Combining Mizar and TPTP Semantic Presentation Tools

Josef Urban*
Charles University in Prague

Steven Trac, Geoff Sutcliffe, Yury Puzis
University of Miami

Abstract

This paper describes a combination of several Mizar-based tools (the MPTP translator, XSL style sheets for Mizar), and TPTP-based tools (IDV, AGInT, SystemOnTPTP) used for visualizing and analyzing Mizar proofs. The combination delivers to the readers of the Mizar Mathematical Library (MML) an easy, powerful, and almost playful way of exploring the semantics and the structure of the library. The key factors for the relative easiness of having these functionalities are the choice of XML as both internal and external interface of Mizar, and the existence of a TPTP representation of MML articles. This clearly shows the great added value that can be obtained by cooperation of several quite diverse (and quite often separately developed) projects, provided that they are based on the same communication standards.

1 Instead of Reading This Paper

Perhaps the first thing a reader of this paper should do is to play with the functionalities that have been implemented. These functionalities provide an easy, powerful, and almost playful way of exploring the semantics and the structure of the Mizar Mathematical Library (MML) [Rud92]. Select one of the HTML files at `http://www.tptp.org/MizarTPTP/`, e.g., the MML article about the Boolean Properties of Sets, `xboole_1`. This will show the HTML rendition of the article, an extract of which is shown in Figure 1. Provided that Java 1.5 is installed and available to the browser, clicking on the palm tree icon next to a theorem will run the Interactive Derivation Viewer (IDV) [TPS06] applet to display the TPTP form [SSCVG06] of the Mizar proof tree.[1]

Figure 2 shows the IDV window for the first theorem (`Th1`) in `xboole_1`. The many IDV functionalities available there are described later in this paper, on the other hand, many of them are quite self-explanatory and easy to explore. One of them, which might be particularly interesting to "semantically oriented" users, is the verification functionality. The sequence of interactions is shown in Figure 3. Click the "white tick" ("show verified formulae") icon (it turns green), and then the "hurricane flags" icon on its right-hand side ("verify all formulae"), accept the default EP system [Sch02] as the verification ATP system in the pop-up window, and click the "GO CANES" icon in the pop-up window. Green ticks will start to appear in the IDV window, denoting that the (TPTP form of the) Mizar inference have been verified by the GDV derivation verifier [Sut06], using EP for checking logical consequences. Click the "hurricane flags" again to stop the verifications.

Going back to the HTML presentation and clicking on the "hammock between palm trees" icon will similarly call IDV, now displaying the overall theorem structure of the article. If you think that this is not especially interesting, click this icon in `jgraph_7` (it

*Supported by Marie Curie International Fellowship within the 6^{th} European Community Framework Programme

[1] See the IDV video - `http://www.cs.miami.edu/~geoff/ResearchProjects/ART/IDVVideo.mov`.

