Reducing abstraction when learning computability theory

Orit Hazzan

Department of Education in Technology & Science

Technion–Israel Institute of Technology

Haifa, Israel 32000

E-mail: oritha@techunix.technion.ac.il

Main Message

- Local: The theme of reducing abstraction (Hazzan, 99).
  - Computability theory
  - Abstract algebra
  - other topics: Differential equations, Data structures

- Global: Illustration of how theories from mathematics education can be used for explaining students' conceptions of computer science ideas.

Organizing Theme: Reducing Abstraction

- Students use ways by which they reduce the level of abstraction of abstract (mathematical) concepts.
- Students make abstract concepts more mentally accessible.
- Students use ways which enable them to make sense of abstract concepts.

How abstraction level is reduced

- Abstraction level as the quality of the relationships between the object of thought and the thinking person (Wilensky, 1991)
  - Retreat to familiar mathematical structures

Example from Computability Theory

- Midterm exam consists of 4 questions, each one having two parts.
- 4 languages are given.

  For each of the languages L_i determine:

  a) Is $L_i \in R$?

  b) Is $L_i \in RE$?

  Prove your claims.
- Each part gets 12.5 points out of 100.

Some Computability Theory

- A recursive (decidable) language L is a language for which a Turing machine that accepts it and halts on all inputs, exists: If $w \in L$ then the Turing machine halts in state q_{accept}; if $w \notin L$ then the Turing machine halts in state q_{reject}.
- A language L is a recursively enumerable (r.e.) language if a Turing machine that accepts it, exists: For all $w \in L$, the Turing machine halts in q_{accept}; for all $w \notin L$ the Turing machine either halts in state q_{reject} or never halts.

Some Computability Theory (Cont.)

- Using set terminology:
  - R = {L | L is decidable};
  - RE = {L | A Turing machine that accepts L, exists}.
- For determining whether a language belongs to R or to RE, one has to understand the nature of the language.
- The better one understands the nature of a given language, the better one may determine to what category it belongs. Then, based on this analysis, one has to choose a method for proving one's claim.

Example of Reducing Abstraction from Computability Theory

Here are two of the languages that appeared in the exam:

$L_1 = \{(\langle M \rangle, x) : \exists M', x \notin L(M) \cap L(M')\}$

$L_4 = \{(\langle M_1 \rangle, \langle M_2 \rangle) : f_{M1}(M_2) = f_{M2}(M_1) = 01001\}$,
where $f_M(x)$ is the function that M calculates with x as input.

Which language is simpler?

Example of Reducing Abstraction from Computability Theory

- It seems that the definition of L_1 is more complex than the definition of L_4.

- The definition of L_1 includes quantifiers and set notations.
- L_4's description uses functions, objects with which students have been familiar for many years.

Example of Reducing Abstraction from Computability Theory

- The data indicates that students do better in addressing L_1 (average 21.13) than in addressing $L_{4.}$ (average 20.02).
- **BUT:** 12 students who worked on L_4 relatively well did not solve L_1.

Example of Reducing Abstraction from Computability Theory

- Such a gap between the solution of L_1 and of L_4 appears in the exams of weak students (see Table).
- Their average grade of the entire exam is 37.4.
- In the exams of these students:
  - The grade of Question 4 is the higher one.
  - In most of the cases students received most of their points from their work on L_4.

Example from Computability Theory

Student #	**Q1**	**Q2**	**Q3**	**Q4**	**Total**
1	0	5	2.5	**12**	19.5
2	0	5	10	**24**	39
3	5	0	15	**15**	35
4	0	0	9	**10**	19
5	0	2.5	4	**12.5**	19
6	5	5	16	**17**	43
7	0	2.5	2.5	**13**	18
8	0	15	16	**19**	50
9	2	0	14.5	**12.5**	29
10	0	5	25	**20**	50
11	0	15	24	**25**	64
12	0	19	20	**20**	59

Example of Reducing Abstraction from Computability Theory

- Possible explanation:
  - L_4's definition is based on functions: A familiar concept for the students on which students relied.
  - Students' (mental) ability of giving L_4 some meaning.

What is abstraction?

- Abstraction level as the quality of the relationships between the object of thought and the thinking person
  - Retreat to familiar mathematical structures
- Abstraction level as reflection of the process-object duality
  - Show strong need for canonical procedures
- Abstraction level as the degree of complexity of the concept of thought
  - Deal with examples instead of the whole

What is abstraction?

- Abstraction level as reflection of the process-object duality
  - Show strong need for canonical procedures

- C*anonical procedure*: A procedure that is more or less automatically triggered by a given problem.

- This can happen either because:
  - the procedure is naturally suggested by the nature of the problem
  - prior training has linked a specific kind of problem with a specific procedure.

