The Learning of Recursive Algorithms and their Functional Formalization

Sylvia da Rosa
Instituto de Computación - Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
darosa@fing.edu.uy

December 2005

Abstract

This thesis presents an instructional proposal for introducing the concept of recursive algorithms in mathematics or computer science courses of pre-University level studies.

The strategy of the proposal is based on the integration of students' construction of both the concept of recursion and the correspondence between it and its formalization into the learning process. The construction of the concept by the students is investigated within the theory of Jean Piaget, Genetic Epistemology. Specifically, the principle according to which the origin of reasoning by recurrence is inherent to the construction of the series of natural numbers (*La Formation des Raisonemments Recurrentiels*, Jean Piaget, 1963), is extended to reasoning on other structures, given rise to the following premise:

> the source of thinking making possible to design recursive solutions to problems lies in elemental forms of reasoning arising from students' comprehension of the relations between the elements to which his/her actions are applied when attempting to solve instances of problems.

The knowledge about recursive algorithms constructed while solving an instance of a problem is *instrumental*, in the sense that the student is unaware of his/her method and of why it is successful.

The empirical work carried out to investigate the transformation of students' instrumental knowledge into a conceptual one and the construction of the correspondence between the concept and its formalization is presented. The proposed formalisms are mathematics and functional programming, on the understanding that the integration of mathematics and computer science and the assembled work of their educators are essential to surmount numerous obstacles to the learning of concepts shared by both disciplines.

The instructional proposal is derived from results of the analysis of collected data from individual interviews and collective classes conducted with students of pre-University level. Even taking into account the experimental character of those instances, some evidence of the impact of the strategy in the effective learning of the concept and its application to designing recursive algorithms has been identified.

Finally, this work takes into account the necessity of providing research in computer science education with strong connections to the theoretical frameworks of educational-related disciplines. In this way it contributes to establish the field as an academic discipline.

Acknowledgements

Special thanks to the members of the tribunal, Ricardo Cantoral, Peter Henderson, Carmen Sessa, Alberto Pardo and Marita Urquhart.
I am deeply grateful to my advisor Ed Dubinsky and to my supervisor Héctor Cancela.
I would like to thank the mathematics department of Kent State University of Ohio, USA, and PEDECIBA Informática, Instituto de Computación (InCo) of Facultad de Ingeniería and Departamento de Ciencias de la Educación of Facultad de Humanidades of Universidad de la República, Uruguay.
Many thanks to my colleagues of Laboratorio de Ciencia de la Computación for their encouragement and to Gustavo Cirigliano for many years of fruitful discussions about computer science education.
Many thanks to all the students who participated in this research.
The support of Rumec (Research in Undergraduate Mathematics Education Community) and of Katina Bittloch who translated students interviews to English is greatly appreciated.
The encouragement of my family was invaluable during these years. Special thanks to my daughters, Angela and Manuela and to my grandson, Nahuel, who was born a few days after this tesis was finished.

Contents

Chapter 1

Introduction

The main goal of the research described in this thesis is to develop an instructional proposal of recursive algorithms in which the construction of the concept by the students is integrated with the introduction of its formalization. The construction of the concept by the students is investigated within the theory of Jean Piaget, genetic epistemology, and the formalization of the concept is given in mathematics and functional programming.

Although two types of recursive algorithms are considered in this work, namely primitive and course-of-values recursion (see Appendix A), just the terms "recursion" or "recursive algorithms" are used.

The general guideline of the instructional proposal is based on distinguishing between the concept (an individual construction) and its formalization (a social construction). The learning of the concept of recursive algorithms as a school subject consists in constructing both the concept and a correspondence between it and its formalization.

The proposal is founded on empirical results obtained from an application of epistemological principles about how the individuals construct their knowledge to the learning of recursive algorithms as a school subject. It is expected that this work contributes to making research in computer science education an academic discipline by means of establishing a solid connection between theoretical aspects related to education and ideas arising from teaching experience.

Regarding the learning of the concept of recursion there exists a broad consensus in computer science education community about three points:

1. the concept is considered powerful and essential in computer science studies;
2. the students experience the learning of this concept as difficult;
3. the main source of difficulty lies in the lack of day-to-day situations which can help in understanding recursion.

Consequently, several proposals to help students understand the concept have been elaborated and put into practice according to teachers' ideas arising from their own experience in class, in most of the cases involving the use of some programming language or computer tool. Students' performance is mainly evaluated by their responses to questions and tests.

Alternatively, the motivation of this research arises from the observation that in many situations in day-to-day life, people successfully use methods to solve problems or perform tasks such as games, ordering of objects, different kinds of searches,etc, in which an action or a sequence of actions is repeated over a sequence of "smaller situations" until a special situation is reached, which can be easily solved by a straight-forward action. People's descriptions include expressions like "I do the same" and "now I know how to do", referring to the cases where they use the same method and they arrive at the easy-to-solve special situation respectively. The point relating these descriptions to recursive formulations of the method is that when asked to which entity "the same is done" (meaning the same sequence of actions is applied), people refer to the remaining part of an object which is another object of the same type. This correspondence between the method and the structure of elements over which it is applied characterizes recursive formulations with respect to formulations in which other variables are involved, for instance, iterative ones.

On the other hand, recursive algorithms are the basis of many procedures that the students are taught to use -for instance, arithmetic operations, Euclid's algorithm to find the gcd of two integers, Ruffini's algorithm to perform the division of two polynomials, etc- evidencing that courses in mathematics offered in high school or in the first University years provide a suitable context where the careful construction of definitions of recursive algorithms can be introduced. However, even advanced students have difficulties dealing with recursion despite the many courses where they have faced the subject.

These observations lead to formulate questions such as "does there exist any connection between the 'know how to' revealed by people solving problems with methods which can be represented recursively and the formal concept of recursive algorithms? If it does, what is the nature of this connection and what is the role of the instrumental knowledge in the learning process? How is this instrumental knowledge generated and how can it be transformed into conceptual knowledge? How can the algorithms that the students learn to use be taken into account as subjects of study? Will the answer to these questions help in improving the teaching-learning of recursion and how should this be done?"

Answering these questions and designing instructional strategies accordingly requires more than knowledge about the topic and long experience as teacher, it requires research in computer science education. Because of the ample diversity of fields in which the terms computer science and computer science education are used, the next sections are devoted to describing the general context of this work, within which the specific meaning of computer science education is defined.

1.1 Computer science, mathematics and education

In this section, the relation between computer science, mathematics and education is defined. The motivation arises from the fact that computer science is developing in many directions and its influence on other sciences expands each day, new paradigms of programming languages are created, and more and more sophisticated tools and applications in diverse fields of society and industry become available.

This expansion has given rise to several perspectives of answering the question *what is computer science*, for instance, some relate it to designing computers, others to areas such as communication, etc. The adopted perspective determines different educational approaches. In this work, the following definition of the computer science discipline taken from Dijkstra: *On the cruelty of really teaching computer science (1988)* is adopted:

> It is -and will always be- concerned with the interplay between mechanized and human symbol manipulation, usually referred to as "computing" and "programming" respectively.

Adopting this point of view has some scientific consequences, one of them -indicated by the author himself- is that in the map of academic disciplines, computer science has to be located in the direction of formal mathematics and applied logic. Another one -increasingly supported by the software industry- is that a way of addressing problems in software construction consists of integrating the application of formal mathematics reasoning.

Regarding education, the concern of computer science educators aligned with this viewpoint, is mainly centered in the integration of formal mathematics reasoning in computer science education, as a mean of making it a standard practice in software construction [Pag01]. However, neither the problem nor the attempts of solving it are new: since the early eighties and perhaps earlier, the introduction of formal mathematics reasoning in computer science courses has been a concern of computer science teachers. The curricula has included courses in the logic of programming, theory of computing, algebraic specifications, etc, but the problem of students' lack of mathematics background has not diminished, on the contrary, it seems to be increasing [Tuc95]. The diversity of computer science theoretical and application areas contributes to broaden the gap between computer science and formal mathematics, leading to usual omission of computer science from the body of basic disciplines of pre-University education.

The approach developed in this thesis consists of looking at the problem from the opposite direction, that is to say, integrating programming into mathematics education, which would provide not only a thorough grounding in computer science, but would also benefit the mathematical background for *all students.*

Mathematics is a discipline of great weight in the curricula since primary school, but the mathematics currently taught has not caught up with the information age -it is primarily based upon the continuous mathematics of the physical sciences- from which focus, methodology of research and education have evolved. Transforming traditional mathematics to the mathematics of the information age - mainly discrete mathematics- requires the assembled work of both mathematics and computer science educators.

Finally, it is necessary to specify the meaning of the expression *computer science education* (CSE). In their work *Research Agenda for Computer Science Education* [HMG01], C.Holmboe, L.McIver, C.George give the following definition:

> The academic discipline CSE consists in focusing research on the application of principles from educational-related disciplines -*pedagogy, episte-*

mology, curriculum studies and psychology- to the teaching and learning of the scientific discipline computer science as a school subject.

Consequently, two types of workers are distinguished:

- *Computer science educators (teachers)*: who need knowledge about their subject **plus** pedagogical content knowledge, which is the knowledge about the way the subject can come to be understood or misunderstood, what counts as understanding, how the individuals experience the subject.

- *Computer science education researchers*: who are teachers especially dedicated to the field of education with the the aim of providing computer science educators with the knowledge to attain the **plus**.

The authors relate CSE to education in long-established scientific disciplines and observe that in contrast to these, there is a lack of research on educational issues, such as pedagogy, psychology, epistemology. They review existing computer science education literature and categorize some areas such as descriptions of courses, development of tools, computer aided learning, expert/novices differences and empirical studies. The authors point out the necessity of providing CSE with strong connections to the theoretical frameworks of educational-related disciplines, in order to establish research in CSE as an academic discipline setting the foundations for answering the traditional *didactical* questions of *why, what, how* and *for whom.*

They add that the strong connection with educational-related disciplines constitutes the theoretical argumentation of the research as a mean of providing evidence of its effectiveness. The authors find that works making references to epistemological theories unfortunately just mention them in the introduction and rarely discuss results within them, which prevent them from becoming academic contributions to the field.

Taking into account those observations, all the parts of the research described in this thesis (designing and conducting the interviews, analyzing the results, stating the conclusions and further work), are incorporated into a theoretical framework based on the tenets of Piaget's theory.

1.2 Epistemological framework

Jean Piaget's theory *genetic epistemology* is adopted as an epistemological framework. In the literature about computer science education very few works supported by Piaget's genetic epistemology have been found, although his name is sometimes mentioned. The works that refer to constructivism for instance, rarely include references to Piaget's and collaborators works, despite the enormous amount of work in which these authors give detailed explanations about the psychological evolution of mathematical concepts and theories.

Piaget created and directed until his death the International Center of Genetic Epistemology in Geneva, where researchers from many fields such as biology, physics, logic, history worked, producing a large number of publications. In the research described in this thesis, Piaget and collaborators works regarding the central questions of the construction of knowledge are studied, especially those related to the formation of recurrence reasoning, from which fruitful ideas are drawn and put in practice both in experimental activities such as conducting students individual interviews and teaching collective classes and in designing instructional strategies.

According to Piaget's theory knowledge consists of a never-ended mental construction governed by laws and tenets similar to those that govern the biological development of human beings. Recent advances in neuro-science studies contribute a great deal to this viewpoint, casting light on the knowledge about how the brain works.

Although the work of Piaget is in great part devoted to the study of the development of the intelligence of children, he stated that the mechanisms and general laws involved in the construction of the cognitive structures are similar from the early stages of development to those of the more advanced theories [PG80] [PB66]. In particular his studies and research about mathematical thought cover all aspects from the genesis of the concepts of number, order relations, space, etc, to the most refined forms of mathematical theories.

Piaget's theory of genetic epistemology is not about pedagogy which was not the main focus in his work. However, the pedagogical implications of both the experiments and the theoretical explanations cannot be ignored in scientific studies about learning-teaching issues.

In this thesis, ideas regarding the concept of recursion developed in [Pia63] are applied. The most relevant is derived from the problem of determining whether reasoning by recurrence is elaborated once the construction of the complete series of natural numbers is achieved or is involved in the intern course of this construction. These authors show the existence of levels of recurrence from elemental forms inherent to the construction of the series, to more complex ones that proceed on the series once it is constructed. They use the expression *reasoning by recurrence* to mean both calculating on the series and inferring properties about its elements. The first of those meanings relates to the mathematical terms 'recursion' and 'recursive algorithms', while the second one relates to the term 'proofs by induction'. The latter is not considered in this thesis.

1.2.1 Specific meaning of terms and expressions used in this thesis

The following list attempts to avoid eventual confusions about the meaning of terms and expressions used in this thesis. The confusion could arise due to the amply diversity of contexts in which these terms and expressions appear.

- Instrumental knowledge: knowledge constructed by the individual while attempting to solve an instance of a problem or perform a task. This knowledge is generated in the interaction of the subject with the specification of the problem, in which he/she pursues a goal and verifies the results of his/her method, but is unaware of what he/she did and why it works or fails.

- Conceptual knowledge or conceptualization: on one hand, the grasp of consciousness of both subject's actions and modifications on the object generates comprehension about the method that have been employed and about the reasons of its success (or failure). On the other hand, facing new problems requires the transformation of the cognitive means to take into account variations and similarities. This gradual process leads to the construction of conceptual knowledge, consisting of new mental structures as a result of the undergone transformation, in which the new operations, deductions, compositions have been assimilated. Describing in natural language what has been done and reflecting on the reasons of success (or failure) helps in improving the conceptualization.

- Concept: cognitive system (mental structure) resulting from the construction of conceptual knowledge.
- Understanding: the result of the application of conceptual knowledge to solve any problem of adequate level of complexity.
- Concept as school subject: a formalized concept, that is, a concept that has been mapped into a universal system of symbols, for instance, mathematics. The understanding of the formalization of a concept is not the same as the understanding of the concept, because cognitive structures for the mapping itself have to be constructed.
- Recursive methods: the expression "people solve problems or perform tasks using recursive methods" means that a sequence of actions, in which one of them is the sequence itself, is done on a sequence of situations in some sense each time smaller, ending in a special case. The actions can be concrete or mental. Recursive algorithms, recursion and recursive definition of functions are the corresponding formal terms used in mathematics and in programming languages. On the other hand, recursive problems are problems admitting recursive methods of solution (often simply called recursive solutions).

1.3 Overview of the research

In this research the principle according to which the origin of reasoning by recurrence is inherent to the construction of the series of natural numbers is extended to reasoning on other structures. Consequently, the following premise is stated:

> the source of thinking making possible to design recursive solutions to problems lies in elemental forms of reasoning arising from students' comprehension of the relations between the elements to which their actions are applied when attempting to solve instances of problems.

The knowledge about recursive algorithms constructed while solving an instance of a problem is *instrumental*, in the sense that the student is unaware of his/her method and of why it is successful. Transforming it into *conceptual knowledge*, implies that the student has to reflect about what has been done and why it works and about how the knowledge can be used (or not) in other situations as well. The *formalization of the concept* is part of the social knowledge that has to be introduced by the teacher, consisting of the representation of the concept in some universal formalism. The construction of the correspondence between student's conceptualization and the formalism -made by the student herself/himself- is the key to the learning of the concept as a school subject. These stages -instrumental, conceptual and formal- of the knowledge about recursion act not sequentially but in a pro and retro-active manner.

The main part of this research is the empirical work carried out to investigate the passage from the instrumental to conceptual knowledge and the construction of the correspondence to the formalization. From the analysis of the results obtained an instructional strategy is designed. The validity of the proposal arises from the theoretical tenets and the empirical work on which it is founded. Further work is required to support the proposal with new data whose analysis can generate evidence of its effectiveness.

From the mathematical point of view, the premise described above reflects the link between the concept of recursion and the concept of induction and well founded relations stated in [Acz77]. Primitive recursion is used to solve problems on an inductively defined structure, for instance a sequential search of an element in a list, and course-of-values recursion is used to solve problems for which a structure of elements in a well founded relation has to be stated, for instance a binary search. The relation between the elements to which students' method is applied corresponds to the well founded relation between the arguments of the successive applications of the recursive method in the formal definitions.

The universal formalism used in this thesis is mathematics and the programming language in which definitions are implemented is Haskell[1].

The application of the epistemological theory is implemented by means of individual interviews and collective activities conducted with students from the two last high school years and from the first University year, aged 16 to 19. The aim of the interviews is to learn about the methods that students employ to solve problems and to induce them to reflect about. The aim of the collective activities is to introduce the formalization of the knowledge already constructed by the students and to help them to construct the correspondence between both.

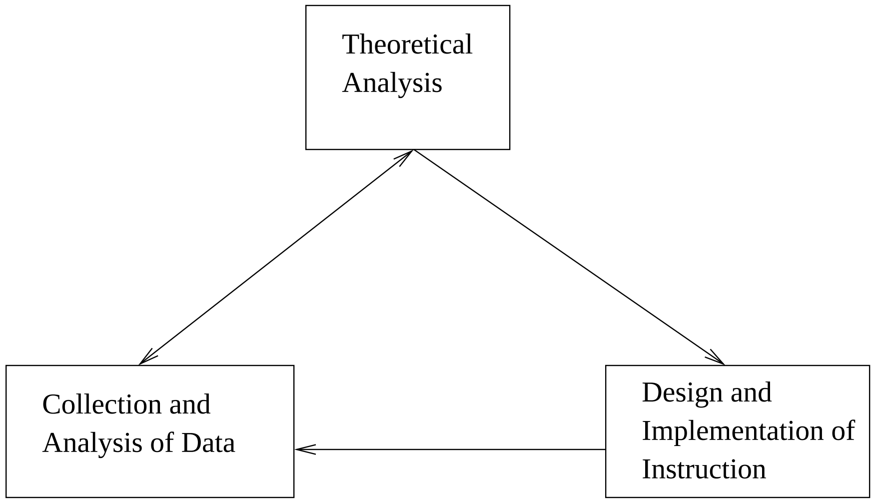
1.3.1 The methodology of the research

The methodology of this research follows the general guidelines of a framework defined by APOS' theory[ABD$^+$00]. APOS theory has been elaborated by mathematics education researchers for the learning of mathematics, based on Piaget's genetic epistemology. The authors use a specific framework for research composed of three components: 1) an initial theoretical analysis of what it means to understand a concept and how that understanding can be constructed by the learner, 2) instructional treatment to make the students do the understanding in (1), and 3) analysis of the gathered data, which can result in a revision of the theoretical analysis and in new iteration of the whole cycle. The framework is graphically described in Figure 1.1.

It is worthwhile saying that the content of the components of the cycle in APOS

[1] www.haskell.org

Figure 1.1: Components of APOS framework



theory is different from the one of this approach. In effect, regarding the theoretical analysis, APOS theory starts from a description of mental constructions that the learner might make in order to understand the concept, called genetic decomposition of the concept. It is built from the researcher's own understanding and modified according to information obtained from students interviews about their understanding from previous instruction.

Alternatively, the starting point of this research is the knowledge that the students develop in solving instances of problems (preferably of real life) and the existing investigations of the psychogenetic evolution of the concept.

In this way, the initial theoretical analysis of this research consists in stating the premise introduced above and preparing a list of questions with the help of a pilot interview conducted with two students that did not participate in the whole project. The result of the theoretical analysis is the content of the first interview of the research, aimed to learn about how students solve some instances of recursive problems, which are the main difficulties they face in conceptualizing their solutions and how they can be helped to surmount these obstacles. The questions and problems do not require previous instruction about the topic.

Regarding the instructional treatment, it is implemented in APOS through a particular pedagogical approach that the authors call ACE teaching cycle, where A stands for activities, C stands for class discussion and E stands for exercises [DF96]. The authors claim that although an action conception is very limited, actions form the crucial beginning of understanding a concept, and therefore their pedagogical approach begins with activities designed to help students construct actions. These activities involve directly working with a programming language, mainly ISETL[2], intended to provide students with experience constructing actions corresponding to

[2] http://isetlw.muc.edu/isetlw/default.asp

selected mathematical concepts. At action level, the programming work consists mostly in making calculations with specific values. The next step is to help the students in reconstructing these actions as processes and in order to do that, they are encouraged to replace code written to perform a specific calculation by a computer function which will perform the calculation for any values. High order functions are called APOS' objects.

In class discussion the students reflect on the computer activities in the lab and the teacher occasionally gives explanations. Exercises to be completed outside of the class are assigned to reinforce the ideas that students have constructed.

The instructional treatment of this research is implemented through a different pedagogical approach, considering that the appropriate formalism from which any implementation can be constructed is mathematics. It consists of introducing formal definitions of students' solutions in a collective class and proposing to the students several exercises to be worked upon after the class. A third individual interview in which the students explain their solutions is conducted aimed at learning about the impact of the collective class and the first interview in students knowledge.

The analysis of the gathered data from these instances generates a new cycle integrating what has been learned about students' knowledge. A collective activity is conducted in which the students are encouraged to discuss their solutions, correct the errors and design the formal solutions to the general problems. All the interviews were audio-taped and transcribed, as in APOS theory and in this case they have also been translated to English. The analysis gave rise to a new cycle and the elaboration of an instructional proposal for introducing recursive algorithms, based on integrating the conceptualization and the formalization of the subject into the process of learning.

1.3.2 Overview of this thesis

The chapters of this thesis are organized as follows:

In Chapter 2, a brief summary of recent studies about the understanding of the concept of recursion is presented. The works are classified according to the underlying epistemological theories of constructivism and mental models.

In Chapter 3 one of the main parts of the empirical study is described, namely, selecting the students and conducting the first interview. The description includes: the material and the criteria used to make the selection of the students, the questions and problems posed in the first interview, selected excerpts of it, analysis of relevant facts and a summary of main results.

Chapter 4 contains the description of on the one hand, a collective class introducing the formalization of the solutions constructed by the students during the first interview and on the other hand, of a new individual interview about the way in which the students solve exercises handed out in the collective class. The description includes: the problems and questions, selected excerpts of the interview, analysis of the responses and a summary of main results.

In Chapter 5, a collective class in which the students correct their own errors from their previous work is described. In this chapter, some aspects of the instructional proposal are present, for instance, the students are encouraged to formalize in mathematics their solutions of the problem. The material and selected excerpts of the interaction developed in the class are included in this chapter.

In Chapter 6 an instructional proposal derived from the research is described in detail using examples from high school mathematics curricula.

In Chapter 7, a summary of both the research and the instructional strategy to introduce the concept of recursive algorithms is presented. Some conclusions and further work are included as well.

The following Appendices are include:

Appendix A: ideas of Piaget's and collaborators' works about the genesis of recursive forms of thought are summarized and mathematical definitions about recursion are included.

Appendix B: some antecedents of this work are briefly described, including references to previous works.

Appendix C: the material used for selecting the students and the complete students interviews are included.

Chapter 2

Learning of recursion: the state of the art

In this chapter a summary of some works about students' understanding of the concept of recursion is presented. Most of them are classified in two groups, according to the underlying dominant learning theory: those based on the concept of mental model and those based on constructivism. Some other works with no reference to any learning theory and a work about learning to program based on phenomenographic research are included as well.

2.1 Works based on mental models of recursion

The term mental model is used by cognitive psychologists such as Johnson-Laird and Norman to define cognitive representations of knowledge. In [GSG03] the authors refer to definitions of Karl Schwamb in "Mental Models: A Survey", who indicates that mental models are subjects' representations of knowledge about particular situations or phenomena. In general it can be said that cognitive psychology uses computational models to account for human cognitive behavior and most of the related research is applied in the field of artificial intelligence.

In the case of learning recursion, several authors refer to mental models to describe the knowledge that the students acquire when introduced to the concept, in most of the cases using some programming language or environment. A mental model is said to be *viable* if it allows the students to accurately and consistently represent the mechanism of recursion and is said to be *non viable* if their representations show misconceptions. On the other hand, a *conceptual model* is designed by the teacher to teach the concept of recursion, while a *mental model* represents the understanding that the learner constructs.

In most of the articles the authors attempt to identify students' difficulties in learning recursion from their solutions or responses to posed problems and give explanations about the underlying misconceptions in terms of mental models. No reference has been found to *how* the understanding represented by mental models is constructed. Some authors point out constructivism as the dominant theory of learning, by which the students actively construct their own knowledge. However, in most of the papers the mental models of students are analyzed once they have been constructed in instruction and interpreted via their responses to questions, exams or tests.

Among the opinions of the researchers there exists a broad consensus about three things: the concept is considered powerful and essential in computer science, it is characterized as a very difficult concept to understand and the main source of difficulty lies in the lack of day-to-day situations which can help in understanding recursion.

In [Kah83] two categories are distinguished: the *copies model*, which is a conceptual model presented in the programming manual of a software environment (SOLO) and is the mental model that experts are supposed to possess, and the *loop model* by which the students understanding is dominated by the notion of iteration, and is the model that novice students in general acquire.

Other mental models have been also identified: in [GSG03], the authors present the analysis of students' responses to exams and tests about tracing Scheme recursive programs execution. Misconceptions about the flow of control in recursive procedures are detected and classified in *active* and *step* mental models. In the same way, misconceptions about the behavior of recursive programs whose syntax has been correctly recognized are identified as not viable models called *syntactic or magic models*.

In [BGM92] the authors study the ways by which novice programmers solve problems using what the authors call *mental methods*, that is, mental models plus solving techniques. Two groups of students, one using a computational system called Petal aimed to assist them in the use of three mental methods (syntactic, analytic and analysis/synthesis) and the other one programming in Lisp, are compared and the first group revealed better performance.

In [CD96] the authors study children's difficulties in learning the concept of recursion in LOGO finding other non-viable mental models and present a general strategy for alleviate misconceptions underlying them.

In [KA86] the authors find that students' great difficulty with programming is flow control and that its origin lies in the inability to develop adequate mental models. The students were taught to program in Simple (a Lisp-like programming language especially designed for studying the acquisition of recursive programming skills) and then they are asked to write iterative or recursive solutions to problems. Their responses and solutions are categorized in terms of mental models such as copies model, loop model, syntactic model, etc. From their study, the authors find that learning about iterative/recursive solutions helps novice students in the learning of recursion later, but that learning recursion first does not help in the learning of iterative procedures.

Taking this results as a start point, in [Wie88] the author attempts to clarify to what extent the learning of iteration helps in understanding recursion, which has been also addressed in other previous work (Anzai and Uesato "Is Recursive Computation Difficult to Learn?", 1982). In order to do so, she poses the following questions: does experience with iteration facilitate learning recursion and vice versa? Can students effectively learn about recursion from analogy to recursive examples? Do students behave differently in learning about recursion understood as a mathematical concept as opposed to a programming technique? The author conducts two experiments: in the first one, students who had no previous knowledge about programming were given some examples of iterative and recursive functions definitions that they have to use to make calculations. She found that in learning mathematical recursion, neither iterative nor recursive similar problems were an aid. The second experiment involved students enrolled in an introductory Pascal programming course and they had to trace and describe Pascal programs. In this case she found that prior work with iterative version of a program to some extent facilitates understanding its recursive version. She concludes that general educational suggestions about the advantages of teaching iteration before recursion can be made.

In [WDB98] the question investigated is: Which of two conceptual models (concrete or abstract) will best help novice programmers with different cognitive learning styles (concrete or abstract) to learn recursion? Conceptual models are defined by teachers and provides accurate, consistent and complete representations of a system that has to be learned. It is *abstract* if it has an abstract base domain, for instance mathematical induction and it is *concrete* if its base domain is concrete, for instance, tracing the process generated by recursive function definitions and stack simulation describing the execution of recursive programs. The students are classified according to their cognitive learning style in abstract learners (who prefer to think when faced with new information) and concrete learning (who prefer to feel and sense when faced with new information). The authors describe an experiment in which the students are first identified as either abstract or concrete learner and then four groups combining the conceptual models and the cognitive learning styles are formed. Recursion was presented to the groups, using the concrete conceptual model known as Russian Dolls and in terms of recursive mathematical definitions and mathematical induction for the case of abstract conceptual models. The performance of students was tested and the conclusions of the authors indicate that concrete conceptual models are better in teaching novice programmers recursion and that teachers should be cautious in adapting or designing such models. On the other hand, they claim that abstract learners perform better that concrete learners in learning recursion, independently of the conceptual model used. They draw no conclusions about the relation between conceptual models and cognitive learning styles.

Some authors focus their research on the impact that visual programming languages or other type of graphical representation are supposed to have on the learning of recursion, finding no evidence supporting the claims [GB96] [Geo00b].

In [Geo00a] the author describes an experiment in which novice students are instructed to use a software visualization tool (EROSI) in tracing the execution of recursive programs, as a mean of helping the students to construct a copies mental model of recursion. After instruction, the students are required to make a test measuring the ability to construct recursive solutions to problems. The author concludes that his empirical observations suggests that this pedagogical approach aids the students in constructing adequate mental models of recursive programs.

2.2 Works explicitly referring to constructivism

The other perspective found in the literature about understanding recursion is that in which the authors refer to constructivism. This term covers a broad range of meanings from ontological issues, related to theories of existence of the real world to epistemological issues, related to the way by which knowledge is obtained, and pedagogical issues, related to how the previous two are applied in teaching.

Constructivist researchers often point out that constructivism arises from Piaget's ideas. Piaget has created an epistemological theory, namely genetic epistemology, which explains the way by which an individual comes to know by constructing mental structures while he or she interacts with the world. The theory also accounts for how this construction is made and the mechanisms, instruments and processes involved in such construction are defined and precisely described in Piaget's and collaborators numerous works, both empirical and theoretical.

What is called constructivism in education refers to pedagogical interpretations of this theory giving rise to several works, especially in science and mathematics education[Dor91], [DL86]. With respect to these, computer science education is in its first stages of development and is in fact striving for achieving the status of an academic discipline [HMG01]. As a consequence, there are works in which Piaget's name is invoked, but no reference to his works appears in their bibliography. An example of the necessity of documenting epistemological affirmations can be found in [BA01]. The author says "these concepts come from the seminal work of Jean Piaget" referring to the fact that "knowledge is acquired recursively" (page 4). Whatever that means, very different explanations of the construction of knowledge can be found in Piaget's works, in particular, the construction of the concept of number obeys *a synthesis* between the concepts of classes and relations.

Computer science education should build on through ways already traveled by science and mathematics education, namely the study of epistemological foundations of the corresponding perspective. In [Dah02] it is stated that ontological issues are not relevant in the same extent.

Other works among the general perspective of constructivism, consider pedagogical approach in which the students are an active part of the learning process and their discourse about phenomena is a source of relevant information for the teacher.

In [Ler88] the author distinguishes between "recursive procedures and the processes they generate" and claims that one of the factors that makes recursion hard is the need to think about the generated process. He claims that when introducing the concept of recursion, the procedure has to be seen as a "description of the product" which is mainly the approach developed by functional programming, in which a program can be derived from a specification of the solution to a problem. He uses the term "procedure" to refer to definition, because he uses definitions in LOGO (procedures).

In [LLP00] and [LL00] the authors work in the context of social constructivism. Their work studies pre-conceptions emerging from the analysis of students' discourse while facing recursive phenomena, as a way of determining the role of such discussions in the process of constructing the concept of recursion. The learners (high school students) are presented some phenomena and asked to classify them according to some criteria of their own choice. The authors say that in this activity, recursion is viewed as an interdisciplinary concept, rooted in every day life and experience, and not merely as a programming tool. They also classify students' pre-conception from the students own words to describe phenomena involving recursion, which can be helpful to teachers when the concept has been introduced. They point out that students widely use analogies and metaphors which stands out in sharp contrast with other authors' arguments about the lack of every day life situations related to recursion as being the source of difficulties of learning this concept. In this research most of the point of views of these authors are shared.

In [BA96] the author presents an idea for introducing the concept of recursion based on using events from real life, (which the author calls dramatizations) and similar programming problems, implemented in Pascal. In this way the students take contact with a traditionally difficult-to-learn concept in an amicable context, which prepares them to a more formal treatment.

In [LZ86] the authors refer to mathematical induction (or complete induction, that is to say, induction over N) and computational recursion, stressing the differences of viewpoints about the dynamic interpretation of the definitions. Their point of view is that recursion and induction are computational and mathematical subjects of study respectively. They use LOGO as implementation language.

2.3 Works without explicit reference to any psychological learning theory

In the following, a number of works studying the learning of recursion without making reference to a psychological learning theory is briefly reviewed.

In [GS99] the authors say that students that master the conventional programming construct of iteration in procedural programming environment find it hard to use recursion. College students were tested over the use of recursion, finding that the students adhere to already learned iterative constructs of the C programming language. The results of their study lead the authors to argue for emphasis on a declarative approach for teaching recursion formulation in a procedural environment. The authors think that the focus should be on the abstract level of problem decomposition irrespective of the implementation. To conduct their experiment, the adopted methodology consists in dividing a group of students in two subgroups and let one of them to learn recursion in the traditional way, that is to say, with emphasis on the implementation level, and the other one with emphasis on the declarative level. Six weeks later they conducted a test, finding that the second group has better performance in formulating recursive procedures in C.

In [HA02] the focus is on the role of base cases in recursion formulation because they have found that identifying the base cases is one of the difficulties in comprehending the concept of recursion. They classify the different types of mistakes of the students about the base cases -ignoring small instances of the problem, avoiding the use of out-of-range values, lack of base cases and redundancy of base cases- and give possible explanations. The research population consists of pre-college students who learned recursion at an introductory level and advanced college students who learned recursive manipulation of data structures and abstract data types. They conclude with some didactic recommendations including emphasizing a declarative approach to introduce the concept, applying diagnostic tests to the students and paying attention to base cases while using concrete models as Russian Dolls.

In [Vel00] the author claims that difficulties in learning recursion come from the learning of the concept in imperative languages, in which several mechanisms other than recursion are involved, such as parameter passing, control stacks, etc. He argues for an approach to teaching recursion where all the mechanisms and involved concepts are identified and isolated from the concept itself and in which recursion is introduced in increasing levels of difficulty from three fields: grammars, functional programming and imperative languages. He has used this approach to teach a course of programming languages for freshmen. This point of view is to some extent shared in this work with the difference that the proposed formalism is mathematics whose courses offer a variety of interesting algorithms to introduce recursion.

In [CM95] and [Bur95], the learning of recursion is investigated in the context of functional programming languages.

In [Boo92] the author takes a point of view which she characterizes as radically different from that of cognitive science while constructivism is not taken into account. Her work follows a phenomenographic tradition of research developed in Sweden, based on exploring and describing the cognitive relations between individuals and the world. In the chapter about recursion, she describes that this topic is taught to students using the programming language SML. Then, the students are asked to solve some problems and answer some questions about their solutions and their works are analyzed and classified into three conceptions of recursion: as a construct in SML, as repetition and as self reference.

2.4 Conclusions

Awareness of computer science education as a new discipline requiring much more than just the practice of teachers is recently emerging. A contribution to clarify the point is the work of [HMG01], taken as a reference to compare this work with the related ones. In it, the authors describe their understanding of what should be the main characteristics of computer science education research in order to transform this field into an autonomous academic discipline. They state that the aim of computer science education research should be

> to describe the different ways in which students come to understand, or not understand, the subject matter. These descriptions, accompanied by knowledge of general pedagogical theory, epistemology and solid subject knowledge will make the foundations for answering the traditional *didactical* questions of *why, what, how* and *for whom.*

The strong connection with educational-related disciplines constitutes the theoretical argumentation of the research as a mean of providing evidence of its effectiveness. The authors find that works in CSE making references to epistemological theories unfortunately just mention them in the introduction and rarely discuss results within them, which prevent them from becoming academic contributions to the field. The situation is completely different in other disciplines, like physics or mathematics, in which the academic field of education is firmly established.

One of the main contributions of this work and one of the most important differences with the related works is the level of importance assigned to the support of an epistemological theory. All the parts of this research (designing and conducting the interviews, analyzing the results, stating the conclusions and further work), are incorporated into a theoretical framework based on the tenets of Piaget's theory.

On the other hand, the authors indicate that works focusing on specific programming phenomena and analizing students' responses to problems and/or questions constitute one of the categories that could be expanded and produce results useful to computer science education. This work clearly pertains to this category, where the specific topic is the concept of recursion/induction.

Regarding the learning of recursion, some specific differences between this work and the cited ones can be indicated.

First of all, in this work students knowledge about recursion is investigated taking as a starting point situations from day-to-day life or from specific problems without any formal introduction to the concept. The students are not even enrolled in the same course and have different study orientations. In contrast, most of the related works, take as a starting point what the students are supposed to have learned in instruction, (the exception are the works of [LLP00] and [BA96].)

Secondly, in this work the concept of recursion is strongly related to inductive structures (see Appendix A), an approach supported by the theory as explained in this thesis.

Finally, in most of the cases, the authors relate the learning of the concept with programming, even those with do not use any computer environment or language. On the contrary, the approach of this research is that the formalization of the concept has to be introduced *mathematically* and implemented in some programming language. Mathematics provides a solid foundation for the concept and the implementation a solid comprehension of the scope of the mathematical definition. The integration of mathematics and programming into a common discipline would benefit the learning for all students. It contributes to provide all students a mathematical background according to the challenges of current social development and constitutes the intrinsic value of computer science as a school subject. On the other hand, improving computer science education has the relevant consequence of improving computer science practitioners' performance.

Chapter 3

The empirical study, Part 1

The experimental part of this work consists of students interviews about solving instances of problems using recursive algorithms. The goal is to gather information about the process of transforming the instrumental knowledge that the students construct while doing that into conceptual knowledge about such algorithms, which can be later formalized. The source of the instrumental knowledge is the interaction between the subject and the structure over which the algorithm is applied[1], the law governing the transformation is the *general law of cognition* [Pia64] and the principal instrument is called *reflective abstraction* by Piaget [PG80]. To start the process the students are faced with problems admitting recursive solutions, encouraged to use recursive methods and induced to reflect about what they did and why it works. In doing so, the students become aware of the relations between the elements and of the influence of these relations on the method to be defined. Students interviews are conducted using a list of questions previously designed, in the manner of Piaget and collaborators, who adapted the clinical method from psychiatry as a tool for doing research in the new field of genetic epistemology. The main tool for obtaining information about cognitive processes is the interview in which the interviewer has a previous hypothesis to be rejected or confirmed. He/she conducts the dialog in a way that prevents deviations or unpredictable situations, taking into account the responses of the interviewee. The latter freely expresses himself/herself and is induced to review his/her claims by the interviewer who observes and intervenes only at the right occasion. These characteristics have been adopted in this research as general guidelines of conducting the interviews.

The students who participate in the interviews were selected from 80 students of four different groups of several orientations (biological, humanistic, scientific and engineering), both from the two last years of high school and the first year of the University of Montevideo, Uruguay. Written material was handed out to the students and according to their responses, 13 students aged between 16 and 19 were selected. The general criteria of selecting consists in including different ways of reasoning and several types of preconceived ideas in the group of participants, as is

[1] The term *structure* is used in two senses: psychogenetically it means *mental structures*, that is to say, the structures of thought that allow the subject to understand or misunderstand concepts, and mathematically it means *structure of elements and their operations*, as usual.

explained in the next section.

3.1 Selecting students

The following is a transcription of part of the written material handed out to 80 students, containing several problem and questions, that the students had to answer. According to their responses, a set of 13 students were selected to participate in the interviews. The complete material (in Spanish) is included in Appendix C.

3.1.1 The material

The goal of this activity consists in selecting students to participate in a project about mathematics teaching. Selected students will participate in three interviews of approximately 45 minutes each. The first and third ones will be individual and the second one will be a collective class, in which some exercises about the treated topics will be handed out. The students have to try to solve the exercises and to explain their attempts in the third interview (complete or partial solutions, correct or incorrect ones). The individual interviews will be recorded for future analysis.
Following, three problems and some questions are presented. Please, pay attention in reading them and answer using your own words and without worrying of previous mathematical knowledge. What is important is the way you reason about the problems and not if your solutions are correct or not. Write your answers on the attached sheet.

3.1.2 Problems and questions

First problem
Two students, John and Peter, have been hired to arrange the window of a store with figures simulating trees. They have to hang red and blue lights from the branches of the trees. The form of the trees is such that from each light (red or blue) zero or two lights (red or blue) have to be hung and the trees are balanced, that is to say, there are as many branches to the left as to the right. All trees have different number of branches.
Here is an example of such a figure (R means red light and B means blue light):

```
       R
      / \
     R   B
    / \ / \
   B  R B  B
```

John puts the lights at random, while Peter follows these rules:
a) the light at the top is red.
b) under each red light he puts a blue to the left and a red to the right.
c) under each blue he puts a red to the left and a blue to the right.
He follows these rules until the tree is completed.

Answer the following question, explaining your reasons:
Peter affirms that his method allows him to count the number of red (or blue) lights of any tree. John says that they both can count the number of red (or blue) lights of any tree without knowledge about the total number of levels. Do you think that he is right? Why? How could be a way to count according to John?
Second problem
Two beachcombers sell what they gather in a day for 2 dollars per kilo. One of them calculates his daily earnings by adding 2 for each kilo he sells, while the other one multiplies by 2 the number of kilos he sells in a day. A mutual friend says to them that it is the same, but the beachcombers refuse to believe that.
1) Which arguments could the common friend employ to convince them?
2) Which arguments could the beachcombers employ to defend their position?

3.1.3 Students selection criteria

The main idea behind the posed problems and questions is to know to which extent the students connect the construction of a structure with its properties and with the possibility of defining methods for calculating values on it.
Problem 1
In the first problem, there are two structures involved: one is the structure of binary trees, and the other is imposed on the trees by the rules followed by the second student (Peter) in placing the lights. The question is aimed to understand how the students connect the structure of binary trees with the computation.
The responses are classified into the following categories:
Category a: to this category belong the responses that attempt to use algebraic formulas, revealing no connection to the structure of trees, attempting to force the problem to be solved in the field of numbers. This way of thinking is pointed out by Piaget as of a lower level in the formation of reasoning by recurrence, for instance when the child instead of taking advantage of the relation between the terms of the series of natural numbers, (being the successor or predecessor or the successor of the successor, etc), to solve some problem, relate the terms to their ordinal position in the series[Pia63]. The great majority of students gave responses in this category, modifying the specification of the problem which explicitly states that the number of branches is unknown.
Examples: "we count the branches and ...", "we count the total amount of lights ...", "if there are n levels ...".
Category b: contains responses referring to "counting one by one" or similar, revealing a weak connection to the structure.
Examples: "he is right because the lights can be counted one by one ...", "it would be complicated because one has to count the lights one by one ...", "it is not possible, unless counting the lights one by one ... perhaps this is the method".
Category c: the answers of type "it is not possible".
Problem 2, questions 1 and 2
In the second problem, it is expected to see whether the students have some idea about the differences between intensional and extensional equality. Distinguishing these notions is an essential factor of the process by which methods playing an in-

strumental role become operations, because an operation is differentiated from its application to arguments. This transformation constitutes the beginning of understanding about function definitions in general and of recursive ones in particular.
Category a: in this category the responses mentioning the differences between the methods in some way are classified.
Examples: "it is the same but applying different methods", "they can say that it is not the same, because one adds and the other multiplies."
Category b: contains responses just referring to the obtained results. Most of students gave responses in this category.
Examples: "it is the same because the obtained results are the same", "let them to look at the results, to see that it is the same."
Category c: responses of the type "they have no arguments."