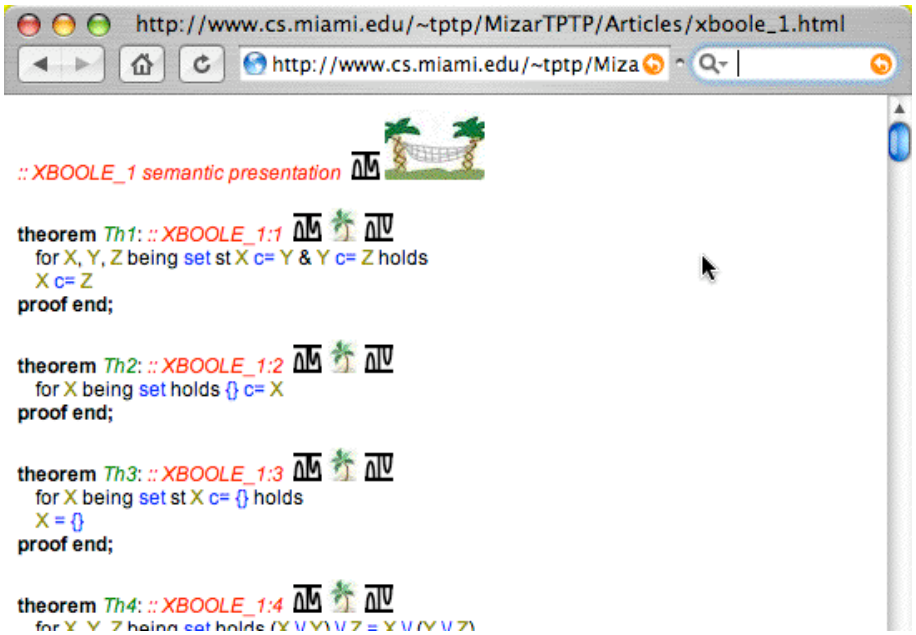

Figure 1: Article `xboole_1`

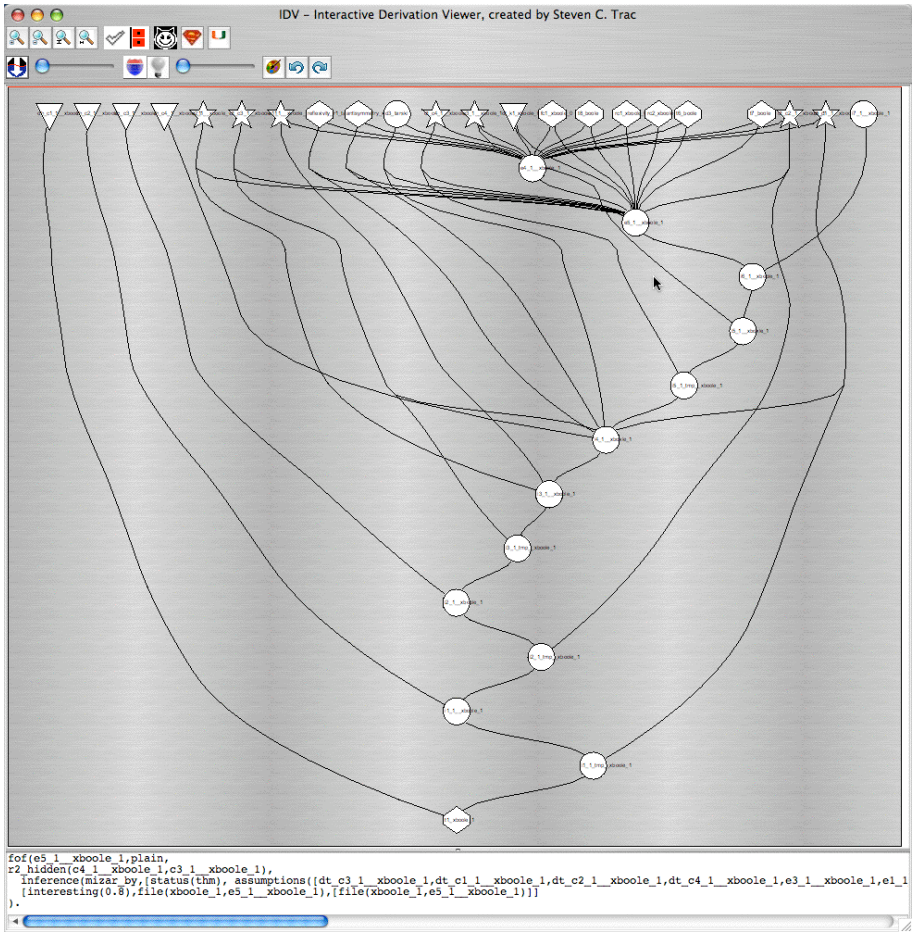

Figure 2: Theorem `Th1` in Article `xboole_1`

will take a while to load the applet), to get the IDV window shown in Figure 4. Would you be able to say just by looking at the HTML (or ASCII) presentation that the Mizar article [NT05] has this particular derivation structure, and be motivated to explore (and perhaps criticize) the reasons why it is so?

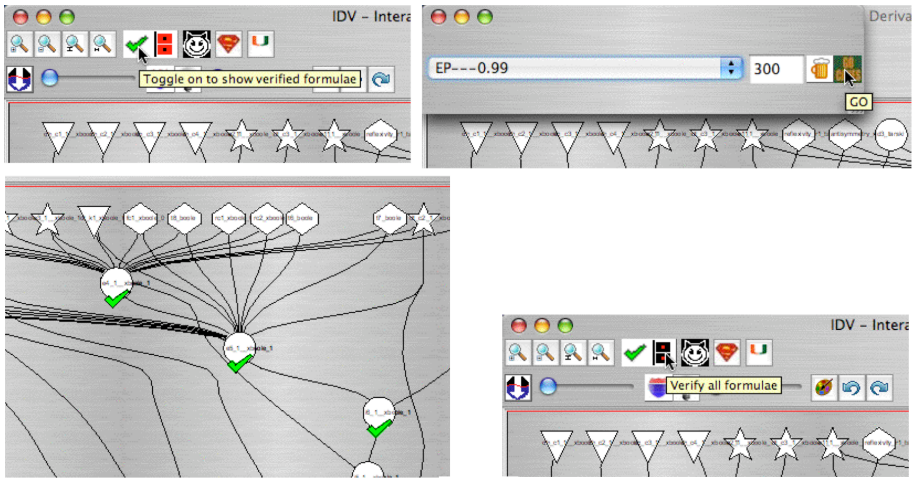

Figure 3: Verifying `Th1` in Article `xboole_1`

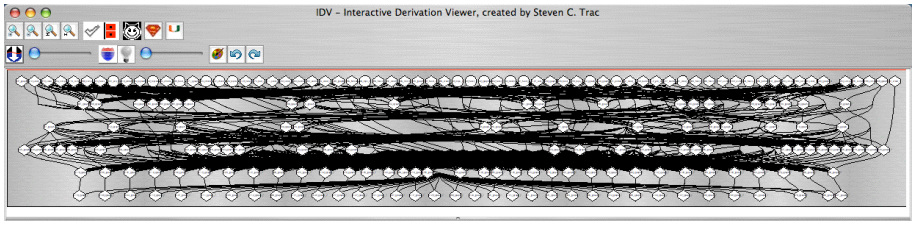

Figure 4: Article `jgraph_7`

2 Motivation and Overview

There has been quite a lot of work recently on translating the MML to the TPTP format, and on making the TPTP format sufficiently rich for this task. The goal is to make the MML accessible to the automated theorem proving (ATP) systems that either directly, or through the TPTP translation tools, understand the TPTP language. The systems can then in turn be used for proof assistance over the MML, its independent verification, refactoring, and many more interesting AI tasks. Similarly, the (XSL-based) HTML presentation of the Mizar library has been continuously developed, with the goal to make it a useful tool for its readers and authors.

