SMT-LIB in XML clothes

Filip Marić and Predrag Janičić

Faculty of Mathematics, University of Belgrade, Studentski trg 16, 11000 Belgrade, Serbia and Montenegro, {filip,janicic}@matf.bg.ac.yu

Abstract

In this paper we propose how to further improve SMT-LIB standards (for the field of decision procedures). We propose using XML for all SMT-LIB standards and we believe that this promising, already widely used framework, would also make SMT-LIB easier to use, more powerful, and more popular. All that should hopefully help in advancing the field of decision procedures. We provide arguments for this proposal and also present the support for SMT-LIB in XML form. This support has been implemented within our ARGO-LIB project.

1 Introduction

After three decades of research in use of decision procedures in automated reasoning, this (sub)field now has a plenty of both theoretical and practical results and is already rather well-established. Most (if not all) of the state-of-the-art proving systems have some support for decision procedures. Nowadays, researchers in the field of decision procedures (and potential users) need ways for easier exchanging of ideas, benchmarks, test results, implementations, abstract representation of algorithms etc.

The SMT-LIB initiative [2] follows these needs and already gathers many (if not most) researchers in this field. The main goal of the initiative, supported by a growing number of researchers world-wide is to produce a library of benchmarks for satisfiability modulo theories and all required standards and notational conventions. Such a library will facilitate the evaluation and the comparison of different approaches for using decision procedures and advance the state of the art in the field. The progress that has been made so far supports these expectations. In SMT-LIB the background logic is first order classical logic with equality (while it also allows sorted logic to more easily express benchmarks). The existing SMT-LIB standard proposes a LISP-like syntax for describing benchmarks and theories.

XML is the Extensible Markup Language. It is called extensible because it is not a fixed format like HTML (a single, predefined markup language). Instead, XML is actually a "metalanguage" — a language for describing other

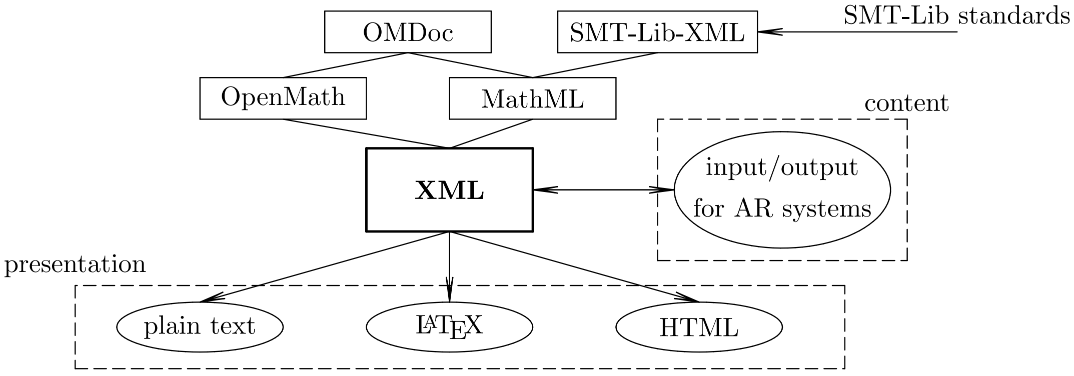

Fig. 1. Logical organization of XML systems

languages, which lets one design his/her own customized markup languages for limitless different types of documents. XML is intended "to make it easy and straightforward to define document types, easy to author and manage documents, and easy to transmit and share them across the Web". However, XML is not just for Web pages: it can be used to store any kind of structured information, and to enclose or encapsulate information in order to pass it between different computing systems. An XML document can carry both presentation (i.e., plausible visualisation) and content information.

XML is a project of the World Wide Web Consortium and is a public format — it is not a proprietary development of any company. Almost all browsers that are currently in use support XML natively. A DTD is a formal description in XML Declaration Syntax of a particular type of document (i.e., of its syntactical restrictions). It sets out what names are to be used for the different types of element, where they may occur, and how they all fit together. This formal description enables automatic verification ("validation") of whether a given document meets the given syntactical restrictions.

XML comes with the XSLT document processing language that is used to transform the input XML documents i.e., the input files to the desired output documents. An XSLT style-sheet declares a set of rules (templates) for an XSLT processor to use when interpreting the contents of an input XML document. These rules tell the XSLT processor how that data should be presented – as an XML document, as an HTML document, as plain text, or in some other form.