13 students were selected as is shown in the tables below. Three of them (Ana, Francisco and Cecilia) did not participate in the third interview. That means that 10 students have participated in the whole project, 5 of them from natural and human sciences orientation and 5 from engineering orientation.

Problem 1		
Category a	Category b	Category c
Felipe	Andrés	Laura
Nicolás	Sergio	Ana
Francisco	Iván	
Cecilia		
Gimena		
Ignacio		
Sofía		
Juan Andrés		

Problem 2 (questions 1 and 2)		
Category a	Category b	Category c
Felipe	Laura	Ana
Andrés	Sergio	Francisco
Sofía	Iván	
	Gimena	
	Cecilia	
	Nicolás	
	Juan Andrés	
	Ignacio	

Students' orientations		
Engineering	Natural Sciences	Social and Human Sciences
Felipe	Andrés	Laura
Nicolás	Sergio	Gimena
Iván	Cecilia	Sofía
Ignacio	Francisco	
Juan Andrés	Ana	

3.2 First Interview

The first interview is about two problems. The first one is well known and automatically solved by the students, so the main goal is to learn about what they know about the employed algorithm and about the reasons of its success. For solving the second one a solution has to be derived, so the main goal is to learn about how the students can do that and identify the difficulties of the process, as well. In both cases, the problems admit recursive solutions.

The list of questions has been prepared with the help of two students that did not participate in the rest of the project. These pilot interviews played a crucial role in introducing the so called "second method" in questions Q7 to Q14 of the next section.

3.2.1 First problem

The first problem is an instance of searching for an element in an ordered list. Firstly the students are required to look up a word in a dictionary and to explain in natural language how they did it and why they succeeded. The search of the reasons of success is the motor of the interaction of the thought between students' actions and the modifications that these impose to the object, that is, the result of their actions. In this interaction the conceptualization of the algorithm is constructed.

Secondly, an ordered list of words is used to simulate a dictionary and the relation between the words is described. The students have to give instructions to "a robot" in order to find a word. The simulation part is aimed to reinforce the conceptualization because the more automatized the action, the more difficult is to become aware of the component actions involved in it.

It is expected that the students solve the problem using the algorithm of binary search, without being aware of what that means.
It is expected that the interview helps the students in grasping the algorithm with respect to:

- the actions which composed it: choosing a word, comparing words and a new instance of the search itself;
- the reasons of success: each new instance of the search is applied on "smaller" parts of the dictionary until a special case is achieved;
- the structure: the "smaller dictionaries" are constructed holding the property of containing the searched word and ending in the special case.

This algorithm is one of the most important methods of searching and it is taught in all courses of programming, traditionally introducing some formalism to implement it, in the form of explanations given by the teacher to the students. In contrast, this approach considers that the source of knowledge is what the students do when they attempt to solve an instance of the problem. The case of searching a word in a

dictionary constitutes a suitable instance of the general problem and an appropriate example of course of values recursion[2] as well.
Moreover, this problem has been chosen because no numerical domain is involved and the role of previous instruction is minimal which diminishes the influence of preconceived ideas.

In the following, sublists of questions are presented, preceded by the motivations behind them. After each sublist of questions, notable facts and illustrative excerpts follow, in which comments from the interviewer are interspersed in italics. Notable words of students responses are indicated in italics as well. The enumerated questions were used as a basis for the interviews. In some cases, other questions were added at the moment or the same question was formulated in another way, depending of the development of each interview, and are indicated just by Q. R means response (of the students).

Questions 1 to 4

The first four questions are aimed, on one hand, to establish a fluid contact between the student, and the interviewer and on the other hand, to induce the student to reflect about the dictionary characterized by the order relation between words, which allows to construct the structure of successive "smaller" dictionaries.
The Spanish word for "cat", that is, "gato", is maintained as a remainder of the original language of the interviews.

Q1:	What is a dictionary?
Q2:	Knowing that a word, for example *gato*, is in the dictionary and also in this novel, where do you think that will it be simpler (easily and quickly) to find it?
Q3:	Why?
Q4:	What makes the difference then between a dictionary and any other book?

Facts

1) Only one student characterizes the dictionary by its alphabetic order relation (Gimena). The other students refer to what one can do with the dictionary or what it is for, in other words, their responses are adequate to questions like "what can you do with the dictionary?" or "what is the dictionary for?". Because of the obvious character of this property, it is more difficult to conceptualize.
2) Faced to questions Q3 and Q4 about reasons, (why is it easier to find a word in a dictionary than in a novel, inferring the difference between both structures), all students recognize the alphabetical order, which reveals that they become aware of the relevant property of the object.

Excerpts of Q1 to Q4:

Andrés:

Q1:
R: A dictionary is a book in which we look for the meaning of words.
Q2:

[2] This type of recursion is described in Appendix A.

R: In the dictionary.
Q3:
R: Because it is in alphabetical order.
Q4:
R: That has a certain order to be ... to find what you are looking for.
Felipe:
Q1:
R1: It is a place where you can find all the words of a language.
Q2:
R: In the dictionary.
Q3:
R: Because they are in alphabetical order.
Q4:
R: That they are organized in a special way.
Gimena:
Q1:
R: It is a book in which the words of a language are in alphabetical order.
Q2:
R: In the dictionary.
Q3:
R: Due to the alphabetical order.
Ignacio:
Q1:
R: A book that has the meaning of the words we use.
Q2:
R: In the dictionary.
Q3:
R: Because it's done in a way that has ... it is in alphabetical order.
Q4:
R: An order.
Iván:
Q1:
R: A dictionary is a book which has the meanings of all the words.
Q2:
R: In the dictionary.
Q3:
R: Because it has an order.
Q4:
R: That they have an order (alphabetical)
Juan Andrés:
Q1:
R: It is a book where we can find the meanings of the words of a language.
Q2:
R: In the dictionary.
Q3:
R: Because the words are in alphabetical order.

Q4:
R: The dictionary is something to consult.
Q3:
R: Because they are in order.
Q4:
R: Due to its practicality.
Q: With respect to the words.
R: Because they are in order.
Laura:
Q1:
R: It is a book where all the answers are, that is ... what you need.
Q: What is it that there is in the dictionary?
R: All the words with which we speak.
Q2:
R: In the dictionary.
Q3:
R: Because it is ordered alphabetically.
Nicolás:
Q1:
R: A book where I can find words which I do not know their meanings.
Q2:
R: In the dictionary.
Q3:
R: Because there is an alphabetical order in the dictionary.
Q4:
R: That I can "create" a word, I can go filtering it.
Sergio:
Q1:
R: A place where answers are found to questions. Words, that is, answers of meaning to questions.
Q2:
R: In the dictionary.
Q3:
R: Because it has an order.
Q4:
R: The order.
Sofia:
Q1:
R: A book that has the definitions of the words of the language.
Q2:
R: In the dictionary.
Q3:
R: Because you know all of them are there and what's more, you have an order to find them.
Q4:
R: The organization, the order.

Questions 5 and 6

The goal of the following questions is to apply Piaget's ideas about the conceptualization of automatically done actions. Interrupting the action and introducing the need for thinking forces the student to direct his/her thought from the result of the action towards the intern mechanism of the coordination of the actions that has given rise to that result. This movement "from the periphery to the center", that Piaget calls *The general law of cognition* [Pia64], allows the subject to construct better representations, on the one hand, of the objects and on the other hand, of his/her own actions which are transformed into operations (methods).

Q5:	Look up the word *gato* in the dictionary.
Q6:	Describe, step by step, how you achieved it. (Eventually, ask them to do it again).

The answers to question Q6 reveal that the students have a very week conceptualization about their method of searching a word, except for Iván. The students apply binary search as expected, but they are not aware of the different actions they do to achieve the result. The analysis of this part is included together with the analysis of Q19, in order to compare the responses, since Q19 is essentially the same question posed at the end of the interview.

Questions 7 to 14

The goal of the following questions is to apply Piaget's ideas about the role of the process of "searching the reasons of success" in the conceptualization. The constant motor impulsing the subject to complete or to replace the observables of facts by deductive or operative inferences is the search of *reasons* for the obtained result [Pia78]. That means that "the search of reasons" impulses students' thought toward the interaction between the coordination of his/her actions and the object, reaching an equilibrium generating the concept of the algorithm.

As mentioned at the beginning of this section, two pilot interviews were conducted while preparing the questions. In them, it was found that inferring the reason by which the search works is hard for the students. In order to help them to surmount this difficulty, they were asked to use another method [3] which consists in asking the students to open the dictionary and to look if the searched word is there. If it is not, (which will happens in essentially), they are asked to close the dictionary and to begin to search for the word again, that is, in the whole dictionary. This strategy was effective: after using the second method, both students of the pilot interviews immediately found the reasons of the success of the search. In the interviews described here, however, two students (Cecilia and Andrés below) were interviewed without using the second method, in order to compare their responses with those of students using the second method and strengthen the evidence of the effectiveness of the strategy.

[3] in the following called "the second method" or "2nd. method"

Q7:	Knowing that a word is in the dictionary, is it always possible to find it?
Q8:	Why?
	(According to their responses, the second method is introduced).
Q9:	Start looking it up again.
	(When the student opens up the dictionary on a letter, he/she will go forwards or backwards to continue the search, there we stop him/her and ask:)
Q10:	Why are you still searching there?
Q11:	What happens with this part of the dictionary?
Q12:	So you do not continue searching through the whole dictionary?
Q13:	What happens with the dictionary then, when you are still searching?
Q14:	Then, why are you sure that you will find it?

Facts
The following facts clearly reveal advances in the conceptualization of the students. The facts are illustrated in the selected excerpts below.
1) All students feel confident of succeeding in finding the word, if it is in the dictionary.
2) None of them knows why this happens.
3) After the questions, all students attribute the success to the method and are aware of the importance of the well founded order relation between the parts of the dictionary (the structure).
4) The use of the second method is crucial for understanding the structure.
5) In some cases, the method is decomposed in its constituents.

Excerpts from students responses to questions Q7 and Q8, before introducing the second method.
Two students were asked without requiring the use of the second method: Cecilia and Andrés. Cecilia did not participate in the third interview and is therefore not considered in the total analysis. The responses of both students are similar and reveal how hard the conceptualization of this algorithm is, which, at the same time, is so successfully applied.
They make the same mistake thinking that the remaining part of the dictionary has more words, which reveals a weak mental representation of the relation between the elements to which the method is applied. Following are excerpts from their interviews showing that point. This part of their interviews is much longer and is included in Appendix C.
Cecilia:
Q8:
R: Because I look for it the same way as I looked for *gato*.
Q: Tell me slowly again.
R: I look for the first letter, after finding the first letter, I start by the second, because they are also going to be in alphabetical order.
Q: We look for the word *gato*. Here, we are searching here, in this piece as you said and, what is it like regarding the whole dictionary?
R: A part.
Q: And one part, what is it like with respect to the whole?
R: Different.

Q: In what sense?
R: That it does not involve all the words.
Q: Has that part more or fewer words?
R: More.
Difficulty in constructing a mental representation of the relation between the parts of the dictionary. After some seconds of silence, she continues with the next response.
R: Fewer ...
Q: When you found the word, how many do you have?
R: One.
Q: Why, then, will you find a word?
R: Because we are shortening the number of words.

Andrés:
Q8:
R: Because it will always be in that order. Because looking for in this way and due to the order the dictionary has, I will find it.
Q: And what is that part with respect to the whole dictionary?
R: The rest.
Q: And how is this rest according to the whole dictionary?
R: More useful for me.
Q: Yes, and with respect to the number of words that has?
R: Greater.
Difficulty in constructing a mental representation of the relation between the parts of the dictionary. Some seconds of silence follow before the next question is formulated.
Q: This part is with respect to the whole dictionary ... ?
R: Smaller than the complete dictionary.
Q: Then what happens with the dictionary when you are still searching?
R: Some parts are being discarded.
Q: And it becomes ...?
R: Smaller.
Q: When you find the word, what happened to the dictionary?
R: It is useless, it only helps me for that word only.
Q: So, what was the dictionary transformed into?
R: Into only one word.
Q: That is why you find it, because you search in sections that become smaller and smaller within the dictionary.
R: Oh, of course.

Excerpts from students responses to questions Q7 and Q8, after introducing the second method.
All the students that were asked to use the second method, quickly improve the conceptualization of the series of "smaller" arguments (the structure). Felipe, Nicolás, Sergio are examples of students whose attention is centered on their actions and have to be directed to the transformations imposed by them to the object, while Ignacio and Iván are examples of attention focused on the object and have to be directed to their own actions. The conceptualization of the structure is attained through the equilibrium between the coordination of subject's actions and the transformations

caused on the object. Some of the students improve the conceptualization of the method decomposing it in its components *choose, compare* and *do the same.*

Felipe:
Q7:
R: Yes.
Q8:
R: Using this method you can find it.
Q: What does this method have that is so special for you to find it?
R: ...
Although he refers to the method as causing the success, he does not know why.
Q: (We use the 2nd method.)
R: We spend three years.
Q: Then, why with the other method you find it more quickly?
R: Because I go on discarding pages, when I go forwards, and you don't need all these ones.
Observe that he is still thinking of his own action. However, to advance in the conceptualization his thought has to move to the transformation produced in the object as a result of his action. That is what next question induces.
Q: Then, in each search, what happens with the dictionary?
R: It becomes smaller.

Nicolás:
Q7:
R: Yes.
Q8:
R: Because I always get the same method to look it up, I always apply the same method by which ... everybody uses it for the dictionary.
Q: And that method you apply and that everybody uses, why does it guarantee you that you will find it?
R: It never fails.
Q: Why?
R: ... because ... because it holds all letters in the dictionary, I mean I always ... never will get lost ... as long as I search one ... then I will always follow the same order until I find the word, by each letter I go on searching.
Although he refers to the method as causing the success, he does not know why.
Q: I propose another method. (we do it with 2nd method).
R: Yes, of course, we never finish.
Q: Which is then the difference between this method and the other?
R: I based myself on the principles that is, I open the dictionary and I based myself on what I find at first sight, (*choose*) then I start checking if I have to go forwards or backwards (*compare*).
Decomposition of the method.
Q: Suppose you go forwards, what happens to the rest?
R: I discard it.
Observe that he is still thinking of his own action. However, to advance in the conceptualization his thought has to move to the transformation produced in the object

as a result of his action. That is what next question induces.
Q: Then, what happens to the dictionary while you are searching?
R: They are being eliminated, it becomes reduced (*"smaller" arguments*).

Sergio:
Q7:
R: Yes.
Q8:
R: Because ... knowing that it is ... What a good question ... If I know that it is there and *I'm going to look for it* ... it is supposed that it is over.
Q: Yes, but why do you find it?
R: Because you look it up.
Although he refers to the method as causing the success, he does not know why.
Q: Another method (we perform the 2nd method).
R: Oh! Because I come back to follow the same process (*do the same*). Always, I mean, first I look for a section (*choose*) and afterwards ... a smaller one (*"smaller" arguments*).
Decomposition of the method.
Q: Why do you find a word, then?
R: I take a sector of the book and I start reducing it until I find it.
Q: So, you told me ?
R: I look for letter by letter.
Observe that he is still thinking of his own action. However, to advance in the conceptualization his thought has to move to the transformation produced in the object as a result of his action. That is what next question induces.
Q: And what happens to the dictionary while you are searching?
R: It becomes smaller.

Ignacio:
Q7:
R: Knowing that it is in the dictionary, yes.
Q8:
R: Due to the order it has.
He attributes the success to the dictionary.
Q9: Good, let's use now another method to find the word *gato* with the same dictionary having the same order. We close the dictionary and I ask you to look up the word *gato*.
R: (He does it).
Q: Is the word *gato* there? (Where he opened).
R: No.
Q: Close it and look up *gato* again.
R: (He does it).
Q: Is it there?
R: No. (It is repeated some times).
Q: With this method the dictionary keeps its order and however, do you think that in this way we'll find the word *gato* easily?
R: No.

Q: Instead, with your method?
R: I find it fast.
Q: What is the difference between the two methods?
R: One is safe and the other isn't.
Q: Why?
R: Because there is an order.
Q: No, in the 2nd method the order remains in the dictionary.
R: But I didn't follow it.
Observe that now he refers to the method and not to the order relation, revealing an advance in conceptualization.
Q: What does "following the order" mean?
R: I discard this part. I know that "g" is not here and I discard it.
Q: What happens with the dictionary?
R: It becomes smaller.

Iván:
Q7:
R: Yes.
Q8:
R: Because if it is in the dictionary I will find it.
Q: Why?
R: Because ... because it has this order ...
He attributes the success to the dictionary.
Q: Good, the dictionary has an order. Now we are going to use another way with the dictionary that has that order (we do it with the 2nd method).
Q: What is the difference between the two methods?
R: Your method is more difficult to find a word, more improbable.
Q: Why?
R: With the 1st one you start discarding parts and I will find it more quickly, not with the other.
Observe that now he refers to the method and not to the order relation, revealing an advance in conceptualization.
Q: Then, what happens with the dictionary?
R: It becomes smaller.
Q: And when you found the word?
R: The search is over.
Q: How many words does the dictionary have?
R: One.
Sofia:
Q7:
R: Yes.
Q8:
R: Because I trust myself, because yes, (she laughs).
Q: You'll find it rather quickly or you will take the whole day.
R: Quick.
Q: Why?

R: Because I have practice.
Q: Now we are going to use another method (2nd method). Open the dictionary (she opens it).
Q: Is *gato* there?
R: No. (We do it again many times).
Q: What do you think of this method with respect to the other?
R: It's much more difficult.
Q: And what is the difference between them?
R: That in the 1st one I start marking between which and which and I can shorten the limits (*"smaller" arguments*) and in the other it is at random.
Q: Why are you sure that with your method you will find it and quickly?
R: I have fewer bounds, fewer limits.
Q: What happens with the group of words in which you search?
R: They start shortening.

Questions 15 and 16
These questions are aimed at inducing the students to identify one of the important methods involved in the search (comparing). The order relation is formalized, preparing for the next activity.

Q15:	Do you agree that you compare the word you look up with the ones in the dictionary? The alphabetical order can be represented saying that one word is "smaller or greater" than another. For example, *gato* $<$ *perro*, *gato* $>$ *dos*, *gato* $<$ *gente*, *gato* $=$ *gato*, etc.
Q16:	Then, which are the possible results of each comparison?

The representation of the alphabetic order with $<$, $>$ and $=$, is always explained in detail, and the possible results expressed as

- Choosen word $<$ searched word.
- Choosen word $>$ searched word.
- Choosen word $=$ searched word.

All students agree.

3.2.2 Second part: Simulating the search on a dictionary

The goal of this part of the interview is to help the students to surmount the obstacle of automatization, by means of repeating the activity and making them to reflect about what they did and why.
The material used is a series of 20 bits of paper with a word written in one side and W_i on the other side, where $1 \leq i \leq 20$. The list of 20 bits of paper is arranged on the table with the side "W_i" up and simulates a dictionary, that is to say, the hidden words are alphabetically ordered. The interviewer acts as a robot that has to find a word in the list following instructions from the student, where the possible ones are (that is to say, what the robot knows how to do):

- Form a list from W_i to $W_j, 1 \leq i, j \leq 20$.
- Choose a word.
- Define an order relation between words.

We give to the student a sheet of paper in which the following is written for 4 or 5 searches:

Search 1:
Result:
Search 2:
Result:
⋮

The student has to write the instructions to the robot following "Search ..." and the obtained result in "Result ..." and then determine the instructions that have to be given for the robot to the next search. The task is done searching for the word *luna*[4] in the list. The activity is successfully performed by all students.
The following questions are posed after the activity is finished.

Questions 17, 18 and 18'
The goal of the questions is make the student experience the methods composing the search -choose a word, compare two words, *do the same*- and the decision implied by the result. The students are induced to infer what does *do the same* mean: on the one hand, making them aware of the fact that one of the components of the method is the method itself, on the other hand, that it is applied to a *a new list* determined according to the result of the comparison. Observe that the worksheet induces the idea that the search is a sequence of searches.

Q17:	How does the result of a search influence the one after?
Q18:	Do you agree that in each search there are things that are done the same way and things that vary?
Q'18:	Could you determine them?

Facts
Four remarkable facts are presented each of which is followed by illustrative excerpts.
1) In contrast to a common belief, the distinction between a method, its argument and its application is in no way trivial and is the source of many misconceptions.
Excerpts
Iván:
Q17:

[4] "luna" is the Spanish word for "moon"

R: The number of words decreases.
Q18:
R: The formation of the list starts shortening, "choose word" is always present and define the relation between also, and the result varies.
Q: And with respect to what the robot does?
R: *It varies, yes, because it has fewer words to choose.*
Observe that he has difficulty in separating the action from its application.
Q: Let's look at what you wrote and let's try to determine what the robot does. Does the robot do different things each time?
R: No.
Laura:
Q18:
R: Yes.
Q'18:
R: *The result varies, the search.*
Observe that she has difficulty in separating the action from its application.
Q: What do you mean with the search?
R: That you see, the search is shorter.
Q: Where do you see that? What varies what?
R: *The list varies.*
She understands that what varies is not the search, but the list.
Q: And what things remain the same?
R: Choosing the word and defining the relation.
Q: Some other thing ? What I do with the list?
R: It does not vary.
Q: Do I do different things?
R: No, the same is always done.

2) The students refer to the invariant property: the lists diminish until the word is found (which always happen) and the searched word belongs to all the lists. This type of properties are discussed in the collective class.
Excerpts
Andrés:
Q: And what remains the same?
R: *That words from a group are always within the greatest group.*
Felipe:
Q: What do we do with the list?
R: *The previous list is always greater than the following.*

3) There is a tendency of considering spatial factors, most of students try "to guess" where in the list the searched word could be. This is revealed because some students change the extremes of the lists according to their guess.
Excerpts
Gimena:
Q17:
R: It influences in the same way that with the dictionary, that as we know the order, one starts getting to it.

Q: Here you had from W1 to W20, we got this result and then why do you put here from W14 to W7 and not for example from W17 to W19 ...
R: Because I suppose more or less that from W7 to W14, knowing the place where letter ”l” is, I guessed that it was a non-intermediate point ..., near the place where I could find the word *luna*.
Spatial factors.
Sergio:
Q17:
R: In reducing, in changing the parameter and the size.
Q: You said 1st reducing and then changing ?
R: Of course, because in fact I would be lucky to get just to the parameter but I could have made a mistake with the distance ...
He refers to the fact that he chose 1st W5 to W11 at random.
Q: How could you make sure that it would always reduce?
R: Taking the greater first.

4) The simulation of the search and the related questions clearly generate advances in the conceptualization of the method: all students realize that the method is the same in each search and that the lists vary.
Excerpts
Gimena:
Q18:
R: Yes.
Q: Determine.
R: *Equal things would be a procedure, but the procedure in itself, one of the things that vary in the procedure is that what W to what W I want.* And another thing that varies is that when I say ”choose word”, the words are different and have a different order respect to the word I am looking for. For example: we found a word that was $>$, another $<$.
Sergio:
Q18:
R: Yes.
Q’18: R: The word we are looking up keeps the same, the words that are $>$ or $<$ change.
Q: Look at what you wrote, what remains the same?
R: *The method.*
Sofía:
Q18:
R: Yes.
Q’18:
R: *The procedure is the same.*
Q: And what varies?
R: The results.
Q: What do you refer to ”procedure”?
R: That 1st you have to form, then choose and after define. It is always the same.
Q: When does that procedure finish?

R: *When the relation I write is "equal".*

Question 19

The segments corresponding to questions Q6 and Q19 are presented together, because they are essentially the same question, posed before and after the activity. In this way, the effect of the interview by comparing both responses can be detected. The question is thus to determine whether the interactive activity has helped the students to improve their levels of conceptualizations of the algorithm. The focus is on, on one hand, to determine if after the interview, the students succeed in decomposing the method in its components. The relevance in the case of recursion is that one of the components is the method itself. The epistemological motivation lies in Piaget's expression "reaching the intern mechanism of subject's actions" pointed out as the way in constructing a concept. On the other hand, to determine to what extent the students advance from referring to a particular word in responses to Q6 to describing a general algorithm in responses to Q19. The epistemological motivation arises from that the ability of detaching the thought from particular cases is considered an advance in conceptualization by Piaget. To illustrate the increased level of conceptualization, the relevant parts of responses to Q6 and to Q19 are indicated in italics.

Q19:	If instead of giving instructions to a robot, you had to explain to a little child -who knows how to read and knows the alphabet- how to look up any word in a dictionary, what would you say?

In Q6 the student is asked to explain in detail how he/she look up the word *gato* in the dictionary.

Facts

In the answers to Q6, the actions choose, compare and "do the same" are implicitly mentioned. The students refer to the success of the method with expressions like "*gato* was going to appear ...", "I will find it ...", "and so on", etc, indicated in italics in the excerpts below.

In the answers to Q19, all students show an improved conceptualization of the method they apply, explicitly talking about *choose, compare, do the same* and in some cases also about the termination condition. Most of all students describe a general algorithm as well.

Excerpts from Q6 and Q19

Andrés:

Q6:

R: I opened more or less in the middle and saw which letter I was, I was in "c", I think, and I went forwards ...

Q: Why did you go forwards?

R: Because "g" is after "c", then ... I went too forwards and came back a little backwards. When I got to "g", I was in "gp" and then I went backwards ... When I got to "ga" and there I looked for ... "gas" ... *and a little bit downwards* gato *was going to appear.*

Q19:

R: He has to check if the 1st letter of the word he is looking at there, where he

opened the dictionary (*choose*), is > or < according to what we determined here, that the first letter of the word that he is looking for and that he goes forwards or backwards in the dictionary as it is > or < (*compare*).
Q: And what does he do afterwards?
R: When he gets to that letter he is looking for (the 1st of his word) within the group of all the words that start with that letter, he does the same but with the 2nd letter (*do the same*).
Felipe:
Q6:
R: I check the beginning letter, with "g", there is an alphabetical order then, more or less I know that it is in the middle upwards, then I check which letter I am, for example: If I open to the "e", I know that "g" is after, that is the order, I go forwards, I go on checking and if I advance too much, I come backwards and so on, *until I find "g" and after that I search.*

Q19:
R: I tell him the same as to the robot, first that he opens the dictionary, that he reads one word (*choose*) and he defines the relation, that is what he tells me if the letter with which he started is before or after than the one he wants to find and he tells me that if it is after (he passes by) I ask him to choose another that is before (*compare*) and in this way successively (*do the same*).
Gimena:
Q6: Describe it to me again.
R: Good ... I open the dictionary (she laughs) and go to a certain letter, then I know, for example, at this moment it is letter "c". I know that from this part of the dictionary backwards, nothing is worth and that I have to go on advancing to be able to find letter "g", then I go on, I come across "h", I know it is ahead from "g" so I have to go back again *and look for* gato *and I find it.*

Q19:
R: First open the dictionary and we find with letter x (*choose*), then we see which the 1st letter of the word we are looking for is, for example *gato*. Then I would ask him "you know towards what place in the dictionary you can find it, knowing that you are stuck on letter x. This is to say letter x is before or after letter "g" (*compare*). And in this way we go on reasoning until we can find the word.
Q: That would be a 1st time, suppose he tells you "I know that *gato* is after letter x, what do I do now?"
R: If you know that it is after that letter, we will locate the letter with which *gato* begins. Once we have it, let's see which letter follows in *gato*, then, supposing that ... the "g" and the "a", he is going to start looking for that way.
Q: Does he do something completely different to what he had been doing?
R: No, he does the same (*do the same*).
Ignacio:
Q6:
R: First knowing the alphabet I started to search, I found letter "h" so that I know that it is one before, and found letter "g", and then I look up "ga" and I found "ge"; I know that "a" is before, so I came back and looked up "gat" *and I found* gato.

Q19:
R: I would say that you check the 1st letter, open the dictionary and focus on the 1st letter of the word that you found in the dictionary (*choose*). If that one is before, you should turn over pages until you find the one you're looking for (*compare*) and once you found it, you do the same with the 2nd, the 3rd ... (*do the same*).
Juan Andrés:
Q6:
R: First I opened the dictionary in a section that seemed near to "g" and went in order always visualizing the alphabet in my mind. I went up to "g" and then with the 2nd letter, which is the "a", and then the "t" *until finding the word.*

Q19:
R: That the word that he wants to find will be in order according to the alphabet that he knows. Then, what he has to search first is the beginning letter of the word. After having found the first letter, he can start with the 2nd. Proceeding also in the same way.
Q: The kid is learning to search ...
Then, when he opens the dictionary, what does he have to do to find the 1st letter?
R: A relation between what he is seeing (*choose*) and what he wants to find and according to this relation (*compare*) he "operates" forwards or backwards.
Q: And when he finds the first letter?
R: He operates again using the same order (*do the same*).
Nicolás:
Q6:
R: First, I opened the dictionary working out the place of letter "g" , that is a little before the middle of the dictionary, then, when finding letter "g", I know the letter following is "a", which is the first letter in the alphabet.
Then, it has to be at the beginning within "g" *and same successively.*

Q19:
Q: So, he opens the dictionary and ... ?
R: If the letter he finds (*choose*) is after, he will have turn over pages forwards, otherwise backwards (*compare*). We look it up. Then he will have to check the second letter ...
Q: And do something completely different?
R: No, the same, only in the section of letter "g". Within the little bundle of letter "g" look up letter "a".
Q: What does it mean "look up letter "a" "?
R: Start to do the same, only in the section of "g" (*do the same*).
Sergio:
Q6:
R: Searching by letter, 1st I searched by the letter, then I did it by the 2nd letter.
Q: What is the meaning of search? The dictionary is closed.
R: First I opened the dictionary, looked for a section, a number of letters, first from "a" to "g", first I looked for the second letter, then the third *and in this way ...*

Q19:
R: First I ask him to look for the 1st letter and then the 2nd and so on.

Q: In this way he will not know how to find.
R: Of course, if they know the alphabet.
Q: Knows the alphabet, but that doesn't mean that they know what to do when you tell them "Look up".
R: Opens the dictionary and start to see that the letters are > ... no, no ... they're <, I mean they're before ... that they open the dictionary and start searching at the beginning ...
He laughs, because he knows it is wrong. It is a typical behavior: faced to something difficult to explain, argue anything (alpha behavior).
Q: That is not the method you used. Let's see, opens the dictionary and what do they do?
R: They look for the 1st letter.. Oh, of course, because first he has to know how to do to find the first letter. Sure, he has to find the first letter.
Q: What does it mean? Suppose that he opens on a page with words with "c".
Observe that, although I suggested the special case "c", Sergio went on describing a general algorithm.
R: Oh, of course that if it is before the letter he searches, he has to go forwards, he has to find the letter (*compare*).
Q: What does he do while going forwards?
R: Takes letters out, diminishes the group, goes on, gets to the letter and ... he has to look for the other letters.
Q: And that means he has to do something completely different from what he has already done.
R: No, he does the same (*do the same*).
Sofía:
Q6: I was checking on the letter according to the order that I have in my mind, first, the first letter, secondly the second *and so on.*
Q'6: Doing what?
R: Discarding the ones didn't help, and looking up the words with the letters I was searching for.

Q19:
R: Take the dictionary and choose one word (*choose*) and then with respect to that one, start looking if it is > or is < (*compare*) until finding the 1st letter and after you found the 1st letter, you have to look for the 2nd , in order as well, and go on until you find the whole word (*do the same*).
Q: Is what he does to find the 1st letter totally different from what he did to find the 1st?
R: It's the same but now with another letter.
Iván:
Q6:
R: I opened the dictionary, I looked at the letter I was on that page (*choose*) and if "g" was before, I went backwards and if it was after I went forwards (*compare*) until finding letter "g" and then the same with the "a", with "t" and "o" (*do the same and termination*).

Q19:

R: That he went searching letter by letter of the word ... that is, he took the 1st letter of the word and searched for it alphabetically.
Q: What does that mean? The kid opens the dictionary ...
R: Reads one word (*choose*) and checks if the 1st letter of the word he is looking for is > than the 1st letter of the word he found, goes backwards in the dictionary (*compare*).
Q: What is he still doing?
R: In this way he looks for the 1st letter, then he does the same but with the 2nd letter and the others (*do the same and termination*).

3.2.3 Summary of main results

As expected all students use the algorithm of binary search to find a word in a dictionary. Their instrumental knowledge of this recursive algorithm is transformed into conceptual knowledge by means of reflecting about what they do and why it works. This transformation develops in the interaction between the coordination of students' actions and the modifications they impose to the object, ending in a form of equilibrium giving rise to the concept of this recursive algorithm. Observe that students' responses to the questions Q7 to Q14, reveal that they understand that they find the word because they apply *the same method* to a sequence of parts of the dictionary *each time smaller* ending in a special case. It is straightforward to state a recursive formulation of students descriptions, from which a recursive function can be defined, as is shown in the next chapter.

On the other hand, questions Q6 and Q19 allow to some extent to measure the advances in students' conceptualization of the algorithm revealed by the decomposition of the method in its constituents methods: *choose a word, compare words* and *do the same* over other element of the structure in which recursion is based.

3.3 Second problem

The second problem posed to the students is an instance of the problem of calculating over an inductively constructed structure using a recursively defined algorithm, according to the way by which the elements are generated [Acz77]. Most of the problems in which operations over inductively defined data types have to be determined admit relatively easy recursive solutions in this way. A brief summary of the mathematical definitions can be found in Appendix A.

The students have previously faced this type of problems in the tree problem of the selection phase and their responses have shown that almost all of them calculate on the number of branches instead on the structure itself. One of the goals of this part of the interview is to make the students aware of the strong relation between the rules generating terms and a definition of the method of calculating something over them.

The following sections describe the theoretical ideas behind the design of this part of the interview, the problem, the questions and the expected results and finally the analysis of the responses including some excerpts of the interviews.

3.3.1 Evolution of recursive reasoning

A brief summary of the main ideas supporting this part of the empirical work is included. More about that can be found in Appendix A.

In [Pia63], the psychogenesis of reasoning over the series of natural numbers is deeply investigated and results are presented. Piaget and collaborators have confirmed the existence of a general mental structure whose elements are terms, transformations *and* a form of reasoning on both (terms and transformations), which is the source of reasoning by recurrence[5]. This form of reasoning evolves from generalizations of iterative inferences to higher levels of reasoning by recurrence. That means that it is the same structure that becomes more flexible and efficient and is enriched in its evolution from childhood to adult age. For instance, the group of numbers does not arise from imposing to the numbers of childhood thought a group structure corresponding to adolescence thought, that is, they are not the numbers that hold one structure or another, but a mental structure corresponding to the numbers of childhood thought that is transformed (terms, operations and reasoning) into a higher one (group structure of numbers) corresponding to adolescence thought.

Piaget distinguishes in the evolution of reasoning by recurrence on the series of natural numbers, several stages from childhood to adolescence in which the numbers become "any thing, *subspecie iterationis*", that is to say, the numbers are abstracted to elements generated by iteration. The evolution of the implicance from an isolated relation between terms to the same relation inserted in a whole structure, plays a relevant role in this construction. Piaget points out that this constitutes the dawn of reasoning by recurrence.

In [Pia63] referring to an experiment by which the subjects have to construct two collections of pearls and answer some questions about it (have the collections the same number of pearls? etc), the authors point out that once the subject establishes a coordination between the succession of his/her actions and their result, a local synthesis specific to these actions is stated between the order of the succession $S_1S'_1, S_2S'_2$, etc. and the growth of the collections C_1, C'_1, C_2, C'_2, etc., extending the construction of the number with an aspect of inferring by recurrence, where the most important generalization is not the passage from 1 to n, but from n to $n+1$. This shows that reasoning by recurrence is from its beginning indissolubly connected to the construction of the series of natural numbers and constitutes its aspect of inference, long before the elaboration of higher forms of reasoning by recurrence.

The mathematical formalization of this result is the set of Peano's axioms (including mathematical induction) defining the natural numbers and their operations. In this thesis, Piaget's results are extended to other types of inductively defined structures, whose formalization is given by the inductive definitions in [Acz77], briefly presented in Appendix A.

The correspondence between the psychogenetic evolution of the concept and

[5] Recall that from the psychogenetic point of view, the expression "reasoning by recurrence" is used in the sense of both calculating and proving and that the term structure is used in two senses, as *mental* constructions and as a mathematical object.

the historical stages of its formalization is a matter of further research. In [PG80] the authors investigate the analogy between the mechanisms of the psychogenetic development of intelligence and those of the socio-genetic development of theories in some scientific domains. The results of this analysis is the essence of Piaget's theory from which unvaluable contributions to pedagogy can be drawn.

3.3.2 The problem

An inductive definition of a set of words is presented to the students and they are encouraged to find a way of calculating some value.

The definition is presented to the students as follows:
Suppose that the inhabitants of an unknown planet have a language such that words are formed just using "a" and "b", according to the following rules:

- ab is a word.
- If * is a word then a*a is a word.
- If * is a word then b*b is a word.
- Only the words obtained by application of the above rules a determined number of times are words of the language.

The rules indicate that each word is either defined in terms of the previous one, or it is an initial given word. In this sense, it is an inductive definition analogous in its construction to the series of natural numbers. In the interview, the students are required to develop a method to count the number of a's of any word and it is expected that they derive a recursive solution to the problem.

According to the premise stated in Chapter 1, page 8, the forms of reasoning allowing to derive recursive computations arise from the construction of the concept of the structure of the words by similar mechanisms that for the case of natural numbers. In such construction the evolution of certain relationships plays a fundamental role. The questions attempt to help the students in understanding those relations between the words.
Induced by questions Q1 to Q6, students focus on the generation of any word from the initial element "ab", that is, on the relationship $ab \rightarrow w_n$, where "ab" plays the role of the natural number 1 in the construction of the series of natural numbers. This level is called *Level 1.*

Q1:	Write some words of the language.
Q2:	Can you determine of these sequences which are words of the language and which are not.
Q3:	Why is this one a word and this one is not?
Q4:	How did you form this word?
Q5:	Let's see this word of the language, which rule did you use to form it?
Q'5:	Which was the last rule used?
Q6:	And this other one?

The goal of the questions Q7 to Q11 is to cause students' thinking advance towards the relationship between any word -represented by * and called "the little symbol" in the interviews- to the next one, that is to say, $w_{n-1} \to w_n$. This generates the most important generalization giving rise to the understanding of the implication of the inductive definition. This level is called *Level 2*.

Q7:	In this one, which would the little symbol be? And in this one?
Q8:	Could the little symbol be like this? (Writing a sequence that does not belong to the language).
Q'8:	Why not or why yes?
Q9:	Then, what does the little symbol have to be?
Q10:	A student said that rules 2 and 3 say that a word is formed from a word formed before. Do you agree?
Q11:	Then in order to know if a sequence of a's and b's is a word, what can we watch?

The possibility of defining recursive algorithms on the elements generated by the rules, arises from the construction of the inverses of the relations of level 1 and level 2. Q12 to Q'13 are aimed at generating both $w_n \to w_{n-1}$ (called *Level 3b*) and $w_n \to ab$ (called *Level 3a*) from which the inductive and the base cases of the definition can be respectively derived. Observe for instance that questions Q13 and Q'13 are aimed at making students' thought to interact between his/her definition of the method and its application to a particular case. The coordination of both defining and applying the method leads to an equilibrium in which the need of the base case to complete the definition is understood.

Q12:	How many a's does this word have?
Q'12:	And the little symbol? And here? (The complete word).
Q"12:	In any word, if we know how many a's the little symbol has, can we determine how many a's the word has?
Q"'12:	How? Write it down please.
Q13:	Determine the a's of ababaaaabbbabbbbaaaababababa by using only what you have written.
Q'13:	What is missing in what you wrote so as to be able to use it until the end?

Expected results
It is expected that the students solve the problem of calculating the a's of any word, developing a recursive algorithm accordingly to the way the words are generated by the rules and that they correctly describe it in Spanish.
It is also expected that the application of the algorithm to a particular case helps in the design of the final solution. The students are required to write down all their work.

3.3.3 Analysis of the responses to the second problem

The goal of the analysis is to identify the levels (1, 2, 3a and 3b) of the construction of the concept of the structure described in the previous section. The relevant question to the identification of level 1 is "with which rule did you form this word?" Six students answer "with the first, the third, the second ..." thinking of the relation between "ab" and the current word (level 1).

The relevant question to identify the passage from level 1 to level 2 is "which is the little symbol in this word?" Eight students answer that it is "ab", revealing confusion in differentiating the initial element from the previous element of any word. In the development of the interview, all students success in differentiating those elements and in recognizing the little symbol as a representation of the previous word in the cases of words generated by rules 2 and 3 (level 2).

The following excerpts show that the greatest difficulty in the construction of mental structures for the rules, is the passage from level 1 to level 2, while level 3 (a and b) is quite quickly attained. The comments of type "He/She writes it well for rules 2 and 3" mean that the student correctly defines the algorithm for the inductive cases. A synthesis of the complete definition of the algorithm derived by the students is included in section 3.4.

Excerpts
Andrés
Q5: Let's see this word of the language, which rule did you use to form it?
R: With the 1st and with the 3rd rule.
Q: This one ?
R: With the 1st, the 3rd and the 2nd.
level 1
Q7: In this one, which would the little symbol be?
R: (He does it quickly and correctly for all the words).
Q8: Could the little symbol be like this?
R: No.
Q: Why not?
R: Because it does not follow the rules (immediate).
Q10: A student said that rules 2 and 3 say that a word is formed from a word formed before. Do you agree?
R: Yes.
Q: And that previous word, what would it be, then?
R: "ab", the rule number 1.
level 1
Q: Are you sure ...?
R: Oh no, it could be the mixture of the 3 rules.
Intermediate from level 1 to level 2.
Q: What was this? (I point to a * that he already marked using a circle).
R: ...
Q: I made you a question and you marked this with a circle, what was the question? Do you remember?
R: What was the * there? (He means in that word).
Q: So this is ...
R: The *.
Q: So, according to what we said of what the student said, which is the previous word so as to form this one?
R: The little symbol.