There are several other projects aimed at translating large formal corpora to TPTP format, and at reaping the benefits from the unified TPTP interface to ATP systems and tools. Examples include the Isabelle proof assistant, [MP06], the SUMO ontology, [NP01], and the Cyc knowledge base [MJWD06]. The advantages of developing and using tools that work directly with the TPTP format are obvious. While the SystemOnTPTP interface for solving ATP problems [Sut00] has been well known in the ATP community for a long time, there has also been a significant recent development of tools working with TPTP

format derivations. IDV is a tool for graphical presentation of TPTP format derivations, and provides an interface for analysis and verification of derivations. Another of IDV's functionalities is its link to the AGInT system [PGS06], which assigns interestingness values to derived formulae, based on several AI heuristics. This can be used by IDV to compact large derivations into smaller presentations of the most interesting facts and the links between them. Graphical presentation of a derivation allows a user to quickly get a feel for the structure of the derivation, and interact with the derivation in a more natural way than is possible with a text presentation.

In short, the work presented here uses the existing (and continuously developed) semantic link between Mizar and TPTP, and capitalizes on that link by re-using the IDV, SystemOnTPTP, and AGInT systems, for additional semantic presentation purposes. In the following section it is explained how this is (relatively easily) technically done by building on the MPTP system [Urb07] and the XSL style sheets for Mizar [Urb05]. In Section 4 we summarize the new features and improvements of the IDV tool that are used for this (and which, by IDV's nature, are generally available for any derivation in the TPTP format).

3 Structure of the System

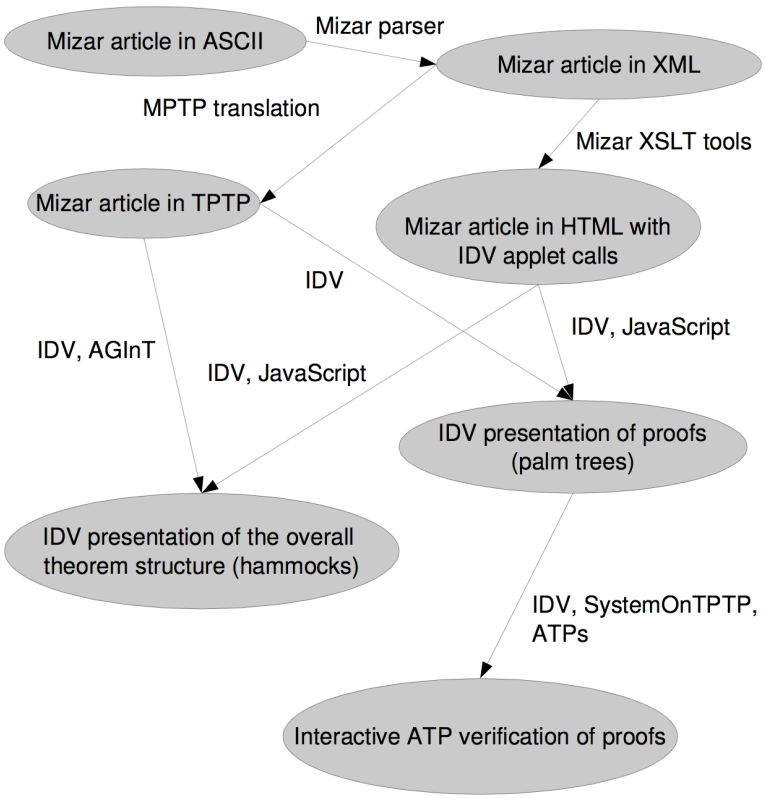

Figure 5: Systems used for the Semantic Presentation

The overall structure of the system is shown in Figure 5. The Mizar parser produces the XML form of a Mizar article, which is then translated by XSLT tools to HTML, and by XSL and the MPTP translator to the TPTP format. A number of additions have recently been made to both the XSL translations. First, the original Mizar identifiers (variables, labels, constants) are kept in the XML, and are thus available for more faithful

HTML presentation. The presentational information is also retained, which allows re-creation of the original logical connectives used in formulae. This is used for both the HTML and TPTP translations. The TPTP translation has been enhanced to contain all the Mizar natural deduction information necessary for recreating the proof structure. A TPTP format extension was implemented for recording proofs that introduce and discharge assumptions, and functions that export the Mizar proofs to this format have been written[2].

The linking to the IDV applet and display of the IDV icons in the Mizar HTML (as in Figure 1) is added if the XSL processing uses the `idv` option. This was a simple extension of the existing XSL style sheets, which have gradually become highly parameterized for producing the Mizar HTML in quite different settings. Together with the `idv` option, the `ajax_proofs` option was used. It puts the Mizar format theorem proofs into separate files, and loads and displays them (via an XMLHttpRequest) when the user clicks on the **proof** keyword. This makes the size of the HTML files much smaller, allowing more eye-candy (colors, titles, etc.), and a faster browsing experience. A new `display_thesis` option has also been implemented, which puts a clickable `thesis` text after each Mizar natural deduction step. This is used to display the implicit thesis (computed by Mizar) after each natural deduction step. It is especially useful for this presentation, because the TPTP counterparts of theses are necessary parts of the corresponding TPTP proofs visualized by IDV.

