DEWAR79

Dewar, Robert B.K. "The SETL Programming Language" (l979), 1176 pages. SETL was first defined by Jacob T. Schwartz [3] in an attempt to bring the conciseness and precision of mathematics, especially that of set theory, into the world of programming. The next ten years saw a great deal of work on and with the language, especially in trying to facilitate and even automate the choice of low-level data structures, invisible to the programmer, that would make SETL programs run nearly as fast as if they had been coded "by hand" into a lower-level language with explicit "computer-oriented" representations such as pointers and has tables. The results were very encouraging, and a great deal was learned about optimization in the process. Unfortunately, however, the only "language manuals" were A SETLB Primer [1], a well-intended document that hid the beauty of the language (actually, a subset thereof) behind the ugly face of the IBM 029 keypunch and its extraordinary mapping to the character set of a 63-character CDC line printer, and Schwartz's own 675-page On Programming [4], a compilation of philosophical musings, language definitions, algorithms, and implementation details which, though impressive in the best sense of the word, presented a formidable front to the novice programmer. Therefore Dewar's tutorial and reference guide, the subject of this review, comes as a welcome introduction to this simple, yet very powerful and truly "high-level" programming language. Chapter 1 gives a sufficient introduction to the language to allow one to begin writing significant programs. It is very exhilarating to read, as it introduces sets, tuples, maps, simple and compound operators, iterators, existential and universal quantified tests, set and tuple formers, and all the usual control structures and basic data types found in most modern "high-level" programming languages. Chapter 2 works through a topological sorting example in order to teach the elements of good SETL programming style. Here we find perhaps the most important piece of advice, "Use maps wherever possible...find the maps, they are always there?" Indeed, Dewar might have been addressing a beginning student of mathematics. Subsequent chapters formalize and make complete the material introduced in Chapter 1, and are well stockedd with examples. As the book unfolds, it becomes increasingly clear that SETL richly deserves the title "very high level language"--algorithms in graph theory, for instance, typically translate directly from their abstract set-theoretical descriptions into SETL, with no more than superficial syntax transliterations. The issue of executional efficiency is addressed in Chapter 6, which (someday!) is supposed to describe the SETL Representation Sublanguage. There are no details of this sublanguage in the current(?) version of the book, apart from the syntax descriptions in the BNF-like grammar at the back. Despite the lack of an index or table of contents, this book is an excellent tutorial, and its logical organization makes it a rather good reference manual for the SETL language. It is a pity that it does not appear to have been formally published. This reviewer has discovered only one article [2] which cites the book by name, and there it is still "to appear". None of the three subsequent SETL-related papers in TOPLAS mentioned it at all, and a search of other current literature proved fruitless also. The only version known to be available at the University of Toronto at the time of this writing (please see David Bacon, Sandford Fleming Building, Room 4008K) is a photocopy of a photocopy (of a photocopy?) made by the systems programming staff at the Computer Centre of the University of British Columbia. (It is still surprisingly easy to read, though getting a little ragged!)

References Mullish, Henry, and Max Goldstein, A SETLB Primer, Courant Institute of Mathematical Sciences, New York University (1973), 201 pages. Schonberg, Edmond, Jacob T. Schwartz, and Micha Sharir, "An Automatic Technique for Selection of Data Representations in SETL Programs", ACM Transactions on Programming Languages and Systems 3, 2(1981), 126-143. Schwartz, Jacob T., Abstract Algorithms and a Set-Theoretic Language for their Expression, Courant Institute of Mathematical Sciences, New York University (1970-1), 289 pages. Schwartz, Jacob T., On Programming, courant Institute of Mathematical Sciences, New York University (1975), 675 pages.