What parsers for (La)TeX mathematics exist outside of the TeX engines?

Inspired by the author's motivation for asking Is there a BNF grammar of tex language.

Are there any well done libraries that can parse some subset of TeX mathematics independently of the TeX engine? Important points to consider in answers:

- How large of a subset of TeX mathematical notation is supported?
- Is the parser portable? Does it have any dependencies?
- Is the parser closely tied to a particular backend or could it easily be used to support multiple output formats. In other words, how easily could it be integrated into a new system that had to support support output to PDF, HTML, PNG, etc.

For example, I know of the following parsers but not much about their applicability outside the use cases for which they were designed (Matplotlib graphics and math rendering in web browsers):

- The mathematical expression handler in Python's Matplotlib.
- The MathJax rendering library for JavaScript.

{math-mode} {tools} {parsing} {mathjax}

edited Nov 21 '10 at 20:56

Stefan Kottwitz ♦
102k 27 343 551

asked Oct 17 '10 at 17:29

Sharpie
7,583 1 27 39

2 Answers

I've been looking into this too, so I'll share some observations that fall rather short of a proper answer, which would really involve looking at a whole lot of source code and asking the right questions about it.

Parsers generating HTML+Math ML

1. Nick Drakos & Ross Moore's Latex2html converter, written in Perl, which I think was the first converter to map equations to Math ML. In 1998, Ross Moore outlined his goals for Latex2html, tied to the now defunct, closed-source WebEq mathematics rendering software, and Webtex, which was an alternative syntax for mathematics designed for use in web pages. From the WebEq documentation: *WebTeX always translates unambiguously into MathML, while LaTeX does not*.
2. itex2mml, in C by Paul Gartside & others, also based on Webtex, but with support for some Latex not supported in Webtex.
3. tex4ht, written in C by Eitan Gurari and other eminent figures. It avoids having to parse Latex source by running `latex` with modified macros that insert specials into the DVI output, and parses the DVI output instead.
4. John McFarlan's Pandoc, as mentioned by Aditya, written in Haskell. Note that Pandoc supports generation of HTML, both with and without Math ML.
5. MathJax allows generation of Math ML besides the usual boxes plus image fonts output. It has an impressive degree of support for Latex, including limited support for user macros.

Parsers generating XML

Jason Blevins has a list of tools that convert Latex documents to XML-based formats, and that handle equations reasonably. Romeo Anghelache's Hermes, which is part of a full Latex parser that generates XML with semantic markup, is worth singling out: like tex4ht, it works by running the Tex engine with macros to put specials in the DVI output, which it then parses; it supports a wider set of semantic markup.

Fragments of Latex or DVI

With the exception of the systems referencing Webtex, there doesn't seem to be much interest in clearly codifying subsets of Latex to be parsed, I guess because these are regarded as moving targets. Instead, lists of commands supported, like that I mentioned for Mathjax, seems to be the way things are done.

With DVI-based converters, the issue of parsing Latex goes away, replaced by the relatively trivial issue of parsing marked-up DVI and the trickier issue of identifying the semantically significant macros and constructing markup-issuing replacements that do not improperly interfere. I haven't looked at how this is done for equational layout. It would be a useful exercise to see how a converter from Tex formulae to those of It's worth noting that the representation of expressions is essentially a superset of that used by Heckmann & Wilhelm (1997) would work.

Syntax highlighting

A completely different kind of parsing is involved in syntax highlighting, where the idea is to help the author see the significance of the parts of the formulae. I don't know of any syntax highlighters that do an interesting job here: Auctex only raisers/lowers super&subscripts, but i haven't really looked.

Reference

Heckmann & Wilhelm, 1997, A Functional Description of TeX's Formula Layout.

edited Oct 19 '10 at 11:47

answered Oct 19 '10 at 10:17

Charles Stewart
12.6k 1 37 77

I know a little about itex2MML. Firstly, it's list of supported commands is very well documented (via the link that you have). Secondly, it's **not** written in Ruby; it's a C library and the current maintainer (Jacques Distler) is most interested in the Ruby extensions. I've successfully compiled Perl, PHP, and Python extensions. Thirdly, its output is naturally MathML but it can be coaxed into producing SVG or PNG. – Andrew Stacey Oct 19 '10 at 10:28

@Andrew – Ruby: oops, fixed; apart from Pandoc and tex4ht, I've not looked at all at the implementation of these, so what I say should be taken with a pinch of salt. About documentation: what I had written was obviously was unclear; what I meant was that Webtex was well-specified, and the two systems, itex2mml and latex2html, that were based on it, were clear as a result. I hope this is now clear in my answer. – Charles Stewart Oct 19 '10 at 11:54

Pandoc uses the Haskell Text.TeXMath.Parser library to parse inline and display math. This is not complete. It only parses most common inline math expressions and does not support amsmath display environments.

- I don't know if there is an official documentation of what subset is supported. The source code will give some idea about that.
- It is as portable as Haskell. So, it should work on most popular OS.
- Pandoc is specifically designed to support multiple output formats. IIRC, the output can be translated to MathML or to images using mimetex, gladtex, etc.

answered Oct 17 '10 at 19:00

Aditya
34.6k 1 61 137

It's worth noting that the representation of expressions is essentially a superset of that used by Heckmann & Wilhelm, 1997, A Functional Description of TeX's Formula Layout. – Charles Stewart Oct 19 '10 at 8:05
