Journal of Symbolic Computation 71 (2015) 1–46

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Towards semantic mathematical editing ☆

Joris van der Hoeven

LIX, CNRS, École polytechnique, 91128 Palaiseau Cedex, France

ARTICLE INFO

Article history:
Received 30 November 2013
Accepted 25 May 2014
Available online 2 October 2014

MSC:
68U15
68U35
68N99

Keywords:
Mathematical editing
Content markup
Packrat parsing
Syntax checker
Syntax corrector
TeXmacs

ABSTRACT

Currently, there exists a big gap between formal computer-understandable mathematics and informal mathematics, as written by humans. When looking more closely, there are two important subproblems: making documents written by humans at least syntactically understandable for computers, and the formal verification of the actual mathematics in the documents. In this paper, we will focus on the first problem.

For the time being, most authors use TEX, LATEX, or one of its graphical front-ends in order to write documents with many mathematical formulas. In the past decade, we have developed an alternative wysiwyg system GNU TEXMACS, which is not based on TEX. All these systems are only adequate for visual typesetting and do not carry much semantics. Stated in the MATHML jargon, they concentrate on presentation markup, not content markup.

In recent versions of TEXMACS, we have started to integrate facilities for the semantic editing of formulas. In this paper, we will describe these facilities and expand on the underlying motivation and design choices.

To go short, we continue to allow the user to enter formulas in a visually oriented way. In the background, we continuously run a packrat parser, which attempts to convert (potentially incomplete) formulas into content markup. As long as all formulas remain sufficiently correct, the editor can then operate both on a visual or semantic level, independently of the low-level representation being used.

An important related topic, which will also be discussed at length, is the automatic correction of syntax errors in existing mathematical documents. In particular, the syntax corrector that we have implemented enables us to upgrade existing documents and test

☆ This work has been supported by the ANR-09-JCJC-0098-01 MAGIX project, as well as a Digiteo 2009-36HD grant sponsored by Région Ile-de-France.

E-mail address: vdhoeven@lix.polytechnique.fr.
URL: http://lix.polytechnique.fr/~vdhoeven.

http://dx.doi.org/10.1016/j.jsc.2014.09.040

our parsing grammar on various books and papers from different sources. We will provide a detailed analysis of these experiments.

1. Introduction

1.1. The main challenge

One major challenge for the design of mathematical text editors is the possibility to give mathematical formulas more semantics. There are many potential applications of mathematical texts with a richer semantics: it would be easier and more robust to copy and paste formulas between a text and a computer algebra system, one might search for formulas on websites such as WIKIPEDIA, various kinds of "typos" in formulas can be detected automatically while entering formulas, etc.

Currently, most mathematicians write their documents in TEX, LATEX, or one of its graphical frontends (Knuth, 1984; Lamport, 1994; Braun et al., 2003; Ettrich et al., 1995; MacKichan Software, 1998). Such documents usually focus on presentation and not on mathematical correctness, not even syntactic correctness. In the past decade, we have developed GNU TEXMACS (van der Hoeven et al., 1998; van der Hoeven, 2001) as an alternative structured wysiwyg text editor. TEXMACS does not rely on TEX or LATEX, and can be freely downloaded from `http://www.texmacs.org`. The main aims of TEXMACS are user-friendliness, high quality typesetting, and its use as an interface for external systems (Grozin, 2001; Audebaud and Rideau, 2004; Grozin, 2005; Mamane and Geuvers, 2006). However, until recently, mathematical formulas inside TEXMACS only carried presentation semantics.

Giving mathematical formulas more semantics can be regarded as a gradual process. In fact, one may distinguish three main semantical levels for mathematical documents (see also Padovani and Zacchiroli, 2006, Figure 1 for a similar classification):

Presentation level. At this level, it is only specified how formulas should be rendered on paper or on a screen. LATEX and presentation MATHML (W3C, 1999) are standard data formats for doing so.

Syntactical (or content) level. At this level, it is also specified how formulas should be parsed from a syntactical point of view. For instance, the formula $a-b+c$ is usually parsed as $(a-b)+c$ and not as $a-(b+c)$. Lisp/Scheme expressions are a convenient way to specify the syntactic structure of formulas, and so is content MATHML. This kind of syntactical semantics is often sufficient for the symbolic manipulation of mathematical formulas using computer algebra software (Maplesoft, 2005–2012; Wolfram, 1991; Axiom, 1971; Maxima, 1998; Stein et al., 2004).

Full semantical level. At this level, the semantics of every formula and subformula is completely specified. For instance, considering the formula $a-b+c$, we might specify that a, b and c are integer variables and that $+$ and $-$ stand for integer addition and subtraction. Full semantics is usually required by proof checkers and automated theorem provers (de Bruijn, 1970; Coquand et al., 1984; Owre et al., 1992; Nipkow et al., 1993).

Various more subtle intermediate levels could also be envisaged, such as one for formulas with presentation semantics, but involving only syntactically non-ambiguous symbols. At such a level, it would for instance be specified whether $\wedge$ is used as a conjunction or as a wedge product (since each interpretation implies a different syntactical parsing rule). However, addition of integers and matrices behaves in a syntactically similar way, so a unique symbol $+$ suffices for this purpose.

Warning. In this paper, we will only be concerned with syntactical semantics of formulas, corresponding to the syntactical level (so roughly speaking to content MATHML). Unless stated otherwise, "semantics" will therefore mean "syntactical semantics".