Q: Then in each word, what is *?
R: The previous word.
level 2.
Next series of questions reveal how slow is this passage, Andrés goes back to level 1.
Q11: Then in order to know if a sequence of a's and b's is a word, what can we watch?
R: If it obeys the 1st rule and then the others.
Q: With respect to the *?
R: It has to follow the rules.
*He does not relate * to the current word*
Q: What is * in this word?
R: "ab".
Level 1.
Next he is again induced to think about the relationship between two consecutive words.
Q: No, what was the *? Remember.
R: The base word.
He means "ab", level 1.
Q: No, (pointing to correct previous answers about *), it was this here, here, this, here this, what was *?
R: The previous word to the last application (of the rules).
level 2.
Q: And what must happen?
R: It must obey the rules.
Q: Does it obey?
R: No.
Q: Why not? How do you do to know?
R: (He marks all the corresponding little symbols correctly). And this one does not have it anymore (he got to the one that did not obey).
Q: What is it that you do?
R: I'm coming backwards.
Q: Watching what?
R: The little symbol in the words.
Level 2 seems to be consolidated, which causes that in the next part about the method of counting the number of a's, Andrés easily discovers the algorithm.
Q12: How many a's does this word have?
R: (He answers particular cases well).
Q"12: In any word, if we know how many a's the little symbol has
can we determine how many a's the word has?
R: No, because depending on the little symbol I applied rule 2 or rule 3.
Q: And can you determine the numbers of a's in every case?
R: ... yes.
Q: How? Write it down.
He writes it well for rules 2 and 3.
Level 3b.
Q13: Determine the a's of ababaaaabbbabbbbaaaababababa by using only what you

have written.
R: I mark the little symbol and has one, two ...
Q: Without counting, always applying what you wrote.
In the following lies the essence of recursion.
R: Oh! Sure, the ones of the little symbol + 2 ... I mark the little symbol again and it will have the same number because it is rule 3, I mark the little symbol again and now the rule I applied that was rule 2. So, it will be 2 +, is this way OK?
Q: Yes, yes.
R: And I now mark again and the rule applied was number 3 so that it remains likes this, as it is, and then I applied rule 2 so it will be 2 + and here I applied rule 2, so 2 +, then rule 3 so, nothing and there is the first.
Q: And how many a's does it have?
R: 1, always + 1 (he sums up).
Q'13: What is missing in what you wrote so as to be able to use it until the end?
R: ... rule 1.
Level 3a. **He completes his algorithm with the base case.**
Q: Let's see if the result is OK.
(We always ask the students to verify.)
Gimena
Q: If this word is formed making use of the rules, and in the rules the little symbol * appears, this means that in your word the * is something. Do you agree?
R: Yes.
Q: Well, which is * in this word?
R: "ab".
level 1.
Q: Are you sure?
R: I could not tell, I don't know.
Q: I want you to make sure. How could you be sure?
R: Good question ...
Q: This word, with which rule did you form it?
R: With the 3rd and also with the 1st , knowing that "ab" is a word.
level 1.
Q: What was the last rule you applied?
R: The 3rd.
Q: If the 3rd rule was the last one you applied and you tell me that "ab" is the little symbol, put "ab" and apply the 3rd rule. (She does it).
Q: Is it the same?
R: No, it is not the same.
Q: So, applying the 3rd rule when "ab" is the little symbol, we obtain this word. Then, which is the little symbol here? (In the original one).
R: The little symbol would be all this. (She does it correctly).
level 2.
Q: In this one?
R: (She does it right for several words).
Level 2 seems to be consolidated, which causes that in the next part about the method of counting the number of a's, Gimena easily discovers the algorithm.

Q12: How many a's does this word have?
R: (Particular cases).
Q"12: Knowing the number of a's of the little symbol, can you determine the number of a's of any word?
R: No, I cannot determine the number of a's because I do not know if that word finishes with "a".
Q: Which is the relation between the number of a's of the whole word and *?
R: Oh! It can be +2 or -2.
Q: (We review particular cases.)
R: Oh! + 2 or the total number of a's. It could be that the number of a's of the word is the same number of a's as there are in *.
Q: What does it depend on, that is one thing or another?
R: It depends on the rules.
Q: Write it down for any word in which the * has any number of a's.
She does it well for rules 2 and 3.
Level 3b.
Q: I give you the word ababaaaabbbabbbbaaaababababa and ask you that using what you wrote, work out the number of a's. (She does it speaking all the time.)
Q: What do we do then?
R: We count the ones from the little symbol.
Q: Very good, come on.
R: One, two, ...
Q: NO, NO, using what you wrote.
R: I can't.
Q: Why not? Count the a's from the little symbol using what you wrote.
In the following lies the essence of recursion.
R: Oh! (Thinking) Yes, again, we do it with *, yes, yes, yes, I understand now, in this case, if I cross this out, I get it this word, then it will be + 2.
Q: Write it. (She put 2 +, following the previous one).
R: Then I do it again and I get 2 + , I do it again and I get 2 + and now I start to do the same again and I get to the first.
Q: And then?
R: I cannot decompose "ab" any more.
Now, she discovers that her algorithm is incomplete.
Q: But how many a's does it have?
R: One.
Q: So, what do we do now?
R: + 1.
Q'13: What is missing in what you wrote so as to be able to use it until the end?
Level 3a. **She completes the algorithm with the base case.**
Sergio
Q4: How did you form this word?
R: With words that are already made.
Q: For example, which rule did you use to form this one?
R: First, the first rule, then the third and then the second.
Level 1.

Q: Was the last rule you applied the second?
R: Yes.
Q: See that the rule talks about *, so, which is the * in this word?
R: (Quickly and without doubting) "ab".
Level 1.
Q: Are you sure?
R: Yes.
Q: You say that the * is "ab" and the last rule you applied is number 2. Let's put the * and apply the second rule. What word we get? Is it the same?
R: No, it is not the same, but it is a word.
Q: Yes, but I asked you which is the * in this word and you told me "ab".
R: Oh, yes.
Q: And that you applied rule number 2, then what would it have to happen?
R: Be the same word.
Q: And is it?
R: No.
Q: Then, what is it that's wrong?
R: That the * is not "ab".
Q: Which is it then?
R: "babb".
Q: Mark it. (He does it well).
Q: Here, who is ?
(It is correctly done many times).
Level 2 seems to be consolidated, which causes that in the next part about the method of counting the number of a's, Sergio easily discovers the algorithm.
Sergio correctly writes the corresponding part of the algorithm for rules 2 and 3.
Level 3b
Q13: Determine the a's of ababaaaabbbabbbbaaaababababa by using only what you have written.
R: The same. (He means "the same as in the previous word, because it is a word that begins and ends with "b").
Q: The same as what?
R: As the previous word.
Q: Which is the previous word? Mark it.
(He does it well).
Q: How many a's are there here? Using what you wrote.
R: 2 fewer.
Q: What did you write?
R: 2 more.
Q: So we put 2 +.
R: But why 2 +?
Q: You told me.
The following part of Sergio's interview shows to what extend the students *know how to do* and are unable of *formalizing it*:
R: Oh no, yes, yes. We put 2 more than the *.

In the following lies the essence of recursion.
Q: And which is the *?
R: (He laughs) 2 +.
Q: Yes.
R: 2 + ...
R: And 1 +.
Q: Let me see if it is OK.
Q'13: What is missing in what you wrote so as to be able to use it until the end?
Level 3a. **He correctly completes the algorithm with the base case.**

Next interview is an example of the relevance of the evolution of the implicance as a relationship inserted in a structure. That means that the relationship $w_n \to ab$ has to be constructed as the inverse of $ab \to w_n$, which in turn is constructed from the rules generating $ab \to w_1 \to w_2 ... \to w_n$. It is not the same to construct $w_n \to ab$ isolated than inserted in a structure. This shows the strong connection between the comprehension of the structure and the method.

Iván
Q1: Write some words of the language.
Q2: Can you determine of these sequences which are words of the language?
R: Yes.
Q: Is this a word or not?
R: It is not.
Q: Why not?
R: Because ... (takes time). A word is always formed from another one and a word will always have equal extremes. Then, I have to see if this is a word. I know that it could be a word because it has both extremes the same. Then if I took the extremes out and I observe if the new extremes are equal and like this successively. When I get to two letters, if they are "ab", it is a word.
Q: In this case?
R: No.
Q: Let's see, doing what you said? (He does it OK).
Q5: Let's see this word of the language, which rule did you use to form it?
R: Rule 2.
Q7: Rule 2 uses the *. In this one, which would the little symbol be?
R: (He does it well with many.)
Q: How would you define the *?
R: It is the word without extremes.
Because of his correct verbalization, no deeper interaction with the construction of the words takes place. However, later in the interview, it is revealed that the role of "ab" was not understood and consequently the meaning of the implicance was not constructed.
Q12: How many a's does this word have?
Q'12: And the little symbol? And this other one? Its little symbol?
Q"12: In any word, if we know how many a's the little symbol has can we determine how many a's the word has?
R: We should know what rule we are using from the * to form the new word.

Q: How do we go about, then?
He writes it well for rules 2 and 3.
Level 3b.
Q13: Determine the a's of ababaaaabbbabbbbaaaababababa by using only what you have written.
R: This is the little symbol, I have to count the number of a's there are.
Q: Yes.
R: Do I count? (he means "count one by one")
Q: No, use what you wrote.
R: But ... To use what I wrote, I have to know the number of a's that the little symbol has.
Q: Does your method help for any word?
R: If I know the number of a's the little symbol has, yes.
Q: Is the * a word?
R: Yes.
Q: Then, please, use what you wrote to work out the number of a's of the *.
R: ... I put the first word ... ("ab")
Observe that the difficulties in confusing the initial word with the previous one arise here. Because of the absence of connection to the first word, the relation between any word and it is not constructed. This prevents Iván to realize the possibility of knowing something for any word if it is known for the previous and *the initial word.*
Q: No. You are not using what you wrote. The quantity of a's of this whole word is going to be, according to what you wrote ...
R: One more ?
Q: Why one more?
R: Because we have the first word.
*His thought is fixed in the belief of the need of knowing the number of a's of * to get a result and not of the need of the* **possibility** *of knowing it. The incomplete interaction with the construction of the words, prevents Iván from reaching the coordination between all the involved factors: the initial word, the previous word of any word, etc, and consequently the relation $w_{n-1} \rightarrow w_n$ is constructed* **isolated** *from the structure $ab \rightarrow w_1 \rightarrow ... \rightarrow w_n$ of the language instead of inserted in it.*
Q: But in what you wrote you do not talk of the first word at all.
R: But to know the number of this *, I have to know the number of previous * and the previous one, and the previous one.
Q: Exactly. Do that, using what you wrote for this word.
R: It will be ... depends on the case I have.
Q: Yes, which rule did you use to form it?
Observe that this is a typical question to understand the difference between the initial and a previous word.
R: With number 2.
Q: So?
R: I add up 2 to the supposed number of a's of the *.
Q: Good, then it is 2+.
In the following lies the essence of recursion.
R: 2+ and now the other word was formed with the * of the other word and the

third rule and if I use this rule, I know that it has the same number of a's, (2+0). The new * is 2+ the previous and the previous is 2+* and now I cannot use what I wrote.
Q: No, but how many are there ?
(He writes 1 and adds up and verifies that the result is right).
Q: It's really true that when you get to "ab" you cannot use what you have written, so what would be missing?
R: What happens is that the 1st word is not formed from the previous one.
Q: But what do you always know?
R: If the word is the 1st , the number of a's is 1.
Iván reaches level 3a. **He correctly completes the algorithm with the base case.**
Q: Would it possible not to get to the first word?
R: I always get to it because from the 1st all the others are formed.
Observe that the indicated reason reveals the construction of $ab \to w_n$ *from* $ab \to w_1 \to w_2 ... \to w_n$.

3.4 Summary of the main results

In this section we have presented the analysis of the gathered information from the responses of the students to the series of questions about the second problem. Evidence arises about the relevance of the construction of the structure stated by the rules as the source of forms of thought that are the beginning of reasoning on recursive methods. Observe that the students themselves derive both the clause for the base case and the clauses for the inductive cases, which is remarked in the excerpts with the comments in bold: "He/she writes it well for rules 2 and 3" and "He/she completes the algorithm with the base case." That means the the students derive a complete recursive formulation of their solution to the problem, which can be synthesized as follows:

```
nr-of-a's-of any word: if the rule is 2, it is 2 + nr-of-a's-of *
                       if the rule is 3, it is nr-of-a's-of *
                       if the rule is 1, it is 1
```

From these informal recursive descriptions, it is straightforward to introduce a recursive function definition, as shown in the next chapter. In this case the domain of the function is the structure of the words given by the inductive definition and the codomain is the set of natural numbers.

Failures and advances are detected through the understanding or misunderstanding of the meaning of the implicance, in the sense of distinguishing the meaning of "*if something is known for any element then it can be known for the next one*" from the meaning of the affirmation "*to know something about any element it has to be known for the previous one*". The meaning of the implicance is precisely to state a *possibility* of obtaining knowledge about any element, *once* it is obtaining about the previous one, while the affirmation means the *necessity* of obtaining knowledge about an element to get it for the next one.

The construction of the result as part of the definition of the method demands coordination of students thinking about both the structure and the result. For instance, the need of the base case becomes understood when the method is applied, as revealed in the interviews.

Finally, the interviews reveal that one of the main difficulties in the construction of the involved relations -and consequently in the meaning of the implicance- lies in the transformation of instrumental into conceptual knowledge. This transformation takes place in a slow and hard process, which is (almost never) considered in traditional teaching of recursive algorithms as school subjects.

By analyzing the information gathered from the responses of the students, levels of conceptualization and cases in which these levels undergo an effective transformation have been identified. For instance, the responses classified as level 1 were unexpected and consequently this relevant information was obtained due to the interviews. This confirms the importance of considering what the students think and how they reason about some problem as the start point of constructing the pedagogical content knowledge of a concept.

Chapter 4

The empirical study, Part 2

4.1 Collective Class

A week after the end of the individual interviews, a collective class was taught to all students. Each student was provided with the sheet containing his/her previous developed work. The purpose is to represent students' descriptions in a formalism similar to mathematics as a means of connecting the concepts to their formalization. The content of the class is described below where explanations are transcribed in italics:

The dictionary and the position of a word in it are represented by [Wfirst - Wlast] and Wword respectively.

A definition of the method of searching a word in a dictionary that you have used in the interviews can be written as:

```
search(word,[Wfirst - Wlast]) =
        let word' = chooseword([Wfirst - Wlast])
        if word = word' then the-list-of-meanings
        else if word' < word then search(word,[Wword' - Wlast])
             else search(word,[Wfirst - Wword'])
```

From this specification of the method, a definition of a mathematical function can be derived. Implementing this function in a programming language is quite easy. One has to make small changes to fit the language syntax, for instance, the = symbol has two meanings in mathematics: "define" like in "word' = chooseword([Wfirst - Wlast])", and "are those values equal?" like in "if word = word' then ..." This is not acceptable in programming, so programming languages usually have another symbol for one of the meanings, for instance == for the equality test.

This function can be applied to find the meaning of the word luna *in the dictionary as follows: ...*

Discussion follows, allowing to connect several important concepts with the activities developed by the students in the first interview. For instance, the series of searches depending on previous results can be "folded" in the corresponding applications of the function to arguments that are actualized with new values according to previous results:

Recall your own words, when asked what changes and what remains the same in each search: "the same is done while the list and the result change". This is represented in the function definition and can be verified with the example of searching luna*, where we can see the way by which the arguments are updated.*
(Carefully development of the application of the method until the result is obtained follows.)

In the interviews, one of the students made an error in determining the list because he puts the extremes in a way such that the searched word was not included. At that occasion, he immediately corrected the mistake. The situation is described in the class and why the erroneous list does not serve for the purpose is discussed. All the students realize the reason and the general property that must be satisfied by all the lists is stated.

Another aspect that is pointed out is the termination of the process because of the list diminishes and the part of the definition determining what to do if the words are equal. The reasons were deeply worked upon in the first interview.

Finally, the following question was discussed: "How can we define a more efficient method 'choose-word' in the searching algorithm?" The term efficient is not specified to see what the students think about it. All the students have answered that they should take the word from the middle, and that this is more efficient because there is more possibilities of discarding a greater number of words.

In the following the method of counting the a's of any word of the language designed by the students, is formalized in this way:

```
count-as(word) = if word = ab then 1
                  else if word = a * a then 2 + count-as(*)
                        else if word = b * b then count-as(*)
```

As before, the function is applied to a particular case, remarking how the arguments are updated in each application.
The question of the domain and the co-domain of those functions is discussed emphasizing that a correct mathematical function definition has to include the signature of the function. In the case of the language it is easier to see because the students are given the formal definition of the language through the rules. In the case of the dictionary, discussion about the domain has arisen leading to the fact that it is a cartesian product.

Finally, some exercises were handed out to the students to be solved individually or in groups and some days afterwards a new individual interview was conducted in which they had to explain their solutions. This is described in the next section.

4.2 Third Interview

The exercises handed out in the collective class are as follows:

Exercise 1
Recall the trees of exercise 1 of the written material handed out before, (see section 3.1.2). The case in which red and blue lights have to be placed at random is considered. A mathematical definition of the set of trees is given by the following rules:

1) R is a tree.
2) B is a tree.[1]
3) if a_1 is a tree and a_2 is a tree, then a_1 R a_2 is a tree.
4) if a_1 is a tree and a_2 is a tree, then a_1 B a_2 is a tree.
5) Only the elements generated by application of these rules a finite number of steps are trees.

Graphically, the trees can be represented as follows:

```
      R               B
     / \             / \
   a1   a2         a1   a2
```

Define a method of counting the red lights (R) of any tree.

Exercise 2

How would you search for the word *gato* in a novel in which you know that the word *gato*[2] exists?

Exercise 3

1) How can the algorithm of searching a word in a dictionary be modified to take into account the case in which the word is not in the dictionary?
2) Given a dictionary and a word that is not in it, define a method of including the word in the dictionary at the right place.

Motivation

At the occasions of the previous interview and the collective class, the students have solved instances of problems and were introduced to the way of formalizing the solutions. In this way, the students have conceptualized their knowledge about these recursive algorithms. One of the goals of this interview is to know about the impact of previous activities in student's reasoning. On the other hand, this conceptual knowledge is attached to the solved cases (intra stage, [PG80]) and in order to make it evolve to an inter stage -characterized by dealing with the methods themselves as transformations to be combined with other ones- the students are faced to new problems presenting similarities and variations with respect to those already solved. The psychological instrument is called by Piaget *generalization* [Pia75] [Pia79] and one of the conditions of effective generalization consists in that the obstacles really can be solved not just by repetition of previous operations but by modifications of these according to the wanted results. The problems proposed to the students present appropriate variations in the structure (trees, novel) with respect to the ones of the first interview and face the students to the necessity of designing new (compositions of) methods (inserting a word, sequential searching).

[1] R stands for "red light" and B stands for "blue light"

[2] "gato" is the Spanish word for "cat"

4.2.1 Analysis of the responses to Exercise 1

Most of the students are not able to design a solution to the problem, if they do not work out the construction of the structure of the trees. Once they begin to do it, they easy and quickly derive a recursive formulation of the method of calculating, stating the clauses both for the base and inductive cases. In the excerpts below, that is indicated with comments in bold and a synthesis of students' solutions is included in section 4.2.3.

The following excerpts of the interviews are organized in two parts. The first one describes the difficulty in understanding the construction of the trees according to the given rules. The obstacles lie in the conceptualization of the meaning of the implication, as in the case of the language problem of the first interview. The students are not capable of making a tree using R and B as previous trees, that is to say, they cannot think of them as a_1 and a_2. While talking about the rules, they say: "having R I have to add two trees ...", thinking of deriving other trees from R (downwards) instead of thinking of a_1 and a_2 to be joined to R (upwards). This reveals not understanding of the meaning of clauses of the form "if a_1 and a_2 are trees then a_1Ra_2 is a tree".

The second part describes how the students -once this obstacle begins to be surmounted- succeed in defining the method in a more advanced way with respect to definitions of the first interview. The advances in the conceptualization of the recursive method are mainly two: firstly, the students reveal understanding of the need of definitions for the base cases, and secondly, the method is correctly referred by a name when deriving clauses for the inductive cases.

Excerpts

Iván correctly solved and formalized the exercises before the interview. Nicolás and Felipe had erroneous solutions and Gimena, Laura and Sofía had partially correct solutions. All students worked on the exercises during the interviews and got correct solutions.

If difficulties understanding the meaning of the implication are detected, the rules are worked out until those are surmounted. Students' previous work about the language exercise is handed out.

In the following, "I" means "instructor" and the responses of the students are indicated by the first (or the two first) letters of their names.

Ignacio

Ig: Count the red lights ...

Part one

I: The rules are those. What trees would you have?

Ig: (He draw some trees).

I: How did you form it, this one for example?

Ig: I put a B to the right and an R to the left.

I: To the left of what? This one for example, which rule did you use to form it?

Ig: By rule 1, R is a tree and by rule 2, B is a tree ...

I: So, what are the first trees you can have?

Ig: B and R.

I: So, which rule did you use for this one?

Ig: I put tree B to the left of R and to the right I put tree R.
Observe that his verbalization is better, he refers to tree R, tree B and a new R as the root.
I: Which is the rule?
Ig: (He reads it well). It is rule 3.
I: And this other one?
Ig: Rule 4. (It's OK).
I: And this one?
Ig: Rule 2.
I: And this one?
Ig: Rule 1.
I: Very good. So, now apply rule 3 to form a new tree with this one and this one.
Ig: With rule 3 ... by rule 3 I have to add other trees to the right and left of R. Having a tree R, adding a tree to the left and a tree to the right.
Observe that Ignacio has not understood the meaning of the implication.
Part two
I: Now we have to define a method to count R lights of any tree.
Ig: ...
I: Do you find any similarities between this and the language exercise?
Ig: Yes.
I: (I show him what he did before, in the language case).
Ig: What goes first is rule 1 or rule 2.
I: Yes, write it down please.
He correctly writes the clauses for the base cases.
Conceptualization of base cases.
Ig: Now I go down a branch and apply rules 1 and 2 again.
I: No ...
Ig: If the rule is 3 I count 1 for the one above and if the rule is 4 ...
I: You haven't finished with rule 3, that says (I repeat rule 3).
Ig: I take a1 as a tree and a2 as a tree.
I: Exactly, how do you write it?
Ig: 1+... I will go on using this rule (he gets confused).
I: Yes, but if the rule is 3, it is 1+ what?
Ig: Plus the trees below.
I: What is it we are counting?
Ig: The number of red lights.
I: So?
Ig: 1+ the number of red lights of the tree (He refers to a1).
I: Good, how do we call the tree?
Ig: a1. (He writes 1+ a1).
I: Write what you told me, that is, it is not the same to add a1 (which is a tree) as the red lights of a1 .
Ig: Yes, it's OK.
I: And what else?
I: 1 + red lights of a1 + red lights of a2.
I: Very good. And if the rule is 4?

Ig: 0+ ... .

He correctly writes the clauses for the inductive cases.

I: Does this method help you to any tree?

Ig: Sure!

I: Let's use it for this tree, for example we call it t. (We take any one that he constructed before).

(He does it well and we check it).

I: Why does the method end?

Ig: By rule 1 and rule 2.

Sergio

Sergio has a good conceptualization of the rules and he is asked to explain how he does.

Part one

I: How would you make some trees using these rules?

S: (He does it well).

I: Which rule did you use to make this one?

S: With 1, 2 and 3 ... a1 is R.

I: Can you explain it? Which are the minimum trees?

S: B and R.

I: Construct another tree with these two and explain how you do it. (Although he builds the trees correctly, he does not know how to explain how he did it).

I: Read rule 3.

S: (He does it).

I: Who is a1?

S: This and this. (He does it well).

I: Good. So build a new tree applying rule 3.

S: (He does it well).

(We repeat it many times, with the different rules. He does it well).

Part two

I: Now, how do we define a method to count the red lights of any tree? (I remind him of the language, what he did, and he reads it).

I: Can you see any similarity between the two methods?

S: Yes, the method is similar.

I: How?

S: If I add some things ... lights are added ...

I: So, how do we go about it?

S: When you apply a rule to the tree ...

I: Let's see.

S: Red ... by each tree that has red lights you add 1.

I: Sorry ... What do you mean?

S: That if the tree is a red one, you add 1.

He correctly writes the clauses for the base cases.

Conceptualization of base cases.

I: Good. What else?

S: With rule 3 I add one red light to the lights that are already in the tree.

I: And what shape does the tree have?

S: One red light plus two trees.
I: Which ones?
S: a1 and a2.
I: So?
S: Red-ones-in a1 + red-ones-in a2.
He correctly writes the clauses for the inductive cases.
I: Very good. Does it help you for all trees?
S: Yes.
I: Let's see.
(He correctly applies the method.)
I: So, what's the name of your method?
S: How is it called?
I: Yes, you have used it to count the red lights of a1 and a2, and you did referring to the method in some way. Could you see which is this way?
S: ... Red ones in the tree.

Andrés

Part one

I: What does the first rule say?
A: (He reads it).
I: So which are the first trees?
A: R and B. (He reads the other rules).
I: Which ones would a1 and a2 be?
A: The same way we have R and B we would have that a1 is a tree and a2 is a tree.
I: But we do not have such a rule. What does it mean "if a1 is a tree and a2 is a tree"?
A: a1 is R and a2 is B ... (He does it well).
I: Now form new trees, applying the rules again.
(He thinks and thinks, but at the end he does it well. We do several cases using rules 3 and 4).
I: Do you think there's any analogy with the exercise of the words?
A: Yes, what it has to do with the little symbol, whenever you finish making a tree you put it as the little symbol ...
Observe how after few questions Andrés improves his understanding of the rules, from "The same way we have R and B we would have that a1 is a tree and a2 is a tree" to "... whenever you finish making a tree you put it as the little symbol ...".

Part two

I: Do you remember how we counted the a's? We had defined a method, here there is yours (I give it to him and he reads it).
I: How can we count the red lights?
A: Counting the number of times that we used rules 3 and 4.
Observe how he relates the construction of the structure of the trees with the method of counting the red lights.
I: Correct, if rule 3 was used, how many R does the tree have?
A: It has one more, the R that is ...
I: That is where?
A: In the tree you have already formed.

I: Good, repeat that please.
A: It has one R plus the number of R that is in the tree.
I: In which tree?
A: In a1 or in a2 .
I: Exactly, only that it is **and** in a2, do you agree? And if it was formed with rule 4?
A: It's the number of R that is in a1 and in a2 .
I: Very good, using that you have already said, let's count the red lights of this tree.
A: (He does it correctly, quickly and secure. We check.)
I: To finish up with this exercise, could you write the complete method?
He correctly writes caluses for the base and inductive cases.
I: Why do we always finish?
A: They're smaller each time.
I: Correct.

Laura

Laura reads the rules and starts constructing trees with different a1 and a2. It is very difficult for her, even when she gets to make a tree, say with the rule 3, she is not able of continuing making new ones with rules 3 and 4. She works on several cases before following questions are posed.

Part one

I: OK, now how would we count the red lights of any tree? How did we count the a's of a word in the language exercise? (I show what she did then and she reads it). What would you focus on?
L: On the root ...
I: Yes, but who talk to you about the root?
L: The rules.
I: So? (We go on seeing according to the rules that R has 1, B has 0 and we get to rule 3. She doubts, we come back to the words and the *).
I: Who would be the little symbol in this case?
L: R.
I: (We see the language again). Who follows the role of "ab"?
L: R and B.
I: Correct, so who's the little symbol?
L: R.
I: No, we already said that it was "ab".
L: a1 and a2 (doubting seriously).

Part two

I: Very good, now suppose that we have a tree formed by rule 3, how do we count the red lights?
L: I would add ... for each R I would add another R.
I: What would you add up?
L: The R of a2 .
I: And what else?
L: And the R of a1 .
I: In the case of rule 4, what do we count?
L: The ones from a1 and the ones from a2 .

Observe Laura's reasoning is correct. Although we think that Laura is able to reach a correct recursive description of the method, we perceive a certain tiredness caused by the hard work about the structure and we do not insist with the interview.
I: Let's use what you tell me to count the red lights of this tree (She does it, but doubts a lot).
I: How many did you arrive at?
L: 4.
I: And does it have 4?
L: Yes (We always check at the end, it is important).
Gimena
Part one
I: Which are then the least trees we can form?
G: R and B.
I: Afterwards?
G: (She reads rule 3 and tells me: "I thought that from R, a1 and a2 went out ...
I: You tell me that R is a tree and that B is a tree and according to the 3rd rule these two can be a1 and a2 , can't they?
G: Yes.
I: So a1Ra2 is a tree; what is a new tree like?
G: (She does it well).
I: Now we have new trees, form a new one applying rule 4.
G: (Reads the rule slowly and thinking, and does it well). Yes, yes, now I understand.
Part two
I: Now we have to count the red lights of any tree.
G: ...
I: What did you focus on, in the language exercise, to count the a's?
G: On the rules.
I: Exactly.
G: R is a tree ...
I: If the rule is that one, how many red lights are there?
G: One, B is a tree, according to the 2nd rule, there's 0 red light.
She correctly writes clauses for the base cases.
Conceptualization of base cases.
I: Let's remember how it was in the language (I show her what she wrote in the language exercise). Which would be the * in this case?
G: R and B.
I: No, R and B that are the least trees, to whom do they correspond?
G: To the 1st word ("ab").
I: So, to whom does the * correspond?
G: Oh! a1 and a2 ! Yes, yes, yes, now yes.
(From this point she goes fast because she understands the role of a1 and a2).
G: According to rule 3 there is a red light more than the ones we already had.
I: Where?
G: What there is in a1 and what there is in a2 ... 1+ the red lights of a1 + the red lights of a2.

(First she puts + a2 and I correct her, she understands immediately, she tells me "sure, a2 is a tree". Then she writes "according to rule 4: 0 + the red lights of a1 + the red lights of a2").

She correctly writes clauses for the inductive cases.

I: Now we are going to use your method to count the red lights of this tree.

(Finally, she goes applying the method and gets to good results.)

Iván

Iván brought the exercise correctly solved: he presented a correct recursive definition of the method. The formal definition of a function is completed indicating its domain and co-domain. Iván's solution is described at the end of the chapter.

I: Did you find difficulties or was it easy?

Iv: It was easy.

I: Did you have a look at what we had done before?

Iv: Yes, yes, I saw it.

I: What similarities did you find with what we had done previously?

Iv: They are all recurrent (he means recursive).

I: And why?

Iv: We based on ... it is like a chain, so as to know something, we based on the previous one until getting to one that you know it is known.

I: Can you do it with any case, for instance, could you do it with the real numbers?

Iv: No, because there is not previous to another one.

I: So, what conditions must the elements have?

Iv: That are finite.

I: What?

Iv: The set of elements.

Observe that despite his good conceptualization of the method, his understanding of the construction of the structure is not completely correct.

I: No, the set not, because, observe that having the two first trees, you can always build another one.

Iv: The tree has finite letters.

I: Exactly, the elements are finite. And besides, what other feature is there?

Iv: That one can be made from the one before.

I: Very good. What do we use to define them?

Iv: Er... the rules.

Sofía

Sofía had had great difficulty in constructing the structure in the case of the language problem, so she had worked out that problem a lot. Because of that, Sofía has constructed a mental structure flexible enough to assimilate this new problem, as can be observed in her responses below, for instance, she has a good conceptualization of the role of the implication.

Part one

I: Read the rules. (She reads them).

S: a1 and a2 can be R or B.

I: Yes, sure. Let's see if you can build some trees (she does it well) which rule did you use to build this one?

S: Rule 3.

I: OK, let's see another.
S: (She does it).
I: Which is the rule?
S: Rule 4. (She correctly makes some more trees.)
I: Which are a1 and a2 in each case?
(She does it well.)

Part two

I: How can we define a method to count the red lights of any tree?
S: You divide by the first one, I mean, let's see, sure, you have ... with rules 3 and 4 is different, with rule 3 it is one thing, and with rule 4 it is another.
I: Good, so let's put what corresponds to each rule. It is what we did with the language ... do you remember?
S: Yes, yes. With rule 1 it is 1, with rule 2 it is 0, ...
Conceptualization of base cases.
... with rule 3 you have 1, you already know that you will have 1 ... and the others, ... they can be infinite ...
I: Not infinite, because we always apply the rules a finite number of steps. The number can be very big. How can it be worked out?
S: With rule 3 you have one more than rule 4.
I: Refer yourself to rule 3.
S: 1+ ... er er ... That is, if a1 and a2 have red ones, + the red lights of a1 and a2.
I: Very good, write that.
S: (She writes).
I: We are adding so we are going to put + instead of "and".
S: (She writes it well).
I: And rule 4?
S: Rule 4 would be ... er ... red ones from a1 + red ones from a2 .
I: Very good. Now what is missing is to indicate that this method is useful to count the red lights of any tree. Here you are using it to count the red lights of a1 and here, the red lights of a2. How is the method called, then?
S: How is it called? I don't know ...
I: You have already used that name because you counted the red lights of a1 or the ones of a2, using the same method to do it, in those cases the trees are a1 and a2 . So what name does the method have?
S: Red ones of ... anything.
I: Very good. Write it down please.
S: Yes, I agree.

She correctly writes clauses for the base and inductive cases.

I: Let's use it to count the red lights of this tree, say t.
S: It would be "red ones of t" ... (She does it well. We check).
I: Why does the method end?
S: Because a1 and a2 go smaller.
I: Until where?
S: Until B or R.

Nicolás and Felipe tried to solve the problem by *forcing* it to fit what they already had done in the problem of the language, revealing α behavior[Pia75]. The generalization is inductive in the sense that the thought remains attached to previous results with no transformation of the constructed operations to be adapted to the differences presented by the new problem. Just the similarities between the two problems are taken into consideration and the operations are repeated without modifications. The construction of concepts demands the integration of both differences and similarities into a new equilibrium reached by the process of constructive generalization. The interview is aimed to help them to start it.

Observe that Nicolás and Felipe's difficulties are related to the representation of the structure, not to the method, for which they have a correct conceptualization. For instance, Felipe correctly uses a name for it (count-R) in both sides of the equal sign and Nicolás can anticipate that the same method has to be used in both sides from the root which is the beginning of understanding the differences between the structures of the language (unary) and this one (binary).

The case of these students is a clear example of the pro and retroactive character of the interaction between the stages of conceptualization and formalization. In the interview, both solved the problem and gave correct recursive formulations of their solutions, described at the end of the chapter.

Excerpts
Nicolás
N: So, what I did was, following these 4 steps (he refers to the rules), I analized the possibilities that could happen. Then, the cases are that from an R we get 2B or that we get an B and an R or that we get 2R. With the B it occurs the same.
So, I had to define a method in order to count the lights R of any tree, I made a recursive method, as we had done before. From each light R could happen that it is the only one and that the other two are B or there is an B and an R or that they are R and R, so the number of R would be 3. And from each blue one the same. What I put is that ... to this I called *, (he wrote R+(2B), R+(B+R) and R+(2R) and called * to the expression between parentheses in each case) then when you see an R, what you do is adding 1, because this is that R plus what there is in the * and if you see an B, it is the same as * because with B you don't add anything.
I: Very good. Now, what you did for 3, is it valid for more?
N: Yes, it goes on opening. You have to do the same with what opens, I mean it is *the same method for those two.*
Felipe
F: count-R from the tree = if the tree is R, it has 1 red light, on the other hand, if the tree has R * R, then it is 2+ count-R(*). But if it has R * B or B * R then it is 1+ count-R(*). In the case of B * B so count-R(*).
I: Did you find any similarity with the language problem?
F: Yes, sure, it is recursive, you define for a base tree, and then you put the cases that the person could have done, if they put R to the right and B to the left or 2R or 2B, etc. It could be complicated if you put R, B and yellow for example.

4.2.2 Analysis to the responses to Exercise 2

In the Exercise 2 of the third interview the following question is posed:
How would you search for the word *gato* in a novel in which you know that the word *gato* exists?
It is an instance of the problem of searching an element in a list which in contrast to the dictionary, is not *alphabetically* ordered. The question is to investigate whether and how the students use the knowledge already constructed. Most of the students answer that they have to read *the whole novel* because there is no order in the novel, moreover some students think that it is not possible to find the word, "I will not find the word because there is no order in the novel".
That means that students' thought focus in the *differences*, that is to say, the factors that prevent them to use a previous successful method without modifications. The goal of the questions of the interview is to induce them to think about the *similarities*, for instance, the novel has *another order* that allows to employ another searching method (sequential) which success by the same reasons that the binary search does, that is to say, the parts of the novel become smaller.
Excerpts
The following excerpts illustrate the above statements.
Ignacio
Ig: I would read the novel.
I: All the novel?
Ig: Until finding it.
I: Oh! Then it is not always all the novel. What does "reading it" mean? I think you are interested in finding the word *gato*.
Ig: Do I look for words at random until I find *gato*?
I: Why at random?
Ig: ... I start by the 1st, then the 2nd, (words) the 3rd, the 4th, the 5th, the 6th.
Transformation of the method.
I: What does this search have as similarities and differences with a search in the dictionary?
Ig: Similarities, the list becomes shorter, the number of words in which I search. And differences ...
I: Did you start by the first word in the dictionary?
Ig: No, because there is an order in the dictionary and there isn't one in the novel.
I: Doesn't the novel have any order?
Ig: Concerning the words, no. (He refers to alphabetical).
I: What order does it have?
Ig: A coherent order concerning the reading.
Laura
(She reads 3rd question).
I: Why cannot we look for it as in the dictionary?
L: Because the words are not in order.
I: OK, then how do we do it? First, will you find it?
L: Yes.
I: How would you do it?

L: I read the whole novel.
I: You are not interested in the novel but in finding the word *gato*.
L: Yes, but as you don't know where it is, you have to read it from beginning to end ...
I: From beginning to end?
L: Well, until you find it.
I: Ahh! Then it might not be the end.
L: No.
I: Are there any similarities with searching in the dictionary? That is, we said that we found the word because the dictionary was reducing, and in this case?
L: You also start discarding words.
I: Which ones?
L: All the ones that are not *gato*.
I: But while you are reading?
L: You discard the ones you are reading.
I: So, what happens to the novel?
L: It becomes smaller as well.

The following excerpts of Nicolás and Gimena interviews are examples of inductive generalization focusing in the differences between the dictionary and the novel, causing erroneous beliefs: Nicolás believes that nothing is discarded and Gimena believes that she will not find the word.

Nicolás

N: Regarding the novel, well, here the novel doesn't have a relation of order among the words. The relation $>$, $<$ or $=$ does not exist so we would have to find it from the beginning ... or ... trying, let's see, let's see ...
I: Despite the relation concerning the order in the novel, do you think that you will find the word?
N: Yes ... what it is possible is to start reading it until finding the word.
I: How would you read it?
N: From the beginning of course ... what I cannot do is reducing as in the dictionary, where I start eliminating.
I: But if you start from the beginning, let's see, what would you do?
N: Well, I open it and start reading, I cannot discard absolutely anything.
I: Do you think you cannot?
N: If I don't know where *gato* is, the author can have *gato* as a last name for instance and I have to read up to the end.
I: Let's suppose that this is the novel, let's see how you do it.
N: I go on reading from top to bottom, I accomplish my order (he remarks). I could do it from bottom to top if I wanted. (He means from the end to the beginning). The intention is not reading the novel but finding the word *gato*.
I: Good, suppose this is the 1st. What do you do?
N: It is not *gato*, I go on.
I: And what do you go on with?
N: With the one below, (he means the following) If it is not *gato*, I go on until it is *gato*.

I: Do you think there is something that diminishes anyway?
N: Yes, of course, it does not make sense to read something and then read it again.
I: So, why do you find it?
N: Because the novel is reduced while I am reading it.

Gimena

G: The only possibility to find the word *gato* in a novel is by reading it, because the words don't have any hierarchical order.
I: You aren't interested in reading the novel but in finding the word *gato*. How would you do it?
G: The thing is that I have no way out, I have to read it ...
I: Read it, what does it mean?
G: To start searching the word *gato* ...
I: Where to start?
G: By the beginning of the novel ...
I: You mean you read the first word ?
G: Yes, yes, up to the last one, if the last one were *gato*, I mean until I find the word *gato*.
I: Does this method have any similarities with the one of looking up in the dictionary?
G: Yes and no.
I: Let's see.
G: Yes because you also search the word, only that in the dictionary you know that you will find it because as it is more accurately in an order ... you know you might find it, whereas in a novel ...
I: In the novel, using the method you say of searching by the 1st, the 2nd, etc, will you find it?
G: No, not because in the novel the words are not in order.
I: But you know that the word is there and you look for the 1st, the 2nd ...
G: Yes, yes, I'll find it, but it will be more difficult.
I: Is there something that goes diminishing?
G: Yes, yes, the number of words, I do not care about the words I have already read, until finding *gato* and the search is over. Yes, distances become shorter. (She refers to each time she searches she's "closer" the word *gato*)
I: So what similarity is there between the two searches while you're looking for, before finding the word?
G: In each of the two, it is the same.

Iván

Iv: Reading each word and if the word is not *gato*, I get rid of it.
I: Do you find the word *gato* with that method?
Iv: Yes.
I: Why?
Iv: Because it is in the novel and I go on using fewer words, the novel starts becoming smaller. The best would be that you take the first word and then the second and after the following, etc. Not words at random.
I: Sure. Oh! I thought that you were telling me that.

Iv: Yes, yes, I say so, but it could also be that you take one word at random, you eliminate it, then you take another at random and so on.
Transformation of the method.
I: Of course. Let's see, is the list in order?
Iv: It is not in alphabetical order.
I: Oh! But is there an order?
Iv: Yes, there is an order.
Felipe
F: I read until I find it.
I: Exactly, this is what you do, but what does that mean? Given that, in fact, we are not interested in reading the novel.
F: I take the 1st word, define the relation, you will get to the fact that it isn't, then you go on with the following ...
I: Is there anything that is diminishing?
F: Yes, sure, the list becomes smaller.
Andrés
A: I have to read the whole novel because it is not in order to look up words.
I: The whole novel always?
A: I would have to read the whole novel if the word *gato* is the last of the novel.
I: And if not?
A: If it is in the middle, up to the middle, it depends on where it is.
I: Anyway, do you think there are similarities with the search in the dictionary?
A: Yes, the space becomes shorter.
Sofía
S: It does not have a defined order so we have to look for word by word until finding it and it could be that it is at the beginning or at the end.
I: So, how would you search?
S: I would start by the beginning or by the end.
I: Are there any similarities with looking up in the dictionary, for example, are you going to find the word *gato*?
S: Yes.
I: Why?
S: Because you tell me it is there.
I: And why do you find it?
S: Because I start discarding the ones that are not.
Juan Andrés
Juan Andrés uses the expensive procedure of relating each element of the structure to its ordinal value, revealing weak conceptualization of the structure. Observe that despite the ennumeration, he searchs at random. This behavior may be caused by previous instruction.
JA: Well, I would number each word of the novel since the 1st word to the end. And *gato* would be the word (he means the word to be searched). We would choose a word from the list, we call it word1 , so if word1 is equal to *gato*, then it is over and if word1 s not equal to *gato*, we eliminate it from the list and we did the same again.
I: You mean that you always take one out at random?
JA: Yes, at random, or ... ok, another method would be to begin with the first word,

then the second, and so on ...
Transformation of the method.
I: Exactly, this is another method.
JA: Yes, but with the first one, (he refers to the random method), the list starts diminishing ... well, in this one too. That is, now I realize that in the other too, if we did not always take out, it wouldn't finish.
I: Is there something similar with the dictionary?
JA: Well, the novel has to be ordered.
I: In fact, that order already exists.
What other similarities are there?
JA: ...
I: Not with respect to the objects, but with respect to what you do.
JA. That are steps that are repeated, we are seeing different examples of something that can always be used.
Sergio
Sergio thinks of the sequential search from the beginning of the interview, his mental structures have already assimilated the novelty posed by this problem.
S: Look for one by one, because there is no order, it is more complicated. Taking a paper and revise if it is not there ...
I: What do you mean by "revise"?
S: Compare with all the words.
I: Is there any similarity with the other search?
S: It is also a method.
I: Why does it work?
S: By discarding.
I: What is discarded?
S: The pages I am reading, I am reducing the set.

