EXPRESSING GENERALITY:

YOUNG CHILDREN, MATHEMATICS AND LOGO

DUNCAN HUGH JONES

Thesis submitted in fulfilment of the requirements for the Ph.D. degree of the University of London

INSTITUTE OF EDUCATION · UNIVERSITY OF LONDON · AUGUST 1998

ABSTRACT

This thesis looks at young children's attempts to express generality within specific learning situations with the aid of carefully designed tools, and at how, given appropriate means of expression, they are able to justify their generalisations. The literature on representation and abstraction leads to a focus on construction of meaning through pupils' concretising of mathematical objects by means of the development of representations and interconnections between representations of those objects.

The study involved eight pairs of ten and eleven year old children. I devised a series of tasks centred on the creation of simple "function machines" expressed in the Logo programming language. The children's work involved the construction and empirical testing of these functions within a game-like situation. Part of the game involved verbally justifying the validity of the Logo procedures to a partner and to the Researcher. These activities provided a window onto children's construction of meaning.

Analysis of the data revealed that within the specific learning situation designed for the study: children were able to make formalised generalisations of mathematical relationships, often webbed by "semi-generalisation"; the expressive powers of the symbolism achieved a more functional role by the symbols' association with a history of specific numerical examples; children constructed situated abstractions for the justification of generality using "generic structuring" and "naturalised formalism" as powerful forms of webbing; and the apparent "rift" between empirical and deductive starting points for generalisation, justification and proving activities appeared less clear than the literature suggests.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

CHAPTER FOUR

CHAPTER FIVE

CHAPTER SIX

CHAPTERELEVEN

ACKNOWLEDGEMENTS

I should like to thank Professor Richard Noss for his support, encouragement and inspiration over the years in which this thesis has taken shape. His helpful criticism of my work and willingness to offer advice, even to the extent of journeying to the depths of Cornwall, has been appreciated greatly.

Such is the time scale of this piece of work, that the children who form the subjects of the study have now left Mount Charles School where they were pupils at the time of the research. However, I should like to thank them and their parents for their time and commitment, and also the staff and former Headmaster for their co-operation.

And then, of course there's my wife. She has had the good grace to say that this research has barely impinged on my home life, but at the risk of contradicting her, this is manifestly untrue. I want to thank Fiona for all her encouragement and patience over these last few years. It is to her that this thesis is dedicated.

CHAPTER ONE: INTRODUCTION

The expression of generality is a process which lies at the very heart of mathematics, yet whilst this centrality of generalisation to mathematics is undisputed, there appear to be many problems associated with children's ability to express that generality within current school curricula. The United Kingdom National Curriculum promotes an approach to generalisation which begins with pattern spotting through scrutiny of number patterns expressed, for example, in spatial arrangements, and leads on to generalisation of that pattern in words and eventually in simple algebra (see below). Two important issues emerge from such an approach: the means of expression of generality; and the extension of activities involving the generalisation of mathematical relationships to those which incorporate justification and eventual proof of that generalisation.

Looking first at problems in the expression of generality, the promotion of pattern spotting activities as points of access to algebraic expression has resulted in problems which differ from, but are as important as the problems which appear to result from the introduction of algebra in less apparently "real" contexts. Where algebra is viewed solely as an abstract language, the manipulation of whose symbols becomes the prime concern of school algebra lessons, pupils fail to apportion meaning to the symbols which they are manipulating; where, however, algebraic expressions are viewed as the goal to be reached through pattern spotting and rule writing, pupils are unable to make the necessary "cognitive leaps" towards this final goal unless they have at their fingertips an algebraic language with which they can explore and identify structure and express their findings. In other words, unless there is an interaction between the manipulation of algebraic syntax and the use of generalising and symbolising activities, neither approach is sufficient in itself to develop pupils' ability to express generality.

Turning now to the extension of generalising activities to embrace justification and proof, school curricula - and, by extension, school texts - appear to place a constraint on what might be thought of as a continuum which stretches from simple pattern spotting through generalisation, empirical testing and justification to formal mathematical proof. Most generalisation activities are looked on as self contained, and few look onward towards justification and proof of the generality.

The common theme which unites these two issues is the role of formalism: the use of formalism to express generality in situations which may begin from an informal basis and where formalism is used as an end result; and the use of formalism as a tool for the justification of generality.

My over-riding concern in this thesis is to look at learning situations in which children are able not only to generalise from the identification of mathematical patterns, but also to justify these generalisations as part of one extended mathematical activity in which algebraic formalism is used both to express generality and to structure justification. I propose a computer programming environment as a possible situation in which this may occur (for reasons I outline below). Thus the central research issue of the thesis is *to investigate how use of a carefully designed computer programming environment structured children's expression of generality and the justification of that expression.*

Having already mentioned the United Kingdom National Curriculum, it is useful to look briefly at the messages it broadcasts. The National Curriculum has appeared in a number of guises (DES 1989, DES 1991, DES 1995) and its contents have been successively slimmed down under pressure from various bodies. The section which refers to generalisation and proof (Attainment Target 1: Using and applying mathematics) in the original curriculum (DES 1989) and in the first revision (DES 1991) most explicitly reflects the current approach to generalisation and proof in

British schools[1]:

Level 1:	making predictions based on experience
Level 2:	asking and responding to questions, e.g. "What would happen if ...?", "Why?"
Level 3:	investigating and testing predictions and general statements; checking results, considering whether they are sensible
Level 4:	using examples to test solutions, statements or definitions; making generalisations or simple hypotheses
Level 5:	generalising from a number of particular examples and carrying out simple tests
Level 6:	making and testing generalisations and simple hypotheses; defining and reasoning in simple contexts with some precision
Level 7:	following a chain of mathematical reasoning; spotting inconsistencies
Level 8:	making statements of conjecture using "if ... then"; defining, reasoning, proving and disproving, using counter-examples; construct an extended chain or argument using "if ... then" appropriately
Level 9:	stating whether a conjecture is true, false or not proven; defining and reasoning; proving and disproving; using symbolisation; recognising and using necessary and sufficient conditions
Level 10:	giving definitions which are sufficient and minimal; using symbolisation with confidence; constructing a proof, including proof by contradiction

[1] In the current (1995) version the levelling is reorganised but the progression of ideas remains largely unaltered.

To put this briefly into perspective, the "average" seven year-old is expected to reach Level 2, the average eleven year-old Level 4 and the average sixteen year-old (on sitting GCSE examinations and completing compulsory mathematics education) is expected to reach around Level 6 or 7.

The emphasis here, at least until Level 7, is entirely on justification by means of empirical methods and at Level 8 the examples given to illustrate the statement are of the type "search for a counter example to demonstrate that this is not always true". At Level 9 the illustrative example involves justifying the solution to a practical problem:

> *When reporting on the best way to stock and transport pipes, explain solutions in terms of costing, volume, surface area and stability.*

Only at the final level is there reference to abstract concepts of proof. One example given is:

> *Follow from a book Euclid's proof by contradiction that $\sqrt{2}$ is irrational, write out a similar proof for $\sqrt{3}$ and find why it breaks down for $\sqrt{4}$.*

Pupils would not seem to be learning about justification and proof in their school curricula. The focus on looking for patterns in either spatial, visual or numerical examples and on attempting to generalise these patterns in natural and then algebraic language rarely leads any further than these activities. The activity stops, "precisely at the point where a central process of mathematics, namely justification and proof, should begin" (Anderson 1995, p. 48).

Now, the overriding impression given by consideration of the UK National Curriculum is one of the apparent importance of levels: the unstated assumption is that pupils must satisfy the requirements of one level before moving onto the next, with the direct implications for teaching that this frame of mind implies. Use of levels, and the location of pupils at some point within a hierarchy of levelled activities, is not confined to school curricula. Much research has at its heart a model involving levels of achievement or levels of abstraction. For example, Sfard[2] describes three stages of abstraction (interiorization, condensation and reification); and Balacheff[3] describes a levelled model for defining types of proof (Naive Empiricism, Crucial Experiment, Generic Example and Thought Experiment).

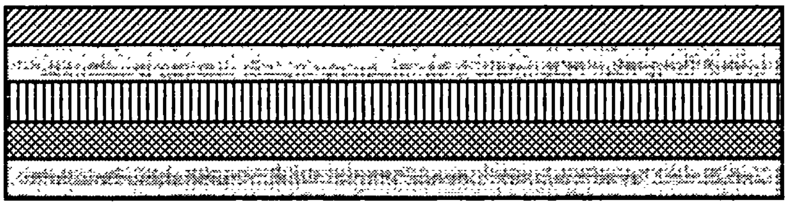


Figure 1.1: Model G - General Learning Principles represented as a cross-section of layers, rather like geological strata of rock types

Much useful research has resulted from an approach which emphasises the importance of hierarchies, but is this always the most appropriate way of thinking about how children build meanings or mathematical (and other) concepts? In talking about the use of theoretical frameworks to describe learning, diSessa describes the notion of G types, that is a version of constructivism which "relies on General, powerful learning principles (like reflective abstraction) and does not pay much attention to the specific schemata that form the naive interpretations out of which reflection (to name one mechanism) builds more adequate ones" (diSessa 1994, p. 251). DiSessa's General Learning Principles might be represented as a cross-section of

[2] See Chapter 2, § 2.3.2

[3] See Chapter 3, § 3.2

layers, rather like geological strata of rock types (Figure 1.1). Pupils move from one level to the next: each level must be thoroughly assimilated before the next level is entered.

However, an alternative view of the construction of knowledge "believes that the naive state, as well as the expert state, needs thorough analysis, both in terms of the Specific naive schemata that form the grist out of which better developed ideas evolve, and also, naturally, in terms of the theoretical categories that describe this thinking" (ibid p. 251). DiSessa terms this the S version. Using my model of a stratified cross-section, specific naive schemata might appear as columns - S-columns - cutting through the layers (Figure 1.2). The S-columns represent specific, situated knowledge which may touch several layers of abstraction: they may cut through the entire body of knowledge or they may touch on one or two layers.

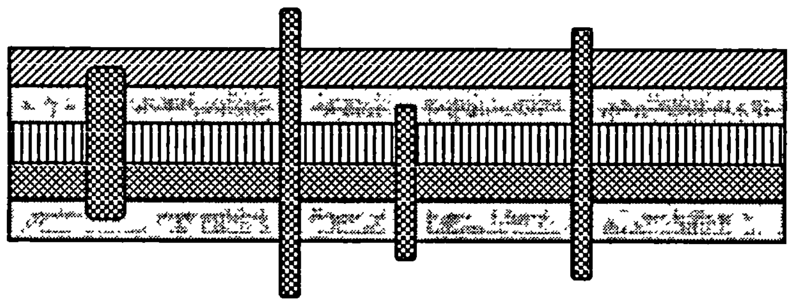


Figure 1.2: Model S - Specific Naive Schemata: situated "bore holes" are shown cutting through the strata of ordered knowledge

Now, if generalisation, justification and proof were represented by such a multi-layered block, the S-columns would then represent specific situated knowledge containing elements from many layers of abstraction, such as generalisation, empirical testing and formal deductive proof, combined within a single, narrowly-focused mathematical experience. A useful way of thinking about these S-columns might be as bore holes, boring into the strata of knowledge. The challenge is to sink some of these bore holes in carefully constructed situations and to investigate the relationships thereby revealed.

The aims of the thesis, then, were to look at how children might construct meanings for generality and justification, and more specifically:

i) to investigate ways in which children expressed generality using the programming medium;

ii) to investigate connections between different expressions of generality made in different modalities and their role in children's creation of mathematical meaning;

iii) to investigate the role of connections between expressions of generality in children's attempts to justify their constructed generalisations.

A programming environment was chosen, using the constructionist tenet that the internal construction of meaning is best seen through the external construction of a mathematical (or other) object, as providing a window onto children's expression of generality and justification. This theme is developed in Chapter Two.

The study centred around pairs of children - all aged between nine and eleven - working on tasks which involved modelling mathematical relationships, presented in natural language, as mathematical functions on a computer using the programming language Logo. A learning situation was constructed in which the two children in a pair worked "competitively" on separate computers with the ultimate aim of convincing each other of the validity of their procedures and justifying why one procedure might work where the other might not. Programming with Logo was chosen as providing an environment where pupils would be able to express mathematical generality through the construction of procedures (a form of algebraic formalism), verify their procedures using specially created features of the medium, and justify their own constructions using the programming language of construction.

The thesis is organised into eleven chapters. In Chapter Two I discuss the major theoretical influences which have a bearing on the stance adopted in the thesis. Chief amongst these are theories of representation and abstraction, a redefinition of what constitutes concrete and abstract including the idea of concretion, the notion of situated abstraction, and the place of microworlds and convivial tools in opening windows onto the construction of meaning.

The thesis is concerned with children expressing formal arguments. Justification and, in a broadly-defined sense, proof of children's generalisations made on the basis of their own mathematical experiences provided a situation in which children were able to make such formal arguments. Thus in Chapter Three I review the Literature on proof, focusing both on a dichotomised view of the teaching of proof which emerges from much of the literature with an accompanying concentration on what children fail to achieve, and on research which focuses on what children can achieve given appropriate tools and carefully constructed learning situations.

In Chapter Four I describe the design of the learning situation used in the study, in particular the computational medium, the mathematical activities and the relationship between pupil and researcher, including an analysis of the important role adopted by the researcher.

In Chapter Five I present the Research Methodology for the initial phases of the study and trace their evolution and the evolution of the design of the learning situation prior to the main study. In Chapter Six I develop some of the ideas on abstraction discussed in Chapter Two in light of the initial phases of the study, defining the terms *expressions of meaning* and *connections*. From here I move on to present a methodology for the main study of the thesis and analyse some preliminary findings from the research data. These findings are used to frame the final research questions and to create four categories in which I analyse the study data in Chapters Seven, Eight, Nine and Ten.

Finally, in Chapter Eleven I draw conclusions and implications from the discussion in the preceding four chapters, and identify areas for further research.

CHAPTER TWO: REPRESENTATION AND ABSTRACTION

2.1 INTRODUCTION

Whether one looks on the acquisition of knowledge as an orderly ascension through pre-determined hierarchies - diSessa's G types described in Chapter One (diSessa 1994 p. 251), or as an altogether less easily categorisable process involving the creation of connections within narrowly situated contexts (the S type model), a central focus common to both these viewpoints is theories of representation and abstraction, and it is to these two processes that I turn first.

In this chapter I look first at two broad interpretations of the term *representation* and at how, evolving from these interpretations, come differing understandings of the role of representations. From this I move on to a discussion of processes of abstraction, introducing an important view of abstraction which reinterprets the nature of abstract and concrete. A consideration of some of the problems associated with prevalent theories surrounding the process of abstraction leads into an exploration of the idea of scaffolding and webbing, looking at how pupils may be able to construct meaning within these cognitive frameworks. Finally I look at how use of computer programming may provide a window onto how pupils construct their own representations.

2.2 REPRESENTATION

2.2.1 Defining Representation

Use of the word *representation* encompasses such a breadth of meaning in the context of mathematics education as to cloud that use in ambiguity. Most writers divide definitions of representations into two broad groups depending on whether they are discussing symbolic systems or the internal mental organisation of knowledge. Janvier (1987c, p. 148), for example, makes this distinction when he uses the term *schematization* or *illustration* to refer to "some material organisation of symbols ... which refers to other entities or 'modelizes' various mental processes"; and the term *conception* to describe mental images and the organisation of knowledge in the human mental system. Concentration on the former type of representation, a viewpoint he shares with Goldin (1987b) and Kaput (1987a), approaches representation from an external aspect, attempting to produce *external representations* which model people's mental representations of the world.

Another group of writers (such as von Glasersfeld 1987a, Mason 1987b) also differentiate between what might be called mental representations and external representations, although they prefer to approach problems of representation from inside, attempting to "capture experience of the inner world by using metaphor and descriptive frameworks that resonate with other people's experience" (Mason 1987 b, p. 208).

I want now to look at what characterises these two interpretations and to consider some of the ensuing implications.

2.2.2 Representation Systems

Kaput, for example, sees representation as a structure of mappings between symbol systems. Representation involves two "related but functionally separate entities", the *representing world* and the *represented world* (Kaput 1987a, p. 23). The act of representation implies some sot of correspondence between these two entities. He defines a *symbol scheme* as a concretely realisable collection of characters together with explicit rules for identifying and combining them; and a *symbol system* as the system which contains a symbol scheme, a field of reference and a systematic rule of correspondence. Mappings between symbol systems (translations) and within symbol systems (transformations) become of paramount importance in demonstrating understanding of a particular problem (see, for example, Lesh, Post and Behr 1987). Kaput sees this ability as the key:

> *The students of the near future will be choosing how to represent given relationships. This skill in choosing or building representations, together with interpretative skills, will soon outstrip computational skills by a wide margin.*
>
> Kaput (op cit, p. 21)

But does it always make sense to talk about translating directly from one external representation to another? There are at least three problems with such an approach. Firstly, the nature of mappings between symbol systems. Symbols system theories are developed with mathematical rigour (Kaput 1987b, Goldin 1987b) with the correspondence between the "represented world" and the "representing world" as an integral element of the structure. But in fact neither the mappings, nor indeed the systems are as well-defined as might at first be thought:

> *Just as the signs and configurations of a given representational system can be ill-defined or only partially specified, so can the correspondence between two representational systems be fuzzy. A mapping in mathematics may permit one system to represent another in a very precise way, as when an abstract mathematical group is represented by linear operators in a vector space. On the other hand, words in a purely verbal representational system can represent*

> *objects in a non-verbal representational system, yet the relationship may defy precise specification.*

(Goldin 1987b, p. 131)

Even mappings between two abstract mathematical representations can be 'fuzzy'. For example Kaput (1987 b, p. 185) shows that extension of the standard base ten place holder symbols system to include negative numbers immediately loses one-to-one correspondence.

Secondly, there appears to be a danger in concentrating on how children translate from one external representation to another of losing sight of children's own mental representations and real world referents to which their mathematical ideas should be firmly linked. Mason (1987 b, p. 214) draws an apt analogy with translating foreign prose into English *word for word.* Such a translation may give a decipherable translation of the meaning of each sentence, but it is unlikely to convey any feeling of the overall style or general understanding of the passage as a whole. Unless translations between representations are carried out with a close regard for the overall mathematical context which the representations are modelling, they may become merely a blind, routine exercise which does little to develop a child's understanding of a particular concept.

Thirdly, to what extent do children learn about the concepts embedded in a variety of representations when performing translations from one to another, or to what extent are they merely learning something about the relationship between these representations? Dufour-Janvier, Bednarz and Belanger find that the use of multiple (external) representations results only in pupils putting representations into correspondence one with another so that all that is constructed are "syntactical rules of correspondence rather than constructing the concept itself via its representation" (Dufour-Janvier, Bednarz and Belanger 1987, p. 113).

There is a feeling in all this literature of merely skimming the surface with a neat, precise, manageable (and therefore superficially attractive) theoretical structure, but of one which never gets to the heart of the problem, one which appears to tell only half a story. The emphasis is on how children encounter and make sense of external representations: there appears to be no room for children's creation of meaning through their *own* experience or of how this creation is shaped by the context in which they encounter the mathematics or the tools they have at their disposal. I want now to look at another theory of representation which places emphasis on the internal construction and reconstruction of knowledge.

2.2.3 Cognitive Structures and Mental Representation

The form mental representations must take is dependent on the manner in which knowledge is acquired and processed by the child. If we accept a Constructivist viewpoint which sees knowledge acquired through children's experience of the real world, then that knowledge must be represented in a form that reflects the fragmentary, ever-changing and developing way in which children gain their experience. Lawler (1985) suggests that the world can only be experienced as a collection of disparate *microworlds*[4] and that correspondingly the mind is made up of a system of disparate, active, cognitive structures, built up through interaction with these microworlds, which he terms *microviews*. Microviews are task-rooted, that is descriptions of actual things in the real world. Learning implies an understanding of what unifies these otherwise diverse and fragmentary experiences.

[4] This term is used by Lawler in a manner distinct from that in which I shall use it later in this chapter.

Thus children build up knowledge not in a clearly defined, ordered progression of facts, but in a random, almost accidental manner, developing not so much a clear picture of a concept, but rather a rough sketch which is refined and given more detail as the child gains more experience. Minsky (1986 p. 245) uses the term *frame* to identify these internal cognitive structures. He likens a frame to a "sort of skeleton, somewhat like an application form with many blanks or slots to be filled". Since mental representations (the structure and fill-able slots of a frame) are built on the basis of an individual's past experience, and since no two people have an identical set of experiences for any one concept, it follows that a mental representation is unique to its owner. This is what von Glasersfeld (1987a, p. 6) terms *subjectivity of meaning*.

Through their own experience, children will develop simple theories, however vague and sketchy, to explain phenomena in real life. These theories may not be entirely correct, but they can form a basis on which skilful teaching can build. DiSessa (1987, p. 84) calls these phenomenological primitives (or p-prims). As the child's knowledge increases, so the p-prims may cease to be primitive and some, it may transpire, will be false. However, p-prims can be used as elements of analysis which partially explain more formal ideas.

Mental representations, then, may be viewed as dynamic, constantly developing and unique interior structures which are constructed and re-constructed on the basis of our own experience of the real[5] world. The emphasis on construction here is vital. In fact, the word *presentation* is preferred to representation by some authors (see Mason 1987b, von Glasersfeld 1987b, Dreyfus 1991) since it refers to "a primary creation, to an act of perceptual or imaginal construction, and there is no prior 'object' that serves as original to be replicated or *re*-presented" (von Glasersfeld 1987b, p. 218).

[5] But see Nemirovsky's point about real and unreal in § 2.3.2.

So these two differing schools of thought place emphasis either on exploring pupils' contact and ability to work with pre-determined external representations of a mathematical idea, or on looking for ways to access the mental representations which pupils themselves construct as they create their own meanings for some mathematical concept.

2.3 ABSTRACTION

Closely related to the idea of representation is that of abstraction. This section looks briefly at the role of abstraction in mathematics and outlines some examples from the literature which suggest how concept formation may occur. Discussion of some problems in these theories leads into the description of an important Vygotskian theory and the application of ideas contained within this theory to mathematical learning situations.

2.3.1 The Role of Abstraction

The verb *to abstract* means to draw away, and much of the power of mathematics lies in its ability to draw away from the particular to express the general (see Mason 1989). People who, for one reason or another, do not like mathematics complain that it is this abstraction that is the cause of their dislike. This may be due to their mathematical experience commonly being confined to using mathematics in an abstracted form (i.e. in an abstraction created by someone else) but never making abstractions for themselves. This is particularly apparent in the mechanical use of symbols in mathematics, an activity all too often treated as sufficient in itself. This results in pupils rarely doing more than manipulating other people's symbols. Whilst practice in the use of symbols is important, its relevance may be questioned if pupils have no experience of symbolising for themselves (i.e. of expressing an idea symbolically in terms of their own mental representations). Since pupils rarely have much experience of the process of abstraction and only of already abstracted ideas, they are likely to view abstracted mathematical concepts as divorced from reality rather than be aware of the process by which they have been drawn from the particular to the general. Thus teaching methods which encourage developing awareness of roots with "real" ideas need to be developed if pupils are to work successfully with abstracted ideas in mathematics.

2.3.2 Processes of Abstraction

Abstraction has been described as a "delicate shift of attention from seeing an expression as an expression of generality, to seeing the expression as an object or property" (Mason 1989, p. 2). How this "delicate shift" comes about is discussed by a number of authors. For example, Sfard (1991, p. 18) describes three stages in the shift from "operational conceptions" (processes, algorithms and actions which are dynamic and sequential) towards "structural conceptions" (abstract, static objects which can be manipulated as a whole).

The problem with such theoretical models which encompass carefully defined stages is that the world being modelled rarely fits neatly into the stages we define. For example, it is interesting to note that in the standard algebraic notation of a function such as $y = 3x^4$, the = sign can be interpreted as a command (operational) or as a symbol of identity (structural). Furthermore an expression such as 6x is both indication of a problem and the name for the answer. Here lies both a strength and a weakness of algebraic symbolism. (For discussions of the ambiguity between process and product see, for example Davis 1975, Sfard and Linchevski 1994, and Gray and Tall 1994.)

Sfard's model of abstraction is essentially recursive, so that the newly formed object itself becomes the focus of interiorization at a higher level[6]. The cyclical nature of concept formation is a feature of models suggested by Mason (1980) and Hiebert (1988) and in fact all three models have much in common in that they highlight the connectedness between each successive layer of abstraction, which exists by virtue of the final stage in each level providing the first stage at the next level. Three important phases are common to each model:

i) manipulation of the referent: the referent (an object at the first level, symbols at subsequent levels[7]) is manipulated with growing confidence. A widening, but inarticulate sense of some idea is developed

ii) creation of a new symbol: a symbolic record of the first stage is created. Because the new symbol is created through manipulation of the referent, it too is easily manipulated, but in ways which closely mirror the ways in which the original referents were manipulated. Increasingly, the referent ceases to be important, a broader view of the process is developed and the process gains meaning at the symbolic level.

iii) acceptance of the symbol as an object: the new symbol becomes an object which can itself be manipulated and provide the referent for the next level of abstraction.

An important feature of Mason's model is that at any point in the spiral process of abstraction (as he represents it), pupils can "fold back" (Pirie and Kieren 1989, p. 9) to a level at which they feel more confident. Essential to the understanding of these models is the interpretation of the terms "object" and "symbol", an interpretation which can only be made subjectively by the learner at any one point in the learning process. In the simplest scenario, the referent changes from an object at the lowest

6 And of course the richness of mathematical symbolism mentioned is a good example of a representation with a "history" at a different level of abstraction, which can serve to remind the pupil of that other level and of the processes associated with that level.

7 But see the discussion of "Concrete" below.

level of abstraction to a symbolic representation of that object at the next level of abstraction and to further symbolic representations with each successive abstraction; in fact the original referent may not be an object in the usual sense - in some areas of mathematics the most "concrete" representation is itself symbolic. The very process of abstraction dictates that at each successive level of abstraction the new symbol becomes the object for the next abstraction.

Now, this structured model of abstraction fits neatly into the view of concept formation as the execution of mappings between and within symbols systems discussed earlier. However, if the alternative view is adopted, that concepts are built up by means of the acquisition and assimilation of random snippets of information and experience (what Lévi-Strauss, 1963-76, calls a process of *Bricolage*), the model of abstraction begins to appear over rigid and the use of the terms "abstract" and "concrete" must be called into question, for if we accept von Glasersfeld's idea of the subjectivity of meaning, then we must also accept that what is and is not "concrete" or "abstract" must equally vary from one pupil to another and from one time and set of experiences to another. Wilensky makes this very point:

> *Concreteness is that property which measures the degree of our relatedness to the object, (the richness of our representations, interactions, connections with the object), how close we are to it, or, if you will, the quality of our relationship with the object.*
>
> (Wilensky 1993, p. 198)

If we accept such a definition, then it is no longer possible to refer to objects as being either "concrete" or "abstract", only concrete (or abstract) to a particular person within a particular situation. And despite the similarities with the model of abstraction described above, this definition (which I shall refer to as the *Subjectivity of Concreteness*) carries with it further important implications. For example, it questions the *hierarchic nature* of abstraction (as exemplified in models such as the ones outlined above). From Wilensky's definition, any objects may be described as being concrete for a particular person if they have "multiple modes of engagement with them and a sufficiently rich collection of models to represent them" (ibid, p. 198). This does not limit them to occupying a position within some externally imposed hierarchy with concrete objects "beneath" and abstract objects "above". This theme is developed in Chapter Six[8].

Noss and Hoyles see the process of abstraction not as an ascension away from meaning but as a process of creation, producing new collections of meaning: "Meaning can be maintained by involvement in the process of acting and abstracting, building new connections whilst consolidating old ones (Noss and Hoyles 1996, p. 49)."

Subjectivity of Concreteness also brings us back to the issue of connections between representations:

> *The more connections we make between an object and other objects, the more concrete it becomes for us. The richer the set of representations of the object, the more ways we have of interacting with it, the more concrete it is for us.*
> (Wilensky op cit, p. 198)

Thus an important element in pupils' construction of meaning is a process of abstraction which has at its heart the building of connections between children's mental representations of an object.

Also called into question is what constitutes "real" and "unreal" situations for mathematical activity. Nemirovsky makes the point that it is necessary to free ourselves from such stereotypes: "often problems are characterised as being decontextualised because they are just about numbers (as opposed to quantities or measures of specific things), as if all the rich background of ideas and experiences that students develop around numbers could not offer a context" (Nemirovsky 1996 b, p.

[8] See Chapter Six, § 6.2.

313). This idea resonates closely with Wilensky's subjective definition of concrete. In the same way that what is concrete for one person may be abstract for another, so what for one person is a "real" problem may be equally "unreal" for another.

What these models of abstraction and concept formation and Wilensky's definition of the concrete have in common is a view of representation centred on the learner: referents are subjective entities, as is the definition of concrete proposed by Wilensky, and any abstraction that takes place on a concept, changes that concept in a way that is unique to the learner and not necessarily shared by other students - in the same way that concepts are uniquely concrete or abstract depending on where one happens to be standing.

2.3.3 Some Difficulties in Abstraction Theory

There are, however, further problems in the idealised models of the abstraction process discussed above. Sfard presents an interesting paradox: in order to develop a structural concept, pupils need to interiorize, condense and reify. Reification is the leap from seeing an entity as a lower level process to seeing it as an object. However, in order to see it as an object, we must have some experience of interiorizing that object at a higher level. But until we can see it as an object, interiorization at that higher level carries little meaning. Or, to put it more succinctly, "the lower level reification and the higher level interiorization are prerequisite for each other" (Sfard 1991, p. 31).

This problem has interesting similarities to Vygotsky's theory of the Zone of Proximal Development in which he identifies the "distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers" (Vygotsky 1978, p. 86) as the zone within which children should be working to achieve maximum potential. Children working at their actual developmental level remain neither stretched nor stimulated and in fact Vygotsky goes on to say that "the only 'good learning' is that which is in advance of development" (ibid, p. 89).

It follows that in the abstraction process, and more particularly at the stage within the abstraction process at which pupils are having to re-present a process as a manipulable object, it is necessary for them to practise higher level manipulation of symbols without necessarily understanding the abstraction required to produce these symbols. In other words, pupils must use algebraic symbols before they truly understand them. And this would seem to bring us back to the problem of pupils being asked to use other people's abstractions of a mathematical concept without having first gained the experience of making their own abstractions.

The key to this paradox lies in Vygotsky's use of the phrase, "under adult guidance or in collaboration with more capable peers." Learning is a social rather than an individual process and "human learning presupposes a specific social nature and a process by which children grow into the intellectual life of those around them" (ibid, p. 88). Through imitation of those around them, children are able to internalise concepts which lie within their Zone of Proximal Development, the emphasis of control shifting gradually from teacher to learner.

The vital role in this learning model is that of the teacher whose job it is to lead and coax and then to allow pupils to express ideas for themselves. Wood, Bruner and Ross (1979) see the teacher's role as one of building scaffolding which offers the learner some degree of assistance in solving a problem and allowing work at the pupil's level of proximal development. This assistance may take a number of different forms: "the adult could act as a memory bank for the pupil, could direct the pupil's attention, or could motivate and encourage the pupil to keep going[9]" (Sutherland 1993a, p. 105). So, pupils can abstract ideas from concepts with which they are already familiar, provided the "cognitive demands of expressing a general rule" (Mason 1989, p. 7) are met by the support of a teacher or some external agency. This support is gradually faded as the participation of the learner increases.

The idea of scaffolding has been extended to computational settings in a form described as *webbing* (Noss and Hoyles 1996, p. 107). In making this extension, a number of limitations in the scaffolding model have to be addressed: the model carries with it connotations of an externally erected structure (whereas I have earlier stated the importance of capturing "experience of the inner world from the inside"); it suggests the idea of a bounded territory; and the concept of fading in a computer environment implies that the support offered by the computer would eventually be replaced by something else. Consequently, addressing these issues, webbing differs from scaffolding through incorporation of the following features:

> i) it is under the learner's control;
> ii) it is available to signal possible user paths rather than point towards a unique, directed goal;
> iii) the structure of local support available at any time is a product of the learners' current understandings as well as the understandings built by others into it;
> iv) the global support structure understood by the user at any time emerges from connections which are forged in use by the user.
>
> (Noss and Hoyles 1996, p. 108)

Webbing in fact *extends* the idea of scaffolding. Noss and Hoyles describe the following important features of webbing:

[9] For further discussion see also the Role of the Researcher in Chapter Four, § 4.4.2.

i) pupils must build their own structures taking what is important from the ambient pedagogical setting, rather than receiving what is given;

ii) webbing is domain contingent;

iii) webbing cedes control of the support structure to the learner.

The crux of the webbing idea is that "the computational medium structures and is structured by the student's emerging mathematical ideas" (ibid, p. 108). Pupils' own construction of meaning lies at the heart of the webbing concept.

Now, if we accept a view of the construction of meaning as being a personal, subjective act based on individual experience (although often within a social setting), I suggest that such a view carries with it further implications relating to the *context* in which these experiences are encountered. If meaning is specific to one person and moreover specific to that person's collection of experiences, does it not follow that meaning may also be specific to the context in which that construction takes place?

Nunes et al (1993) describe how untrained "street mathematicians" (such as farmers, fishermen and carpenters) will frequently develop their own methods of performing the calculations necessary for their work (which may include ratio and proportionality); "school mathematicians" - students who have learnt formal methods of calculation - frequently have difficulty performing similar problems using the methods they have learnt. The context in which the street mathematicians construct their meanings would appear to contain elements which help in the concretisation of the mathematical concepts in question in a way that the school mathematicians' context does not. Whether or not the mathematics is concretised appears to depend upon the context in which it is encountered.

Interestingly, although Nunes shows that the same streets mathematicians were able to apply their working methods to new problems within the same environment, it does not necessarily follow that the notion of concrete as I have defined it here, is transferable to other unrelated environments. Rather, if concreteness is an individual and unique construction formed through particular and unique experiences, it follows that concretisation is specific only to those experiences and that specific situation: what is concrete in one context may not be concrete in another. In other words, concreteness is domain specific.

This idea of concrete concepts being situated resonates closely with two important theories from the literature: theorems-in-action and situated abstractions. In mathematical behaviour, pupils often "choose the right thing to do without being able to mention the reasons for it" (Vergnaud 1990, p. 20). Observation of such behaviour suggests the existence of Theorems-in-Action, "those properties of relationships grasped and used by the student in problem-solving situations, though of course that does not necessarily mean that they are capable of making them explicit or justifying them" (Vergnaud 1981, p. 11). Thus the development by pupils of a theorem-in-action implies that they can operate with relational invariants for specific values of variables. (To this definition, Balacheff (1987) adds the conditions that pupils should take into account the validity conditions of the theorem; and that they should have an expectation of the effect of an action which is matched by the actual effect. If these two conditions are not met, Balacheff suggests in preference the term "Rule of Action".)

Theorems-in-action, however, imply no explicit expression of mathematical relationships (or of attempts to justify them). Once such an understanding develops, they begin to lose their "action" limitation. As an intermediary stage where pupils demonstrate an ability to operate beyond the specific mathematical experience and a conscious understanding of generalised relationships, Noss and Hoyles (1996, p. 122) describe "situated abstraction". This is not so much an object as the "process of abstracting in situ", as "(re)thinking-in-progress" (ibid). To perform the process of situated abstraction, pupils "constructively generate mathematical ideas which are articulated in terms of the medium of construction" (Hoyles and Noss 1993, p. 84). The process is characterised by the following criteria:

i) they entirely encapsulate a mathematical relationship;

ii) they are bound up by their setting;

iii) they are mediated by the technology in which they are situated and its associated language;

iv) they lack universality, particularly that element of unmediated abstraction present in a mathematical discourse;

v) they articulate a general relationship structured by the environment in which it is expressed;

vi) they are constructed by a learner who may have no access to the semantics and syntax of general mathematical language, but who is given the opportunity by the computational environment in which he is working.

To sum up, I wish to embrace a theory of the construction of meaning based on the concretising of mathematical concepts through the creation of a rich set of connections between objects, underpinned by children's own experience. This incorporates a view of the concretisation of ideas as firstly a *subjective* process whereby what is concrete to one need not be concrete to another, and secondly a *situated* process whereby what is concrete in one situation need not necessarily be concrete in another.

2.3.4 Creating Windows on Meaning: Microworlds and Convivial Tools

I want now to consider how best to obtain a picture of children making connections between objects, for it is in such a picture that we might hope to find some pointers towards how children construct meaning. As Wilensky says:

> *The pivotal point on which the determination of concreteness turns is not some intensive examination of the object, but rather an examination of the modes of interaction and the models which the person uses to understand the object.*
> (Wilensky 1993, p. 198)

In order to understand the internal construction of meaning, the internal creation of connections and "modes of interaction" which pupils appear to carry out, we need a context in which they are *able* to carry this out and a context in which it is possible to *observe* them carrying out these processes. We need appropriate external tools through the use of which something of these internal processes may be revealed. Noss and Hoyles suggest that use of computers for programming purposes fulfils this function, that the computer can act as "a window onto children's thinking and to offer a means by which children can express themselves mathematically" (Noss and Hoyles 1996, p. 71).

I want to use the idea of a *microworld* as the basis for exploration of pupils' construction of mathematical meaning. The term microworld was first used by "artificial intelligence workers to describe a small, coherent domain of objects and activities implemented in the form of a computer program and corresponding to an interesting part of the real world" (Weir 1987 p.12). Since then the term has evolved in meaning[10] but here I adopt the definition put forward by Noss and Hoyles (op cit p. 65). They suggest that the idea of a microworld "involves an intention to develop an open and investigative stance" to mathematical enquiry. Within this stance learning is regarded as "a consequence of breakdowns - incidents where predicted outcomes are not experienced" and that consequently in order to develop a microworld it is necessary to predict where such breakdowns may occur. Thus at the core of any microworld there is "a model of a knowledge domain to be investigated by interaction with the software". They continue: "Exploration is necessarily constrained but in ways designed to promote learning; knowledge is not simplified, it is recognised as

[10] See Noss and Hoyles (1996 p. 63 - 65) for discussion of the changing meaning of the term *microworld*.

complex, interrelated and evolving in action". An important feature of the system is its extensibility - "the extent to which the elements of the microworld can be combined, recombined and extended to form new elements".

Four important components of a microworld as defined by Noss and Hoyles are:

i) the **pupil** component: the existing understandings and partial conceptions which the child brings to the learning situation;
ii) the **technical** component: the software or programming language and a set of tools which provides the representational system for understanding a mathematical structure or a conceptual field;
iii) the **pedagogical** component: all the didactical interventions that take place during the programming activity; and
iv) the **contextual** component: the social setting of the activities.

To this may be added the **mediating role** of the computer (Noss and Hoyles 1996).

Now, given the idea of a microworld in which pupils may carry out mathematical activities and explore mathematical ideas, the next question from a research angle is how to *observe* pupils' work within a microworld.

Noss and Hoyles (op cit p. 54) describe two very different kinds of software which have appeared in mathematics education circles over the past years. One kind, which has been highly prevalent in schools and has done little for the cause of introducing computers into the classroom, simply replicates the sort of thing which is already familiar from school textbooks but in an animated form: the emphasis is on the computer teaching the pupil a mathematical process. The other kind points "outward towards new, more learnable mathematics; towards a redefinition of what school mathematics might become and who might become involved in it" (ibid). The activity of programming belongs to the latter kind because it involves the programmer in expressing and articulating the mathematical relationships involved: "it is in the process of articulation that a learner can create mathematics and simultaneously reveal this act of creation to an observer" (ibid).

It is not a quality of the computer itself which somehow determines how it is used:

> *The problem with asking questions regarding 'the effect' of 'the computer' is that such questions presume that the computer itself can somehow directly affect thinking and learning, that the computer, solely by virtue of its being a computer, can change the way people think and learn.*
>
> (Falbel 1993, p. 29)

The crucial question is that of *how* the computer is used. In activities such as programming, use of the computer may create meanings for its user. Such use of a tool has been described by Illich as *convivial*:

> *Tools foster conviviality to the extent to which they can be used, by anybody, as often or as seldom as desired, for the accomplishment of a purpose chosen by the user. ... They allow the user to express his meaning in action.*
>
> (Illich 1973, pp. 22-23)

It is important to emphasise that there is nothing in the computer - or indeed in any tool which may be described as convivial - that is *inherently* convivial: conviviality is both subjective and context dependent. Noss and Hoyles make two important observations about convivial tools:

> *First tools are cultural objects. Tools are not passive, they are active elements of the culture into which they are inserted. Second, the extent to which a tool is convivial is determined by the relationship of user to tool, not by any ontological characteristic of the tool itself.*
>
> (Noss and Hoyles, op cit, p. 58)

Thus a microworld, which has at its heart activities centred on programming, not only provides a rich environment for pupils to *create* mathematics (much as Papert describes in *Mindstorms* - Papert 1980) through the use of a tool which has a strong potential for conviviality, but also provides a means through which an observer may *observe* this creation of mathematics.

2.4 CONCLUSION

The three research aims, stated in Chapter One, were as follows:

i) to investigate ways in which children expressed generality using the programming medium;
ii) to investigate connections between different expressions of generality made in different modalities and their role in children's creation of mathematical meaning;
iii) to investigate the role of connections between expressions of generality in children's attempts to justify their constructed generalisations.

The literature offers a clear focus for investigation of these aims, centering on pupils' concretising of mathematical objects through the development of representations and interconnections between representations of those objects. By looking at how pupils concretise abstract objects it may be possible to throw some light on their construction of meaning. Any light that is thrown must be seen within the limitations of the subjectivity of meaning and context discussed in this chapter.

Three concepts, discussed in this chapter, emerge as central to the theoretical basis of the thesis:

i) *concretion* (Wilensky 1993) - objects becoming concrete by virtue of the learner's relationship with that object;
ii) *webbing* (Noss and Hoyles 1996) - a structure which learners can draw upon and reconstruct for support; and

iii) *situated abstraction* (Noss and Hoyles 1996) - the way in which learners draw on the webbing of a setting to construct mathematical ideas and the way in which that webbing shapes the expression of ideas.

Computer programming is proposed as providing a convivial tool, which pupils may use in order to create their own meaning for the mathematical relationships with which they are working; moreover, carefully constructed computational microworlds would appear to provide not only a setting in which pupils may express their emergent mathematical concepts, but also a window through which it may be possible to observe children's construction of meaning.

In Chapter Three I look at the literature on proof and proving processes as a window onto generalisation and justification. I describe a dichotomised slant to research which gives rise to the apparent existence of a rift between various types of proof with a consequent focus on children's failure to carry out proof. I contrast this with a discussion of research into proof which focuses on what children *are* able to achieve, building design issues and foci for research on the basis of this research.

CHAPTER THREE: WINDOWS ON GENERALISATION

3.1 INTRODUCTION

If we want to look at how pupils express formal arguments, we need first to place them in learning situations which equip them with something to express. By placing them in situations which are centred around generalising from their own mathematical experiences, the justification and proof[11] of these generalisations can then provide just such an experience of expressing formal arguments. Thus proof, in a broad sense which includes justification, can provide an important window onto generalisation. A review of the literature on proof provides the focus for this chapter.

This chapter focuses on two very different approaches which emerge from a review of the literature. A significant part of the literature focuses on a dichotomised view of proof, leading to descriptions and analysis of the way students fail to progress in proof and proving processes. Another section of the literature concentrates on what students *are* able to achieve and looks at ways forward from these successes. I shall look first at the various dichotomies which feature in research on proof and suggest the consequences of such views. Then I shall look at what students are able to achieve and describe ways in which my own research is shaped - in terms of design issues and foci for research - by these findings. To begin, however, I look briefly at definitions and functions of proof.

[11] I use this word loosely for the time being.

The word "proof" carries a variety of preconceptions both within and without the mathematical world. As Tall (1989, p. 28) observes, proof can imply "beyond reasonable doubt" to a judge and jury; "occurring with a certain probability" to a statistician; and can be the result of empirical investigation to a scientist. The term "proof" within the mathematical world also embodies a variety of concepts. There are at least three senses in which the term is used: it is used as a *verification* or *justification* of the truth of a proposition; it is used in the sense of an *illumination* conveying an insight into why a proposition is true; and it is used to refer to the *systematisation* of results, that is the organisation of results into a deductive system of axioms, major concepts and theorems, and minor results derived from these (Bell 1976, p. 24). Its usual mathematical meaning is the third of these definitions: a rigorous statement in formal mathematical language derived from axioms or other already-proven statements.

However, in relation to mathematics, it is, perhaps, useful to think of "proof" in terms of *proving processes*, which are concerned with understanding, and *proof* itself, which is concerned with communication. In fact the French literature (e.g. Balacheff 1988) makes this distinction in its use of the terms *preuve* (i.e. the proving process) and *démonstration* (i.e. the proof product). Although preuve is not strictly speaking mathematical proof in the sense used by the Mathematical Community, it is in fact discussed as a type of proof within mathematics education literature, for example by Balacheff in his definition of proof types (Balacheff 1988). It is in this broader sense, encompassing both mathematical proof and proving processes such as the justification of generality, that the term proof is used here.

It is often said that proof lies at the heart of mathematics, but why is it perceived as being so important? Tall sees proof as important for two reasons:

> *i) (Local) Based on explicit hypotheses, a proof shows that certain consequences follow logically;*

> *ii) (Global) Such logical consequences themselves can be used as "relay results" (Hadamard 1945) to build up mathematical theories.*
>
> (Tall 1995, p. 27)

These reasons place emphasis on the verification and systematisation of results: two aspects rightly associated with formal proof (Bell op cit). However, de Villiers suggests that the functions of proof are considerably more numerous and varied:

> i) verification (concerned with the **truth** of a statement);
> ii) explanation (providing insight into **why** it is true);
> iii) discovery (the discovery or invention of **new** results);
> iv) systematisation (the **organisation** of various results into a deductive system of axioms, major concepts and theorems);
> v) intellectual challenge (the **self-realisation/fulfilment** derived from constructing a proof); and
> vi) communication (the **negotiation of meaning** and **transmission** of mathematical knowledge).
>
> (de Villiers 1995, p. 155)

3.2 PROOF DICHOTOMIES

Much of the literature relating to proof is characterised by dichotomies, perhaps the most significant of which is an apparent rift between types of mathematical activity common (if not in fact predominant) in British (and other) schools, and the construction of formal mathematical proofs. On the one hand school curricula advocate approaches to algebraic generalisation and proof which place emphasis on observation, interpretation and generalisation of physical and visual patterns leading to empirical investigation, justification and proofs which are reliant on empirical data and are thus mathematically unacceptable.

