Conserving Implicit Mathematical Semantics in Conversion between TEX and MathML

Stephen M. Watt
Ontario Research Center for Computer Algebra
University of Western Ontario, London, Canada N6A 5B7
Stephen.Watt@uwo.ca
http://www.csd.uwo.ca/~watt

Abstract

MathML[1] is an XML representation for mathematical objects, allowing expressions to be stored in databases, transmitted between applications and operated upon by programs. MathML can be used to express mathematical content in web pages and digital libraries, and has become an accepted form for input and output of computer algebra systems. There have been several efforts to design software for conversion of mathematical expressions from TEX to MathML and vice versa. We consider the problem of how conversion between formats can conserve any mathematical semantics implied by the markup of the original document.

Both TEX and MathML admit macro mechanisms, natively so in TEX and via XSLT[2] with MathML. Authors may use pre-defined style-sheets or define their own abbreviations, effectively extending the vocabulary of the environment. Macros are typically used as shorthands for lengthy expressions or to maintain notational independence. A simple example of notational independence would be, e.g., to define `\Vector` to expand either to `\mathbf{v}` or `\vec{v}` or something else, depending on the style sheet used.

Any serious converter between TEX and MathML must support macros. The standard approach has been to expand macros, and then perform the translation from low-level TEX to MathML (or vice versa). We see this approach as undesirable, as the use of macros in practice captures mathematical semantics within expressions. For example, if the form `\BesselJ{\alpha}{z}` expands to $J_\alpha(z)$, and translates to

```
<mrow>
    <msub><mi>J</mi><mi>&alpha;</mi></msub>
    <mfenced><mi>z</mi></mfenced>
</mrow>
```

then we have lost the fact that J is a Bessel function. The MathML-processing application will have no way to determine that this is a Bessel function, and not something else, e.g., an angular momentum or a jet bundle.

Our approach has been quite different: We assume that in many interesting cases it shall be possible to map macros in one setting to corresponding macros in another, thus conserving implied semantics. We have adopted the hypothesis that certain TEX style or class files will naturally have counterpart XSLT style-sheets. Particular TEX macros then correspond to specific XSLT template definitions. The conversion process in either direction must be able to recognize and use these correspondences. Additionally, we provide fine control over which macros are to be carried over to the target markup and which are uninteresting shorthands to be expanded.

This paper presents the overall architecture of a pair of programs — one converting mathematical TEX to MathML[3] and one converting MathML to TEX[4]. Both of these programs permit macros to be translated at a high semantic level. These may be specified individually, or as sets corresponding to style files. Other macros are expanded to primitive forms for translation.

Declarative mapping files are used to specify the correspondence between TEX and XML forms, typically for TEX macros and XSLT templates, and the same file may be used for conversion in either direction. Multiple mapping files may be used, if desired, corresponding to multiple macro definition files. We discuss how this architecture can be used with content, presentation or combined MathML markup, and how the programs can be used in conjunction with other tools for conversion between TEX and other XML formats.

— — ∗ — —

References

[1] Mathematical Markup Language (MathML) Version 2.0, D. Carlisle, P. Ion, N. Poppelier and R. Miner (eds), R. Ausbrooks, S. Buswell, S. Dalmas, S. Devitt, A. Diaz, R. Hunter, B. Smith, N. Soiffer, R. Sutor and S. Watt, World Wide Web Consortium Recommendation, 21 February 2001, http://www.w3.org/TR/2001/REC-MathML2-20010221.

[2] XML Transformations (XSLT) Version 1.0, J. Clark (ed), World Wide Web Consortium Recommendation, 16 November 1999, http://www.w3.org/TR/1999/REC-xslt-19991116.

[3] A TEX to MathML Converter, I. Rodionov and S. Watt, http://www.orcca.on.ca/MathML/texmml/textomml.html.

[4] A MathML to TEX Converter, E. Smirnova and S. Watt, http://www.orcca.on.ca/MathML/texmml/mmltotex.html.