The analysis of the responses to exercise 3 is included in the next chapter.

4.2.3 Summary of main results

The mental structures constructed by the students through the work of the first interview and the collective class allow them to solve problems presenting differences and similarities with respect to the problems solved before. At the same time, these structures are transformed into more advanced conceptual knowledge of recursive algorithms by means of equilibrating the differences and similarities presented by the new problems.

The structure of the trees of the first problem is similar to the one of the language problem in the sense that it is defined by inductive rules and it is different in the sense that it is a binary structure. The method to be defined is similar because it is a calculation over the structure.

The structure of the novel of the second problem is similar to the dictionary in the sense that it is a list of words and it is different in the sense that it is not alphabetically ordered. The method is in both cases a search.

Commonly the students try to generalize the constructed knowledge to the new situations without transforming it and attempting to change the situation; in other cases, the differences act as an obstacle generating in the students the feeling that it is not possible to solve the new problem. The conceptualization is consolidated by successing in equilibrating both factors and integrating them into new and more flexible structures, allowing the students to derive correct recursive formulations of their solutions to problems.

For the case of the exercise of defining a method of calculating the numbers of red lights of any tree of the given inductively defined set, the solution of the students can be synthesized as follows:

```
count-R (t) = if t = R then 1 else
               if t = B then 0 else
                if t = a1Ra2 then 1 + count-R (a1) + count-R (a2)
                             else count-R (a1) + count-R (a2)
```

Remarkable facts are on the one hand, that all students begin their descriptions of the method specifying the clauses for the base cases and on the other hand, that they correctly use a name for the method.

By adding the corresponding signature, this description becomes a recursive function definition for the solution method. The process of putting into correspondence students' recursive descriptions with formal recursive function definitions is presented in the next chapters. The construction of the concept of recursive algorithms as a school subject is attained through the conceptualization of such a correspondence.

Chapter 5

Instructional Proposal, Part 1

5.1 Introduction

The exercise 3 handed out to the students at the collective class is as follows:
Exercise 3
1) How can the searching algorithm be modified to take into account the case in which the searched word is not in the dictionary?
2) Given a dictionary and a word that is not in it, define a method of including the word in the dictionary at the right place.

In the case of the first question, students' responses reveal that they are confused about what would happen is the word is not in the dictionary. The second question is answered by most of the students saying that they would use the searching method (previously used and defined) and "at the end, the list has two words and the word to be inserted goes between those two". All students are absolutely confident about the correctness of their answers.

However, this solution is not formally correct and a way of correcting the errors is to attempt to formalize the solution. A collective class was designed and conducted[1], in which every student received a sheet including a synthesis of his/her own answers to the questions to exercise 3, and a worksheet with new exercises and questions. The goal is to encourage them to correct the errors in their solutions and formalize their conclusions as well.

Each problem is collectively discussed and then each student has to write down his answers in the worksheet. Explanations are given only when absolutely necessary.

5.1.1 Collective class

Since the first question of exercise 3 revealed that the students are confused about what happen if the word is not in the dictionary, they are encouraged to answer extra questions. They are as follows:
0. Recall the definition of the searching method below. What happen if

[1]This class took place one year after the first interview. Only four students from the original group of ten participated. From the remaining students, two had emigrated and four said that they would participate but never did.

the word is not in the dictionary?
1. What would a robot do to search (moon, [cat, day, house, sun]) for instance?
2. How does this method work in real life?

```
search (word, [Wf ... Wl]) =




                 let word' = chooseword ([Wf ... Wl])
                 if word = word' then return the-meaning-of-word
                 else if word < word'
                          then search (word, [Wf ... word']
                       else search (word, [word' ... Wl])
```

It was found that the students do not follow the definition step by step in the case of absence of the word. Only one student does so and his answer is the unique correct: "the robot will continuously search". Two of the students say that nothing is returned and the other one gives a surprising answer: he says that the robot "returns the word that is closer to the searched one". This is a case of α behavior, that is to say, the student changes the "reality" to fit his expectations. In this case the behavior is probably due to preconceived ideas: perhaps the student associates this case with some mathematical problem in which the solution is an "approximation" to some value. Another preconceived idea is that a process has to return a value or "nothing", but not to be "undefined", that is to say, never ending. The answers are discussed and all students agree that the correct one is that the robot will continuously search.
All students recognize the need of adding a termination condition for the case of absence of the word to get the method work as in real life and all of them do it correctly.

After those reflections the first question of the exercise 3 is posed, that is,
How should the definition above be modified in order to take into account the case in which the word is not in the dictionary?
The responses are synthesized saying that "if we arrive at a list with just two words none of which is the searched one, the search must be stopped and the message has to be 'the word is not there'". All students agree with this synthesis and they are encouraged to modify the definition of search (in the blank spaces of the definition above), to care about the case of the word not being in the dictionary *according to their descriptions.* They answer questions 3, 4 and 5 of the worksheet, namely:

3. Which are the possible results of search, depending the word is or is not in the dictionary?

4. Is there other possible result?

5. Which is the type of the result?

Question number 6 of the worksheet is
6. What does this mean mathematically?
It is the only one that has to be explained introducing the meaning of the expression **search(word, dic)**, that is always a **message**, either the constant message "the word is not in the dictionary" or a variable one corresponding to the meaning of the founded word.

5.1.2 Inserting a word in the dictionary

For the question
How to insert a word in a dictionary?
students' answers are synthesized saying that to insert a word in a dictionary, the searching method is applied and "the list is reduced to two words none of which is the searched one, then we insert the word between those two words of the list."
All students agree and are required to answer questions 7 and 8 below according to this synthesis.
7. Write below the definition of the method of insertion a word in a dictionary (the word is not there)
insert (word, [Wf ... Wl]) = ...

8. Use your definition to answer:

1. *insert (moon, [cat, day, house, sun]) =*
2. *Why do we get this result?*
3. *What kind of result should we get?*

Some of students' responses follow (relevant parts are indicated in italics):
Student 1
Question7:
insert(word, [Wf ... Wl]) = search (word, [Wf ... Wl])
when the message is "the word is not there", insert word between Wf and Wl *of the last list used in searching.*

Question 8.1:
insert(moon, [cat, day, house, sun]) = search (moon, [cat, day, house, sun]) = "the word is not there" ⇒ *(if I know the last list)* then return [house, moon, sun] else return [cat, moon, sun].

Student 2
Question 7:
insert(word, [Wf ... Wl]) = search (word, [Wf ... Wl]) ⇒ *new list* = [Wf' ... Wl'] ⇒ Wf' < word < Wl'

Question 8.1:
insert(moon, [cat, day, house, sun]) = search (moon, [cat, day, house, sun]) = "the word is not there" ⇒ *new list* = [house, sun] ⇒ house < moon < sun ⇒ insert(moon, [cat, day, house, sun]) = [cat, day, house, moon, sun].

Student 3
Question7:
insert(word, [Wf ... Wl]) = search (word, [Wf ... Wl])
length ([Wf ... Wl]) = 2 $\Rightarrow$ insert word | Wf < word < Wl $\Rightarrow$ length ([Wf ... Wl]) = 3.

Question 8.1:
insert(moon, [cat, day, house, sun]) = search (moon, [cat, day, house, sun]) = [house,sun] $\Rightarrow$ "the word is not there" insert moon | house < moon < sun = [house, moon, sun].

The first two students reveal a correct conceptualization of the problem: they correctly use the name of the methods, while the third student does not (see that she writes search (moon, [cat, day, house, sun]) = [house, sun], although immediately before it has been stated that the expression search (moon, [cat, day, house, sun]) is *a message*). α behavior has appeared again in the response of student 2 to question 8.1 where he returned the whole list to fit the correct answer.

Subquestions 2 and 3 of question 8 generate a fruitful discussion because all students refer to the modified list (in italics in students' responses) and *all of them suspect that something is wrong.* However, they are apparently satisfied with the obtained result (a partial list) because they attribute the error to something else and not to their own definition.

To help them to find the error, the students are recalled that *they have previously indicated* that the type of the result of searching a word in a dictionary (a list) is a message and *not* the modified list (see questions 5 and 6). In this way, the insertion method never accesses to the list modified by the searching method because it is not part of its result. Moreover, if the modified list were part of the result of the searching, it is not anyway the whole dictionary plus the new word as required. The importance of considering the result *that has to be constructed* is reinforced by emphzising that the goal of inserting a word in the dictionary is to obtain the whole dictionary, including the new word.

5.1.3 Another method

During the interview about the question of inserting a new word in the dictionary[2], one student (Sergio) has proposed another method, namely, sequential insertion, by which the new word is compared with the first, the second, etc, until finding one that is greater and placed before it. Questions 9.1 and 9.2 of the worksheet deal with this as described below:

Question 9:

"I compare it with the first one and if it is greater, I compare it with the second and if it is greater with the third, and so on until it is less and I put it in front or ..."

1. Replace the ... by an answer:

[2] Recall that this exercise was handed out in the collective class after the first interview. The students were asked about their solutions in the third individual interview after that class.

2. Define this method according to the description above. We use the following informal notation:

   Wf - the first word in a list of words

   Wl - the last word in a list of words

   Wnxt - the words between the first word and the last word

   Wf[Wnxt ... Wl] - the list whose first word is Wf and its tail is [Wnxt ... Wl]

   [] - the empty list

3. Given the word moon and the dictionary [cat, day, house, sun], how do you get the dictionary including the word moon in the right place?

These are relevant questions of the activity, because it is expected that the students have conceptualized the recursive method and are able to formalize it. Question 9.1 would help them to assimilate the need of the termination condition, and question 9.2 would evidence how far they can go in the formalization stage. Observe that no hint is given, in order to force the students to think about the complete definition (that is to say, if one writes "insert(word, Wf[Wnxt ... Wl]) =" the students miss something important (put the name, start the definition)).
All the students answer well question 9.1 revealing understanding of the base case, a fact already confirmed in the tree problem of exercise 1 (see section 4.2.1).
Responses to question 9.2, are analized considering the following factors:

1. the student uses a name ("insert" or something similar) to refer to the algorithm in both sides of the = symbol,
2. the arguments are correctly manipulated, that is to say, they are updated in each application of the method.
3. the result is well constructed regarding if Wf is forgotten or not.
4. the base case is present.

Student 1
He writes:
"insert(word, Wf[Wnxt...Wl]) = if word < Wf, it goes in front of the list else insert(word, Wnxt[Wnxt2...Wl]) and call Wnxt Wf, Wnxt2 Wnxt.
If insert(word,[]) then the word goes at the end."

Observe that he writes the name of the method in both sides of = (factor 1), the arguments are updated (factor 2) and the base case is present (factor 4). Although the result is not formally well constructed, his conceptualization is correct (Wf is taken into account, that is, factor 3 is present).

Student 2
Uses the name of the algorithm only in the left hand side of = and in the right hand side he writes the specification in Spanish using recursion as in previous activities (that is saying "the process is repeated") and he forgets Wf.
None of the factors above are completely present.

Student 3 and 4
They write:
"insert(word, Wf[Wnxt...Wl]) = if the list is empty put the word at the end else if word < Wf put it in front else Wf[... word ...].

They use the name only in the left hand side of = and in the right hand side, they concentrate in *the construction of the result*, putting the first word (factor 3). The base case is present (factor 4).

The following facts are observed: students who concentrate in some aspects of the problem, do not consider other ones and the solution is not completely correct. That means that there is an insufficient coordination between all involved aspects. On the other hand, the students are aware of the fact that their solutions do not construct the result appropriately.

In order to attain a complete coordination the students are encouraged to find themselves the correct answers by applying their definitions to a particular case (last part of question 9) and reflect about what has to be done to construct the right result. As expected, all students have a correct conceptualization of the recursive method of sequential insertion, while its formalization is difficult. For instance, the solution that students 3 and 4 above have presented expresses the result as *Wf[... word ...]* for the case in which *word* (the word to be inserted) is greater than the first word. Students' conceptualization is correct, as the following extract from the interaction developed in class shows:

Q: What does *Wf[... word ...]* mean?
R: That the word has to be inserted in the list after the first word.
(Observe that this description refers to the application of the insertion method to the tail of the list, which characterizes this type of recursion. The structure of the list is correctly used by the students.)
Q: The list after the first word is indicated by *[Wnxt ... Wl]*, is it not? Why don't you use that?
R: ... *Wf* .... (They focus in that the first word has to be part of the result.)
Q: You have to insert *word* in *[Wnxt ... Wl]*, how do you indicate that *word* has to be inserted in a list? (Observe that they have already written an instance of that in the left hand side of the equal sign.)
R: *insert(word, [Wnxt ... Wl])*.
Q: Ok. Which is then the result for the case that we are considering?
R: ...
Q: You have said that your expression *[... word ...]* actually is *insert(word, [Wnxt ... Wl])*, haven't you? Then, make the corresponding substitution in your solution, please.

The final solution of the students can be synthesized as follows:

```
insert(word, Wf[Wnxt ... Wl]) =
               if Wf[Wnxt ... Wl] = [] then [word]
                else if word < Wf then word Wf[Wnxt ... Wl]
                      else Wf insert(word, [Wnxt ... Wl])
```

To verify the correctness of their recursive specification the students are asked to write down the instances of the method by which the word moon is inserted in the list [cat, day, house, sun], that is:

```
insert(moon, [cat, day, house, sun]
                        -> cat insert(moon, [day,house, sun])
                        -> cat day insert(moon, [house, sun])
                        -> cat day house insert(moon, [sun])
                        -> cat day house moon sun []
```

Observe that some details are informally treated, for instance that the obtained result is actually the list [cat, day, house, moon, sun]. In instruction, the lists and their operations are formally introduced, as described in the examples of Chapter 6.

5.1.4 Conclusions

While previous work focus on helping the students in conceptualizing the structure of the involved elements and the application of the method to it, in the activity described in this chapter the emphasis is on increasing students coordination of the operations and the construction of the result.

The stage of formalization plays a relevant role, forcing the students to rigorously describe what they mean. All of them finally get correct answers to the questions revealing satisfactory conceptualization of this type of recursive algorithms for the studied cases. Constructing the correspondence between that conceptualization and its formalization appears as a hard stage of the process. The passage from describing the solutions in natural language to a formalism similar to mathematics is gradually accomplished. Despite the obstacles the validity of the strategy is evidenced by the work developed during the activity. Specially encouraging are the results obtained facing the students with the challenge of correcting their own errors.

Regarding students behavior, they reveal a passive attitude in which, faced with errors, they seldom take the initiative of searching the causes themselves. They are not willing to review their methods with which they are satisfied, as if the mistakes have been produced by "something" extern to their reasoning. In many cases, α behavior is detected, by which the students "change the reality" to fit their (erroneous) thinking.

An explanation is that in traditional teaching the students are not used to verify their responses neither to ask themselves about the reasons of failure or success. They are trained to answer questions or tests posed by teachers and to wait for explanations if they face any difficulty. At the same time, students performance is evaluated through their responses to exams or tests, although the strong evidence against, on the one hand, the pedagogical model of "the teacher explains-the student listens" as a way of generating knowledge and on the other hand, the consideration of correct answers as guarantee of effective learning [Sch02], [Bat99].

A remarkable fact is that while in traditional teaching it is amply acknowledged that teachers' explanations have to be repeated year after year, because the students "forget" what they "have learned", in the collective class described in this chapter which took place *one year after students' first interviews*, the students had no problem working with the material. For instance, no difficulties arise about the formalization of the searching method, no repetition of previous work was necessary and the class developed as if the previous class had taken place the day before.

This activity is the final part of the empirical work. The examples remain to some extent attached to the specific cases, that is, the way leading to solve and formalize the general problems is not include. For instance, from searching a word in a dictionary, the general problem of finding an element in an ordered list can be introduced.

In the next chapter the instructional proposal derived from the research is described in detail through examples from a course of discrete mathematics -including implementations in functional programming- for high school mathematics teachers.

5.2 The material

In this section, the material handed out to the students for working in the class is described:

- a worksheet called "Complementary Activity" containing the questions aimed at helping the students to find and correct their errors from previous work.
- a worksheet called "Questions and Answers" containing students' answers to the exercise 3 and to extra questions and synthesis of those answers used to work during the activity.

5.2.1 Complementary Activity

Definition of the method of searching a word in a dictionary from previous activity.

```
search (word, [Wf ... Wl]) =



                 let word' = chooseword ([Wf ... Wl])
                 if word = word' then return the-meaning-of-word
                 else if word < word'
                          then search (word, [Wf ... word']
                       else search (word, [word' ... Wl])
```

1. What happen if the word is not in the dictionary?

2. What would the robot do to search (moon, [cat, day, house, sun]) for instance?

3. How does this method work in real life?

4. Which are the possible results of search, depending the word is or is not in the dictionary?

5. Is there other possible result?

6. Which is the type of the result?

7. What does this mean mathematically ?

8. Write the definition of the method of inserting a word in a dictionary (the word is not there), according to answer 3.

```
insert (word, [Wf ... Wl]) =
```

8. Use your definition to answer:

   1. insert (moon, [cat, day, house, sun]) =

   2. Why do we get this result?

3. What kind of result should we get?

Question 9:

"I compare it with the first one and if it is greater, I compare it with the second and if it is greater with the third, and so on until it is less and I put it in front or ..."

1. Replace the ... by an answer:

2. Define this method according to the description above. We use the following informal notation:

   Wf - the first word in a list of words

   Wl - the last word in a list of words

   Wnxt - the words between the first word and the last word

   Wf[Wnxt ... Wl] - the list whose first word is Wf and its tail is [Wnxt ... Wl]

   [] - the empty list

3. Given the word moon and the dictionary [cat, day, house, sun], how do you get the dictionary including the word moon in the right place?

4. Given the word moon and the dictionary [cat, day, house, sun], how do you get the dictionary including the word moon in the right place?

5.2.2 Questions and Answers

The exercise 3 handed out at the collective class pose two questions. Some of the answers of the students are selected and synthesized.

How should the definition of searching be modified in order to take into account the case in which the word is not in the dictionary?

Students' selected answers:

Answer 1:
St: Write a sentence that if word first and word last are next to one another, that is, if they do not have words in between, then there he can finish.

Answer 2:
St: If there are two consecutive words and with the two ones, one is greater and the other is less, then the word is not in the dictionary.

Answer 3:
St: If there is a greater word and a less word but it is never equal, then that it says that the word does not exist.

Answer 4:
St: Er... if a list remains formed by two words and none of the two is the one, it is not there.

The answers can be *synthesized* saying that if we arrive at a list with just two words none of which is the searched one, the search must be stopped and the message have to be "the word is not there".

Do you agree?.

We verify that a list has only two words with the expression "length ([Wf ... Wl]) = 2".

Modify the definition according to that description and answer the questions from 3 to 6 of Complementary Activity.

How to insert a word in a dictionary?

Students' selected answers:

Answer 1:

I: We are going to represent the dictionary like this (I write word1 < word2 < word3 < ... < wordn).

St: I choose any word, say word4 , if word4 is greater than my word, then I will insert my word from word1 to word4 .

I: And then?

St: Well, there I have another list and I choose another word, say word2 and I get less than, then I make a list again and I get word3 ... no, ... no (the problem he has is that he got to one word, but he corrects well).

St: I include word4 but I don't use it and I include word2 but I don't use it, then if word' < word3 , goes between word2 and word3 and if it greater, it goes between word3 and word4.

Answer 2:

St: Oh, I take any word and I see if it is > or < , and if the word I want to include is < than the one I chose, then that is the minor quota, then I discard all the rest and I take that from the last and take another word of that new list and see if it is > or < and if the one I take is > then I discard the previous ones and like this I go on shortening the list until I get to two and there, in between those two, the word is inserted.

Answer 3:

St: We use the searching method from the previous class and since the word is not there, a list with just two words will be formed and then we insert the word in between.

Answer 4:

St: Given the word, you act as if it were in the dictionary.

I: You can represent the dictionary. (He puts word1 < word2 , etc).

St: I choose one (word3), I define the relation with word' and we do it again, until it is between 2. That is, with word1, you get it greater and with word2 you get it less (he refers to the extremes of the list, in fact), then, the word goes there.

Answer 5:

St: Well, to include a word in the dictionary, the first step is to choose a word from the list at random, we call it word1 and if word1 is less than the lost word, a new list of word1 to word lost is defined and if it is greater, the list will be from word first to word1 and step 3 would to repeat it until the list is of a Wn to a Wn+1 and then the lost word would be between Wn and Wn+1 .

The answers can be *synthesized* saying that to insert a word in a dictionary, all the students apply the searching method in the following way: the list is reduced to two words none of which is the searched one, then we insert the word between those two words of the list.

Do you agree?

Do you think that there is some difference between answer number 3 and the others?
Answer questions 7 and 8 of Complementary Activity.

Extra questions

Extra questions, selected answers and synthesis:

What happen if the word is not in the dictionary?

Answer 1:
St: It's going to have a reduced list to two words, in which yours isn't going to be, it's going to be greater than one and less than the other.

Answer 2:
I: In this function we had defined, we had this condition (word=word') that helps when the word is in the dictionary. What happens if the word is not there?
St: I would finish without finding the word.
I: With the same definition?
St: There will be two words between which you can't choose any.
I: Look at what happens with word=word'.
St: I don't know, it isn't going to work.

Answer 3:
St: There is going to be a point in which a list will remain with extremes and empty (he means to only having the extremes), a list with 2 ... He is not going to know what to do .. the robot ... will have to choo se only the extreme words.

Answer 2 says that it is not possible to choose, answer 3 says that the robot would just choose the extremes. All the answers coincide in that the method does not work, but there are different opinions about what happens.
Answer questions 1 and 2 of Complementary Activity.

We want to obtain the modified dictionary

During the interviews another method without using the searching one arose.

Answer 1:
St: to insert a word in a dictionary, a word that is not in it, we have to respect the order of the alphabet. The first is to see with which letter it begins. When this letter is inserted, I have to see the following letter and so on until the word is inserted in the right place.

Answer 2:
St: I go comparing one by one.
I: From which one?
St: From the first one.
I: And when do you insert it?
St: I compare it with the first, if it is greater, I compare it with the second, and if

it is greater, with the third and like this until it is less than the one I put in front.

Answer 3:
St: Comparing.
I: Which ones?
St: The first one with the one I am looking for.
I: And if we suppose that it is less?
St: Word' is less?
I: Suppose.
St: Then it would go forwards.
I: And if you get it greater?
St: I would compare with the following ...
I: If you get it less?
St: I insert it there.
I: And if it is greater?
St: I would go on with the following.
I: Until?
St: Until finding it a place, until comparing it with all of them ... that is, until finding its place.

The answers can be *synthesized* saying that I compare my word with the first in the dictionary, if it is greater, I it compare with the second and if it is greater, I compare it with the third and so on until it is less and I out it in front or ..."
Answer question 9 of Complementary Activity.

Chapter 6

Instructional Proposal, Part 2

6.1 Introduction

The examples described in this proposal are based on selected problems from the current high school mathematics curriculum and have been used in a discrete mathematics course for high school mathematics teachers. The course is aimed at explaining the relationship between computer science and discrete mathematics through concepts common to both disciplines -logic, functions, algorithms, inductive definitions, etc,- contrasting the differences between solving problems using the approach described in this thesis and the traditional way.

The emphasis of the course is put in that students' own techniques of solving problems have to be considered as the starting point of introducing a concept and their informal solutions have to be transformed into algorithmic specifications in natural language. These specifications are put into correspondence with formal mathematical definitions and finally, these mathematical definitions are implemented in a programming language. All these stages occur in an interactive way and the effective participation of students is encouraged. It is worth remarking that in most of the cases, the students already know an algorithm to solve particular cases of the problem. For instance, they know how to find the gcd of two natural numbers using Euclid's algorithm or they know how to sort the elements of a given set according to an order relation. The student himself/herself construct conceptual knowledge from this instrumental "know how", with the help of the teacher. In the course these ideas are put in practice and the teachers are required to introduce a topic in their high school classes applying the approach.

This proposal derives from both the gathered experience of several editions of this course and the research described in this thesis. The examples are developed focusing on the relevant aspects related to the understanding of the concept of recursion learned in the research.

6.1.1 Counting Problems

Counting techniques are described in discrete mathematics books from the point of view of deriving formulas to get the number of different combinations of sets of objects. One of these cases is to count the number of ways of selecting m elements without replacement from a set of n elements.

This point is usually illustrated by building general observations from small problems. For example counting the number of words of three letters without repetitions from four letters a, b, c, d.

Whichever algorithm is used in class to solve the problem, it is traditionally considered a means of deriving a formula for the case and introducing the general calculation formula of k-permutations $P(n,k) = n \times (n-1) \times ... \times (n-k+1)$, where n is the number of elements of the set from which groups of k elements are selected. The traditional way of teaching counting techniques does not take into account the algorithms themselves as objects of study.

However, in developing the algorithm the students acquire a fundamental knowledge, called in our epistemological framework *knowledge in the plane of actions*, which is the source of both the conceptualization and the formalization of the algorithm.

The stage of conceptualizing

Conceptualizing means to transform the instrumental knowledge acquired by attempting to solve instances of problems into conceptual knowledge by which the employed methods, the reasons of success (or failure) and the constructed result are grasped. The coordination of all these involved factors allows students to develop formal solutions to general problems. This transformation implies individual mental re-constructions of students cognitive structures.

To induce this process, several questions about the algorithms are formulated focusing in what has been done to achieve a result. The students have to describe their answers in natural language as precisely as possible, likely generating new questions.

Regarding the problem stated above, the students commonly tend to join a letter with the pairs formed with the other ones without repetitions. After several attempts their actions can be summarized as:

I) take the first letter in the set and *add it to each of the pairs of letters without repetition not containing the first letter*,
II) take the second letter in the set and *do the same* as in I) but adding it to the pairs without repetition not containing the second letter,
III) do this until there are *no more letters in the set.*

Observe that the students develop a recursive solution to the problem, arising from their way of thinking of a set as a sequence: they refer to "the first letter, the second, etc" of the set, although there is no order in a set. This structure allows them to specify points II and III above, in which lies the basis of recursion. Algorithmically speaking, sets are often treated as sequences and actually, in our examples, we use sequences with no repetitions to represent sets *because of students' thinking.*

Marked in italics in point I above is the following subproblem: given a letter and a sequence of pairs, select from the latter those pairs not containing the given letter, and add it to them. What is marked in italics in points II and III is the subproblem above with respect to each of the letters of the sequence.

The most important fact here is that the students become aware that these subproblems *are already present in their descriptions*, which naturally permits the following refinement:

1. take the first letter x in the sequence A,
2. select from the sequence of pairs without repetition the words that do not contain x,
3. add x to those words,

Reflecting about what has to be done after working out the first letter in the sequence A, the students strive to:

4. do the same with the remaining letters of A until no letters remain.

Observe that this description is correct but incomplete, because it is centered in the employed method and nothing is said about how to construct the result. To transform this instrumental knowledge into conceptual knowledge, the students are encouraged to think about *the result* and how it is constructed. They have to reflect on questions such as "Why do you think that your method works? What do you do when no letters remain? What kind of object is the solution? How can it be constructed?" This interaction between the usage of the method and the construction of the result is a crucial aspect of understanding recursion, because the final result is constructed from partial results. The operation taking these partial results and returning the final one *has to be conceptualized as part of the definition of the method*. In students' descriptions this operation is still not completely clear and answering the questions above helps students to become aware of the need of including one more point about the result:

5. construct the sequence of all obtained words (the new solution).

Observe that students are at first not aware of the importance of point 5, because what matters in mathematics while solving this problem is to count the words and not to construct the sequence of all of them. This approach on the contrary, consists in constructing the sequence of words and using the computer to count the number of its elements.

The stage of formalizing

The meaning of formalizing is to put into correspondence mental constructions (concepts) with some universal system of symbols[Pia75]. Traditionally, in computer science teaching algorithms are described using different formalisms (pseudo-codes, diagrams, flow-charts). But these formalisms are not universal systems of symbols, while mathematics is. Mathematical objects representing computer science concepts can be implemented in almost any programming language. Most importantly, the mathematical methods can be used to prove properties about them, for instance, correctness. Actually, one of the goals of functional programming has been to bring programming closer to mathematics [Bir98]. This approach takes a step in the opposite direction, getting mathematics closer to programming. This implies, among many things, to adopt a notation suitable for describing computer science concepts in mathematical language. Functional programming denotation is adopted and the students are introduced to concepts such as lists and their basic operations (map, filter, ++, etc), and application and composition of functions as well. A description of terms and expressions used in the formalization of the algorithms follows:

Sets

Seq(A) denotes the inductively defined set of all sequences of elements taken from a set A.
Seq$^+$*(A)* denotes the set of non empty sequences of elements of a set A.
N denotes the inductively defined set of natural numbers.
N^+ denotes the natural numbers ≥ 1.
Bool denotes the set of Booleans values, true and false.
- and $\cap$ stand for set difference and set intersection respectively.
$\#X$ stands for the number of elements of a set X.

Sequences

The symbol [ ] denotes the empty sequence and $[x]$ denotes a sequence whose unique element is x.
cons stands for the constructor function for sequences, that is to say, *cons(x,s)*, (with $x \in A$ and $s \in Seq(A)$), denotes a non empty sequence.
head denotes the function taking a non empty sequence and returning its first element.
tail denotes the function taking a non empty sequence and returning the sequence without its first element.
++ denotes the function (used as infix operator) taking two sequences s_1 and s_2 and returning a sequence with the elements of s_1 followed by the elements of s_2.
map denotes the function taking a function $f : A \rightarrow B$ and a sequence $s \in Seq(A)$ and returning a sequence $\in Seq(B)$ formed by the applications of f to the elements of s.
filter denotes the function taking a predicate $p : A \rightarrow Bool$ and a sequence $s \in Seq(A)$ and returning a sequence $\in Seq(A)$ with the elements of s satisfying the predicate p.
elem denotes the function taking an element x and a sequence s and returning the value of $x \in s$.
length denotes the function that given a sequence returns its number of elements.
[1 .. n] stands for the sequence of consecutive natural numbers from 1 to n, $n \geq 1$.

The symbol $=$ stands for testing equality and the symbol $\stackrel{\text{def}}{=}$ stands for introducing definitions.
High order functions are defined and used as in functional programming. For instance, *(elem x)* is a function taking a sequence y and returning the value of $x \in y$.

Formalizing the algorithm

Recall the specification of the algorithm given at the end of the section about conceptualization, composed using 5 steps. Steps two and three can be represented by the following function g, that given an element x and a sequence ss of sequences, adds x to all the sequences of ss not containing x:

g: $A \times Seq(Seq(A)) \rightarrow Seq(Seq(A))$

$$g(x,ss) \stackrel{\text{def}}{=} \underbrace{map\ (add\ x)}_{3}\,\underbrace{(filter\ (not \circ (elem\ x))\ ss)}_{2}$$

$$where\ add\ x\ y \stackrel{\text{def}}{=} y \mathbin{+\!\!+} [x]$$

Steps 4 and 5 of the algorithm above describe the recursive solution. Students are encouraged to derive a formal meaning of expressions like "do the same", "the remaining letters of" and "no letters remain" from their previous responses, using the notation above. First of all, they are induced to determine over what kind of objects the actions are done, that is, over what "do the same" is performed. In this way, students establish the well founded relationship between pairs -formed by a sequence and a sequence of sequences- (s_1, ss_2) and (s_1', ss_2) such that they belong to the relationship if and only if s_1' is the tail of s_1 and the least element is $([\,], ss_2)$. Secondly, students are induced to reflect about that their expressions "Do" and "Do the same" indicate that the same method has to be applied to different pairs. This is represented by the function f below which returns the constructed result from applying g to the elements of the relationship, until the least one is reached. The result is constructed using the operation $\mathbin{+\!\!+}$. The definition of f completes the formalization of steps 1, 4 and 5 of the algorithm:

$$f : Seq(A) \times Seq(Seq(A)) \rightarrow Seq(Seq(A))$$

$$f(s_1, ss_2) \stackrel{\text{def}}{=} \begin{cases} [\,] & ;\quad if\ \underbrace{s_1 = [\,]}_{4} \\ g(\underbrace{x}_{1}, ss_2) \underbrace{\mathbin{+\!\!+}}_{5} \underbrace{f(tail(s_1), ss_2)}_{4} & ;\quad otherwise \end{cases}$$

$$where\ x \stackrel{\text{def}}{=} head(s_1)$$

The first element in the pair is the given sequence, what about the second one? The students are asked about what they have already stated in their descriptions of the solution of the problem. Reasoning in this way, they discover that "the pairs (words of two letters) without repetition" present in their descriptions is the solution of a *previous instance of the same problem.* Another formulation of the sentence is: *if the sequence of pairs without repetition is given, say subs(s,2), then the solution for the triad, that is subs(s,3), is f(s,subs(s,2))*, that is to say, f above is applied to the sequence and the previous solution. The solution of the general problem is derived from that and formalized by function *subs* below:

$$subs : (Seq^{+}(A)) \times (N^{+}) \rightarrow Seq(Seq(A))$$

$$subs(s,k) \stackrel{\text{def}}{=} \begin{cases} [[x] \mid x \in s] & ;\quad if\ k = 1 \\ f(s, subs(s, k-1)) & ;\quad otherwise \end{cases}$$

That is, function *subs* takes a non-empty sequence s and a natural number $k \neq 0$ and returns the result of applying f to s and the previous solution for $k - 1$.

Applying *subs* to *[1..n]* and k, a sequence of sequences of k elements without repetition formed from n elements, is computed. So, the function that computes the number $P(k, n)$ of k-permutations is defined by:

$$perm : N^+ \times N^+ \rightarrow N$$
$$perm\ (k,n) \stackrel{\text{def}}{=} length\ (subs([1..n],k))$$

Once the functions are defined and implemented, they are used to solve other instances of the problem and experiment with other types of problems as well.

It is worthwhile remarking that, on the one hand, the students know how to solve the problem and they do it using a recursive method, and on the other hand, the concepts to be formalized and generalized are present in their descriptions. The proposal derived from the research implies to guide the process of transformation of this knowledge into concepts.

6.1.2 Sorting

Sorting a set of elements according to some order relation is a common problem in real life. Most people have, at one time or another, solved this problem for some particular case (sorting objects according to their size, children to their heights, pictures to their colors, sorting words lexicographically, numbers in increasing order, etc). The students are asked to solve some instances of the problem which they do using a recursive algorithm:
Given the following sets, construct sets with their elements sorted according to each of the indicated order relationships.

{3,2,-1,0,4,-3} and $\leq$
{b,f,d,y,z,a,m} and the lexicographic order
{{a,b,c},{},{b},{a,b}} and $\subseteq$
{{0,1,2},{a},{True, False}} and #X $\leq$ #Y

Then, they are induced to reflect upon their methods and to abstract an algorithm for the general problem. Basically, one algorithm described by the students is:

1. take the least element in the set and put it first,
2. take the least element in the remaining set and put it next,
3. do "the same" with the remaining elements of the set, until there are no more elements in it,
4. construct the result as a sequence.

In this description, "do the same" refers to applying the same method, which has not been formally defined yet nor assigned a name. Observe that this sequence of actions can be formalized as an algorithm admitting both iterative and recursive formulations. Since the focus of this work is on the learning of recursion the second one is developed.

Point 1 states a subproblem: given a finite and not empty set X and an order relation R, obtain the least element of the set, that is, the element x of X such that, forall elements y in X, x R y holds. To solve this problem the students define a function, (say *ordR*), taking a non-empty sequence X and an order relation R and returning the head of a sequence whose unique element is the "least" element of X according to R. This question poses the problem of representing a binary order relation so that it can be an argument of a function. It is represented by a function : (A x A) $\rightarrow$ Bool taking a pair of elements and returning true if the pair belongs to the relation, false otherwise.

ordR is defined as follows[1]:

$$ordR : (Seq^{+}(A)) \times ((A \times A) \rightarrow Bool) \rightarrow A$$
$$ordR\ (X,R) \stackrel{\text{def}}{=} head\ [x :\ x \in X \mid \forall y \in X,\ x\ R\ y]$$

Points 2, 3 and 4 of the description of the algorithm contains the relevant aspects of recursion. As in the previous example, the students are induced to determine the structure over which "do the same" is applied. In this case, the well founded relationship is between the pairs (X,R) and (X',R) where X' is the set X without the least element according to R, ending in the pair ([ ],R). The result is constructed by the function *cons*.

From the algorithm described by the students, the following definition of sort is derived, in which the steps, are indicated by a corresponding number below each subexpression.

$$sort : Seq(A) \times (A \times A \rightarrow Bool) \rightarrow Seq(A)$$

$$sort(X,R) \stackrel{\text{def}}{=} \begin{cases} [\,] & ; \quad if\ \underbrace{X = [\,]}_{3} \\ \underbrace{cons}_{4}(\underbrace{x}_{1}, \underbrace{sort}_{3}\underbrace{(X - [x], R)}_{2} & ; \quad otherwise \end{cases}$$

$$where\ x \stackrel{\text{def}}{=} ordR(X,R)$$

Once implemented, the algorithm is applied to each of the instances above.

6.1.3 Summary of the main points

In both examples the students have to recognize subproblems and define functions representing their solutions for them. "What has to be done on each element" is represented sometimes by functions admitting non recursive definitions. In the case of the counting problem is the sequence of actions represented by the function g and in the case of sorting is the way by which the least element of each of the set is obtained. The recursive method applies these functions for each element *belonging to a certain structure*, which is the cause of both the way in which the method works and its success. One of the main points to be understood is that it is the relationship between any element and its previous one *inserted* in a structure that guarantees the success of the method, because the structure has an initial element in which the process ends. Formalizing the concept, which means to formally define the algorithm *and* to apply it to particular cases, helps in understanding on the one hand that what students "do" and "do the same" are expressions which can

[1]recall that we use sequences without repetitions as sets

be referred by the same name and on the other hand that the result is constructed combining the solutions of the partial results and the solution of the subproblems.

6.1.4 Euclid's Algorithm

In this example, functions like *mod*, *div*, *division* and *divisors* commonly used in programming languages are introduced as mathematical functions. Observe that concepts such as the quotient and the remainder of a division are widely used in mathematics courses, and the students know how to compute each for given values but have no idea about the general algorithm they apply due to the fact that these concepts are seldom introduced *as functions*. However, deriving mathematical definitions for these functions from the algorithms developed by the students is an illustrative example of pedagogically introducing recursion and hence it was presented in the course for mathematics teachers.

The problem is to define functions *div* and *mod* calculating the quotient and the remainder of the division of two natural numbers respectively.

Several attempts and collective discussion allow us to arrive at a satisfactory description of the algorithms in natural language which are described here after the mathematical definitions for clearity reasons. The mathematical definitions are as follows:

$$div : N \times N^{+} \rightarrow N$$

$$div(n,m) \stackrel{\text{def}}{=} \begin{cases} 0 & ; \quad if\ n < m \\ div(n-m,m)+1 & ; \quad otherwise \end{cases}$$

$$mod : N \times N^{+} \rightarrow N$$

$$mod(n,m) \stackrel{\text{def}}{=} \begin{cases} n & ; \quad if\ n < m \\ mod(n-m,m) & ; \quad otherwise \end{cases}$$

$$division : N \times N^{+} \rightarrow N \times N$$
$$division\ (n,\ m) \stackrel{\text{def}}{=} (div(n,m),\ mod(n,m))$$

In solving particular cases the students are induced to reflect about *what they do and why*. They become aware that if n is to be divided by m, $(n > m)$ a sequence of groups of m elements taken from the n ones is constructed and every time such a group is formed, one is counted and the same method is applied to the new group. When no group can be formed any more, the questions to be answered are such as *why not?* and *what has to be done in this case?* Other questions such as *what does "m elements taken from n ones?" mean?* and *what does "count one" mean?* arise and the students are encouraged to find mathematical expressions for their answers. In the interaction between the conceptualization described in natural language, the formalization in mathematical notation and the application of the method to particular cases, the recursive definitions above are constructed by the students themselves.

Euclid's algorithm provides a suitable opportunity for introducing what is called "the negation of the concept", meaning a non-recursive definition of the same function to be compared with the recursive one, contributing to constructively generalize the concept [Pia78]. Firstly, the following recursive definition of the function for Euclides' algorithm is stated:

$$gcd : Z \times Z \rightarrow Z$$

$$gcd(a,b) \stackrel{\text{def}}{=} \begin{cases} b & ; \quad \textit{if } a \bmod b = 0 \\ gcd(b, a \bmod b) & ; \quad \textit{otherwise} \end{cases}$$

The importance of the base case and the application of the function to a structure of pairs of natural numbers holding a well founded relationship are discussed.

A non recursive definition can easily be derived from the the meaning of "the greatest common divisor of two numbers", in order to contrast both types of definitions of the same function and state the need of proving the equivalence between them.

Defining an instance of $ordR$ where the order relation is "x is greater or equal than y", the maximum value of a set of numbers can be obtained. The new definition of gcd, say gcd', is straightforward:

$$\begin{array}{l} gcd'\!: Z \times Z \rightarrow Z \\ gcd'\ (a,b) \stackrel{\text{def}}{=} max\ (divisors(a) \cap divisors(b)) \\ where \\ divisors : N \rightarrow Seq(N) \\ divisors\ (n) \stackrel{\text{def}}{=} [x : x \in [1 \,..\, n] \wedge n \bmod x = 0] \end{array}$$

Having two definitions of the function that computes the greatest common divisor of two integers, the need for arguing the behavioral equivalence of two definitions naturally arises, that is to say, the proof of equivalence:

$$\begin{array}{l} \forall\, a,\ b\ \in\ Z,\ gcd(a,b)\ =\ gcd'\,(a,b), \text{ that is to say:} \\ \forall\, a,\ b\ \in\ Z,\ gcd(a,b)\ =\ max(divisors(a) \cap divisors(b)) \end{array}$$

Proofs are presented in the usual way.

The stage of implementing

The implementation of the mathematics definitions in a programming language is constructed completing the stage of formalizing. Although functional programming languages are the most suitable for our purpose, languages from other programming paradigms can be used.