The traditional pattern of mathematical proof as the term is used at undergraduate level and above, is what has been described (Leron 1985a, p. 7) as Linear Proof. The initial proof of lemmas from axioms and previously proven theorems leads on to a complete proof of the desired theorem. This process may continue through many layers so that the proof of other theorems in a similar manner leads in turn to the proof of a larger result. Talking about his proof of Fermat's Last Theorem[12], Andrew Wiles says:

> *You work your way around the room and you begin to feel objects here and there and suddenly the light goes on and you can see everything clearly. Then you move into the next room.*

Wiles's analogy of groping around a darkened room is apt: when a group of lemmas has been proved and the light has illuminated one stage in the overall proof, it is time to move to the next layer which is still shrouded in darkness.

Traditional mathematical proofs belong to the second category, yet most of the mathematics taught at school in the United Kingdom prior to 'A' level or even undergraduate courses belongs to the first category. (Proofs of Euclidean Geometry, formerly taught in this country to 14 year olds and still on many national syllabuses, such as those of France, belong to the second group). In fact the image promoted by the U.K. National Curriculum (see Chapter 1) and consequently by commercial texts used in schools, supports and encourages empirical approaches up to the age of sixteen, and relegates formal proof to the periphery of examination papers in the years immediately prior to university entrance. The universities, on the other hand, call for formal proof methods to appear again on the school syllabuses (LMS 1995).

It is not only in terms of United Kingdom "school" and "university" approaches that

[12] *The Guardian*, 8th April 1995.

the literature is presented as a dichotomy. For example, Balacheff makes a distinction between two distinct types of proof: the *pragmatic*, i.e. those which have recourse to actions; and the *conceptual*, i.e. those which do not involve action but instead depend on the formulation of the properties in question and the relationships between them (Balacheff 1988, p. 217).

Balacheff further subdivides pragmatic proofs into Naive Empiricism, Crucial Experiments and Generic Examples:

Naive Empiricism

Asserting the truth of a result after verifying several cases. It is characterised by a fatalism and impatience and as a form is resistant to generalisation.

The Crucial Experiment

A case is looked upon as that which will decide whether or not a rule is general, i.e. "if it works in the following case, it will always work".

The important difference between these two classes of proof is that in progressing from the former to the latter, the pupil is moving from "truth asserted on the basis of a statement of fact to one of an assertion based on reasons" (ibid, p. 228). A similar break exists between the Generic Example, which Balacheff sees as a "transitional stage in moving from pragmatic to conceptual proofs" (ibid, p. 229) and the Thought Experiment which he categorises as conceptual proof:

The Generic Example

The reasons for the truth of an assertion are made explicit by means of operations or transformations on an object chosen not in its own right but as a characteristic representative of its class. Problems may well be presented here by the difficulty on the part of pupils to express mathematical relationships either in natural language or in formal algebraic language.[13]

[13] It is worth pointing out that a generic example is unique to an individual at any one point in time: it is not possible to talk about something or to teach it as "a generic example", only to consider it as an example used for its generic properties by a particular pupil at given point in a given task.

The Thought Experiment

This does not invoke particular situations; instead pupils must make decontextualised, complex cognitive and linguistic constructions. They must be able to accept an abstraction as an object to be operated upon in order to construct a proof. Frequently the necessary removal of particularities from a statement may fail to conserve previously established relationships.

In addition Balacheff describes a further type of conceptual proof:

Calculations on Statements

These have nothing to do with experience: they are intellectual constructions based on more-or-less formalised, more-or-less explicit theories of the ideas in question in the solution of the problem. Proofs are the result of inferential calculations on statements and rely on definitions or explicit characteristic properties.

Tall (1995) places proof into three categories: enactive, visual and manipulative. These correspond closely to Bruner's three modes of learning : enactive, iconic and symbolic (Bruner 1966). Enactive proof requires physical movement to demonstrate the relationships required by the proof. "Such a proof invariably involves either specific examples, or specific examples seen as *prototypes* of a class of examples" (Tall op cit, p. 31). It is thus a pragmatic form of proof which may or may not incorporate the use of generic examples. Visual proof is largely dependent upon identifying examples as being generic: seeing the general in a particular iconic representation. The vital role of the teacher in such cases is often involved with re-focusing the learner's attention by means of visual rearrangement.

Manipulative proof is concerned with the manipulation of algebraic symbols and does not involve physical movement or iconic representation.

There are important differences between Tall's and Balacheff's categorisations. For example, the difference between enactive proofs and pragmatic proofs is that between physical movement and action. The "physical movement" of an enactive proof involves some sort of rearrangement to show a relationship; the "action" of a pragmatic proof may involve the testing of numerical examples of a general rule. Thus both visual and manipulative proofs may contain elements of pragmatism.

In fact Tall's and Balacheff's categorisations serve different purposes: Tall's categorisation is one of types of proof and is useful in identifying the range of proofs which might be used in a particular mathematical situation; Balacheff's categorisation is one of the processes of proof and can be used to identify strategies used by pupils in particular situations.

Bell (1976) also places emphasis on proof processes, but chooses to categorise proofs not on the basis of whether or not they involve action but on whether or not the pupil shows any evidence of deduction. Thus Bell identifies *empirical* responses to proof, which are based on action and therefore pragmatic, and *deductive* responses to proof, which although largely conceptual in nature, can be linked with action, provided they also contain some deductive qualities.

Bell makes a detailed subdivision of his two broad categories of proof response. He divides Empirical responses as follows:

i) Failure to generate examples or to comply with given conditions.

ii) Extrapolation: truth of general statement inferred from a subset of the relevant cases; any apparent reasons are either assertions that the conditions have been complied with, or added fragments. The basis of the inference is clearly empirical.

iii) Non-systematic: finds some of the required cases, no complete subsets, ignores the requirements to find all.

iv) Partially systematic: finds some partially complete subsets of cases; has some awareness of the requirement to find all.

v) Systematic: finds at least some complete subsets of cases, is clearly attempting to find all.

vi) Check of full finite set of cases.

(Bell op cit, p. 28)

Coe and Ruthven discover, in a group of pupils in their last year of school before going to university, a view of proof as a kind of "'supercheck' - a more rigorous and comprehensive analogue of testing individual cases to see if a rule works" (Coe and Ruthven 1994, p. 49). Bell's six types of empirical proof response might be seen as a continuum from the inadequate - a "subcheck", if you like - to the fully comprehensive - a "supercheck".

Bell sub-divides Deductive proof responses as follows:

i) Non-dependence: one or more examples correctly worked, but not used to test the general statement; lack of awareness of connection between conclusion details of the data,

ii) Dependence: attempts to make deductive link between data and conclusion, but fails to achieve any higher category.

iii) Relevant (general restatement): makes no analysis of the situation, mentions no relevant aspects beyond what is actually in the data, but re-presents the situation as a whole, in general terms, as if aware that a deductive connection exists but unable to expose it.

iv) Relevant (collateral details): makes some analysis of the situation, mentions relevant aspects which could form part of a proof, possibly identifies different subclasses but fails to build them into a connected argument; is fragmentary.

v) Connected (incomplete): has a connected argument with explanatory quality, but is incomplete.

vi) Connected ('side-step'): failing only because it appeals to facts or principles which are no more generally agreed than the proposition itself.

vii) Complete Explanation: derives the conclusion by a connected argument from the data and from generally agreed facts and principles.

(Bell op cit, p. 28)

In fact the first two of Bell's subdivisions seem to be typified by failure to carry out deduction rather than by its use. Otherwise these subdivisions appear to move from generic example to conceptual proof. Coe and Ruthven (op cit, p. 44) choose to divide deductive proofs into two parts: *weak deductive* (some attempt to suggest an underlying reason) and *strong deductive* (an attempt at a clear, logical argument with reasonably explicit links between the starting assumptions and a clearly defined solution).

Thus, a significant proportion of authors present proof in terms of a dichotomy, be that between "school" and "university" approaches or between proof types of their own definition, such as pragmatic and conceptual, or empirical and deductive.

3.3 CONSEQUENCES OF PROOF DICHOTOMIES

Broad categories of proof presented in the ways described in Section 3.2 lead to a polarisation of proof types. The consequence of this polarisation is that a large body of the literature focuses on the product of the proving process and on what children fail to achieve rather than on the process itself and what they might or can achieve given more effective learning environments.

The literature is characterised by gloomy litanies of failure. For example, Coe and Ruthven (1994) suggest that those pupils educated since the publication of the Cockcroft Report (Cockcroft 1982) tend to use predominantly informal proof strategies to explain rules or generalisations. This, they imply, has major consequences for the perception of proof held by students embarking on university courses:

> i) few students were concerned to explain why rules or patterns occurred, or to locate them within some wider mathematical system;
> ii) students' proof strategies were primarily and predominantly empirical, with a very low incidence of strategies that could be described as deductive;
> iii) students' primary concern was to validate conjectured rules and patterns which, for most, took the form of testing them against a few examples;
> iv) in tackling starting points, students tended to employ routinely a standard repertoire of investigational techniques from their textbook, primarily designed for the analysis of numeric data;
> v) numeric data generated from a starting point quickly became the object of investigation, with the original situation being abandoned.
>
> Coe and Ruthven (op cit, p. 52)

From the United Kingdom National Curriculum "programme of study" (described in Chapter One) it is easy to see why students who have followed it hold these perceptions of proof.

Even the valuable aspects of school-based investigation and empiricism are not, however, all they seem. Such work may become routinised, often to fulfil the demands of time and assessment: "Through prototypical examples of investigative tasks, and standardising templates for their conduct and reporting, ... codes [emergent through mathematics course work] seek to make the process of enquiry more teachable and test-able" (ibid, p. 52).

Turning now to students who follow "traditional" courses solely in formal, deductive proof, all is far from well here. Moore (1994) looked at first year undergraduate students who were following a formally taught course on mathematical proof (with little or no experience of deductive proof at school). He found the following:

> i) the students did not know the definitions, that is they were unable to state the definitions;
> ii) they had little intuitive understanding of the concepts;
> iii) their concept images were inadequate for doing the proofs;
> iv) they were unable, or unwilling, to generate and use their own examples;
> v) they did not know how to use definitions to obtain the overall structure of the proofs;
> vi) they were unable to use mathematical language and notation;
> vii) they did not know how to begin proofs.
>
> (Moore 1994, p. 251)

To this list Lerman et al (1993) add:

> i) failure to follow a chain of reasoning;
> ii) misunderstanding a mathematical concept;
> iii) misunderstanding the language used;
>
> (Lerman et al 1993, p. 259)

where they define "misunderstanding" as "failing to interpret in the same sense as the teacher".

Students also do not appear to relate other mathematical experiences they have encountered with the mathematics of proof. In an analysis of students solving mathematical problems, Schoenfeld (1985) finds that students do not use their mathematical knowledge because they do not perceive it as useful. Additionally, he finds that students are all too often unaware of the purposes of proof, perceiving proof as unnecessary argumentation about the intuitively obvious, or the ritual verification of something which they are told at the outset is true.

Correct use of mathematical language and notation has long been a stumbling block in the understanding of proof. Mathematicians aim for clarity and simplicity in their proofs and in fact clarity and simplicity of argument are two principal factors in the acceptance of a proof (Dreyfus and Eisenberg 1986), but ironically, it is in pursuit of these qualities that mathematicians frequently and deliberately omit large amounts of details in the exposition of formal proofs (Bell 1976). This is detail which is unnecessary for a mathematician and which would in fact only serve to cloud the issue by making it difficult to detect the relevant argument for extraneous detail. Indeed, the *suppression* of detail is necessary for acceptance by the mathematical community (Tall 1995). For the uninitiated, however, use of conventions such as "similarly" to describe some operation parallel to one previously carried out, can only serve to exclude. At a higher level, mathematicians conversant in the language of one subset of mathematics are excluded by the conventions and language of another subset (Thurston 1995).

There is a second form of "exclusion through omission" and this lies in the assumption of certain "standard" results. The results, of course are standard to writers of the proof, and because they are writing for a particular audience for whom the results are also standard, they may safely assume that they and their audience share a common bank of references. Yet immediately by doing so they have excluded the wider audience. This exclusion of the uninitiated may be seen in the familiar sequence of proofs of Euclidean theorems, once standard fare in British schools. Knowledge is assumed of all theorems in the sequence once proven, so that anyone “entering the sequence” part way through must accept previous theorems without proof or be excluded from the proof currently under consideration.

An interesting example of this is described by John Aubrey in his *Brief Life* of the seventeenth century mathematician, Thomas Hobbes[14]:

---

[14] Lawson-Dick 1992, p. 150.

> *He was 40 years old before he looked on Geometry; which happened accidentally. Being in a Gentleman's Library, Euclid's Elements lay open, and 'twas the 47 El. libri I. He read the Proposition. By G ---, sayd he (he would now and then sweare an emphaticall Oath by way of emphasis) this is impossible! So he reads the Demonstration of it, which referred him back to such a Proposition; which proposition he read. That referred him back to another, which he also read. Et sic deinceps [and so on] that at last he was demonstratively convinced of that trueth. This made him in love with Geometry.*

Whilst many pupil might share Hobbes's initial reaction, it would seem safe to say that for the majority, the teaching of formal proof alone will not help them reach the same conclusion.

Additionally many students appear not to see any need for proof and regard methods such as proof by induction as a "confidence trick" (Anderson 1995, p. 48). Students even find difficulty identifying what is and what is not a proof: Finlow-Bates describes the confusion experienced by first year mathematics undergraduates when shown in the first case an informal proof followed by some examples and in another case some examples followed by an informal proof. The students described the examples as the proof of the mathematical statement and were more convinced by these than by the informal proof, which they variously referred to as "examples", "comments", "summary" and "notes" (Finlow Bates 1994, p. 349). A tendency for students to mistrust a formal proof to the extent of checking it against empirical evidence, has been observed elsewhere (Vinner 1983, Porteous 1990, Simpson 1995, Chazan 1993).

Van Dormolen (1977) suggests that encountering proof initially through formal methods may carry the expectation of pupils working at a cognitive level which is too high for them to have any understanding of the concepts involved:

> *I am convinced that what in the past was called giving a proof was in fact for many students a forced operating in the second level*[15] *of thinking, whereas they had not as yet reached or sufficiently explored the first.*
>
> (van Dormolen 1977, p. 33)

Simpson (1995, p. 40) paints a picture of incompatibility in his description of two Routes to Proof: the *Natural Route* which he defines as "proof through reasoning" and which leads the student along a path progressing from Exploration to Proof via Discovery, Finding Patterns, Explaining, Justifying and Formalising; and the *Alien Route* which he defines as "proof through logic" which involves progression along a path leading from Sums to Proof via Drill, Testing, Techniques, Calculations and Predicate / Propositional Calculus. He suggests that

> *as teachers increasingly provide experiences from the "proof through reasoning" route - investigative work, modelling, explorations - and move away from what mathematicians might see as the foundation for the traditional "proof through logic" route, we may get a broken route: the foundations of a "proof through reasoning" route provided by a school education, followed by the "top end" of a "proof through logic" route.*
>
> (Simpson 1995, p. 42)

Simpson's routes seem over-simplistic and it is difficult to agree with the clear cut division between reasoning and logic that he describes, but in fact the literature goes further. From presenting these two diverse and apparently incompatible routes, some authors compartmentalise children's thinking processes into two equally diverse types. For students, the important development in moving from empirically based proof to deductively based proof is an awareness of the difference between "asserting truth on the basis of empirical evidence, and proving it is true by logical deduction from known facts" (Tall 1989, p. 29). This is a major change in thinking for students. The empirical gathering of data is a normal, everyday procedure: it is a Constructivist

[15] Van Dormolen is referring to van Hiele's three levels of thinking (see van Hiele *Begrip en inzicht*, Purmend, 1973)

view of the way in which children learn. In an empirical context the most important consideration is to produce an efficient solution rather than to be rigorous (Balacheff 1988); the emphasis in deductive proofs is conversely on rigour and the production of knowledge.

The implication here is that empirical and logical ways of reasoning are not only different but, at least at an early age, virtually incompatible: formal proof is completely outside the main stream of behaviour, so for a purely empirical thinker, formal proof has no meaning (Fischbein 1982). (Cobb (1986) refers to "self-generated mathematics" and "academic mathematics".)

But is it as simple as this? Such a viewpoint leaves many questions unanswered. Do children in fact gather data empirically more readily than they reason logically? Under certain conditions might not children at least as readily reason logically as use empirical methods to gather data?

3.4 BEYOND SIMPLE DICHOTOMIES

The previous section looked at the result of research where the existence of a dichotomy appeared to be at the forefront of the researcher's mind and where, consequently, the focus was very much on what children *fail* to achieve. In this section I shall look at research where the emphasis is placed on what children *are* able to achieve. I begin with a look at generalising activities and how these might be extended and enriched. Then I describe research which examines means to promote proof within the school curriculum and to make it accessible.

3.4.1 Generalisation and the link with proving

In the introduction to this chapter I described how I intended reviewing the literature on proof in order to open a window onto generalisation and justification. In this section I explore the link between generalisation, justification and proving.

Mason suggests that "generalisation is the life-blood, the heart of mathematics" (Mason 1996, p. 74) and from a look at the published texts at least, generalisation would appear to occupy an important position in the UK primary school curriculum (see, for example, Blair et al 1976, or more recently Hatch et al 1992). There are at least three forms of generalisation (Harel and Tall 1991 - quoted in Mason 1996, p. 81):

i) expansive: extend the range of an existing schema, assimilating the new particular in an old generality;

ii) reconstructive: rebuilding, accommodating the old generality to subsume the new particular;

iii) disjunctive: adjoins the new particular as an extra case.

Two pedagogic considerations would appear to play a part in the effective teaching of generalisation in the classroom: the use of exploratory approaches which Bell (1996, p. 184) describes within a problem-based course of algebra; and the willingness and ability to reflect on the relationship between particularity and generality: "it is essential that you pause and consider ... the need to read implicit generality in what often appears to be very particular assertions that are being put forward" (Mason 1996, p. 68). I explore this relationship in Section 3.5.6.

Generalisation occurs in many forms - not all mathematical - and appears to be well within the reach of comparatively young children (i.e. eleven and younger). Nemirovsky (1996a) describes work carried out with 9 and 10 year olds' generalisations of mathematical behaviour shown in graphs, such as the growth of a plant. He suggests that "it is through the construction of mathematical narratives that children's experience with change, that is, with the different ways in which change occurs, becomes the subject of mathematical generalisation" (Nemirovsky op cit, p. 217). Boero found that, under the guidance of teacher-led discussions, 11 year olds could make "significant observations" concerning the generality and conditionality of statements (Boero 1995, p. 10). In fact he hypothesises that it is possible to get children of this age "constructively involved in approaching statements of theorems in an adequate educational context, depending on the role of the teacher for what concerns the approach to the essential, epistemological characteristics of statements (conditionality and generality)" (ibid p. 2).

All too often, school generalisation activities come to a dead end, however. "Many students do come to value general and explanatory arguments through ... investigative activities, but this fertile ground is not exploited to introduce mathematical proof and face students with the challenge of setting out a mathematical argument in a coherent and logical manner" (Healy and Hoyles 1998, p. 6). This is a matter of major concern (and one shared by, for example, Anderson 1995, p. 48), but under the right conditions there seems to be every reason why pupils' activities could be extended to introduce mathematical proof. Two ways forward would appear to be through consideration of verification and of justification.

A number of authors find that the statement of generality and the verification of that statement are intimately bound up together. For example Mason suggests that: "the thrust of expressing generality is that students take over more and more responsibility for recognising and expressing generality, and verifying the associated conjectures" (Mason 1996, p. 77). Radford makes a similar point: "generalisation as a didactic device cannot avoid the problem of validity" (Radford 1996, p. 109). Boero goes further and links precision of generality with verification of validity: "the problem of the greater precision in the formulation of the statement was intimately weaved [sic] with the problem of the verification of its validity, even if gradually, through numerical examples, they started to realise that the property might be true" (Boero 1995, p. 10). Sutherland, on the other hand, suggests that a feature of computer environments which supports pupils in the development of an algebraic approach to problem solving in mathematics is the fact that "the computer frees pupils from the process activity of evaluating an expression, thus enabling them to focus more on the structural aspects of a situation" (Sutherland 1995). This opens up the question of the role of verification in construction: to what extent are the two processes inter-dependent and to what extent may a computer environment change the balance of such an inter-dependence?

If investigative generalisation activities are to be "extended" to make a link with mathematical proof, some element of justification of the generality must be introduced: "the logical base underlying generalisation is that of justifying the conclusion. It is a proof-process, which moves from empirical knowledge to abstract knowledge that is beyond the empirical scope" (Radford 1996, p. 111). Porteous suggests that this in fact should be central to the curriculum:

> *The proof types used by children are naturally informal, but it would be a mistake to devalue them because of this. Proving, in the sense of explaining generalities, should be part of the normal activity of mathematics.*
>
> (Porteous 1990, p. 597)

The key to justification would appear to lie in the relationship between the *means of expression* of generality and the *means of expression* of justification of that generality. Bell cites an example where a pupil expressed relationships between numbers in a calendar square algebraically and then justified the relationship she had defined. "She uses the formation and manipulation of her expressions not only to express her conjecture but also to prove or, in this case, to disprove its generality" (Bell 1996, p. 177).

Boero also makes this link. Through consideration of what he defines as relational and procedural statements of generality and the different approaches to proof to which these statements lead, he suggests that "in the case of a statement produced (or assumed) by a student, its proving process may naturally evolve from it as a 'textual development' of the statement itself" (Boero 1995, p. 2). For example, he found that from the relational statement *a number and the number immediately after have no common divisors except for the number 1*, students found all the divisors of a number and its next and verified that there were no common divisors except for 1. Although this did not help the students with justification of the statement, it did help them to get an opinion of validity. What is important, however, is that the attempt was "influenced by the formulation of the attempt that they had considered" (ibid, p. 11).

An approach to proof which begins with generalisation and verification might begin at a young age. Almeida (1995, p. 61) outlines three essentially internal proof processes encountered by the early learner:

> i) establishing the truth of an assertion on intuitive grounds (e.g. the intuitive belief that the shortest distance between two points is a straight line);
> ii) seeing why a statement is true (e.g. a visual insight into why a geo-strip triangle is always rigid obtained through construction);
> iii) inductive proof (e.g. proving that all objects fall at the same rate on the basis of empirical evidence).

At this early stage the pupils are merely convincing themselves: there is no explanatory function to these early proving processes and no means of or need for communication. But importantly, Almeida suggests that "giving recognition to prototypical proof practices is a necessary condition to enable the learner to become aware of (and then later work with) non-empirical proof" (ibid, p. 67).

One approach to the introduction of justification to investigative generalisation activities might be through comparison of equivalent expressions of generality. Bell suggests that a development to the situations he describes might be for a teacher to record pupils' expressions, using examples of the same relation expressed in different ways: "discussion of this could provide the beginnings of the awareness of the equivalencies that form the basis of manipulations" (Bell 1996, p. 176). Perhaps this area might be even more fertile than he suggests: might not the discussion of equivalencies also form the basis for justification of generality?

3.4.2 Contexts for proving: Accessibility of Mathematical Content and Links with the Curriculum

Central to all the examples quoted in § 3.5.1 is the accessibility of the mathematics being used to the pupil who is expected to use it. All too often proof is taught using statements involving mathematical concepts with which the learner is still struggling. It is like asking a learner driver to manoeuvre an articulated lorry into an awkwardly situated parking space: ask the driver to practise reversing a small, familiar motor car into the same space and he or she will soon acquire familiarity with the difficulties of the manoeuvre. In time he or she will be so familiar with the process of reversing into the space that when general management of the articulated lorry is mastered, performing the difficult manoeuvre will become a realisable possibility.

Almeida (1995, p. 70) finds that "given the exhortation to justify an *accessible*[16] conjecture (such as the sum of two odd numbers is always an even number), the majority of learners seemed willing to offer an explanation even if some of these may have been clumsy or insufficiently understood (for example, "an odd number is a minus and an even number a plus and two minuses give a plus")."

Also, if we want proof to be considered by the pupil as a natural and essential part of mathematics - as a central element of mathematical culture - it is necessary for examples of proof to be closely related to the curriculum: in this way there is a greater likelihood that, when a curriculum topic is revisited, the possible application of a proof within that topic may present itself spontaneously. If proof is introduced through isolated mathematical examples, then proof itself may be perceived as an activity in isolation. Similarly, the mathematical situation should contain an apparent

[16] My italics.

need for proof: "explanation must be necessary, technically and sociologically; if the result is obvious or ... generally accepted, only a recipe is obtained" (Brousseau 1988, p. 3).

If we accept the importance of creating a link with the curriculum, it is difficult to know where ideas such as Austin's example of an addition sum - acting as a generality ripe for proof - fit in (Austin 1995, p. 75). He proposes giving a pupil the sum

$$\begin{array}{r} M\ A\ T\ H\ S \\ +\ P\ R\ O\ O\ F \\ \hline T\ H\ E\ M\ E \\ \hline \end{array}$$

where different letters stand for different digits. Pupil are required to "find all the answers and give a proof to show the list of answers is correct." This cannot be solved easily by an exhaustive proof so, it is hoped, pupils will begin to look for patterns and from these make generalisations. What message does such an activity give to pupils? Rather than relating to their curriculum or involving them in the creation of a generalised expression, it would appear to reinforce the idea of proving (and indeed algebraic expression) as a pointless game.

In considering the place of the curriculum in relation to proof and processes of proof, it seems possible that the speed with which it is necessary to cover the curriculum in British schools combined with a tendency to "compartmentalise" the curriculum, rarely allows for the depth of study required to tackle proof at anything beyond a superficial level.

3.4.3 Prerequisites to the Use of Formal Manipulation

Austin (1995, p. 76) suggests that prior experience of using the relevant mathematics is necessary as a prerequisite to formal proof. He states that the "first two steps on what should be the pupil's ascent[17] to proof are omitted" and proposes that Step 1 should show pupils what a proof is all about and allow them to appreciate

> *a) that a mathematical proof is an argument just like other everyday arguments - it involves everyday, sensible thinking,*
> *and*
> *b) that a mathematical proof is an argument unlike all other arguments - it involves an absolute certainty that other arguments lack.*

(It is difficult to see how both these statements can be true: the latter would seem to be what proof is about; therefore the former is not.) For this step he uses as an example the following problem, which he claims may be solved by means of a simple exhaustive proof:

> *Write the numbers 1, 2, 3, 4, in a row so that the sum of any two numbers that are next to each other in the row is at least five. Find all the answers and give a proof to show your list of answers is correct.*

Leron suggests that "most non-trivial proofs pivot round an *act of construction*" (Leron 1985b, p. 323) and in Step 2 Austin advocates giving pupils the experience, the feel, of constructing a proof for themselves. It allows pupils to appreciate

> *a) that constructing a mathematical proof is a creative act,*
> *and*
> *b) this creative act involves hard and careful work.*

Austin uses the example quoted in § 3.5.1 of the addition sum

[17] Choice of the word "ascent" indicates that Austin makes some interesting assumptions.

M A T H S + P R O O F = T H E M E

where different letters stand for different digits, and challenges the pupil to "find all the answers and give a proof to show the list of answers is correct." This cannot be solved easily by an exhaustive proof so, it is hoped, pupils will begin to look for patterns and from these make generalisations.

The act of construction is also highlighted by Goldenberg and Hazzan (1995, p. 109) who suggest that construction need not be interpreted as physical manipulation of concrete objects but may include construction through manipulation of symbols including those which constitute a computer programming language. This ties in closely with the idea of the subjectivity of concreteness described in Chapter Two[18], so that what is a dauntingly "abstract" concept for one person may lie sufficiently within another's experience and have a sufficiently rich set of representations to be seen and treated by that person as "concrete". What it does not include is the "deconstructive" proof by contradiction, which contains no element of showing how a mathematical object is made.

However, since Austin bases his examples on empirical methods of proof (since these are most accessible to the pupils they are aimed at), the question remains how to make a transition to conceptual proof methods. Examples where exhaustive proof becomes clearly impractical, serve to highlight the necessity for conceptual proof, as do examples which might reasonably be expected to be generic for many pupils, but they fail to equip the pupil with language or notation for expression, justification or formalisation.

Another interesting example, this time involving very young children using proof in

[18] See Chapter 2, § 2.3.2.

the early stages of mathematics, is described by Davis and Maher. In a study involving children finding the number of arrangements of two colours in towers of three, four and five bricks, they observe that organisation of their solutions enabled the children to develop "proof by cases" (i.e. proof by means of empirical testing) and "proof by mathematical induction" (Davis and Maher 1993, p. 111). No use of algebraic or any other type of notation is made as all the work is carried out verbally, so again the exercise appears to come to a dead end: the children have not been equipped with a language in which to express their ideas in any way more formalised than that required for the empirical work in which they have been engaged.

The same goes for the Action Proofs described by Sémadéni. She defines them as an "idealised, simplified version of a recommended way in which children can convince themselves of the validity of a statement" (Sémadéni 1984, p. 32). The emphasis here is simply on the children convincing themselves: there is no question of the social acceptability of a proof or even of Tall's (1989, p. 30) "convince yourself, convince a friend, convince an enemy" idea, let alone any element of formal deduction. An action proof of a statement S proceeds as follows:

> *Choose a special case of S. The case should be generic, not too complicated, and not too simple. Choose an enactive and/or iconic representation of the case or a paradigmatic example. Perform certain concrete, physical actions so as to verify the statement in the given case.*
>
> *Choose other examples, keeping the general schema permanent but varying the constants involved. In each case verify the statement, trying to use the same method as in 1.*
>
> *When you no longer need physical actions, continue performing them mentally until you are convinced that you know how to do the same for many other examples.*
>
> *Try to determine the class of cases for which this method works.*
>
> (Sémadéni op cit, p. 32)

This leaves a great many questions unanswered, not least of which is how one ensures that the special case of S is generic (1) for the pupil concerned. Also the question of language is not broached at all: in what language is the statement verified (2), being convinced expressed (3) and determination of the class of cases expressed (4). Sémadéni has provided an interesting general framework in which to explore empiricism but it is difficult to know where one might progress from here.

Brousseau suggests that "the passage from natural thought to the use of logical thought like that which regulates logical reasoning, is accompanied by construction, rejection and the use of different methods of proof: rhetoric, pragmatic, semantic or syntactic" (Brousseau 1988, p. 12). The place of construction stands out again, as it does in Tall (1989) where he suggests that pupils need to have the informal experience of *using* ideas as a prerequisite to manipulating formal statements. Pupils should begin by producing convincing arguments in practical situations so that they may progress to making logical deductions in more general situations (Tall op cit, p. 31). Empiricism and deduction are combined within the same example: the students' empirical work is seen not as an end in itself but as the beginning of deductive reasoning.

As an example, Tall suggests getting pupils to make statements of the form "If I know something, then I know something else" in cases where the second statement may hold where the first does not, for example (ibid):

> *if* $x + 1 = 3$, *then* $(x + 1)(x + 1) = 9$;
> *but if* $(x + 1)(x + 1) = 9$, $(x + 1)$ *need not be 3.*

Use of general statements such as

> *If* $x > 5$, *then* $x > 3$

which are likely to produce universal agreement, provide some interesting possibilities. By labelling '$x > 5$' as P and '$x > 3$' as Q, it is clear that whenever P is true, Q is also true. If, however, $x = 4$, P is false and Q is true; and in the case $x = 2$, neither P nor Q are true. Tall suggests that reasoning of this type with easily understood concepts provides a good introduction to formal mathematical proof.

It seems that whilst a certain amount of research has been carried out into what might be labelled "introductory activities" to formal methods of proof, not enough thought has been given to how these activities will progress into formal proof. The advantage of Tall's work over the examples by Austin discussed above is that the preliminary mathematical experiences which Tall describes contain the mathematical language and notation needed for the ensuing logical reasoning and deduction. It is where the language and notation of the formal proof differs significantly from that of the practical mathematical experience that students are unable to make the transition from the pragmatic to the conceptual.

3.4.4 Particularity as a Window onto Generality

Seeing the general through the particular is a theme which runs through the literature. Careful use of particular examples which are representative of their class (generic examples) appears to enable pupils to see beyond the particular to the general. As Mason puts it, "particularity has to be used as a window to be looked through, rather than as a wall to be looked at" (Mason 1996, p. 70). This use of particularity as a window onto generality is important in the expression of both generality and of proof. So whilst Mason suggests that "generalisation is usually taken to be an inductively empirical activity in which one accumulates many examples and detects patterns" (Mason 1996, p. 77), he also cites Davydov (1990) who refers to "mastery of a single example that, with appropriate stressing and consequent ignoring of special features, serves as a generic example from which the general can be read" (Mason, op cit, p. 77).

Looking at the relationship between generality and particularity in terms of the introduction of algebraic formalism, Bell suggests that by introducing a letter to stand initially for a specific value, "we avoid the incomprehension that often arises when manipulative exercises are given in the absence of the meaning-giving formulation and interpretation stages" (Bell 1996, p. 179). This thinking may also be seen in the use of spreadsheets (Sutherland 1992). Thus, through consideration of the particular, pupils seem able to approach the general.

Where problems exist it is often because there are two types of expression: if these two languages could be brought closer together or even better incorporated into one multi-purpose language, then the problems might disappear.

> *While most of the work in maths education seeks to improve the learning and communication of mathematics by supplementing or bypassing mathematical formalism, it is also important to consider at the same time how the formalism itself might be improved to become more communicative of the ideas behind it.*
>
> Leron (1985b, p. 321)

Balacheff suggests that "the movement to conceptual proofs lies essentially in taking account of the generic quality of those situations previously envisaged" (Balacheff 1988, p. 217). He goes on to say that such a movement "requires an altered position: the speaker must distance herself from the action and the processes of solution of the problem". This is partly at odds with the view of abstraction discussed in Chapter Two, where the emphasis is placed on creation of ever-richer meanings rather than a drawing away from one meaning towards another. However, the importance of the generic quality of a situation remains.

A number of authors recognise the importance of using generic examples. In an attempt to make a link between the "natural" empirical process and the "artificial" deductive process (hints here of Simpson's "routes to proof" - Simpson 1995), Tall (1979) and Steiner (1976) offer the concept of a *generic proof* - one which "works at the examples level but is generic in that the examples chosen are typical of the whole class of examples and hence the proof is generalisable" (Alibert and Thomas 1991). As an example, Tall gives a generic proof for the irrationality of $\sqrt{\frac{5}{8}}$:

> *We will show that if we start with any rational p/q and square it, then the result p^2/q^2 cannot be 5/8.*
>
> *On squaring any integer n, the number of times that any prime factor appears in the factorisation of n is doubled in the prime factorisation of n^2, so each prime factor occurs an even number of times in n^2. (For instance, if $n = 12 = 2^2 \times 3$, then $12^2 = 2^4 \times 3^2$.)*
>
> *In the fraction p^2/q^2, factorise p^2 and q^2 into primes and cancel common factors where possible. Each factor will either cancel exactly or we are left with an even number of appearances of that factor in the numerator or denominator of the fraction. The fraction p^2/q^2 can never be simplified to 5/8 for the latter is $5/2^3$, which has an odd number of 5's in the numerator (and an odd number of 2's in the denominator).*
>
> (Tall 1979, p. 204)

Tall found that students showed a significant preference for the generic proof over a "standard" proof by contradiction on the grounds of ease of understanding and lack of confusion. The generic proof combines empirical examples with deductive reasoning.

Movshovitz-Hadar attempts to make a link between empiricism and deduction, but starts from another angle: in her *generic-example assisted proof* she appends an example "small enough to serve as a concrete example yet large enough to be considered a non-specific representation of the general case" (Movshovitz-Hadar 1988, p. 17). She uses as an example the following theorem:

> *For any nxn matrix, n a positive integer, such that the rows form arithmetic progressions with the same common difference d, then the sum of any n elements, no two of which are in the same row or column, is invariant.*

The example she uses to assist in the proof is that of an 8x8 matrix. Because the proof for the 8x8 case "is kind of 'transparent', one can see the general proof through it because nothing specific to the 8x8 case enters the proof" (ibid, p. 19). Clearly what such an approach loses in generality it gains in explanatory power.

As in the earlier discussion of generalisation and justification[19], the form and context of the means of expression pupils are required to use are all important. Mason introduces an interesting concept: generality in action (Mason 1996, p. 81), that is children acting as if they perceive generality, although unable to express it. Although this is an observation of pupils' mathematical behaviour and an interesting perspective on how pupils use particularity to express generality, it also highlights the vital importance of pupils having at their fingertips the means with which to express and formalise their emergent concept of generality and indeed justification. Bell describes an activity called Line Patterns in which students are asked to generalise relationships between sets of numbers. He observes that "they are generalised numbers, but have concrete support. The cognitive demand is therefore low. But these letters are being used in symbolic algebraic statements of generalisation. The students are using algebraic language in a situation where it forms a natural means of communication" (Bell 1996, p. 175). What is needed is surely learning situations which incorporate the means to express the relationships contained within them, i.e. an auto-expressive language.

Leading on from this point, it is interesting to look more closely at the use of language in generalising and proving activities. Formal mathematical proof is expressed in formal mathematical language. In fact, a paradox surrounds the use of language: informal proof methods use informal language and thus fail to equip pupils with a mathematical language in which to express their ideas - formalisation is then often presented as a "tack-on" activity at the end of empirical work; pupils working with formal proof methods which have formal language at their heart, are often either

[19] § 3.4.1

unable to use the language and notation (Moore 1994) or misunderstand the language used (Lerman et al 1993). In formal proof the language for many - but not all - pupils is purely a code in which the ritual of the proof is carried out and not a means of expression.

Goldenberg and Hazzan see a tension between the role of language and the nature of students' uses of language. The role of language is to bring "focus and abstraction to actions and processes" (Goldenberg and Hazzan 1995, p. 112), so in the translation of physical constructions into words, students free their construction from specific details. The use of different types of language affects students' concepts of the development of proof:

> *Informal language captures meanings in ways that students find familiar and in tune with their intuitions. Formal language helps make description more precise, but also helps strip it down to those features that are most likely to be central to reasoning and proof. Differences between these two styles of expression may represent hurdles that students encounter when they present formal arguments.*
>
> (Ibid, p. 112)

Balacheff sees a tension between "the language of the everyday, whose main support is natural language" and a language which "must become a tool for logical deductions and not just a means of communication" (Balacheff 1988, p. 217). He sees a necessity to move from one to the other in order to progress from the pragmatic to the conceptual.

Given this tension between formal and informal language, it is particularly interesting to look at what lies between the two, in neither the domain of informal languages nor of formal mathematical languages. In observing a pupil called Chloe, Noss says: "her proof lives in a world somewhere between natural language and mathematical formalism; a world which allows abstractions to be formulated and explored by those who have not yet completed their journey into the rigorous world of mathematics, but which carries them a little nearer to it." (Noss 1997, p. 43) Thus looking at how pupils use the particular to construct and interpret the general provides a useful window onto the relationship between formal and informal language.

If didactic situations can be devised where the language forms an integral part of the proving process, then pupils will be equipped with a means of formal expression for their ideas: articulation by means of the working language of the activity. Expression of generality and its proof might then itself become an integral part of an activity rather than an extra activity tacked on after the investigation is at an end. Such an approach removes the exclusive and off-putting burden of translating a result into another language. Noss and Hoyles (1996, p. 69) term this use of language "auto-expression": a computational world which "contains the elements of a language to talk about itself". Sutherland identifies Logo as an effective example of an auto-expressive language in the context of generalisation and justification: "in Logo there does not have to be a gap between pupils' informal methods and the formal representation of this method. Within the Logo context, pupils, through interacting with the computer and discussion with their peer, are able to develop their intuitive understanding of pattern and structure to the point where they can make a generalisation and formalise this generalisation in Logo" (Sutherland 1989, p. 341).

The issue of auto-expression, of what is an auto-expressive language, is at the core of the whole issue of language for the construction, expression and justification of proof. Goldenberg and Hazzan (1995a, p. 123) talk about "successive refinements" to the language of construction "as the complexity and sophistication of the argument requires them". This is not true auto-expression, but a suggestion that situations can be created where there is one language existing in some kind of informal version - a dialect if you like - for the construction phase of a mathematical situation, and also in a more formalised dialect which is better suited to formal deductive proof. Because these dialects are part of the same language, "successive refinements" can mould the first into the second. This view needs caution: not all languages are accessible to refinement and if the dialects of construction and discussion are too far removed from one another, the hurdle of translation rears its head again. Professor Higgins successfully trained Liza Doolittle's accent and enunciation, but initially she let him down through unfamiliarity with the idioms of her new type of speech: "informal" language can be refined but may remain unidiomatic if it is too far removed from "formal" language.

Refinement also carries undesirable overtones of progressing from one outlook to another rather than the ideal integration of construction and justification. Given the crucial roles of language, a truly auto-expressive language would be the single most important factor in linking empiricism and deduction.

3.4.5 Acceptance

Fischbein (1982) identifies three types of conviction held by pupils as to whether or not a statement is true. These convictions may be based upon formal argumentation of the type traditionally found in axiomatic mathematical proof; they may be the results of practical investigation - findings which support a conclusion previously reached; and they may be of an intuitive, intrinsic type, imposed by the structure of the situation. This last category Fischbein calls "cognitive belief" (Fischbein op cit, p. 11). In the acceptance of a proof by students, the first type of proof carries little prominence. In fact rigorous, formal proofs of a theorem are a relatively insignificant factor in its acceptance (Hanna 1991). Lerman, Finlow-Bates and Morgan (1993) find that clarity, usefulness, consistency and the extent to which proofs are convincing and easily understood are all considered of greater importance by students than logical argument and rigour.

So something needs to be done to make formal proof appear relevant and necessary to students. One step is for them to "accept" the proof. Mathematicians accept a new theorem if they understand it, if it is significant and consistent with accepted mathematical results, if there is a convincing and familiar mathematical argument for it, and if its author has a reliable reputation as a mathematician (Hanna op cit). When new theorems appear, only those which appear significant will be scrutinised by the mathematical community and have their proofs checked; others, however sophisticated their proof, will not be tested if they appear to be insignificant theorems. "A proof becomes a proof after the social act of 'accepting it as a proof'. This is true of mathematics as it is of physics, linguistics and biology" (Manin 1977). In fact Volmink (1988) suggests that greater emphasis should be placed on the social criteria for acceptance of mathematical truth at the expense of purely formal criteria.

Now, this social nature of proof is of importance not only in the greater Mathematical Community, but also within what one might term the closed mathematical community of a school classroom or lecture theatre. The idea of acceptance is discussed by Alibert and Thomas (1991) in the context of scientific debate. In order to make students see proof as a necessary part of the scientific process, students' scientific statements are written on the blackboard, thus placing them in the public domain. These statements are then discussed by fellow students who take a vote on their validity: those which are validated become theorems; those which are not validated are preserved as false statements. This approach to proof coincides with Davis's (1986) view of proof as "an argument needed to validate a statement; a debating forum" and Alibert's (1988) view of the importance of creating a scientific debate providing opportunity to discuss arguments made by a proof.

The idea of acceptance has wider connotations. In the same way that a proof which satisfies, say, a scientist would not be acceptable to a mathematician as a mathematical proof, what is acceptable as a proof in one branch of mathematics in one particular situation may not be acceptable in another mathematical situation: the social construction and acceptance of proof is dependent upon and shaped by the society which constructs it. This opens up the question of the relationship between proofs in differing mathematical situations.

3.4.6 Explanatory Quality

We have seen that formal deductive proofs are likely to contain little in the way of explanation as to why a result is true. For example, a proof by induction *begins* with a result and involves the manipulation of the symbols in which that result is expressed; proof by contradiction begins with an "opposite" result and through symbolic manipulation demonstrates its impossibility. In neither proof is there any element of explanation as to *why* the result being proved is true. Proofs which contain an explanation in addition to merely proving a theorem (but which also contain the mathematical rigour of a traditional proof) might be used in an attempt to give meaning for students (Hanna 1989b, Wittman and Mueller, 1988). Thus proofs which explain play two important roles:

> *An explanatory proof illuminates at the same time that it dispels doubt. When such a proof is used successfully in the classroom, the students acquire not only a "knowledge* ***that****", a knowledge of* ***what*** *is true, but also "knowledge* ***why****", an understanding of* ***why*** *it is true.*
>
> (Hanna 1995b, p. 134)

As an example of an explanatory proof, Hanna (1989, p. 48) compares two proofs of the theorem that the sum of the first n integers S_n is given by

$$n\,(n + 1)/2$$

Proof by induction demonstrates the truth of the statement but fails to show why it is true:

i) If $n = 1$, $S_n = 1$

ii) If $n = k + 1$, $S_{k+1} = k\,(k + 1)/2 + k + 1$

$$= (k^2 + k + 2k + 2)/2$$
$$= (k + 1)\,(k + 2)/2$$

The result is true for $n = 1$ *and* $n = k + 1$*, so it is true for every n.*

Gauß's proof, on the other hand:

$$S_n = 1 + 2 + \ldots + n,$$

and

$$S_n = n + (n-1) + \ldots + 1,$$

so

$$2S_n = (n+1) + (n+1) + \ldots + (n+1),$$

$$= n(n+1)$$

and

$$S_n = n(n+1)/2$$

is explanatory "because it uses the property of symmetry (of the different representations of the sum) to show why the statement is true. It makes reference to the property of symmetry and it is evident from the proof that the results depend on this property" (Hanna 1989b, p. 48).

A proof only explains when it "reveals and makes use of the mathematical ideas which motivate it" (ibid, p. 47). Proof by induction, for example, makes use of a standard mathematical "trick" which may be applied to a wide variety of results, regardless of the mathematical ideas contained within them. Steiner (1978) talks about a "characteristic property" of an explanatory proof:

> *An explanatory proof makes reference to a characterising property of an entity or structure mentioned in the theorem, such that from the proof it is evident that the results depend on the property. It must be evident, that is, that if we substitute in the proof a different object of the same domain, the theorem collapses; more, we should be able to see as we vary the object how the theorem changes in response.*
>
> (Steiner op cit, p. 143)

Chazan (1993, p. 383) suggests that "the explanatory aspect of proofs is a useful starting point for a discussion of the value of deductive proofs."

3.4.7 Providing an Overview

The problem students encounter in traditional mathematical proofs is partly one of confusion and a lack of understanding of what the proof is trying to achieve. The holistic conveyance of the ideas in a proof, making them intelligible and convincing, is of greater importance than the formal adequacy of the logic (Hanna 1991). One method of proof proposed as being easily understood is the Structural Method described by Leron (1985), that is one where the proof is constructed in levels proceeding from the top down. Each level is autonomous and contains one idea of the proof, an idea which enables one to gain an overall view of that subsection. For example, Level 1 gives a global view of the proof; an 'elevator' allows for informal discussion (which may itself be a part of the proof) and takes one down to the next level of the proof. Any necessary lemmas are assumed and proved *at the end of the overall proof.* This is the opposite of the traditional (bottom-up) proof methods where all lemmas are proved initially and only when this has been done can the proof progress. In a Structural view of proof, emphasis is instead placed on giving the pupil an overview of what is trying to be achieved. Leron outlines the format for structured proofs as follows:

> *Introduce the pivot as a system of constraints (i.e. define it implicitly by postulating its properties);*
>
> *Without actually solving the system, use the pivot as introduced in step one to derive the conclusion of the theorem;*
>
> *Discuss heuristically the solution of the system to find how the pivot might be constructed;*
>
> *(Recursion step) Solve the system repeating steps 1-4 if necessary. That is construct (or prove the existence of) the pivot, then prove that it satisfies the postulated properties. If some of the sub-proofs are themselves complicated, introduce sub-pivots and repeat the four-step procedure.*
>
> (Leron op cit, p. 12)

For the deductive thinker, this approach provides purpose and direction whilst never leaving the realms of a purely deductive proof. It perhaps removes the sense of groping around in a darkened room described by Andrew Wiles.

