EXPLORING ALGEBRA AS A LANGUAGE-IN-USE:

A STUDY WITH 11-12 YEAR OLDS USING GRAPHIC CALCULATORS.

TENOCH ESAÚ CEDILLO ÁVALOS

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENT

FOR THE Ph.D. DEGREE OF THE UNIVERSITY OF LONDON

UNIVERSITY OF LONDON INSTITUTE OF EDUCATION

DECEMBER, 1996.

ABSTRACT

The thesis presents a research that focuses on how children's learning processes occur when algebra is introduced as a language-in-use. The research incorporates graphic calculators as a means for providing children with a computing environment where communication is held by using a symbolic language similar in syntax and notation to the algebraic code. The use of calculators is shaped by a set of tasks specifically designed for this study. The tasks are arranged in order to simulate the social processes through which children learn the mother tongue.

The design of the learning environment is based on Bruner's research on children's language acquisition. According to this, the major aim of the study is to investigate the ways in which the calculator's symbolic code shapes children's expressions of general relationships, and more specifically the kinds of notions and strategies that children develop through using calculator language. The study seeks for an explanatory framework that might provide a better understanding of the potential of technological resources in the teaching of algebra.

The study drew promising results that provide evidence for an alternative approach to teaching algebra. The thesis offers a discussion of the theoretical background and its relationship with the teaching method. It also provides an analysis of children's achievements and difficulties.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to thank my supervisors, Prof. Rosamund Sutherland and Prof. Richard Noss, for their generous advice and guidance throughout the preparation of this thesis during my stages at the Institute of Education as well as during their visits to the Department of Mathematics Education, CINVESTAV, in Mexico. I want to thank Dr. Teresa Rojano, who had always time and patience to read my work and gave valuable advice. I am specially grateful to Dr. Eugenio Filloy, I would have not made this research without his decisive support.

I am indebted to the group of pupils from the school Centro Escolar Hermanos Revueltas in Mexico City and in particular to the case study pupils who were always ready to collaborate in the study.

Thanks are also due to my colleague Marcela Santillán for her continual enthusiasm and encouragement, and for establishing a working environment in which the ideas of the thesis could be fostered. I am grateful to Jan Winter and Susie Molyneux from the University of Bristol, who very kindly read the thesis and provided valuable assistance in the preparation of the manuscript.

I am indebted to the Subsecretaría de Educación Superior for their financial assistance during the preparation of this thesis. I thank the Universidad Pedagógica Nacional for allowing me to undertake this research in the course of my employment as a lecturer in the Department of Mathematics and Statistics.

Finally, I could not have made this research without the support of my family, who were always supportive and often they had to take care of themselves and help each other when I disappeared into my work.

CHAPTER 1
INTRODUCTION AND BACKGROUND

Introduction

This thesis is developed around a specific approach to introducing the study of algebra using graphic calculators. The research addresses the notions and strategies that 11-12 year old children may develop whilst they work on a set of specially designed activities which shape the classroom setting so that the calculator code is met by children as a language-in-use.

The research carried out until the early 80's on the transition from arithmetic to algebra has shed light on the teaching and learning of mathematics, and the recent advent of computerised resources within the mathematics classroom has opened promising new alternatives. Of particular interest to this thesis are those studies which have focused on: pupils' interpretations of the letters used in the algebraic code, the implications of children's previous arithmetic experience to their approach to algebra, pupil's approaches to algebra problem solving, and the potential of new technological resources in the teaching and learning of algebra.

Some of the research carried out prior to the incorporation of computerised resources into the mathematics classroom (Collis, 1969-1975; Küchemann, 1978-1981; Booth, 1984; Clement et al, 1979-1982; among others) has suggested that some algebraic concepts, such as the notions of variable and function, require from the children a level of maturity in their intellectual development, that they have not apparently reached at the age when the study of algebra traditionally starts (12-13 years old).

In some countries (for example, USA, and UK) these results have led to delaying the study of algebra or to the decision that algebra should not be taught to those children who have not showed an acceptable performance in mathematics. However, the current availability of microcomputers and graphic calculators has encouraged the development of educational research which has provided empirical evidence that challenges the current structure of the school algebra curriculum and suggests new promising approaches to the teaching and learning of this subject.

The present study attempts to take advantage of the vast experience developed by the research done prior to and after the incorporation of computing tools, and aims to explore the potential of the graphic calculator as a tool which may provide support for introducing algebra as a language-in-use, in a way similar to that in which one learns the mother tongue. As will be further discussed in Chapters 3 and 4, the approach to algebra as a language-in-use was influenced by the theoretical and empirical work by Bruner on language acquisition, some research results from Bruner's investigation were recast to shape the classroom tasks used in this study and to analyse the ways in which children confronted them.

The chapter is organised in three sections: first those ideas that constitute the background for this study are discussed, then the aims of this researched are presented; finally, the structure of the thesis is described.

1.1. Background

This section presents some antecedents that constitute the raw material on which this research was developed, and succinctly discusses those central issues which provide an overview for the study. The initial ideas that finally led to the present study have their origin in some conjectures developed by the author of this thesis. These conjectures rely on empirical findings obtained during a long experience of teaching.

One of these conjectures consists of the idea that children develop a better grasp, whether of a notion or a mathematical strategy, once they have previously met and used it as a tool while dealing with a problem situation. As children use these notions and strategies in different and more complex situations they can gradually reformulate their explanatory framework.

What seems to be a major difficulty in such a teaching position is how to match the students' present level of knowledge so that they are likely to become engaged enough with the mathematical activity. This position seems to find support in Vygotsky's concept of zone of proximal development.

Another conjecture comes from an informal exploratory study carried out in 1990, in which programmable calculators were used as a means of introducing 11-12 years old children to use the algebraic symbolism. During the study it was observed that a special kind of interaction took place between children and the calculator. It seemed that children were motivated by a sort of curiosity to know more about the facilities offered by the machine. Children's work suggested that the requirement of learning the intricacies of the calculator's programming code did not seem to be a serious obstacle. The way in which the children carried out the tasks suggested that they came into an environment in which, in order to inquire what else they could do with these machines, the children learned to use the formal code that the calculator 'understands'. This experience suggested that most of these children were learning how to use the algebraic symbolism used by the calculator in something like a 'communicative setting'. It seemed that the need to establish communication with the calculator was what led them to cope with the algebraic symbolism as a language-in-use.

The idea of conceiving algebra as a language has been addressed by various authors (see, for example, Papert, 1980; Mason et al 1985, Pimm, 1987; Sutherland & Rojano, 1992; Bell, 1992; Shoenfeld, 1993; Tall, 1993). The positions put forward by these authors suggest that the metaphor of algebra as a language may be a fruitful vein for research on algebra. The present study is intended to explore a particular view of this metaphor so as to investigate the potential of the graphic calculator within a highly framed learning environment specially designed for pupils to meet algebraic code as a language-in-use. Since these issues form the building block on which this thesis develops some of its central ideas are briefly discussed below and are further elaborated in the rest of the thesis.

Particularly relevant to this research is the work done by Papert (1980), Sutherland (1987), Hoyles and Sutherland (1989), and Mason et al. (1985). Papert's idea of a microworld influenced this study in the sense of creating a computer-based environment which encourages children to use a formal language to express and communicate mathematical ideas. The research carried out by Hoyles and Sutherland on Logo-based environments mainly influenced this study with regard to the role of the teacher; their work provided background to this study in the sense of focusing attention on how children learn through interacting within computer environments as a necessary step in designing a computer-based classroom setting. Hoyles and Sutherland found it necessary to modify Papert's original idea of encouraging pupils to create their own 'project goal' which was based on the principle of that "children learn to speak, learn the intuitive geometry needed to get around in space, and learn enough of logic and rhetoric to get around parents -all this without being taught" (Papert 1980, p. 7). Sutherland's work showed that teaching intervention is more important than Papert's original work suggested, for structuring and guiding children's learning within the Logo environment. In this respect, this study took up Sutherland's findings and adopted Bruner's concept of *format*, which was recast so as to serve as a building block from which to delineate the teacher's intervention within the calculator-based setting designed for this research (this point is discussed in more detail in chapters 3 and 4).

Mason's work influenced this study with regard to the approach used in the design of the tasks, which is based on the idea of 'expressing generality'. Mason's work on this topic as one of the routes to algebra was recast so as to incorporate the use of the graphic calculator. The use of the calculator was guided by the principle of expressing generality by means of the calculator's code, this principle was exploited to create a mathematical environment intended to allow pupils to meet the algebraic code as a language which helps them explore the behaviour of number patterns and to produce algebraic expressions which encapsulate and govern these type of general relationships. The tight link between general number relationships and the production of algebraic expressions mediated by the calculator is meant to constitute a shared context between pupil's previous arithmetic experience and the new algebraic code.

1.2. Aims of the Study

The major aim of this thesis is to investigate the ways in which the calculator's symbolic code shapes children's expressions of general relationships. More specifically, the main purposes of this study are to investigate:

1. The notions that pupils may develop for algebraic language when they meet it through using calculator code.
2. The extent to which the use of the calculator language helps pupils cope with simplifying similar terms within linear expressions, inverting linear functions, and transforming a linear algebraic expression to obtain a target expression.
3. The strategies that children may develop through working with the calculator.
4. The extent to which the use of the calculator language as a means of expressing general rules governing number patterns, helps children grasp that the algebraic code can be used as a tool for coping with problem situations.

1.3. Structure of the Thesis

The remainder of the thesis is organised in eight chapters which are now briefly described. Chapter 2 discusses the relevant algebra research literature which documents those research findings which more closely relate to the major issues addressed in this thesis, in particular, those topics that take place in the transition from arithmetic to algebra, and technology-based studies which deal with introductory algebra.

Chapter 3 addresses the theoretical perspective adopted in this study. The chapter presents an overview of Bruner's theoretical perspective on language acquisition and concludes with an account of how Bruner's theory was recast so as to be applied to the case of the teaching and learning of early algebra using graphic calculators.

Chapter 4 deals with methodological issues that guided the implementation of this research, and provides a detailed account of how the theoretical perspective described in Chapter 3 was made concrete in a set of articulated tasks. These were intended to structure pupils' work throughout a lengthy learning process which assumed little knowledge about algebra and aimed at using the calculator language to negotiate problem solutions.

Chapter 5 presents a vertical analysis of two case-study files which offers a detailed account of the work done by the children during the fieldwork. Since the thesis approaches the teaching and learning of algebra as a language-in-use, the ways in which pupils' command of calculator language evolves in time offers a relevant research perspective to this study. This chapter provides a chronological analysis of how children's use of the calculator language evolves, and how this use frames children's algebraic notions and strategies throughout the study. The chapter addresses some issues intended to start shaping an explanatory framework for analysing children's achievements.

Chapter 6 presents a brief account of the work done by the other five children who were closely followed throughout the study. The major aim of this chapter is to complete the view provided by Chapter 5 and to provide background for the horizontal analysis carried out in Chapter 7.

Chapter 7 offers a wider view of the results presented in the preceding chapters. This chapter provides a horizontal analysis of children's work which focuses on the main difficulties, insights, notions and strategies presented by the seven children who were closely observed throughout the fieldwork.

Chapter 8 presents the conclusions of this study which are formulated around the main contributions of this study to the research of algebra, its limitations, and the implications of the results drawn from the thesis to further research.

CHAPTER 2
REVIEW OF THE ALGEBRA RESEARCH LITERATURE

Introduction

This chapter reviews those findings documented in the algebra research literature that have addressed from various perspectives the major issues investigated in this thesis. As was stated in Chapter 1, this study focuses on the transition from arithmetic to algebra, and of special interest to this thesis is the research that has studied this facet of mathematics education before and after the availability of computers and graphic calculators in the classroom. Also relevant to this thesis are those studies and positions that have addressed the metaphor of algebra as a language and its implications to the teaching and learning of introductory algebra.

The chapter is organised in six sections. Section 1 discusses some of the research that have focused on children's difficulties in generating correct notions for algebraic expressions, and some implications to children's approaches to solve word problems. Section 2 deals with some research findings obtained from studies which, taking advantage of preceding studies on children's difficulties with school algebra, has included in their research method a teaching phase aimed at alleviating the problems children have. Section 3 discusses some studies that have addressed the issue of providing a theoretical background to the teaching and learning of algebra. Section 4 discusses research findings obtained from technology-based studies that have addressed issues concerning the transition from arithmetic to algebra. In particular, technology-based studies that have focused on children's notions of algebraic expressions, and how certain technology-based approaches (algorithmics in school mathematics, Logo, Spreadsheets, Basic, and graphic calculators) may help pupils make sense of the algebraic code and use it in negotiating solutions for algebra word problems. Section 5 discusses various positions around the metaphor of algebra as a language. Finally, Section 6 presents the conclusions derived from the review made in the preceding sections, and discusses some possible implications to the approach of algebra as a language-in-use adopted in the present study.

2.1. Children's notions of algebraic expressions.

A number of research studies have shown that the interpretation of algebraic expressions, particularly of the letters used in the algebraic code, is not an easy matter for many children (Collis, 1969-1975; Küchemann, 1978-1981; Booth, 1981; Booth 1984; Clement et al, 1979-1982; among others). From a range of different perspectives and emphases these studies have found that the majority of 15-year-olds appear to be unable to interpret algebraic letters as generalised numbers or even as specific unknowns. Large scale studies (Küchemann, 1978-1981) have documented that a great proportion of students ignore the letters, replace them by numerical values, or regard them as shorthand of names or measurement labels. A considerable research effort has been made to investigate the underlying reasons in children's difficulties to cope with the new use of letters introduced in the study of algebra. Perhaps the most immediate explanatory frame is provided by children's arithmetic experience in elementary school. For instance, whilst in arithmetic children have experienced that letters can denote measurements, for example 10m to denote 10 metres, in algebra such expression may denote 'ten times an unspecified number'. Traditionally, children's experience with letters in elementary school is restricted to equations such as $A=l\times w$, which seems to reinforce the arithmetic use of letters as labels (l for length and w for width), this interpretation of letters as measurement labels seems to explain the students' tendency to treat numerical variables as if they stood for objects rather than numbers.

The ambiguity of the use of letters in algebra also represents a difficult challenge for children to fulfill. In algebra the same letter can be used to represent different numbers within different situations; the same number can be represented by different letters in the same situation; a letter can also represent a whole class of numbers; and, what seems the most difficult part in all this; these letters represent unknown or unspecified numbers. In what follows, particular research approaches to these issues are discussed.

Collis (1969, 1971, 1973, 1975a, 1975b), hypothesised that the difficulties children have in algebra relate to the abstract nature of the elements involved. He elaborated on

Piagetian findings on concrete and formal thinking to explain children's achievement in algebra. This framework allowed Collis to differentiate between concrete and formal operational thinkers on the basis of a child's degree of reliance on 'reality'. According to Collis, a concrete operational thinker is restricted to concrete-empirical experience, a concrete operational thinker considers only what is empirically verifiable; that is, concrete and formal thinkers differ in their ability to handle abstract elements and operations (Collis, 1969, 1971). This view proposes the existence of a progression in the children's ability to cope with small numbers of immediate experience, then larger numbers which lie outside the immediately verifiable range, and algebraic elements, which would in turn be viewed initially as representing specific individual values, and only later as generalised number, and finally 'variables' (Collis, 1975c). Perhaps the main conclusion made from Collis' analysis was that development of understanding in algebra may correspond to a progression in the ways in which letters are interpreted.

The work done by Collis influenced the research carried out during the early 80's on children's conceptions of algebraic symbolism. A study by Küchemann (1981), as part of the Concepts in Secondary Mathematics and Science (CSMS) project, investigated the performance of school students aged 11-16 on test items concerning the use of algebraic letters in generalised arithmetic. This study showed that most students in the large sample were unable to cope with items that required interpreting letters as generalised numbers or specific unknowns. Küchemann (1978, 1980, 1981) used the framework developed by Collis to analyse and describe students responses, and found that most of the students' errors were likely to be produced by their interpretations of letters. Küchemann (1981) studied different students' conceptions of letters in algebra with a sample of 3000 British students (13-15 year olds). The study produced results that showed that 73% of the 13 year olds, 59% of the 14 year olds, and 53% of the 15 year olds, dealt with letters in expressions and equations as objects; few were able to consider letters as specific unknowns, and fewer still as generalised numbers or variables. The study showed that students' misunderstanding of letters seem to be reflected in their approach to symbolising the relevant relationships in problem solutions. In this respect Küchemann reported that all students were asked the question: Blue pencils cost 5 pence each and red pencils cost 6 pence each. I buy some blue and some red pencils and altogether it costs me 90 pence. If *b* is the number of blue pencils bought, and *r* is the number of red pencils bought, what can you write down about *b* and *r*? The percentages of correct responses within each age group were 2%, 11% , and 13% respectively. The most common response was $b+r=90$, this mistake suggests a strong students' tendency to conceive letters as labels denoting specific sets, which seems to be a result of the students' attempt to accommodate their previous arithmetic experience with letters to the new meanings assigned to letters within an algebraic context.

Similar results were found in the National Assessment of Educational Progress, which was carried out with 70,000 American pupils (9, 13, and 17-year-olds). The students were asked the problem: Carol earned D dollars during the week. She spent C dollars for clothes and F dollars for food. Write an expression using D, C and F that shows the number of dollars she has left (Carpenter et al., 1981). Though in this problem the literal symbols represent specific quantities that do not need to be related by an equation (as is required in the problem used by Küchemann), a third of the 17 year old students (who had had one year of algebra) and over a quarter of those who had two years of algebra did not provide an acceptable answer. These results shed light on the problem of translating from one symbol system (natural language) to another (algebraic code) which shows the difficulty that novice algebra students have when using a new symbol system when, according to Küchemann's results, they are not yet familiar with its semantic structure.

Other studies have shown that symbolising the relationships in a problem situation is not only difficult for novice students (Clement, 1982; Clement, Lockhead, & Monk, 1981; Clement, Lockhead, & Soloway, 1979). Clement et al investigated the responses of 150 freshman engineering students to the Students-Professors problem: Write an equation using the variables *S* and *P* to represent the following statement: There are six times as many students as professors at this university. Use *S* for the number of students and *P* for the number of professors. The results showed that only 63% of these students could correctly solve the problem, where 68% of the errors consisted of the reversal situation ($6S = P$ instead of $6P = S$).

Among others, these findings produced a striking impact on the ways in which the teaching and learning of algebra have been traditionally conceived, and suggested the development of new research lines oriented by the idea that investigating the meanings that children attach to literal symbols may provide results that lead us to propose teaching approaches aimed at improving the learning of algebra. This research style is encouraged by the idea that though the errors already identified (and others) may be familiar to any algebra teacher, the causes of such errors should be identified, otherwise it is difficult to improve the teaching of algebra so as to enhance student's understanding. The development of further research in this direction is discussed in the next two sections.

2.2. Research studies incorporating a teaching module

Wagner (1979, 1981a) suggested that encouraging students to solve problems involving expressions and equations that are purposely similar in form may help in identifying particular learning difficulties associated with these forms. She proposed an analytical framework for investigating student's understanding of variables aimed at generating tasks that provide measures of the clarity and difficulty of the different roles of letters.

The framework is based on the assumption that, similarly to words of verbal language, the symbols for mathematical variables acquire meaning only as they appear in some *context* and represent some *referent.* Wagner proposed that, as in verbal language, the symbol and its referent determine the syntactic role of the variable, and argued that all three components: symbol, referent and context, as well as all the three aspects (the semantic role, the syntactic role and the mathematical role) combine to contribute to the student's interpretation of variables. A change in any of these components may or may not, depending on the nature of change, cause a corresponding change in each related aspect of the variable. Her study was designed to investigate whether the students (12-17 year-olds) perceived the equivalence between pairs of equations which were presented using different letters as unknowns. The results showed a different confusion over the use of letters: some children did not appear to realise that the value of an unknown is independent of the letter used. These children seemed to believe that changing the letter implied that the value was also changed (Wagner, 1981, 1981b). There were some children whose work showed that they established a sort of correspondence between the order in which letters appear in the alphabet and the number system: letters towards the end of the alphabet were assigned a higher value than those nearer the beginning.

Another research project that included a teaching module was carried out within the SESM project (Strategies and Errors in Secondary Mathematics Project, Booth, 1984a). The SESM was a sequel to the 1974-79 programme Concepts in Secondary Mathematics and Science (CSMS), and consisted of a longitudinal study of students in seven classes. Its aim was to investigate in depth some of the problems commonly experienced by secondary school pupils in the area of mathematics and examined to what extent these problems could be alleviated by specifically designed teaching modules. The SESM addressed the errors in generalised arithmetic identified by the earlier CSMS project and was based on the framework provided by the following hypotheses: (1) The errors observed depend (in part) on the child's interpretation of the letters involved, and (2): An error may also arise as a consequence of the procedures that the child uses in solving arithmetical problems of a similar kind.

A general result obtained from this study is that the teaching module designed to overcome particular difficulties in early algebra was more successful with high ability students than with students in middle and lower streams. Booth suggested that this finding provides further evidence for the link between students' cognitive levels and their ability to understand the meanings of algebraic letters and to formalise mathematical methods.

One of the specific issues addressed in the SESM was the children's approaches to formalising their methods. She suggested that the construction of formalised procedures constitutes a relevant part of mathematical activity. A teaching strategy was implemented based on the idea of a 'mathematics machine', which involves the use of a four-operations calculator and a pencil and paper 'mathematics machine' which extended the calculator facilities to include operations with letters. The rationale for the formalisation was provided by a notational machine, complete with input pad, start button, processor, store locations and output pad. She hypothesised that the advantages of such an approach are based on the need for explicit and rigorous representation of procedures; the use of the 'mathematics machine' was intended to provide a rationale for the use of letters as a means for controlling the machine. The results obtained from this teaching experiment showed that children improved their skills in formalising their methods and developed some insights about the role of algebraic symbolism. Booth reported that most of the difficulties children have in formalising their methods are likely to be due to children's tendency to use informal procedures which have been proved to be successful in facing arithmetic tasks but failed in the algebraic case. These results led Booth to suggest that teachers should be aware of the informal methods used by children and to encourage their use until these methods prove not to be efficient. Booth suggests that this strategy may help children accept the formal methods that the teacher is willing to teach. This is particularly so with regard to the understanding of letters as generalised numbers, where Booth reports that children presented a strong resistance to assimilating the idea of letter as generalised number even in the context of the teaching programme specially designed to help pupils overcome this difficulty. In this respect, the use of technology offers an interesting alternative (as will be discussed in the following section). There are various studies that have shown that computer or calculator-based environments may be arranged to encourage children to explore mathematical situations using their own strategies (see, for example, Ruthven 1990; Sutherland & Rojano, 1993). Technology-based studies have reported that the formality of the code of the technological resource helps children to structure a more formal approach on the basis of their initially informal methods. This point is taken up later in this chapter.

Another specific issue addressed by the SESM was that of children's interpretation of letters. In this respect they reported that "whilst some of the misconceptions that children have may be due to inadequacies in the teaching-learning situation, some of the difficulty which children have appears to be related more to a 'cognitive readiness' factor. This is particularly so with regard to the apprehension of letters as representing generalised number rather than specific unknowns" (p. 87). The results obtained from using the 'mathematics machine' and a simple calculator suggest that the introduction of letters as generalised numbers seems to be a promising starting point for children to achieve the different meanings assigned to letters in algebra.

Booth's findings strongly suggest that many of the difficulties that children have in algebra are not difficulties in algebra as such, but rather difficulties in arithmetic. Booth argues that "if algebra (or generalised arithmetic) is regarded as the writing of general statements representing given arithmetic rules, operations and procedures, the non use of and non-recognition of those structures in arithmetic would be likely to have a considerable effect on children's performance in algebra" (p. 89). For example, children's reluctance to accept unclosed algebraic expressions may be explained in terms of their lack of experience with such type of expressions in arithmetic; conceiving of a+b as an object in algebra should have some numerical precursor in arithmetic. The 'acceptance of lack of closure' is reported by Booth (1984a) in items where students were asked to mark all equivalent expressions among several options. She observed that, some children thought that x+y was as equivalent to xy and found the same type of answers when children were asked to find the area of rectangles (for example, 7f3 or f21 for the area of a rectangle with length 7 and height f+3). This point is taken up again in Chapter 6.

The link between the structure of arithmetic procedures with the structure of algebraic expressions was also investigated by Chaiklin and Lesgold (1984). They investigated the children's reactions when working with three-termed arithmetic expressions involving a subtraction and an addition (for example, 685-492+947, 947+492-685, 947-685+492, 947-492+685). The inquiry consisted of asking sixth graders to find equivalencies without computing the totals. Chaiklin and Lesgold found that, despite the explicit instruction of not to calculate the totals, the prevalent method used by the pupils was to calculate the sums in order to decide the equivalence of expressions. Using similar tasks, Collis reported that younger children were able to succeed only when the number of items involved were easy for them to calculate. Collis described the ability to work with the expressions without reducing them by calculating as 'acceptance of lack of closure'.

The 'acceptance of lack of closure' was investigated within a geometrical context by Chalouh and Herscovics (1988). They used geometrical diagrams to help students make sense of and use algebraic expressions. Rectangular arrangements were included as part of an instructional phase with sixth and seventh-grade students. The teaching sequence allowed students to construct algebraic expressions for referring to the area of rectangles like the items used by Booth. The results showed that this approach supported children in using algebraic expressions to denote the area and perimeter of rectangular shapes. Nevertheless, children's responses still showed a sort of reluctance to accept unclosed algebraic expressions as real answers. The children considered that expressions like 4x+4y were some how incomplete, and so they completed the expressions expressing them as part of an equality (i.e., Area=4x+4y for a rectangle of height 4 and length x+y).

2.3. Theoretical Approaches to the Teaching and Learning of Algebra.

Many studies have been carried out with the aim of providing theoretical background to the teaching and learning of algebra. Of particular interest for this thesis is the work done by Herscovics (1989) on the concept of cognitive obstacle, the research carried out by Sfard (1991, 1992,1994) on the historical-epistemological development of algebra, and the work done by Mason et al (1980, 1984, 1985, 1988, 1989, 1991, 1993) on expressing generality as a particular route of algebraic thinking. These approaches are succinctly reviewed below focusing on those features that more closely relate with the approach to school algebra adopted in this thesis.

The concept of cognitive obstacle

Herscovics (1989) analysed the concept of cognitive obstacle as a way of providing a different explanation for the difficulties children have in facing the study of algebra. He pointed out that the arguments offered to explain the extensive rate of failure in high school algebra may be summarised in two points: (i) students' failure as a result of inadequate teaching, which suggests that improved teacher training programmes may solve the problem. This position suggests that all the problems involved in the learning of algebra can be solved, and, consequently, that the main problem relies on a failure in communication; and (ii) another type of explanation is given by the argument that mathematics, and consequently algebra, was never intended for the general population. This argument suggests that the teaching and learning of mathematics should not be a question of concern, since there have always been people who have learnt mathematics regardless of the way it is taught.

The concept of cognitive obstacle provides an argument which is rooted in historical, epistemological and psychological points of view, which in Herscovics' view "denigrates neither the teacher's professionalism nor the students intellectual potential" (p. 60). The concept of obstacle was approached first within the context of science where it was referred to as an epistemological obstacle. Later on, Brousseau (1986) re-cast the concept of epistemological obstacle in order to applied it to mathematics education. He characterised a cognitive obstacle as "a piece of knowledge that has in general been satisfactory for a time for solving certain problems, and so becomes anchored in the student's mind, but subsequently that knowledge proves to be inadequate and difficult to adapt when the student is faced with new problems" (quoted by Tall, 1991).

The concept of cognitive obstacle is illustrated by the difficulties that many children have at the beginning of their initial algebra course. For example, pupils' notions and strategies which proved to be successful in arithmetic but later lead them to misconceptions in the construction of meanings for algebraic expressions, such as the interpretation of letters as denoting measurements (Booth, 1984). Children's apparent reluctance to accept unclosed expressions (Collis, 1974) provides another example of a cognitive obstacle. Herscovics (1989) used the framework of cognitive obstacle for examining three distinct situations: obstacles induced by instruction, obstacles of an epistemological nature, and obstacles associated with the learner's process of accommodation (in Piaget's sense). He puts forward that the obstacles induced by instruction are often due to a formalistic presentation of mathematics, in this case the problem is caused because the student cannot relate his/her existing knowledge with the new notions being introduced. His findings suggests that this type of obstacle remains with the student until he/she is provided with additional means to bridge the gap between the new material and his/her existing knowledge. The main difficulty here is that this gap is not always evident.

An epistemological obstacle is associated with those violent shifts occurring in the historical development of a discipline that had to be overcome for any growth, such as the concepts of limit and continuity of functions in the context of differential calculus. Those obstacles associated with the learner's process of accommodation seem to be the most challenging in pedagogical terms. Herscovics suggests that "no matter how much goodwill and care the teacher provides, these structural changes cannot be conveyed by mere transmission of information. Each learner is condemned to alter the mental structure in his or her own mind" (p. 83).

A historical-epistemological approach to algebra

Sfard developed a theoretical perspective from which to analyse the role of mathematical concepts in mathematical thinking that finally suggests a teaching approach to school algebra. She grounds her study in an ontological-psychological approach and analyses different mathematical definitions and representations which led her to conclude that abstract notions, such as number or function, can be conceived of in two fundamentally different, but complementary, ways: structurally as objects and operationally as processes. From this viewpoint, the algebraic representation of a function may be interpreted both ways: it may be explained operationally, as a concise description of some computation, or structurally, as a static relation between two magnitudes. The dual meaning of equality sign (Behr et al., 1976; Kaput, 1979; Kieran, 1981) also illustrates the procedural/structural framework, the equality sign can be used as a symbol of identity, or as a 'command' for executing a series of arithmetic operations.

Sfard's historical analysis proposes that for many people the operational conception is the first step in the acquisition of new mathematical notions. This framework relies on the assumption that the transition from computational operations to abstract objects (concept formation) takes place through a process accomplished in three steps: *interiorisation*, *condensation* and *reification*. The latter is considered to be a complex phenomenon, which seems inherently so difficult that it may remain practically out of reach for certain students. These three steps constitute a schema that suggests a hierarchy, where one stage cannot be reached before all the former steps are taken. Each stage in the process of concept formation may be summarised as follows: Interiorisation is characterised as a process performed on already familiar objects. Condensation has to do with the stage in which this process turns into an autonomous entity. Reification relates to the ability to see this new entity as an integrated object that has been acquired.

Sfard establishes the framework of reification on the basis of historical evolution of numerical systems from natural to complex numbers and the history of the concept of function, this framework proposes that various processes had to be converted into compact static wholes to become the basic units of a new higher level theory. Accordingly, such a hierarchy emerges in a long sequence of reifications.

Sfard suggests that the theory of reification can be applied to develop a curricular approach to algebra. Her proposal relies on the argument that the structural approach is more abstract than the operational, which implies that a person could hardly arrive at a structural conception without previous operational understanding. On this view the historical analysis of algebra suggests that algebraic thinking has its origin in operational processes and evolves in a sequence that goes on towards gaining generality and structure: from rhetorical to syncopated algebra, from this to Vietean symbolic algebra, and finally abstract algebra.

Sfard (1994) carried out an empirical investigation which provided further evidence for the procedural-structural view. The study gave "not very encouraging conclusions ... pupils cannot cope with problems which do not yield to the standard algorithms ... it became clear that the functional approach is not easily accessible even for the better students" (p. 223). The unsatisfactory results obtained from this study were explained as a consequence of the Israeli curriculum, where the study of algebra begins by introducing the use of letters as variables: This, according to Sfard's model, disrupts the historical-epistemological order in which the use of letters evolved (first as unknowns and much later as variables). However, technology-based research has shown that the multiple representation resources provided by computer environments encourage the introduction of algebra from the notion of function (Kaput, 1989; Dreyfus and Halevy, 1988; Leitzel, 1989; Kieran, 1991; Confrey, 1993). In this respect Sfard suggests that "it is fairly possible that the massive use of computer graphics in teaching functions will reverse the 'natural' order of learning (suggested by the historical analysis) so that the structural approach to algebra will become accessible even to young children" (p. 224).

To summarise, Sfard's theory of reification leads to a hierarchical model which, she proposes, may be used to explain most of the difficulties students have in learning algebra. The model may also be used to design teaching approaches that may help students reach the stage of reifying the central concepts in algebra. In Particular, Sfard's framework points out that disrupting the natural way in which algebra evolved through history may lead to a pseudo-structural conception of algebra, which may be the case of introducing algebra through the notion of function.

Expressing generality

The theme of generality has been addressed by various authors (Piaget, 1988, 1985; Krutetskii, 1976; Polya, 1965), and more recently by Mason et al. (1980, 1984, 1985, 1988, 1989, 1991, 1993) in the United Kingdom. Among the various perspectives on generality provided by these authors, the work done by Mason is of special interest to the present thesis, particularly due to Mason's orientation of integrating expressing generality into the everyday life of the mathematics classroom.

Mason (1993) conjectures that "when awareness of generality permeates the classroom, algebra will cease to be a watershed for most people ... unless and until, expressing generality becomes natural and spontaneous in the conduct of mathematics" (p. 1). Though Mason acknowledges that there does not seem to be a single programme for learning algebra through the expression of generality, his work seeks for awakening and sharpening sensitivity to the potential of generality for algebraic thinking. Mason (1988) offers a wide range of examples of tasks which have been exploited in this way (some of them were recast and used in the present research, see Appendix).

According to Mason, activities intended to develop awareness of generality should encourage the abstraction and concretisation of experiences that promote seeing a generality through the particular, and seeing the particular in the general. Seeing the particular in the general implies recognising the particular as an instance, that is, seeing the particular in the general involves maintaining attention on the general as well as on the particular. Mason (1989) proposes a framework for analysing the process of moving from the particular to the general and vice versa. The framework may be summarised as a four steps schema which evolves through a spiral which represents the development from confident manipulation, to getting-sense-of, to articulating that sense, to that articulation becoming itself in turn confidently manipulable.

Perhaps the strongest argument to place generality at the core of the teaching and learning of algebra is that "the facility in manipulation of generality follows as confidence in expression develops and as multiple expressions for the same thing arise, and that use of algebra to solve problems depends on confident expression of generality using the as yet-unknown supported by awareness of the role of constraints on variables" (Mason, 1993, p. 2).

A point which is emphasised in Mason's work is that generalisation seems to have such a tacit role in algebra that the experts no longer notice its presence in what is, for them, elementary, but it is precisely the shifts of attention that they have integrated into their thinking which are problematic to the novice. This view brings a cultural dimension to the approach of generality in the classroom, where generality is much more than a topic to be taught (if this were possible). It is rather a manner of thinking and acting which should shape the activity in the classroom, a setting in which teachers feel comfortable in acting mathematically with and in front of their pupils so as to help their pupils confront mathematics as a way of thinking and expressing, "just as naturally as they are into listening to and speaking their native tongue" (p. 3).

The framework of manipulating, getting-a-sense-of and articulating was obtained by re-casting Bruner's (1966) notion of enactive, iconic and symbolic representations. Though the proof of such framework is in future experience, Mason (1993) has applied it in analysing and describing experiences with teachers, and reports that these experiences can develop into potent sensitisers to future classroom opportunities. Particularly, he has found the framework informative when thinking of these experiences as phases in a developing spiral, in which manipulation, whether of physical, mental or symbolic objects, seemed to provide the basis for getting a sense of patterns, relationships and generalities, and that as you become articulate, your relationship with the ideas changes. A shift takes place in the form and structure of your attention. What was previously abstract becomes increasingly confidently manipulable. Mason's spiral model attempts "to connect similar yet different states, while suggesting that manipulation changes as pattern is sensed, that attempts at articulation may cause re-thinking and re-manipulating, but that through a fluid almost symbiotic process, increasing facility and confidence develop" (p. 15).

Expressing generality has been placed as one of the roots of algebra as well as one of the routes to learning algebra (Mason et al., 1985). The impact of this approach has been reflected in that during the last few years mathematics educators have come to regard working with generality as one of the characteristics of algebraic thinking. In a good deal of countries school algebra is nowadays a manifestation of generalisation about numbers. The study and description of patterns and general rules has been included within the algebra curriculum in some countries. In Australia, for example, the National Statement on Mathematics for Australian Schools (1991) and the Curriculum and evaluation Standards (1989) recommended that algebra learning should be introduced by investigating geometric patterns, number sequences and function tables (MacGregor and Stacey, 1993).

Perhaps number patterns have been the most common resource used to introduce young children to expressing generality. For example, the National Curriculum in the UK has integrated number pattern with the development of algebraic symbolism (Tall, 1993). The use of a number pattern approach is being generally applied within a sequence consisting of asking children to find the underlying pattern, expressing it in words, and then use algebraic notation to describe the pattern (or a shorthand algebraic notation).

Though the introduction to algebra through number patterns-based approaches seems to be successful for the more able pupils a number of research studies have found that this approach presents difficulties for the majority of children (Stacey, 1989; Herscovics, 1989; Arzarello, 1991; MacGregor and Stacey, 1992; MacGregor and Stacey, 1993; Stacey and MacGregor, 1996). These researchers have addressed the effects of introducing school algebra through the study of patterns and rules relating two variables and have reported that students have difficulties in generating algebraic rules from patterns and tables. The results indicate that the ability to perceive a rule does not imply that the students can easily express it algebraically.

Herscovics (1989) in reviewing the results of the national testing in the USA comments that although most students could recognise a pattern presented in a table of ordered pairs connected by simple rules (for example, add 4), the majority were unable to express these rules algebraically (for example, $y=x+4$). Arzarello (1991) also found that generating algebraic rules from number patterns was a difficult task. When students had generated a table of ordered pairs from a geometric pattern, they focus on the difference between successive values of each variable. The students looked for a recurrence rule that would help them predict a number from the value of its predecessor rather than a functional relationship linking pairs of numbers.

MacGregor and Stacey (1993) carried out a questionnaire-based research with a sample of 143 students who were in their third year of learning algebra. A further 15 students were interviewed individually. The results show that more students were able to find and use a relationship for calculating than could describe it in words or algebraically.

They found that the major causes of difficulty were: (i) focusing on recurrence patterns in one variable rather than on relationships linking two variables; (ii) inability to articulate clearly the structure of a pattern or relationship using ordinary language. The items they used in the questionnaire were designed to find out what aspects of the task of recognising patterns and describing them algebraically present most difficulty. Their results show that "whereas most students can perceive patterns in tables easily, many do not perceive the functional relationship" (p. 181). Interview protocols showed that a substantial proportion of those students who 'see' the relationship clearly (so as to calculate with it), could not express it neither algebraically or in natural language.

MacGregor and Stacey (1996) in reviewing how students go about expressing general relationships between variables, concluded that "a patterns-based approach does not automatically lead to better understanding; the way students are taught and the practice exercises that they do may promote the learning of a routine procedure without understanding" (p. 3). Among the student's difficulties they found is that most of the students guided their procedures by natural language descriptions which could hardly help them structure an algebraic expression to properly describe the relationships between two variables.

2.4. Computers and calculators in the classroom.

The incorporation of computerised resources in the mathematics classroom have shed light on new promising avenues for research in mathematics education. Studies based on using computers and calculators have shown that these new tools lead to different approaches in the teaching and learning of mathematics, in diSessa's (1995) terms 'coming to see differently arises from being able to interact differently'. This section briefly reviews some of the findings from technology-based studies that address issues concerning the transition from arithmetic to algebra. The section is organised following a chronological order with regard to the incorporation of technology into the mathematics classroom. First, the role of algorithms in school mathematics is discussed; second, an approach using BASIC is presented; third, results from Logo-based environments are discussed; fourth, some spreadsheets-based studies are presented; finally, the section provides an overview of calculators-based studies.

The role of algorithms in school mathematics

The present research exploits the programming mode of the graphic calculator to encourage children to produce rules governing number patterns, which implies the production of a particular type of algorithm (functional rules). In this sense Johnson's (1991) conception of algorithms as an educational topic is of special interest for this study.

Johnson (1991) discusses the "why", the "what" and the "how" for algorithmics in school mathematics, both in developing mathematical concepts and processes and as a subject of study in its own right. Algorithmics consists of the design and analysis of procedures and "provides a new way to approach and view or express many 'traditional' mathematical concepts and relationships" (p. 331), such as primeness, solving equations, generating values for parameters and variables, and evaluating or graphing functions or relationships. Johnson argues that many mathematical ideas become more dynamic and useful when described as a procedure.

According to Johnson, the potential of algorithms in the teaching and learning of school mathematics relies on the process of producing them, which involves design and analysis. Design may include: (i) implementation of a known procedure to be applied using a programmable calculator or a spreadsheet, for example, to generate the successive terms in an arithmetic or geometric sequence, (ii) modification of a given procedure to carry out a new but related task, such as extending a procedure used to evaluate or generate results to a inquiry for particular values that satisfy given conditions, and (iii) development of a new procedure to help solve a problem, for instance, searching for Pythagorean Triples.

Analysis is "the process of determining how long an algorithm takes to run" (p. 333), which involves the question of efficiency. Analysis implies a high level outcome which is traditionally taught within a Discrete Mathematics course in the upper high-school grades, whilst the design of algorithms seems to be a suitable task for the elementary, middle and lower high-school grades.

While much about algorithms may look like computer science or teaching programming, Johnson's essential idea is that "the contexts for applying (and learning about) these concepts will come from, and within, school mathematics ... The constructs, concepts and relationships in algorithmics should develop naturally along with the development of mathematics" (p. 334). From this view, algorithms may constitute a suitable domain for introducing mathematical concepts; the design of algorithms may be exploited to provide pupils with a new way to express a mathematical idea or relationship, for exploration, investigation, and problem solving.

According to Johnson, the role of algorithms differs depending on their use in a teaching and learning environment or in software engineering applications. In teaching and learning the approach is based on the sequence 'Run-Understand-Debug-Edit'. This sequence is based on the assumption that the user generally experiments with programming to "increase understanding of a problem rather than to produce a polished, commercially viable product" (p. 336), Logo work in schools illustrates this approach. In this sense, the process of constructing an algorithm is a "bottom-up" or from "inside-out". This approach contrasts with the strategy generally used with software engineering, which may be described by the sequence 'Specify-Prove-Implement-Verify'. Within this approach, the procedure is written until the problem has been completely specified. The approach is based on specification and proof, and may be described as a "top-down" approach. The approach is more "conventional" and the focus is on specific examples which are used to illustrate the central ideas and techniques.

With regard to elementary and middle school activity, where a major aim is to help pupils understand key mathematical concepts and relationships, Johnson proposes that algorithms can be better exploited using the 'Run-Understand-Debug-Edit' approach. He enhances the fact of the current increasing availability of programmable graphics calculators which "substantially facilitates individual or small group participation in the *design* activity of algorithmics" (p. 339). The 'Specify-Prove-Implement-Verify' approach may offer crucial support when facing open-ended problems or investigations where the development of an algorithm can aid exploration.

To summarise, Johnson proposes that a rationale ("the why") for algorithms in school mathematics, consists of their potential in developing mathematical concepts and processes. With regard to "the what" for the teaching and learning, the big ideas which underpin algorithmics are assignment, selection, iteration, recursion, and input-output; pupils seem more likely to grasp these big ideas within a teaching and learning environment based on the 'Run-Understand-Debug-Edit' approach. In terms of "the how", he highlights the approach to prime numbers "in terms of both the role of a dynamic (procedural) mathematical description and the extensions, explorations and investigations of mathematical relationships" (p. 342).

Programming with Basic.

Tall and Thomas (1991) questioned that most of the precedent research has been carried out under the assumption that the mathematics curriculum will remain as it is as present, and "one must consider to what degree the conclusions may remain relevant in a new computer paradigm" (p. 88). They investigated some of the cognitive obstacles documented in the research literature within the pre-computer curriculum, and inquired if such obstacles would remain within a computer-based school setting. In order to carry out their research they created a learning environment based on combining the use of a programming language (BASIC), a sequence of physical activities carried out by the students which simulate the storage of variables and manipulation of variables when programming in BASIC ('cardboard maths machine'), and the use of a piece of software specially designed for their study which allows students to evaluate algebraic expressions using standard mathematical notation (Generic Organiser).

This environment was used to investigate the extent to which the students may overcome what has been reported as cognitive obstacles related to the understanding of the concept of variable. A cognitive obstacle is conceived here "as a piece of knowledge that has in general been satisfactory for a time for solving certain problems, and so becomes anchored in the student's mind, but subsequently that knowledge proves to be inadequate and difficult to adapt when the student is faced with new problems". (Tall, 1989, p. 88). Using this characterisation of cognitive obstacle Tall and Thomas investigated the potential of the learning environment they designed in helping novice algebra students overcome the obstacles described below. Their inquiry relies on the hypothesis that the cognitive obstacles previously documented in the research literature were found within the context of the pre-computer curriculum, so "if there are certain fundamental obstacles that occur for us all, they would therefore apply in a new paradigm" (p. 88). The cognitive obstacles addressed by Tall are:

(i) The *parsing* obstacle, which consists of students' proclivity to read from left to right expressions such as 2+3a which may lead them to evaluating them incorrectly. This obstacle seemed to originate because in most western civilisations language and algebra are both read and written from left to right, but the latter not always is processed from left to right. This conflict is also manifested when a child reads a+b as '*a* and *b*', thus he simplifies it as *ab*, which is interpreted as a+b. Another manifestation of this conflict is when the child reads an expression like 5+4a from left to right, 5+4 gives 9, and considers the full expression as 9a.

(ii) The *expected answer* obstacle. An explanation for this obstacle may be found in the previous children's arithmetic experience, where activities usually end with obtaining a numerical answer, thus a child sees an expression like 5+6 as 'an invitation to calculate', whereas an expression like 5+6b cannot be calculated unless the value of b is known. This erroneous expectation seems to be an explanation of why the child does not accept an open expression as an answer, because an answer cannot be expressed as something which still looks like a process.

(iii) *The product-process obstacle*. This obstacle is caused by the fact that an expression like *2+3a* represents both the process by which the computation is carried out and the product of that process. If a child thinks only in terms of process, he/she will conceive the expressions *3(a+b)* and *3a+3b* quite differently, because in the former expression the child requires the addition of *a* and *b* before multiplication of the result by 3, while the latter requires him to multiply both *a* and *b* by 3 before adding the results. Tall and Thomas have observed that though such a child may be asked to observe that both expressions lead to the same product when being evaluated, the child still needs to realise that, for example, an expression like *4a+5* comprises any process consisting of multiplying a number by four and then adding five to this result. They argue that for the child to overcome that obstacle, the child needs to 'encapsulate' the process as an object so that he/she can talk about it without the need to carry out the process with particular values for the variable. When the encapsulation has been performed, two different encapsulated objects can be compared and regarded as being the same object if they always give the same product.

The research carried out by Tall and Thomas consisted of two experimental studies which compared children's achievements using a control/experimental group methodology, and used teaching activities aimed at (i) encouraging the mental image of a letter as store label representing a single number which could be changed and, by extension, could hold any of the variety of numbers, and (ii) evaluating expressions (including equivalent expressions) in BASIC and in standard algebraic notations. Their results showed that those children who were exposed to the computer algebra work were better able to cope with the parsing obstacle. This obstacle was faced "by discussion between teacher and pupils, then encouraging pupils to reflect on the possible meaning of different expressions using Basic and the algebraic maths machine to build theories as to the meaning of equivalent expressions" (p. 135).

The process-product obstacle was faced by seeing that the expression represented both process (in the cardboard maths machine) and product (in the algebraic maths machine and in BASIC). The fact that the computer performed the process in BASIC helped pupils concentrate on the notation as product and to think of it as a conceptual entity. The results showed that the experimental pupils were more likely to consider expressions such as 2(a+b) and 2a+2b to be the same. In addition, they were better at conserving equations. For example, the experimental pupils were more likely to be able to holistically grasp information, for instance in seeing that p+1 can be treated as the variable in the equation 2(p+1)−1=5, whilst the control pupils showed a proclivity to process the information by multiplying out, collecting terms, and solving the equation.

These results suggested that "the beginning phase of the subject -giving meaning to the variable concept and devising ways of overcoming the cognitive obstacles- is fundamental to laying a foundation for meaningful algebraic thinking" (p. 127). In this respect Tall and Thomas point out that though the initial difficulties in the learning of algebra cannot be totally avoided by the incorporation of computers into the mathematics classroom, "these difficulties are exaggerated by the teaching of algebra in a context in which the symbolism does not make sense to the vast majority of pupils and that the success rate can be significantly improved by giving a coherent meaning to the concepts by using a computer" (p. 128).

Logo-based research approaches

Papert's contribution of the Logo programming language and his idea of a Logo Microworld has probably been the most documented computer-based approach to mathematics education (see, for example, Noss, 1985; Noss, 1986; Hillel and Kieren 1988; De Corte and Verschaffel 1989; Hoyles and Sutherland 1989, Hoyles and Noss, 1992, Ursini, 1994). These studies focus on investigating the notions of angle, length, variable and function, which are the central concepts when working with the Logo programming language. Perhaps the most striking result which has emerged from these studies is that the role of the teacher, for structuring and guiding children's learning within the Logo environment, is more important than Papert's original work suggested (Sutherland, 1987).

The effects of using Logo in the developing of children's algebraic conceptions was studied by Noss (1986) as a sequel of a longitudinal investigation of the mathematical environment created through Logo programming (Noss, 1985). Noss (1986), carried out an exploratory study aimed at examining the extent to which Logo experienced children are able to exploit their Logo-based knowledge to construct meaning for elementary algebraic concepts. The study focused on the possible influence of children's Logo learning in facilitating the conceptualisation of algebraic variable, and their ability to formalise in a non computational context.

Noss' (1986) research was based on the conjecture that "symbolic representation of mathematical concepts in the form of computer programs, engages the learner in the *doing* of mathematics" (p. 335); the study intended to clarify what else the children may be learning through programming with Logo. The research was carried out with eight children who had been the subjects of case-study (Noss, 1985). These children were working with Logo during 18 months (approximately 50 hours). The method consisted of interviews with children as they solved pencil and paper rule-formulation problems. The results of the study showed "that the metaphor of typing in a value at the keyboard can be viewed as a means of conceptualising a range of values while only necessitating the consideration of specific values (one at a time)" (p. 350), and that certain aspects of Logo syntax were used by the children in the construction of formalised rules. Noss concluded that, provided appropriate conditions, the Logo programming experience may help children make use of the algebra embedded in the Logo environment to develop algebraic ideas within a non-computational context. On the basis of the results of this study, Noss suggested that Logo programming may be used to help young children in forming primitive conceptions of algebraic notions, in time, children integrate them as part of a system of algebraic understandings, and that "the challenge consists of finding ways of creating Logo settings which are sufficiently transparent and flexible to enable the learner to gain control over the embedded concepts" (p. 355).

Within the context of a large study (Hoyles and Sutherland, 1989), which was aimed at providing a framework to organise and analyse children's work within the classroom, Sutherland (1987) investigated the hypothesis that certain experiences in Logo will provide pupils with a conceptual basis of algebraic ideas which will enhance their work with paper and pencil. This research led Sutherland to formulate categories for analysing different features of children's work within the Logo environment, these categories clearly show how the results of previous research were taken up to be re-studied within a technological-based setting: acceptance of the idea of variable; understanding that any variable name can be used; understanding that a variable name represents a range of numbers; understanding that different variable names can represent the same value; acceptance of 'lack of closure' in a variable dependent expression; ability to establish a second-order relationship between variable dependent expressions; ability to use variables to formalise a general method.

Hoyles and Sutherland found it necessary to modify Papert's principle of encouraging pupils to create their own 'project goal' and decided to intervene to provide pupils with specific goal-oriented 'Logo projects' to work with. Sutherland (1993) reports that "towards the end of the first year of the Logo Maths Project we changed our strategy and intervened with teacher-directed tasks which introduced pupils to the idea of variable in Logo. This change in direction resulted from our ongoing analysis of the data, that is the finding that pupils were not choosing projects in which it was appropriate to introduce the idea of variable. We were now structuring the situation for the pupils" (p. 98).

Their analysis through the framework described above allowed them to design and put into practice various modifications to the classroom principles advocated by Papert. For example, Hoyles and Sutherland highlighted the role of teacher within the Logo environment. Among the implications to using Logo in the classroom Hoyles and Sutherland recommend that the teacher should: be aware of the different styles of interacting with Logo so he/she can promote that one particular style does not predominate; be familiar with student's misconceptions about the programming language, be alert to intervene to remediate them; intervene to encourage students to predict and reflect, and, if necessary, take the initiative making explicit the mathematical features of activities in the Logo environment.

Hoyles and Sutherland (1989) reported that the Logo environment, with appropriate teacher intervention, can help students overcome fears and restrictions generated by previous experience while working in mathematics. Particularly, they emphasised the potential of the Logo environment as a source for experimenting with an idea, as promoting a dynamic representation of mathematical concepts, and the role of the computer feedback in provoking a cognitive conflict which results in child's reassessing his/her mathematical concepts.

As far as the notion of variable is concerned, Hoyles and Sutherland (1989) reported that children did not necessarily use variables in programming tasks nor choose projects which needed this idea. Thus, they suggest that the use of variables might be introduced through simple general procedures within undemanding contexts. They point out that once the pupil is engaged in such a task he/she is in a better position to "become confortable with the idea that a letter can stand for a range of numbers and then, with a wider range of experience, can develop their intuitive understanding of pattern and structure to the point where they can make a generalisation and formalise this in a Logo program" (p. 223).

Later on, these guiding lines to shape teaching intervention were put into practice (Healy, Hoyles & Sutherland, 1990;). They investigated the role of peer group discussion in linking their understanding of algebraic ideas and their Logo experience. Their findings indicate that those pupils who could operate on a variable performed better on the algebra interview than those who could not. The majority of these pupils who could operate on a variable understood that a variable represent a range of numbers and accepted lack of closure in an algebraic expression.

Ursini (1996) investigated the potential of Logo in helping children construct algebraic concepts prior to formal algebraic instruction. She found that the use of Logo favoured the development and the formal expression of the intermediate steps that initially supported pupil's formal expression of a general method. A crucial point in supporting pupil's activity was to encourage them to focus on the method for solving a problem as opposed to focusing on the result obtained. The pupil's numeric background and their capability for working with particular cases were basic elements supporting this approach.

Another relevant approach using Logo was carried out by Cuoco (1995). Cuoco addressed the function concept from two points of view: "an epistemological perspective that describes how the function concept may develop in students, and the various ways functions are used in mathematics" (p. 79). He elaborates on the basis of a framework which consists of three levels of abstraction: action, process, an object concepts of function. These levels come from Dubinsky's framework (to appear) which has its roots in the genetic epistemology of Piaget. In order to carry out experimental work, Cuoco used three different computational media that support both the different uses and levels of the function concept: The Function Machines, Logo and ISETL (Interactive Set Language).

The Function Machine allows students to start with isolated calculations and to gradually interiorise calculations into procedural entities. Logo is used to provide an environment in which students build an experiment with processes, compare them and begin to manipulate them as data. ISETL is intended to support expressions for higher order functions allowing students to manipulate functions in a mathematical way.

The experiments addressed the notion of algebraic equivalence between functions, the basic idea was that if two different expressions produced for every input the same outputs the functions are equivalent. The students were given a tabulating primitive that allows them to see a table of values for any function they model as a Logo procedure, and asked to find several different Logo procedures that produce the same table. Another type of activity was to tackle algebra word problems which can be modelled by means of a function. The students could then find the output which corresponds to a given input of that function. This procedure entails solving an equation as a particular case of a function.

Cuoco reports that the Function Machines prove to be a useful tool for students to find processes where all they could see before as isolated calculation (for example, 3x+2 conceived as multiplying by three, then adding 2), and Logo as a medium to express students' procedural thinking (directly typing to f :x; output 3* :x+2; end). Using the Logo command 'tabulating primitive' students can produce a table of values, for example, TAB "f 1 10, produce the table corresponding to the pairs (x, f(x)) from x=1 to x=10.

Cuoco emphasises that the level of mathematics of the students he was working with does not determine the computational media used. "What is important is the degree to which students have interiorised the functions under consideration; students who are able to model a situation with a Logo procedure are already viewing the function at hand as a process" (p. 90). Another result reported was that the activity of finding various Logo procedures which produce the same table proved to be a powerful resource in helping students to build, experiment, and compare processes and it guided them to the idea of function as an object. ISETL was used to confront students with higher levels of complexity tasks, such as operating with functions. An important characteristic of ITSEL is that it uses construction and notation that "are as close as possible to the language of mathematics" (p. 92).

Cuoco concluded that none of these three computational media allowed students to experience every facet of the function concept. A second conclusion was that students experience with the three computational media helped them developed a broad sense of mathematical function. The final conclusion puts forward the idea that the constraints of the medium strongly influence the ways in which students think of the situation being modelled.

Spreadsheet-based studies

Though spreadsheets were originally designed for business administration, their mathematical nature has been exploited so as to create environments that have been used to study issues concerning modelling problem situations in mathematics and sciences

(Matos and Carreira, 1994; Matos, 1995; Neuwirth, 1995; Sutherland, Rojano, Ursini; Molyneux, and Jinich, 1996; Sutherland, Rojano, Mochón, Jinich, and Molyneux, 1996, among others), and coping with algebra word problems concerning features of the transition from arithmetic to algebra (Sutherland and Rojano, 1993). Due to the aims of this thesis, the review will concentrate in those spreadsheet-based approaches that have a focus on introductory algebra.

The spreadsheet's column/row display and its crunching number facilities have been used to help children organise the involved relationships in word problems, where the spreadsheet code has been exploited to help children structure and express their reasoning by means of a formal code (i.e., A5*3 in one cell and (A5*3)^2 in another). Neuirth (1995) describes the spreadsheet environment by means of a spatial arrangement metaphor for representing structural relationships. He claims that creating formulas in spreadsheets can be a completely different process to dealing with formulas using paper and pencil and emphasises that the most important difference is that in the spreadsheet the variables do not have names (as it occurs in Logo), the role of variables is played by cells. The cell itself corresponds to the concept of a variable whereas the contents of the cells corresponds to the current values of the variable. The spreadsheet program allows us to combine variables into new values using the mouse-driven-interface, that is, creating formulas is done by pointing at the cells containing the values needed. Neuirth describes this technique as "the gestural representation of mathematical formulas" (p. 172).

Demana and Leitzel (1988) focused attention on using various numerical computation tools in helping students make the transition from arithmetic to algebraic reasoning. They used spreadsheets and calculators to investigate the relations among variables. Their strategy was to emphasise the search for patterns in tables of values for related variables to introduce students to formal algebraic expressions for describing such relationships. The results obtained by Demana and Leitzel suggest that this computation transition to the abstractions of algebra is noticeably more effective than traditional approaches.

Sutherland and Rojano (1993-1996) have designed activities that allowed them to investigate the potential of the spreadsheet in helping young children (10-15-year-olds) move from non-algebraic strategies to more algebraic approaches when coping with negotiating algebra word problems solutions. On the basis of their previous results using computer environments, Sutherland and Rojano have been trying out different strategies to help overcome the children's reluctance to spontaneously work with the unknown when facing situations involving generality. Their results have shown that work with the spreadsheet helped pupils to accept the idea of working with the unknown, "an idea which most pupils find difficult" (Sutherland, 1996a, p. 5). Their findings suggest that the algebra-like spreadsheet symbolic code can be used to mediate the algebraic approach. They argue that, in a spreadsheet, a critical feature in helping children move from a non-algebraic approach to a more algebraic strategy is that the pupils first use a cell to represent the unknown by a cell reference (for example, x), then other mathematical relationships are expressed in terms of this unknown. Then pupils can use pointing with the mouse to support the expression of mathematical relationships. When a given problem has been expressed in the spreadsheet code pupils can vary the unknown either by coping down the rules or by changing the number in the cell representing the unknown. This method has drawn encouraging results. Sutherland and Rojano (1993) reported that with appropriate teacher's intervention, the use of spreadsheets help pupils move from non-algebraic to more algebraic solutions of algebra word problems. They have also stressed the idea of using the spreadsheet cell as a variable to successfully introduce the notions of algebraic equivalence and inverse function (undoing a spreadsheet rule).

Sutherland (1996) in reviewing the results obtained from her previous research conjectures that "work with computers was doing more than allowing to pupils progress along stages. Pupils seemed to transgress any traditional sequential learning path and approached problems in ways which did not fit easily a number of aspects of this way of looking at algebra learning" (p. 7).

Calculators in the classroom

The use of calculators in the mathematics classroom has been a theme of great debate during the last two decades. One of the main topics addressed has been the teacher's concerns about the rapid introduction of calculator use into a teaching tradition strongly oriented to help children develop arithmetic skills in elementary school. Nowadays we have evidence from large scale studies which have shown that the use of calculators does not necessarily harm pupils' basic arithmetic skills.

Hembree and Dessart (1986) studied the effects of hand-held calculator use in precollege mathematics. This study was intended to elucidate why, despite the apparent potential of the calculator facility in supporting student's problem solving skills and concept development, and encouraging discovery, exploration and creativity, the calculator had caused little impact on redirecting curriculum development and had failed to enter most of mathematics classrooms. By the early 1980's less than 20% of the elementary teachers and less than 36% of the secondary teachers in the United States have employed the calculator in mathematics instruction (Suydam, 1982). Hembree and Dessart carried out their investigation by using a meta-analysis method to integrate the findings of 79 studies on the use of calculators. These studies were carried out in the United States during the late 1970's and the early 1980's; and focused on the effects of using calculators in Grades K-12.

In each study one group of students had been permitted to use calculators within a period of thirty school days, the machines were used for computation or to help develop concepts and problem solving strategies. During the same period each study included a comparison group which received instruction on the same mathematical topics but with no access in-class to calculators. The meta-analysis showed that the use of calculators in concert with traditional mathematics instruction apparently improves the average student's basic skills with paper and pencil, both in working exercises and in problem solving. Particularly important for a rationale to calculator use in the classroom is that Hembree and Dessart found that calculators greatly benefit low-and high-ability students in problem solving. Among other findings they emphasised the potential of using calculators during tests. In this respect they found that, across all grades and ability levels, the support provided by the calculator helped students achieve much higher scores than paper-and-pencil efforts. In particular they observed the most positive effects in low-level and high ability students in problem solving. The study found also beneficial effects of using calculators on student's attitude toward mathematics; Hembree and Dessart reported that "across all grade and ability levels the students using calculators showed a better attitude toward mathematics and better self-concept in mathematics than students not using calculators" (p. 83).

Hembree and Dessart (1992) analysed nine additional research studies on calculator use in precollege mathematics. The new data strengthen the conclusion that "using the calculator during instruction may improve paper-and-pencil skills for low average and high-ability students in addition to those of average ability" (p. 26). With regard to using calculators during tests, they found continued advantage from calculators in computation and better advantage from devices in problem solving. In terms of students' attitudes the new data supported previous findings that calculators help promote a better attitude toward mathematics and specially better self-concept in mathematics. With regard to prevailing uses of the calculator in different school levels they found that in the early grades the use of calculators is frequently for familiarisation, for checking work, and for problem solving, whilst in the senior high school the emphasis is made on using the machines as tools for calculation and reference.

To what school policies for calculator use is concerned, Hembree and Dessart found that most schools tend to have a single classroom set of the devices, whilst it seems clear that for most efficient use, a calculator should be made available for each student. Ruthven's work (1992, 1995) gives further support in this respect. The Graphic Calculators in Mathematics Project provided evidence for the advantages of calculator use as an individual resource (Ruthven, 1992, 1992a, 1993a, 1993b). The data showed that the calculator promotes that the learning takes place "more privately and informally than on a classroom microcomputer with its public display exposed to scrutiny by hovering enthusiasts all too ready to offer advice or take control" (p. 92). This project brought to light the fact that, despite the potential of computing in teaching and learning mathematics, a major obstacle is that most mathematics classrooms have a single computer, which makes the access to computing resources very limited. Additionally, this fact encourages many teachers to use the computer for demonstration purposes with students' access to the machine being under the teacher's control. This situation means that few students are able to make spontaneous use of computing facilities. In this respect the graphic calculator offers an alternative: a class set of calculators is comparable in expense to a single microcomputer. Ruthven (1995) explored the potential of the calculator as a personal resource across the curriculum. He reported on a project where each Year 7 pupil in two Cambridgeshire school (1993/1994) was provided with a graphic calculator for their personal use. The project drew encouraging data with regard to the feasibility of school policies to give pupils personal ownership of calculators. The pupils showed a degree of involvement and initiative that suggests that such a school policy exerts a positive influence on pupils attitudes to technology and its use.

Graphic calculator environment

The graphic calculator offers facilities previously available only on micro computers, as production of graphics of functions and relations; some advanced prototypes provide also facilities to perform symbolic manipulations (Shumway, 1988). The large screen display, exploratory functions of graphing, and multiline display offered by graphic calculators are now being exploited to design tasks and teaching approaches intended to promote student's creativity and exploration (see, for example, Usiskin, 1987; Demana and Leitzel, 1988; Shumway, 1990; Winter et al., 1991, Hector, 1992; Ruthven, 1992a, Burill, 1992).

The potential of graphic calculators is described by Vonder Embse (1992) who focuses on the facilities provided by the large screen display and the editing and graphing functions. He conceives the graphic calculator as "the ideal environment for middle school students to learn prealgebra concepts, to explore patterns and processes, and to solve problems" (p. 65). Vonder Embse emphasises the facilities offered by the large screen display of the graphic calculator, it allows us to show multiple inputs and outputs, which helps students explore and experiment in ways that help better understand the various parts of a process, patterns or problem situation through the integration of numerical and graphic representation. These large screen functions (like a computer screen) allows us to show the entire key sequence of complicated string of numbers like $3+4(5-6\div7)$, this facility may help novice users to visualise and hold in memory operational and grouping symbols as they are entered into the calculator. In particular, he argues that the screen display of the keying sequence closely resembles the actual mathematical expression, which may help students link the work with the calculator to their work with paper-and-pencil. Multiple problems and answers can be shown at the same time as the keying sequence, which may help students grasp the functional idea of input and output and the concept of variable.

The editing functions offered by the graphic calculator, like *Insert* and *Delete,* allows correction or changing the input rather than retyping a problem. The *Replay* function allows the user to repeat the last command line that was calculated, to edit the command, and to recalculate a new result. These editing functions may encourage and support 'guess-and-check' problem solving strategies; for example, the student can re-enter a single problem as many times as he needs within a process of guessing and refining a problem solution. The graphing calculator provides an environment which may help students to integrate the numerical, graphical and symbolic representations of mathematical relationships. This environment may help students understand the relationship between numerical values in a table, the symbolic rule relating table values, and the graphical representation of the table and rule.

Graphic calculator-based studies

Research studies have critically examined the possible benefits provided by the graphic calculator. The results drawn from these studies have suggested that the graphic calculator environment influence students' strategies and mathematical attainment. Ruthven (1990) carried out an investigation which consisted of comparing the mathematical performance of upper secondary school mathematics students who were provided with graphic calculators as a standard mathematical tool, with that of students of similar background without access to graphing technology. The study examined students' responses to two types of item: (i) symbolisation items, which call for an algebraic description of some cartesian graph, and (ii) interpretation items, which call for the extraction of information from verbally contextualised graphs.

The study was carried out within the context of a large project (Graphic Calculators in Mathematics). This project enabled six small groups of classroom teachers to work with at least one class of students having access to graphic calculators during their two-year advanced-level mathematics course. The teachers were free to plan the work in their own classes and periodically met together to exchange ideas and review progress. Toward the end of the first year the students' performance was examined using the items described above. There were classes parallel to the project class in four of the six schools, which provided a sample of 87 students, 47 were in project classes having regular access to graphic calculators; the 40 students in the non-project classes had no access to graphing technology, with exception of seven students who bought their own graphic calculators. A questionnaire was administered by each class teacher, the first section of the questionnaire gathered general information about each student, the second section was a 40 minute test containing 12 items where students were allowed to use the computing and graphing technology. The items were designed to examine issues which cannot be faced by directly using automatic graphic procedures so as to give no direct advantage to calculator users.

The results indicated that the project group was better at recognising a graph as a particular case of a family of functions, and they were better in producing a precise symbolic description of the graph using the relevant information. A statistical analysis of the results showed that the project group substantially outperformed the comparison group in symbolisation items, but there were no significant differences between the groups with regard to interpretation items. This finding suggested that "the treatment effect is not an artefact of the design of the study, but genuinely attributable to the use of graphic calculators in the project classes" (Ruthven 1990, p. 447). This study led Ruthven (1992) to emphasise how the use of calculators influenced students' strategies and mathematical attainment. The research data suggested that as the students gained command on the calculator's automatic procedures they developed variations of both graphic-trial and numeric-trial approaches. In this respect Ruthven emphasises the value of informal approaches, particularly in helping students who might have no other resource to face problem situations: "although the trial-and-improve approach has limitations it often gives students a means of tackling a problem that would otherwise be intractable. Moreover, it is an example of a *moderated* procedure, dependent for its efficiency and success on intermediate judgements made by the user" (p. 97). The work done by some of the students showed that as they become familiar with specific mathematical situations their judgements become more sophisticated; students' informal approaches indicated that they were not guessing blindly but were interpreting graphic feedback in the light of crucial mathematical principles. These results suggests that, under appropriate conditions, the graphic calculator may play the role of a tool for cognitive growth.

The fact that graphic calculators naturally demand the use of unambiguous notation has been exploited to create calculator-based settings which may bring benefits to symbol manipulation. Ruthven (1992) analyses certain graphic calculator features which seemed to highlight the role of the machine as a cognitive tool. He puts forward that calculators offer more than a simple mechanism for calculating, that the calculator can be used so that it rather plays the part of a "medium for thinking and learning" (p. 95). One of the salient features of the graphic calculator is that it uses a symbolic language which is situated in the calculation environment, this feature gives an operational referent to the calculator's formal language (Ruthven 1993a). For example, the *Ans*[1] key can be used to design tasks which require the pupils to produce expressions using a symbolic algebra-like language (i.e., the expression *Ans*×2 produces the sequence 2, 4, 8, 16). Another facility offered by the calculator symbolic language consists of using any of the alphabet

[1] The ***Ans*** function automatically stores the last result of a numerical calculation so that we can operate on that number by typing a program which uses the ***Ans*** symbol as variable. For example, if the last computed result was 5, the keying sequence ***2Ans+1EXE*** produces 11 as a result. Pressing ***EXE*** again will produce 23, and so on.

letters as variables to produce algebraic expressions which can be evaluated by giving numerical value to the letter(s) used.