Show strong need for canonical procedures

- The availability of a canonical procedure enables students to solve problems without analyzing properties of mathematical concepts, and to automatically follow the step-by-step algorithm the canonical procedure provides.

Show strong need for canonical procedures

Three main methods for solving the exam:

a. Constructing a Turing machine that accepts L (to show that $L \in$ RE) or a Turing machine that accepts L and halts on all inputs (to show that $L \in$ R).

b. Using Rice theorem: Rice theorem provides criteria for determining for a given language L of a certain form whether $L \notin$ R and whether $L \notin$ RE.

c. Defining a reduction between two languages L' and L'', for one of which it is known that it is or it is not in R or in RE. (a reduction from L' to L'' indicates that L'' is at least as hard as L'.)

Show strong need for canonical procedures

$$L_3 = \{<M>: \exists M', <M'> \in L(M) \wedge L(M') \neq \phi\}$$

- For determining weather $L_3 \in R$ students could:
  - use Rice theorem or
  - construct a reduction from one of the non recursive languages.
- A comparison of the details involved in these solutions indicates that relying on Rice theorem should be a simpler and shorter process.

Show strong need for canonical procedures

- However, the data indicates that students prefer building a reduction over using Rice theorem (see next slide).

- This observation may be explained by the fact that a reduction is sometimes defined in an automatic process without understanding the details.

Solving Question 3A

- **No. of students who used Rice theorem:** 36.

- **No. of students who built a reduction**: 72.

The students used different languages as the source language of the reduction.

HP	45 students	
$L_{(\phi)'}$ or $(L_\phi)'$	12	(Note: The symbol ' represents set complement.)
L_d	12	
L_u	3	

Show strong need for canonical procedures

- The dominance of HP as a source of the reduction:
  - Immediate explanation: Students practiced this kind of reduction in class and in the tutorials.
  - Another explanation: There is a pattern that one can apply when a reduction from HP is constructed.

Show strong need for canonical procedures

- A student: "*When I do a reduction from HP I know what to do in each case, when it halts and when it does not halt. […] When I learn for the exam [and solve problems from previous years] instead of complicating the solution, showing that a property is a non-trivial property of languages in RE [and using Rice theorem], I always do a reduction from HP*".
- Interpretation: "A reduction from HP can sometimes be constructed automatically without understanding the subtle details".

Show strong need for canonical procedures

- Theories which mainly distinguish between a process conception and a object conception. (Breidenbach et al, 1992; Douady, 1985; Dubinsky, 1991; Dubinsky et al, 1994; Sfard, 1991, 1992; Thompson, 1985).
- Process conception of a mathematical entity reflects a lower abstraction level than its conception as an object. (Beth & Piaget, 1966).
- A canonical procedure may reflect a process conception.

What is abstraction?

- Abstraction level as the degree of complexity of the concept of thought
  - Deal with examples instead of the whole class
- Students reduce the level of abstraction by mentally manipulate simpler objects than those which should be used to solve the problems.
- An example is presented in the paper.

Reducing abstraction and constructivism

- One learns new concepts by constructing mental objects.
- One solves a problem by mentally manipulate objects (or structures) involved.
- When one lacks a mental object to hang onto, one reduces the level of abstraction.

Summary

- Students' tend to think and work on a lower abstraction level than the one which is expected by the instructors.
- Constructivism perspective: A natural mental approach people always engage in when thrust into an unknown situation.
- This perspective cannot be taken by the students themselves. The level of abstraction is reduced unconsciously.

Appendix

Some Computability Theory

- A typical computability course deals with the issue of assessing how difficult is to solve computational problems.
- With respect to a given problem we ask:
  - Is it *possible* to design an algorithm which solves the problem?
  - Is it possible to design an *efficient algorithm* which solves the problem?

Some Computability Theory

- The discussion here is limited to questions in which one has to determine whether a given language is recursive or recursively enumerable.

Show strong need for canonical procedures

Three main methods for dealing with the exam questions:

a. Constructing a Turing machine that accepts L (for showing that $L \in$ RE) or a Turing machine that accepts L and halts on all inputs (to show that $L \in$ R).

b. Using Rice theorem: Rice theorem provides criteria for determining for a given language L of a certain form whether $L \notin$ R and whether $L \notin$ RE. Using Rice theorem requires that one's argument be based on identifying a non-trivial property of languages in RE.

Show strong need for canonical procedures

Three main methods:

c. Defining a reduction between two languages *L'* and *L''*, for one of which it is known that it is or it is not in R or in RE:

- If a reduction from *L'* to *L''* exists, it indicates that *L''* is at least as hard as *L'*.
- A construction of a reduction consists of two steps. First, finding the reduction, and second, proving that the reduction satisfies three conditions.