3.4.8 Expectations

Healy and Hoyles, in the results of a survey into high-attaining Year 10 students' views of and competencies in mathematical proof (Healy and Hoyles 1998) offer some pointers towards extending students' experiences of proof within the school curriculum. Although they find that students are "unable to distinguish and describe mathematical properties relevant to a proof and use deductive reasoning in their arguments" they also say that the majority "also recognise that a valid proof is general and accord high status to formally-presented arguments" (ibid, p. 6).

Healy and Hoyles go on to suggest that more challenge and more attention to proving could enhance performance and that this might be achieved through explicit efforts to engage students with proof "while discussing with them the idea of proof at a meta-level, in terms of its meaning, generality and purposes. This would involve finding ways of balancing the need to produce a coherent and logical argument with the need to provide one that explains, communicates and convinces" (ibid, p. 7). They conclude: "our evidence suggests that students could well respond positively to the challenge of attempting more challenging and rigorous proofs alongside informal argumentation" (ibid, p. 7).

3.5 CONCLUSION

This chapter has presented two contrasting views of the literature on proof: firstly from a dichotomised viewpoint where emphasis was placed on the product of proof teaching; and secondly from a viewpoint where the processes of proof formed the focus. From this latter viewpoint, a number of important points emerge as being conducive to proving and generalising in school mathematics. Messages from research into proof have important implications for how, from a wider perspective, the processes of generalisation, verification and justification are taught in school mathematics. These implications are summarised below and presented partly as design issues for the study and partly as foci for research:

3.5.1 Design Issues

Design issues fall into three groups: those affecting the participants in the study (i.e. researcher and pupils) and their respective roles, those affecting the mathematical activities around which the study was centred, and those affecting the computational medium in which they were required to work. These issues may be summarised as follows:

A computational medium:

- for construction and expression which incorporates a visual dimension;
- which contains a link between the process of construction and a means of verification of that construction;
- which contains a common language for construction and for expression; and
- which facilitates the use of generic examples.

Mathematical activities:

- in which the need to generalise and justify arises as an integral part of the activity;
- which incorporate manipulation of "concrete" objects as an integral part; and
- the content of which is accessible and related to the curriculum.

Pupil/researcher relationship:

- group organisation which facilitates the ability to make arguments in practical situations; and
- group organisation which facilitates negotiation of acceptance of mathematical

statements by the "mathematical community"[20].

These design issues are considered in Chapter Four where the design of the programming microworld is discussed.

3.5.2 Foci for Research

Research foci may be summarised as follows:

i) The relationship between construction, expression, generalisation and justification;

ii) The connection between children's visualisation of mathematical relationships (within the specific learning situation) and their expression of generality and justification;

iii) The role children's acts of construction play in effecting the abstraction of mathematical relationships;

iv) Ways in which children's exploration of particularity leads to generality;

v) Ways in which children link ideas at varying levels of abstraction and ways in which they use these relationships;

vi) How children use "movement" between levels of abstraction;

vii) The relationship between the language of construction, the language of expression and the children's own natural language within a specific learning situation;

viii) The role of language in the construction and expression of ideas.

These foci are addressed in Chapter Six and used, in conjunction with initial findings from the preliminary phases of the study, to frame questions for research.

[20] The "community" is made up of the pupils and the researcher, with the researcher occupying an essential but clearly defined role within that community. This idea is discussed in § 4.4.

CHAPTER FOUR: DESIGN OF THE PROGRAMMING MICROWORLD

4.1 INTRODUCTION

In Chapters Two and Three I developed a theoretical framework for the study and looked at issues which affect the expression of generality and the justification of generalisations. In this chapter I build on these points in order to put forward a rationale for the design of the microworld and activities used in the study. The design of the various components of the study was iterative in nature, that is rather than any one component being devised, tested in a pilot study and then used as a vehicle for gathering data in a main study, the final design evolved through the first two phases of the study, only then reaching the form it was to take in the final and main part of the study (hence the organisation of the study into phases). Thus, for example, the various forms of researcher intervention were identified as a result of analysis of the initial phases. To present a clear picture of the rationale behind the design, the final design is presented in this chapter; the iterations through which this final design was reached may be traced in Chapters Five and Six.

In Chapters Two and Three the specification of Specific Learning Situations in which to observe pupils' attempts at generalisation and justification was discussed. In order to consider this specification I shall look at the three components of the learning situation and at the relationship between these components. The components comprise the participants in the study (i.e. researcher and pupils) and their respective roles, the mathematical activities around which the study was centred, and the computational medium in which they were required to work.

The criteria for consideration within these three components are reproduced here:

A Computational Medium:

- for construction and expression which incorporates a visual dimension;
- which contains a link between the process of construction and a means of verification of that construction;
- which contains a common language for construction and for expression; and
- which facilitates the use of generic examples.

Mathematical Activities:

- in which the need to generalise and justify arises as an integral part of the activity;
- which incorporate manipulation of "concrete" objects as an integral part; and
- the content of which is accessible and related to the curriculum.

Pupil/Researcher Relationship:

- group organisation which facilitates the ability to make arguments in practical situations; and
- group organisation which facilitates negotiation of acceptance of mathematical statements by the "Mathematical Community"[21].

These criteria are addressed in the sections which follow.

[21] The "community" is made up of the pupils and the researcher, with the researcher occupying an essential but clearly defined role within that community. This idea is discussed in § 4.4.

4.2 DESIGN OF THE COMPUTATIONAL MEDIUM

4.2.1 Introduction

In this section I consider the design of the computational medium through discussion firstly of the mathematical focus of the study. This leads into a description of the rationale behind the setting adopted in the study. Finally I present my reasons for the choice of Logo as the computational language used by the pupils and describe the software developed for use in the research.

4.2.2 The Mathematical Focus

Simple function machines were chosen as the vehicle for investigating pupils' construction of meaning through the statement of generalisation and justification. Such function machines were familiar from the mathematics texts used at the school and would have been encountered first when the pupils were approximately seven years old. Thus pupils would be coming to the tasks with a relatively long history of familiarity with and experience of using function machines, satisfying the criteria that the mathematical activities used should be easily accessible and linked to the pupils' curriculum and that pupils should approach any activities from a basis of their own experience rather than one of school taught algorithms. Given the mathematical experience of the pupils taking part and the degree of contact they would have had with the concept of function machines it was reasonable to assume that this concept was relatively concrete for most of the pupils. Function, and more particularly function machines, may be modelled through a variety of representations: given the right computational medium, pupils could be offered a rich set of models around which they might construct their own understanding of mathematical function.

Function machines were also chosen as being an appropriate means by which to approach both generalisation and justification: verification of the ability of a constructed function machine to model some pre-defined mathematical relationship may be demonstrated by repeated testing of cases, yet additionally, generalised statements may be made based on the algebraic expression of function machines by means of a relatively rudimentary grasp of algebra. The simple concept of function machines is, of course, a tremendously powerful one and so there was no problem in providing through it mathematical experiences which, whilst rooted in their own experience, would challenge the pupils taking part in the study.

4.2.3 The Setting

As a starting point it is interesting to consider Brousseau's (1981) description of three situations which encompass corresponding forms of knowledge functioning:

> **action situations** - requiring an implicit use of knowledge;
> **formulation situations** - requiring the making explicit of concepts;
> **verification situations** - requiring the justification of what has been made explicit.

This model neatly underpins the rationale behind the setting of the mathematical tasks used in the study. Pupils were grouped in pairs, each member of which had his or her own computer to work on. Each task took place within a game-like context, whereby pupils were challenged to construct a procedure from information presented in a written form, working not in collaboration but in competition with a partner. In this stage, which constitutes Brousseau's *action situation*, pupils worked independently of one another on separate computers.

The game context of the setting was designed so as to encourage pupils to ensure (prove[22] to themselves) that their procedure was correct, and to assert (prove to their partner) that their procedure was either the equal of their partner's or better than it. It was hoped that the challenge element contained in the game would provide the motivation for ensuring that pupils would check, to the best of their ability, that their procedure would always work (within the conditions described in the question). This checking process, which I termed *auto-verification*, normally involved testing of cases and so provided useful and relevant experience of empiricism. Auto-verification was in fact an integral part of the construction process: in the "ideal" scenario pupils constructed a procedure and carried out auto-verification, which, if negative, informed the re-construction of the procedure, and if positive, confirmed the validity of the procedure thus completing the first stage of the game.

The next stage of the game required pupils to prove[23] that their partner's procedure did or did not work (again within the conditions described in the question); and further to explain, analyse and justify their own procedure construction. This stage contained two elements: the empirical testing of the partner's procedure, a process which I have termed *altero-verification*; and the analytical discussion of the written procedures, which I termed *justification*. Pupils had the opportunity of producing convincing arguments in practical situations, and in those situations, justification was an integral part of the mathematics and directly connected to the mathematical experiences encountered within the task. Justification might occur spontaneously as a result of the game context of the tasks, or it might be prompted by suitable researcher interventions[24] .

[22] "Prove" used here to indicate empirical verification or verbal justification.

[23] Again used loosely.

[24] See § 4.4.2

An additional stage in the model was illumination: it was expected that the experience of construction and proof would inform those same processes in subsequent, similar tasks. Consideration of this element was not, however, a research aim of the thesis.

Thus, the proposed model contained action, formulation and verification situations (in the sense described above by Brousseau). The design of the learning situation and the choice of computational medium addressed potential problems highlighted by Laborde (1989, p. 35): "The spontaneous formulations of pupils in mathematics often contain implicit information and ambiguities." Use of the programming language forced the explicit, unambiguous statement of mathematical relationships and the game or challenge nature, in which the construction and testing of procedures was embedded, created both the necessity and the desirability of making any procedure as clear as possible (for if the mathematical relationships contained therein were self evident by virtue of the construction of the procedure, then so too was the fact that the procedure would work and that the pupil had won the game). This description resonates with a remark made by Laborde:

> *Two characteristics are used to generate ... a construction [of precise and unambiguous formulations]: the social dimension of the language activity and its finality. The pupil's formulation is aimed at a peer who needs it to carry out a subsequent activity which cannot be carried out without this formulation. The fact that a peer is addressed encourages the pupils to be careful about the quality of their formulation so that their classmate can manage the activity which depends on it.*
>
> (Ibid, p. 35)

The "subsequent activity" in this case was the verification (altero-verification) of the other pupil's procedure, but the relationship between quality of formulation and the peer as addressee had marked similarities with the thinking behind the game situation of the tasks.

The necessity both to construct and verify and to justify meant that there was a sense in which the pupils' work had a dual focus: whilst construction and verification was centred around the computer, the requirement to justify their procedures provided "a forum where these ideas can be discussed and evaluated away from the computer" (Hoyles, Healy and Pozzi 1994, p. 214).

The components of the setting are summarised in Table 4.1.

Stage	Elements	Pupil Activity
Introduction		Pupils receive a stimulus in the form of a game to design a Logo procedure for their partner to use.
Generalisation	Procedure Construction Auto-Verification Researcher Interventions	Pupils attempt to construct the required procedure and ensure its validity
Justification	Altero-Verification	Pupils test each other's procedure to their satisfaction
	Verbal Justification	In discussion, pupils justify and explain their own & criticise their partner's procedure
Illumination		In subsequent, related activities, pupils use newly acquired knowledge and skills

Table 4.1: Components of the Setting

4.2.4 Logo and the Mathematical Concept of Function

The computer can help bridge the gap between formal knowledge and intuitive understanding:

> *It stands "betwixt and between the world of formal systems and physical things: it has the ability to make the abstract concrete"*
>
> (Turkle and Papert 1991, p. 346)

Considerable claims have been made for Logo. For example, Ortiz and MacGregor (1991, p. 47) claim that it provides a "concrete framework" of procedures and visual images which help to create mental imagery for the concept of variable and facilitate the application of the concept to other concepts outside the computer environment (Nelson 1987, Feurzeig 1986); also pupils using Logo retain their acquired algebraic understanding (Nelson 1987) due to the mental imagery of algebraic concepts they have developed (Ortiz and MacGregor 1991). These claims are couched in terms which speak of mappings between external, teacher-imposed representations and also make reference to transfer of understanding to other representations. Whether doing "Logo algebra" provides the same understanding of algebra as doing "paper and pencil algebra" is unclear. Noss suggests that Logo may have a role in forming "primitive conceptions of algebraic notions" which may form part of a pupil's system of algebraic understanding and it is this idea that "Logo algebra" may be one element in the conceptual field[25] of Algebra that is adopted here (Noss 1986, p. 354). Certainly pupils would seem to have to experience variables in a variety of Logo contexts before any transfer of concepts to "traditional" algebra occurs (Sutherland and Hoyles 1986, Sutherland 1987). In fact Roberts, Carter, Davis and Feuerzeig (1989) find little evidence to suggest that transfer of concepts occurs at all.

[25] See Vergnaud (1982)

Logo provides a model of function and variable which is reasonably consistent with mathematical usage (Noss 1986). Research suggests that there are strong links between Logo and "traditional" concepts of variable in the context of functions and function machines. Logo is a functional language, i.e. one which interprets a program as an "operation formed from simpler operations by functional composition" (Klotz 1986, p. 16). In a sense, any Logo procedure may be said to have an output, which may include effects on the screen such as drawing and printing. (Indeed the command FD 100 in turtle geometry has an output in this respect.) A more restricted meaning of output is a Logo object which can be passed from one procedure as an input to another procedure (Leron and Zazkis 1986, Sutherland 1987). Such an output is generated by function machines which closely model "traditional" paper and pencil definitions of a function. For example, a Logo procedure such as

```
to f :x
op :x * 2 + 3
end
```

models the function

$$f(x) = 2x + 3.$$

Logo was enhanced by the creation of TAB, which calculated output values for any input value of a particular function defined in this way.

Simple function machines can be used to introduce ideas associated with formal algebra. Logo imposes a domain of definition by means of error messages which reject certain inputs (Leron and Zazkis 1986) and inverse functions can be generated (see Feurzeig 1986 and Leron and Zazkis 1986). Children's difficulty in accepting lack of closure (Booth 1984a, Collis 1974) and desire to make all expressions "equal to something" (Chalouh and Herscovics 1988, Kieran 1983) might effectively be addressed in a Logo context where they appear to accept lack of closure through use of simple function machines like the above: "Providing pupils with relevant Logo experiences could be an important step in helping them to manipulate "unclosed" expressions in the algebraic domain" (Sutherland 1989, p. 337).

4.2.5 Design of the Software

As described in § 4.2.4, TAB made it possible to tabulate values for a function defined in the way described. For the purposes of this study, this facility was harnessed and placed at pupils' disposal in the form of the *Calculator*, a means of trying single values in a previously constructed procedure (see fig 4.1). Numbers were entered in the IN box and the input and output appear in the OUT box. For example, the function mentioned above (op :x * 2 + 3) given the input 5, outputs

5 13

Using the Calculator, a collection of values could be built up in the output box. This simple context combined the advantages of a clear visual display with the requirement that pupils select and have responsibility for the *values*, the *range* of values and the *number* of values to be used for verification. In the initial phases of the study a more complex program was available which could output a series of values within a range of values for the input and at intervals determined by the pupils (similar to TAB). This was rejected for the last phase of the study as the responsibility for selecting which values to use in verifications was largely removed from the pupils who tended to use the same range of inputs in each successive task[26].

This format had the advantage of enabling pupils to verify their procedures empirically both quickly and easily, thus effecting a close link between the expression of the mathematical relationship defined by the algebraic language in which the procedure was written, and the tabular format of the Calculator output which had close connections with the visual format of function machines as encountered by

[26] See Chapter 5 § 5.2.4 for an account of the evolution of the Calculator.

pupils previously in their school mathematics. Thus the software gave pupils the opportunity to express their ideas both formally (a condition Hoyles, Healy and Pozzi (1994) describe as promoting successful group work) and in an informal, empirical manner.

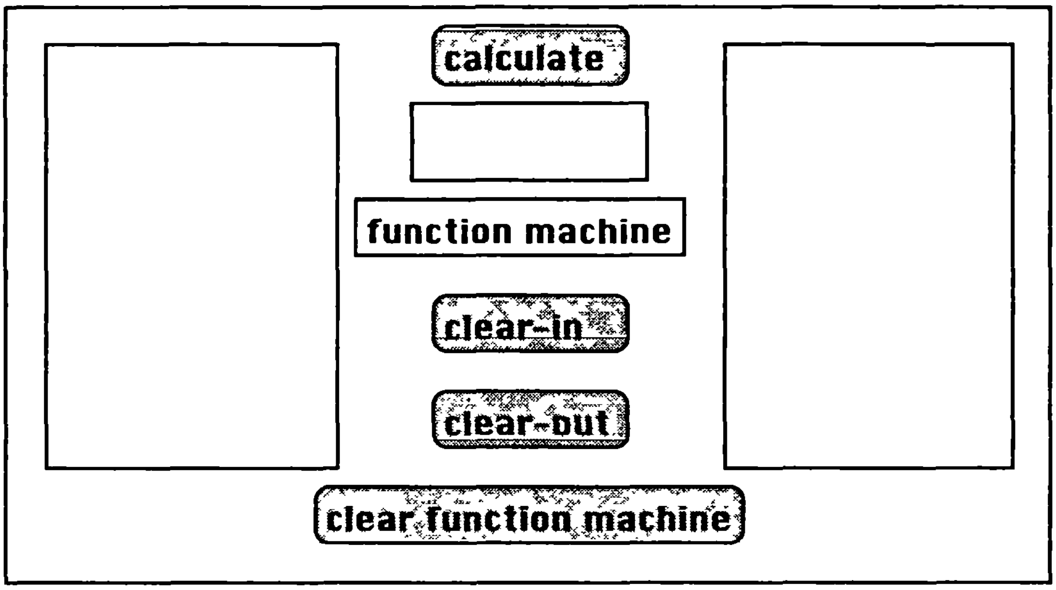


Figure 4.1: The Calculator Software

4.3 THE MATHEMATICAL ACTIVITIES

4.3.1 The Tasks

As with other features described in this chapter, the eventual content and order of the tasks were the subject of much testing and revision, which is described in Chapter Five. Here the final versions of the tasks are described in the order in which they appeared in phase III of the study; the full set of tasks appears in Appendix Two. The rationale behind the choice of function machines is described in § 4.2.2 of this chapter. Each set of tasks was designed to last approximately one hour.

Activity Type One: Function Machines

Since all the tasks in the study were based on function machines, the first group of tasks in the series was designed as an introduction to function machines (a familiar concept) in the context of Logo programming (a largely unfamiliar concept). Pupils were shown how a simple function machine could be expressed in Logo and then asked to write procedures to model function machines presented as in Figure 4.2

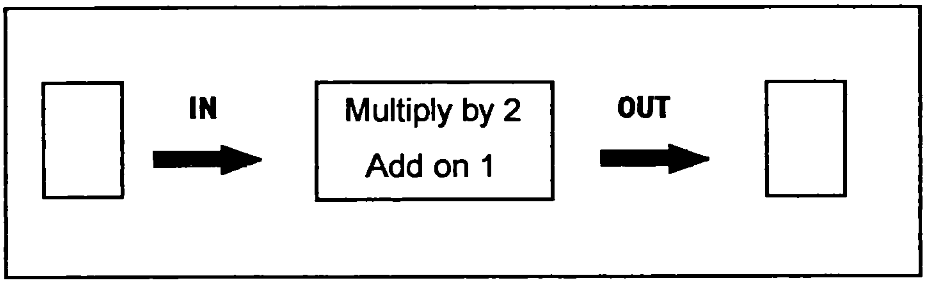


Figure 4.2: Paper Representation of Function Machines

In the subsequent three tasks in the first session, pupils were given a table of values (from 0 to 10 inclusive) and asked to construct a Logo function machine which produced these values, e.g. (Task 1.4):

Input	*Output*
0	3
1	5
2	7
3	9
4	11
5	13
6	15
7	17
8	19
9	21
10	23

The functions (in Logo and *f(x)* notation) used in the first session were as follows:

Task	Logo notation	f(x) notation
1.1	op 2 * :x + 1	$f(x) = 2x + 1$
1.2	op 3 * :x + 2	$f(x) = 3x + 2$
1.3	op 3 * :x + 4	$f(x) = 3x + 4$
1.4	op 2 * :x + 3	$f(x) = 2x + 3$
1.5	op 4 * :x + 1	$f(x) = 4x + 1$
1.6	op 5 * :x + 2	$f(x) = 5x + 2$
Extra	op :x * :x + 2	$f(x) = x^2 + 2$

Table 4.2: Logo notation / f(x) notation comparison for tasks in Session 1

Activity Type Two: Amaze Your Friends

The second collection of tasks switched the focus from constructing function machines on the basis of tables of values to modelling a function machine on the basis of a written description in natural language of a mathematical relationship. This relationship was expressed in terms of a puzzle typified by Task 2.1:

> *Amaze Your Friends ....*
>
> *Pick a number between 1 and 10. Multiply it by 2. Add on the number you first thought of. Divide by 3. Divide by the number you first thought of.*
>
> *The Answer is .... ONE !!!*

Activities such as this clearly extended the idea of a function machine from those in the first session and introduced opportunities for empirical verification and for justification of why the function behaved as it did. Each question was defined within the domain of input $x \in \{1, \ldots 10\}$ in order to avoid division by zero. The expected procedures in the second set of tasks were as follows:

Task	Logo notation	f(x) notation
2.1	op (:x * 2 + :x) / 3 / :x	$f(x) = \dfrac{2x + x}{3x}$
2.2	op (:x * 2 * 2 - 4) / 4 + 1	$f(x) = \dfrac{4x - 4}{4} + 1$
2.3	op ((:x + 2) * 4 - 8) / :x	$f(x) = \dfrac{4(x + 2) - 8}{x}$

Table 4.3: Logo notation / f(x) notation comparison for tasks in Session 2

Activity Type Three: Consecutive Numbers

The tasks in the third session called for interpretation of a number of functions centred round properties of consecutive numbers. For example, Task 3.1 read as follows:

> *Write a Logo procedure which adds the input to the next consecutive number, e.g. if you input 5, it adds 5 and 6 and outputs 11.*

Here it was anticipated that empirical verification would become more difficult because empirical results were not well known to the pupils in the way that they had been in the previous groups of tasks. Thus justification of the validity of a procedure by analytical means became a more important issue in these and later tasks. (The issue of the shifting balance between empirical verification and justification is examined in § 4.3.2). The expected procedures in this group of tasks were as follows:

Task	Logo notation	f(x) notation
3.1	op :x + :x + 1	$f(x) = x + x + 1$
3.2	op :x + :x + :x + 2	$f(x) = x + x + 1 + x + 2$
3.3	op :x * (:x - 1)	$f(x) = x (x - 1)$
Extra	op :x * (:x - 1) * (:x + 1)	$f(x) = x (x - 1) (x + 1)$

Table 4.4: Logo notation / f(x) notation comparison for tasks in Session 3

Activity Type Four: Odds and Evens

The final group of tasks centred around properties of odd and even numbers. These functions were probably the least familiar to the pupils. For example, the first task (Task 4.1) was stated as follows:

> *Write a procedure which will always output an **even** number, whatever number you use as an input.*

Whilst the fact that even numbers are divisible by two would have been familiar to the pupils, asking them to *generate* even (and then, in Task 4.2, odd) numbers in this fashion would have been a new idea for them. The final problems in this session (Tasks 4.3 - 4.5) were the most difficult. They introduced two variables and required pupils to use the knowledge they had acquired in Tasks 4.1 and 4.2 for generating even and odd numbers to produce functions for generating the sums of two even numbers, two odd numbers and finally an even numbers and an odd number.

The expected procedures in the fourth series of tasks were as follows:

Task	Logo notation	f(x) notation
4.1	op :x * 2	$f(x) = 2x$
4.2	op :x * 2 + 1	$f(x) = 2x + 1$
4.3	op (:x * 2) + (:y * 2)	$f(x, y) = 2x + 2y$
4.4	op (:x * 2 + 1) + (:y * 2 + 1)	$f(x, y) = 2x + 1 + 2y + 1$
4.5	op (:x * 2) + (:y * 2 + 1)	$f(x, y) = 2x + 2y + 1$

Table 4.5: Logo notation / f(x) notation comparison for tasks in Session 4

The tasks in the latter stages of the study were designed to engender discussion: in each there were a number of issues which could be raised by the researcher which might draw the pupils beyond simple empirical verification to using the structure of a procedure as a tool for analysing its mathematical behaviour. This important feature is examined in the following section.

4.3.2 The Verification/ Justification Balance

The nature of the tasks created a changing role for verification: a series of situations was created through which it was intended that pupils should feel the need to shift the emphasis successively away from empirical testing towards analysis. The learning situation provided both a framework wherein pupils might carry out empirical testing (i.e. empirical testing controlled by the pupil), and a structured medium through which they were able to justify, to explain and to structure their ideas.

Initial activities (Function Machines: Tasks 1.1 - 1.3) did little more than present opportunities for simple empirical verification, but led into those tasks where exhaustive verification (within the limited domain defined by the question) was possible (Function Machines: Tasks 1.4 - 1.6). Information was presented in a tabular form similar to that which appeared on the computer, so a direct comparison might be made between question and computer output. (It was, of course, necessary only to verify two examples in these linear functions in order to demonstrate that they were correct. Non-linear functions were also included - e.g. the extra task in Function Machines and tasks in the Amaze Your Friends section - although it was not anticipated that the type of function would influence perceptions of what constituted sufficient empirical testing amongst the pupils taking part in the study.)

The second group of activities (Tasks 2.1 - 2.3) also provided the opportunity for exhaustive verification within the domain defined by the question, but here there was no tabular presentation for comparison: individual outputs needed to be checked against mental calculation from the written question details; also the range defined by the domain had to be interpreted from the natural language presentation of the question.

Exhaustive verification was impossible in subsequent tasks, and this, combined with unfamiliarity with the expected outputs, reduced the relevance of empirical testing. For example, in Task 3.3 (a procedure for generating the product of two consecutive numbers), pupils were not instantly familiar with the product of a number and that immediately preceding it, and consequently had no strong expectation of what their procedure would output.

With a reduced relevance on empirical testing as a means of persuasion (remembering still the game-like context of the learning situation and the necessity for pupils to convince one another of the validity of their procedures), another means of persuasion now became necessary, and it was here that the emphasis shifted from verification by empirical means to explanation and justification by consideration of the constructed procedure. This persuasion was necessarily supported by interventions from the researcher: the researcher's role is discussed in the following section. The shifting balance between empirical verification and justification is summarised in Table 4.6

Verification	Task	Justification
Two or three values tested	Function Machines 1.1 - 1.3	No apparent necessity
Exhaustive	Function Machines 1.4 - 1.6	No apparent necessity
Exhaustive; central to construction process	Amaze Your Friends 2.1 - 2.3	Leads on from exhaustive verification; brackets initiate justification
Less relevant as results are not well known	Consecutive Numbers 3.1 - 3.3	Necessary for understanding of 3.3
Relevant, but refined use possible	Odds and Evens 4.1 - 4.2	Simple but convincing
Confusing: results totally unfamiliar	Odds and Evens 4.3 - 4.5	Extension of above, and less confusing than verification

Table 4.6: The Validation/Justification Balance

4.4 THE PUPIL/RESEARCHER RELATIONSHIP

4.4.1 Introduction

Laborde (1989) describes a number of constraints which influence the didactic system. Two of these provide useful points for consideration in describing the relationship between pupils and researcher within the setting:

i) the pupils' concepts, their mode of cognitive development which condition access to new knowledge; and

ii) the teacher-learner asymmetry in relation to the knowledge embedded in teaching situations.

To provide an insight into pupils' existing concepts at the outset of the main part of the study (i.e. Phase III) I devised a pre-task interview. This interview is described in detail in Chapter 6 (§ 6.3.2) and the full questionnaire appears in Appendix One. An indication of the pupils' ability level is given in § 6.3.1 (using standardised tests administered by the school) and where relevant their previous experience of programming using Logo is summarised.

As Laborde suggests, the relationship between researcher and pupils is one of the utmost importance. From the outset, the researcher's role was seen as an active one, with an important interactive part to play throughout the course of the activities. In Chapter Three, reference was made to the "mathematical community" and, in the learning situation described in this study, that community is made up of the pupils and the researcher, with the researcher occupying an essential but clearly defined role within that community. Thus the role of the researcher is an extremely important one and one which I describe in considerable detail in the following section.

4.4.2 The Role of the Researcher

The role of the researcher was developed and evaluated through the course of Phases I and II of the study. This role varied widely, depending on the nature of the learning situation. Whilst the researcher's primary role was usually one of observation, in research situations such as that described in this study, an active role in the provision of cognitive support carried almost equal weight. Cognitive support came through intervention on the part of the researcher, and here important distinctions must be made between types and purposes of intervention. Intervention, whilst generally supportive in nature, contains one or more of three elements: instruction, support and exploration. Interventions, then, can be categorised as: those which in some way instruct pupils; those which support pupils' work with the aim of helping them progress further; and those which seek to explore more openly pupils' perceptions and understandings of a concept.

Within each category a number of sub-categories may be identified:

Instructive Intervention

- **introduction of a new concept** may occur at the beginning of an activity or part way through it and is characterised by direct, un-searching explanation, necessary to equip pupils with the information to proceed further;
- **complete explanation** generally concludes an activity (or at any rate that part of the activity which is useful for research purposes).

Supportive Intervention

- **partial or complete explanation** may be used to facilitate the next stage in an activity;
- **attention focusing** may be used to highlight one particularly fruitful avenue for enquiry, or a particular bug in a procedure, particularly if the pupil's attention has lingered unduly on an unfruitful or unhelpful area;
- if the learning situation involves a language other than the pupil's natural language, then a **syntactic explanation** may remove the problem of awkward syntax clouding the mathematics which should provide the true focus of the research;
- if a pupil makes a statement, a request for **clarification by the pupil** may provide a clearer re-statement (for the researcher's benefit) or give the pupil a second opportunity to correct any mistakes or misleading phraseology previously used, or even to shift the focus of attention;
- **clarification by the researcher** helps to make clear to the researcher what the pupil has just said and gives the pupil feedback on whether or not he has

communicated his ideas clearly. This re-statement may be verbatim (to lead to clarification) or re-phrased (to re-focus attention);

- simple **confirmation or rejection** gives pupils confidence to plan their next move;
- offered at vital moments to stimulate further activity, **encouragement** may involve re-focusing attention;
- if pupils are struggling with one particular representation at one particular level of abstraction, the researcher can offer support by a **re-presentation** of the idea in a form which the pupil may find more approachable.

Exploratory Intervention

One of the chief thrusts of the study was an investigation into how pupils' expression of mathematical concepts was mediated by the computational medium. Spontaneity was never anticipated as forming a major factor in the initiation of discussion, the researcher's intervention coming to the forefront in such situations: "Argumentation does not evolve in a proof spontaneously, but the specificity of the environment to which the argumentation is related and the direct guide of the teacher may determine this evolution" (Mariotti 1995, p. 187). The following types of intervention are thus integral to the initiation and development of fruitful discussion:

- the researcher can use **open questioning** in an attempt to prompt the statement of previously unarticulated thoughts, or to open up new avenues for exploration (in a form of attention re-focusing);
- **closed questioning** can also be used to explore and probe and to re-focus attention;
- **re-presentation** of an idea can be used to explore pupils' ability to express their ideas at a more or less abstract level.

The nature of intervention varied according to stage: at the construction stage, interventions tended to be more widely instructive and supportive; in the justification stage there was a shift of emphasis towards exploratory interventions.

4.5 CONCLUSION

In this chapter I have put forward a rationale for the design of the microworld and activities used in this study, drawing on the issues discussed in Chapters Two and Three. I have identified the use of activities centering on construction of simple function machines expressed through the use of Logo within a game-like context as providing a suitable means of offering a window onto children's expression of generality and justification. An integral feature of the design of the activities is the intended balance between empirical verification and verbal justification. The important role of the researcher has also been highlighted as being of particular significance.

In Chapter Five I trace the evolution of the design of the learning situation through the first two phases of the study.

CHAPTER FIVE: RESEARCH METHODOLOGY FOR THE PRELIMINARY PHASES

5.1 INTRODUCTION: METHODOLOGICAL ISSUES

The principal research focus for the thesis was to investigate how use of a carefully designed computer programming environment structured children's expression of generality and the justification of that expression. Rather than concentrating on the setting out and testing of a predetermined hypothesis, the approach to research was one involving observation of pupils' thought and learning processes. The main focus for the research was the observation of pupils engaged in a number of mathematical activities within carefully constructed learning situations. The methodology chosen for the study was "illuminative evaluation" where the "primary concern is with description and interpretation rather than measurement and prediction" (Parlett and Hamilton 1972, p. 8). Such a methodology centres around "observation, interviews with participants, questionnaires and analysis of documents and background information, examined within the school context or *learning milieu*" (ibid, p. 11). The researcher's role was viewed as one of participant rather than passive observation. Case studies were made and episodes from these appear in the data analysis chapters (Chapters Seven - Ten) to illustrate points made in the analysis. Techniques of data gathering used during the course of the study are described in this chapter together with descriptions of the initial phases of the study and an account of the evolution of the learning situation.

5.2 EVOLUTION OF THE LEARNING SITUATION

5.2.1 Overview of the Phases of the Study

The empirical part of the research was organised into three phases with Phases I and II used for the iterative refinements of ideas in preparation for Phase III in which the data used for analysis was collected. In the initial phases activities were devised, tried out and, where necessary, their content was adjusted or their order re-sequenced.

	Phase I	Phase II	Pre-Task Interview	Phase III
Date of Empirical Work	May 1995	June - July 1995	October 1995	October 1995 - January 1996
Number of Sessions	Three	Four	One	Four
Number of Pupils (working in pairs)	Two	Six	Sixteen	Sixteen

Table 5.1: Summary of the Study Phases

Phase I was intended to provide a clarification of the theoretical orientation of the thesis. It involved looking at possible tasks, their context and their relative timings. It also looked at the software and provided some pointers for the eventual evolution of the learning situation. Phase II continued the adjustment of software, facilitating the trialing of a new piece of software. It also involved the re-sequencing of tasks. The main purpose of Phase II, however, was to develop a provisional scheme of categorisation which would eventually form the categorisation for the analysis of the main findings in Phase II of the study. A summary of the study phases is given in Table 5.1. The following sections consider some preliminary thoughts on learning situations and the balance between teaching and research and then trace the development of the learning situation through descriptions of Phases I and II of the study. The categorisation which resulted from the second phase is described in detail in Chapter Six.

5.2.2 Preliminary Thoughts: The Research - Pedagogic Balance

The original conception for the study was one of several self-contained microworlds, each of which contained instructions for its use and a clearly laid out course of action for the pupils to follow. For example, a microworld was developed for constructing procedures to output odd numbers and even numbers (Figure 5.1).

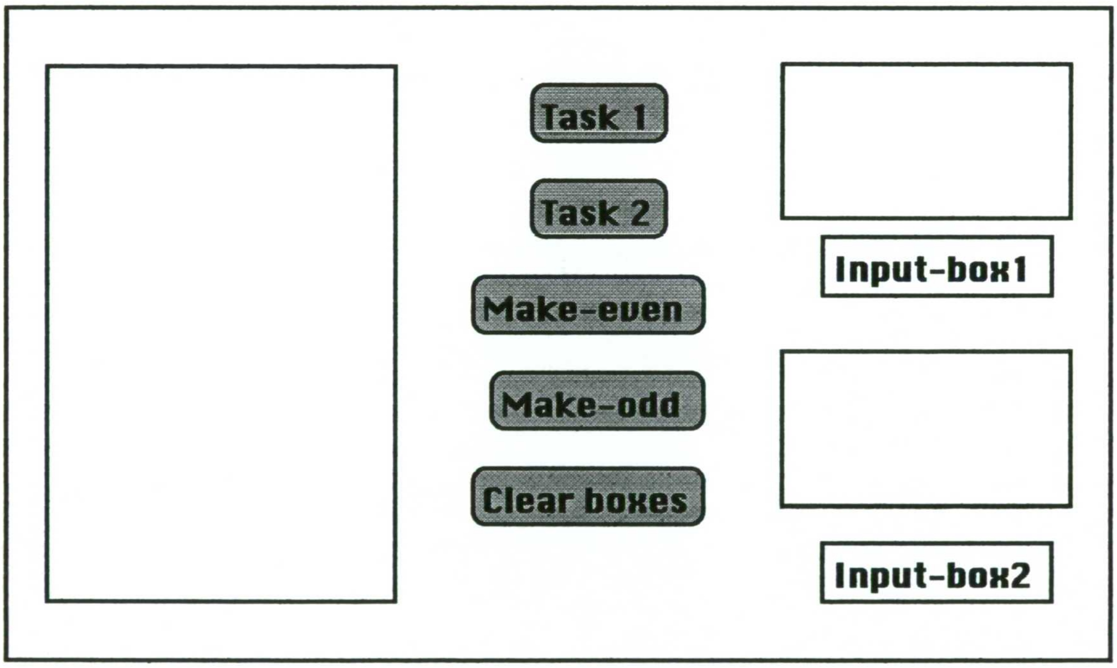


Figure 5.1: Initial Attempt at Software

The button "Task 1" printed out the first task for the pupils to attempt in the left hand text box:

> Odds and Evens 1
>
> You have to write a procedure which changes any number into an even number.

Call this procedure **even**.

If you have been successful, when you write a number in the input-box and press the button "make-even", an even number will appear in the output-box.

The button "Task 2" printed out the second task:

> Odds and Evens 2
>
> You have to write a procedure which changes any number into an odd number this time.
>
> Call this procedure **odd**.
>
> If you have been successful, when you write a number in the input-box and press the button "make-odd" an odd number will appear in the output-box.

Although such a self-running format might have a place within the classroom, where the sole desired outcome might be for pupils to learn about the process of changing numbers into odd numbers or even numbers, it falls down on a number of counts as a vehicle for research. Chief amongst these is that pupils are working within what amounts to a prescriptive straight jacket of the researcher's perceptions of the task, rather than in a situation where they are able to pursue their own ideas in directions of their own choosing. This directly contradicts the position taken in Chapter 2 ("to try to capture experience of the inner world from the inside ... by metaphor and descriptive frameworks that resonate with other people's experience" Mason 1987b, p. 208). Also, as Sutherland points out, "most of the potential of programming within mathematics education will be lost if teachers over-direct students' problem solutions by an overemphasis on pre-written macros" (Sutherland 1994, p. 186). Thus very early on, the philosophy behind the construction of the learning situation was modified to encompass a freer approach in which pupils were given greater room for self expression.

5.2.3 Phase I

Purpose

Phase I of the Study involved observation of two pupils' work during three sessions, each of approximately one hour. First came a session on function machines, designed to introduce the pupils to the use of function machines in a Logo context and to the use of the Tabulator[27]. In the second session, work involved concepts of odd and even numbers. The pupils first devised definitions of odd-ness and even-ness and then rules about odd and even numbers under addition and multiplication. When this had been done, they were asked to write Logo procedures which modelled their previously written definitions and rules. The third session was in two parts, focusing on generalisation of number puzzles of the type "pick a number, add 2, multiply by 3 ..." and on defining and operating on consecutive numbers in the context of Logo function machines. These activities are summarised in Table 5.2.

Session	Activity
One	Creating simple function machines from natural language descriptions and tables of data
Two	Creating Logo procedures to model definitions of odd-ness and even-ness
Three	i) Generalisation of number puzzles ii) Defining and writing procedures to model relationships between consecutive numbers

Table 5.2: Phase I Activities

[27] The software designed for the study.

Subjects

The participants, Joe and Daniel, were both eleven at the time of Phase I and in their final year of primary school. Daniel enjoyed and was good at mathematics, whilst Joe was about average for his year group in the standardised NFER[28] tests used at the school. In their previous work with Logo, Joe had shown a greater willingness to experiment with the computer and to make guesses which might well lead to the wrong solution, whereas Daniel proved to be more cautious and more methodical. Their previous experience of using Logo is summarised in Table 5.3.

Activity	Logo Concepts
Drawing shapes in direct mode	Use of Logo primitives: FD, BK, RT, LT, HOME, CS, LIFT, DROP, ST, HT, SETPC
Use of REPEAT for drawing regular polygons	Use of REPEAT
Use of Logo procedures to name previously drawn shapes	Use of TO .... and editor
Drawing shapes with fixed number of sides and fixed angle, but variable side length.	Procedures containing variables
Drawing shapes with fixed side length and fixed angle, but variable number of sides	Procedures containing variables
Spirals	Self-naming procedures, recursion

Table 5.3: Pupils' Previous Experience of Logo (Phase I)

---

[28] National Foundation for Educational Research

Observations

Several important points emerged from Phase I relating to the tasks, the learning situation and the ordering of the tasks.

Session One

In Session One an interesting dichotomy emerged relating to the use or otherwise of turtle geometry. Should a link be made between operations on a variable in a function machine context and the use of a simple variable in a turtle graphics context? For example, might an intermediary step be the introduction of procedures such as

```
to square :side
repeat 4 [fd :side * 2 + 3 rt 90]
end
```

On the other hand, given the pupils' total lack of reference to turtle graphics examples, their apparently hazy recollection of the use of variable in that context and their quick acceptance of :x as a variable in the context of function machines, was there any need for pupils to have previous experience of turtle graphics in order to use function machines? Certainly, familiarity with Logo procedures was clearly necessary, but even that familiarity might be learnt within a function machine context.

Daniel and Joe had considerable experience of turtle geometry and with nothing to compare them with, the issue seemed important at this stage. In Phase II of the study there was a spread of experience and inexperience (and in Phase III by chance exactly half the pupils had experience of turtle geometry and half had none or next to none) and no appreciable difference was observed between the programming ability of those with experience and those with none. This was not a main theme of the study, but the implication from casual observation would appear to be that previous experience of programming in the context of turtle geometry had little or no effect on pupils' ability to program function machines in Logo. Certainly all pupils were treated as "beginners" in programming in all three phases of the study and taught about the use of variable with the assumption of no prior knowledge.

That said, once the pupils had embarked on the tasks, a reasonable degree of practice in writing function machines seemed necessary to ensure familiarity with the language, practice not afforded by the order of the tasks at this stage. Thus, in order to provide this practice, the number puzzles in part three were placed directly after the initial section on function machines.

It was hoped that the various tasks would engender interaction between discussion, paper and pencil work and use of the computer. As it turned out, Joe and Daniel appeared to do most of their working out in their heads and used Logo merely to *translate* their previously worked out solution into the required format. The Tabulator served as a checking device at the end. This implied a problem with either the difficulty level or nature of the tasks, or indeed with the Tabulator software. Given the scale of Phase I of the study it was difficult to reach conclusions about the appropriateness of the level of difficulty, so it was decided to monitor this and the use of the Tabulator closely in Phase II.

Session Two

The pupils' mathematical knowledge was greater than anticipated, so that they were able to work out definitions and rules on paper and actually found interaction with the computer in places a hindrance rather than a help. For this reason, this was in many ways the least fruitful session. In fact, as far as the Logo work was concerned, the pupils seemed out of their depth. This led to the decision that a position later in the programme of tasks for the work on odd and even numbers would yield more of interest to the research from the point of view of their familiarity with and therefore willingness and ability to use the computer.

Observation of work carried out in Phase I provided a system for coding pupils' responses to the tasks. This system is summarised in Table 5.4 (some examples being taken from later in the study). This coding was then trialed in Phase II.

Terminology	Explanation	Example
term-by-term analysis	pupils discussed a procedure in terms of its constituent parts, very often pointing at individual terms on the computer screen.	Daniel: *Because look, it's :t plus :t would be ... um ... say you had 4 that would be 8 and then it would just add :t again. It would only add the one you started with.*
empirical verification	pupils checked that a procedure produced an expected outcome either through mental calculation or by means of running the procedure on the computer.	Sean: *Now what should the output be if it works?* Sandy: *9 + 10 is 19.* (They tried 9 in the Calculator and gave an output of 19)
procedure construction	the construction of a computer program on the screen	
natural language	everyday, typically spoken, language used in the description of mathematical ideas	Joe: *Every time you get a number and you times it by eight it comes out even.*
formalised language	language concurrent with some accepted mathematical formalism such as f(x,y) functions.	Sean: *The reason it hasn't worked is because you told it to add 6 to :m. It says the next consecutive number, so it wouldn't be :m + 6, would it? It's more likely to be something like op :m + (:m + 1).*
equivalent procedures	procedures which performed the same function but were expressed differently.	to mmm :p op :p * 2 end to ilil :y op :y + :y end

Table 5.4: System for Coding Pupils' Responses

informed construction and verification	procedure construction which was somehow shaped by what had gone before it.	Richard: *Well what I did, I looked at the very first one and it's 0 and I saw that it came out 3. So I put in 2 times, then the side and then I added 1 for starters. And that came out as - I've forgotten what that came out as. It didn't work, so then I put in - I kept the 2 times the same but I changed the plus to 2 instead of 1 and it was one number away. Then I put 3 there and it did work.*

Table 5.4 (contd): System for Coding Pupils' Responses

Session Three

In this session a lack of rigour became apparent in the pupils' work, particularly in the boys' attempts to express their ideas verbally. From a desire to make pupils think about what they were saying, the idea of a game context for the study emerged. In such a context, the two pupils would be competing with one another and it would be necessary for them to convince one another of their mathematical discoveries through the power of their verbal reasoning (thus demanding a greater degree of rigour). Such a context would also have the power of eliminating the element of 'pleasing sir' which was also noticeable in Phase I.

The function machines which were constructed by Joe and Daniel in the "Amaze Your Friends" problems constituted an obvious extension of the basic function machines developed in the first session. For this reason it was decided to place this section as the second activity in later phases of the study. It was interesting to note that the pupils clearly very much enjoyed this work as it related to puzzles they had already come across, thus creating close links with their school curriculum.

In summary, Phase I of the study helped to determine a new order for the tasks, summarised in Table 5.5:

Session	Activity
One	**Function Machines**: Creating simple function machines from natural language descriptions and tables of data
Two	**Amaze Your Friends**: Logo generalisation of number puzzles
Three	**Consecutive Numbers**: Defining and writing procedures to model relationships between consecutive numbers
Four	**Odd and Even**: Creating Logo procedures to model definitions of odd-ness and even-ness

Table 5.5: Phase II Activities

In addition, the game-like context for the tasks was born and the difficulty level of the tasks and the use of the Tabulator became two major areas for review in Phase II.

