The Learning of Recursive Algorithms from a Psychogenetic Perspective

Sylvia da Rosa

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

Abstract. The ideas presented in this paper concern computer science education research within Jean Piaget's theory *genetic epistemology.* Results from Piaget and collaborators investigations about the recursive reasoning on the series of natural numbers are extended to learn about entering students recursive reasoning on other inductively defined structures. In this paper the main aspects of that extension are described using a selected example.

1 Introduction

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 recursion have been presented and put into practice according to various 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, we have observed 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 them 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 phrases 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.

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?" The motivation of the research described in [7] – from which the example presented in this paper is extracted – arises from the above observations and questions. The main goal of the research is to develop an instructional proposal of recursive algorithms in which the construction of the concept of recursion 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 which explains both the construction of knowledge and the evolution of cognitive instruments [2, 4], accounting for the psychogenetic approach of the research.

1.1 The starting premise

Answering the above questions requires doing research in computer science education *within* the framework of educational related disciplines. The lack of this type of research is pointed out as one of the problems of computer science education in [8]. The authors relate computer science education 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. 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 these observations, our investigations about the learning of the concept of recursion [7] are incorporated into a theoretical framework based on Jean Piaget's theory *genetic epistemology*, especially Piaget and collaborators works regarding the formation of recurrence reasoning [3]. In that work, 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. 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[1] 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 on 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 implication 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.

Referring to an experiment by which the subjects have to construct two collections of pearls and answer some questions (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 of actions 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 formalization of that construction is given by Peano's axioms. In our work the above result is extended to reasoning on other structures – isomorphic to the structure of natural numbers – whose formalization is given by the inductive definitions in [1].

The starting point of the research is therefore the following premise:

> the source of thinking which permits the design of recursive solutions for 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.

[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 a concept, and mathematically it means *structure of elements and their operations*, as usual.

The methodology of research includes conducting students interviews to learn about the evolution of students' thinking. The students are encouraged to formalize their solutions in mathematics.

The paper is organized as follows: Section 2 addresses the following topics: the problem posed in the selected example, the basic questions used in the interviews, the guidelines of the analysis of students responses, some excerpts of the interviews and a summary of the activity of formalizing students solutions. In Section 3 some related works are described. Sections 4 and 5 present some conclusions and further work respectively.

2 The example

One of the problems posed to the students in the interviews 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 in the sense of [1]. Most of the problems in which operations over inductively defined data types have to be determined admit relatively easy recursive solutions in this way.

An inductive definition of a set of words is presented to the students and they are encouraged to find a way of calculating a value, as described below. 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:

1. ab is a word.
2. If * is a word then a*a is a word.
3. If * is a word then b*b is a word.
4. 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 interviews, the students are required to develop a method to count the number of a's in any word. It is expected that the students solve the problem 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. 13 students aged between 16 and 18 and selected from different groups both from the last year of high school and the first year of the University have participated in the whole activity.

2.1 The questions

According to the premise stated in subsection 1.1, the forms of reasoning allowing to derive recursive computations arise from the construction of the concept of the structure of the language by similar mechanisms that for the case of natural numbers. In such construction the evolution of the relationships between the initial word "ab" and any word, and between any word and the next one plays a fundamental role. Those relationships are denoted $ab \to w_n$ and $w_{n-1} \to w_n$ respectively. The possibility of defining recursive algorithms on the elements generated by the rules, arises from the construction of the inverses of those relations, that is to say, $w_n \to w_{n-1}$ (any word to the previous one) and $w_n \to ab$ (any word to the initial one) from which the inductive and the base cases of the definition can be respectively derived. The questions attempt to help the students in understanding those relations and consequently designing a solution to the problem. The lists of questions described below were previously designed and during the interviews other questions were added, according to the responses of the students. Comments not included in the questions are indicated between parentheses.

Q1: Write some words of the language.
Q2: Can you determine of these sequences which are words of the language and which are not? (Several sequences are presented)
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?
Q6: And this other one?

Induced by questions Q1 to Q6, students focus on the generation of any word from the initial element "ab", that is, on the relationship $ab \to w_n$, where "ab" plays the role of the natural number 1 in the construction of the series of natural numbers. In the following list of questions, the expression "the little symbol" refers to the symbol * of the rules, that represents the previous word of a word.