In this respect, Ruthven (1993b) reports advantages of exploiting the above features of the graphic calculator. A class of 13-year-olds worked with activities based on iterative patterns defining sequences of numbers. In particular, he found that activities based on number sequences that can be represented by linear expressions like $Ans \times 3+2$, lead students to produce different but equivalent expressions. This situation encourages students to ask questions of whether and why these rules are the same, which provides "a highly motivating context for discussion, out of which I could develop two key algebraic ideas: the distinction between seeing such expressions as descriptions of a calculating procedure, and seeing them as descriptions of the results of a procedure" (p. 23). He proposes a second type of activity which consists of simply programming the machine using letters[2]. These techniques can be used to help students to cope with problem situations as 'how many hours are there in any given number of days' by assigning a value to the letter D, and calculating $24 \times D$ or $24D$ (p. 24). This calculator facility can also be used to design activities focusing on algebraic equivalence, for example, establishing the equivalence of expressions as 2a+2b and 2(a+b) by means of comparing their numerical value.

2.4. The metaphor of algebra as a language.

Though within a wide range of perspectives, a good deal of mathematics educators have assumed the position of considering algebra as a language. Mason et al (1985) declared: "algebra is firstly a language, a way of saying and communicating (p. 1) ... If algebra can even be usefully referred to as a thing at all, then it must be as a language in which to express your thoughts and awareness of patterns" (p. 54). Pimm (1987) interprets the claim that mathematics is a language in a particular way, "namely as a metaphor" (p.

[2] The graphic calculator uses letters to design each of the 26 immediately available memories. A letter can be used within the *computing mode* by providing it with a numerical value, for example if 2 is assigned to A, the expression A+5 will display 7 as a result. A letter can be used in the *programming mode*, which automatically updates the value of the variable as a number is entered in the calculator.

xiv). For Sutherland & Rojano (1992) "algebra is the language of mathematics, a language which can be used to express within mathematics itself, or within other disciplines" (p. 2). Bell (1992) goes further and proposes that the algebraic language should be learned "in a way more similar to that in which the mother tongue is learnt" (p. 11-12). Shoenfeld (1993) talks of the formal stuff of algebra as the syntax of a rich language, on this view algebra is seen as the language of abstraction and manipulation of symbolic entities.

The view of algebra as a language has been changed and broadened by technology. The availability of different representations for expressing quantitative relationships such as graphics and tables has influenced the ways in which mathematics educators conceive the teaching and learning of algebra. From this view algebra can be seen as a language with various dialects: symbols, graphs and tables. Particularly, the new technological resources seem to strengthen the view of algebra as a language for generalising arithmetic. In this respect Tall (1993) declares that he advocates "introducing algebraic symbolism by using it as a language of communication with the computer, through programming in a suitable computer language ... it develops a meaningful algebraic language which can be used to describe number patterns, and it gives a foundation for traditional algebra and its manipulation" (p. 38). Tall suggests that using the symbolic code of a computer language is "to speak it in a context where the algebraic language is seen to make sense" (p. 39). Tall compares Logo and BASIC as environments where the children may use variables closely related with the idea of a letter standing for a number. He suggests that whilst Logo is a better language for children to explore ideas, BASIC symbolism is closer to traditional algebra. For example, a BASIC command such as a=3 followed by the command PRINT a+1 will produce 4, which seems to be an easy task for children to predict what happens with the command PRINT a+2. The results of such a command can be predicted and then tested. Although BASIC language is close to traditional algebra there are differences, for example the multiplication sign (i.e. 2*x instead of 2x or 2×x).

The first, and perhaps the most clear attempt to approach the learning of mathematics based on the metaphor of learning natural language was made by Papert. Papert (1980) wrote:

> I take from Piaget a model of children as builders of their own intellectual structures. Children seem to be innately gifted learners, acquiring long before they go to school a vast quantity of knowledge by a process I call 'Piagetian learning' or 'learning without being taught'. For example, children learn to speak, learn the intuitive geometry needed to get around in space, and learn enough of logic and rhetoric to get around parents -all this without being 'taught'. We must ask why some learning takes place so early and spontaneously while some is delayed many years or does not happen at all without deliberately imposed formal instruction (1980, p. 7).

The idea of language as means of communication is at the kernel of Papert's conception of 'Logo programming microworld':

> Programming a computer means nothing more or less than communicating to it in a language that it and the human user can both 'understand'. And learning languages is one of the things that children do best. Why then should a child not learn to 'talk' to a computer?. The computer can be a mathematics-speaking entity. We are learning how to make computers with which children love to communicate. When this communication occurs, children learn mathematics as a living language ... (Papert 1980, pp. 5-6).

A weak point in Papert's metaphor relating the Logo environment to the learning of the mother tongue relies on the assumption that "the interaction between student and computer will have similar qualities to the interaction between child and caregiver" (Ruthven, 1993, p. 194). Ruthven argues that the caregiver-child interaction allows the child to receive and produce linguistic code, and stresses the role of that interaction in helping the child develop reception skills which in the long term supports him to develop producing skills. He claims that this characteristic of the caregiver-child interaction strongly contrasts with the child-computer interaction, because when working with the computer the child may produce programming utterances but he never receives programming code from the computer. Ruthven also argues that the computer within the Logo environment cannot fulfil the role of the caregiver as an interlocutor "who is able to handle a considerable degree of ambiguity and unorthodoxy in the code produced by the child and still respond appropriately" (p. 194). On this view the caregiver plays a crucial role in promoting effective communication meanwhile the child comes to master the language, and this feature of children-adult interaction is not fulfilled by the computer within the Logo environment.

The studies and positions above mentioned have influenced the design of the calculator-based environment used in the present research, particularly with regard to the role played by the teacher within the calculator setting. The results drawn from investigating the Logo environment led to carefully design goal-oriented tasks intended to structure and guide children's actions within the calculator-based environment used in this study. For example, though it was expected that children might 'spontaneously' develop notions for algebraic equivalence and inverting linear functions, specially designed activities were included to help them develop these notions (formats 3 and 4). In the same vein, Ruthven's criticism of Papert's metaphor of computer programming as 'learning to talk to the computer' led to the use in the present study of a specific method intended to accomplish as much as possible the caregiver-child interaction. The method consisted of marking children's work after every classroom session and letting children have the teacher's observations before they dealt with new tasks. This strategy helped the teacher have an updated view of each child's work during the whole study and allowed him to give the children opportune feedback delivering 'new calculator code' to children when they needed, for example, when children were not able to produce by themselves expressions of the form ax+b, and in the case of using parenthesis (this point is further discussed in Chapter 4).

Contrasting with Piaget's constructivist conception of development of language as a by-product of the development of other, non-linguistic cognitive operations, Bruner's (1982) investigations led him to conclude that language acquisition takes place through a 'teaching' process which, roughly said, consists of a highly framed child-adult interaction where "the child is hugely aided in his mastery of linguistically mediated requests by the social interactions into which he enters with his mother and other adults" (p. 1).

This thesis is strongly influenced by Bruner's view so his view is discussed in more detail in Chapter 3.

Other authors have subscribed a position that relates to this theme, some of them do not present algebra as a language either directly or metaphorically, but deal with the semantics and syntax of algebra (for example, Kirshner, 1989).

The above conceptions and theoretical positions will be reviewed when discussing the results of this thesis in Chapter 7. The present study focuses on exploring possible ways that allow us to set up a mathematical environment where algebra could be learned in a way more similar to that in which the mother tongue is learnt. In order to do this, we investigate the potential of graphic calculators as means of creating an algebraic context which helps achieve connection with 12-13 years old children's arithmetical background. A critical feature of such an environment is that of allowing the use of algebraic code so strongly attached to the number realm that children can permanently check the algebraic utterances they produce by means of basic arithmetic facts. We have arranged that environment trying to mirror Bruner's concept of Language Acquisition Support System to be discussed in Chapters 3 and 4.

The research findings discussed in the above sections encourages the hypothesis that the resources offered by computers and calculators may allow the study of algebra as a language-in-use, and, furthermore, that the use of algebraic code helps the students generate meanings for that symbolic language that allow them to use it for problem solving and expressing and justifying generalisations.

2.6. Concluding remarks and implications

This section discusses the ways in which the research approaches and results discussed in the preceding sections have influenced the present study. The research previously carried out influenced fundamental parts of the present thesis, in particular: (i) its aims, (ii) the choice of graphic calculators as a computing device, (iii) the research method, and (iv) the theoretical approach. These points are further discussed in what follows.

Implications for the aims of the study

The aims of the present study were influenced by those contrasting results obtained from the research developed on the transition from arithmetic to algebra, before and after the incorporation of technological resources to school mathematics. The research carried out prior to the incorporation of technological resources into the mathematics classroom reports that children have strong difficulties in interpreting the letters as algebraic entities and how these difficulties may affect their approach to symbolising the relationships in problem solutions. These results shed light on the problem of translating from one symbolic system (natural language) to another (algebraic code) which shows the difficulty that novice algebra students have when using a new symbolic system.

In contrast, many different technological-based approaches to the teaching and learning of algebra have reported that pupils do not seem to present the previously reported difficulties in understanding the role of letters as variables (see section 2.3 in this chapter). Though technological-based approaches have provided promising results on pupils' achievements in modelling algebra problem situations, there has not been carried out research (or not published yet) on the effects of computerised-based environments on how pupils, who have not had previous algebra instruction, may evolve from their first pre-algebraic steps, to algebraic manipulation, to the stage of coping with algebra word problems using the algebra-like code provided by a computing device. This poses, at least, the following questions:

- May the apparent benefits of computerised-based environments in helping children develop the notion of variable provide support for children to understand the role of algebraic symbolism as a tool for coping with algebraic manipulation and problem solving?
- Which teaching strategies seem to be suitable in helping children develop algebraic notions and strategies that support them to cope with algebra problem solving?

The search for answers to these questions determined the aims of the present study set out in Chapter 1, the study intends to investigate:

1. The notions that pupils develop for algebraic language when they meet it through using calculator code.
2. The extent to which the use of the calculator language helps pupils cope with simplifying similar terms within linear expressions, inverting linear functions, and transforming a linear algebraic expression to obtain a target expression.
3. The algebraic strategies that children develop through working with the calculator.
4. The extent to which the use of the calculator language as a means of expressing general rules governing number patterns, helps children grasp that the algebraic code can be used as a tool for coping with problem situations.

According to these aims, this research is intended to document how pupils' notions of letters as mathematical entities evolve, how pupils cope with algebraic equivalence and inverting linear functions, and the extent to which pupils are able to negotiate solutions for algebra word problem through using the calculator code.

The choice of the graphic calculator

The choice of graphic calculators as the technological device on which to create the learning environment was determined by results drawn from the research reviewed in the previous sections. The most influential factor determining the use of calculators in this study was the constraints imposed by the existing differences between the computer code and the algebraic symbolic system, such as requiring the student to link the actions made by using the computer code with their work with standard algebraic notation (see, for example, Cuoco and Tall in this chapter). BASIC programming language has more similarities with the algebraic code than Logo language and spreadsheet code, but still presents noticeable differences with the standard algebraic symbolism. This fact encouraged the investigation of the potential of the graphic calculator in helping pupils to develop algebraic notions and strategies. An influential factor in this respect were those results that have showed the potential of calculators as a personal resource which provides a symbolic language that presents strong similarities with the algebraic code, both in notation and syntax (Ruthven 1992, 1992a, 1993a, 1993b). This point will be further discussed in this section when dealing with methodological issues.

Implications to the methodological approach

The choice of the method used for data gathering and analysis was mainly influenced by the qualitative approaches employed by Noss (1985), Sutherland (1987), Hoyles & Sutherland (1989) and Hoyles and Noss (1992). Their work has showed the potential of the case-study method in investigating in depth children's notions and strategies within computer-based environments. As will be further discussed in Chapter 4, individual interviews were chosen to be the main source of data for this research.

Other crucial methodological feature consisted of designing the tasks to be done in the classroom. There were two main features involved in designing the tasks: determining their content, and the ways in which the content may help exploit the use of calculator language. On the one hand the content of the tasks was influenced by the use of the calculator, but the final form of the tasks was mainly determined by the work developed by Mason on expressing generality as a route to algebra. In deciding the content of the tasks Mason's claim was assumed that "the facility in manipulation of generality follows as confidence in expression develops and as multiple expressions for the same thing arise, and that use of algebra to solve problems depends on confident expression of generality using the as yet-unknown supported by awareness of the role of constraints on variables" (Mason, 1993, p. 2). This assumption led to the design of tasks involving expressing generality, in particular, tasks which require pupils to describe the rules governing given number patterns by means of calculator language (see Formats 1 and 2, Chapter 4, section 4.6); then, tasks which involve manipulating multiple expressions for the same thing were included (see format 3, Chapter 4, section 4.6); finally, tasks focusing on using the calculator code to solve problems which depend on confident expression of generality using the as yet-unknown were included (see, formats 5 and 6, Chapter 4, section 4.6). Though various studies have found that the introduction to algebra through number patterns-based approaches present difficulties for the majority of children (Stacey, 1989; Herscovics, 1989; Arzarello, 1991; MacGregor and Stacey, 1992; MacGregor and Stacey, 1993; Stacey and MacGregor, 1996), technology-based approaches (Ruthven, 1993a, 1993b), have provided results which suggest that the support provided by graphic calculators to handle symbolic representation may help to take advantage of number patterns for introducing school algebra. Johnson's (1991) view that mathematical ideas become 'more dynamic and useful when described as a procedure' provides further support for the position taken in the present research. In particular, Johnson's position that 'the potential of algorithms in the teaching and learning of school mathematics relies on the process of producing them', which is precisely the building block in helping children learn the calculator code as a language-in-use.

With regard to the ways in which the calculator code would be used to tackle the tasks, the work by Booth (1984a), Johnson (1991), Tall (1993), and Ruthven (1993a, 1993b), were important antecedents. Booth's work influenced the present study in the sense of paying special attention to the difficulties that children have in formalising their methods, which are likely to be due to children's tendency to using informal procedures that have proved to be successful in facing arithmetic tasks but failed in the algebraic case. This finding led to the inclusion of tasks where the pupils, from the beginning of the study, confront number patterns which must be described by expressions of the form $ax+b$. Expressions of this form involve typing a string of operations 'in one piece', which are intended to break with children's experience in elementary school (see Appendix, Format 1). With regard to the issue of how to link pupils' previous arithmetic experience to the new algebra-like calculator code, Ruthven's claim that calculator language 'is situated in the calculation environment' and this provides an operational referent to the calculator formal code, provides support to the way in which the calculator language used in the present research. In this respect, an original feature of the present research is that it exploits the programming calculator language, which currently, seems to be the only study that has devised in this way the fact that graphic calculators naturally demand the use of unambiguous notation.

Implications to the theoretical approach

As has been mentioned earlier, the work by Papert (1980) was the major theoretical antecedent in the direction of taking advantage of the computer as an interlocutor for children to communicate with by using a formal language. Papert's position may be briefly described by the metaphor of 'learning French in France". This study was primarily inspired by Papert's view and further refined by later research findings from working with Logo-based environments in the classroom (Sutherland, 1987, 1989;1993 Hoyles and Sutherland 1989). Sutherland's research brought to light that teaching intervention plays a more important role for structuring and guiding children's learning within the Logo environment, than Papert's original work suggested. Another important view in this sense was Ruthven's (1993) criticism of Papert's metaphor relating the Logo environment to the learning of the mother tongue. Ruthven pointed out that Papert's original position relies on the assumption that "the interaction between student and computer will have similar qualities to the interaction between child and caregiver" (p. 194). Ruthven argues that the computer cannot play the caregiver's role of delivering linguistic code to the child, and stresses the role of the caregiver-child interaction in helping the child develop reception skills which in the long term will support him to develop language producing skills.

Both, Sutherland (1995) findings and Ruthven's view, influenced the present study in the sense of searching for a theoretical background that supports a teaching position that allows the introduction of algebra as a language-in-use. Chapter 3 addresses this point.

Final Remarks

The preceding section has addressed the immediate implications of the algebra research literature for the major structural parts of the present study. Most of the research and theoretical positions reviewed in this chapter will be taken up later when discussing the results of this study (Chapter 7).

CHAPTER 3

THEORETICAL BACKGROUND: A LINGUISTIC-BASED APPROACH TO INTRODUCTORY ALGEBRA

Introduction

The aim of this chapter is to discuss those theoretical issues that were used to shape the research on which the present thesis is based. The theoretical referent adopted in this study mainly relies on Bruner's research on the acquisition of the mother tongue (1978, 1985, 1980, 1982, 1983, 1990). Some outcomes and principles drawn from Bruner's work were borrowed both to inform the study on children's algebraic achievements and to provide support for the design of a mathematical environment within which the teaching of algebra could be approached attempting to simulate the ways in which children learn the rudiments of natural language. Bruner's research revolves around "how the young child acquires the uses of his native language and how by using language first for limited ends the child comes finally to recognise its more powerful, productive uses" (Bruner 1983, p. 7). There are two major research questions in Bruner's work: How does a child acquire language, and what may facilitate this learning? The theoretical issues this chapter deals with do not attempt to provide an exhaustive review of Bruner's work on language acquisition, rather it offers a concise discussion of some selected topics that are more relevant to the research approach adopted in the present study.

The adoption of children's language acquisition as a theoretical referent for this thesis was inspired by the following ideas: (i) as has been mentioned in Chapter 1, the author of the present thesis had found empirical evidence that encourages the idea of conceiving school algebra as a language which allows us to communicate and sort out mathematical tasks; (ii) the graphic calculator offers facilities that allows us to put the pupil in the position of using the calculator's language without having previous instruction about its structure and syntax rules. More specifically, the graphic calculator allows the user to type and evaluate algebraic expressions by means of a code which presents strong similarities with the algebraic notation, this calculator's facility offers a link between numerical facts and the algebraic notation and its syntax rules: it is hypothesised that if the learning environment is suitably arranged, such a link may provide the children with a referent that helps them deal with the algebraic sign system being supported by their previous arithmetic knowledge.

The main idea borrowed from Bruner's work in this research was that language is 'taught', that is, language is neither a by-product of intellectual development or a result of a sort of children's imitation of adult's speech; Bruner's investigation proposes that the process of language acquisition actually starts earlier than the child is able to produce his/her first lexical utterances, that language acquisition comprises a lengthy period of preparation where children acquire those clues which enable him/her to make sense of what it is talking about, and later to decode what initially appears as a continuous flow of language utterances. As will be further discussed throughout the chapter, such a preparing stage for language acquisition relies on a highly framed adult-child interaction which was recast to shape the classroom setting so that pupils can meet the calculator code as a language-in-use; the use of calculator code is put into a context which may help pupils negotiate meanings for the new language by numerically exploring the effects of using specific 'calculator's utterances'; such a classroom setting intends to provide a 'pre-algebraic' stage which includes those uses of algebraic code that constitutes the essence of an introductory algebra course, as expressing generality, symbolic manipulation and negotiating problem solutions.

This chapter is organised as follows. First an overview of Bruner's theoretical view and his main empirical results are presented. Second, some prior approaches to language acquisition are discussed and commented on in terms of Bruner's position. Third, a summary of Bruner's results and concepts is offered. Finally, the chapter is closed by discussing how Bruner's principles and empirical results were recast to make a link between language acquisition and the learning of algebra. This discussion is further elaborated in Chapter 4, which describes in detail the methodological approach adopted in this thesis.

3.1. Bruner's Approach to Language Acquisition: An overview.

Bruner's research throughout the 1970's addressed questions about what it is, beyond a splendid nervous system, that makes it possible for a child to acquire language so swiftly and effortlessly. From this view the acquisition of language not only addresses a psychological issue, it "is also a thorn in the side of linguistics, a testing ground for theories in the philosophy of mind, and a major enterprise in that part of anthropology and sociology that concerns itself with how a culture gets passed on" (Bruner, 1980, p.155). A central premise within such a theoretical position is that, to become a member of a linguistic community, "an aspirant human being must not only learn about language as a system of well-formed, rule-bound utterances about the world, but how to get things done with words in the language in the world" (p. 156). It is in this sense that the calculator-based environment in this study is meant to be: such an environment should help the pupil to become a member of the 'algebraic community'. In order to do this the classroom environment was artificially arranged intending that the pupil meets the algebraic code as a language that allows him to do things through 'algebraic speech' within the 'mathematical world' framed by the classroom activity.

Bruner considers the syntactic, the semantic, and the pragmatic facets of language as constituting three great problem spaces in language acquisition:

- *Syntax* deals with the problem of how we acquire our facility in managing well-formed utterances governed by a grammar.
- *Semantics* concerns the nature of the relation between words and possible worlds as we know such worlds.
- *Pragmatics* has to do with the manner in which we finally come to use well-formed utterances about possible worlds to affect others.

Bruner's investigation suggests that the pragmatic will be first and the syntactic will be last, but more importantly, that the acquisition of language-in-use strongly depends upon the interdependence of well-formedness, meaning and reference, and conventions of use. He emphasises that syntax, semantics and pragmatics "are not derivable each from the other, but rather that each serves as a scaffold for aiding in mastery of the others" (p. 155).

For Bruner, language is learned by using it, and central to its use are what he called *formats*, which are highly framed interactions between mother and child. His research focused principally on two of the great functions fulfilled through language by native speakers: *indicating* and *requesting*. In particular, Bruner studied the ascent of these functions from a pre-linguistic beginning to a level of linguistic proficiency where speakers are at the take-off point that will lead them into ordinary or conventional language use. His findings led him to conclude that since the child masters some general aspects of communicative use before making much progress in either the semantic or syntactic domain, pragmatics provides the most general support system for mastery of the more formal aspects of language. In terms of the present thesis, the concept of format is borrowed to frame the teacher-pupil and the pupil-calculator interactions. Bruner's concept of format was recast so as to help children develop the referring function of language by confronting pupils to describing general number patters by means of the calculator language, and to develop the requesting function by encouraging the pupils to gain experience in using calculator language to answer particular questions about general number patterns. These points are further discussed in section 3.4. in this chapter (see Mathematical Formats).

3.2. An overview of different approaches to language acquisition.

This section is intended to place the pragmatic approach to language acquisition within the framework provided by other approaches to this topic. The intention is to discuss those issues which point at the boundaries between different theoretical positions in order to get a better position from which to recast Bruner's pragmatic view of language in terms of the learning of introductory algebra.

Behaviourist approach to language

Bruner (1978) strongly criticised the positions taken towards language acquisition as existed up to the late 1950's, from what he considered its first enunciation by St.

Augustine to its last in Skinner (1957). This view conceived the process of language acquisition as a particular way of learning and was explained by general theories. According to Bruner, most behaviourist learning theories operated with principles and with experimental paradigms that had little to do with the phenomena of language. Their principles and their research paradigms were not derived from the phenomena of language but from general behaviour, which leads one to see language like any other behaviour, and to explain it as just another set of responses. For instance, transfer of response from one stimulus to another was assured by the similarity between stimuli. In Bruner's (1983) terms "language learning was assumed to be much like, say, nonsense syllable learning, except that it might be aided by imitation, the learner imitating the performance of the 'model' and then being reinforced for correct performance" (p. 32). The Behaviourist learning theory put the emphasis on *words* rather than on *grammar*. Consequently, it omitted almost entirely the combinatorial and generative effect of having syntax which made possible the routine construction of sentences never before heard and which did not already exist in the adult speech to be imitated.

Cognitive Developmental Approach

The most important study within this position was provided by Piaget. Bruner (1982) considered that the development of language was for Piaget "a by-product of the development of other, non-linguistic cognitive operations ... Language, so to speak, was simply a symptom of the automatic semiotization of those growing cognitive operations that achieved reversibility and made possible such things as object constancy, and so on" (p. 10). Karmiloff-Smith (1979) points out that for Piaget, language did not constitute a separate problem space. Karmiloff-Smith did not find in Piaget's work a clear explanation of how exactly these non-linguistic cognitive operations can support the ability "to recognise and use predicational grammar or the definite marking system of anaphora or to generate only well formed sentences". In this respect, Bruner questioned how the egocentric child mastered the shifter pronominals, like *I* and *You*, when he was supposed not to be able to take another's perspective. Bruner considered that "the origin of Piaget's massive intellectual scotoma about accounting for language acquisition was his stout resistance to the idea that language could lead to or even nourish non-linguistic cognitive development ... Therefore he made it a by-product of non-linguistic development" (1982, p 11). Bruner argues that the only way in which language acquisition could be shown to be an indicator of cognitive growth was by showing that it correlated with language development. "But then, so too does body weight ..." (p. 12).

Bruner (1982) shows that the last decade of research strongly supports the view that language acquisition is aided by the acquirer gaining world knowledge, that language is aided by maturation; and it is aided by a privileged interaction between the child and a caregiver who is somewhat well tuned to his linguistic level. In this view, children's learning of language is seen as a result of a highly framed social interaction rather than solely a result of children's cognitive development.

A syntactic Approach

Perhaps Chomsky's (1957) work is the best representative of the syntactic approach. His thesis was that the acquisition of the structure of language depended upon a recognition device which Chomsky called the Language Acquisition Device (LAD). Briefly said, the LAD is what allows one to accept the surface structure of any natural language as input and to recognise its deep structure by virtue of the affinity of all natural languages with a universal linguistic deep structure that humans knew innately. The output of the LAD was the grammatical rules of the language by which the learner was enabled to generate well-formed utterances and none that were ill-formed.

A radical interpretation, that according to Bruner is attributable more to his psychological followers than to Chomsky himself, claims that acquisition of the formal, syntactic structure of language is completely independent of either world knowledge or of social interaction with speakers of the language. In this view LAD is basically a recognition mechanism "by which the infant speaker is enabled to recognise the deep regularities in the surface structure of the local language to which he is exposed by virtue of knowing already the nature of the deep structure of all languages" (Bruner, 1982, p.2). The radical Chomsky's followers propose that the child simply recognises the realisation of these universals in the local language, although he may have encountered only degenerate instances of it. In summary, syntax was independent of any kind of knowledge of the world and of communicative function, thus the acquisition of syntax is only constrained by limitations of performance, like the child's limited attention and memory span.

Bruner's perspective on this radical view was that, the detail of language acquisition is entirely a matter of performance rather than competence, which he considers as a part of the innate child's endowment. Thus, the growth of performance depends entirely on the growth of attention span and information processing capacity. Bruner (1982) points out that "nobody has ever been very clear about whether this 'performance variable' grew simply with maturation (the more radical exponents rather implied it did) or whether it depended as well upon the acquisition of other forms of non-linguistic knowledge" (p. 3).

Bruner considers that the radical view of Chomsky's theory is undoubtedly wrong, mainly due to the fact that the LAD may require priming in order to operate. In any case, Bruner acknowledges that Chomsky succeeded in getting people to look afresh at language acquisition and to look at it as the acquisition of real language rather than in the form of nonsense syllables.

A semantic approach

According to Bruner (1980), the basic assumption that distinguishes the semantic from the syntactic approach to language is that the former proposes that children had a working knowledge of the world before they acquire language, and that such knowledge of the world assist them in learning the language. Bruner (1982) thinks that this is not an unreasonable start: "if you know what it is that you are trying to distinguish in the real world, you will presumably be alerted in some way to linguistic distinctions that reflect or map into or simply accompany those distinctions" (p. 6). However, he points out that it is a weak claim in the sense that there is nothing about conceptual distinctions between, for instance, phases of a goal-directed activity, that will give the child any hint as to how grammatical or lexical distinctions are realised linguistically. Bruner proposes that if the child did not have some code breaking device to aid him with the language, knowledge of the real world would aid him very little. Bruner's point is that without such a code breaker the child would be back in the behaviourist-type position, operating by pure induction: "we know that is an impossible position if only for the formal reason that for any finite utterance, an infinite number of grammars is applicable and nobody lasts long enough to narrow down the contending field by pure induction" (1982, p. 7).

There were many efforts to develop a generative semantics out of which grammatical hypotheses could be derived. They were oriented by the principle that a knowledge of the world, organised in terms of a system of concepts, might give one hints as to where distinctions could be expected to occur in the language.

Bruner points out that the linguistic distinctions and their mode of being realised (whether morphologically or syntactically) have to be acquired as well, and that the issue of whether rules of grammar can somehow be inferred or generalised from the structure of our knowledge of the world still conveys open questions which deserves further research.

Bruner (1983) thinks that the semantic position does not seem to explain how the child gets to the point of being able to put together verbal strings so as to create utterances that assign appropriate perspectives to scenes, in Bruner's terms "the hypothesis is interesting but in that special way in which, as in Japanese prints, landscapes are interesting by virtue of being enshrouded in mist" (p. 159). Nevertheless he points out that the semantic approach incorporates child's actions and this new element provides a different dimension for explaining language acquisition. For example, Fillmore (1977) hypothesises that meanings depend upon scenes and this involves an assignment of perspective. Particular words used impose a perspective on the scene and sentence decisions are perspective decisions. If, for example, the agent of action is forefronted in perspective, the nominal which represents it must be the 'deep subject' of the sentence. In this sense a child's action "comprises a set of universal, presumably innate, concepts which identify certain types of judgements human beings are capable of making about the events that are going on around them ... who did it, who it happened to, and what got changed" (p.24). The basic structures are these action categories, and different languages go about realising them in different ways: by function words, by inflectional morphemes as in the case endings of Latin, syntactic devices like passivization, and so on. These grammatical forms are the surface structures of language that depend for their acquisition on an understanding of deep semantic concepts about action.

Nelson (1975) offers an argument that the child approaches the task of acquiring language already equipped with concepts related to action. In essence, she proposed that the child came to language with a store of familiar concepts of people and objects that were organised around the child's experience with these things. "Because the child's experience was active, the dynamic aspects would be the most potent part of what the child came to know about the things experienced. It could be expected that the child would organise knowledge around what he could do with things and what they could do. In other words, knowledge of the world would be functionally organised from the child's point of view" (pp. 4-5).

Bruner, while investigating the child's acquisition of prelinguistic and linguistic means for making requests, found that requests are in the deepest sense dependent upon the child's understanding of action and on enlisting another in carrying out one's own actions. In this sense Bruner's findings coincide with the semantic-oriented studies in that action provides the child with a set of formats that permit him to organise his concepts sequentially in a sentence-like form. He points out that the capacity to do this is a "basic form of representation that the child uses from the start and gradually elaborates. In effect, it is what guides the formation of utterances beyond the one-word stage" (p. 160).

A pragmatic Approach to Language Acquisition

This section deals first with pragmatics in language generally, then with language acquisition in particular. In Bruner's view the syntactic model assumes the child as simply a consumer of linguistic input; the semantic model assumes him as a rather lone problem solver sorting out the world around him in terms of his actions upon it and generalising that knowledge to language; and the pragmatic approach takes us directly into the issue of the social context of language.

For Bruner (1982) pragmatics is "the study of how speech is used to accomplish such social ends as promising, humiliating, assuaging, warning, declaring, requesting ... Its elements do not 'stand for' anything: they are something ... even silence, though it cannot be specified syntactically or semantically, may speak volumes in the context where it occurs. It is certainly not just like a grammatical deletion rule where patterned absence implies presence" (p. 13). From this perspective, language is conceived of as a vehicle for doing things with and to others.

The implication is that pragmatics necessarily relates to discourse and, at the same time, is always context dependent, dependent upon a shared context. Discourse presupposes a reciprocal commitment between speakers that includes at least three elements: (i) a shared set of conventions for establishing speaker intent and listener uptake, (ii) a shared basis for exploiting the deictic possibilities of spatial, temporal, and interpersonal context; and (iii) conventional means jointly for establishing and retrieving presuppositions. These three elements, announcement of intention, regulation of deixis, and control of presupposition, give discourse its future, present, and past orientations.

A great many acts of discourse are found to be ways of 'tuning' these forms of reciprocal commitment. Within a radical position some linguistic theorists have proposed that the grammatical categories of language exist to ensure such tuning as well as to assure reference and meaning. For example, Benveniste (1971) raised the question of the function served by personal pronouns, a universal feature of all known languages. Why are they needed, he asked, when in fact we could accomplish the same semantic ends more reliably by using nominals to specify people or objects rather than having to employ shifter pronominals. His answer was that shifters like *I* and *you* serve as economical ways of sharing and regulating the perspectives of two speakers through reciprocal role shift.

Bruner's stance is that the pragmatics of discourse cannot be based upon ordinary grammatical categories alone. For grammar is traditionally based upon the concept of the sentence and on sentence parts. But the rules of discourse depend for their power upon the privileges of occurrence of expressions in discourse, not just in individual sentences.

There is another sense in which interaction motivates grammatical rules. One of the major tasks in interacting with another is the regulation of joint attention, early interaction abounds in procedures for regulating attentional perspective on scenes in the form of vocatives, demonstratives, pointing gestures, and intonational contours, employed by both adult and child.

The publication of Austin's *How To Do Things With Words* in 1962 seemed to encourage the emergence of pragmatics. Austin's work addresses the issue that utterances cannot be understood in terms of their propositional content. Utterances also have an operational function based on convention. In this view, mastering a language involves not only knowing how to string together propositions, but also how to meet the conditions on the appropriate making of utterances.

Bruner (1980) analysed the utterance 'would you be so kind as to pass the salt?' to exemplify Austin's claim. Bruner argues that it is not designed to "prove the limits of the listener's compassion, but rather is a conventionalised request for the condiment named that also takes into account certain conditions imposed on discourse, for example, that the voluntarism of the addressee be recognised in the framing of a request" (p. 161).

In Austin's view, an utterance can be thought of as containing not only a propositional form, its locution, but also an illocutionary force whose uptake by an interlocutor guides his assignment of interpretation to the locution. The speaker's communicative intentions are relevant issues for pragmatics, though the relation between the form of the locution and its force remains obscure.

The role of discourse within a pragmatic approach establishes a link with how language is acquired. There are at least two questions crucial to acquisition: communicative intention and shared presuppositions. From the pragmatic perspective the intention is what needs to be decoded in speaking and understanding a language. For using a language not only depends upon a shared grammar and a shared lexicon that makes it possible for speaker and hearer to map each other's utterance into context if they are to extract meaningful propositions from talk. Communication also depends upon shared notions about conditions of utterances. Searle (1969) calls 'speech acts' the combined locutionary form and illocutionary force and describes them as having at least three conditions: a *preparatory condition* (laying appropriate ground for the utterance), an *essential condition* (meeting the logical conditions for performing a speech act, as for example being uninformed as a condition for asking for information related to a matter), and *sincerity conditions* (meeting the psychological requirement that, for example, you really want the information you are asking for).

In this respect, Bruner (1983) reports that the learning of speech acts seems, somehow, more clear than the learning either of syntax or semantics. Syntactic rule-following is rarely followed by corrective feedback. And even semantic mastery often seems strikingly unassisted. Speech acts, on the contrary, work or don't work and are openly corrected. What is striking about them, too, is that they are present in some recognisable form even before lexico-grammatical speech develops. The child learns how to realise his intentions communicatively by conventionalised gestural or vocal means before he ever learns to do so by the use of locutions. In this sense, primitive speech act patterns may be established in the child's repertory as a kind of matrix into which syntactic and semantic achievements can be set.

Bruner found that what is very apparent in examining any early corpus of discourse (not just speech, but discourse in which the mother is included) is that the child manages quite well in making his intentions clear, and that the mother is very much more preoccupied with teaching the child how, when, and where to make appropriate utterances than she is with issues of syntax or meaning. In contrast, it is very rare to find any early instances of syntactic correction and there is even some suspicion (Nelson, 1973) that semantic corrections may lead to the suppression of the lexical items that produced the difficulty. It suffices to note only that there is a large investment of time and energy during acquisition in helping the child learn how to say things in a fashion appropriate to the discourse, even if syntax is ragged and semantics hazy.

This brings us directly to the heart of the problem relating to a pragmatic approach to language: the role of the 'tutor' in language acquisition. The pragmatician's stress on shared convention and presupposition requires a more active role for the adult in the child's language acquisition than just being a 'model'. The pragmatic route requires that the adult be a partner. The research until the late 70's (a great deal of it summarised in Snow and Ferguson, 1977) indicate that parents play a far more active role in language acquisition than simply modelling the language and providing input for Chomsky's LAD. A suitable way of characterising the parents role is by the current phrase: the parent's role is 'fine-tuned' to the level of their children. Bruner (1980) reported that "semantically, syntactically, lexically, intonationally, in terms of sentence length and complexity, parents get down to the level on which their children are operating and move ahead with them at a rate that shows remarkable sensitivity to their children's progress" (p. 160). Parents are very skilful in using the 'baby talk' register and the level of their speech matches strikingly well the level of their children's speech. This children's approach to language seems to pose a dilemma. As Brown (1977) puts it, how do you teach children how to talk by talking baby talk with them at a level which they already understand? In this respect Bruner's research indicates that the answer has got to be that the important thing is to keep communicating with children for by so doing one allows them to learn how to extend the speech that they have into new contexts, how to meet the conditions on speech acts, how to maintain topics across turns, how indeed to regulate turn-taking and adjacency pairing, and so on. Above all, children are learning what is worth talking about, when and how. They are learning scripts about interaction with others through communication.

3.3. Bruner's Research Findings

The research by Bruner on the process of language acquisition cover a vast range of fundamental psychological issues. According to the specific purposes of the present thesis, the most relevant result drawn from Bruner's theoretical and empirical work may be summarised in the existence of a Language Acquisition Support System (LASS).

Bruner claims that there is a LASS that makes it possible for the infant to enter the linguistic community, a system that "frames the interaction of human beings in such a way as to aid the aspirant speaker in mastering the uses of language. It is that system that provides the functional priming that makes language acquisition not only possible, but makes it proceed in the order and pace in which it ordinarily occurs" (p. 120).

Bruner's work sustains the hypothesis that "in order for the young child to be clued into the language, he must first enter into social relationships of a kind that function in the manner consonant with the uses of language in discourse -relating to intention sharing, to deictic specification, and to the establishment of presupposition" (1982, p. 15). He calls such a social relationship a *format.*

A format "is a rule-bound microcosm in which the adult and child do things to and with each other. In its most general sense, it is the instrument of patterned human interaction" (p. 16). The formats are crucial vehicles in the passage from communication to language; formats pattern communicative interaction between infant and caregiver before lexico-grammatical speech begins.

"A format entails a contingent interaction between two acting parties. It is contingent in the sense that the responses of each member can be shown to be dependent upon a prior response of the other" (Bruner, 1982, p. 16). Since each member of the pair has a goal and a set of means towards its attainment, it is necessary that the format fulfils two conditions: first, that a participant's successive responses are instrumental to that goal, and second, that there is a discernible order in the sequence indicating that the terminal goal has been reached.

Formats, defined in this sense, represent a simple instance of a 'scenario'. Formats, however, grow and can become as varied and complex as necessary. Their growth is affected in several ways. They may in time incorporate new means or strategies for the attainment of goals, including symbolic or linguistic ones. They may move toward coordination of the goals of the two partners not only in the sense of agreement but also with respect to a division of labour and a division of initiative.

Formats are also modular in the sense of being manageable as subroutines for incorporation in larger scale, longer term routines. A greeting format, for example, can be incorporated in a larger scale routine involving other forms of joint action. In this sense, any given format may have a hierarchical structure, parts being interpretable in terms of their placement in a larger structure. The creation of higher order formats by incorporation of subroutine formats is one of the principal sources of presupposition. What is incorporated becomes implicit or presupposed. Formats, except when highly conventionalised, cannot be identified independently of the perceptions of the participants. In this sense, they have the property of contexts generally in being the results of definition by the participants. The communal definition of formats is one of the major ways in which a community controls the interaction of its members. Once a format is conventionalised and 'socialised' it comes to be seen as having exteriority and constraint and is seen as having objective status. Eventually, they provide the basis for speech acts and can be reconstituted as needed by linguistic means alone.

Bruner suggests that one special property of formats involving an infant and an adult, "is that they are asymmetrical with respect to the 'consciousness' of the members, one 'knowing what's up,' the other not knowing or knowing less. The adult serves as model, scaffold, and monitor until the child can take over on his own" (1982, p.18).

A final feature of Bruner's (1983) research which has been relevant for this thesis is that his findings suggest that referring and requesting are the main linguistic functions that young children develop in the process of mastering language as means of communication. Referring has to do with "the causal historical chain that links an introductory *referential event* (when a person tries to indicate, however crudely, what he has on his mind) and some later referential episode (when each member of the communicating pair assigns a referential interpretation to a message that passes between them)" (p. 67). This framework presupposes four things: (i) that individuals can signal to each other 'that they have a referential or communicative intent', (ii) that two parties to a conversation may refer to the 'same' topic with widely different degrees of precision, (iii) that reference is a form of social interaction having to do with the management of joint attention, and (iv) that there is a goal-structure in referring , "referring is sustained not only by intent to refer, but by appropriate means for doing so and by specification as to one has succeeded" (p. 68). To where requesting is concerned, Bruner characterises the object of request as "to get somebody to deliver the goods" (p. 91). His research findings suggests that requesting seems to be the form of language use which most deeply depends upon context.

3.4. Implications of Bruner's theory for the teaching and learning of algebra.

This section is aimed at discussing how some theoretical principles drawn from the pragmatic approach to language, and more particularly from Bruner's research, were recast in order to outline a teaching approach to introductory algebra which make it possible to observe how children learn algebraic code as a language-in-use.

As has been said earlier, the major aim of this thesis is to explore the learning of algebra within the pragmatic paradigm of language acquisition. This implies conceiving a teaching approach in which the learning environment mirrors, as much as possible, those social circumstances which frame the acquisition of the mother tongue. Accordingly, such an approach must be different from both a syntactic or semantic teaching-oriented approach. In order to make this clear an attempt is made to characterise such teaching trends, which in no sense tries to imply that one particular trend might be more effective than another.

A syntactic-oriented approach implies a teaching position in which the pupil plays the role of 'consumer of linguistic input', more specifically, a consumer of those rules governing the use of algebraic code. In this approach the teacher mainly acts as the model for pupils to imitate, and the teaching contents are characterised by introducing algebra starting with the study of polynomial expressions and their rules to perform legal transformations, then equations and finally functions. A good deal of text books exemplify this approach: first the contents supported by examples, then a list of exercises and problems to be solved.

A semantic-oriented approach may be characterised by the pupil playing the role of 'a lone problem solver sorting out the world around him in terms of his actions upon it and generalising that knowledge to algebraic language'. This kind of approach relies on supporting the introduction of algebraic syntax by providing pupils with 'meanings' for the symbolic system. The teacher is the most active person in the classroom and plays the role of a model to be followed by pupils. The teacher tries to offer as many different approaches to problem solving as possible intending to help pupils induce general properties or rules from a limited number of examples. In some countries (at least in Mexico), this approach is generally based on introducing algebra by confronting the pupil with problem situations, then equations as a means for modelling these problems, then polynomial expressions, and finally functions.

A pragmatic-based approach must allow pupils to enter into algebra by using its code, this principle marks the main difference with the other approaches. Though it seems paradoxical to propose starting to use a formal symbolic language before we know at least some definitions about it, there is a good example: children learn their native tongue without any previous knowledge of grammar rules or definitions. As will be discussed next, Bruner's work on language acquisition can be recast so that it sheds light on the use of certain technological facilities offered by graphic calculators and how these facilities support a pragmatic entry into the learning of algebra.

In principle, both natural language and school algebra deal with learning to use a sign system. One of the most overt differences between acquiring these sign systems is that natural language is learnt within the rich environment provided by adult-child interaction, it embodies a learning process which, as has been discussed, is hugely aided by what Bruner calls a Language Acquisition Support System (LASS). In this respect, the present thesis proposes that, following Bruner's concept of LASS, the school setting can be artificially arranged to create an *Algebra Acquisition Support System* (AASS), a system in which the teacher's expertise in using algebraic code is strengthened by incorporating a technological component (graphic calculator) that allows him/her to achieve a milieu where children encounter the algebraic code as language-in-use for expressing and negotiating mathematical ideas. In other words, the calculator use is shaped so that it allows children to use the algebraic code as 'a vehicle for doing things with and to others'. In such a milieu, the calculator code not only allows children to use a symbolic system to describe mathematical relationships, but, furthermore, the use of the calculator code positively conveys algebraic actions since this notational code is embedded within a sign system governed by algebraic rules. The role assigned to the calculator language resonates with Bruner's view that 'if the child did not have some code breaking device to aid him with the language, the knowledge of the real world would aid him little ... without such a code breaker the child would be back in the old empiricist position, operating by pure induction'.

Finally, the mathematical content included in the particular pragmatic approach in this study centres around the notion of function. Without following a rigid sequence, syntactic and semantic algebraic features are treated, including certain types of problem situations, which involve modelling and solving equations. These features have more to do with methodology so a more detailed discussion of this is made in Chapter 4.

The setting up of such an Algebra Acquisition Support System relies on three theoretical assumptions of a pragmatic nature. The first has to do with providing a context which makes discourse possible, a context that helps children use 'algebraic speech' to accomplish mathematical ends. The second assumption relies on the feasibility of creating 'mathematical formats' which allow the framing of both teacher-pupil and pupil-calculator interaction. The third deals with the role of the teacher within a pragmatic approach to the teaching of algebra. These assumptions are discussed in more detail in the following paragraphs.

A context for algebraic discourse

As has been discussed earlier, discourse is always context dependent, that is dependent upon a shared context. In terms of encouraging 'algebraic discourse' the experimental tasks were designed so that children can resort to their prior arithmetic knowledge to make sense of and negotiate answers for the involved questions. That is, it was hypothesised that children's command of *elementary arithmetic facts* may play the role of a shared set of conventions which supports teacher's communicative intent and pupil's language uptake. The interaction speaker-listener is twofold: teacher↔pupil, and pupil↔calculator, which implies translating from one sign symbol system (natural language) to another sign symbol system (calculator code). The role assigned to arithmetic as a shared symbol system attempts to fulfil the preparatory condition for discourse: laying appropriate ground for the utterance in the process of translating from one symbol system to another. In summary, the arithmetic context is intended to provide conventional means for establishing and retrieving presuppositions so as to help children negotiate meanings for the symbolic code in-use. Here, meaning is conceived of as "a culturally mediated phenomenon that depends upon the prior existence of a shared symbol system ... where symbols depend upon the existence of a 'language' that contains an ordered or rule-governed system of signs" (Bruner, 1990, p. 69).

Mathematical formats

The second assumption upon which the Algebra Acquisition Support System depends, consists of the feasibility to create *mathematical formats*. Recasting Bruner's concepts, a mathematical format is conceived of as a routinised and familiar setting which frames teacher-child and child-calculator interaction to make communication effective, 'fine-tuned'. Such formats must allow children to enter into mathematical tasks that function in a manner consonant with the uses of language in discourse (algebraic code), and shape a microcosm where the child's successive responses are instrumental to a goal, and there is a discernible order in the sequence indicating that the terminal goal has been reached. Following Bruner's terms, these formats must also grow and become as varied and complex as necessary (see Appendix, Formats 1-6).

An important feature that shapes the structure of the mathematical format is that its intention is to embody the development of the two basic linguistic functions referring and requesting discussed earlier in this chapter. In structuring the mathematical formats an attempt has been made to help children develop these linguistic functions. Recasting Bruner's concepts into mathematical terms they consist, respectively, of *using* algebraic code to (a) describe general relationships (referring) and (b) negotiate problem solutions (requesting). Thus, every format was structured so that it includes a section which requires children to describe a general number relationship by means of calculator code (*referring*), and another section that requires the children to answer questions and negotiate problem solutions by means of calculator code (*requesting*). That is, to 'get the calculator to deliver the goods'. The 'referring section' always has the same form, while 'requesting' varies in content, this structure allowed the creation of higher order mathematical formats by incorporating subroutine formats, which, in theory, provide the pupil-teacher interaction with a basis for shared presupposition (what is incorporated becomes implicit or presupposed). A more specific discussion of the content and structure of the various mathematical formats used in this research is carried out in Chapter 4.

The role of the teacher

The teacher fulfils the role of proficient user of the calculator's language. It is assumed that his/her command of the language lets him/her guide the way it is used so as to fine tune it to the children's present level of knowledge. Among the specific responsibilities implied by this heavy burden, the teacher must be aware that the calculator only affords a good environment for children to produce and test algebraic utterances but does not provide children with 'new words' that allow their 'algebraic discourse' to flow. Thus, the teacher must fulfil the crucial role of helping children produce those expressions that go beyond their creative initiative.

Another important teacher's responsibility is to handle and respond appropriately to those unorthodox or ambiguous utterances that children may eventually produce. In this respect the teacher's intervention is guided (in theory) by the pragmatic principle that children acquire language with the intention of communicating; it is in the interest of fulfilling his communicative intentions that the child recognises new structural tricks relating to language. An intent of this kind uses whatever props are available. In this sense, the only errors recognised or responded to by the teacher should be those that he knows the child knows he can correct if challenged.

3.5 Conclusions

This chapter has discussed Bruner's main theoretical principles and empirical findings that have influenced the present research. Bruner's theoretical background has been primarily recast to design a mathematical environment intended to help children develop algebraic notions and strategies through using the calculator code, that is, an environment which is intended to allow children to learn about the calculator language through using such a language. As will be discussed in Chapters 4 and 5, this theoretical background will be also used to inform the study on children's algebraic achievements throughout the fieldwork.

The major theoretical premise of Bruner that has been exploited in the present study is that children acquire the mother tongue through social interaction, an interaction which is highly framed and artificially arranged by the adult so that the adult's use of language is fine-tuned to the present level of child's intellectual development. Bruner's theoretical principles have been interpreted to create a learning setting which we have called an Algebra Acquisition Support System (AASS); it is hypothesised that such a system may provide a context which will help pupils attach meanings to the symbolic calculator code so that they can use the new formal code as a tool to cope with describing general number relations and negotiating solutions for algebra word problems.

The pedagogical approach proposed in this study strongly relies on the concept of *mathematical format*. The concept of format was the building block for designing the tasks used in this study. The AASS proposed in this study has been meant to be a pragmatic approach to the teaching and learning of algebra in the sense that it places the mathematical activity into a context where children can use calculator language without having previous definitions or rules governing the use of that formal code; definitions and rules are supposed to be developed through using the calculator language. That is, the AASS relies on the hypothesis that the *use* of language is what *provides meanings* and support *proper production* of linguistic utterances, as opposite to the position which sustains that definitions and rules are what determine appropriate uses of language.

The manner in which Bruner's concept of format was recast into algebraic activities is discussed in more detail in Chapter 4.

CHAPTER 4

METHODOLOGY

Introduction

As was stated in Chapter 1 the general aim of this study is to investigate those learning events that take place when children encounter the algebraic code as a language-in-use. This aim, seen within the theoretical view adopted in this study, has implications for the research method.

First, the research has to be carried out within the mathematics classroom, which is the natural setting in which children learn mathematics. Second, as a necessary condition for the research to be done, the method should include the creation of a teaching/learning setting which places children in the position of users of algebraic language. Third, though observation of children's achievements play a relevant role in this study as this provides empirical evidence for the research, such achievements say nothing in isolation from the learning processes in which they arise, certainly such processes are what provide its explanatory framework.

Consequently, the method used in this study sets out, on the one hand, to obtain information which allows the analysis of each child's possible development throughout the study. On the other hand, the method also needs to allow a wider view of children's development as a 'linguistic community'.

This chapter is organised as follows. It begins with a discussion of the approach to qualitative analysis adopted in this study, then the pilot study is briefly described; finally the following components of the main study are discussed: (i) the calculator's role, (ii) the school setting, (iii) the classroom setting, (iv) the tasks used in the study, (v) the subjects taking part in the study, (vi) the sources of data and data gathering, and (vi) the analytical framework.

4.1. Method

Following Bruner's approach to language acquisition, this research was carried out within the clutter of the mathematics classroom, the natural milieu where children are supposed to learn mathematics. The kind of data drawn from such study is essentially based on children's work episodes, which led to the adoption of a research method based on qualitative analysis. The model proposed by Miles & Huberman (1984) influenced the methodological approach in this research. This model is described in figure 1.

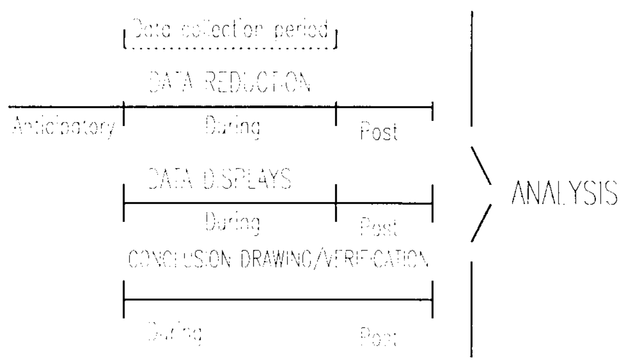


Fig. 1

Miles & Huberman's model proposes that "analysis consists of three concurrent flows of activity: data reduction, data display and conclusion drawing/verification" (p. 21). The model assumes that data reduction and the creation and use of displays must not be taken separately from analysis, rather they are constituent parts of analysis. Data reduction refers to "the process of selecting, focusing, simplifying, abstracting, and transforming the raw data obtained from fieldwork" (p. 21). This process takes place throughout the research study.

Before the data is actually collected, Miles' model suggests an anticipatory phase for data reduction. In this research the anticipatory phase consisted of a pilot study which was intended to refine the tasks and interview protocols (the pilot study is discussed in more detail later on in this chapter). Data reduction was a process that continued after field work. This process included doing summaries and selecting extracts from interview transcripts (Chapters 5, 6 and 7 are examples of how this principle was applied in this research). Data reduction was in fact "a form of analysis that sharpens, sorts, focuses, discards and organises data in such a way that 'final' conclusions can be drawn" (p. 21).

Data display is the second major component of analytic activity. According to Miles & Huberman (1984) a display "is an organised assembly of information that permits conclusion drawing and action taking" (p. 21). In this research data displays were used to help understand what was happening during classroom sessions and interviews so as to allow further analysis based on that understanding. The displays used in this study were matrices showing children's work throughout the study, and narrative text, used to discuss and analyse children's reactions during interviews. This kind of data analysis is presented in Chapters 5, where two case-study files are discussed. Chapter 6 presents a cross-analysis which is more based on narrative text.

The third stream of analysis considered in Miles' model is conclusion drawing and verification. This level of analysis consists of "beginning to decide what things mean, in noting regularities, patterns, explanations, possible configurations, causal flows, and propositions". This level of analysis is carried out both in chapters 5, 6 and 7. In chapters 5 and 6 each set of data is followed by a discussion in which some explanations for children's achievements are put forward. Chapter 7 intends to set up regularities among children's work, and some conjectures are discussed on the basis of the empirical data drawn from the field work.

Data reduction, data displays and the drawing of conclusions interweave before, during and after data collection. (Fig. 2). This interrelation consisted in moving back and forth between different types of data: everyday children's work, video tapes and transcripts from interviews, and notes

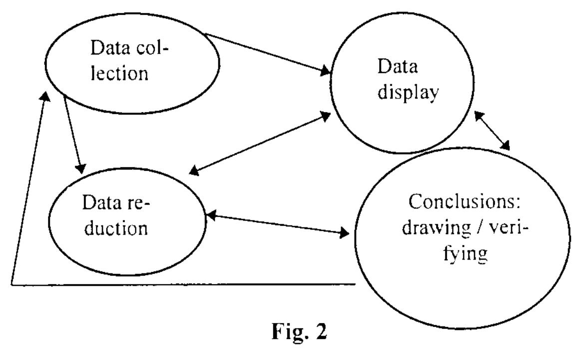


Fig. 2

taken by the researcher after each classroom session. In this sense, qualitative data analysis is an iterative process. Issues of data reduction, of display, and of conclusion drawing/verification result in analytical episodes following each other. The following sections discuss in more detail the different features included in the methodological approach.

4.2. Pilot study

The pilot study was carried out in three phases. The pilot study's major purposes were to refine the aims of the research, to test the strategy for collecting data, and to refine the design of the tasks. The different phases of the pilot study are described next.

- The first phase was carried out with two groups of 25 children (11-12 years old) over 12 weeks (two 50 minutes sessions per week). A sample of two boys and two girls were followed and interviewed during the last two weeks. This phase was carried out in Mexico City in a private school[1].

- The pilot experience held in Mexico led to refinement of the strategy and the adoption of a case-study approach in order to improve the quality of data drawn from the study. Accordingly, a second phase of the pilot study was carried out in London with a 12 year old girl, during twenty sessions of 45 minutes each. Every session was audio taped. This phase allowed refinement of the protocol for individual interviews and incorporation of a number of tasks.

- The third phase was implemented with a school class of 25 children during eight sessions (50 minutes each). This experience helped achieve the final design for the tasks and interview protocols.

Conclusions from the pilot study

The main conclusions from the pilot study are summarised next.

[1] The characteristics of the school are described when dealing with the school setting.

a) The requirement of learning the calculator programming code did not seem to be a serious obstacle for the children. They seemed to be motivated by a sort of curiosity to know more about the calculator facilities that led them to become competent users of the programming mode.

b) The activities designed as input/output number tables were encouraging enough to engage children in the solution of the proposed tasks. The empirical data suggested that these kind of activities match well with the 11-12 years old children's arithmetic background and served to support children to face calculator language as language-in-use. The pilot experience also offered opportunities for refining the experimental tasks.

c) The empirical data obtained from the pilot study suggested that children learned to use the programming code not only as a computing recourse but as a language that allowed them to tackle mathematical tasks by making the machine 'do what they were looking for'. This data suggested that the calculator played the role of a mediational tool that gave support to children in making the transition from a step by step strategy to a more relational-based way of working. The final version of the tasks was achieved on the basis of these findings (see Appendix 1).

4.3. The Main Study

The main study was carried out as a part of the regular one year course which is given in the First Grade of Secondary School in Mexico and was implemented in the same school where the pilot study was held. The main study (fieldwork) lasted eighteen sessions of fifty minutes each where the researcher acted as the teacher[2]. The following sections describe the methodological aspects considered in this research:

- Calculator's role
- School setting
- Classroom setting

---

[2]From now on references to "the researcher" will be made when interviews and data analysis is concerned. References to "the teacher" will be made when dealing with issues concerning the work during classroom sessions.

- Subjects
- Tasks,
- Data gathering
- Analytical framework
- Final remarks.

The calculator's role.

The calculator played a relevant part in this study. A crucial factor determining its role relies on the similarity between its programming code and the algebraic symbol system, both in rules and notation. Another key factor is that the calculator's programming mode allows the children to type and evaluate algebraic expressions, which gives immediate operational and numerical referent to the calculator's algebra-like language. This feature allows children to link an algebraic expression with their numerical value; this fact was the building block from which to offer the calculator's language as a language-in-use to children who have not had any previous algebra instruction. This point is further discussed below.

The calculators used in this study allow at least two ways of representing functional relationships[3]: the analytic expression, used to type a program (figure 1); and the tabular representation, obtained on the calculator's screen by inputting a range of values to the program's variable (figure 2). The tabular representation was used as an arithmetic referent for the analytic expressions. These characteristics of the machine's operation were exploited to create a mathematical work environment immersed in the context of communication. In it, the calculator's formal code is available to anyone with basic arithmetic skills. This environment was shaped as follows:

Fig. 1

? → A:	2A+5
Declaring variables	Programming expression

Fig. 2

1	7
3	11
6	17
7	19

- A teacher, who fulfils the role of proficient user of the calculator's language. It is hypothesised that his command of the language allows him to guide the way it is used so as to fine tune it to the children's present level of knowledge.

- A group of 23 children (11-12 years old), each with their own calculator. This acts as a tool with a new sign system that provides them with a way of expressing algebraically arithmetic procedures.
- Previously designed activities which structure the children's use of the calculator's language. Children use this code to make the calculator do their bidding; that is, when the child is using the calculator's programming mode, he is in a situation where 'by using words' that the calculator 'understands' he achieves his ends.

Work begins by showing the student how to write a program for the calculator and what the program does when it runs. The activity consists of a game-like task in which the children 'guess' someone else's program. Pupils must recognise the numeric pattern shown in a table and program the calculator to produce this table. The activity's structure is given by the game itself, it contains the activity's rules and goals. The features used to shape the tasks so as to mirror Bruner's concept of format are discussed in section 4.6.

To summarise, the activities revolve around learning a code which allows you to express numeric relationships and make calculations with them. The interaction occurs on two levels: student-machine and student-teacher. The underlying hypothesis is that pupils, *through use*, create meanings for the calculator's sign system, somehow emulating the process through which we acquire the basics of our native tongue.

When they engage in these activities the children are using the programming code as the language that the calculator 'understands'. Arithmetic plays the role of context that helps them set up and verify conjectures which they express through the calculator's language. These activities are intended to provide a work environment in which the language is so strongly tied to context that corrections in its use can be made constantly through the context itself. This close relationship between form and context (tables-analytic expressions) is similar to a fundamental characteristic of learning the native

[3] Graphics resources were not used.

tongue, where the child would find it nearly impossible to use linguistic forms without learning in context.

School Setting

The main study was carried out in the same school where the pilot study was implemented. This school was chosen because of the possibility to work with a group of pupils in an innovative way throughout the whole school year. The school is characterised by what in Mexico is called "active school", which is guided by the principle that discipline within the classroom must be derived from the kind of work that the each teacher promotes, that is, the work within the classroom must be motivating enough so as to encourage children to become engaged in the activity.

Another characteristic of this school is that the criteria for accepting a child as a pupil is based on a psychological test which provides elements to judge if an applicant may be of benefit for the school environment and if the school environment may be of benefit for such a pupil. That is, the criteria for accepting a pupil is not based on a pupil's attainment, but in certain pupil's attitudes towards school[4]. On the basis of this fact the Principal of the school considers that their pupils constitute a 'mixed-ability' population. The researcher's previous experience (ten years as a mathematics teacher in State Secondary Schools) suggests that the students population in this school does not significantly differs from the students population within the State Schools in Mexico.

The group of children who took part in the study were 11-12 year olds who were in the First Grade of Secondary School. In order to work with these pupils (they had not had any previous algebra instruction) it was necessary to make special arrangements because the Mexican Curriculum does not include any algebra in the First Grade. This fact made it necessary to submit the project for approval to the Ministry of Education Supervisors Board. Once the approval was given a Secondary School Inspector was regularly visit-

[4] Pupil's attainment is judged on the basis provided by the Primary School Certificate, which is the basic prerequisite that every applicant to Secondary School has to fulfil. As well, every aspirant has to pass an examination given by the school, which covers basic topics of Arithmetic and Spanish Language.

ing the Mathematics class in order to assure that the goals signalled by the Mexican Curriculum were being met. As well as this, the Principal of the school required the researcher to teach the whole course[5], in order to fulfil the Ministry of Education Supervisors Board requirements.

On the one hand, the above conditions led to design the research so that it was a relatively small part of the regular one year course (18 fifty minute sessions out of two hundred sessions) and still have enough time to cover the regular syllabus. On the other hand, these restrictions placed the researcher in a situation which helped him more closely know each pupil in the class and have a better control of the mathematical content they were confronted with during the whole school year, (before the main study the pupils were not taught any topic that relates to pre-algebra, this condition was also respected by the teachers in charge of Science and Computing).

Classroom setting

Each child in the class was given a calculator when the course began, this helped them master the machine's keyboard and computing functions. The programming mode was saved for the experimental period, which began three months after classes started.

Activities were introduced in worksheets (a total of 55) which were organised in six packages called *formats* (after Bruner's concept). This way of presenting the tasks was intended to respect (as much as possible) each child's pace, which in fact is the way in which language acquisition occurs. Nevertheless, since the research had to be carried out during a pre-established period it was expected that some children would not have time to complete the whole set of tasks. Thus, the tasks were designed so that the central features were included within the first 60% of the activities in each format. The remaining 40% (within each format) was thought of in terms of those children who work

[5] The syllabus of the First Grade of Secondary School consists of eight chapters. Three chapters deal with Arithmetic (Algorithms of the basic arithmetic operations, Divisibility, and Fractions); four chapters deal with Geometry (Triangles and quadrilaterals; Angles, parallelism and perpendicularity, Plane representation of three dimensional figures, and Perimeter, area and volume). The last chapter deals with basic notions of probability and statistics.

more rapidly. In order to do this the following working routine was set up: at the beginning of the class each child was delivered an envelope containing a format's sheets without being told how many should be completed. They returned it at the end of the class. In the next class they collected their envelopes, finding their work marked by the teacher along with the sheets they had not completed. Marking the work was aimed at providing the researcher/teacher with a current view of each child's work throughout the study. This was also intended to simulate the care-giver-child interaction and to provide the pupil with an interlocutor who could understand unorthodox expressions and help him understand why these expressions do not work in the calculator's formal language. Teacher's feedback was outlined following Bruner's findings, particularly, it was intended that the only errors recognised or responded to by the teacher should be those that he knows the child knows he can correct if challenged. Thus, the kind of feedback given to the children was characterised by asking them new questions that made the mistakes evident.

Subjects

In order to obtain data which might provide a more complete view of the effects of calculator use on children with different levels of mathematical ability, eight[6] children were chosen to be observed during the experimental phase using a case-study methodology. They were selected according to their mathematical attainment prior to the experimental phase. This was done as follows: (i) a boy and girl of below average attainment, (ii) two boys and two girls of average attainment, and, (iii) a boy and girl of above average attainment.

[6] The below average boy got sick at the middle of the study and was out of school for two months.

Tasks[7]

The use of the calculator programming code determined the activities' contents. The structure and the sequence of the tasks were defined by reinterpreting Bruner's concept of format into mathematical terms. Activities were organised into six groups called *formats*. Format 1 contains the 'raw material' on which formats 2 to 6 elaborate. In this format, expressions containing letters are introduced as the mathematical language that allows children to control the calculator. For example, running the program 2×A+1 for A=2, 5, 9 outputs the table shown in figure 3[8]. The activity is handled as a game in which pupils are given a table (in a simulation of the calculator's screen). They are then asked to:

Fig. 3

?	
2	
	5
?	
5	
	11
?	
9	
	19

- Find how the input is operated on to get the output, and express that in natural language.
- Program the calculator to reproduce the worksheet's table.
- Complete another table given with the same program.

This game contains the basic elements used to mirror Bruner's concept of format and constituted the communication platform on which increasingly complex activities were designed. As has been discussed in Chapter 3, a format is a highly routinised way of interaction between children an adult, a format is the instrument of patterned interaction that allows the child to enter into social relationships of a kind that function in the manner consonant with the uses of language in discourse. In other words, a format shapes communicative interaction between infant and caretaker before lexico-grammatical speech begins. Among other features, the concept of format was intended to mirror by keeping the children working on tasks which have the same structure, such an structure is provided by the game 'guess my rule'. Throughout Formats 1 to 5 the children confronted different mathematical activities based on the 'guess my rule' structure while their content changes (they are asked to use the calculator language to: (i) describe

[7]A sample of the tasks is shown in the Appendix .

[8]Although the calculator recognises expressions like 2A+1, the arithmetical notation formerly known by the children was respected.

number patterns, (ii) produce number patterns, (iii) produce equivalent expressions, (iv) reverse linear functions, and (v) describe whole-part relationships). This structure was intended to help children gain self confidence in using the new mathematical 'words' involved in the calculator's code and start making sense of the new formal code in-use. For example, expressions of the form $ax+b$ were like 'new words' for children; the use of these expressions imply leaving some calculations in suspense which is something that they seldom confronted when working arithmetically. This apparent routine was used to softly introduce new elements which were intended to keep children's interested in doing the tasks as they gained experience in dealing with the new code to cope with different mathematical tasks. Each worksheet included a new element, be it numerical, with a sign or decimal point, or structural, like 'two step' rules, for example 3×D is a 'one step' rule and 3×D+1 is a 'two step' rule. As was described above, the new elements incorporated throughout the six formats designed for this study included algebraic equivalence, inverting linear functions and certain problem situations.

These tasks attempt to mirror Bruner's (1982) findings that formats, grow and can become as varied and complex as necessary. Their growth is affected in several ways. The formats may in time incorporate new means or strategies for the attainment of goals (including symbolic or linguistic ones). They may move toward co-ordination of the goals of the two partners not only in the sense of agreement but also with respect to a division of labour and a division of initiative. These tasks also resemble the structure of 'speech acts' (Searle, 1969): the child knows where the game starts, its rules, where the task ends; and can check on his own whether his work is correct or not.

So far the internal structure of the mathematical formats has been discussed. Below, each format used in this study is described as is the sequence in which they were introduced. This description gives a more concrete discussion that attempts to make clear how the creation of higher order mathematical formats by incorporation of subroutine formats is used to provide teacher-pupil and pupil-calculator interactions with a powerful source of presupposition: what is incorporated becomes implicit or presupposed.

Format 1

This format consists of 15 worksheets that are aimed at introducing the use of the calculator programming code (see Appendix 1, worksheets 1-15). Here, the children were supposed to learn 'how to say' to the calculator the rules of linear functional relationships which were presented in its tabular form. In analysing the children's work attention is focused both on the role of tables as a means of encouraging children to use the new code and on children's specific achievements as clues of their development of language reception and language production skills. The time allowed to complete the format was 5 sessions of 50 minute each.

Format 2

This format consists of five worksheets. The tasks are aimed at encouraging pupils to construct a functional rule before visualising a numerical pattern, which in terms of language acquisition corresponds to 'let the child use language to get his own ends' (see Appendix 1, worksheets 16-20). The rules they constructed were used to create a table which they give as a clue for a fellow pupil to guess what program was being used to produce such a table. The time allowed to complete this format was one 50 minutes session.

Format 3

This format consists of 10 worksheets aimed at introducing the notion of equivalence between algebraic expressions. The worksheets are presented as follows: firstly, children are asked to program the calculator so that it duplicates a given table. Then pupils are required to construct at least four more programs which must display the same table (see appendix 1, worksheets 21-30). In terms of language acquisition, these tasks are aimed at helping children reach a higher linguistic level, from using tables as referents to 'talking' about tables using the algebraic code. It is intended to fulfil this aim by putting children in the position of extending their prior experience, based on reading tables and reproducing them with a calculator program, to using tables to compare and construct different algebraic expressions that produce the same table (synonymy). The time allowed to complete this format was three sessions of 50 minutes each.

Format 4

This format consisted of 10 worksheets, its content is based on finding rules of decreasing functions, which in fact introduced children to using 'new words' (expressions of the form *b-ax*). Besides the typical tasks about finding a program to reproduce a given table, in this format the child is confronted with story-based problems which can be solved by means of a calculator program (see Appendix 1, worksheets 31-40). These problems require the pupil to symbolise part-whole relationships, for example, arbitrarily cutting in two parts a piece of wire with length 16 cm, if one of these parts is called x, the other should be called $16-x$. In terms of language acquisition this format is aimed at encouraging children to start linking quantitative relationships expressed in natural language with algebraic expressions. Three classroom sessions were assigned to this format.

Since the number patterns produced by decreasing functions were completely new to the children, worksheets 31-32 include a table which might help the pupil link the problem statement with his previous experience. Problems in worksheets 33-34 require the pupil to grasp the functional variation suggested by the problem statement and relate it to the notion of a program as a functional rule.