Data Collection

Each session was audio-taped from beginning to end and the result later transcribed. The researcher observed the sessions closely and made field notes, recording in particular what the pupils were typing onto the computer (including iterations in procedure construction) and other interactions with the computer (for example pointing at specific terms on the screen). All procedures created by the pupils were retained as Logo files and a print out of these was used to aid the transcription.

5.2.4 Phase II

Purpose

The main purpose of Phase II was to review the use of the Tabulator software, developed as a result of Phase I, to consider the difficulty level of the tasks and, most importantly, to begin to form categories which would be used for analysis in the final phase of the study.

Adjustments in Light of Phase I

In Phase II of the study, each of the three groups worked for approximately an hour at each of the four sessions. The first session remained unchanged from Phase I and dealt with writing and then devising function machines. This session remained subsequently unchanged in Phase III. The second session centred on the *Amaze your friends* type of problem and was identical to the set of tasks used subsequently in Phase III[29] with the addition of an open question inviting pupils to devise their own puzzle. This was dropped from Phase III due largely to lack of time: in fact in Phase III of the study no pupils had time to go on to this question. The third session dealt with consecutive numbers and in addition to the Phase III question contained the following:

> *d) Here's an "Amaze Your Friends" type puzzle using consecutive numbers:*
>
> *Pick a number between one and ten. Add on one. Multiply this new number by itself. Call this number **Answer A***
>
> *Now go back to the number you first thought of. Multiply it by the number that*

[29] See Appendix Two.

*is two bigger. Call this new number **Answer B**.*

*Now take **Answer B** away from **Answer A**.*

Try different values for the input (use the Tabulator). What do you notice?

This question was also dropped from Phase III due to lack of time and also due to the complexity of the question and the degree of explanation and intervention needed from the researcher. It was felt that the remaining tasks provided enough data of sufficient quality to warrant the omission of this task.

The final set of tasks, centering on the properties of odd and even numbers, was substantially remodelled between Phases II and III. In Phase II the computer-based work was prefaced with a number of questions designed to ascertain some idea of the pupils' existing understanding of odd and even numbers. In Phase III these, along with other questions relating to the other parts of the study, were collected together in the pre-task interview. The computer-based part of this section was confined to a single open-ended question:

> *e) What happens when you add, subtract, multiply and divide odd and even numbers? Can you think of five or six sentences to say what happens. You could begin*
>
> ***When you add two odd numbers together ...***

In the event this proved to be *too* open ended, with the result that either what the pupils produced was of little interest, or the researcher was involved (as in the final consecutive numbers task) in an excessive amount of intervention. Consequently the series of structured tasks used in Phase III of the study was devised. Changes to the tasks as a result of Phase II are summarised in Table 5.6.

Session	Activity	Changes
One	Function Machines	None
Two	Amaze Your Friends	Removal of final open-ended question inviting pupils to devise their own puzzle
Three	Consecutive Numbers	Complex puzzle question removed
Four	Odd and Even	a) verbal questions removed to pre-task interview b) original open-ended question amplified and made more specific

Table 5.6: Changes to Phase II Activities

Subjects

Phase II of the study involved observation of six pupils working in three groups of two. Each group worked for approximately an hour at each of the four sessions. The pupils who took part in Phase II were from a greater range in terms of age and ability than those in Phase I, or indeed those in Phase III itself. Sean and Sandy were both able mathematicians[30], but Matthew and Richard were less so, with Richard below average, according to his class teacher. These boys were all aged ten at the time of the study. The third group consisted of Scott and Daniel who were thirteen and twelve respectively at the time of the study. Whilst Scott was reasonably able mathematically, Daniel was well below average. (It may be apparent that all the pupils used this far were boys. This was due to their availability and willingness to give up their own time to take part in the two studies. However, the pupils who took part in Phase III were boys and girls in equal number. The distribution of boys and

[30] Gauge of ability here was based on comments from class-teachers backed up by NFER test results.

girls is summarised in Table 5.7)

	Boys	Girls
Phase I	2	0
Phase II	6	0
Phase III	8	8

Table 5.7: Distribution of Boys and Girls taking parts in the Study

Observations

The main points to emerge from Phase I concerned the setting up of the game context, the difficulty level, the order of tasks and the role of the computer software. These issues are addressed in the following discussion of Phase II, and other issues are touched upon which emerged in the course of Phase II and later provided foci for observation and research in Phase III.

<u>Construction and Verification</u>

Evidence from Phase II suggested there was considerable interaction between pupil and computer during the construction phase. The rarity of examples of pupils using the computer merely as a means of recording their ideas coupled with the high frequency of this behaviour in Phase I suggested that the level of difficulty of the tasks was just about right, if children of comparable ability were to take part in the final phase of the study. The process of auto verification (discussed below) emerged as an important element in the construction process and implied that the processes of construction and verification were less two discrete processes than two elements of one process. Types and frequencies of verification would also need to be considered.

At an early stage it became apparent that verification had two elements: verification by pupils of their own procedures for the purpose of improving the construction of their procedure; and verification by one pupil of another's procedure. Whilst, given the environment in which the pupils were working, it seemed that the latter form of verification (what I have termed altero verification) was an ordered, discrete process, the former type of verification (auto verification) was an integrated element in the construction of the Logo procedures which made up the study.

Perhaps the most apparent distinction which emerged through Phase II was that between verification which fulfilled the requirements of the question and that which did not. Analysis of this distinction proved to be clouded by the software (as was feared in Phase I) and re-development of the Tabulator was required.

The Tabulator software tabulated and output values of a function within a range defined by the pupil (Figure 5.2). Whilst a useful and effective teaching tool, this failed as a research tool for the purpose of opening a window onto pupils' emergent understanding of what constituted empirical verification. Built into the Tabulator was the means to regulate both the range of the input values and the difference between successive inputs. This meant that it was possible to consider, for example, single values in isolation, a range of values ascending in integers, or successive even integers (by setting the initial value to, say, 2 and the interval to 2).

In practice, however, the range of values for the input tended to be left unchanged and therefore no thought was put into which and how many input values were needed (in the pupils' eyes) for adequate verification of the procedure to occur. In fact since most pupils appeared to forget about the facility to change the range of input values, they treated the values previously set as constant, unchangeable and as *the* range of values necessary for verification. In the *Amaze Your Friends* tasks (Tasks 2.1 - 2.3), no pupils in Phase II recognised that the range defined in the questions ("Pick a number between 1 and 10") and the range of inputs previously set in the Tabulator (inputs between 0 and 10) were at variance. (Here the range was important as an input of zero led to division by zero in some procedures).

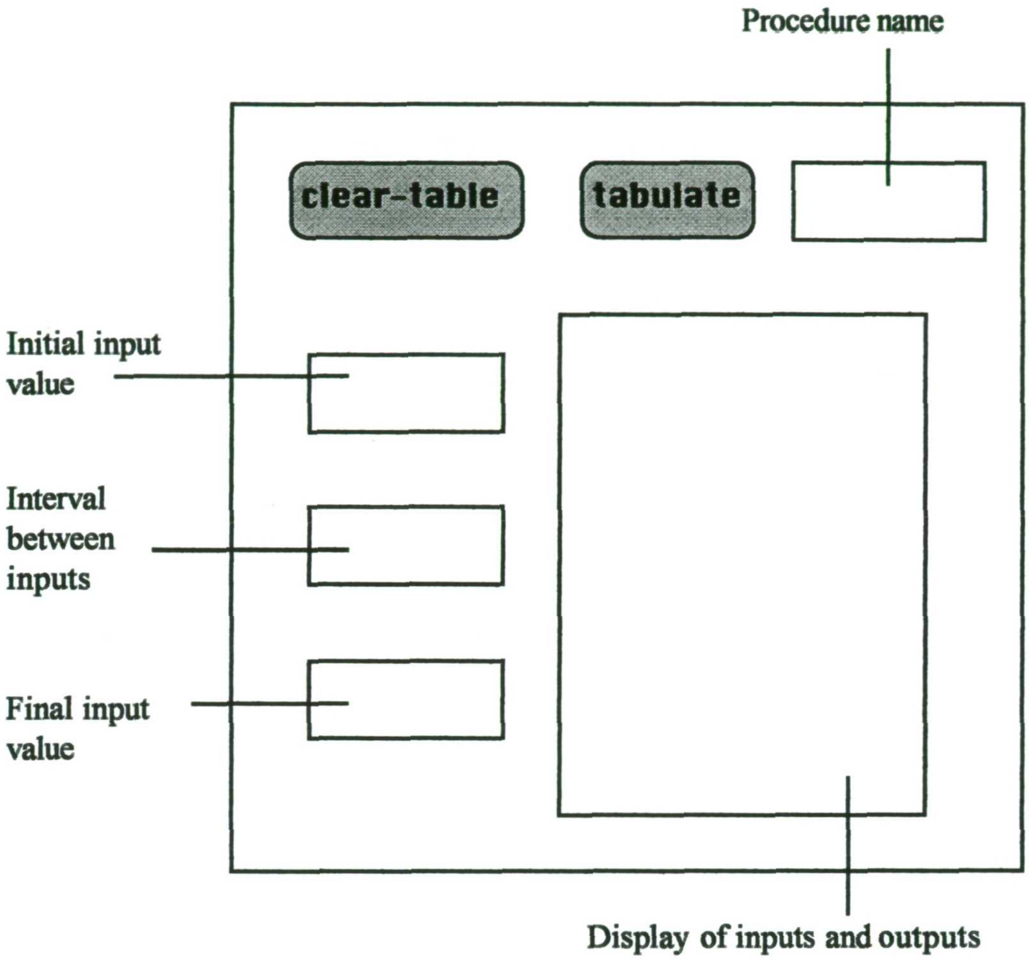


Figure 5.2: The Tabulator Software

The Tabulator, with inputs treated as unchangeable, also failed to make adequate distinction between odd and even inputs in the *Odds and Evens* Tasks (Tasks 4.1-4.4). Here it was important to consider independently the effect of odd and even numbers and it was important for pupils consciously to select, test and observe the effect of odd numbers in their procedures as distinct from even numbers. Whilst the Tabulator in its invariant form included both odd and even inputs, they appeared in numerical rather then type order.

Now, it could be argued that pupils were misled by the previously set values and that if no values were set in the first place, pupils would be forced to consider their own values for each verification. Phase II suggested, however, that once a set of values had been arrived at (and it tended to be values from 0 to 10 since the *Function Machines* examples were illustrated with tables of values with inputs from 0 to 10), then pupils showed no inclination to change them.

In short, then, the Tabulator, through its misuse (or, more precisely, *uninformed* use), failed to give pupils control over, and even responsibility for, their own verification. Yet the visual impact and the potential for selective inputs were features worth making use of. What was needed was something which combined visual impact with the necessity to consider and select input values. To meet these demands the Calculator was developed (Figure 5.3). Based on, but much simpler than the Tabulator, this only allowed the selection of one input value at a time, but had the advantage of displaying and retaining input and corresponding output values in a large textbox in the order in which they were entered, so that, for example, a table of inputs from 1 to 10 or a collection of even inputs and their corresponding outputs could be built up *if that was what the pupil saw as the appropriate set of inputs for verification* in a particular case.

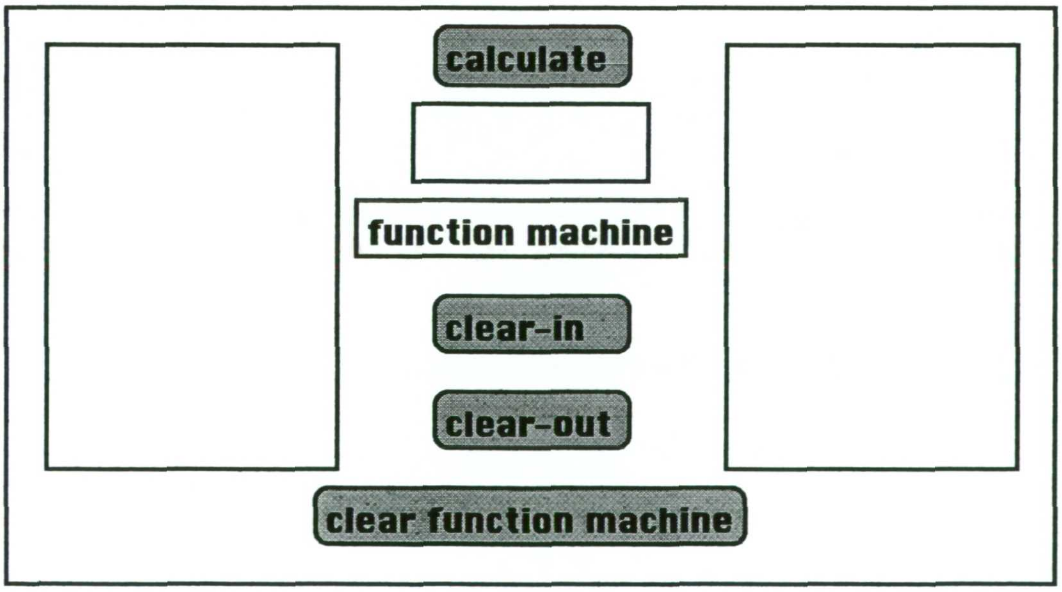


Figure 5.3: The Calculator Software

Justification

Phase II certainly appeared to suggest that justification, be it through spontaneous discussion or through researcher-given prompts, developed the ability to make reasoned statements about procedures which went beyond trivial term-by-term descriptions, to draw out general statements from the consideration of specific examples, and to talk in a generalised way about procedures without reference to specific examples, and to express notions of justification.

The important focus for Phase III would be identifying the role of the computer in facilitating these activities.

Some pupils experienced confusion in the *Amaze Your Friends* problems over the difference between specific examples and the general case; if in the initial oral examples they had chosen, say, 7 as the input, then there was a temptation to use 7 in the procedure as the number to be operated upon. This anomaly (which I later refer to as semi-generalisation[31]), rather than posing a problem as I first feared it would, turned out to be an important source of data in the analysis of Phase III.

It became increasingly clear that in the majority of cases, justification and verification, whilst easily discernible in analysis, were difficult to separate in practice. The distinct phases outlined in the model of the components of the setting[32] simply did not occur as such. This did not invalidate this model, but highlighted potential pitfalls during the analysis phase. It implied importantly that whilst analysis of construction, verification and justification might be carried out in isolation (at least initially), the relationship between these elements would also be of importance, and that the

[31] See Chapter 8, § 8.2.

[32] Table 4.1

common themes running through these elements would provide foci for analysis. The balance between verification and justification came through strongly as an area for further investigation.

Researcher Intervention

The precise nature of the role of the researcher emerged through the course of Phase II and this role is described in detail in Chapter Four[33]. Although an introduction was envisaged at the outset of each task, the number of additional interventions which were needed at further points during the construction phase, which was originally seen as being free of interjections, was not envisaged. As has been described above, certain tasks were modified or dropped because of the difficulty level and the consequent need for excessive researcher intervention. Expectations of spontaneous discussion ensuing from the verification process proved exaggerated; in reality, the situation was too staged to allow a great deal of spontaneity[34] and it was often necessary to illicit pupils' views by means of careful intervention. The rationale behind the interventions is also examined in Chapter Four.

One intervention which became standard was in connection with the use of brackets. Although all children had encountered and used brackets in the course of their school mathematics lessons, they had not previously encountered them in a Logo context and an interjection proved necessary in the course of the *Amaze Your Friends* tasks (Task 2.1). This was done at a point where pupils had produced an otherwise correct procedure but were getting apparently inexplicable answers due to the lack of brackets. The interjection was simple: pupils were asked to work out the following two expressions

$$(2 \times 3) + 1 \text{ and } 2 \times (3 + 1)$$

[33] See Chapter 4, § 4.4.2

[34] Although important exceptions to this are discussed in Chapter 11, § 11.7.1.

and then the following two:

$$(2 + 3 + 5) / 5 \text{ and } 2 + 3 + (5 / 5)$$

It was then suggested that they consider the use of brackets in their procedures. A second intervention concerned the use of brackets within brackets, i.e. the enumeration of the innermost set of brackets first[35].

In summary, the following points emerged as foci for analysis in Phase III of the Study:

- types of verification and choice of values used;
- frequency of verification;
- relationship between verification and construction;
- the role of the software in the verification process;
- the changing role of verification;
- semi-generalisation;
- relationship between justification and verification.

Data Collection

The processes of data collection in Phase II of the Study remained the same those in Phase I:

- audio-tapes taken from beginning to end and the result later transcribed;
- close observation of the sessions by the researcher;
- field notes made by the researcher (recording what the pupils typed and how they interacted with the computer);
- Logo files of pupils' work made and printed out as an aid to transcription.

[35] This was referred to as the Inverse Dinner Party Analogy, given that use of brackets is precisely opposite to the use of cutlery at a dinner party consisting of several courses.

5.3 CONCLUSION

In Chapter Five the evolution of the learning situation has been charted from early ideas through the initial two phases of the study to the point where the final phase was ready to be undertaken. The main points of development concerned:

i) the difficulty level and order of the tasks;

ii) the coding of the data;

iii) the emergence of the game-like context;

iv) the evolution of the software to enable a better insight into children's creation of meaning; and

v) the role of the researcher.

In the next chapter, a new theoretical framework is discussed, arising as a result of these initial phases. This is followed by a description of the methodology of the main study (Phase III), after which an initial analysis of the main study data leads to the formulation of research questions.

CHAPTER SIX: FRAMEWORK AND METHODOLOGY FOR THE MAIN STUDY

6.1 INTRODUCTION

In the foregoing chapters of the thesis I have identified some important theoretical concepts through a discussion of literature on representation and abstraction (Chapter Two), I have explored the literature proof and processes of proof in order to gain a perspective on processes of generalisation and justification (Chapter Three) and I have described the design of the programming microworld used in the study (Chapter Four).

Chapter Six looks, Janus-like, back into the old and forward into the new: back in the sense that as a result of some initial findings from Phases I and II of the study (described in Chapter Five), I shall draw together strands from the preceding chapters in order to present my own theoretical stance which will then provide the focus for research in the final phase of the study; and forward in the sense that I shall then describe the methodology for the final phase of the study and identify broad categories for analysis of data which then form the remaining chapter headings of the thesis. The various elements of this chapter will also be brought together to frame a set of research questions.

6.2 TOWARDS A THEORETICAL FRAMEWORK AS A RESULT OF ANALYSIS OF PHASES I AND II

In the introduction to the thesis I used what I termed the S-column metaphor to offer a possible alternative to the ordered progression through layers of meaning in some predefined sequence which characterises so many theories of learning and which, consequently, epitomises so many commercial mathematics texts and courses for school children. I suggested that rather than ordered vertical progression through layers of ever more abstract knowledge - entry to each layer being dependent on complete mastery of that preceding it - children might acquire meaning through examples of situated knowledge which simultaneously incorporated several layers of abstraction. The emphasis here would be not so much a progression from one layer to the next, but the interplay *between* layers. This I represented pictorially by a narrow column cutting through the broad strata of knowledge - a bore-hole - (Figure 6.1) to signify at once the limited nature of the concept confined to its specific situation, and the inclusion within that concept of multiple levels of abstraction.

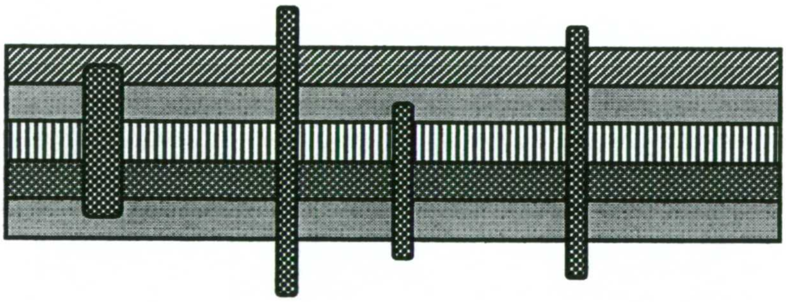


Figure 6.1: Model S - Specific Naive Schemata: situated "bore holes" are shown cutting through the strata of ordered knowledge

Since then, in Chapter Two, I have described Wilensky's subjective definition of concrete (and abstract) and his use of the term *concretion*, developing a theory where children construct meaning in an apparently unique fashion, building their own mental representations and making all-important links between these representations, within the important parameters of situation and context. Within this theory, then, I am suggesting that meaning is subjective and that perception of what is and what is not concrete is similarly subjective. In Chapter Two I also described the *webbing* theory developed by Noss and Hoyles (1996). As part of this theory they take a view of abstraction which - resonating closely with Wilensky's ideas - extends the S-column metaphor to view the process of abstraction not as the replacement of one kind of meaning by another but rather as the creation of connections between ways of knowing. Meanings "become reshaped as learners move the focus of their attention onto new objects and relationships within the setting" (p. 49). I want to concentrate particularly on ways pupils use these new objects and relationships in their attempts to express proof and generality, using the idea of a "web of meaning" as a framework within which to make sense of these activities. This in fact is what Noss and Hoyles term *situated abstraction*, that is where learners draw on the webbing of a setting to construct mathematical ideas, and where the webbing shapes the expression of those ideas. Thus the three theories of concretion, webbing and situated abstraction underpin the theoretical basis of this thesis.

Now, the hierarchic ordering of representations - an interpretation perpetuated by the S-column metaphor - becomes more and more at odds with this developing theoretical basis, for it would appear difficult to reconcile such a hierarchy of representation as depicted in the strata of figure 6.1 - an observer's uniquely subjective creation - with the uniquely subjective creation of meaning and concreteness of each pupil as I have described in this thesis. However, there is still a question of the place of *local* as opposed to *global* hierarchies in understanding how pupils link their representations, so for this reason I prefer to leave the question of hierarchy open until the concluding chapter.

At the centre of the theoretical basis I have described - and at the centre of the research focus for this thesis - lie children's representations of knowledge and the ways in which they link these representations. It is not my wish, however, to describe these constructs as "representations" and their connections "translations" between representations, as with these terms comes a theoretical stance which I have previously rejected[36] where the focus for research is on mappings from one external, teacher-imposed representation to another. As has already been described, this view of representation and abstraction is at odds with the pupil-constructed theory of representation which lies at the heart of this thesis - and indeed the very term representation lies uncomfortably with the idea of unique creation of ideas as opposed to the replication of those of another. A further problem of terminology arises since to remain entirely in sympathy with the webbing metaphor and also with Wilensky's essentially subjective definition of abstract and concrete[37] it becomes difficult to describe certain objects as being "more abstract" or "less abstract" than certain other objects.

So in order to define a terminology consistent with the theoretical framework, I want to adopt the terms **Expression of Meaning** and **Connections** to signify the ways in which pupils express their developing meanings in different forms, and the means through which they connect these different expressions. I shall also use the verb to **re-express** to signify the expression of a mathematical relationship in a new form. Thus in general terms the focus for research in this study may be described as how pupils shape meaning and this is viewed in terms of the creation of expressions of meaning, their re-expression and the connections made between them.

How, then, do generality and justification fit into this theoretical framework? The overall aim of the thesis was stated in Chapter One as follows:

---

[36] See Chapter 2, § 2.2.2.

[37] See Chapter 2 Section 2.2.3.

> To observe children's creation of meaning through the expression of mathematical generality and through the justification of that generality.

Thus the focus for the thesis is on how children were able to express generality and justification and on how, by forging connections between these expressions, they concretised the concepts of generality and justification. The construction of a learning situation which incorporated a programming microworld and the active participation of the researcher was designed to create webbing which allowed them to generalise and justify, the webbing shaping those generalisations and justifications. Thus the aims of the thesis may be stated as follows:

i) to investigate ways in which children expressed generality using the programming medium;

ii) to investigate connections between different expressions of generality made in different modalities and their role in children's creation of mathematical meaning;

iii) to investigate the role of connections between expressions of generality in children's attempts to justify their constructed generalisations.

Thus the concept behind the design of the learning situation was to place within pupils' reach a number of means through which they might explore, generalise and justify mathematical relationships: the construction of function machines in a formal computer programming language; the empirical testing of constructed functions within a competitive environment; the tabular display of ordered pairs; the verbal and empirical justification (in natural or other language) of constructed functions, within both a competitive situation and a collaborative situation, the latter involving both pupils and researcher. It was my wish not to force my expressions of meaning of the tasks I had set on the pupils who were carrying them out, but at the same time it was necessary to balance this desire with the need to make available to pupils a variety of possible means of expressing their ideas.

Now, Noss and Hoyles (op cit) confine their discussion of the webbing theory largely to an exploration of what I have defined as expressions of meaning, rather than to the means of making connections between these expressions of meaning. Focusing on how pupils create connections between expressions of meaning for generality is an important and necessary development towards gaining further insights into how children make sense of and express generality and justification within a situated context. It is through focusing on the creation of connections that this thesis aims to extend the notion of webbing and of situated abstraction.

The research foci from Chapter Three provide useful pointers to areas which may provide glimpses into pupils actively engaged in the activity of creating expressions of meaning for generality and justification. Chapter Three highlights the following foci for possible research. Partially rephrased in the new terminology defined in this chapter, these are as follows:

i) The relationship between construction, expression, generalisation and justification;

ii) The connection between children's visualisation of mathematical relationships (within the specific learning situation) and their expression of generality and justification;

iii) The role children's construction of programs plays in effecting the abstraction of mathematical relationships;

iv) Ways in which children's exploration of particularity leads to generality;

v) The creation of connections between expressions of meaning;

vi) The role of connections in the shaping of meaning;

vii) The relationship between the language of construction, the language of expression and the children's own natural language within a specific learning situation;

viii) The role which computer-mediated language plays in the construction and expression of generality and justification.

From consideration of these foci from Chapter Three and from tentative findings from the initial phases of the study, the following five research questions emerge:

i) How does the use of empirical verification mediate the expression of generality through the construction of formalised procedures?

ii) How do tensions in the balance between empirical verification and justification mediate the creation of connections between expressions of meaning?

iii) How and to what extent does formalised language act as a means of connection or dislocation between expressions of meaning?

iv) How does the contrast and comparison of equivalent procedures shape the exploration of structure within mathematical relationships?

v) How does the computer mediate the creation and use of generic examples as a bridge between the specific and the general?

As a preliminary to attempting to answer these questions, I felt it necessary to carry out an initial analysis of the data from the final phase of the study. This was in order to look at ways in which pupils appeared to express generality and justification in their work so as to build up a picture of some of the possible elements in individual webs. Although meaning is subjective and consequently each web is unique, there was every reason to suppose that individual webs would contain similar expressions of meaning. The identical learning situations in which pupils worked and the similar mathematical backgrounds from which they came suggested some commonality of means of expression. This initial analysis appears in Section 6.4 and leads to a refinement of the research questions.

6.3 METHODOLOGICAL STRUCTURE OF PHASE III

In this section I look at the organisation of the third and final phase of the study.

6.3.1 Subjects

Phase III of the Study involved eight pairs of children, selected largely on the basis of standardised scores obtained in NFER[38] Mathematics Tests administered annually by the school the children attended (see Table 6.1).

Group	Pupil's Name	NFER Score	Age
A	Kimberley	121	10
	Vicky	122	10
B	Rachael	129	9
	Hannah	111	9
C	Kathryn	122	9
	Sarah	103	9
D	Matthew	119	9
	Ben	133	9
E	Luke	133	10
	Jamie	119	10
F	Jenny	125	10
	Sophie	126	10
G	Edward	121	10
	Nicholas	111	10
H	David	115	9
	Tom	137	9

Table 6.1: Pupils participating in Phase III of the Study

These scores place all children involved above the average (100), with most within the top five percent. The children were all aged nine or ten at the time of the study and volunteered to take part in the study, which was carried out in the children's own time

[38] National Foundation for Educational Research

(lunch hours and after school or at weekends). The nine year old children had a little experience of turtle geometry having been taught the previous year by the researcher who was also a teacher at the school; the ten year old pupils had no experience of Logo. Pairings were chosen by the pupils on the grounds of friendship and consisted of four pairs of boys and four pairs of girls, two each from Year 5 (Groups B, C, D and H) and two each from Year 6 (Groups A, E, F, and G)[39]. During the research sessions (each lasting approximately an hour) each group attempted one of the four sets of tasks.

6.3.2 The Pre-task Interview

Each child received an individually administered pre-task interview which was recorded on audio tape. The interview had four objectives. It sought to gain an insight into:

- pupils' understanding of the mathematical concepts involved in the study tasks;
- pupils' ability to express their ideas in natural language or any formalised algebraic language of which they had knowledge;
- pupils' perceptions of what constituted empirical testing; and
- pupils' ability to explain and analyse mathematical relationships.

To achieve these objectives, the interview was divided into seven sections, the main focus of each being as follows:

- explanation of how simple function machines work;
- working out of a rule for generating a set of tabulated values and describing criteria for being sure it works;

[39] U.K. compulsory school education is currently spread over twelve years: pupils enter school at the age of 5 and subsequently spend Years 1 - 6 at primary school and Years 7 - 11 at secondary school

- mentally calculating a string of operations and explaining why the answer is constant whatever the starting number;
- definition of consecutive numbers;
- recognition and definition of even numbers;
- recognition and definition of odd numbers; and
- generalisation of changing odd to even (and vice versa) and generation of odd and even numbers.

Results of the interviews were used to inform subsequent analysis of data and references to the interviews appear in the data analysis chapters. The full questionnaire appears in Appendix One.

6.3.3 Summary of Tasks

The rationale behind the tasks is described in Chapter Four, and the evolution of the tasks is described in Chapter Five. Table 6.2 contains a summary of the final order and content of the tasks used in the main study. The complete set of tasks is shown in Appendix Two.

Session One: Function Machines	
Tasks 1.1 - 1.3	Construction of procedures from visual representations of function machines
Tasks 1.4 - 1.6	Construction of procedures to model tabulated pairs of values
Session Two: Amaze Your Friends	
Tasks 2.1 - 2.3	Construction of procedures to model number puzzles described in natural language

Table 6.2: Summary of main study tasks

(although there are a number of local variations in organisation).

Session Three: Consecutive Numbers	
Task 3.1	A procedure to output the sum of two consecutive numbers
Task 3.2	A procedure to output the sum of three consecutive numbers
Task 3.3	A procedure to output the product of two consecutive numbers (described as a number and the number preceding it)
Extra	A procedure to output the product of three consecutive numbers (described as a number and the numbers following and preceding it)
Session Four: Odd and Even	
Task 4.1	A procedure to output an even number
Task 4.2	A procedure to output an odd number
Task 4.3	A procedure to output the sum of two even numbers
Task 4.4	A procedure to output the sum of two odd numbers
Task 4.5	A procedure to output the sum of an even and an odd number

Table 6.2 (contd): Summary of main study tasks

6.3.4 The Role of the Researcher

The important role of the Researcher is discussed in Chapter Four (§ 4.4.2)

6.3.5 Collection of Data

Pupils were observed working at the computer-based tasks and field notes were made during these sessions. Each session was audio-taped and the recording later transcribed onto paper. This transcription constituted both a record of their work and a record of the researcher's intervention and the pupils' responses to those interventions. In addition, computer files of the pupils' programs were retained and printed out.

6.3.6 Phases of Analysis

Analysis of the data took place in three stages. Following the empirical sessions of the study, each session was transcribed in full using the audio tapes, the files of Logo procedures and the researcher's handwritten notes in conjunction. This done, detailed case studies of each pair's work were made, shaped by the emergent categories for analysis described earlier in this chapter. One such case study is reproduced complete in Appendix Three.

The eight case studies were then used as a pool of data from which to draw pertinent examples for the ideas discussed in Chapters Seven to Ten. Each example used within these chapters is prefaced by a brief description of its content (presented as the title in an underlined format).

Explanatory note on the presentation of data in the thesis

Much of the children's work involved the empirical verification of procedures. This frequently involved writing a procedure, testing it using the Calculator, rewriting the procedure and so on. Where this occurred and is relevant to the analysis, iterations of construction and verification are shown as follows:

op :u * 5 + 2

Input	*Output*
10	52

op :u * 5 + 1

Input	*Output*
5	26

op :u * 1

Input	*Output*
4	4

op :u * 1 + 1

What this shows is that the pupil constructed the procedure op ;u * 5 + 2 and tried the value 10 in the Calculator. This gave the output 52, displayed as an input/output pairing. Because this was incorrect, the pupils changed the procedure to ;u * 5 + 1 and tried the value 5 which produced the output 26. And so on. Such a method of presenting the data was felt to be more concise and easier to analyse for trends and patterns.

6.4 INITIAL ANALYSIS OF DATA

This section is an initial analysis of the data from the main study. Its purpose was to get a feel for the data, to provide categories for analysing the data and to refine the research questions.

Early on in the study, the expectation that pupils rarely worked within neatly-defined expressions of meaning was confirmed. There was an "untidiness" in most pupils' working patterns in that they were constantly re-expressing their ideas in different forms. As such it was difficult to define categories for purposes of analysis in terms of individual expressions of meaning; the overlap between expressions of meaning was so great that it was necessary to begin with a few broad delineations and to look for further categorisations within these. These broad delineations, chosen as a result of the initial phases of the study, were as follows:

- Construction and Verification
- General and Particular
- Equivalence
- LanguageTypes and Roles

6.4.1 Construction and Verification

Consideration of the data from the initial phases of the study suggested that a view of construction and verification as two distinct phases in the expression of generalisation and proof was misguided. In fact the relationship appeared far more complex and on the basis of Phases I and II it seemed more sensible to look on verification and construction as two elements of one greater process of *informed construction*. These two elements appeared bound together, one directly influencing and informing the other. If this was so, this complex relationship between acts of construction and empirical verification, and ways in which the computer mediated these two elements, provided an important window onto how pupils made connections between different ways of expressing meaning.

It also became apparent that construction/verification was not one "stage" in some linear model of expression and justification, but rather an element integral to the *whole process* of generalisation.

Turning first to the relationship between construction and verification, Sarah, in Task 4.2 (in which she was required to write a procedure which would output odd numbers) adopted an approach in which construction and verification were closely linked. She began with the common assumption that multiplying by three would produce an odd number and typed in the following procedure:

```
to usa :u
op :u * 3
end
```

She tried the following two values in this procedure:

Input	*Output*
2	6
3	9

Not satisfied (rightly) with this, she reconstructed her procedure a total of nine times. The verification appeared to inform her re-construction and after this first attempt she reconstructed her procedure whenever she got an even number as an output. She tried the following second line in the procedure **usa**:

```
op :u * 5
```

and then the following values in the new procedure (shown here with their accompanying outputs):

Input	*Output*
57	285
10	50

```
op :u * 5 + 2
```

Input	*Output*
10	52

```
op :u * 5 + 1
```

Input	*Output*
5	26

op :u * 1

Input	Output
4	4

op :u * 1 + 1

Input	Output
10	11
102	103
5	6

op :u * 2

Input	Output
5	10

op :u * 3

Input	Output
0	0
4	12

op :u * 3 + 1

Input	Output
3	10

op :u * 3 + 0

Input	Output
45	135
2	6

This example pointed to the formation of meaning through the complex dual process of procedure construction and verification. Sarah adopted a loose strategy: three times she tried fixing the multiplicative factor (as 5, 1 and 3) and then experimenting with the addition of various numbers (including zero). Unfortunately, in the case where the addition of an odd number would answer the question (i.e. when she chose two as the multiplicative factor) she failed to experiment and passed on to another coefficient of :u.

In this example verification *informed* construction to a certain extent; in others there appeared to be little or no connection between construction and verification, which nevertheless remained integral parts of one process. What other forms might this relationship take? It was felt necessary to obtain a full picture of this complex relationship and the computer's role in its mediation. So this relationship between construction and verification formed one significant category for analysis in Chapter Seven. Part and parcel of the construction-verification relationship was the connection between this *use* of structure, in verification and re-construction, and the *expression* of structure through the expression and justification of generality. In Chapter Seven I look at the part structuring elements had to play in construction and verification as a window onto connections between expressions of meaning.

A related feature was that of the relationship between use of the computer to verify procedures and the use of mental arithmetic for the same purpose. For example, Tom, in Task 2.1, verified his procedure using the computer and then checked this verification mentally. The task was given as follows:

> *Amaze Your Friends ....*
>
> *Pick a number between 1 and 10. Multiply it by 2. Add on the number you first thought of. Divide by 3. Divide by the number you first thought of.*
>
> *The Answer is .... ONE !!!*

He expressed this in terms of a Logo procedure:

```
to nat :v
op (:v * 2 + :v)  / 3 / :v
end
```

and checked that this worked by means of empirical testing of cases:

Input	*Output*
6	1
7	1
50	1
100	1

He followed up this testing on the computer by trying out some of the same values as above mentally, using his written procedure for guidance:

Tom: *Seven times two plus seven equals 19, divided by three. Eh?*
Researcher: *Are you happy with that?*
Tom: *Oh, I'll try a different number. Six, times two is twelve, plus six is 18, divided by three is six, divided by six is one. Oh. Fifty times two is 100, add 50 is 150, divided by three is 50, divided by 50 is one. It works.*

There appeared to be a tension here between expectation and verification; if the expectation was not "built into" the question or easily ascertained, then the provision of an expectation and the accompanying verification of results became a factor. Thus the relationship between mental calculation and computer verification was another interesting element in the wider relationship between construction and verification.

In Chapter Four[40], the intended shifting balance between empirical verification and justification was discussed. The preliminary phases of the study suggested that this shift did in fact take place; the important issue here was how this shifting balance

[40] Section 4.3.2

influenced the expression and justification of generality. As the balance shifted, did pupils perceive a *need* for justification? Would encountering problems in the execution of meaningful empirical verification lead to a *desire* for an apparently more satisfactory form of explanation? If this proved to be the case, might the creation of situations where there was a manufactured tension between empirical verification and justification create connections between these two expressions of meaning?

Thus the relationship between construction and empirical verification contained a number of interesting features. In summary, these included:

- term-by-term analysis - analysing the behaviour of a function;
- the computer's dual role in the verification process;
- visualisation of structure through procedure construction and verification;
- empirical verification as an integral part of the construction of procedures;
- verification versus justification - connection through tension?

6.4.2 General and Particular

A central and recurring theme in the pupils' work was the ever-shifting balance between particular and general. Take, for example, Tom's explanation of why the puzzle

> *Amaze Your Friends ....*
>
> *Pick a number between 1 and 10. Multiply it by 2. Add on the number you first thought of. Divide by 3. Divide by the number you first thought of.*
>
> *The Answer is .... ONE !!!*

output one. In the pre-task interview he had offered the following explanation:

Tom:	*Really, you'll get one because you're taking the number you first thought of and you're timesing it by two and adding on, which is just like timesing it by three, and then dividing it by three and then of course if you divide it by itself you'll get one.*

After constructing the procedure

```
to nat :v
op (:v * 2 + :v)  / 3 / :v
end
```

and verifying it empirically (see above), he gave a similar explanation, although with the small addition of a specific example (input = 5) at the end:

Tom:	*Really, by multiplying by two and adding on the first number you thought of, then it's just like multiplying it by three. And you're dividing it by three, so you've got the number you first started off with. And divided by the number you first started off with - five divided by five - and five divided by five is one.*

Fortunately (for research purposes) his partner David had not been paying attention so there was the opportunity for Tom to repeat his description. He apparently thought that James could not follow such a generalised description so used a specific example to illustrate it:

Researcher:	*Did you get that, James?*
David:	*I didn't hear it.*
Tom:	*I'll show you, I'll show you. It says pick a number between one and ten, times it by two. So, say it was seven, you'd get a seven putting on another seven. Add on the number you first thought of, which is seven - you're adding on another seven which is three sevens. So it's just like multiplying it by three. Divide by three and you get seven, divide by the number you first thought of - seven divided by seven - is one.*
David:	*I get that.*

This description was less generalised than the previous descriptions in that it made use throughout of a specific numerical example to illustrate how the procedure worked. However, this value whilst apparently specific (i.e. = 7) was not chosen for its specificness. Rather, it was chosen as a value typical of its class. The seven-ness of seven was of no interest here: it was the fact that seven was perceived to behave typically that made it Tom's choice. It was, in fact, an example chosen for the genus of seven.

This example is typical of the study in showing the considerable amount of interplay between general and particular expressions, mediated by the computer. The shift between general and particular appeared to be another relationship which might open a window onto ways in which pupils made connections between elements of the web, and this relationship forms a focus for analysis in Chapter Eight.

6.4.3 Equivalence

The design of the learning situation facilitated a variety of solutions to each task, and a variety of routes to those solutions. The following example shows two procedures which were equivalent. This simple example of equivalence shows Sophie's and Jenny's procedures for generating even numbers (Task 4.1): both explained how their procedures worked and they saw that either procedure would work equally well.

Sophie and Jenny wrote the following procedures, respectively[41].

```
to mmm :p
op :p * 2
end
```

[41] Details of construction and empirical verification and not included here.

```
to ilil :y
op :y + :y
end
```

In discussion they compared their procedures and concluded that they were equivalent:

Researcher:	*Can you explain, Sophie, what you have done and why you have done it like that.*
Sophie:	*The number that you put in is timesed by two, because whether it's odd or even, if you times it by two it will always come out as an even number.*
Researcher:	*Jenny, what have you done?*
Jenny:	*Well, I've done the same as Sophie, really, but I've put add the same number as you put in instead of timesing it by two.*
Researcher:	*Does it matter which you do?*
Sophie:	*No.*
Jenny:	*No.*

The contrast and comparison of procedures appeared to be fertile ground for the exploration and analysis of how pupils expressed and structured their ideas of generality. This perspective is explored in Chapter Nine.

6.4.4 Language Types and Roles

The role of language appeared to be central to the connection of disparate elements of the web, so that analysis of this role forms a major part of the data analysis chapters. A number of fruitful areas for investigation presented themselves.

Using the example of Tom and David working at Task 2.1[42], a variety of language types and uses may be discerned. In the pre-task interview, Tom described the puzzle's constant output of one as follows:

[42] Quoted above in Section 6.4.2.

Tom: *Really, you'll get one because you're taking the number you first thought of and you're timesing it by two and adding on, which is just like timesing it by three, and then dividing it by three and then of course if you divide it by itself you'll get one.*

This was a natural language explanation described entirely in general terms and was what might have been expected from the more mathematically experienced pupils at this stage, given that Tom had had experience of neither traditional paper and pencil algebra nor Logo programming as possible alternative means of expression. By the discussion stage at the end of Task 2.1, the web of meaning had grown. Tom had now encountered the natural language statement of the question and written his own version of this question statement in a formalised algebraic language (his Logo procedure). After procedures had been verified empirically and further discussion had occurred, the researcher asked Tom to consider what he had written in his procedure.

Researcher: *Can you explain that in terms of the procedure you've done on the computer, then?*
Tom: *[looking at the screen] Well, number times two plus number is three of the numbers. Divided by three is the number and divided by the number is one.*

If for "number" we were to read some standard algebraic symbol such as x, Tom had proved that the puzzle would always come to one in concise, formal, algebraic language. In fact this statement was in a kind of dialect of formal algebraic language, but it was a dialect of the language of construction. Thus Tom had expressed generality and engaged in a proving process in the language in which he had constructed the mathematical relationship (or at any rate in a dialect version of that language).

Weaving through this example are the multi-faceted roles of language, roles which appeared to influence and be influenced by all parts of the learning situation, and which were shaped by the mediating influence of the computer. Not least amongst these was the part discussion took in the creation of meaning, and as part of that discussion, the catalysts for engendering discussion, chief amongst which was the researcher.

The role of language, which is explored further in Chapter Ten, appeared to encompass at least the following aspects:

- a tool for the construction of mathematical relationships;
- a medium for the expression of mathematical ideas;
- a resource for discussion and analysis;
- a means of formalising mathematical relationships;
- a means of communication.

These four areas of construction and verification, general and particular, equivalence and language types and roles provide foci for analysis.

6.5 RESEARCH QUESTIONS AND THE REMAINDER OF THE THESIS

The aims of the thesis, as described in Chapter One, were defined as follows:

> i) to investigate ways in which children expressed generality using the programming medium;
>
> ii) to investigate connections between different expressions of generality made in different modalities and their role in children's creation of mathematical meaning;
>
> iii) to investigate the role of connections between expressions of generality in children's attempts to justify their constructed generalisations.

From the initial analysis of data from the main study (§ 6.4), a number of research questions may be framed which address these aims.

The role of empirical verification featured prominently in the children's work reviewed, in terms of its relationship with the construction process and with analysis of a function's behaviour. This empirical verification, an accessible and prominent feature of the microworld, appeared to underlie pupils' ability to construct procedures and to express generality, and provided an insight into how pupils created their unique expressions of meaning. Tensions between empirical verification and the use of justification, underpinned by analysis and even deduction, also appeared to offer an insight into how pupils constructed connections between expressions of meaning. Two related research questions emerged:

- How did the use of empirical verification mediate the expression of generality through the construction of formalised procedures?
- How did tensions in the balance between empirical verification and justification mediate the creation of connections between expressions of meaning?

These questions are addressed in Chapter Seven under four organisational themes:

i) the act of symbolisation - the role of construction in creating meaning for generality;

ii) analysis of structure - how this was facilitated by the programming medium;

iii) construction strategies - detailed examination of different ways in which pupils constructed their procedures, focusing particularly on their use of empirical verification;

iv) perceptions of verification - pupils' perceptions and expectations of the role of empirical verification and the relationship between verification and justification.

A feature of children's work in the initial analysis of the main study was the use of generic examples, so a resultant feature of the research was to look at ways in which children appeared to create and use generic examples in order to provide a bridge between specific and general. The related research question was:

- How did the computer mediate the creation and use of generic examples as a bridge between the specific and the general?

This question is considered in Chapter Eight under the following themes:

i) semi-generalisation - a form of webbing already identified as effecting construction of generality;
ii) generic structuring - a term introduced to describe ways in which the structure of a procedure was explored through the use of generic examples.

Children's exploration of the equivalency of procedures appeared to provide a focus for examining the structure of mathematical relationships and again provided an insight into children shaping their own meaning. This focus provided the following research question:

- How did the contrast and comparison of equivalent procedures shape the exploration of structure within mathematical relationships?

This question is addressed in Chapter Nine through consideration of ways in which pupils used equivalence to explore mathematical relationships.

Finally the use and role of language, be it natural, algebraic or computer-programming, and be its purpose one of construction, explanation or communication, was a dominant theme throughout. Of particular interest was the use of language which was in some way mediated by the setting and a particular focus of the research was this mediating role. This prompted the following research question:

- How and to what extent did formalised language act as a means of connection or dislocation between expressions of meaning?

This question is explored in Chapter Ten under the following organisational themes:

i) comparing the use of natural and formal language - formal language as opposed to natural language used as a common means of expression and communication within the context of the study;
ii) naturalised formalism - a term introduced to describe how mathematical relationships were frequently shaped by the structure of formal procedures but expressed in natural language.

Thus the following four chapters consist of analysis of the main study data. In these chapters I draw out ways in which children express meaning for generality and justification and identify connections they make between their expressions of meaning.

Chapter Eleven is the concluding chapter and in it I draw together strands from the analysis of the data in an attempt to answer the research questions. In addition I make some implications for teaching and learning and suggest areas for further research.