The TPTP format proofs of the theorems in each article are available under the TSTP icon to the right of each theorem header in the HTML presentation (as in Figure 1). The Mizar-to-TPTP translation is also easy to do in real time, and we hope to make this service available in the near future. Note that these proofs are in a format that is intended to be really verifiable by ATP systems. That means that the necessary background information used implicitly by the Mizar proof checker has been added to the problems as axioms. In advanced domains this can make the axiom set quite large, which is unsuitable for direct IDV display. That's one reason why the IDV "red line" functionality for hiding axioms (see Section 4) was developed, and is used for presenting such problems. An interestingness rating was added to each step in each theorem's TPTP format proof, based on the level of nesting the Mizar proof. The "lightbulb" icon and slider in IDV (see Figure 2) allow the user to interactively set an interestingness threshold for the derivation display, and hide nodes whose interestingness is below the threshold, thus displaying a proof synopsis (see Section 4).

The TPTP format problem corresponding to the Mizar problem, as generated by the MPTP system [Urb07], are available under the TPTP icon to the right of the palm tree of each theorem header in the HTML presentation (as in Figure 1). The TPTP problem is an independent translation of the Mizar problem, which can be attempted by any ATP system. Of course the derivation obtained by an ATP system is unlikely to be the same as the TPTP format proof formed by the translation of the Mizar proof of the theorem.

In the same way that individual theorem proofs are translated, the system is used to produce the overall theorem structure of each Mizar article, in the form of summarized TPTP derivations. In these summarized derivations each theorem is a node of the derivation, and its parents are the axioms, definitions, and theorems from which it was proved in Mizar. These are available under the TSTP icon to the right of each article header in the HTML presentation (as in Figure 1). The goal of this presentation is to provide structural information about dependencies between articles' "main results". The example `jgraph_7` given in Section 1 shows that the visual information about this high-level structure can be very useful (to the authors, reviewers, or just readers of the Mizar library). This information is intended to be purely presentational, and as such the background information

[2]This will be described in more detail in a paper about ATP verification of Mizar.

necessary for "high-level" verification is not added. This would actually be very easy to do, but users probably would not like to try to verify these high-level steps because the success rate (in a reasonable time limit for an ATP system) is obviously much lower than for the simple inferences in the individual theorem's proofs. The AGInT system was used on these overall presentations to add an interestingness rating to each theorem, so that IDV can display a synopsis of the overall structure. That again can produce new insights while viewing the high-level derivation structure.

4 Presenting with IDV

IDV is a tool for graphical rendering of derivations that are written in the TPTP format. A number of additions and improvements have recently been done (since [TPS06]) to provide the functionalities needed for the the presentation of the Mizar library and beyond. A description of the features, many of them new, useful for viewing the Mizar proofs is provided here: the summarization, subderivation extraction, and verification functionalities.

4.1 Summarization

The TPTP format proofs, and in particular the article summaries, are very large, and typically have a very high proportion of axioms. Such large derivations are difficult to display in full detail, for three reasons. First, IDV runs as a Java applet, which limits its speed. Second, it is hard to see a single formulae node when there are a few thousand of them on the screen. Third, when there is a very high proportion of axioms the display is necessarily very wide because of the axioms lined up across the top, which requires zooming out a great deal to see the whole proof, and the nodes become very small. For derivations that are very large, IDV offers two mechanisms to make the derivation easier to view.

The first mechanism is proof synopsis. As explained in Section 3, an "interestingness" value can be associated with each formulae, either in advance by some external criteria, or by the AGInT system. AGInT may be used in advance (as is done for the Mizar articles), or can be called from within IDV by toggling on the "light bulb" ("show IDV synopsis") icon. When the light bulb is on, nodes are resized proportionally to their interestingness. Moving the interestingness slider to the right increases the interestingness threshold, and nodes with lower interestingness are hidden (with edges being extended from their children to their unhidden ancestors). By default leaf nodes are protected from being hidden, but the new "police badge" ("toggle protection of uninteresting axioms") icon can be used to turn off this protection, thus making it possible to hide large numbers of uninteresting axioms. The "artists palette" ("redraw") icon redraws the derivation with only the displayed nodes, to provide a synopsis of (a part of) the derivation. An example synopsis of the first theorem in `xboole_1` is shown in Figure 6.

The second mechanism is unconditional hiding. The new "diver down" ("hide formulae above the red line") icon and slider allow the user to unconditionally hide formulae above a chosen depth from the axioms. This is particularly useful for (and was motivated by) Mizar article summaries that have a very high proportion of axioms. For example, the overall theorem structure of the article `jgraph_7` is slow to display because of the large number of axioms. Using the red line slider to hide the axioms (the top of Figure 7), and then doing a redraw (the bottom of Figure 7), provides a summary of the lower, probably more important, parts of the derivation.

