A LaTeX to HyperCard Translator *

Panagiotis T. Metaxas and Kanyi Zhao
Computer Science Department,
Wellesley College
pmetaxas@wellesley.edu, http://www.wellesley.edu/CS/pmetaxas.html

1 Introduction

Electronic publications are becoming widely available, and it is expected that in the not-so-distant future most printed books will have a hypermedia version. Many commercial editors have appeared in the market; almost none of them, however, deals effectively with mathematical text in the context of a multimedia system. Some of them are being used as extensions to word processing programs, while others produce postscript output that is not editable and essentially is good only for previewing. Mathematicians and computer scientists overwhelmingly use TeX or LaTeX [L86] as the editor of preference when writing their papers. The advantage of LaTeX is that it contains a set of easy-to-memorize commands to describe the logical structure of the article. Its main drawback is that, like TeX, it is not a *wysiwyg* language, and its source code looks uninviting to users. In general, LaTeX text looks nice when typesetted and printed out on a postscript printer but not when created.

When one needs to use mathematical notation in a hypermedia system, as we did in [G+93, C+94] there are no easy solutions available. This paper describes an effort to build, on top of HyperCard, a system that can read and display mathematical text from LaTeX source. Such a system will be useful in multimedia mathematics books in the future.

2 Our Approach

We decided to use the HyperTalk 2.2 language to produce text in standard fields making the translator portable to other stacks. The steps we followed are essentially those of creating the parser of a compiler. The actions taken by this parser are HyperTalk commands that format individual lines, fields, or even characters of the input.

We edited a few sample fonts that correspond to the mathematical notation we wanted to tackle. Then, we created a grammar that closely imitates the actions taken by a LaTeX typesetter. Failure of parsing these strings indicates erroneous input. The grammar actions correspond to HyperTalk commands. A set of additional handlers is included, that implements actions to be taken at different points of the processing. We have found that our translator can correctly parse typical (yet restricted) LaTeX input. Given the object-oriented style of implementation, it is possible to extend the parser so that it parses more complicated mathematical expressions. Its limitations are essentially the limitations of the hypercard environment.

We note that, since the code of the translator is interpreted, it is very slow. (We did not use XCMDs so that the code can be ported to ToolBook.) To compensate for this, we display the output as it is generated by the translator, and the user watches an animation of the typesetting process. For example, the text first appears on the screen unformatted, and then some portions change size and style; the mathematical notation appears as if it is "typed" by the program. Even though it is slow, it is a process that needs to be performed only once. If speed is important, an XCMD implementation in C is more appropriate, using tools like lex and yacc.

3. References

[L86] Leslie Lamport *LaTeX User's Guide & Reference Manual*, Addison-Wesley, Reading, MA 1986.
[G+93] P. Gloor, F. Makedon, J. Matthews, D. Johnson and P. Metaxas (eds.) *Parallel Computing*, (CD-ROM), TELOS/Springer-Verlag, CA, 1993.
[C+94] M. Cheyney, P. Gloor, D. Johnson, F. Makedon, J. Matthews, and P. Metaxas. *CONFERENCE ON A DISK: A successful experiment in Hypermedia Publishing (extended abstract)*, Educational Multimedia and Hypermedia, 1994, pp. 129-134

* Partial support for this project was provided by a summer grant from the Wellesley College Science Center and by a grant from the New England Consortium on Undergraduate Science Education.