CHAPTER SEVEN: CONSTRUCTION AND VERIFICATION

7.1 INTRODUCTION

Chapter Seven is the first of four chapters which examine the study data and from it develop themes relating to the research questions. In this chapter, the relationship between construction and empirical verification, which appeared central to the process of generalisation and proof within the learning situations defined in this study, is examined in detail. In Chapter Six I outlined an example where verification appeared to inform construction, albeit to a limited extent; here I explore the relationship between construction and verification, so obtaining a full picture of this complex relationship and, importantly, the computer's role in its mediation. As suggested in Chapter Four, the connection between the *use* of structure, in verification and re-construction, and the *expression* of structure through the expression of generality and proof is an integral part of this relationship and one which I explore. The part structuring elements have to play in construction and verification as a window onto connections between expressions of meaning is also an important focus of this section. A related feature is that of the relationship between use of the computer to verify procedures and the use of mental arithmetic for the same purpose.

The chapter begins with two important areas which emerged from phase III of the study as providing connections between expressions of meaning. These were acts of symbolisation and the analysis of structure. Each area is discussed by means of looking closely at data from the study and factors which influenced their execution are identified. The third section focuses on construction strategies employed by the pupils and the final section looks at pupils' perceptions of verification, discussing pupils' use of empirical verification in conjunction with their own expectations of a function's behaviour, and the balance between empirical verification and justification.

7.2 THE ACT OF SYMBOLISATION

In this section I look in detail at two examples where the act of symbolisation was central to pupils' ability to express their mathematical ideas and show how it connected two diverse expressions of meaning for generality.

Example: Sophie and Jenny finding the visual structure of their procedures helpful in shaping their description and in clarifying the mathematical relationship contained therein - 50 (F 2.1)[43]

In Task 2.1, Sophie and Jenny both wrote similar procedures;

```
to Jenny :h
op (:h * 2 + :h) / 3 / :h
end
```

These had been constructed by means of a series of attempts which had been tested empirically for various values of input using the Calculator. For the moment the details of this act of construction will be skimmed over; they are taken up again in Section 7.4. In the discussion that follows verification, the structure of the procedure provided a focus;

Researcher:	*Can you say why it always comes out as one?*
Jenny:	*Um, because you...*

[43] This method of numbering was adopted in the original transcription of sessions from audio tape and from the researcher's notes.

Sophie:	*Because the number you get has always got either a two or a three in it and a two and a three, you times it by two and divide it by three.*
Researcher:	*How do you mean it's always got a two or a three?*
Sophie:	*When you... Say the number was ten. When you times it by two which is 20, it's always got a two in it*[44] *and then you add ten again which comes to 30. So the first number's always got a two in it and the second number's always got a three in it*[45].
Researcher:	*Right, I see what you mean.*
Jenny:	*So you're timesing it by three and then you're dividing it by three so it comes out.*
Researcher:	*But it doesn't come out as the same number, it comes out as one.*
Jenny:	*Well then you divide it by its own number so that makes it one.*
Researcher:	*Had you worked that out before or does looking at the Logo procedure help you to work that out?*
Sophie:	*Yes, the Logo procedure.*
Researcher:	*How?*
Sophie:	*It's more spaced out so I can see it clearer.*
Jenny:	*It reminds you of what the sum is.*

This extract provides two interesting insights into Sophie's thinking:

Firstly, the visual appeal of the procedure; it was spaced in such a way (structured) as to allow Sophie to consider, identify and express *how* it functioned. Sophie was referring to the symbolic procedure structure rather than the natural language structure of the written question she had worked from initially. One of the functions of mathematical symbolism is to present relationships in as clear a way as possible, and it is this function that Sophie had identified here. Although in this example she may have expressed her ideas through use of Logo symbolism (i.e. using terms such as ":h times two"), from what she said it would seem that her description of what the function did is largely shaped by consideration of the symbolic procedure rather than of the natural language statement. Her descriptions were in a kind of formalised natural language which commonly served as an intermediary language between the

[44] i.e. divisible by two.

[45] i.e. divisible by three.

natural language of the question and formal algebraic expression[46].

Secondly, there is Jenny's statement that it provided a fixed, clear, concise record of "what the sum is". This, of course, is another function of mathematical symbolism. Because Jenny had constructed the symbolism herself, it was richly endowed with meaning. For her - for both of them - the translation into symbolic form was a *helpful* act, creating clarity. This makes an interesting comparison with conceptions of formal symbolism which are characterised by mistrust and misunderstanding. Use of mathematical symbolism which is accompanied by little experience of the associated mathematics would seem to create this mistrust and misunderstanding; the learning situation experienced by Jenny and Sophie where an act of construction used the symbolism as an integral part of a "concrete" mathematical experience, appeared to create not just an *acceptance* of symbolic mathematical expression, but a perceived *need* for it.

The act of symbolisation for some students created a connection between a problem expressed in natural language and the expression of that problem in formal algebraic language. There was a connection, then, between the visual nature of the algebraic function and use and expression of its structure. What was it that caused pupils to become aware of the visual structure of a procedure? It would appear to have been the term-by-term construction of the procedure and the computer-centred verification of that procedure which in turn required (or encouraged) term-by-term analysis of the function's behaviour. The power of this act of symbolisation - the building of procedures on the computer screen - extended beyond simply forging links between two expressions of a mathematical statement to the transformation of symbols into a new means of expression.

[46] This idea of an intermediary language is developed in Chapter 10, § 10.3.

Example: Tom manipulating the relationships defined by a procedure in order to express a proof - 39 (H 4.4)

In the work carried out by Group H in Task 4.4 the researcher steered the discussion towards consideration of the output from the sum of any two odd numbers. At first the two boys established that the sum of two odd numbers is an even number by drawing on their own experience of summing two such numbers in the verification phase of procedure construction:

Researcher:	*What sort of number do you always get?*
David:	*An odd number.*
Tom:	*Is that right?*
Researcher:	*Go back and have a look [at the results of verification on the screen].*
Researcher:	*It's odd plus odd, so what does that come to?*
David:	*Odd plus odd has to be odd.*
Tom:	*No. Nine plus nine is eighteen.*
David:	*Oh yes.*
Tom:	*An odd plus an odd is an even.*

The researcher then asked them to explain this, focusing their attention on the procedure which Tom had already constructed:

```
to bob :k :j
op :k * 2 + 1 + :j * 2 + 1
end
```

Researcher:	*Why is it always an even number? Can you say by looking at your procedure?*
Tom:	*I know. An odd number add an odd number. An odd number is an even number add one - plus one. Add another even number plus one equals another odd number, but you have two even numbers, but you still have the two little ones left over and that makes another even number.*
Researcher:	*So you end up with how many even numbers, then? Three even numbers. Do you understand that, David?*
David:	*I wasn't really listening.*

David's apparent lack of concentration (or his disinclination to admit to a lack of understanding) was fortunate, as it gave Tom the opportunity to repeat his explanation. In fact he restructured it:

Tom:	*David, I'll show you [pointing at the procedure]: to make an odd number you need an even number plus one. So what you do is you add two even numbers, but because they're odd numbers you have to add the one - the two little ones still left over. So from an even number you count one on to an odd number and another one on to an even number. See?*
David:	*Yes.*

The first explanation might be written more formally by saying

If $k * 2$ is even $\forall k \in \mathbf{Z}$,

then $k * 2 + 1$ is odd $\forall k \in \mathbf{Z}$.

Thus the sum of two odd numbers may be written

$$(x * 2 + 1) + (y * 2 + 1),$$

which can be written

$$(x * 2) + (y * 2) + 1 + 1.$$

Since $1 + 1$ is even, this makes three even numbers and the sum of three even numbers is another even number.

Tom's "repeat" explanation to David was markedly different. Expressed in a similar format, it might read

If $(k * 2)$ is even $\forall k \in \mathbf{Z}$,

then $(k * 2) + 1$ is odd $\forall k \in \mathbf{Z}$.

Thus the sum of two odd numbers can be written

$$(x * 2) + (y * 2) + 1 + 1.$$

Since $(x * 2) + (y * 2)$ is even,

$(x * 2) + (y * 2) + 1$ is odd and

$(x * 2) + (y * 2) + 1 + 1$ is even.

In both explanations Tom established the evenness of even + even (although from different starting points). In the first explanation he then focused on the fact that 1 + 1 = 2, an even number. In the second, he began with the fact that even + even is even and then counted on one to make an odd number and another to make an even number (interestingly using the "alternate" idea of even and odd numbers which David described in his pre-test interview: this explanation must have resonated with David).

How had Tom developed the facility to manipulate the structure of this procedure? To answer this question it is necessary to go back to Tasks 4.1 and 4.2 through which he had developed an understanding of oddness and evenness. In Task 4.1 (the construction of a procedure which outputs an even number) he constructed the following procedure:

```
to reg :w
op :w * 2
end
```

This was done almost spontaneously: pausing only for thought, Tom typed the procedure straight onto the computer screen. He then tested the procedure with an input of 7 in the Calculator which gave an output of 14. In all he tried the following five values:

Input	*Output*
7	14
31	62
4	8
16	32
5	10

Following this verification he declared

Tom: *All you have to do is times it by two: it couldn't be simpler.*

The researcher asked him why this was so:

Tom: *It's easy, though, because every even number is divided by two - can be divided by two. That means you can divide - you can times every single number in the whole world by two, which means that every single number in the whole world you can make an even number by timesing it by two. Because everything is divided by two. Is nought always divided by two? I know what I'm saying.*

Researcher: *Can you say that again? Why does your procedure make even numbers?*

Tom: *Because even numbers are divided by two, then that means that every single number times two will make an even number, because it would be like dividing an even number by two.*

Although Tom may have known this before (and the evidence from his pre-task interview indicates this to be the case) he appeared excited by demonstrating and expressing his knowledge through procedure construction and discussion. He was re-expressing his ideas in a manner influenced by the form of the procedure: it is in this way that this expression differs from that of the pre-task interview. He also constructed and verified a procedure in a similar manner in Task 4.2 (the construction of a procedure which outputs an odd number). Using the following procedure

```
to sid :s
op :s * 2 + 1
end
```

he tried six values for the input in the Calculator

Input	*Output*
5	11
7	15
1	3
88	177
999	1999
0	1

On being asked to construct a second procedure which would serve the same function he wrote

```
to jim :l
op :l * 2 + 243
end
```

and tried the following values:

Input	*Output*
5	253
4	251
400	1043

Thus Tom demonstrated his understanding of the generality of his procedure. He commented:

Tom: *So as long as it's got an odd number on the end it's alright.*

Thus in Tasks 4.1 and 4.2 Tom had implicitly established the following Lemmas (here re-expressed):

i $\forall\, k \in \mathbf{Z}$, $(k * 2)$ is even;

ii $\forall\, k \in \mathbf{Z}$, $(k * 2) + 1$ is odd;

From these, in Task 4.4 he had constructed and proved the Theorem

$$(x * 2) + 1 + (y * 2) + 1 \text{ is even } \forall\ x, y \in \mathbb{Z}.$$

He proved this theorem through the manipulation of his procedure. He did not manipulate the *symbols* as such, as might be the case in a traditional paper and pencil example, but manipulated the relationships defined by the symbols in natural language (in what I shall describe as "naturalised formalism"[47]). Here the symbols had acquired a meaning different from the traditional symbols of paper and pencil algebra in two ways: experience of constructing and verifying the Lemmas had transformed the symbols into a *means of expression*; and the verification process had endowed the symbols with a history of several specific numerical examples, although they remained a generalised expression of a mathematical relationship. Thus, using the symbols as both a means of expression and as abstractions of a recently concrete experience, Tom was able to restructure his procedure (i.e. manipulate its symbols) into a new form from the structure of which its proof became clear. This, of course, is the process of formal proof. The difference here is that the restructuring of the theorem statement necessary to its proof was facilitated by the expressive quality of the abstract symbols, albeit expressed in "naturalised formalism" rather than in the language of construction itself.

From these two examples, the act of symbolisation emerges as an important connection between mathematical statements expressed in very different forms (namely natural language and formal algebra) and between formal mathematical statements and expressions of generality, the starting points for which are those formal mathematical statements.

[47] See Chapter 10, § 10.3

An important influence on an ability to symbolise appeared to be the relationship between construction and verification, a relationship which was prominent in the computer-based learning situation designed for this study. Before looking at this important relationship in depth I want to look at some examples where pupils explored the structure of the procedures they had constructed and how, through this exploration, they made generalisations about their procedures.

7.3 ANALYSIS OF STRUCTURE

In this section I describe three examples which show how children's analysis of their own procedure construction was influenced by the process of construction supported by empirical verification, by the tabular presentation of verification data and by term-by-term verification.

<u>Example: Edward's symbolism acquiring meaning through iterative construction and empirical verification - 105 (G 3.3)</u>

Task 3.3 involved writing a procedure which multiplied the input by the integer immediately preceding it. In his attempt at this task, Edward wrote a procedure which was correct in all respects apart from his use of brackets:

```
to jerrrr :j
op (:j * :j) - 1
end
```

Edward attempted to verify this empirically using the Calculator and then continued to attempt to position the brackets correctly. At each step the verification informed his next adjustment to the procedure construction. So after trying an input of 8, which resulted in an output of 63

Input	*Output*
8	63

Edward adjusted his procedure as follows

op (:j *) :j - 1

and continued testing values and adjusting his procedure:

Input	*Output*
8	-

op (:j * :j) - 1

Input	*Output*
2	3

op :j * (:j - 1)

Input	*Output*
8	56
2	2

When Edward was asked to explain his procedure he did so in a way directly influenced by the preceding construction and verification:

Edward: *It's :j times and then the second part is say, um, :j take one is to do with that number, so it's :j timesed by :j take one, but ... it's take one from the second :j so that instead of it being eight times eight, it's eight times seven, because there's one taken away from the second :j.*

Edward's explanation of the procedure referred directly to its structure, a structure which had become clearly defined through the process of positioning brackets. It is through that process that he had come to appreciate the difference between

$$(:j * :j) - 1$$

and

$$:j * (:j - 1)$$

and why one modelled the task and the other did not. The learning situation required pupils to formalise mathematical relationships in the form of a Logo procedure: the construction and verification processes in this example shaped their understanding of the structure of that procedure with the result that they were able to discern the difference small adjustments to the symbolism could make. The effect of manipulating symbols could be seen clearly through the empirical testing of different values of the input.

Edward seemed to have learnt through *using* the learning situation that

$$(j * j) - 1 \neq j\,(j - 1)$$

For Edward the symbolism appeared to have acquired layers of meaning through his experience of procedure construction[48]. Term-by-term analysis of the procedure and systematic restructuring, supported by verification, appeared to promote his ability to generalise, to analyse and to justify.

[48] It is interesting to speculate about the possibility of introducing standard algebraic notation and demonstrating this inequality: such a step might well follow because the symbolism involved is the same language as that with which he has constructed his procedure and the result is one which he has just demonstrated empirically.

<u>Example: Ben's justification of a mathematical relationship influenced by the visual qualities of the verification - 45 (D 4.5)</u>

In Task 4.5 (summation of an odd and an even number), Ben constructed the following procedure:

```
to mixed :a :b
op (:a * 2) + (:b * 2 + 1)
end
```

and tried the following input values using the Calculator:

Inputs		*Output*
0	1	3
1	0	3
1	4	11
4	1	11

The interest here clearly lies in Ben's choice of pairs: 0, 1 and 1, 0; 1, 4 and 4, 1. When the researcher asked the pupils to explain how their procedures worked, Ben's response was shaped by his verification of the procedure:

Ben: *Sir, sir, if you put two numbers ... whatever order you put two numbers in, the output will always be the same.*

Ben was clearly excited by this discovery, the discovery of the Associative Law of Addition in action. He was saying that a and b are interchangeable, that

```
op (:a * 2) + (:b * 2 + 1)
```

and

```
op (:b * 2) + (:a * 2 + 1)
```

are the same thing. The discussion continued:

PAGE MISSING IN ORIGINAL

This analysis of the structure of the procedure would appear to have shaped the proof of why the procedure always output an odd number. The proof was restated as a generic example:

Researcher:	*Yes. Can you say whether that always comes to odd or even.*
Ben:	*Odd.*
Researcher:	*An even number plus an odd number.*
Ben:	*Yes.*
Researcher:	*Why?*
Ben:	*Well, as I said before, every odd number is an even number add one, so it would be two even numbers together make an even number and then add one makes an odd number.*
Matthew:	*Yes.*
Researcher:	*Good. Can you be a little more explicit? Can you say a little more?*
Ben:	*Oh, I'll give you an example. If you had four add five, it comes to nine which is an odd number, because it's the same as four add four - you add two even numbers together is an even number - eight, and add on one.*
Researcher:	*You said just now that you've got an even number plus an even number makes another even number plus one makes an odd number. Can you just say what are the even numbers there.*
Ben:	*Oh, :a * 2 and :b * 2.*

This example brings together a number of different threads. Firstly, Ben's verification of his procedure demonstrated the fact that the order in which the inputs were placed made no difference to the output: Ben's investigation of the behaviour of the inputs appeared to be influenced by the tabular output of the Calculator, making the result particularly clear to him. Secondly, discussion and analysis of the structure of Ben's procedure highlighted the relationships which were needed for the ultimate expression of proof. Ben's construction and verification highlighted a mathematical property of his procedure which he investigated further through analysis of his procedure and which he finally proved.

<u>Example: Rachael's and Hannah's rehearsal of the behaviour of the function under specific values appearing to highlight the structure of the function - 3 (B 2.1)</u>

In discussion of why the procedure in Task 2.1 produced a constant output, Rachael and Hannah were not able to explain immediately, any more than they had been in the pre-task interview. For them, simply working with the computer had not been enough to enable them to make abstract mathematical statements. But from vague descriptions outlining some sort of intuitive theory:

Rachael:	*Well, is it because you're adding loads of numbers and then you see how many numbers you take away and then you take away exactly the same numbers in a different sum.*
Hannah:	*Every number you start with, you just sort of add it on and then you just take it all away again.*

the structure of the Logo procedure (and the focus that structure provides for their verbalising) and careful intervention from the researcher provided sufficient support to build up a clear expression of why the procedure produced an output of one every time.

```
to freda :y
op (:y * 2 + :y) / 3 / :y
end
```

Researcher:	*Why does this always come to one?*
Rachael:	*Do you choose the number, and add a certain number and then take away the number you first started with to get back to that number?*
Researcher:	*Try a number in it.*
Rachael:	*[pointing at the procedure] Say three. Three times two is six. Add :y.*
Hannah:	*Nine.*
Rachael:	*Then divided by three.*
Hannah:	*Three.*
Rachael:	*Divided by three is one.*
Hannah:	*Try five.*
Rachael:	*Five times two add five is fifteen. Divided by three is three...*
Hannah:	*Because whenever you divide it always makes three.*

Rachael:	*Six times two: twelve...*
Hannah:	*When you're there [after /3] you'll always have three.*
Rachael:	*It always comes to three.*
Hannah:	*Yes.*

The girls had found a rule, so the researcher asked them to demonstrate its validity:

Researcher:	*Give me an example, then. Take five.*
Rachael:	*O.K. Five timesed by two is ten, add the number you started with.*
Hannah:	*Five.*
Rachael:	*Fifteen. Divided by three is three.*
Hannah:	*Three.*
Researcher:	*Divided by three is three?*
Rachael:	*Oh, five.*

Such was the pupils' strength of belief in the rule they had devised that they did not appear even to notice this simple error. In order to encourage the girls to analyse their procedure, the researcher focused their attention on an important part of that procedure:

Researcher:	*Have a look at this first bit: :y * 2 + :y. If you times a number by two...*
Hannah:	*You're doubling it.*
Researcher:	*You're doubling it and add on the number again. You multiply by two and add on the number you first thought of.*
Hannah:	*Is it the same as timesing by three?*
Researcher:	*Can you go any further?*
Hannah:	*And then if you divide it again you get back to your number.*
Rachael:	*And then if you divide by your number you get one.*

In this example the structure of the procedure seemed to become apparent through rehearsal of the behaviour of the procedure under specific values of the input.

Looking now at the three preceding examples, chosen to illustrate how pupils explored the structure of their procedures, it may be seen from the first example that analysis of the procedure structure appeared to be facilitated by the process of construction/ verification in which Edward engaged. The characteristic of that process here was the iterative nature of the (re)construction and verification which proceeds throughout the extract. In the second example it was also the construction/verification process which facilitated analysis of procedure structure, but here it appeared to be the tabular format of the verification which was significant in shaping Ben's analysis of the procedure. In the third example the feature of verification which predominated was a term-by-term approach, where use of the computer did not appear to feature highly.

Thus exploration of structure, an important connection between formal mathematical expression and the expression of generality about that mathematical statement, appeared to be influenced by the construction and verification of procedures, and particularly

i) iterative (re)construction and verification
ii) the tabular presentation of verification
iii) term-by-term verification

7.4 CONSTRUCTION STRATEGIES

Given that the construction/verification relationship appeared to be a major influence both on the ability to symbolise and on the ability to explore the structure of procedures (two connections between expressions of meaning), I want in this section to look more closely at the construction/verification process, with particular emphasis on construction strategies employed by pupils, the part empirical verification played in them and the ways in which the computer was used to mediate these processes. Construction strategies were characterised by both the degree and the nature of interplay between the processes of construction and verification, and by the value which information derived from the verification process was accorded. Two broad types of construction strategy emerged: those based on a process of trial and error, where pupils used the computer to construct successive iterations of a procedure; and prior mental construction, where a more or less correct procedure was worked out mentally and the computer used merely as a means of recording the procedure, perhaps with some element of verification. Both these broad categories included examples where empirical verification informed re-construction of a procedure, and examples where it did not.

7.4.1 Trial and Error

Perhaps the highest degree of interplay was evident where pupils adopted a trial and error approach. In some cases verification did not inform subsequent procedure construction, beyond simply demonstrating whether a procedure gave a correct or incorrect output. The computer was used as a convenient means of evaluating inputs. Frequently trial and error was structured by a template (appropriate or otherwise) determined by the experience a particular pupil had had with questions of a similar type.

<u>Example: Sarah using a template to model her procedure construction but failing to inform subsequent empirical re-construction through verification - 95 (C 1.5)</u>

Sarah's strategy in Task 1.5 was to select one input value and simply to attempt to generate that number within the template she had accepted as the norm. She began by typing into the computer the following procedure:

```
to a :t
op :t * 2 + 4
end
```

This was because the previous example, and the example initially given to pupils by the researcher, was of the form

```
op :x * 2 + c
```

She chose 10 as her input value and stuck to this value throughout the Task. Empirical testing of her first procedure gave an output of 24:

Input	*Output*
10	24

The succession of procedures she wrote, and more particularly the verifications she attempted, apparently failed to inform revisions of the procedure. The following sequence gives some idea of this seemingly random choice of procedure re-construction, each one constructed within the more generalised template

op :x * a ± c

So, multiplying by two gave twenty something, multiplying by three gave thirty something and multiplying by six gave sixty something:

```
to a :t
op :t * 2 + 8
end
```

Input	*Output*
10	28

```
to a :t
op :t * 3 + 6
end
```

Input	*Output*
10	36

```
to a :t
op :t * 6 + 4
end
```

Input	*Output*
10	64

Yet it did not occur to her that multiplying by four would give her forty something (the goal). Instead, she subsequently latched onto a multiplicative factor of six and attempted to "close in" from there, still failing to reach the correct solution[49].

In some examples, using a template in this way allowed pupils to stumble on correct solutions: not by complete chance, but by a kind of chance which existed within the parameters of the template. A benefit of working with the computer here was the ease of verification it afforded: construction and verification become integrated into a greater act of construction, that is verification became part of the construction process and not an extra phase through which pupils had to pass[50]. The interplay between construction and verification would also appear to have shaped pupils' understanding of the structure of procedures. The constant rehearsal of values in a procedure and the search for bugs in that procedure forced pupils to examine each term and its relationship with other terms, and to analyse this relationship in terms of how it affected the behaviour of values of the input.

However, use of a template also had drawbacks. The template structure could actually hinder the natural expression of pupils and give them the impression that *all* procedures must conform to the format defined by the template. This was unfortunate because one of the strengths of the learning situation was the non-uniqueness of solutions and, more importantly, routes to solutions[51]. The template, whilst supportive, might also constrain pupils into conforming to a pattern which might not be appropriate and which they might not feel to be entirely appropriate. This was frequently the case where pupils misunderstood a question or had no clear expectation of the output from any input they tried. It is important to note that

[49] It seems unhelpful to quote the example in full here.

[50] Although in this particular category of construction, verification did not inform procedure construction.

[51] See, for example, discussion on Equivalent Procedures in Chapter 9.

successful and effective use of the computer required an expectation of likely outputs from the inputs chosen[52] .

The degree of interplay between construction and verification was also high where each successive attempt at a procedure was *informed by the construction and empirical testing of the previous attempt.* Sarah's attempts (above) using trial and error were largely random: Sarah failed to adjust her strategy in the light of information which could be gleaned from the successive testing of her procedures. Informed trial and error was characterised by the interplay between two elements: construction and verification. The validity of the construction was confirmed or rejected by the verification process; the verification process itself informed the next phase of construction. In fact the two elements were inextricably linked, attention passing continuously from one to the other until the process was successfully verified and the construction phase was at an end. Informed trial and error was essentially an iterative process: it was characterised by an initial procedure written through guess work or based on a template, with each subsequent iteration informed by the previous verification.

<u>Example: Kimberley's re-constructions informed by empirical verification - 89 (A 1.6)</u>

In task 1.6 Kimberley made what appeared to be a guess at a procedure using the standard template:

```
to monty :x
op :x * 5 + 7
end
```

It seems clear that the initial procedure was largely guess-work and not worked out carefully in her head, as even her first input value using the Calculator (x = 3) did not give the expected output:

[52] The question of expectation of output is discussed in § 7.5.2

Input	*Output*
2	22

Since this had given the output that should be associated with :x = 4, she tried that input[53] :

Input	*Output*
4	27

She then reconstructed her procedure and tested the same values:

```
to monty :x
op :x * 4 + 7
end
```

Input	*Output*
3	19
4	23

She wanted the output when :x = 4 to be one less, so adjusted her procedure accordingly:

```
to monty :x
op :x * 4 + 6
end
```

Input	*Output*
4	22

This worked, so she tried another value

[53] In fact Kimberley bases her re-constructions first round the testing of :x = 3, then :x = 4.

Input	*Output*
5	26

which did not. At this point, unfortunately, she gave up. However, each reconstruction was informed by the foregoing verification: there is a feeling of gradually homing in on the correct procedure, even though in this case it was never actually reached. Kimberley referred to this feeling of homing in on the answer when she said later on:

Kimberley: *I was getting closer but I couldn't work it out.*

7.4.2 Prior Mental Construction

Construction strategies in this category differed from what has been described above as an iterative trial and error approach, in that the pupil thought of a likely procedure first mentally, then typed it onto the screen and then, either, by means of verification of that procedure, adjusted it until it was correct, or simply made a "token" verification and no adjustment.

The important word here is likely, the intention being to convey some element of deliberate forethought to the initial procedure. Trial and error approaches (whether informed or not) were characterised by the guessed nature of the initial procedure construction, which was often based on a template. The key elements of this approach, then, were initial mental construction of a procedure and amendments to that procedure informed by verification.

<u>Example: Kathryn constructing a procedure mentally *before* using the computer - 97 (C 2.2)</u>

In Task 2.2 (the second of the Amaze Your Friends puzzles), Kathryn began by clarifying the meaning of the question and making her first attempt:

Kathryn:	*What does it mean by double?*
Researcher:	*What do you normally mean when you talk about doubling something?*
Kathryn:	*It means add another one onto it.*
Sarah:	*Add one on.*
	[Pause]
Kathryn:	*I know.*

She wrote down the following procedure quickly, fluently and silently as if typing out something already clear in her mind:

```
to fgtyuight :v
op (:v * 2 * 2) - 4 + 1
end
```

This done, she verified her procedure using the Calculator with the following input value and corresponding output

Input	*Output*
4	13

and checked through the procedure again mentally to see why it had not worked. She re-read the question, apparently saw her omission and amended the procedure (interestingly removing the brackets first, as if to concentrate on one thing at a time):

```
op :v * 2 * 2 - 4 / 4 + 1
```

She tried to verify this on the computer:

Input	*Output*
4	16

Under the (correct) impression that all the terms were now correct she made another attempt at introducing brackets:

```
op (:v * 2 * 2 ) - 4 / 4 + 1
```

and verified this

Input	*Output*
4	16

Informed by this attempt at verification, her second attempt yielded success

```
op (:v * 2 * 2  - 4 ) / 4 + 1
```

and she verified the now correct procedure:

Input	*Output*
4	4
6	6
6780965	6780965
1	1

This example shows how a pupil built up the procedure on the computer screen from a starting point which had been thought through mentally without using the computer. The computer's role here was first as a means of jotting down her idea, and second as a means of testing that idea. Importantly, the computer would still run the procedure even in its partially correct state. This meant that the pupil could "home in" on the correct procedure by iterative verification/reconstruction. The verification of these partially correct procedures informed construction of the pupil's next version of the procedure.

Example: Hannah and Rachael discussing use of the computer for constructing procedures - 94 (B 1.2)

Researcher:	*Do you think that doing it with the computer makes it any easier?*
Hannah:	*Yes.*
Rachael:	*Yes.*
Researcher:	*Why do you think that is?*
Hannah:	*Because the computer does all the work for you!*
Researcher:	*What does it do for you, then?*
Rachael:	*Well, it does the sum for you.*
Hannah:	*It can't always be right, I suppose...*
Rachael:	*So you've got to work it out before.*
Researcher:	*Can you think what it is that makes it easier to do on the computer than in your head?*
Hannah:	*It makes me more confident. You can like type stuff down but in your head you've just got to try and think of it.*
Rachael:	*I find it easier to write stuff down so you can actually read it.*
Researcher:	*Why do you think writing down helps you?*
Rachael:	*Because you can see.*
Hannah:	*Because you can see the numbers.*

Given the fact that in this case there was little interaction between pupil and computer in the sense of a construction-verification-adjustment-verification sequence (not reproduced here), these answers are interesting. The pupils described the process of verification and recognised that the computer had limitations: although it would do the work for them, they had first to know what answer to expect. The second half of this exchange is of particular interest. Hannah put forward the notion of a medium which was able to extend her working memory: a tool which allowed her to fix her ideas, if only temporarily, on the screen. Rachael was making a similar point when she described being able to read rather than mentally envisage her ideas. Tied in with this was the ability to see inputs and their accompanying outputs. Rachael said, "because you can see": perhaps by this she meant because you can detach yourself from the mechanics of calculating the problem and look at its structure clearly, free from the burden of mental calculation.

For some pupils, verification only appeared to be carried out because it was perceived to be part of the game: a ritual which had to be gone through to satisfy the researcher. Verification was largely nominal and inconsequential.

Example: Ben's non-use of the computer for purposes of construction and verification - 99 (D1.5)

Ben was asked how he had constructed his procedure in Task 1.5:

```
to five :e
op :e * 4 + 1
end
```

Researcher:	*Can you say how you worked your procedure out?*
Matthew:	*If you times one by a particular number, like one you get...*
Ben:	*When I saw that [the pair of values (0, 1)] I instantly thought it was times any number add on one and I saw that [the pair of values (1, 5)] and thought times four, add on one.*
Researcher:	*And how did you check those?*
Ben:	*I checked one or two of them in my head and then I did them on the computer.*
Researcher:	*Can you just describe what the procedure does.*
Ben:	*Times four add one, really.*
Researcher:	*Can you describe what the different bits do in it?*
Ben:	*Well, all these here are odd numbers.*
Matthew:	*So it's probably going to be an odd number to times.*
Ben:	*They're going up in fours, except starting on one.*

Ben's ability to work out simple functions was exceptional amongst members of the eight groups of the study, but he neatly described the process many of the pupils went through who were able to construct a rule mentally before attempting anything on the computer. The writing of the procedure on the computer screen was here a *translation* from a rule he had constructed in natural language: the computer had no role in the construction of the rule; it was used merely to record the translation and, almost as an after thought, a few values were tested. They were not tested to check that the rule was correct, but to check that the translation of the rule was correct.

Using the computer, for Ben at this stage, was an additional part of the task: one which actually added to the complexity of the task.

The role of empirical verification in these examples varies. The next section looks briefly at pupils' perceptions of the role of verification and then focuses on two verification-related issues: expectation of result and the shifting balance between verification and justification.

7.5 PERCEPTIONS OF VERIFICATION

7.5.1 Pupil Perceptions of Verification

Verification was a complex and widely varying process. The evolution of the Calculator to provide pupils with a means of verifying procedures easily and quickly and to present inputs and outputs in a clear, tabular format which was nevertheless under the control of pupils, has been described in Chapter Five[54]. However, pupils' perceptions of what constituted adequate verification varied with task and time: different tasks produced different degrees of empirical verification and what was perceived as adequate varied in degree as a result of pupils' interaction with the learning situation, i.e. experience of the programming medium, of the mathematical content of the tasks and of the process of verification itself. Interesting comparisons may be made between pupils' perceptions of adequacy in the pre-task interview and the actual verification carried out in tasks similar to the interview questions. In general the degree of empirical verification carried out in the actual tasks was less than that advocated in the pre-task interview.

[54] See Chapter 5 § 5.2.4

Pupils displayed consciousness of wanting to make their choice of values in empirical verification representative in some way. This could take the form of a spread of values within the domain defined by the question. In the fourth set of tasks (construction of procedures which output odd and even numbers), some pupils displayed an appreciation of the need to try a representative sample of both odd and even numbers, whilst other pupils failed to appreciate the need even to distinguish between the two types of input. A common belief held by pupils was that the use of large numbers in the verification process was somehow more demanding than the use of small numbers.

Some pupils began with exhaustive verification (sometimes in line with thoughts expressed in the course of their pre-task interview), but as they became familiar with the learning situation, their verification became partial. Some pupils were directly prompted by the learning situation, particularly by the presentation of the data in tabular form and by the instruction to pick a number between one and ten (in the second set of tasks) into attempting exhaustive verification of the domain defined by the question. It should, however, be noted that since most of the functions used in the four sets of tasks were linear, there was generally, of course, no need to test more than two values (as has already been noted). Some pupils developed strategies involving the use of a specific number of input values but it seemed unlikely that these were based on anything more than guesswork and experience.

Occasionally pupils strayed outside the domain defined in the question: some did this without any apparent awareness that they were going beyond the requirements of the question; some treated such departures as mistakes (which they frequently were in typing a value for the input) and ignored them. Only in some of the second set of tasks did testing values outside the domain cause a problem since here the possibility of division by zero arose.

7.5.2 Verification and Expectation

In examples where the result was familiar or clearly stated (e.g. Task 2.1 where the output was always one) or where the output could be checked off against a printed table of values (e.g. Tasks 1.4 - 1.6), verification was a simple process of checking the computer output against the expected output. In examples where the result was unfamiliar (e.g. in addition of consecutive number examples - Task 3.1 - 3.3), verification became a duality involving both the ascertaining of the value that was to become the expectation and its testing. This appeared to take one of two forms: either the computer verification would precede the pupil's mental check, so the computer output became the expected outcome; or the expectation was derived from calculations performed mentally (or otherwise) by the pupil, and then verified on the computer.

Example: Tom's use of computer verification as a creation of expectation and as a check of mental calculation - 85: H 3.2

Tom had constructed a procedure for Task 3.2 (the sum of three consecutive numbers):

```
to odo :k
op :k + :k + 1 + :k + 2
end
```

He described his thoughts audibly as he worked, beginning by making a prediction through mental calculation:

Tom: *So if it was three you put in: seven. Twelve. It ought to be twelve.*

He then checked this prediction, now the expected outcome, by verification on the computer:

Input	*Output*
3	12

For the remainder of the verification of his procedure, Tom chose to precede his mental calculations by verification on the computer.

Input	*Output*
7	24

Tom: *Seven, add eight is 15, add nine is 24. Yes.*

Input	*Output*
1	6

Tom: *The Calculator says six: one add two is three, add four is six. Yes.*

Input	*Output*
8	27

Tom: *Eight, add nine is 17, add ten - I'm happy.*

7.5.3 Verification and Justification: a Shifting Balance

Changing expectations of the outcome of empirical verification led to a shift in the balance between justification and empirical verification.

Example: Rachael's shift from empirical verification to justification effected by the difficulty of mental calculation - 6 (B 4.4)

When Rachael tried values in Hannah's procedure for Task 4.4, she had to go through an elaborate feat of mental calculation to interpret what was happening. Here she had only the following set of values (which she had chosen herself), resultant from her attempt to verify Hannah's procedure using the Calculator software, and the question statement on which to base her analysis:

Input		*Output*
10	11	43

Researcher: *What do you think?*

Rachael: *I'll try it again. Two times ten is 20 add one is 21. Two elevens are 22, and 21 add 22 is 43. I thought it came out as an even number.*

Researcher: *An odd number plus an odd number.*

Rachael: *One add one is two.*

Researcher: *Yes.*

Rachael: *And three add three is six, and five add five is ten, and seven and seven is 14. Oh, I know what I've done. Oh, no. (Pause) Oh no, she's done it wrong..., hasn't she? I forgot to add the one, because two elevens are 22 add one is 23, and 23 add 21 is 44. So that one's wrong.*

Researcher: *The thing is, is it producing an even number?*

Rachael: *No.*

For the computer verification to be effective, the pupil would have to go through mental calculations of a complexity which would admit errors. Rachael actually found it much easier to explain, with little reference to specific examples, where Hannah had gone wrong when she saw the procedure on the computer screen:

```
to wwww :x :y
op (:x + :y) * 2 + 1
end
```

Rachael: *Oh, I know what you've done. She's thought that you were getting two odd numbers and then it comes out as an odd*

> *number, because you did roughly the same as last time, but you've added one. Like, you take two numbers like three add four is seven, times by two is fourteen, and you add one, fifteen. 'Cos it could work on that one - just on the first one, the even one, without the adding one, but it wouldn't work with the one because you've got to get out an even number if you add two odd numbers together.*

Now this was hardly a perfect explanation of why Hannah's procedure was wrong, although it was an adequate explanation of why the output was an odd number rather than an even number. But it was important in that Rachael was *explaining* how the procedure worked rather than relying on empirical testing. Resort to explanation rather than testing was typical of a number of examples from the data where verification had become largely meaningless due to the difficulty of mental calculation or to unfamiliarity with expected outcome. In this example, the difficulty level would appear to have pushed a pupil towards justifying her procedure in terms of natural (or other) language rather than attempting to verify it with input-output pairs which carried little meaning.

7.6 CONCLUSION

Consideration of the data from Phase III of the study so far points towards three expressions of meaning through which pupils articulated their mathematical ideas. The first is a natural language statement of a mathematical relationship, by which I mean some mathematical idea which has been expressed in everyday language, but which nevertheless conveys the import and integrity of an equivalent statement in algebraic language (although inevitably in less concise a form). The initial question statement of each task in the study was expressed in such language, but here, of course the expression of meaning was the researcher's not the pupil's. However it was the starting point for all tasks and an expression which pupils had to *use*, albeit not an expression of their own *creation*. Natural language statements featured highly throughout the study, either as starting points for tasks or as expressions of meaning made within a task by pupils themselves.

The Logo procedure which lay at the centre of each task was another expression of meaning. A Logo procedure is a formalised statement of a mathematical relationship expressed in algebraic language. Re-expression of natural language statements in terms of a formalised Logo procedure was the central action of each task in the study. The connection between these two expressions of meaning appeared to be the act of symbolisation which constituted each re-expression. Closely related to the act of symbolisation was the analysis of the structure of a procedure. This exploration of structure was itself a connection, this time between a formal Logo procedure - a formalised statement of a mathematical relationship expressed in algebraic language - and the expression of *generality* about that relationship. Thus pupils appeared to express meaning in at least three ways: through natural language statements, through Logo procedures and through generalisations about those Logo procedures (or even about the natural language statements). Acting as connections between these three expressions of meaning were acts of symbolisation and the analysis of structure.

How did the computer mediate the creation of these connections? Central to both acts of symbolisation and the exploration of structure was the construction process of procedures on the computer. The construction process formalised mathematical relationships and structured them. This structuring process provided the starting point and focus for further analysis of the procedure. Construction of a procedure was also important in that through it pupils gained ownership of the mathematical problem. The process of writing a procedure, even of making a simple translation from natural language to symbolic language, appeared to give pupils an involvement with the task which was not achieved when the task began with a symbolisation which had been carried out by someone else. The act of symbolising seemed to be of central importance. It is through this act that symbols became concrete, and this concretisation effected subsequent manipulation of the symbols.

However, it was not through construction alone that symbolism acquired meaning and structure was explored. Where the construction process used was one where verification informed construction, it was the interplay between construction and verification which revealed and highlighted the structure of the procedure. Term-by-term analysis and systematic restructuring supported by verification appeared to promote pupils' ability to generalise, to analyse and ultimately to prove. Systematic verification highlighted mathematical relationships and how those relationships behaved with varying values of the input. Importantly, verification endowed symbols with a history of specific examples and values. Furthermore, pupils manipulated symbols in an empirical context with the constant possibility of validating the result of that manipulation. This experience appeared to equip them with an ability to talk about these symbols and to manipulate them in contexts removed from the immediacy of the construction/verification environment, namely in discussion about and analysis of the structure and generality of the procedure in the justification stage of the task.

In Chapter Eight I look at the relationship between the particular and the general, a relationship which has already emerged through this chapter's discussion of generalised Logo procedures and specific numerical verifications of those procedures. In particular I focus on how pupils used particularity in order to express generality.

CHAPTER EIGHT: PARTICULAR AND GENERAL

8.1 INTRODUCTION

Having looked at the relationship between construction and verification in Chapter Seven, I move on in this chapter to a second significant relationship which emerged from analysis of the study data. An important component of many pupils' work during the study was the relationship between the particular and the general. Frequently this relationship was characterised by ambiguity: at one moment pupils might be thinking and speaking in particular terms about a procedure and the way it behaved in a *particular* circumstance; the next they might be making a general observation about the procedure and how it behaved in a *set* of circumstances. This chapter concentrates on how pupils used ambiguity between the particular and the general to express generality, and how this relationship was mediated by the computer.

From analysis of children's work I identify two forms of webbing, which I describe as *semi-generalisation* and *generic structuring*. Semi-generalisation was a means of using a procedure, general only in terms of a strictly limited domain, as a framework within which to express complete generality. Generic structuring was a means of structuring and exploring a procedure through the consideration of generic examples.

8.2 SEMI-GENERALISATION

To explore semi-generalisation I want to focus on an example taken from Phase III of the Study. It features the members of Group G (Nicholas and Edward) working at Task 2.1 in which they were required to write a procedure to model the puzzle

> *Amaze Your Friends ....*
>
> *Pick a number between 1 and 10. Multiply it by 2. Add on the number you first thought of. Divide by 3. Divide by the number you first thought of.*
>
> *The Answer is .... ONE !!!*

The approach to the question adopted by both Edward and Nicholas demonstrated how the computer might provide support to enable pupils to work at a level at which their ideas were only partially thought through and later to "tighten up" these emergent concepts. It demonstrated what Wilensky describes as *concretion*, i.e. "the process of the new knowledge coming into relationship with itself and with prior knowledge, and thus becoming concrete" (Wilensky 1991 p. 201) The computer allowed them to work simultaneously at a specific and a general level; it enabled them to work in such a way that they could see the general in the particular.

<u>Example: Edward using semi-generalisation to web his construction of generality - 54 (G 2.1)</u>

Edward read the question carefully and immediately typed the following procedure:

```
to jem :d
op :d * 2 + 5 / 3 / 5
end
```

He then tested this with an input of 5 using the calculator. It gave an output of 10.33334:

Input	*Output*
5	10.33334

This was not a fully generalised procedure because particular values had been used within it for what should be general terms. He could not believe that this was wrong (and of course within a domain where the input - :d - has the single value 5 it is correct, brackets apart) so checked his workings, comparing the printed words with his procedure on the screen:

Edward: *What did I do wrong? Think of a number: yeah. Put it in, times it by two: yeah. Add on the first number you thought of: yeah. Divide by three: yeah. Divide by number you first thought of: yeah. I'll try again: put in five, press Calculate.*

He tested this again with an input of 5 using the calculator. Again it gave an output of 10.33334:

Input	*Output*
5	10.33334

Edward: *No! I don't want that answer, I want one.*

At this point there was a choice of routes open to Edward: making the procedure fully generalised and then sorting out the brackets; or making this partially generalised procedure work through the use of brackets and then making it fully generalised. Either approach was possible because for the computer to function (and, by extension, to inform Edward's re-constructions), it was not necessary that everything should function perfectly at one and the same time. The ability to preserve and use a partly completed structure in its half unfinished state meant that attention might be focused on one aspect or another as appeared more appropriate. Here, provided the domain remained as one where the input (:d) had the single value 5, Edward could begin by bracketing his partially generalised procedure, and later on shift his attention to its generality. As this was the approach Edward adopted, the researcher gave him the first input on brackets[55]. Edward experimented with the positioning of brackets, verifying his procedure each time with an input of five. The strategy he adopted was trial and error, the criterion for success the proximity of the output to one. The procedure, however, remained partially generalised. The iterations in Edward's design are shown here with his accompanying attempts at verification (the value he used as an input paired with the corresponding output) and any comments he made:

op :d * (2 + 5 / 3) / 5

Input	*Output*
5	3.666666

Edward: *I'm getting closer.*

op :d (* 2 + 5 / 3) / 5

Input	*Output*
5	5

Edward: *I'm so close.*

op (:d * 2 + 5 / 3) / 5

Input	*Output*
5	2.333334

(op :d * 2 + 5 / 3) / 5

Input	*Output*
5	11.66666

op (:d * 2 + 5 / 3 /) 5

---

[55] See Chapter 5, § 5.2.4

Input	*Output*
5	) in wrong place in jem

op (:d * 2 + 5 / 3) / 5

Input	*Output*
5	2.333334

Edward: *I'm getting even closer.*

The researcher gave the second input on brackets[56] and Edward straight away wrote

op (:d * 2 + 5) / 3 / 5

He verified this empirically and expressed his satisfaction:

Input	*Output*
5	1

Edward: *Yes.*

The first stage was complete. The researcher now shifted the focus of attention to the lack of generality by adopting a form of exploratory intervention[57] :

Researcher: *What about other numbers?*
Edward: *[Pause] I know what I've done wrong. Get rid of that 5 .... [unintelligible muttering]*

Edward immediately replaced the two fives with variables (:d)

op (:d * 2 + :d) / 3 / :d

[56] See Chapter 5, § 5.2.4

[57] See Chapter 4 §, 4.4.2

and verified that his procedure worked for a variety of input values:

Input	*Output*
8	1
10	1
56	1
10000000	1
2342	1

The apparent ease with which Edward completed the task suggests that the shift of attention had already begun: perhaps as he worked on the brackets Edward was beginning to question what would happen with other inputs.

