A review of "The Evolution of Programs"

Nachum Dershowitz, "The Evolution of Programs,
Birkhäuser, Boston-Basel-Stuttgart, 1983. 357 pp.

The printing of this book is atrocious enough to decide never to open again a book published by Birkhaüser. Type founts and the layout of pages, lines, and formulae are ugly and the individual characters are often printed with a quality that reminds one of The Compact Edition of the OED, viewed through a magnifying glass. For all, except the blind, this book is a pain to read. (If the author has provided the publisher with a camera-ready text, he shares the blame.)

The main chapters have the titles "General Overview", "Program Modification and Debugging", "Program Abstraction and Instantiation", "Program Synthesis and Extension", "Program Annotation and Analysis"; then we have 5 appendices "Global Transformations", "Program Schemata", "Synthesis Rules", "Annotation Rules", and "Implementation".

To give the flavour of the book I quote the opening paragraph of the Preface:

"Programs are invariably subjected to many forms of transformation. After an initial version of a program has been designed and developed, it undergoes debugging and certification. In addition, most long-lived programs have a life-cycle that includes modifications to meet amended specifications and extensions for expanded capabilities. Such evolutionary aspects of programming are the topic of this monograph. We present formal methods for manipulating programs and illustrate their application with numerous examples. Such methods could be incorporated in semi-automated programming environments, where they would serve to ease the burden on the programmer."

and from the chapter "Program Modification and Debugging" the Overview:

"For program modification, one needs a program satisfying its input-output specification along with the specification for a new program. Comparison of the two specifications may suggest a transformation of the given program that will bring it [in] line with the new specification. Even if the transformed program does not exactly fulfil the goal, it may serve as a basis for constructing the desired program."

Given a program and a specification, the claim that the former meets the latter has the status of a theorem without its proof. Not surprisingly, the author discovers that for the justification of modifications one usually needs the correctness proof of the initial program as well. And that is regarded as awkward: "Even if [the programmer] uses on of the current prototype verification systems, he is required to supply most, if not all, of the necessary invariant assertions, in addition to guiding the theorem prover. The goal of automatic program annotation is to relieve the programmer of some of this burden." The central position of the program as the programmer's product is also reflected in his remark "Ultimately [sic], it is the responsibility of the programmer to ensure the correctness of his product."

But it is a severe mistake to think that a programmer's products are the programs he writes; the programmer has to produce trustworthy solutions, and he has to produce and present them in the form of convincing arguments. Those arguments constitute the hard core of his product and the written program text is only the accompanying material to which his arguments are applicable.

By accepting as programmer someone that works under the motto "Code first, think later.", Dershowitz adopts the mistake, like so many of his confraters in Artificial Intelligence who try to develop "tools" for obsolete programming techniques. His saving graces are that he knows that part of his work is "putting the cart before the horse" and that he absolutely refuses to oversell what those wonderful "programming environments" might do for us in the future. And we may be grateful to him for his reminder that Artificial Intelligence does not operate in the forefront of Computing Science.

A final remark. György Pólya has now been accepted by so many —and Dershowitz is no exception— as the high priest of problem solving that doubting the wisdom of his teachings amounts to blasphemy. But when I read Pólya, each page showed me to have been written by a man who had never programmed and since then I am expecting the moment that we shall clearly see the extent to which Pólya's recommendations are products of his time. I see in Dershowitz's book an indication that we are approaching that moment.

Plataanstraat 5
5671 AL NUENEN
The Netherlands

27 February 1984
prof. dr. Edsger W. Dijkstra
Burroughs Research Fellow