Bibliography Prettyprinting and Syntax Checking

Nelson H. F. Beebe
Center for Scientific Computing
Department of Mathematics
University of Utah
Salt Lake City, UT 84112
USA
Tel: +1 801 581 5254
FAX: +1 801 581 4148
Internet: beebe@math.utah.edu

Abstract

This paper describes three significant software tools for BibTeX support. The first, `bibclean`, is a prettyprinter, syntax checker, and lexical analyzer for BibTeX files. The second is `biblex`, a lexical analyzer capable of tokenizing a BibTeX file. The third is `bibparse`, a parser that analyzes a lexical token stream from `bibclean` or `biblex`.

The current BibTeX implementation (0.99) is based on a vague and ambiguous grammar; that situation *must* be remedied in the 1.0 version under development. Rigorous lexical analyzer and parser grammars are presented in literate programming style, and implemented as `biblex` and `bibparse` using modern software tools to automatically generate the programs from the grammars. `bibclean` also implements these grammars, but with hand-coded parsers that permit it to apply heuristics for better error detection and recovery.

Extensions of the current BibTeX for comments, file inclusion, a `Periodical` bibliography entry, and ISSN and ISBN entry fields, are proposed and supported in these tools.

The impact of much larger character sets is treated, and grammatical limitations are introduced to ensure that an international portability nightmare does not accompany the move to these character sets.

`bibclean` is extensively customizable, with system-wide and user-specific initialization files, and run-time-definable patterns for checking BibTeX value strings. A customized pattern-matching language is provided for this purpose. `bibclean` can also be compiled to use regular-expression patterns, or none at all.

All code is written in the C programming language, and has been tested for portability with more than 40 C and C++ compilers on several major operating systems. The distribution includes a large suite of torture tests to check new implementations. It is not necessary for installation to have the lexical analyzer and parser generator tools that process the grammars; their output code is included in the distribution.

The complete paper is too long for the TUG'93 Conference Proceedings issue; it will instead appear in the next issue of *TUGboat*.