Q7: In this word, 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).
Q9: Why not or why yes?
Q10: Then, what does the little symbol have to be?
Q11: A student said that rules 2 and 3 say that a word is formed from a word formed before. Do you agree?
Q12: Then in order to know if a sequence of a's and b's is a word, what can we watch?

The goal of the questions Q7 to Q12 is to cause students' thinking advance towards the relationship between any word 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.

Questions Q13 to Q18 are aimed at generating the relationships $w_n \to w_{n-1}$ (any word to the previous one) and $w_n \to ab$ (any word to the initial one). Observe for instance that questions Q17 and Q18 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, as the excerpts from the interviews included below show.

Q13: How many a's does this word have?
Q14: And the little symbol? And here? (The complete word).
Q15: In any word, if we know how many a's the little symbol has, can we determine how many a's the word has?
Q16: How? Write it down please.
Q17: Determine the a's of ababaaaabbbabbbbaaaababababa by using only what you have written.
Q18: What is missing in what you wrote so as to be able to use it until the end?

2.2 Analyzing students responses

The students are required to write down all their work, while responding to the questions from the interview. The goal of analyzing students' responses is to observe how the students solve the problem of counting the number of a's of any word. The observation is guided by the identification of levels in the evolution of students' thinking while constructing the concept of the relationships described above. The levels are denoted as follows: $ab \rightarrow w_n$ (*level 1*), $w_{n-1} \rightarrow w_n$ (*level 2*), $w_n \rightarrow ab$ (*level 3a*) and $w_n \rightarrow w_{n-1}$ (*level 3b*).

Faced to the question "with which rule did you form this word?", eight students answer "with the first, the third, the second ..." thinking of the relation between "ab" and the current word, that is $ab \rightarrow w_n$, (*level 1*). The relevant question to identify the hard evolution to the next level is "which is the little symbol in this word?" Ten students answer that it is "ab", revealing confusion in differentiating between the initial element and the previous element of any word. In the development of the interview, all students succeed 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. This means that the relationship $w_{n-1} \rightarrow w_n$ of level 2 has been constructed.

On the other hand, faced to the question of how the number of a's of any word can be determined, all the students succeed in formulating a solution to the problem in Spanish, following the way the words are generated by the rules 2 and 3. This means that the inverse of $w_{n-1} \rightarrow w_n$ has been constructed – that is $w_n \rightarrow w_{n-1}$ – and that students' thinking has evolved to level 3b.

Finally, when required to apply their algorithm to a particular case, the students discover the essence of recursion and experience the need of defining the clause for the base case constructing the relationship $w_n \rightarrow ab$ (level 3a).

Excerpts of the interviews are included below to show the development of students' responses in detail. The analysis of students' responses interspersed with the series of questions (indicated in italics) illustrates the identification of the levels.

2.3 Excerpts from students interviews

In the following, excerpts from two of the interviews are included (Andrés and Gimena). The unnumbered questions were spontaneously posed during the interviews. Here R indicates a student's response and comments not included in the questions are indicated between parentheses.

Answering questions Q1 to Q4 generates a list of sequences of a's and b's, some written by the student, some presented by the interviewer; some are words of the language, (for instance "aabaababaa"), others are not, (for instance "baababbbab"). The excerpts below refer to sequences in that list.

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? (The same for several words follows).
The student marks using a circle the part of the sequence corresponding to the previous word. He does it quickly and correctly for all the words.
Q8: Could the little symbol be like this?
The interviewer points to a sequence that is not a word.
R: No.
Q: Why not?
R: Because it does not follow the rules (immediate).
Q11: 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.
Observe that despite marking correctly the previous word before, the student confuses "previous" and "initial word".
Q: What was this? (The interviewer points to a sequence that the student has already marked using a circle).
R: ...
Q: I made you a question and you marked this sequence (for example "ababba") with a circle, what was the question? Do you remember?
R: What was the * there? (He means in the whole sequence, for example "babab-bab").
Q: So this is ... (pointing to "ababba")
R: The * (of "bababbab").
Q: So, according to what we said of what the student said, which is the previous

word so as to form this one? (pointing to "bababbab")
R: The little symbol.
Q: Then in each word, what is *?
R: The previous word.
level 2.
Next series of questions reveals how slow and difficult is the passage from level 1 to level 2, Andrés goes back to level 1.
Q12: 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 (saying something like "it has to be formed from applying a rule to *"), level 2 is weakly constructed.*
Q: What is * in this word?
R: "ab".
Back to 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, it was this here, (pointing to correct previous answers about *) here, this, here, this, what was *?
R: ... the previous word to the last application (of the rules).
level 2.
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.
Q13: How many a's does this word have?
R: (He answers several particular cases well).
Q15: 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 perfectly writes the cases of the algorithm for rules 2 and 3, attaining level 3b.
Q17: Determine the a's of ababaaaabbbabbbbaaaababababa by using only what you have written.
R: I mark the little symbol and has one, two ...