Format 5

The time allowed to complete this format was two sessions, and it consisted of five worksheets. The tasks were aimed at introducing the notion of 'inverse programs' (inverse functions) and were delivered as follows: for a given table, pupils were asked to find a program that outputs it, then a program that outputs the inverse table (see Appendix 1, worksheets 41-45). They were also asked to find the inverse of a given program when its rule is given. In terms of language acquisition these tasks are aimed at encouraging children to start operating symbolically with algebraic expressions.

Format 6

This format consists of 10 worksheets and the time allowed to complete them was four sessions. Its aim was to observe the extent to which children can extend their experience in Formats 1-5 to facing new situations where children are required to use calculator language in negotiating problem solutions (see appendix 1, worksheets 46-55). These tasks are designed to provide the major source of data regarding the pragmatic facet of language acquisition. In other words, the tasks are aimed at obtaining a closer view of children's strategies when confronting new situations using the language they possibly acquired while describing number patterns. In order to provide an overview of this format, a succinct description of the tasks is made in what follows.

- Worksheets 46-48 deal with sequences presented by geometrical patterns. The pupils are asked to program the calculator so that it helps them to obtain any specific member of the sequence (which in fact corresponds to finding the general form of any term in the sequence).
- Worksheets 49-51 and 54-55 require the children to translate word-based problem situations into algebraic expressions (like the length is 30 meters larger than twice the width). These tasks involve procedures to calculate the perimeter or the area of rectangular shapes. For example, to negotiate a solution the pupils need to program the calculator to obtain the cost of any window frame of a whole class of rectangular shapes. More specifically, worksheet 54 focuses on calculating the length and width where a relationship between them and the perimeter are given. Worksheet 55 asks for the dimensions of a rectangular piece of land with maximum area where its perimeter is given.
- Worksheets 51-53 concern problem situations which involve the notion of percentage. These tasks also deal with translating word-based statements into algebraic expressions. For example, children are asked to program the calculator to obtain the regular and the special price of any merchandise when the discount amount is given and the bargain offers 15% off.

Data gathering

The main sources of data were the following:

a) Children's written work throughout the fieldwork.
b) Individual interviews which were carried out with the case-study children.
c) Notes taken by the researcher after each classroom session during the fieldwork addressing relevant children's interventions.

a) *Children's written work throughout the fieldwork*

a) *Children's written work throughout the fieldwork*

The content and aims of the tasks were already described (*Tasks* section). Approximately 1000 worksheets completed by the whole class were collected and marked. From these worksheets, around 350 were analysed (those completed by the case-study children), the rest of the worksheets were used eventually to observe with more detail some particular features of children's work [9].

b) Individual interviews

The major aim of the interviews was to obtain more precise and specific information about each case-study pupil and tackle some issues which were not directly addressed in the worksheets, such as algebraic transformation, and specific pupils' approaches to problem solving. The interviews were task-based sessions which lasted 50 minutes each (at most), and were centred on specific activities that closely related to the tasks the pupils had carried out during the classroom sessions[10]. As the interviewee was sorting out a task he/she was asked specific questions to explain his/her reasoning. The same questions were asked of each pupil and more specific questions for each children were prepared in advance on the basis of each child's written work and the notes taken by the researcher at the end of every classroom session. In each interview were incorporated some questions according to specific characteristics of each of the case-study subjects.

[9] Appendix 3 presents a sample of the work done by children who did not take part as case-study subjects.
[10] Appendix 2 presents the protocols used to carry out each interview.

Each of the case-study children was interviewed three times, twice during the study, and once at the end. Every interview was video recorded and transcribed. Approximately 30 hours of video taped interviews and 650 pages of interview transcripts were collected.

An outline of the mathematical issues addressed in each interview is described below.

Interview 1

This interview was carried out once the children had completed five classroom sessions programming the calculator and centred on the activities in Formats 1 and 2. The interview addressed the following aspects:

- The notions the child might have developed about letters and symbolic expressions.
- Child's use of parentheses and priority of operations.
- Child's strategies for transforming linear function rules.

Interview 2

This interview was carried out once the children had completed 12 classroom sessions programming the calculator and centred on the activities in Formats 3, 4 and 5. The interview addressed the following aspects:

- Transforming an expression to obtain another given expression.
- Simplifying linear expressions.
- Inverting a given program.

Interview 3

This interview was carried out once the children had completed 18 classroom sessions programming the calculator and centred on the activities in Format 6. The interview addressed the following issues:

- Interpreting algebraic expressions (geometrical context).
- Simplifying linear expressions.
- Inverting linear expressions.
- Children's strategies of coping with problem situations involving generality.

c) *Notes taken by the researcher*

The researcher took notes after each classroom session throughout the fieldwork. These were aimed at recording the interventions made by the teacher and pupils which could be used later during interviews or to provide further support data analysis.

When the study began it was intended to pick up children's interventions both when talking to each other and when talking to the teacher. In order to do this the teacher/researcher carried with him an audio tape recording machine while children were working. After a few sessions this tactic was abandoned. The main reason for having made this decision was that the classroom environment made it difficult to identify individual interventions and particular situations which could be used effectively within a case-study based inquiry. It was observed that in order to take advantage of tape recording during classroom sessions it was necessary to introduce new constraints which could have inhibited children's spontaneous interventions.

According to preliminary results obtained from the pilot study, the type of data derived from following the teacher's intervention was still not sufficient to thoroughly study the role of the teacher.

Organisation of data gathering

Data gathering was organised in three phases throughout the field work, each of them ending with an interview. These phases were determined by the content of the mathematical formats in which the classroom tasks were organised and are described below.

1. *Children's entry into language*, which consists of data provided by children's work in Formats 1 and the information provided by Interview 1.
2. *Children's entry into algebraic transformation*; which consists of data provided by children's work through Formats 3, 4 and 5, and Interview 2.
3. *Children's entry into problem solving*, which consists of data provided by children's work through Format 6 and Interview 3.

Chapter 5 presents a sample of children's work during all classroom sessions and interviews, the chapter consists of a chronological analysis of the work done by two case-study children. Chapter 6 presents a brief overview of the other five children that took part in the study. Chapter 7 analyses from a more general view a set of selected excerpts of the work done by all the case-study children.

Analytical framework

The analytical framework was intended to follow children's work through their development as users of calculator language. In order to do this the framework was based on the syntactic, semantic and pragmatic features of language. Though this framework allowed the researcher to anticipate and identify a good many different aspects of children's achievements, it was during the main study that the final features to be observed were more clearly defined so as to attempt to establish some regularities and put forward possible explanations. The qualitative nature of the research offered the advantage of incorporating different issues drawn from each case-study child, which provided a rich set of data (see, for example, Chapter 5), but it also made it impossible to follow a uniform scheme for describing children's work as a whole.

The analytical framework was derived from Bruner's view of the syntactic, semantic and pragmatic facets of language discussed in Chapter 3. As has been mentioned earlier, Bruner (1980) conceives the syntactic, the semantic, and the pragmatic as three great problem spaces in language acquisition. In Bruner's terms *syntax* is concerned "with the problem of how we acquire our facility in managing well-formed utterances governed roughly by a grammar". *Semantics* concerns "the nature of the relation between words and possible worlds as we know such worlds". And *pragmatics* has to do with "the manner in which we come finally to use well-formed utterances about possible worlds to affect others" (p. 156). Since this study attempts to investigate children's mathematical activity from the perspective of language acquisition, the linguistic concepts of syntax, semantics, and pragmatics were recast so that they serve to observe the children's mathematical progress through the lenses of language acquisition. These categories were used to characterise the ways in which children approached the mathematical tasks they confronted during the study. The way in which these concepts were recast is shown below.

- *Syntax*: How children acquire their facility in managing well-formed algebraic utterances governed by the formality of calculator language.

In terms of children's mathematical activity, this category was assigned to observe children's understanding of the priority of operations and the use of parentheses, and the ways in which they cope with producing algebraic expressions that conform to syntax rules.

- *Semantics*: The nature of the meanings that children develop for algebraic utterances as they use calculator language to explore number patterns, deal with algebraic transformations, and negotiate problem solutions.

This category was assigned to deal with data that provides evidence for those notions developed by children for the literal terms and algebraic expressions used when programming the calculator, and the notion of algebraic equivalence.

- *Pragmatics*: The manner in which children come finally to use well-formed algebraic utterances (produced while describing number patterns) to confront new problem situations within different contexts.

In terms of children's mathematical activity, this category was taken to analyse: (i) the strategies used by the children to 'make the calculator do the work' when confronting problem situations, and (ii) the ways in which different contexts influence children's work when negotiating problem solutions. Pragmatics also has to do with the ways in which children put into play any available notion, be it of semantic or syntactic nature, when confronting similar terms simplification, transforming a given expression to obtain a target expression, and inverting linear functions.

The thesis adopts the theoretical view that these three facets of language seem to be learned interdependently and that the three facets are inseparable in the process of acquisition. Thus, syntax, semantics and pragmatics are taken not to be derivable each from the other, but rather that each serves as a scaffold for aiding in mastery of the others (Bruner, 1980, p. 156).

Final remarks

The method used in this research consists of a qualitative analysis of the work done by the children during the field work. A central component of this method is the way in which the classroom setting was artificially arranged so that children can meet the calculator language as a language in-use. In this respect it is important to make clear that Bruner's findings on children's language acquisition have been used to implement the field work and analyse children's work, but the tasks are of a mathematical nature. So this research does not intend to confirm Bruner's theory, what is relevant to this study are those algebraic notions and strategies that children may develop through using the calculator language within the particular pragmatic approach recast from Bruner's work. As was discussed earlier in this chapter, the classroom tasks were shaped following as closely as possible Bruner's findings, which he analysed in terms of a theory of pragmatics, this is the main reason why the analytical framework is based on categories of a linguistic nature.

It is also important to take into account that these categories were applied both to report and analyse episodes from children's work, so the criteria for assigning some particular episode to some particular category necessarily bears some researcher's subjectivity. Nevertheless, during the data analysis an effort has been made to be consistent in the way in which the categories were used.

CHAPTER 5

CHRONOLOGY: VERTICAL ANALYSIS

The cases of Diego and Jenny

Introduction

This chapter presents a vertical analysis of the work done by the children within the eighteen classroom sessions and the three interviews carried out during the field work. This vertical analysis is intended to demonstrate the ways in which children's acquisition of "calculator language" evolved in time, which, due to the nature of this research, fulfils a fundamental part of the study. As will be seen throughout the chapter, a single case-study file includes a huge amount of data, so in order to show a chronological view of children's work only two of the seven case-study files available will be analysed. These files were chosen to "represent" an "average view" of the seven case-study children. That is, the selected files analysed in the chapter are of those children who did not reach either the highest or the lowest achievement during the field work.

One of these files presents the analysis of the work done by a boy (Diego, 11 years old). Diego was considered to be within the "average strand" of the whole class according to his mathematical attainment during the first months of the course. The other file corresponds to the case of a girl (Jenny, 12 years old), who in terms of the whole class was considered to be within the "above average strand".

These two files allow us to observe in detail the different ways in which children may approach the same learning situations, and the ways in which the specific calculator-based environment used in this study helped them build on the basis of their particular approaches and progressively refine their strategies and notions so as to successfully confront the most complex tasks delivered to them during the study. Jennifer was able to correctly complete all the tasks included in the study and Diego missed some of them. Nevertheless, from a qualitative point of view it is difficult to establish big differences between these two children in terms of the effectiveness of their responses to mathematical situations. As will be seen throughout the chapter, the differences between these children rather corresponds to different styles of approaching the tasks.

The chapter consists of two major sections, respectively: Diego's case, and Jennifer's case. Each of these sections is split into three phases, which respectively correspond to the algebraic topics around which the tasks were arranged (as was described in Chapter 4): Phase 1, Children's entry into Calculator Language, provides an analysis of a child's work in formats 1 and 2, and in the first interview; Phase 2, Children's entry into Algebraic Manipulation, provides an analysis of a child's work throughout formats 3, 4 and 5 and in the second interview; Phase 3, Children's entry into Problem Solving, provides an analysis of the work done by the child in Format 6 and interview 3. Finally, each case-study file is closed by a more general discussion which is intended to provide a background for the general results of this study that are discussed in Chapter 7.

5.1. THE CASE OF DIEGO

PHASE 1: Diego's Entry Into Calculator's Language.

Phase 1 is organised as follows: First Diego's written work is analysed, then a summarised transcription of his work is presented intending to give further support for the preceding analysis; finally Diego's work during Interview 1 is analysed.

FORMAT 1: Discussion of Diego's work

Semantics: Diego's notion of literal terms and algebraic expressions.

Diego was interested in doing the tasks and completed 12 of the 15 worksheets included in Format 1 (the time allowed was five sessions). Diego's work indicates that he is beginning to grasp how to read a table and to express it as a function rule (calculator program).

It is important to notice the way in which Diego refers to the function rule using natural language. For example, "*I multiplied by 2 and added 1*" which does not explicitly involve either the variable or the number on which he performs these operations. Nevertheless, he can express correctly the rule using calculator language (see, for example, worksheet 4). The way Diego is learning to formalise his methods seems to be strongly influenced by the calculator environment. The research data suggests that work with the calculator leads Diego to think of the tasks "operationally", that is, thinking of what computations have to be done to the input in order to obtain the given output. This seems to be the reason why Diego correctly expressed the relevant relationships algebraically, though he was not able to do it verbally. Diego's responses suggest that a teaching approach which advises children to first verbally describe the relevant relationships, making explicit reference to the number or variable they are operating on, before expressing symbolically their method, (e.g. I multiplied the number by 2, then added 1 to it) needs requestioning. This teaching approach is currently used in Mexico and probably in some other countries (see for example; Mason's (1980) recommendations and MacGregor & Stacey, (1993 and 1996) reports). It seems that this approach is induced by the constraints imposed by the paper and pencil environment, where natural language is the most available tool for expressing mathematical relationships. Diego's work suggests that the use of calculator language provides another means of expressing such relationships, if the child does not respect the calculator's constraints the algebraic expressions they produce simply do not work. Accordingly, Diego's work within a paper and pencil environment was observed during the rest of the study.

Pragmatics: Diego's approach to inverting linear functions

From the very beginning the tasks in Format 1 confronted the children with three types of questions. The first consisted of uncovering the underlying number pattern that governs a set of numbers presented in a table. The second question asked the child to describe that pattern, using calculator language in order to construct a program. The third question involved a new task, which required the child to sort out how to use that program to find the numbers input when the outputs were given. This question is aimed at making the child reflect on the sense of the algebraic expression he/she has produced. From now on these tasks will be called "producing inverse values".

In this respect, Diego's work showed that, particularly at the beginning, teacher's feedback is crucial for Diego, because, when he runs a program for values which he has not already generated, he usually does not anticipate which results are going to be obtained. He just does what is required to: run the program and fill in the blanks. Thus, the task may result in a blind computing procedure. Diego's work also indicates that producing algebra-like expressions appears not enough for him to gain awareness of their general nature. It seems necessary that the child both **produces** and **uses** the expression. The following extract illustrates this:

Diego had some trouble in producing inverse values, particularly when the expression involved was of the form $ax+b$. Once the teacher had given Diego feedback in the form of marked work he managed to correct and find a way to sort out the remaining tasks. Teacher's feedback consisted of marking the errors and writing down a note like "the program you made does not give these results, check it back". This helped him fill in the blanks correctly in worksheet 4, where A×2+1 was the involved rule (Diego's answers in bold).

Input	1.3	2.8	14	50	81	274	**1st attempt: 161.5** **2nd attempt: 162**	**1st attempt: 209** **2nd attempt: 209.5**
Output	**3.6**	**6.6**	**29**	**101**	**163**	**549**	325	420

Let us look at the case of 325, at the first attempt he produced 161.5 by dividing 325 by 2, 162.5, then he took 1 away and got 161.5. He did the same in the case of 420, which shows a trend of inverting operations following the order in which they appear (×, +, then ÷, −). This allows us to see that he has grasped the inverse role played by these numbers in the table, but he has not yet realised that he might have used the program A×2+1 as a source of feedback, which suggests a lack of awareness of the relationship between the arithmetic procedure executed by the program and the procedure he used to find the inverse values. In fact, Diego could answer the item by guessing with the program A×2+1 until he got the desired numbers.

Pragmatics: Diego's approach to negative numbers

Operating with negative numbers was new for all the children at the beginning of the calculator sessions. Previous to this only those aspects about order and the use of negative numbers were treated in representing magnitudes like temperature, income and debts, altitude with respect to the sea level, etc. Worksheets 6 and 7 includes tables whose inputs are decimal negative numbers, which imply operating with such numbers to find the functional rules that generate these tables (see Diego's work summary). Diego found these rules by exploring with the calculator how to operate with negative numbers and was able to work out correctly the two worksheets. It seems relevant that Diego resorted to using the negative sign twice (i.e. A– - 5, worksheet 6) which required him to distinguish between the minus sign for operating and the minus sign for denoting a negative number. He also combined the adding and subtracting signs (A+–1.5, worksheet 7). This way of working provided evidence of Diego's level of engagement with the calculator's modes of computing, and how this experience enabled him to get support from the machine.

FORMAT 1: Summary of Diego's work

WS[1]	Clues given						Rule written in natural language[2]	Programming expression produced	Completing the table
1.	In	1	2	3	4	5	*"I added 4"*	?→A: A+4	He had some trouble in completing the blanks corresponding to inverse values.
	Out	5	6	7	8	9			
2.	In	7	8	9	15	18	*"I multiplied by 2"*	?→A: A×2	OK even with decimal numbers.
	Out	14	16	18	30	36			
3.	In	2.5	3.1	4	4.2	5.3	A×3	?→A: A×3 ?→A: A÷3	OK even with decimal numbers.
	Out	7.5	9.3	12	12.6				
4.	In	1.1	2.5	3	4.3	5	*"I multiplied by 2 and added 1"*	?→A: A×2+1 Diego was given a hint: combine two operations, as multiplying and subtracting, dividing and adding, etc.	He had some trouble in completing the blanks corresponding to inverse values.
	Out	3.2	6	7	9.6	11			
5.	In	1	2	3	4	5	*"I multiplied by 2 and took away 1"*	?→A: A×2–1	OK
	Out	1	3	5	7	9			
6.	In	-10	-9.7	-7.8	-6.2	-5.3	Not required	?→A: A– – 0.5	OK.
	Out	-9.5	-9.2	-7.3	-5.7				
7.	In	-15	-14.5	-12.4	-10.2	-5.8	*"I added minus 1.5"*	?→A: A+–1.5	OK, including inverse values involving negative numbers.
	Out	-16.5	-16	-13.9	-11.7				

[1]Worksheets 11-15 were given only to those children who completed the work more rapidly.
[2] The tasks required the child to express the rules using "their own words".

WS[3]		Clues given					Rule written in natural language[4]	Programming expression produced	Completing the table
8.	In	10.5	14.42	15.3	16.7	20.1	*"I divided by 2"*	?→A: A÷2	Not required.
	Out	5.25	7.21	7.65	8.35	10.05			
9.	In	6	8	14	15	18	*"I divided by 2 and multiplied by 3"*	?→A: A÷2×3	OK, including calculating inverse values.
	Out	9	12	21	22.5	27			
10.	In	4	6	9	10	12	*"To multiply by 1.01"*	?→A: A×1.01	OK, including calculating inverse values.
	Out	4.04	6.06	9.09	10.1	12.12			
11.	In	7	9	10	12	16	*"I multiplied by 3 and added 2"*	?→A: A×3+2	OK, including calculating inverse values.
	Out	23	29	32	38	50			
12.	In	7	7.5	8.2	9	9.6	*"I multiplied by 3 and took away 1"*	?→A: A×3−1	OK, it was included calculating inverse values involving decimal numbers.
	Out	20	21.5	23.6	26	27.8			
13.	In	10	15	20	25	30			He did not have time to do this worksheet.
	Out	2.5	3.75	4	6.25	7.5			
14.	In	2	3	4	5				He did not have time to do this worksheet.
	Out	5	7.5	10	12.5				
15.	In	0.15	0.27	0.3	1.5	2.03			He did not have time to do this worksheet.
	Out	0.015	0.027	0.03	0.15	0.203			

FORMAT 2: Discussion of Diego's work

Semantics: Diego's notion of literal terms and algebraic expressions.

In four of the five worksheets Diego constructed programming expressions based on combining two arithmetic operations. Since the kind of expressions to be used was Diego's choice, this suggests that Diego was able to give sense to this type of expressions.

Syntax: Diego's notions of parentheses and priority of operations.

Diego's work highlights the value of allowing children to use the calculator language to get their own goals. In order to do this, the child may create a number-based generic example to produce a general procedure, or he may just construct an algebraic expression and see how it works. Diego chose the former strategy, and due to his lack of awareness of priority of operations he first made a different table from the one he finally gave to his fellow partner (see tables below). Anticipating a general relationship gave him a method for verifying the correctness of the expression he had built.

[3]Worksheets 11-15 were given only to those children who completed the work more rapidly.

Program that Diego used to build the table
? →A: A+–5×7

Table obtained by mental calculation

Input	1	3	5	8	10	20
Output	–28	–14	0	21	35	105

Table obtained using the calculator

Input	1	3	5	8	10	20
Output	–34	–32	–30	–27	–25	–15

Diego did not expect the outputs he obtained with the program A+–5×7 because he was still operating from left to right. For example, he expected 10+–5×7=35 instead of –25. This made him inquire why he got such different results. At the time of delivering the task to a partner Diego had already learnt how to explain it in terms of order of arithmetic operations: "*it first does –5×7, –35, so it always takes 35 away from the number you input*" (the researcher was the fellow he was working with). Nevertheless, data from interview 2 shows that this experience was not enough for him to be always aware of the role of order of operations.

FORMAT 2: Summary of Diego's work

Work sheet	Program that Diego used to build the table	Clues given by Diego						
16.	? →A: A×5÷2	Input	1	3	5	8	10	20
		Output	2.5	7.5	12.5	20	25	50
17.	? →A: A×9+1.5	Input	1	3	5	8	10	20
		Output	10.5	28.5	46.5	73.5	91.5	181.5
18.	? →A: A×13.1	Input	1	3	5	8	10	20
		Output	13.1	39.3	65.5	104.8	131	262
19.	? →A: A×5.3+1	Input	1	3	5	8	10	20
		Output	6.3	16.9	27.5	43.4	54	107
20.	? →A: A+–5×7	Input	1	3	5	8	10	20
		Output	–34	–32	–30	–27	–25	–15

INTERVIEW 1: Discussion of Diego's work

Summary

At the date the interview was given, Diego has been engaged in programming tasks during five classroom sessions (four in Format 1, one in Format 2). This interview was focused on observing the following aspects:

[4] The tasks required the child to express the rules using "their own words".

- The notions the child might have developed about letters and symbolic expressions.
- Child's use of parentheses and priority of operations.
- Child's strategies for transforming linear function rules. This was aimed at observing whether the programming experience may have helped the child in confronting symbolic manipulation with algebraic expressions (this task is placed into calculator context, for example, transforming 4×A in order to make it equivalent to 3×A). These points are discussed in what follows.

Semantics: Diego's notion of literal terms and algebraic expressions.

Diego's responses indicate that his experience using the calculator has led him to develop the notion of letters as symbols that represent a range of numbers. Diego has also grasped that letters can be chosen arbitrarily, that what matters is the structure of the expression that embodies the literal symbol. The following extracts illustrate this process.

Being asked what the letter he used in a program meant to him, Diego answered: "*the letter personifies the number I want to make the program with ... they personify any number, once you put the letter you can input any quantity ... you can run the program for any number you want ... the output changes depending on the number you put in*" (I1: 93-96)[5]. He has also grasped that a programming expression does not depend on the letter he uses. For instance, he had written the program A×5÷2, and was asked what would happened if someone else wrote M×5÷2. He answered: "*A×5÷2 does the same as M×5÷2 because the number I am operating with may take the form of any other letter*" (I1: 15-17).

It is also important to notice that the way in which Diego links literal symbols with numbers allowed him to successfully deal with tasks about symbolic manipulation.

Syntax: Diego's notions of parentheses and priority of operations.

Diego gained awareness of the priority of operations once he realised that the calculator did not give the results he was expecting to get. As will be discussed in Jenny's case, it was crucial that Diego was able to make the computations in advance, otherwise he would not have realised anything from the outputs produced by the calculator. The following extract illustrates this.

Diego was asked to program the calculator so that it first takes 1 away, then it multiplies the outcome by 3. He showed that he understood the question and mentally tried a few specific examples: "*if I put 7 ... 6 by 3, 18 ... If I put 11, 10 by 3, 30*" (I1: 19-22). However, Diego typed the program A–1×3. He showed surprise when it produced 1 after having run the program for A=4, "*because I expected it to output 9*". Being asked why the program produced this, he ran it for a few values and found that "*it is taking away 3*" and explained that "*the program first makes 1×3 then it takes this away*". After several failed trials where he did not show any sign of knowing about parentheses he was told by the teacher/researcher how to get the desired program by using them. (I1: 23-64). What is relevant from this episode is how he grasped the idea of what parentheses are used for. From then on he used them as often as he could in answering the remaining questions in this interview (see the section about transforming linear function rules).

This episode also suggests that for children to grasp the conventions of syntax it is necessary to make evident their purpose in terms of being more suitable means than their previous available resources.

Semantics: Diego's notion of algebraic equivalence

This topic was presented immediately after Diego had faced the question of using parentheses. It seems that Diego's fresh knowledge about parentheses allowed him to face items on transforming algebraic expressions. The specific question was: I wanted to type the program 11×B but unwittingly I typed 10×B. Can you correct it without deleting anything of what I have already typed? Diego's first attempt was to make the program 10+1×B, he followed the rule of not deleting, he inserted **+1**. Without needing to run the program Diego realised that it would not work, because "*the calculator multiplies B by 1 first, then adds 10*". Then he inserted brackets: (10+1)×B (I1: 66-69).

[5] The code '*In: a-b*', is meant to denote 'Interview *n*, interventions *a* to *b*'.

Diego's number-based strategies helped him use his notion of letters as symbols that represent a range of numbers to develop the notion of letters as manipulative entities. For example, Diego was asked to transform the program 10×B so that it produces the same as 7×B without either deleting or inserting. He immediately said "*again using brackets*" but the imposed restriction of not inserting made him abandon this idea. He kept thinking for a few moments and finally typed the program 10×B–3×B. He then ran it for a few values to check it back (he was really happy verifying his finding). Upon being asked, he explained: "*The program had to multiply by three numbers less than ten, so we have to decrease B three times when we multiply*" (I1: 72-77).

After this Diego was required to do it using brackets: he typed (10–3)×B. Then he was told that there were now three different programs that he says are the same: 7×B, 10×B–3×B and (10–3)×B. He asserted "*they would all be the same*". Then he was asked to choose from a list those programs he thought could be equivalent to 7×B. The list was the following: 11×B–4xB; (14–6)×B; 9×B–2×B; 6×B+1; 6×B+B and 6+1×B. He chose them correctly (he explained he had done it by mentally substituting values for variables) (I1: 80-83).

PHASE 2: Diego's entry into algebraic manipulation

FORMAT 3: Discussion of Diego's work

Diego completed correctly the 10 worksheets included in Format 3. Nevertheless, it is relevant how he put into play the new tools he had just dealt with three days before (interview 2), such as operating with the literal terms and using parentheses, see, for example, worksheets 22, 26 and 28.

Semantics: Diego's notion of algebraic equivalence

Diego's approach to algebraic equivalence was hugely aided by his syntactic notions as will be discussed next. Diego resorted to using parentheses to operate with the coefficients of an algebraic expression in order to obtain a new equivalent expression. For example, in worksheet 22 he built B×(1+0.5) to be equivalent to B×1.5 (see also worksheets 23-26). Although Diego's way of working shows his methods to be tightly linked to numerical computing it indicates as well that he is grasping what parentheses are used for.

It seems worth noticing that Diego operated with the variable only when coefficients were integer numbers and where multiplication was involved (worksheets 21, and 28). Where division or fractional coefficients were included he operated on the independent term (worksheets 22-27 and 29-30). Diego's apparent limitation in manipulating algebraic terms relates to his approach to transforming algebraic expressions, which consists of exploring the behaviour of an algebraic expression through number-based generic examples (similarly to Jenny's case[6]). This strategy strongly relies on mental computing which is good as long as the target expression involves numbers and operations he can mentally operate with, that is, when he can anticipate the outcomes by himself. For example, in worksheet 21 he set B×2+B×2, B×5–B and B×3+B as equivalent programs to B×4, while in worksheet 22 he worked differently, there he set B×1×1.5, B×(1+0.5), B×(2–0.5) and 1.5×B×1 to be equivalent to B×1.5.

It is also interesting how Diego extended his strategy of adding zero and multiplying by 1 to the algebraic case. For example, in worksheet 27 he built B×1+1+1–2 to be equivalent to B÷1. In worksheet 28 he wrote B×B+B–B and B×B+B–B+B^2–B×B as equivalent expressions to B^2. It is this strategy which can explain how he performed more difficult algebraic transformations later on (see interview 2).

Semantics: Diego's notion of literal terms and algebraic expressions.

When working in Format 3 Diego made a shift from describing his method using natural written language to describing it by means of the language of calculator expressions. For instance, In Format 1 his descriptions were of the type "I multiplied by 4", while in Format 3 they were of the type "B×4". This shift suggests Diego's progress where translation from natural language to algebraic code is concerned.

FORMAT 3: Summary of Diego's work

Work sheet	Given table	Rule expressed in natural language	Programming expression produced	Equivalent expressions produced.
21.	In 1 1.5 3 . 5 Out 4 6 12 20	D×4	?→B: B×4	?→B: B×1×4 ?→B: B×2+B×2 ?→B: B×5−B ?→B: B×3+B
22.	In 2 4 8 10 Out 3 6 12 14	B×1.5	?→B: B×1.5	?→B: B×1×1.5 ?→B: B×(1+0.5) ?→B: B×(2−0.5) ?→B: 1.5×B×1
Work sheet	**Given table**	**Rule expressed in natural language**	**Programming expression produced**	**Equivalent expressions produced.**
23.	In 1 2 3 4 Out 0.25 0.5 0.75 1	B÷4	?→B: B÷4	?→B: B÷4×1 ?→B: B÷(3+1) ?→B: B÷(5−1) ?→B: B÷(356−352)
24.	In -1 3 7.4 17 Out -0.5 1.5 3.7 8.5	Not required	?→B: B÷2	?→B: B÷1×2 ?→B: B÷1÷2 ?→B: B×2÷2÷2 ?→B: B÷(9−7)
25.		CANCELLED		
26.	In 1 3 5 9 Out 6 10 14 22	Not required	?→B: B×2+4	?→B: B×2+2+2 ?→B: B×2×2÷2+4 ?→B: B×2÷2×2+2+2 ?→B: (B+4÷2)×2
27.	In 15 16 17 18 Out 15 16 17 18	Not required	?→B: B÷1	?→B: B×0+B ?→B: B+1−1 ?→B: B×1+1−1 ?→B: B÷2×2−4−1+3+2
28.	In 1 3.2 5 9 Out 1 10.24 25 81	Not required	?→B: B^2	?→B: B×B ?→B: B×B+B−B ?→B: B×B+B−B+B^2−B×B
29.	Here, the program ?→N: 3.5×N was provided instead of giving a table.		?→N: N×3.5×1	?→N: N×7÷2 ?→N: N×14÷2÷2 ?→N: N×10.5÷3 ?→N: N×3.5÷2×2
30.	The program 1.02×Z was provided instead of a table.	Not required	?→A: A×1.02×1	**?→A: A×1+0.02** ?→A: A÷2×2×1.02

FORMAT 4: Discussion of Diego's work

In this format Diego just completed 5 out of the 10 tasks which reflects the difficulty he had in coping with a new kind of number pattern. Diego's reactions suggest that expressing rules of the form *b-ax* goes beyond that of just describing a pattern and somehow implies operating on the algebraic term. This conjecture seems to be supported by the fact that Diego confronted the task of recognising the underlying number pattern from a numerical exploration, that is, from finding what operations should be done with

[6] See section 5.2 in this chapter.

the input so as to obtain the output. This approach appears to add an important feature to symbolic manipulation, for in the case of describing functions of the form b-ax, the child not only is encouraged to operate with the as yet unknown but also he has to keep in suspense the as yet unknown, for example, he has first to find out the constant "b" and then subtract "ax" from this. This point is further discussed in the next paragraph.

Semantics: Diego's notion of literal terms and algebraic expressions.

In worksheet 32 Diego described the operations he made first verbally as "I took **A** away from 20" then, in the same row, as 20–A. Since the way of expressing the rule is the child's choice, this suggests that he is in the process of adopting algebraic language as a suitable means to describe quantitative relationships. From another perspective, Diego used expressions like 20–A to describe his method which seems to be related to the issue of acceptance of unclosed algebraic expressions.

Syntax: Diego's notion of using parentheses

Contrasting with what Diego did in Formats 3 and 5, he did not use parentheses in this format. Worksheet 34 requires their use but he could not complete it. This suggests that children's learning of syntactic conventions consists of a lengthy process during which children need to confront different uses of the calculator language within different contexts. This point is taken up again in the last section of Diego's case.

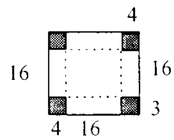
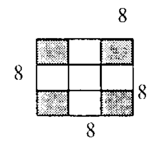
Pragmatics: Diego's approach to negative numbers

Diego correctly completed worksheet 35 which involves operating with negative numbers to find the function rule (1–A). His written explanation indicates that he is at the stage of anticipating results of this kind without using the calculator: "*I took the number I want away from 1, I knew this because of the negative outputs in the table*". What he meant by saying "the number I want" is the input number in a program.

Pragmatics: Diego's use of algebraic language to negotiate problem solutions.

In this format the children met for the first time the use of calculator language to negotiate problem solutions. Diego was able to algebraically represent quantitative relationships involved in three out of four story-based problems (worksheets 31-33). He could not make sense of the problem posed in worksheet 34 (the square box of maximum volume).

FORMAT 4: Summary of Diego's work.

WS	Worksheet content	Expression produced
31.	My grand father owns a hardware store. In helping him I programmed my calculator so that every time that some amount of wire is sold the program tells you how much wire is left. The table below is an example of how my program works. Can you guess what is it? **Sold** 1.7 2.4 3.1 4.06 5.2 **Left** 8.3 7.6 6.9 5.94 4.8	Answered correctly ?→A: 5+5−A
32.	**32.1.** Can you program the calculator so that it produces the following table? **Input** 1.3 2.5 3.8 4.4 5.9 **Output** 18.7 17.5 16.2 15.6 14.1 **32.2.** What does it happen when you input a negative number?	Answered correctly ?→A: 20−A *"I took away A from 20"* *"Instead of taking A away from 20 the program adds 20"*
33.	I have some pieces of wire, all are of length 16 cm. I want to cut them all into two pieces in different ways, for example, 12 cm and 4 cm, 11 cm and 5 cm, and so on. Can you program the calculator so that if I input the length of one small piece it prints out the length of the other one?	Answered correctly ?→A: 16−A *"Because in earlier worksheets I was taking away from numbers and I did the same here"*
34.	I want to make a box with a square piece of cardboard. I can make the box by cutting squares off the corners and bending up the pieces that are left jutting out. The **base** and **height** of the box, are determined by the length of the sides of the squares I cut off. Figures 1 and 2 show two possible ways of making the box. Can you program your calculator so that it allows to calculate the volume of any box I could build?  	Did not complete this worksheet.
	Program your calculator so that it duplicates the table below. **Input** 1 2 3 4 5 **Output** 0 −1 −2 −3 −4	Answered correctly ?→A: 1−A *"I took the number I want away from 1. I knew this because of the negative outputs in the table"*
35.	Program your calculator so that it produces the table below. **Input** 1 2 3 4 5 **Output** 4 9 14 19 24	Answered correctly ?→A: A×5−1 *"Because when I multiplied by 5 it always got one number less than the product"*.
36.	Program your calculator so that it produces the table below. **Input** 1 2 3 4 5 **Output** 0.5 −0.5 −1.5 −2.5 −3.5	Did not complete this worksheet.
37.	Program your calculator so that it produces the table below. **Input** 1 2 3 4 5 **Output** 8.5 6.5 4.5 2.5 0.5	Did not complete this worksheet.
38.	Program your calculator so that it duplicates the table below. **Input** 1 2 3 4 5 **Output** 0 0 0 0 0	Did not complete this worksheet.
39.	Program your calculator so that it produces the table below. **Input** 1 2 3 4 5 **Output** −1 −2 −3 −4 −5	Did not complete this worksheet.

FORMAT 5: Discussion of Diego's work

Pragmatics: Diego's approach to inverting linear functions

Diego completed four out of five worksheets. His strategy for inverting rules of the form f(x)=ax+b consisted of reversing operations in the order in which they were written. After doing so, he adjusted the expression by adding or subtracting a constant to obtain the desired behaviour of the expression. For example, he inverted A×2−1 as (A÷2)+1−0.5 (worksheet 44). In the same worksheet he could not invert the program B×3+1, quite probably due to the complexity of the expression, which makes it more difficult to adjust B÷3−1 by adding a constant.

Syntax: Diego's notion of using of parentheses

Diego's work shows that he is not yet at the stage of discriminating when parentheses are strictly necessary (see for example worksheet 44). Besides, he could not extend his notion of using parentheses so as to apply then to get the inverse rule of functions of the form f(x)=ax+b.

FORMAT 5: Summary of Diego's work

WS	Content	Program produced	Inverse program
41.	Input 10.4 16 19 23.5 37 Output 4.9 10.5 13.5 18 31.5	?→A: A+5.5	?→A: A−5.5
42.	Input 11.4 19 23.1 38 50 Output 17.5 25.1 29.2 44.1 56.1	?→A: A−6.1	?→A: A+6.1
43.	43.1 Input 0.13 0.17 0.65 3.8 9.28 Output 0.26 0.34 1.3 7.6 18.56 43.2.Program your calculator so that it produces the inverse of M×3. 43.3.Program your calculator so that it produces the inverse of N×1.5.	?→A: A×2	?→A: A÷2 ?→M: M÷3 ?→N: N÷1.5
44.	44.1. Input 3 7 10 11 15 Output 5 13 19 21 29 44.2.Invent a program so that it "undoes" the one you have just found. 44.3.Can you type a program so that it undoes the program B×3+1?	?→A: A×2−1	?→A: (A÷2)+1−0.5 Did not complete it.
45.	45.1. Input 2 5 7 8 10 Output 4 25 49 64 100 45.2.Invent a program so that it "undoes" the one you have just found. 45.3.For the following programs construct one which undoes each of them. ?→A: A×1.5+1 ?→K: 0.5×K−1 ?→X: 0.25×X+2 45.4.Did you find a method to undo programs? Say what it consists of.	Did not complete it. Did not complete it Did not complete it Did not complete it	

INTERVIEW 2: Discussion of Diego's work

The interview was carried out around the following features[7]:

- Transforming an expression in order to obtain another given expression.
- Simplifying linear expressions
- Inverting linear functions.

Semantics: Diego's notion of literal terms and algebraic expressions.

Diego's strategy of transforming algebraic expressions shows that he is not seeing letters as entities on which he can operate, he rather conceives them as symbols that represent a range of numbers. Diego's responses suggest that his most basic strategy relies on conceiving algebraic expressions as instruments that serve to describe and perform arithmetical procedures. This notion enables him to proceed from the general to the particular and vice versa. This appears to be the main reason why he is able to cope with algebraic transformations based on specific cases without losing the general nature of the expressions. Here, both the exploratory background and the explanatory referent are provided by basic arithmetical facts, which strongly contrast with the way of proceeding within a formal algebraic transformation approach, where validity exclusively relies on correct application of syntax rules. The following extract illustrates this claim.

Diego was asked to do something with A^3 so that it produces the same as A^2. His first attempt was $A^3–A$, "*because it is like having A multiplied three times by itself so I have to delete an A*". He silently ran the program for a couple of values and quickly changed it to $A^3–2\times A$. In this case he did not run the program, he kept watching the expression for a few moments and said: "*It does not work ... I was only thinking of 2 ... because 2^3 gives 8, minus 4, (2×2), gives 4 ... the same as A^2*". He then was asked why he thought this was wrong, he explained "*because it* (the program) *has to work with any other number*" (I2: 53-67). Although Diego was able to see A^3 as A×A×A he could not think of dividing by A to get A^2. He finally found a way to do it. He inserted parentheses to write $A^{(3-1)}$ (I2: 68-78).

[7] It was planned to ask questions about simplifying similar terms within linear expressions but there was no time to do this. These questions were included in Interview 3.

The above episode deserves further analysis. Diego's reasoning about transforming A^3 to make it equivalent to A^2 was "to delete an A". He formalised this idea as A^3-A, which is pure symbolic manipulation based on his failed interpretation of "deleting" as "taking away" instead of "dividing". The important point here is that, once he did this, he checked it back numerically which allowed him to realise his mistake. After this his trials were all number-based which, although they did not produce the expected answer ($A^3 \div A$), finally led him to successfully carry out the question. This episode shows Diego's first steps in symbolic manipulation; it should be noticed that his strategy consists of exploring on the basis of an initial conjecture he made, which is quite different from blind guessing. Once he could not go ahead with his first idea he abandoned it, and looked for another strategy being guided by the numerical behaviour of the algebraic expression.

The episode also shows that Diego's notion of programming expressions being like devices to calculate and express arithmetic procedures enables him to cope with questions about algebraic expressions. In other words, Diego's previous experience of programming the calculator seems to help him in making sense of symbolic manipulation and refining his conjectures.

Syntax: Diego's notions of parentheses and priority of operations

Diego's approach to determining if two algebraic expressions are equivalent allows us to see the crucial role that priority of arithmetical operations plays in helping him make sense of the question and in breaking down an algebraic expression into its simplest terms. The following episode illustrates this.

Diego was asked to compare 7×B with 11×B−4×B. After some mental calculation he found that "*they are the same because here (*pointing at 11×B*) you are multiplying by 11, but we want 7, so you need to take 4 times this away, this would give 7×B*" (I2: 33-34). After this Diego was given an algebraic expression to be rewritten without parentheses when parentheses were used and vice versa. The expression I gave to him was C×(5−4), he easily rewrote it as C×1. Being asked for another way to do it, after an overt struggle with mental calculation, Diego produced C×5–C×4 and explained: "*I thought of C×(5–4) without parentheses ... I mean C×5–4, if C was 5, C×5 would give us 25, 25–4 doesn't give the same as C×1, I needed minus 20 ... it is C×4 ... it works*" (I2: 39-45). Diego could not have thought of 5×5–5×4 as a specific case of C×5–C×4 without being aware of priority of operations. It is also worth observing that by establishing the equivalence C×(5–4)=C×5–C×4, he is somehow applying the distributive law as a "theorem in action" (in Vergnaud's sense)

Semantics: Diego's notion of algebraic equivalence

Diego's approach to algebraic manipulation seems to be strongly supported by his syntactic notions. As will be seen here, Diego's command on using parenthesis combined with the strategy of exploring the numerical behaviour of an algebraic expression led him to learn new features about algebraic manipulation. Diego confronted algebraic transformation resorting to simplifying similar terms. As will be shown below this feature of Diego's work emphasises the role of the calculator as an environment that encourages children to move back and forth between the particular and the general.

Diego was asked to transform A^3, to make it equivalent to A^2 without using parentheses (he had already done it as $A^{(3-1)}$). The interviewer made him notice that A^3=A×A×A, trying to suggest that he could be doing A^3÷A. Diego seemed simply not to have heard any of this and kept following his own line of reasoning. His first reaction was to type the program A^3–A^2+A^2: "*I input 2 ... it only raises it to cube*" (I2:79-83). Diego mentally did some calculations and said: "*I got it! When I put this it deleted it*" (he meant –A^2+A^2). Immediately he retyped the program as A^3–A^3+A^2 (I2:84-90). It is worth recalling that Diego had used this strategy before (adding zero, Format 3). This shows how his strategies evolved. First, he resorted to adding zero in the numerical case, he then extended it to the algebraic case. For example, B^2=B×B+B–B+B^2–B×B (worksheet 28). Finally he extended this strategy to "reducing to zero", which is the way he used to set A^3 = A^3–A^3+A^2. The episode shows that his strategies are context dependent, but also shows that with time Diego can extend his findings to different contexts.

The above extract indicates that making Diego observe that $A^3=A\times A\times A$ meant nothing to him because he was engaged with the idea of deleting as "taking away". Thus, he explored again using symbolic manipulation and got $A^3-A^2+A^2$. It was not until he ran the program and contrasted it numerically that he abandoned his initial trial. This going back to numbers was what enabled him to make sense of the whole expression ("*When I put this* ($-A^2+A^2$) *it deleted it*"). Then he used his new finding to type $A^3-A^3+A^2$, moving from a specific algebraic case ($-A^2+A^2=0$) to general symbolic manipulation.

As in interview 1, Diego showed that he is more confident in multiplying than he is in dividing. When multiplication is involved he does not appear to have any difficulty in transforming algebraic expressions. This emphasises the importance of children's arithmetical background when the teaching of algebra is based upon arithmetic manipulation. What follows provides evidence for this point.

Diego was asked to transform A^2 so that it gave the same as A^3. Without any hesitation he answered "*A^2 times A*". Next he was asked to transform A^2 so that it gave the same as A^4, he immediately wrote down $A^2\times A\times A$ and commented: "*It is easy if I want a higher power, I don't know how to do it when the power is lower*" (I2:93-108).

Pragmatics: Diego's approach to inverting linear functions

The interview allows us to see that priority of operations is a topic that requires the pupil to work on for a long period before he abandons his previous left to right way of operating. Diego's reactions also suggest that teacher intervention is needed for him to use parentheses in inverting linear expressions. As has been shown earlier, his spontaneous tendency to invert function rules consists of reversing the arithmetic operations in the order in which they appear disregarding their priority. So far, Diego's responses make evident that his experience in programming the calculator has not been enough for him to link the notion of parentheses as devices that serve to break down the operating order imposed by the calculator with the notion of inverting as "doing first what was the last and vice versa". The following extract illustrates this.

Diego was asked what the program A×2−1 does. He said "*it takes 1 away from 2 then it multiplies this by something*". Then he was told to run it; without doing it he said "*no, it first multiply by 2, then takes 1 away*" (I2: 111-114). His response indicates that he is still at the stage of getting confused between his old habit of calculating from left to right and the new notion of priority of operations.

Once he answered the above question correctly he was asked to invert the program. He said "*everything all the way round*", built the program A÷2+1 and ran it. Then he realised that if he input 3 the program gave 2.5, then he input 9 and got 5.5, which differs from the result he was expecting (A×2−1 would give 3 if A=2, if A=5 would give 9). He became aware of this and adjusted the program by taking away 0.5 (A÷2+1−0.5) and said: "*now if I input 3 it will give 2, and so on ... that's it*" (I2:116-121). His answer shows that he is fully aware of the inverse nature of these programs but this is not enough to make him realise their inverse structure.

PHASE 3: Diego's entry into Problem Solving

FORMAT 6: Discussion of Diego's work

This format consists of 10 worksheets and its aim was to observe the extent to which children can extend their experience in Formats 1-5 to facing new situations which require them to use the calculator language to negotiate problem solutions. Diego completed seven of the ten worksheets included in this format

Pragmatics: The role of the context

The work done by Diego shows that he can more easily confront problem situations whose wording involves number patterns than those whose content is described by means of written natural language. This may be seen as a direct consequence of the kind of tasks the children were doing throughout formats 1 to 5. Though this conjecture seems to be plausible, the research data suggest that those children with a better command of syntax conventions are more likely to be more able when confronting algebra word problems. As it will be shown in section 5.2, Diego's approach to problem solving

contrasts with Jenny's, for whom facing new situations seemed to encourage her to develop new strategies and notions.
A major difference between "number pattern based problems" and "word problems" is the nature of the verifying referent. Where a number pattern is given, it serves as a referent for the child to verify his responses. That is, the child can check whether the program he has made duplicates (or not) the given number pattern. In the case of "word problems" the problem's statement is what plays the role of the referent. This requires the child to be able to decode it and properly describe the relationships involved by means of calculator language. If that happens successfully the pupil either will get a correct response at the first attempt or, if necessary, he may debug his initial trial by himself. In this respect it is crucial that once the child has made a calculator program he runs it in order to check it with the numerical values he has already obtained by direct calculation. Otherwise, the child may not realise that his algebraic expression is a wrong description of his (quite frequently correct) arithmetic procedure. The research data shows that the most frequent mistake Diego makes in posing a problem is to follow his trend of calculating from left to right disregarding the formal constraints of calculator language.

For example, Diego's written work in worksheets 46-48 suggests that he worked fluently to completion of the tasks despite the difficulty implied by the number patterns. A possible explanation for (apparent) Diego's success is that, in these worksheets, the rules involved do not require the child to be aware of priority of operations ($A\times2-1$, $A\times3-2$, $A\times4+4$). He even successfully solved worksheet 49 at his first attempt (presented as a story-based situation) which required him to break the order of operations just once: $(A\times3\times2+A\times2)\times53$, but he could not successfully do this in worksheets 50 and 55 as discussed below.

The expression $(A\times3\times2+A\times2)\times53$ describes, step by step, a linear way of reasoning such as:

> If A denotes a short side of the rectangle, $A\times3$ denotes its largest side, so $A\times3\times2+A\times2$ allows him to compute the perimeter of the rectangle. He then mul-

tiplied the perimeter by the cost per metre ($53.00), which led the child to obtain the total cost of any window frame.

However, in worksheets 50 and 55, he needed the teacher's feedback to realise his mistakes. The problem situation in worksheet 50 has the same structure as the one in worksheet 49, but the relationship between the large and the short sides requires the child to break the order of operations twice ("the height is 50 centimetres shorter than three times the width"). At his first attempt Diego typed the program (A×3–0.50×2+A×2)×62. Upon being asked, he computed the cost for a specific value without using the calculator, then he compared this result with the one given by the program he had made. Not until then did Diego realise that it was necessary to put in another pair of brackets: ((A×3–0.50)×2+A×2)×62. This extract enhances the value of the calculator both as feedback supplier and as a mathematical environment. It was the formality of calculator's code that made the child produce an expression that conforms to algebraic syntax rules. Otherwise Diego may have kept on working incorrectly, because he could follow with paper and pencil the wrong procedure (A×3–0.50×2+A×2)×62 and still get a correct outcome.

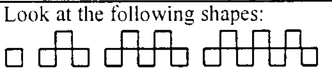
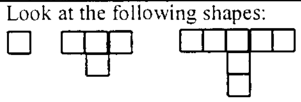
Worksheet 55 (building a square box with maximum volume) also illustrates this point. Here, the way in which Diego completed the table indicates that he fully understood the problem's constraints (see summary of Diego's work). But, despite the fact that Diego's arithmetic procedure was correct he could not properly describe it by using calculator language. This seemed to be due to a lack of awareness of priority of operations at the time he typed the program, instead of typing (100–A)÷2×A, he typed 100–A÷2×A. It should be noticed that Diego might have run the program to check it with the table he had just correctly completed. He did not do this because he tackled the problem guided by the values displayed in the table, thus writing down the program was rather a formal requirement in the task.

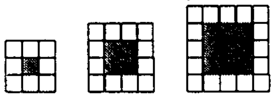
The above episodes suggest that Diego is still not able to confront by himself problem situations which require him to discern when and how to use parentheses in order to produce suitable algebra-like expressions to negotiate problem solutions. Up to this point Diego's reactions show that he still needs help from a more competent person (be it the teacher or a fellow pupil) in order to use calculator language as an efficient tool.

Pragmatics: Diego's approach to inverting linear functions

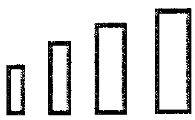
Diego was able to obtain the inverse values included in worksheets 46-50, where the rules involved were as complex as A×2−1, A×3−2, A×4+4, (A×3×2+A×2)×53, ((A×3−0.50)×2+A×2))×62 and A−A×15÷100 respectively. He could not get the inverse programs but resorted to finding the desired values by successive approximations using the above programs. This point is further discussed in the concluding remarks section.

FORMAT 6: Summary of Diego's work.

WS	Problem situation: "Figurative patterns"	Diego's responses
46.	Look at the following shapes:  46.2. How many squares are needed to build up the shape that goes in the 17th place?	Completed without teacher's feedback. **33**
	46.3. How many squares are needed to build up the shape that goes in the 100th place?	Completed without teacher's feedback **199**
	46.4. Explain how you reasoned to answer the questions above.	Completed without teacher's feedback *"I realised that, when I multiplied by 2, I got too many squares, one square more, then I had to take one away and it worked"*
	46.5. Can you program your calculator to complete the following table?	Completed without teacher's feedback **?→A: A×2−1**
	Place: 48 | 75 | 12 3 | **176** | **206** | **254** **No. of squares**: **95** | **149** | **24 5** | 351 | 411 | 507	Completed correctly (Diego's answers in bold)
47.	Look at the following shapes:  47.2. How many squares are needed to build up the shape that goes in the 9th place?	Completed without teacher's feedback **25**
	47.3. How many squares are needed to build up the shape that goes in the 17th place?	Completed without teacher's feedback **49**
	47.4. Explain how you reasoned to answer the questions above.	Completed without teacher's feedback *"It is the same as the worksheet before but here I added 1 to the program before* (A×2−1) *to see if it works, and it worked"*.
	47.5. Can you program your calculator to complete the following table?	Completed without teacher's feedback **?→A: A×3−2**
	Place: 48 | 75 | 12 3 | **143** | **157** | **201** **No. of squares**: **142** | **223** | **36 7** | 427 | 469 | 601	Completed correctly (Diego's answers in bold)

WS	Problem situation: "Figurative patterns"	Diego's responses
48.	Look at the following shapes:  48.2. How many squares are needed to build up the shape that goes in the 27th place?	Completed without teacher's feedback **112**
	48.3. How many squares are needed to build up the shape that goes in the 40th place?	Completed without teacher's feedback 164
	48.4. Explain how you reasoned to answer the questions above.	*"I made a program with the squares then I answered the questions".*
	48.5. Can you program your calculator to complete the following table?	Completed without needing teacher's feedback **?→A: A×4+4**
	(table below)	Completed correctly (Diego's answers in bold)

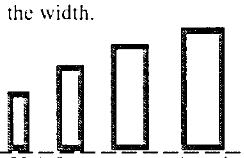
Place	48	75	123	**175**	**192**	**209**
No. of squares	**196**	**304**	**496**	704	772	840

WS	Problem situation: "Rectangular shapes"	Diego's responses
49.	The windows have different dimensions but in all of them the height is three times the width. 	Completed without needing teacher's feedback
	49.1 Can you complete the following table?	Correctly completed (Diego's answers in bold)

Width	0.75	0.86	1.28	**1.17**	**1.41**
Height	**2.25**	**2.58**	**3.84**	3.51	4.23

	49.2. The windows frame is made of wood which cost is $53 per metre. a) How much does a window frame cost whose width is 1.5 metres?	Completed without needing teacher's feedback $ 636.00
	b) What did you do to answer the question above?	*"I multiplied the cost per meter by the height plus the width".*
	49.3. Can you program the calculator to obtain the cost of any window frame?	Completed without teacher's feedback **?→A: (A×3×2+A×2)×53.00**
	49.4. Complete the following table using the program you have just made.	Correctly completed (Diego's answers in bold)

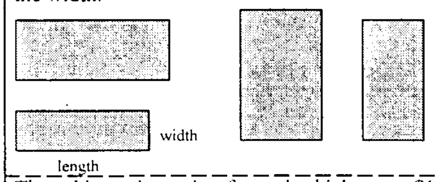
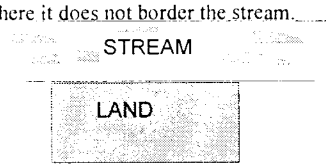
Width	0.68	0.80	0.95	1.15	**1.25**
Cost	**288.32**	**339.2**	**402.4**	**487.6**	530

WS	Problem situation: "Rectangular shapes"	Diego's responses
50.	The windows (below) have different dimensions but in all of them the height is 50 cm less than three times the width. 	
	50.1 Can you complete the following table?	Correctly completed (Diego's answers in bold)

Width	0.30	0.45	1.30	**1.65**	**2.35**
Height	**0.40**	**0.085**	**3.40**	4.45	6.55

	50.2. Can you program the calculator to obtain the cost of any window frame?	At first attempt Diego did not put the outside parentheses. **?→A: ((A×3−0.50)×2+A×2))×62.00**
	50.3. Fill in the blanks using the program you have just made.	Observe that 0.80 is just an approximation.

Width	0.35	0.65	0.84	1.20	**0.80**
Cost	**111.6**	**260.4**	**354.64**	**533.2**	334

WS	Problem situation: "Rectangular shapes"	Diego's responses
51.	The tabletops (below) have different dimensions but in all of them the length is 1 meter greater than twice the width.  The tabletop is made of wood which costs $155 per square meter. Can you program your calculator to obtain the cost of any tabletop?	Diego did not complete this worksheet
54.	A good number of pieces of land are for sale. They all have the following characteristics: the length is 30 meters greater than the width.	Did not complete this worksheet
	54.1. Mr. Pérez needed 132 meters of wire fence to limit his land. What are the dimensions of his land?	
	54.2. Mr. González bought a land whose width is 76 meters. How many meters of wire fence does he need?	
	54.3. Did you program the calculator to solve these problems?	
55.	A man has a piece of land by a stream. He bought 100 metres of barbed wire to fence his land where it does not border the stream.  The man wants to use the stream as a border so that his 100 metres of barbed wire yield the biggest possible rectangular area. It depends on the size of the sides. Can you complete the following table?	He completed the table without using the program he made. Thus, he could not realise that the program was. not properly written.

Large side	50	**40**	60	**80**	70	**84**	65	55.5	54.8
Short side	**25**	30	**20**	10	**15**	8	**17.5**	**22.5**	**22.6**
Area	**1250**	**1200**	**1200**	**800**	**1050**	**672**	**1137.5**	**1234.875**	**1238.48**

	55.1. Can you program the calculator to complete the table?	**?→A: 100-A÷2×A**
	55.2. How long should the long and short sides be to get the biggest area of land?	Short side: 50.2 m Large side: 24.9 m Area: 1249.58 m²

WS	Problem situation: "Percents"	Diego's responses
52	The Music Centre is having a Special Sale: All records 15% off. The discount will be applied on the labelled price.	
	52.1. Can you complete the following table?	Completed correctly

Label Price	$ 34	$ 18.75	$ 126.50	$ 28.50	$ 150	$ 72.35	$ 29.40	(Diego's responses in bold type).
Discount	**5.1**	**2.8125**	**19.02**	**4.275**	**22.5**	**10.8525**	**4.41**	
Special Price	**28.9**	**15.9375**	**107.78**	**24.225**	**127.5**	**61.4975**	**24.99**	

	52.2. Can you program the calculator so that it prints out the Special Price every time you input the Label Price?	**?→A: A−A×15÷100**
	52.3. Use the program you have just made to complete the table.	Completed correctly

Label Price	$84	$28.75	$226.50	$ 29.60	$140.00	**$168.00**	**$170.00**	(Diego's responses in bold type).
Special Price	**71.40**	**24.4375**	**192.78**	**25.16**	**119**	$142.80	$144.50	

WS	Problem situation: "Percents"								Diego's responses
53.	A Book Store is having a Special Sale: All titles 25% off. The discount will be applied on the ticket price.								
	53.1. Can you complete the following table?								
	Label Price	**75.00**	**24.00**	**36.00**	**86.00**	**35.00**	**26.00**	**46.00**	He completed the table correctly.
	Discount	$18.75	$6.00	$9.00	$21.50	$8.75	$6.50	$11.50	
	Special Price	**56.25**	**18**	**27**	**64.5**	**26.25**	**19.50**	**34.50**	
	53.2. Program the calculator so that it prints out the Special Price every time you input the Amount Discounted.								He did not make a program
	53.3. Program the calculator so that it prints out the Amount Discounted every time you input the Label Price								He did not make a program

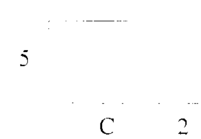
INTERVIEW 3: Discussion of Diego's work

The interview was concerned with the following issues:

- Interpreting algebraic expressions: (geometrical context).
- Simplifying linear expressions.
- Inverting linear expressions.
- Children's strategies to cope with problem situations involving generality.

Pragmatics: The role of the context

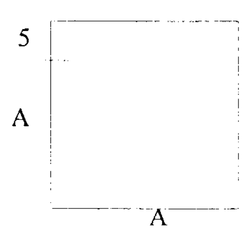
Diego's responses suggest that he was not able to use geometrical diagrams to obtain support to work algebraically. This strongly contrasts with the case of Jennifer to whom diagrams offered a rich context for her reasoning. In Diego's case the visual perception of shapes seems to dominate his reasoning making him disregard quantitative information. What follows illustrates this fact.



He was asked to program the calculator to compute the perimeter of the rectangle shown on the right. He thought that 2 must be used as a scale measurement, so "*to obtain the length of the large side I have to multiply 2 as many times as it fits into the line*" (I3: 5-26). Up to that point he was ignoring the letter C. Then he was told that the "small line" measures 2 and the other part "measures C", and was encouraged to find a way of computing the perimeter of the shape . Diego thought of it for a while and said "*The perimeter is all this line around ... first 5×2 ... No, I don't know how to do that ... I cannot recall the formula for the perimeter*". Could you put it as a calculator program? "*No, I* *don't recall the formula*". Next, he was told that a pupil from another class believes that he can get the perimeter using the program 5×2+(C+2)×2. Immediately Diego agreed because "*with 5×2 you get these two sides ... the height ... then (C+2)×2 gives the other two sides, and that's it*" (I3: 27-30). Then he was asked to program the calculator to compute the area of such rectangle, he fluently did it.

The episode illustrates how well Diego reacted on reading an algebraic expression. At that point he easily accepted that C+2 denotes a length and the problem he had with recalling the formula for the perimeter did not appear at all. The following episode seems to corroborate that he really runs into problems when geometrical contexts appear.

He was asked to program the calculator to compute the perimeter of the rectangle on the right hand. He said it was (A+5)×4 and explained: "*it looks like the one before and all the sides are equal*" (I3: 41-44). Even after he had tried with specific values he hardly accepted the sides were different "*because the shape looks like a square*" (I3: 45-56).



Furthermore, once he had accepted that the shape was a rectangle he asked to be allowed to try the program for the perimeter again. He wrote down A+5+A×3 "*because one side has a length of A+5, the length of the other three sides is A*".

This episode suggests that basic geometrical information is not as intuitive for all children. It also suggests that we need to know enough about children's geometrical background before relying on a geometrical context to introduce algebraic notions. Even children like Diego, who showed himself to be able to correctly read and handle algebraic expressions as complex as 5×2+(C+2)×2 may be disturbed when algebra is put into a geometrical contexts. Diego's lack of geometrical skills may explain the long time he spent trying to deal with the problem about building a square box with maximum volume (Format 4).

Semantics: Diego's notion of literal terms and algebraic expressions.

Diego was given the following story-based question:

> A pupil from another class says that $(A+B)^2$ is the same as A^2+B^2. What do you think about this?

This item involves an expression containing two variables which was completely new for the child[8]. Our aim was to observe the extent to which Diego's experience with calculator language might allow him to make sense of the question and how he faced it.

Diego's responses suggest that he was becoming able to interpret this kind of algebraic utterance. His notion of literal terms as "representing any number" helped him to relate algebraic expressions like A^2+B^2 and $(A+B)^2$ to numbers. This question was made within the paper and pencil environment, that is, he was not asked to program the calculator using these type of expression. However, his first attempt suggests that he was able to grasp the whole expression's structure so as to compare both expressions on the basis of a global estimation of their number value. What follows is aimed at illustrating this.

Diego said after some inspection: "*Here* (pointing at A^2+B^2) *it adds separately the numbers but already raised to square ... Here* (pointing at $(A+B)^2$) *it adds first, then raises it to square ... The latter must be greater*" Were you thinking of a number? "*No, I was just thinking of this* (pointing at the expressions)" Do you agree with this pupil? "*No ... See, 2 plus 3, 5, 5 times 5, 25 ... 2 times 2, 4, 3 times 3, 9, ... 13 ... No, I don't agree*" (I3: 233-238).

Syntax: Diego's notions of parentheses and priority of operations.

Diego's answers while working with geometrical shapes indicate that there is still a long way to go before he incorporates syntax conventions into his working routines. Almost every time he used an algebraic expression working with paper and pencil he ignored the priority of operations. An important point here is that Diego correctly produced

---

[8] Though the Mexican syllabus for elementary school includes formulas for calculating the perimeter and the area of some polygons, the children are never confronted with unclosed expressions such as A^2+B^2.

these expressions when he was asked to work them out using the calculator. Diego's resistance to incorporate mathematical formalities enhances the role of the calculator as an environment where the child must properly produce algebraic utterances. The following extracts illustrate this.

Diego was asked for a way to compute the area of the rectangle (height 5, width C+2) mentioned above. He wrote down 5×2+C. Then he was required to explain what the calculator would do if he tried this expression. He explained: "*it'd do 5 times 2 first, then it'd add C to this*". Then he said: "*if I had to do it with the calculator I'd type 5×(2+C)*" (I3: 32-38). Later on, when working with the rectangle A, A+5, he did exactly the same. After having discussed with him that the opposite sides of a rectangle have the same length he finally wrote down the expression A+5×2+A×2 but correctly computed the perimeter for a specific value by mental calculation. Being asked what the calculator would do if he entered such an expression he explained: "*It would first compute 5×2 and A×2, then would add them altogether and finally would add A to all this*", next he rewrote it as (A+5)×2+A×2 "*because this is the way I'd do it with the calculator*". Finally, being asked how he could know that his answer is correct, he explained "*it's OK because I did it exactly as the calculator would do it*" (I3: 61-68).

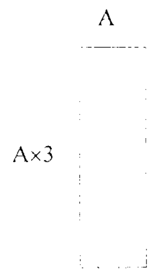
The above excerpts show that priority of operations plays a critical role not only in children's production of utterances that conform to algebraic rules, but also in the development of children's algebraic reading skills. This is clearly exemplified by the above episode where Diego fluently read the expression 5×2+(C+2)×2: "*with 5×2 you get these two sides ... the height ... then (C+2)×2 gives the other two sides, and that's it*". He could not have read such an expression without being aware of the priority of operations. It seems obvious that syntax rules are needed to decode a string of symbolic utterances, but the point I want to emphasise here is that Diego's incipient acquisition of syntax rules took place on the basis of semantic notions developed through using calculator language.

Semantics: Diego's notion of algebraic equivalence

Diego's responses show that, when asked to, he can simplify similar terms within linear algebraic expressions. So far his strategy to do this is twofold. On the one hand he seemed to be developing something like the known method of "adding up the coefficients then put down the variable". On the other hand he resorts to verifying the correctness of his outcomes through comparing the simplified expression with the original one giving specific values to the literal term. Diego's work shows how his simplifying/verifying strategies enable him to know whether an expression can be simplified or not.

In this respect it is important to note that Diego showed a tendency to generate misrules which, during the interview, he could debug by checking with the calculator. This suggests that a teacher's careful supervision is necessary until the child realises that the calculator can be used to get feedback.

For example, Diego made the program A×2+A×3×2 to compute the perimeter of the rectangle shown on the right. Then he was asked if that program might be written in shorter way. His first reaction was A×2+A×6. When being asked to, he made it shorter (A×8). He explained his reasoning as follows: "*I know it gives the same as A×2+A×6 because here these two are multiplying* (pointing at the A's)... *so I make 2+6 ... well ... see ... suppose A was 2, then 2×2, 4, 2×6, 12, 12+4, 16, ... 2×8, 16... it gives the same as A×8*" (I3: 74-86).



Then Diego was told the following story to inquire about the extent to which he has grasped the role of the literal terms within an algebraic expression: "A pupil from another class says that 3×A+2×B gives the same as 5×A×B. What do you think about this?" He answered: "*that pupil was wrong ... because instead of multiplying by B he should have added B, like this: 5×A+B ... let's see ... if A was 2 and B was 3 ... I'd have 3×2, 6 ... 2×4 ... 10. He'd get 5×2, 10, times B, 30 ... that's it, he is wrong ... Then mine ... 5×2, 10, plus B, 13 ... I am wrong too! ... There is the problem* (pointing at 3×A+2×B)

... you have two letters ... you have to enter two numbers ... You cannot make it shorter" (I3: 87-98).

Up to this point Diego seemed to be able to sort out some algebraic simplification. Later on he was asked again to simplify some other expressions. Diego's answers suggest that he is on his way to generating an algorithm but, meanwhile, he prematurely tries to get rid of numerical feedback and generates misrules. For example, he was asked if the program A+2+A+5+A might be written shorter. He correctly simplified it as A×3+7 by visual inspection, but he did not in the case of 2×A+3+A×4+5. He got A×6×2+8 and checked it by comparing the two expressions for A=2. He saw he was wrong but could not correct the mistake. Nevertheless, while explaining his reasoning he found how to debug it: "*Here* (pointing at 2×A and A×4) *you have two multiplications, then I added them ... it gives A×6, then I multiplied it by 2 because A appears twice ... I've already realised that I didn't need to multiply by 2 ... it'd just be A×6+8, that's it*" (I3: 189-198). After this he fluently simplified 3×B+5+4×B+2+B×3 which looks even more complex than the items he faced before (I3: 203-204).

The last item of this type was 7×M+4−2×M+6−1. Diego's response to this question shows that his experience during this interview made him gain self confidence. His first reaction was "*I'm not sure if I can ... but I'll try*". His first approach was 9×M+10−M×2−1 and compared the two expressions for M=2. Then he got 7×M+10−M×2−1 and explained: "*with the program before I got a difference of 4, that's 2×M, so I just took 2×M away from 9×M*". When asked he finally got its simplest form: 5×M−9 (I3: 205-208).

Pragmatics: Diego's approach to inverting linear functions

Diego was asked to produce the inverse program of 4×D+7×D. This question was intended to observe if Diego could resort to simplifying similar terms in order to obtain the inverse rule. We will see next that Diego finally was able to answer the question, nevertheless his approach suggests that for him, symbolic manipulation was not a task he takes up spontaneously. This episode shows that for the child is not enough to have some knowledge about symbolic manipulation to use it as a tool to face problem situations.