The functional programming language Haskell[2] has been used in discrete mathematics courses of computer science University studies [dRC98], while Isetl [3] has been used in discrete mathematics courses for high school teachers [dR02]. Discussing

[2] www.haskell.org

[3] http://isetlw.muc.edu/isetlw/

these alternatives is beyond the context of this thesis, although it is worthwhile mentioning some features:
Isetl has been designed for the purpose of using it in the learning of mathematics, so its syntax is similar to mathematics. For instance, sets are primitive objects and their operations are predefined. It is not a typed language, it is not functional and it does not allow pattern matching.
On the other hand, Haskell is a functional programming language of general purpose, it is a strong typed language, equational definitions of functions using pattern matching are allowed, as well as users defined algebraic types. The implementation of recursive function definitions in Haskell is straightforward, as is shown in the following.

Some of the predefined functions in Haskell are implemented by students (for instance, *forall*, *gcd*), others are not (for instance *div* and *mod*).

Sets functions:

```
--intersection of two sequences
inter :: Eq a => [a] -> [a] -> [a]
inter [] _ = []
inter _ [] = []
inter (x:xs) ys = case (elem x ys) of
                     True  -> x : inter xs ys
                     False -> inter xs ys

-- removing a given element from a sequence
remove :: Eq a => a -> [a] -> [a]
remove x [] = []
remove x (y:ys) = if x == y then remove x ys else y:remove x ys

-- checking if all elements of a sequence hold a given property
forall :: (a -> Bool) -> [a] -> Bool
forall p [] = True
forall p (x:xs) = (p x) && (forall p xs)
```

k-permutations example:

```
g :: Eq a => (a ,[[a]]) -> [[a]]
g (x,xss) = map (add x) (filter (not.(elem x)) xss)
              where add x xs = xs ++ [x]

f :: Eq a => [a] -> [[a]] -> [[a]]
f [] l1 = []
f (x:l) l1 = (g (x,l1)) ++ (f l l1)

subs :: Eq a => ([a], Int) -> [[a]]
subs (xs,1) = [[x] | x <- xs]
subs (xs,k) =  f xs (subs (xs,k-1))
```

```
perm :: (Int,Int) -> Int
perm (k,n) = length (subs([1..n],k))
```

Sorting example:

```
ordR :: [a] -> (a -> a -> Bool) -> a
ordR xs r = head [x | x <- xs, forall (r x) xs]

-- sort is predefined in Haskell, so sorti is used instead.
sorti :: Eq a => (a -> a -> Bool) -> [a] -> [a]
sorti r [] = []
sorti r xs = y : sorti r (remove y xs)
               where y = ordR xs r
```

Euclid's algorithm example:

```
-- divisors is defined for integers different from 0.
divisors :: Int -> [Int]
divisors 0 = error "infinite list"
divisors n = [x | x <- [1..abs(n)]++[-abs(n)..(-1)], mod n x == 0]

-- gcd is predefined in Haskell, so mcd is used instead.
mcd :: (Int,Int) -> Int
mcd (n,m)
    | mod n m == 0   = m
    | otherwise      = mcd (m,mod n m)

-- max is predefined in Haskell, so may is used instead.
may :: Ord a => [a] -> a
may (xs) = ordR xs maxi
            where maxi x y = x >= y

mcd' :: (Int,Int) -> Int
mcd' (a,b) = may(divisors(a) `inter` divisors(b))
```

After implementing the functions above, the students are encouraged to pose different types of questions and to investigate properties, as well. Some examples are: Which is the result of $perm(k,n)$ in which $k > n$? Which are the divisors of 0 according to the mathematical definition? Why the case of 0 returns an error message in the divisors program? These questions provide the opportunity for discussing the limitations of the computer. Why does Haskell use the $==$ symbol instead of $=$ like mathematics? The use in mathematics traditional teaching of the same symbol $=$ with different meanings (defining and testing equality) unconsciously confuses the students. A programming language avoids this type of confusion and provides the opportunity for discussing the semantics of languages.

Observe that in the stages we do not care about efficiency issues. Our goal is to induce thinking about the methods (algorithms) that the students make contact with in many problems of mathematics courses.

In this way, students are provided with a solid background to the learning of advanced topics in computer science, in particular those related to the concept of recursive algorithms. Implementations in other languages using iterative expressions can be introduced as other type of representation of mathematical objects, reinforcing the relationship between mathematics and computer science.

Chapter 7

Conclusions and Further Work

7.1 Summary of the research

The research work described in this thesis focuses on elaborating an instructional proposal aimed at providing computer science educators with a strategy for introducing the concept of recursive algorithms (primitive and course-of-values). The approach is founded on results from empirical work developed according to epistemological principles.

The epistemological principles are taken from Jean Piaget's theory and the empirical work is mainly an implementation of these to the learning of the concept combined with the experience acquired during many years of teaching discrete mathematics and functional programming courses in computer science University studies.

In particular, the theory gives detailed and satisfactory explanations about the source of elementary forms of reasoning by recurrence on natural numbers. On the other hand a general principle of the theory states that the mechanisms and general laws governing the construction of cognitive structures are the same in all stages of their evolution. From these principles the premise of this work is derived, stating that the approach of this research consists of extending the results of Piaget's theory about recurrence on natural numbers to the investigation of the construction of the concept of recursive algorithms on other structures.

A concept is an individual construction and the formalization is its representation in a universal system of symbols. The formalization is socially constructed and transmitted by educational systems. Learning a concept as a school subject means to individually construct a correspondence between the concept and its formalization.

The instructional proposal derived from this research consists in a strategy to integrate into the process of the learning of the concept of recursive algorithms both the construction of the concept and the construction of the correspondence to its formalization. In this way, the understanding of the formalization of the concept is dialectically achieved with the understanding of the concept.

The empirical knowledge from many years of teaching discrete mathematics and functional programming courses in computer science has led to the reflection on educational issues such that the role of mathematics education in computer science education. Serious difficulties to understand concepts such as high order functions, recursion, induction have been detected in students of computer science courses. On the other hand, those concepts are widely used by the students of mathematics courses: they work with derivatives, (high order functions), they use recursive algorithms (Euclid's Algorithm, arithmetic and polynomials operations) and they make proofs using mathematical induction (induction). This reflection and the fact that mathematics teachers have detected similar difficulties, have led to the conviction that one of the challenges that education currently faces is to succeed in integrating mathematics and computer science, by means of the assembled work of educators of both disciplines. In the last few years, the author of this thesis has participated in various activities in that direction[1], for instance, she has been teaching a discrete mathematics course for high school mathematics teachers using the programming language ISETL following the approach of this research. Some of the examples described in this thesis have been used in several editions of the course. Although a systematical evaluation of the proposal cannot be made from these activities, they allow us to draw some encouraging conclusions [dR02], [dR04].

The next two sections briefly describe the stages of the construction of the concept of recursive algorithms as a school subject that have been identified in the empirical work and the instructional proposal. The latter is developed in detail in Chapter 6.

7.1.1 The stages of the construction of the concept of recursive algorithms as a school subject

Three stages in the construction of the concept of recursive algorithms as a school subject has been identified:

- An *instrumental* stage in which the students construct knowledge interacting with an instance of the problem without consciousness about how the solution is achieved and why it works. This knowledge constitutes the source of conceptual knowledge. For instance, the individuals solve the problem of searching a word in a dictionary (an instance of searching an element in an ordered list) but have difficulties in describing the steps and are unaware of the reasons of the success.

- An stage of *conceptualization* with two sub-stages:
  1) the evolution from the instrumental knowledge to conceptual knowledge starts by the grasp of consciousness and the reflection about the actions involved in the developed method and about the reasons of the success (or failure). This reflection leads to an understanding of the relationship between the structure of the elements over which the method is applied, the method itself, and the obtained result. This process ends in the comprehension of the recursive algorithm. For instance, in the language problem of the first interview, the students derive the method of counting the a's of any word from reflecting both about the structure of words and the result to be obtained.
  2) On the other hand, once the particular algorithm is understood, students'

[1] http://imerl.fing.edu.uy/secutuniv/

reasoning attempts to generalize what has been successfully constructed to all the situations, which is the prototype of inductive generalization. Observe, for instance, that all students (except Sergio) apply binary search to insert a word in a dictionary. The mental construction of the algorithm has not yet attained the status of a concept. In this sub-stage of the conceptualization, the students face new instances of problems presenting variations and similarities with the old ones, causing a desequilibrium of their cognitive structures. The necessity of assimilate the novelties generates the progress of the grasp of consciousness and of the reflective abstraction, giving rise to constructive generalizations and a new equilibrium. In this process the mental constructions corresponding to the algorithms are transformed, combined and differentiated from other operations and integrated in a more complex cognitive system. Examples of this process are students' solutions to the tree problem and the search of a word in a novel described in Chapter 4.

- A stage of *formalization* which consists of constructing a correspondence between students' concept and a universal system of symbols, for instance, mathematics. This is the stage that transforms a concept into a school subject. The students become aware of ambiguities and/or errors in their specifications and correct them gradually approaching a formal definition of the solution of the problem. Examples are presented in Chapters 3 and 5 and in the proposal of Chapter 6 in which the formalization is completed with the implementation of the solutions in Haskell. The choice of mathematics to formalize the knowledge *before* the implementation is not casual: this approach aligns itself with the school encouraging formal mathematics in the learning of programming [Pro97] [BvW99] [Pag01] [Pag03] [BH02]. The implementation of the solutions in a programming language provides an experimental context in which the mathematical objects can be represented generating new sources of higher forms of knowledge. The role of the computer is clearly established as a powerful tool which calculates what the student has designed as a solution of a problem. The importance of developing skills in formally doing these designs (programs) is evident.

These stages act in a pro and retro-active manner, influencing the development of each other. For instance, the retro-active role of the stage of formalization has become evident in the problem of determining a method to insert a word in a dictionary using the searching method: attempting to formalize their solutions, all the students have understood that the modified lists of the searching method are not part of the returned result and that their solutions have to be improved (see the collective class described in Chapter 5). On the other hand, the interaction between defining the method and applying it to particular cases allows both to refine the definition and to construct the proper result. One of the most significative results of this research lies in that the students have themselves become aware of the need of defining the method for the base case in the language problem of Chapter 3, and in subsequent problems they have begun to define the methods specifying the base cases, as in the tree problem of Chapter 4. Observe that usually, the base cases are introduced with no regard of students thinking. In [HA02], the correct formulation of base cases is pointed out as one of the main difficulties in the learning of recursion.

7.1.2 Instructional Proposal

The analysis of the empirical work within the framework of Piaget's theory has allowed us to develop an instructional proposal for introducing the concept of recursive algorithms. The examples of Chapter 6 and the activities of Chapter 5 illustrate the strategy, summarized in the following steps.

1. Get students to solve instances of problems, preferably those that they know from day-to-day life or those that they have already faced in mathematics courses.

2. Get students to reflect on:

    - what they did, inducing them to specify their solutions in natural language using expressions such as "do" and "do the same" (or similar),
    - why their solutions work (or why not, in which case they are encouraged to correct the errors), making them aware of the well founded relationship between the elements over which the method is applied as the reason of the success,
    - the difference between the method (which is always the same) and its successive applications to arguments (that always vary). From that, the students are induced to think that "do" and "do the same" can be referred through a name, which helps in the design of the formal definition of the method.

3. Face the students with problems' instances presenting variations and similarities with respect to those already solved. Induce the students to solve the new problems taking into account both the differences and the analogies in order to derive the adequate transformations to be imposed to the operations.

4. Introduce formal terms and expressions corresponding to students' concepts in a pro and retro-active manner, that is, backing to students' descriptions of the solutions when necessary, approaching a formalization of the general solution for the general problem. The formalization corresponds as much as possible to students' specifications of the solutions, emphasizing that they have already solved the problem using an algorithm that can be formalized by a recursive function definition. In particular, the occurrence of the name of the function to both sides of the equal sign is the formal mathematics representation of students' expressions "do" and "do the same".

5. Get students to verify the correctness of the solutions or to correctly detected errors by applying the methods to new particular cases.

In all the points, *students descriptions are seriously taken into account.* For instance, in the formalization of an algorithm the students learn how their informal solutions can be expressed in mathematics and implemented in Haskell, as presented in Chapter 6.

7.1.3 Summary of main contributions and further work

This work specially contributes to the study of the evolution of the concept of recursion from the stage in which it is used as an instrument to solve problems or perform a task to the stage in which it becomes a school subject. The contributions specially relevant are:

- all the empirical investigations and the derived instructional strategy are theoretical justified,
- the concept of recursive algorithms is defined from the educational perspective as the cognitive structure constructed integrating sub-structures (other concepts) corresponding to:
  - the well founded relation between the elements over which the algorithm is applied,
  - the relationship between the successive instances of application of the algorithm to decreasing arguments,
  - the operations involved in the construction of the result, being one of them the method itself,
- the relationship between conceptual and formal knowledge is stated,
- as a result of the analysis of the data collected from individual interviews and collective classes, an instructional strategy to introduce the design of recursive algorithms for solving problems is proposed. The proposal has been partially applied with a small group of students and in a course of discrete mathematics for high school mathematics teachers, with encouraging results.

Further research is needed to investigate the impact of the instructional strategy in the effective learning of the concept of recursive algorithms. In this work some evidence is revealed because of students' understanding of the need of base cases, of the reference to a recursive algorithm using a name in different instances in the formal definition, and of the sequence of "smaller" arguments holding a property as the reason of the success. Students' descriptions of what they do to solve a recursive problem allow us to establish a correspondence with the main characteristic of recursive formulations of the solutions, which are derived by the students themselves. Examples of this are described in Chapter 3 (nr-of-a's), Chapter 4 (count-R), Chapter 5 (insert) and Chapter 6 (subs, sort, div, mod, gcd).

New interviews and experiments have to be conducted. It is not obvious how to prepare and conduct such activities, on the contrary, computer science education researchers need to be supported in the learning and training on these tasks. The collaboration of researchers from educational related disciplines is indispensable. Support is also needed in get students engaged in such experiments and projects. One of the main difficulties of this research has been to convince the students to seriously participate in the project.

Further investigation is required about, on the one hand, the learning of structures and their representations to take care of the difficulties discovered in this research, for instance about the role of the implication, and on the other hand, the learning of the method of proof by induction, based on both the structures and the recursive methods defined on it.

The investigation of the historical evolution of the concept of recursion from the perspective of the analysis developed by Jean Piaget and Rolando García in "Psychogenesis and the History of Sciences" will surely contribute a great deal to investigations on the learning and teaching of the concept. In effect, Piaget and García have stated a parallelism between the mechanisms involved in the psychogenesis of a concept and those of the stages of its historical development whose implications for pedagogy are unvaluable. In the case of the concept of recursion, one of the interesting subjects of investigation is the process of its early use, -from ancient times- to the formalization of the concept by Gödel in the 30's and the recently developed theories introducing the dual concepts of co-recursion and co-induction and their applications in programming.

Bibliography

[ABD$^+$00] Mark Asiala, Anne Brown, David DeVries, Ed Dubinsky, David Mathews, and Karen Thomas. A Framework for Research and Curriculum Development in Undergraduate Mathematics Education. 2000.

[Acz77] Peter Aczel. *Handbook of Mathematical Logic.* Studies in Logic and the Foundations of Mathematics, Vol 90. North-Holland Publishing Company, Amsterdam-NewYork-Oxford, 1977.

[BA96] Mordechai Ben-Ari. Recursion: From Drama to Program. 1996.

[BA01] Mordechai Ben-Ari. Constructivism in Computer Science Education. *Journal of Computers in Mathematics and Science Teaching, Vol. 20, Issue. 1, 2001, pp. 45-73*, 2001.

[Bat99] Michael Battista. How Do Children Learn Mathematics? Research and Reform in Mathematics Education. *Proceedings of the conference Curriculum Wars: Alternative Approaches to Reading and Mathematics. Harvard University*, 1999.

[BGM92] Shawkat Bhuiyan, Jim E. Greer, and Gordon I. McCalla. Supporting the Learning of Recursive Problem Solving. 1992.

[BH02] Doug Baldwin and Peter B. Henderson. The Importance of Mathematics to the Software Practitioner. *IEEE Software*, 2002.

[Bir98] R.S. Bird. *Introduction to Functional Programming using Haskell.* Series Editors C.A.R. Hoare and Richard Bird, Prentice Hall, 1998.

[Boo92] Shirley Booth. Learning to Program. A phenomenographic perspective. *Göteborg Studies in Educational Sciences, ISSN o436-1121*, 1992.

[Bur95] C.T.P. Burton. Conceptual Structures for Recursion. *Lecture Notes in Computer Science, vol. 1022, Springer-Verlag*, 1995.

[BvW99] Ralph-Johan Back and Joakim von Wright. Structured Derivations: a method for Doing High School Mathematics Carefully. *TUCS Technical Report No 246*, 1999.

[CD96] John Close and Darina Dicheva. Misconceptions in Recursion: Diagnostic Teaching. 1996.

[CM95] C. Clack and C. Meyers. The dys-functional student. *Lecture Notes in Computer Science, Nr.1022, Springer-Verlag*, 1995.

[Cut80] Nigel Cutland. *Computability: An Introduction to Recursive Function Theory.* Cambridge University Press, 1980.

[Dah02] Bettina Dahl. A Synthesis of Different Psychological Learning Theories?- Piaget and Vygotsky. 2002.

[DF96] Ed Dubinsky and William Fenton. *Introduction to Discrete Mathematics with Isetl.* Springer, 1996.

[DL86] Ed Dubinsky and G. Lewin. Reflective Abstraction and Mathematics Education: The Genetic Decomposition of Induction. *Advanced Mathematical Thinking*, 1986.

[Dor91] W. Dorfler. Forms and Means of Generalization in Mathematics. 1991.

[dR02] Sylvia da Rosa. The Role of Discrete Mathematics and Programming in Education. *Workshop: Functional and Declarative Programming in Education, technical report of University of Kiel*, 2002.

[dR04] Sylvia da Rosa. Designing Algorithms in High School Mathematics. *Lecture Notes in Computer Science, vol. 3294, Springer-Verlag*, 2004.

[dRC98] Sylvia da Rosa and Gustavo Cirigliano. Matemática y Programación. *Anales del Congreso Iberoamericano de Educación Superior en Computación*, 1998.

[GB96] J. Good and P. Brna. Novices Difficulties with recursion: Do Graphical Representations Hold the Solution? 1996.

[Geo00a] Carlisle E. George. EROSI - Visualising Recursion and Discovering New Errors. *ACM 1-58113-213-1/00/0003*, 2000.

[Geo00b] Carlisle E. George. Experiences with Novices: The importance of Graphical Representations in Supporting Mental Models. *In A.F.Blackwell and E.Bilotta (Eds). Proc. PPIG 12, pp 33-44*, 2000.

[GS99] Davit Ginat and Eyal Shifroni. Teaching Recursion in a Procedural Environment - How much should we emphasize the Computing Model? *ACM 1-58113-085-6/99/0003*, 1999.

[GSG03] Tina Gotschi, Ian Sanders, and Vashti Galpin. Mental Models of Recursion. *ACM 1-58113-648-X/03/0002*, 2003.

[HA02] Bruria Haberman and Haim Averbuch. The Case of Base Cases: Why are They so Difficult to Recognize? Student Difficulties with Recursion. *ACM 1-58113-499-1/02/0006*, 2002.

[HMG01] Christian Holmboe, Linda McIver, and Carlisle E. George. Research Agenda for Computer Science Education. *In G.Kadoda (Ed). Proc. PPIG 13, pp 207-223*, 2001.

[KA86] Claudius M. Kessler and John R. Anderson. Learning Flow of Control: Recursive and Iterative Procedures. *Human-Computer Interaction, 1986, Volume 2, pp 135-166, Lawrence Erlbaum Associates, Inc.*, 1986.

[Kah83] Hank Kahney. What do Novice Programmers Know about Recursion. *ACM 0-89791-121-0/23/012/0235*, 1983.

[Kle77] Stephen Kleene. *Handbook of Mathematical Logic.* Studies in Logic and the Foundations of Mathematics, Vol 90. North-Holland Publishing Company, Amsterdam-NewYork-Oxford, 1977.

[Kle81] Stephen Kleene. Origins of recursive function theory. *Annals of the History of Computing, Vol. 3, Number 1*, 1981.

[Ler88] Uri Leron. What Makes Recursion Hard? *Proceedings of the Sixth Interbational Congress of Mathematics Education (ICME6), 1988*, 1988.

[LL00] Dalit Levy and Tami Lapidot. Recursively Speaking: Analyzing Students' Discourse of Recursive Phenomena. *ACM 1-58113-213-1/00/0003*, 2000.

[LLP00] Dalit Levy, Tami Lapidot, and Tamar Paz. "It' just like the whole picture, but smaller": Expressions of gradualism, self-similarity, and other pre-conceptions while classifying recursive phenomena. 2000.

[LP98] Harry Lewis and Christos Papadimitriou. *Elements of the Theory of Computation.* Prentice Hall, (2nd. edition), 1998.

[LZ86] Uri Leron and Rina Zazkis. Computational Recursion and Mathematical Induction. *For the Learning of Mathematics, Vol. 6, No.2, pp 25-28*, 1986.

[Pag01] Rex L. Page. Formal Methods Education and Programming Effectiveness. *Project supported by NSF(ITR) Award 0082849 and by the University of Oklahoma*, 2001.

[Pag03] Rex L. Page. Software is Discrete Mathematics. *ICFP'03*, 2003.

[PB66] Jean Piaget and Evert Beth. *Mathematical Epistemology and Psychology.* D.Reidel Publishing Company, Dordrecht-Holland, 1966.

[PG80] Jean Piaget and Rolando Garcia. *Psychogenesis and the History of Sciences.* Columbia University Press, New York, 1980.

[Pia63] Jean Piaget. *La Formation des Raisonnements Recurrentiels.* Presses Universitaires de France, 1963.

[Pia64] Jean Piaget. *La Prise de Conscience.* Presses Universitaires de France, 1964.

[Pia75] Jean Piaget. *L'équilibration des Structures Cognitives, Problème Central du Développement.* Presses Universitaires de France, 1975.

[Pia78] Jean Piaget. *Success and Understanding.* Harvard University Press, 1978.

[Pia79] Jean Piaget. *Recherches sur la Généralisation.* Presses Universitaires de France, 1979.

[Pro97] Viera K. Proulx. *The Role of Computer Science and Discrete Mathematics in the High School Curriculum.* DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 36, 1997.

[Rog87] Hartley Rogers. *Theory of Recursive Functions and Effective Computability.* MIT Press, 1987.

[Sch02] Alan H. Schoenfeld. Making Mathematics Work for All Children: Issues of Standards, Testing, and Equity. *Educational Researcher, Vol. 31, No. 1*, 2002.

[Tuc95] Allen B. Tucker. *Fundamentals of Computing I: Logic, Problem Solving, Programs and Computers.* Mc Graw-Hill Series in Computer Science, 1995.

[Vel00] J.Angel Velazquez Iturbide. Recursion in Gradual Steps (Is Recursion Really that Difficult?). *ACM 1-58113-213-1/00/0003*, 2000.

[WDB98] Cheng-Chih Wu, Nell B. Dale, and Lowell J. Bethel. Conceptual Models and Cognitive Learning Styles in Teaching Recursion. *SIGSCE 98 0-89791-994-7/98/2*, 1998.

[Wie88] Susan Wiedenbeck. Learning Recursion As a concept and As a programming Technique. *ACM 0-89791-256-X/88/0002/0275*, 1988.

Appendix A

Recursion from Psychogenetic and Mathematical viewpoints

A.1 Summary of Piaget and collaborators' works about Recursion

The main ideas presented in the work "La Formation des Raisonnements Recurrentiels" by Piaget and collaborators published in the sixties by the Series of "Etudes d'Epistemologie Genetique" are summarized.

This work contains complete descriptions and analysis of several experiments including selected parts of conducted interviews. In the experiments several types of problems are posed to determine the psychogenetic evolution of reasoning by recurrence with respect to the intern course of constructing the structure of natural numbers (terms and operations). By reasoning by recurrence the authors refer to the structures of thought by which the subject is capable of solving problems involving both calculations and inferences, applying specific properties of the series as a whole. For instance, children are faced to questions that require to reflect about the recurrent procedure of the construction, or to establish relations between a number and its successor or/and predecessor, or questions aimed to develop some method of calculation or questions in which, given a number with some property, the other members of the series holding the same property have to be found. In some cases the material is concrete (pins, boxes, etc) and in other ones symbols are used.

A.1.1 Main Ideas of "La Formation des Raisonnements Recurrentiels" by Piaget and collaborators

The main problem to which the genetic analysis can give support consists in investigating if the specific numerical reasoning (reasoning by recurrence) is a late way of inference elaborated once the series of natural numbers has been constructed from zero to infinity or if it is involved from the beginning of number construction as a constituent mechanism of the series. The authors explain that since the number is constructed by the synthesis of the groupments of classes and relations, it can be argued that to this specific synthesis corresponds a specific way of reasoning. This is reasoning by recurrence founded in the inclusion between classes and in the serializations, which differs from inferences just based on the simple transitivity.

The authors pose the hypothesis of the existence of levels of recurrence from the elemental forms inherent to the intern construction of the series to more complex ones that proceed on the series once it is constructed.

First at all, the authors summarize previous works about structures such as modus ponens, modus tollens, the transitivity of the implication, the INRC group [PB66] and the notions of any number and any operation with respect to the problem. Benjamin Matalon, who conducts a physical experiment aimed to investigate the role of structures of modus ponens, modus tollens and the transitivity of the implication in reasoning by recurrence, indicates that there are differences between the reasoning about the implication between two elements and the implication in a series. This is due to the fact that an isolated relation is not identifyable from the psychological point of view, with the same relation inserted in a structure of set.

Regarding modus ponens and modus tollens structures, Pierre Greco in "Recherches sur quelques formes d'inferences arithmetiques et sur la comprehension de l'iteration numerique chez l'enfant", Etudes d'Epistemologie Genetique XI, states that the structure of modus ponens appears at the moment in which the series of numbers can be constructed by iteration (8-9 years), and the structure of modus tollens is of hypothetical-deductive character and then appears much later. Pierre Greco states that the main instrument of knowledge is the INRC group by which the subjects are capable of reasoning about the series as a whole coordinating the cardinal and ordinal transformations involved in its internal construction. This explains why the reasoning about recursion appears so late: the group INRC appears at the hypothetical-deductive period, while recursive methods are used from childhood. Other important conclusion added by Greco is that the INRC group does not lead to the construction of new structures of this type of reasoning, but simply allows reconstruct by reflective abstraction what has been involved in the intern construction of the series.

The coordination of the actions leading to the construction of a series has to be composed with other relations in order to evolve to iterative inferences: the understanding of the notion of "any number" and "any operation" and the development of structures of modus ponens, modus tollens and the transitivity of a relation (in this case the implication). This underlying logic is not a block, but can be analyzed in several "paliers" that appear at different moments of subject's development. In the case of the series of natural numbers, its iterative character gives rise to a confidence in its regularity from the first levels, but the understanding of the notion of "any" number appears much later. The authors explain this delay saying that the notion of "any" is linked to the evolution of the combinatorial from the operation of "vicariance"[PB66]. Working about the series and not just generating it demands the flexibility of a combinatory structure. The identified levels in the elaboration of the notion of "any" are include below.

A.1.2 The experiments

The experiments are based on requiring the children to repeat the same action and asking them about relations on the results. The material consists in elements sometimes discrete and equal to each other (pearls), sometimes discrete but different (pins with different lengths), sometimes continuous (liquids). The elements are contained in holders with the same or different forms and the actions are adding or taking out elements and putting the collections (C_n and C'_n) into correspondence.

The children are asked about whether the relation between the collections (equality or inequality) is conserved or not in the course of actions. The central problem investigated is: does the knowledge about the conservation arise from a transmission by recurrence, that is to say, a passage from the repetition of the actions to an iteration of the results or from just a recapitulation of the result of putting into correspondence, with no influence of the succession of actions?

Piaget classifies three levels:
A) no generalization of the relation (for instance, $C_n = C'_n$ is recognized if the holders have the same form and denied otherwise).
B) intermediate cases: the thought of the subject oscillates between the dynamism of the action which is understood and the recapitulation of the result of the action, that is to say, they think in terms of the constructed collections. Then the subject is able to subordinate the collection to its forming process, that is to say, to the successor operation (Piaget says "the $(n+1)th$ operation"). In other words, they grasp the sense of the constructive iteration, which is not immediate and in which figurative aspects interfere.
C) Once the subordination of the collection to the process of its construction is reached, nothing prevents the reasoning from being generalized to whatever collections formed in the same way, leading to the consolidation of the constructive sense of the iteration, opposed to just the consideration of the already created collections.

The thought moves from considering the result of the actions and centering on the collections independently of the process of its creation, towards reasoning in terms of the actions themselves and their iteration. The problem is to establish the nature of the reasoning by which the subjects affirm that two collections are equal and will infinitely be, if they are constructed by a correspondence one by one.

The main problem studied here is the evolution of the transmission from the repeated actions ($S_1S'_1...S_nS'_n$) to the iterative results ($C_{1,2,3,..} = C'_{1,2,3,...}$).
That is to say, $S_1S'_1...S_nS'_n \to C_{1,2,3,..} = C'_{1,2,3,...}$ and from an anticipated equality C_n = C'$_n$ to understanding that if $(S_n = S'_n) \to (C_n = C'_n)$, it will necessarily be $(S_{n+1} = S'_{n+1}) \to (C_{n+1} = C'_{n+1})$.

In other words, the problem is focused on the relation established by the subject between his/her own action and the implied results. In levels A) and B) above, the transmission is more apparent than real because the subject is centered either on his/her own action independently of the transformations introduced on the objects or on the final results independently of the process of transformation due to this action. In the first case, the subject becomes aware of this repetition of the action and affirms that the collections are equal, but without effective transmission due to the lack of reflection about the introduced transformations on the objects. On

the second case, focusing on the collections (the objects) causes that the action by which they have been constructed is forgotten. On the other hand, in level C) the subject coordinates his/her action and the obtained results. In effect, each action repeated after the previous one, differs from it because of its rank in the ordering of the succession and at the same time it adds one object to what has been joined in the previous iteration. Once the subject establishes a coordination between the succession of his/her actions and their result, a local synthesis specific to these actions is stated between the order of the succession $S_1S'_1, S_2S'_2$, etc and the increasing of the collections C_1, C'_1, C_2, C'_2, etc., extending the construction of the number with an aspect of inferring by recurrence, in which the most important generalization is not the passage from 1 to n, but from n to $n+1$.
This shows that reasoning by recurrence is from its beginning indissolubly connected to the construction of the series of natural numbers and constitutes its aspect of inference, long before the elaboration of higher forms of reasoning by recurrence.

To sum up, the elementary form of recurrence described in this work constitutes the aspect of inference of the synthesis between class inclusion and serialization. The interest of this study is to show the strong relation between the mechanism forming the number and the mechanism of the recurrence itself. Consequently it is possible admit the existence of an elementary form of recurrence typical to the progressive and spontaneous number construction, different from higher and explicit forms of reasoning by recurrence on the series once its construction is reached.

A.1.3 The problem of "any"

Piaget says: "One thing is to construct an ordered set over which general functions and relations will be stated and another thing is to reason directly on such transformations, by complete assimilation of terms and operations." In the first case, the subject reasons about *the numbers* and in the second one, he/she does it about *any number*.

In formal thought a structure can be constructed after its elements and operations, but in natural thought, terms and relations are simultaneously and reciprocally defined. In the same way as for the child the judgements do not consist in relating concepts previously elaborated, the structures do not consist in organizing elements that would be firstly constructed by operations different from these structures. In this way, the group of numbers does not arise from imposing to the numbers of childhood thought a group structure corresponding to adolescence thought. In other words they are not the numbers that hold one structure or another, but a structure corresponding to the numbers of childhood thought is transformed (terms and operations) into a higher one (group structure of numbers) corresponding to adolescence thought. The structures of thought are structures of transformations, their elements are not only the terms over which they are applied but also the operations themselves. The question is thus to investigate which is the new operative organization corresponding to "any number". This investigation involves the study of genetic levels of iterative inferences.

Several experiments are conducted and analyzed giving rise to conclusions summarizing the evolution of the iterative thought, which are briefly presented in next section. From the experiments, the authors learned not only that the arithmetic inferences progress in the framework of the general structures of the intelligence but also that in each level of this development the notion of number becomes enriched.

A.1.4 The evolution of iterative relationships

Piaget distinguishes three levels between 5 and 9 years old to which correspond a certain way of understanding and using the relations generated by the iteration[1].

Level 1: 5-6 years old, The constructive iteration

The child has an elementary and practical operation that allows him to construct one by one the successive cardinals, to identify a number in the series according to its ordinal position and even more to admit sometimes that the arithmetic difference between two consecutive terms is constant. The composition of the relations is not complete, not reversible and not associative. From $k+1 \rightarrow Sk$, it is not always inferred $S(k+1) = SSk$ nor $SSk = k+2$ nor $(k+1)+1 = SSk$ nor $Sk - k = 1$.

Level 2: 6-7 years old: The qualitative iteration

The progress is from k+1=Sk to (n) (n+1 = Sn), that is to say "for all n", but it is not always possible to substitute n by Pn+1 or Sn-1 when it is required by the question. The synthesis between cardinality and ordinality is not achieved yet. Consequently, the subject uses the expensive procedure by which the iteration is employed for fixing the cardinal value of the terms and the operations are over these values, instead of reasoning over the iteration itself. There is an advance in coordination, but not yet enough to become a set structure in which all the inferences can be made without recoursing to the same beings that this structure organizes.

Level 3: 8-9 years old: The operative iteration

It is close to 9 years old that this structure reaches its equilibrium, by the fusion of cardinality and ordinality, at the same time that the operations based on class inclusion or order relations composition do. The iteration becomes operative and constitutes the series of natural numbers as structure of a set of ordered cardinals, although it does not work as independent structure, detached from cardinal objects that it constructs and organizes.

However, according to the different situations in which iterative inferences have to be applied, these are still subject to certain figurative intuitions, as is testified by some spatial errors or coming back to one by one procedures.

Therefore, the existence of a general structure is confirmed: on the one hand, the errors and doubts do not disappears until the adult age, which means that surmounting these difficulties is not a matter of a new structure or a different structure of higher level, on the contrary, it is a matter of flexibility and efficiency. On the other hand, although limited, the generalization of these iterative inferences is possible in the various considered situations. The relations recognized as valid in some region of the series can be transported to another region (if $k + p = k'$, $S(k+p) = Sk'$ and $P(k+p) = P(k')$ for all p, etc). Before this level, the successor operation just generates similar but different numbers. In this level, all the numbers are elements of a structure in which a unique operative tenet guarantees the general properties

[1]in the following, S and P stand for successor and predecessor respectively.

and the homogeneity.

What has been said before does not lead to the belief that inference reasoning will not face further difficulties, for instance, those due to relations composition or serial and figurative illusions. In the course of the progress of inference reasoning, it becomes organized in more and more stable, flexible and general systems. Numbers to which this reasoning is applied become other numbers. From the progress of inference reasoning not only hypothesis about the transformation of the manipulated objects can be induced, but also solution procedures and spontaneous formulations of problems can be examined to obtain information about the evolution of mathematical beings. An example of the genetic circle appears: the concept allows the inference, but what if not the inference itself formulates the concept? The teacher? The collective unconscious? The socio-cultural environment?

Level 3 (9-10 years old) The generalized iteration

The iterative generalization is in this level "horizontal", what means that it needs the concrete support of a series of terms. The iteration is aimed to construct the terms and not to calculate or directly reason.

Spatial errors or "figurative" intuition show up the vulnerability of iterative inferences. In those cases, the errors interfere with the reasoning or the calculations bringing it to pre operative forms for justifying it, for instance, all the subject agree that no term can be equal to its successor but not all are capable of reviewing their methods of calculation. Many children are not shocked by erroneous affirmations.

The weakness have two sources: the insufficiency of a complete operative structure to calculating with relations (logical insufficiency) and the lack of ability of using iterative relations out of the serial context of enumerated objects. To sum up: in this level the number is operative, the iterations is operative, but the operations are still concrete operations.

Level 4 (10-12 years old): The iteration that generalizes and the qualified "any"

The progress is characterized by two essential novelties: first at all, the iterative operation becomes a tool of the inter operators generalization, that is to say, most of the subjects begin to surmount the previous difficulties. In the plane of inferences Piaget speaks of iteration that generalizes and in the plane of the number itself, he speaks of an integration of cardinals and their operators in the same system. Observe that Piaget works with numerical series which are represented with different type of objects: boxes, pins, etc. Some of the experiments are made using symbols, as well.

Secondly, the iterative relation imposes to the series of terms a stable structure which protects the calculations of figurative errors. The spatial errors diminish (not disappear) and when these errors lead to contradictions, the question is reconsidered. In this level, the iteration conserves not only the relations between terms but also the operations over these relations. Iterative succession becomes the tool of calculating: subjects reason by direct inferences over S and P, inclusive arithmetically. In this way, in this level, the iteration is more than the property of a series, it is the foundation of an inference structure. The numbers have lost their individuality, they are "any thing 'sub specie iterationis'" (something that is created by iteration). It is the dawn of the personal power of the cardinals and at the same time it is the

dawn of the recurrence.
Level 5 (12-14 years old):
Iteration by recurrence and formal arithmetic
Until now, the term recurrence has been avoided because of the risk of giving it a too strong sense (recursion, complete induction) or a too weak sense and to reduce the recurrence to the iteration. The recurrence is based on the "any" number and the pure order. This conditions are detached by 12 years old and are effective at level 5 at the adolescence. Arbitrary substitutions mark at level 4 the decadence of numerical absolutes and the equivalence of all the sequences of cardinals is admitted (13, 14, 15, ...) or (21, 22, 23, ...). But for calculating, the cardinal is needed. At level 5, this need disappears and the order is abstracted. The iteration by recurrence appears at the end of a long process of constructing and abstracting. This process can be described in two phases: from level 1 to 3 and from level 3 to 5. In the latter, the "any" number has been consolidated and the inference by recurrence founded.

A.2 Mathematical Definitions

Several authors define recursive functions (primitive, general, μ-recursive) referring to numerical functions [Cut80], [Rog87], [LP98], [Kle77]. These definitions can be summarized as follows:
Recursion is a method of defining a function by specifying each of its values in terms of previously defined values, and possibly using other already defined functions.
$f : N^n \to N$ is an n-ary partial function. If the domain is N^n, the function is total.
Theorem: let $x = (x_1, x_2, \dots, x_n)$ and the functions
$f : N^n \to N$
$g : N^n \times N \times N \to N$
Then there is a unique function
$h : N^n \times N \to N$
$h(x, 0) = f(x)$
$h(x, y + 1) = g(x, y, h(x, y))$
Where $n = 0, h(0) = a, a \in N$ and $h(y + 1) = g(y, h(y))$.

The functions that can be defined by this method are primitive recursive. There are functions that are not primitive recursive, for instance, the Ackerman function, where the argument to a recursive application is itself obtained by involving the same function:
$f : N \times N \to N$
$f(0, y) = y + 1$
$f(x + 1, y) = f(x, 1)$
$f(x + 1, y + 1) = f(x, f(x + 1, y))$

These functions are called general recursive functions. They can not be defined as the h function above: they are generated by the operation of minimalization or the minimalization operator μ.

Rózsa Péter published in 1976 "Recursive Functions in Computer Theory" in which she defines recursive functions that are used in programming languages, although the concept of induction-recursion as used in modern programming languages was introduced in Peano's axioms in 1889, 1891. In 1931 the concept of recursive functions was formally defined by Kurt Gödel. Thereafter several papers of Rózsa Péter about various levels of recursive functions were published and her theory is expounded in her book "Recursive Functions" published in 1951 [Kle81].

In [Acz77], the concept of recursive function is defined, using the way by which the elements in a inductively defined domain are generated. The definitions are summarized below.

A.2.1 Peter Aczel's Definitions

1) A rule is a pair (X, x) where X is a set of premises and x is the conclusion. The rule is usually written $X \to x$.
2) if Φ is a set of rules (rule set) the a set A is Φ-closed if each rule in Φ whose premises are in A also has its conclusion in A.
3) If Φ is a rule set then $I(\Phi)$, the set inductively defined by Φ is given by
$I(\Phi) = \bigcap\{A \mid A \text{ is } \Phi\text{-closed}\}$
Example: $N = \{0, 1, 2, 3, ...\}$ is the smallest set containing 0 and closed under the successor function $N = I(\Phi_N)$ where $\Phi_N = \{\{\} \to 0, \{n\} \to suc(n) \text{ for } n \in N\}$
This definition of N justifies the principle of mathematical induction: if P is a property that holds for 0 and holds for $suc(n)$ whenever it holds for n, then P holds for all natural numbers.

Stating an induction principle can be generalized to any rule set: to each rule set Φ there is a principle of Φ-induction:
If P is a property, such that whenever $\Phi : X \to x$ and $\forall\, y \in XP(y)$ then $P(x)$, then $P(a)$ holds for every $a \in I(\Phi)$.
The above principle is the natural tool to use in proving properties of $I(\Phi)$.

A.2.2 The Well Founded Part of a Relation

Let $<$ be a binary relation on a set A. The well founded part of $<$ is the set $W(<)$ of $a \in A$ such that there is no infinite descending sequence $a > a0 > a1 > a2 \ldots$.
The relation $<$ is a well founded relation if $A = W(<)$. $W(<)$ can be inductively defined as follows:
Let $\Phi_<$ be the set of rules $(< a) \to a$ for $a \in A$, where $(< a) = \{x \in A \mid x < a\}$
$W(<) = I(\Phi_<)$
Definition:
The rule set Φ is deterministic if $(\Phi : X_1 \to x \text{ and } \Phi : X_2 \to x) \to X_1 = X_2$
For deterministic Φ, functions on $I(\Phi)$ can be defined by recursion on the way objects in $I(\Phi)$ are generated.
An example is the operation $s : F \to F$ (F = Set of well formed formulas of a formal language), such that $s(\varphi(v)) = \varphi(t)$ obtained by substituting a term t for all free occurrences of a variable v in a formula $\varphi(v)$ is defined by recursion on the way the formula is generated.
If the rule set is not deterministic, the following warning holds: a function on it is only guaranteed to be well-defined if each element generated by the set can be

constructed just in *only one way*, that is only be a unique combination rule applied to unique elements generated by the rules. So this kind of recursive definitions will work fine for fully bracketed arithmetic expressions, binary trees, lists and recursive trees, but might not yield a well-defined function if the definition of f is the elements generated by a set of rules S such as the following, for instance:
$\epsilon \in S$
if $x \in S$ then $axb \in S$
if $x \in S$ then $bxa \in S$
if $x, y \in S$ then $xy \in S$
The reason is that some elements of S can be constructed in more than one way, according to the last rule.

To the forms of defining functions by recursion, induction principles are associated to prove properties over them. For instance, the principle of mathematical induction and the principle of complete induction are used to prove properties of functions defined on the set of natural numbers by primitive recursion and course of value recursion respectively. Similar principles are stated for other domains as well, for instance the principle of structural induction over lists or trees.