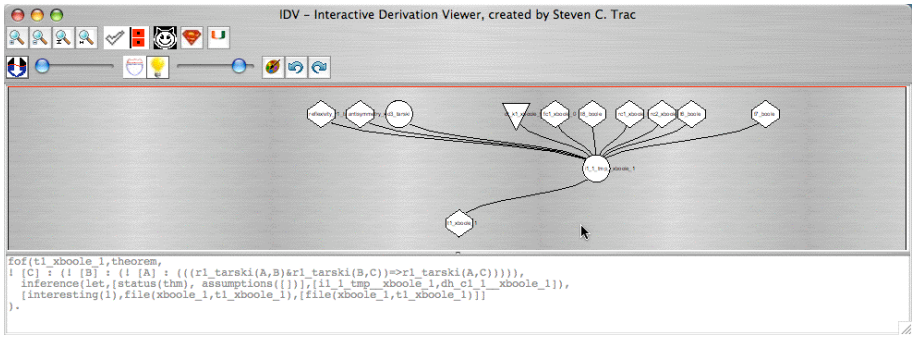

Figure 6: Synopsis of `Th1` in Article `xboole_1`

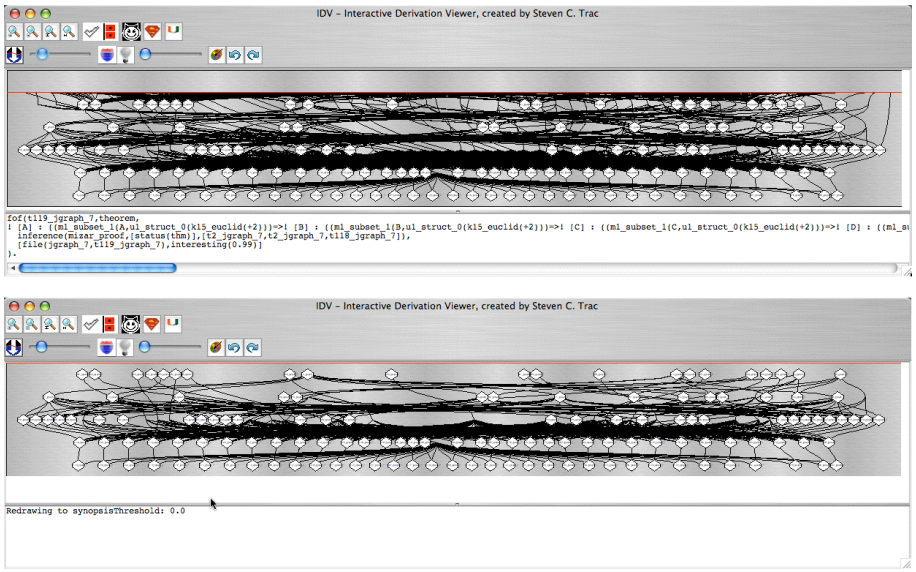

Figure 7: Summary of Article `jgraph_7`

The functionalities described above have been combined to automatically summarize very large derivations that are given to IDV. If there are more than 256 nodes in the derivation, then IDV

1. Adds interestingness (by calling AGInT), unless already supplied by user.
2. Sets the interestingness threshold (i.e., moves the interestingness slider) to try reduce the number of nodes to less than 256.
3. If more than 256 nodes remain unhidden, sets the axiom protection off.
4. If more than 256 nodes still remain unhidden, moves the red line down as many levels as necessary.
5. Does a redraw, so that the hidden formulae do not affect the current drawing. The user can move the sliders back and toggle/untoggle buttons to show hidden formulae later.

This automatic summarization can be seen, e.g., in the display of the article `jgraph_4`.

Together with an optimization of the AGInT system on very large data (thousands of derivation steps are now rated within seconds), these mechanisms have largely sped up and improved the display of the Mizar proofs.

4.2 Subderivation Extraction

There are now more ways to interact with the IDV graph, making it easier to explore the proof in different ways, and render selected extracts.

The first group of extracts are determined by mouse clicks on a chosen node. A left mouse click opens a pop-up window with the text of the annotated formula and its parents. This window also allows verification of that inference, as explained in Section 4.3. Note that the parents shown are of the "what you see is what you get" nature, where hidden ones from non-interestingness and the red line are not shown.

A control left mouse click on a node opens a new IDV window showing the formula and its parents, with the full functionality of any IDV window. A shift-control left mouse click opens a new IDV window showing the formula and all its ancestors. This is useful for extracting a subderivation - Figure 8 shows the subderivation rooted at the node `e5_1__xboole_1` of the first theorem in `xboole_1`. A control right mouse click opens a new IDV window showing the formula and its immediate descendants and their parents. A shift-control right mouse click opens a new IDV window showing the formula and all its descendants and their parents. This is useful for seeing what formulae depend on the clicked one - Figure 9 shows the descendant derivation hung from the node `t7_jgraph_7` of the `jgraph_7` article.