Diego's first approach was to make sense of the question by exploring the numerical value of 4×D+7×D for D=2: "*let's see, 4×2, 8 ... plus 14, 22 ... Now, if I input 22 I have to get 2*" (I3: 209-210). At this point it seemed that he would easily get the inverse program (D÷11). However, he faced the question based on the fact that multiplication/division and addition/subtraction are inverse operations. He simply changed multiplication for division and wrote the terms of the expression in the order they originally appeared (4÷D+D÷7). He tried it for D=2, he saw this was wrong and made another trial: 4÷D–D÷7. Here he said "*I forgot to change addition for subtraction*" and tried it again for D=2, he kept on calculating for a while until he got mixed up (I3: 211-213). The teacher was about to suggest that he gives up the task when he said, "***11 divided by D***". Being asked he explained: "*I got 11 from 7+4, I thought of making shorter 4×D+7×D, that's 11×D ... then I inverted it ...* " (I3: 214-220). Then he tried it for D=2 and found that "*the eleven must take the place of D ... then it should be D÷11*" (I3: 221-224).

The above episode shows how his initial exploration with numbers allowed him to realise his mistakes. That is why he abandoned his first trials. Even in his last trial, where he was much more confident, he resorted to checking with numbers.

Diego's work showed again a trend to blindly operate with symbolic expressions. His tendency to prematurely operate leads us to think that despite the availability of the calculator, exploring the numerical behaviour of the expressions involved is a heavy burden for the child. This emphasises the importance of letting children have enough opportunities to work with algebraic expressions within a numerical environment until they get to the position of using numbers to verify their conjectures. Here it is important to note that, throughout Diego's learning phase, a teacher's careful supervision played a critical role. Otherwise, he may have kept on generating misrules being unaware of their mistakes as has been found with students working in a paper and pencil environment (see Matz, 1980).

Diego's responses also emphasise the potential of introducing the study of algebra with no previous rules but the ones the child may have got from his arithmetic experience. His reactions indicate that confronting symbolic manipulation is rather an invitation to explore instead of a heavy burden of working out routines. Although exploration may start as blind guessing it finally results in consistent strategies based on basic number manipulation facts. With time, these incipient children's achievements may result in solid algebraic process. Our data provides evidence that the calculator-based environment does help the child to value numerical exploration as a powerful tool to generate and validate algebraic syntax rules, overcoming what seems a child's natural tendency to premature symbolic manipulation.

Pragmatics: Diego's approach to expressing and justifying generality.

Diego was given two questions aimed at observing the extent to which he was able to extend his prior experience with calculator language to using it to express and justify generality within different contexts. The two questions were focused on numerical relationships, but required the child to use algebra to justify their general validity.

The first question was the following puzzle-like situation:

> Think of a number, add 10 to it and write down the result. Now take the number you thought of away from 10 and write down the result. Now add the first result to the second one ... May I try to guess the final result you got? It must be 20. Then he was asked if he could find out why the final result could be predicted.

The second question was a story-based situation:

> A pupil from another school says that every time he sums two consecutive numbers he gets an odd number. What do you think about this? Will it be true?

The way in which Diego dealt with the above questions indicates that the experience he has had with calculator language enabled him to describe algebraically the relationships involved. His answers suggest that he is aware of why he used algebraic code and what he uses it for in these particular situations. His work also suggests that he is becoming aware of the general nature of a programming expression and that letters serve to represent a range of numbers. As discussed below it seems that a key point for him to be able to face these questions was the numerical referent, which relates to his experience using the calculator. While producing the algebra-like expressions he was thinking aloud "think of a number, say 7 (wrote down *A*) add ten to it and write down the result, that's 7 plus 10, 17 (wrote down A+10), Now take the number you thought of away from ten and write down the result, that's 10 minus 7, 3 (wrote down 10–A), add the first result to the second one, 17 plus 3, 20 (wrote down A+10+10–A)"(I3: 116-126).

However, Diego's work on these two questions makes evident that, to him, there is a difference between using algebraic code to express generality and using it to justify generality. Perhaps, a major problem here is that justifying generality may require the child to know how to operate algebraically and also be aware that symbolic manipulation may lead him to produce fresh information. Diego's reactions show that he can perform some algebraic manipulation but cannot reach by himself the stage of transforming algebraic expressions with the aim of justifying generality. Nevertheless, his experience with calculator language allowed him to justify the general validity of the first question with some scaffolding from the interviewer. For the question about consecutive numbers he gave a good argument but he did not realise that he could relate it to the algebraic expression he had built ("*cause if you put first an even number and then an odd number ... then you have to obtain an odd number when you add them*" I3: 153-154).

Expressing generality, as in the questions we are discussing, was not a topic which had ever been taught to him. Nevertheless, Diego's responses suggest that he could sort it out on his own on the basis of his prior experience of describing algebraically number patterns. Additionally, Diego's reaction to the question about the sum of two consecutive numbers indicates that he is beginning to develop a skill of algebraically describing numerical relationships that go beyond just translating step by step a sequenced procedure, such as that involved in the "think of a number question". The difficulty Diego had to make the step between *expressing* generality and justifying *generality* suggests that using algebraic code to justify generality is a topic that needs more teaching support than that offered by the set of worksheets given throughout this research. The following extracts provide evidence for this.

"Think of a number" question

At the time Diego was given this question he followed it mentally operating with the number he had thought of. Then he was asked if he would always get 20 as a final outcome. He tried unsuccessfully to explain it verbally: "*Yes, because when you take a number away from 10 you get a number ... then you add the number you had thought of to 10 ... it will give 20*". Then he was asked if there would be a general form to express this. Diego immediately said "*with the calculator*", but he did not use the machine, he wrote down A+10+10−A. When being asked to, he explained that the "A" he put in the program "*means a number between 1 and 10*" (<u>I3: 116-122</u>)[9]. After this it was suggested to Diego to look at the program in order to find why it always gives 20. He could not give a clear verbal explanation to this question and was asked to make the program shorter. Since he could not make sense of this question he was asked for an equivalent program and in response he quickly wrote down "*A+20−A*" (<u>I3: 125-126</u>). The next question was if the latter program might explain why 20 will always be the final outcome. He kept on looking at the expression and said "*it is like if it were just 20 ... the result of all this ... Because I took A away, so the A means nothing ... it is like if it were zero ... nothing is left but 20* " What about the program A+10+10−A, "*it would give the same, they are equivalent*" (<u>I3: 134-141</u>).

---

[9] In order to prevent children to become confused with the appearance of negative numbers, they were asked to think of a number between 1 and 10. This explains Diego's reference to "a number between 1 and 10"

"Consecutive numbers" question.

Diego immediately said that he agreed with the other pupil "*cause ... if you put first an even number and then an odd number ... then you have to obtain an odd number when you add them*" (I3: 153-154). Then he was asked if he could represent this in general, for instance with a calculator program. Diego at once said "a program that add two consecutive numbers", he kept on thinking for a few moments and said "*A plus A plus 1*". Being asked to explain, he said "*well, at the beginning I did not know very well how ... because I did not know how to get a consecutive number and add it to A ... then I realised that a consecutive number is one number more ... that's why I added 1 to A*" (I3: 160-162). After this he was asked about a program for the case of three consecutive numbers. At once he wrote down "*A+A+1+A+2*" (I3: 163-164).

This episode suggests that he started reasoning from a perspective of generality. He thought first of a general number (A), then went to find out how to symbolically refer to the number that comes after A. There he resorted to analysing specific cases and went again to generality when finding the relationship between any two consecutive numbers: "*then I realised that a consecutive number is one number more ... that's why I added 1 to A*". It could be seen that it was the idea of constructing a program which put him on the plane of generality. Then it was his experience of finding out number patterns which led him to explore with specific cases to find a general relationship.

The above excerpts show that Diego's experience with number patterns presented as tables (formats 1-5) helped him to cope with general number relationships in other contexts. His acquaintance with symbolically expressing functional relationships enabled him to successfully confront certain problem situations using algebraic code. But he was still a long way from justifying generalisations using his own algebraic code.

DIEGO'S CASE: Concluding remarks

The potential of the calculator within a pragmatic approach

Diego's mathematical achievements during the field work provide evidence for the pragmatic approach to learning a new sign system by using it, and for the potential of the computing device as a basic support for this activity. We discuss this claim in more detail next.

Diego's' semantic notions

It seems that the major semantic notion developed by Diego was about algebraic expressions. He conceives of an algebraic expression as a device to describe and compute general arithmetic procedures. From this notion he derived two other relevant notions:

i. Letters as symbols that serve to "*personify*" any number to him in order to make the calculator's general procedures work.
ii. Algebraic equivalence which he conceives as two or more programs that produce the same outputs for the same inputs.

The work done by Diego suggests that his mathematical notions strongly relate to the way in which the calculator-based tasks were designed. In what follows some conjectures are made with the intention of explaining how he learned about these notions and how he was able to use them as tools to express generality and negotiate problem solutions.

Diego's interpretation of algebraic expressions and literal terms.

Worksheets 1 to 3 require the child to find number patterns generated by function rules of the form f(x)=kx. Since these rules consist of only one operation (i.e. A+4 or A×2) their algebraic description did not disturb Diego's prior arithmetic experience. From worksheet 4 on, most of the rules involved are of the form ax+b.

These tasks lead the child from the very beginning to express his reasoning by means of a 'one-piece' string of operations such as A×2+1[10]. In fact, Diego needed support from the teacher to find this rule (when he could not produce it on his own he was told "to try combining two different operations"). That was the most difficult part Diego had to overcome, once he found the rule he did not show any apparent difficulty in algebraically expressing it. It is discussed below how the calculator's way of working made it sensible for the child to express his reasoning in that form.

The tasks are all goal-oriented activities intended to provide a numerical referent for the child to produce algebraic expressions. It underlines the fine tuning involved in orderly language learning and allows the child to proceed from learning how to refer to objects (arithmetic procedures) to learning to make a request using calculator's language (negotiate problem solutions). From his first encounter with the calculator's language the child is required to express his method by means of the canonical utterances required by the machine. For instance, to carry out the tasks in worksheet 4 (Format 1) he needed to:

i. Find out the underlying pattern shown in a table. For example:

Input	1.1	2.5	3	4.3	5
Output	3.2	6	7	9.6	11

To do that the child had to carefully explore what computations have been carried out with 1.1 to obtain 3.2. Then he had to verify if those computations let him obtain 6 when applied to 2.5, and so on. Diego first described it as "*multiply by 2 then add 1 to this*".

ii. Communicate to the calculator how to multiply **any** of these numbers by 2 and then add 1 to the outcome.

The above process required the child to do something new: to formalise his method. The child accepted the formality of an expression like A×2+1 because it is the way that the

[10] It was observed during the first three months of the school year that his spontaneous way of proceeding

calculator works. Based on this experience Diego developed the notion of letters as symbols that "*personify any number*" (interview 2). The design of the tasks also led the child to extend this notion beyond the set of numbers displayed in the table. This was done by asking the child to complete a new table using the program he had built. To do this, he needed to find the input when the output are given. This made the child analyse more carefully how the program proceeded and, consequently, think of the role played by the letter he was using. For instance, he was asked to complete the table below using the program A×2+1 (Diego's answers in bold type).

Input	1.3	2.8	14	50	81	274	**162**	**209.5**
Output	**3.6**	**6.6**	**29**	**101**	**163**	**549**	325	420

Throughout formats 1 to 4 tables were the only way to encourage pupils to use the calculator language. This was a key point in helping children develop skills to receive and produce calculator's language utterances. In this context, representing an arithmetic procedure goes beyond the mere encoding of what is represented, it is rather a result of an interaction with the known according to a goal.

Diego's work in formats 1 and 3 provide evidence for the way in which his "calculator language" evolved. Most of the worksheets include a task that requires the child to write down the rule he found. In Format 1 he did it using natural language (i.e. *"I divided by 2 and multiplied by 3"*). But in Format 3 he used calculator language to answer these questions. During a classroom session he was asked why he did so, to which he explained *"it is more comfortable ... it requires me less writing and says the same"*. This was his first approach to translating natural language utterances into algebraic expressions. Later on, in formats 4, 5 and 6 he showed that he was able to deal with story-based problems using calculator language.

was not to express his reasoning as 1.1×2+1. He usually performs 1.1×2, 2.2, then 2.2+1, 3.2.

Diego's notion of algebraic equivalence

The research data shows that numerical substitution helped Diego develop a notion for algebraic equivalence. He grasped that two or more programs are equivalent "if they produce the same results" (see Format 3). This notion has to do with synonymy, it was the use of the calculator's language which allowed the child to grasp that a calculator's program may have different representations, but they are equivalent if they "*produce same outputs for same inputs*". An episode from interview 4 illustrates this. Diego built the program A+10+10–A to describe the relationships in the puzzle about "summing to 20". Upon being asked he made the equivalent program A+20–A; this helped him realise why any input number would give 20 and affirmed that the same would happen with A+10+10–A "*because they are equivalent programs*". This notion of equivalence was a building block in Diego's strategies when confronting any kind of algebraic transformation. Certainly, Diego could not have dealt with algebraic transformation without having a notion of algebraic equivalence (transforming an algebraic expression so as to make it equivalent to a target expression).

The next sections will analyse how, in time, Diego's semantic notions allowed him to make sense and successfully cope with algebraic manipulations, such as inverting linear functions, constructing equivalent algebraic expressions and simplifying similar terms within linear expressions. His work shows that he confronts symbolic manipulation by exploring the numerical behaviour of algebraic expressions (that is, as a semantic activity). Later on he began to set up initial syntax rules. This relationship between semantics and syntax allowed Diego to develop strategies to verify his incipient syntax rules.

Diego's syntactic notions: Priority of operations and use of parentheses

Diego's work emphasises the role of tasks designed to let the child use calculator's language to produce his own ends (Format 2). While working out these tasks Diego did not only learn "how to say it" but also what is canonical. For instance, trying to make something difficult for a fellow pupil to guess, Diego made a table following the rule A÷–5×7 (without using the calculator). Then he ran the program and found that the calculator proceeds differently from himself. It was really his first encounter with the priority of arithmetical operations. Diego engaged himself in disentangling it and found why this occurred (see interview 2). For this to happen, it was crucial that Diego was able to carry out the calculations involved in advance, otherwise he would not have noticed that the calculator proceeds in a different way. It is worth noticing that it was his notion of the calculator's program as a device to calculate which allowed him to make sense of this syntactic convention.

During the study Diego repeatedly showed that he was unaware of priority of operations when working with paper and pencil. But he could always correct on his own if he was required to carry out the calculations by means of a calculator's program. This suggests that a longer period using the calculator is needed before the child incorporates syntax conventions into his everyday computing routines (see section on priority of operations, interview 3).

The episode about the perimeter of the rectangle with height $A+5$ and width A illustrates this (interview 3). Diego produced the complex expression (A+5)×2+A×2. The process by which Diego produced such an expression shows that he values syntax restrictions only when the context demands them. Before he got to correctly express the perimeter he wrote A+5×2+A×2; nevertheless he used this expressions to compute the correct perimeter for several specific values. The incorrect syntax did not perturb him because he was following his own reasoning: "*first the side's length plus 5, that's the height, then multiply it by 2, then the side's length plus 2, then add together the results*". However, when being asked to do this with the calculator he typed without any hesitation (A+5)×2+A×2 "*because that's the way I'd do it with the calculator*".

Using parentheses was a feature Diego needed to be taught about. This took place once he realised that the calculator proceeds differently from him (certainly the tasks were not designed for him to find what parentheses are for). Once Diego understood the function of parentheses he used them as frequently as he could. Since the tasks required Diego to produce algebraic expressions his experience with parentheses was focused on using them, later he showed he was able to read them too.

Diego's work in Format 3 supports what has been discussed above. In worksheet 22, he used parentheses to obtain equivalent programs, for example, from B×1.5 he built the programs B×(1+0.5) and B×(2−0.5). Later on (interview 3), when transforming A^3 so that it "gives the same" as A^2, he resorted to using parentheses: $A^{(3-1)}=A^2$. In fact, Diego was faced with reading parentheses only once (interview 3). There, he was asked to judge somebody else's claim: "a pupil from another class says that he can compute the perimeter of a rectangle with height 5 and width C+2 using the program 5×2+(C+2)×2, what do you think about that". Diego properly read this sophisticated expression which provides evidence for his achievement in this respect. This suggests that Diego's pragmatic encounter with syntax conventions, such as order of arithmetic operations and use of parentheses, helped him to acquire them as tools for his reasoning. He would not have been likely to have produced these achievements by a passive listening to teacher's explanations and reading expressions containing parentheses.

Diego's strategies: Numerical substitution

Diego's work shows that numerical substitution was his strongest strategy for coping with algebraic transformation. He used it wherever algebraic transformation was required: in operating with a given expression to make it equivalent to another expression, in inverting function rules and in simplifying algebraic expressions.

Diego used numerical substitution to make sense of algebraic manipulation, that is, he essentially used numerical substitution as a semantic recourse. This fact seems to validate the role assigned to arithmetic as a "shared context" within the study. Diego's strategy of numerical substitution helped him go back and forth between the general and the particular. This process helped the child gain awareness of the general nature of programming expressions and of the role played by specific cases as generic examples. In fact, Diego resorted to numerical substitution to explore generality whether to validate or to refute conjectures. For example, being asked to type an equivalent program to C×(5−4) he got C×5−C×4 (interview 2). He certainly did not have in mind the distributive law (he did not know about it). What he did was to "*think of C×(5−4) without parentheses ... I mean C×5−4, if C was 5, C×5 would give us 25, 25−4 doesn't give the* *same as C×1, I needed minus 20 ... it is C×4 ... it works*". This shows his transition from the general to the particular to the general. During interview 3 he was asked to analyse another child's statements, like "a pupil from another class says that $(A+B)=A^2+B^2$". Though this type of expression was never used before he made sense of it with numbers and concluded that it was wrong: "*Here* (pointing at A^2+B^2) *it adds separately the numbers but already raised to square ... Here* (pointing at $(A+B)^2$) *it adds first, then raises it to square ... The latter must be greater*" When asked if he would agree with this pupil he resorted to numerical substitution to give stronger evidence: "*No, I don't ... See, 2 plus 3, 5, 5 times 5, 25 ... 2 times 2, 4, 3 times 3, 9, ... 13*".

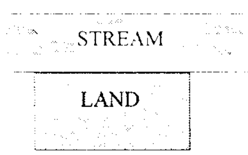
Diego's approach to symbolic manipulation.
Interview extracts cited earlier show that Diego was able to handle the task of transforming an algebraic expression to make it equivalent to another. Here we will try to make some conclusions of how he became able to do this.

Diego tackled algebraic manipulation based on his incipient experience of using parentheses and operating with the algebraic terms. For instance, when transforming 10×B to make it equivalent to 11×B, Diego inserted parentheses to operate with the coefficient: (10+1)×B. Next he was asked to transform 10×B so that it makes the same as 7×B without using parentheses. He immediately started to manipulate the algebraic terms: 10×B–3×B and explained: "*first the program multiplies by 10 ... then I had to make it multiply by three numbers less than ten, so we have to decrease B three times when we multiply*" (interview 2). Later on, his answer to worksheet 47 tells us about his tendency to symbolic manipulation. In worksheet 46 he had built the program A×2–1, then he produced the program A×3–2 by "*adding 1 to A×2–1*" (in fact, he added A–1, but he could not explain it).

At the time Diego was confronted with algebraic transformations he had just completed five sessions programming the calculator (Formats 1 and 2). It seems that Diego's apparent proclivity to algebraic manipulation led him to proceed too rapidly from exploring with numbers to operating symbolically. An episode cited earlier (interview 2) illustrates this. Despite Diego's initial approach to obtain the inverse of 4×D+7×D was to explore its numerical value, he abandoned this strategy and inverted the expression as 4÷D−D÷7. He could have left it like this, but the interviewer's questions made him review the expression until he finally got a correct answer.

Diego's approach to negotiating problem solutions

We have already discussed Diego's achievements in problem solving when reviewing his work in Format 6. Diego's work shows that, despite the difficulty involved in most of the problem situations, he was able to make sense of a problem's constraints and worked out some of them successfully (see table below). It shows that his experience describing number patterns allowed him to use the calculator language to describe the relevant relationships.

WS	Problem situation	Diego's answer
33.	I have some pieces of wire, all are of length 16 cm. I want to cut them all into two pieces in different ways, for example, 12 cm and 4 cm, 11 cm and 5 cm, and so on. Can you program the calculator so that if I input the length of one small piece it prints out the length of the other one?	**Correctly answered** ?→A: 16−A *"Because in earlier worksheets we were taking away from numbers and I did the same here"*
49.	Programming the calculator to find the cost of a class of rectangular window's frames whose height is three times as its length. The cost per metre is $53.	**Correctly answered** (A×3×2+A×2)×53.00
50.	Programming the calculator to find the cost of a class of rectangular window frames whose height is 50 centimetres less than three times its length.	**First approach:** (A×3−0.50×2+A×2)×62 **Second approach:** ((A×3−0.50)×2+A×2))×62
52.	Programming the calculator to find the special price of any merchandise when 15% is off and the regular price is known.	**Correctly answered** A−A×15÷100
55.	Programming the calculator to obtain the dimensions of a "three sides" rectangle with maximum area whose "three sides" perimeter is 100 metres. 	He missed putting in parentheses 100-A÷2×A

Perhaps the most important point in this respect is the crucial role played by the calculator in helping the child to proceed from learning "how to say it" to "saying it" according to algebraic canonical forms. Diego's work shows that understanding the problem's constraints was not enough for him to correctly express them, but it is also necessary that the child masters syntactic conventions. His first approach to worksheet 50 illustrates this (see table above). He did not notice anything wrong with this expression because he used it to outline his calculations with paper and pencil. That is, such an expression fits well his line of reasoning. A the teacher's intervention was needed to make him realise his mistake.

Diego's achievement shows that the approach to algebra based on describing number patterns enabled him to make sense of and cope with some story-based problem situations. We found that the major step between describing number patterns and negotiating problem solutions is that, in the former, algebraic representation is a result of an interaction with the known (arithmetic facts) according to a goal (making the machine do what the child expects). This kind of interaction helps the child to check by himself the correctness of his algebraic utterances. This interaction (content-context) hardly occurs when the child fails in representing algebraically the relationships within a story problem situation, because, there, the role of the known (shared context) is played by quantitative relationships verbally delivered. This suggests that encouraging children to explore by specific cases should be of great help for them in getting an algebraic representation to negotiate problem solutions. Then, the formality of the calculator's code may aid them to find what is canonical.

Diego's approach to expressing and justifying generality

Diego's reactions during interview 4 shows that his experience describing number patterns helped him gain confidence in using calculator language as a means to describe quantitative relationships. With no apparent struggle he produced the expressions A+10+10–A and A+A+1 to refer respectively to the questions of "think of a number" and "consecutive numbers". This shows that Diego's use of calculator language goes beyond a mere description of arithmetic procedures, he is certainly expressing generality. For instance, he was asked if the "think of a number" puzzle may be described in general, he immediately answered "with the calculator" and typed the program

A+10+10–A. Furthermore, upon being asked, he explained that the letter A represents "*any number between 1 and 10*". A similar situation took place with the "consecutive numbers" item, for instance, he explained that the difficulty he had in making a program was "*to find out how is the number that comes after A*" (the letter he was using to program the calculator).

The research data indicate that there is a big step between using calculator language to express generality and using it to justify generality. Diego's experience with the set of worksheets used in this study was not enough for him to grasp that calculator language can help him to justify generalisations.

Diego's conceptions

Diego's work shows that simplifying algebraic expressions deserves special teaching attention. As well as Diego, the case-study children -except those with an above average attainment- showed a tendency to generate false rules when simplifying algebraic expressions. Since this trend was never present with other forms of algebraic transformation I think that the major reason for this to happen is that, when simplifying, the child has only one expression to operate with. This differs from the case of transforming a given expression to make it equivalent to another, where both the source and the target expressions are available. Diego's approach to inverting function rules corroborates this. There, he was also given only one expression to operate on and in this case also generated misrules. What follows is intended to make this clear.

When transforming a given expression to make it equivalent to another, for example, transforming B×7 to make equivalent to B×10, Diego invariably obtained a specific value for the target expression; having this as a clue, he went on to operate with the source expression. Finally he compared the numerical value of the target expression with the new one and repeated the process as necessary until the desired result was obtained.

Now, let us look at the task of simplifying an algebraic expression. There, the child is required to construct the target expression having the source expression. To do this, Diego obtained a numerical value for the source expression. If the expression was of the form $ax+bx$, he easily found a shorter equivalent expression. For example, B×5+B×4=18 for B=2, this suggested to him that the rule should be "multiplying by 9". In that way he began to construct initial rules like A×3+A×2=A×5. But, exploring the numerical behaviour of expressions like 2×A+3+A×4+5, appeared to be too heavy a burden for him to fulfil (though he could have done it). It seems that the expression's complexity encouraged him to try out his own rules for algebraic manipulation, for example, 2×A+3+A×4+5=A×6×2+8. He explained this as follows: 2×A+A×4=6×A×2 because "*2+4=6, **but A appears twice**, it must be 12 ... 12×A*". This led him to fail even with questions he had shown he could master before.

Diego's strategy of counting the "A's" relies on his previous findings about letters as entities to operate with. The following illustrates this. Just before the question we discussed above, Diego correctly simplified A+2+A+5+A. Despite the expression's complexity, obtaining how many "A's there are" did not make him run into problems. This suggests that his error arose from applying this strategy where coefficients are different from 1. To debug this mistake Diego needed the interviewer to question his work. It was not until he tested his "rule" upon the expression's numerical value that he agreed that something was wrong. It is worth noticing one more feature about Diego's approach to algebraic simplification: He never showed any evidence of making mistakes such as $a+b=ab$, $a+a=a^2$, $2\times a=a^2$, $5\times a+3=8\times a$, nor $(a+b)^2=a^2+b^2$. I think this was due to the strong link between numbers and letters provided by the calculator environment. This suggests that these types of children's misunderstandings are induced by particular teaching approaches.

General remarks:

Priority of operations

Diego's work suggests that mathematical conventions such as priority of operations and use of parentheses play a crucial role in bridging the gap between arithmetic and algebra. The research data show that he attained a good level of command of these conventions. Nevertheless, there were also a good number of episodes that showed his resistance to incorporating priority of operations into his everyday ways of working, particularly when working with paper and pencil.

Mental calculation

Diego's approach to symbolic manipulation strongly relies on mental calculation. Diego's first successful attempts to operate with literal terms took place where coefficients were positive integers and addition or multiplication were involved. The interview data suggested that his skills of mental calculation had much to do with both finding equivalent expressions and simplifying similar terms. This leads us to suggest that special teaching attention should be paid to mental calculation if it is intended to introduce algebra from an arithmetic approach.

5.2. THE CASE OF JENNY

PHASE 1: Jenny's entry into Calculator's Language.

Phase 1 is organised as follows: First Jenny's written work is analysed, then a summarised transcription of her work is presented intending to give further support for the preceding analysis; finally Jenny's work during Interview 1 is analysed.

FORMAT 1: Discussion of Jenny's work

This part reports on the work done by Jennifer within formats 1 and 2 and the first interview. A discussion of Jenny's written work is presented first, then a summarised transcript of her work is presented. Finally Jenny's reactions during the first interview are analysed.

Pragmatics: Jenny's approach to inverting linear functions

Children were not explicitly required to use inverse programs to produce the numbers given as outputs. Nevertheless Jenny tried out inverse programs from the very beginning, see, for example, worksheet 3. At this stage she made this explicit only in cases where the functional rule involved just one operation. Though she did not show how she calculated the inverse numbers when a "two step" expression was involved she managed herself to do it correctly without having typed the inverse program. For example, see the table below which Jenny completed using the rule A×2+1 (the numbers in bold type are Jenny's responses, worksheet 4).

Input	1.3	2.8	14	50	81	274	**162**	**209.5**
Output	**3.6**	**6.6**	**29**	**101**	**163**	**549**	325	420

Semantics: Jenny's notions of literal terms and algebraic expressions.

Jenny kept doing the tasks quite motivated and correctly completed the 15 worksheets prepared for this format. Only 5 out of the 23 children in the class completed all the worksheets (the time allowed was five sessions). Her work shows that she has grasped how to read a table and express the relevant function rule as a program.

Pragmatics: Jenny's approach to negative numbers

This theme was new for all the children when the course commenced. Only those aspects concerning the order of negative numbers and their use to represent magnitudes (like temperature, and profits and debts) were treated in classroom sessions. The topic of how to operate with negative numbers was not treated in order to observe the extent to which the calculator might help the children cope with it. In this respect Jenny correctly completed worksheets 6 and 7 which include tables whose inputs are decimal negative numbers. Her answers show that she was able to operate with such numbers, otherwise she could not have managed to find the functional rules that generate these tables (see the summary of Jenny's work below). Jenny explained that she found how to operate with negative numbers through exploring their behaviour with the calculator. For example, she found that –10+0.5=–9.5 by a trial and refining strategy, like subtracting 0.5 from minus 10, since this did not work, she tried adding 0.5. Then she systematised this experience so as to find how to program the calculator to produce the given table.

FORMAT 1: Summary of Jenny's work

Work sheet	Clues given						Rule written in natural language	Expression produced	Completing the table
1.	**In**	1	2	3	4	5	*Add 4 to each number*	?→A: A+4	OK.
	Out	5	6	7	8	9			
2.	**In**	7	8	9	15	18	*Multiply the number by 2*	?→Z: Z ×2	OK even with decimal numbers.
	Out	14	16	18	30	36			
3.	**In**	2.5	3.1	4	4.2	5.3	*Multiply the number by 3, sometimes to divide it by 3.*	?→J: J ×3	OK even with decimal numbers.
	Out	7.5	9.3	12	12.6				
4.	**In**	1.1	2.5	3	4.3	5	*Multiply the number by 2 and add 1 to it.*	?→L: L×2+1	OK, it included calculating inverse values involving decimal numbers.
	Out	3.2	6	7	9.6	11			
5.	**In**	1	2	3	4	5	*Multiply the number by 2 and take away 1.*	?→L: L×2−1	OK
	Out	1	3	5	7	9			
6.	**In**	-10	-9.7	-7.8	-6.2	-5.3	*Not required*	?→Y: Y+.5	OK.
	Out	-9.5	-9.2	-7.3	-5.7				
7.	**In**	-15	-14.5	-12.4	-10.2	-5.8	*Taking away 1.5*	?→H: H−1.5	OK, it included calculating inverse values involving negative numbers.
	Out	-16.5	-16	-13.9	-11.7				
8.	**In**	10.5	14.42	15.3	16.7	20.1	*Dividing the number by 2*	?→C: C÷2	Not required.
	Out	5.25	7.21	7.65	8.35	10.05			

Work sheet	Clues given						Rule written in natural language	Expression produced	Completing the table
9.	**In**	6	8	14	15	18	*I multiplied the number by 3 and divided the result by 2.*	?→C: C×3÷2	OK, it included calculating inverse values.
	Out	9	12	21	22.5	27			
10.	**In**	4	6	9	10	12	*Multiply the number by 1.01*	?→C: C×1.01	OK, it included calculating inverse values involving decimal numbers.
	Out	4.04	6.06	9.09	10.1	12.12			
11.	**In**	7	9	10	12	16	*I multiplied the number by 3 and added 2 to it.*	?→D: D×3+2	OK, it included calculating inverse values involving decimal numbers.
	Out	23	29	32	38	50			
12.	**In**	7	7.5	8.2	9	9.6	*I multiplied the number by 3 and took away 1 (I did some of them the other way round)*	?→B: B×3−1	OK, it included calculating inverse values involving decimal numbers.
	Out	20	21.5	23.6	26	27.8			
13.	**In**	10	15	20	25	30	*I divided the number by 4 (and vice versa)*	?→E: E÷4	OK
	Out	2.5	3.75	4	6.25	7.5			
14.	**In**	2	3	4	5		*Multiply the number by 2.5*	?→Z: Z×2.5	OK
	Out	5	7.5	10	12.5				
15.	**In**	0.15	0.27	0.3	1.5	2.03	*Multiply the number by .100*	?→X: X×.100	OK, it included calculating inverse values involving decimal numbers.
	Out	0.015	0.027	0.03	0.15	0.203			

FORMAT 2: Discussion of Jenny's work

Syntax: Jenny's notions of using parentheses and priority of operations.

She started using parentheses to express functional rules on her own initiative. Jenny constructed expressions which required parentheses, for example, *"first take away 1, then multiply this by 3"* (see worksheet 1). Such expressions are structurally different from the ones she confronted in Format 1 in which the functional rules involved were of the type f(x)=ax+b. The way in which she cope with using parenthesis is discussed later (interview 1). The following table summarises her work in Format 2.

Work sheet	Program that Jenny used to build the table	Clues given by Jenny						
16	? →A: (A-1)×3	**Input**	1	3	5	8	10	20
		Output	0	6	12	21	27	57
17	? →A: (A+7)÷2	**Input**	1	3	5	8	10	20
		Output	4	5	6	7.5	8.5	13.5

INTERVIEW 1: Discussion of Jenny's work

At the time the interview was given Jennifer had been engaged in programming tasks during five classroom sessions (four in format 1, one in Format 2). The interview was focused on observing three points: (i) Semantics: the notions the child might have developed about letters and symbolic expressions used in calculator programs, (ii) Syntax: parentheses and priority of operations, and (iii) Syntax: whether the programming experience may help the child in facing symbolic manipulation of algebraic expressions. This task is placed into the calculator context as transforming, for example, to manipulate 4×A somehow in order to make it equivalent to 3×A. These points are discussed in what follows.

Semantics: Jenny's notion of literal terms and algebraic expressions.

Jennifer's notion of letters is that "*they stand for any number*". She has also grasped that a programming expression does not depend on the letter used to express it. It is also important to notice that this notion of letters allowed her to successfully work out the task about transforming a function rule to make it equivalent to another function rule. The notions she developed about letters and its role in programming expressions were derived from the ways she used them in Formats 1-2. These notions were never discussed in class. What follows provides evidence for this.

She was asked what do the letters she used in the program (A+7)÷2 mean for her, she said: "*any number ... A can be any number ...*". (I1: 31-32). Going further on this point she was asked what does the program (Z+7)÷2 do. She said "*it does the same as (A+7)÷2, because when you put the letter in the calculator it doesn't matter if it is A or Z. A may be 1 and Z may be 1 as well and so on for any number, it is the same regardless the letter you put in*". (I1: 33-38). Jenny's reaction suggests a potential problem which she may have to face when dealing with polynomials or equations containing two or more variables. This point is further discussed in Chapter 8.

Jenny's notion of algebraic expressions also shows that she has grasped its generality. For instance, she was asked how she knew that a program is correct. She answered "*when everything goes well with the same code (*the same programming expression*) ... not only one or two (*input-output pairs*) have to go well, but all of them*" (I1: 25-28).

Syntax: Jenny's notions of parentheses and priority of operations.

Jennifer gained awareness of the use of parentheses and priority of operations once she realised that the calculator did not give the results she was expecting to get. For this it was crucial that she had made the computations in advance, otherwise she would not have been able to notice that difference. It is fair to say that this happened because she engaged herself in constructing a program which adds or subtracts before multiplying or dividing. In other words, it seems that it was this goal-oriented task that impelled her to accept the value of using parentheses and accept them as conventions imposed by the formality of calculator code. What follows provides evidence for this.

In the interview she was asked why she used programs containing parentheses in Format 2 (worksheets 1 and 2). Her answer shows how she came to grasp the role of parentheses and its relation with the priority of operations. "*I had already mentally calculated the outputs and put them in the table ... but when I tried with the calculator it didn't do what I had in mind. I wanted to take 1 away first, then multiply ...*" What do you mean by "it didn't do what I had in mind". "*As I don't think as the calculator does, I said 1-1, 0, then 0×3, 0. The machine said: –1×3, -3, 1-3, -2. I had to know how to make the calculator do what I wanted (*it was a task requirement*) ... I put in parentheses (*she wrote down the program (A–1)×3*) ... It does what I had in mind*". Let's suppose you want to explain to one of your fellows about using parentheses, how would you do it. "*Parentheses serve for the calculator not to perform the two operations at a time ... I mean ... first one then the other*". (I1: 1-12).

Semantics: Jenny's notion of algebraic equivalence.

Although it may seem trivial to transform 4×B in order to make it equivalent to 3×B the research data shows that it is not. Jenny's reactions suggest that it was the notion that she developed for letters as symbols that represent a range of numbers which helped her firstly, to make sense of what the question was about, and secondly to successfully work it out.

Jenny's answers indicate that the experience she has had with the calculator programming language led her to develop a notion of algebraic expressions as devices to describe and perform arithmetic procedures. Particularly, this notion comprises the use of letters as symbols that represent a range of values.

The research data suggest that an approach to mathematical functions as devices to carry out arithmetic procedures makes algebraic manipulation an action of semantic interpretation rather than a process of applying syntax rules. Jenny's responses indicates that this kind of encounter with symbolic manipulation allowed her to generate initial rules to face these tasks as well as to develop strategies to check their correctness.

To make sense of the question Jenny needed to explore the numerical behaviour of the relevant expressions. She first tried with B=2 and, having in mind that 3×B=6, compared the two expressions and found that 4×B–2=3×B, for B=2. Here, what was crucial was the fact that she was aware that the relation 4×B–2=3×B was only a particular case. At this point she knew that something should be taken away but it took her some time to realise that it was enough to subtract B from 4×B. She needed to explore more carefully the question and tried with B=1, B=4 and B=5. It was then she realised that there were an underlying pattern: the number she took away from 4×B was the only one value of the variable that makes it equivalent to 3×B. Say, 4×B–1=3×B, for B=1, and 4×B–5=3.B, for B=5. Thus, as she wanted both expressions to be equivalent regardless of the value she input, that "thing" to be taken away from 4×B must be the variable itself. Once Jenny had this insight she still needed to repeat this experience with a number of items before beginning to sketch a rule to perform such algebraic transformations. Her work suggests that Jenny started to make a shift from her first notion of letters as "representing any number" to the new notion of letters as entities on which she can operate[1]. The following extracts illustrate what is above.

> **Question:** I wanted to type the program 3×B but I made a mistake and typed 4×B. Can you correct my program without deleting anything of what I have typed?

Her first attempt was 4×B–2, she said "*it wouldn't work ... neither it would if I take 1 away*". Being asked why, she said: "*4×B-1 works only if B=1*" (I1: 41-44). She kept thinking and said: "*If I tried 4×B-5 it would only work for B=5*" Then she was given a hint: when you tried 4×B–4 it only worked for B=4. What can we do to make it work for 1, 5 and other numbers. She said "*This way ...* " and typed B×4-B. (I1: 41-60).

When asked why she explained "*it will now work for all the numbers ... because B is any number, 4 times that number and then it takes away ... well, this thing*[2] *... well, I mean ... if B is 6, 6×4, 24, as we know we want it to be 18, then minus 6, 18*". (I1: 66)

Then she was asked to correct the program 10×C so that it makes the same as 7×C. She immediately said: "*It's the same as the one before, isn't it?*". Surprisingly she got mixed up (she was expected to do it easily). Thus, the question was changed to transforming 10×C so that it made the same as 9×C. It was then that she really related this item with the one before. Finally Jenny was asked again to transform 10×C so that it makes the same as 7×C and 4×C. She faced them without hesitation (typed 10×C-3×C and 10×C-6×C respectively). (I1: 67-100).

PHASE 2: Jennifer's entry into Algebraic Manipulation

This section reports the work done by Jennifer within formats 3, 4 and 5 and interview 2. Similarly to the section before, Jenny's written work will be first discussed, then a

---

[1] This issue will be taken up again in interviews 2 and 3.
[2] It is interesting to notice that she actually does not know how to refer to the variable ("*this thing*") but she does know the role that the letter is playing in the expression.

summarised transcript of her work is presented to provide evidence for the analysis done in the preceding discussion. Finally Jenny's work during the third interview is analysed.

FORMAT 3: Discussion of Jenny's work

Jennifer completed correctly the whole set of worksheets which these three formats include. It seems relevant that she put into play the new tools she had just met, such as operating with literal symbols and using parentheses (see worksheets 21, 26, 27). She also used negative numbers to find equivalent expressions (worksheet 27). Her work shows that she has been exploring (by herself) with negative numbers using the calculator since operations involving laws of signs were not treated in class before (i.e. $-5 \div -5$). These points are next described in more detail.

Syntax: Jenny's notions of using parentheses and priority of operations.

She used them unnecessarily but not incorrectly in worksheets 21-28, and completely correctly in worksheet 26.

Semantics: Jenny's notion of algebraic equivalence.

Jennifer operated with literal symbols to obtain equivalent algebra-like expressions where the coefficients were integer numbers (worksheets 21, 26, 27 and 28). Where fractional coefficients were involved she just operated on the independent term (worksheets 22, 23, 29 and 30). The way in which Jenny carried out the tasks indicates that she is starting to operate with literal symbols within algebraic expressions. It seems that she was only able to transform expressions algebraically when coefficients were integers because this is directly connected with her skill in mentally operating with integer numbers.

Semantics: Jenny's notion of literal terms and algebraic expressions.

It is worth noticing the way in which she used the notion of literal symbol as representing a range of numbers in the program ?→A: A (worksheet 27). It seems that she is conceiving the letter as an entity (range of numbers) because she did not need to accompany the letter with any number (though it might well also be due to the fact that she is seeing A+0=A).

FORMAT 3: Summary of Jenny's work

Work sheet	Given table	Rule written in natural language	Programming expression produced	Equivalent expressions produced.
21.	In 1 1.5 3 5 Out 4 6 12 20	Multiply the number by 4	?→W: W×4	?→A: A×5−A ?→X: X×3+X ?→Y: Y×2+(Y×2) ?→C: C×6−(C×2) ?→D: D×7−(D×3)
22.	In 2 4 8 10 Out 3 6 12 14	Multiply the number by 3 and divide this by 2	?→N: N×3÷2	?→A: A÷2×3 ?→O: O×2÷2×1.5 ?→Q: Q×1.5 ?→A: A×6÷4

Work sheet	Given table	Rule written in natural language	Programming expression produced	Equivalent expressions produced.
23.	In 1 2 3 4 Out 0.25 0.5 0.75 1	Divide the number by 4	?→A: A÷4	?→A: A×0.25 ?→A: A×2÷8 ?→A: A×3÷12 ?→A: A×0.50÷2
24.	In -1 3 7.4 17 Out -0.5 1.5 3.7 8.5	Not required	?→A: A÷2	?→A: A÷4×2 ?→A: A×2÷4 ?→A: A÷6×3 ?→A: A÷120×60
25.		CANCELLED		
26.	In 1 3 5 9 Out 6 10 14 22	Not required	?→A: A×2+4	?→A: A+A+4 ?→C: C+4+C ?→A: A+A+(2×2) ?→B: B+B+(2×4÷2)
27.	In 15 16 17 18 Out 15 16 17 18	Not required	?→A: A+0	?→A: A+A−A ?→A: A×120÷120 ?→B: B+5−6+1 ?→C: C×−5÷−5 ?→A: A÷A×A ?→A: A−A+A ?→A: A
28.	In 1 3.2 5 9 Out 1 10.24 25 81	Not required	?→A: A×A	?→A: A^2 ?→A: $(A^2\times2)\div2$
29.	Here, the program ?→N: 3.5×N was given instead of giving a table.		?→A: A×7÷2	?→A: A÷4×14 ?→A: A÷8×28 ?→A: A÷16×56 ?→A: A÷10×35
30.	Here the program ?→Z: 1.02×Z was given instead of giving a table.	Not required	?→A: A×2.04÷2	?→A: A×4.08÷4 ?→Z: Z×1.02×8÷4÷2 ?→Z: Z÷2÷4×8×÷1.02 ?→Z: Z×8.16÷8

FORMAT 4: Discussion of Jenny's work

Jenny completed eight of the ten worksheets within the time allowed (three sessions). She completed the last two worksheets later. Jenny's work is described in more detail in what follows.

Pragmatics: Jenny's approach to negative numbers

Jennifer's responses (worksheets 35, 37, 38 and 40) show that her explorations with negative numbers have enabled her to operate with them supported by the calculator. It is worth noticing that this implies the acceptance of formal restrictions imposed by the machine's code. Particularly, the use of the minus sign for subtracting and the minus sign assigned to declare a negative number. See, for example, Jenny's explanation of how she found the program (A×−1)−A+10.5: "*I multiply the number by −1 -it gives the same number but negative, then I take away the same number and add 10.5*" (worksheet 38).

Pragmatics: The role of the context

Jenny's work shows how a new semantic referent influenced the way in which she tackles the tasks. Worksheets 31 and 33 are presented as word problems which suggest that the rule to be found is of the form f(x)=k−x, where the value of k can be found by adding any input value to its associated output in the table. Her written explanation in worksheet 33 proves that she grasped this clearly: "*I realised that all pieces of wire gave a total of 16 cm, then you put a number and the calculator gives how many units are left to get to 16*" (her program was 16−Z). She still used this strategy to deal with worksheet 32 which consisted of constructing a program that duplicates a given table. Nevertheless she did not used this strategy to work out the rest of the tasks. In worksheets 35-40, where the referent was a table, she produced rules as sophisticated as A×−1+1, A×−1+1.5 or (A×−1)−A+10.5 instead of 1−A, 1.5−A and 10.5−2×A respectively. In this case her strategy consisted of finding out what operations she had to perform on the input number (A) so that they produced the output number (f(A)).

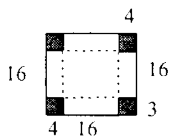
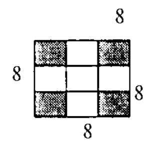
Although eight out of the ten worksheets have the same structure (f(x)=k–ax) Jenny did not seem to realise this. That is why she worked them out differently according to the way in which each task was delivered. This suggests that in addition to repeatedly confronting the child with algebraic utterances it is needed to help her recognise generic algebraic forms.

Pragmatics: Jenny's use of algebraic language to negotiate problem solutions.

In this format the children met for first time the use of calculator language to negotiate problem solutions. Jennifer was able to represent algebraically quantitative relationships involved in story-based problems (worksheets 31, 33 and 34). During the classroom session she explained to the teacher how she completed worksheet 34 (the box with maximum volume). "*It looks like the other problems ... in this case the side of the cardboard has a length of 24 centimetres ... The A is the length I'm cutting out ... well, it is also the height of the box ... so, to calculate the area of the box I have to multiply its side by itself ... that is $(24\text{-}2\times A)^2$ ... and I multiply this by the height ... that's it*".

FORMAT 4: Summary of Jenny's work

Work sheet	Worksheet content	Expression produced
31.	My grand father owns a hardware store. In helping him I programmed my calculator so that every time that some amount of wire is sold the program tells you how much wire is left. The table below is an example of how my program works. Can you guess what is it? **Sold** 1.7 2.4 3.1 4.06 5.2 **Left** 8.3 7.6 6.9 5.94 4.8	?→A: 10–A
32.	**32.1** Can you program the calculator so that it duplicates the following table? **Input** 1.3 2.5 3.8 4.4 5.9 **Output** 18.7 17.5 16.2 15.6 14.1 **32.2** What happens when have as input a negative number?	?→A: 20–A "*Instead of taking A away from 20 the program adds A on 20*"
33.	I have some pieces of wire, all are of length 16 cm. I want to cut them all into two smaller pieces in different ways, for example, 12 cm and 4 cm, 11 cm and 5 cm, and so on. Can you program the calculator so that if I input the length of one small piece it prints out the length of the other?	?→Z: 16–Z

Work sheet	Worksheet content	Expression produced
34.	I want to make a box with a square piece of cardboard. I can make the box by cutting squares off the corners and bending up the pieces that are left jutting out. The **base** and **height** of the box, are determined by the length of the sides of the squares I cut off. Figures 1 and 2 show two possible ways of making the box. Can you program your calculator so that it allows to calculate the volume of any box I could build?  	?→A: (24−A×2)^2 ×A
35.	Program your calculator so that it duplicates the table below. **Input** 1 2 3 4 5 **Output** 0 −1 −2 −3 −4	?→A: A×−1+1
36.	Program your calculator so that it produces the table below. **Input** 1 2 3 4 5 **Output** 4 9 14 19 24	?→A: A×5−1
37.	Program your calculator so that it produces the table below. **Input** 1 2 3 4 5 **Output** 0.5 −0.5 −1.5 −2.5 −3.5	?→A: A×−1+1.5
38.	Program your calculator so that it produces the table below. **Input** 1 2 3 4 5 **Output** 8.5 6.5 4.5 2.5 0.5	?→A: (A×−1) − A+10.5
39.	Program your calculator so that it duplicates the table below. **Input** 1 2 3 4 5 **Output** 0 0 0 0 0	?→A: A×0
40.	Program your calculator so that it produces the table below. **Input** 1 2 3 4 5 **Output** −1 −2 −3 −4 −5	?→Z: Z×−1

FORMAT 5: Discussion of Jenny's work

Jenny completed the five worksheets included in this format. In the case of inverting rules of the form f(x) = ax+b her first trials consisted of simply inverting operations in the order in which they appear. After doing this she adjusted the expression by adding or subtracting a constant (for example, she inverted A×2−1 as A÷2+0.5, worksheet 44). In the same worksheet she produced the inverse program by inverting the whole structure of the expression, for example, (B−1)÷3 to invert B×3+1. This suggests that delivering the task by giving the program expression instead of a table and including numbers more difficult for her to adjust her first trial, led Jenny to abandon her initial strategy and take up a more general one. From then on she used this new strategy to work out the tasks (see worksheet 45). This point is taken up in more detail in interview 2.

Syntax: Jenny's notion of using of parentheses

Jenny managed to use parentheses herself for constructing the inverse of an algebraic expression (tasks 44.2, 44.3 and 45.3). Jenny's work illustrates how her pragmatic approach to inverting linear functions seems to scaffold her to reach a higher level in algebraic manipulation. This is exemplified by Jenny's production of inverse functions of the form ax+b. This point is further discussed next.

Pragmatics: Jenny's approach to inverting linear functions

Throughout Format 1 to Format 4 Jenny worked with tables as a clue giving sources to construct an arithmetic procedure which she then described by means of the calculator language. But this strategy did not lead her to inverting algebraic expressions of the form ax+b. For example, Jenny showed herself to be able to obtain the underlying pattern suggested by the numbers in a table and to describe it using calculator language, like A×2−1 (see worksheet 44 below). But she could not have produced the inverse rule (A+1)÷2 using the same strategy. What she did was the following. Since she knew it had to be the inverse rule she started by simply inverting the relevant operations (A÷2+1). Then she ran this program, saw that it did not work and managed to make it give the desired outcomes (A÷2+1+0.5).

She got an idea of how to use parentheses once her first strategy proved not to be enough to invert programs like B×3+1 (see worksheet 44, task 44.3). Jenny's new strategy was based on the metaphor of "*doing first what was the last and vice versa*" (see interview 2). This was the key point that made her link the task of inverting a program with her notion of parentheses as symbols that serve "*to say to the calculator what to do first*".

FORMAT 5: Summary of Jenny's work

Work sheet	Content						Program produced	Inverse program
41.	Input	10.4	16	19	23.5	37	?→A: A+5.5	?→A: A−5.5
	Output	4.9	10.5	13.5	18	31.5		
42.	Input	11.4	19	23.1	38	50	?→A: A−6.1	?→A: A+6.1
	Output	17.5	25.1	29.2	44.1	56.1		

Work sheet	Content	Program produced	Inverse program
43.	43.1 Input 0.13 0.17 0.65 3.8 9.28 Output 0.26 0.34 1.3 7.6 18.56 43.2.Program your calculator so that it produces the inverse as M×3. 43.3.Program your calculator so that it produces the inverse as N×1.5.	?→A: A×2	?→A: A÷2 ?→M: M÷3 ?→N: N÷1.5
44.	44.1. Input 3 7 10 11 15 Output 5 13 19 21 29 44.2.Invent a program so that it "undoes" the one you have just found. That is, that produces the following table: Output 5 13 19 21 29 Input 3 7 10 11 15 44.3.Can you type a program so that it undoes the program B×3+1?	?→A: A×2−1	?→A: A÷2+0.5 and ?→A: (A+1)÷2 ?→B: (B−1)÷3
45.	45.1. Input 2 5 7 8 10 Output 4 25 49 64 100 45.2.Invent a program so that it "undoes" the one you have just found. 45.3.For the following programs construct one which undoes each of them. ?→A: A×1.5+1 ?→K: 0.5×K−1 ?→X: 0.25×X+2 45.4.Did you find a method to undo programs? Say what it consists on.	?→A: A^2 *"To use the inverse operations. Sometimes is necessary to put in parentheses".*	 ?→A: √A ?→A: (A−1)÷1.5 ?→K: (K+1)÷0.5 ?→X: (X−2)÷0.25

INTERVIEW 2: Discussion of Jenny's work

- The interview was concerned with tasks that rely on the notion of algebraic equivalence. These tasks were the following: (a) Transforming an expression to obtain another given expression, (b) Simplifying linear expressions, and (c) Inverting a given program. Jenny's reactions to these questions are discussed next.

Semantics: Jenny's notion of algebraic equivalence

Our data confirm that the notion of algebraic expressions as devices to describe and carry out arithmetic procedures promotes a strong relationship between semantic interpretation and syntax transformation of algebraic expressions. Jenny's strategy to cope with algebraic transformation illustrates this. It consisted of clearly distinguishing the role played for each of the expressions involved: the one to be transformed and the one which plays the role of target. Then she gave a value to the variable in the target expression. Having this in mind she went on to transform the other expression so that it gave the same value as the target one. That is, Jenny needed a semantic analysis of the algebraic expression to make sense of the question of transforming, then when Jenny grasped what the question of transforming was about, she started to extend her initial strategy to operate on the variables.

Jenny worked out successfully these kind of items in interview 1 (20 days before). In the second interview she showed she had not yet developed a systematic way to face the task (in fact she was not asked about it again until this interview). Nevertheless she had made remarkable progress. In this interview she dealt with the question of transforming, for example, 15×A into 2×A, by operating on the algebraic expression instead of exploring with specific values as she did in the first interview. Although she could not complete the task in this way her attempts show a higher level of understanding of what these algebraic expressions mean. What follows illustrates this: she first tried 15×A÷A×2 "*because 15×A gives any number ... as I don't want this I remove everything ... well, I divide it by A, it removes everything ... then I multiply by 2*" (<u>I2: 4</u>). After a number of failed trials the question was changed to transforming 15×A into 14×A. She did it correctly (15×A–A): "*I remember, that was the way I did it before*" (<u>I2: 18</u>). After this she answered successfully even more difficult questions, for instance, transforming 10×A+5×A into 18.3×A and 12×B+5×B–2×B into 4×B (<u>I2: 41-47</u>). It is worth noticing that she worked out the last questions first by operating on the coefficients, then she checked her responses by running the programs for specific values. This makes evident the role of the calculator in giving feedback to the child's conjectures.

Simplifying similar terms is another relevant issue which relates to the notion of algebraic equivalence. This matter was implicitly involved in the question of transforming an expression with two or more terms into an expression with only one term (i.e. transforming 10×A+5×A into 18.3×A). To do this Jenny simplified 10×A+5×A to 15×A in order to find the missing term 3.3×A. She did this by mentally operating with the expressions but she could not proceed in the same way when transforming 12×B+5×B–2×B into 4×B, apparently this was due to its complex structure. The following extract illustrates this. Her first reaction was to say "*It might be ... getting the result of all this*

(12×B+5×B−2×B) ... If B were 1 it would be 15 ... It multiplies by 15! Then it must be 12×B+5×B−2×B−(B×11)" (12: 47-54). The episode shows that she is developing a rule to simplify similar terms but when she meets a more difficult item she goes back to giving specific values to the variable which finally guides her to find a solution. At this stage the strongest resource she had for simplifying linear expressions was to explore the expression's numerical value. In this respect, her choosing of 1 seems to be her key strategy since it directly leads to simplifying the coefficients.

Pragmatics: Jenny's approach to negative numbers

The use of negative numbers was abruptly introduced in Format 4 (worksheets 32, 33, 35, 37, 40). The aim was to observe the extent to which the pupils, being supported by the calculator, may develop some notions about operating with such numbers. Jenny appeared to be one of those children to whom the topic of negative numbers was of special interest. The following episode shows the extent to which she had gained confidence in working with negative numbers. She was asked to transform 15×A into 25×A. She had the program 15×A−(A×10) on the calculator screen (she made it to obtain A×5) and just changed the expression between parentheses, she thought of it for a moment and typed 15×A−(A×−10). She explained this as follows: "*because if I take A×10 away from 15×A I get A×5 ... but, if I want to add I put −10*". Some moments after she said "*it might be done in other way as well ... 15×A+ A×10*" (I2: 32). The episode shows that she unwittingly restricted herself to a more difficult situation, like solving an equation of the form 15x-bx=25x where the unknown is b. She could finally manage the situation on the basis of her previous experience with negative numbers (see worksheet 32).

Pragmatics: Jenny's approach to inverting linear functions

The interview questions were focused on how Jenny began to use parentheses to find the inverse programs (see worksheets 44 and 45). In the interview she showed she had mastered how to invert linear functions of the form f(x)=ax+b, but she failed in doing this with the expression A−4×5 where awareness of priority of operations is needed. As we will see next, priority of operations plays a crucial role in the child's learning process towards formalising her methods. In that process she had to make a shift from constructing a symbolic expression by copying step by step a number-based model to globally visualising the structure of the whole symbolic expression so as then to operate on it. The episodes cited below suggest on the one hand that priority of operations needs to be met several times before the child gets to be aware of its necessity. On the other hand they provide evidence for the value of calculator feedback, it seems that the child could hardly find by herself why her responses are wrong working with no computing support.

Jenny explained how she built the program (A+1)÷2 as the inverse of A×2−1: "*I first tried A÷2+1 ... I saw it didn't work, then I adjusted it until get the outputs I had in mind, I got A÷2+0.5 ... Later I noticed that it was easier to put A+1 between parentheses, then divide it by 2*" (12: 68) ... ... *It is fully inverting the program ... it is not only inverting operations but putting in parentheses ... I have to do it all the way round ... because to invert a program I do first what I did last and vice versa*" (12: 102).

After this she showed that she was able to get the inverse of programs with the same structure as A×2−1 but she failed with the expression A−4×5. In the previous questions she was able to find the inverse of a program at a glance. In the case of A−4×5 Jenny hesitated: "*Is it A÷5+4? ... Should I use brackets*?" Then she typed the program A−4×5 and ran it for A=10. She got −10 but expected to get 30, "*because 10-4, 6, 6 times 5, 30*" (12: 106-108). Jenny coped with this question by finding out why the calculator output was minus 10: "*I don't know what is going on ... It first should take away 4... No, no ... That's it! It takes 4×5 away from 10*!" (12: 114). Then she typed A+4×5 as the inverse of A−4×5. Jenny worked on the expression 4×5 as a whole entity, she did not read 4×5 as 20. This shows that Jenny analysed the structure of the expression rather than exploring it with specific number values.

The above episode illustrates how, once the child has grasped what algebraic transformation is about, she is in a position to gain a deeper semantic conception of algebraic expressions promoted by tasks based on symbolic manipulation.

PHASE 3: Jenny's entry into problem solving

This section reports the work done by Jennifer in Format 6 and interview 4. Similarly to the sections before, Jenny's written work will first be discussed, then a summarised transcript of her work is presented to provide evidence for the analysis done in the preceding discussion. Finally Jenny's work during the third interview is analysed.

FORMAT 6: Discussion of Jenny's work

Format 6 presents three types of problem situations which require the child to formulate general relationships between quantities and represent such relationships using calculator language to negotiate solutions for particular cases. That is, to confront these problems using calculator language, the child has to use a function to model the problem and then to look for a particular case of that function (equation) to answer particular questions. One type of these problems consists of situations which require the child to answer questions about number sequences presented by means of figurative patterns. The second type of problems requires the child to use calculator language to represent and compute either the perimeter or the area of rectangular shapes described by general relationships between their sides. The third type of problem consists of situations that require the child to establish general relationships among quantities linked by percentual relationships. Jenny completed the ten worksheets included in this format. Her work is discussed in more detail below.

Syntax: Jenny's notions of using parentheses and priority of operations.

She used parentheses correctly every time they were needed, see, for example, worksheets 48, 49, 50, 51, 52, 53, 54 and 55. It is worth noticing that in worksheets 49, 50, 51,53 and 55 she correctly produced nested expressions. Jenny's confident use of brackets seems to be due to the fact that she approached on her own to that kind of expressions (see Format 2). Jenny's work suggests that an spontaneous approach to the priority of operations favour that children gain a better understanding of the function of brackets. Diego's case seems to confirm this conjecture, because it took him longer to appraise brackets as tools to break down the established order of arithmetic operations. As was discussed earlier in this chapter, Diego was guided by the researcher so that he met expressions that needed the use of parentheses. Though he understood the function of brackets he frequently needed the teacher to intervene in order to make him realise what parentheses serve for and incorporate their use into his every day computing routines.

Semantics: Jenny's notion of algebraic equivalence.

In worksheet 48, Jenny made the program $(A+2)^2-(A\times A)$. Then she engaged in inverting it to complete the table for the cases where outputs were given. She could not do this but found another way of interpreting the number pattern and produced the equivalent program $A\times4+4$. She accepted the equivalence "*because both programs output the same values for same inputs*".

The following episode illustrates how Jenny has extended her notion of equivalence to simplifying similar terms. Jennifer produced the program $((A\times3)\times2+(A\times2))\times53$ (worksheet 49). When giving feedback the teacher included a note wondering if that program might be expressed in a shorter form. She simplified it to $((A\times6)+(A\times2))\times53$, then as $(A\times8)\times53$.

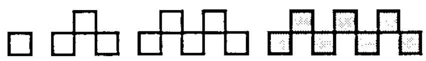
Pragmatics: Jenny's approach to inverting linear functions

Jenny's work in this format shows that she has gained enough confidence both in looking for equivalent expressions and in inverting linear expressions. In worksheet 48 (see Summary of Jenny's work below) she produced the expression $(A+2)^2-(A\times A)$. As could be seen the structure of the expression is quite sophisticated, that is to say, she really struggled to produce such an expression. Nevertheless, since $(A+2)^2-(A\times A)$ was not suitable for her to calculate the inverse values when the outputs were given, she went on and found an equivalent linear expression which she knew she would be able to invert: $A\times4+4$. Then she found two equivalent rules that reverse this expression: $A\div4-1$ and $(A-4)\div4$. She accepted the equivalence because both programs output the same values for the same inputs. This point is further discussed in the next section.

Pragmatics: The role of the context

The geometrical-based number patterns influenced Jenny's work. Earlier, Jenny had already worked out the rule L×2−1 (worksheet 4) where the clues were given by a table. There, Jenny's strategy was to investigate what operations should be carried out with any input number in order to obtain the associated output.

This contrasts with the way she worked out worksheet 4 where the function rule was the same as in worksheet 46. Here she used a different strategy; the outputs (figurative patterns) suggested to her a decomposition of the output number, for example, 5 as 2+3, 7 as 3+4, and so on. Thus, she related this to the number of the figure in the sequence.



Her explanation illustrates this: "*I realised that in shape 4 there are 3 squares above and 4 below ... that is, above, there is one square less than the number of the shape and below there are exactly as many squares as the number of the shape. So, I multiply the number of the shape by 2 and take away 1, which is the same as adding the number of the shape to itself but taking away 1*". She used a similar strategy in worksheet 48. It could be seen that she first tackled the question by taking away the area of the grey square from the area of the whole square (fig.1): $(A+2)^2-(A\times A)$. When trying to find a different way of expressing the rule she saw the shape as "a cross" (fig. 2) which allowed her to count the number of squares surrounding the grey square, then added the four squares on the corners: A×4+4.

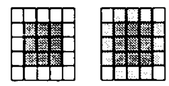
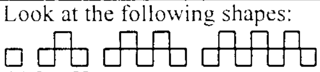
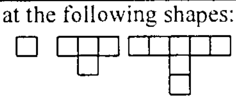
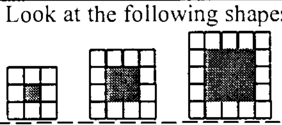


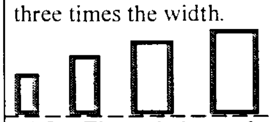
fig.1 fig. 2

FORMAT 6: Summary of Jenny's work

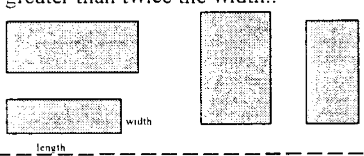
WS	Problem situation "Figurative patterns"	Jenny's responses
46.	Look at the following shapes: 	
	46.2. How many squares are needed to build up the shape that goes in the 17^{th} place?	33
	46.3. How many squares are needed to build up the shape that goes in the 100^{th} place?	199
	46.4. Explain how you reasoned to answer the questions above.	"*I realised that, for example, in shape 4* (the shapes were not numbered, she has this in her mind) *there are 3 squares above and 4 below. I mean, one square less than the number of the shape. Below there are exactly as many squares as the number of the shape. So I multiply the number of the shape by 2 and take away 1 from it, which is the same as adding the number of the shape to itself but taking 1 away*"

<table>
<tr><th>WS</th><th>Problem situation
"Figurative patterns"</th><th>Jenny's responses</th></tr>
<tr><td></td><td>46.5. Can you program your calculator to complete the following table?
<table><tr><td>Place</td><td>48</td><td>75</td><td>123</td><td>176</td><td>206</td><td>254</td></tr><tr><td>No. of squares</td><td>95</td><td>149</td><td>245</td><td>351</td><td>411</td><td>507</td></tr></table></td><td>Completed without teacher's feedback
?→A: A×2−1
Completed correctly (Jenny's answers in bold)</td></tr>
<tr><td>47.</td><td>Look at the following shapes:

47.2. How many squares are needed to build up the shape that goes in the 9th place?</td><td>25</td></tr>
<tr><td></td><td>47.3. How many squares are needed to build up the shape that goes in the 17th place?</td><td>49</td></tr>
<tr><td></td><td>47.4. Explain how you reasoned to answer the questions above.</td><td>"I multiply the number of the figure by 3 and take away 3 from this result. It gives the number of squares of this shape. I tried it out with all the others and it works".</td></tr>
<tr><td>48.</td><td>Look at the following shapes:
</td><td></td></tr>
<tr><td></td><td>48.2. How many squares are needed to build up the shape that goes in the 27th place?</td><td>112</td></tr>
<tr><td></td><td>48.3. How many squares are needed to build up the shape that goes in the 40th place?</td><td>164</td></tr>
<tr><td></td><td>48.4. Explain how you reasoned to answer the questions above.</td><td>"I added 2 to the number of the figure and raised to square and I take away the number of the figure multiplied by itself. This gives the result required".</td></tr>
<tr><td></td><td>48.5. Can you program your calculator to complete the following table?
<table><tr><td>Place</td><td>48</td><td>75</td><td>123</td><td></td><td></td><td></td></tr><tr><td>No. of squares.</td><td></td><td></td><td></td><td>704</td><td>772</td><td>840</td></tr></table></td><td>?→A: $(A+2)^2-(A\times A)$

Since she could not invert the program above, she engaged herself in constructing a new program she was able to invert. She got the program ?→A: A×4+4. The programs below are the inverse ones she did:
?→A: A÷4−1
?→A: A÷4−4+3
?→A: (A−4)÷4</td></tr>
</table>

<table>
<tr><th>WS</th><th>Problem situation
"Rectangular shapes"</th><th>Jenny's responses</th></tr>
<tr><td>49.</td><td>The windows (below) have different dimensions but in all of them the height is three times the width.
</td><td></td></tr>
<tr><td></td><td>49.2. The windows frame is made of wood whose cost is $53 per metre.
a) How much does it cost a window frame whose width is 1.5 meters?
b) What did you do to answer the question above?</td><td>$ 636.00

"I first multiplied 1.5 by 3 in order to get the height. I then calculated the perimeter, then multiplied this result by 53".</td></tr>
<tr><td></td><td>49.3. Can you program the calculator to obtain the cost of any window frame?</td><td>?→A: ((A×3)×2+(A×2))×53
Later she was encouraged to make this program shorter. Jenny did the following:
?→A: ((A×6)+(A×2))×53
?→A: (A×8)×53</td></tr>
</table>

WS	Problem situation "Rectangular shapes"	Jenny's responses
50.	The windows (below) have different dimensions but in all of them the height is 50 cm less than three times the width. 	As in worksheet 49 she was asked to program the calculator to obtain the cost of any window frame. She answered incorrectly. The program she used was: ?→A: ((A×3)−0.50)×62. She missed adding the width to the height and doubling the sum.
51.	The tabletops (below) have different dimensions but in all of them the length is 1 metre greater than twice the width.. 	
	The table top is made of wood which costs $155 per square metre Can you program your calculator to obtain the cost of any table top?	?→A: ((A×2+1)×A))×155 She also correctly used this program to complete a table which involved "inverse values".
54.	A number of pieces of land are for sale. They all have the following characteristics: the length is 30 meters greater than twice the width.	
	54.1. Mr. Pérez needed 132 metres of wire fence to fence his land. What are the dimensions of his land?	length: 9 width: 48 (Tasks 54.2 and 54.3 are similar to this one).
	54.2. Mr. González bought a piece of land whose width is 76 meters. How many meters of wire fence does he need?	668 meters
	54.3. Explain how you reasoned to answer the questions above.	*"I took away 60 from the number that you gave me then I divided it by 8. This gives the width. To obtain the length I multiplied the width by 2 and added 30 to this".*
	54.4. Did you program the calculator to solve these problems?	*"Yes, the program I used is* (((A-60)÷8)×2)+30"
55.	A rectangular piece of land can be limited as shown in the figure. 	The land-holder wants to take advantage of having the stream to limit one side of the land. Therefore he has only to fence the other three sides. The total facing available is 100 metres. He'd like to do it so that the land has the maximum area which depends of the measure of the land sides. To do this he will try with the following table:

Large side		60		70		65	58	55.5	54.8	49.7	She completed the table.
Short side	30		10		8						
Area											

WS	Problem situation	Jenny's responses
	55.1. Can you program the calculator to complete the table above?	?→A: ((100-A)÷2)×A
	55.2. What are the measurements that make the land have maximum area?	*Short side: 50 m (49.999...)* *Large side: 25 m* *Area: 1250 square meters.*

<table>
<tr><th>WS</th><th>Problem situation
"Percentages"</th><th>Jenny's responses</th></tr>
<tr><td>52.</td><td>The Music Centre is having a Special Sale: All records 15% off.
The discount will be applied on the labelled price.</td><td></td></tr>
<tr><td></td><td>52.1. Can you complete the following table?
<table>
<tr><td>Label Price</td><td>$ 34</td><td>$ 18.75</td><td>$ 126.50</td><td>$ 28.50</td><td>$ 150</td><td>$ 72.35</td><td>$ 29.40</td></tr>
<tr><td>Discount</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Special Price</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td><td>She completed the table correctly.</td></tr>
<tr><td></td><td>52.2. Can you program the calculator so that it prints out the Special Price every time you input the Label Price?</td><td>?→A: A–(A×0.15)</td></tr>
<tr><td>53.</td><td>A Book Store is having a Special Sale: All titles 25% off.
The discount will be applied on the labelled price.</td><td></td></tr>
<tr><td></td><td>53.1. Can you complete the following table?
<table>
<tr><td>Label Price</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Amount Discounted</td><td>$18.75</td><td>$6.00</td><td>$9.00</td><td>$21.50</td><td>$8.75</td><td>$6.50</td><td>$11.5</td></tr>
<tr><td>Special Price</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td><td>She completed the table correctly.</td></tr>
<tr><td></td><td>53.2. Program the calculator so that it prints out the Special Price every time you input the Amount Discounted.</td><td>?→A: A×3</td></tr>
<tr><td></td><td>53.3. Program the calculator so that it prints out the discounted amount every time you input the Label Price</td><td>?→A: (A÷25)×100, or
?→A: A×4</td></tr>
</table>

INTERVIEW 3: Discussion of Jenny's work

The interview was concerned with the following issues:

- Interpreting algebraic expressions used to denote measurements in diagrams
- Simplifying linear expressions
- Inverting linear functions
- Working out problem situations involving generality by expressing number relationships algebraically.