A.2.3 Course-of-values recursion

The terms recursion and induction are strongly related. Briefly, it can be said that a function can be defined by recursion and a property can be proved by induction. That is to say, recursion is a certain form of defining and induction is a way of making proofs. There are several types of recursion as well and in general, for each of them a principle of induction can be stated. Properties involving functions defined by primitive recursion, for instance, can be proved by the principle of *mathematical induction*, in which the application of the induction hypothesis is allowed only on the immediate predecessor of the induction variable.

The principle known in mathematics as *principle of complete induction* is also called *course-of-values induction*. According to it, the induction hypothesis can be applied on any element smaller than the induction variable. For instance, course-of-values for lists allows to apply the induction hypothesis on any sublist of the original list.

To this induction principle, corresponds a form of defining functions by recursion called course-of-values recursion. In it, the inductive case of the definition states the value of the function for a sepecific argument in terms of values of the function for smaller arguments. For instance, the algorithm of binary search on a list is defined in terms of searches on sublists of the original list.

It can be shown that every function specified using course-of-values recursion can be implemented by primitive recursion.

Appendix B

Antecedents

B.1 Introduction

The author of this thesis has been teaching courses in functional programming and discrete mathematics as part of the curriculum of computer science studies at Universidad de la República and Universitario Autónomo del Sur (Montevideo), for several years. It was observed a need for the adequate preparation of entering students, in particular that the students have serious difficulties to understand concepts such as high order functions, recursion, induction, type systems and logic in general. Some groups of students at the first computer science year were surveyed and some results that confirm this fact were found. For instance, they never describe a function including domain and range (all is R), they hardly ever use either quantifiers carefully in proofs or notions of logic in calculations (what counts are the results of calculations), they treated induction as a recipe to solve certain problems (often involving sums of numbers). The difficulties in learning the concepts are not exclusive of computer science education, on the contrary, they are also pointed out by mathematics educators.

Several efforts are made to improve the learning process from both mathematics' and computer science's side. Although these efforts are really useful and positive they are not sufficient; they are isolated attempts to solve a problem that is common both for mathematics and computer science education. What is needed is the assembled work both of mathematics and computer science educators to create and teach new courses to be introduced as part of the basic mathematical knowledge required to pursue scientific or technical studies. In some sense, computer science has to play an anlogous role to that physical sciences played in the past influencing the directions of mathematics education.

B.2 Previous work

In 1996 new contents and methodology of the course in discrete mathematics to computer science students were designed. The main novelty consisted in the integration of programming and mathematics, using the functional programming language Haskell as a formalism to implement mathematical concepts, in a similar way of what is described in projects like Scheme[1] or Beseme[2]. Results from putting these ideas in practice can be found in [1] [2] [3] [4] [5] (all in Spanish).

At the same time, contacts with mathematics educators both from high school and University took place and an informal working group -called Secutuniv- was established[3]. In it, mathematics and computer science educational problems were analyzed, several opinions were discussed, reports were elaborated and talks were given.

One of the outcomes of these activities consisted in making contact with Dr. Ed Dubinsky, who has worked in mathematics education since many years. In particular, he and other teachers in USA use a programming language called Isetl as a tool for helping students to learn mathematics in calculus, algebra and discrete mathematics University courses.

In 1999, supported by Pedeciba[4] and Instituto de Computación de la Facultad de Ingeniería of Montevideo, a project involving the preparation of high school mathematics teachers was designed and conducted. In particular, topics in discrete mathematics using Isetl as programming language were introduced to reinforce important concepts. The experience was successful and the course is given every year since then.

Dr. Dubinsky became interested in the work and a fruitful interchange of e-mails took place. In 2001, at the occasion of an edition of the Relme meeting[5] held in Buenos Aires, he was invited to discuss with Pedeciba's authorities in Montevideo about the possibility of tutoring the research described in this thesis.

The plan involved, on the one hand, the study of educational-related subjects: the first year, three courses in Facultad de Humanidades in Montevideo (Psicología Evolutiva, Psicología de la Educación and Procesos Grupales y Aprendizaje) and one more with Dr. Dubinsky about four works of Piaget (The Grasp of Consciousness, Success and Understanding, Mathematics Epistemology and Psychology and Psychogenesis and the History of Sciences) were taken.

The second year the author of this thesis stayed in Kent State University in USA where a course of Dr. Michel Battista (Research in Math Education) was taken. As final work, a project about the learning of recursion including the first experience in conducting students interviews was carried out. A summary of its results can be found in [6]. In USA, discussions with Dr. Dubinsky about the specific topic of the thesis took place and it was finally determined that it would be on the concept of recursion, which is important both for mathematics and computer science education.

Immediately after the return to Montevideo, the empirical work was designed and put in practice. The time for each of the activities was approximately as follows:

---

[1] http://www.teach-scheme.org/

[2] http://www.cs.ou.edu/ beseme/

[3] http://imerl.fing.edu.uy/secutuniv/

[4] Programa de Desarrollo de las Ciencias Básicas

[5] Reunión Latinoamericana de Matemática Educativa

- Jun 03 - Aug 03: selecting the students and preparing the interviews,
- Sep 03 - Nov 03: conducting the interviews,
- Dec 03 - May 04: analyzing the gathered data,
- Jun 04 - Aug 04: designing and conducted the complementary activity,
- Sep 04 - Nov 04: analyzing the gathered data,
- Jul 04 - Aug 05: designing and writting the thesis.

It is worthwhile saying that Rumec community[6] provided financial support for the translation to English of the recorded interviews.

[6]Research in Undergraduate Mathematics Education Community

Bibliography

[1] Sylvia da Rosa, Gustavo Cirigliano: *Matemática y Programación*
VI Congreso Iberoamericano de Educación Superior en Computación, EDUC98, PUCE (Pontificia Universidad Católica del Ecuador), Quito, Ecuador, octubre 1998.

[2] Sylvia da Rosa, Gustavo Cirigliano: *Ensayo sobre Matemática aplicada a Computación*
Anales del VII Congreso Iberoamericano de Educación Superior en Computación, Paraguay, 1999.

[3] Sylvia da Rosa: *Matemática y Computación: hacia una disciplina integrada*
Congreso Informática en la Educación de la Conferencia Internacional Informática 2000, Universidad de La Habana, Cuba, 2000.

[4] Sylvia da Rosa: *La Matemática Discreta como Formación Básica*
Reunión Latinoamericana de Matemática Educativa, organizada por el Comité Latinoamericano de Matemática Educativa, Buenos Aires, 2001, http://webs.sinectis.com.ar/ccrespo.

[5] da Rosa Sylvia y Gustavo Cirigliano: *Formación en Matemática Discreta usando Lenguajes de Programación*
Anales del Congreso Iberoamericano de Educación Superior en Computación, Uruguay, 2002.

[6] Sylvia da Rosa: *Psychogenesis of the Concept of Recursion*
Congreso Iberoamericano de Educación Superior en Computación, CIESC 2003, La Paz, Bolivia.

Appendix C

The Interviews

This Appendix contains the complete description of the interviews and the material used for selecting the students (in Spanish).

C.1 Questions Posed at the First Interview

Searching a word in a dictionary:

Q1:	What is a dictionary?
Q2:	Knowing that a word, for example *gato*, is in the dictionary and also in this novel, where do you think that will it be simpler (easily and quickly) to find it?
Q3:	Why?
Q4:	What makes the difference then between a dictionary and any other book?
Q5:	Look up the word *gato* in the dictionary.
Q6:	Describe, step by step, how you achieved it.
Q7:	Knowing that a word is in the dictionary, is it always possible to find it?
Q8:	Why?
Q9:	Start looking it up again.
Q10:	Why are you still searching there?
Q11:	What happens with this part of the dictionary?
Q12:	So you do not continue searching through the whole dictionary?
Q13:	What happens with the dictionary then, when you are still searching?
Q14:	Then, why are you sure that you will find it?
Q15:	Do you agree that you compare the word you look up with the ones in the dictionary?
	The alphabetical order can be represented saying that one word is "smaller or greater" than another.
	For example, *gato* $<$ *perro*, *gato* $>$ *dos*, etc.
Q16:	Then, which are the possible results of each comparison?
Q17:	How does the result of a search influence the one after?

Q18:	Do you agree that in each search there are things that are done the same way and things that vary?
Q'18:	Could you determine them?
Q19:	If instead of giving instructions to a robot, you had to explain to a little child -who knows how to read and knows the alphabet- how to look up any word in a dictionary, what would you say?

Counting the number of a's of words of a language:
The following definition and questions are given:
Suppose that the inhabitants of an unknown planet have a language such that words are formed just using "a" and "b", according to the following rules:

- ab is a word.
- If * is a word then a*a is a word.
- If * is a word then b*b is a word.
- Only the words obtained by application of the above rules a determined number of times are words of the language.

Q1:	Write some words of the language.
Q2:	Can you determine of these sequences which are words of the language and which are not.
Q3:	Why is this one a word and this one is not?
Q4:	How did you form this word?
Q5:	Let's see this word of the language, which rule did you use to form it?
Q'5:	Which was the last rule used?
Q6:	And this other one?
Q7:	In this one, which would the little symbol be? And in this one?
Q8:	Could the little symbol be like this? (Writing a sequence that does not belong to the language).
Q'8:	Why not or why yes?
Q9:	Then, what does the little symbol have to be?
Q10:	A student said that rules 2 and 3 say that a word is formed from a word formed before. Do you agree?
Q11:	Then in order to know if a sequence of a's and b's is a word, what can we watch?
Q12:	How many a's does this word have?
Q'12:	And the little symbol? And here? (The complete word).
Q"12:	In any word, if we know how many a's the little symbol has, can we determine how many a's the word has?
Q"'12:	How? Write it down please.
Q13:	Determine the a's of ababaaaabbbabbbbaaaabababa by using only what you have written.
Q'13:	What is missing in what you wrote so as to be able to use it until the end?

C.1.1 First Interview

The complete interviews follow.

C.1.2 Interview with Andrés

Q1:
R: A dictionary is a book in which we look for the meaning of words.
Q2:
R: In the dictionary.
Q3:
R: Because it is in alphabetical order.
Q4:
R: That has a certain order to be ... to find what you are looking for.
Q5:
R: (He reads the meaning)
Q6:
R: I opened more or less in the middle and saw which letter I was, I was in "c", I think, and I went forwards ...
Q: Why did you go forwards?
R: Because "g" is after "c", then ... I went too forwards and came back a little backwards. When I got to "g", I was in "gp" and then I went backwards ... When I got to "ga" and there I looked for ... "gas" ... and a little bit downwards *gato* was going to appear.
Q7:
R: Yes.
Q8:
R: Because it will always be in that order. Because looking for in this way and due to the order the dictionary has, I will find it.
Q: When you tell me that you open the dictionary on letter "c", you went forwards. What happens with what there is before "c"?
R: They are words that start with "c" or "b" or with "a", then they do not help me.
Q: Then when you go on searching letter "g", Do you use the whole dictionary?
R: No.
Q: Where do you look for?
R: Up to "g", in "h" ...
Q: And what is that in the whole dictionary?
R: What?
Q: So, for example the words that are before letter "c" do not interest you, you get rid of them.
R: Yes.
Q: And you will go on searching.
R: Yes.
Q: So you will not search in the whole dictionary, do you agree?
R: Yes.
Q: In what thing of the dictionary do you look for the word *gato*.?
R: (He does it)

Q: Where are you there?
R: In "f".
Q: Then, what happens with all this part? (pointing to what there is before "f").
R: I do not need it.
Q: And then where will you continue searching?
R: Forwards.
Q: And what is that in respect to the whole dictionary?
R: The rest.
Q: And what is this rest according to the whole dictionary?
R: More useful for me.
Q: Yes, and with respect to the number of words that has?
R: Greater (it is wrong!).
Q: This part is with respect to the whole dictionary?
R: Smaller than the complete dictionary.
Q: And now, how do you go on searching?
R: (he does it)
Q: And now what happens with all this part?
R: I do not need it.
Q: And where will you search?
R: Only on this page.
Q: And what is this page like according to the previous section you had?
R: Much smaller.
Q: Then what happens with the dictionary when you are still searching?
R: Some parts are being discarded.
Q: And it becomes ...?
R: Smaller.
Q: When you find the word, what happened to the dictionary?
R: It is useless, it only helps me for that word only.
Q: So, what was the dictionary transformed into?
R: Into only one word.
Q: That is why you find it, because you search in sections that become smaller and smaller within the dictionary.
R: Oh, of course.

Simulation

Q: Which are the possible results of a comparison between two words?
R: $>$ or $<$ or $=$
Q: (Sample of chosen $<$ searched, etc).
Q: Which is the 1st instruction do you give me to look for the word *luna* in this list?
R: Turn over some slips of paper.
Q: I only know how to follow these instructions ...
R: Oh! Choose word W10.
Q: Where?
R: ...
Q: The robot does not know how to determine where to choose.
R: In these slips of paper.

Q: Which would the instruction be?
R: Do I have to follow the order? (He refers to "how" the instructions are written).
Q: The robot is stupid and only follows your instructions. How do we go about the dictionary to find a word?
R: ... Form a list from W1 to W20
(I stay there)
R: And now, form the list from W4 to W16.
(I stay there)
R: Choose word ... P13 *perro* $>$ *luna*.
Q: Write it in the result.
R: I did not define the relation.
Q: No, I define the relation and give you the result.
Q: I haven't found it yet. So?
R: Now ... reduce ... form the list again, W5 to W12. Choose word. Define relation. W8 $<$ *luna* so form the list again from W8 to W2, choose word, define relation, the result ...
Q: W10 $>$ *luna* and the previous one had been W8, then what list do I make?
R: Form a list from W9 to W9, choose word, define relation.
R: The result is W9 $=$ *luna*.
Q17:
R: It is relevant because I am reducing the number of words within the ones I am looking for.
Q18:
R: Yes.
Q'18:
R: The number of words varies among the ones I am looking for, numbers vary.
Q: And what remains the same?
R: That words from a group are always within the greatest group. (This is very interesting, he is saying the invariant, we use that later).
Q: Is there anything else that remains the same?
R: The order in which I am looking for.
Q19:
R: First of all, the alphabet; then that you have to start by the 1st letter of the word, then the 2nd and in the same way successively.
Q: Do you think that with this explanation the boy will know how to find a word?
R: Do we suppose he knows the alphabet?
Q: Yes, let's do so.
R: Then, when he wants to look for a word, he should take the 1st letter, place it in the dictionary and then he takes the 2nd letter of the word ...
Q: What do you mean by placing? Suppose that he opens the dictionary.
R: He has to check if the 1st letter of the word he is looking at there, where he opened the dictionary is $>$ or $<$ according to what we determined here, that the first letter of the word that he is looking for and that he goes forwards or backwards in the dictionary as it is $>$ or $<$.
Q: And what does he do afterwards?
R: When he gets to that letter he is looking for (the 1st of his word) within the

group of all the words that start with that letter, he does the same but with the 2nd letter.
Second activity.
Q1:
R: (He does it)
Q: Why this one is from the language?
R: Because it has "ab" and to the right and to the left it has one "a" and to the right and to the left it has one "b".
Q2: (I give him sequences.)
R: Yes.
Q: How?
R: (He does it).
Q5:
R: with the 1st and with the 3rd rule.
Q: This one ?
R: With the 1st, the 3rd and the 2nd.
Q7:
R: (He does it quickly and correctly for all the words).
Q8:
R: No.
Q: Why not?
R: Because it does not follow the rules (immediate).
Q: So what does it have to happen with the little symbol?
R: It has to follow the rules.
Q10:
R: Yes.
Q: And that previous word, what would it be, then?
R: "ab", the rule number 1. (That is, he gets confused with "previous" and "initial").
Q: No ...
R: Oh yes, it could be the mixture of the 3 rules.
Q: What was this? (I point to a * that he already marked).
R: (As he didn't put "*", it is difficult for him to see that the * is a sequence).
Q: I made you a question and you marked this with (a circle) , what was the question? Do you remember?
R: What was the * there? (He means in that word).
Q: So this is ...
R: The *.
Q: So, according to what we said of what the student said, which is the previous word so as to form this one?
R: The little symbol.
Q: Then in each word, what is *?
R: The previous word.
Q11:
R: If it obeys the 1st rule and then the others.
Q: Knowing that it was formed based on the *.

R: ...
Q: What does it have to happen with * of each word?
R: It has to follow the rules.
Q: What is * in this word?
R: "ab" (it is wrong)!! (That is, it is not sequential, after answering well many questions, he is asked again and answers one of them wrongly).
Q: No, what was the *? Remember.
R: The base word.
Q: No, it was this here, here, this, here this, what was *?
R: The previous word to the last application (of the rules).
Q: And what must happen?
R: It must obey the rules.
Q: Does it obey?
R: No.
Q: Why not? How do you do to know?
R: (He marks all the corresponding little symbols) and this one does not have it anymore (he got to the one that did not obey).
Q: What is it that you do?
R: I'm coming backwards.
Q: Watching what?
R: The little symbol in the words.
Q12:
R: (Answers the particular cases well).
Q"12: No, because depending on the little symbol I applied rule 2 or rule 3.
Q: Write it down (He does it).
Q13:
R: I mark the little symbol and has one, two ...
Q: Without counting, always applying what you wrote.
R: Oh! Sure, the ones of the little symbol + 2 ... I mark the little symbol again and it will have the same number because it is rule 3, I mark the little symbol again and now the rule I applied that was rule 2. So, it will be 2 +, is this way OK?
Q: Yes, yes.
R: And I now mark again and the rule applied was number 3 so that it remains likes this, as it is, and then I applied rule 2 so it will be 2 + and here I applied rule 2, so 2 +, then rule 3 so, nothing and there is the first.
Q: And how many "as" does it have?
R: 1, always + 1 (he sums up).
Q: Let's see if the result is OK.
R: (He counts 9 "as").
Q: If we know how many the symbol has ... ?
R: Yes, yes.
Q'13:
R: ... rule 1.

C.1.3 Interview with Cecilia

Q1:
R: It is a book where there are words with their meanings.
Q2:
R: In the dictionary.
Q3:
R: Because it is in alphabetical order and it is possible to look up in the words with "g" and there you find it.
Q4:
R: That they are ordered alphabetically.
Q5:
R: (She finds it).
Q6:
R: I searched letter g.
Q: First of all, the dictionary is closed, what do you do?
R: I open it up, look for letter g , (she does it).
Q: Why while looking for it do you go that way?
R: Because of the alphabetical order. I look for it then I look for letter "a".
Q7:
R: I think so.
Q: You can imagine that you can't?
R: No, I will find it.
Q8:
R: Because I look for it the same way as I looked for *gato*.
Q: Tell me slowly again.
R: I look for the first letter, after finding the first letter, I start by the second, because they are also going to be in alphabetical order.
Q: And where do you search?
R: ...
Q: What happens with the rest? Close the dictionary and open it again.
Q: Why are you going backwards now?
R: Because of the alphabetical order.
Q: And why do you get until there?
R: Because I do not care about the rest.
Q: Then, do you look up in the complete dictionary?
R: No, on the letter I'm interested in.
Q: What is it from the dictionary?
R: One page.
Q: One page or ... here you do not have one page (they are many), what do you have from the dictionary?
R: A piece.
Q: What is this piece like with respect to the whole dictionary?
R: It is different.
Q: In what sense?
R: (She does not realize) It has other words.

Q: We look for the word *gato*. Here, we are searching here, in this piece as you said and, what is it like regarding the whole dictionary?
R: A part.
Q: And one part, what is it like with respect to the whole?
R: Different.
Q: In what sense?
R: That it does not involve all the words.
Q: That means, what is it?
R: ...
Q: And if you go on searching?
R: I go backwards.
Q: And where do you search?
R: In this part.
Q: What is this part like with respect to the previous one?
R: Different.
Q: In what sense?
R: It has different words.
Q: More or fewer?
R: The same number, no, I don't know.
Q: When you start looking for in the whole dictionary, how many words do you have?
R: Oh! One million.
Q: And in the piece?
R: Some ...
Q: And if you go on looking for, in the next part?
R: Different other ones.
Q: More or fewer?
R: More.
Q: ...!? (I could not avoid to be really surprised!)
R: Fewer ...
Q: When you found the word, how many do you have?
R: One.
Q: Why, then, will you find a word?
R: Because we are shortening the number of words.

Simulation

Q: When you start searching, where do you do it?
R: In the whole dictionary.
Q: So which would the 1st instruction be?
R: From W1 to W20. Now I have to choose.
Q: You give me the order.
R: It would be W11 ...
Q: Why?
R: Oh! Then oh no.
Q: I have this list, then now, what would I do? What do you do when you search the dictionary?

R: I start by the letter that concerns me.
Q: Of course, but here we do not know.
R: I go on looking.
Q: I cannot look.
R: Oh!
Q: What would I do? These are the things I can do.
R: I can choose a word.
Q: You can give me the order, I do it.
R: Yes.(She does it).
Q: And now?
R: Define the relation between words.
Q: I give you the result *luna* $>$ W7
Q: What do I do now?
R: Form the list from W7 to W2 now.
Q: And now?
R: Choose another word.
Q: I choose this one.
R: Define the relation.
Q: *luna* $<$ W14, and now?
R: The list will be from W7 to W14 and now choose another word.
Q: I choose this one and define the relation *luna* $<$ W11, and now?
R: Form the list from W7 to W11... (she writes "choose a word and define ..., as well)
(I choose another word.)
Q: I define the relation and I tell you *luna* = *luna*. And now ?
R: The search is over.
Q17:
R: We have fewer options and we are going to get the result.
Q: How many searches did it take us?
R: 4.
Q18:
R: Yes.
Q'18:
R: The other word remains, the space between words varies until we get to the result, in fact what varies is the words that remain until I get to the result.
Q: And what remains?
R: The word we are looking for.
Q: And what else?
R: Choose another word.
Q: And what do we do?
R: ...
Q: Do we do different things in each search?
R: Oh, no we always do the same. (Observe that when one says: "we do something different?" immediately they say: "no, we always do the same" in a way that shows that for them it is obvious and that is why they would not say at once).
Second activity

Q1 and Q2.
(Cecilia answers using the little symbol in the same way that Gabi[1] and Sofía did, that is, indicating that a*a, for instance, is a concrete word). (We repeat Q2.)
R: I know that "ab" is a word.
Q: Where is that rule? (She writes wrong!)
R: I cannot add "a" and "b" separately ...
Q: Exactly, let's read the rules again. Is there a rule that says you can add a "b" and an "a"?
R: No.
Q: So?
R: (Goes putting.)
Q: Which is the rule because of that you formed this word?
R: I will not have much to form.
Q: Oh! You can go on adding ... (She had not understood that she could go on applying the rules.)
Q: Does this word correspond to the 2nd rule or the 3rd?
R: the 2nd.
Q7:
R: All this (she does it well).
Q: In this other one?
R: This would be the 3rd rule.
Q: And which is the little symbol?
R: This one (she does it right).
(We see the little symbol in many cases).
Q: What must the little symbol do? Could it have this form, for example? (breaking the rules).
R: No.
Q: Why not?
R: It must always have the same letter at the end as at the beginning.
Q: That is ... ?
R: It must follow the rules.
Q: How many "as" do we have here? Here, here. She answers 5, 3, etc.
Q"12: If we know how many "as" the little symbol has ...?
R: Yes (immediately)
Q: Why?
R: Because I added 2 to the little symbol.
Q: And in this case, how many did you add here?
R: None.
Q: That means that you add 2 or nothing.
Q: Then if we know the number of "as" of the little symbol, how many will a following word have?
R: One more (Observe that I think this is the influence of a preconception linking "following" with "one more", she says it very quickly, without thinking, "one more" and suddenly she corrects herself and says: "2 more, sorry").

[1] Gabi is one of the students who participated in the pilot interviews.

Q: Or?
R: Or none, say, the same.
Q: Is the number of "as" even or odd?
R: odd
Q: Why?
R: Because I will always have one more.
Q: Of what?
R: Of "as".
Q: Why are you sure that it is always odd?
R: Because I start with "ab".
Q: How many "as" are there here?
Q: One and I add 2 more and then we have 3 and 2 more and we get 5 ...
Q: And if you do not add anything?
R: I will get the same.
Q: And it is.
R: odd
Q: Why?
R: (Laughs).
Q: So you know that is odd. Why?
R: Because it is as I start the word.
Q: Because at the beginning, how many "as" do you have?
R: One.

C.1.4 Interview with Felipe

Q1:
R1: It is a place where you can find all the words of a language.
Q2:
R: In the dictionary.
Q3:
R: Because they are in alphabetical order.
Q4:
R: That they are organized in a special way.
Q5:
Q6:
R: I check the beginning letter, with "g", there is an alphabetical order then, more or less I know that it is in the middle upwards, then I check which letter I am, for example: If I open to the "e", I know that "g" is after, that is the order, I go forwards, I go on checking and if I advance too much, I come backwards and so on, until I find "g" and after that I search.
Q: After finding the "g" ?
R: I must find the "a".
Q: And to do so, what do you do?
R: The same.
Q7:
R: Yes.

P8:
R: Using this method you can find it.
Q: What does this method have that is so special for you to find it?
R: ...
Q: (We use the 2nd method.)
R: We spend three years.
Q: Then, why with the other method you find it more quickly?
R: Because I go on discarding pages, when I go forwards, and you don't need all these ones.
Q: Then, in each search, what happens with the dictionary?
R: It becomes smaller.

Simulation

R: Form list from W1 to W20 ... (following, he does it well)
W1 to W16
W8 to W16 (Here he made a mistake and we used it for Q17, see below (*luna* is W9[2])
W11 to W8 (he corrects)
Q17:
R: Directly.
Q: What makes you determine it?
R: I discard the others, make the dictionary smaller.
Q: Making it smaller in what way? Because if I am here, according to what you have said (when he got wrong), I made it smaller, but what was going on? (Here is the error).
R: *luna* wasn't in the ones I don't discard.
Q18:
R: Yes.
Q'18:
R: The word you are looking at, remains the same, the list and result vary.
Q: What do we do with the list?
R: The previous list is always greater than the following.(Observe that this is related to the previous error)
Q: And what do I do with the list?
R: You choose a word.
Q: And what else?
R: Define relation ... that remains the same.
Q19:
R: I tell him the same as to the robot, first that he opens the dictionary, that he reads one word and he defines the relation, that is what he tells me if the letter with which he started is before or after than the one he wants to find and he tells me that if it is after (he passes by) I ask him to choose another that is before and in

[2]In the collective class we explained the importance of this property that Andres expressed so well and in which Felipe made an error. We explained that always exists in those problems properties that holds in each application of the method and that are called invariants

this way successively.
Q: "This way successively", what does it mean?
R: Can I set an example?
Q: Yes.
R: (He does it with an example) ... until.
Q: Every time he goes forwards or backwards he does something completely different from what he had been doing?
R: Yes.
(And here now he says that yes, that it is different !!!).
Q: why?
R: Yes, because if he passes by he doesn't have to go forwards ... He discards them.
Q: Yes but what he is going to do with "d" is different from what he did with "g"?
R: Oh! You want to tell me if he is going to look up the letter again ... No, he discards it. (He is stuck with the variation of the list. That is to say, he looks at the list and not at the action. That is why he says that it is different. Here, see what Piaget verified about the stability in a relation and in the coordination of actions).
Q: And to look up "a" , what does he do?
R: He doesn't come back here.
Q: Then, what does he do?
R: He discards.
Q: Yes, for the list to be different but on that different list, his action, what is he going to do ?
R: It is the same.
Q: Are you sure?
R: Yes.
Second activity.

Q1:
R: (He puts one with "ba").
Q: No, these are the rules.
Q2:
Q3:
Q4:
Q: how do you know if this is a word?
R: First I would count the digits.
Q: Sorry?
P2: I count the number of letters and if it is odd, it isn't a word. If it is even ... I check the extremes of the word and they have to be equal, if they are different, it is not a word, then I go on with the letter inwards and they have to be equal. When you find this one, that is different, it's not a word.
Q6: (Many times).
Q7:
R: ...
Q: (I repeat).
R: (He marks it well).
Q: So, what is the little symbol in each word?

R: The previous word.
Q: So, what does the * have to accomplish?
R: It has to be a word.
Q: This sequence, for example: why isn't it a word?
R: Because the * is not a word.
Q12:, Q'12: , Q"12:
R: Yes.
Q: How?
R: For example: the * has a certain number of "as", do I have to tell you the number of "as" of the whole word?
Q: Yes.
R: For example, if a * has 5 "as", I tell you that one word has 5 "as" or 6 "as".
Q: 6?
R: No, 7 "as".
Q: What does it depend on that it has 5 or 7 "as"?
R: 7 if the last rule was the second and ...
Q: Write it down for any word in which the * has X "as".
R: If the last rule is the second ... (He writes it well).
Q13: (While he is calculating, he puts 2+ Xa and always uses Xa with which he gets mixed up, which is typical of parameter passings. At a moment he puts X-2 with which I make him watch that in what he wrote it never says "- ". We try with another word. By putting 2+ Xa we see that he should work out Xa (The number of "as" of the *) and puts 2+ Xa again, so he uses the same Xa for all different * and he himself says: "Calling them always the same I get confused", (see below). When I ask him why he put minus, he says: "because I was finding (the number of "as" of) the smallest word that was remaining. Spatial with - : as the word diminishes, he believe that the operation has to be subtraction.)
Q: But what does what you wrote say? Do you agree that it is useful for any word? (We try another case ... When trying again and writing at the same time, he says: "This * is this one, this is this", and he goes on marking and says: "It will have fewer than the others".)
Q: Do not forget that you mark this little symbol in order to count the words (marking), "as" in this one. (This oversight is in fact that he focused on one relation, he doesn't coordinate them.)
R: Oh! Of course, by calling always the same I get confused, that is, this Xa is different from this one, so I would have put ... (he wants to change what he wrote).
Q: No, what you wrote is OK, read it.
R: (He reads it).
Q: What happens is that by applying what you wrote many times, each time you use the same name for different things.
R: Yes (Here he is stuck on the referential transparency, Xa is always Xa, which is true, but in the case of recursion it means "it is always the current argument". It is one of the greatest difficulties in recursion).
Q: And the last one, how many "as" does it have?
R: 1.
Q: So, what do we do at the end?

R: Xa = 1.

C.1.5 Interview with Gimena

Q1:
R: It is a book in which the words of a language are in alphabetical order.
Q2:
R: In the dictionary.
Q3:
R: Due to the alphabetical order.
Q5:
R: (She does it while speaking).
Q6:
R: The first thing I did was to remember where the letter "g" was in the dictionary.
Q: Let's search again.
R: I opened the dictionary and remember the first letter and I start to search.
Q: Why do you go that way, which letter are you in?
Q: In "e".
Q: So?
R: I know that "g" is after "e". So I have to turn over the pages to find it.
Q: And what happens with all this part of the dictionary?
R: I do not care. I still look for it.
Q: Now why are you going to the other side?
R: Because I know that "h" goes after "g", then I passed by in my search ...
Q: And what happens with this part of the dictionary?
R: I do not care either at this moment. So I search, I start to look for the "a" and found *gato*.
Q: Describe it to me again.
R: Good ... I open the dictionary (she laughs) and go to a certain letter, then I know, for example, at this moment it is letter "c". I know that from this part of the dictionary backwards, nothing is worth and that I have to go on advancing to be able to find letter "g", then I go on, I come across "h", I know it is ahead from "g" so I have to go back again and look for *gato* and I find it.
Q7:
R: Yes, because the dictionary has all the words from the language.
Q8: If I tell you the following: "Look for the word *gato*" (the 2nd method).
R: In what sense?
Q: Look for the word *gato*.
R: (She opens it again).
Q: I close it and look for it again.
R: (She laughs).
Q: What is the difference between the two methods? (We do it several times so she understands the 2nd method).
Q: With my method I open the dictionary and search where I suppose that it is going to be.

Q: That is, it is not in the whole dictionary.
R: No.
Q: And with my method?
R: Yes, I guess so, because you, when I open the dictionary, tell me on what page the word *gato* will be.
Q: No, I do not tell you that it is going to be on that page, when you open the dictionary, I, without saying anything, close it up and ask you to search again. What is the difference between the two methods?
R: Suddenly, it closes up and I cannot search ...
Q: And where do you start looking again?
R: eh...eh... approximately in the letter ... In the dictionary.
Q: And with your method where do you start searching again?
R: In letter "g".
Q: So, is it in the whole dictionary?
R: No.
Q: Where in the dictionary?
R: On a page.
Q: Page or?
R: In a part of the dictionary.
Q: What happened to the dictionary when you are still searching?
R: I don't understand the question.
P7: So you open the dictionary, there you have some words and where do you go on searching?
R: Forwards or backwards.
Q: Is it the whole dictionary what you are using?
R: No, no, no. They are specific pages.
Q: So ...
R: It becomes shorter, I need only ...
Q: When you found the word. What happened?
R: It depends on what for I was searching, I will give it certain ...
Q: With respect to what you were telling me about the shortening of the dictionary.
R: Oh! The other words become totally useless, that is, not useless but in this search I do not need them, I only need that determined word.

Simulation

R: When you look for the word *gato* in the dictionary, we start with the whole dictionary. We are going to search in the same way. We do not know where the word *luna* is.
R: To form a list from W1 to W20.
(She does the task perfectly well).
Q17:
R: It influences in the same way that with the dictionary, that as we know the order, one starts getting to it.
Q: Here you had from W1 to W20, we got this result and then why do we put here from W14 to W7 and not for example from W17 to W19 ...
R: Because I suppose more or less that from W7 to W14, knowing the place where

letter "l" is, I guessed that it was a non-intermediate ... point, near the place where I could find the word *luna*.
(Observe that what she says, have nothing to do with what she does and besides she says: "non-intermediate point", I think that preconceived notions are playing their role here).
Q18:
R: Yes.
Q: Determine.
R: Equal things would be a procedure, but the procedure in itself, one of the things that vary in the procedure is that what W to what W I want, for example: to choose the word. (observe that this is very good!). And another thing that varies is that when I say "choose word", the words are different and have a different order respect to the word I am looking for. For example: we found a word that was >, another <.
Q19:
R: If one knows the alphabet ... the first thing I would do is that you say it up to letter "l" for example (here she goes on with confusing things but it is that she had not understood that it is in the dictionary and not in the list).
(Explanation and we start again.)
Q19:
R: First open the dictionary and we find with letter x, then we see which the 1st letter of the word we are looking for is, for example *gato*. Then I would ask him "you know towards what place in the dictionary you can find it, knowing that you are stuck on letter x. This is to say letter x is before or after letter "g". And in this way we go on reasoning until we can find the word.
Q: That would be a 1st time, suppose he tells you. I know that *gato* is after letter x , what do I do now?
R: If you know that it is after that letter, we will locate the letter with which *gato* begins. Once we have it, let's see which letter follows in *gato*, then, supposing that ... the "g" and the "a", he is going to start looking for that way.
Q: Does he do something completely different to what he had been doing?
R: No, not at all. That is, he does the same as I did or what any person does.
Q: And in respect to what he himself does, let's say, to get to letter "g" he does a certain thing. With the "a" does he do something different?
R: No, he does the same.

Second activity.

Q1, Q3, Q4:
R: (the 1st word that she writes is wrong, she puts "aba", something like that).
Q: (I ask her to read the rules.)
R: Oh! "a" at the end and at the beginning (while reading rule 2, she corrects. She writes others applying the rules 2, then 3, etc.)
Q: Why are these ones from the language?
R: Because they follow the rules of construction of words (she repeats.)
Q2: (I give her a particular sequence.)
R: (She immediately realizes that it is not).

Q7 reformulated: If this word you formed it making use of the rules, and in the rules the little symbol appears *, this means that in your word the * is something. Do you agree?
R: Yes.
Q: Well, who is * in this word?
R: "ab".
Q: Are you sure?
R: I could not tell, I don't know.
Q: I want you to make sure. How could you be sure?
R: Good question ... (She smiles).
Q: This word, with which rule did you form it?
R: With the 3rd and also with the 1st , knowing that "ab" is a word. (They do not see that every time I apply one rule, I have a new word).
Q: What was the last rule you applied?
R: The 3rd.
Q: If the 3rd rule was the last one you applied and you tell me that "ab" is the little symbol, put "ab" and apply the 3rd rule. (She does it).
Q: Is it the same?
R: No, it is not the same.
Q: So, applying the 3rd rule when "ab" is the little symbol, we obtain this word. Then, which is the little symbol here? (In the original one).
R: The little symbol would be all this. (She does it correctly).
Q: In this one?
R: (She does it right).
Q: In this one?
R: (She does it right).
Q: Could the little symbol be anything?
R: It depends if it follows the rules to form words.
Q: Then, could it be whatever?
R: No, it can't.
Q: What does it need to accomplish?
R: The 2nd and 3rd rule.
Q: In this case which is the *?
R: "ab" is a word. So it is the first rule, the 3 rules must be accomplished.
Q: In this case? (I show her one that is not).
R: In this case it is not a word, there cannot be *.
Q: And what must occur with *?
R: It must accomplish the rules.
Q12:
R: (Particular cases).
Q"12: Knowing the number of "as" of the little symbol?
R: No, I cannot determine the number of "as" because I do not know if that word finishes with "a".
Q: If you have the entire word and the number of "as" of *.
R: I know the number of total letters. (We review the particular cases, this is important. It is important to see many particular cases).

Q: Which is the relation between the number of "as" of one and the other (whole word and *).
R: Oh! It can be +2 or -2.
Q: We review cases.
R: Oh! + 2 or the total number of "as". It could be that the number of "as" of the word is the same number of "as" as there are in *.
Q: What does it depend on, that is one thing or another?
R: It depends on the rules.
(I ask her to write it down and she does it).
Q: I give you this word and ask you that using what you wrote, work out the number of "as". (She does it speaking all the time. She observes that it is the number of as of * (3rd rule)).
Q: What do we do then?
R: We count the ones from the little symbol.
Q: Very good, come on.
R: One, two, ...
Q: NO, NO, using what you wrote.
R: I can't.
Q: How come? Count the "as" from the little symbol using what you wrote.
R: Oh! (Thinking) Yes, again, we do it with *, yes, yes, yes, I understand now, in this case, if I cross this out, I get it this word, then it will be + 2.
Q: Write it. (She put 2 +, following the previous one).
R: Then I do it again and I get 2 + , I do it again and I get 2 + and now I start to do the same again and I get to the first.
Q: And then?
R: I cannot decompose "ab" any more.
Q: But how many "as" does it have?
R: One.
Q: So, what do we do now?
R: + 1. (She puts 2+2+2+1=*)
(We count and check that it is right).
Q: What was it that you did?
R: Counting the number of "as".
Q: So it is not = *, = what?
R: Yes, yes, = the number of "as".
Q: In what way have you to modify what you wrote in order to cover all the words, including "ab"?
R: (She does it well).

C.1.6 Interview with Ignacio

Q1:
R: A book that has the meaning of the words we use.
Q2:
R: In the dictionary.

Q3:
R: Because it's done in a way that has ... it is in alphabetical order.
Q4:
R: An order.
Q5:
R: (He looks it up).
Q6:
R: First knowing the alphabet I started to search, I found letter "h" so that I know that it is one before, and found letter "g", and then I look up "ga" and I found "ge"; I know that "a" is before, so I came back and looked up "gat" and I found *gato*.
Q7:
R: Knowing that it is in the dictionary, yes.
Q8:
R: Due to the order it has.
Q9: Good, let's use now another method to find the word *gato* (2nd method). We close the dictionary and I ask you to look up the word *gato*.
R: (He does it).
Q: Is the word *gato* there? (Where he opened).
R: No.
Q: Close it and look up *gato* again.
R: (He does it).
Q: Is it there?
R: No. (It is repeated some times).
Q: With this method the dictionary keeps its order and however, do you think that in this way we'll find the word *gato* easily?
R: No.
Q: Instead, with your method?
R: I find it fast.
Q: What is the difference between the two methods?
R: One is safe and the other isn't.
Q: Why?
R: Because there is an order.
Q: No, in the 2nd method the order remains in the dictionary.
R: But I didn't follow it.
(Observe that this is an example of directing the attention of the student from the object towards his actions).
Q: What does "following the order" mean? Let's use your method again.
R: (He does it.)
Q: Why are you browsing in that direction?
R: Because there is letter "e" and I know that letter "g" is after.
Q: What happens then with all this part of the dictionary?
R: I discard it. I know that here "g" is not and I discard it.
Q: Let's go on.
R: (Goes on searching and he passes by).
Q: What do you do with all this part of the dictionary?
R: Also, I know that it is not in this part.

Q: What happens with the dictionary?
R: There are parts which I know that it is not there.
Q: And what do you do with them?
R: I discard them.
Q: So what happens to the dictionary?
R: It becomes smaller.

Simulation

With the instructions, he tells me: "Choose one at random".
Q: I don't know where.
R: Oh! I have to tell you one.
Q: No, I know how to choose at random, but I don't know where.
R: Oh! OK! Choose a word from W1 to W20.
Q: That's not the instruction, check well.
R: Form list W1 to W20 (he does it well and we continue up the end. While he is writing he says: "they're always the same steps" and when I ask him to repeat, he says: "the 1st step changes the W and the others are the same".)
Q17:
R: If the chosen word is minor than the one you are looking for, all the ones that are behind are eliminated and if the chosen word is greater, you discard the ones before.
Q'17: What does "eliminate" mean?
R: Subtract to the previous list the words that go before or after.
Q18:
R: Yes.
Q'18:
R: Choose words and define relations remain the same. Form list between which words and what word varies.
Q: That is, does the action change? What I do in the 1st one.
R: No, the action doesn't change.
Q: What is it that changes?
R: The limit, the place where you have to work.
Q19:
R: I would say that you check the 1st letter, open the dictionary and focus on the 1st letter of the word that you found in the dictionary. If that one is before, you should turn over pages until you find the one you're looking for and once you found it, you do the same with the 2nd, the 3rd ...
Q: When do I stop?
R: When you achieve in finding the word.

Second activity

Q1:
R: (Ignacio writes words)
Q2:
R: Yes.
Q3: How?

R: Because I have the rules.
Q7:
R: Unless I see that this is a word ... (he changes the rules).
Q: No, the rules cannot be changed.
R: All this? (all the word)
Q: No. Why not?
R: Because I start from the **outside** towards **inside**, don't I?
(Observe that when he works out the "as", actually he does it the other way around, I mean from **inside** to **outside**).
R: I see that here they're "as", these ones are "bs" and then I come across that this one is "a" and this one is "b" , then it doesn't follow the rules.
Q4:
R: According to the rules, I had a previous one and I put "bs" on the sides.
Q: And this one?
R: I put "as".
Q: How did you form this word?
R: I put an "a" to the previous one, to the sides.
Q: Which rule did you use?
R: The 2nd.
Q: Good, the rule talks about the little symbol * and you tell me that this one was formed using the rule, so I ask you: Which is the little symbol in this word?
R: This one (he does it well).
Q: And in this one.
R: This one (he does it well). (In one that is not a word, he says immediately "it is not a word").
Q9:
R: To keep.
Q: To keep what?
R: The order ... it has to be a word.
Q11:
R: In the letters of the extremes the 1st one is "b", the 2nd is "a" (he tries to keep an order that is not so).
Q: Why is the 2nd an "a"? If it is like this, suppose.
R: Oh, you mean ... ? Oh, it is possible, OK, OK. (He understands that there is no order to apply the rules, observe that right now, after having solved the previous things well, however, he hadn't understood the explanation of the rules correctly). Oh! here I used the 2nd rule.
Q12: Q'12: Q"12:
R: It depends, I have to know besides the "as" the little symbol has, what rules I'm using. If I use the 2nd (an "a" at the beginning and the end) I know that I have to add 2 and with the 3rd rule (I add a "b" at the beginning and at the end) I get to the same.
Q: write it down, (he writes it).
Q13:
R: (I don't know why he starts with "ab" and he says "the number of "as" of the asterisk" and then he goes using the rules and he gets 1+2+?. What is wrong is

that he begins as if the * were "ab".

