On Abstraction and Informatics

Tom Verhoeff, *T.Verhoeff@tue.nl*
Dept of Math & CS, Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract

One often hears, and less often reads, the claim that informatics and its application is so difficult because it involves demanding abstractions. Abstractions in informatics supposedly are even harder than those in mathematics and the physical sciences. If abstraction is so important in informatics, then you would expect that we have good ways of dealing with it and communicating about it. To some extent, this is indeed the case, but unfortunately these ways are not widespread. It is our duty to get to grips with abstraction, and especially to address it in the teaching of informatics.

In this article, I will not solve the problems posed by abstraction, and certainly not the problem of teaching abstraction. But I would like to put it more prominently on the agenda. In order to deal with abstraction, you will have to investigate it, dissect it, analyze it, establish terminology, etc. I will give a, somewhat personal, overview of abstraction, showing that it is not a single, atomic concept, but a diverse complex of interrelated concepts. It is my hope that this will help in embedding abstraction more explicitly in the informatics curriculum.

In the development process, you	**In backward-chaining, you learn to**
(1) Draw up a contract for an abstraction	(1) Use an abstraction, given its contract
(2) Validate the contract	(2) Test an abstraction given its contract
(3) Design and implement the contract	(3) Review a given design/implementation
(4) Review the design/implementation	(4) Design/implement a given contract
(5) Test the implementation	(5) Validate a given contract
(6) Use the abstraction	(6) Draw up a contract for an abstraction

Table 1. Forward chain of the development process (left); learning order in backward-chaining (right)

Concerning the teaching of abstraction in informatics, consider the case of teaching abstraction mechanisms like parameterized procedures and Abstract Data Types. In a traditional approach, you teach such mechanisms by explaining and practicing the steps involved in the order of the development process; see Table 1 (left). The student then always works toward the unknown. An alternative approach is *backward chaining*, where a process consisting of several steps is taught in reverse order, by starting with the *last* step, and successively adding *preceding* steps, one by one; see Table 1 (right). In backward chaining, the student always works toward familiar steps. Backward chaining is also applied successfully in sports, music, and military training.

Keywords

Abstraction, informatics, mathematics, physics, teaching, backward chaining