(Observe that the student attempts to count one by one the a's in the sequence).
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 now I mark again and the rule applied was number 3 so that it remains like 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).
Q18: What is missing in what you wrote so as to be able to use it until the end?
R: ... rule 1.
He completes his algorithm with the base case, attaining level 3a.
Gimena
(The interviewer points to a word constructed by the student, say "bababbab").
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? (always referring to "bababbab").
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, obtaining the word "babb").
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? (pointing to "bababbab").
R: The little symbol would be all this. (She does it correctly, marking "ababba" with a circle).
level 2.
Q: In this one? (pointing to another word).
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.
Q13: How many a's does this word have?
R: (She answers several particular cases well).
Q15: 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, to induce the student to become aware that the number of a's is actually the same or increased by 2).
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 perfectly writes the cases of the algorithm for rules 2 and 3, attaining level 3b.
Q17: I give you the word ababaaaabbbabbbbaaaabababa and ask you that using what you wrote, work out the number of a's.
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.
Q18: What is missing in what you wrote so as to be able to use it until the end?
She completes her algorithm with the base case, attaining level 3a.

When the students are required to write down their descriptions of the method for counting the a's of any word, special care to the way they refer to the method is taken. For instance, they are required to begin with "the number of a's of any word is" and to reflect about what has to be written after "is", because the students tend to write "2 + *" instead of "2 + the number of a's in *", although they correctly *say* that. All the students succeeded in deriving a complete recursive formulation of their solution to the problem in natural language. These 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
```

2.4 Formalizing students solutions

After the interviews instructional instances in the form of collective classes take place. The goal of these instances is to encourage the students to discuss their solutions, and to find and correct errors and to formulate mathematical definitions of their informal descriptions. Every student gets his/her worksheet and they have to try to represent their solutions mathematically. New questions are posed to help the students in constructing a correspondence between their descriptions and the mathematical language. For instance, to focus on the form of the word according to the number of the rule, questions like "how can you represent the expression 'if the rule is ...' using the word in the left hand side of your description?" are posed. In this way students' thinking is directed *from* the definition of the algorithm *to* the inductive definition, reinforcing the idea stated in the premise of subsection 1.1, regarding the fact that both the concept of the inductive definition and the concept of the definition of the recursive algorithm evolve as aspects of the same mental structure.

The need of describing the domain and codomain of the function as part of its definition is naturally assumed by the students, and the language defined by the inductive definition is called "Set-of-words". The mathematical function definition described by the students is as follows:

```
nr-of-as : Set-of-words -> N
nr-of-as (word) = if word = ab then 1
                  else if word = a * a then 2 + nr-of-as (*)
                       else if word = b * b then nr-of-as (*)
```

This dialectic process allows the students to experience the correspondence between their construction of the algorithm and its formalization in mathematics.

The students that have participated in this activity were selected from groups of several orientations (engineering, natural sciences and social and human sciences). Therefore, no implementation in any programming language of the mathematical definitions is introduced in this activity.

3 Related Work

Several approaches to the learning of recursion have been found in the literature and are described in [7]. The most relevant aspects of that description are presented in this section.

Some authors refer to the term "mental model" used by cognitive psychologists to define cognitive representations of knowledge about particular situations or phenomena. In general it can be said that cognitive psychology uses computational model to account for human cognitive behavior and most of the related research is made in the field of artificial intelligence. In the case of the learning of recursion, several authors refer to mental models to describe the knowledge that students acquire when introduced to the concept, in most of the cases using some programming language or environment. 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 [23, 21, 19, 26, 22]. No reference has been found to *how* the understanding represented by mental models is constructed.

Some authors investigate the learning of recursion with respect to the learning of mathematical induction [15, 14, 25], and other ones with respect to iterative procedures [17, 24, 22, 23]. Other approaches are: emphasizing a declarative approach [12, 11, 9, 29, 20], focusing on the impact that visual or graphical representations are supposed to have on the learning of recursion [27, 13, 30] and following a phenomenographic tradition of research [28].

Other works consider pedagogical approaches 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 [10, 16, 18]. The general perspective of the research is often called constructivism and constructivist researchers often point out that constructivism arises from Piaget's ideas. However, (almost) none reference to Piaget's work appears in the bibliography.

One of the aspects that differentiates our work from the related works is the level of importance assigned to a theoretical framework, in this case, Piaget's *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 it are defined and precisely described in Piaget's and collaborators numerous works, both empirical and theoretical. On the other hand, in our work students' knowledge about recursion is investigated without any formal introduction to the concept. In contrast, most of the related works take as starting point what the students are supposed to have learning in instruction, often using programming environments. Finally, our approach relates the introduction of the concept of recursion with the concept of inductive structures, supported both by the mathematical and the epistemological theory [1, 3].

4 Conclusions

The example presented in this paper illustrates the application of a methodology of research developed from principles of Jean Piaget's theory *genetic epistemology*. In this example, the design of recursive algorithms taking into account the interaction between the students and the structure of the elements over which the algorithm is to be applied is investigated. From the analysis of the responses of the students evidence about the relevance of the construction of the concept of the structure as the source of forms of thought that facilitate reasoning on recursive methods arises. This claim is supported by two main results: on the one hand, regarding the evolution of the concept of the structure of the language, the most important passage is from the relationship between the initial word "ab" and any word ($ab \to w_n$) to the relationship between any word and the next one ($w_{n-1} \to w_n$). On the other hand, the students quite easily succeeded in constructing the inverse relationships $w_n \to w_{n-1}$ (any word to the previous one) and $w_n \to ab$ (any word to the initial one). Consequently they correctly formulated the clauses for the inductive cases and the base case of the algorithm respectively.

Failures and advances are detected in the construction of the involved relations related to the transformation of instrumental into conceptual knowledge [2, 5]. This transformation takes place in a slow and hard process, which is (almost) never considered in traditional teaching of recursive algorithms as school subject. This explains, in our opinion, the observation that despite the many courses where students have been introduced to the concept, even advanced students have difficulties dealing with recursion. 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 greatest common divisor 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 could be introduced [6].

Carefully introducing the construction of recursive definitions means that any pedagogical approach should begin by considering what the students think and how they reason when solving instances of problems, and how they themselves can achieve the formalization of their solutions. Research regarding these issues within the framework of educational related disciplines is of the highest importance, as the example presented in this paper shows.

5 Further work