C.1.7 Interview with Iván

Q1:
R: A dictionary is a book which has the meanings of all the words.
Q2:
R: In the dictionary.
Q3:
R: Because it has an order.
Q4:
R: That they have an order (alphabetical)
Q5:
Q6:
R: I opened the dictionary, I looked at the letter I was on that page and if "g" was before, I went backwards and if it was after I went forwards until finding letter "g" and then the same with the "a", with "t" and "o".
Q7:
R: Yes.
Q8:
R: Because if it is in the dictionary I will find it.
Q: Why?
R: Because ... because it has this order ...
Q: Good, the dictionary has an order. Now we are going to use another way with the dictionary that has that order (we do it with the 2nd method).
Q: What is the difference between the two methods?
R: Your method is more difficult to find a word, more improbable.
Q: Why?
R: With the 1st one you start discarding parts and I will find it more quickly, not with the other.
Q: Then, what happens with the dictionary?
R: It becomes smaller.
Q: And when you found the word?
R: The search is over.
Q: How many words does the dictionary have?
R: One.

Simulation

W1 to W20, W1 to W15, W5 to W15, W5 to W12 (he does it well)
Q17:
R: The number of words decreases.
Q18:
R: The formation of the list starts shortening, choose words is always present and define the relation between also, and the result varies.
Q: And with respect to what the robot does?

R: It varies, yes, because it has fewer words to choose. (Observe the difficulty they have on separating the action from its application).
Q: Let's look at what you wrote and let's try to determine what the robot does. Does the robot do different things each time?
R: No.
Q19:
R: That it went searching letter by letter of the word ... that is it took the 1st letter of the word and searched it alphabetically.
Q: What does that mean? The kid opens the dictionary ...
R: Reads one word and checks if the 1st letter of the word he is looking for is > than the 1st letter of the word he found, goes backwards in the dictionary.
Q: What is he still doing?
R: In this way he looks for the 1st letter, then he does the same but with the 2nd letter and the others.

Second activity

Q1: and Q2:
R: Yes.
Q: Is this a word or not?
R: It is not.
Q: Why not?
R: Because ... (takes time). A word is always formed from another one and a word will always have equal extremes. Then, I have to see if this is a word. I know that it could be a word because it has both extremes the same. Then if I took the extremes out and I observe if the new extremes are equal and like this successively. When I get to two letters, if they are ab, it is a word.
Q: In this case?
R: No.
Q: Let's see, doing what you said? (He does it OK).
Q5:
R: Rule 2.
Q: Rule 2 uses the *.
Q7:
R: (He does it well with many.)
Q: How would you define the *?
R: It is the word without extremes.
Q12:, Q'12:, Q"12:
R: We should know what rule we are using from the * to form the new word.
Q: How do we go about, then?
R: (He says it and writes it well).
Q13:
R: This is the little symbol, I have to count the number of "as" there are.
Q: Yes.
R: Do I count? (he means "count one by one")
Q: No, use what you wrote.
R: But ... To use what I wrote, I have to know the number of "as" that the little

symbol has.
Q: What you wrote, helps you to work out the number of "as" of any word?
R: If I know the number of "as" the little symbol has, yes.
Q: Is the * a word?
R: Yes.
Q: Then use what you wrote to work out the number of "as" of the *.
R: ... I put the first word ... ("ab")
Q: No. You are not using what you wrote. The quantity of "as" of this whole word is going to be, according to what you wrote ...
R: One more ?
Q: Why one more?
R: Because we have the first word.
Q: But in what you wrote you do not talk of the first word at all.
R: But to know the number of this *, I have to know the number of previous * and the previous one, and the previous one.
Q: Exactly. Do that, using what you wrote for this word.
R: It will be ... depends on the case I have.
Q: Yes, which rule did you use to form it?
R: With number 2.
Q: So?
R: I add up 2 to the supposed number of "as" of the *.
Q: Good, then it is 2+.
R: 2+ and now the other word was formed with the * of the other word and the third rule and if I use this rule, I know that it has the same number of as (2+0). The new * is 2+ the previous and the previous is 2+* and now I cannot use what I wrote. (He got to "ab" and he is right, observe that Gimena got to the same and said "I cannot decompose this one").
R: So it is ?
(We add up and verify that the result is right).
Q: It's really true that when you get to "ab" you cannot use what you have written, so what would be missing?
R: What happens is that the 1st word is not formed from the previous one.
Q: No, and then what would we put?
R: If the word is the 1st , the number of "as" is 1.
Q: Would it possible not to get to the first word?
R: I always get to it because from the 1st all the others are formed.

C.1.8 Interview with Juan Andrés

Q1:
R: It is a book where we can find the meanings of the words of a language.
Q2:
R: In the dictionary.
Q3:
R: Because the words are in alphabetical order.

Q4:
R: The dictionary is something to consult.
Q3:
R: Because they are in order.
Q4:
R: Due to its practicality.
Q: With respect to the words.
R: Because they are in order.
Q5: Q6:
R: First I opened the dictionary in a section that seemed near to "g" and went in order always visualizing the alphabet in my mind. I went up to "g" and then with the 2nd letter, which is the "a", and then the "t" until finding the word.
Q7:, Q8:
R: Because if it is a good dictionary we would have to be certain that if the word is there we will find it.
Q: Why?
R: Because that is the function of the dictionary .
(Observe how he refers to the object when in fact it is his action what guarantees that he finds it and it is what I try to make him see with the 2nd method).
Q: I show you another way (we do it).
Q: What do you think about this way of looking for the word *gato*?
R: It is not rational.
Q: What is the difference between the 2 methods?
R: That this is and random method.
Q: What can happen?
R: That we never find it.
Q: Which is then the difference?
R: ...
Q: Let's look for *gato* again with your method.
R: (He does it).
Q: Why are you going that way?
R: Because I am in letter "b".
Q: And what happens with all this? (From the pages where he is to the beginning).
R: I know that "g" is not there, because it has an order.
Q: Where do you continue searching then?
R: That way (he goes on until he passes by the word) and says "we passed by" ...
Q: And now why are you going backwards?
R: Because letter "g" is before letter "h".
Q: And what happens with all this? (after "h").
R: I know that it is not there.
Q: Where will you search now?
R: Here, in between.
Q: and what is that part with respect to the whole dictionary?
R: Bounded. (Observe how he tries to use words that are not from the ordinary language but from Math).
Q: According to the size?

R: Much shorter, smaller.
Q: Then, while you are searching, what happens with the dictionary?
R: It becomes smaller.
Q: When you find the word, what does the dictionary become?
R: One word.

Simulation

He tells me: "Choose a word".
Q:I do not know where to do that.
R: Form the list from W1 to W20.
Q:
Q15 $>$ *luna*...
R: (He goes on well up to the end, he says "we bound now", and "we put again" and "we give command 2".
Q: W6 $<$ *luna*, (etc. Up to W9 $=$ *luna*)
R: As soon as the word is found, the search is over.
Q17:
R: The way in which it influences is that in each search we do, we achieve the shortening of the field, it is bounding the search.
Q: What does the result determine?
R: A shortening of ... (he does not say the list but he talks about "commands". Almost no student referred to "the list" although I introduced this name myself).
Q18:
R: Yes.
Q'18:
R: What is always the same is the order of commands and what changes is the results.
Q: And with respect to what the robot does?
R: The actions are always the same.
Q19:
R: That the word that he wants to find will be in order according to the alphabet that he knows. Then, what he has to search first is the beginning letter of the word. After having found the first letter, he can start with the 2nd. Proceeding also in the same way.
Q: The kid is learning to search ...
Then, when he opens the dictionary, what does he have to do to find the 1st letter?
R: A relation between what he is seeing and what he wants to find and according to this relation he "operates" forwards or backwards.
Q: And when he finds the first letter?
R: He operates again using the same order.

Second activity

Q1:
R: (after writing some words) I have run out of words.
(He had not understood that the rules can be applied many times)
Q: Are these words?

R: Yes.
Q: Then why do you say that there are no more? What can you continue doing with them?
R: I can go on adding (He does it).
Q: (The rules are applied repeatedly).
Q2:
R: Yes.
Q: (I give him one that is).
R: Yes it is.
Q: Why?
R: I start from the center outwards (that is he takes "ab" and "aaba", etc).
Q: (I give him one that is not).
R: It is not.
Q: For a sequence to be a word, what do we observe?
R: The extremes have to be always equal and then .. (He notices that something else is missing) ... from the nucleus of the word the successive letters in front and at the back have to be equal.
Q: How did you form this word?
R: I took the nucleus.
Q: In fact you did not do so (because he formed it directly from a previous one that was not "ab").
R: I took this one and added ...
Q: With respect to the rules?
R: Rule 2.
Q: And in this one?
R: Rule 3.
Q7: Q8: (He does it well)
Q9:
R: If it is always a word, it has to be a word.
Q: Can we know the number of "as" of any word?
R: I think so (Observe that I do not say: "knowing the number of as of the *).
R: We can count them, as long as it is not very big.
Q: Exactly, the matter is for any word, as big as we want.
R: ...
Q: (I give him the clue of *).
Q: With respect to the number of as of *, how many does the whole word have?
R: 2+.
Q: Always?
R: No, or none, say or the same.
(He writes it).
Q13:
R: (He starts pointing to "ab").
Q: Using what you wrote ... Where does what you wrote talk about "ab"?
R: First I have to know if it is a word ...
Q: We already know that.
R: (He reads his rule but marks as * "ab" !!).

Q: Which is the * of this word?
R: To start with, I've chosen it. ("ab")
Q: Let's see, what was the * in each word?
(We look the previous cases over).
(Here the confusion is that there is the need to clarify that what counts is the rule that was applied **at last**).
Q: Do you agree that the * is a word?
R: Yes.
Q: And what you wrote is useful to any word?
R: Yes.
Q: Then, use what you wrote to work out the number of "as" (he is doing it).
R: This would be * (he does it right) and then up to get to "ab".
Q: How many "as" does it have?
R: One, let's add up 1.
Q: In fact this is not within what you wrote (about "ab").
R: Yes, here we would not know what to do.
Q: Then, what would you add?
R: (He puts ab=1).
Q: Actually, it would be: if the word is ab, it has 1 a.
(He checks that the result is OK, (it always happens)).

C.1.9 Interview with Laura

Q1:
R: It is a book where all the answers are, that is ... what you need.
Q: What is it that there is in the dictionary?
R: All the words with which we speak.
Q2:
R: In the dictionary.
Q3:
R: Because it is ordered alphabetically.
Q5:
R: (she does it)
Q6:
R: I search around where the letter is in the alphabet, if it is at the beginning or at the end and ...
Q: Again (opens the dictionary). Why are you going forwards?
R: Because "g" is after.
Q: What do you do with all this part of the dictionary?
R: I skip it, it doesn't help.
Q: Why are you going backwards now?
R: Because of the vowels, "e" is after "a".
Q: Then, what happens with this part of the dictionary?
R: It is after, it is not there, let's say (I suppose she wants to say the "a" will not be there).

Q: Let's recap, after opening the dictionary and decide that you should go forwards, what did you do with the first part of the dictionary?
R: I got rid of it.
Q: And then, going backwards, what did you do?
R: I also eliminated it.
Q7:
R: Yes.
Q8:
R: Because it is a word.
Q: (we do it with 2nd method). What is difference between the two methods?
R: In the 1st one you start eliminating little by little and with the 2nd you discard very soon.
Q: What happens in the 2nd?
R: If you do not open on that page (where *gato* is) you have to start again.
Q: Do you think that you will find the word with the 2nd method?
R: There is almost no possibility.
Q: Why?
R: Because you eliminate little by little.
Q15:
Q16:
R: (she does it well)

Simulation

Q: Which would the 1st instruction be?
R: To form a list from W7 to W17.
Q: And now?
R: To choose one word.
Q: I choose, and now?
R: Define the relation.
Q: Exactly, I compare *luna* with *día* and I give you the result. What do I do now?
R: A new search.
Q: What would it be like?
R: Between W7 and W15.
Q: What do I do now?
R: Choose word.
Q: And now?
R: Define the relation.
Q: I compare it and this is the result.
R: (She writes it).
Q: And now?
R: Form a list from W7 to W10 ... choose words, ... define relation.
R: Writes the result... (she does it).
Q: And now?
R: Form the list from W7 to W9 ... choose words ... define the relation ...
R: (She writes the result that in this case is W9=*luna*).
Q: And now?

R: It is over.
Q17:
R: You know where *luna* is approximately and you know if the following word you will search is before or after, if it is > or <. (Observe that she performs the action perfectly, however, the last answer is completely wrong with respect to what she does).
Q: How does it influence on the Ws that you put? For example, why did you put here W7 to W17 and here you put W7 to W15?
R: Because as W16 was > *luna* , I eliminated it.
Q: So what can you do with the list with each result?
R: To make a shorter list.
Q18:
R: Yes.
Q'18:
R: The result varies, the search.
Q: What do you mean with the search?
R: That you see, the search is shorter.
Q: Where do you see that? What varies what?
R: The list varies.
Q: And what things remain the same?
R: Choosing the word and defining the relation.
Q: Some other thing ? What I do with the list?
R: It does not vary.
Q: Do I do different things?
R: No, the same is always done.

Second activity

Q1: Q2:
R: This does not.
Q: Why not?
R: Because it has to be repeated. So far it is OK, but here it is wrong, here there is a "b", here there has to be a "b" but there is an "a".
Q: Why are your words part of the language?
R: Because they follow the rules.
Q6:
R: (she does it well).
Q7:
R: "ab".
Q: If that is the little symbol and you use the 2nd rule, what would the word be like?
R: Like this.
Q: And is it the word?
R: No.
Q: Then which is the * in this word?
R: What it is in the middle.
Q: Read me the 2nd rule.

R: (She reads it wrong saying an "a" is a word instead of a*a is a word).
Q: an "a"?
R: a*a. (She marks it right).
Q: And in this one?
R: This (she marks it wrong).
Q: That?
R: Not from the second rule. From the 3rd ... (That is, she thinks that the little symbol is something different in each rule, then, she believes that the little symbol depends on the rule).
Q: On all the rules? (I explain to her again, making her make new words again and see the rules).
Q: Which is the little symbol in this word?
R: This one (she does it right).
Q: Can the little symbol be anything?
R: As long as it has the "as" and the "bs".
Q: That is ?
R: That it has the rules.
Q: In this case for example: Which would be the little symbol?
R: This one.
Q: And does that little symbol follow the rules?
R: No.
Q: Why not?
R: Because it is wrong, there is "a" and "b".
Q: Would you say that this word is not from the language? Why?
R: Because it is not a word.
Q: Who?
R: The little symbol.
Q: So, to know whether a word belongs to the language, what do we watch?
R: The little symbol.
Q: What exactly?
R: If it is correct, if it follows the rules.
Q12: (Perfect particular cases).
Q'12:
R: Yes.
Q: How?
R: Because * is in the middle.
Q: Let's count again (we come back to particular cases).
Q: Given a certain word, how is the amount of "as" with respect to the number of "as" of the little symbol?
R: It will always have 2 more.
Q: Always? In this case ?
R: Oh, it can have equal or greater, or 2 more.
Q: Knowing the quantity of "as" from *, work out the quantity of "as" of the whole word.
R: If the 2nd rule is accomplished, I add 2, otherwise it is what contains *.
Q: Write it down.

Q13:
R: I look for *, we count the ones that the little symbol has, ...
Q: Always using your writing.
R: I add 2.
Q: Which is the * now?
R: This one.
Q: Which rule does it follow?
R: The 2nd.
Q: Then?
R: I add 2.
Q: In this *.
R: It does not follow the 2nd rule so I do not put anything.
(Goes on like this until "ab").
Q: And the 1st?
R: I add up 1.
Q: Is that your result?
R: 7.
Q: Is it OK?
R: Yes.

C.1.10 Interview with Nicolás

Q1:
R: A book where I can find words which I do not know their meanings.
Q2:
R: In the dictionary.
Q3:
R: Because there is an alphabetical order in the dictionary.
Q4:
R: That I can "create" a word, I can go filtering it.
Q5: (Everybody finds it fast).
Q6:
R: First, I opened the dictionary working out the place of letter "g" , that is a little before the middle of the dictionary, then, when finding letter "g", I know the letter following is "a", which is the first letter in the alphabet.
Then, it has to be at the beginning within "g" and same successively.
Q: If you open the dictionary and do not open in the letter "g"? What do you do?
R: It is automatic, I mean if the letter I opened on is after letter "g" in the dictionary, I start to go backwards and if it is before "g", I start to go forwards.
Q7:
R: Yes.
Q8:
R: Because I always get the same method to look it up, I always apply the same method by which ... everybody uses it for the dictionary.
Q: And that method you apply and that everybody uses, why does it guarantee you

that you will find it?
R: If it is in the dictionary?
Q: Yes.
R: ...
Q: What does this method have that guarantees you find it?
R: It never fails.
Q: Why?
R: ... because ... because it holds all letters in the dictionary, I mean I always ... never will get lost ... as long as I search one ... then I will always follow the same order until I find the word, by each letter I go on searching.
Q: I propose another method. (we do it with 2nd method).
R: Yes, of course, we never finish.
Q: Which is then the difference between this method and the other?
R: I based myself on the principles (he is thinking of his own actions now), that is, I open the dictionary and I based myself on what I find at first sight, then I start checking if I have to go forwards or backwards.
Q: Suppose you go forwards, what happens to the rest?
R: I discard it.
Q: Then, what happens with the dictionary while you are looking up?
R: I start discarding until I find the word I am looking for.
Q: Then, what happens to the dictionary while you are searching?
R: They are being eliminated, it becomes reduced.

Simulation

R: Form list from W5 to W12. Choose ... Define ... Form list from W5 to W10. We are reducing again in another fashion.
Q: Why in another fashion?
R: Different from the dictionary.
Q: Why?
R: Because I cannot see the words here.
Q: Because you are using a robot.
R: Form list from W8 to W10 ... (we go on until the end).
Q17: Do you agree that the result influences in the later search?
R: Of course, I was based on that.
Q: How? Let's see.
R: It starts bounding.
Q: What is it that is bounded?
R: The amount of paper.
Q: What have we called that?
R: The list.
Q18:
R: It varies the endings of the list, what keeps the same is to choose a word at random and the relation will be always the same. (it's wrong). It will always be $>$ or $<$ but the relation is the same (in fact he doesn't mean that).
Q: And according to what the robot does?
R: He does the same, he is methodical. What he starts observing is that if the word

is before or after in the dictionary.
Q: That was done in the first search, what about the second?
R: If he did it in the second, he did it in the third.
Q: So?
R: He observes if the relation is $>$ or $<$.
Q: And according to that, does he do something completely different?
R: No (surprised, laughing).
Q: What does he do then?
R: The same, he is methodical.
Q19:
R: First I would ask him to look up the first letter of the word in the dictionary. That is, he should look up the words that start with that letter, then he should look up the second ...
Q: So he should look up this letter in the dictionary, what its meaning is. Suppose the kid opens the dictionary.
R: Opens the dictionary and goes to find the word *gato* for example. Well, he should look for letter "g" ...
Q: What do you mean by "looking up letter "g" " in the dictionary, he will not understand because you are teaching him how to look up.
R: The dictionary order is the alphabetical one. First he has to locate the letter of the word, letter "g" , in the dictionary.
Q: So, he opens the dictionary and ... ?
R: If the letter he finds is after, he will have turn over pages forwards, otherwise backwards. We look it up. Then he will have to check the second letter ...
Q: And do something completely different?
R: No, the same, only in the section of letter "g". Within the little bundle of letter "g" look up letter "a".
Q: What does it mean "look up letter "a" "?
R: Start to do the same, only in the section of "g".

Second activity

Q1:
Q4:
R: I formed with the previous one and an "a" at the beginning and an "a" at the end.
Q: Given any sequence of "as" and "bs", can we determine if it is a word from the language?
R: Yes.
Q: How? With this sequence for example (I give him one).
R: It is not from the language.
Q: Why not?
R: Here there is one that accomplishes.
Q: How did you do it?
R: When starting in the middle of the word, knowing that "ab" must be followed ... This one, if it is very long we will have lo follow another procedure. Afterwards, we start extending until we get to where there is an "a" and "a" ... We know that

there is an "a" at the beginning, so it is OK.
Q7:
R: It is a * in the middle that would be this, then an "a" and an "a" (he does not do it well).
Q7: (Again).
R: What it is in the middle, (he writes "ab", it is wrong).
Q: (We look attentively how he formed a word until he discovers * is not "ab").
R: Of course, you are right the * is all this (he does it right).
Q: Can the * be anything?
R: The asterisk has to follow the rules.
(We see the sequence again that was not and we determine that it is not a word because there is a * that does not follow the rules).
Q"12:
R: No. (Whenever I forget to say "knowing the number of "as" of *" they say "no" immediately and if I do not forget and say it, they say "yes" immediately).
Q"12: (Adding the above condition).
R: Yes.
Q: How?
R: It will always be a relation between the number of "as" of the * and the number of "as" of the word. If the word starts with "a" , it will have 2 "as" more than the *.
Q: And if not?
R: The number of * will be the number of "as".
Q: (Write it down).
R: With words?
Q: Yes.
R: (doubts, but he writes it). (Then he uses it to count the "as" of a word, putting 2 + * + * + 2 + * +... (and says when he gets to "ab" "and so long").
Q: What happens now?
R: ...
Q: You are counting the number of "as".
R: Plus one (he does it well).
Q: (As that sequence is of 2+ and * , I ask him: "Do you agree on the fact that adding the numbers of that sequence you have the number of "as" of the word?").
R: Yes.
Q: How many are they? (He counts, checks, the result is OK.)
Q: Do you agree on the fact that whatever word I give you, you can work out the "as" using what you wrote?
R: Yes.

C.1.11 Interview with Sergio

Q1:
R: A place where answers are found to questions. Words, that is, answers of meaning to questions.

Q2:
R: In the dictionary.
Q3:
R: Because it has an order.
Q4:
R: The order.
Q5:
R: (He looks it up and finds it).
Q6:
R: Searching by letter, 1st I searched by the letter, then I did it by the 2nd letter.
Q: What is the meaning of search? The dictionary is closed.
R: First I opened the dictionary, looked for a section, a number of letters, first from "a" to "g", first I looked for the second letter, then the third and in this way ...
Q7:
R: Yes.
Q8:
R: Because ... knowing that it is ... What a good question ... If I know that it is there and I'm going to look for it ... it is supposed that it is over.
Q: Yes, but why do you find it?
R: Because you look it up.
Q: Another method (we perform the 2nd method).
R: Oh! Because I come back to follow the same process. Always, I mean, first I look for a section and afterwards ... a smaller one.
Q: Why do you find a word, then?
R: I take a sector of the book and I start reducing it until I find it.
Q: So, you told me ?
R: I look for letter by letter.
Q: And what happens to the dictionary while you are searching?
R: It becomes smaller.

Simulation

(When I tell him about the possible results, he says: well, it could also happen that the word is not there. I explain that I'm not studying that possibility now).
R: From the list W5 to W11.
Q: Why did you choose this list?
R: At random, because It seemed to me.
(We do all the process. He puts the "minor" word).
R: W7 to W11, now I change the parameter again, W7 to W10, ...
Q17:
R: In reducing, in changing the parameter and the size.
Q: You said 1st reducing and then changing ?
R: Of course, because in fact I would be lucky to get just to the parameter but I could have made a mistake with the distance ... (He refers to the fact that he choses 1st W5 to W11 at random).
Q: How could you make sure that it would always reduce?
R: Taking the greater first.

Q18:
R: Yes, because I was right, if I had taken W1 to W8 I wouldn't have gotten to the result.
Q: And there?
R: I would have changed and there it would always reduce.
(He went on thinking like that so I asked Q18 again, it is as if he hadn't heard (interesting)).
Q18 (again):
R: Yes.
Q'18: R: The word we are looking up keeps the same, the words that are > or < change.
Q: Look at what you wrote, what remains the same?
R: The method.
Q19:
R: That I started searching (the same of what I did).
Q: Let's see ?
R: That I started to look for the region, first ...
Q: You tell me the same of what I did, but one thing is to know how to do it and a very different one is to know how to explain it correctly.
R: First I ask him to look for the 1st letter and then the 2nd and so on.
Q: In this way he will not know how to find.
R: Of course, if they know the alphabet.
Q: Knows the alphabet, but that doesn't mean that they know what to do when you tell them "Look up".
R: Opens the dictionary and start to see that the letters are > ... no, no ... they're <, I mean they're before ... that they open the dictionary and start searching at the beginning ... (he laughs, because he knows it is wrong).
Q: That is not the method you used. Let's see, opens the dictionary and what do they do?
R: They look for the 1st letter.. Oh, of course, because first he has to know to do to find the first letter. Sure, he has to find the first letter.
Q: What does it mean? Suppose that he opens on a page with words with "c".
R: Oh, of course that if it is before the letter he searches, he has to go forwards, he has to find the letter.
Q: What does he do while going forwards?
R: Takes letters out, diminishes the group, goes on, gets to the letter and ... he has to look for the other letters.
Q: And that means he has to do something completely different from what he has already done.
R: No, he does the same.

Second activity

Q1:
R: They're all the same, no, it would be like this, no ...
Q2:
R: Yes.

Q: Can you show me? This one.
R: No.
Q: Why?
R: Because here there's an "a-a" "b-b", but here there's "a-b" so that it is not a word.
Q: This is not a word, but the whole word, why isn't it a word?
R: Because if it is not a word the one inside, the one outside, the total, can't be a word.
Q: What happens in this part in the middle?
R: That is not a word, it doesn't follow the rules.
Q4:
R: With words are already made.
Q: For example this one, with which rule?
R: 1st the 1, then the 3, after the 2.
Q: The last one you applied is the second?
R: Yes.
Q: Check that rule 2 talks about the *, who's * in this word?
R: (Quickly and without doubting) "ab" (it's wrong).
Q: Are you sure?
R: Yes.
(Observe that before he said and did everything right and this is wrong).
Q: You say that the * is "ab" and the last rule you applied is number 2. Let's put the * and apply the second rule. What word we get? Is it the same?
R: No, it is not the same, but it is a word. (Here I would have to see that the students understand that once the * is determined, we get to the word, by the application of one of the rules once. This means that the relation between the * and the word is one only rule once. Behind is the concept of reducing in "one step" or in "several steps").
Q: Yes but I asked you who the * is in this word and you told me "ab".
R: Oh, yes.
Q: And that you applied rule number 2, then what would it have to happen?
R: Be the same word.
Q: And is it?
R: No.
Q: Then, what is it that's wrong?
R: That the * is not "ab".
Q: Which is it then?
R: "babb".
Q: Mark it. (He does it well).
Q: Here, who is ?
R: With respect to the last rule?
Q: Yes, yes. It is always with respect to the last rule. (he does it well).
Q: Here? (It is done many times).
Q: Can the * be whatever?
R: No, it has to be a word (immediate) it has to follow the rules.
Q12:, Q'12:, Q"12:

R: Yes.
Q: How?
R: Adding 2 to each word.
Q: To each one?
R: To each word that uses the 2nd rule.
Q: And if you don't use the 2nd rule?
R: I don't add anything.
Q: Write it down.
R: ...
Q: How do you do to work out the "as" of a word?
(He does it) (It is not very well written, though. We try to improve the writing).
Q13:
R: The same.
Q: The same as what?
R: As the previous word.
Q: Which is the previous word? Mark it.
Q: How many "as" are there here? Using what you wrote (in the *).
R: 2 fewer.
Q: What did you write?
R: 2 more.
Q: So we put 2 +.
R: But why 2 +?
Q: You told me.
R: Oh no, yes, yes. We put 2 more than the *.
Q: And which is the *?
R: (He laughs) 2 +.
Q: Yes.
R: 2 + ...
R: And 1 +.
Q: Let me see if it is OK.
(He does it correctly).

C.1.12 Interview with Sofía

Q1:
R: A book that has the definitions of the words of the language.
Q2:
R: In the dictionary.
Q3:
R: Because you know all of them are there and what's more, you have an order to find them.
Q4:
R: The organization, the order.
Q5:
R: (She does it).

Q6: I was checking on the letter according to the order that I have in my mind, first, the first letter, secondly the second and so on.
Q'6: Doing what?
R: Discarding the ones didn't help, and looking up the words with the letters I was searching.
Q"6: What do you do with those letters ... Let's see, open the dictionary, what letter are you ...
R: "c".
Q: So what do you do?
R: I turn pages over.
Q: Where to?
R: Forwards.
Q: Why?
R: Because it is the alphabetical order.
Q: What do you do both with the word you're looking up (*gato*) and the ones that are on the page you opened?
R: I located them. (Observe what difficult is to realize that there is a comparison there, no student did it).
Q: And so you go forwards (she passes by). There, why do you come backwards?
R: Because I passed by, because I am looking for the location of the 1st letter in the alphabet.
Q: And now, check where you have your hand (she has a part of the dictionary in her hand), where will you go on searching *gato*?
R: Where I have it marked because I know that it's neither before nor after.
Q7:
R: Yes.
Q8:
R: Because I trust myself, because yes, (she laughs).
Q: You'll find it rather quickly or you will take the whole day.
R: Quick.
Q: Why?
R: Because I have practice.
Q: Now we are going to use another method (2nd method). Open the dictionary (she opens it).
Q:Is *gato* there?
R: No. (We do it again many times).
Q: What do you think of this method with respect to the other?
R: It's much more difficult.
Q: And what is the difference between them?
Q: That in the 1st one I start marking between which and which and I can shorten the limits and in the other it is at random.
Q: Why are you sure that with your method you will find it and quickly?
R: I have fewer bounds, fewer limits.
Q: What happens with the group of words in which you search?
R: They start shortening.

Simulation

R: Choose word.
Q: I don't know where.
R: Oh! First you have to know where so form list from 1 to 20.
(Now it's hard for her to say: "choose word" and when she says she says it again, so there are two instructions "choose word").
Q: Another? And what shall I do with the first one?
R: Form list ...
Q: I still don't know what to do with the first word ... (explanation a little bit more until she defines the result and puts it).
Q17:
R: ...
Q: What happens with that?
R: It modifies, it becomes smaller.
Q: According to?
R: The results you start obtaining.
Q18:
R: Yes.
Q'18:
R: The procedure is the same.
Q: And what varies?
R: The results.
Q: What do you refer to "procedure"?
R: That 1st you have to form, then choose and after define. It is always the same.
Q: When does that procedure finish?
R: When the relation I write is =.
Q19:
R: Take the dictionary and choose one word and then with respect to that one, start looking if it is > or is < until finding the 1st letter and after you found the 1st letter, you have to look for the 2nd , in order as well, and go on until you find the whole word.
Q: Is what he does to find the 1st letter totally different from what he did to find the 1st?
R: It's the same but now with another letter.

Second activity

Q1 and Q2
(We work with concrete words and she uses the * the same as Gabi and Cecilia (they're the same age), they wrote a*a as a particular word.)
Q: that what you wrote is **any** word and I want a concrete word.
R: With any letter?
Q: No, with letters "a" and "b" and following the rules.
R: Like this. (She does it well).
Q: Is this one?
R: Let's see (she wrote "a", "ab", "ababa") no, they are 3 words together.
Q: It's not a word because who would be the * in this word?

R: This (she does it wrong, marks one of the "abs").
Q: Only this?
R: ...
Q: Who's the * in each word?
R: "ab".
Q: No ...
R: What is in the middle between "a" and "a".
Q: So here, who would be the *?
R: (She does it wrong).
Q: Repeat what you've just said: who is the little symbol?
R: What is between the 2 "as".
Q: Any 2 "as"?
R: The two last ones.
Q: So, which is the * in this word?
R: What is equidistant to the 2 "as".
Q: You didn't say that before ...
R: What is in the middle (she marks the 3 "abs" separately as 3 *).
Q: There you have 3 little symbols, but I ask by one (she marks it well).
Q: Which of the two solutions is the correct one according to the rules?
R: This one (she marks it right).
Q: Why?
R: Because ... "ab" is a word ...
(She does it wrong again and says "the * is "ab" between the 1st and the last "as"". We read the rules again, she reads them and says "here it would be a*a and here as well (and she does it well).
Q: So, what does it have to happen with the *.
R: It has to be a word.
Q: And is it, in this case?
R: "ab" is a word and "ab" is a word, look "abab".
Q: But is there any rule that tells you that all that is a word?
(Note: how difficult it is for her to see this).
R: That it is the same, maybe, that it is equal (she is inventing a rule herself).
Q: Check in the rules: is "abab" a word?
R: It has to finish with "a" ... no, it is not a word.
Q: If this is not a word, could this one be? ("aababa").
R: Not, because the * has to be a word.
Q: And is it in this case?
R: No.
Q: Why?
R: Because what it is inside has to be a word ("aababa" is not a word).
(Now she seems to have understood that "what it is inside" is the * and it is not a word in this case).
Q2:
R: Yes.
Q: How?
R: I have to check if what is between the first and the last "a" is a word.

Q: Let's do it (with a sequence). (Now she finds something of the form b*b and says that it is not a word "what is inside" because it has to start with "a" and finish with "a". Amazing: she is not able to coordinate her reading of the 3 rules and stayed stuck on the 2nd rule that was the one worked in the last steps).
Q: Why? (We read the rules again).
R: Now yes, it can start and finish with "b" and here she does it (in fact she's looking a * wrongly).
Q: Finishing with "b"?
R: No, but here as there's a little symbol, I check "b" and then ...
Q: Oh! But let's see in this sequence, which would the * be?
R: "ab" (it is wrong). Oh! It would be all of it (she corrects).
Q: Up to where?
R: (She marks it right).
Q: So, what does it have to happen with that * ?
R: It has to be a word.
Q: Then, how do we do to know if it is a word or not?
R: ...
Q: What did you do before to know if the total was a word, that is, what we did already?
R: I marked inwards.
Q: Good. Now you have another sequence what would you do to know if it is a word?
R: I mark again (she does it right but with doubts).
Q: What does it have to happen with that?
R: It has to be a word.
Q: And what do we do to know if it is so?
R: Another little symbol?
Q: Exactly, which is it?
R: (She does it well, doubting).
Q: Well and this little symbol, is it a word?
R: It starts with "b" and finishes with "a".
Q: So?
R: It's not a word.
Q: So, what do we do to know whether a sequence of "as" and "bs" is a word?
R: I'm considering 1st a little symbol and see if it is a word or not, if it is a word I go on each time shortening more, marking each time a little symbol ... (she says it with doubts), if it's a word I can go on searching ...
(It is amazing how this is so hard for her and for Gabi too (review)).
Q1: (She doesn't apply the rules, she does it peculiar and the result is wrong).
Q: Let's see if you realize why it's wrong.
R: Here, it doesn't accomplish.
Q: Write another (she does it well).
Q12:, Q'12:
R: (is ok)
Q"12:
R: ... No, I guess that not.

(We see the whole relation again, the word and the *. It's hard for her to understand the question).
Q"12:
R: One you know you already have?
Q: And you know something more, you know the number of "as" of the *. (We do some more cases). What relation is there between the number of "as" of the word and the number of "as" of the * ?
R: That if it is "a" they're 2- and if it is "b" it is the same.
Q: Write it down.
(She writes).
Q: Is there any other case?
R: No.
Q: And the 1st word?
R: It has 1.
Q: Work it out using what you wrote ? but we change 2- by 2+.
R: (She does it right saying: Now again inwards and then here. No, here again (she does it well) and finally 1
Q: Is the result OK?
R: Yes.
Q: Do you think that with this method if I give you any word you can count the number of "as" without counting one by one?
R: Yes, but anyway it is hard.
Q: In the collective class we will see how we can do to make it easier.

C.2 Problems Worked on the Third Interview

In the collective class (Chapter 4) three exercises were posed and in the third interviews students' solutions investigated. In the following the exercises and the complete interviews are included.
Exercise 1
Recall the trees of exercise 1 of the written material handed out before, (see section 3.1.2). The case in which red and blue lights have to be placed at random is considered. A mathematical definition of the set of trees is given by the following rules:

1) R is a tree.
2) B is a tree.[3]
3) if a_1 is a tree and a_2 is a tree, then a_1 R a_2 is a tree.
4) if a_1 is a tree and a_2 is a tree, then a_1 B a_2 is a tree.
5) Only the elements generated by application of these rules a finite number of steps are trees.

Graphically, the trees can be represented as follows:

[3] R stands for "red light" and B stands for "blue light"

```
      R              B
     / \            / \
   a1   a2        a1   a2
```

Define a method of counting the red lights (R) of any tree.
Exercise 2
How would you search for the word *gato* in a novel in which you know that the word *gato*[4] exists?

Exercise 3
1) How can the algorithm of searching a word in a dictionary be modified to take into account the case in which the word is not in the dictionary?
2) Given a dictionary and a word that is not in it, define a method of including the word in the dictionary at the right place.

C.2.1 Third Interview

C.2.2 Interview with Andrés

Exercise 1

I: What does the first rule say?
A: (He reads it).
I: So which are the first trees?
A: R and A. (He reads the other rules).
I: Which ones would a1 and a2 be?
A: The same way we have R and A we would have that a1 is a tree and a2 is a tree.
I: No, R is a1 and A is a2 , for example. (I already see what happened to him, he confuses the statement "R is a tree" with the conditional "if a1 is a tree ...". So I have to make them realize that we have a tree when we state it, for instance, in "R is a tree" or in "a1Ra2 is a tree", but we never say "a1 is a tree" but "if a1 is a tree", and it's quite different).
I: a1 and a2 represent any tree. (Here we have the problem of variables and the constants, I mean, it is a problem of type. Observe that when in a program language we define a type, we use for instance Const N | Var Char, with which we're distinguishing variables and constants **explicitly**. While in Math this distinction is **implicitly** made, that is we know that 2 is a constant and x is a variable. In this case we have that R is a constant and a1 is a variable, but the student doesn't know how to make that distinction (And it's OK they don't know, in fact)).
I: How would you construct a new tree with rule 3?
A: Could it be any combination?
I: If a1 is R and a2 is A, for example?
A: (He does it well).
I: Now form new trees, applying the rules again (it is important to remark that the rules can be applied more than once. It's difficult for him, indeed. He thinks and

[4] "gato" is the Spanish word for "cat"

thinks, but at the end he does it well. We do several cases using rules 3 and 4).
I: Do you think there's any analogy with the exercise of the words?
A: Yes, what it has to do with the little symbol, whenever you finish making a tree you put it as the little symbol ...
I: Do you remember how we counted the "as"? We had defined a method, here there's yours (I give it to him and he reads it).
I: How can we count the red lights?
A: Counting the number of times that we used rules 3 and 4.
I: Correct, if rule 3 was used, how many R does the tree have?
A: It has one more, the R that is ... (he stops, that is, it is difficult to see what happens when the structure is binary, or say, not unary.
I: That is where?
A: In the tree you have already formed.
I: Good, repeat that please.
A: It has one R plus the number of R that is in the tree.
I: In which tree?
A: In a1 or in a2 .
I: Exactly, only that it is **and** in a2 , and if it was formed with rule 4?
A: It's the number of R that is in a1 and in a2 .
I: Very good, using that you have already said, let's count the red lights of this tree.
A: (He does it correctly, quickly and secure. We check.)
I: To finish up with this exercise, could you write the complete method? (He does it).
I: (I explain that from this a definition of the mathematical function can be constructed and I name the method using the name he put in the recursive applications and tell him that it can be done because the trees are built with the rules. He seems understand this perfectly).
I: Why do we always finish?
A: Because there's a limited number of trees.
I: How are the a1 and a2 with respect to the original tree?
A: The same way ...
I: Yes, but what is it that makes the method end?
A: They're smaller each time.
I: Correct. (I remind him of the dictionary and we also see it with the factorial function.)

Exercise 2

I: How do we locate a word in its place?
A: I search by the method we had already seen ...
I: You don't have to search, you have a word.
A: Anyway, I have to look for the place in which it could be.
I: How do you look for that place? Let's see: if we represent the dictionary like this (I write it).
A: I choose any word, say word5 and determine if it's greater or less than word' (the one we want to locate) and if it's greater for example I go a little backwards and choose another, for example word3 .

I: So, you repeat the same once you determine that word' < word5 ?
A: I shorten the place where I'm looking for and make another search and then I choose another, word3, and I get that it's word3 < word', so I shorten the place where I'm looking for the following to word3 and the previous to word5 .
I: How do you locate it between word3 and word5 ?
A: I go on shortening like this until I get two consecutive words, that is, my space is reduced to 2 words.
I: And then?
A: Word', goes in between.
I: Very good. And the other part (the line of kids).
A: Yes, the same way.
I: And the 3rd ?
A: I have to read the whole novel because it isn't in order to look up words.
I: The whole novel always?
A: I would have to read the whole novel if the word *gato* is the last of the novel.
I: And if not?
A: If it's in the middle, up to the middle, it depends on where it is.
I: Anyway, do you think there are similarities with the search in the dictionary?
A: Yes, the space becomes shorter.

C.2.3 Interview with Felipe

Exercise 1

(Felipe asked for more explanations about the statement in exercise 1, and time to work it out before having the interview, which was given.)
F: Count-R from the tree = if the tree is R, it has 1 red light, on the other hand, if the tree has R * R, then it is 2+ count-R(*) (Observe that he changes the rules!). But if it has R * A or A * R then it is 1+ count-R(*). In the case of A * A so count-R(*). What was complicated was the way to write a tree, that is, if the tree is, for instance:

```
      R
     / \
    A   R
   / \ / \
  R  A A  R
```

this can be RAARARR or leaving blanks, (he wrote something like that). It can be both ways, or leaving blanks, I don't know when the tree finishes.
I: So you flatten the tree. That's why it isn't convenient for you, it's better to do it like this and you add to the sides.
I: (We study his method of flattening and it seems to be OK, we see that it is recursive, etc. The problem of Felipe is that he does NOT use the rules that I gave him, although he thinks about them, as his following explanation shows). Did you find any similarity?
F: Yes, sure, it is recursive, you define for a base tree, and then you put the cases

that the person could have done, if they put R to the right and A to the left or 2R or 2A, etc. It could be complicated if you put R, A and yellow for example.
(Observe that he thinks of changing the problem too!)
I: Well, but there the trees would be different. (I explain that he changes the rules and I explain the method using the rules that I give him. Observe that Felipe, who achieved establishing a method, finds it very difficult too, the same as the other students, to understand my rules and build trees with rules 3 and 4, considering a1 and a2. Finally, Felipe solves the exercise with my method).

