# VizAR: The Proof Navigator *

Jan Jakubův[1] and Cezary Kaliszyk[2]

[1] Czech Technical University in Prague, Prague, Czech Republic
jakubuv@gmail.com
[2] University of Innsbruck, Innsbruck, Austria and INDRC, Prague, Czech Republic
cezary.kaliszyk@uibk.ac.at

We present a system for the visualization of proofs originating from Automated Theorem Provers for first-order logic. The system can hide uninteresting proof parts of proofs, such as type annotations, translate first-order terms to standard math syntax, and compactly display complex formulas. We demonstrate the system on several non-trivial automated proofs of statements from Mizar Mathematical Library [1, 2, 5] translated to first-order logic, and we provide a web interface where curious users can browse and investigate the proofs.

With the increasing power of *Automated Theorem Proving* systems (ATPs), the size and complexity of the proofs they output are also increasing. This additionally implies that analyzing such automatically generated proofs is becoming more daunting for users. This is of particular importance for proofs that originate from machine-learning-guided provers. The guided version of E, ENIGMA [7] can automatically find proofs of many theorems that have previously been provable only with long manual proofs. A large number of such proofs have been discussed in our recent work on machine learning for Mizar [6]. To allow users to inspect and analyze such proofs conveniently, we developed and present the VizAR system:

http://ai.ciirc.cvut.cz/vizar/

The system can hide uninteresting parts of proofs (such as Mizar soft type system annotations and reasoning about them), translate first-order terms to standard math syntax (such as presenting $\mathsf{Element}(x, y)$ as $x \in y$), and compactly display complex formulas. The system provides several ways to visualize complex proofs. In the full proof view, the proof is displayed as an interactive SVG image. In order to simplify orientation in large proofs, the system features a conjecture-centered view which helps to identify essential proof steps. Finally, the proof step view allows the user to interactively browse individual proof steps and reveal the proof essence hidden in their symbols.

VizAR can display an arbitrary proof in the TPTP language. In addition, it integrates extended support for proofs of Mizar statements coming from the MPTP [10] translation of Mizar to first-order logic. A large amount of MPTP proofs has been recently generated by ATPs (E [9] and Vampire [8]) with machine learning guidance [6]. Selected proofs can be investigated on the VizAR web page. VizAR shows the original Mizar statements for every conjecture and assumption, and it provides links to Mizar proofs and symbol definitions.

**Symbol Translation.** VizAR uses Unicode symbols to display terms and predicates in standard mathematical notation when possible. For example, the MPTP symbol `m1_subset_1(X,Y)` corresponds to the Mizar symbol $\mathsf{Element}(X, Y)$ and in VizAR it is presented as $X \in Y$. Another example is the MPTP symbol `r2_wellord2(X,Y)` corresponding to the Mizar symbol $\mathsf{are\_equipotent}(X, Y)$ which is written as $|X| = |Y|$ in VizAR.

**Clause Visualization.** In the VizAR syntax, clauses are displayed as sequents, for example, the clause $A \mid B \mid \sim C \mid \sim D$ is considered as the logically equivalent sequent $C, D \Rightarrow A, B$. The

*Supported by ERC-CZ grant no. LL1902 POSTMAN and EU Regional Development Fund under the Czech project AI&Reasoning no. CZ.02.1.01/0.0/0.0/15_003/00004.

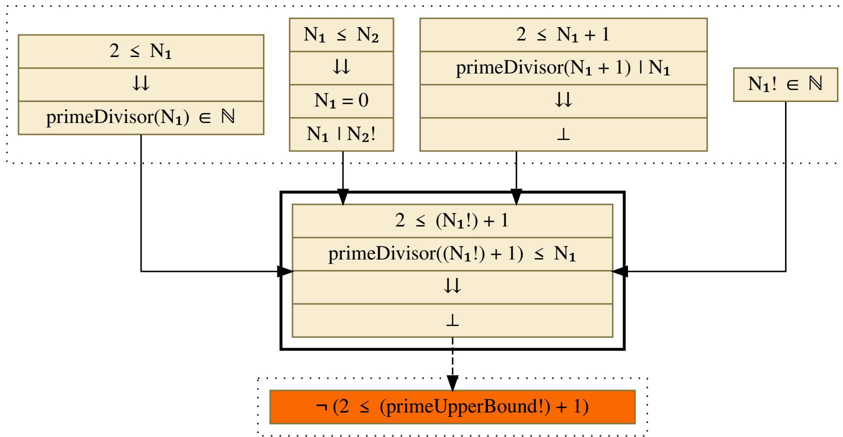


Figure 1: Visualization of a proof step from the proof of MPTP theorem `t72_newton`.

sequents are visualized as boxes with the content displayed vertically (top-down) as demonstrated in Figure 1. Clauses without negative literals (for example, $A \mid B$) are displayed simply as $A, B$ instead of $\top \Rightarrow A, B$. Clauses without positive literals (for example, $\sim C \mid \sim D$) are displayed as $C, D \Rightarrow \bot$.