Syntax: Jenny's notion of using of parentheses

The episode about the rectangle C+2 by 5 provides evidence of Jenny's correct use of parentheses. However, when she was asked to transform 20×D in order to make it equivalent to 17×D she showed a lack of awareness of priority of operations while working with paper and pencil. When the question was asked she immediately answered: 20–3×D "*because 20 minus 3, 17, then 17 times D*" (I3: 131-134). Then it was suggested that she typed this in the calculator. She said (without using the calculator):

"*It would give 3 times the number I input for D, then it would do 20 minus this ... I need to put in parentheses*" (I3: 135-136). Then she went to the calculator and typed (20–3) ×D. Next, she was asked to find another way of getting 20×D equivalent to 17×D. She verbally proposed 20×D–3, when explaining what the program would do she let D=5 and found in this way what was wrong, finally she got the expression 20×D–3×D (I3: 139-145). On the one hand the episode enhances the role of the calculator as source of feedback that helps the child gain awareness of syntax conventions. On the other hand, the episode suggests the need of combining more frequently the work within calculator and paper and pencil environments.

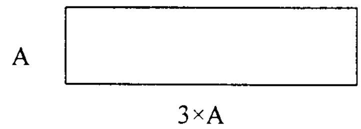
Semantics: Jenny's notion of algebraic equivalence.

Twenty days previously Jenny had successfully answered questions about simplifying similar terms. To probe this further she was asked again to do this in this interview. At this time Jenny's responses showed that she had gained acquaintance of simplifying linear expressions, for example, during the interview she made wrong conjectures but corrected them by herself.

The way she worked out this type of question suggests that her experience of using calculator programs enabled her to cope with algebraic simplification, and provides her with a referent that allows auto correction. Here, it seems worth remarking on a couple of points. First, that her correct use of parentheses relies on a good understanding of the priority of operations. Second, that it was still necessary to suggest to her to simplify an expression, otherwise she does see the need to do this. Although it may be obvious to have expected this to happen due to her incipient algebraic experience, Jenny's unawareness of the usefulness of simplifying leads one to think that simplification of similar terms should be taken within a context where algebraic transformation best shows its potential (this point is reviewed again in the section assigned to inverting linear functions).

It is also relevant that Jenny's reactions suggest that the number-based approach to simplifying seems to have prepared her for understanding the distributive law. In fact, Jenny seems to be stating this law when she explains her strategy to simplify similar terms. The following extract illustrates this.

She typed the program (A×2)+(A×3)×2 to calculate the perimeter of the rectangle shown on the right. Then she was asked whether the program may be expressed in a shorter way or not. She answered by simplifying it as (A×2)+(A×6) and said: "*it could also be A×12*" (I3: 43-44). But immediately she corrected this saying: "*A×8 ... I thought it was 6 times 2 times A, but it does not work because it is A×2 first... then A×6 to make shorter (A×3)×2 ... then it gives A×8, because A times 2 plus A times 6 gives A times 8 ... well, it would be easier with numbers ... like 1×2 would give 2, and 1×6 would be 6, and 2 plus 6 gives 8 ... then, instead of adding once and multiplying twice, I can just do 1 times 8*" (I3: 48).



It is interesting to notice how differently Jenny reacts to algebraic transformation depending on the context. Jenny quite correctly worked out the item that was presented geometrically. However her work was not as fluent, and some times she became confused, when symbolic manipulation relies solely on her notion of algebraic expressions, even with items that seemed less complex than the one presented in the extract above. This suggests that, for this child, her notions of letters and algebraic expressions are better exploited being supported by diagrams involving geometrical notions.

Jenny's work confirms that the notion of letters as "representing any number" is the most powerful tool she has acquired to deal with this situations. In fact, this notion seems to be what leads her to substitute letters for specific values. The research data shows that Jenny can always resort to number substitution both to work out simplifications and, which seems more important, to validate/correct her answers, despite of not having developed solid algorithms to reduce similar algebraic terms. Here, again, the relevant role played by the calculator environment is critical in supporting the child's reasoning whether as clue giving source or as feedback supplier. The following interview excerpts illustrate this.

Jenny was asked if the program 3×A+4×A+A could be made shorter. By visual inspection she got 13×A. While explaining her answer she changed it to 12×A+A, "*because 3×4=12, 12 plus A ... but I cannot add A ... for example, 12×2, 24, 24+2, 26*" (I3: 147-152). Then she was asked about another way of checking the correctness of this answer, to which she replied "*the two programs have to give the same*". She let A=2, mentally obtained 16 as the number value of 3×A+4×A+A, and noticed that "*adding 10 to this program the two programs would be the same, but only if A=2*" (I3: 153-158). From this she found where and why she had made an error and produced a correct answer for this question as well as with other even more complex expressions than the first.

Pragmatics: Jenny's approach to inverting linear functions

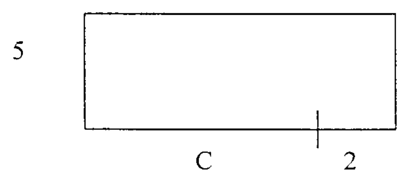
In the second interview (20 days before) Jenny showed that she had developed a consistent strategy to invert linear functions. In the present interview it was intended to inquire whether the experience she had at that time may have helped her gain awareness of both simplification and inversion not just as syntactic transformations to be performed but as tools to face problem situations involving algebraic expressions. So, she was asked to invert the program 4×D+7×D which implies simplifying as a previous step to inverting it.

Despite the fact that Jenny had clearly shown that she was able to simplify even more complex expressions than 4×D+7×D, at the moment of having as a goal obtaining its inverse function, she resorted to giving values to the literal symbol. Jenny's reaction suggests that her strongest strategy to face algebraic transformation is to explore the behaviour of the expression by substituting the variable for particular values. In this the way she found that 4×D+7×D is equivalent to D×11, then she found its inverse, D÷11 (I3: 103-114).

The above episode indicates that, in fact, substituting was the building block from which Jenny generated initial rules to simplify algebraic expressions. Nevertheless, rules like the one we are discussing, need time to be established as tools. Even the most simple rules, like dividing as the inverse operation to multiplying seem to need to be intensively used before the child becomes able to master them. Very simple details may disturb children's attention and inhibit their progress, like the order in which the numbers must be input to perform a division. All this allows us to see the critical role of continuous feedback from the calculator. What follows exemplifies this.

Once Jenny had found that 4×D+7×D=D×11, she immediately said the inverse was "*dividing by 11*", but she typed the program 11÷D . Then she realised that the program did not output what she expected. She had to make some effort to find what was wrong. She finally explained her mistake "*I forgot that in the division the numbers cannot change places as opposite to the case of multiplication*" (I3: 116-128).

Pragmatics: Jenny's use of algebraic language to negotiate problem solutions.
Jennifer's answers suggest that the close relationship between the symbolic expressions used for programming the calculator and its function as a means of numerical computing, enables her to algebraically represent general procedures despite the fact that she conceives calculating only when it is to be performed with numbers. Jenny's way of proceeding evokes children's non acceptance of unclosed algebraic expressions, but Jenny had no difficulty in formalising her methods. What follows aims to clarify this.



Jenny answered the question of calculating the perimeter of the rectangle shown in the diagram: "*it is not possible to calculate the perimeter of such a rectangle, is it? ... if it were a program ... I can do it with a program*". Then, she quite naturally typed the program (A+2)×2+5×2 and explained: "*this (pointing at (A+2)×2) is for the sides below and above altogether, and this (pointing at 5×2) is two times the height*" (I3: 1-6). Then she was asked to build a program to calculate the area of such a rectangle, she typed (A+2)×5 and commented: "*First, I add 2 to the number I put here (pointing at A) to make the program give the base* (of the rectangle) *then I multiply this by 5, which is its height*" (I3: 7-10).

The episode illustrates that the calculator's programming mode provides a fine-tuning link between Jenny's arithmetic way of reasoning and her initial steps towards using algebraic code. Her reactions show that she is actually working with unclosed algebraic expressions being supported by the underlying notion of numerical computation. That is the only apparent reason why she is able to cope with the question. This suggests that Jenny's experience in expressing the rule that governs a number relationship being guided by an input-output table seems to be a good precursor to the ability to algebraically represent the structure of problems.

Semantics: Jenny's notions of literal terms and algebraic expressions.

The type of question such as the "rectangle C+2 by 5" had never been used before in this study, therefore it was thought necessary to inquire whether the children could make sense of it. Thus, Jenny was asked what the diagram might be about. She answered: "*It is wanted to know how much this measures (pointing at the C) ... so as to obtain the area*" (I3: 1-2). Can you calculate the perimeter of this rectangle? "*It is not possible, is it? ... but with a program*" (I3: 3-4). The episode suggests that the experience of using letters while programming the calculator helped her to develop an ability to see the particular in the general and vice versa. More specifically, her first reaction was to see the letter as representing a specific value but, without any apparent difficulty, she could make a shift from this to the notion of a literal symbol as representing a range of values when programming the calculator.

It also seems important that she used the letter A instead of C in the program she built to obtain the perimeter. This clearly shows that she has grasped that changing the literal symbol does not change the expression (Wagner, 1981) and shows the range of generality of Jenny's notion about letters as representing a range of numbers.

Another remark in this respect concerns the development of children's skills for interpreting algebraic expressions. The use of algebraic expressions both as a means to represent general relationships and as tools to perform numerical computing seems to allow her to acquire elements from which to read algebraic expressions. Though most of the time the child builds a programming expression by copying step by step an arithmetic procedure, the completion of this process comprises the formalisation of her method which seems to be the clue for her to acquire a global vision of symbolic expressions. Particularly, questions of the type of transforming expressions compels the child both to analyse step by step the algebraic expression and to look at the whole expression as an entity. In order to clarify this two interview extracts will be discussed next.

As was discussed earlier in this section, Jenny started to develop a misrule to simplify similar terms. She proposed 12×A+A as a simpler form of 3A+4×A+A because "*3 times 4 is 12 and it is not possible to add A*". Though wrong, this shows that she was working on the basis of a whole view of the expression. She debugged this error by giving specific values to the variable and comparing the outputs obtained from the simplified and the original expressions. After this she could easily simplify more complex expressions, for example, 5×B+3×B+B+2×B as 11×B explaining "*I added those numbers which are not B because it is not possible to add B*" (I3: 167-172) (it is obvious that somehow she added the coefficient of B to 5+3+2, otherwise she could not have got 11, but she cannot explain this). Then she was asked to simplify the expression 5×B+3×B+1×B+2×B. She reacted immediately "*it is the same as the one before because B alone is equal to 1×B*" (I3: 173-174). It seems that the only way in which she could get this is by having a global view of the two expressions.

After this she was asked to make shorter the program 5×B+2×A, she said: "... *shorter? ... It couldn't be shorter ... all the values are different here ... unless A and B had the same value ... or A was zero ... or B was zero*" (I3: 176-182).

It seems relevant that Jenny shows neither reluctance to accept unclosed expressions nor any tendency to conjoin terms which could have happened when she had been asked to simplify 5×B+2×A. All this seems to relate to the arithmetic notation used throughout the whole study and to the fact that the calculator leads her to express her reasoning in terms of arithmetic operations. This not necessarily happens within a paper and pencil environment, where the child may consider enough to express her reasoning in natural language. This point will be further discussed in Chapter 7.

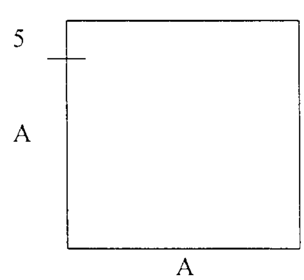
Another type of item related to interpreting algebraic expressions was the following: A pupil from another class says that $(A+B)^2$ gives the same as A^2+B^2. What do you think about this?

Despite the fact that the expression involves two variables and it was completely new to Jenny she did not show any hesitation either in making sense of the expression nor in finding a way to answer the question. Her answer was: "*He is wrong, because to square a number means to multiply it by itself, for example, if A was 1 and B was 2 we'd get 1+4, 5, here* (pointing at A^2+B^2) *and* 3^2, *9, there* (pointing at $(A+B)^2$)" (I3: 183-188). Then she was told that the pupil who said this said also that he had examples of his statement. After a moment of reflection she said, "*he might have been thinking of A and B having both zero as a value ... or A was zero and B was 1*" (I3: 189-192).

Pragmatics: Use of algebraic language to negotiate problem solutions.

Since questions of the type involved in this section were not posed to the children before, it is assumed that Jenny's responses rely solely on her classroom experience using algebraic language as calculator-language expressions. The question we will deal with next is to consider a problem situation under the criteria that the child does not have immediate available resources to confront it.

A diagram like the one on the right was shown to Jennifer. She was asked if the diagram provides her with some information. The point that will be focused on here is how the generality into which the diagram is embedded led this child to use the general expression A+5−A to answer what seemed a simple question: finding out the difference between the two marked sides in the shape.



Though incorrect at the beginning, Jenny was able to read the diagram relating metric relationships (A and A+5) to her geometric notion of square shapes. Jenny's first reaction was to say "*there is something wrong, it looks like a square, so it must have all its sides equal*" (I3: 12). Then she proposed to "arrange the figure" whether "*by taking 5 away or adding five to the other side because the A's have the same value*" (I3: 13-16). She was asked to suppose that the shape is not a square, and if she could say which is the greatest side. She did not have any difficulty in answering the question, "*because in both sides the A values the same ... let's suppose that A was 9, here the two sides are the same, and 14 is greater than 9*" (I3: 20-22). Here we get to the point that seems relevant, she was asked to say exactly how much greater is one side than the other. After some reflection she said "*in the program A+5 ... Oh! A+5–A would be the difference ... it depends on the value you give to A*". The interviewer wrote down A+5–A and was interrupted by Jenny who said "*No, it would always give 5 ... doesn't matter what value you input for A you would do nothing*" ... In terms of numbers, what does A–A mean? ..."*Zero, zero plus five gives five*" (I3: 23-36).

On the one hand, the episode shows that despite the fact that the diagram provides enough information to answer that question Jenny did not make any connection between it and the question she was asked, which suggests that some care is needed to rely on giving information by means of diagrams. On the other hand, it is relevant to notice that when calculating was required she used the calculator language to negotiate a problem solution, to do this she produced the expression A+5–A without being guided by any number-based pattern, this suggests that she was working purely with unclosed algebraic expressions. Though she might have done it with the idea of programming the calculator to numerically explore the situation, she finally was able to find a solution apparently by simple inspection of the symbolic expression.

A second item was given to observe Jenny's use of calculator language to negotiate problem solutions. The item was the following:

> Look at this list of numbers: 5, 9, 13, 17, 21, ... If I keep on writing down numbers in the list, will I get the number 877?

Jenny used the calculator to divide 877 by 4 and obtained 219.25, then she did it with paper and pencil. Jenny explained that 877 must be in the list "*because I noticed that for each number in the list 1 is left when you divide them by 4 ... so 877 must be in the list because if you divided it by 4 there is also 1 left*" (I3: 193-200). Then she was asked for the place in the list that 877 would have. She found it is "*219 ... 887 must appear in this place because 4×219+1=877*". After this she typed the program A×4+1 in order to show a way of producing any number in the list (I3: 201-208).

Pragmatics: Jenny's approach to expressing and justifying generality.

So far Jennifer's responses indicate that the experience she has had exploring and describing number patterns has enabled her to use the calculator's language to represent number relationships.

However, Jenny's work shows that she still needs more experience using calculator language before she becomes able to use it to justify generalisations. In fact, Jenny did not resort to using algebraic expressions spontaneously to confront justifying generality, she did it only when she was specifically asked to do so. Nevertheless, her answers show that she has made important progress because it seems unlikely that she might use algebraic language as an argument without being able to represent the relationships involved symbolically. To summarise, it could be said that the acquaintance she has developed with representing number patterns algebraically seems to be a good starting point for her to begin exploring the use of calculator language as a means of expressing and justifying generality. It seems promising that she needed less interviewer's help in the second item about generality than in the first one, which leads us to conjecture that Jenny is on her way to grasping how the calculator language may help her get fresh and more general information. The following two extracts are aimed at illustrating the above.

The following puzzle-like situation was verbally posed to Jennifer:

> Think of a number, add 10 to it and write down the result. Now take the number you thought of away from 10 and write down the result. Now add the first result to the second one ... May I try to guess the final result you got? It must be 20.

Then she was asked if she could find out why the final result can be predicted. Jenny seemed to have understood why 20 would always be the final outcome but, after some trials, she could not offer any intelligible explanation from generalising numerical examples and said "*you are using a trick so it always gives 20*" (I3: 49-58). Then she was asked to try to program the calculator so that it may help to explain that "trick". Without any apparent difficulty she typed the program (A+10)+(10–A) and was asked to observe the programming expression looking for an explanation of why it always gives 20. She was about to be told to abandon the task when she said "*that's it, you are doing nothing ... right there (pointing at the A's in the expression (A+10)+(10–A)) ... A minus A gives zero so it just leaves 10 plus 10, that is why it gives 20 ... does not matter what number you have thought of*" (I3: 59-68).

A second item was the following story-based situation:

> A pupil from another class says that every time he sums two consecutive numbers he gets an odd number. What do you think about this?

This time she attempted a general reasoning: "*I agree with him, because if you have two consecutive numbers one of them must be even and the other odd ... for the sum to be even you would need two even numbers... since you have one even and one odd numbers the sum must be odd*" (I3: 73-74). We should notice that if she had included in her reasoning the case of two odd numbers she would have had the skeleton of what we would accept as an indirect mathematical proof. After this she was asked to represent in general the sum of two consecutive numbers. Since she appeared not to make sense of this, the question was changed to asking for a program to represent the sum of two consecutive numbers. She immediately said (she did no type it): "*A plus A plus one*" and explained: "*because A is a number, then A+1 is the one which follows ... Let's suppose A was 4, 4+1 is the next number*" (I3: 77-80). When asked she easily expressed the sum of three consecutive numbers as A+A+1+A+2 because "*I just added A+2 to what I have already done*" (I3: 81-84).

It is worth emphasising that although Jenny could not make sense of the question posed in terms of generality this seems to be due to the novelty of the type of statement; her reasoning, though based on numbers, is founded on generality.

After this she was asked if the expression A+A+1 might offer some explanation for the sum of consecutive numbers. She said it did, because "*when the machine does A+A it is adding the same number twice, say, you input an even number, it does A+A and it's already an even number, say 4+4, 8, plus one, it gives an odd number, 9.*" What about if I input an odd number. ... "*Well ... A may be also an odd number ... say 3, 3+3, it is the same as the one before, because it gives 6, plus one 7 ... it always gives an odd number*" (I3: 87-90).

THE CASE OF JENNY: Concluding Remarks

This section aims to put forward some conjectures intended to explain how Jenny developed the notions and strategies discussed in the previous sections. In order to do this some episodes will be re-analysed in terms of some learning processes which may have occurred within the framework provided by the tasks which she confronted during the field work.

The potential of the calculator within a pragmatic approach

Jenny's mathematical attainment before and during the field work lead us to think that her algebraic achievements stem from her working experience within the calculator-based environment designed for the research. Jenny's mathematical attainment throughout the study provides evidence for the implicit conjecture around which this research was implemented:

> Computing devices which use a programming language a similar code to algebraic sign system can help create a mathematical environment which encourages the children to learn the algebraic sign system **as a language-in-use,** in a similar way as we learn the rudiments of the mother tongue.

In what follows this statement will be discussed in more detail.

Semantic notions developed by Jenny

The use of calculator language helped Jenny develop two basic notions:

i. Algebraic expressions as devices to describe and compute general arithmetic procedures.
ii. Letters as symbols that allow her to represent any number and serve to make the calculator's general procedures work.

Here there are two important points to discuss: how she learned about these notions and how she was able to use them as tools to express generality and negotiate problem solutions. In this respect the claim is put forward that Jenny's mathematical notions strongly relate to the way in which the tasks were designed.

Let us look at the tasks in more detail. They all are goal-oriented activities whose main characteristic consists of providing a numerical referent for the child to produce algebraic expressions. From her first encounter with calculator's language Jenny was required to formalise her method. For instance, to carry out the tasks in worksheet 4 (format 1) she was required to:

i. Find the underlying pattern shown in a table.

Input	1.1	2.5	3	4.3	5
Output	3.2	6	7	9.6	11

According to the research data, Jenny confronted this task by carefully inquiring what computations had been carried out with 1.1 to obtain 3.2. Then she had to check out if those computations allowed her to obtain 6 when applied to 2.5, and so on.

ii. Communicate to the calculator how to multiply any of these numbers by 2 and then add 1 to the outcome. This requires the child to produce a "one-piece" symbolic utterance so that it describes and carries out her reasoning.

The above process led the child to formalise her method. It was observed during the first three months of the school year that her spontaneous way of proceeding was not to express her reasoning as 1.1×2+1. She usually did 1.1×2, 2.2, then 2.2+1, 3.2. Jenny accepted the formality of an expression like A×2+1 just because it is the way that the calculator works. The research data indicates that she gained understanding of the generality of such an expression since she had constructed it. From this experience the child develops the notion for letters as symbols that "*represent any number*". This notion is extended beyond the set of numbers displayed in the table by requiring the child to fill in the blanks in a new table using the program she had built. There, it is particularly important that she needs to find the inputs when outputs are given. This makes the child carefully analyse the way her program proceeds and, consequently, think of the role played by the letter she is using. For instance, in the same worksheet she completed the table below using the program A×2+1 (Jenny's answers in bold type).

Input	1.3	2.8	14	50	81	274	**162**	**209.5**
Output	**3.6**	**6.6**	**29**	**101**	**163**	**549**	325	420

As has been said earlier, it was the teacher who introduced children to the calculator language. Worksheet 1 was used to exemplify this. Worksheets 2 and 3 helped children become familiar with programming operational details. From then on, tables were the only way used to encourage children to use calculator language. In fact, what children received were challenges to finding number patterns which they had to describe by means of calculator language. This was a key point in helping children develop skills to receive and produce calculator language utterances. Here, the game-like structure of the tasks also played an important role.

With time, Jenny's semantic notions allowed her to make sense of and successfully deal with algebraic manipulation, like inverting linear functions, constructing equivalent algebraic expressions and simplifying linear function's rules. Her work shows that she confronts symbolic manipulation by exploring the numerical behaviour of algebraic expressions (that is, as a semantic activity). Later on she started to construct initial syntax rules. An important point here is that this relationship between semantics and syntax allowed Jennifer to develop strategies to verify her incipient syntax rules.

Syntactic notions developed by Jenny

Jenny learned about priority of operations and use of parentheses while using the calculator's language. The case of Jennifer is unique in this respect, her fellow pupils had to be guided by the researcher so that they met expressions which made them realise the priority of operations. It was Jenny's intellectual curiosity which led her to realise that the calculator "*thinks differently from her*" (format 2).

Jenny gained awareness of the priority of arithmetic operations and use of parentheses for the only reason that it is the way the calculator works. Jenny's work shows that the calculator-based tasks create a milieu governed by mathematical rules that helped her accept their conventions. For this to happen, it was crucial that Jenny was able to carry out the relevant calculations in advance, otherwise she would not have noticed that the calculator proceeds differently. For instance, she thought of "*a program that first takes 1 away, then multiplies it by 3*" (see interview 1) and calculated by herself several input/output pairs. But she could not make the calculator work in that way. The fact that Jenny had met such conventions while trying to produce her own goals (see format 2) seemed to make her value the role of syntax rules. This suggests that more suitable tasks should be designed in order to favour such children's spontaneous approach to syntax.

Our data shows that Jenny's command of priority of operations plays a critical role both in decoding an expression like $A\times3+5+2\times A$ and in constructing rules for algebraic transformations. This relationship between algebraic manipulation and priority of operations is due to the fact that algebraic notation was introduced as a means of describing arithmetic procedures. In that fashion the child starts producing and reading algebraic expressions seeing behind them an arithmetic procedure. Then, being unaware of priority of operations is quite likely to result in misunderstandings. For instance, not to distinguish the terms $A\times3$, 5 and $2\times A$ in the above expression.

Strategies developed by Jenny

Jennifer's work shows that her strongest strategy to cope with algebraic transformation was numerical substitution. She used it wherever algebraic transformation was required: to operate on a given expression to make it equivalent to another expression, to invert function rules and to simplify algebraic expressions. Each of these cases are specifically discussed in what follows.

Jenny's approach to symbolic manipulation

The interview extracts cited earlier show that Jenny was able to cope with the task of transforming an algebraic expression to make it equivalent to another expression. Here we will discuss how she was able to do this and how a fusion of Jenny's semantic and syntax notions took place.

It is worth noticing that Jenny had just completed five sessions programming the calculator when she was confronted with this kind of algebraic transformation (interview 1). Her work shows that the strategy she used relies on her prior experience describing number patterns by calculator language. In order to make it clear let us analyse the question of transforming 4×B to make it equivalent to 3×B.

First she made sense of the question. That is, she clearly distinguished between the expression to be transformed (4×B) and the target expression (3×B). Then she let B=2 and produced 3×B=6. Having this in mind she put forth 4×B–2=3×B. Jenny did not accept this because "*it works only for B=2*". Up to this point it is clear that her strategy was based on having a numerical referent as a departure point so that she might compare both expressions by means of their numerical value. Then she tried with other values for the variable until she produced the expression 4×B–B=3×B, because:

"*4×B–2=3×B only works if B=2,*
4×B–1=3×B only works if B=1,
4×B–5=3×B only works if B=5,
then it must be 4×B–B=3×B".

This strategy is an extension of her previous experience with number patterns. The task's structure in that case was the same. She had a number to be transformed (input) and a target number (output). Having the latter in mind she looked for those computations that helped her transform the input so as to obtain the output. Then she made a first conjecture and tried it with other pairs of input/output. Once her conjecture proved to work she expressed it as a calculator's program. This indicates how her notion of algebraic expressions as devices to calculate enabled her to extend her experience to cope with new situations.

Later, when harder questions were asked, she started to sketch out syntax rules. For example, without having yet a systematic way of proceeding she was able to operate on 10×A+5×A to make it equivalent to 18.3×A, and 12×B+5×B−2×B into 4×B.

Jenny's approach to simplifying similar terms

Interview 3 shows that Jennifer was able to simplify similar terms. Here it will be explained in more detail how she was able to do that. Our first conclusion about this is that her incipient awareness of priority of operations was a key point for her to simplify similar terms. But there are also some experiences that seem to have mediated her learning.

The question about simplifying similar terms was given in the third interview. Before this she had some experiences that relate to algebraic simplification. Specifically, she had completed the worksheets about equivalence in format 3 and had confronted the questions about transforming algebraic expressions twice, first in interview 1 and then in interview 2. I consider that these antecedents play an important role in explaining Jenny's strategies.

For example, in format 3 (worksheet 21) she built the programs A×5−A, X×3+X, Y×2+(Y×2), C×6−(C×2) and D×7−(D×3) as equivalent to W×4. These were her first steps towards algebraic manipulation. Thus, when she was asked to simplify similar terms she had gained some acquaintance with algebraic manipulation. Her work confirms this as we will see next. Jenny faced quite naturally the question of transforming 10×A+5×A to make it equivalent to 18.3×A. To do this she **operated with** the expressions, she saw that "*10×A+5×A gives the same as 15×A ... it must be 10×A+5×A+3.3×A*" (interview 3). Her strategy reflects her prior experience of equivalence. Nevertheless, she did not face the question of simplifying 12×B+5×B−2×B in this way. Here she explored the numerical value of the expression. She let B=1, mentally operated and realised that "*it multiplies by 15*"[3]. We should notice that this was possible thanks to her awareness of priority of operations. It was this notion that enabled her to decompose this expression into "12×B, 5×B and 2×B". In this way she got 12×1+5×1−2×1=12+5−2=15, which she then related to 15×B as a simpler form of 12×B+5×B−2×B.

Jenny's work suggests a second conclusion: she has not yet developed a method for simplifying linear expressions. Up to this point she resorts to using a twofold strategy, when she was able to make sense of the expression, she proceeded adding up the coefficients. When she was not able to make sense of the expression (for example, 12×B+5×B−2×B), Jenny went on to explore its numerical behaviour in order to obtain further clues to guide her reasoning.

Jenny's approach to inverting linear functions

Jenny's strategy evolved according to the following pattern:

i. When a table and the program that produces it are both available she uses the program to "chase" the input by successive approximations. For example, to find the associated input to 325 (see table below) Jenny used the program A×2+1 as follows: since 163<325<549 she looked for an A value so that 81<A<274, say 100. Then she went on trying until she got 162.

Input	1.3	2.8	14	50	81	274	**162**	**209.5**
Output	**3.6**	**6.6**	**29**	**101**	**163**	**549**	325	420

ii. When just the program to be inverted is available, she inverted the operations in the order in which they appear, then she adjusts the result. For example, in worksheet 44 she inverted the program A×2−1 as A÷2+1. Then she ran the program and saw its outcomes exceed by 0.5 the results she expected. For example, A×2−1=5,

[3] She sometimes substitutes the variable for other values.

for A=3, and A÷2+1=3.5, for A=5. So, she adjusted the program taking 0.5 away (A÷2+0.5).

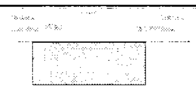
iii. When inverting operations proved not to be a suitable strategy, she found how to use parentheses to invert the expression. To do this Jenny related her notion of parentheses to her conception of inverting function rules. What follows aims to clarify this.

While working with the tasks in format 2 she realised that parentheses are symbols that "*... serve for the calculator not to perform the two operations at a time ... I mean ... first one then the other*" (interview 1). Later on, in format 5, she found that inversion consists of "*doing first what you did last and vice versa ... I have done it with the calculator ... it works*" (interview 3). These are the notions that enabled her to use parentheses when inverting linear function rules. For example, worksheet 44 required the child to invert the program B×3+1. She first tried inverting the operations (B÷3−1). When she ran the program the output numbers were too difficult to grasp because of the periodic expansion obtained when dividing by 3. Then she looked for another way to do it.

The above process provides a good example of how Jenny's command of the calculator's language evolved. It was Jenny's use of calculator language through a range of goal-oriented tasks which helped her to assign meanings to algebraic expressions and parentheses. Based on these notions she started making trials on symbolic manipulation like just inverting operations. Later on, she made her idea about inverting more concrete ("*we have to do it all the way round ... doing first what you did last ...*") by using the calculator language. This shows a higher level of linguistic competence: from describing arithmetic procedures to pure symbolic manipulation.

Jenny's approach to negotiating problem solutions

Jenny's classroom work shows the extent to which she was able to use calculator language in negotiating problem solutions. This section aims to discuss how she developed strategies to cope with problem solving. To do this we will carefully look at the sample of Jenny's work shown below.

WS	Problem situation	Jenny's answer
34.	Programming the calculator to build a square box with maximum volume (from a square cardboard with length 24 cm).	$(24-2\times A)^2$
49.	Programming the calculator to find the cost of a class of rectangular window frames whose height is three times its length.	((A×3)×2+(A×2))×53
50.	Programming the calculator to find the cost of a class of rectangular window frames whose height is 50 centimetres less than three times its length.	((A×3)−0.50)×62. She missed adding the width to the height and doubling the sum.
51	Programming the calculator to find the cost of a class of table tops whose length is 1 metre greater than twice its width.	((A×2+1)×A))×155
52.	Programming the calculator to find the special price of any merchandise when 15% is off and the regular price is known.	A−(A×0.15)
53.2	Programming the calculator to find the special price for any merchandise when the discount obtained (25%) in the purchase is known.	A×3
53.3	Programming the calculator to find the regular price when the discount (25%) is known.	(A÷25)×100, or ?→A: A×4
55.	Programming the calculator to obtain the a "three sides" rectangle with maximum area whose "three sides" perimeter is 100 metres. 	((100-A)÷2)×A "*(100-A) ÷2 represents the large side*"; "*(A×2) represents the two short sides*".

The algebraic representation of the relevant relationships in these problems involve a great level of difficulty, at least for a brief introductory study of algebra with 12-13 year old children. Situations like the ones in worksheets 34 and 55 usually appear in introductory Differential Calculus textbooks[4] (though these type of problems are also included in many secondary school textbooks, but not for the same purpose). Worksheets 49-51 require the child to algebraically describe sophisticated relationships like "the height is 50 centimetres less than three times the length" that usually appear in introductory algebra textbooks. Worksheets 51-53 require the child to use the notion of percentage so as to construct a program to compute the cost and the tax applied when its total is given. Additionally all the problems refer to a whole class of things, that is, they are modelled by functions. An important point here is that the children were not helped by the teacher to do these tasks. Thus, what they did stemmed from resources and strategies they developed through working with the calculator.

[4] For example, A. Cruse & M. Granberg, 1975. Lectures on Freshman Calculus. Addison Wesley Publishing Company, London.

To confront these tasks Jenny put into play every strategy and notion she had learned about. Jennifer's work suggests that what goes on in tasks like games describing number patterns by calculator language can tell us much about learning algebra. The point that will be addressed here is how Jenny's mathematical attainment evolved from describing number patterns to using algebraic language to negotiate problem solutions.

Jenny's transition from finding a rule like "multiplying by 3 then adding 2" to expressing it as B×3+2 seems to be her starting point towards formalising her reasoning. This task provides the child with a bridge that links her prior step-by step arithmetic way of proceeding to an algebraic way of working. It was this arithmetically-exploring → algebraically-describing structured activity that gradually let her gain confidence in using the calculator formal code. Let us carefully look at the child's actions while facing these tasks.

She was free to operate with numbers in a fashion she had already mastered when finding the rule that governs the numbers in a table. Then she expressed that rule using calculator language. The fact that Jenny had made the relevant calculations in advance enabled her to verify (by herself) the correctness of her calculator language utterances. Also, she could resort to teacher's support when she ran into problems, as it happened when she could not make the calculator "*add 1 first then multiply this by 2*". Through using calculator language (33 worksheets before she faced problem solving) she was assigning meanings to programming expressions and getting awareness of some syntax notions.

Later (when transforming), I introduced expressions in which the variable is used more than one time (for example, 3×A+4×A+5). Jenny made sense of these expressions by exploring numerically and began to sketch some syntax rules to manipulate them symbolically. In this process she found that parentheses allowed her to invert function rules of the form $f(x)=ax+b$. There she went further to algebraically describing number patterns and started to operate on algebraic expressions. This process shows a continuous going back and forth from semantic to syntactic notions, from the particular to the general.

Finally, in worksheet 34 (box of maximum volume) she offered a first sample of how her prior experience using calculator language helped her describe general quantitative relationships. To do this she had to work with the as yet unknown, and furthermore, when the as yet unknown was a magnitude which was varying. Her work in worksheet 55 illustrates the extent to which Jenny uses the calculator code to cope with problem solutions. Though the expression she produced properly describe the relevant relationships according to problem constraints, and she used it correctly to negotiate the problem solution (expression 1), the explanation she gave later allows us to see that there is still a lack of command of the calculator code (expressions 2 and 3)). She produced expression 1 to compute the area of the rectangle, and produced expressions 2 and 3 (see table above) to explain her response. Expression 1 shows that Jenny used the letter A to represent the **large side**, consequently ((100−A)÷2))×A represents the product of the large and the short sides (the area of the rectangle). She built expression 1 on the basis of the numerical calculations she made when completing a table that required her to find the length of the short side when the large side was given. That table also included some rows where the short side was given and the large side had to be found. Eventually Jenny decided to algebraically express the relationships by naming A as the large side. But, when explaining how she reasoned to produce expression 1, she reversed the relationship between the sides: her written explanation says that (100−A)÷2) represents the **large side** (which in fact is the short side), and that (A×2) represents the two short sides (which in terms of her initial program would represent **two times the large side**).

Worksheet 55
1. ((100-A)÷2)×A
2. *"(100-A) ÷2 represents the large side".*
3. *(A×2) represents the two short sides".*

There are various plausible explanations for Jenny's confusion. One possible explanation comes from the common fact that many beginner algebra students are better able to cope with problem situations than they are to explain their reasoning. A second is that she had worked flexibly with both the unknown as long side and unknown as short side, and she became confused about what she had used in expressing the relationships. Another possible explanation is that Jenny might have been using the letter as an object (in Küchemann's sense, 1981), thus she might have just been using the letter "A" as a "name" for the side of the rectangle, without relating that name with numbers (the length of the side). This interpretation of letters could have led Jenny to get confused so as to reverse the relationships between the sides of the rectangle. The research data throughout the study seems to contradict such explanation. During the study Jenny always used the letters to represent numbers, that is the way in which she program the calculator. Every time she dealt with a letter a number should have been evoked, otherwise the work with the calculator does not make sense. Consequently, it seems unlikely that Jenny had become confused when explaining her reasoning because she had been thinking of a letter as a name for the side of the rectangle.

The above discussion illustrates the kinds of subtle details that Jenny still have to refine before she becomes a more proficient user of the algebraic language as a tool for negotiating problem solutions.

Jenny's approach to expressing and justifying generality

Interview 3 provides evidence for Jenny's acquaintance with expressing algebraically general statements. As has been discussed in reviewing interview 3, Jenny easily produced the expression A+A+1 to describe the "sum of any two consecutive numbers". She also, without any difficult, produced the expression (A+10)+(A–10) to describe the situation of "think of a number add it to 10 and write down the result. Now take your number away from ten and add the first result to the second one". In fact, making sense of and algebraically referring to these situations seem to be simpler than dealing with the problem's statements cited in the above section. But, in contrast with her attainment in problem solving, where she showed she had grasped how to use calculator language to negotiate problem situations, she could not spontaneously use the calculator's language to justify generality. To do this she needed the teacher's intervention. Jennifer's case suggests that there is a distance between using algebra in problem solving and using it to argue mathematically. Nevertheless, Jenny's skilfulness in describing general relationships seems to be an important antecedent to justifying generalisations since it seems unlikely that the child could do it without being able to communicate using algebraic language.

General Remarks

Priority of operations

Jennifer's case suggest that mathematical conventions such as priority of operations and the use of parentheses play a crucial role in bridging the gap between arithmetic and algebra. Jenny's work shows that she obtained a good level of command of these conventions and her progress tells us about her talent and favourable attitude towards mathematics. Nevertheless, there were also a good number of episodes that show her resistance to incorporate priority of operations into her everyday ways of working, particularly when working with paper and pencil.

This suggests that special teaching attention should be paid to these features. Let us describe an experience with 30 mathematics school teachers during a workshop recently carried out. The teachers were asked to calculate $2+3\times5$. At first they looked a bit confused about the question due to its elementary nature. A few of them let us know their answer, it was 25. Then we asked for the numerical value of $2+3a$ when $a=5$. This time they all got 17, "because that is algebraic substitution, so you know where to start from ... there you clearly have two terms, 2 and 3a". In the other case they appear to be dominated by the arithmetic fashion of calculating from left to right. We know this situation cannot lead us to any generalisation but, at least it shows that these school mathematics teachers were not aware of priority of operations in the arithmetic case. This suggests that there is a teacher's tendency to separate arithmetic from algebra, so we would not be surprised that pupils show a similar proclivity.

Mental calculation

As well as Diego, Jenny's approach to symbolic manipulation shows a strong relation with mental calculation. This remark is derived from observing that Jenny's first successful attempts to operate with literal terms took place where the coefficients were positive integers and addition or multiplication were involved. Interview data confirmed that her skills to mental calculation had much to do with both finding equivalent expressions and simplifying similar terms. This leads us to suggest that special teaching attention should be paid to mental calculation if it is intended to introduce algebra from an arithmetic approach.

CHAPTER 6

THE REST OF THE CASE-STUDY PUPILS: AN OVERVIEW

Introduction

Chapter 5 presented a detailed analysis of the data provided by two of the seven children that took part as case-study subjects throughout the study. The detailed account made in Chapter 5 provides background to the present chapter, which is intended to provide a more complete view of the data collected by presenting a succinct account of the work done and the strategies used by the other five children that were closely observed during the study. This review will be carried out by taking up those features that best characterise each pupil. The chapter discusses those issues concerning pupil's attitude towards mathematics, their mathematical attainment prior to the fieldwork, the work done in terms of the number of worksheets completed, and their strategies to work out different tasks.

The chapter is organised as follows. First, the case-study pupils with below average attainment are presented. Second, the case-study pupils with average attainment are described. Third, the case-study pupils with above average attainment are presented. The chapter ends with a section which presents some preliminary conclusions that outline the horizontal analysis to be made in Chapter 7.

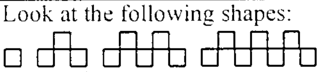
6.1. Lower attainment pupils.

Rocío

As has been said in Chapter 4, two children with below average attainment (Rocío and Omar) were initially chosen to be followed as case-study subjects. Unfortunately, Omar got sick in the middle of the study and was out of school for the rest of the school year. Thus, Rocío was the only pupil with below average attainment that was observed during the study.

Rocío is a 12 year old girl, her work prior to the study showed that she was good at performing arithmetic operations with whole positive numbers but she had strong difficulties when solving arithmetic problem situations. A characteristic trait in Rocío's attitude was that she always tried to present her work to the teacher until she was convinced her work was correct. This attitude encouraged her from the beginning of the study to find out how to take advantage of the calculator's feedback.

Probably due to her reluctance to showing mistakes in her work, Rocío worked slower than the majority of her fellow pupils, for example, in Format 1 she only completed five worksheets whilst the majority of children had completed ten worksheets (see summary table at the end of this section). Nevertheless, her work throughout the study shows that she was progressing. For example, the table below shows the way in which she tackled a problem situation in Format 6. Her answers to this problem show the extent to which she had grasped how to use the calculator code as a tool for representing the relevant relationships within a problem situation and to negotiate a solution.

WS	Problem situation "Figurative patterns"	Rocío's written explanation
46.	Look at the following shapes:  46.2. How many squares are needed to build up the shape that goes in the 17th place?	33
	46.3. How many squares are needed to build up the shape that goes in the 100th place?	199
	46.4. Explain how you reasoned to answer the questions above.	*"I realised that in any figure the row below has one square less than the row above ... I multiplied by 2, because there are two rows, then I took 1 away"*
	46.5. Can you program your calculator to complete the following table?	Completed without teacher's feedback **?→V: V×2−1**

Her explanation suggests that she was referring to a general rule: "I multiplied by 2 because there are 2 rows, then I took 1 away"). This extract suggests that she did not need to refer to a specific case because in "any figure" happens the same: "the row below has one square less than the row above". Rocío's explanation illustrates how the calculator environment influenced her thinking so as to help her look for a general relationship that she can describe using the calculator code: V×2−1.

In worksheet 47 she went beyond correctly describing the underlying pattern: (V×3)−2: at this time she also tried to produce the "inverse program" to answer questions where she needed to calculate the inputs when the outputs were given: (V÷3)+2. Though her attempt was not correct, she finally produced a correct expression with the teacher's feedback (the metaphor of "undoing a route" was a successful hint: "doing last what you did first"). Later on (interview 3) she showed the extent to which she grasped the notion of inverse function when she correctly discerned that the inverse rule was needed to answer the question: "Will the number 877 appear in the sequence 5, 9, 13, 17, ...?" (see also Chapter 7, section 6.4.a).

The following table summarises Rocío's work throughout the study.

Number of worksheets completed • **C**: completed correctly • **F**: completed correctly after having teacher's feedback • Denominators indicate the total number of worksheets in each format.					
Format 1	**Format 2**	**Format 3: Equivalence**	**Format 4: Decreasing functions**	**Format 5: Inversion**	**Format 6: Problem solving**
4/15, **C** 1/15, **F**	2/5, **C**	2/9, **C** 1/9, F	3/10, **C**	3/5, **C**	1/10; **C** 4/10, **F**

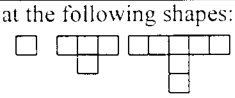
6.2. Average attainment pupils.

Four children were chosen as average attainment pupils according to their work prior to the fieldwork. These children were two girls, Jimena and Erandi, and two boys, Raúl and Diego. Since Diego's case was already presented in Chapter 5 this section will analyse the work done by Jimena, Erandi and Raúl.

Jimena

Jimena is a 12 year old girl, mathematics is among her favourite school subjects. Perhaps the most evident trait in Jimena's work was a tendency to produce algebraic expressions "just to see what happens". For example, in Format 2 (where children were asked to produce calculator program for a fellow pupil to guess), Jimena started, on her own, to produce expressions using the variable more than once; for instance: A×A+A, A÷A×2, and A+5÷A. This "exploratory" approach helped her gain understanding of the dual character of letters in algebra: as symbols that represent "any number" and as symbols on which she can operate. This experience influenced Jimena's work throughout the study, for example, she faced the question of transforming an algebraic expression to make it equivalent to a target expression by directly operating with the variable, for example, by adding 2×A to 7×A to obtain 9×A (this point is discussed in more detail in Chapter 7, equivalence).

Later on (Format 6), Jimena's work suggests that this tendency led her to develop a "trial and refining" strategy based on operating with the literal terms. For example, the figurative patterns included in Format 6 induced for the first time the idea of recurrent rules (see table below), and Jimena's first approach was to use a recurrent rule, for instance, A+3, because "*every new figure has 3 squares more than the figure before*", then, with teacher's feedback, she realised the rule did not work as a calculator program and tried by "multiplying by 3", she checked again until she finally produced the program A×3−2 (see Jimena's answer below). From then on she consistently used this strategy to cope with number sequences. As can be observed this method consists of finding out the slope of the linear function f(x)=***a***x+***b***, once ***a*** is determined, ***b*** can be found by giving specific values for x and f(x). Though she was lucky in producing this method, Jimena's work shows how her "exploratory insight" of operating with the literal terms led her to find a consistent method to face this kind of task.

WS	Problem situation: "Figurative patterns"	Jimena's responses
47.	Look at the following shapes: 	
	47.2. How many squares are needed to build up the shape that goes in the 9th place?	25
	47.3. How many squares are needed to build up the shape that goes in the 17th place?	49
	47.4. Explain how you reasoned to answer the questions above.	*"Every new figure has 3 squares more than the one before, so I thought I had to add 3* (3+A) *As this did not work I multiplied by 3* (3×A)*, but the result gave 2 more than the number I was looking for, so I took 2 away from 3×A"*.

Jimena's work is further discussed in Chapter 7, in particular, her approach to using parentheses, how she developed the notion of inverse function, and how she cope with algebra word problems using the calculator code. A global view of the work done by Jimena during the study is presented below.

Number of worksheets completed • **C:** completed correctly • **F:** completed correctly after having teacher's feedback • Denominators indicate the total number of worksheets in each format.					
Format 1	**Format 2**	**Format 3: Equivalence**	**Format 4: Decreasing functions**	**Format 5: Inversion**	**Format 6: Problem solving**
8/15, **C** 2/15, **F**	5/5, **C**	6/9, **C** 3/9; **F**	4/10, **C** 2/10, **F**	3/5, **C** 1/5, **F**	6/10, **C** 2/10, **F**

Erandi

Erandi is a 13 year old girl. She did not have a successful experience with arithmetic in the elementary school. In particular, she was not good at operating with fractions, be it common or decimal fractions. This experience led her not to like mathematics and to be reluctant to participate in the study. Despite Erandi's difficulties with arithmetic calculations she was considered an "average attainment pupil" because, prior the fieldwork, she showed she was able to delineate a strategy for facing word problem situations, though she frequently could not obtain the solution due to computing mistakes.

Though Erandi was reluctant to participate at the beginning of the study, the way in which she engaged in carrying out the tasks suggests that the computing support offered by the calculator was a crucial source of motivation. Since the burden of carrying out the calculations was left to the calculator she was in a better position to concentrate on using the calculator code to express her reasoning. Erandi was gradually gaining self confidence during the study, this is shown by the increasing number of tasks she was completing as the study progress. For example, in Format 1 she completed 10 of 15 worksheets; in Format 2 she just work out two of five tasks, but started using "two step" expressions (for example $A\times5-1.5$). In Format 5 she showed she had gained confidence on operating with decimal numbers, question that she did not want to know about at the beginning of the study. For example, she correctly work out worksheet 42, where recognising the number pattern strongly depended upon correctly operating with decimal numbers (see Erandi's explanation below).

Worksheet 41						ERANDI'S ANSWERS
I typed a program that produces the table below. Can you make a program that produces the same as mine? Explain what you did to answer this question.						?→E: E−6.1 *"I realised that the difference in the decimal numbers was always 1, after this I just counted how many spaces there were between the whole numbers, it was 6, so the rule must have been taking away 6.1"*
Input	11.4	19	23.1	38	50	
Output	17.5	25.1	29.2	44.1	56.1	

Finally, in Format 6 (negotiating problem solutions), she correctly work out 9 of 10 worksheets. The work done by Erandi throughout the study is summarised below.

Number of worksheets completed

- **C**: completed correctly
- **F**: completed correctly after having teacher's feedback
- Denominators indicate the total number of worksheets in a format.

Format 1	Format 2	Format 3: Equivalence	Format 4: Decreasing functions	Format 5: Inversion	Format 6: Problem solving
7/15, **C** 2/15, **F**	2/5, **C**	4/9, **C** 2/9, **F**	4/10, **C** 2/10, **F**	3/5, **C** 1/5, **F**	8/10, **C** 1/10, **F**

The main strategy used by Erandi to cope with the tasks consisted of exploring numerically the problem situations. She used this strategy to make sense of the question, be it about transforming algebraic expressions, inverting linear functions, or negotiating solutions for algebra word problems. The ways in which Erandi confronted algebraic manipulation and coped with algebra word problems are discussed in more detailed in Chapter 7, sections 7.1 (algebraic equivalence), and 7.4.a, respectively.

Raúl

Raúl is a 12 year old boy, his work prior to the fieldwork shows that he was good at operating with numbers but he had difficulty when dealing with arithmetic word problems. Though mathematics was not his favourite school subject he did not dislike it. In a similar way as Jimena did, Raúl showed a tendency to produce programs using the variable more than once from the beginning of the study. For example, in Format 1 he produced the programs A+A×2−1 and A÷2+A (worksheets 5 and 9), whilst the majority of his fellow pupils produced A×3−1 and A×1.5 to respectively describe the same items.

Raúl's particular way of approaching the tasks shows, on the one hand, an incipient attempt to operate with the literal terms; on the other hand, Raúl's work indicates that he was starting to grasp the role of the letters as symbols that represent a range of numbers, because, when constructing an expression such as A+A×2−1, Raúl was thinking of operating "*on the same number ... that is, I took first a number, say 2, then I multiplied this number by 2, and finally took 1 away*" (interview 1). The extent to which Raúl extended these notions so as to cope with algebraic equivalence and algebraic transformation is further discussed in Chapter 7, section 7.1. The ways in which Raúl coped with algebraic equivalence and algebraic transformation showed that his major strategy consisted of numerical substitution. The experience of describing number patterns seems to be the building block from which Raúl developed the numerical substitution strategy as a tool to cope with algebraic tasks. Possible explanations of how Raúl and his fellow pupils developed this strategy are discussed in more detail in chapter 7., section 7.4.

Raúl's work during the study is summarised below.

Number of worksheets completed

- **C**: completed correctly
- **F**: completed correctly after having teacher's feedback
- Denominators indicate the total number of worksheets in a format.

Format 1	Format 2	Format 3: Equivalence	Format 4: Decreasing functions	Format 5: Inversion	Format 6: Problem solving
12/15, **C** 3/15, **F**	4/5, **C**	7/9, **C** 2/9, **F**	5/10, **C** 2/10, **F**	4/5, **C**	4/10, **C** 4/10, **F**

6.3. Above average attainment pupils.

Two children were chosen as "above average attainment pupils" according to their work prior to the fieldwork. These children were Jenny and Iván. Jenny's case was already analysed in Chapter 5, thus, Iván's will be the only case discussed in this section.

Iván

Iván is a 12 year old boy, he very quickly grasped how to describe number patterns and symbolise problem situations using the calculator code. Iván was the only pupil that correctly completed the whole set of worksheets. From the beginning of the study he grasped how to obtain feedback from the calculator and his work was generally correct, only 4 of the 55 worksheets were given back for him to correct (see summary table).

Iván grasped the notion of letters as representing a range of numbers from the beginning of the study. It seemed that the formality of the calculator code matched well his way of conceiving mathematics, for example, let us observe the formality of his written explanation in worksheet 1: "*the program adds 4 to the number you enter*"; certainly, he was the only child who used such a formal expressions to explain his reasoning. In worksheet 2 he explained: "*I made a program that multiplies by 2*" (A×2), which shows the extent to which he has grasped the general nature of the expressions used to program the calculator. From worksheet 2 on he explained his reasoning by just writing down the program he had made.

As well as Jenny, Iván engaged on his own in producing expressions which led him to realise the need for using parentheses, for example, "*a program that adds 2 first and then multiplies by 3*". Once he was told to use parentheses he used them correctly when facing new situations (see for example worksheet 54 later in this section).

Similarly to Jimena and Raúl, Iván produced programs using the variable more than once, for example A+A÷100 (worksheet 10) to symbolise the number pattern below (Iván's answers in bold):

Input	1	**2.2**	3.1	**4.3**	9	**12**	32
Output	**1.01**	2.222	**3.131**	4.343	**9.09**	12.12	**32.32**

It seems that this "spontaneous" approach to operate with the literal term helped Iván to cope with questions involving algebraic equivalence. Iván worked out so fluently the questions involving algebraic equivalence that the researcher decided to confront him with more complex situations than the ones given to the other children. For example, upon being asked, he was able to program the calculator in order to produce five digit palindrome numbers (i.e., 10000×B+1000×C+100×A+10×C+B), which implies analysing the structure of the base ten numbers and symbolise the relationships among the digits using three variables; he was also able to work with non linear expressions, for instance, transforming A^2 to make it equivalent to A^6, and vice versa.

The strategy used by Iván to confront algebraic transformation was to directly operate with the literal terms, for example, upon being asked to transform A×8 to make it equivalent to A×15, he immediately proposed A×8+A×7. Nevertheless, when the involved expression was more complex, for example, involving two or more similar terms, he resorted to giving specific values to the literal term, which suggests that the main strategy developed by Iván to confront algebraic equivalence was numerical substitution. This point is further discussed in Chapter 7, section 7.3.b.

To what negotiating algebra word problem solutions is concerned, Iván was the only child who correctly completed worksheet 54 (see Iván's answers below).

WS	Problem situation: "Rectangular shapes"		Iván's responses
54.	A real estate firm is selling lots with the following dimensions: A depth of 30 metres more than twice the width. Answer the following using these data.		
	54.1.	Mr. Pérez needed 132 metres of barbed wire to fence his land. Give the dimensions of the plot he bought.	Length: 12 width: 54 (Tasks 54.2 and 54.3 are similar to this one).
	54.2.	Mr. González bought a plot of land 76 metres wide. How many metres of barbed wire does he need if he intends to keep people out of it?	516 meters
	54.3.	Explain your reasoning about the previous questions.	"*I made a program to compute the perimeter according to the given measurements*".
	54.4.	Did you program your calculator to solve the problems? Show your program if you did.	"*The program I used is* ((E×2+30)+E)×2"

Iván answer shows that he was able to recognise that all these problems have the same structure, which allowed him to produce a program to tackle all of them. It seems worth noticing that the task 54.2 can be solved by directly running that program, but the task 54.1 requires the child to use the inverse relationship. His explanation (see task 54.3) allows us to see that he accepted the relationship "A depth of 30 metres more than twice the width" as a given measurement which he symbolises as "E×2+30". Possible explanations of how the children were able to negotiate problem solutions using the calculator code are discussed in Chapter 7, section 6.4.

The work done by Iván throughout the study is summarised below.

Number of worksheets completed

- **C**: completed correctly
- **F**: completed correctly after having teacher's feedback
- Denominators indicate the total number of worksheets in a format.

Format 1	Format 2	Format 3: Equivalence	Format 4: Decreasing functions	Format 5: Inversion	Format 6: Problem solving
14/15, **C** 1/15 **F**	4/5, **C**	8/9, **C** 1/9, **F**	10/10, **C**	5/5, **C**	8/10, **C** 2/10, **F**

6.4. Teacher's feedback throughout the study

The paragraphs below describe the kinds of teacher's feedback offered to children during the study. Teacher's feedback was given both during the calculator-based sessions and through marking the written work that the children did every session. Besides, the teacher gave more specific and direct support to the children during interviews.

Guiding principles

As was stated in Chapter 4, the teacher's feedback was guided by the principle of "answering a question only when he considered that the child would not be able to give an answer by him/herself". When it was found that a child was not able to answer a question, he/she was not given an answer but a new question that might encourage the child to reflect on the task so as to sort out the underlying difficulties. This "feedback method" was intended to fit the pragmatic approach used to shape the activity in the classroom which was based on doing, in the sense that the child was expected to learn the calculator code through use, consequently, the teacher's feedback should help the child to learn from it through using the calculator code.

The teacher's feedback was always given to children, when their work was correct they were given encouraging comments such as "well done, continue working in this way".

If there were mistakes in their work, the children also were given encouraging comments. The teacher highlighted the positive points in their work and invited them to do something with the calculator that may lead them to see the mistakes.

Feedback within classroom sessions

During classroom sessions the teacher/researcher gave feedback only when the children explicitly requested him to do it. An example of how teacher's feedback was given during classroom sessions is provided by the following episode. In worksheet 46 (see figure), Jimena found that the rule governing the number of squares for every shape in the sequence was B+2, "because any new shape has two more squares than the shape before" (recursive rule). Jimena then went to the teacher/researcher and proudly showed her work to him. She was told that was fine, but it was suggested that she run the program B+2 and observe if it produced the number of squares in any shape. She realised that the program did not work as she expected and that her recursive rule was not very suitable for her to find out how many squares would be needed to draw shape number 100 in the sequence. Then she finally built the program B×2−1, which better helped her to answer such a question.

Feedback to children's written work

At the beginning of every new classroom session each child received an envelope containing the work they had done, which had been marked by the teacher. The children were emphatically told that they must take first the work marked wrong and correct the highlighted mistakes; they could not go on with the rest of the tasks till they had tried correcting their work. If a child found that he/she could not correct the mistakes by him/herself, the teacher's verbal feedback would be available.

Children's mistakes were marked underlining the mistake(s) and some comments were written down by the teacher following the idea of making evident to the child the mistake he/she has made. In general, the teacher's comments suggested to do something and then think about the results of doing that thing. For example, the children were asked to run the program they had built and check if it produced the expected results.

The teacher's feedback concerning the children's written work was crucial, particularly at the beginning of the study. An illustrating situation is provided in Chapter 5 (p. 107) when Diego's work was discussed. In general, the worksheets in Formats 1 to 5 required the children to produce a calculator program and then use this program to complete a given table. It was found that a good number of children did just what was asked: to run the program they had produced, so they did not pay enough attention to the results produced by the program, particularly where the given table required them to find out the input when the output was given. This activity showed that many children confronted the question without checking back if the program worked correctly. At this point, what helped them realise the mistake was the teacher's feedback, which consisted of asking them to run the program so that they might observe the incorrect answers. Then the children were encouraged to find out why that had happened. A detailed episode that further illustrates the teacher's feedback to children's written work can be found in Chapter 7, p. 212.

Feedback during interviews

This type of feedback confronted the teacher/researcher with a more difficult situation. Though the interviews were carried out following a ready made protocol, children's reactions frequently led the interview to unexpected situations that required the teacher to react on the course of the events. For example, when Erandi failed to reduce the similar terms in the expression A×2+A×3+A×5 despite the fact that she had shown in the preceding interview that she had found a "method" for doing this consisting of "adding up the numbers and then just attach them to *A*". As was discussed in Chapter 7, she found that 10+3×A was equivalent to A×2+A×3+A×5, because "2+3+5=10 and the 'A' appears three times". Erandi's reaction suggests that if the teacher had only told her that she was wrong, she could hardly learn from this situation why she was wrong. Fortunately, the teacher decided to ask her to explain how she found this result, trying to not make evident that he had noted something wrong in her answer. This teacher's attitude seemed to favour a more natural reaction from the child, who proudly offered the explanation mentioned above. Meanwhile the teacher gained time to look for a better strategy to help the child realise the mistake. The teacher then asked her for another way of checking the correctness of her answer. Since Erandi got stuck she was asked to run both programs (A×2+A×3+A×5 and 10+3×A) to verify whether they were equivalent, the child said "it was not necessary because she had correctly applied the 'method' she had found". This child's reaction suggests that if the teacher had not made her to run the programs and compare the results, she could hardly accept she was wrong, if she did, it would likely to have been as a result of obeying the teacher's suggestion but not on the basis of a better understanding of the situation.

Concluding remarks

These kind of situations provided relevant research data, these data were obtained thanks to a suitable (and sometimes fortunate) teacher's reaction when giving feedback. Nevertheless, it cannot be assured that all of these precious moments when interacting with children have been exploited during the study, quite likely some were lost. However, the experience throughout the study shows that the general principles for guiding the teacher's feedback were crucial supporting points for the teacher when interacting with children, whether during classroom sessions, when checking back their work, and in the course of individual interviews.

The research data suggest that perhaps the most important feature of the teacher's feedback was to offer support when the children were confronted with syntactic issues, like transforming algebraic expressions and using parentheses. Though the calculator can offer support for children to make sense of algebraic expressions by allowing them to link such expressions with their numerical value, operating algebraically seems to encourage children to prematurely get rid of numbers and start generating rules for algebraic transformation.

On the one hand, this seems to be profitable because children generate their own rules, which quite likely conform better with their way of reasoning than the rules taught by the teacher. But, on the other hand, the research data show that a good number of pupils generate mal rules, which apparently, without the teacher's intervention, the children would keep on using such mal rules being unaware of the involved misconceptions.

Finally, the list below describes the situations where the children most frequently required the teacher's feedback.

- Situations which implied awareness of order of operations and consequently of the use of parentheses.
- Tasks which required the children to produce expressions of the form ax+b.
- Encouraging the children to become aware of the possibility of obtaining feedback from the calculator. For example, using the calculator to verify algebraic transformations by means of typing the corresponding programs and running them to check back the equivalence of the involved expressions.
- Tasks where children confronted similar terms simplification. In such kind of activity the children showed a tendency to generate their own syntax rules which sometimes may lead them to generate misconceptions.

6.5. Summary

The succinct report of the work done by Rocío, Jimena, Erandi, Raúl and Iván has been intended to provide a more complete view of the detailed analysis of Jenny's and Diego's cases made in Chapter 5. The data presented in Chapter 5 and 6 provide elements which lead to the following general conclusions.

a) The calculator-based environment helped children develop:

- The notion of letters as symbols that represent a range of values.
- A notion of algebraic equivalence: "two calculator programs are equivalent if they produce the same inputs for same outputs". They used this notion to cope with simplification of similar terms and transforming an algebraic expression to make it equivalent to a target expression.
- A notion of inverse function, which they used to cope with problem situations.

b) With different levels of attainment, the children were able to use the calculator code to cope with algebra word problems on the basis provided by their experience of describing number patterns.

c) The research data suggest that the children would not make mistakes such as $a+b=ab$, $a+a=a^2$, $2\times a=a^2$, $5\times a+3=8\times a$, nor $(a+b)^2=a^2+b^2$. This conjecture points to a relevant feature of using calculators in the classroom and deserves further research.

d) The research data shows that the teacher's feedback is crucial, particularly when children confront symbolic manipulation (simplifying similar terms). In such kind of activity the children showed a tendency to create their own syntax rules which sometimes may lead them to generate misconceptions.

Chapter 6: The rest of the case-study pupils: an overview

CHAPTER 7
RESULTS: HORIZONTAL ANALYSIS.

Introduction

This chapter discusses from a more general perspective the results analysed in Chapters 5 and 6. In order to fulfil this aim the chapter offers a horizontal analysis which was carried out by examining selected excerpts of children's work involving what seemed to be their most significant features in terms of the aims of this study. The main results of the thesis constitute the core of this chapter and analysis is focused on the calculator-based environment as a mediating tool in children's learning of algebraic code as a language-in-use. Some excerpts of work done by children when dealing with algebraic transformation and negotiating problem solutions provide the raw material for discussion. Since the children mastered some general aspects of arithmetic use before they make much progress in either the semantic or syntactic algebraic domain, the chapter will outline why pragmatics seems to provide the most general support system for mastery of the more formal aspects of algebraic language.

Here, the term pragmatics refers to a highly framed teacher-child and child-calculator interaction, it is hypothesised that this highly framed classroom environment will provide a context that gives continuous feedback to the children's use of calculator language so that they can negotiate meanings for the calculator code through using it. This pragmatic approach is not based on syntactic rules or definitions (which characterise a syntactic approach) nor on rich examples for children to be followed and later on induce generalisations (which characterise a semantic-based approach). The pragmatic approach adopted here is founded on a tight relation between context and language use, so that the use of language can always be checked upon context itself. It is the context which encourages language exploration and structures it. A context which helps children make sense of what is being done in the classroom and provides support for negotiating meanings so as to help them use the calculator code as a means of communication.

The analysis was carried out by thoroughly looking at how children's acquisition of calculator language evolves through the syntactic, semantic, and pragmatic facets of language. As has been discussed in Chapter 4, these facets of language are used as categories which may inform the particular ways in which different children approach the specific tasks used in this study. Since this framework will be applied throughout the chapter the characterisation for syntax, semantics and pragmatics are presented below.

- *Syntax*: How children acquire their facility in managing well-formed algebraic utterances governed by the formality of calculator language.
- *Semantics*: The nature of the meanings that children develop for algebraic utterances (words) as they use calculator language to explore number patterns and face algebraic transformation and problem solving (possible worlds).
- *Pragmatics*: The manner in which children come finally to use well-formed algebraic utterances (produced while describing number patterns) to face new problem situations within different contexts.

The chapter presents a picture of the calculator's role in mediating children's learning processes from no previous experience of algebra to negotiating problem solutions using algebraic language. As has been showed in Chapter 5, children's work involves highly intertwined features, therefore, some issues will be referred to in more than one of the above categories. In order to better appraise the results of this research it must be remembered that all the instruction that the children received to cope with the experimental tasks was only that necessary to handle the calculator programming mode.

The content of the chapter is organised as follows: Section 7.1 deals with those results that relate with the semantic notions developed by the children during the study, these notions correspond to those meanings that children developed for the letters and expressions they used when programming the calculator, the notion of algebraic equivalence, and the notion of inverse function. Section 7.2 addresses the ways in which the context influenced children's approach to negotiate solutions for algebra word problems. Section 7.3 discusses the strategies that children used to cope with algebraic transformation and problem solving. Finally, Section 7.4, Final Remarks, presents a general discussion of the results presented in the preceding sections. The discussion includes a view of the results of this research through the lenses of the research literature of the teaching and learning of algebra.

7.1. The calculator's role in children's acquisition of semantic notions

The notion of Algebraic Equivalence

While using the calculator language to describe number patterns the children grasped that "*two programs are equivalent if both give the same outputs for same inputs*". This notion resembles the criterion for equivalence between functions: If f and g are functions, and $f(x)=g(x)$ for every x, then f and g are equivalent. The ways in which children acquired a notion of equivalence has to do with synonymy, that is, while exploring with the calculator they found that the valuable thing in all this was its numerical value, because the numerical values is what provides feedback to them, in other words, the numerical value is what provides the meaning for the involved algebraic expression. This process helps children learn that a calculator program may have different expressions, but these expressions remain equivalent if they produce the same results for all values.

A relevant feature of children's notion of algebraic equivalence is that the children were able to apply such a notion to confront new algebraic expressions containing two variables both in linear and quadratic expressions. This provides information on the potential of such a notion. For example, in interview 4 the children correctly answered the question: "A pupil from another class says that $(A+B)^2$ is equivalent to A^2+B^2, what do you think about that?" This fact relates to Bruner's (1983) hypothesis that "formats eventually migrate from their original situational moorings and are generalised to activities and settings in which they have never before occurred" (p. 121). This point is further discussed at the end of this section.

The discussion will now more closely analyse how the calculator-based environment supported children to cope with situations involving the notion of algebraic equivalence. As will be discussed below there were specific features in the classroom setting that helped children gradually confront situations involving algebraic equivalence. First, the simple fact that children could use any letter to represent a variable within a calculator program induced a first insight of equivalence. This fact helped them grasp that the numerical value of a program does not depend on the letter used as a variable. The description of children's work presented in Chapter 5 widely illustrates how children resort to using different letters when programming the calculator. This fact was observed in the work done by all of the children. An example of the extent to which children grasped that the letter used does not affect the results of a calculator program is that all the children answered correctly the question of comparing programs such as (A+7)×2 and (L+7)×2 (see Interview 1, Diego's case, Chapter 5). This result has also been reported in other computer-based studies (Hoyles and Sutherland (1989) using Logo, Sutherland and Rojano using spreadsheets (1993), Tall (1991) using Basic). The fact that this result has been found from different computer-based environments strongly suggests that children's belief that changing the letter changes the value of the unknown (Wagner, 1981) comes from issues concerning the learning of algebra within a paper-and-pencil environment.

Children's reactions indicate that what enabled them to cope with algebraic equivalence was the meanings they developed for letters and algebraic expressions through programming the calculator. The following extract illustrates what seems to be the first antecedent for children to develop the notion of equivalence. Raúl[1] was asked to compare the programs (A+1)×3 and (L+1)×3. He answered: "*they are the same because with the calculator the number you input stands for the letter you have put ... That's ... the calculator removes the letter and puts the number you input in its place*". Raúl's reaction allows us to see the richness of the meanings he has developed through using calculator language. First, his answer shows that he looked at the whole structure of the expressions rather than only conceiving them as mere procedures. Second, it shows he has gained awareness of the arbitrary choice of letters when using algebraic code. Third, his reaction shows an incipient notion of letters as symbols that represent a range of

[1] Average mathematical attainment pupil

values, and algebraic expressions as a means of representing and computing general arithmetic procedures.