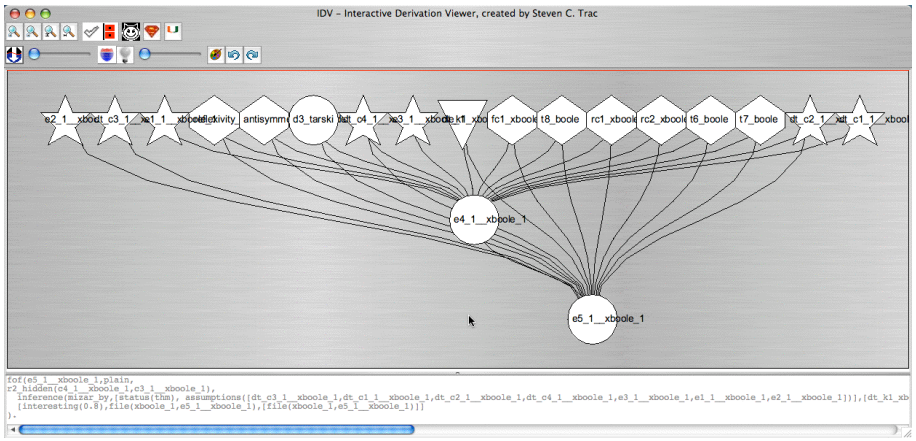

Figure 8: Extract from `Th1` in Article `xboole_1`

The new "good/evil cat" ("show only axioms and lemmas") icon hides nodes that are not axioms or logical consequences of the axioms. In proofs by contradiction, which negate the conjecture, this hides formulae that are derived from the negated conjecture. In proofs that have assumptions, this hides the assumptions and any formulae that have undischarged assumptions. This is useful for identifying formulae that can be used, e.g., as lemmas from the axioms.

4.3 Verification

A derivation displayed by IDV can be verified in three ways. Regardless of how the verification is done, if a node has been verified and the "green tick" ("show verified formulae") icon is on, verified nodes will have a green tick on them (see Figure 3). The verified status is reset if proof summarization changes the apparent parents of the node.

The first way to verify nodes is as illustrated in Section 1. IDV iteratively calls either GDV or a chosen ATP system to verify each node of the derivation. The choice of whether

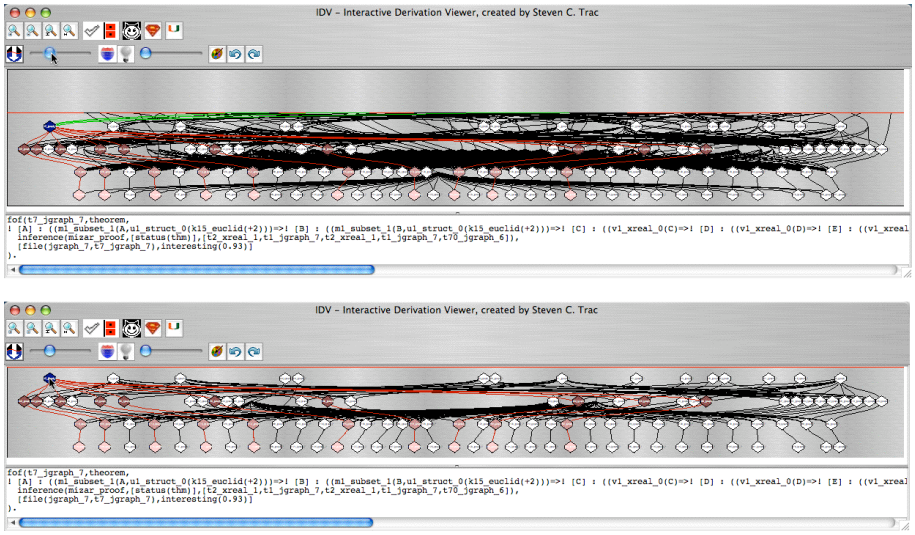

Figure 9: Extract from Article `jgraph_7`

to use GDV or an ATP system is controlled by the "beer" - GDV - and "cocktail" - ATP system - icon in the pop-up window shown in the upper right of Figure 3. If an ATP system is used directly then only steps of logical consequence are verified, by proving them using the ATP system. If GDV is used then all inference steps are verified, proving steps of logical consequence using the ATP system, and using other techniques in other situations. In particular, GDV is able to verify the propagation and discharge of assumptions.

The second way is to verify an individual node from it's pop-up window, produced by clicking on the node (see Section 4.2). Again, there is a choice of using GDV or an ATP system. If an ATP system is used and a TPTP format proof is returned by the ATP system, the "palm tree" ("new IDV window") icon will open a new IDV window displaying the verifying ATP system's proof. Figure 10 show's EP 0.99's verification proof of the final node `t1_xboole_1` of the first theorem in `xboole_1`.

The third way is to use the "superman" ("SystemOnTSTP") icon, which exports the derivation to the SystemOnTSTP interface[3], which in turn provides access to a range of derivation analysis and display tools. The GDV tool is available there for a complete verification of the derivation, including structural checks that are not done from within IDV. This interface is shown in Figure 11.

5 Conclusion

This paper describes a combination of Mizar- and TPTP- based tools used for visualizing and analyzing Mizar proofs. The combination delivers to the readers of the MML an easy, powerful, and almost playful way of exploring the semantics and the structure of the library. The key factors for the relative easiness of having these functionalities are the choice of XML as both internal and external interface of Mizar, and the existence of a TPTP representation of MML articles.