In this example, Edward worked through three stages: the focus for the first was the structure of the procedure; in the second attention shifted to questions of syntax; in the third the question of generality became the focus[58] .

In this example Edward constructed a procedure which was only partially correct. What singled it our from other partially correct procedures, however, was that although it functioned correctly for only one input value, its general structure was nevertheless largely correct. The procedure was in fact *semi-generalised.*

Semi-generalisation was characterised by an approach to construction whereby a partially correct procedure was created and used as a basic framework - a partly-functioning prototype. This model was then improved bit by bit: pupils' attention might shift from one aspect to another and they did not need to tackle all aspects of the procedure simultaneously in order to correct the procedure. They could test the validity of a partially correct procedure within any parameters they defined as

[58] This was not a unique route and one of the strengths of working within this learning situation was the variety of solutions and paths to those solutions available to the pupil.

necessary for the procedure to be semi-generalised, for example a severely limited domain of input (commonly confined to one value). The importance of the computer was its role in capturing the working model in a kind of limbo which nevertheless made available the possibility of empirical validation. Completion of the procedure then became a process of homing-in, again supported by easily achieved validation. The contrast with paper and pencil methods is striking.

By defining the domain in this way, a procedure was as generalised as necessary for further work on that procedure, be that work further efforts towards complete generalisation, or further work on the structure of the procedure. The computer preserved this "semi-generalised" procedure and as long as the input remained constant, it could be seen as an object which could itself be manipulated. The advantage is that of being able to adjust an already functioning machine in order to get it to work properly, over that of trying to get the machine to function in the first place.

The generalisation of the procedure might not proceed beyond this level, or it might happen that when the bugs in the procedure had been removed, the pupil would in fact make the procedure fully generalised. This might also occur as part of the justification process, so a semi-generalised procedure provided a good focus for discussion.

Semi-generalisation provided a connection between two expressions of meaning, namely between the statement of a mathematical relationship in natural language (usually the question statement) and the expression of that mathematical relationship in formalised algebraic language (i.e. the Logo procedure).

8.3 GENERIC STRUCTURING

Continuing with Nicholas's and Edward's work in Task 2.1, I want now to focus on ways in which pupils bridged the gap between specific evaluations of their procedure and the articulation of general statements about that procedure.

Example: Edward and Nicholas using specific examples to explore the general nature of a procedure - 55 (G 2.1)

In Task 2.1 Edward and Nicholas used their formalised procedure as a resource to express their proof of why this puzzle always comes to one. They worked through several examples using specific values, but here the pupils were using them as a means of describing general patterns of behaviour through consideration of the particular. The degree of generality changed as the work proceeded, sometimes with a shift of attention focused by the researcher. The example began with a consideration of Edward's procedure described above:

```
to jem :d
op (:d * 2 + :d) / 3 / :d
end
```

Researcher: *Can you work out from that why it always comes to one?*

Nicholas made an initial attempt at an explanation which set the discussion in motion and acted as a base line from which the pupils could only move up:

Nicholas: *Whenever you put three or whatever the number is, it just cancels out.*

Edward began to generalise but instead referred to an example:

Edward: *Because whenever you put a number in, if you times it by two, it'll equal - say I put in seven, it would equal fourteen and divide by three, it would equal [laughs and stops].*

Researcher: *Try again, because I think you were on the right lines.*

He used another example, this, like the last, chosen for its generality rather than its specificity:

Edward: *Just say that the number was five, times by two equals ten, then add five equals fifteen and then you divide it by three so that equals the number you started off with. Then if you divide by the number you started off with, it will always equal one.*

The researcher then attempted to prompt Edward to generalise his explanation from the specific case of five: he shifted attention away from the particular towards the general.

Researcher: *Can you say why it always equals one? You say it works with five.*

Edward went on to prove the result, expressing his ideas by means of naturalised formalism[59]:

Edward: *It works with any number ... [looking at the procedure on the screen] because whenever you times a number which you started off with by two then add the number you started off with, it's more or less like timesing by three.*
Researcher: *Right. Carry on.*
Edward: *Because then you divide it by three, equals the number you started off with - it's just like doing the three times table. Then if you divide by the number you started off with it will always equal one.*
Researcher: *Great.*

In this example Edward used specific examples (i.e. evaluations of procedures using particular values) in order to understand the general nature of that same procedure. I want to refer to this phenomenon as generic structuring: *generic* because it made use

[59] An idea I develop in Chapter 10, § 10.3.

of examples which were chosen not for their specific values, but as representatives of their class; and *structuring* because it was used as a means of shaping experience of specific empirical verification into the statement of some general statement, or for the examination and clarification of a partially-emergent generalisation. Generic structuring was a process: it was the connection between two expressions of meaning, namely the verification of a procedure with a specific value and a generalised statement about that procedure, expressed in natural language, formal algebraic language, or indeed in naturalised formalism.

It is not necessarily straightforward to define what did and what did not constitute generic structuring. Trying out a numerical example in a procedure could work at two levels: it could be merely the verification of that one particular example; or it could be the exploration or explanation of the structure of the procedure through the testing of a value. When the latter was the case, use of this example became generic structuring for the pupil because through the example's particularity he or she was able to see something of the procedure's generality.

<u>Example: Luke using generic structuring to explore how his procedure would work for the class of cases of which his value was representative - 47 (E 2.2)</u>

In an attempt to explain why the input and output in Task 2.2 were the same number, Luke, who had written the following procedure:

```
to plom :p
op (:p * 2 - :p) - 4 / 4 + 1
end
```

questioned how his procedure functioned:

Luke:	*Times by two and times by two again... What happens if it won't divide by four? What happens if you get a number like nine?*
Researcher:	*What does happen if you get a number like nine?*
Luke:	*So if you put in a number like one: times two is two, times two is four. It always comes to even, won't it. If you put in an odd*

> *number it always becomes even. So it won't - it'll always divide by four...*

Use of generic structuring - and it clearly *was* generic since Luke had chosen the number one to represent odd numbers (presumably for ease of calculation) in preference to the nine which he first mentioned - had made it clear to Luke that whether the input was odd or even, it would be made even by the first operations in the procedure - a fact he equated here with divisibility by four. He had not made the explicit statement that the input had become a multiple of four, but it would seem from his consideration of this generic example that this is precisely what he had discovered.

It is not always easy to know whether use of an example cited by pupils was for them a case of generic structuring (i.e. whether the example was generic), or whether they were merely rehearsing how a function operated with a particular value for the input. (This rehearsal of a function's operation, of course, was an important point along the route to understanding why the function operated as it did.)

<u>Example: Establishing what constituted genericity in Rachael's discussion of her procedures - 44 (B 4.4)</u>

In Task 4.4 (the summation of two odd numbers), Rachael was able to explain her own procedure

```
to qwer :k :l
op (:k * 2 + 1) + (:l * 2 + 1)
end
```

in terms of a numerical example:

Rachael: *Well, you have your number and you times it by two, and if you have the number three, two times three is six and add one is seven, and that comes out as an odd number always. And then you have your next number, eight: two eights are sixteen, add one is seventeen. And then you just add those together. And they will come up as an odd number.*

Is this example generic? Were the particular values of the inputs used in such a way as to describe the generality of the procedure (ignoring the fact that the final statement was incorrect)? Insofar as Rachael used the input three to make the point that "two times three is six and add one is seven, and that comes out as an odd number always," this might be counted as an example of generic structuring: she was establishing the oddness of :k * 2 + 1. However the remainder of her explanation said little about the structure of the procedure or the general nature of its output: "And then you just add those together. And they will come up as an odd number." Although this example has generic elements, it is largely a description of what happens when two particular values are used as inputs.

Later, with support from the researcher, Rachael produced another description:

Researcher: *:k * 2. Is that even or odd?*
Hannah: *Even.*
Rachael: *Even.*
Researcher: *What about :l * 2?*
Rachael: *That would be even as well. Four times two is eight.*
Researcher: *What have you got left?*
Hannah: *Adding one.*
Rachael: *Adding two ones.*
Researcher: *Which makes?*
Hannah: *Two.*
Rachael: *Two. So you add two onto it and it comes out as an even number.*
Researcher: *Can you explain that again.*
Rachael: *Well, you have your number and you do like :k times two - could be five: five times two is ten. And then on the :l times two - six - six times two is 12. You've got two ones left over so one add one is two and you add two to that so it comes out as an even number.*

Is *this* example generic? Again her explanation made use of particular values: nowhere did she state explicitly that :k * 2 and :l * 2 produced even numbers, but her description of the behaviour of the function with using two particular values demonstrated this (and whether consciously picked, her choice of one even and one odd input added weight to the generic nature of the example). The example continued with an element of analysis, i.e. the separation of the two ones, their summation and the conclusion that their sum would add a further even number to the existing even numbers (these last five words not stated explicitly). The use of particular values in this example demonstrated both how the function behaved as it did and why: this, then, is generic structuring.

An essential element of generic structuring was the rehearsal of procedure behaviour under specific values of input: it was through this process that examples changed from specific to generic. How was this rehearsal facilitated? Central to rehearsal of procedure behaviour was the procedure itself. Now, whilst an algebraic function in a paper and pencil context could serve the same purpose, the difference between that context and the learning situation developed in this study is the fact that pupils had ownership of and familiarity with the procedure they were using because they themselves had constructed it: the act of symbolisation was their own.

Pupils appeared to use generic structuring in a variety of ways. In many examples pupils attempted to express their ideas in entirely general terms. If this proved too difficult, they used generic structuring as a kind of reference point to which they could fall back, and from which they could embark on their generalised description once again: rather like attempting to recite a poem from memory, a quick glance at the overall structure of the poem when one's memory fails, restores enough to complete the recitation.

Example: Ben dipping into the particular to structure his generalised explanation - 11 (D 2.2)

In Task 2.2, Ben had constructed the following procedure

```
to astound :a
op ( ( ( ( :a * 2 ) * 2 ) - 4 ) / 4 ) + 1 )
end
```

During subsequent discussion, Ben described why the output always equalled the input. He began by discussing it in a generalised form in natural language, but when he got confused he resorted to a generic example:

Ben:	*Double it, double it again ...*
Matthew:	*Whatever you put in ...*
Ben:	*Oh yeah. If you double it and double it again, it should come out four times. And then take away four. You take away four of the, ah ... [pause] Say that the number you put in was one, it would come out as four; take away four. It would take away one from each of the numbers.*

In the last sentence Ben was attempting to express his description more generally again, and in fact from here, the researcher tried to get Ben to re-express his ideas in the language of the procedure:

Researcher:	*Try using :a as the number you are multiplying by. Would that help?*
Ben:	*If you had four times :a and take away four... If you've got four :a's, then it would be a bit like taking away one from each :a. And then if you divide it, you would have :a without the one on the end. And if you add on one you get back to the number you started with.*
Researcher:	*Difficult to explain, isn't it.*
Ben:	*Yes!*

The use of a generic example had enabled Ben to restructure his explanation in more general terms: the process of generic structuring.

Clearly the researcher's role was important here. In the same way that here use of the language of the procedure was suggested, so the use of generic restructuring could be prompted by the researcher (although it had to be borne in mind that what was perceived as generic by the researcher might not be perceived as such by the pupil).

In this next example, generic structuring provided a means of exploring the structure of Rachael's procedure (which summed an odd and an even number), and the exploration of structure led to a conceptual re-structuring of the procedure, necessary for an appreciation of why the procedure behaved as it did.

Example: Rachael using generic structuring to restructure her procedure conceptually - 109 (B 4.5)

In Task 4.5 Rachael had constructed the following procedure:

```
to uszssss :x :y
op (:x * 2) + (:y * 2 + 1)
end
```

Researcher:	*Why does this always come out as an odd number?*
Rachael:	*Oh.*
Hannah:	*Because...*
Rachael:	*Well, you have one number, five, and you times it by two which gives you ten and then you have another number, six. Times it by two - twelve - and add it on. So one's an odd number, one's even. You add them together and you get nine.*

On being asked again, Rachael made the following statement:

Rachael:	*Because you times it by two first and then you times it by two. Then you've got one left over and you add the one.*

What is interesting here is that the first statement was not in fact an exact model for the second. It was not a case of the formulation of a generic example forming the model for a situated abstraction. The statement "one's an odd number and one's even" came directly from trying values in the

*(:x * 2) + (:y * 2 + 1)*

structure; the second statement restructured the procedure as

$$(:x * 2) + (:y * 2) + 1,$$

the all-important re-grouping necessary to appreciate the oddness of the expression. But how did Rachael get from one to the other?

She appeared to have taken out the "times it by two" statements from the first statement and isolated them. Perhaps it was the rehearsal, the verbalisation, the repetition which highlighted a pattern (or in this case a common property) which might be isolated and detached from its surroundings. When this was seen as an object - a chunk of evenness - Rachael could generalise the effect of the addition of the extra one. Rachael had in fact concretised the chunk and was able to manipulate it as an object in its own right.

8.4 SUMMARY

This chapter highlights two important connections between expressions of meaning, two forms of webbing which pupils used to construct mathematical meaning. Both were concerned with the fluid, ambiguous relationship between particular and general. The first - semi-generalisation - made use of the ability to construct a procedure which would work only for one value of input, but, under certain circumstances, would behave as if it functioned for all values. Thus semi-generalisation provided a connection between the statement of a mathematical relationship in natural language (usually the question statement) and the expression of that mathematical relationship in formalised algebraic language (i.e. the Logo procedure) through the breaking down of the construction task into manageable chunks, the partially generalised procedure behaving in a manner which precisely and usefully modelled the final generalised construction.

The second connection - generic structuring - made use of the relationship between particular examples, generic examples and generalisations about formalised procedures. Through the restructuring of a procedure, the structure of which use of generic examples illuminated, generic structuring provided a connection between the formalised mathematical relationship defined in a Logo procedure (backed up by empirical testing) and generalised observations on that mathematical statement (albeit generalisation which existed within a situated context).

Having in this chapter considered the relationship between particularity and generality, in Chapter Nine I move on to look at the role which exploration of equivalent procedures played in creating meanings for generality and in attempting to justify that generality.

CHAPTER NINE: EXPLORATIONS OF EQUIVALENCE

9.1 INTRODUCTION

Chapter Eight focused on the important relationship between general and particular. Another interesting and prominent feature of pupils' work throughout the study was the construction and analysis of equivalent procedures. Chapter Nine looks at pupils' construction of equivalent procedures and charts how discussion of equivalence appeared to lead to a deepening awareness of mathematical concepts. I describe how pupils were able to make situated abstractions of a number of mathematical relationships through discussion of equivalence, and suggest that these examples further support a view of abstraction as the creation of connections between expressions of meaning as opposed to a view which sees abstraction as the moving away from the "concrete" through an externally-imposed hierarchy of abstraction.

9.2 ROUTES TO EQUIVALENCE

Although each task may have had an obvious solution in the eyes of someone with more extensive mathematical experience than the pupils concerned, the construction of the learning situation facilitated alternative solutions and alternative routes to those solutions. Also, given the "competitive" element of the learning situation, where pairs of pupils independently constructed procedures, there was likely to be a degree of divergence in the procedures they constructed. This divergence provided an interesting source of discussion for researcher and pupils. Why did these two procedures have the same output? What was the relationship between two superficially different algebraic expressions of the same mathematical relationship? From the researcher's point of view, the interesting question concerned what pupils would discover in their attempts at answering these questions, since the comparison and analysis of equivalent procedures appeared to provide fertile ground for the development of mathematical concepts, and how the computer mediated these discoveries.

Equivalence was not always the result of independent working, however - what one might call chance equivalence.

<u>Example: Ben making a prediction about the nature of his partner's procedure and analysing why this prediction and his partner's actual procedure were in fact equivalent - 18 (D 3E)</u>

Matthew made the following attempt at the extra task in the third set of tasks:

```
to aboo :a
op ((:a - 1) * :a) + 2 * :a
end
```

Ben tested his own set of values in Matthew's procedure as follows using the Calculator:

Input	*Output*
0	0
1	2
2	6
3	12
4	20
5	30

From scrutiny of these values, Ben predicted that Matthew's procedure had the function

op :a * (:a + 1)

He explained this as follows:

Ben:	*What this is doing is it's timesing :a, the input, by the number above it, :a + 1.*
Researcher:	*You're saying it's :a times (:a + 1). O.K., have a look at Matthew's procedure.*

Although this was not Matthew's function, Ben was able to explain, partly through generic structuring, and partly through analysis of the formal procedures using their language of construction as the means of expression and communication, that what Matthew had written was equivalent to what Ben had imagined:

Ben:	*[looking at the procedure] Yes.*
Researcher:	*Can you explain it.*
Ben:	*:a - 1 times :a. Say the input was three, :a - 1 is two, times :a is six. Um, the times :a. That would make six.*
Researcher:	*[clarifying what Ben is pointing at] You mean (:a - 1) times :a.*
Ben:	*Yes, that's six. I think what would happen ... Yes it would add on two times :a, which would make it twelve. [Conclusively] And that is three times four.*
Researcher:	*Which is the same as saying what you said - :a times (:a + 1)*[60].

Ben was articulating, with no knowledge of formal algebraic manipulation at his disposal, the equivalence of

$$(a - 1) * a + 2 * a \text{ and}$$

$$a * (a + 1)$$

In fact he was making a situated abstraction. Ben's analysis was dependent on the context: his prediction was made on the basis of scrutiny of Matthew's tabular

[60] The Researcher finished off Ben's unstated conclusion.

verification (the tabular display acting, it seems, as a powerful means of visual organisation); his exploration relied heavily on the use of generic examples whilst being expressed partly in the formal language of construction.

Example: Ben expressing the spontaneous desire to construct and test a procedure which he already had in mind and analysing the reasons for its equivalence with a previously constructed procedure, leading to a situated abstraction of a mathematical relationship - 17 (D 3.3)

In Task 3.3 Ben constructed his first procedure spontaneously:

```
to con3 :a
op (:a - 1) * :a
end
```

He verified this using the Calculator with the following pairs of values:

Input	*Output*
5	20
1	0
3	6
10	90

Ben described his procedure as follows:

Ben: *Well, it just does :a minus one first and times it by :a, really.*

and

Ben: *So it would be the input - I mean one number less than the input, and then it timeses it by :a.*

Matthew: *It timeses it by the input.*

Ben himself asked to construct a second procedure:

Ben:	*Sir, could I do another procedure which is different but does the same thing?*
Researcher:	*Mm, you can. Have you thought of one?*
Ben:	*Um, yes.*

He wrote the following, a procedure he had clearly already worked out in his head:

```
to con3x :a
op :a * :a - :a
end
```

What had prompted this procedure? Although it was equivalent to con3, it no longer modelled the question statement (i.e. the product of an integer and the preceding integer). This implied that Ben had constructed this new procedure based on the number relationships he had identified in the four pairs of values resultant from his empirical verification of his procedure. If this was the case, then it highlights the importance of the visual nature of verification: Ben was very capable of verifying procedures such as con3 and con3x in his head; it is, however, less certain that he would have spotted the relationship necessary for construction of the latter had he not verified the former on the computer using the tabular nature of presentation which the software made possible. The nature of the empirical verification appeared to shape Ben's awareness of the existence of an equivalent procedure. He gave some indication of this consideration of numerical relationships using the tabular presentation of his own verification in his next statement:

Ben:	*Um, I was thinking of ... I thought if it was :a times :a, what would you have to take away: just sort of experimenting. And, um ...*

Matthew clarified what Ben had said in his own mind:

Matthew:	*So it squares and takes whatever you've thought of.*
Ben:	*Takes away, so ... [long pause] it's hard to explain. :a times one less than :a. If you times :a by itself and then ... You times :a by itself and you take away - so you'll have lots of :a's, and*

> *you take away one of the :a's so it'll be um ... So the output will be the same as :a times (:a - 1). It's hard to explain.*

Ben had explained the equivalence between his two procedures by analysis and description because he did not possess the formal manipulative algebraic skills to show by the "usual" means that

$$a^2 - a = a\,(a - 1).$$

Again, he was making a situated abstraction. His analysis began from enquiry into the relationship between a^2 and $a(a - 1)$. a^2 became his starting point and thence he adjusted his procedure. This was done in his head and the procedure which he eventually typed on the computer was the final correct version. He then tried four values in order to test the procedure. The learning situation had enabled him to write what for him was an obvious first procedure, which acted as a bench-mark against which he could investigate another procedure which he had at the back of his mind, derived from the verification data. Experience of constructing, verifying and explaining the procedures gave him the means to express why they were equivalent.

The exploration of equivalence was a connection, but not in exactly the same sense that this term has been used previously where it has denoted the linking of distinct expressions of meaning. Here the link was between a mathematical concept and a deeper, richer understanding of that concept, plus, in some cases, a link to other related mathematical concepts. Now in fact this use of the term connection is not dissimilar to the previous use: here the edges between expressions of meaning became a little blurred; the mental image of a connection acting as a neat bridge between two expressions of meaning was subsumed into one consisting of many links within and between mathematical concepts.

Thus in this previous example, through creation and analysis of a procedure equivalent to his first, Ben had moved from modelling a function which multiplied an integer by the previous integer (as stated in the initial question), to a function which was equivalent but based on the relationship between ordered pairs of the first function, to an explanation of the equivalence of the two functions. In fact the equivalency was a statement of the Distributive Law of Multiplication. What the exploration of equivalence in this situated context appeared to do was to develop for an individual pupil a set of deep structural relationships for that initial function, and, by the same token, make the newly discovered mathematical relationships concrete, in the sense that these relationships were richly and meaningfully connected to other relationships with which an individual pupil was familiar and had had experience of symbolisation and empirical verification.

To illustrate further the power of explorations of equivalent procedures to build connections between expressions of mathematical relationships, I want to look at two groups' work in Task 3.2, which asked pupils to find a function which sums three consecutive numbers.

<u>Example: Tom's exploration of equivalence leading to a situated abstraction of an additive relationship - 64 a) (H 3.2)</u>

Tom's first procedure

```
to odo :k
op :k + :k + 1 + :k + 2
end
```

was one which suggested that on reading the question Tom immediately perceived the structure of how to generate successive consecutive numbers and was able to express this structure in the organisation of the terms in his procedure. He explained the procedure to David:

Tom:	*O.K. Five, plus five plus one, which is six, so it will be five plus six. Plus any old number plus two which will be five add two is seven, so it will be five plus six plus seven. See?*

David: *No.*

This presentation of the procedure was explicitly structured in that terms were grouped and ordered. One might expect a pupil with knowledge of traditional paper and pencil algebra to use brackets to help define this structure

$$x + (x + 1) + (x + 2)$$

Tom did not use brackets but it emerged from later discussion that he had considered their use and that these brackets would have been used in a way similar to that of the traditional paper and pencil pupil. In the following exchange he described how consecutive numbers were generated and began (although unfortunately interrupted) to describe how his brackets would have been used:

Researcher:	*David, if you put five in there [pointing to odo], which part would be the five?*
David:	*That one [pointing at the first :k].*
Researcher:	*So where's six, then?*
David:	*The second :k?*
Researcher:	*No, :k is going to be five if you put five in.*
Tom:	*You see you're adding one onto :k, making six. I thought you might need brackets because it's :k + :k + 1 is eleven...*

At this point he was interrupted by David. Where might Tom's train of thought have led from here had he not been interrupted? It seems reasonable to conjecture that Tom had intended bracketing k + k + 1 as the first part of his procedure to be worked out, and perhaps k + 2 as the next, giving the following structuring:

$$(x + x + 1) + (x + 2)$$

or even

$$x + (x + 1) + (x + 2)$$

It certainly appeared to be this structuring that he had in mind when he began to explain the procedure to David (in the extract above). David, however, saw the first part of the structure differently:

David:	*If that's five, though...*
Tom:	*It makes the same thing anyway!*
David:	*If that's five, it's add five again and that makes ten, then one, that makes eleven...*

Here he was thinking of a structure which might be represented as follows:

$$(x + x) + 1 + \ldots$$

Tom took up his train of thought:

Tom:	*Plus :k is 15, plus two is ... No, plus five is sixteen, plus two is eighteen.*
David:	*Yes.*
Tom:	*So whichever way you want to put it, it makes the right answer.*

This structuring makes a good starting point for the consideration of equivalence, since

$$x + x + 1$$

is seen originally by Tom as

$$x + (x + 1)$$

and by David as

$$(x + x) + 1.$$

By the end of the discussion both have taken part in expressing the procedural equivalent of the following equality:

$$x + (x + 1) = (x + x) + 1,$$

i.e. the Associative Law of addition.

Example: Tom and David articulating a further additive relationship through consideration of equivalence - 64 b) (H 3.2)

After his first procedure Tom was challenged to write a second. This time, rather than structuring it as the sum of discrete expressions, each describing one in a series of three consecutive numbers, he chose to group the variables together and the constants together:

```
to qwark :h
op :h + :h + :h + 1 + 2
end
```

Tom continued the discussion on equivalence by describing qwark:

Tom:	*Because even if you add all those [pointing at variables :h in qwark] up together rather than saying five plus five plus one is six, five plus six plus five plus two is seven - five plus six plus seven - just add all of them together.*
Researcher:	*So, what's qwark, then?*
Tom:	*Because it works both ways with just adding things... Because whichever way you put it, it's adding all of them together. That one works just the same, so five plus five plus five is fifteen. So it could have been plus two plus one.*
David:	*Exactly the same thing written in a different way.*

Here he was attempting to articulate the following relationship:

$$x + x + 1 + x + 2 = x + x + x + 1 + 2$$

Tom was expressing the Commutative Law of Addition when he said, "Because it works both ways with just adding things," and he used it to explain why his procedures were equivalent.

David was brought back into the discussion through a procedure he had written but rejected during the construction process:

```
op :x + :x + :x + 3
```

Researcher:	*David, can you tell me one of the procedures that you did?*
David:	*I put any number add one and add one again.*
Researcher:	*But what did you have before that?*
David:	*Rather like that, actually [pointing at qwark].*
Researcher:	*You had :x + :x + :x + 3. Would that have been alright, do you think?*
Tom:	*It would have worked. It ought to have worked. Did it work?*
Researcher:	*It gave five which came out as 18.*
Tom:	*Five which came out as 18: but did it work for all the others? [Pause]*
Researcher:	*He tried five and then gave up. So do you think that would have worked?*
Tom:	*It ought to have done. Yes.*

Working with the computer enabled Tom to explore equivalent ways of expressing the same relationship. He appeared to begin by tackling the problem through a highly structured approach, considering the use of brackets to define individual terms. By writing a second (and considering a third) equivalent procedure, he realised that it was possible to rearrange the order of terms, in fact grouping terms in other ways. Tom had discovered through constructing procedures and testing their validity, that it was possible to rearrange the (added) terms in a procedure without affecting its function.

<u>Example: Edward and Nicholas considering equivalence between multiplication and repeated addition - 58 G (3.2)</u>

Task 3.2 was expressed as follows:

Write a Logo procedure which adds three consecutive numbers, e.g. if you input 3, it outputs the sum of 3, 4 and 5.

Discussion of Nicholas's procedure written in Task 3.2

```
to loo :k
op :k + :k + :k + 3
end
```

led onto a spontaneous request to investigate equivalent procedures:

Edward:	*I know a shorter way you could do it. You do double dot whatever you want to put in, then you put times three, then add three.*
Researcher:	*Right.*
Nicholas:	*Will that be right?*
Researcher:	*Do you think that would work?*
Edward:	*Yeah.*
Nicholas:	*I'm not sure.*
Researcher:	*Do you want to try it?*
Edward:	*Yeah.*
Nicholas:	*O.K.*

Edward wrote the following procedure immediately:

```
to jerrr :k
op :k * 3 + 3
end
```

and tried a single value in it:

Input	*Output*
3	12

Nicholas wrote the following:

```
to hj :j
op :j * 3
end
```

again trying a single value:

Input	*Output*
1	3

Nicholas:	*Is yours right?*
Edward:	*Yeah.*
Nicholas:	*Mine's wrong, then, isn't it?*
Researcher:	*Have a look at your procedure, then. Can you work out what's wrong with it?*
Edward:	*Oh yeah.*
Researcher:	*Can you say, Nicholas?*
Nicholas:	*Um.*
Edward:	*I can, I can.*
Researcher:	*Go on, then.*
Edward:	*You've got to add on three at the end.*

Nicholas:	*Add on three? So times three, add three?*
Edward:	*Ten times three equals 30, add three equals 33.*
Researcher:	*Which one do you think is better?*
Edward:	*Um, the whatever letter, times three add three because it's shorter and quicker.*
Nicholas:	*I find the other way better.*
Researcher:	*You think it's shorter and quicker.*
Nicholas:	*Yeah, but I don't understand this one.*
Researcher:	*Do you understand the first one?*
Edward:	*It's just a shorter version because instead of putting, say it was three, three add three add three, you just put times three, instead of adding it three times. It's the same thing. Like in a maths test would you rather have six add six add six add six than ...*
Nicholas:	*Six times four.*
Edward:	*Six times four.*

The two procedure types (additive and multiplicative) were of different degrees of abstraction from the original expression of the task. Nicholas understood the procedure he wrote initially:

op :k + :k + :k + 3

which more closely modelled the idea of adding on consecutive numbers, but could not understand Edward's more mathematically concise procedure

op :k * 3 + 3

The former procedure more closely modelled Nicholas's thinking process - the practical add on, add on method which Nicholas had described as his method of constructing the procedure. He had concretised this expression of the mathematical relationship, but as yet Edward's expression remained outside his collection of concrete expressions of meaning. Construction of the latter procedure required recognition of a mathematical relationship which at that point was outside his experience.

Looking at Examples 64 a), 64 b) and 58, it is possible to discern five ways in which the mathematical relationship around which the task was built was expressed. These were the original natural language expression of the question and four differing but equally well-behaved Logo procedures constructed by the children:

Write a Logo procedure which adds three consecutive numbers, e.g. if you input **3**, it outputs the sum of **3**, **4** and **5**.	Expression 1 (Natural language statement of question.)
$k + k + 1 + k + 2$	Expression 2 (Tom's first procedure)
$k + k + k + 1 + 2$	Expression 3 (Tom's second procedure)
$k + k + k + 3$	Expression 4 (David's, Nicholas's & Edward's first procedures)
$k * 3 + 3$	Expression 5 (Edward's second procedure)

This collection of expressions provides an interesting perspective on the uniqueness of each pupil's web of meanings. For each pupil, a different collection of expressions had become concretised through its construction and verification. For Tom, the concretion process incorporated expressions 1[61], 2 and 3; Edward's collection of expressions comprised 1, 4 and 5; David's and Nicholas's 1 and 4. Such variance does not correspond with hierarchic views of abstraction involving a "moving away" from the "concrete", but rather resonates closely with a view of abstraction as the creation of meaning through the forging of connections. This suggests that within this learning situation, the concretion of expressions of meaning was dependent not on

[61] Although the natural language statement was an expression "imposed" by the researcher.

passing through the stages of some externally imposed hierarchy of abstraction, but rather on the building up of connections between other concrete expressions of meaning with the consequent strengthening of the web of connections.

9.3 CONCLUSION

Thus equivalence appeared to be an important connection, not so much between distinct expressions of meaning - although this might be the case - but between and within mathematical concepts, connecting various closely related expressions of meaning together in order to build up a tightly-knit collection of concretised ideas. This chapter helps to sharpen our understanding of the nature of connections. Paradoxically this sharpening leads to a blurring: the previous two chapters have promoted the idea of a connection as a neat, well-defined link between two expressions of meaning; this chapter suggests that this linkage was a complicated, even ambiguous affair. Using the mathematical imagery of functions, there was no one-to-one mapping between expressions of meaning: the relationship between expressions was as varied as the people who created them.

This chapter also demonstrates clearly the non-hierarchic nature of expressions of meaning, emphasising Wilensky's point that expressions become concrete through their relationships with other concrete expressions and not by virtue of their position in a hierarchy of abstraction.

In Chapter Ten, the final analysis chapter, I turn to the relationship between types of language which featured in the study, and look at the various roles which these language types took.

Chapter Ten: Roles and Types of Language

10.1 Introduction

Throughout the study, pupils expressed mathematical relationships (including generality, justification and proof) in a number of ways using various forms of language. In fact, the use of language underpinned all the other relationships described in Chapters Seven, Eight and Nine and I have touched on its use in those chapters. In Chapter Ten I look at the various types of language which were a feature of the study and attempt to define their roles in the expression of generality and justification.

In order to examine these roles I want to look in turn at the different forms of language which pupils used to express their ideas. I begin by contrasting the use of natural and formal language. Where natural language was used to explain mathematical relationships contained within a generalised procedure, explanations were frequently unstructured and it appeared difficult for pupils to communicate with one another; where pupils used the formal language of the procedure, explanations appeared clearer and communication between pupils - and with the researcher - became more effective.

I then introduce the concept of naturalised formalism, a form of webbing used by pupils to express mathematical ideas, structured by use of a formal Logo procedure but expressed in natural language.

10.2 COMPARING THE USE OF NATURAL AND FORMAL LANGUAGE

In some examples, pupils attempted to explain mathematical relationships contained within their procedures using the natural language of the question statement. The discrepancy between the formal mathematical structure of, for example, a Logo procedure and the grammatical conventions of written (or worse, spoken) English might account for the ambiguity of statements such as the following attempt by Matthew to explain how a procedure worked.

Example: Matthew expressing mathematical relationships contained within his procedure by means of natural language, although failing to communicate his meaning effectively - 15 (D 3.2)

In Task 3.2 (writing a procedure to find the sum of three consecutive numbers) Matthew had written the following procedure:

```
to task2 :a
op :a + :a + 1 + :a + 2
end
```

On request he attempted to explain how his procedure worked in terms of a generic example:

Matthew: *Well, first you've got to get the four and plus the three and four, so any number plus one. So that'll do three again plus one, and that'll add that to three. Then you've already got four so then you plus four again. Then plus two.*

Such an explanation may be analysed as follows, but although clearly influenced by the question statement, it remains largely unstructured:

Well, first you've got to get the four...	4
...and plus the three and four...	$3 + 4$
...so any number plus one.	$a + 1 = 4$
So that'll do three again plus one...	$3 + 1$
...and that'll add that to three.	$3 + 3 + 1$
Then you've already got four...	$3 + 4$
...so then you plus four [three][62] again...	$3 + 4 + 3$
Then plus two.	$3 + 4 + 3 + 2$

There was certainly no element of illumination in Matthew's explanation. In fact, such informal language created a barrier to communication and frequently failed as a means of expressing mathematical relationships.

It is interesting to contrast this with the next two examples which focus on formal rather than natural language. In the first example, a pupil made use of formal language to express mathematical relationships and in the example which follows, the formal language became a means of communication between the two pupils involved and the researcher.

Example: Edward expressing a justification of his generalised procedure in the language of construction - 34 (G 4.3)

In Task 4.3 (a procedure for summing two even numbers) Edward constructed his procedure as follows:

```
to jeeeem :j :h
op :j + :j + :h + :h
end
```

He verified this with the following values:

[62] Matthew presumably meant "three" rather than "four" at this point.

Inputs		*Output*
6	8	28
5	7	24
9	4	26

He was able to prove that this procedure would always output an even number in an expression of proof that was entirely structured by the structure of that procedure and couched in formal algebraic language, i.e. the language of construction:

> *Edward:* *I've just added the :j to the :j which will equal an even number; then I've added :h to :h which will equal an even number and I've added them both together and equalled an even number.*

This proof was expressed using formal language because it proceeded from his construction and verification of the procedure and because it equipped him with the metaphor of input and output. The act of symbolisation by which the procedure was created had endowed the symbolism with a rich association of meaning, so that rather than the symbols (the formal language) being a barrier to expression of the mathematical relationship they defined, they became the obvious means of expressing that relationship. Because of their rich association of meanings, they were a resource to be used for further exploration of the mathematics and as a means of expressing generality and proof. It is probably fair to say that Edward could not have made this simple proof had he not first represented the mathematical relationship in a formal way such as a Logo procedure.

In fact Edward had to restructure his procedure in order to prove its validity: he could not read from left to right but had to view the structure of the procedure as a whole in order to extract the fact that the sum of the first two terms was even, that the sum of the second two terms was even, and that the sum of these two even terms was itself even. The ability to stand back from a procedure, to see it as a whole - an object - and then to manipulate that object implied that the symbolic procedure was a concrete entity in Edward's eyes. This concretion was the result of Edward's experience of symbolisation and verification.

So the use of formal language here facilitated the expression of generality and proof. It was in fact a connection between the formalised statement of a mathematical relationship as defined by a Logo procedure and the articulation of some generalisation about that relationship. That the language of expression was also the language of construction was a significant feature of this environment. This phenomenon has been described as "auto-expression", i.e. a language used for construction which contains the means for its own expression (Noss and Hoyles 1996).

Formal language also provided a means of communication. Standard approaches to the teaching of algebra rely either on the formal language of "paper and pencil" algebra as a means of communication common to teacher and pupil (although in practice pupils often have insufficient facility with formal algebra to use it in this way), or on semi-formal symbolisations designed by pupils themselves in order to express emergent concepts of generality (which, by the very nature of their subjective genesis, contain few elements of commonality). Logo, the symbolic language in which procedures were written and mathematical ideas and relationships expressed, appeared to provide that commonality. Using the idea of the "mathematical community of the classroom", the use of Logo provided the members of that community - in the case of this study the two pupils in any one group plus the researcher - with a common means of communication.

<u>Example: Ben, Matthew and the Researcher sharing a common language for communication in Logo procedures - 20 (D 4.2)</u>

In Group D's work in Task 4.2, which called for a procedure to generate odd numbers, Ben first wrote

```
to odd :a
op :a * 2 - 1
end
```

and then on request from the Researcher attempted a second, equivalent procedure:

```
to odd2 :a
op :a * 2 + 1
end
```

Matthew wrote

```
to odddd :a
op :a * 2 + 1
end
```

Matthew verified his own procedure **odddd** as follows

Input	*Output*
1	3
9	19
6	13
4	9

and later tested Ben's procedure **odd** with the following values:

Input	*Output*
0	-1
4	7
2	3
9	17
7	13

These two sets of data derived from empirical verification served to highlight and define the difference in the two procedures:

Matthew:	*I can guess what the difference is.*
Researcher:	*What do you think it is before you look.[at the procedures]*
Matthew:	*Well, instead of ... I'm doing plus one in mine and I think he's doing times nine, take one, or something. An odd number, take ... a number.*

Ben:	*No.*
Researcher:	*Have a look.*
Ben:	*Well, it isn't. It is take one. The one that you were using was* ***odd.***
Researcher:	*How do these work, then?*
Ben:	*Well, it's just :a times two is in the two times table - it's an even number.*
Matthew:	*So add one or take one: that'll change it to an odd number.*
Researcher:	*Can you think of yet another procedure that'll do the same thing?*
Ben:	*:a times two add three? [laughing] :a times two add nine. It's :a times two add or take any odd number.*
Researcher:	*Could you be even more general?*
Ben:	*No.*
Researcher:	*Yes, you could!*
Ben:	*Or :a times any even number take away any odd number.*
Researcher:	*Brilliant!*

This nice little exchange again shows Logo as a common language, it shows how empirical verification of a Logo procedure structured the pupils' thinking and their arguing, it shows how the procedure structure which they had created provided a focus for further discussion and it shows how this procedure structure could be used as a model for extending and a resource for expressing their ideas. Interestingly, Ben *chose* formal language to express his ideas in the last few lines, a language which was easily and precisely communicated to the other members of the mathematical community.

It is, of course, the power of algebra that it encapsulates mathematical relationships in a clear and concise manner, provided, that is, that one is sufficiently conversant in the relevant algebraic language to make sense of this clarity and conciseness. The comparison between :a * 2 + 1 and :a * 2 - 1, two algebraic expressions which neatly encapsulate the mathematical essence of oddness, focuses attention on the added/ subtracted constant and so leads to the further discussion and articulation of generalisation. The important point again, however, is that these expressions were the result of pupils' experience of procedure construction, supported very evidently here, by the empirical testing of values. The ultimate expression of oddness as ":a times any even number take away any odd number" was a product of the learning situation.

One indicator of the rich association of meanings which the symbolism acquired was the apparent lack of confusion between process and result. An area highlighted as problematic in traditional paper and pencil algebra is appreciation that, say, 2x can denote both a process (2 multiplied by x) and a product[63]. The set of tasks in the second session, for example, asks pupils to interpret natural language statements as formal functions, and then to explain the effect of the function. As one of the pupils observed, it was necessary for them to interpret the verb "double" as x * 2 or as x + x, so making explicit the mathematical relationship contained within the term. Similarly, statements of the type: "Well, number times two plus number is three of the numbers. Divided by three is the number and divided by the number is one[64] " (Tom working at Task 2.1), demonstrate an easy fluency in interpretation between process and result.

10.3 NATURALISED FORMALISM

There was, however, a third way in which language was used to explain and to communicate. Here an explanation was shaped by the formalised procedure which formed the starting point for the mathematical relationship under discussion and not by the natural language statement of the question, yet the explanation was expressed in natural language. This use of language, which I shall refer to as *naturalised formalism*, was neither purely natural language nor a formal mathematical statement, but something in between the two. It appeared to provide a linguistic medium in which formal relationships might be structured and explored without requiring a rigorous command of mathematical formalism. Consider the following two examples.

---

63 For a discussion of the dilemma between process and product see Davis (1975), Sfard and Linchevski (1993) and Gray and Tall (1994)

64 "Number" used here without article is the variable :v on the screen in front of the pupils.

Example: Edward explaining the behaviour of his procedure by means of naturalised formalism - 55 (G 2.1)

In Task 2.1 Edward attempted to explain why his procedure

```
to jem :d
op (:d * 2 + :d) / 3 / :d
end
```

always produced an output of one:

Edward:	*It works with any number ... [looking at the procedure on the screen] because whenever you times a number which you started off with by two then add the number you started off with, it's more or less like timesing by three.*
Researcher:	*Right. Carry on.*
Edward:	*Because then you divide it by three, equals the number you started off with - it's just like doing the three times table. Then if you divide by the number you started off with it will always equal one.*
Researcher:	*Great.*

Example: Luke and Jamie describing how Luke's procedure worked, first (Jamie) in terms of a generic example, and then (Luke) in naturalised formalism - 55 (E 2.2)

In Task 2.1, Luke constructed the following procedure:

```
to plum :o
op (:o * 2 + :o) / 3 / :o
end
```

Researcher:	*Can you explain why this procedure always comes to one?*
Jamie:	*Five, times two is ten, add five is fifteen, divide by three is five, divide by the number you first thought of is one.*
Luke:	*Well, because if you do the number you thought of divided by the number you thought, it equals one. By doing this procedure you've got the number you first thought of times the number you first thought of, which is the same as times three. Then you've got divided by three, then you divide by the number you first started with which is one.*

In these examples pupils did not manipulate the symbols of an algebraic function as might be the case in a paper and pencil example (or indeed in the examples of formal language quoted in § 10.2), but manipulated the relationship defined by the symbols. This relationship was expressed in a kind of formalised natural language which commonly served as an intermediary language between the natural language of the question and formal algebraic expression. This language might be termed "naturalised formalism", a description which seems more apt than "formalised naturalism" since it owed more to the formalised structure of the procedure than to the original natural language expression of the question. A feature of naturalised formalism was that through it pupils were able to express generality and even prove a result, as Luke and Edward did in the statements quoted above.

These two examples demonstrate statements in naturalised formalism as final statements in the course of completion of a task. Expression of a mathematical relationship in naturalised formalism might also lead to the re-expression of that same relationship in formal terms.

<u>Example: Tom using naturalised formalism to re-shape his generalised expression and ultimately to re-express it in formal language - 40 (H 4.5)</u>

In this example Tom's naturalised formalism explanation acted rather like generic structuring[65] in that through it he shaped his expression of the function's properties to such an extent that he was then able to re-express the relationship in formalised language. The Researcher's role in encouraging him to articulate a formalised statement was clearly of central importance.

Tom's procedure in Task 4.5 (the summation of an odd and an even number) was as follows:

[65] See Chapter 8, § 8.3

```
to pip :e :m
op :e * 2 + :m * 2 + 1
end
```

He had verified this as follows using the knowledge that the sum of an odd number and an even number is an odd number:

Tom: *It ought to come out as an odd number: odd plus even is odd. Yes, because even plus even makes even, so even plus odd must make one more or less than even.*

Inputs		*Output*
2	4	13

Tom: *Four becomes eight; eight becomes 17; eight add 17 is 25*

Inputs		*Output*
4	8	25

Tom: *Yes. Right, six and nine: six will become 12; nine becomes 19: 31.*

Inputs		*Output*
6	9	31

Tom: *I'm satisfied*
Researcher: *What sort of number are you getting coming out?*
David: *Odd.*
Tom: *An odd number.*
Researcher: *Can you explain from looking at your procedure why?*
Tom: *Because you're making an even number and an odd number and an even and an odd number is an odd number.*
Researcher: *Yes, but that doesn't explain why that works.*
Tom: *Because you know that if you times something by two it becomes an even number, so you times by two again to get the even number and we know that one more than an even number is an odd number. So times two makes an even number plus one*

	makes an odd number: add the odd number and the even number and you get an odd number.
Researcher:	*That doesn't quite explain why. :e times two is an even number, yes. David. any ideas?*
David:	*No.*

On being pressed by the researcher for an explanation, Tom restructured his procedure to make the following statements:

Tom:	*Well, it would be just like adding two numbers times two together only adding one on the end.*
Researcher:	*Right. Say that again.*
Tom:	*So it's just like taking two even numbers, timesing them both by two and adding one.*
Researcher:	*Can you say that in relation to the :e's and the :m's.*
Tom:	*:e times two is an even number; :m times two is an even number; and you're adding those both together and adding one.*

Here naturalised formalism was acting as a connection between expressions of meaning: it was providing a link between the formal Logo procedure (which Tom himself had constructed) and his formalised articulation (referring to that procedure) of why it would always output an odd number (i.e. an explanation of the mathematical relationship contained within that procedure).

Naturalised formalism displayed close parallels with generic structuring: generic structuring involved the expression of generality through the exploration of a procedure's behaviour in terms of specific and generic examples; naturalised formalism was used to shape the expression of generality in formal terms - expression through naturalised formalism clarified and defined relationships before they were expressed in formal language.

10.4 CONCLUSION

Language, mediated by use of the computer, was used in at least two ways as a connection between expressions of meaning. Formal language, rather than acting as a barrier, became a connection between formalised algebraic statements and articulation of generality or even proof of that statement. What seemed to be important here was the association of meanings the formal language acquired through the construction process: the formal symbolism was concretised through acts of symbolisation and was thus a natural means of expression when the procedure was analysed and ideas of generality and proof discussed.

What I have described as naturalised formalism adds an extra element to this relationship. The commonality of the language of construction and the language of expression appeared to enable pupils to express generality using their constructed procedures as a resource for expression. However, naturalised formalism seemed in some cases to act as an intermediary stage between a constructed Logo procedure and the articulation of generality and justification in formal language: it webbed the expression of formal relationships. Generic structuring was a means of exploring and clarifying structure through recourse to expressions which pupils might find more concrete; in a similar way, naturalised formalism combined the concrete associations of natural language with the formal structure of a procedure (itself concretised through an act of symbolisation).