Another relevant feature concerning the calculator environment is that the strong similarity between the calculator code and the algebraic code seems to provide an advantage in comparison with the use of Logo or spreadsheet codes. The experience with the calculator code helped the children make sense of questions involving expressions like $(A+B)^2=A^2+B^2$. When confronting this question the children did not need any explanation about the expression, whereas in the case of Logo or spreadsheet children still have to sort out the subtle differences between Logo or spreadsheet code with the algebraic symbolism. Even in the case of Basic, where the language more closely matches the algebraic code, children still have to link the expression (a+b)^2=a^2+b^2 with the standard algebraic notation.

A second factor that might influence the development of the notion of equivalence was the social milieu offered by the classroom activity. Though most of the children decided to work individually, the feasibility of producing different representations for the same program was present from the very initial tasks, for example, the children eventually witnessed that someone else's program was different from theirs but produced the same outputs. This fact mainly emerged when the children built a program for a fellow pupil to guess (Format 2). These tasks encouraged the children to discuss whether two calculator programs were equivalent or not. A third influencing factor were the tasks in Format 3. These tasks consisted of asking children to make various programs that produce the same table (see Format 3, Chapter 5). Children's responses showed that they had grasped that two programs are equivalent "if they produce the same outputs for the same inputs". Their responses indicate that the majority of the pupils resorted to operating with the independent terms to construct equivalent programs, for example $3\times A+2=3\times A+8\div 4$ (see Format 3, Chapter 5). Similar results have been reported by Sutherland (1996), working with spreadsheets the children were able to construct equivalent spreadsheet rules without needing the teacher's intervention. In this case the children also concentrated on transforming the constant terms.

This sort of invitation was taken up by several children and produced complex expressions such as D×7–D×3 to be equivalent to W×4 (see, Jenny's case, Format 3, Chapter 5). In this respect Ruthven (1993a), using graphic calculators with a group of 13-year-olds, reports that using the key *Ans* to explore the behaviour of number sequences led the children to produce different expressions that encouraged rich discussion about equivalence. Though children's notion of equivalence is correct when constructing expressions like D×7–D×3 to be equivalent to W×4 (as functions), this example suggests a potential problem that children may have to face when they deal with polynomial expressions with more than one variable and with equations containing more than one unknown. This situation deserves further research.

The notion of equivalence between functions has also been investigated by Cuoco (1995). Cuoco used a similar approach as the one adopted in the present study (tables, algebraic rules) and used three different environments to introduce the idea of equivalence between functions: The Function Machines allows students to start with isolated calculation and gradually interiorise calculations into procedural entities. Logo provides an environment in which students build an experiment with processes, produce tables, compare them and begin to manipulate them as data. ISETL supports expressions of higher order functions allowing students to manipulate functions in a mathematical way. The findings obtained by Cuoco are similar to the results of the present research: the students found that two functions are equivalent if they produce the same outputs for the same inputs. If the limitations of the graphic calculator are not crucial factors, such as screen size, reduced data storage capability, and lack of symbolic manipulation, the graphic calculator may be used as a media which provides the three different computerised media used by Cuoco: using the calculator computing mode pupils can perform isolated calculations; using the calculator programming code the pupils are allowed to use standard mathematical notation and produce tables of values for the functions used to make the programs.

Algebraic equivalence as a tool for coping with algebraic transformation

Algebraic equivalence was a powerful semantic notion developed by the children. Chapter 5 provides evidence for the extent to which Jenny and Diego were able to cope with situations which involved the notion of algebraic equivalence (see equivalence in Interviews 1-3, and in Concluding Remarks). This section analyses further episodes with the rest of the children and discusses how the notion of algebraic equivalence allowed them to confront algebraic transformation and simplification of similar terms within linear algebraic expressions. These episodes also indicate the potential of introducing the calculator code as language-in-use, particularly how children actually use the calculator language as a means of learning about a new situation, that is, as a means for negotiating new meanings in order to cope with new complex problem situations.

The data obtained during the fieldwork suggests that the use of mathematical functions as devices to carry out arithmetic procedures makes algebraic manipulation an action of semantic interpretation rather than a process of applying syntactic rules. The research data indicates that this kind of encounter with symbolic manipulation led children, in time, to generate initial rules to face these tasks as well as to develop strategies to check their correctness. The following episode illustrates this. The children were asked the question:

> "I wanted to type the program B×8 but I made a mistake, instead of that I typed B×7. Can you correct it without deleting anything of what I typed?"

Erandi faced the above question by exploring the numerical behaviour of the given expressions, which resemble her prior experience in describing number patterns using calculator language. She used numerical exploration to make sense of the question and finally recognised an underlying pattern which she could symbolise using calculator language. The following documents her approach.

The child tried with B×7+4 and checked it for B=2 (she expected to get 16, i.e., B×8). She ran the program and saw it gave 18, she said "*I thought it would work ... I had* *mentally checked it for B=4, it was right* (i.e. B×7+4=B×8, for B=4). *That is why I thought it would give 16 when I input 2 ... But it multiplies by 7 ... then it adds 4, 7×2, 14, plus 4, 18 ... it works all right for 4 ... but wont do it for 5 ... it should work for all the numbers ... there must be something to do for making it work for 5, 6, and so on*". After a few moments of reflection Erandi found how to do it: "*I got it, it must be 7×B+B*". She checked the program and clearly happy explained: "*I realised that the number I needed to add is the one you put here* (pointing at B in 7×B)". Later on, she easily worked out other symbolic manipulations like obtaining 9×B by adding B+B to 7×B, and subtracting 3×B from 10×B to obtain 7×B. Erandi explained that "*at the beginning I thought that it wasn't possible to repeat and repeat the letter within a program ... I mean, to put B more than once*". It is worth noticing that she answered the last questions working with paper but applying the numerical exploration strategy induced by using the calculator.

This episode highlights a relevant result of this research, a result of encouraging pupils to meet the algebraic code as language-in-use. Erandi's work shows that she did not use the calculator code to formally describe a fully polished idea. She used the calculator code to explore the situation in order to make sense of the question; by using the calculator language she was learning about the question, the feedback offered by the machine supported her in successively structuring-re-structuring her reasoning. Erandi's work shows that she was actually using the calculator language as a means of communication, communication with an interlocutor (the calculator) that can help her gain insights about the problematic situation she was facing. Other children confronted this task in a similar way as Erandi did (Jenny, Diego, Rocío). Other children's approaches are discussed later in this section.

Erandi's explanation suggests that expressions like 7B+B were like "new words" for the children. It is also interesting to note that these "new words" can be generated from previous children's experience of using calculator language to describe number patterns. That is, the language used to work out these tasks provided elements for Erandi to extend her "vocabulary". Erandi's work suggests that as she gained further insight into the language as a codified system of representation, she came to operate not only on concrete events, but upon possible combinations derived from operations on the language itself.

Another important factor that influenced the ways in which children developed strategies to cope with algebraic equivalence was the way in which the tasks were administered, particularly the intention of respecting each child's pace[2]. This feature was aimed at simulating the ways in which children acquire natural language. Respecting each child's pace was a crucial aspect of the learning processes, it encouraged children to focus attention on their own work, which allowed them to keep on using, and progressively refining, their own strategies without being disrupted by someone else's interventions, particularly, general teacher interventions. This way of working resulted in a range of children's different strategies to face algebraic transformation, some of them are illustrated below.

An alternative children's approach to transforming algebraic expressions consisted of directly operating with the literal term. It is significant that this strategy was used only by those children who, from the beginning of the study, spontaneously modelled number patterns with expressions containing the literal term more than once. Those children (like Jimena[3], Raúl and Iván[4]) who built expressions involving the literal term more than once from the beginning of the study (for example, A–0.75×A, Iván, Format 3), seemed to have gained awareness of the dual character of letters in algebra, both as devices to represent "any number" and as entities on which they can operate.

Raúl's work illustrates a version of this approach. He faced the question using a whole-part strategy. For example, to transform A×9 to obtain an equivalent expression to A×10, Raúl directly typed A×9+A because "*A is one tenth of A×10, so it is just missing an A to complete A×10*". Nevertheless, Raúl resorted to a similar strategy as Erandi's one, when he was confronted with more complex situations, for example, transforming

[2] A detailed account of this point is provided in Section 4.4., Classroom Setting, Chapter 4.
[3] Average attainment pupil

A×3+A×5+A to make it equivalent to A×13. Raúl's approach resembles Küchemann's category of "letters as objects". Raúl's responses suggests that he was not thinking of a letter as representing a range of numbers, rather, his "holistic" view of the expression (*A* as one tenth of *A×10*) suggests that he is dealing with the letter as an entity. As was stated before, the whole-part approach was found as a spontaneous children's strategy, (that is, this strategy was neither encouraged or suggested by the tasks included in the study). This fact suggests that the interpretation of letters as objects was apparently derived from the whole part tactics and has little to do with the calculator-based experience they had during the study.

Despite Raúl's whole-part tactics which helped him to easily set up incipient rules for algebraic transformation like 3×B+4×B=7×B, he could not look at A×3+A×5+A×2 as A×10, which would have put him in the position of adding A×3 to obtain A×13 (as he did before). Raúl's reactions suggest that his "whole-part" strategy is more context dependent than the "number-based" strategy, which appears to be of a more general nature.

Jimena and Iván faced also the question by operating directly with the literal term. In the same way as Raúl did, they resorted to exploring with specific values when facing more complex situations. This shows that exploring the numerical behaviour of the algebraic expressions was the strongest strategy the case-study children acquired to face algebraic transformation.

It was mentioned at the beginning of this section that children's notions of letters and algebraic equivalence allowed them to confront more complex questions. The following episode provides evidence for this claim. In the fourth interview the children were asked the question: "A pupil from another class says that $A^2+B^2=(A+B)^2$, what do you think about that?" Though they had never been confronted with this kind of expression, the children were able to make sense of the question and successfully faced it by giving specific values to the letter. Jenny went beyond this and found that "*this pupil is wrong*

---

[4] Above average attainment pupil

... *but he possibly might have been thinking of A=0 or B=0*". Iván and Diego gave a "holistic answer": "*see,* $(A+B)^2$ *must be greater because you are adding first the numbers and then you are raising it to square*". Then they substituted the variables for specific values to strength their argument.

These children's achievements agree with Bruner's (1983) view that for children to develop meanings for linguistic utterances they must take part in communicative interaction, a setting in which "the child is hugely aided in his mastery of linguistically mediated requests by the social interactions into which he enters with his mother and other adults" (p. 1). Here, the role of calculator as a supplier of immediate feedback played also a relevant part, as has been shown in the above extracts.

7.2. The calculator's role in children's acquisition of syntactic notions.

The results of this research about children's insight into syntactic conventions centre around the following issues: a) How the calculator environment enhances the instrumental character of syntax conventions; b) The role of priority of arithmetic operations in simplifying similar terms; c) Pros and cons of children's proclivity to create their own syntactic rules for symbolic manipulation; and d) Children's awareness of the role of letters within algebraic expressions. These points are discussed below in terms of specific results and the role played by the calculator-based environment.

a) How the calculator environment enhanced the instrumental character of syntax conventions.

The research data suggest that children more effectively learned syntax conventions instrumentally, that is as instruments for carrying out certain previously operative functions and objectives. For this to happen it was crucial that children used the calculator language to express their own reasoning; while doing this the children make the involved computations in advance which helped them realise that the calculator proceeds differently from them. It was observed during the study that children are not concerned with priority of operations and use of parentheses as long as they work with paper and pencil. In contrast, they were aware of these conventions while working with *the calculator. This fact enhances the role of the calculator environment in helping children gain awareness of algebraic syntax. The children inquired about syntax conventions when they ran into problems to express their reasoning using calculator language. The research data suggest that it was the need to use syntax conventions as a means of communication which made children realise their value.*

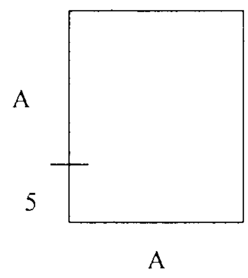
To summarise, using parentheses was a feature that the children needed to be reminded about. This took place after children realised that the calculator proceeded differently from themselves. Once the children understood the function of parentheses they used them as frequently as they could. Since the tasks required the children to produce algebraic utterances their experience with parentheses was focused on using them, later they showed they were able to read them too. This suggests that the children's pragmatic encounter with order of arithmetic operations and use of parentheses, helped them acquire these syntax conventions as tools for their reasoning. These issues are more carefully discussed in what follows.

There were children who met syntax conventions while trying to program the calculator. The key point in this kind of encounter is that these children have done the involved computations in advance, otherwise they could not have realised that the calculator did not work as they expected. This children's approach relates with Shatz's (1982) conjecture that "some forms of syntax may be derived from prior semantic representations that achieve deep structure by being transformed by social interaction" (quoted by Bruner, 1982, p. 26)

It is worth noticing that the children who spontaneously met syntax conventions seemed to have grasped a better notion of priority of operations and use of parentheses than those children whose encounter with syntax was provoked by the researcher. Perhaps the greatest difference between these approaches was that, in the former case, the entry into syntax came out as a response to children's own enquiry, whilst in the latter, the children had to follow not their own reasoning but the researcher's.

An extract from Jenny's work illustrates a spontaneous approach to syntax. Jenny engaged herself in posing an item for a fellow pupil to guess (format 2). Her idea was to program the calculator so that "*it first took 1 away, then multiplied this by 3*". Her first attempt was to make the program B−1×3. Jenny's explanation nicely shows how the formality of calculator code helped her gain awareness of the priority of operations and use of parentheses: "*the calculator did not do what I meant ... As I don't think as the calculator does, I said 1-1, 0, then 0×3, 0. The machine said: −1×3, -3, 1-3, -2. I'd like to know how to make the calculator do what I want*". Then the teacher/researcher suggested she used parentheses. The way in which this pragmatic approach was assimilated by Jenny was shown later when she was asked to explain what parentheses serve for: "they *serve to make the machine do first one operation then the other*". The kind of expressions she produced later (Format 6) allows us to see the extent to which she could apply these notions, for example, ((A×2+1)×A))×155 (see Chapter 5, worksheet 51).

The last point to be discussed about this result is that of children's apparent reluctance to incorporate syntax conventions into their everyday computing routines, particularly when working with paper and pencil. This point highlights the calculator's role as a mediational tool in children's learning processes. Working with the calculator encouraged the acceptance of parentheses use and operation priority as conventions necessary to "communicate" procedures to the calculator. Children's reactions suggest that, for them, the calculator played the role of an interlocutor that requires formal use of syntax to produce the expected results. This kind of interaction took the children, from being competent readers of expressions containing parentheses, to being competent in using these symbols as instruments of their thought. The children did not really understand parentheses use until they needed to *use* them as a means of expression rather than to just *read* them. Their answers show that parentheses use only becomes reasonable when they have to express a procedure algebraically. Seemingly, new rules and procedures must prove absolutely necessary or at least more effective than those already known, otherwise, the children treat them as teacher impositions and thus forget them quickly.



An episode with Diego illustrates well this children's trend. He was shown the figure on the right hand and asked if he could compute its perimeter (interview 4). After some struggle the child said that this could be made with a calculator program and eventually produced the expression (A+5)×2+A×2. The process by which he came to produce such an expression indicates that he values syntax restrictions only when the context demands them. Before Diego correctly expressed the perimeter he wrote the incorrect expression A+5×2+A×2. Nevertheless he unerringly used it to compute the perimeter for several specific values. Formal syntax restrictions did not perturb him because he was following his own reasoning: "*first the side's length plus 5, that's the height, then multiply it times 2, then the side's length plus 2, then add together the results*". However, when being asked to do it with the calculator he typed without any hesitation (A+5)×2+A×2 "*because that's the way I'd do it with the calculator*". This episode suggests that if the terms involved are just numbers, the child does not see the need to group them, since he can always operate on two of them to get just one.

The use of parentheses implies delaying some operations. For example, with expressions like (3+A)×2 this delay is required. The following episode illustrates this situation. Parentheses use and its relationship with operation priorities was explained when the calculator's keyboard was introduced (at the beginning of the school year). The children then completed quite a few activities. Nevertheless, most of them did not recall this experience later when they needed to use parentheses to program the calculator.

Children's work shows that they find it unnecessary to group terms because they do the operations themselves. They were asked, for example, to compute (with paper and pencil) the perimeter of a 5 cm by 8 cm rectangle and show their work in just one expression. They used the expression 5+8×2, while getting the right answer, 26 cm. Clearly, their reasoning and calculations were correct, their problem was expressing correctly their procedure. Apparently, parentheses use and operation priorities make no sense to them, in spite of knowing about them. This is because their arithmetic procedures do not lead them to any questionable results. The following problem was given to them to prove this: "A student from another school uses the operations 2×5+8 to compute the rectangle's perimeter. What do you think about that?". The children thought that was all right, because "*you know what it's about, first you add 5 and 8, 13, and then you multiply that times 2*". To contradict this they were asked to run it through the calculator. They saw that something was wrong, and remembered that they needed to use parentheses. Still, this did not seem like a relevant experience; ultimately, they could thrive without using parentheses, so their use was too sophisticated.

It was said above that the real need for parentheses use first arose when students were building number patterns for their classmates to duplicate with a program (Format 2). Given the nature of the activity, students were expected to try to build complex number patterns, which was encouraged; but they were asked not to base that complexity on big numbers or complicated fractions. Thus, to build number patterns which were hard to find, they started to vary the expressions' structure. In these sessions several students asked questions like Jennifer did, for example Jimena: "*I want the calculator to add 1 and then multiply times 2. I programmed A+1×2, but it doesn't do what I want, it just adds 2, why?*"

From similar situations the children noticed that they needed parentheses to modify pre-established order and operation priorities. Pupils were very enthusiastic about the chance to use different expressions. Soon, this information had spread and most of the class were using parentheses in their expressions. Later, when we worked on inverting programs, it was noticed that they did not mention parentheses use when they explained what they had done; though, in their worksheets they did use them. They were asked why they did this. The following incident with Jenny exemplifies this: "*I don't think like the calculator ... neither do you ... you can understand me ... If I want the calculator to understand me I have to use parentheses, otherwise it will make a mess*"..

b) The role of priority of arithmetic operations in simplifying similar terms.

The research data suggests that the priority of operations plays a central role in children's understanding of similar terms simplification. This notion helped children to distinguish the terms within an algebraic expression and find a relationship among its coefficients which eventually led them to generate rules for simplifying similar terms.

The major issue discussed in this section is the role of children's awareness of priority of operations in their way of generating rules for simplifying similar terms. Children's work showed that once they have successfully simplified expressions containing two terms, they tend to create rules for simplifying like terms, for example, B×4+B×2=B×6. Nevertheless, this achievement was not enough for them to simplify expressions containing more than two linear terms. The children faced these kinds of expressions by giving particular values to the variable, so their success strongly depended on their computing skills. The children did so because they could not realise the feasibility of applying their rules for simplifying two terms more than once, for example:

$$5\times A+2\times A+4\times A=7\times A+4\times A=11\times A.$$

The research data suggest that a lack of awareness of the associative property of addition was not at the core of the difficulty they had, this was confirmed by the fact that they could not cope with the question even after they were reminded of the way they usually carry out that operations with numbers. The following extract from Raúl's interview illustrates this fact: "*I know what to do with additions like 5+2+4, that is 11, but there (pointing at 5×A+2×A+4×A) you have something else ... those numbers the A stands for ... you still have to multiply ... that's where I get mixed up*".

Children's reactions indicate that the problem they had to sort out was a lack of awareness of priority of operations. The successful strategy they used was to type a program to evaluate the algebraic expression. They then ran the program to explore its numerical behaviour and finally found how the calculator performed these operations. The following extract from Iván's third interview illustrates this. He was asked to simplify 12×B+5×B−2×B: After some failed trials working with paper and pencil he said: "*It* *might be ... if I get the result of all this* (he typed the program 12×B+5×B−2×B) ... *If I input 1... it gives 15 ... It multiplies by 15! Then it is ... see, 12 times 1, 12, 5 times 1, 5, altogether gives 17, minus 2, 15*! ... I can type it shorter: 15×A" Then he explained that he had tried to do that by mental calculation but he "*got mix up because 12 times 2 gives 24, then I multiplied 24 times 5, then plus ... I don't know*". After this he found that he could just add "*the numbers that are not B*". The next section tell us about some pros and cons of children's proclivity to generate syntax rules.

c) Children's proclivity to create syntactic rules for symbolic manipulation

Similar terms simplification was the only task in which the children generated malrules. Once the children start generating syntax rules they tend to completely rely on them, which in the case of malrules may prevent them from achieving future success in mathematics.

An important outcome regarding similar terms simplification is that it was the only task in which the children generated malrules. In other tasks, like transforming expressions or inverting linear functions, they sometimes generated non elegant or less effective rules, but never false rules, as occurred in the case of simplifying similar terms. A possible explanation is that when transforming an expression to make it equivalent to another, the children have two expressions to work with, a "target" expression and a "source" expression, which helped them get clues to relate each other. While in the case of simplifying they had just one expression, the simplified expression was something to be found. Though they could compare the original expression with the simplified one by numerical substitution, they did not, because they had faced the question using their own rules. This children's trend presents pros and cons. The following excerpt of Erandi's fourth interview documents this.

Erandi had faced this type of question in interview 3 and successfully started to make rules to simplify two similar terms. In the fourth interview she was asked a more difficult question: to "make shorter" the program A×2+A×3+A×5. She immediately simplified it to A×3+10, so the interviewer asked her to program the calculator to check her answer. She explained: "*I don't need to do that, it is all right ... I counted the A's, there are three, then I added this, 3+2+5*".

The above excerpt shows why she thought it was not sensible to use the calculator: she already had a rule to work out the task and she was even able to explain it. On the one hand, this episode enhances the fact that the child had used her own rule, it compelled her to defend it, which finally led her to get a better understanding of the question. Once the child checked her answer with the calculator she realised her error and did not want to abandon the task until finding the mistake she had made. On the other hand, the episode allows us to see the importance of a teacher's opportune intervention. This resembles Bruner's assertion that "the pragmatic route requires that the adult be a partner". (Bruner, 1980, p. 163).

Erandi's answer also suggests that symbolic manipulation prompts the child to shift their conception of letters as "representing any number" to a new conception of "letters as entities to operate with". It seems that numerical exploration implies a too heavy burden for the child to fulfil, so she eventually decided to detach numerical meaning from the letter in order to operate with it. This episode points to a potential problem that children may have to eventually overcome, because the pragmatic approach to algebra adopted in this study does not provide children with syntactic rules, rather the approach encourages children to generate such rules through their experience use calculator language. This lack of rules represents on the one hand the advantage of encouraging the children to create their own rules which may fit better their own ways of reasoning. But, on the other hand, this lack of rules represents a potential risk, which may result in children becoming lost unless they are supported by opportune teacher's feedback. This discussion highlights the major role that the teacher has to play within a pragmatic approach to learning and teaching school algebra.

d) Children's awareness of the role of letters within algebraic expressions

The research data suggest that the children would not make some of the most common mistakes reported in the research literature on algebra. Particularly those mistakes about erroneous concatenation of terms, like $3+a=3a$, $a+a=a^2$, $3a+5=8a$ or $a+b=ab$.

Throughout the study the children were asked questions that relate with items that have been used to investigate children's misconceptions (Matz (1980, 1982), Küchemann (1978, 1980, 1981), Booth (1984, 1984a), Peirera (1987), Kieran (1988, 1990, 1992). The questions used in the present study were arranged so that they fit the pragmatic approach adopted in the study. For example, algebraic transformation was put in the situation of "doing something to a calculator program in order to make it equivalent to another program" or judging the feasibility of "make it shorter" a calculator program. Other example is provided in Chapter 5, Interview 3, where the children were asked to interpret the information given by a diagram where letters were used to denote unknown measurements. These diagrams were used to observe whether the child were able to relate their experience of programming the calculator to questions about calculating the perimeter or the area of rectangles involving unclosed algebraic expressions. Though the contexts of paper and pencil and the calculator environment are different, children's reactions in the present study provided data which suggest that they would not make the mistakes reported by previous research when working within a paper and pencil environment. These points are further discussed next.

The first has to do with the fact that the arithmetic notation for multiplication was respected throughout the study, for example, A×3 instead of 3A. The use of this notation helped children distinguish terms containing variables from constant terms. For example, the children were asked whether the program 5×A+3 could be expressed shorter. Children did not present any apparent difficulty to explain that it could not. They then were told that "a pupil from another class says that 5×A+3=8×A". Some children went beyond noticing the error and looked for the specific case when 5×A+3=8×A. For example, Raúl explained that "*if A were 1 the program 5×A+3 could be typed as 8×A ... but only in this case*". When they were asked this question involving two variables children's reactions were similar. For instance: "A pupil from another class says that A×3+5×B gives the same as 8×A×B". Rocío, by visual inspection, answered: "*he is wrong because you have two different letters, they have different values ... if both letters had the same value he is right ... but, why should I make a program with two letters if they will always have same value?*".

Although Rocío's answer is not correct because she did not explicitly provide the number value that the variables must have to make the expressions A×3+5×B and 8×A×B equivalent (A=B=1), her answer does provide us with an interesting episode to analyse. Rocío's response informs us of the progress she has made in understanding the role of letters within a calculator program on the basis of the pragmatic experience she has had using the machine. This experience led her to question the feasibility of having a program involving two variables, not because she considered they could not eventually take the same value, but because it does not make sense to program the calculator using two variables that will always represent the same value ("*why should I make a program with two letters if they will always have same value?*"). A relevant point in terms of the issue we are discussing here, is that Rocío's answer (though wrong) shows that the experience using the calculator has provided her with incipient algebraic notions that she used to judge conjectures involving algebraic equivalence. From a pragmatic perspective she has decided that it does not make sense for A and B to be the same value, in this example, however this does not imply that she has developed a misconception -that A can never be equal to B within an algebraic expression.

The other factor that seemed to have prevented children from making these mistakes is the numerical-based approach to algebra induced by the calculator environment. Children's answers show that they used numerical substitution as a tool for making sense of questions involving algebraic expressions. The support given by numerical substitution prevented them far from interpreting, for example a+b as "a and b", which seemed to produce the error a+b=ab (Tall, 1991). In paragraph 6.2 it was discussed how children face the false identity $A^2+B^2=(A+B)^2$, which implies the analysis of a more complex structured expression.

This result shows that, at this stage, these children do not present the dissociation between arithmetic and algebra reported by Lee and Wheeler (1986, 1989). The apparent strength of numerical-based notions developed by these children suggests that they will not have the problems pointed out by Lee and Wheeler, nevertheless this hypothesis needs to be proved. This fact points at an issue to be attended by future research.

This result also relates to the children's apparent reluctance to accept unclosed expressions reported by Booth (1984). The way in which children approached the questions mentioned above shows that they analysed the algebraic expressions as a whole and not only as descriptors of general arithmetic procedures, which suggest an acceptance of unclosed expressions. A possible explanation for children's apparent readiness to work with unclosed expressions is the fact that from the beginning of the study they used such an expression to program the calculator. They accept these expressions because "that is the way the calculator works". That is, the use of the calculator language seems to imply a tacit children's acceptance of syntactic conventions imposed by its formal code. This point will be taken up again in the next section.

7.3. Pragmatics: The manner in which children come to produce well-formed algebraic utterances to face new problem situations.

An analysis of children's pragmatic approaches to new problem situations makes a dense summary of the notions and strategies they developed throughout the study. The research data seem to conform to Bruner's (1990) view that "with an appreciation of context, the child seems better able to grasp not only the lexicon but the appropriate aspects of the grammar of the language" (p. 71). This section focuses on the apparent influence of different contexts on children's insights when negotiating problem solutions, and is organised around the following features: a) the context provided by inverting linear functions b) visual images as supportive context for children's insights, and, c) the major strategies used by the children throughout the study.

a) The context provided by inverting linear functions

Despite the fact that most of the children could not grasp the canonical way of inverting functions, it is relevant that they developed a pragmatic notion that allowed them to grasp what inverting a function serves for.

Rocío's work in interview 4 illustrates this result. She was asked if the number 877 would appear in the sequence 5, 9, 13, 17, ... After some struggle she said: "*If I input 1 it must give 5 ... I got it, it is 4×B+1! ... But it will not tell me if 877 is in the list ... What I want is the program that undoes this, because I need to know if there is a place for 877 in the list ... If I input 877 and it gives an "exact number" (sic) it will be the place in which 877 appears in the list*". Rocío needed some interviewer's help to find the inverse function, but the relevant point is that she knew in advance what to do with (B−1)÷4. Once she had this program she ran it for B=877 and got 219, and explained: "*219 is an "exact number*" (she meant a whole number), *so 877 is in the list ... it comes in the 219th place*".

Rocío's answer is a result of a pragmatic learning process where she first found that, in the same way in which she could program the calculator to reproduce an "X→Y" table, she can also construct a program to produce a "Y→X" table. Based on this experience she grasped the role of the variable used in a calculator program and the role of the program as a means for representing the relationship between the sets of numbers that was delivered to her in its tabular form.

It seems significant that only a few children were able to find a systematic way of inverting linear functions. This suggests a deep step in linguistic competence: from describing algebraically arithmetic procedures to pure symbolic manipulation. For example, Jennifer and Iván found the standard form when "adjusting the results" was more difficult, like in A×3−1. Jennifer explained her finding as follows: "*I realised that it was not enough to reverse operations, I had to fully reverse it ... I mean, doing last what I did first ... since the last thing was to take 1 away, I added 1 first, then I divided it by 3*

... That's done with parentheses ... they serve to do first one operation then the other". Other children, like Jimena and Erandi learned from them how to do it.

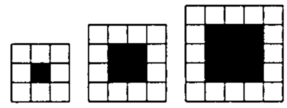
Most of the children used a rather intuitive approach, whether by using the program to be inverted until finding through successive approximations the "x" value corresponding to the given "y" value, or reversing operations and adjusting the results. For example, Diego inverted B×2+1 as B÷2−1+0.5. To obtain the latter expression Diego first reversed the operations (B÷2−1), he then tried with specific values and adjusted the "inverse program". Jimena's work exemplifies the case in which the child found how to invert the function ("*first taking away 1, then dividing by 2 the result*") but could not program the calculator due to a lack of command on using parentheses. She did the involved calculations by hand "*because the calculator did not understand what I want to do*". Once she learned about parentheses she resorted to using them whenever it was needed.

b) Visual images: a supportive context for children's insights

Visual patterns and simple geometrical relations helped children obtain equivalent algebraic representations according to problem constraints. This allowed them to ignore intricate details of syntax.

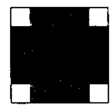
Perhaps the most relevant issue raised when children confronted problem solving is how they resorted to using their incipient notions and strategies as tools for negotiating solutions, particularly those notions about algebraic equivalence and inverting a given function. For example, Jennifer and Jimena, using different strategies, looked for an equivalent expression to obtain fresh information to face a problem situation. The following extracts provide a detailed account of how the context helped children go back and forth between the concrete and the general until they had a better position to face the problem situation.

Fig. 1



Jenny made the program $(A+2)^2-(A\times A)$ to obtain the number of white squares in any member of the sequence shown in figure 1 (worksheet 48). She then became engaged, on her own, in inverting the program to complete the table for the cases where outputs were given. The complexity of the expression did not allow her to do it but she finally found another way of interpreting the number pattern and produced the equivalent program A×4+4. She then built the "inverse program" to complete the task: (A–4)÷4.

Fig. 2



Jenny explained that she made the program $(A+2)^2-(A\times A)$ "*taking away the area of the grey square from the area of the whole square ... The "A" is the length of the grey square* (fig. 1) ... *I found a different program when I saw the shape as a cross*" (fig. 2). This allowed her to count the number of squares surrounding the grey square, then added the four squares on the corners: A×4+4.

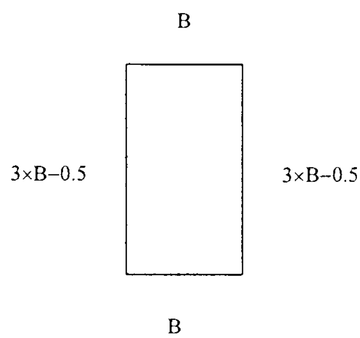
This episode indicates that Jennifer used calculator code as a language that allowed her to cope with different situations. Her approach relates with synonymy, when she could not properly tackle the problem situation with certain expression, she looked for a different expression. Her explanation provides evidence for this: "*I was trying to say the same to the calculator ... I knew I could use any of these programs because both output the same values for the same inputs*".

Jimena's work provides another interesting example. It illustrates how context provides support for children's insights when children are ready to face new problem situations. Similarly to Jenny's approach, Jimena became engaged in inverting a complex expression to complete a table where the outputs were given. Her attempt led her to "uncover" the distributive law. She had made the program (A×2+A×3×2)×53 to compute the cost of any window wooden frame which "they all are three times as high as they are wide and the price per metre is $ 53.00" (worksheet 49). When working out the inverse function she found that (A×2+A×3×2)×53=106×A+318×A. She explained it as follows "*if I* *had two sides which cost 53 each, altogether should cost 106 times the length of one ... I did the same with the other two sides of the window ... I then checked it with the calculator and saw it works*".

Erandi's approach exemplifies another children's strategy. The problem situation was the following:

> "In the sculptures parlour of a certain Art Gallery, the windows have the following features: Their sizes vary, but, in all of them the height is 50 cm less than three times the width. The material used to build the frames costs $62 per metre. Can you program the calculator so that it helps you compute the cost of any window frame? " (worksheet 50).

It is relevant that Erandi, being supported by context, used her incipient knowledge about algebraic simplification in producing an expression that properly describes these relationships. She built the program ((B+B)+(B×6−1))×62 to compute the cost of any window's frame. To explain how she obtained this program Erandi sketched a diagram like the one on the right to explain: "*The width is B ... there is another B on the top ... the height is 50 cm less than three times the width, that is ... 3×B−0.5 ... the opposite side is the same ... Then I computed the perimeter, that's B plus B plus the other two ... they are six times B but one metre less* (pointing at 0.5)... *all this multiplied by 62 gives the cost*". She proceeded similarly to face task in worksheet 49.



Another illustrative situation in this respect is provided by Jimena's solution to the problem briefly posed below:

> Find the length and width that gives the maximum area for the "three sides" rectangle with perimeter 100 metres" (see Appendix, worksheet 54).

Jimena made the program (100–A)÷2×A. The problem statement did not required her to explain her reasoning nor did the researcher. Nevertheless she wrote the following explanation "to assure" that the researcher could understand what she did: "*(100–A)÷2 is going to give the short side, if I multiply it by "A", which is to be the large side, I will get the area*".

Since the children were not given examples that directly relate problem solving with their previous experience (describing number patterns), their achievements document the potential of putting them in the position of learning a language, not in the role of spectator, but through use.

The ways in which the children faced problem solving agrees with Bruner's (1990) claim that "certain communicative functions or intentions are well in place before the child has mastered the formal language for expressing them linguistically" (p. 71).

In terms of this research, the "communicative functions" referred to above are those resources that children had access to when using calculator language to represent quantitative relationships and to set up eventual dependence among variables. The above extracts provide evidence of how children cope with the as yet unknown so as to "request the calculator" to provide fresh information to face certain problem situations. Each excerpt suggests that the use of the calculator seems to help children learn principles rather than algorithms, and how "sensitive to context" these learning processes are.

c) The major strategies used by the children throughout the study

Numerical substitution was the children's strongest strategy to cope with generalisation and algebraic transformation.

The research data discussed earlier in the chapter show that children resorted to exploring the numerical behaviour of algebraic expressions either to cope with algebraic manipulation or express generality.

Children used numerical substitution essentially as a semantic recourse; the data from the study shows that they used it to make sense of the question. This way of working led the children to go back and forth between the general and the particular. This process helped them gain awareness of the general nature of programming expressions and of the role played by specific cases as generic examples. In fact, the children resorted to numerical substitution to explore generality whether to validate or refute conjectures. For example, Diego built the program C×5−C×4 to be equivalent to C×(5−4) (interview 3). He certainly did not have in mind the distributive law (the children did not know about it at all). What he actually did was to "*think of C×(5−4) without parentheses ... I mean C×5−4, if C was 5, C×5 would give us 25, 25−4 doesn't give the same as C×1, I needed minus 20 ... it is C×4 ... it works*". This makes it evident how he went from the general-to the particular-to the general.

7.4. Final Remarks

Children's algebraic attainment throughout the study provides empirical evidence for the approach to learning a new sign system by using it, and for the potential of the graphic calculator as a fundamental support in the fulfilment of this enterprise.

The results discussed in the above sections led to the conclusion that a highly framed calculator environment, based on expressing generality, provides a promising alternative for introducing the algebraic code as a language-in-use. The conjunction calculator-generality helped pupils to make sense of algebraic expressions so as to allow them to cope with basic algebraic activities such as initial symbolic manipulation and negotiating problem solutions.

The work done by the children during the study conforms to Bruner's (1990) hypothesis that "being exposed to the flow of language is not nearly so important as using it in the midst of 'doing'. Learning a language is learning 'how to do things with words' ... The child is not learning simply what to say but how, where, to whom, and under what circumstances" (p. 71).

The research data shows that children's arithmetic background played the role of a 'shared symbol system' which supported them to generate meanings for the algebraic code and gain insights of how to use it while moving within the 'classroom culture' shaped by the calculator based-environment. The term *meaning* is used here as a "culturally mediated phenomenon that depends upon the prior existence of a shared symbol system ... In this sense, symbols depend upon the existence of a "language" that contains an ordered or rule governed system of signs" (Bruner, 1990, p. 69).

This research sheds some light on the enterprise of taking advantage of technology to fill in the gap between arithmetic and algebra. Children's mathematical attainment during the study indicates that, while working with the calculator, they were not only learning about using the specific calculator's facilities to face specific mathematical tasks. Chapter 5 shows that children were progressively developing algebraic notions and strategies that allowed them to use calculator language to face genuine algebraic situations, like transforming a linear algebraic expression to obtain a target expression, simplifying similar terms within linear algebraic expressions, inverting linear functions, and negotiating certain problem solutions using algebraic language. These points are further discussed in the following sections.

A crucial issue in the children-calculator interaction was that children used the calculator's language as a means of *expressing their reasoning*. In this way, the child-calculator interaction constituted an act of communication mediated by a sign system similar to algebraic code. The fact that the child is the one who produces the algebraic expression prepares him/her to receive feedback from the calculator's outputs. This allows the child to anticipate the calculator's outcomes which led him/her to reflect on the task and test his/her own symbolic descriptions. This interaction was supported by the calculator-based environment, which was arranged to provide goal oriented tasks to encourage children to put forward a conjecture.

Jennifer's work illustrates the above, in particular, how arithmetic played the role of a 'shared symbol system' that allowed her to analyse the calculator code. She was engaged in programming the calculator so that "*it first took 1 away, then multiplied this by 3*". Since Jenny did not know about using parentheses she had tried the program B−1×3. She explained she had already made the computations in advance and filled in a table "*but the calculator did not do what I meant ... As I don't think as the calculator does, I said 1-1,.0, then 0×3, 0. The machine said: −1×3, -3, 1-3, -2. I'd like to know how to make the calculator do what I want*".

Jenny's reaction illustrates how the interaction with the calculator encouraged her to go forward and backward from the general to the particular (intending a program that first takes 1 away ... then producing a table by mental calculation).

Though the calculator provides an excellent environment for children to produce algebraic expressions, the machine cannot confront children with the flow of language, so children's utterances revolve within the limits of their own creativity. This limitation of the calculator-based setting was confirmed by the fact that most of the children ran into problems when facing for the first time number patterns where the rule was of the form ax+b (worksheet 4).

It seems that children's previous arithmetic experience led them to express a string with more than two operations following a step by step procedure. It was observed during the first three months of the school year that they performed, for example, 2×3+2, computing first 2×3=6 with the calculator; then 6+2=8. They kept on working in this way despite the fact that they were encouraged to express the whole string of operations in one line and then compute them. This tendency led them to run into problems when facing a rule like "multiplying by 3 and adding 2", they could not conceive how to type a calculator program for representing such an expression, for example, some tried expressions like "3×A=+2", which made the calculator produce a "syntax error" message. This process required the children to do something new: to formalise their method. Their work shows that they accepted the formality of an expression like A×3+2 because "*that's the way that the calculator works*". At this point the researcher/teacher intervention was crucial. This point is further discussed in Chapter 8, section 8.4.

From the beginning of the study the calculator tasks encouraged the children to produce and use algebraic expressions. This process helped them develop the notion of letters as symbols that "*represent any number*" (see, for example Chapter 5, interview 2). The tasks led the children to extend their notion of letters beyond the set of numbers displayed in the table. This was done by asking them to complete a new table using the program they had built. They then needed to find the inputs when outputs were given. This led the children to analyse more carefully how the program proceeds and, consequently, think of the role played by the letter they were using.

Children's work in formats 1 and 3 shows how their use of calculator language evolved. Most of the worksheets include a task that requires the children to write down the rule they found. In format 1 they did it using natural language (i.e. *"I divided by 2 and multiplied by 3"*). But in format 3, most of them used calculator language to answer these questions. Later on, in formats 4, 5 and 6 the pupils showed they were able to confront story-based problems using calculator language.

The results presented in this chapter allows us to see that, along the use of the calculator in the classroom, the articulated activities used in the study help create an environment in which children, through working on the tasks, might develop their first conceptualisations to cope with algebraic issues. Pupil's interaction with the mathematical content was highly framed (in Bruner's sense) intending to scaffold them in the transition from arithmetic to algebra. This approach to teaching and learning algebra strongly contrasts with the approach traditionally adopted in the school, where children's learning heavily depends on the teacher's discourse. These points are further reviewed in Chapter 8 (summary of results).

CHAPTER 8: CONCLUSIONS

Introduction

The conclusions of this thesis were derived from reviewing those aspects that more clearly characterise its nature and research outcomes. The chapter is organised in five sections. The first section, Summary of Results, is aimed at providing an outline which may guide the reader for further analysis and discussion throughout the different chapters of the thesis. The second section, Contributions, discusses the possible contributions of this thesis to the research on the teaching and learning of algebra; this section centres on the most general features of the theoretical approach to teaching and learning algebra adopted in the study and the role played by the graphic calculator in the research. The third section, Limitations, discusses those factors that seemed to have caused the most evident limitations of this research, and some possible alternative actions which may help in overcoming these limitations are put forward. The fourth section, Findings, addresses pedagogical issues which were beyond the aims of this research, in particular, the section discusses some relevant aspects with regard to the role of the teacher within the calculator-based environment. Finally, the fifth section, Further Research, outlines some of the possible lines for future research that may lead to a more solid theoretical and methodological position with regard of taking advantage of the symbolic facilities offered by the graphic calculator.

8.1. Summary of results

Chapter 5 has presented a detailed analysis of two case-study subjects (vertical analysis). Chapter 6 provided an overview of the work done by the other five case-study subjects, the overview was intended to provide a more complete picture of those individual learning events that took place throughout the study. Chapter 7 provided a horizontal analysis focusing on selected episodes that inform how the calculator-based environment shaped children's expressions of general relationships and helps them to develop algebraic notions and strategies.

The present section summarises those research results that can be attributed to the majority of the children that participated in the study. The summary of results is presented in terms of the general aims of the study. As was set out in Chapter 1, this study is intended to investigate:

1. The notions that pupils may develop for algebraic language when they meet it through using calculator code.
2. The extent to which the use of the calculator language helps pupils cope with simplifying similar terms within linear expressions, inverting linear functions, and transforming a linear algebraic expression to obtain a target expression.
3. The strategies that children may develop through working with the calculator.
4. The extent to which the use of the calculator language as a means of expressing general rules governing number patterns, helps children grasp that the algebraic code can be used as a tool for coping with problem situations.

Notions that pupils developed throughout the study (Aim 1)

♦ *The children's notions of letters within algebraic expressions*

Children's work during the study suggests that the use of the calculator language to describe number patterns helped them develop the notion of letters as symbols that "represent any number", and the notion of 'computing devices' for the algebraic expressions used to produce calculator programs.

The data analysed in chapters 5, 6 and 7 provides support for the above claim, the ways in which the children worked out the tasks throughout the study suggest that a key point for them to develop these notions was the fact that the calculator allowed the children to use the programming code both as a means of describing and calculating. Besides, the activity based on describing number patterns helped pupils to link their previous arithmetic experience with the new formal code they were using to program the calculator. The tight link between the tasks and the computing tool helped children find out and formally express the general behaviour of number patterns by producing algebra-like expressions.

The fact that the calculator code is situated within the computing environment helped children to develop a notion of programming expressions as 'calculating' tools. The children's strategy of numerically validating/refuting the symbolic expressions they produced provides evidence of this notion (see Chapter 7, section 7.3c). The ways in which the children coped with expressing rules governing general number relationships suggest that algebraic representation, within the calculator environment, is more than simply encoding what is represented, in this context representation is rather a result of an interaction with the known (arithmetic) according to a goal (make the calculator produce a given table). This activity allowed the children to move from the particular (analysing a particular pair $a \to b$) to the general (verifying the validity of the rule they found for every pair $x \to y$ in the table). This experience was the building block for them to develop the notion of letters as "representing any number", and for algebraic expressions (calculator programs) as "*things that allow you to make the same operations as many times as you want with the numbers you want*" (Rocío[1], interview 2). The ways in which children used the calculator algebra-like code are the best evidence of the notions they developed (see Chapter 5). The children's verbal descriptions illustrate their idiosyncratic conceptions of letters as algebraic entities. Diego's description of the letters he was using to program the calculator encapsulates the notion developed by the children: "*the letter personifies the number I want to make the program with ... they can personify any number, once you put the letter you can input any quantity ... you can run the program for any number you want ... the output changes depending on the number you put in*" (Diego, Interview 1).

The children also showed that they grasped that the value of an algebraic expression does not depend on the letter used. To investigate this feature the children were asked the question: "What does the letter used in the program (A+7)÷2 mean for you? Jenny's answer captures the essence of the notions developed by the children: "*any number ... A can be any number ...*". Going further on this point she was asked: "What will the program (Z+7)÷2 produce?". She said "*it does the same as (A+7)÷2, because when you put the letter in the calculator it doesn't matter if it is A or Z, A may be 1 and Z may be 1 as*

[1] Below average attainment pupil

well ... and so on for any number, it is the same regardless of the letter you put in" (see chapters 5, 6 and 7 for further evidence).

Along with the notion of letters as "*representing any number*" the children became aware that different letters represent different values within the same expression but can also represent the same value within the same expression. The following episode provides support for this assertion. The children were asked whether $(A+B)^2$ could be equal to A^2+B^2. The question was completely new for them, this fact highlights the finding that all of them (including the child with low average attainment) were able to cope with the question by giving specific values to the variables. Some went further and found that the assertion would be correct if "A or B were zero" (see Chapter 7, section 7.1). This finding suggests that the calculator environment helped the children to develop awareness of the role of the letter within an algebraic expression and grasp: (i) the existence of a relationship between the variable and the function through their numerical value, (ii) that the same letter always represent the same value within an expression, (iii) that, in general, different letters represent different values, but can represent the same value in the same expression (a more detailed discussion is presented in Chapter 5, Interview 4; Chapter 5, Final Remarks; Chapter 7, section 7.1).

The notions for letters developed by the children closely relate to the notion of variable. Their work during the study shows that they did not only associate a range of values to a given letter, but they consistently associated such range of values (inputs) to another set of values (outputs), which was the building block for them to explore and verify conjectures about number relationships. This notion is a result of a specific way of working, where the calculator played a central part.

This finding contrasts with Küchemann's (1981) results, which suggests a hierarchical categorisation for children's interpretation of letters: as objects, as unknowns, as generalised numbers, and finally as variables. Following Piagetian principles, Küchemann associated these different roles played by the letters within algebraic expressions to different stages of children's intellectual development, which suggests that the notion of variable can only be grasped when the child reaches the stage of 'formal operations'. Accordingly, the other notions of letters should precede the notion of variable. The results of the present study suggest that children can grasp the notion of letters as variables without having as antecedent other notions (letters as objects, as unknowns or as generalised numbers). Rather, a good number of episodes reported on in chapters 5, 6 and 7, show that the children eventually shifted from the notion of letters as representing a range of numbers, to the notion of letters as unknowns (for example when finding out a specific value for the input when the output is given). This result seems to indicate that the notion of variable does not exclusively depend on issues related with intellectual development, rather it seems that the development of such notion strongly depends on specific teaching approaches and activities.

♦ *The children's notions of algebraic equivalence*

The pupils developed a notion of algebraic equivalence based on exploring the numerical value of algebra-like calculator expressions. This strategy indicates that such a notion of equivalence was derived from the use of calculator language to describe number patterns.

The research data suggests that along with the notion of variable the children were developing a notion of algebraic equivalence, which, in time, proved to be one of the most powerful tools developed by children to cope with a range of different tasks, such as algebraic transformation and inverting linear functions. The research data suggests that the children's experience of using the calculator language to describe number patterns helped them develop a notion of algebraic equivalence. The notion of algebraic equivalence developed by the children was that two calculator programs (linear expressions) are equivalent if "*they produce the same output for the same input*". The work carried out by the children throughout the study shows that this notion helped children to cope with algebraic equivalence within situations that they had never met before. For example, they were able to face questions involving quadratic expressions with two variables. An example of these tasks was given in the above section (i.e., $(A+B)^2=A^2+B^2$?). Other illustrating episodes can be found in Diego's and Jenny's cases, Chapter 5, interviews 2 and 3, and Chapter 7, Section 7.1.

The children's notion of algebraic equivalence was the building block for them to cope with algebraic transformation. The data analysed in Chapters 5, 6 and 7 suggest that such notion of equivalence was a result of their work within a highly framed classroom setting. A detailed discussion of this point is made in Chapter 7, Section 7.1.

The data analysis also allows us to see that this notion of equivalence still needs to be refined. This feature points to the need to investigate the ways in which this numerical-based notion may help/obstruct a more formal approach to algebraic equivalence.

- *The children's notions of priority of operations and use of parentheses*

The children more effectively learned syntax conventions instrumentally, that is as instruments for carrying out certain previously operative functions and objectives. For this to happen it was crucial that the children used calculator language to express their own reasoning; while doing this the children make the involved computations in advance which help them realise that the calculator proceeds differently from them. It was observed during the study that children are not concerned about priority of operations and use of parentheses as long as they work with paper and pencil. In contrast, they were aware of these conventions while working with the calculator.

This fact enhances the role of the calculator environment in helping children gain awareness of algebraic syntax. The children inquired about syntax conventions when they ran into problems to express their reasoning using calculator language. The children's reactions suggest that it was the need to use syntax conventions as means of communication which made children appraise their value (examples of the extent to which children use correctly parentheses are provided in Chapter 5, Children's work, Format 6; this feature is discussed in more detail in Chapter 7, section 7.2).

How children coped with symbolic manipulation (Aim 2)

♦ *Algebraic transformation*

The children's experience of using the calculator code to describe number patterns helped them cope with transforming a linear algebraic expression in order to make it equivalent to a target expression. The ways in which the children tackled these tasks suggests that numerical exploration played a crucial part in the children's approaches to algebraic manipulation.

In order to investigate this issue the children were asked questions such as the following:

> I wanted to type the program B×8 but I made a mistake, instead of that I typed B×7. Can you correct it without deleting anything I typed?

The children's reactions to this question are further discussed in this section when analysing the children's strategies.

♦ *Simplifying similar terms*

The research data suggest that the experience of using the calculator code helped the children cope with simplifying similar terms within linear expressions. A relevant finding in this respect is that similar terms simplification was the only task where children tended to generate misrules.

The typical question asked to the children was like the following: "Can you make shorter the program A×7+A×3?". Children's initial strategy was to give specific values to the variable, they then found, for example, that the program "*just multiply by 10*" and typed the expression A×10 (a more detailed discussion is made in Diego's and Jenny's cases, Chapter 5, Interviews 2 and 3, Chapter 5, Final Remarks, and Chapter 7, section 7.1).

The research data shows that once the children start generating syntax rules they tend to completely rely on them, which in the case of misrules may prevent them from achieving future success in mathematics. Erandi's response to this question captures the essence of the mistake that children tended to make. She found that A×13 was equivalent to A×2+A×3+A×5 because "*the numbers 2, 3, and 5 give 10 altogether, then you must add 3 to 10 because you have three A's there, it gives 13 times A*". A detailed discussion of this feature is made in Chapter 7, Section 7.2.c.

♦ *How children coped with inverting linear functions*

Despite the fact that most of the pupils could not grasp the canonical way of inverting functions, it is relevant that all of them grasped what inverting a function serves for.

The majority of the children used a strategy that consists of reversing the order in which the operations appeared, they then checked their trial and 'adjusted the results'. For example, to invert the rule A×2−1, most of the children produced the program A÷2+1, after running the program they realised that it did not produce the expected results, they then went on adjusting the results (trial-and-refining) until they produced a program like A÷2+1−0.5. Only the children with above average attainment were able to find a systematic way of inverting linear functions (i.e., (A+1)÷2). Nevertheless, all the case-study pupils grasped what inverting a function serves for. An episode with Rocío (a 'below average attainment pupil') provides evidence for this conclusion. Rocío was asked if the number 877 would appear in the sequence 5, 9, 13, 17, ... After some struggle she program the calculator to produce this number pattern (A×4+1), but immediately recognised that the inverse rule was what she needed to answer the question. Rocío needed some help to find the inverse function, but the relevant point is that she knew in advance what to do with (B−1)÷4 (a more detailed discussion is made in Chapter 6, Diego's and Jenny's cases; Chapter 7, Section 7.3,a).

Children's strategies (Aim 3)

Through using the calculator code the majority of the children developed informal strategies which enabled them to cope with algebraic manipulation. These informal strategies suggest that the children used the calculator code as a means of making sense of, and negotiating solutions for new problem situations, rather than using it to repre-

sent a ready made idea. This result seems to contrast with the use of algebraic language within a paper-and-pencil environment, where the students usually use the algebraic code as the *<u>final step</u> of a process of reasoning instead of <u>a means of</u> reasoning.*

When facing the tasks of algebraic manipulation, the children presented the following strategies: (i) trial-and-refining through numerical substitution, and (ii) operating directly with the variable terms. Some of the children used the first strategy, for example, Jenny, Erandi and Diego gave specific values to the variable (i.e., B=1 in B×8, since it did not work they tried with B=2, and so on). By structuring and restructuring their reasoning the children finally realised that the expression they wanted was B×8–B (see Chapter 5 and Chapter 7, section 7.1). Other children (Jimena, Raúl and Iván) operated directly with the variable, for example, B×10–3×B to make B×10 (the source expression) equivalent to B×7 (the target expression). Nevertheless, all of them finally resorted to giving values to the variable when confronting more difficult tasks, for example, transforming a three term linear expression to make it equivalent to a one term linear expression (see Chapter 5, Final Remarks; Chapter 7, Section 7.1). This fact suggests that numerical substitution was the strongest strategy developed by the children.

The trial-and-refining strategy used by the children highlights the role of the calculator language as a mediational tool in the learning of introductory algebra. The ways in which children confronted the task indicate that, in general, they did not use the calculator code to describe a polished idea, the children rather used the calculator code as a means for making sense of the question and progressively refine their reasoning. The research data suggest that the children used the calculator code to communicate with an interlocutor (the calculator), the calculator's feedback helped them to structure and restructure their reasoning. This approach indicates that children used that formal code as a transactional tool, which strongly contrasts with paper-and-pencil work, where algebraic language is, in general, used as the final step within a process of anticipating and establishing the involved relationships. A detailed account of the children's informal approaches is made in Chapter 5, Interviews 2 and 3, Diego's and Jenny's cases; Chapter 5, Final Remarks; Chapter 6, Jimena's case; and Chapter 7, Section 7.1.

How children used the calculator code when confronting problem situations (Aim 4)

With different level of attainment, all the case-study children were able to use the calculator code to cope with algebra word problems.

The experience of describing number patterns using the calculator language helped pupils make sense of traditional algebra word problems and provided pupils with a formal code to negotiate problem solutions. This result strongly contrasts with outcomes obtained in studies that have investigated the effects of introducing school algebra through describing number patterns (Stacey, 1989; Herscovics, 1989; Arzarello, 1991; MacGregor and Stacey, 1993; Stacey and MacGregor, 1996). These studies reported students' difficulties in generating algebraic rules from patterns and tables. MacGregor and Stacey (1996) concluded that "a patterns-based approach does not automatically lead to better understanding; the way students are taught and the practice exercises that they do may promote the learning of a routine procedure without understanding" (p. 3). They reported that students were able to recognise and describe the involved quantitative relationships, but their approach was rather a rhetorical description (in the sense of Harper, 1987) which leave children far from describing the problem algebraically.

This result involves a relevant contribution of this thesis, therefore the rest of this section discusses those issues that provide an explanatory framework of why children were able to take the step between using calculator language to describe number patters, and using the calculator code to negotiate problem solutions (further evidence is provided in Chapter 5, Format 6: children's work; Chapter 6; Chapter 7, section 7.3).

There are various factors that may explain the strong contrast observed in the findings of this thesis and the outcomes obtained by MacGregor and Stacey. What seems the most immediate explanation is that the students reported by MacGregor and Stacey worked within a paper and pencil environment, where apparently natural language is the most immediately available tool for children to structure their reasoning when facing a problem situation. MacGregor and Stacey (1996) found that most of the students guided their procedures by natural language descriptions. Nevertheless, they conclude that this approach hardly helps them structure an algebraic expression to properly describe the relationships between two variables.

This contrasts with the fact that the calculator programming language is situated within the computing environment, this feature allows the user to produce algebra-like expressions and use them both to describe numerical relationships and to calculate with these expressions. The operative nature of the calculator language places the children within a milieu where algebraic formulation becomes an inherent part of the problem situation to be solved. The use of the calculator language leads children to <u>describe the relationships in a problem situation operationally</u>, <u>even if they make this description in natural language</u>. When working with the calculator the children do not look for the relationship between the "x" and "y" variables to find out the underlying pattern (which was the question used by MacGregor and Stacey); the calculator environment allows us to make the same question so that the children are led to think of what operations they can make with the input in order to produce the correspondent output. The data obtained from the present study provide evidence for this assertion: when the children were asked to use natural language to describe the relationship involved in a number pattern, they used expressions which always include an operative description, for example, "*I multiplied by 2*" which they expressed as A×2 to program the calculator. When the rules were more sophisticated, they ignored the constraint of using natural language and directly used calculator language, for example, 3×A+2, "*because the calculator language makes it easier to explain this*" (see, for example, Chapter 5, Diego's and Jenny's work, Format 1). This use of the calculator code allowed the children to focus on the operational structure of the calculator expressions, whether describing number patterns or describing the relationships involved in story-based problem situations. This operational approach does not necessarily occur when the children work within a paper and pencil environment, where natural language is the immediate means of communication, this situation seems to lead children to see the use of algebraic code as a sophisticated teacher's imposition.

The mathematical content and the sequence of the tasks used during the study provide another source of explanation for pupils' achievements in problem solving. As was discussed in Chapter 4, the content of the tasks addressed general mathematical notions as opposed to specific topics. Specifically the tasks addressed the following issues: expressing generality (formats 1 and 2), algebraic equivalence (Format 3), inversion (Format 4), decreasing linear functions (Format 5), and problem solving (Format 6). A close look at the tasks provides an explanatory framework for how the children developed such notions and strategies which finally they exhibited when coping with negotiating problem solutions. This review is intended to provide support for the conclusion that these tasks shaped a didactic 'route to algebra problem solving'.

The results discussed in Chapters 5, 6 and 7 show that the tasks in Formats 1 and 2 allowed the introduction of calculator language as a language-in-use. The main feature of these tasks was to place children in the position of using the calculator code to fulfil their communicative intention. The tasks guided the children to gain awareness of the inherent generality of the algebraic expressions they were using from the beginning of the study. The tasks in these formats also introduced children to the use of parentheses and the idea of inverse function (finding the input when the output was given).

The tasks in Format 2 introduced children to the notion of algebraic equivalence. The children's work showed that, spontaneously, they construct equivalent expressions operating with the independent terms (for example, $3\times B+4=3\times B+8\div2$). That is, they did not spontaneously operate with terms containing variables to construct equivalent expressions. Nevertheless, during individual interviews they showed they were able to operate with algebraic expressions when the task was changed to that of transforming an algebraic expression to make it equivalent to a target expression. The work carried out by the pupils in Format 6, where they produced expressions as $((A\times3)\times2+(A\times2))\times53$, suggests that the experience of transforming algebraic expressions was a key point in helping children gain awareness of the feasibility of using expressions of the form $ax+bx+c$ (see Chapter 5, Format 6).

The tasks in Format 4 required the children to deal with inverting linear functions. These tasks encouraged the children to look for a systematic way of inverting linear functions, and to refine their notion about using parentheses. The children's responses to questions about number sequences (see worksheets 46-48, Chapter 5, Format 6) show how their previous experience with inverting linear functions helped them cope with problem situations which required them to apply their incipient notion of inverse relationships.

Finally, the tasks in Format 5 introduced the children to new number patterns generated by linear decreasing functions. These tasks were intended to introduce children to the use of algebraic expressions to describe part-whole relationships. The children's responses to worksheet 55, where they produced expressions like ((100–A)÷2×A, provide evidence of the extent to which their experience in producing decreasing functions influenced the ways in which they used the algebraic code to negotiate solutions.

As a final remark, it must be mentioned that the notions and strategies that children used when negotiating problem solutions can provide a basis for further development of algebraic ideas. As was discussed in Chapters 5 and 7, the pupils will still need to refine some notions and strategies in order to become more competent users of the algebra-like calculator language. They sometimes became confused when explaining how they produced symbolic expression to describe the relationships within algebra problems.

8.2. Contributions to the teaching and learning of algebra.

The children's algebraic attainment throughout the study provides empirical evidence for the approach to learning a new sign system by using it, and for the potential of the graphic calculator as a fundamental support in the fulfilment of this enterprise.

This study provides empirical evidence for a pragmatic approach to the teaching and learning of algebra that offers a promising vein for exploiting the symbolic facilities offered by the graphic calculator. The study stresses the potential of technological devices in envisioning new possible routes to school algebra (this issue is discussed in more detail in Chapter 7, section 7.4).

The theoretical background used in the study contributes to fill in the gap between a constructivist approach to using technology in the classroom and the need of a more informed teaching intervention. The results of this study show that the way in which Bruner's research findings were recast gives promising guiding lines for using calculators in the mathematics classroom. Perhaps the most relevant pedagogical feature of the material used in the study consists of having motivated children in a way that once they grasped what the few first worksheets required them to do, they were in the position of working out a set of progressively complex tasks on their own. The research data shows that, along with the crucial support provided by the calculator, the tasks used in the study motivated a favourable pupils' attitude toward mathematics. This pupils' attitude was a critical issue in getting to work the approach to algebra as a language-in-use. In fact, language acquisition could hardly take place without a children's attitude of willing receipt. The fact that children have shown that readiness to work indicates that the tasks allowed a productive teacher-pupil and pupil-calculator interaction, where both teacher's and pupil's presuppositions and intentions were clearly exposed so as to create a solid platform of communication.

The set of articulated teaching materials used in the study made it possible to document a lengthy children's learning process which addresses fundamental algebraic activities: expressing generality, symbolic manipulation and problem solving. Though the tasks used in the study are widely known by the mathematics educator's community (much of it due to the work done by Mason, 1988), the materials were rearranged in order to take advantage of the calculator facilities. The research data suggests that the children's achievements came as a result of the particular way of using the calculator adopted in this study. The work carried out by the children throughout the study shows that they were able to cope with issues that have been earlier reported on as problematic, such as the acceptance of unclosed expressions, the connection between arithmetic and algebra.

the use of parentheses, and most important: the use of algebraic code to negotiate problem solutions (see chapters 5, 6 and 7).

8.3. Limitations

Methodological issues

One of the limitations of the present study derives from its qualitative nature, which does not allow us to generalise its results. Consequently, the research outcomes provided by this study should be taken as an empirical evidence which documents a promising approach to using graphic calculators in the teaching and learning of algebra.

Another methodological factor that limited the present research was the fact that, in order to investigate the learning processes that take place when the study of algebra begins, the fieldwork had to be carried out during a short time period. The time constraints made it impossible to follow the children within a longer time period, which would have let us investigate the children's informal strategies more thoroughly. The results obtained by the present research showed that allowing children to develop and recreate their own strategies provides them with a powerful tool which seemed to place them in a better position to cope with the learning of algebra. These results suggest that investigating in depth the children's informal strategies must provide more solid outcomes which would surely strengthen and refine the conclusions derived form this study.

The data drawn from this study shows that the approach to algebra as a language in use helped children use the calculator language to negotiate solutions for algebra word problems, and develop a notion of algebraic equivalence that allowed them to confront tasks involving algebraic manipulation, such as simplifying similar terms, transforming an algebraic expressions to make it equivalent to another, and inverting linear functions. The study suggests that the children have reached what seems a promising starting point to confront more traditional school algebra.

Nevertheless, there are still many aspects of algebra which the children did not encounter within this study. In particular, a number of research questions should be faced in order to refine/consolidate the results of the existing study. Among the major issues leading to further research are the following:

In which sense may the pragmatic approach to teaching and learning algebra help/obstruct:

a) children's learning of formal syntactic rules for algebraic manipulation?
b) children's learning when confronting algebra problem solving which involves using equations?
c) children's learning of more formal methods for establishing algebraic equivalence (for instance, algebraic transformation)?
d) children's learning of a more formal approach to the notion of function?
e) children's learning of graphs as another way of representing number relationships?
f) children's learning of formal features concerning negative numbers?
g) children's understanding that a conjecture about number relationships cannot be validated on the basis of the results obtained from specific cases?
h) children's understanding of the value of counterexamples as a means of proof/refuse mathematical conjectures?

The above research questions tell us about the potentialities and limitations of the present study. About its potential because these questions give an account of the wide range of algebraic topics that children experienced during a relatively short school time (about 18 hours). These questions tell us of the limitations of the present study because they bring to light issues that still have to be investigated before setting up stronger claims about the potential of the approach to learning and teaching of algebra as a language in use, and the support provided by the symbolic capabilities of the graphic calculator to fulfil such an enterprise.

Theoretical issues

The theoretical referent shaped for the study gave encouraging results in terms of providing promising guidelines to take advantage of the symbolic facilities offered by the graphic calculator. Nevertheless, the theoretical background needs further development. So far, the theory provided strong support for shaping the teacher-pupil and pupil-calculator interactions (the design of the tasks), and for interpreting the ways in which the children tackled specific problem situations (syntactically, semantically or pragmatically). Nevertheless, the theory, perhaps due to the researcher's limitations, could not be suitably exploited to analyse the children's mathematical achievements in terms of language acquisition. The causes of this theoretical limitation should be thoroughly reviewed in order to search for a stronger link between the linguistic-based approach and the mathematical realm.

8.4. Findings: Pedagogical issues

Though the calculator provides an excellent environment for children to produce algebraic expressions, the machine cannot confront children with the flow of language, so children's utterances revolve within the limits of their own creativity. This limitation of the calculator-based setting was confirmed by the fact that most of the children ran into problems when facing for the first time number patterns where the rule was of the form ax+b (worksheet 4).

A key factor in 'getting to work' the calculator-based environment was the presence of an 'expert user of calculator language'. The role of the teacher was particularly relevant when children faced the task of expressing number patterns governed by rules of the form ax+b. This issue seems to be on the border line between children's arithmetic experience and their entry into algebra. Expressions like A×3+2 were like "new words" that children needed to learn from a more competent peer, these "new calculator words" mark a crucial point in their way to formalising their methods.

It seems that the children's previous arithmetic experience led them to express a string with more than two operations following a step by step procedure. It was observed during the first three months of the school year that they performed, for example, 2×3+2, computing first 2×3=6 with the calculator; then 6+2=8. They kept on working in this way despite the fact that they were encouraged to express the whole string of operations in one line and then compute them. This tendency led them to run into problems when facing a rule like "multiplying by 3 and adding 2", they could not conceive by themselves how to type a calculator program for representing such an expression, for example, some tried expressions like "3×A=+2", which made the calculator produce a "syntax error" message.

At this point the researcher/teacher intervened suggesting that they type the rule "altogether, as a one-piece string of operations". From then on, the children's prior reluctance to work in this way practically disappeared. This suggests that the formality of calculator language helped them realise the value of these "new words" mainly due to their instrumental function. Children's reactions resemble Bruner's (1980) claim that "mastering a language involves not only knowing how to string together propositions, but also how to meet the conditions on the appropriate making of utterances" (p. 161).

8.5. Further research

The fact that the researcher acted as the teacher during the fieldwork induces particular conditions which must have influenced the research. In order to obtain more elements whether to reinforce or refine the theoretical background and the results of the research, it appears suitable to carry out a replica of the study where the researcher plays only the role of observer. This new research stage should provide an opportunity to test both the theoretical and methodological approaches when the classroom activity takes place under conditions derived from the mathematics teacher's conceptions and his/her own interpretation of the calculator-based approach.

The present study was carried out following the hypothesis that the symbolic capabilities of the graphic calculator can be exploited to introduce the learning of algebraic code as a language-in use, but the acquisition of a language takes place within a long time period (maybe an entire life!). This hypothesis suggests an important vein for further research: to carry out a longitudinal study throughout the secondary school mathematics, a study aimed at investigating the ways in which the specific approach to algebra as a language-in-use may influence children's learning when they face more formal aspects of algebra.

Finally, a more specific topic that deserves further research is that of the effects of introducing algebra as a language-in-use on the learning of negative numbers. As was discussed in Chapter 5, the children developed interesting informal strategies to cope with tasks that required them to operate with negative numbers. Time constraints led the researcher to the decision of exploring just a few issues regarding the children's approaches to negative numbers being supported by the use of calculators. Thus, an interesting step for future research is to investigate the potential of the graphic calculator as an aid in the learning of negative numbers.

REFERENCES

Arzarello, F., 1991. "Pre-algebraic problem solving". Paper presented at seminar on problem solving. Viana do Castelo, Portugal, April 27-30.

Behr, M., et al., 1976. "How children view equality sentences". (PMDC Technical Report No. 3). Florida State University (ERIC Document Reproduction Service No. DE144802).

Bell, .A., 1992. Purpose in School Algebra. Plenary Lecture to the Algebra Working Group, *Sixteenth Conference of the International Group for Psychology of Mathematics Education, North American Chapter*. University of New Hampshire, Durham, NH (USA).

Benveniste, E., 1971. *Problems in general linguistics*. Coral Gables: University of Miami Press.

Booth, L., 1981. "Child-Methods in Secondary Mathematics". *Educational Studies in Mathematics*. Kluwer Academic Publisher, 12, 29-41.