The system integrates so many components that it naturally behaves as a large debugger for the various tools[4]. This has resulted in battle hardening of the tools, and a robust

[3] `http://www.tptp.org/cgi-bin/SystemOnTSTP`

[4] Just a recent example: While randomly inspecting the large number of Mizar derivations in IDV, it has turned out that some cannot be verified, because of a recently introduced incompatibility between the

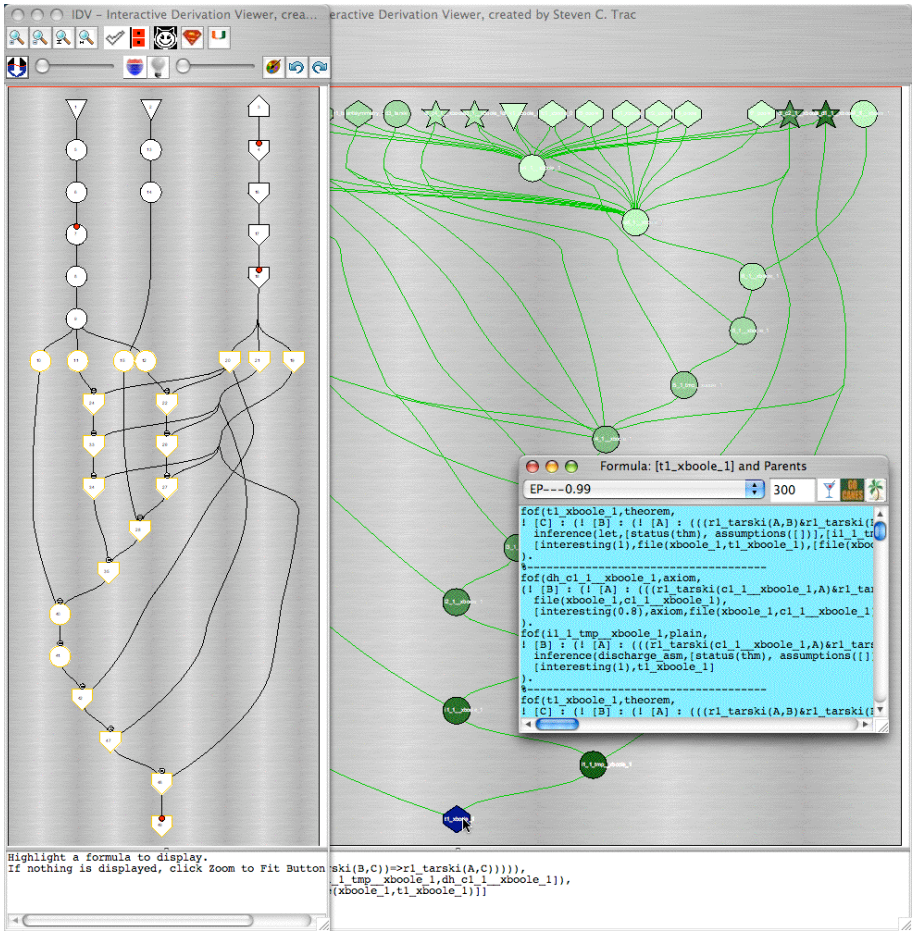

Figure 10: Node Verification from `Th1` in Article `xboole_1`

and reliable interface. The combination shows the great added value that can be obtained by cooperation of several quite diverse (and, quite often, separately developed) projects, provided that they are based on the same communication standards. This places the system alongside other work based around a combination of component reasoning systems, e.g., [ZMSZ04, ZA06, SMMC06].

Although much of work done was motivated by the desire to view the structure of Mizar proofs and articles, all of the work is general and immediately available for any derivations in the TPTP format. As such all the tools are now part of the general SystemOnTSTP interface.

References

[MJWD06] C. Matuszek, Cabral J., M. Witbrock, and J. DeOliveira. An Introduction to the Syntax and Content of Cyc. In Baral C., editor, *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, pages 44–49, 2006.

TPTP Java parser used by IDV, and the TPTP parser used by the E prover.