Chapter Ten is the last of the four chapters which deal with analysis of data. In Chapter Eleven, the final chapter, I draw conclusions from these analyses and suggest areas for further research.

CHAPTER ELEVEN: CONCLUSION

11.1 INTRODUCTION

In this final chapter I look once again at the aims of the research and summarise the findings in the light of these aims. I then look in more detail at these findings. This section I group under three headings, numbered 11.3.2 - 11.3.3:

- creating generality;
- structuring generality; and
- webbing generality and justification.

Finally, after a review of the important role of the researcher in the thesis, I look at the limitations and implications of the research and discuss possible areas for further work.

11.2 AIMS & PRINCIPAL FINDINGS OF THE THESIS

The overall aim of the thesis was to observe children's creation of meaning through the expression of mathematical generality and through the justification of that generality. Review of the literature suggested that this might best be observed through consideration of children's *expressions of meaning* and through the *connections* they made between these expressions of meaning.

Three theoretical concepts - elaborated in Chapter Two - were dominant in determining the design of the learning situation:

i) *concretion* (Wilensky 1993) - objects becoming concrete by virtue of the learner's relationship with that object;

ii) *webbing* (Noss and Hoyles 1996) - a structure which learners can draw upon and reconstruct for support; and

iii) *situated abstraction* (Noss and Hoyles 1996) - the way in which learners draw on the webbing of a setting to construct mathematical ideas and the way in which that webbing shapes the expression of ideas. A Logo programming environment was chosen as providing a window onto pupils' construction of meaning

Based on these theoretical considerations, the overriding research issue was stated as being to investigate how use of a carefully designed computer programming environment structured children's expression of generality and the justification of that expression, and from this issue three aims were identified:

i) to investigate ways in which children expressed generality using the programming medium;

ii) to investigate connections between different expressions of generality made in different modalities and their role in children's creation of mathematical meaning;

iii) to investigate the role of connections between expressions of generality in children's attempts to justify their constructed generalisations.

These investigations were carried out through the creation of a learning situation which centred round the creation of Logo procedures to model simple functions. This setting was chosen as opening a window onto children's constructions of meaning for generality. The starting point for each activity was a mathematical relationship expressed either in natural language or in a table of input-output pairings. The activity thus involved generalising that relationship in terms of a Logo procedure. A further dimension was added - one designed to explore children's justification of their generalisations - in that the activities were treated as a competitive game in which it was necessary for children to prove or demonstrate to their partner that their procedure was either as good as or better than their partner's.

The findings of the thesis weave around the dual themes of generality and particularity; abstract and concrete; and formal and informal. They focus not on these relationships as dichotomies but throw light on them as complementary partnerships in the creation of meaning. The principal findings were as follows:

i) within the learning situation defined in the study children were able to make formalised generalisations of mathematical relationships, often webbed by what I have termed "semi-generalisation";
ii) the expressive powers of the symbolisation, which was both a goal and a means of expression within the activities, achieved a more functional role by the symbols' association with a history of specific numerical examples;
iii) children constructed situated abstractions for the justification of generality webbed by what I have defined as "generic structuring" and "naturalised formalism";
iv) the apparent "rift" between empirical and deductive starting points for generalisation, justification and proving activities appeared less clear than the literature suggests.

Each of these points is discussed in the following section.

11.3 DISCUSSION

11.3.1 Introduction

In determining how children concretise a mathematical object, I have introduced Wilensky's idea that this is achieved not through "some intensive examination of the object, but rather an examination of the modes of interaction and the models which the person uses to understand the object" (Wilensky 1993, p 198)[66]. In the following sections I explore through data from the study these modes of interaction and the models the pupils used. In §11.3.2 I describe the symbolisation process through which pupils engaged with the initial mathematical problem in each of the tasks, and examine the process of semi-generalisation which they frequently used to web the symbolisation process. I draw parallels with a form of webbing described by Noss and Hoyles (1996) as *flagging* and suggest how semi-generalisation extends this concept.

In the next section (§11.3.3) I look at how the children used the auto-expressive quality of the computational environment to analyse and justify their procedures, suggesting how my research brings a new dimension to the idea of auto-expression. I also look at how the computational environment itself appeared to motivate the participants' justification of generality.

In the third section (§11.3.4) I explore two important processes which children used to concretise mathematical relationships: generic structuring and naturalised formalism.

11.3.2 Creating Generality

Act of Symbolisation

The central act of each task carried out by the pupils in the study involved the creation of a procedure expressed in symbolic algebraic language. Creation of such a procedure required a generalisation of the mathematical relationship defined in the task, that relationship - the starting point for each of the tasks - being a problem

[66] See Chapter 2.

expressed in natural language (or in a few cases a collection of numerical values). Thus each task involved an act of symbolisation which attempted to create a connection between a problem expressed in natural language and the expression of that problem in formal algebraic language. The feature of the programming medium which appeared to facilitate this act of symbolisation was the ability it afforded pupils to verify empirically the emergent procedure at each iteration of its construction.

In addition, the act of symbolisation appeared to carry with it a number of important consequences:

i) it produced descriptions of procedure behaviour shaped by the formalised procedure rather than the natural language statement of the question. When discussing a task, pupils referred and pointed to - and sometimes used the language of - their procedures, rather than referring back to the natural language statement of the original question;

ii) it provided a rich association of meanings for the symbolism. Pupils had used the symbolism as a means of constructing the procedure and they had verified the symbolism empirically;

iii) translation into symbolic form was perceived as a helpful act - for many pupils it appeared to clarify mathematical relationships;

iv) it appeared to create in pupils not just acceptance of but a perceived need for symbolic mathematical expression

Semi-generalisation

The programming medium provided a powerful piece of webbing for the construction of generalisations in what I have termed semi-generalisation. Semi-generalisation is the creation of a partially correct procedure which is used as a basic framework - what might be termed a partly-functioning prototype - for further work. There were three important features of semi-generalised procedures:

i) a procedure was as generalised as necessary for further work on that procedure, be that work further efforts towards complete generalisation, or further work on the structure of the procedure;
ii) the importance of the computational medium was its role in capturing the working model in a kind of limbo between specific and general which nevertheless made available the possibility of empirical verification;
iii) with the computer's preservation of the semi-generalised procedure and with use of a constant input, the procedure could be seen as an object which could itself be manipulated;

It is also possible to identify a number of features relating to the process of construction and to the use of semi-generalisations:

i) further adjustments to the procedure might proceed in a piecemeal fashion: pupils' attention could shift from one aspect to another and they did not need to tackle all aspects of the procedure simultaneously in order to correct the procedure;
ii) validity of a partially correct procedure might be tested by pupils within any range of parameters defined as necessary for the procedure to be semi-generalised, for example a severely limited domain of input (commonly confined to one value);
iii) completion of the procedure became a process of homing-in, supported by easily achieved verification;
iv) complete generalisation of the procedure might not proceed beyond this level, or it might happen that when the bugs in the procedure had been removed, the pupil would in fact make the procedure fully generalised;
v) generalisation of a semi-generalised procedure might also occur as part of the justification process, so a semi-generalised procedure might provide a good

focus for analysis and discussion.

Semi-generalisation contains close parallels with what Noss and Hoyles describe as *flagging* (Noss and Hoyles 1996 , p 110), in that the "critical features of the problem were marked, modified and finally built into a mathematical structure, but one that could be modified later." Where semi-generalisation extends the notion of flaggingis in the close connection which is maintained between generality in the form of the emergent Logo procedure and the specificity of the empirical verification process; and in the way children were able to use the semi-generalised procedure as an object which could be manipulated prior to completion of the complete generalisation. There is a sense of fluidity in the learning process here: unbound by the fetters of routine and acknowledged procedure, but with the gap between what lay within the children's experience and the final expression of mathematical relationships webbed by the computational medium, children appeared able to explore and broaden their expression of generality in diverse, yet ultimately complementary ways.

Expression of generality in the study took the form of re-expressing natural language statements as a formalised mathematical relationship through writing a Logo procedure. The connection between natural language statements and formalised Logo procedures was the act of symbolisation, this act frequently webbed by semi-generalisation.

Figure 11.1 depicts this relationship. The two expressions of meaning - natural language and Logo procedure - are represented as being linked by an act of symbolisation. The double-headed arrow is used to show that the relationship is two way, the act of symbolisation both formalising the natural language statement in terms of a Logo procedure, and enriching the Logo procedure in terms of the symbolism's strong links with the original natural language statement. Semi-generalisation is shown as "floating" between the two to depict the support offered by this powerful piece of webbing. It should be noted that the placing of the various components of the diagram is not designed to convey a sense of hierarchy.

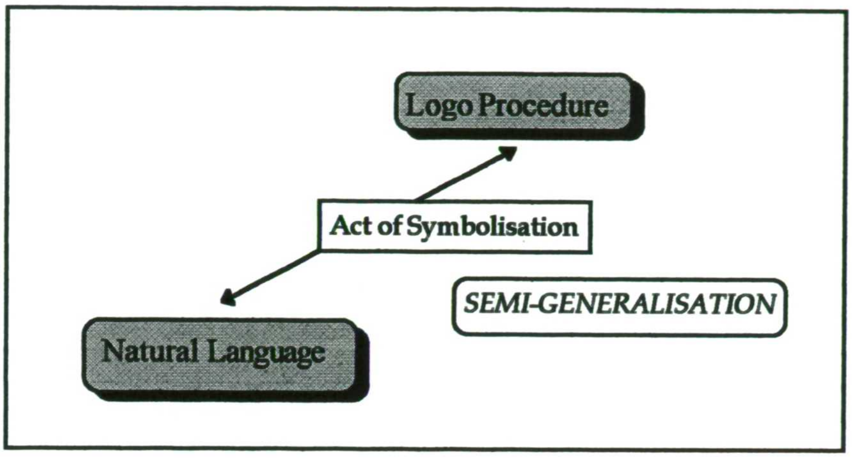


Figure 11.1: Semi-generalisation webbing symbolisation - the arrows represent two expressions of meaning linked by the act of symbolisation, the process webbed by semi-generalisation.

11.3.3 Restructuring Generality

The design of the learning situation incorporated a game context whereby pupils' procedure construction was intimately bound up with the necessity to demonstrate the superiority of one procedure over another. In many cases this demonstration consisted of pupils testing values, i.e. empirically verifying their procedure; where, in addition, some element of analysis of the procedure structure was involved, pupils were involved in the *justification* of that procedure. In this situation all analysis of procedure was in fact also a justification of that procedure's validity, and consequently, analysis of procedure structure and the motivation behind that analysis are vital windows onto pupils' ability and desire to justify. It is interesting to note that there was no clear distinction between use of empirical verification as a means of demonstrating the validity of a procedure and the use of analysis of procedure structure. Frequently both were used side by side, one backing up the other:

empirical verification might be used to "back up" verbal justification; verbal justification might be used to enhance a point made during empirical verification. In this situation a least, the apparent gulf between empirical and deductive failed to appear.

Analysis of Structure

The central importance of the relationship between construction and verification has already been highlighted. This relationship also heavily influenced the ability to analyse the structure of a procedure. Features of construction and verification which promoted the analysis of structure were:

i) iterative term-by-term (re)construction and verification where the empirical verification of a procedure informed the subsequent reconstruction of that procedure; and

ii) using the calculator to create a tabular presentation of empirical verification, i.e. the fact that the input/output pairings of the verification process appeared in a tabular format, the ordering of which was under the control of an individual pupil;

Another key to the effective analysis of structure was term-by-term verification, carried out largely through mental evaluation of a procedure under a specific input value. Here less direct use was made of the computer.

Roles of Symbolism

Of primary importance to the analysis of a procedure was the role and use of the symbolism itself. The power of the act of symbolisation - the building of procedures on the computer screen - extended beyond simply forging links between two expressions of a mathematical statement. The data provides examples of the symbols of the constructed procedures acquiring a significance which was clearly structured by the medium in which the language was embedded:

i) experience of constructing and verifying the procedure had transformed the symbols into a *means of expression*;

ii) the verification process had endowed the symbols with a history of several specific numerical examples, although they remained a generalised expression of a mathematical relationship.

Using the symbols in this way - as both a means of expression and as abstractions of a recently concrete experience - pupils were able to restructure a procedure (i.e. manipulate its symbols) into a new form, from the structure of which justification and even proof of that procedure appeared to become clear. This, of course, is a key facet of the process of formal proof. The important characteristic of these cases was that the restructuring of the generalised statement (the procedure) necessary to its justification was facilitated by the expressive quality of the abstract symbols, albeit expressed in naturalised formalism rather than in the language of construction itself.

Analysis of procedure structure exhibited the following features:

i) Logo was used as a common language - a common reference point for discussion and a common means of communication; common that is, to the pair of pupils participating in any one task and to the researcher;

ii) empirical verification of a Logo procedure structured some pupils' thinking and their arguments;

iii) procedure structure provided a focus for further discussion; and

iv) procedure structure was used as a model for extending and a resource for expressing pupils' ideas.

Although not a central theme of this thesis, it is interesting to note in passing that some pupils *chose* to use the formal language of their procedure to express their ideas in preference to natural language.

The formalised procedure provided a framework for analysis and discussion: its structure appeared to focus discussion, whether spontaneous discussion on the part of the pupils, or the result of an artificial focus created by the researcher. (The commonality of the language provided an important starting point for dialogue between researcher and pupils). The algebraic expression (in contrast to the natural language expression) made consideration of the function bit by bit possible: it provided reference points for analysis. Pupils often appeared to grapple with the natural language expression of the question in a manner lacking focus and direction; the formal expression of the procedure lent itself to concentration on small chunks (the focus often provided by the researcher). As a means of support, the researcher could isolate a section of the procedure and either explain it outright, or illicit an explanation through discussion with the pupils. Attention might then shift (or be shifted) to another section which pupils might then find easier to explain. The researcher could explain one chunk as an aid to pupils' progress. Also chunks might be rearranged in order to investigate mathematical relationships contained within the function. Now, of course this is a routine part of traditional paper and pencil algebra, and it would be wrong to ascribe any unique advantage to Logo algebra over paper algebra in respect of symbol manipulation. The value of the learning situation lay in the rich association of meanings the Logo symbols acquired through their use in the construction process and through empirical verification of the procedure and the way in which the Logo meaning provided an anchor for analysis and the subsequent expression of justification.

New Perspectives on Auto-Expression

A computational medium in which the language of construction is also the language of expression and justification of generality (i.e. one which contains the "elements of a language to talk about itself") may be described as auto-expressive (Noss and Hoyles 1996, p. 69). However, the use of language in this study adds an extra dimension to the idea of auto-expression. For here it is not merely the *language* which facilitates expression of generality, but it is the use of that language interpreted through its relationship with the specificity of recent numerical verification. The symbolism of the language is a generalised expression of a mathematical relationship, but it has extra power by virtue of the symbols' association with a history of specific numerical examples. Now this association increases the situatedness of a procedure in that it is firmly linked in a pupil's experience to specific numerical examples, but it nevertheless remains a generalised expression, as does the justification of that generality, within the specific learning situation.

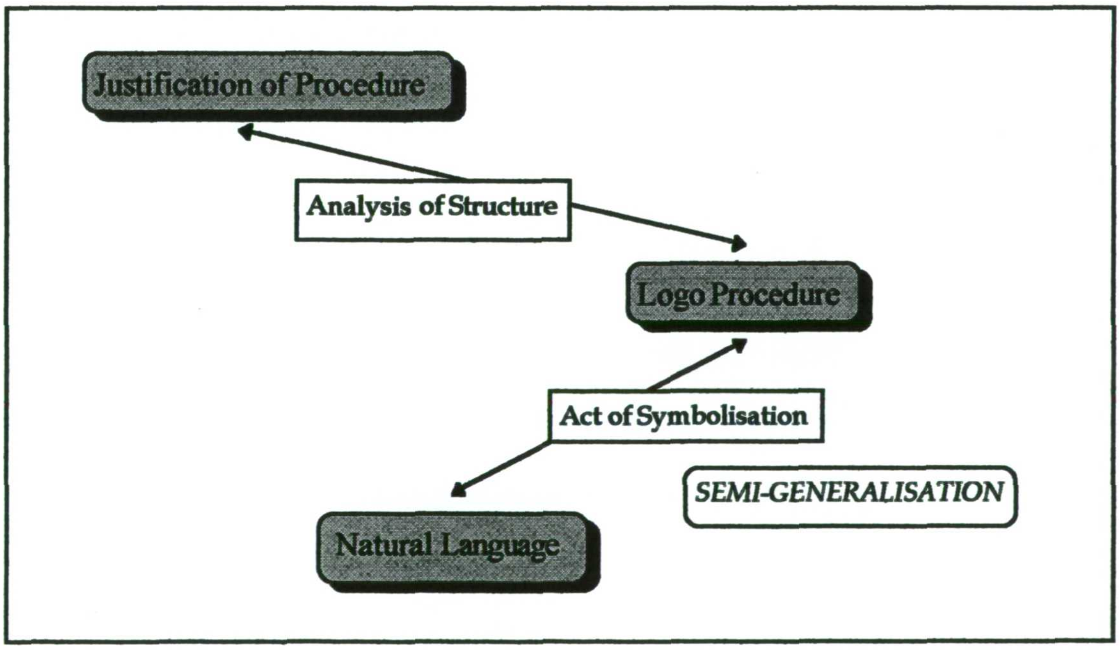


Figure 11.2: Justification of Generality - the arrows represent expressions of meaning linked by the act of symbolisation and analysis of structure.

In this situation, then, use of the auto-expressive language became more powerful by its situation; by a paradox the ability to generalise is heightened by a stronger relationship with the specific.

Figure 11.2 extends the web of expressions of meaning and connections, shown in Figure 11.1, to include justification of procedure generalisation effected through analysis of structure. Again the two-headed arrow is intended to denote a two-way relationship: analysis of procedure structure leading to justification of a procedure's generality; and justification of the procedure confirming the generality of the procedure.

11.3.4 Webbing Generality and Justification

Generic structuring

Generic structuring is the process of gaining a clearer picture of a procedure's structure through recourse to examples which give an insight into the general through the particular; it is the connection between two expressions of meaning, namely the verification of a procedure with a specific value and a generalised statement about that procedure, expressed in natural language, formal algebraic language, or indeed in naturalised formalism. I use the word *generic* because it makes use of examples which are chosen not for their specific values, but as representatives of their class; and *structuring* because it is used as a means of shaping experience of specific empirical verification into the expression of some general statement, or for the examination and clarification of an emergent generalisation.

Generic structuring is intimately bound up with the rehearsal of a procedure's behaviour with particular inputs from which the pupils obtained a general feeling of the function and through which they highlighted patterns. As the pattern was spotted, the example ceased to be simply a *description* of what happened with a particular value in a particular case. The particular value became representative of values in its class and the example became a generic *explanation* of how the function behaved, using a value as an example. This use of procedure structure and the formation of generic examples constitutes generic structuring.

The point at which an example took on a generic, explanatory quality rather than a particular descriptive quality was not always easy to discern. In fact identification of pupils' use of examples as generic could rest on subtleties of emphasis and tone of voice: a particular description might be recited almost monotonously; the use of a generic example might be characterised by a purposeful, directed tone of voice, leading to a "triumphant" conclusion, as if to say *quod erat demonstrandum*. This might sound fanciful, but responsiveness to tone of voice is, I believe, an essential facet of the researcher's observational role in work such as this.

The most important element necessary in any generalisation would appear to be an awareness of the structure of the procedure being explained. This is another criterion which may help to distinguish between generic and particular examples: an explanation which casts some new light on the structure of a procedure (for example by rearranging the written order of terms), but uses a particular input, is very likely using that input as representative of its class and is a result of generic structuring.

Generic examples were expressed in both natural language and in the formal algebraic language of procedure construction. At no point in the study did pupils appear to lack the means to express generic examples. This is in direct contrast to Balacheff's expectation of problems in expression. Balacheff, in speaking of generic examples, says:

> *The reasons for the truth of an assertion are made explicit by means of operations or transformations on an object chosen not in its own right but as a characteristic representative of its class. Problems may well be presented here by the difficulty on the part of pupils to express mathematical relationships either in natural language or in formal algebraic language.*
>
> (Balacheff 1988, p. 219)

Scrutiny of the data suggests that the apparent relative ease of expression found in the study might be put down to the closeness in form between generic example and particular example, and the impression that the expression of particular examples is so heavily dependent on procedure structure, which itself is the result of pupils' construction and testing of their own procedures - the result of their own act of symbolisation and empirical testing.

Generic structuring took a variety of forms: sometimes through the use of generic examples it was used as a starting point for the expression of generality, sometimes as a fall back in cases where pupils were unable to sustain a more generalised explanation, but more usually it underpinned the formation of emergent generalisations and justifications of generality, with the attention continually passing from the specific to the general. In this last case, pupils sometimes used generic examples as a reference point: if they lost their bearings in a more abstract explanation with which they were not as yet fully comfortable, the generic example might provide familiar, concrete territory from which to build further.

Naturalised formalism

In examples of what I have called naturalised formalism, an explanation was *shaped* by the formalised procedure which formed the starting point for the mathematical relationship under discussion and not by the natural language statement of the question, yet the explanation was *expressed* in natural language.

Naturalised formalism was characterised by the following features:

i) pupils did not manipulate the symbols of an algebraic function as might be the case in a paper and pencil example, but manipulated the relationship defined by the symbols;

ii) it expressed a mathematical relationship in a kind of formalised natural language which commonly served as an intermediary language between the natural language of the question and formal algebraic expression; and

iii) through its use pupils were able to express generality and even prove a result.

Examples from the data show naturalised formalism acting as a connection between expressions of meaning, acting as a link between a formal Logo procedure and a pupil's formalised articulation of why that procedure would always behave in a certain manner (i.e. an explanation of the mathematical relationship contained within that procedure).

The parallel here with generic structuring is striking: in examples of generic structuring pupils expressed generality and justifications of generality through the exploration of a procedure's behaviour in terms of specific and generic examples; naturalised formalism acted as a shaping medium for the expression of generality in formal terms - expression through naturalised formalism helped clarify and define relationships prior to their expression in formal language.

Naturalised formalism appeared to act as an intermediary stage between a generalised Logo procedure and the articulation and justification of generality in formal language. Rather in the way that generic structuring provided a means of moving towards expressions which pupils might find more concrete in order to explore and clarify the structure of a procedure, so naturalised formalism appeared to provide a means of exploring the structure of a Logo procedure through a means of expression which combined the concrete associations of natural language with the formal structure of a procedure which had itself become concretised through an act of symbolisation, namely the construction of that procedure.

If concretion is the "process of the new knowledge coming into relationship with itself and with prior knowledge" (Wilensky 1993, p. 201), then generic structuring and naturalised formalism are two important means of interpreting that process. They also enrich our understanding of the idea of situated abstraction (Noss and Hoyles 1996) in that they provide an insight into two ways in which pupils appeared to web generality and justification within this particular setting. In this study, expressions of generality and justification were clearly webbed by generic structuring and naturalised formalism, and indeed by semi-generalisation.

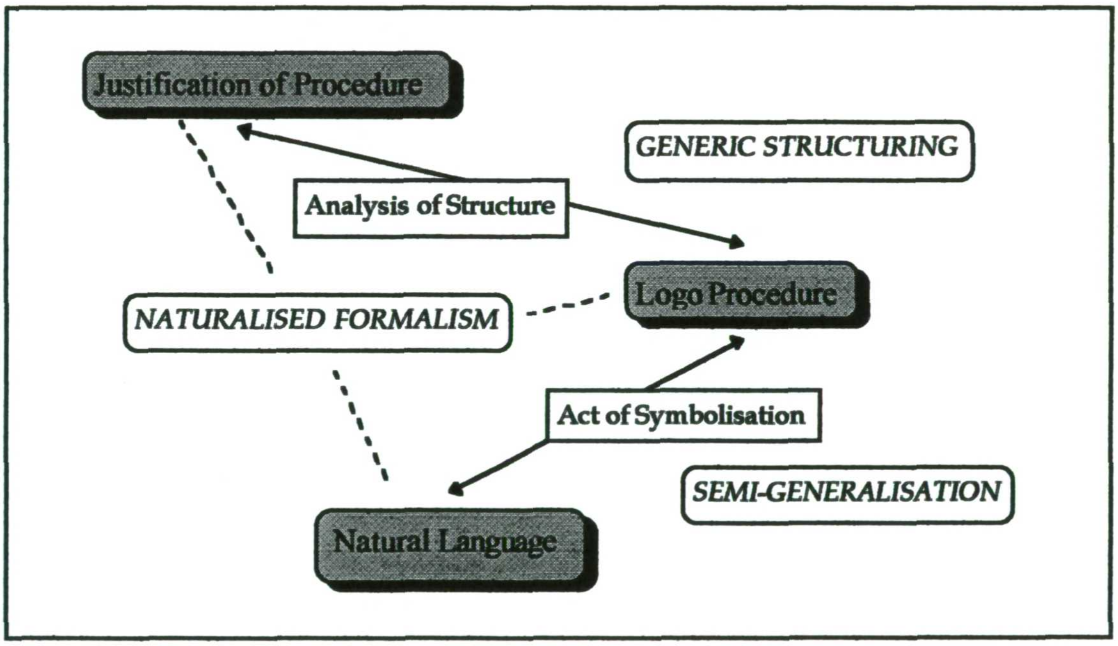


Figure 11.3: Justification webbed by generic structuring and naturalised formalism - building on Figures 11.2 and 11.3, generic structuring and semi-generalisation are shown webbing analysis of structure and act of symbolisation, respectively, and naturalised formalism is depicted, by means of a dotted line, as providing a form of webbing which links natural language, formalised Logo procedures and justifications of the generality of those Logo procedures.

Figure 11.3 expands the two previous diagrams to include generic structuring and naturalised formalism. Like semi-generalisation, these are depicted as "floating" in order to show that they constitute much used but by no means indispensable means of webbing justification of generality. Naturalised formalism is deliberately placed within the triangle of *natural language - logo procedure - justification of procedure* and joined to them by dotted lines to depict how through naturalised formalism, justification of a procedure's generality might be shaped by the logo procedure, yet expressed in natural language.

11.4 A REVIEW OF THE RESEARCHER'S ROLE

The role of the researcher in this study had a crucial significance. A rationale behind the interventions made by the researcher is described in Chapter Four[67]. Maintaining the right balance proved a difficult job: the balance between instructive, supportive and exploratory intervention was important and one determined to a great extent by the researcher's goals. As Noss and Hoyles observe:

> *When the connections to the teacher's goal are dominant, students may simply imitate a procedure or try to guess what the teacher wants; when they are not made at all, the mathematical agenda might be bypassed (although if we recognise diversity and reject the notion of a unique line of development, the situation is not so brittle).*
>
> (Noss and Hoyles 1996, p. 109)

Whilst stressing again that, the researcher's role was all along intended as participative - and I must emphasise that nowhere would I wish to claim that the findings obtained in this study would necessarily be replicated in a similar study in which the researcher took a more passive role - it is nonetheless interesting to speculate how the activities might have progressed had it indeed been more passive. It seems likely that pupils would have constructed some form of generalised Logo procedures with no

[67] See § 4.4.2

intervention from the researcher (bar initial explanations of the game context, etc.) and that pupils would have proceeded to carry out auto- and altero-verification, given that these were the rules of the game. It is less certain, however, that much spontaneous discussion would have ensued and that what would have occurred would have led anywhere of great mathematical import.

That said, what I describe in Section 11.7.1 as *auto-concretion* is of particular interest in this context. In that section I speculate that for some children, carefully constructed learning situations can in themselves motivate pupils to go in search of a broader understanding of the mathematical concept contained within them. If this speculation were supported by further evidence, there would be important implications here for the role of the researcher/teacher.

11.5 LIMITATIONS IN SCOPE OF THE RESEARCH

Clearly the research described in this thesis, whilst containing a number of interesting findings, is also limited in the scope of the findings:

i) the children who took part in the study were chosen on the basis of their generally high ability in mathematics as identified by standardised tests administered by the school[68];

ii) the children were generally well-motivated - each chose to take part in the study, all of which took place in the children's own free time;

iii) only a relatively small sample of children, which was nonetheless sufficient for exploring the aims of the study, was used;

iv) all the children were white (due to the locality) and most came from middle class backgrounds.

[68] See Chapter 6 § 6.3.1.

11.6 IMPLICATIONS OF THE RESEARCH

At the beginning of this chapter I suggested that the findings of this thesis explore relationships between generality and particularity, abstract and concrete and formal and informal, looking on these not as dichotomies but as complementary partnerships in the creation of meaning. I consider implications for research under the heading of each of four major findings.

Finding 1

> Within the learning situation defined in the study, children were able to make formalised generalisations of mathematical relationships, often webbed by what I have termed "semi-generalisation".

In Chapter Two I discussed the idea of situated abstraction: "how learners construct mathematical ideas by drawing on the webbing of a particular setting which, in turn, shapes the way the ideas are expressed" (Noss and Hoyles 1996, p 122). The research data very clearly reveals three forms webbing appeared to take - semi-generalisation, generic structuring and naturalised formalism - in the construction of generalisation and justification. These forms of webbing all exploited relationships between generality and particularity: through the particular - and especially through consideration of the *structure* of the particular - pupils were able to achieve powerful insights into generality, and, furthermore, by means of the language of construction, explain, justify and even prove their generalisations.

Returning once more to the S-column analogy and diSessa's specific naive schemata first described in Chapter One, it is apparent that the hierarchic ordering of representations, on which the "rift" viewpoint relies, is at odds with the subjective and uniquely personal ways in which the pupils in this study have been observed concretising mathematical objects and relationships: expressing and re-expressing meaning, and forging connections between their constructed expressions of meaning for generality and justification. Perhaps, if the idea of hierarchy is inappropriate, then the metaphor of specific naive schemata as a kind of bore-hole through strata of meaning is also inappropriate. The power of that visual metaphor lay in the slim column cutting through wide strata - a narrow, situated aspect of some much greater field of knowledge.

Now, it is possible to identify local as opposed to global hierarchies within this study. However, these local hierarchies were fluid and ever changing - consideration of the way in which pupils appeared to shift between the particular and the general demonstrate this in my description of generic structuring and naturalised formalism. For most pupils a generalised Logo procedure of a mathematical relationship began at a "higher level of abstraction" than the natural language description of the same. But, as I have described in this chapter, at other times the Logo procedure through analysis became a more powerful and *more concrete* expression than natural language. This shifting feeling of what at any one time constitutes concreteness resonates strongly with Wilensky's discussion of concrete and abstract (Wilensky 1993).

Finding 2

> The expressive powers of the symbolisation, which was both a goal and a means of expression within the activities, achieved a more functional role by the symbols' association with a history of specific numerical examples.

In Chapter Three I reviewed research (Anderson 1995, p 48) which suggested that approaches to mathematics, which involve pupils in constructing their own generalisations from pattern spotting activities, may be criticised for stopping at the point where a generalisation is produced, which is also the point where proving might begin. Algebra (if it is used) becomes the final goal of the activity. It is interesting to compare this criticism with the use of formal algebraic language in this study. The

Logo procedure expressed in formal algebraic language (a formal expression of generality) was the goal of the construction stage of any task. However, formal language then became a *resource* for use in the exploration and explanation of the structure of mathematical relationships, and for use as a means for justifying generality. Thus formal language served the dual role of medium for construction of generality and medium for the justification of that generality.

The findings thus corroborate research which stresses the importance of an act of construction (e.g. Tall (1989) and Goldenberg and Hazzan (1995)) and which stresses the relationship between the means of expressing generality and the means of expressing justification of that generality (e.g. Bell 1996, p. 177). In fact Boero's finding, that in children's attempts at justification, the "proving process may naturally evolve from it as a "textual development" of the statement [of generality] itself" (Boero 1995, p. 2), is extended in my research as it demonstrates that this evolution is possible at an even earlier age than Boero's findings show, i.e. amongst nine, ten and eleven year olds. Sutherland's findings that children can "make a generalisation and formalise this generalisation in Logo" (Sutherland 1989, p. 341) are also extended by the participation in my study of younger children, but more significantly by the identification of the important relationship between particularity and generality in this Logo programming context: the constantly evolving relationship between the general, in terms of the Logo expressions, and the particular, in terms of the symbols' association with a history of specific numerical examples was a result of the strong link between empirical verification and construction.

This strong link between particularity and generality was a recurring feature of pupils' work. Boero has made a link between precision of generality and verification of validity (Boero 1995, p. 10); my findings (casting in a new light Sutherland's view that the computer frees pupils from the evaluation of an expression - Sutherland 1995) suggest that construction and verification are not so much two distinct processes which are closely linked, but that they are two discernible features of one greater, more complex process of construction, and that consequently for many children in learning situations of this type, effective construction of generality is dependent on the interplay between construction and verification: structural aspects of a situation are focused by the process of verification.

The findings also underline the power of learning situations which are built around an auto-expressive language (Noss and Hoyles 1996), the relationship between particularity and generality adding an extra dimension to the idea of auto-expression. Recognition of the power of auto-expression, or at least a strong connection between generalising and justifying, also questions the value of activities such as those described by Austin (1995, p. 74) and even Maher and Davis (1995, p. 87), which are essentially dead-ended, having no expressive means for justification or proof. Perhaps it is these sorts of activities which Anderson (op cit) is criticising.

Finding 3

> Children constructed situated abstractions for the justification of generality webbed by what I have defined as "generic structuring" and "naturalised formalism".

Generic structuring highlights the central importance to justification of generality of seeing the general in the particular (Mason 1996, p. 70). A number of authors describe the use of generic examples in the justification of generality and expression of proof (e.g. Balacheff 1988, Tall 1979, Steiner 1976, Movshovitz-Hadar 1988). Ideas such as generic proof (Tall op cit) and generic example-assisted proof (Movshovitz-Hadar op cit) involve the use of a particular example, representative of its class (i.e. generic), chosen to demonstrate a proof so that a non-generic proof may be stated as an extension of the generic proof. Generic structuring differs from these in that pupils choose to "dip into" a generic example - often momentarily - in order to give particularity to the generality of an expression they have constructed. They use this experience to help structure their justification of generality. Generic structuring *supports* pupils in their concretising of a generality and is within the *control* of a pupil.

Noss (1997) suggests that proof inhabits a world which lies between natural language and mathematical formalism. My notion of naturalised formalism provides a powerful insight into that world, a world where generality and justification may be expressed in a natural language structured by mathematical formalism. Sutherland gives the example of why for one pupil "it was the experience of symbolising mathematical experiences with a spreadsheet language which helped him to begin to express his ideas in natural language" (Sutherland 1993b, p. 46). Naturalised formalism helps to explain - within a Logo programming context - why the process of symbolisation influences expression.

Furthermore, naturalised formalism satisfies Balacheff's criterion that language must become a tool for logical deduction - for this is the role of naturalised formalism - but by forming a bridge between natural language and mathematical formalism (as manifested in Logo) it obviates Balacheff's need for abstraction in the sense of moving away from one form of language to embrace another. In other words naturalised formalism resonates closely with and extends the view of abstraction (Noss and Hoyles 1996, p. 48) which sees the process as the creation of meaning (the creation of connections) rather than a moving away from one thing to the replacement of it by another.

Finding 4

> The apparent "rift" between empirical and deductive starting points for generalisation, justification and proving activities appeared less clear than the literature suggests.

In Chapter Three I described what is presented in much of the literature as a rift between formal, axiomatic proof teaching (*démonstration*) and teaching methods which focus on processes of proving (*preuve*), beginning with pattern spotting activities and leading through generalisation to justification (Simpson 1995). My research suggests that such a view is unnecessarily blinkered. I do not claim to have constructed a learning situation in which children somehow encounter and integrate both *preuve* and *démonstration* (although why should this not be achieved given more mathematically experienced children and more advanced mathematics within a carefully constructed learning environment?) and come away with a rounded, comprehensive understanding of all that proof entails. What this research does show, however, is that given appropriate tools which afford them adequate webbing, it is possible for pupils to explore generality through a computational medium thereby building a powerful, situated collection of meanings to enrich their developing conceptions of generality, and for them to attempt, again supported by the medium, to justify these generalisations.

Simpson's gloomy picture of "alien" and "natural" routes as two parallel paths, mutually exclusive until, in an apparently non-Euclidean Universe, they converge at the Infinity of Proof, seems particularly inappropriate. Kenneth Ruthven[69] suggests the analogy of two flywheels spinning on a single machine: each wheel is distinct but the two are linked by a band which is forever passing from one to the other. Whilst the idea of representing the proving process by a wheel forever going round in circles but never getting anywhere certainly has its attractions, I prefer to portray the two routes as two winding, constantly crossing paths: the route to proof then becomes a journey involving constant interchange between routes, first one path and then another as most appropriate at any particular stage in the journey. Such a route would not be unique: one pupil might progress quickly along the "alien" path, barely

[69] Oral Reaction (Justifying and Proving in School Mathematics Conference, University of London Institute of Education, December 1995)

straying onto the "natural" path; for another the emphasis might be quite different with much crossing and retracing of routes. Importantly neither path would be a one-way street; back tracking would be a constantly available reality and the crossing points would be frequent enough to allow easy comparison between the experiences gained from travelling each path at similar points of the journey.

11.7 PEDAGOGICAL IMPLICATIONS

Some issues emerged from the study data which have a bearing on classroom practice. Because this area was not one on which the thesis aimed to focus, these issues can form no more than bases for speculation: further testing would be necessary to give them any weight, but they nonetheless remain interesting pointers towards possible further research. The chief area for speculation in this section is what features of the learning situation motivated re-structuring of procedures and what motivated justification of generality.

11.7.1 Promoting Justification

Whereas some tasks dealt with functions which had input-output pairings which were easily calculable or which were known as facts by the pupils, there were others where for the computer verification to be effective, the pupil would have to go through mental calculations of a complexity which would admit errors. Where for some pupils verification lacked relevance due to the difficulty of mental calculation or to unfamiliarity with expected outcome, there were examples of pupils choosing to explain and justify how a procedure worked through analysis of the procedure's structure, rather than relying on empirical testing. In these examples the difficulty level, i.e. the complexity level of the functions, would appear to have pushed pupils towards justifying their procedures in terms of natural (or other) language rather than attempting to verify them with input-output pairs which, because of the pupils' unfamiliarity with the mathematical relationships defined by the functions, in that instance held little or no relevance.

This is a surprising result and has to be interpreted within the context in which it was observed. However, were such a finding to be corroborated by further research, the implications would be highly significant. For there would then appear to be a critical balance of difficulty level which, when exceeded, rather than proving a disincentive to the further, richer construction of meaning, actually *promoted* what is usually felt to be a harder, less natural and less spontaneous form of reasoning. In Chapter Three I questioned the assumption that empirical rather than logical reasoning is necessarily the most natural form in which children attempted to make sense of generalisation and justification[70]; this finding would seem to suggest that for some children in this particular learning situation, empirical reasoning might have been neither their first choice, nor what they perceived as the most appropriate choice.

In Chapter Nine I looked at ways in which pupils explored equivalence of function when this occurred through their pursuance of independent routes to the same common goal. The main value of pupils exploring and discussing equivalent procedures lay in the enrichment of mathematical concepts such activities appeared to promote - the enrichment of the quality of relationship these children were developing with a mathematical object and the diverse collection of connections it wove between and within many expressions of meaning.

Thus children's exploration of the equivalence of procedures provided a valuable window onto their concretion of those procedures, and above all underlined the uniqueness of children's webs of connections: it demonstrated the unpredictability of the manner in which children may make those connections, even in a learning situation which is carefully constructed and tested such as this one. The relationship between even just a pair of children in this study (who shared a common mathematical background and a common school and who received common input in the study tasks, even working together in parts of them) and the mathematical objects with which they were working remain unique to each of the pair.

Comparison of procedures also appeared to *promote* analysis of procedure structure and therefore justification of generality, and this was most marked where the procedures being compared were in fact equivalent or where pupils went on to construct second procedures equivalent to their first (supporting my speculation in Chapter Three that Bell's ideas on using equivalence as a focus for discussion - Bell 1996, p. 176 - might be fertile ground for justification of generality). The feature of the game context of the learning situation suddenly becomes important: undoubtedly the competitive edge (for these children) which was inherent in having to demonstrate that one procedure was superior to another - or more subtly one was as effective as the other, only different - promoted the comparison of procedures and necessarily the analysis of procedure structure and, ipso facto, justification of generality.

Turning again to the question of motivation, I do not wish to think of this in the sense of children wanting to carry out some activity because of a sense of some intrinsic worth to be found in that activity; rather I am interested in the possibility of there being some element inherent in the construction of the learning situation (and therefore children's interaction with that learning situation) which will, if children make the right connections and encounter this element at a propitious juncture in their concretion of the overriding mathematical relationship under investigation, somehow *promote* the forging of further connections. In this study there appeared to be three such elements:

i) the game context: satisfying the requirement to demonstrate to one another that one procedure was superior to another motivated analysis and

[70] See Chapter 3 § 3.3 where I question Fischbein (1982).

justification (or at least demonstrative empirical verification) of the procedures;

ii) the difficulty level of function: analysis and justification were motivated by the inappropriateness of empirical verification, whether due to children's unfamiliarity with empirical results or the complexity of accompanying mental calculation;

iii) the comparison of equivalent procedures: the ability (even likelihood) of pupils pursuing unique paths towards the solution of the mathematical activities motivated the comparison of one another's procedures and the consequent analysis and justification of those procedures.

It must be emphasised that these elements did not feature in each pupil's encounter with the learning situation, in the same way that Logo was not an auto-expressive language for all pupils. However, for those who encountered these elements at the optimum point in their creation of meaning for the mathematics being explored, their relationship with the tools was such that analysis and justification appeared to become the natural by-products of their activity. I suggest that it may be possible to construct learning situations which contain within them elements which motivate their own concretion: *auto-concretion*, perhaps?. That is, there may be situations in which the forging of connections between expressions of meaning motivates further exploration of further expressions.

It is worth pointing out that such a view is not at variance with Wilensky's statement about the determination of concreteness not turning on intensive examinations of the object (Wilensky 1993, p 198), for auto-concretion is entirely dependent on pupils' interaction with the learning situation.

It is also worth stressing again that this is a phenomenon observed in a few cases in this study. However, the notion of auto-concretion is one which would be repaid by further study.

11.7.2 Expectations

The children in this study were aged nine, ten and eleven. It is interesting to note that their attempts at generalisation, justification and even proof, expressed in something approaching mathematical formalism within this carefully designed computational setting, were far in advance of any "usual" expectations for this age. For example, the UK National Curriculum (DES 1991)[71] expects above average eleven year olds (Level 5) to be "generalising from a number of particular examples and carrying out simple tests," and cites the example

> *Having investigated the difference between six two-digit numbers and their reverses (e.g. 82 and 28), make the conjecture that the difference is always in the nine times table, and decide to check three other numbers to test it.*

There is a strong implication here that expectation of mathematical achievement, certainly as far as early experience of the power of mathematical formalism is concerned, is unnecessarily low. Why do we deprive children of this age (and older) of mathematical experiences which, given the right settings, appear to be within their reach?

Following on from this point, it is interesting to look again at the work of Healy and Hoyles (1998 - reviewed in Chapter Three) in the light of this study. They found that in constructing proofs the most popular form of argument is empirical verification and students are unlikely to use deductive reasoning (F28 and F32). In this study, although empirical verification was the prevalent form of justification, it clearly was possible for children (much younger than those in Healy and Hoyles's survey) to engage in deductive reasoning - often through the researcher's interventions - and in some cases to *choose* to reason deductively rather than empirically.

---

[71] See Chapter 1.

11.8 CONCLUDING REMARKS AND POSSIBILITIES FOR FURTHER RESEARCH

This study shows that, given appropriate tools and support, young children are able to generalise mathematical relationships in formal mathematical language and to justify and even prove these generalisations. Further research needs to build on this positive view of what children can achieve, given the existence of carefully designed learning situations. How can we develop new tools and new learning situations to develop and enrich children's experience of generality, justification and proof?

The relationship between the expressive powers of the symbolisation and the symbols' association with a history of specific numerical relationships in this study has highlighted the importance of looking at generality *through* particularity. Research into children's ability to express generality and to justify it within a programming environment, needs to consider how particularity may be used most effectively both as a resource by children and as a tool by teachers.

Semi-generalisation, generic structuring and naturalised formalism were three methods which children used to web their creation of meaning for generality and justification. By unravelling these forms of webbing, it becomes possible to identify the important features of a learning situation - here the ability to access the general through the particular and the mutually supportive relationship between natural and formal language. So identifying and analysing forms of webbing within carefully constructed computational learning situations provides a window onto the essential features of that learning situation. The implication for research is to identify and analyse ways in which children web mathematical relationships so as to develop an ever richer understanding of children's creation of mathematical meaning.

REFERENCES

Alibert, D. (1988)
Towards New Customs in the Classroom. For the Learning of Mathematics 8(2): 31-35.

Alibert, D. and Thomas, M. (1991)
Research on Mathematical Proof. In D. Tall (Ed) Advanced Mathematical Thinking. Kluwer, Dordrecht.

Almeida, D. (1995)
Making Proof Accessible in the Mathematics Classroom. Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

Anderson, J. (1995)
The Legacy of School - Attempts at Justifying and Proving among New Graduates. Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

Austin, K. (1995)
Proof in Mathematics - The First Two Steps. Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

Balacheff, N. (1987)
Processus de preuve et Situations de Validation. Educational Studies in Mathematics 18: 147-176

Balacheff, N. (1988)
Aspects of Proof in Pupils' Practice of School Mathematics. In D. Pimm (Ed) Mathematics, Teachers and Children. London, Hodder and Stoughton.

Bednarz, N., Kieran, C. and Lee, L. (1996)
Approaches to Algebra: Perspectives for Research and Teaching. Kluwer, Dordrecht.

Bell, A. (1976)
A Study of Pupils' Proof Explanations in Mathematical Situations. Educational Studies in Mathematics 7: 23-40

Bell, A. (1993)
Problem Solving Approaches to Algebra. ESRC Seminar Group on Algebraic Processes and the Role of Symbolism, Institute of Education, University of London.

Bell, A. (1996)
Algebraic Thought and the Role of Manipulable Symbolic Language. In Bednarz, N., Kieran, C. and Lee, L. (Eds) *Approaches to Algebra: Perspectives for Research and Teaching.* Kluwer, Dordrecht.

Bell, A and Malone, J. (1993)
Learning the Language of Algebra. Proceedings of the Seventeenth International Conference for the Psychology of Mathematics Education, Japan: Vol 1 (pp 138-145)

Binns, L. and Mason, J. (1993)
Explorations of Vergnaud's Theorem-in-Action in the Context of Algebra. ESRC Seminar Group on Algebraic Processes and the Role of Symbolism, Institute of Education, University of London.