The aspect of proving properties by induction is strongly related both to the definition of inductive structures and to the definition of recursive algorithms. In [3] the authors consider the development of reasoning by recurrence both for the case of calculating some value and proving some property on the series of natural numbers. Further work consists in stating problems such as "how can a property like 'for any word of Set-of-words, the number of a's is odd'" be proved, and observe the methods that students develop to find an answer. It is expected that the students informally describe proofs by induction and that they are capable of attaining its formalization.

The stage of implementing the mathematical definitions in a programming language is not included in the activity described in this paper, partially due to the heterogeneity of students' interests. It is our concern to investigate how computer science students work to obtain Haskell[2] programs from mathematical definitions of recursive algorithms, in the manner of [6]. In that article the solutions of some problems selected from a discrete mathematic course for high school mathematics teachers are implemented in Haskell.

Finally, it would be interesting to consider aspects from approaches based on other learning theories, particularly arising from social constructivism as those presented in [16, 10, 18].

6 Acknowledgements

The author would like to thank Peter Henderson for correcting the English and the referees for their profitable comments.

References

1. Peter Aczel. Handbook of Mathematical Logic. Studies in Logic and the Foundations of Mathematics, Vol. 90. North-Holland Publishing Company, Amsterdam-New York-Oxford (1977)
2. Piaget Jean and Rolando García. Psychogenesis and the History of Sciences. Columbia University Press, New York (1980)
3. Piaget Jean. La Formation des Raisonnements Recurrentiels. Presses Universitaires de France (1963)
4. Piaget Jean. La Prise de Conscience. Presses Universitaires de France (1964)
5. Piaget Jean and Evert Beth. Mathematical Epistemology and Psychology. D.Reidel Publishing Company, Dordrecht-Holland (1966)
6. Sylvia da Rosa. Designing Algorithms in High School Mathematics. Lecture Notes in Computer Science, Vol. 3294 (2004)
7. Sylvia da Rosa. The Learning of Recursive Algorithms and their Functional Formalization. PEDECIBA 05-12 (2005). Doctoral thesis, can be found at http://www.fing.edu.uy/˜darosa/
8. Christian Holmboe, Linda McIver and Carlisle E. George. Research Agenda for Computer Science Education. In G.Kadoda (Ed). Proc. PPIG 13 (2001) 207–223
9. J.Angel Velazquez Iturbide. Recursion in Gradual Steps (Is Recursion Really that Difficult?) Proc. SIGCSE 31. ACM Press, NY, USA (2000) 310–314
10. Dalit Levy and Tami Lapidot. Recursively Speaking: Analyzing Students' Discourse of Recursive Phenomena. Proc. SIGCSE 31. ACM Press, NY, USA (2000) 315–319
11. Bruria Haberman and Haim Averbuch. The Case of Base Cases: Why are They so Difficult to Recognize? Student Difficulties with Recursion. Proc. 7th Annual Conference on Innovation and Technology in Computer Science Education. ACM Press, NY, USA (2002) 84–88
12. Davit Ginat and Eyal Shifroni. Teaching Recursion in a Procedural Environment - How much should we emphasize the Computing Model? Proc. SIGCSE 30. ACM Press, NY, USA (1999) 127–131

[2] www.haskell.org

13. Carlisle E.George. EROSI - Visualising Recursion and Discovering New Errors. Proc. SIGCSE 31. ACM Press, NY, USA (2000) 305–309.
14. Susan Wiedenbeck. Learning Recursion As a Concept and As a Programming Technique. Proc. SIGCSE 19. ACM Press, NY, USA (1988) 275–278
15. Uri Leron and Rina Zazkis. Computational Recursion and Mathematical Induction. For the Learning of Mathematics, Vol. 6, No.2 (1986) 25–28
16. Dalit Levy and Tami Lapidot. "It's just like the whole picture, but smaller": Expressions of gradualism, self-similarity, and other pre-conceptions while classifying recursive phenomena. In G.Kadoda (Ed). Proc. PPIG 13 (2001) 249–262
17. Susan Wiedenbeck. Learning Iteration and Recursion from Examples. International Journal of Man-Machine Studies, Vol. 30, No.1 (1989) 1–22
18. Mordechai Ben-Ari. Recursion: From Drama to Program. Journal of Computer Science Education, Vol. 11 No.3, (1997) 9–12,
19. Shawkat Bhuiyan, Jim E. Greer, and Gordon I. McCalla. Suporting the Learning of Recursive Problem Solving. Interactive Learning Environments, Vol. 4, No.2 (1992) 115–139
20. C.T.P. Burton. Conceptual Structures for Recursion. Lecture Notes in Computer Science, Vol. 1022 (1995)
21. Tina Gotschi, Ian Sanders, and Vashti Galpin. Mental Models of Recursion. Proc. SIGCSE 35, ACM Press, NY, USA (2003) 346–350
22. Claudius M. Kessler and John R. Anderson. Learning Flow of Control: Recursive and Iterative Procedures. Human-Computer Interaction, Vol. 2, Lawrence Erlbaum Associates, Inc. (1986) 135–166
23. Hank Kahney. What Do Novice Programmers Know About Recursion. Proc. SIGCHI Conference on Human Factors in Computing Systems. ACM Press, NY, USA (1983) 235–239
24. Uri Leron. What Makes Recursion Hard? Proc. 6th International Congress of Mathematics Education (1988)
25. Cheng-Chih Wu, Nell B. Dale, and Lowell J. Bethel. Conceptual Models and Cognitive Learning Styles in Teaching Recursion. Proc. SIGCSE 29, ACM Press, NY, USA, (1998) 292–296
26. DiCheva, D. and Close, J. Mental Models of Recursion. Journal of Educational Computing Research. Vol. 14, No. 1 (1996)
27. Good, J. and Brna, P. Novice Difficulties with Recursion: Do Graphical Representations Hold the Solution? Proc. of the European Conference on Artificial Intelligence in Education, (1996) 364–371
28. Shirley Booth. Learning to Program. A phenomenographic perspective. Göteborg Studies in Educational Sciences, ISSN o436-1121 (1992)
29. C.Clack and C.Meyers. The dys-functional student. Lecture Notes in Computer Science No. 1022 (1995)
30. 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 (2000) 33–44