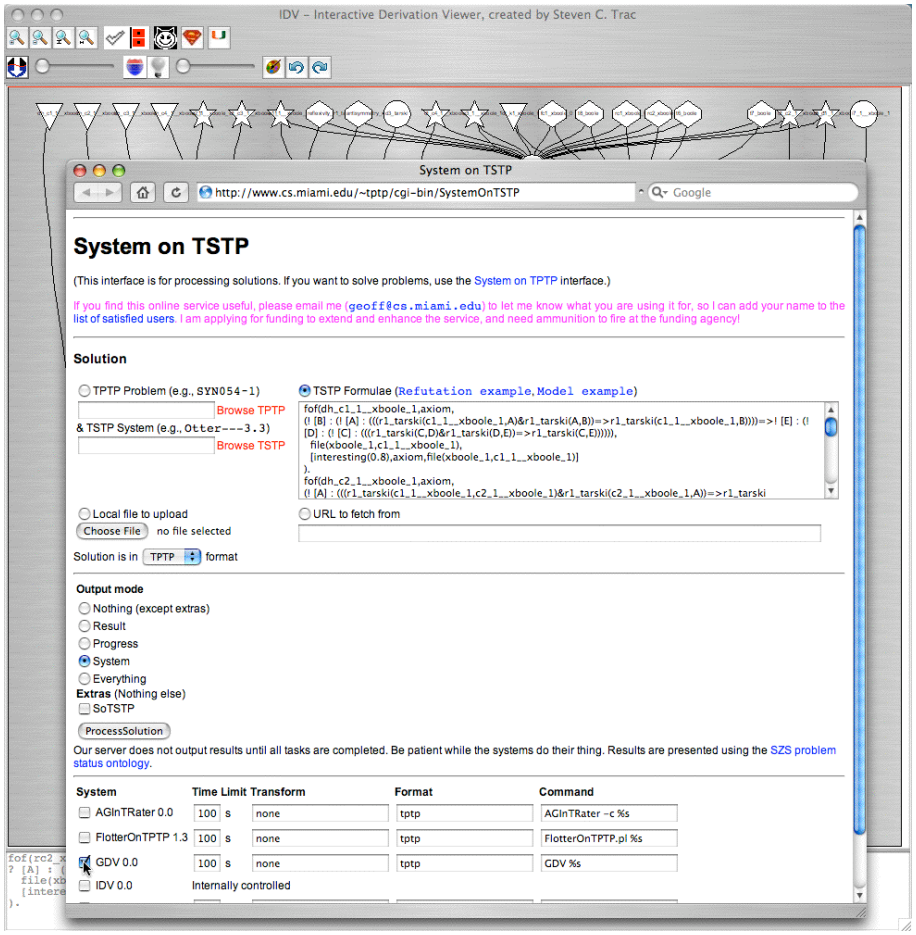

Figure 11: SystemOnTSTP for `Th1` in Article `xboole_1`

[MP06] J. Meng and L. Paulson. Translating Higher-Order Problems to First-Order Clauses. In G. Sutcliffe, R. Schmidt, and S. Schulz, editors, *Proceedings of the FLoC'06 Workshop on Empirically Successful Computerized Reasoning, 3rd International Joint Conference on Automated Reasoning*, volume 192 of *CEUR Workshop Proceedings*, pages 70–80, 2006.

[NP01] I. Niles and A. Pease. Towards A Standard Upper Ontology. In C. Welty and B. Smith, editors, *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*, pages 2–9, 2001.

[NT05] Y. Nakamura and A. Trybulec. The Fashoda Meet Theorem for Rectangles. *Journal of Formalized Mathematics*, 13(2):199–219, 2005.

[PGS06] Y. Puzis, Y. Gao, and G. Sutcliffe. Automated Generation of Interesting Theorems. In G. Sutcliffe and R. Goebel, editors, *Proceedings of the 19th International FLAIRS Conference*, pages 49–54. AAAI Press, 2006.

[Rud92] P. Rudnicki. An Overview of the Mizar Project. In *Proceedings of the 1992 Workshop on Types for Proofs and Programs*, pages 311–332, 1992.

[Sch02] S. Schulz. A Comparison of Different Techniques for Grounding Near-Propositional CNF Formulae. In S. Haller and G. Simmons, editors, *Pro-*

ceedings of the 15th International FLAIRS Conference, pages 72–76. AAAI Press, 2002.

[SMMC06] V. Sorge, A. Meier, R. McCasland, and S. Colton. Automatic Construction and Verification of Isotopy Invariants. In U. Furbach and N. Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning*, number 4130 in Lecture Notes in Artificial Intelligence, pages 36–51. Springer-Verlag, 2006.

[SSCVG06] G. Sutcliffe, S. Schulz, K. Claessen, and A. Van Gelder. Using the TPTP Language for Writing Derivations and Finite Interpretations. In U. Furbach and N. Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning*, number 4130 in Lecture Notes in Artificial Intelligence, pages 67–81, 2006.

[Sut00] G. Sutcliffe. SystemOnTPTP. In D. McAllester, editor, *Proceedings of the 17th International Conference on Automated Deduction*, number 1831 in Lecture Notes in Artificial Intelligence, pages 406–410. Springer-Verlag, 2000.

[Sut06] G. Sutcliffe. Semantic Derivation Verification. *International Journal on Artificial Intelligence Tools*, 15(6):1053–1070, 2006.

[TPS06] S. Trac, Y. Puzis, and G. Sutcliffe. An Interactive Derivation Viewer. In S. Autexier and C. Benzmüller, editors, *Proceedings of the 7th Workshop on Workshop on User Interfaces for Theorem Provers, 3rd International Joint Conference on Automated Reasoning*, volume 174 of *Electronic Notes in Theoretical Computer Science*, pages 109–123, 2006.

[Urb05] J. Urban. XML-izing Mizar: Making Semantic Processing and Presentaion of MML Easy. In M. Kohlhase, editor, *Proceedings of the 4th Integrated Conference on Mathematical Knowledge Management*, volume 3863 of *Lecture Notes in Computer Science*, pages 346–360, 2005.

[Urb07] J. Urban. MPTP 0.2: Design, Implementation, and Initial Experiments. *Journal of Automated Reasoning*, 37(1-2):21–43, 2007.

[ZA06] J. Zimmer and S. Autexier. The MathServe System for Semantic Web Reasoning Services. In U. Furbach and N. Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning*, number 4130 in Lecture Notes in Artificial Intelligence, pages 17–20, 2006.

[ZMSZ04] J. Zimmer, A. Meier, G. Sutcliffe, and Y. Zhang. Integrated Proof Transformation Services. In C. Benzmüller and W. Windsteiger, editors, *Proceedings of the Workshop on Computer-Supported Mathematical Theory Development, 2nd International Joint Conference on Automated Reasoning*, Electronic Notes in Theoretical Computer Science, 2004.