Booth, L., 1984. "Misconceptions leading to error in elementary algebra". *Journal of Struct. Learning*, 8, pp. 125-138.

Booth, L., 1984a. *Algebra: Children's Strategies and Errors in Secondary Mathematics Project*. NFER-NELSON, London.

Brousseau, G., 1986. "Fondements et méthodes de la didatique des mathématiques". *Recherches en didactique des mathématiques*, 7(2), 33-115.

Brown, R., 1977. Introduction to C. E. Snow and C. A. Ferguson (Eds.), *Talking to Children: Language input and acquisition*. Cambridge University Press, Cambridge.

Bruner, J., 1964. "The course of cognitive growth". *American Psychologist*, 19, 1-15.

Bruner, J., 1969. *Hacia una teoría de la Instrucción*. UTEHA, México.

Bruner, J., 1978. The role of dialogue in Language Acquisition. In A. Sinclair, R.J. Jarvella and W.J.M. Levelt (Eds.). *The Child's Conception of Language*. Springer-Verlag, Berlin.

Bruner, J., 1980. *The social context of language acquisition*. Witkin Memorial Lecture, Educational Testing Service, Princeton, New Jersey.

Bruner, J., 1982. "The formats of Language Acquisition". *American Journal of Semiotics*, Vol. 1, No. 3 ,1- 16.

Bruner, J., 1983. *Child's talk.* New York: Norton.

Bruner, J., 1985. Vygotsky: a historical and conceptual perspective. In J. V. Werstch (Ed), *Culture, Communication and Cognition: Vygotskian perspectives.* Cambridge University Press.

Bruner, J., 1990. *Acts of Meaning.* Harvard University Press, Cambridge, Massachusetts.

Burill, G., 1992. "The Graphing Calculator: A Tool for Change". In J. T. Fey and C. R. Hirsch (Eds.), *Calculators in Mathematics Education*, Yearbook. National Council of Teachers of Mathematics, Reston, VA.

Carpenter, T., Corbitt, M., Kepner, H., Jr., Lindquist, M., and Reys, E., 1981. *Results from the second mathematics assessment of the National Assessment of Educational Progress.* National Council of Teachers of Mathematics, Reston, VA.

Clement, J., 1982. "Algebra word problem solutions: Thought processes underlying a common misconception". *Journal for Research in Mathematics Education*, 13, 16-30.

Clement, J., Lockhead, J., and Monk, G., 1981. "Translation difficulties in learning mathematics". *American Mathematical Monthly*, 88, 286-290.

Clement, J.; Lochhead, J.; and Soloway, E. (1979*). Translations between symbol systems: Isolating a common difficulty in solving algebra word problems.* COINS, Technical report No. 79-19. Amherst: Department of Computer and Information Science, University of Massachusetts.

Collis, K., 1969. "Concrete operational and formal operational thinking in mathematics". *Australian Mathematics Teacher*, 25(3), 77-84.

Collis, K., 1971. "Study of concrete and formal reasoning in school mathematics". *Australian Journal of Psychology*, 23, 289-296.

Collis, K., 1973. *Study of concrete and formal operations in school mathematics.* Unpublished PhD Thesis, University of Newcastle, N.S.W., Australia.

Collis, K., 1975a. "Cognitive development and mathematics learning". *Psychology of Mathematics Education Workshop*, Chelsea College.

Collis, K., 1975b. "The development of formal reasoning". Draft paper, University of Newcastle, N.S.W., Australia.

Collis, K., 1975c. *A study of concrete and formal operation in school mathematics: A Piagetian view point.* Melbourne: Australian Council for Educational Research.

Confrey, J., 1993. The role of technology in reconceptualising functions and algebra. Invited address to the Group for the Psychology of Mathematics Education, North American Chapter. Asilomar; California.

Cuoco, A., 1995. Computational Media to Support the Learning and Use of Functions. In A. DiSessa, C. Hoyles and R. Noss (Eds.), *Computers and Exploratory Learning.* Springer-Verlag.

Chaiklin, S., and Lesgold, S., 1984. "Prealgebra students' knowledge of algebraic tasks with arithmetic expressions". Paper presented at the Annual Meeting of the *American Educational Research Association*, New Orleans, LA.

Chalouh, L., and Herscovics, N., 1988. "Teaching algebraic expressions in meaningful ways". In A. Coxford (Ed.), *Algebraic concepts in the curriculum, K-12.* National Council of Teachers of Mathematics, Reston, VA.

Chomsky, N., 1957. *Learning and the Structure of Language.* University of Chicago.

De Corte, E. and Verschaffel, L., 1989. "Logo: a Vehicle for Thinking". In B. Greer, G. Mulhern (Eds.), *New Directions in Mathematics Education.* London: Routledge.

Demana, F., and Leitzel, J., 1988. "Establishing fundamental concepts through numerical problem solving". In A. F. Coxford and A. P. Shulte (Eds.), *The Ideas of Algebra K-12*, Yearbook. National Council of Teachers of Mathematics, Reston, VA.

diSessa, A., 1995. Computers and Exploratory Learning: Setting the scene. In A. diSessa, C. Hoyles, and R., Noss (Eds.), *Computers and Exploratory Learning.* Springer-Verlag.

Dreyfus, T., and Halevy, T, 1988. "A case study of pupil-computer interaction". *ICME 6*, Budapest, Hungary.

Fillmore, C., 1977. "The case for case reopened in syntax and semantics". In P Cole and J. M. Sadock (Eds.), *Grammatical Relations*, Vol. 8. New York: Academic Press.

Greeno, G. J., 1983. "Conceptual entities". In D. Genter and A. L. Stevens (Eds.), *Mental Models*, 227-252.

Harper, E., 1987. "Ghost of Diouphantus". *Educational Studies in Mathematics*. Kluwer Academic Publisher, 18, 75-90.

Healy, L., Hoyles, C., and Sutherland, R., 1990. *The role of peer group discussion in mathematical environments* (Final Report). Leverhulme Trust.

Hector, J. H., 1992. "Graphical Insight into Elementary Functions". In J. T. Fey and C. R. Hirsch (Eds.), *Calculators in Mathematics Education*, Yearbook. National Council of Teachers of Mathematics, Reston, VA.

Hembree, Ray, and Donald J. Dessart, 1986. "Effects of Hand-held Calculators in Precollege Mathematics Education: A Meta-Analysis". *Journal of Research in Mathematics Education*, 17: 83-99.

Hembree, Ray, and Donald J. Dessart, 1992. "Research on Calculators in Mathematics Education". In J. T. Fey and C. R. Hirsch (Eds.), *Calculators in Mathematics Education*, Yearbook. National Council of Teachers of Mathematics, Reston, VA.

Herscovics, N., 1989. Cognitive obstacles encountered in the learning of algebra. In S. Wagner and C. Kieran (Eds.), *Research Issues in the Learning of Algebra*. National Council of Teachers of Mathematics, Reston, VA.

Hillel, J. and Kieren, C., 1988. "Schemas used by 12-year olds in solving selected turtle geometry tasks". *Recherches en didactique des mathematiques*, 8, 1, 61-102.

Hoyles, C. and Noss, R., 1992. "A Pedagogy for Mathematical Microworlds". *Educational Studies in Mathematics*. Kluwer Academic Publisher, 23, 31-57.

Hoyles, C. and Sutherland, R., 1989. *Logo Mathematics in the Classroom*. London: Routledge.

Johnson, D. C., 1991. "Algorithmics in School Mathematics: Why, What and How?". In I. Wirszup and R. Streit (Eds.), *Proceedings of the Third UCSMP International Conference on Mathematics Education*. National Council of Teachers of Mathematics, Reston, VA.

Kaput, J., 1979. Mathematics and Learning: Roots of epistemological status. In J. Lochhead, and J. Clement, (Eds.), *Cognitive Processes Instruction*. Franklin Institute Press.

Kaput, J., 1989. Linking Representations in the Symbol Systems of Algebra. In S. Wagner and C. Kieran (Eds.), *Research Issues in the Learning of Algebra*. National Council of Teachers of Mathematics, Reston, VA.

Karmilof-Smith, A., 1979. *A Functional Approach to Child Language*. Cambridge University Press, Cambridge.

Kieran, C., 1981. "Concepts associated with the equality symbol". *Educational Studies in Mathematics*. Kluwer Academic Publisher, 12(3), 317-326.

Kieran, C., 1988. "Relationships between Novices' View of Algebraic Letters and Their Use of Symmetric and Asymmetric Equation-Solving Procedures". In J.C. Bergeron and N. Herscovics (Eds.), *Proceedings of the Fifth Annual Meeting of PME-NA*, Vol. 1, Montreal, Canada.

Kieran, C., 1990. Cognitive Processes Involved in School Algebra. In P. Nesher and J. Kilpatrick (Eds.), *Mathematics and Cognition: A Research Synthesis* by the International Group for the Psychology of Mathematics Education. Cambridge University Press, Cambridge.

Kieran, C., 1991. *CARAPACE* (software). Canada: University of Quebec at Montreal.

Kieran, C., 1992. The Learning and Teaching of Algebra. In D. A. Grouws (Ed.), *Handbook of Research on Mathematics Teaching and Learning*. National Council of Teachers of Mathematics, Reston, VA.

Krutetskii, V. A., 1976. *The Psychology of Mathematical Abilities in Schoolchildren*. Chicago University Press.

Küchemann, D.E., 1978. "Children's Understanding of Numerical Variables". *Mathematics in School*, 7(4), 23-6.

Küchemann, D.E., 1980. *The Understanding of Generalised Arithmetic by Secondary School Children*. Unpublished Doctoral Dissertation, Chelsea College, University of London.

Küchemann, D.E., 1981. Algebra. In K. Hart, (Ed.) *Children's Understanding of Mathematics: 11-16*. London: Murray, 102-19.

Lee, L., and Wheeler, D., 1986. "High school students' conception of justification in algebra". *Proceedings of the 8th Annual Meeting PME-NA*. East Lansing, Michigan, USA.

Lee, L., and Wheeler, D., 1989. "The arithmetic connection". *Educational Studies in Mathematics*. Kluwer Academic Publisher, 20, 41-54.

Leitzel, J. R., 1989. Critical considerations for the future of algebra instruction. In S. Wagner and C. Kieran (Eds.), *Research Issues in the Learning and Teaching of Algebra*. National Council of Teachers of Mathematics, Reston, VA.

MacGregor, M. and Stacey, K., 1992. "A comparison of patterns-based and equation solving approaches to algebra". In B. Southwell, K. Owens and B. Perry (Eds.), *Proceedings of the Fifteenth Annual Conference of the Mathematics Education* Research Group of Australasia, MERGA, Brisbane, pp. 362-371.

MacGregor, M. and Stacey, K., 1993. "Seeing a Pattern and Writing a Rule". *Proceedings of the XVII PME Annual Meeting*, Japan, pp 181-188.

Mason, J., 1980. "When is a Symbol Symbolic?". *For the Learning of Mathematics*, 1 (2), pp. 8-12.

Mason, J., 1988. *Expressing Generality*. Open University, Milton Keynes.

Mason, J., 1989. "Mathematical Abstraction Seen as a Delicate Shift of Attention". *For the Learning of Mathematics*, 9 (2), pp. 2-8.

Mason, J., 1991. *Supporting Primary Mathematics: Algebra*. Open University, Milton Keynes.

Mason, J., 1993. "Expressing generality and roots of algebra". Open University. Draft paper presented at PME-Algebra Working Group.

Mason, J., Gower, N., Graham, A., and Pimm, D., 1985. *Routes to/Roots of Algebra*. Open University, Milton Keynes.

Mason, J.; Burton, L; and Stacey, K., 1984. *Thinking Mathematically*. Addison Wesley, London.

Matos, J. and Carreira, S., 1994. Cognitive processes and representations involved in applied problem solving. In C. Sloyer, I. Huntly and W. Blum (Eds.), *Advances and Perspective in the Teaching of Mathematical Modelling and Applications*. Chichester: Ellis Horwood.

Matos, J., 1995. Spreadsheets as a tool for mathematical modelling. In A. diSessa; C. Hoyles, and R. Noss (Eds.), *Computers and Exploratory Learning*. Springer-Verlag.

Matz, M., 1980. "Towards a computational theory of algebraic competence". *Journal of Mathematical Behaviour*, 3, 93-166.

Matz, M., 1982. Towards a Process Model for High School Algebra Errors. In Sleeman and Brown (Eds.), *Intelligent Tutoring Systems*. Academic Press, London.

Miles, M. and Huberman, A., 1984. *Qualitative Data Analysis, a Sourcebook of New Methods*. SAGE Publications, London.

Nelson, K. (1975). The syntagmatics and paradigmatics of conceptual development. In S. Kuczaj (Ed.), *Language, Cognition and Culture*. Lawrence Erlbaum, Hillsdale, NY.

Nelson, K., 1973. "Structure and Strategy in Learning to Talk". *Monographs of the Society for Research in Child Development*, 38, Serial no. 149.

Neuirth, E., 1995. Visualising Formal and Structural Relationships with Spreadsheets. In A. diSessa, C. Hoyles and R. Noss (Eds.), *Computers and Exploratory Learning*. Springer-Verlag.

Noss, R., 1985. *Creating a Mathematical Environment through Programming: A Study of Young Children Learning Logo*. Doctoral Dissertation, Chelsea College, University of London.

Noss, R., 1986. "Constructing a Conceptual Framework for Elementary Algebra through Logo Programming. *Educational Studies in Mathematics*. Kluwer Academic Publisher, 17, 335-357.

Noss, R., 1995. "Programming and Mathematics: Equation or Inequality". Draft paper prepared for *A Framework for Constructing a Vision of Algebra*. Algebra Working Group, National Council of Teachers of Mathematics, Reston, VA.

Papert, S., 1980. *Mindstorms*. The Harvester Press Limited, Sussex, UK.

Peirera, L., 1987. "Error patterns and strategies in algebraic simplification". In Bergeron and. Herscovics (Eds.), *Proceedings of the 11th International Conference PME*, Vol. 1, pp. 331-337, Montreal, Canada.

Piaget, J., 1985. *La construcción de lo real en el niño*. Grijalbo, México.

Piaget, J., 1988. *La psicología de la Inteligencia*. Grijalbo, México.

Pimm, D. 1987. *Speaking Mathematically*. Routledge. London.

Polya, G., 1965. *Cómo plantear y resolver problemas*. Trillas, México.

Rojano, T. and Sutherland, R.(1992). "A new approach to algebra: Results from a Study with 15 year-olds". *Cuarto Simposio Internacional sobre Investigacion en Educación Matemática.* PNFAPM/Univ. Aut. de Ciudad. Juárez, Vol. Enseñanza del Álgebra.

Ruthven, K., 1990. "The Influence of Graphic Calculator use on Translation from Graphic to Symbolic Forms". *Educational Studies in Mathematics.* Kluwer Academic Publisher, 21, 431-450.

Ruthven, K., 1992. "Personal Technology and Classroom Change: A British Perspective". In J. T. Fey and C. R. Hirsch (Eds.), *Calculators in Mathematics Education.* Yearbook, National Council of Teachers of Mathematics, Reston, VA.

Ruthven, K., 1992a. "Calculator Strategies". *Micromath*, Spring.

Ruthven, K., 1993. Technology and the Rationalisation of Teaching. In C. Keithel, and K. Ruthven, (Eds.) Learning from Computers: Mathematics Education and Technology. Springer-Verlag, London.

Ruthven, K., 1993a. "Changing key: The calculator and the transition to algebraic thinking". *Paper presented at the ESRC Seminar Algebraic Processes and the Role of Symbolism*, Institute of Education, University of London.

Ruthven, K., 1993b. "Developing algebra with the supercalculator". *Micromath*, Spring.

Ruthven, K., 1995. "Graphic Calculators as a personal resource in the lower secondary school". *Micromath* Summer.

Searle, J. R., 1969. *Speech Acts: An Essay in the Philosophy of Language.* Cambridge University Press, Cambridge.

Sfard, A., 1991. "On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin". *Educational Studies in Mathematics.* Kluwer Academic Publisher, 22, 1-35.

Sfard, A., 1992. "The development of algebra- Confronting historical and psychological perspectives". *ICME 7*, Quebec, Canada.

Sfard, A., 1994. "The gains and the pitfalls of reification - The case of algebra". *Educational Studies in Mathematics.* Kluwer Academic Publisher, 26, 191-228.

Shatz, M., 1982. Relations between cognition and language acquisition. In W. Deutsch (Ed.), *The Child's Construction of Language.* New York and London: Academic Press.

Shumway, R., 1988. "Technology, Mathematics and the International Congress". Special Report on *ICME-6*, Budapest, Hungary.

Shumway, R., 1990. "Supercalculators and the Curriculum". *For the Learning of Mathematics*, 10(2), 2-9.

Skinner, B.F. 1957. *Verbal Behaviour*. Prentice Hall, Englewood, NJ.

Snow, C. E. and. Ferguson, C. A (Eds.), 1977. *Talking to Children: Language Input and Acquisition*. Cambridge University Press, Cambridge.

Stacey, K., 1989. "Finding and using patterns in linear generalising problems". *Educational Studies in Mathematics*. Kluwer Academic Publisher, 20, 147-164.

Stacey, K., and MacGregor, M., 1996. Curriculum Reform and Approaches to Algebra. To be published in Sutherland R. and Rojano T. (Eds.) *Algebraic Processes and Structure*, PME Working Group,. Kluwer.

Sutherland, R. and Rojano, T., 1993. "Spreadsheet Approach to Solving Algebra Problems". *Journal of Mathematical Behaviour*, 4, 353-383.

Sutherland, R., 1987. *A Longitudinal Study of the Development of Pupils' Algebraic Thinking in a Logo Environment*. Unpublished Doctoral Dissertation, Institute of Education, University of London.

Sutherland, R., 1989. "Providing a computer based framework for algebraic thinking". *Educational Studies in Mathematics*. Kluwer Academic Publisher, 20(3), 317-344.

Sutherland, R., 1993. "Connecting Theory and Practice: Results from the Teaching of Logo". *Educational Studies in Mathematics*. Kluwer Academic Publisher, 24, 317-344.

Sutherland, R., 1995. "The Influence of Teaching on the Practice of School Algebra: What can we learn from Work with Computers?". *Rendiconti del Seminar Matematico* del Universitia e del Politecnio di Torino.

Sutherland, R., 1996. Reflections on the Role of the Computer in Developing Algebraic Understanding. To be published in Sutherland R. and Rojano T. (Eds.) *Algebraic Processes and Structure*, PME Working Group,. Kluwer.

Sutherland, R., 1996a. "A dramatic shift of attention: From Arithmetic to Algebra". Personal communication.

Sutherland, R.; and Rojano, T.; Mochón, S.; Jinich, E.; Molyneux, S., 1996. "Mathematical modelling in the sciences through the eyes of Marina and Adam". *Proceedings of the PME XX*, Valencia, Spain.

Sutherland, R.; Rojano, T. Ursini, S.; Molyneux, S.; Jinich, E., 1996. "Ways of solving algebra problems:.The influence of school culture". *Proceedings PME XX*, Valencia, Spain.

Suydam, M. N., 1982. *The use of calculators in pre-college education: Fifth annual state-of-the-art review*. Columbus, OH: Calculator Information Center. (ERIC Document Reproduction Service No. ED 220 273).

Tall, D. and Thomas, M., 1991. "Encouraging versatile thinking in algebra using the computer". *Educational Studies in Mathematics*. Kluwer Academic Publisher, 22(2), 125-147.

Tall, D., 1989. Different cognitive obstacles in a technological paradigm. In S. Wagner and C. Kieran (Eds.), *Research Issues in the Learning and Teaching of Algebra*. National Council of Teachers of Mathematics, Reston, VA.

Tall, D., 1993. "School Algebra and the Computer". *Micromath*, Spring, 38-41.

Ursini, S., 1994. *Pupil's approaches to different characterisations of variable in Logo*. Unpublished PhD Thesis, University of London Institute of Education.

Ursini, S., 1996. General Methods: A way of entering algebra word. To be published in Sutherland R. and Rojano T. (Eds.), *Algebraic Processes and Structure*, PME Working Group,. Kluwer.

Usiskin, Z., 1987. Lessons learned for the first eighteen months of the secondary component of the University of Chicago School Mathematics Project. In I. Wirzup and R. Streit (Eds.), *Developments in school mathematics around the world*. National Council of Teachers of Mathematics, Reston, VA.

Vonder Embse, C. B., 1992. "Concept Development and Problem Solving Using Graphing Calculators in the Middle School". In J. T. Fey and C. R. Hirsch (Eds.), *Calculators in Mathematics Education*, Yearbook. National Council of Teachers of Mathematics, Reston, VA.

Wagner, S., 1979. "Mathematical variables and verbal "variables": An essential difference". *Proceedings of the Third International Conference for the Psychology of Mathematics Education*. Warwick, UK.

Wagner, S., 1981. "Conservation of equation and function under transformations of variable". *Journal for Research in Mathematics Education*, 12, 107-118.

Wagner, S., 1981a. "An analytical framework for mathematical variables". *Proceedings of the Fifth Conference of the International Group for the Psychology of Mathematics Education*. Grenoble, France. Laboratoire IMAG.

Wagner, S., 1981b. "Investigating Learning Difficulties in Algebra". *Proceedings of the Third Annual Meeting of the PME-NA*.

Winter, J., et al., 1991. "Graphic Calculator Working Group". *Mathematics Teaching*, June.

APPENDIX 1

TASKS

WORKSHEET 1 (Format 1)

NAME ______________________________________ DATE ______________

I programmed my calculator to do the following:

?
1
5
?
2
6
?
3
7
?
4
8
?
5
9

What will the calculator output if I input the number 5? ______What if I input 10 ?____

What happens if I input 70?____________

Which operations did you use to make these predictions? ________________________

____

Can you program the calculator to do that? Write your program in the following space.

Use your program to complete the following table.

17	35.02	89.73	107.06	299.1	307.09		
						511	613.03

WORKSHEET 2 (Format 1)

NAME ________________________________ DATE ____________

I programmed my calculator to do the following:

```
?
7
              14
?
8
              16
?
9
              18
?
15
              30
?
18

              36
```

What will my program output if I input 5 into the calculator? ____ If I input 25 instead?__________________ Or if I input 17? __________________________

Which operations did you use to get these answers?__________________________
____

Can you program the calculator to do that? Write your program in the following space.

Use your program to complete the following table.

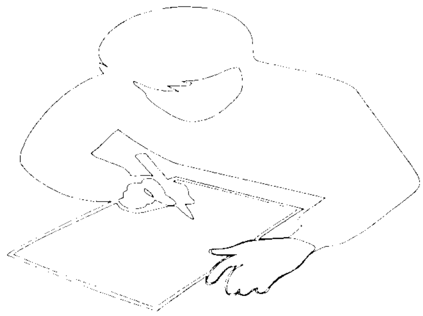
25	37.03	59.83	117.18	136.1	200.79		
						551	653.38

WORKSHEET 3 (Format 1)

NAME ______________________________________ DATE ______________

2.5	7.5
3.1	9.3
4	12
4.2	12.6
5.3	
6.2	
	47.4
73	

Complete the following table.



Which operations did you use to complete the table? ____________________________
____

Can you program the calculator to find the right hand column numbers? Write your program in the following space.

Does your program output the numbers shown in the table? ______________________

Use your program to complete the following table.

9	17	18.04	47.01	50.4	63.9		
						89.1	92.4

WORKSHEET 4 (Format 1)

NAME ________________________________ DATE ____________

?
1.1
3.2

?
2.5
6

?
3
7

?
4.3
9.6

?
5
11

I programmed my calculator to do the following:

What will the calculator output if I input the number 50? ____________________

What if I input 81? __________ What happens if I input 274? ______________

Which operations did you use to make these predictions? ____________________

____

Can you program the calculator to do that? Write your program in the following space.

Use your program to complete the following table.

1.3	2.8	14	50	81	274		
						325	420

WORKSHEET 5 (Format 1)

NAME ______________________________________ DATE ______________

I programmed my calculator to do the following:

? 1	1
? 2	3
? 3	5
? 4	7
? 5	9

What will my program output if I input 6 into the calculator? ___________________
If I input 7 instead? ________________ Or if I input 15? ____________________

Which operations did you use to get these answers?___________________________
____

Write a program that copies mine. Show your program in the right hand square.

Is my program's output identical to yours?_________________________________

Use your program to complete the following table.

10	11	15	19	27	24.9	136.5	259.14

WORKSHEET 6 (Format 1)

NAME ________________________________ DATE ______________

Fill in the blanks in the following table.

-10	-9.7	-7.8	-6.2	-5.3	-4.6	-0.7	0	1.3	12.4
-9.5	-9.2	- 7.3	-5.7						

Can you program the calculator to do the work?

Once you have done that show your program in the right hand square.

Use your program to copy the previous table. Did you get the same numbers?____

____

Use your program to complete the following table.

-20	-14.7	-13.8	-12.3	-10.8	-9.6	-0.5	0

WORKSHEET 7 (Format 1)

NAME ____ DATE _______________

I am making this table.
Can you complete it?

-15	-14.5	-12.4	-10.2	-5.8	-4.6	-0.9	0	1.3	15.4
-16.5	-16	-13.9	-11.7						

The numbers in the previous table follow a rule. Can you state it? ________________

____

____

Can you program said rule in your calculator?

Write it in the square once you have finished.

Use your program to copy the previous table. Did you get the same numbers?________

____

Use your program to complete the following table.

-20		-13.8		-10.83		-0.5	
	-17.3		-11.9		-9.72		10

WORKSHEET 8 (Format 1)

NAME ______________________________ DATE ______________

I programmed my calculator to do the following:

?
10.5
5.25
?
14.42
7.21
?
15.3
7.65
?
16.7
8.35
?
20.1
10.05

If the input is 6, what will my calculator output? ______________________

If the input is 19.3, what will my calculator output? ______________________

If the input is 56, what will my calculator output? ______________________

If the input is 177, what will my calculator output? ______________________

Explain your answers.____

____

Can you program the calculator to do the same thing? When you are finished write your program in the following space.

WORKSHEET 9 (Format 1)

NAME ____ DATE ______________

I programmed my calculator to do the following:

?
6
9
?
8
12
?
14
21
?
15
22.5
?
18
27

What will my program output if I input 10 into the calculator? ____________________

If I input 13.4 instead? ________________ Or if I input 15.6? ____________________

Explain your answers.____

____

Can you program the calculator to do the same thing? When you are finished write your program in the following space.

Use your program to complete the following table.

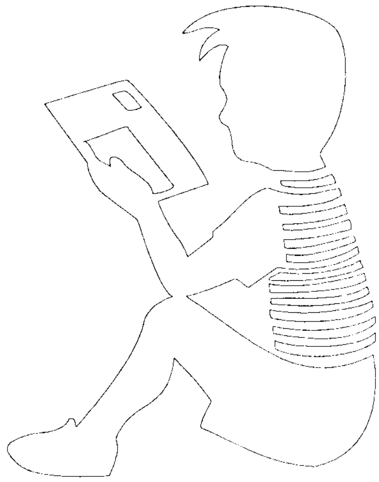
20		35		44		72	
	33		57		75		123

WORKSHEET 10 (Format 1)

NAME ____ DATE ________________

I am completing this table. Can you find the missing numbers?

4	4.04
6	6.06
9	9.09
10	10.1
12	12.12
15.5	
17.8	
19.2	
20.4	
50.2	



The numbers in the table above were found following certain rule. Which is this rule?

____

____

Can you program the calculator using such rule? Write down your program on the right.

Did your program produce the same numbers as the ones that appear in the table? _____

Use your program to complete the following table.

1		3.1		9		32	
	2.222		4.343		12.12		38.784

WORKSHEET 11 (Format 1)

NAME ______________________________________ DATE ______________

I programmed my calculator to do the following:

?
7
23
?
9
29
?
10
32
?
12
38
?
16
50

What will my program output if I input 11 into the calculator? ____________________
If I input 13.4 instead? ____________________

The calculator output 76. Which number did I input? _________

Explain your answers.____
____

Can you program the calculator to do the same thing? When you are finished write your program in the following space.

Use your program to complete the following table.

1		5.1		9.4		22	
	17		20.9		33.5		83

WORKSHEET 12 (Format 1)

NAME ______________________________ DATE ____________

I programmed my calculator to do the following:

?
7
20
?
7.5
21.5
?
8.2
23.6
?
9
26
?
9.6
27.8

What will my program output if I input 11 into the calculator? ________ If I input 12 instead? ________ Or if I input 15.6? ____________________

The calculator output 17.5. Which number did I input? ______

Explain your answers.____

____

Can you program the calculator to do the same thing? When you are finished write your program in the following space.

Use your program to complete the following table.

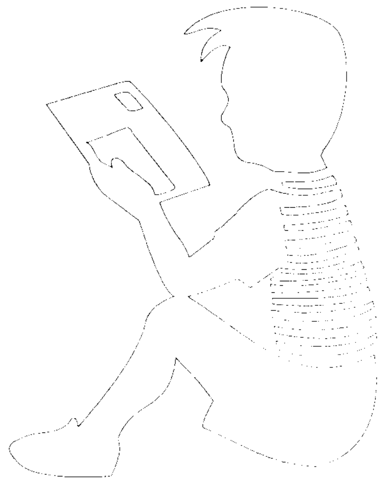
3		5.1		9.4		22	
	17		15.2		32.6		80

WORKSHEET 13 (Format 1)

NAME ______________________________________ DATE ______________

I programmed my calculator to do the following:

?
10
2.5
?
15
3.75
?
20
4
?
25
6.25
?
30
7.5



What will my program output if I input 56 into the calculator?

____

The calculator output 87. Which number did I input? ______

Explain your answers.____

____

Can you program the calculator to do the same thing? When you are finished write your program in the following space.

Use your program to complete the following table.

3		5.1		9.4		22	
	1		1.65		2.7		8.75

WORKSHEET 14 (Format 1)

NAME: ______________________________ DATE: ____________

?	
2	
	5
?	
3	
	7.5
?	
4	
	10
?	
5	
	12.5

I programmed my calculator to do the following:

If the input is 6, what will my calculator output? ______________________

If the input is 7, what will my calculator output? ______________________

If the input is 55, what will my calculator output? ______________________

Which operations did you use to get these answers? ______________________

____

Can you program the calculator to do the same thing? When you are finished write your program in the following space.

Use your program to complete the following table.

3		5.1		9.4		12.2	
	8.5		15.5		23.5		35

WORKSHEET 15 (Format 1)

NAME ______________________________________ DATE ______________

?
0.15
0.015

?
0.27
0.027

?
0.3
0.03

?
1.5
0.15

?
2.03
0.203

I programmed my calculator to do this:

What will the calculator output if I input the number 10____________________________

If the calculator outputs 37, what number did I input? ____________________________

Which operations did you use to make these predictions? __________________________

____

Can you program the calculator to do that? Write your program in the this space.

Use your program to complete the following table.

3		5.1		9.4		12.2	
	0.4		0.63		1.18		35

WORKSHEET 16 (Format 2)[1]

GUESS MY PROGRAM

Date ____

Name of the programmer: ____________________________________

Who is guessing? ____

IMPORTANT: IF ITS YOUR TURN TO MAKE THE PROGRAM, YOU MUST WRITE IT DOWN. YOU MUST GIVE ONLY A FEW HINTS FOR GUESSING. WHEN YOUR PARTNER MAKES A GUESS, YOU MUST VERIFY IT.

Clues for guessing:

Input	Output
1	
3	
5	
8	
10	
20	

If you are guessing you must program your calculator.
If your program works show it in the square below.

How many times did you try to guess? ____________Did you need extra clues?____

If so, which ones? ____

____

____

[1] Worksheets 17 to 20 are identical to this worksheet.

WORKSHEET 21 (Format 3)

NAME ______________________________________ DATE _______________

1. I programmed my calculator to do the following:

?	
1	
	4
?	
1.5	
	6
?	
3	
	12
?	
5	
	20

If the input is 7, what will my calculator output? ________________________

What if I input 10 ? __________________ What happens if I input 70? _____________

Which operations did you use to make these predictions? ________________________

____

2. Can you program the calculator to do the same thing? Test your program in your calculator once you have it. Write it below if it works.

3. Can you program the calculator to do it in a different way? Test your program in your calculator and write it below.

Can you program the calculator to do it in other different ways? Test them, if they work, write them below.

WORKSHEET 22 (Format 3)

NAME ______________________________________ DATE _______________

?	
2	
	3
?	
4	
	6
?	
8	
	12
?	
10	
	15

1. I programmed my calculator to do the following:

If the input is 5, what will my calculator output? ________________________

What if I input 6 ? __________________ What happens if I input 15? _____________

Which operations did you use to make these predictions? _________________________

____

2. Can you program the calculator to do the same thing? Test your program in your calculator once you have it. Write it below if it works.

3. Can you program the calculator to do it in a different way? Test your program in your calculator and write it below.

4. Can you program the calculator to do it in other different ways? Test them, if they work, write them below.

WORKSHEET 23 (Format 3)

NAME ______________________________ DATE ______________

1. I programmed my calculator to do the following:

?	
1	
	.25
?	
2	
	0.5
?	
3	
	0.75
?	
4	
	1

If the input is 5, what will my calculator output? ______________________

What if I input 6 ? ______________ What happens if I input 15? ___________

Which operations did you use to make these predictions?

____

____

____

2. Can you program the calculator to do the same thing? Test your program in your calculator once you have it. Write it below if it works.

3. Can you program the calculator to do it in a different way? Test your program in your calculator and write it below.

4. Can you program the calculator to do it in other different ways? Test them, if they work, write them below.

WORKSHEET 24 (Format 3)

NAME ____ DATE ________________

1. I programmed my calculator to do the following:

?
-1
-0.5
?
3
1.5
?
7.4
3.7
?
17
8.5

If the input is 5, what will my calculator output? ___________________________

What if I input 6 ? __________________ What happens if I input 15? ______________

Which operations did you use to make these predictions? _________________________

____

2. Can you program the calculator to do the same thing? Test your program in your calculator once you have it. Write it below if it works.

3. Can you program the calculator to do it in a different way? Test your program in your calculator and write it below.

4. Can you program the calculator to do it in other different ways? Test them, if they work, write them below.

WORKSHEET 25 (Format 3)

NAME ______________________________________ DATE _______________

I programmed my calculator to do the following:

?	
4	
	6
?	
6	
	9
?	
10	
	15
?	
18	
	27

If the input is 12, what will my calculator output? ____________________________

What if I input 20 ? __________________ What happens if I input 50? _____________

Can you program the calculator to do that? Test your program in your calculator once you have it. Write it below if it works.

Can you program the calculator to do it in a different way? Test your program in your calculator and write it below.

Can you program the calculator to do it in other different ways? Test them, if they work, write them below.

WORKSHEET 26 (Format 3)

NAME ____ DATE _______________

I programmed my calculator to do the following:

?	
1	
	6
?	
3	
	10
?	
5	
	14
?	
9	
	22

If the input is 10, what will my calculator output? ______________________________

What if I input 20 ? __________________ What happens if I input 50? ______________

2. Can you program the calculator to do that? Test your program in your calculator once you have it. Write it below if it works.

3. Can you program the calculator to do it in a different way? Test your program in your calculator and write it below.

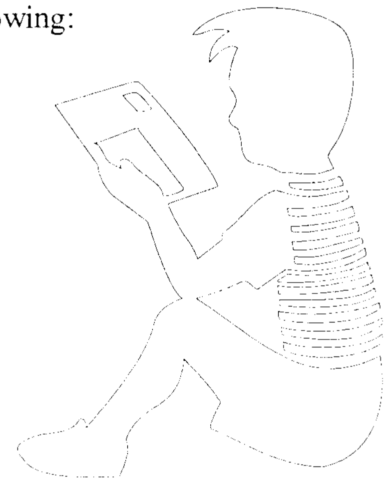
4. Can you program the calculator to do it in other different ways? Test them, if they work, write them below.

WORKSHEET 27 (Format 3)

NAME ____ DATE __________________

?	
15	
	15
?	
16	
	16
?	
17	
	17
?	
18	
	18

1. I programmed my calculator to do the following:



If the input is 1, what will my calculator output? ________________________________

What if I input 2 ? ____________________ What happens if I input 55? _______________

2. Can you program the calculator to do that? Test your program in your calculator once you have it. Write it below if it works.

3. Can you program the calculator to do it in a different way? Test your program in your calculator and write it below.

4. Can you program the calculator to do it in other different ways? Test them, if they work, write them below.

WORKSHEET 28 (Format 3)

NAME ____ DATE ________________

1. I programmed my calculator to do the following:

?
1
1
?
3.2
10.24
?
5
25
?
9
81

If the input is 7, what will my calculator output? ______________________________

What if I input 10 ? __________________ What happens if I input 25? ______________

2. Can you program the calculator to do that? Test your program in your calculator once you have it. Write it below if it works.

3. Can you program the calculator to do it in a different way? Test your program in your calculator and write it below.

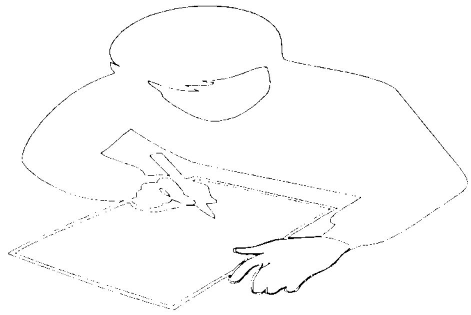
4. Can you program the calculator to do it in other different ways? Test them, if they work, write them below.

WORKSHEET 29 (Format 3)

NAME ______________________________________ DATE ______________

I produced the following program:

? → N: 3.5 × N



If the input is 8, what will my calculator output? ____________________________

What if I input 14 ? __________________ What happens if I input 29? ______________

2. Can you program the calculator to do that? Test your program in your calculator once you have it. Write it below if it works.

3. Can you program the calculator to do it in a different way? Test your program in your calculator and write it below.

4. Can you program the calculator to do it in other different ways? Test them, if they work, write them below.

WORKSHEET 30 (Format 3)

NAME ________________________________ DATE ______________

I produced the following program:

? → Z: 1.02. × Z

If the input is 7, what will my calculator output? ______________________

What if I input 10 ? ______________ What happens if I input 26.7? __________

2. Can you program the calculator to do that? Test your program in your calculator once you have it. Write it below if it works.

3. Can you program the calculator to do it in a different way? Test your program in your calculator and write it below.

4. Can you program the calculator to do it in other different ways? Test them, if they work, write them below.

WORKSHEET 31 (Format 4)

NAME ______________________________ DATE ______________

1. There is a roll of wire in my grandfather's hardware shop which is sold as it weights.

I programmed the calculator in order to help him register how much wire is left. If you input the amount of **sold** wire, the program outputs the amount of wire that is **left**.

?	
1.7	
	8.3
?	
2.4	
	7.6
?	
3.1	
	6.9
?	
4.06	
	5.94
?	
5.2	
	4.8

According to the table above, how many kilos of wire there are in a roll of wire?

____

2. Can you program your calculator so that it produces the same thing as mine? Test your program in your calculator and write it below.

3. Use your program to complete the following table

2.83	3.03	3.5	4.8				
				5.01	6.2	7.04	7.32

WORKSHEET 32 (Format 4)

NAME ______________________________ DATE ____________

1. I programmed my calculator to do the following:

?	
1.3	
	18.7
?	
2.5	
	17.5
?	
3.8	
	16.2
?	
4.4	
	15.6
?	
5.9	
	14.1

If the input is 6, what will my calculator output? ______________________

What if I input 7 ? ______________ What happens if I input 9? ____________

Which operations did you use to make these predictions? __________________

____

2. Can you program the calculator to do the same thing? When you are finished write your program in the following space.

3. Use your program to complete the following table.

2.83	3.03	- 3.5	- 4.8				
				5.01	6.2	27.04	37.32

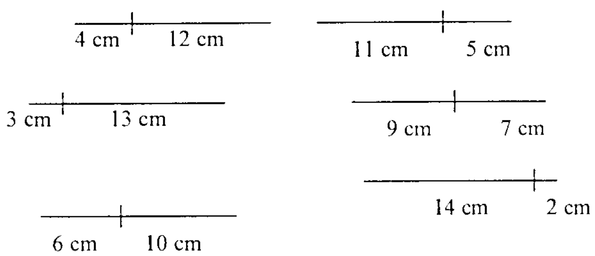
4. What happens when you input a negative number? ______________________

____

WORKSHEET 33 (Format 4)

NAME ______________________________ DATE ____________

I have several pieces of wire, all them with a length of 16 cm. I want to cut them into two parts in different ways. Some possibilities are shown below.



1. Can you program the calculator so that if you input the length of one piece of wire the machine outputs the length of the other piece?

Write your program down in the following space.

2. Explain how did you reason to produce that program ____________________

3. Use your program to complete the following table.

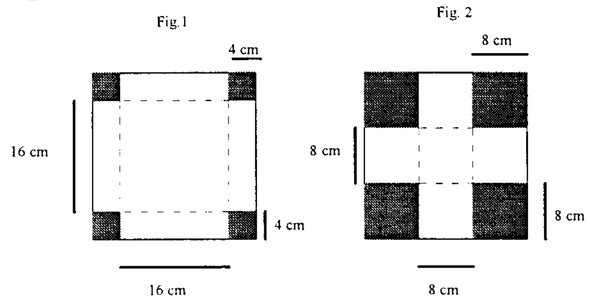
1.7		3.8		6.8		7.9	
	12.8		14.9		15.6		17.4

WORKSHEET 34 (Format 4)

NAME ______________________________ DATE ____________

I want to make a box with a square piece of cardboard. I can make the box by cutting squares off the corners and bending up the pieces that are left jutting out.

The dimensions, the **base** and **height** of the box, are determined by the length of the sides of the squares I cut off. Figures 1 and 2 show two possible ways of making the box.



1. How long is the side of my cardboard square? ______________ What is its area? __________ Write down the operations you used. ______________________________

____

2. Complete the following table:

	According to Figure 1	According to Figure 2
Area of the base		
Height of the box		
Volume of the box		

3. I would like a box with the **biggest possible volume**. Having only one cardboard, I can try only once. Can you program your calculator to find the volume of any box made like this? Write your program here.

4. Use your program to find the desired base and height for a box with the biggest volume. Write the answers here.

Base	Height	Volume

WORKSHEET 35 (Format 4)

NAME ______________________________ DATE ____________

1. I programmed my calculator to do the following:

?	
1	
	0
2	
	-1
3	
	-2
4	
	-3
5	
	-2

If the input is 6, what will my calculator output? ______________________

What if I input 7 ? ______________ What happens if I input 9? ___________

What about if I input 17? ______________________________

How did you obtain these results? ______________________________

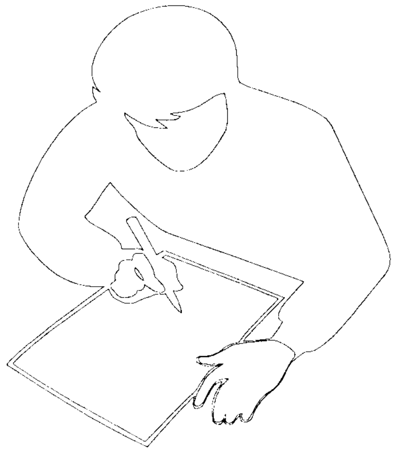
____

2. Can you program the calculator to do the same thing? When you are finished write your program in the following space.

WORKSHEET 36 (Format 4)

NAME ______________________________ DATE ____________

1. I programmed my calculator to do the following:



?	
1	
	4
2	
	9
3	
	14
4	
	19
5	
	24

If the input is 6, what will my calculator output? ______________________

What if I input 7 ? ______________ What happens if I input 9? __________

What about if I input 17? ______________________________

How did you obtain these results? ______________________________

____

2. Can you program the calculator to do the same thing? When you are finished write your program in the following space.

WORKSHEET 37 (Format 4)

NAME ______________________________ DATE ____________

1. I made a program that produced the following:

?	
1	
	0.5
2	
	-0.5
3	
	-1.5
4	
	-2.5
5	
	-3.5

If the input is 6, what will my calculator output? ______________________

What if I input 7 ? ______________ What happens if I input 9? ___________

What about if I input 17? ______________________________

How did you obtain these results? ______________________________

____

2. Can you program the calculator to do the same thing? When you are finished write your program in the following space.

WORKSHEET 38 (Format 4)

NAME ________________________________ DATE ____________

?	
1	
	8.5
2	
	6.5
3	
	4.5
4	
	2.5
5	
	0.5

1. Can you guess what program I made? It produced the following table

If the input is 6, what will my calculator output? ________________________

What if I input 7? ______________ What happens if I input 9? ____________

What about if I input 17? ______________________________________

How did you obtain these results? ________________________________

____

2. Can you program the calculator to do the same thing? When you are finished write your program in the following space.

WORKSHEET 39 (Format 4)

NAME ______________________________ DATE ____________

1. I programmed my calculator to do the following:

```
?
1
                 0
2
                 0
3
                 0
4
                 0
5
                 0
```

If the input is 6, what will my calculator output? ______________________

What if I input 7 ? ______________ What happens if I input 9? ____________

What about if I input 17? ______________________________

How did you obtain these results? ______________________________

____

2. Can you program the calculator to do the same thing? When you are finished write your program in the following space.

WORKSHEET 40 (Format 4)

NAME ______________________________ DATE ____________

1. I programmed my calculator to do the following:

?	
1	
	-1
2	
	-2
3	
	-3
4	
	-4
5	
	-5

If the input is 6, what will my calculator output? ______________________

What if I input 7 ? ______________ What happens if I input 9? ____________

What about if I input 17? ______________________________

How did you obtain these results? ______________________________

____

2. Can you program the calculator to do the same thing? When you are finished write your program in the following space.

WORKSHEET 41 (Format 5)

NAME: ______________________________ DATE: ____________

I programmed my calculator
to do the following:

?
10.4
4.9
?
16
10.5
?
19
13.5
?
23.5
18
?
37
31.5

1. Guess my program. Try your program on your calculator, if it works like mine write it below.

2. Can you program your calculator to do the **opposite** of what my program does? It should output the number that the following table shows:

4.9	10.5	13.5	18	31.5
10.4	16	19	23.5	37

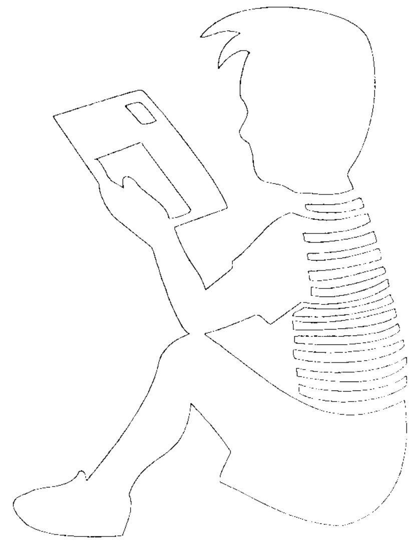
If you could write a program that undoes mine, show it in the square below.

WORKSHEET 42 (Format 5)

NAME: ______________________________ DATE: ____________

?
11.4
17.5
?
11.4
17.5
?
19
25.1
?
23.1
29.2
?
38
44.1
?
50
56.1

I programmed my calculator to do the following:



1. Guess my program. Try your program on your calculator, if it works like mine write it below.

2. Can you program your calculator to do the **opposite** of what my program does? It should output the number that the following table shows:

17.5	25.1	29.2	44.1	31.5
11.4	19	23.1	38	37

If you could write a program that undoes mine, show it in the following space.

3. Explain what you did to build a program that "undoes" mine.

____

____

____

____

WORKSHEET 43 (Format 5)

NAME: ______________________________________ DATE: _____________

?
0.13
0.26
?
0.17
0.34
?
0.65
1.3
?
3.8
7.6
?
9.28
18.56

1. Guess my program. Try your program on your calculator, if it works like mine write it below.

2. Can you program your calculator to do the **opposite** of what my program does? Try your program on your calculator, if it works write it below.

3. Can you program your calculator to do the **opposite** of what the following program does?

   ? → M : M × 3

   Try your program on your calculator, if it works write in the following space.

4. Can you program your calculator to do the **opposite** of what the following program does?

   ? → N : N × 1.5

   Try your program on your calculator, if it works write in the following space.

WORKSHEET 44 (Format 5)

NAME: ______________________________ DATE: ____________

I programmed the calculator to do the following:

?
3
5
?
7
13
?
10
19
?
11
21
?
15
29

1. Guess my program. Try your program on your calculator, if it works like mine show it below.

2. Can you program your calculator do the **opposite** of what my program does? It should output the number that the following table shows:

5	13	19	21	29
3	7	10	11	15

If you could write a program that undoes mine, show it in the following space.

3. Can you program your calculator to do the **opposite** of what the following program does?

? → B : B×3+1

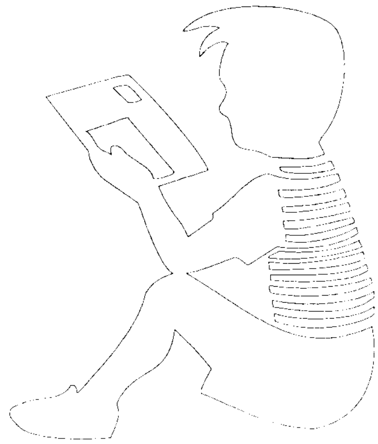
Try your program on your calculator, if it works write in the following space.

WORKSHEET 45(Format 5)

NAME: ______________________________________ DATE: _____________

I programmed the calculator to do the following:

```
?
2
                 4
?
5
                25
?
7
                49
?
8
                64
?
10
               100
```



1. Guess my program. Try your program on your calculator, if it works like mine show it in the following space.

2. Can you program your calculator to do the **opposite** of what my program does? It should output the number that the following table shows:

4	25	49	64	100
2	5	7	8	10

If you could write a program that undoes mine, show it in the following space.

3. Produce programs that "undo" every program in the following list. Test the programs you made in the calculator, if they work show them in the spaces below.

? → A : A × 1.5 + 1	
? → K : 0.5 × K - 1	
? → X : 0.25 × X + 2	

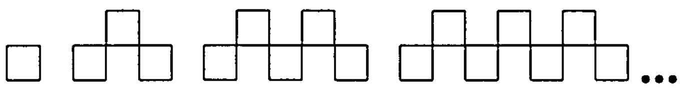
4. Did you find a method for "undoing programs"? Explain what your method consists of. ____

____

WORKSHEET 46 (Format 6)

NAME ____ DATE _____

Look at the following shapes.



1. Draw the next two shapes of the sequence in the space below.

2. How many squares would be needed to draw shape number 17? ______________________

3. How many squares would be needed to draw shape number 100? ________________

4. Explain how you reasoned to answer question 2 and 3. ____________________
____
____
____
____

5. Can you program your calculator to complete the following table?

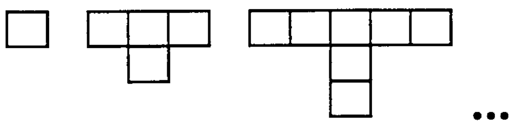
The shape's number in the sequence.	Number of squares needed to draw it.
48	
75	
123	
	351
	411
	507

Write your program here.

WORKSHEET 47 (Format 6)

NAME ____ DATE ______

Look at the following shapes.



1. Draw the next two shapes of the sequence in the space below.

2. How many squares would be needed to draw shape number 9?

____

3. How many squares would be needed to draw shape number 17?

____

4. Explain how you reasoned to answer question 2 and 3. ____________________
____
____
____

5. Can you program your calculator to complete the following table?

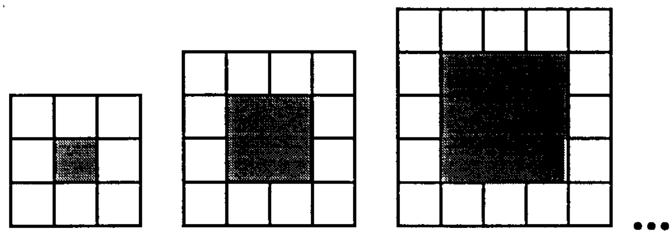
The shape's number in the sequence.	Number of squares needed to draw it.
48	
75	
123	
	427
	469
	601

Write your program here..

WORKSHEET 48 (Format 6)

NAME ____ DATE ______

Look at the following shapes.



1. Draw the next two shapes of the sequence in the space below.

2. How many squares would be needed to draw shape number 27?

   ____

3. How many squares would be needed to draw shape number 40?

   ____

4. Explain how you reasoned to answer question 2 and 3. ____________________

____

____

____

5. Can you program your calculator to complete the following table?

The shape's number in the sequence.	Number of squares needed to draw it.
48	
75	
123	
	704
	772
	840

Write your program here.

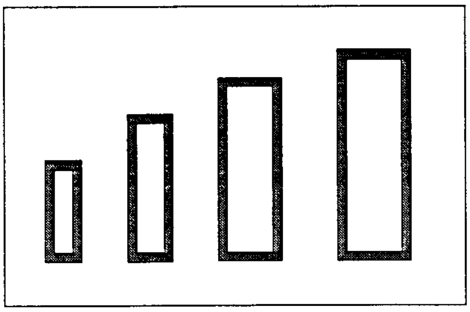
WORKSHEET 49 (Format 6)

NAME ______________________________ DATE ______________

WINDOWS

In the sculptures parlour, of a certain gallery, windows have the following features:

Their sizes vary, but, they all are three times as high as they are wide.



1. Can you complete the table?

Width of the window	0.75 m	0.86 m	1.28 m		
Height of the window				3.51 m	4.23 m

2. The window's frames are wooden, their price per metre is $ 53.00.

   a) What is the price of the frame if the window is 1.5 metres tall? ______________
   ____

   b) Which operations did you do to compute the cost? ______________
   ____

3. Can you write a program that gives you the frame's price for any of the windows in the gallery? Show your program below.

4. Use your program to complete the following table.

Width of the window	0.68 m	0.80 m	0.95 m	0.98 m	1.15 m	
Price of the frame	$	$	$	$	$	$ 530

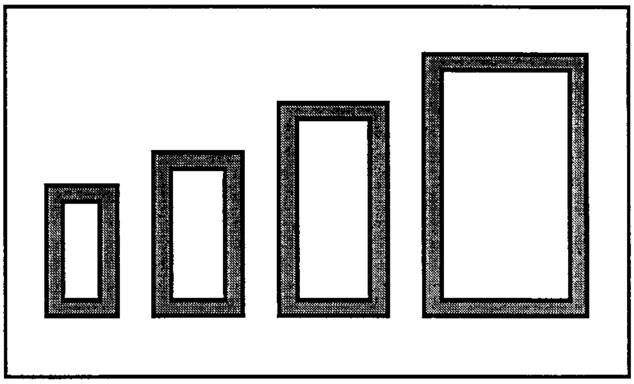
WORKSHEET 50 (Format 6)

NAME ______________________________________ DATE ________________

MORE WINDOWS

In the Modern Art Museum windows have the following features:

Their sizes vary, but, but in all of them the height is 50 cm less than three times the width.



1. Can you complete the following table?

Ancho	0.30 m	0.45 m	1.30		
Altura				4.45	6.55 m

2. The window's frames are wooden, their price per metre is $ 62.00.

a) What is the price of the frame if the window is 1.5 metres wide? _____________?

____

b) Which operations did you do to compute the cost? _________________________

____

3. Can you write a program that gives you the frame's price for any of the windows in the museum? Show your program below.

4. Use your program to complete the following table.

Width of the window	0.35 m	0.65 m	0.84 m	1.20 m	
Cost					$ 334.00

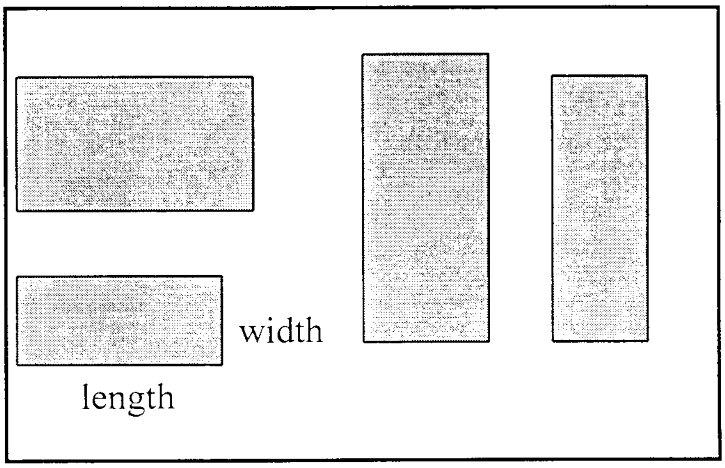
WORKSHEET 50 (Format 6)

NAME ______________________________ DATE ______________

SCALE MODELS

An exposition of Scale Models is held in the Modern Art Museum in order to show different designs for the new airport. The scale models are located on special tables with the following characteristics.

Their sizes vary but in all of them the length is one metre more than the width.



1. Can you complete the following table?

With of the table	Length of the table
1.40 metros	
2.55 metros	
3.45 metros	
	2.75 metros
	6.5 metros
	4.4 metros
	8.3 metros

2. The tables are wooden, the price per square metre is $155.00. Can you program your calculator so that it allows to obtain the cost of the any of these tables? If you could show your program in the space below.

3. Use your program to complete the following table.

Width	1.20 m	1.70 m	1.85 m			
Cost				$ 1413.00	$ 1692.00	$ 2157.6

WORKSHEET 52 (Format 6)

NAME____DATE _________

A library and record store is making the following offer:

15% OFF IN ALL OUR MERCHANDISE
The discount will be taken off the ticket price at the counters.

1. Complete the following table.

Ticket price	Amount discounted	Discount price
£ 34.00		
£ 18.75		
£ 126.80		
£ 28.50		
£ 150.00		
£ 72.35		
£ 29.40		

2. Can you program your calculator to do the following?
If you input the **ticket price**, it should output the **discount price.** Write your program in the square below.

1. Use your program to complete the following table.

Ticket price	**Discount price**
£ 84.00	
£ 28.75	
£226.80	
£29.60	
£ 140.00	
	£ 142.80
	£ 144.50

WORKSHEET 53 (Format 6)

NAME ____ DATE ________________

A stationary store is making the following offer:

25% OFF IN ALL OUR MERCHANDISE
The discount will be taken off the ticket price at the counters.

1. Complete the following table.

Ticket price	Amount discounted	Discount price
	£18.75	
	£ 6.00	
	£ 9.00	
	£21.50	
	£ 8.75	
	£6.50	
	£ 11.50	

2. Program your calculator to do the following. If you input the **ticket price**, it should output the **discount price.** Write your program: ____________________

3. Program your calculator thus: if you input the **amount discounted**, it should output the **ticket price.** Write your program: ____________________

4. Use your programs to complete the following tables.

a)

Amount discounted	$ 15.40	$ 18.75	$ 8.90	$ 10.00	$ 14.35
Discount price					

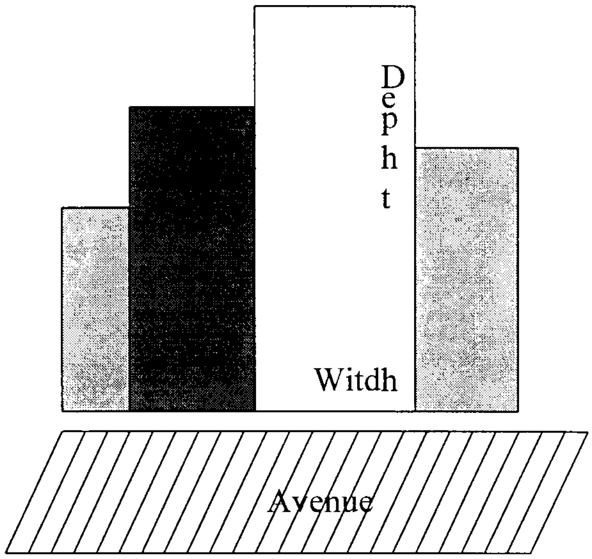
b)

Amount discounted	$ 11.70	$ 6.75	$ 8.90	$ 8.40	$ 9.60
Ticket price					

WORKSHEET 54 (Format 6)

NAME ____ DATE ___________

A real estate firm is selling lots with the following dimensions: A depth of 30 metres more than twice the front.



Answer the following using these data.

1. Mr. Pérez needed 132 metres of barbed wire to fence his land. Give the dimensions of the plot he bought. ____

2. Mrs. Gómez used 168 metres of barbed wire to fence her recently acquired plot. How long are its front and depth? ______________________________________

3. Mrs. Rodríguez built a white fence 156 metres long around her lot. What are the dimensions of said lot? ____

4. Mr. González bought a plot of land 76 metres wide. How many metres of barbed wire does he need if he intends to keep people out of it? _________________________

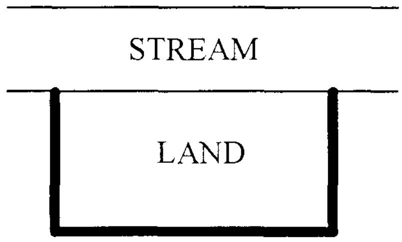
5. Explain your reasoning about the previous questions. ___________________________
____
____
____
____

6. Did you program your calculator to solve the problems? Show your program if you did.
____

WORKSHEET 55 (Format 6)

NAME ______________________________ DATE__________

A man has a piece of land by a stream. He bought 100 metres of barbed wire to fence his land where it does not border the stream.



The man wants to use the stream as a border so that his 100 metres of barbed wire yield the **biggest possible** rectangular **area**. It depends on the size of the sides.

1. Complete the right hand table.

2. Can you program the calculator to complete the table faster? If you did so, show the program below.

____

Long side	Short side	Area
50		
	30	
60		
	10	
70		
	8	
65		
58		
55.5		
54.8		
53.4		
50.2		
49.7		

3. How long should the long and short sides be to get the biggest area of land?

Long side = ____________ metres

Short side = ____________ metres

Area = ____________ square metres.

APPENDIX 2

INTERVIEW PROTOCOLS

INTERVIEW PROTOCOLS

General issues

- The researcher should encourage children to feel free of pressure. This was attempted by considering the following guidelines:
  - The interviewee must be told the purposes of the interview (for example, that these sessions were aimed at knowing better how the pupils were learning in order to improve the teaching strategies during the course).
  - The interviewer should not take seat in front to but beside the interviewee.
  - The interview should start by asking questions about situations where the interviewee was successful during the previous classroom sessions.
  - All the pupils should be asked the same (or similar) questions and the interviewer must be alert so as to introduce new suitable questions in order to follow not expected situations that may surge from specific pupil's answers.

- The interviews were task-based sessions and were carried out within the school schedule (out of the Mathematics class). The first and second interviews should last 50 minutes each (at most). In some cases, the third interview might take up to 90 minutes.
- The interviews were given individually.
- All interviews were video taped and transcribed.
- The questions posed within each interview must clearly relate to the major aims of the study. As was stated in Chapter 1 these aims are the following.

  To investigate:

  1. The notions that pupils may develop for algebraic language when they meet it through using calculator code.
  2. The extent to which the use of the calculator language helps pupils cope with simplifying similar terms within linear expressions, inverting linear functions, and transforming a linear algebraic expression to obtain a target expression.
  3. The strategies that children may develop through working with the calculator.
  4. The extent to which the use of the calculator language as a means of expressing general rules governing number patterns, helps children grasp that the algebraic code can be used as a tool for coping with problem situations.

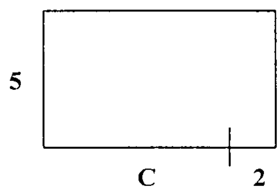
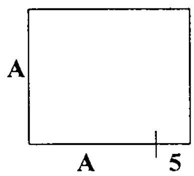
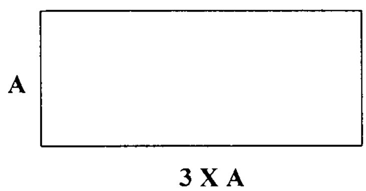
FIRST INTERVIEW

ISSUES	QUESTIONS	RELATED AIMS
1. The notions the child might have developed about letters and symbolic expressions.	1.1. You have produced a calculator program. What do the letter you used in such program mean to you? 1.2. You had written the program A×5÷2, what would happened if someone else wrote M×5÷2?	**1 and 3**
2. Child's use of parentheses and priority of operations.	2.1. Can you program the calculator so that it first takes 1 away, then it multiplies the outcome by 3? If the child produced a wrong program (for example, A−1×3), he/she would be asked to mentally calculate the result when, for example, A=2. Then he/she would be asked to run the program he/she had produced and explain why the program proceeds differently. 2.2. Explain how you come to use parentheses (this question was asked to those pupils who "spontaneously" came to see the need to use parentheses in Format 2.	**1 and 3**
3. Child's strategies for transforming linear function rules.	3.1. I wanted to type the program 3×B but I made a mistake and typed 4×B. Can you correct my program without deleting anything of what I have typed? 3.2. I wanted to type the program 10×C but I made a mistake and typed 7×C. Can you correct my program without deleting anything of what I have typed? 3.3. I wanted to type the program 10×C but I made a mistake and typed 4×C. Can you correct my program without deleting anything of what I have typed? 3.4. Look at the programs in the following list. Which of them are equivalent programs? 11×B−4xB (14−6)×B 9×B−2×B 6×B+1 6×B+B 6+1×B	**2 and 3**

SECOND INTERVIEW

ISSUES	QUESTIONS	RELATED AIMS
1. Transforming an expression to obtain another given expression.	1.1. Can you do something with 15×A so that it produces the same as 2×A? 1.2. Can you do something with 12×B+5×B−2×B so that it produces the same as 4×B? 1.3. Can you do something with 10×A+5×A so that it produces the same as 18.3×A? 1.4. Can you do something with 15×A so that it produces the same as 25×A? 1.5. Can you do something with A^3 so that it produces the same as A^2?	**2 and 3**
2. Simplifying linear expressions.	This issue was implicitly observed when the children were sorting out tasks like 1.2. and 1.3 in the section above.	**2 and 3**
3. Inverting a given program.	3.1. Can you produce a program so that it "undoes" what the program A×2−1 does? 3.2. Can you produce a program so that it "undoes" what the program A÷2+1 does? 3.3. Can you produce a program so that it "undoes" what the program A−4×5 does?	**2 and 3**

THIRD INTERVIEW

ISSUES	QUESTIONS	RELATED AIMS
1. Interpreting algebraic expressions used to denote measurements in diagrams.	1.1. Which information does this diagram give to you?  1.2. Can you calculate the perimeter/area of this rectangle? **The following questions were made when a pupil could not make sense of question 1.2.** • Can you program the calculator so that it allows you to compute the perimeter of this rectangle? • Can you program the calculator so that it allows you to compute the area of this rectangle? 1.2. Do the diagrams below provide any information?   	**1 and 3**
2. Simplifying linear expressions	2.1. Might the program $A+2+A+5+A$ be written shorter? 2.2. Might the program $A\times2+A\times3\times2$ be written shorter? 2.3. Might the program $3\times A+4\times A+A$ be written shorter 2.4. Might the program $2\times A+3+A\times4+5$ be written shorter? 2.5. Might the program $2\times A+3+A\times4+5$ be written shorter? 2.6. Might the program $3\times B+5+4\times B+2+B\times3$ be written shorter? 2.7. Might the program $7\times M+4-2\times M+6-1$ be written shorter? 2.8. A pupil from another class says that $3\times A+2\times B$ gives the same as $5\times A\times B$. What do you think about this?	**2 and 3**
3. Inverting linear functions.	3.1. Can you type a program that makes the inverse as the program: $?\rightarrow D$: $4\times D+7\times D$ **If the pupil cannot make sense of this question he/she will be asked the following question:** 3.2. Can you "make shorter" the program $?\rightarrow D$: $4\times D+7\times D$? 3.3. If the pupil answer question 1.1, he/she will be asked to invert the new program.	**2 and 3**

4. Problem situations involving generality.	4.1. Think of a number, add 10 to it and write down the result. Now take the number you thought of away from 10 and write down the result. Now add the first result to the second one ... May I try to guess the final result you got? It must be 20. Would you like to try other numbers? Can you explain why you always come to 20? 4.2. A pupil from another class says that every time he sums two consecutive numbers he gets an odd number. What do you think about this? • The interviewee will be asked to justify his/her answer. • If the pupil tried to justify by means of specific examples he/she will be asked to program the calculator so that it allows him/her to face this question. • If the pupil can make a program describing the sum of two consecutive numbers, he/she will be asked to explain what the symbols he/she used mean and to try to "make shorter" that program. 4.3. Look at the following sequence: 5, 9, 13, 17, 21, ... Will the number 877 appear if I continued writing numbers down in such list? 4.4. A pupil from another class says that $(A+B)^2 = A^2+B^2$. What do you think about this?	**3 and 4**

APPENDIX 3

CHILDREN'S WORK: A SAMPLE

HOJA DE TRABAJO NUMERO 2 (Formato 1)

NOMBRE Enrique Parker FECHA 11-Abril-94.

?
7
14
?
8
16
?
9
18
?
15
30
?
18
36

En mi calculadora escribí un programa que hace lo siguiente:

1. ¿Qué resultado me va a dar la calculadora si escribo en mi programa el número 5? 10

¿ Y si escribo el número 25? 50 ¿Si escribo el número 17? 34

¿Qué operaciones hiciste para obtener esos resultados multiplicar X 2 el número q' se desea.

2. ¿Puedes programar tu calculadora para que haga lo mismo? Escribe tu programa en el cuadro de abajo.

?→A:AX2

3. Usa el programa que hiciste para encontrar los números que faltan en la tabla.

25	37.03	59.83	117.18	136.1	200.79	275.5	326.69
50	74.06	119.66	234.36	272.2	401.58	551	653.38

¡Muy bien! Ahora apúrate con las siguientes hojas

HOJA DE TRABAJO NUMERO 2 (Formato 1)

NOMBRE Edna Paola Rodriguez Hernandez FECHA 11-04-94

?	
7	
	14
?	
8	
	16
?	
9	
	18
?	
15	
	30
?	
18	
	36

En mi calculadora escribí un programa que hace lo siguiente:

1. ¿Qué resultado me va a dar la calculadora si escribo en mi programa el número 5? 10

¿ Y si escribo el número 25? 50 ¿Si escribo el número 17? 34

¿Qué **operaciones** hiciste para obtener esos resultados multiplique por 2

2. ¿Puedes programar tu calculadora para que haga lo mismo? Escribe tu programa en el cuadro de abajo.