MathML is the Mathematical Markup Language. It is an XML application for describing mathematical notation and capturing both its structure and content. It provides a much needed foundation for the inclusion of mathematical expressions in Web pages. However, MathML is not only used in Web pages but also for storing general purpose mathematical documents. Several markup schemes for mathematical documents have been developed. For example, OMDoc is a scheme for describing various mathematical documents including articles, textbooks, interactive books and courses. OMDoc uses MathML, and a similar scheme called OpenMath to describe mathematical formulae (see Figure 1).

2 SMT-LIB-XML

We believe that the syntax of SMT-LIB benchmarks and theory descriptions should be adapted and put into the XML framework. All (or almost all) restrictions stated in the SMT-LIB specifications can be formally represented in a DTD document, saying what syntactical constraints each benchmark has to meet. We also propose using a subset of MathML for describing the formulas themselves in the benchmarks. The main reason for this is the fact that MathML is becoming a standard markup-scheme for mathematical formulae, and the fact that it has wide support in existing software which is constantly growing. The general MathML can be restricted with respect to SMT-LIB requirements. This DTD can then be used, in conjunction with the generic XML validation mechanism, for verifying whether a given benchmark is legal. The same applies to representation of theories (or other operational content, such as rewrite rules).

By means of XSLT, XML representation of benchmarks and theories can be easily transformed into HTML format that is convenient for human-readable display in browsers. It can also be transformed into any specific input/output representation, required by any automated reasoning system (see Figure 1).

3 ARGO-LIB

ARGO-LIB[1] is being developed as a flexible, modular, and efficient generic platform for using decision procedures aimed at *realistic* use both in academia and in industry [1]. It provides support for a range of decision procedures (for a range of theories) and also for different techniques for using decision procedures. ARGO-LIB is implemented in the standard C++ programming language. It can work stand-alone but can also be simply integrated into some other tool (e.g., a theorem prover, constraint solver, model checking system etc.).

ARGO-LIB has support for a library of theories and conjectures and support for benchmarking. In this sense, ARGO-LIB builds on motivations, ideas, and standards promoted by the SMT-LIB initiative. ARGO-LIB uses the SMT-LIB format as its native input format (for theory representations and benchmark representations), while there are still several additions to SMT-LIB.

ARGO-LIB also provides a prototype support for SMT-XML as proposed in §2. This support includes required DTD and XSLT files and a suite of tools for converting from and to XML from different formats (e.g. SMT-LIB format).

[1] The web page for ARGO-LIB, including source files and documentation is at: `www.matf.bg.ac.yu/~janicic/argo`. There is also available a longer version of this paper.

4 Conclusions and Future Work

In this paper we proposed and advocated the use of XML in SMT-LIB. We gave a brief description of SMT-LIB, XML and MathML, and why the use of XML would be beneficial for SMT-LIB. Some of the main advantages of using XML and MathML would be:

- instead of raw, plain text representation, SMT-LIB files will be stored in strictly structured files; these files will be easy to parse, process, and convert into different forms and formats;
- input/output tasks will be supported by generic, external tools and different automated reasoning systems will communicate easily;
- easier communication and exchange of material with the rest of mathematical and computer science community;
- wide and growing support for XML;
- different sorts of presentation (text form, LaTeX form, HTML) easily enabled;
- strict content validation of documents with respect to given restrictions.

We are planning to further improve the current ARGO-LIB support for XML. We are planning to use only XML files (for theories, benchmarks, rewrite rules, etc.) both for input and for output, while we will use XSLT for converting output files into LaTeX format, text format, or HTML format. This would make the whole system very flexible and all presentation issues would be subject to changes in external (XSLT) files and not in the source code itself. At the same time, different sorts of validations of content will be possible. We will also work on further potentials of the validation mechanism and try also to verify some semantic conditions.

By this paper we advocate moving all SMT-LIB standards to the XML framework and we also call upon the SMT-LIB interest group to support that move.

References

[1] Filip Marić and Predrag Janičić. ARGO-LIB: A generic platform for decision procedures. In *The 2nd International Joint Conference on Automated Reasoning (IJCAR-2004)*, LNAI, Springer, 2004.

[2] Silvio Ranise and Cesare Tinelli. The SMT-LIB Format: An Initial Proposal. 2003. on-line at: http://goedel.cs.uiowa.edu/smt-lib/.