Project:

Title: Review of "An Introduction to Formal Methods"

Ref: DS/FMU/RBJ/181 *Issue:* 1.1 *Date:* 1991-01-19

Status: informal *Type:* review

Author:

Name	*Location*	*Signature*	*Date*
R.B.Jones	WIN01		

Abstract: A review of "An Introduction to Formal Methods" by Antoni Diller.

Distribution: R.B.Jones ICL

0 DOCUMENT CONTROL

0.1 Contents List

0 DOCUMENT CONTROL **2**
0.1 Contents List 2
0.2 Document cross references 2

1 **2**

2 **2**

0.2 Document cross references

1

[?]

2

This book is presented as an introductory text on formal methods using the specification language Z. It is derived in part from courses given to first and second year undergraduates at the University of Birmingham. These parts cover, briefly *The Philosophy of Formal Methods*, an extended *Tutorial Introduction*, and more extended *Cases Studies*.

The book also covers more advanced topics, having a section on *Methods of Reasoning*, a section on *Specification Animation* and a *Reference Manual*.

The brief preliminary section on "Philosophy" serves as a prospectus against which I propose to evaluate the remainder of the book. The author expresses his conviction that "it is the fact that you can reason mathematically about Z specifications and prove results about them that is its main advantage over alternative specification methods". Here it is claimed that "specifications in Z are precise, unambiguous, concise and amenable to proof", and Z is advertised as "a pretty successful attempt.. ..to devise a notation for building models of software systems and for proving that programs meet their specifications".

The tutorial section (which accounts for about a third of the book) contains very painstaking explanations and illustrations of the basic features of Z and their use. The author has avoided any use of formality in the description of Z, even in the description of the syntax. This makes the book unsuitable as a reference work, since points of detail about the legality of various constructs cannot be settled. Constants defined in the Z library are introduced with informal descriptions rather than their formal definitions (the formal definitions may however be consulted in the reference section).

It is in the section on "Methods of Reasoning" that we might hope to find some support for the claims about Z which Diller makes in his introductory philosophising. This section consists of four chapters dealing in turn with *Formal Proof*, *Rigorous Proof*, *Immanent Reasoning* (reasoning about or within a single specification) and *Reification and Decomposition*.

The chapter on Formal Proof gives an incomplete account of formal reasoning in a language which appears to be a 'typed' first order predicate calculus. I say *appears to be* because no precise account of the syntax of the language is given, and though some of the rules have side conditions relating to the types of terms, no account whatsoever is given of how the type of a term is to be established. The term structure is not precisely defined, and important concepts are either completely untouched upon or given wholly inadequate treatment (such as that of *a free occurrence of a variable*, the description of which makes no mention of free occurrences in atomic formulae or in terms). Diller's account of formal reasoning, contrary to his claims, is quite inadequate for formal reasoning about Z specifications.

Diller only begins to treat reasoning in a language richer than first order predicate logic in his next chapter *Rigorous Proof*. It is suggested that rigorous proofs are capable of transformation into formal proofs, and Diller gives examples of reasoning about sets, and inductive reasoning about natural numbers and sequences. No explanation is offered of how such reasoning can be transformed into formal proofs in a first order language containing no mention of either sets or numbers.

Diller's further applications of these techniques in the remaining chapters of this section give no better grounds for confidence in the techniques he is offering. The very simple example specifications he is working with contain both typographical and substantive errors. The proofs avoid the most serious deficiencies in his formal system mainly because the examples are too trivial to bring them out, and yet there are non-trivial errors (he falsely claims, for example, on page 155, that "$\#d' \leq Max \land d' = \{\}$" *is equivalent to* "$\#\{\} \leq Max$") in informal proofs of plausible propositions.

In describing *Reification and Decomposition*, the statements which are alleged to show that a design correctly implements a specification would be readily provable if the predicate determining the system state in the design was inconsistent. No mention is made of an obligation to show this (or any other part of the specification) consistent.

The section on *Specification Animation* might better have been described as *case studies in prototyping*. It provides examples of implementations of Z specifications in *Miranda*^{TM} and *Prolog*. Animation has been advocated (elsewhere) as a way of improving ones understanding of the meaning of a specification. It may well do this if the animation is undertaken by a tool which faithfully reflects the semantics of the specification language. Here however, animation is achieved by manual implementation of the specification in a suitable programming language. This technique is therefore likely to implement what the author had intended to specify rather than what he actually did specify. This may well be valuable for identifying errors in the detail of the specification, but might better be described as prototyping than as animation.

The reference section covers broadly the same ground as the library descriptions in Spivey's reference manual, supplemented in some areas, particularly in relation to additions to the library which Diller has introduced for his own purposes. It does not provide any precise description of the syntax of Z, and for this reason alone fails to provide an adequate alternative to Spivey's reference manual. If the reader is expected to have a copy of the reference manual to hand then it is difficult to see why the author has not provided just those supplementary definitions which are not to be found there.

The appendices provide an account of the variable naming conventions consistently applied throughout the book, answers to all of the exercises, and an annotated bibliography.

It is regrettable that by trying to take more seriously the justification of formal specification languages by their suitability for formal reasoning Diller has been drawn into an indefensible position. Whatever

the intentions of those who have contributed to the development of Z, the language is not currently defined with the degree of precision necessary to support formal reasoning about specifications. Spivey's *Z reference manual* does not provide a complete account either of the static or the dynamic semantics of Z. His previous work, *Understanding Z*, provides a complete semantic account for only part of the language described in the reference manual and is clearly contradicted by the reference manual in various points of detail. Diller's contribution does not improve the situation. To move from first order predicate logic to *many sorted* first order predicate logic, even if his account of this system were complete, would be at best a neutral move so far as relevance to Z is concerned. Arguably it is misleading, because it introduces a notation (type assignments) which appears to be but is not the same as the superficially similar notations in Z (which are set membership constraints, not derivable in Z's type inference system, and not decidable).