? →B : B×2

Corrigió

3. Usa el programa que hiciste para encontrar los números que faltan en la tabla.

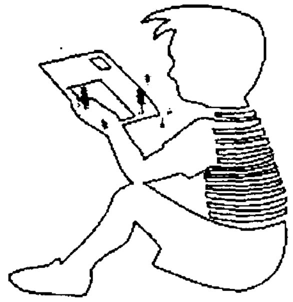
25	37.03	59.83	117.18	136.1	200.79	1102	206.37
50	74.06	119.66	234.36	272.2	401.58	551	653.38

HOJA DE TRABAJO NUMERO 4 (Formato 1)

NOMBRE José Ramón J Rodríguez FECHA 13/4/94

?	
1.1	
	3.2
?	
2.5	
	6
?	
3	
	7
?	
4.3	
	9.6
?	
5	
	11

En mi calculadora escribí un programa que hace lo siguiente:



¿Qué resultado me va a dar la calculadora si escribo en mi programa el número 50? 101

¿ Y si escribo el número 81? 163 ¿Si escribo el número 274? 549

¿Qué operaciones hiciste para obtener esos resultados? x2-1

¿Puedes programar tu calculadora para que haga lo mismo? Escribe tu programa en el cuadro de abajo.

?→D:Dx2-1

El programa no corresponde a los números que encontraste

Usa el programa que hiciste para completar la siguiente tabla.

1.3	2.8	14	50	81	274	163.6	211
3.3	6.6	29	101	163	549	325	420

3.6

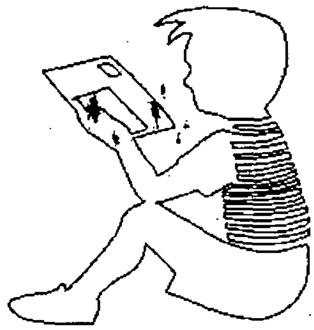
Explícame que pasó y corrige esta hoja

HOJA DE TRABAJO NUMERO 4 (Formato 1)

NOMBRE Enrique Parkés FECHA 15-Abril-94

?	
1.1	
	3.2
?	
2.5	
	6
?	
3	
	7
?	
4.3	
	9.6
?	
5	
	11

En mi calculadora escribí un programa que hace lo siguiente:



¿Qué resultado me va a dar la calculadora si escribo en mi programa el número 50? 101

¿ Y si escribo el número 81? 163 ¿Si escribo el número 274? 549

¿Qué operaciones hiciste para obtener esos resultados? multipliqué los números marcados X 2 y luego les sumé 1.

¿Puedes programar tu calculadora para que haga lo mismo? Escribe tu programa en el cuadro de abajo.

7→D:D×2+1

Muy bien!

Usa el programa que hiciste para completar la siguiente tabla.

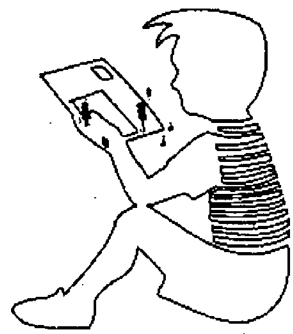
1.3	2.8	14	50	81	274	161.5	209
3.6	6.6	29	101	163	549	325	420

HOJA DE TRABAJO NUMERO 4 (Formato 1)

NOMBRE Anabell Rosas Navarrete FECHA 13/04/94.

?
1.1
3.2
?
2.5
6
?
3
7
?
4.3
9.6
?
5
11

En mi calculadora escribí un programa que hace lo siguiente:



¿Qué resultado me va a dar la calculadora si escribo en mi programa el número 50? 101

¿ Y si escribo el número 81? 163 ¿Si escribo el número 274? 549

¿Qué operaciones hiciste para obtener esos resultados? multiplicar por 2 y sumar 1

¿Puedes programar tu calculadora para que haga lo mismo? Escribe tu programa en el cuadro de abajo.

?→B:B×2+1

Corrige

Usa el programa que hiciste para completar la siguiente tabla.

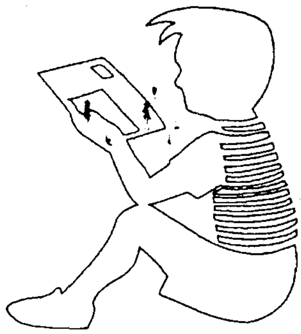
1.3	2.8	14	50	81	274	163.5	211
3.6	6.6	29	101	163	569	328	428

HOJA DE TRABAJO NUMERO 4 (Formato 1)

NOMBRE Priscila Vera FECHA 11/04/94

?
1.1
3.2
?
2.5
6
?
3
7
?
4.3
9.6
?
5
11

En mi calculadora escribí un programa que hace lo siguiente:



¿Qué resultado me va a dar la calculadora si escribo en mi programa el número 50? 101

¿ Y si escribo el número 81? 163 ¿Si escribo el número 274? 549

¿Qué operaciones hiciste para obtener esos resultados? sume el mismo numero y le aumento 1

¿Puedes programar tu calculadora para que haga lo mismo? Escribe tu programa en el cuadro de abajo.

? -> D: D+D+1

Bien!

Usa el programa que hiciste para completar la siguiente tabla.

1.3	2.8	14	50	81	274	162	204.5
3.6	6.6	29	101	163	549	325	420

HOJA DE TRABAJO NUMERO 6 (Formato 1)

NOMBRE Pedro San Juan do Ando FECHA 4/15/94

Escribe en la tabla los números que faltan

-10	-9.7	-7.8	-6.2	-5.3	-4.6	-0.7	0	1.3	12.4
-9.5	-9.2	-7.3	-5.7	-4.8	-4.1	-0.2	0.5	1.8	12.9

¿Puedes programar tu calculadora para que haga el trabajo de completar la tabla?. Si corrige

Una vez que lo hayas hecho escribe tu programa en el cuadro de la derecha

```
?→E:E+.5
[illegible]
```

Usa el programa que hiciste para obtener los números que se muestran en la tabla anterior.
¿Pudiste obtener los mismos números? Si

Completa la siguiente tabla usando el programa que hiciste.

-20	-14.7	-13.8	-12.3	-10.8	-9.6	-0.5	0
-19.5	-14.2	-13.3	-11.8	-10.3	-9.1	0	0.5

Este programa no te da estos números. Tú no corriste el programa. Debes hacerlo para asegurarte que tu respuesta es correcta.

HOJA DE TRABAJO NUMERO 7 (Formato 1)

NOMBRE Jose Luis Zaldívar F. FECHA 15/IV/94

Estoy haciendo esta tabla. ¿Puedes obtener los números que faltan?

-15	-14.5	-12.4	-10.2	-5.8	-4.6	-0.9	0	1.3	15.4
-16.5	-16	-13.9	-11.7	-7.3	-6.1	-2.4	-1.5	-.2	+13.9

Los números que aparecen en la tabla anterior se obtuvieron siguiendo una regla. ¿Cuál es esa regla? -1.5

¿Puedes programar tu calculadora usando esa regla?.

Una vez que lo hayas hecho escribe tu programa en el cuadro de la derecha

? → C; C - 1.5

Usa el programa que hiciste para obtener los números que aparecen en la tabla. ¿Tu programa produce los mismos números? no

Completa la siguiente tabla usando el programa que hiciste.

-20	-15.8	-13.8	-10.4	-10.83	-8.22	-0.5	11.5
-21.5	-17.3	-15.3	-11.9	-12.33	-9.72	-2	10

HOJA DE TRABAJO NUMERO 7 (Formato 1)

NOMBRE Priscila Vera FECHA 15/Abril/94.

Estoy haciendo esta tabla. ¿Puedes obtener los números que faltan?

-15	-14.5	-12.4	-10.2	-5.8	-4.6	-0.9	0	1.3	15.4
-16.5	-16	-13.9	-11.7	-7.3	-6.1	-2.4	1.5	-0.2	13.9

Los números que aparecen en la tabla anterior se obtuvieron siguiendo una regla. ¿Cuál es esa regla? sumarle 1.5

¿Puedes programar tu calculadora usando esa regla?.

Una vez que lo hayas hecho escribe tu programa en el cuadro de la derecha

? →D: D-1.5

Usa el programa que hiciste para obtener los números que aparecen en la tabla. ¿Tu programa produce los mismos números? Si

Muy bien!

Completa la siguiente tabla usando el programa que hiciste.

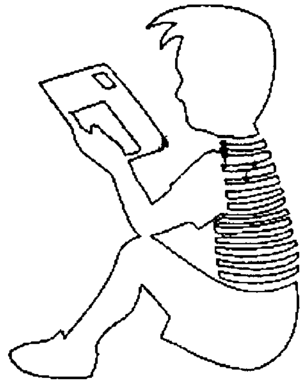
-20	-15.8	-13.8	-10.4	-10.83	-8.22	-0.5	11.5
-21.5	-17.3	15.3	-11.9	-12.33	-9.72	-2	10

HOJA DE TRABAJO NUMERO 10 (Formato 1) ✓ D

NOMBRE Georgina [illegible] FECHA 15-04-94

Estoy completando esta tabla. ¿Puedes encontrar los números que me faltan?

4	4.04
6	6.06
9	9.09
10	10.1
12	12.12
15.5	15.655
17.8	17.978
19.2	19.392
20.4	20.604
50.2	50.702



Los números que aparecen en la tabla anterior se obtuvieron siguiendo una regla. ¿Cuál es esa regla? multiplicar x 1.01

¿Puedes programar tu calculadora usando esa regla? Escribe tu programa en el cuadro de la derecha

? → K : K x 1.01

Muy bien!

¿Tu programa produce los mismos números que aparecen en la tabla? si

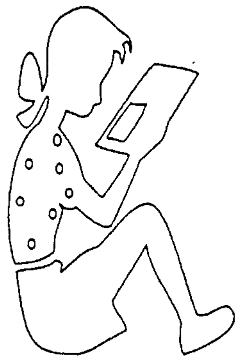
Usa el programa que hiciste para completar la siguiente tabla.

1	2.2	3.1	4.3	9	12	32	38.4
1.01	2.222	3.131	4.343	9.09	12.12	32.32	38.784

HOJA DE TRABAJO NUMERO 12 (Formato 1)

NOMBRE Juan Alejandro Hdz. Herrera FECHA 18/4/97

En mi calculadora escribí un programa que hace lo siguiente:



?	
7	
	20
?	
7.5	
	21.5
?	
8.2	
	23.6
?	
9	
	26
?	
9.6	
	27.8

Si escribo el número 10 ¿qué número va a dar como resultado la calculadora? 29

Si escribo el número 12, ¿qué número va a dar como resultado la calculadora? 35

La calculadora me dio como resultado 17.5. ¿Cuál número le di como entrada? 51.5 6.17

Explica cómo obtuviste esos resultados X3 − 1

¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe tu programa en el cuadro de la derecha.

? → A. A x3 − 1

Usa el programa que hiciste para completar la siguiente tabla.

3	6	5.1	5.4	9.4	11.2	22	27
8	17	14.3	15.2	27.2	32.6	65	80

Muy bien!

HOJA DE TRABAJO NUMERO 14 (Formato 1)

NOMBRE Martha Balcazar Santa Ana FECHA 18/04/94

?	
2	
	5
?	
3	
	7.5
?	
4	
	10
?	
5	
	12.5

En mi calculadora escribí un programa que hace lo siguiente:

Si escribo el número 6 ¿qué número va a dar como resultado la calculadora? 15

Si escribo el número 7, ¿qué número va a dar como resultado la calculadora? 17.5

Si escribo el número 55, ¿qué número va a dar como resultado la calculadora? 137.5

¿Qué operaciones hiciste para obtener esos resultados? Multiplica 2.5

¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe tu programa en el cuadro de la derecha.

? → A: A×2.5

Usa el programa que hiciste para completar la siguiente tabla.

3	3.4	5.1	6.2	9.4	9.4	12.2	14
7.5	8.5	12.75	15.5	23.5	23.5	30.5	35

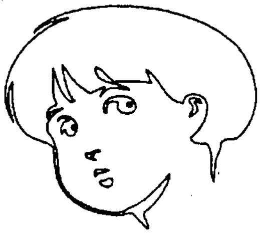
Muy bien

HOJA DE TRABAJO NUMERO 15 (Formato 1)

NOMBRE Reyes y Morales José FECHA 18/Abril/94

En mi calculadora escribí un programa que hace lo siguiente:

?	
0.15	
	0.015
?	
0.27	
	0.027
?	
0.3	
	0.03
?	
1.5	
	0.15
?	
2.03	
	0.203



¡Muy bien!

Si escribo el número 10 ¿qué número va a dar como resultado la calculadora? 0.1

La calculadora me dio como resultado 37, ¿qué número le dí como entrada? 3.7

¿Qué operaciones hiciste para obtener esos resultados Multiplicando por punto cien

¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe tu programa en el cuadro de la derecha.

? → A: A × 0.100

Usa el programa que hiciste para completar la siguiente tabla.

3	4	5.1	6.3	9.4	11.8	12.2	350
0.3	0.4	0.51	0.63	.94	1.18	1.22	35

HOJA DE TRABAJO NUMERO 16 (Formato 2)

ADIVINA MI PROGRAMA

Fecha 20 / Abril / 94

Nombre del que inventó el programa: Rosas y Morales José

Nombre del que adivinó el programa: Jenny Scott

IMPORTANTE: EL QUE INVENTA EL PROGRAMA NO DEBE ESCRIBIRLO, SOLO DEBE DAR ALGUNAS PISTAS PARA QUE SU COMPAÑERO (A) LO ADIVINE. CUANDO SU COMPAÑERO (A) PROPONGA UN PROGRAMA, EL QUE LO INVENTO DEBE CHECAR SI ADIVINO CORRECTAMENTE.

Pistas para adivinar el programa:

Número de entrada	Número de salida
1	1.5
3	4.5
5	7.5
8	12.
10	15.
20	30.

El que está adivinando el programa debe escribirlo en su calculadora y si su programa funciona bien debe anotarlo en el siguiente cuadro.

?→ A:A × 3 ÷ 2

¿Cuántos intentos hiciste para adivinar el programa? 1

¿Necesitaste que te dieran más pistas? ¿Cuáles? no

HOJA DE TRABAJO NUMERO 17 (Formato 2)

ADIVINA MI PROGRAMA

Fecha 20/04/94

Nombre del que inventó el programa: Renata Elizabeth Arias Lopez

Nombre del que adivinó el programa: Jose Luis Zaldivar Fujigaki

IMPORTANTE: EL QUE INVENTA EL PROGRAMA NO DEBE ESCRIBIRLO, SOLO DEBE DAR ALGUNAS PISTAS PARA QUE SU COMPAÑERO (A) LO ADIVINE. CUANDO SU COMPAÑERO (A) PROPONGA UN PROGRAMA, EL QUE LO INVENTO DEBE CHECAR SI ADIVINO CORRECTAMENTE.

Pistas para adivinar el programa:

Número de entrada	Número de salida
1	9.1
3	27.3
5	45.5
8	72.8
10	91
20	182

El que está adivinando el programa debe escribirlo en su calculadora y si su programa funciona bien debe anotarlo en el siguiente cuadro.

? → B : B × 9.1

¿Cuántos intentos hiciste para adivinar el programa? 1

¿Necesitaste que te dieran más pistas? ¿Cuáles? no, ninguna

HOJA DE TRABAJO NUMERO 17 (Formato 2)

ADIVINA MI PROGRAMA

Fecha 20/ABRIL/94

Nombre del que inventó el programa: GUILLERMO OLVERA

Nombre del que adivinó el programa: Enrique Parker

IMPORTANTE: EL QUE INVENTA EL PROGRAMA NO DEBE ESCRIBIRLO, SOLO DEBE DAR ALGUNAS PISTAS PARA QUE SU COMPAÑERO (A) LO ADIVINE. CUANDO SU COMPAÑERO (A) PROPONGA UN PROGRAMA, EL QUE LO INVENTO DEBE CHECAR SI ADIVINO CORRECTAMENTE.

Pistas para adivinar el programa:

Número de entrada	Número de salida
1	21
3	27
5	33
8	42
10	48
20	78

El que está adivinando el programa debe escribirlo en su calculadora y si su programa funciona bien debe anotarlo en el siguiente cuadro.

? → B : B×3+18

¿Cuántos intentos hiciste para adivinar el programa? 9

¿Necesitaste que te dieran más pistas? ¿Cuáles? si, de 2 operaciones

HOJA DE TRABAJO NUMERO 17 (Formato 2)

ADIVINA MI PROGRAMA

Fecha 20/4/94

Nombre del que inventó el programa: Felipe Tomás Carreon D.

Nombre del que adivinó el programa: Jose Ramon J. Rodriguez F.

IMPORTANTE: EL QUE INVENTA EL PROGRAMA NO DEBE ESCRIBIRLO, SOLO DEBE DAR ALGUNAS PISTAS PARA QUE SU COMPAÑERO (A) LO ADIVINE. CUANDO SU COMPAÑERO (A) PROPONGA UN PROGRAMA, EL QUE LO INVENTO DEBE CHECAR SI ADIVINO CORRECTAMENTE.

Pistas para adivinar el programa:

Número de entrada	Número de salida
1	1
3	1.4
5	1.8
8	2.4
10	2.8
20	4.8

El que está adivinando el programa debe escribirlo en su calculadora y si su programa funciona bien debe anotarlo en el siguiente cuadro.

2 → A : Δ ÷ 5 + .8

¿Cuántos intentos hiciste para adivinar el programa? 4

¿Necesitaste que te dieran más pistas? ¿Cuáles? Si ÷ y decimales

HOJA DE TRABAJO NUMERO 21 (Formato 3)

NOMBRE Alma Nathalie CQ FECHA 25/IV/94

En mi calculadora escribí un programa que hace lo siguiente:

?
1
4
?
1.5
6
?
3
12
?
5
20

Si escribo el número 7, ¿qué número va a dar como resultado la calculadora? 28

¿ Y si escribo el número 10? 40 ¿Si escribo el número 70? 280

¿Qué operaciones hiciste para obtener esos resultados? multiplicación

2. ¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe el programa que hiciste en el cuadro de abajo.

?→V: V×4

3. ¿Puedes escribir otro programa que haga lo mismo? Una vez que lo hayas hecho escríbelo en el cuadro de abajo.

?→N: N×3×1

4. ¿Puedes encontrar otros programas ~~distintos que~~ hagan lo mismo? Escríbelos abajo.

?→X: X×2×2	?2: 2: 1×4	

¿80% esta hoja en una clase?

HOJA DE TRABAJO NUMERO 21 (Formato 3)

NOMBRE Georgina Haina FECHA 25-05-94

En mi calculadora escribí un programa que hace lo siguiente:

?	
1	
	4
?	
1.5	
	6
?	
3	
	12
?	
5	
	20

Si escribo el número 7, ¿qué número va a dar como resultado la calculadora? 28

¿ Y si escribo el número 10? 40 ¿Si escribo el número 70? 280

¿Qué operaciones hiciste para obtener esos resultados? multiplicar x 4

2. ¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe el programa que hiciste en el cuadro de abajo.

? → K : K x 4

3. ¿Puedes escribir otro programa que haga lo mismo? Una vez que lo hayas hecho escríbelo en el cuadro de abajo.

? → K : K x 8 ÷ 2

y 16 ÷ 4

4. ¿Puedes encontrar otros programas distintos que hagan lo mismo? Escríbelos abajo.

? → L : L x 16 ÷ 4		

Trata de encontrar otros programas. Haz un esfuerzo.

HOJA DE TRABAJO NUMERO 21 (Formato 3)

NOMBRE José Luis Zaldívar FECHA 25/IV/93

En mi calculadora escribí un programa que hace lo siguiente:

?
1
4
?
1.5
6
?
3
12
?
5
20

Si escribo el número 7, ¿qué número va a dar como resultado la calculadora? 28

¿ Y si escribo el número 10? 40 **¿Si escribo el número 70?** 280

¿Qué operaciones hiciste para obtener esos resultados? X 4

2. ¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe el programa que hiciste en el cuadro de abajo.

? → C₀ C X 4

3. ¿Puedes escribir otro programa que haga lo mismo? Una vez que lo hayas hecho escríbelo en el cuadro de abajo.

? → C₀ C X 2 X 2

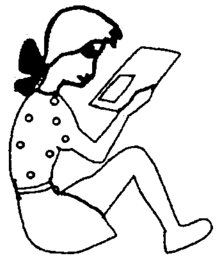
4. ¿Puedes encontrar otros programas distintos que hagan lo mismo? Escríbelos abajo.

? → C₀ C X 8 ÷ 2 | ? → C₀ C X 16 ÷ 4 | ? → C₀ C X 24 ÷ 6

HOJA DE TRABAJO NUMERO 21 (Formato 3)

NOMBRE Uirsha Balcazar FECHA 25/04/94

En mi calculadora escribí un programa que hace lo siguiente:



?	
1	
	4
?	
1.5	
	6
?	
3	
	12
?	
5	
	20

Si escribo el número 7, ¿qué número va a dar como resultado la calculadora? 28

¿ Y si escribo el número 10? 40 ¿Si escribo el número 70? 280

¿Qué operaciones hiciste para obtener esos resultados? multipliqué por 4

2. ¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe el programa que hiciste en el cuadro de abajo.

? → G: G x 4

3. ¿Puedes escribir otro programa que haga lo mismo? Una vez que lo hayas hecho escríbelo en el cuadro de abajo.

? → R: R x 5 − R

4. ¿Puedes encontrar otros programas distintos que hagan lo mismo? Escríbelos abajo.

? → h: h x 3 + h	? → U: U x 2 + U x 2	? → T: T x L (x2)

Este programa no lo entiendo. EXPLICAMELO

Completar

HOJA DE TRABAJO NUMERO 22 (Formato 3)

NOMBRE Felipe Tomás Carreón P. FECHA 20/4/94

?	
2	
	3
?	
4	
	6
?	
8	
	12
?	
10	
	15

En mi calculadora escribí un programa que hace lo siguiente:

Si escribo el número 5, ¿qué número va a dar como resultado la calculadora? 7.5

¿ Y si escribo el número 6? 9 **¿Si escribo el número 15?** 22.5

¿Qué operaciones hiciste para obtener esos resultados? x3÷2

2. ¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho pruébalo en tu calculadora y si funciona escríbelo en el cuadro de abajo.

? → B : B x 3 ÷ 2

3. ¿Puedes escribir otro programa que haga lo mismo? Pruébalo en tu calculadora y si funciona escríbelo en el cuadro de abajo.

? → B : B x 6 ÷ 4

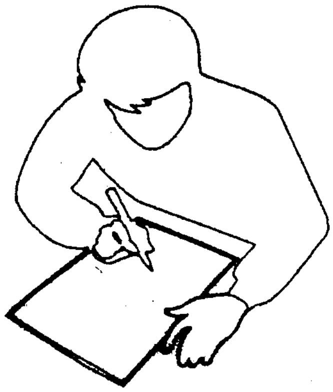
4. ¿Puedes encontrar otros programas distintos que hagan lo mismo? Pruébalos en tu calculadora y si funcionan escríbelos abajo.

? → B : B x 6 ÷ 4		? → B : B x 36 ÷ 24

HOJA DE TRABAJO NUMERO 26 (Formato 3)

NOMBRE [illegible] José FECHA 17/Abr./94

En mi calculadora escribí un programa que hace lo siguiente:



?	
1	
	6
?	
3	
	10
?	
5	
	14
?	
9	
	22

Si escribo el número 10, ¿qué número va a dar como resultado la calculadora? 74

¿Y si escribo el número 20? 44 ¿Si escribo el número 50? 104

2. ¿Puedes programar tu calculadora para que haga lo mismo? Si tu programa funciona escríbelo en el cuadro de abajo.

?->A:A×2+4

3. ¿Puedes escribir otro programa que haga lo mismo? Pruébalo en tu calculadora, si funciona escríbelo en el cuadro de abajo.

?->A:A+A+4

4. ¿Puedes encontrar otros programas distintos que hagan lo mismo? Escríbelos abajo.

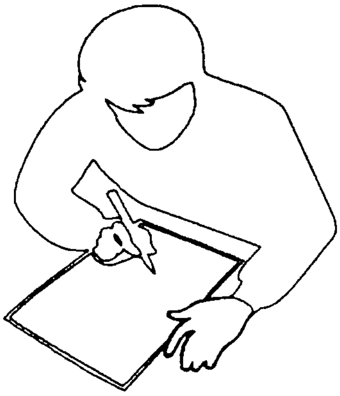
?->A:A+A(2×4÷2) | ?->A:A+A(2×4-2) | ?->A:A+4+A

Muy bien!

HOJA DE TRABAJO NUMERO 32 (Formato 4)

NOMBRE Enrique Parker FECHA 16-Mayo-94

En mi calculadora escribí un programa que hace lo siguiente:



```
?
1.3
                18.7
?
2.5
                17.5
?
3.8
                16.2
?
4.4
                15.6
?
5.9
                14.1
```

Si escribo el número 6 ¿qué número va a dar como resultado la calculadora? 14

¿Y si escribo el número 7? 13 ¿Si escribo el número 9? 11

¿Qué operaciones hiciste para obtener esos resultados? sumé 20 sobre el número indicado y luego le resté D.

2. ¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe tu programa en el cuadro de abajo.

?→D : 10+10-D

3. Usa tu programa para completar la siguiente tabla.

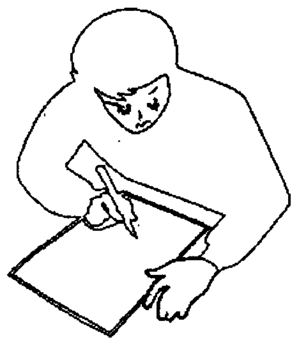
2.83	3.03	- 3.5	- 4.8	15.01	26.2	47.04	57.32
17.17	16.97	23.5	24.8	5.01	6.2	27.04	37.32

4. ¿Qué ocurre cuando introduces como entrada a tu programa un número negativo? lo vuelve positivo

HOJA DE TRABAJO NUMERO 32 (Formato 4)

NOMBRE Anabell Rosas Narciso FECHA 9/05/99

En mi calculadora escribí un programa que hace lo siguiente:



?	
1.3	
	18.7
?	
2.5	
	17.5
?	
3.8	
	16.2
?	
4.4	
	15.6
?	
5.9	
	14.1

Si escribo el número 6 ¿qué número va a dar como resultado la calculadora? 14

¿Y si escribo el número 7? 13 ¿Si escribo el número 9? 11

¿Qué operaciones hiciste para obtener esos resultados? a 20 le reste el numero que necesita.

2. ¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe tu programa en el cuadro de abajo.

? -> F : 20 - F

3. Usa tu programa para completar la siguiente tabla.

2.83	3.03	-3.5	-4.8	14.99	13.8	-7.04	-17.32
17.17	16.97	23.5	24.8	5.01	6.2	27.04	37.32

4. ¿Qué ocurre cuando introduces como entrada a tu programa un número negativo? en vez de restar lo sumo

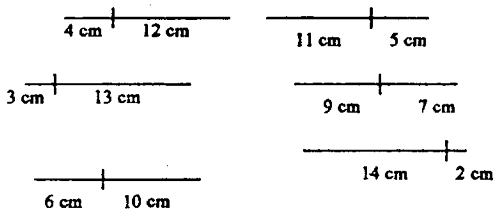
MUY BIEN

SIGUELE ASÍ, VAS BIEN

HOJA DE TRABAJO NUMERO 33 (Formato 4)

NOMBRE ______________________________ FECHA 6-Junio-94

Tengo varios trozos de alambre, todos miden 16 cm. de largo. Los quiero cortar en dos partes de dsitintas maneras. En la siguiente figura se muestran algunas posibilidades:



1. ¿Puedes programar tu calculadora de manera que si le das la medida de una de las partes te dé como resultado la medida de la otra parte?

   Escribe el programa que hiciste en el cuadro de abajo.

? -▷ B : 16 - B

2. Describe cómo razonaste para construir tu programa. Busqué un número q' diera por resultado el número puesto ahí.

3. Usa el programa que hiciste para completar la siguiente tabla.

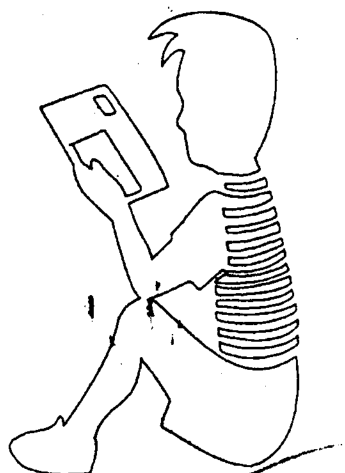
1.7	28.8	3.8	30.9	6.8	31.6	7.9	33.4
14.3	12.8	12.2	14.9	9.2	15.6	8.1	17.4

Muy bien!

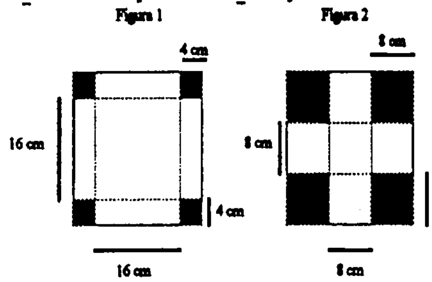
HOJA DE TRABAJO NUMERO 34 (Formato 4)

NOMBRE ______________________________ FECHA __________

Tengo una pieza cuadrada de cartón y quiero usarla para hacer una caja. Si recorto cuadrados en cada esquina de la pieza de cartón y luego doblo hacia arriba formaré la caja que quiero.



El tamaño de los lados de los cuadrados que recorte determinan cuánto va a medir la base de la caja y también cuánto va a medir su altura. Las figuras 1 y 2 muestran dos posibles formas para armar la caja.



1. ¿Cuánto mide por lado la pieza cuadrada de cartón que tengo ? 24 cm ¿Cuál es su área? 576 Escribe las operaciones que hiciste. 24 x 24

2. Completa la siguiente tabla:

	De acuerdo con la Figura 1	De acuerdo con la figura 2
Area de la base	256	64
Altura de la caja	4	8
Volumen de la caja	1024 cm²	810 cm²

3. Quisiera formar una caja de manera que tenga el mayor volumen posible. Como sólo tengo esta pieza de cartón solamente puedo hacer un intento. ¿Puedes programar tu calculadora para obtener el volumen de cualquier caja que forme cortando cuadrados en las esquinas? Escribe tu programa en el cuadro de abajo.

9→C : (24 − B x2)² x B

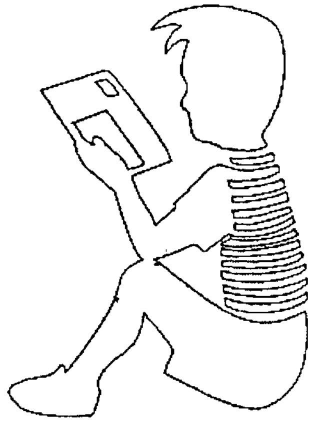
4. Usa tu programa para obtener cuánto deben medir el lado de la base y altura de la caja para obtener un volumen máximo. Escribe las medidas que encontraste en el cuadro de abajo.

	Volumen máximo	Aprox. por defecto	Aprox. por exceso
Lado de la base	13.5	23	24
Altura de la caja	,25	~5	6

HOJA DE TRABAJO NUMERO 34 (Formato 4)

NOMBRE José Luis FECHA 8/VI/99

Tengo una pieza cuadrada de cartón y quiero usarla para hacer una caja. Si recorto cuadrados en cada esquina de la pieza de cartón y luego doblo hacia arriba formaré la caja que quiero.



El tamaño de los lados de los cuadrados que recorte determinan cuánto va a medir la **base de la caja** y también cuánto va a medir su **altura**. Las figuras 1 y 2 muestran dos posibles formas para armar la caja.

Figura 1

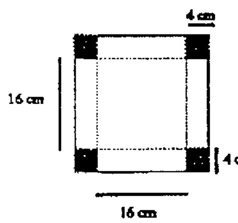
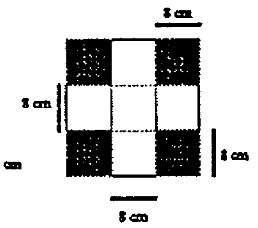


Figura 2



1. ¿Cuánto mide por lado la pieza cuadrada de cartón que tengo ? 24 ¿Cuál es su área? 576 Escribe las operaciones que hiciste. 24 X 24

2. Completa la siguiente tabla:

	De acuerdo con la Figura 1	De acuerdo con la figura 2
Area de la base	256	64
Altura de la caja	~~16~~ 4	~~64~~ 8
Volumen de la caja	~~4096~~ 1024	512

3. Quisiera formar una caja de manera que tenga el **mayor volumen posible**. Como sólo tengo esta pieza de cartón solamente puedo hacer un intento. ¿Puedes programar tu calculadora para obtener el volumen de cualquier caja que forme cortando cuadrados en las esquinas? Escribe tu programa en el cuadro de abajo.

? → A: A² X (24 − A) ÷ 2

4. Usa tu programa para obtener cuánto deben medir el lado de la base y altura de la caja para obtener un volumen máximo. Escribe las medidas que encontraste en el cuadro de abajo.

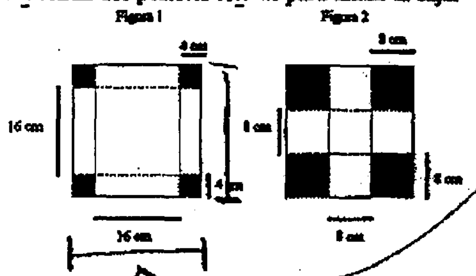
	Volumen máximo	Aprox. por defecto	Aprox. por exceso
Lado de la base	16	15.9	16.1
Altura de la caja	4	4.05	3.95

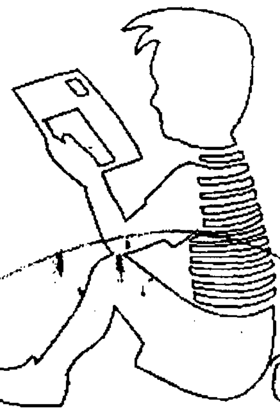
HOJA DE TRABAJO NUMERO 34 (Formato 4)

NOMBRE PRISCILA VERA FECHA ______

Tengo una pieza cuadrada de cartón y quiero usarla para hacer una caja. Si recorto cuadrados en cada esquina de la pieza de cartón y luego doblo hacia arriba formaré la caja que quiero.

El tamaño de los lados de los cuadrados que recorte determinan cuánto va a medir la base de la caja y también cuánto va a medir su altura. Las figuras 1 y 2 muestran dos posibles formas para armar la caja.





1. ¿Cuánto mide por lado la pieza cuadrada de cartón que tengo? 24 ¿Cuál es su área? 576 Escribe las operaciones que hiciste. 24 × 24

OJO ESA NO ES LA MEDIDA

2. Completa la siguiente tabla:

	De acuerdo con la Figura 1	De acuerdo con la figura 2
Area de la base	256	64
Altura de la caja	4	8
Volumen de la caja	1024 cm³	512 cm³

3. Quisiera formar una caja de manera que tenga el mayor volumen posible. Como sólo tengo esta pieza de cartón solamente puedo hacer un intento. ¿Puedes programar tu calculadora para obtener el volumen de cualquier caja que forme cortando cuadrados en las esquinas? Escribe tu programa en el cuadro de abajo.

$2 \to D: (24 - D + (-D))^2 \times D$

4. Usa tu programa para obtener cuánto deben medir el lado de la base y altura de la caja para obtener un volumen máximo. Escribe las medidas que encontraste en el cuadro de abajo.

	Volumen máximo	Aprox. por defecto	Aprox. por exceso
Lado de la base			
Altura de la caja			

Vas muy bien. Continúa así y termina esta hoja. Esta es la más difícil. Lo demás te va a parecer muy sencillo.

HOJA DE TRABAJO NUMERO 35

NOMBRE Mirska Baldizon FECHA ____________

En el área de programación número 5 de mi calculadora escribí un programa que hace lo siguiente:

Próg 5	
?	
1	
	0
2	
	-1
3	
	-2
4	
	-3
5	
	-2

Si escribo el número 6 ¿qué número va a dar como resultado la calculadora? -5

Si escribo el número 7, ¿qué número va a dar como resultado la calculadora? -6

Si escribo el número 9, ¿qué número va a dar como resultado la calculadora? -8

Si escribo el número 17, ¿qué número va a dar como resultado la calculadora? -16

¿Por qué pudiste saber esos resultados? MULTIPLICO EL NUMERO -1

2. ¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe el programa que hiciste en el cuadro de abajo.

2 → K: K × -1 + 1

HOJA DE TRABAJO NUMERO 35

NOMBRE ______________________ FECHA __________

En el área de programación número 5 de mi calculadora escribí un programa que hace lo siguiente:

Prog 5	
?	
1	
	0
2	
	-1
3	
	-2
4	
	-3
5	
	-2

Si escribo el número 6 ¿qué número va a dar como resultado la calculadora? -5

Si escribo el número 7, ¿qué número va a dar como resultado la calculadora? -6

Si escribo el número 9, ¿qué número va a dar como resultado la calculadora? -8

Si escribo el número 17, ¿qué número va a dar como resultado la calculadora? 716

¿Por qué pudiste saber esos resultados? por que lo multiplique x -1 y le sume 1

2. ¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe el programa que hiciste en el cuadro de abajo.

?→D:D×-1+1

HOJA DE TRABAJO NUMERO 39

NOMBRE [illegible] FECHA 9/Enero/94

En el área de programación número 9 de mi calculadora escribí un programa que hace lo siguiente:	Prog 9 ? 1 0 2 0 3 0 4 0 5 0

Si escribo el número 6 ¿qué número va a dar como resultado la calculadora? 0

Si escribo el número 7, ¿qué número va a dar como resultado la calculadora? 0

Si escribo el número 9, ¿qué número va a dar como resultado la calculadora? 0

Si escribo el número 17, ¿qué número va a dar como resultado la calculadora? 0

¿Por qué pudiste saber esos resultados? Lo multiplico por 0

2. ¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe el programa que hiciste en el cuadro de abajo.

?→A=A×0

¡Bien!

HOJA DE TRABAJO NUMERO 40

NOMBRE Guillermo [illegible] FECHA [illegible]

En el área de programación número 0 de mi calculadora escribí un programa que hace lo siguiente:

Prog 0	
?	
1	-1
2	-2
3	-3
4	-4
5	-5

Si escribo el número 6 ¿qué número va a dar como resultado la calculadora? -6

Si escribo el número 7, ¿qué número va a dar como resultado la calculadora? -7

Si escribo el número 9, ¿qué número va a dar como resultado la calculadora? -9

Si escribo el número 17, ¿qué número va a dar como resultado la calculadora? -17

¿Por qué pudiste saber esos resultados? LO RESTE Y LO MULTIPLIQUE

2. ¿Puedes programar tu calculadora para que haga lo mismo? Una vez que lo hayas hecho escribe el programa que hiciste en el cuadro de abajo.

?→A A-(A×2)

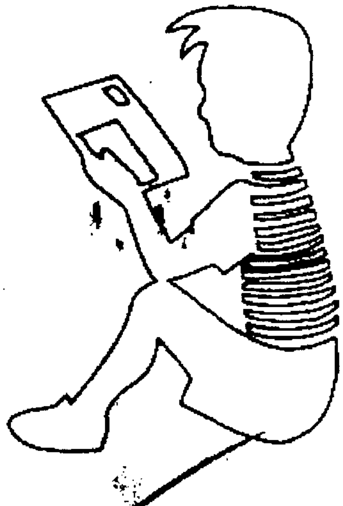
¡Bien!

HOJA DE TRABAJO NUMERO 42 (Formato 5)

NOMBRE Oscar Rojo Aranda ____________ FECHA 2 Marzo 94

?
11.4
17.5
?
11.4
17.5
?
19
25.1
?
23.1
29.2
?
38
44.1
?
50
56.1

En mi calculadora escribí un programa que hace lo siguiente:



1. Adivina cuál es el programa que hice. Escríbelo en tu calculadora y si funciona igual que el mío escríbelo en el cuadro de abajo.

2→H:H+H+H+6.1

2. Ahora inventa un programa que haga lo inverso de lo que hace mi programa ? Es decir que te dé los números que se muestran en la siguiente tabla:

17.5	25.1	29.2	44.1	31.5
11.4	19	23.1	38	27

Si pudiste hacer el programa escríbelo en el cuadro de la derecha.

2→H:H/H+H−6.1

3. Describe qué hiciste para construir el programa que "deshace" el mío ____________

____

____

____

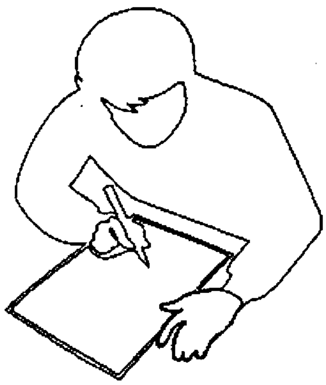
____

Completa

HOJA DE TRABAJO NUMERO 44 (Formato 5)

NOMBRE Enrique Parker FECHA ________

En mi calculadora escribí un programa que hace lo siguiente:



?
3
5
?
7
13
?
10
19
?
11
21
?
15
29

1. ¿Puedes adivinar qué programa hice? Si lo encuentras pruébalo en tu calculadora y escríbelo en el cuadro de abajo.

$? \rightarrow O : O \times 2 - 1$

2. Ahora inventa un programa que "deshaga" lo que hace mi programa ? Es decir que te dé los números que se muestran en la siguiente tabla:

5	13	19	21	29
3	7	10	11	15

Si pudiste inventar un programa como el que se pide escríbelo en el cuadro de la derecha.

$? \rightarrow O : (O + 1) \div 2$

~~? → O (O + 2 +~~

3. ¿Puedes hacer un programa que "deshaga" al siguiente programa?

$? \rightarrow B : B \times 3 + 1$

Prueba tu programa en tu calculadora y asegúrate que funcione como esperas, después escríbelo en el cuadro de abajo.

$? \rightarrow B : (B - 1) \div 3$

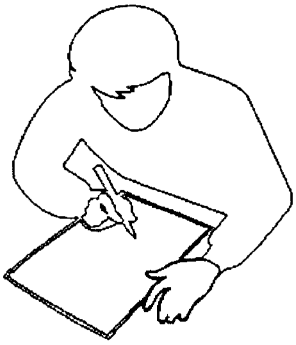
Bien!

Corregir

HOJA DE TRABAJO NUMERO 44 (Formato 5)

NOMBRE Jose Ramón J. Rodríguez FECHA 2/5/94

En mi calculadora escribí un programa que hace lo siguiente:



?	
3	
	5
?	
7	
	13
?	
10	
	19
?	
11	
	21
?	
15	
	29

1. ¿Puedes adivinar qué programa hice? Si lo encuentras pruébalo en tu calculadora y escríbelo en el cuadro de abajo.

?→D:D×2-1

2. Ahora inventa un programa que "deshaga" lo que hace mi programa ? Es decir que te dé los números que se muestran en la siguiente tabla:

5	13	19	21	29
3	7	10	11	15

Si pudiste inventar un programa como el que se pide escríbelo en el cuadro de la derecha.

?→D:D÷2+.5 ✓

3. ¿Puedes hacer un programa que "deshaga" al siguiente programa?

? → B : B × 3 + 1

Prueba tu programa en tu calculadora y **asegúrate** que funcione como esperas, después escríbelo en el cuadro de abajo.

?→B:B÷3-1 X

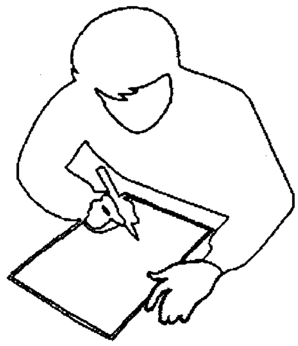
Corrige
Debes correr el programa para estar seguro de tu respuesta.

Corregir

HOJA DE TRABAJO NUMERO 44 (Formato 5)

NOMBRE Pedro San Juan de Anda FECHA 2/4/94

En mi calculadora escribí un programa que hace lo siguiente:



?	
3	
	5
?	
7	
	13
?	
10	
	19
?	
11	
	21
?	
15	
	29

1. ¿Puedes adivinar qué programa hice? Si lo encuentras pruébalo en tu calculadora y escríbelo en el cuadro de abajo.

?→C:C×2-1 ✓

2. Ahora inventa un programa que "deshaga" lo que hace mi programa ? Es decir que te dé los números que se muestran en la siguiente tabla:

5	13	19	21	29
3	7	10	11	15

Si pudiste inventar un programa como el que se pide escríbelo en el cuadro de la derecha.

?→C:C÷2+.5 ✓

3. ¿Puedes hacer un programa que "deshaga" al siguiente programa?

$? \rightarrow B : B \times 3 + 1$

Prueba tu programa en tu calculadora y asegúrate que funcione como esperas, después escríbelo en el cuadro de abajo.

~~[illegible]~~ ?→B:B÷2+.5 ✓

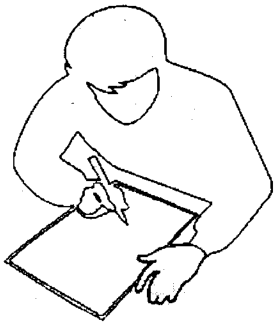
Corre este programa y ve qué pasa.

Corregir

HOJA DE TRABAJO NUMERO 44 (Formato 5)

NOMBRE ________________________________ FECHA __________

En mi calculadora escribí un programa que hace lo siguiente:



?	
3	
	5
?	
7	
	13
?	
10	
	19
?	
11	
	21
?	
15	
	29

1. ¿Puedes adivinar qué programa hice? Si lo encuentras pruébalo en tu calculadora y escríbelo en el cuadro de abajo.

? → A : A × 2 − 1

× 2 − 1

2. Ahora inventa un programa que "deshaga" lo que hace mi programa ? Es decir que te dé los números que se muestran en la siguiente tabla:

5	13	19	21	29
3	7	10	11	15

Si pudiste inventar un programa como el que se pide escríbelo en el cuadro de la derecha.

? → A : A ÷ 2 + .5

3. ¿Puedes hacer un programa que "deshaga" al siguiente programa?

? → B : B × 3 + 1

Prueba tu programa en tu calculadora y asegúrate que funcione como esperas, después escríbelo en el cuadro de abajo.

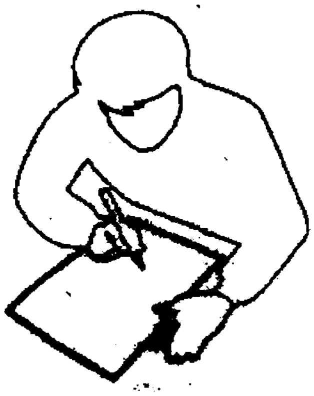
? → B : B ÷ 3 + .5

Corre este programa y vas a ver que no invierte al programa B × 3 + 1

HOJA DE TRABAJO NUMERO 44 (Formato 5)

NOMBRE PRISCILA UEDA FECHA ________

En mi calculadora escribí un programa que hace lo siguiente:



?	
3	
	5
?	
7	
	13
?	
10	
	19
?	
11	
	21
?	
15	
	29

1. ¿Puedes adivinar qué programa hice? Si lo encuentras pruébalo en tu calculadora y escríbelo en el cuadro de abajo.

?→D:D×2-1

2. Ahora inventa un programa que "deshaga" lo que hace mi programa ? Es decir que te dé los números que se muestran en la siguiente tabla:

5	13	19	21	29
3	7	10	11	15

Si pudiste inventar un programa como el que se pide escríbelo en el cuadro de la derecha.

?→D:D÷2+.5

Bien !

3. ¿Puedes hacer un programa que "deshaga" al siguiente programa?

? → B : B × 3 + 1

Prueba tu programa en tu calculadora y asegúrate que funcione como esperas, después escríbelo en el cuadro de abajo.

?→D:D÷5+1

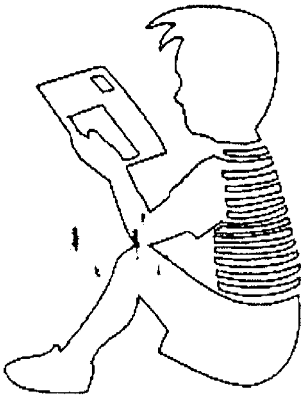
Corregir

HOJA DE TRABAJO NUMERO 45 (Formato 5)

NOMBRE Pedro San Juan de Aida FECHA 2/5/95

?	
2	
	4
?	
5	
	25
?	
7	
	49
?	
8	
	64
?	
10	
	100

En mi calculadora escribí un programa que hace lo siguiente:



1. ¿Puedes adivinar qué programa hice? Si lo encuentras pruébalo en tu calculadora y escríbelo en el cuadro de la derecha.

?→C:C²

2. Ahora inventa un programa que "deshaga" lo que hace mi programa ? Es decir que te dé los números que se muestran en la siguiente tabla:

4	25	49	64	100
2	5	7	8	10

Si pudiste hacer el programa escríbelo en el cuadro de la derecha.

?→C:C√ √C

3. Para cada uno de los siguientes programas inventa otro programa que los "deshaga"? Prueba cada uno de tus programas en tu calculadora y asegúrate que funcionen como esperas, después escríbelos en los cuadros de la derecha

Debes correr tus programas para estar seguro de tus respuestas.

? → A : A × 1.5 + 1	?→A:A÷1.5−1
? → K : 0.5 × K − 1	?→K:0.5÷K+1
? → X : 0.25 × X + 2	?→X:.25÷X+2

4. ¿Encontraste un "método" para invertir programas? Explca en qué consiste.

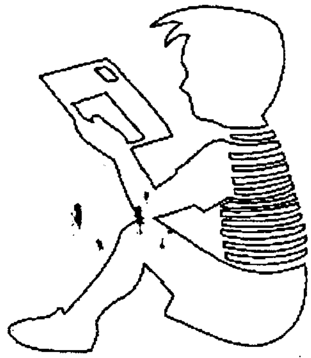
Para que lo hagan las operaciones se cambian así se encuentran los programas

HOJA DE TRABAJO NUMERO 45 (Formato 5)

NOMBRE Georgina Merino FECHA ________

?	
2	
	4
?	
5	
	25
?	
7	
	49
?	
8	
	64
?	
10	
	100

En mi calculadora escribí un programa que hace lo siguiente:



1. ¿Puedes adivinar qué programa hice? Si lo encuentras pruébalo en tu calculadora y escríbelo en el cuadro de la derecha.

   ?→E:E×E

2. Ahora inventa un programa que "deshaga" lo que hace mi programa ? Es decir que te dé los números que se muestran en la siguiente tabla:

4	25	49	64	100
2	5	7	8	10

Si pudiste hacer el programa escríbelo en el cuadro de la derecha.

?→E:E÷E ?

3. Para cada uno de los siguientes programas inventa otro programa que los "deshaga"? Prueba cada uno de tus programas en tu calculadora y asegúrate que funcionen como esperas, después escríbelos en los cuadros de la derecha

?→A : A × 1.5 + 1	?→A (A−1) ÷ 1.5
?→K : 0.5 × K − 1	?→K : (K+1) ÷ 0.5
?→X : 0.25 × X + 2	

Completa

4. ¿Encontraste un "método" para invertir programas? Explca en qué consiste.

____

____

____

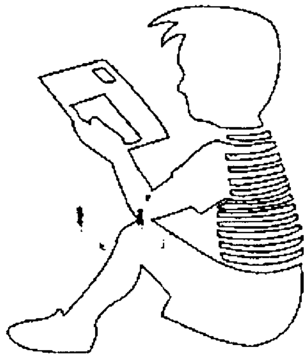
?

HOJA DE TRABAJO NUMERO 45 (Formato 5)

NOMBRE GUILLERMO OLVERA FECHA 6/MAYO/94

?	
2	
	4
?	
5	
	25
?	
7	
	49
?	
8	
	64
?	
10	
	100

En mi calculadora escribí un programa que hace lo siguiente:



1. ¿Puedes adivinar qué programa hice? Si lo encuentras pruébalo en tu calculadora y escríbelo en el cuadro de la derecha.

   ?→A:A×A

2. Ahora inventa un programa que "deshaga" lo que hace mi programa ? Es decir que te dé los números que se muestran en la siguiente tabla:

4	25	49	64	100
2	5	7	8	10

Si pudiste hacer el programa escríbelo en el cuadro de la derecha.

?→A:√A

3. Para cada uno de los siguientes programas inventa otro programa que los "deshaga"? Prueba cada uno de tus programas en tu calculadora y asegúrate que funcionen como esperas, después escríbelos en los cuadros de la derecha

? → A : A × 1.5 + 1	?→A:(A−1)÷1.5
? → K : 0.5 × K − 1	?→A:(A+1)÷0.5
? → X : 0.25 × X + 2	?→X:(X−2)÷0.25

4. ¿Encontraste un "método" para invertir programas? Explca en qué consiste.

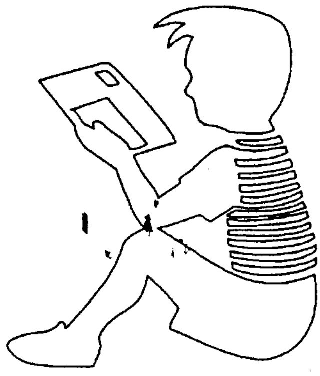
   HAY QUE INVERTIR LOS SIGNOS Y USAR PARENTESIS.

HOJA DE TRABAJO NUMERO 45 (Formato 5)

NOMBRE Rojas y Morales José FECHA 2/May/94

?	
2	
	4
?	
5	
	25
?	
7	
	49
?	
8	
	64
?	
10	
	100

En mi calculadora escribí un programa que hace lo siguiente:



1. ¿Puedes adivinar qué programa hice? Si lo encuentras pruébalo en tu calculadora y escríbelo en el cuadro de la derecha.

   ? → A: A²

2. Ahora inventa un programa que "deshaga" lo que hace mi programa ? Es decir que te dé los números que se muestran en la siguiente tabla:

4	25	49	64	100
2	5	7	8	10

Si pudiste hacer el programa escríbelo en el cuadro de la derecha.

? → A: √A

3. Para cada uno de los siguientes programas inventa otro programa que los "deshaga"? Prueba cada uno de tus programas en tu calculadora y **asegúrate que funcionen** como esperas, después escríbelos en los cuadros de la derecha

? → A : A × 1.5 + 1	? → A: (A − 1) ÷ 1.5
? → K : 0.5 × K − 1	? → K: (K + 1) ÷ 0.5
? → X : 0.25 × X + 2	? → X: (X − 2) ÷ .25

Bien!

4. ¿Encontraste un "método" para invertir programas? Explica en qué consiste.

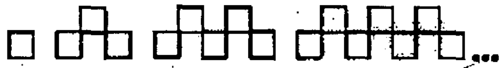
en poner la operacion inversa y

HOJA DE TRABAJO NUMERO 46 (Formato 6)

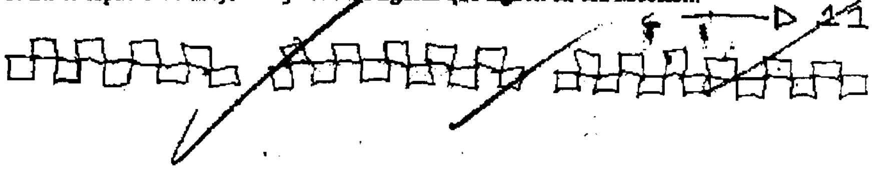
NOMBRE José Ramón J. Rodríguez F. FECHA 18/5/94

Observa las siguientes figuras.

1 ——▷ 1
2 ——▷ 3
3 ——▷ 5
4 ——▷ 7
5 ——▷ 9
6 ——▷ 11



1. En el espacio de abajo dibuja las dos figuras que siguen en esa sucesión.



2. ¿Cuántos cuadrados se necesitan para construir la figura que va en el lugar número 17?
33

3. ¿Cuántos cuadrados se necesitan para construir la figura que va en el lugar número 100?
199

4. Explica cómo razonaste para responder las preguntas 2 y 3.
haciendo un programa que hiciera lo mismo

5. ¿Puedes programar tu calculadora para completar la siguiente tabla?

Lugar que ocupa la figura en la sucesión	Número de cuadrados que se necesitan
48	95
75	149
123	245
[illegible]	351
[illegible]	411
1015	507

Escribe aquí el programa que hiciste.

? → A : A + A + 1

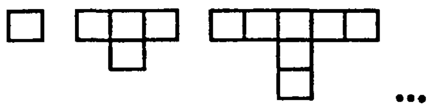
No lo entiendo, explícamelo en la clase.

Continúa para que termines esta hoja.

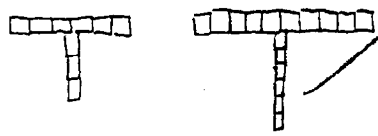
HOJA DE TRABAJO NUMERO 47 (Formato 6)

NOMBRE Pedro San Juan de Ande FECHA 5/18/94

Observa la siguiente sucesión de figuras.



1. En el espacio de abajo dibuja las dos figuras que siguen en esa sucesión.



2. ¿Cuántos cuadrados se necesitan para construir la figura que va en el lugar número 9?
   25

3. ¿Cuántos cuadrados se necesitan para construir la figura que va en el lugar número 17?
   49

4. Explica cómo razonaste para responder las preguntas 2 y 3.
   Siguindo la sucesion fue como encontre el Resultado

5. ¿Puedes programar tu calculadora para completar la siguiente tabla?

Lugar que ocupa la figura en la sucesión	Número de cuadrados que se necesitan
48	142
75	223
123	367
143	427
157	469
201	601

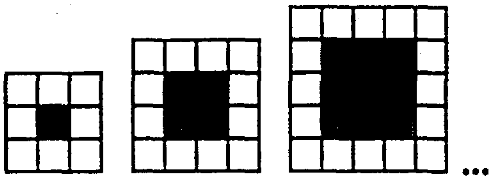
Escribe en este cuadro el programa que hiciste.

7→A: A×3-2

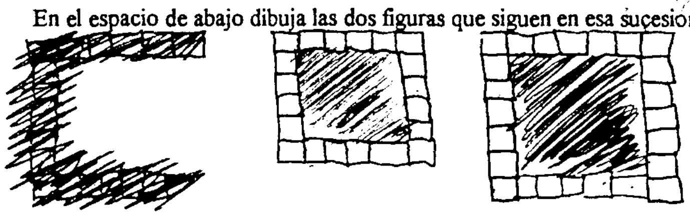
HOJA DE TRABAJO NUMERO 48 (Formato 6)

NOMBRE Rojas y Morales José FECHA 20/May/94

Observa la siguiente sucesión de figuras.



1. En el espacio de abajo dibuja las dos figuras que siguen en esa sucesión.



2. ¿Cuántos cuadrados se necesitan para construir el marco del cuadrado gris en la figura que va en el lugar número 27?
 112

3. ¿Cuántos cuadrados se necesitan para construir el marco del cuadrado gris en la figura que va en el lugar número 40? 164

4. Explica cómo razonaste para responder las preguntas 2 y 3.
multiplique el numero por 4 y sume 4

5. ¿Puedes programar tu calculadora para completar la siguiente tabla?

Lugar que ocupa la figura en la sucesión	Número de cuadrados que se usan en el marco
48	196
75	304
123	496
175	704
192	772
209	840

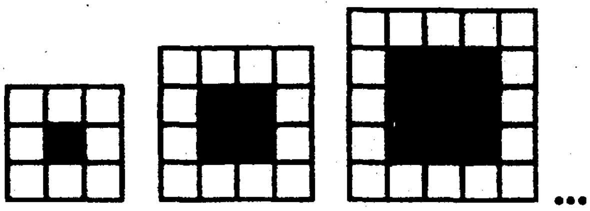
Escribe en este cuadro el programa que hiciste.
?→A: A×4+4
?→A: (A−4)÷4
?→A: A÷4−1
?→A: A÷4−4+3

MUY BIEN

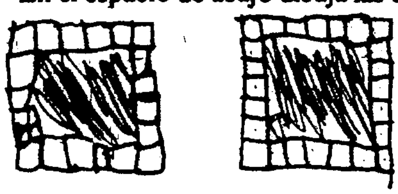
HOJA DE TRABAJO NUMERO 48 (Formato 6)

NOMBRE José Luis Zaldivar **FECHA** 29/V/94

Observa la siguiente sucesión de figuras.



1. **En el espacio de abajo dibuja las dos figuras que siguen en esa sucesión.**



2. **¿Cuántos cuadrados se necesitan para construir el marco del cuadrado gris en la figura que va en el lugar número 27?** 112

3. **¿Cuántos cuadrados se necesitan para construir el marco del cuadrado gris en la figura que va en el lugar número 40?** 164

4. **Explica cómo razonaste para responder las preguntas 2 y 3.**

el numero de la fig. x 4 + 8

5. **¿Puedes programar tu calculadora para completar la siguiente tabla?**

Lugar que ocupa la figura en la sucesión	Número de cuadrados que se usan en el marco
48	196
75	304
123	496
175	704
192	772
209	840

Escribe en este cuadro el programa que hiciste.

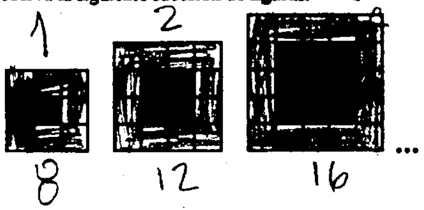
? → A0 A x 4 + 4

Bien

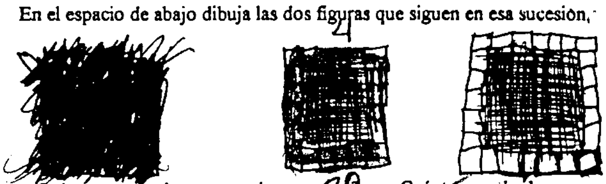
HOJA DE TRABAJO NUMERO 48 (Formato 6)

NOMBRE [illegible] **FECHA** 23/05/94

Observa la siguiente sucesión de figuras.



1. En el espacio de abajo dibuja las dos figuras que siguen en esa sucesión.



2. ¿Cuántos cuadrados se necesitan para construir el marco del cuadrado gris en la figura que va en el lugar número 27? 112

3. ¿Cuántos cuadrados se necesitan para construir el marco del cuadrado gris en la figura que va en el lugar número 40? 124

4. Explica cómo razonaste para responder las preguntas 2 y 3.

segui la sucesion de las figuras hasta llegar a la que se me indicaba

5. ¿Puedes programar tu calculadora para completar la siguiente tabla?

Lugar que ocupa la figura en la sucesión	Número de cuadrados que se usan en el marco
48	108
75	164
123	262
179	704
204	772
252	840

Escribe en este cuadro ~~el~~ programa que hiciste.

? → A: A + 4

Corrige este programa

El programa te debe dar el número de cuadros en el marco de las figuras

¿Qué le embarraste al sobre? Está hecho un mugrero.

Corregir y completar

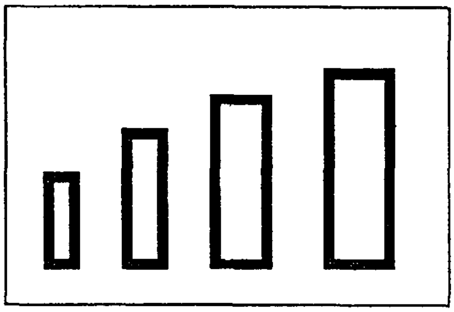
HOJA DE TRABAJO NUMERO 49 (Formato 6)

NOMBRE Georgina Merino PA FECHA 10/05/94

VENTANAS

En la sala de escultura de un museo de Arte Moderno las ventanas tienen las siguientes características:

Las ventanas tienen distintas medidas, pero en todas la altura mide el triple de lo que mide el ancho



1. ¿Puedes completar la siguiente tabla?

Ancho de la ventana	0.75 m	0.86 m	1.28 m	1.17	1.41
Altura de la ventana	2.25 m	2.58 m	3.84	3.51 m	4.23 m

2. Los marcos de las ventanas están hechos con madera cuyo precio por metro es $ 53.00.

   a) ¿Cuál es el costo del marco de una ventana que mide 1.5 metros de ancho? $636

   b) ¿Qué operaciones hiciste para calcular ese costo? multiplicas 53 por el perímetro

3. ¿Puedes hacer un programa que te permita calcular el costo del marco para cualquiera de las ventanas de esa sala del museo? Escribe tu programa en el cuadro de abajo.

   7 → F : (Fx3) + (Fx2) x 53.00

   7 → F : (Fx3)x2+(Fx2)+53.00

   CORREGIR

4. Usa el programa que hiciste para completar la siguiente tabla.

Ancho de la ventana	0.68 m	0.80 m	0.95 m	0.98 m	1.15 m	1.75 m
Costo del marco	$ 288.32	$ 339.20	$ 402.80	$ 415.52	$ 487.60	$ 530

Corregir

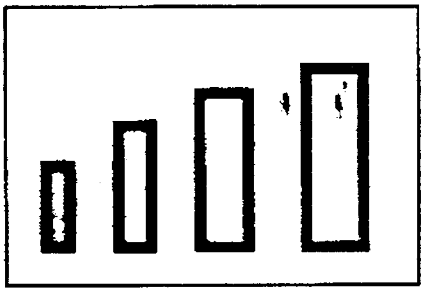
HOJA DE TRABAJO NUMERO 49 (Formato 6)

NOMBRE José Luis Zaldívar FECHA 20/5/99

VENTANAS

En la sala de escultura de un museo de Arte Moderno las ventanas tienen las siguientes características:

Las ventanas tienen distintas medidas, pero en todas la altura mide el triple de lo que mide el ancho



1. ¿Puedes completar la siguiente tabla?

Ancho de la ventana	0.75 m	0.86 m	1.28 m	1.17	1.41
Altura de la ventana	2.25	2.58	3.96	3.51 m	4.23 m

2. Los marcos de las ventanas están hechos con madera cuyo precio por metro es $ 53.00.

a) ¿Cuál es el costo del marco de una ventana que mide 1.5 metros de ancho?
636

b) ¿Qué operaciones hiciste para calcular ese costo? primero multiplique X3 para sacar el perímetro y multiplique X los metros por el precio

3. ¿Puedes hacer un programa que te permita calcular el costo del marco para cualquiera de las ventanas de esa sala del museo? Escribe tu programa en el cuadro de abajo.

A x 53

Fíjate que este programa no te da este resultado. Tienes que corregirlo, piensa como le hiciste para obtener 636 y trata de hacer el programa

4. Usa el programa que hiciste para completar la siguiente tabla.

Ancho de la ventana	0.68 m	0.80 m	0.95 m	0.98 m	1.15 m	10 m
Coste del marco	$36.04	$42.4	$50.35	$51.94	$60.95	$ 530

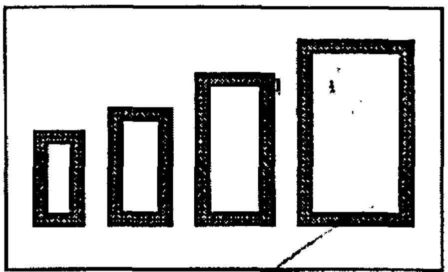
HOJA DE TRABAJO NUMERO 50 (Formato 6)

NOMBRE José Luis Zaldivar F. FECHA 27/V/94.

MAS VENTANAS

En la sala de arquitectura del Museo de Arte Moderno las ventanas tienen las siguientes caracteristicas:

Las ventanas tienen distintas medidas, pero en todas su altura mide 50 cm. menos que el triple de lo que mide el ancho.



1. ¿Puedes completar la siguiente tabla? 1.65

Ancho	0.30 m	0.45 m	1.30	~~[illegible]~~	2.35
Altura	~~[illegible]~~ .4	~~[illegible]~~ .85	~~[illegible]~~ 3.4	4.45	6.55 m

2. Los marcos de las ventanas están hechos de madera cuyo precio es $ 62.00 por metro.
   a) ¿Cuál es el costo del marco de una ventana que mide 1.3 metros de ancho?
   $582.80
   b) ¿Qué operaciones hiciste para calcular ese costo? ~~[illegible]~~ ~~x 62~~ Multiplicar, sumar y restar

3. ¿Puedes programar tu calculadora para obtener el costo del marco para cualquiera de las ventanas de esa sala del museo?

   Escribe el programa que hiciste en el cuadro de la derecha.

?? A°(A + A X3 - 50) X2X62

4. Usa el programa que hiciste para completar la siguiente tabla.

Ancho de la ventana	0.35 m	0.65 m	0.84 m	~~1.20 m~~ 1.15
Costo del marco	11.60	5505.6	5141.94	5307.2	$ 334.00

HOJA DE TRABAJO NUMERO 52 (Formato 6)

NOMBRE Mirsha Balcázar Saldaña FECHA 27/05/94

En una tienda de libros y discos están haciendo la siguiente oferta.

15% DE DESCUENTO EN TODA LA MERCANCIA
El descuento se aplica en la caja sobre el precio marcado en la etiqueta.

1. Completa la siguiente tabla.

Precio en la etiqueta	Cantidad que se descuenta	Precio de oferta
$ 34.00	5.1	28.9
$ 18.75	2.8125	15.9375
$ 126.80	19.02	107.78
$ 28.50	4.275	24.225
$ 150.00	22.5	127.5
$ 72.35	10.8525	61.4975
$ 29.40	4.41	24.99

2. ¿Puedes programar tu calculadora para que haga lo siguiente?
Si le das **el precio de etiqueta** te dé por resultado **el precio de oferta.**

Escribe el programa que hiciste en el cuadro de la derecha.

?→A:A-(A×.15)

1. Usa el programa que hiciste para completar la siguiente tabla.

Precio en la etiqueta	**Precio de oferta**
$ 84.00	71.4
$ 28.75	24.4375
$ 226.80	192.78
$ 29.60	25.16
$ 140.00	119
168	$ 142.80
170	$ 144.50

B

HOJA DE TRABAJO NUMERO 53 (Formato 6)

NOMBRE Guillermo Olvera FECHA 27/Mayo/94

En una papelería están haciendo la siguiente oferta.

25% DE DESCUENTO EN TODA LA MERCANCIA
El decuento se aplica en la caja sobre el precio marcado en la etiqueta.

1. De acuerdo con esa información completa la siguiente tabla.

Precio en la etiqueta	**Cantidad que se descuenta**	**Precio de oferta**
75	$ 18.75	56.25
24	$ 6.00	18
36	$ 9.00	27
86	$ 21.50	64.5
35	$ 8.75	26.25
26	$ 6.50	19.5
46	$ 11.50	34.5

2. Programa tu calculadora para que haga lo siguiente: Si le das al programa la **cantidad que se descuenta**, te debe dar como resultado el **precio de oferta**. Escribe tu programa: (Ax4)-A

3. Programa tu calculadora para que haga lo siguiente: Si le das al programa la **cantidad que se descuenta**, te debe dar como resultado el **precio marcado en la etiqueta**. Escribe tu programa: Ax4

4. Usa los programas que hiciste para completar las siguientes tablas.

a)

Cantidad que se descuenta	$ 15.40	$ 18.75	$ 8.90	$ 10.00	$ 14.35
Precio de oferta	61.6	75	35.6	40	57.4

b)

Cantidad que se descuenta	$ 11.70	$ 6.75	$ 8.90	$ 8.40	$ 9.60
Precio marcado en la etiqueta	35.1	20.25	26.7	25.2	28.8