Exercise 2

F: (Reads the statement). Given the word, you act as if it were in the dictionary.
I: You can represent the dictionary. (He puts word1 < word2 , etc).
F: I choose one (word3), I define the relation with word' and we do it again, until it is between 2. That is, with word1, you get it greater and with word2 you get it less (he refers to the extremes of the list, in fact), then, the word goes there.
I: Very good, let's see part 2.
F: Yes.
I: What is it that changes?
F: Instead of letters, they're the heights.
I: Good, the relation of order is another, and what is it that keeps the same?
F: I do the same procedure until getting to two kids where one is taller and the other is shorter (than the one to be located).
I: Good, part 3?
F: I read until I find it.
I: Exactly, this is what you do, but what does that mean? Given that, in fact, we aren't interested in reading the novel.
F: I take the 1st word, define the relation, you will get to the fact that it isn't, then you go on with the following ...
I: Is there anything that is diminishing?
F: Yes, sure, the list becomes smaller.
I: Exactly.
F: When you work with a computer, do you do that?
I: Yes, of course. (I explain to him about the searches, binary and sequential).

Questions posed in the collective class

I: Which is the best way to choose it?
F: Dividing in equal parts.
I: Why?
F: Because in that way we discard the biggest part.
I: Exact.
(The other question) I: Does this algorithm work if the word is not in the dictionary?
F: It's going to have a reduced list to two words, in which yours isn't going to be, it's going to be greater than one and less than the other.
I: You're telling me what there has to be added for the method to work?
F: Yes.

C.2.4 Interview with Gimena

Exercise 2

G: I start with exercise 2.
(She brought something written and reads): "to place a word in a dictionary, a word that is not in it, we have to respect the order of the alphabet. Supose that we want to locate the word "gato" (if it is not in the dictionary), the first is to see with which letter it begins. When this letter is located, I have to see the following letter and so on until the word is located in the right place."
Comments: the conceptualization of Gimena is very week: she refers to the same example used before and has not conceptualized the comparing method.
(About the kids): Yes I can, because its the same procedure but with their height.
(3rd question): The only possibility to find the word *gato* in a novel is by reading it, because the words don't have any hierarchical order.
I: You aren't interested in reading the novel but in finding the word *gato*. How would you do it?
G: The thing is that I have no way out, I have to read it ...
I: Read it, what does it mean?
G: To start searching the word *gato* ...
I: Where to start?
G: By the beginning of the novel ...
I: You mean you read the first word ?
G: Yes, yes, up to the last one, if the last one were *gato*, I mean until I find the word *gato*.
I: Does this method have any similarities with the one of looking up in the dictionary?
G: Yes and no.
I: Let's see.
G: Yes because you also search the word, only that in the dictionary you know that you will find it because as it is more accurately in an order ... you know you might find it, whereas in a novel ...
I: In the novel, using the method you say of searching by the 1st, the 2nd, etc, will you find it?
G: No, not because in the novel the words are not in order.
I: But you know that the word is there and you look for the 1st, the 2nd ...
G: Yes, yes, I'll find it, but it will be more difficult.
(Observe that the examples are relevant: both the dictionary and the novel have a certain order relation, that is to say are structures, which is more evident in the dictionary).
I: Is there something that goes diminishing?
G: Yes, yes, the number of words, I do not care about the words I have already read, until finding *gato* and the search is over. Yes, distances become shorter. (She refers to each time she searches she's "closer" the word *gato*)
I: So what similarity is there between the two searches while you're looking for, before finding the word?
G: In each of the two, it is the same.

Exercise 1

(She hadn't understood that what I put as an example is an example. I mean these students do not understand what an example is. I explain it to her, she understands).
I: Let's read the rules. (She reads them).
I: Which are then the least trees we can form?
G: R and A.
I: Afterwards?
G: (She reads rule 3 and tells me: "I thought that from R, a1 and a2 went out ...
I: Yes.
G: ...
I: Apply the 3rd rule with R as a1 and A as a2.
G: ...
(She doesn't understand the roles of a1 and a2. In fact, the rule is not understood, that is to say that the 3rd rule gives the R of the root and the 4rd gives the A of the root. That is to say, as pointed out before, they think that R (or A), *are given* and a1 and a2 **are obtained** and it is exactly the contrary).
I: You tell me that R is a tree and that A is a tree and according to the 3rd rule these two can be a1 and a2 , can't they?
G: Yes.
I: So a1Ra2 is a tree; what is a new tree like?
G: (She does it well).
I: Now we have new trees, form a new one applying rule 4.
G: (Reads the rule slowly and thinking, and does it well). Yes, yes, now I understand.
I: Now we have to count the red lights of any tree.
G: ...
I: What did you focus on, in the language exercise, to count the "as"?
G: On the rules.
I: Exactly.
G: R is a tree ...
I: If the rule is that one, how many red lights are there?
G: One, A is a tree, according to the 2nd rule, there's 0 red light. (She writes it). After (reads rule 3, always slowly) and puts "according to rule 3 there's one red ... and after ... 2 red lights ..."
I: No, let's remember how it was in the language (I show her what she wrote in the language exercise). Which would be the * in this case?
G: R and A.
I: No, R and A that are the least trees, to whom do they correspond?
G: To the 1st word ("ab").
I: So, to whom does the * correspond?
G: (Reads rule 3 again)... a1 and a2 are the letters that we add ...
I: No, check well. (We look at the words again). How did you form this tree? Who plays the role of the *?
G: Oh! a1 and a2 ! Yes, yes, yes, now yes.
(From this point she goes fast. That is, understanding the role of a1, a2, *, is something essential). According to rule 3 there is a red light more than the ones we

already had.
I: Where?
G: What there is in a1 and what there is in a2 . It would be then 1+2.
I: That wasn't what you told me.
(Observe how amazing is that she gets mixed up with what he said with 1+2 !! Because of there are two things, she puts 1 + 2!).
G: 1+ the red lights of a1 + the red lights of a2.
(First she puts + a2 and I correct her, she understands immediately, she tells me "sure, a2 is a tree". Then she writes "according to rule 4: 0 + the red lights of a1 + the red lights of a2").
(She writes it immediately, but she gets confused, she tends to put only a1. Observe how she says it perfectly well, but she writes it wrongly, this is one of the greatest difficulties in conceptualizing).
I: Now we are going to use your method to count the red lights of this tree (I see that she has this difficulty: in the tree, she says "R is a tree" (the first rule) and looks **at the root** of a tree like:

```
          R
         / \
       A     R
      / \   / \
     A   R R   A
```

that is to say, a tree that is not formed by application of rule 1 last. It is difficult for her to see that the tree has a1 and a2. It is difficult for her to see how the tree was formed. Actually, the difficulty lies in seeing that what counts is the rule applied last, which was found in the language exercise, as well.)
Finally, she goes applying the method and gets to good results, with some errors, above all because she puts A1, A2 instead of a1 and a2 and gets confused with A. (Observe what important is the role of the notation). Goes on saying "now I do the same, but this one is a1 and this is a2". When she finishes, says "now I make the total addition").
I: OK, does the method you used seem to you similar to the one we used in the previous exercises?
G: To the one of the words yes, totally, it was the one that helped me.

C.2.5 Interview with Ignacio

Exercise 1

Ig: Count the red lights ...
I: The rules are those. What trees would you have?
Ig: (He does it).
I: How did you form it, this one for example?
Ig: I put an A to the right and an R to the left.
I: You are inventing this rule. This one for example, which rule did you use to form it?
Ig: By rule 1, R is a tree and by rule 2, A is a tree ...

I: So, what are the first trees you can have?
Ig: A and R.
I: So, which rule did you use for this one?
Ig: I put tree A to the left of R and to the right I put tree R.
I: Which is the rule?
Ig: (He reads it well). It is rule 3.
I: And this other one?
Ig: Rule 4. (It's OK).
I: And this one?
Ig: Rule 2.
I: And this one?
Ig: Rule 1.
I: Very good. So, now apply rule 3 to form a new tree with this one and this one.
Ig: With rule 3 ... by rule 3 I have to add other trees to the right and left of R. Having a tree R, adding a tree to the left and a tree to the right.
I: Sure. Now I tell you, this is a tree and this is a tree, so apply rule 3 to form a new tree. (In spite of saying it perfectly well, he doesn't know how to do it directly, I have to guide him and so he does it).
Ig: Oh! (He does it well).
(We repeat it with rules 4 and 3 some more times) (He does it well).
I: Now we have to define a method to count R lights of any tree.
Ig: ...
I: Do you find any similarities between this and the language exercise?
Ig: Yes.
I: (I show him what he did before, in the language case).
Ig: What goes first is rule 1 or rule 2.
I: Yes, write it (he does it well).
Ig: Now I go down a branch and apply rules 1 and 2 again.
I: No ...
Ig: If the rule is 3 I count 1 for the one above and if the rule is 4 ...
I: You haven't finished with rule 3, that says (I repeat rule 3).
Ig: I take a1 as a tree and a2 as a tree.
I: Exactly, how do you write it?
Ig: 1+... I will go on using this rule (he gets confused).
I: Yes, but if the rule is 3, it is 1+ what?
Ig: Plus the trees below.
I: What is it we are counting?
Ig: The number of red lights.
I: So?
Ig: 1+ the number of red lights of the tree (He refers to a1).
I: Good, how do we call the tree?
Ig: a1. (He writes 1+ a1).
I: Write what you told me, that is, it is not the same to add a1 (which is a tree) as the red lights of a1 .
Ig: Yes, it's OK.
I: And what else?

Ig: Now I take a1 again as a tree ...
I: we didn't finish with this tree that is a1Ra2.
I: 1 + red lights of a1 + red lights of a2.
I: Very good. And if the rule is 4?
Ig: 0+ ... (he writes it well).
I: Does this method help you to any tree?
Ig: Yes, but if the rule is 1 ... (he gets confused because he wants to add in rule 1).
I: No, if the rule is 1, the tree is R, if the tree is R and something else, then you are not in rule 1.
Ig: Sure!
I: Well, then let's assign a name to it, that must be this one you already used, red-lights-of (see line above) and we use it for any tree t. Let's use it for this tree, for example (we take any one that he constructed before).
Ig: I start at the top.
I: This is the tree, what rule does it correspond to?
Ig: Rule 4.
I: Good, so?
(He does it well and we check it).
I: How are the trees you are applying the function?
Ig: Each time smaller.
I: Why does the method end?
Ig: By rule 1 and rule 2.

Exercise 2

I: We go to exercise 2.
Ig: (He reads part 1). It's similar to find words. Instead of looking up a word, I search the smallest possible ... I look for the less or equal ...
I: We are going to represent the dictionary like this (I write word1 < word2 < word3 < ... < wordn).
Ig: I choose any word, say word4 , if word4 is greater than my word, then I will locate my word from word1 to word4 .
I: And then?
Ig: Well, there I have another list and I choose another word, say word2 and I get less than, then I make a list again and I get word3 ... no, ... no (the problem he has is that he got to one word, but he corrects well).
Ig: I include word4 but I don't use it and I include word2 but I don't use it, then if word' < word3 , goes between word2 and word3 and if it greater, it goes between word3 and word4.
(His confusion derives from the fact that he doesn't realize that, according to his method, he needs two words to locate a word in the middle and, choosing one by one, he believes he only has one. He corrects himself fast and correctly).
I: Why does your method finish?
Ig: The list is getting shorter and shorter.
I: Well, the 2nd question.
Ig: Yes.
I: If we have a collection of objects ordered according to their size and we want to

insert one?
Ig: It is the same method.
(Now he reads question 3.)
Ig: I would read the novel.
I: All the novel?
Ig: Until finding it.
I: Oh! Then it is not always all the novel. What does "reading it" mean? I think you are interested in finding the word *gato*.
Ig: Do I look for words at random until I find *gato*?
I: Why at random and not beginning with the 1st one?
Ig: Yes, I start by the 1st, then the 2nd, (words) the 3rd, the 4th, the 5th, the 6th.
I: What does this search have as similarities and differences with a search in the dictionary?
Ig: Similarities, the list becomes shorter, the number of words in which I search. And differences ...
I: Did you start by the first word in the dictionary?
Ig: No, because there is an order in the dictionary and there isn't one in the novel.
I: Doesn't the novel have any order?
Ig: Concerning the words, no. (He refers to alphabetical).
I: What order does it have?
Ig: A coherent order concerning the reading.

Questions posed in the collective class

I: In the search in the dictionary, what is it that we always know?
Ig: In the dictionary the "a" is always the first and the last one is "z". So if I am going to look up a word with l, I must look up around the middle.
I: And in the other cases? What do we always know of each list?
Ig: The first and the last ones.
I: So, knowing the first and the last, is there any way to choose the word but not at random?
Ig: We choose the one in the middle.
I: Why is it more efficient for all the cases?
Ig: It's a way of shortening the way.
I: Exactly.
(We work on the last question now.We have the material from the collective class.)
I: In this function we had defined, we had this condition (word=word') that helps when the word is in the dictionary. What happens if the word is not there?
Ig: I would finish without finding the word.
I: With the same definition?
Ig: There will be two words between which you can't choose any.
I: Look at what happens with word=word'
Ig: I don't know, it isn't going to work.
I: Exactly, what would we have to add for it to work?
Ig: If there are two consecutive words and with the two ones, one is greater and the other is less, then the word is not in the dictionary.
I: Perfect.

C.2.6 Interview with Iván

(Iván brought the exercises done. The first one was OK).
Iv: I did the second one by writing. (He means in Spanish).

Exercise 1

I: Did you find difficulties or was it easy?
Iv: It was easy.
I: Did you have a look at what we had done before?
Iv: Yes, yes, I saw it.
I: What similarities did you find with what we had done previously?
Iv: They are all recurrent (he means recursive).
I: And why?
Iv: We based on ... it is like a chain, so as to know something, we based on the previous one until getting to one that you know it is known.
I: Can you do it with any case, for instance, could you do it with the real numbers?
Iv: No, because there is not previous to another one.
I: So, what conditions must the elements have?
Iv: That are finite.
I: What?
Iv: The set of elements.
I: No, the set not, because, observe that having the two first trees, you can always build another one.
Iv: The tree has finite letters.
I: Exactly, the elements are finite. And besides, what other feature is there?
Iv: That one can be made from the one before.
I: Very good. What do we use to define them?
Iv: Er... the rules.
I: We go on to the second.

Exercise 2

Iv: (He reads what he did, it is OK, he wrote it in Spanish).
I: What would the condition be then, that we should use in the method (by formalizing it).
Iv: Er... yes, the list... it would be, first word and last word, that's all.
I: Yes, it is a condition about the list... without using the searching method ... Does another method come to your mind?
Iv: ...
I: Suppose that this is the dictionary (I represent it).
Iv: I go comparing one by one.
I: From which one?
Iv: From the first one.
I: And when do you locate it?
Iv: I compare it with the first, if it is greater, I compare it with the second, and if it is greater, with the third and like this until it is less than the one I put in front.
I: Exactly.

Questions posed in the collective class

I: How can we choose words in a way that the algorithm is the most possibly efficient?
Iv: That the chosen word is in the middle.
I: Why is it the most efficient?
Iv: Because you discard a bigger piece.
I: What happens if the word is not in the dictionary?
Iv: You are not going to find it.
I: What should we do to take this case into account?
Iv: Er... if a list remains formed by two words and none of the two is the one, it is not there.
I: Very good. Let's see now the second part of exercise 2.
Iv: If it is the same (method).
I: What is it that changes?
Iv: The object, instead of locating words, we have to locate a child.
I: And what else? Do you compare the kids alphabetically?
Iv: The order.
I: The relation of order. And the third?
Iv: Reading each word and if the word is not *gato*, I get rid of it.
I: Do you find the word *gato* with that method?
Iv: Yes.
I: Why?
Iv: Because it is in the novel and I go on using fewer words, the novel starts becoming smaller. The best would be that you take the first word and then the second and after the following, etc. Not words at random.
I: Sure. Oh! I thought that you were telling me that.
Iv: Yes, yes, I say so, but it could also be that you take one word at random, you eliminate it, then you take another at random and so on.
I: Of course. Let's see, is the list in order?
Iv: It is not in alphabetical order.
I: Oh! But is there an order?
Iv: Yes, there is an order.

C.2.7 Interview with Juan Andrés

Exercise 1

(Juan Andrés has studied a course in imperative programming and his responses are evidently influenced by it).
JA: For the first exercise, I thought about it in the following way: I tried to define something that worked as a counter that counts the red lights. I did it with 2 methods. Firstly a step that would be to select the tree, then find the root of it and the third step would be if it is R we define R as R+1 and if the root is equal to A, then R=R and the 4th would be to come back to step 1. This method doesn't choose well ... Afterwards, I had this idea which is similar, up to the 3rd , it is identical, then the 4th step says "confirm the existence of branches", so, if branches exist and the root is R, then I define R as R+1 and if branches exist and the * is

A then R would be R+2 because from this A, 2 red ones go out, due to the way it was defined. (He does not use the given rules, but others. So, in fact, this does not make use. Anyway, we write it down so as to see if we get useful data).
I: With respect to these rules, they are rules of construction of the trees, did you use them to define the method?
JA: No ... I mean a1 and a2 would be the branches.
I: Correct.
JA: We saw it to see how they were constructed and in this case to know that R is already a tree, we also had to look here (rule 1).
I: According to what we saw previously, the exercises of the 1st interview and what I explained in the collective class, what could you say?
JA: For example, what is there that is similar?
I: Yes.
JA: It could be that one method that is applicable a number of times to get to a final result.
I: Exactly. With respect to exercise 2?

Exercise 2

JA: Well, to include a word in the dictionary, the first step is to choose a word from the list at random, we call it word1 and if word1 is less than the lost word, a new list of word1 to word last is defined and if it is greater, the list will be from word first to word1 and step 3 would to repeat it until the list is of a Wn to a Wn+1 and then the lost word would be between Wn and Wn+1 .
I: Did you try it to see if it worked?
JA: Yes, yes.
I: Let's see and what does this mean? (They wrote "inductive").
JA: Oh! Inductive means that it is repetitive.
I: OK, we call them recursive. And the other questions?
JA: (Line of kids): Yes, the height of the kids is a list in order, so it lets us have an order.
I: So, for what can we generalize the method?
JA: For any list that is in order.
I: And the other question?
JA: Well, I would number each word of the novel since the 1st word to the end. And *gato* would be the word (he means the word to be searched). We would choose a word from the list, we call it word1 , so if word1 is equal to *gato*, then it is over and if word1 s not equal to *gato*, we eliminate it from the list and we did the same again.
I: You mean that you always take one out at random?
JA: Yes, at random, or ... ok, another method would be to begin with the first word, then the second, and so on ...
I: Exactly, this is another method.
JA: Yes, but with the first one, (he refers to the random method), the list starts diminishing ... well, in this one too. That is, now I realize that in the other too, if we didn't always take out, it wouldn't finish.
I: Is there something similar with the dictionary?
JA: Well, the novel has to be ordered.

I: In fact, that order already exists.
(Discussion on the ordered and the non-ordered sets).
I: What other similarities are there?
JA: ...
I: Not with respect to the objects, but with respect to what you do.
JA. That are steps that are repeated, we are seeing different examples of something that can always be used.

Questions posed in the collective class

I: With respect to choosing a word when searching in the dictionary?
JA: Is that a clue for the robot?
I: Yes, a clue or a way to do it, for example, what does the robot know in all cases?
JA: The word he choses and the two extremes.
I: Exactly, can we take advantage that he knows the extremes to give it a concrete form, instead of doing it at random? (Perhaps a better question should be: which is the way in which what is discarded is the maximum in all cases?)
JA: That he chooses the extreme following the word.
I: Yes but if you have a word at the end or at the beginning, it is more ineffective than at random.
JA. Yes, sure. (He thinks ...) The center. The center is the best.
I: Exactly. The other question was what happens if the word is not in the dictionary.
JA: There is going to be a point in which a list will remain with extremes and empty (he means to only having the extremes), a list with 2 ... He is not going to know what to do .. the robot ... will have to choose only the extreme words.
I: Of course, how can we change the algorithm to include this case?
JA: Write a sentence that if word first and word last are next to one another, that is, if they do not have words in between, then there he can finish.

C.2.8 Interview with Laura

(She thought only the 2nd, the 1st one was difficult for her.)

Exercise 2

I: Let's see the 2nd then.
L: ... I don't know ... I check letter by letter and look for the words that are similar because of the letters and ... I put it there.
I: (I remind her of how we represented the dictionary: word1 $<$ word2 $<$ ... and I tell her that we have to locate word'.)
L: I go on searching between the words it should be.
I: What do you do to do so?
L: I look at the letters that it has.
I: Let's suppose we know what the result of comparing two words is.
L: Between the greater one and the one that is less than it.
I: If you have to put it there, so how do you do it?
L: I go on comparing them ...
I: Show me, for instance.
L: Word' with word3.

I: And then?
L: I see if it is greater or less and ...
I: How would the relation among these words be so that you put word' between word2 and word3 ?
L: Word' should be greater than word2 and less than word3.
I: How would we do to know that this is the place?
L: Comparing them.
I: When do you finish your comparison?
L: When I find that relation.
I: Good, we go on with exercise 2, next part.

L: (She reads about the kids and says yes.)
I: What similarity is there between the two things?
L: That it has to be greater than the one before and less than the one behind (She says it incorrectly, the same error than before, see line above, but thinks about it. Reads 3rd question).
I: Why can't we look for it as in the dictionary?
L: Because the words aren't in order.
I: OK, then how do we do it? First, will you find it?
L: Yes.
I: How would you do it?
L: I read the whole novel.
I: You aren't interested in the novel but in finding the word *gato*.
L: Yes, but as you don't know where it is, you have to read it from beginning to end ...
I: From beginning to end?
L: Well, until you find it.
I: Ahh! Then it might not be the end.
L: No.
I: Are there any similarities with searching in the dictionary? That is, we said that we found the word because the dictionary was reducing, and in this case?
L: You also start discarding words.
I: Which ones?
L: All the ones that are not *gato*.
I: But while you are reading?
L: You discard the ones you are reading.
I: So, what happens to the novel?
L: It becomes smaller as well.

Exercise 1

I: Good, from exercise 1 you didn't think anything?
L: No, I get confused.
I: Let's see. Let's read the rules (She reads them. We start constructing trees with different a1 and a2. It's very difficult for her, even when she gets to make a tree, say with the rule 3, she's not able of continuing making new ones with rules 3 and 4).
I: OK, now how would we count the red lights of any tree? How did we count the

"as" of a word in the language exercise? (I show what she did then and she reads it).
I: What would you focus on?
L: On the root ...
I: Yes, but who talk to you about the root?
L: The rules.
I: So? (We go on seeing according to the rules that R has 1, A has 0 and we get to rule 3. She doubts, we come back to the words and the *).
I: Who would be the little symbol in this case?
L: R.
I: (We see the language again). Who follows the role of "ab"?
L: R and A.
I: Correct, so who's the little symbol?
L: R.
I: No, we already said that it was "ab".
L: a1 and a2 (doubting seriously).
I: Very good, now suppose that we have a tree formed by rule 3, how do we count the red lights?
L: I would add ... for each R I would add another R.
I: For each R you count 1 and then what would you add up?
L: The R of a2 .
I: And what else?
L: And the R of a1 .
I: In the case of rule 4, what do we count?
L: The ones from a1 and the ones from a2 .
I: Let's use what you tell me to count the red lights of this tree (She does it, but doubts a lot).
I: How many did you arrive at?
L: 4.
I: And does it have 4?
L: Yes (We always check at the end, it is important).

C.2.9 Interview with Nicolás

Exercise 1

N: Exercise 1 ... the exercises are related to what we saw before, so I treated them with a certain link to that. In this case, I called the method "to locate".
I: That was the idea.
N: What I noticed in exercise 1 is that there is no order; in what we saw before, a pattern was followed between what R was and what A was, (he refers to the first material I handed out to the students for the selection, see section 3.1). In this case, it is much more open. So, what I did was, following these 4 steps (he refers to the rules), I analized the possibilities that could happen. Then, the cases are that from an R we get 2A or that we get an A and an R or that we get 2R. With the A it occurs the same. So, I had to define a method in order to count the lights R of any tree, I made a recursive method, as we had done before. From each light R could

happen that it is the only one and that the other two are A or there is an A and an R or that they are R and R, so the number of R would be 3. And from each blue one the same. I mean that it is the same but subtracting 1 to the possibilities. So, what I put is that ... to this I called *, (he wrote R+(2A), R+(A+R) and R+(2R) and called * to the expression between parentheses in each case) then when you see an R, what you do is adding 1, because this is that R plus what there is in the * and if you see an A, it is the same as * because with A you don't add anything.
I: Very good. Now, what you did for 3, is it valid for more?
N: Yes, it goes on opening.
I: Exactly. Now, you told me that you followed the rules and called * to this. In fact, if we look at the rules, we see that we have "the basis" (the root of the tree) and then it opens in such a way that what you called *, actually we have two * in the case of the trees: we have what opens to this side and what opens to this side (left and right).
N: You have to do the same with what opens, I mean it is **the same method for those two**.
I: Exactly.
N: This is the method that allows you to work out the lights R.

Exercise 2

N: I did something similar to the task of looking up in the dictionary but I called it "to locate", and I want to locate a word between two others. I defined a chosen word (word') between the 1st and the last. Two things can happens: if the chosen word is less than the word to be located, then I have to locate it between that one and the last because chosen word < locate word.
And if word' is greater, then I have to locate it between the 1st and word'.
I: The difference with looking up is that we had a case that allows you to finish the process.
N: Yes, I know, it happened to me that I couldn't finalize.
I: Exactly, here, your process doesn't finish, it is what is missing to you.
N: There will be a time in which it is going to be always between two. But I didn't find the opportunity to put it, I could have been written it with words: **when this is the case, locate the word there**.
I: Very good.
N: It is the same with the line of kids in order, what would have to be changed is "the data in the machine" to say that if the kid is taller (he remarks this) than another ...
I: So, what does it change in this case?
N: The information I have to give to the machine.
I: And the method?
N: The method is the same, exactly. Regarding the novel, well, here the novel doesn't have a relation of order among the words. The relation >, < or = doesn't exist so we would have to find it from the beginning ... or ... trying, let's see, let's see ...
I: If we don't want to try but define a method that despite the relation is non-existent concerning the order in the novel, it allows you to find the word.

N: What it's possible is to start reading it until finding the word.
I: How would you read it?
N: From the beginning of course ... what I can't do is reducing as in the dictionary, where I start eliminating.
I: But if you start from the beginning, let's see, what would you do?
N: Well, I open it and start reading, I can't discard absolutely anything.
I: Do you think you can't?
N: If I don't know where *gato* is, the author can have *gato* as a last name for instance and I have to read up to the end.
I: Let's suppose that this is the novel, let's see how you do it.
N: I go on reading from top to bottom, I accomplish my order (he remarks). I could do it from bottom to top if I wanted. (He means from the end to the beginning). The intention is not reading the novel but finding the word *gato*.
I: Good, suppose this is the 1st. What do you do?
N: It's not *gato*, I go on.
I: And what do you go on with?
N: With the one below, (he means the following) If it's not *gato*, I go on until it is *gato*.
I: Do you think there's something that diminishes anyway?
N: Yes, of course, it doesn't make sense to read something and then read it again.
I: So, why do you find it?
N: Because I don't skip anything and the novel is reduced while I am reading it.

Questions posed in the collective class

I: How would it be the most efficient way to choose the word for all the cases? (After several attempts, I explain it to him).
N: I totally agree, yes, yes, yes. Yes, because when trying one says "I bet to the middle and see what happens".
I: Exactly. Now, about that the word is not in the dictionary.
N: Word' will always be different from word.
I: And then how does the process end?
N: You will have an empty result. I mean, you will be reducing the list until having the words which are just before and after the one you search, but this one is not there, then the result is empty. I mean word' and last word or first word and word' will be just the one that is before and the one that is after the one you are looking for, but the one you search won't be there.
(Observe that they always refer to what would happen (the result) and never to what must be done (the action).)
I: How could we change our method to include this case?
N: Good question ... er ... er
I: Nicolás, you have already told me, you already know it. How would it be what you told me adapted to this notation? (The algorithmic one).
N: I know how to say it in words but ...
I: Let's see, what's going on with these values? (Word first and word last).
N: They reduce.
I: So ?

N: We would have to add a condition that checks if the list between the word I chose and first one or the last one, is empty, that finishes the procedure.

C.2.10 Interview with Sergio

Exercise 1

I: How would you make some trees using these rules?
S: (He does it well).
I: Which rule did you use to make this one?
S: With 1, 2 and 3 ... a1 is R.
I: Can you explain it? Which are the minimum trees?
S: A and R.
I: Construct another tree with these two. (Although he builds the trees correctly, he does not know how to explain how he did it; apparently, he does not apply the rules, or he applies them the wrong way, in spite of the correct trees).
I: Read rule 3.
S: (He does it).
I: Who is a1?
S: This and this. (He does it well).
I: Good. So build a new tree applying rule 3.
S: (He does it well).
(We repeat it many times, with the different rules. He does it well).
I: Now, how do we define a method to count the red lights of any tree? (I remind him of the language, what he did, and he reads it).
I: Can you see any similarity between the two methods?
S: Yes, the method is similar.
I: How?
S: If I add some things ... lights are added ...
I: So, how do we go about it?
S: When you apply a rule to the tree ...
I: Let's see.
S: Red ... by each tree that has red lights you add 1.
I: Sorry ... What do you mean?
S: That if the tree is a red one, you add 1. (He refers to R (rule 1). He writes it well for rules 1 and 2).
I: Good. What else?
S: With rule 3 I add one red light to the lights that are already in the tree.
I: And what shape does the tree have?
S: One red light plus two trees.
I: Which ones?
S: a1 and a2.
I: So?
S: Red ones in a1 + red ones in a2. (He writes well with rule 4).
I: Very good. Does it help you for all trees?
S: Yes.
I: So, what's the name of your method?

S: How is it called?
I: Yes, you've used it to count the red lights of a1 and a2, so how have you used it if it does not have a name?
S: Oh! Yes, recurrence. That is, I recur to this.
I: How do you recur to your method?
S: ...
I: (I explain for a1 and a2). So, what's your method's name?
S: Red ones in the tree.
I: Perfect, then we put that name to the method and call "any tree" with a variable, for example: "a". So we can say that we have a mathematical function "red-ones-in", applied to "a" that is any tree and that is the same as all this (I point out to what he wrote). That is to say, the domain is the set of all trees formed by the rules, the range is N and the rule of correspondence is what you wrote.
(Sergio gets astonished when he sees how it looks).

Exercise 2

S: It is the same as the other.
I: Sorry ... (I represent the dictionary). How do you locate word'?
S: Comparing.
I: Which ones?
S: The first one with the one I am looking for.
I: And if we suppose that it is less?
S: Word' is less?
I: Suppose.
S: Then it would go forwards.
I: And if you get it greater?
S: I would compare with the following ...
I: If you get it less?
S: I locate it there.
I: And if it is greater?
S: I would go on with the following.
I: Until?
S: Until finding it a place, until comparing it with all of them ... that is, until finding its place.
I: Very good. This way of locating has nothing to do with the search. Or does it?
S: Yes, it has to do with it.
I: Let's see.
S: In both of them you start reducing the set.
I: Until when?
S: Until locating it in the place or until finding the word.
I: OK, part 2.
S: Yes, it is the same. Except if two children are the same height.
I: It is true, good, and are there any differences?
S: That one.
I: Yes, that one, is there another difference?
S: Which is the order?

I: Exactly. It is called order relation. OK, we still have the third.
S: Look for one by one, because there is no order, it is more complicated. Taking a paper and revise if it is not there ...
I: What do you mean by "revise"?
S: Compare with all the words.
I: Is there any similarity with the other search?
S: It is also a method.
I: Why does it work?
S: By discarding.
I: What is discarded?
S: The pages I am reading, I am reducing the set.
Questions posed in the collective class

I: To finish up we see how we could choose words in a way that is not at random? In a way that is equally efficient for all cases?
S: The middle point.
I: What happens if the word is not there?
S: ...
I: What happens with the list if the word is not there?
S: The list does not contain the word.
I: Which is the limit of that list?
S: One or two words.
I: What should we add to the method?
S: In fact it is always one.
I: So, when the list has one word, what happens?
S: It is not there if it is not the same.

C.2.11 Interview with Sofía

Exercise 1

I: Read the rules. (She reads them).
S: a1 and a2 can be R or A.
I: Yes, sure. Let's see if you can build some trees (she does it well) which rule did you use to build this one?
S: Rule 3.
I: OK, let's see another.
S: (She does it).
I: Which is the rule?
S: Rule 4. (She makes some more that I ask her to do. I explain to her the notation we use to describe the trees. She does it well).
I: Which are a1 and a2?
(She does it well. We do many cases. Sofía explains that she took **the example as the model**, that is, she didn't think of the example as an example, (other students did the same, as well. I think that this is an important difficulty in learning.))
I: How can we define a method to count the red lights of any tree?
S: You divide by the first one, I mean, let's see, sure, you have ... with rules 3 and

4 is different, with rule 3 it is one thing, and with rule 4 it is another.
I: Good, so let's put what corresponds to each rule. It is what we did with the language ... do you remember?
S: Yes, yes. With rule 1 it is 1, with rule 2 it is 0, with rule 3 you have 1, you already know that you will have 1 ... and the others, ... they can be infinite ...
I: Not infinite, because we always apply the rules a finite number of steps. The number can be very big. How can it be worked out?
S: With rule 3 you have one more than rule 4.
I: Refer yourself to rule 3.
S: 1+ ... er er ... That is, if a1 and a2 have red ones, + the red lights of a1 and a2 .
I: Very good, write that.
S: (She writes).
I: We are adding so we are going to put + instead of "and".
S: (She writes it well).
I: And rule 4?
S: Rule 4 would be ... er ... red ones from a1 + red ones from a2 .
I: Very good. Now what is missing is to indicate that this method is useful to count the red lights of any tree. Here you are using it to count the red lights of a1 and here, the red lights of a2. How is the method called, then?
S: How is it called? I don't know ...
I: You have already used that name because you counted the red lights of a1 or the ones of a2, using the same method to do it, in those cases the trees are a1 and a2 . So what name does the method have?
S: Red ones of ... anything.
I: Very good. (We put it at the beginning with a - between "red" and "of" (I explain why and we apply it to any tree.))
S: Yes, I agree.
I: Let's use it to count the red lights of this tree.
S: It would be 1+ ... (She does it well. We check).
I: Why does the method end?
S: Because the tree is over, in the last branches there are no more ... because it counts a1 and a2 ...
I: Where is what you told me "there are no more branches" taken into account?
S: I count the ones from a1 and a2 ... and there are no more.
I: Which are the clauses of your method that indicate it?
S: From "a" ... ("a" is any tree).
I: Which is the relation between "a" and the rest of the trees to which you apply the method?
S: That all of them have lights R and A.
I: Yes, but what makes it different from a1 and a2 and "a"?
S: "a" is the set of a1 and a2 .
I: And then how are they with respect to "a"?
S: Smaller.
I: So, why does the method finish?
S: Because a1 and a2 go smaller.
I: Until where?

S: Until A or R.
I: And which are the clauses in which your method take these cases into account?
S: Rule 1 and rule 2 (She refers to cases A and R of her method).
I: Are there any similarities, for instance, with the search in the dictionary?
S: Each time you shorten things, it goes from the biggest to the smallest.
I: Until when?
S: Until there are no more ...
I: OK, we go to the second exercise.

Exercise 2

S: First we have to make a list of words.
I: (We represent it).
S: Then after you make the list, we have to find the least and the greatest.
I: How do we do that? Let's see.
S: We have the list and at the end it has to be between two.
I: How would you do to find that place?
S: Because it has an order ... First, the first letter, then the second ...
I: Let's suppose that given any word of the list, we compare it with word' (the one we want to locate) and we know if it is greater or less.
S: Oh, I take any word and I see if it is $>$ or $<$, and if the word I want to include is $<$ than the one I chose, then that is the minor quota, then I discard all the rest and I take that from the last and take another word of that new list and see if it is $>$ or $<$ and if the one I take is $>$ ... wait ... I think I said it the other way around.
I: Yes. You said it the other way around.
(Observe the difficulties of imagined reasoning, Sofía has the idea perfectly well, but when expressing it, she gets wrong because she must "imagine" what she does, it is one of the steps of conceptualization).
S: ... then I discard the previous ones and like this I go on shortening the list until I get to two and there, in between those two, the word is located.
I: Very good. The second question?
S: Yes, sure, it is the same , because we have the line of children and you take out one child, measure him and if he is shorter, ... , until two children are left and there is the place.
I: What is it that changes with respect to the dictionary?
S: That with the kids there are measurements of height and in the dictionary it is an abstract order.
I: And what we do?
S: It is the same.
I: OK, and the third?
S: It does not have a defined order so we have to look for word by word until finding it and it could be that it is at the beginning or at the end.
I: So, how would you search?
S: I would start by the beginning or by the end.
I: Are there any similarities with looking up in the dictionary, for example, are you going to find the word *gato*?
S: Yes.

I: Why?
S: Because you tell me it is there.
I: And why do you find it?
S: Because I start discarding the ones that are not.
Questions posed in the collective class

I: There were other little questions ... How could we choose the word in the search of the dictionary in such a way that it is not at random?
S: What is convenient is to look for the one that is equidistant, let's say.
I: So, where is it?
S: In between word1 and wordlast.
I: Why do you think this is the best?
S: Because you have more possibilities of discarding a greater number of words. If you choose the next to word' (the word to be located) and you find that it is less than it, you just discard one, otherwise if you choose the middle you can discard more.
I: Very good. And the other question? What happen if the word is not in the dictionary?
S: You will not find it ...
I: What happen with the method?
S: It does not work.
I: How can we modify the method in a way that if the word is not, we know that and the method ends?
S: if there is a greater word and a less word but it is never equal, then that it says that the word does not exist.

C.3 Material for selecting the students

In this section the original material handed out to students for selecting those who participated in the interview is included (in Spanish).

Proyecto sobre enseñanza de matemática.
Sylvia da Rosa, docente del Instituto de Computación de la Facultad de Ingeniería.
Universidad de la República.
e-mail: darosa@fing.edu.uy

El objetivo de esta tarea es seleccionar estudiantes para participar en un proyecto de investigación sobre la enseñanza de matemática. Los estudiantes seleccionados participarán en 3 entrevistas de aproximadamente 45 minutos de duración cada una. La primera y la tercera entrevistas son individuales, mientras que la segunda es una clase colectiva. En dicha clase se repartirán 2 o 3 problemas sobre el tema dado en la clase, que los estudiantes deberán intentar solucionar y en la tercera entrevista explicar sus soluciones (completas o parciales, correctas o incorrectas). Aparte del tiempo de las entrevistas y del de la solución de los problemas, no se requiere más dedicación por parte del estudiante. Las entrevistas individuales serán grabadas para su posterior análisis. Los estudiantes seleccionados, que participen en la totalidad del proyecto, recibirán un libro de estudio de su elección.

A continuación se presentan 3 problemas y algunas preguntas. Lee con atención los enunciados y responde las preguntas usando tus propias palabras y sin preocuparte de utilizar conocimientos matemáticos anteriores. Tampoco es importante si tus respuestas son correctas o no, sino cuáles son las ideas que los problemas te sugieren. Utiliza la hoja adjunta para responder.

Primer problema

Dos estudiantes han sido contratados para adornar las vidrieras de una tienda con figuras que simulan árboles, a las cuales deben colocar luces rojas y azules. Si R significa "luz roja" y A "luz azul", ejemplos de figuras con luces son:

```
     A                  R
    / \                / \
   R   R              R   A
                     / \ / \
                    A  R R  R
```

Observar que los árboles tienen la siguiente forma: de cada R y de cada A, salen dos ramas, una a la izquierda y otra a la derecha. Siempre que se ponga una luz en una de las ramas debe ponerse en la otra también, salvo en las últimas de las cuales no sale ninguna rama ni para la izquierda ni para la derecha. Las figuras son de distintos tamaños.
Uno de los estudiantes coloca las luces al azar, siempre completando el árbol según la forma descrita. El otro estudiante decide hacer su tarea metódicamente siguiendo las siguientes reglas:
a) en la punta superior coloca una luz roja.
b) debajo de cada luz roja coloca a la izquierda una azul y a la derecha una roja.
c) debajo de cada luz azul coloca a la izquierda una roja y a la derecha una azul.
Sigue estas reglas hasta completar la figura de árbol.
Contesta las siguientes preguntas, explicando las razones de tu respuesta:
1. El primer estudiante le dice al segundo que para sus árboles, siempre se cumple que el número de luces rojas es uno más que el de azules. El segundo le pregunta cómo puede estar seguro que si el árbol tiene mil ramas esa relación sigue existiendo. Cómo podría el primer estudiante responder esa pregunta?
2. El primer estudiante dice que colocando las luces a su manera es más fácil contar el número de luces rojas o azules empleadas en cualquier árbol. El segundo estudiante dice que ambos podrían emplear un método para contar las luces rojas (o azules) de cualquier árbol, sin saber el número de ramas. Te parece que tiene razón? Por qué? Cómo sería ese método al que se refiere?

Segundo problema

Dos hurgadores de basura venden lo que han juntado en el día a 20 pesos el quilo. Uno de ellos calcula sus ganancias diarias sumando 20 por cada quilo que vende, el otro multiplica 20 por el número de quilos que vendió en el día. Un amigo común les comenta que es lo mismo, pero ellos se niegan a creerle.
1. Con qué argumentos podría el amigo común convencerlos?

2. Con qué argumentos podrían ellos defender su posicón y rebatir lo que el amigo les dice?

Tercer problema[5]

El método de prueba por inducción (completa o matemática) puede describirse de la siguiente manera:
Para probar que una propiedad P se cumple para todos los números naturales, probamos las dos proposiciones siguientes:
a) P se cumple para 0.
b) Si P se cumple para un n cualquiera, entonces se cumple para el sucesor de n.
Un estudiante dice que en b), al suponer que P se cumple para un natural cualquiera, estamos suponiendo lo que queremos probar y que entonces el método no es válido.
1. Estás de acuerdo con él? Si lo estás, cómo explicas que se use un método de prueba no válido? Si crees que no tiene razón, cómo se lo explicarías?
2. Si sustituimos "los nmeros naturales" por "los números reales mayores o iguales que 0", obtendríamos un método de prueba válido? Por qué si o por qué no?

Por cualquier duda o consulta, enviar e-mail a darosa@fing.edu.uy.
Nombre del estudiante:
Número de Teléfono:
e-mail:

[5] This problem was not considered in the research.