**Clause Simplifications.** MPTP first-order translations of Mizar statements typically use soft type guard predicates to specify types of variables. VizAR hides the type guards applied to variables, and introduces a different variable symbol for each type predicate, for example, $N$ for natural numbers and $R$ for real numbers. As a second step, all negative occurrences of type guard predicates (even with a non-variable argument) are completely hidden. This is because they typically provide no interesting information from the human point of view.

**Proof Transformations.** After symbol translations and clause simplifications, two consequent proof graph nodes might represent syntactically equal clauses. For example, the Mizar statements $\mathsf{Element}(X, \mathsf{NAT})$ and $\mathsf{natural}(X)$ are both represented as $X \in \mathbb{N}$ in VizAR. In these cases, to further simplify the proof graph, we unify consequent nodes labeled with the same VizAR expression and merge their respective source and destination edges.

**Proof Visualizations.** VizAR uses Graphviz [3, 4] to render proof graphs while the web interface is implemented by the static site generator Jekyll. VizAR web interface provides several ways to investigate ATP proofs. In the *full proof view*, the whole proof graph is displayed as an SVG image with hyperlinks. Graph leaves corresponding to assumptions are displayed in blue and all the nodes inferred from the negated conjecture are displayed in orange. Since the full proof view might be very complex, VizAR features a *conjecture-centered view* where only the statements derived from the conjecture are displayed. This view can help the user inspect how the negated conjecture leads to a contradiction. In the *proof step view*, only a single proof graph node is displayed with its immediate parents and children. Fig. 1 shows an example proof step in VizAR. The ATP proved the Mizar theorem `t72_newton`, which states that there is no upper bound on the prime numbers.

We have developed the VizAR ATP proof visualization system and we publish its web interface on GitHub pages with a custom domain redirect. The web interface currently features selected ATP proofs of MPTP statements. In the proof gallery, we present *featured proofs* with improved VizAR syntax for all relevant Mizar symbols. Moreover, the *other proofs* section of the page contains a large number of proofs where Mizar names are used for selected symbols.

## References


[1] Grzegorz Bancerek, Czesław Byliński, Adam Grabowski, Artur Korniłowicz, Roman Matuszewski, Adam Naumowicz, and Karol Pąk. The role of the Mizar Mathematical Library for interactive proof development in Mizar. *J. Autom. Reason.*, 61(1-4):9–32, 2018.

[2] Grzegorz Bancerek, Czesław Byliński, Adam Grabowski, Artur Korniłowicz, Roman Matuszewski, Adam Naumowicz, Karol Pąk, and Josef Urban. Mizar: State-of-the-art and beyond. In Manfred Kerber, Jacques Carette, Cezary Kaliszyk, Florian Rabe, and Volker Sorge, editors, *Intelligent Computer Mathematics - International Conference, CICM 2015*, volume 9150 of *LNCS*, pages 261–279. Springer, 2015.

[3] John Ellson, Emden R. Gansner, Eleftherios Koutsofios, Stephen C. North, and Gordon Woodhull. Graphviz - open source graph drawing tools. In *GD*, volume 2265 of *LNCS*, pages 483–484. Springer, 2001.

[4] John Ellson, Emden R. Gansner, Eleftherios Koutsofios, Stephen C. North, and Gordon Woodhull. Graphviz and dynagraph - static and dynamic graph drawing tools. In *Graph Drawing Software*, pages 127–148. Springer, 2004.

[5] Adam Grabowski, Artur Korniłowicz, and Adam Naumowicz. Mizar in a nutshell. *J. Formalized Reasoning*, 3(2):153–245, 2010.

[6] Jan Jakubův, Karel Chvalovský, Zarathustra Amadeus Goertzel, Cezary Kaliszyk, Mirek Olšák, Bartosz Piotrowski, Stephan Schulz, Martin Suda, and Josef Urban. MizAR 60 for Mizar 50. *CoRR*, 2023. Accepted for ITP 2023.

[7] Jan Jakubův, Karel Chvalovský, Miroslav Olšák, Bartosz Piotrowski, Martin Suda, and Josef Urban. ENIGMA Anonymous: Symbol-independent inference guiding machine (system description). In *IJCAR (2)*, volume 12167 of *LNCS*, pages 448–463. Springer, 2020.

[8] Laura Kovács and Andrei Voronkov. First-order theorem proving and Vampire. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification - 25th International Conference, CAV 2013*, volume 8044 of *LNCS*, pages 1–35. Springer, 2013.

[9] Stephan Schulz. E – A Brainiac Theorem Prover. *AI Commun.*, 15(2-3):111–126, 2002.

[10] Josef Urban. MPTP 0.2: Design, implementation, and initial experiments. *J. Autom. Reason.*, 37(1-2):21–43, 2006.