Blair, J., Carpy, J., Clark, L, Duncan, A., Henderson, P., MacCallum, A., McCulloch, D., Mcinnes, D., Mackinlay, J., Preston, C., Tait, W. (1976)
Stage 4 Mathematics: A Development Through Activity. Scottish Primary Maths Group, Heinemann Educational Books, London and Edinburgh.

Boero, P. (1995)
Approaching Theorems: Backwards and Forwards from Historical, Epistemological and Cognitive Analyses to Classroom Experiences. Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

Booth, L. (1984a)
Misconceptions Leading to Error in Elementary Algebra. Journal of Structural Learning 8: 125-138

Booth, L. (1984b)
Grade 8 Students' Understanding of Structural Properties in Mathematics. In Proceedings of the Thirteenth International Conference for the Psychology of Mathematics Education, Paris.

Brousseau, G. (1981)
Problèmes de didactique des décimaux. Recherches en Didactiques des Mathématiques 18 (2): 147-176. Cited in Laborde, C. (1989) *Audacity and Reason: French Research in Mathematics Education.* For the Learning of Mathematics 9 (3): 31-36.

Brousseau, G. (1986)
Fondations et méthodes de la didactique des mathématiques. Recherches en Didactiques des Mathématiques 7 (2): 33-115.

Brousseau, G. (1988)
Etude locale des processus d'acquisition en situations scolaires.

Bruner, J.S. (1966)
Toward a Theory of Instruction. Cambridge, Massachusetts, Harvard University Press

Byatt, S. (1994)
Blindfolding the Train-Spotters. Micromath 10: 25-26

Chalouh, L. and Herscovics, N. (1988)
Teaching Algebraic Expressionsin a Meaningful Way. In A.F. Coxford (Ed) The Ideas of Algebra (1988 Year Book) NCTM, Reston VA, pp 33-42

Chazan, D. (1993)
High School Geometry Students' Justification for their Views of Empirical Evidence and Mathematical Proof. Educational Studies in Mathematics 24: 359-387.

Cobb, P. (1986)
Concrete and be Abstract: a Case Study. Educational Studies in Mathematics 17: 37 - 48.

Cockroft W. H. (1982)
Mathematics Counts. London: HMSO.

Coe, R. and Ruthven, K. (1994)
Proof Practices and Constructs of Advanced Mathematics Students. British Educational Research Journal 20 (1): 41-53.

Collis, K.F. (1974)
Cognitive Development and Mathematics Learning. Paper prepared for the Psychology of Mathematics Workshop, published at the Shell Mathematics Unit Centre for Science Education, Chelsea Colege, University of London.

Davis, R.B. (1975)
Cognitive Processes Involved in Solving Simple Algebraic Equations. Journal of Children's Mathematical Behaviour 1(3): 7 - 35.

Davis, R.B. (1984)
Learning Mathematics: The Cognitive Science Approach to Mathematics Education. London and New York, Routledge.

Davis, R. B. (1986)
Algebra in Elementary Schools. In Proceedings of the Fifth International Conference for the Psychology of Mathematics Education, Birkhauser, Boston.

Davis, R. and Maher, C. (1993)
Children's Development of Methods of Proof. In Proceedings of the Seventeenth International Conference for the Psychology of Mathematics Education, Japan.

Davydov, G. (1990)
Types of Generalisation in Instruction. In Kilpatrick, J. (Ed) *Soviet Studies in Mathematics Education* (J. Teller, trans). Reston, VA: National Council of Teachers of Mathematics.

de Villiers, M. (1995)
Why proof in Dynamic Geometry? Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

DES (1989)
Mathematics in the National Curriculum. London: HMSO.

DES (1991)
Mathematics in the National Curriculum (1991). London: HMSO.

DES (1995)
Mathematics in the National Curriculum. London: HMSO.

diSessa, A.A. (1987)
Phenomenology and the Evolution of Intuition. In C Janvier (Ed) *Problems of Representation in the Learning and Teaching of Mathematics*, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

diSessa, A.A. (1994)
Comments in Ed Dubinsky's Chapter. In Schoenfeld, A. (Ed) *Mathematical Thinking and Problem Solving.* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Dreyfus, T. (1991)
On the Status of Visual Reasoning in Mathematics and Mathematics Education. In Proceedings of the Fifteenth International Conference for the Psychology of Mathematics Education, Assisi: Volume 1: 33-48.

Dreyfus, T. and Eisenburg, T. (1986)
On the Aesthetics of Mathematical Thought. For the Learning of Mathematics 6 (1): 2-10.

Dufour-Janvier, B., Bednarz, N. and Berlanger, M. (1987)
Pedagogical Considerations Concerning the Problem of Representation. In C Janvier (Ed) *Problems of Representation in the Learning and Teaching of Mathematics*, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Falbel, A. (1993)
The Computer as a Convivial Tool. In Harel, I. and Papert, S. (Eds) *Constructionism.* Ablex Publishing Corporation, Norwood, New Jersey.

Feurzeig, W. (1986)
Algebra Slaves and Sgents in a Logo-based Mathematics Curriculum. Instructional Science 14: 229 - 254.

Finlow-Bates, K. (1994)
First Year Mathematics Students' Notions of the Role of Informal Proof and Examples. In Proceedings of the Eighteenth International Conference for the Psychology of Mathematics Education, Lisbon.

Fischbein, E. (1982)
Intuition and Proof. For the Learning of Mathematics 3: 9-24.

Goldenberg, E.P. and Hazzan, O. (1995)
Proof as Explanation in Geometry. Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

Goldin, G.A. (1987a)
Levels of Language in Mathematical Problem Solving. In C Janvier (Ed) *Problems of Representation in the Learning and Teaching of Mathematics*, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Goldin, G.A. (1987b)
Cognitive Representational Systems for Mathematical Problem Solving. In C Janvier (Ed) *Problems of Representation in the Learning and Teaching of Mathematics*, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Gray, E. M. and Tall, D. O. (1994)
Duality, Ambiguity and Flexibility: A Proceptual View of Simple Arithmetic. Journal for Research in Mathematics Education 26 (2): 115 - 141.

Hanna, G. (1989a)
More then Formal Proof. For the Learning of Mathematics 9 (1): 20-23.

Hanna, G. (1989b)
Proofs the Prove and Proofs that Explain. In Proceedings of the Thirteenth International Conference for the Psychology of Mathematics Education, Paris.

Hanna, G. (1991)
Mathematical Proof. In D.Tall (Ed) Advanced Mathematical Thinking. Dordrecht: Kluwer.

Hanna, G. (1995a)
Challenges to the Importance of Proof. For the Learning of Mathematics 15 (3): 42-49.

Hanna, G. (1995b)
Proof as Explanation in Geometry. Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

Harel, G. and Tall, D. (1991)
The General, the Abstract and the Generic in Advanced Mathematics. For the Learning of Mathematics 11(1): 38 - 42.

Hatch, G., Baker, M., Cockett, P., Marriott, G. and Micklewright, W. (1992)
Maths Chest. Thomas Nelson and Sons Ltd, London.

Healy, L. and Hoyles, C. (1998)
Justifying and Proving in School Mathematics: Summary of the Survey Results. Institute of Education, University of London.

Healy, L., Pozzi, S. and Hoyles, C. (1995)
Making Sense of Groups, Computers and Mathematics. Cognition and Instruction 13 (4): 505-523.

Hiebert, J. (1988)
A Theory of Developing Competence with Written Mathematical Symbols. Educational Studies in Mathematics 19: 333-355.

Hoyles, C. (1996)
The Curricular Shaping of Students' Approaches to Proof. For the Learning of Mathematics 17 (1): 7-16.

Hoyles, C., Healy, L. and Pozzi, S. (1994)
Groupwork with Computers: an Overview of Findings. Journal of Computer Assisted Learning 10: 202-215.

Hoyles, C. and Noss, R. (1992)
Learning Mathematics and Logo. MIT Press, Cambridge Massachusetts.

Hoyles, C. and Noss, R. (1993)
Out of the Cul-de Sac? In Proceedings of the Fifteenth International Conference for the Psychology of Mathematics Education (North America), pp 83-90.

Illich, I. (1973)
Tools for Conviviality. London, Calder and Boyars. Cited in Noss, R. and Hoyles, C. (1996) *Windows on Mathematical Meanings*: Learning Cultures and Computers. Dordrecht, Boston and London. Kluwer Academic Publishers.

Janvier, C. (1987a)
Translation Processes in Mathematics Education. In C Janvier (Ed) Problems of Representation in the Learning and Teaching of Mathematics, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Janvier, C. (1987b)
Representation and Understanding: The Notion of Function as an Example. In C Janvier (Ed) Problems of Representation in the Learning and Teaching of Mathematics, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Janvier, C. (1987 c)
Conceptions and Representations: The Circle as an Example. In C Janvier (Ed) Problems of Representation in the Learning and Teaching of Mathematics, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Jones, K. (1995)
Dynamic Geometry Contexts for Proof as Explanation. Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

Kafai, Y. and Resnick, M. (1996)
Constructionism in Practice: Designing, Thinking and Learning in a Digital World, Mahwah, New Jersey, Lawrence Erlbaum Associates.

Kaput, J.J. (1987 a)
Representation Systems in Mathematics. In C Janvier (Ed) *Problems of Representation in the Learning and Teaching of Mathematics*, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Kaput, J.J. (1987 b)
Towards a Theory of Symbol Use in Mathematics. In C Janvier (Ed) *Problems of Representation in the Learning and Teaching of Mathematics*, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Kieran, C. (1983)
A Comparison between Novices' Views of Algebraic Letters and their Use of Symmetric and Asymmetric Equation-Solving Procedures. In Berberon, J. C., Hersovics, N. & Kieran, C. (Eds) Proceedings of the Eleventh International Conference for the Psychology of Mathematics Education. Montral: Université de Montréal. pp 221-227.

Klotz, F.S. (1986)
When is a Program like a Function? Proceedings of the Second International Conference for Logo and Mathematics Education (pp 14 - 24). London, University of London Institute of Education.

LMS (1985)
Tackling the Mathematics Problem. London Mathematical Society with IMA and RSS.

Laborde, C. (1989)
Audacity and Reason: French Research in Mathematics Education. For the Learning of Mathematics 9 (3): 31-36.

Lawler, R.W. (1985)
Computer Experience and Cognitive Development: A Child's Learning in a Computer Culture. Chichester, Ellis Horwood Ltd.

Lawson-Dick, O. (1992)
Aubrey's Brief Lives. Reprint of 1949 edition. Mandarin: London.

Lerman, S., Finlow-Bates, K. and Morgan, C. (1993)
A Survey of Current Concepts of Proof Held by First Year Mathematics Students. In Proceedings of the Seventeenth International Conference for the Psychology of Mathematics Education, Japan.

Leron, U. (1985a)
Heuristic Presentations: the Role of Structuring. For the Learning of Mathematics 5 (3): 7 - 13.

Leron, U. (1985b)
A Direct Approach to Indirect Proofs. Educational Studies in Mathematics 16: 321-325.

Leron, U. and Zazkis, R. (1986)
Functions and Variables: A Case Study of Learning Mathematics through Logo. Proceedings of the Second International Conference for Logo and Mathematics Education (pp 186-220). London, University of London Institute of Education.

Lesh, R., Post, T., Behr, M. (1987)
Representations and Translations among Representations In Mathematics Learning and Problem Solving. In C. Janvier (Ed) *Problems of Representation in the Learning and Teaching of Mathematics*, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Lévi-Strauss, C. (1963 - 76)
Structural Anthropology (2 vols). New York, Basic Books. Cited in Papert, S. (1980) *Mindstorms.* Basic Books, New York.

Maher, C. and Davis, R.B. (1995)
Children's Explorations Leading to Proof. Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

Manin, Y.L. (1977)
A Course in Mathematical Logic. New York: Springer Verlag. Cited in Hanna, G. (1991) *Mathematical Proof*. In D.Tall (Ed) Advanced Mathematical Thinking. Dordrecht: Kluwer.

Mariotti. M.A. (1995)
Introducing Pupils to Proof: A Teaching Experiment. Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

Mason, J.H. (1980)
When is a Symbol Symbolic? For the Learning of Mathematics 1 (2): 8-12.

Mason, J.H. (1987a)
What do Symbols Represent? In C Janvier (Ed) Problems of Representation in the Learning and Teaching of Mathematics, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Mason, J.H. (1987b)
Representing Representing: Notes following the Conference. In C Janvier (Ed) *Problems of Representation in the Learning and Teaching of Mathematics*, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Mason, J.H. (1989)
Mathematical Abstraction as the Result of a Delicate Shift of Attention. For the Learning of Mathematics 9 (2): 2-8.

Mason, J. (1996)
Expressing Generality and Roots of Algebra. In Bednarz, N., Kieran, C. and Lee, L. (Eds) *Approaches to Algebra: Perspectives for Research and Teaching*. Kluwer, Dordrecht.

Minsky, M. (1986)
The Society of the Mind. New York: Simon and Schuster.

Moore, R.C. (1994)
Making the Transition to Formal Proof. Educational Studies in Mathematics 27: 249-266.

Movshovitz-Hadar, N. (1988)
Stimulating Presentation of Theorems followed by Responsive Proofs. For the Learning of Mathematics 8 (2): 12-19, 30.

Nelson, G. (1987)
Using Microcomputer-Assisted Problem Solving to Explore the Concept of Literal Symbols - A Follow-Up Study. In Berberon, J. C., Hersovics, N. & Kieran, C. (Eds) Proceedings of the Eleventh International Conference for the Psychology of Mathematics Education. Montral: Université de Montréal. pp 221-227.

Nemirovsky, R. (1996a)
Mathematical Narratives, Modelling and Algebra. In Bednarz, N., Kieran, C. and Lee, L. (Eds) *Approaches to Algebra: Perspectives for Research and Teaching.* Kluwer, Dordrecht.

Nemirovsky, R (1996b)
A Functional Approach to Algebra: Two Issues that Emerge. In Bednarz, N., Kieran, C. and Lee, L. (Eds) *Approaches to Algebra: Perspectives for Research and Teaching.* Kluwer, Dordrecht.

Nesher, P. and Kilpatrick, J. (1990)
Mathematics and Cognition: A Research Synthesis by the International Group for the Psychology of Mathematics Education. ICMI Study Series. Cambridge: Cambridge University Press.

Noss, R. (1986)
Constructing a Conceptual Framework for Elementary Algebra through Logo Programming. Educational Studies in Mathematics 17: 335-357.

Noss, R. (1997)
New Cultures, New Numeracies. Inaugural Professorial Lecture, University of London Institute of Education.

Noss, R. and Hoyles, C. (1996)
Windows on Mathematical Meanings: Learning Cultures and Computers. Dordrecht, Boston and London. Kluwer Academic Publishers.

Nunes, T., Schliemann, A.D. and Carraher, D.W. (1993)
Street Mathematics and School Mathematics. Cambridge University Press, Cambridge.

Ortiz and MacGregor, S.K. (1991)
Effects of Logo Programming on Understanding Variables. Journal of Educational Computing Research.

Parlett, M. and Hamilton, D. (1972)
Evaluation as Illumination: A New Approach to the Study of Innovatory Programs. Occasional Paper.

Papert, S. (1980)
Mindstorms. Basic Books, New York.

Pirie, S. and Kieren, T. (1989)
A Recursive Theory of Mathematical Understanding. For the Learning of Mathematics 9 (3): 7 - 11.

Pirie, S. and Kieren, T. (1994)
Beyond the Metaphor: Formalising in Mathematical Understanding within Constructivist Environments. For the Learning of Mathematics 14: 39-43.

Porteous, K. (1990)
What Do Children Really Believe? Educational Studies in Mathematics 21: 589-598.

Radford, L. (1996)
Some Reflections on Teaching Algebra through Generalisation. In Bednarz, N., Kieran, C. and Lee, L. (Eds) *Approaches to Algebra: Perspectives for Research and Teaching.* Kluwer, Dordrecht.

Roberts, N., Carter, R., Davis, F. and Feuerzeig, W. (1989)
Power Tools for Algebra Problem Solving. Journal of Mathematical Behaviour 8: 251-265.

Schoenfeld, A.H. (1985)
Mathematical Problem Solving. Orlando, Academic Press.

Sémadéni, Z. (1984)
Action Proofs in Primary Mathematics Teaching and in Teacher Training. For the Learning of Mathematics 4 (1): 32-34.

Sfard, A. (1991)
On the Dual Nature of Mathematical Conceptions: Reflections on Processes and Objects as Two Sides of the same Coin. Educational Studies in Mathematics 22: 1-36.

Sfard, A. and Linchevski, L. (1994)
The Gains and the Pitfalls of Reification - The Case of Algebra. Educational Studies in Mathematics 26: 191 - 228.

Simpson, A. (1995)
Developing a Proving Attitude. Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

Steiner, M. (1976)
Mathematical Explanation. Mimeographed Notes, Columbia University. Cited in Alibert, D. and Thomas, M. (1991) *Research on Mathematical Proof.* In D. Tall (Ed) Advanced Mathematical Thinking. Kluwer, Dordrecht.

Steiner, M. (1978)
Mathematical Explanation. Philosophiocal Studies 34: 135-151.

Sutherland, R. (1987)
What are the Links between Variable in Logo and Variable in Algebra? Recherches en Didactiques des Mathématiques 8: 103-130.

Sutherland, R. (1989)
Providing a Computer Based Framework for Algebraic Thinking. Educational Studies in Mathematics 20: 317-344.

Sutherland, R. (1992)
Thinking algebraically: Pupil models developed in Logo and a spreadsheet environment. London: University of London Institute of Education.

Sutherland, R. (1993a)
Connecting Theory and Practice: Results from the Teaching of Logo. Educational Studies in Mathematics 24: 95-113.

Sutherland, R. (1993b)
Consciousness of the Unknown. For the Learning of Mathematics 13: 43-46.

Sutherland, R. (1994)
The Role of Programming: Towards Experimental Mathematics. In R Bichler, R W Scholz, R Sträßer, B Winkelmann (Eds): *Didactics of Mathematics as a Scientific Discipline.* Dordrecht: Kluwer Academic Publishers: 117-187.

Sutherland, R. (1995)
Algebraic Thinking: The Role of the Computer. In Burton, L. & Jaworski, B. *Technology in Mathematics Teaching - a bridge between teaching and learning.* Charwell. Bratt, Sweden, pp 275-288.

Sutherland, R. and Hoyles, C. (1986)
Logo as a Context for Learning about Variable. In Proceedings of the Tenth International Conference for the Psychology of Mathematics Education (pp 301-306), London: University of London Institute of Education.

Tall, D. (1979)
Cognitive Aspects of Proof with Special Reference to the irrationality of $\sqrt{2}$. Proceedings of the Third International Conference for the Psychology of Mathematics Education, Warwick: 203-205.

Tall, D. (1989)
The Nature of Mathematical Proof. Mathematics Teaching 127: 28-31.

Tall, D. (1995)
Cognitive Development, Representations and Proof. Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

Thurston, W.P. (1995)
On Proof and Progress in Mathematics. For the Learning of Mathematics 15(1): 29-37.

Turkle, S. and Papert, S. (1990)
Epistemological Pluralism: Styles and Voices within the Computer Culture. Signs: Journal of Women in Culture and Society 16(1): 345-377.

van Dormolen, J. (1977)
Learning to Understand what Giving a Proof Really Means. Educational Studies in Mathematics 8: 27-34.

Vergnaud, G. (1981)
Quelques orientations Theoriques et Methodologiques des Recherches Francaises en Didactiques des Mathématiques. In Proceedings of the 5th Annual Meeting of the International Group for Psychology of Mathematics Education, edited by Equipe de Récherche Pedagogique, Grenoble, pp 7 - 17.

Vergnaud, G. (1982)
Cognitive and Developmental Psychology and Research in Mathematics Education: Some Theoretical and Methodological Issues. For the Learning of Mathematics 3(2): 31-41.

Vergnaud, G. (1987)
Conclusion. In C Janvier (Ed) *Problems of Representation in the Learning and Teaching of Mathematics*, Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Vergnaud, G. (1990)
Epistemology and Psychology of Mathematics Education. In Nesher, P. and Kilpatrick, J. (Eds) *Mathematics and Cognition: A Research Synthesis by the International Group for the Psychology of Mathematics Education.* ICMI Study Series. Cambridge: Cambridge University Press.

van Dormolen, J. (1977)
Learning to Understand what Giving a Proof Really Means. Educational Studies in Mathematics 8: 27 - 34.

Vinner, S. (1988)
Visual Considerations in College Calculus - Students and Teachers. Theory of Education III, Antwerp: 109 - 116.

Volmink, J. (1988)
The Role of Proof in Students' Understanding of Geometry. Paper presented at the annual meeting of mathematics teachers, Germany. Cited in Hanna, G. (1989) *Proofs the Prove and Proofs that Explain.* In Proceedings of the Thirteenth International Conference for the Psychology of Mathematics Education, Paris.

von Glasersfeld, E. (1987a)
Learning as a Constructive Activity. In C Janvier (Ed) *Problems of Representation in the Learning and Teaching of Mathematics,* Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

von Glasersfeld, E. (1987b)
Preliminaries to any Theory of Representation. In C Janvier (Ed) *Problems of Representation in the Learning and Teaching of Mathematics,* Hillsdale, New Jersey and London, Lawrence Erlbaum Associates.

Vygotsky, L.S. (1962)
Thought and Language. Cambridge MA, MIT Press.

Vygotsky, L.S. (1978)
Mind in Society: The Development of Higher Psychological Processes. Cambridge MA, Harvard University Press.

Weir, S. (1987)
Cultivating Minds: A Logo Casebook. Harper and Row. In Sacristan Rock, A. I. *Windows on the Infinite: Constructing Meanings in a Logo-Based Microworld.* Unpublished Doctoral Thesis, University of London Institute of Education.

Wilensky, U. (1993)
Abstract Meditations on the Concrete and Concrete Implications for Mathematics Education. In Harel, I. and Papert, S. (Eds) *Constructionism.* MIT. Ablex Publishing Corporation, Norwood, New Jersey.

Wittmann, E.C. and Müller, G. (1990)
When is a Proof a Proof? Bulletin de la Société Mathématique de Belge 1: 15-40. Cited in Hanna, G. (1995b) *Proof as Explanation in Geometry.* Justifying and Proving in School Mathematics Conference, Institute of Education, University of London.

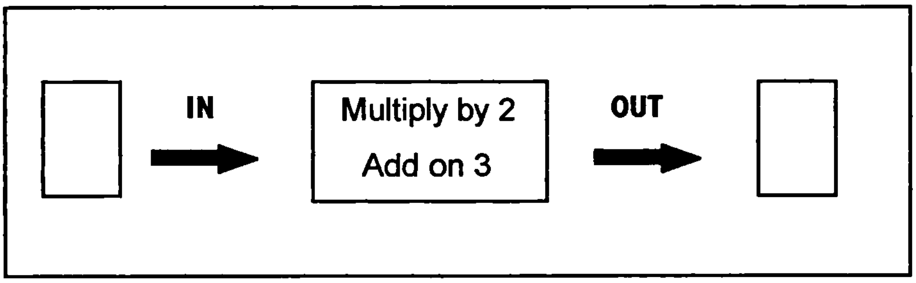
Wood, D., Bruner, J. and Ross, G. (1979)
The Role of Tutoring in Problem Solving. Journal of Child Psychology and Psychiatry 17: 89-100. Cited in Noss, R. and Hoyles, C. (1992) *Learning Mathematics and Logo.* MIT Press, Cambridge Massachusetts.

APPENDIX ONE: PRE-TASK INTERVIEW

Appendix One contains the questions which made up the pre-task interview administered to all pupils prior to taking part in the main study.

Part One: Function machines I

Here is a machine which changes one number into another. If you put a number in from the left, it multiplies it by 2 and adds on 3.



1. Can you say what would happen if you put a 5 into that machine?
2. Try some other numbers of your own.
3. Can you explain in your own words how this machine works?
4. Imagine you had to explain to a child in Year 5/4 how this works.
   What would you say?

Part Two: Function machines II

Now look at this table of values. Just one machine has changed all the numbers on the left into the numbers on the right.

Input	*Output*
0	1
1	4
2	7
3	10
4	13
5	16
6	19
7	22
8	25
9	28
10	31

1. Can you work out what must go in the box in the machine?

If a rule was worked out, the following questions were asked about that rule; otherwise the pupil was shown the rule "multiply by 3 and add on 1" and the questions were asked about this.

2. Are you sure that this rule always works?
3. Why are you sure?
4. Can you explain how this rule works?
5. How would you explain that to a child in Year 5/4?

Part Three: Amaze Your Friends

Task 2.1 was administered orally and the pupil asked for the answer.

Pick a number between 1 and 10. Multiply it by 2. Add on the number you first thought of. Divide by 3. Divide by the number you first thought of. What do you get?

If the right answer was reached (one) the pupil was asked to pick a new number and try again; if the wrong answer was reached, the procedure was gone through out loud with the interviewer interjecting as necessary. The pupil was then shown a written copy of the puzzle.

1. Do you notice anything about this puzzle?
2. Do you think that the answer will always be one?
3. How can you make sure you're right?
4. *Why* do you think the answer is always one?

Part Four: Consecutive Numbers

Look at these three sets of numbers:

8, 9, 10, 11, 12, 13

223, 224, 225, 226

1, 2, 3, 4, 5, 6, 7

1. What can you say about these numbers?

If the word "consecutive" or a description of that property had been reached the following questions were asked:

2. Can you say anything else about consecutive numbers?
3. How would you describe consecutive numbers to a child in Year 5/4?

Part Five: Odd and Even Numbers I

Look at these numbers: 2, 38, 6, 174.

1. What can you say about all these numbers?
2. What (else) do you know about even numbers?
3. How would you describe an even number to a younger child?

Part Six: Odd and Even Numbers II

Now look at these numbers: 5, 91, 17, 653.

1. What can you say about all these numbers?
2. What (else) do you know about odd numbers?
3. How would you describe an odd number to a younger child?

Part Seven: Odd and Even Numbers III

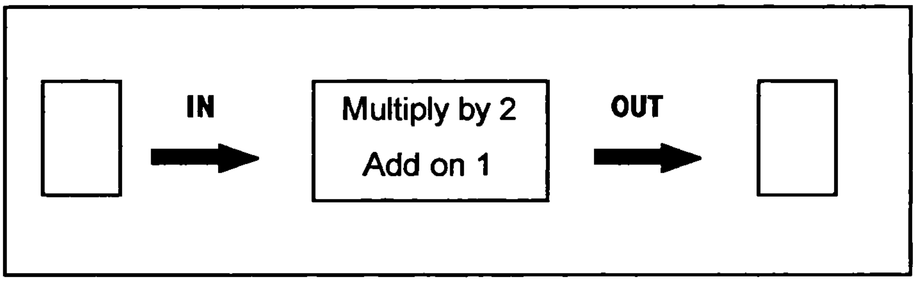
1. How would you change an even number into an odd number?
2. How would you change an odd number into an even number?
3. Do you remember the function machines we talked about in the beginning? Well can you think of a function machine which would change any number you put into it into an even number?
4. Can you think of a function machine which would change any number you put into it into an odd number?

Appendix Two: The Main Study Tasks

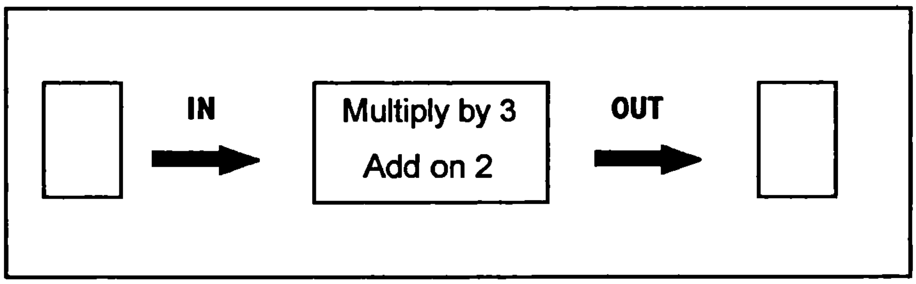
Session One: Function Machines

Pupils were asked to construct Logo procedures to model the function machines in Tasks 1.1 - 13, and to construct function machines which would output the values in the accompanying tables in Tasks 1.4 - 1.6.

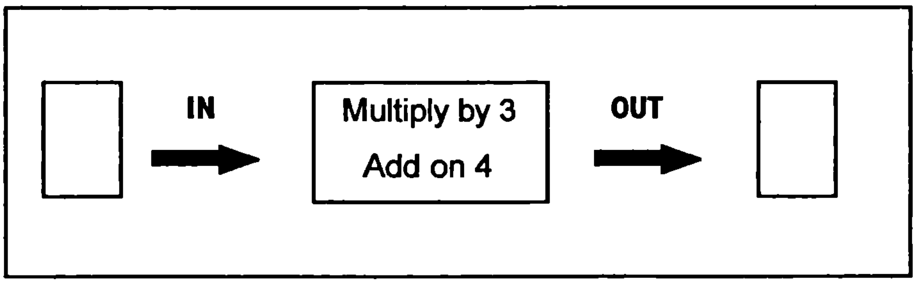
Task 1.1



Task 1.2



Task 1.3



Task 1.4

Input	*Output*
0	3
1	5
2	7
3	9
4	11
5	13
6	15
7	17
8	19
9	21
10	23

Task 1.5

Input	*Output*
0	1
1	5
2	9
3	13
4	17
5	21
6	25
7	29
8	33
9	37
10	41

Task 1.6

Input	*Output*
0	2
1	7
2	12
3	17
4	22
5	27
6	32
7	37
8	42
9	47
10	52

Task 1 Extra

Input	*Output*
0	2
1	3
2	6
3	11
4	18
5	27
6	38
7	51
8	66
9	83
10	102

Session Two: Amaze Your Friends

Pupils were asked to construct Logo procedures to model the following "puzzles" expressed in natural language:

Task 2.1

Amaze Your Friends ....

Pick a number between 1 and 10. Multiply it by 2. Add on the number you first thought of. Divide by 3. Divide by the number you first thought of.

The Answer is .... ONE !!!

Task 2.2

Astound Your Friends ....

Pick a number between 1 and 10. Double it. Double it again. Take away 4. Divide the answer by 4 and add on 1.

The Answer is .... THE NUMBER YOU STARTED WITH !!!

Task 2.3

Thrill Your Friends ....

Pick a number between 1 and 10. Add on 2. Multiply by 4. Take away 8. Divide by the number you first thought of.

The Answer is .... FOUR !!!

Session Three: Consecutive Numbers

Task 3.1

Write a Logo procedure which adds the input to the next number, e.g. if you input **5**, it adds 5 and 6 and outputs **11**.

Task 3.2

Write a Logo procedure which adds three consecutive numbers, e.g. if you input **3**, it outputs the sum of **3**, **4** and **5**.

Task 3.3

Now write a procedure which multiplies the input by the previous number, e.g. if you input **8**, it multiplies **8** by **7** and outputs **56**.

Task 3 Extra

Write a procedure which multiplies the input by the number before it and the number after it.

Session Four: Odds and Evens

Task 4.1

Write a procedure which will always output an **even** number, whatever number you use as an input.

Task 4.2

Write a procedure which will always output an **odd** number, whatever number you use as an input.

Task 4.3

Write a procedure which adds two even numbers together.

Task 4.4

Write a procedure which adds two odd numbers together.

Task 4.5

Write a procedure which adds an even number and an odd number.

APPENDIX THREE: CASE STUDY

This Appendix illustrates the third stage in the gathering and analysis of data. The first stage was to collect the data by means of audio recording and observation. This was followed by transcription of the spoken word and description of action. In the third stage, these transcriptions and descriptions were worked into episodic case studies in order to facilitate the identification of common themes for the four data analysis chapters (Chapters Seven - Ten). Appendix Three contains this episodic treatment of Group F - Jenny and Sophie. Because the purpose of these case studies was to provide data for further analysis, details such as brief descriptions of each task when it is mentioned do not feature here as they do in the data analysis chapters.

Session One: Function Machines

Episode 76: Strategies for Validation - F 1.1

Typical of the pupils' choice of test values in the validation process were those in Task 1.1. For the procedure

```
to jenny :j
op :j * 2 + 1
end
```

Sophie tried

Input	*Output*
567	1135
24	49
20	41
90	181

and Jenny tried

Input	*Output*
12	25
5	11
78	157
50	101

Researcher:	*When you are trying values in there to check that they work, how many values do you think you need to do and what sorts of numbers do you use?*
Jenny:	*I normally do around four and I do high ones and low ones and middle ones.*
Researcher:	*Why do you do high, low and middle?*
Jenny:	*Because then you know that if the high ones work and the low ones work and the middle ones work, quite a few of the rest could be right.*
Researcher:	*How about you, Sophie?*
Sophie:	*About four or five times and round numbers.*
Researcher:	*Like?*
Sophie:	*Fifty and sixty.*
Researcher:	*I notice you've taken mainly high numbers. Is there a reason for that?*
Sophie:	*Because most of the function machine numbers are low numbers and it's not too hard to work out the hard numbers.*
Researcher:	*You mean the numbers that are inside the function machine.*

For these pupils at this early stage empirical verification was a rational, well thought out process. Jenny's idea of "spread" is common, but, Sophie's unusual. Her ability at mental arithmetic was perhaps better than some other pupils'. In Tasks 1.4 - 1.6 both girls continued with their strategies, although Jenny adapted hers to include only values (but still a spread of values) within the domain indicated by the question. Sophie initially checked 10 as an input (within the domain) but all her other values lay outside the domain. This defeated the object of validation in this case, since apart from 10, there was no pair of values to compare with the expected values on paper.

For example, both girls wrote a similar program to Sophie's:

```
to dad :t
op :t * 4 + 1
end
```

which Sophie checked with

Input	*Output*
10	41
60	241
30	121
80	321

and Jenny with

Input	*Output*
1	5
5	21
10	41
8	33

Episode 78: The computer/mental arithmetic interface - F 1.2

The Tasks in section one produced an interesting contrast in the way the two girls worked. Sophie saw the computer as an interactive tool, but one where mental arithmetic had the final say: the computer generated values, and these could be checked mentally:

Researcher: *Do you think it's easier doing these using the computer than working them out in your head?*
Sophie: *I think it's better.*
Researcher: *Why?*
Sophie: *Because you're partly working it out in your head and on the computer, and you're checking it in your head.*

Jenny, on the other hand, saw the computer as a means for checking her own mental calculations:

Researcher: *So do you think the computer helps you, Jenny?*
Jenny: *It does help because sometimes you make mistakes in your head.*
Researcher: *So it checks your calculations.*
Jenny: *Yes.*

Episode 77: The role of the computer in verifying procedures - F 1.3

Jenny repeated her claim that she used the computer to check up on her mistakes.

Researcher:	*So was the computer helpful there?*
Sophie:	*Yes.*
Researcher:	*Can you say again why the computer is helpful?*
Jenny:	*Well, I make a lot of slip ups, and it helps me to check my mistakes, but when I'm writing it on my own it doesn't matter if I accidentally don't do a space, but it does on the computer.*
Researcher:	*You've got to be very exact on the computer.*

On the other hand, of course, the syntax might actually get in the way of the mathematics.

Session Two: Amaze Your Friends

Episode 102: Trial and error informed by procedure construction - F 2.1

Sophie's re-construction of her procedure in the following example (Task 2.1) was informed by successive attempts at validation:

```
to mum :d
op (:d * 2) + :d / 3 / :d
end
```

Input	*Output*
10	20.33334

```
op :d * (2 + :d) / 3 / :d
```

Input	*Output*
10	4

```
op :d * 2 + (:d / 3) / :d
```

Input	*Output*
10	20.33334

op (:d * 2 + :d) / 3 / :d

Input	*Output*
10	1
2	1
4567890	1

Episode 50: The visual appeal of procedure structure and an appreciation of mathematical formalism - F 2.2

In Task 2.1, Sophie and Jenny had both written correct procedures, e.g. Jenny's:

```
to Jenny :h
op (:h * 2 + :h) / 3 / :h
end
```

Researcher:	*Can you say why it always comes out as one?*
Jenny:	*Um, because you...*
Sophie:	*Because the number you get has always got either a two or a three in it and a two and a three, you times it by two and divide it by three.*
Researcher:	*How do you mean it's always got a two or a three?*
Sophie:	*When you... Say the number was ten. When you times it by two which is 20, it's always got a two in it*[72] *and then you add ten again which comes to 30. So the first number's always got a two in it and the second number's always got a three in it*[73].
Researcher:	*Right, I see what you mean.*
Jenny:	*So you're timesing it by three and then you're dividing it by three so it comes out.*
Researcher:	*But it doesn't come out as the same number, it comes out as one.*
Jenny:	*Well then you divide it by its own number so that makes it one.*
Researcher:	*Had you worked that out before or does looking at the Logo procedure help you to work that out?*
Sophie:	*Yes, the Logo procedure.*

[72] i.e. divisible by two.

[73] i.e. divisible by three.

Researcher:	*How?*
Sophie:	*It's more spaced out so I can see it clearer.*
Jenny:	*It reminds you of what the sum is.*

This extract provides two interesting clues to the value of the learning situation:

Firstly, the visual appeal of the procedure: it was spaced in such a way (structured) as to allow Sophie to consider, identify and express *how* it functioned. Sophie was referring to the symbolic procedure structure rather than the natural language structure of the written question she had worked from initially. One of the functions of mathematical symbolism is to present relationships in as clear a way as possible, and it is this function that Sophie had identified here. Although in this example she might not "speak" in Logo (i.e. use terms such as :h times two), from what she said it would seem that her description of what the function did was largely shaped by consideration of the symbolic procedure rather than of the natural language statement. Her descriptions were in a kind of naturalised formalism (or formalised naturalism) which commonly served as an intermediary language between natural and formal. It is "naturalised formalism" rather than "formalised naturalism" because it owed more to the formalised structure of the procedure than to the original natural language expression of the question.

Secondly, Jenny's statement that it provided a fixed, clear, concise record of "what the sum is". This, of course, is another function of mathematical symbolism. Because Jenny had constructed the symbolism herself, it was richly endowed with meaning. For her - for both of them - the translation into symbolic form was a *helpful* act, creating clarity. This might be compared with conceptions of formal symbolism which are characterised by mistrust and misunderstanding. Learning situations such as the one experienced by Jenny and Sophie where an act of construction used the symbolism as an integral part of a "concrete" mathematical experience, appeared to create not just an *acceptance* of symbolic mathematical expression, but a perceived *need* for it.

Episode 51: Generic Examples - F 2.3

Jenny and Sophie attempted to describe how Task 2.2 worked entirely through generic examples:

```
to jenny :j
op (:j * 2 * 2 - 4) / 4 + 1
end
```

Researcher:	*Can you say why that always comes back to the number you started with?*
Jenny:	*Because you double it and double it again... Say you pick ten. You double it - that's 20 - and you double that - that's 40 - and then you take four, so it's 36. Then if you take four away... [Pause]*
Sophie:	*It's like that other one. The numbers that you use, like four, four, one and two you usually get the number - one of those numbers in it.*
Researcher:	*Can you be a bit more precise. Jenny, you were doing quite well then.*
Jenny:	*Well, when you double it and double it again - use five this time - you double it and it's ten and you double it again and it's 20, then you take four off it and if you divide it by four, then it always comes to, um you always divide it by four and it comes equally out and then you just add on one [laughing].*
Researcher:	*Can you add anything to that?*
Jenny:	*Because you go up and then you just go down... and then you go down too far so you add one.*

Session Three: Consecutive Numbers

Episode 52: Change in level of abstraction - F 3.1

In Task 3.2 Jenny moved from a generalised description of how her procedure behaved to a generic example when she was no longer able to continue with the generalised description:

Jenny:	*You could have done :t times three, which would have been the three main numbers and then you add the three, because :t, say is five, then six and there's one and then there's two more, so you add that onto the one. So that's the three.*

Episode 26: Ability to explain structured by procedure construction - F 3.2

In the discussion which took place as a part of the justification process in Task 3.2, the Researcher asked the girls to consider Jenny's procedure for Task 3.1:

```
to cat :o
op :o + :o + 1
end
```

Researcher:	*[after the girls had tried a few values] What do you notice about all the outputs?*
Sophie:	*They're timesing by two and then you just add on one.*
Researcher:	*What do you notice if you look at the numbers.*
Jenny:	*They're all odd numbers.*
Researcher:	*Can you say why they're always odd?*
Jenny:	*Because when you times by two it's always even and when you add on one it's odd.*
Researcher:	*How about in ooo [Task 3,2]?*

```
to ooo :p
op :p + :p + :p + 3
end
```

Jenny:	*I don't know.*
Researcher:	*Try some values again.*
Jenny:	*They're different.*
Researcher:	*Some are even and some are odd. Why?*
Jenny:	*Because you've timesed by three, not two.*
Researcher:	*And what does timesing by three do? [Pause] Does it make them all even or all odd?*
Jenny:	*Because if you timesed it by two and then you just added on the number. So when you timesed it by two, it's an even and so then if you... like five, you times it by two, you get ten. Then you're just adding on another five and five's odd, so it comes out as 15, so it's odd. So if you times it by two and then add on an even number, it will come out as an even.*

Researcher:	*And if you times it by two and add on an odd number?*
Jenny:	*It will come out odd.*

Jenny's explanations were slightly unfocused.

Session Four: Odds and Evens

Episode 53: Simple consideration of equivalence - F 4.1

In Task 4.1 Sophie wrote

```
to mmm :p
op :p * 2
end
```

and Jenny wrote

```
to ilil :y
op :y + :y
end
```

In the discussion they compared their procedures and concluded that they were equivalent:

Researcher:	*Can you explain, Sophie, what you have done and why you have done it like that.*
Sophie:	*The number that you put in is timesed by two, because whether it's odd or even, if you times it by two it will always come out as an even number.*
Researcher:	*Jenny, what have you done?*
Jenny:	*Well, I've done the same as Sophie, really, but I've put add the same number as you put in instead of timesing it by two.*
Researcher:	*Does it matter which you do?*
Sophie:	*No.*
Jenny:	*No.*

Episode 103: Construction strategies: non-interactive - F 4.2

In Task 4.2 Jenny wrote down the following:

```
to yol :h
op :h * 3
end
```

She thought about this procedure, apparently verifying it mentally, and then changed it to the following, without verifying the first procedure on the computer:

```
op :h * 2 + 1
```

This she then validated empirically:

Input	*Output*
6	13
8	17
4	9
90	181
3	7

Jenny was using the computer as an extension of her working memory - as a kind of note pad. Paper and pencil would very likely have served just as well.

Episode 27: Generalisation through construction, verification and discussion - F 4.3

In Task 4.2, Sophie produced the procedure

```
to pop :m
op :m * 2 - 1
end
```

Jenny wrote the procedure

```
to yol :h
op :h * 2 + 1
end
```

After brief verification of the two procedures, the girls predicted what their friend's procedure would be. Both predicted correctly:

Researcher:	*Can you say what Jenny's procedure does?*
Sophie:	*Each time she's timesing by two and adding on one.*

and

Researcher:	*What do you think Sophie has done, Jenny?*
Jenny:	*I think she's timesing by two and taking one.*

Jenny was now able to make a generalisation about functions which output odd numbers:

Researcher:	*Does it matter that they're different?*
Sophie:	*No.*
Jenny:	*No.*
Researcher:	*What do you have to do to make an odd number, then?*
Jenny:	*Well, if you produce an even number, all you have to do is add one or take one.*

Construction, verification and discussion would appear to effect this generalisation.

Episode 28: Explanation structured by procedure construction - F 4.4

In Task 4.3 Jenny explained why her procedure

```
to yuck :e :s
op :e * 2 + :s * 2
end
```

always output an even number:

Jenny:	*Because you're timesing it by two.*
Researcher:	*What are you timesing by two?*
Sophie:	*The two numbers that you put in.*
Researcher:	*And then?*
Jenny:	*Which makes them an even number and then you add both even numbers which makes another even number.*

Episode 29: Formalism as a resource: expression of proof structured by procedure construction - F 4.5

In Task 4.4, Sophie wrote

```
to zich :n :p
op :n * 2 - 1 + :p * 2 - 1
end
```

and Jenny wrote

```
to yach :e :s
op :e * 2 + 1 + :s * 2 + 1
end
```

The Researcher asked them to explain their procedures.

Jenny:	*I've done exactly the same as before, apart from I've put one onto both of the even numbers to make them both odd.*
Researcher:	*So :e * 2 + 1 is odd. What sort of number does it come out as?*
Jenny:	*An even number.*
Researcher:	*Why does it come out as an even number?*
Jenny:	*Because odd plus odd equals even.*
Researcher:	*That's true, but can you explain why that is true by looking at that [the procedure]? [Pause] What does :e * 2 make?*
Jenny:	*Even.*
Researcher:	*Can you go on?*
Jenny:	*And so then you do :s * 2 which makes another even and then you plus them together which makes another even and then you plus two to that and so that makes another even.*

Jenny had proved that the sum of two odd number was even by consideration of her procedure. Sophie was able to make a similar statement:

Sophie:	*It's exactly the same as Jenny's, except I've taken away one instead of adding one.*
Researcher:	*So why does that one always come to an even number, then?*
Sophie:	*Because it's just the same as the ones we did before with one number, only it's got two numbers.*
Researcher:	*Again :n * 2 will be even.*
Sophie:	*So :p * 2 will be even and then you're adding on another two.*

Researcher: *But you're taking away. Does that matter?*
Sophie: *No, because one add one is two, so you're just adding on another even number.*

<u>Episode 30: Formalism as a resource - F 4.6</u>

In Task 4.5 Sophie wrote the following procedure

```
to zuch :n :p
op  :n * 2 + :p * 2 - 1
end
```

and Jenny the following:

```
to yock :e :s
op :e * 2 + 1 + :s * 2
end
```

Researcher: *What sort of number does that always come out to?*
Jenny: *An odd number.*
Researcher: *Can you say why?*
Jenny: *Because you've made your even number and then you've made your odd number and when you do even add even it makes an even and so if you do an even plus an odd - it will be taking one or adding one - it will make it an odd.*

This is an interesting general description, incorporating as it did the two ides about creating an odd number which had emerged over the course of the tasks in section four. The Researcher led the girls towards making a generalisation:

Researcher: *Where are the even bits and where are the odd bits in there? [pointing to the procedure]*
Jenny: *Well the first bit's the odd bit. You times your number by two and then you add the one, and then the second bit's the even bit where you just times it by two.*
Researcher: *But if you said :e * 2 is even, that's an even bit.*
Jenny: *But I did that bit as the odd bit and I did that bit as the even bit.*
Researcher: *Does it matter the order that you do it in?*

Researcher: *Let's have a look at Sophie's.*

Sophie:	*It's the same as Jenny's again, except that I've got my even first and my odd last.*
Researcher:	*So, if :n * 2 is even, :p * 2 is even aswell, isn't it? So altogether, what part of that is even?*
Sophie:	*All of it apart from the taking away of the one.*
Researcher:	*So you could say that :n * 2 plus :p * 2 is a great big even chunk. So it's like saying an even bit take away one, which of course makes it odd.*

The language of construction formed a common language in which Researcher and pupils were able to discuss how the procedures worked.