An Overview of Methods for Large-Theory Automated Theorem Proving

(Invited Paper)

Josef Urban
Radboud University Nijmegen
josef.urban@gmail.com

Abstract

This is an attempt at a brief initial overview of the state of the art in the young field of first-order automated reasoning in large theories (ARLT). It is necessarily biased by the author's imperfect knowledge, and hopefully will serve as material provoking further corrections and completions.

1 Why Large Theories?

Why should we want to (automatically) reason in large theories and develop them instead of small theories? Here are several answers:

- Mathematicians work in large theories. They know a lot of concepts, facts, examples and counter-examples, proofs, heuristics, and theory-development methods.
- Other scientists (and humans in general) work with large theories. Consider physics, chemistry, biology, law, politics, large software libraries, Wikipedia, etc. Our current knowledge about the world is large.
- In the last years, more and more knowledge is becoming available formally by all kinds of human efforts (interactive theorem proving, common-sense reasoning, knowledge bases for various sciences, Semantic Web, etc.). This is an opportunity for automated reasoning to help with the sciences and tasks mentioned above.
- Existing resolution/superposition automated reasoning systems often derive large numbers of facts, even from small initial number of premises. Managing such large numbers can profit from specialized large-theory techniques.

Automated reasoning in large theories is today often about increasing the comfort of users of automated reasoning methods: It is typically possible to *manually* select premises from which some conjecture should follow. Often this is even a significant part of one's formal reasoning wisdom. But ultimately, *manual* is the opposite of *automated.*

1.1 Large Formal Theories Are Not Our Enemy

However, *automated selection of relevant facts* is only the very first step that recently made existing ATP methods usable and useful in large theories. This premise-selection view treats large theories to a large extent only as an ATP person's *enemy*: We need to select the few right facts from the large pile of less relevant facts before we get down to the "real science" of "doing ATP".

This is in the author's opinion a very limited view of the large-theory field. The bigger reason for making large complex theories and knowledge bases available to the automated reasoning world is that they can contain a large amount of domain-specific problem-solving knowledge, and likely (in less explicit form), also a large amount of general problem-solving knowledge that the automated reasoning field should reveal and integrate into its pool of methods.

For this, however, another limited view needs to be overcome: Large theories (and theories in general) are not just random collections of usable facts. Mathematical theories in particular have been developed by smart people over centuries, and quite likely such theories are the best, deeply computer-understandable corpus of *abstract human thinking* that we currently have. It seems negligent to ignore the internal theory structure, and the problem-solving and theory-engineering knowledge developed by mathematicians so far. Especially when we know that first-order ATP is an undecidable problem, and that the current ATP methods are on average far behind what trained mathematicians can do.

Thus, large complex formal theories and knowledge bases are not an enemy, but an opportunity. Not just an opportunity to reason with the knowledge of many already established facts, but also an opportunity to analyze and learn how smart people reason and prove difficult theorems, develop their conceptual space, and how they find surprising connections and solutions. In short, large formal theories are a great new playground for developing general AI. But because general AI (and theorem-proving oriented AI in particular) has been in the second half of the 20th century labeled as unproductive, general AI research in this field should go hand-in-hand with practical applications and usability testing. So far, this has fortunately often been the case in this young field.

2 Corpora

Several large formal knowledge bases have become recently available to experiments with first-order automated reasoning tools. To name the major ones (in alphabetic order):

- The CYC (OpenCyc, ResearchCyc) common-sense knowledge base [16]
- The Isabelle/HOL mathematical library [10]
- The Mizar/MML mathematical library [25]
- The SUMO (and related ontologies) common-sense knowledge base [13]

It is likely that more will follow (or already are available). For example, the HOL Light/Flyspeck [4, 5] large mathematical library should benefit from similar first-order translation techniques as the Isabelle/HOL library. More common-sense knowledge bases like YAGO [21] might be produced by semi-automated methods, and bridges to all kinds of specialized scientific databases are being build, spearheaded by systems like Biodeducta [19]. The LogAnswer project [3] has already started to reason over the first-order export of the *full texts* of German Wikipedia.

The corpora differ in their purpose/origin, size, complexity, consistency, completeness, and the extent to which they cover various large-theory aspects. The common-sense ontologies contain a lot of classification/hierarchical knowledge, resulting typically in simple Horn clauses, and also a lot of concept definitions with relatively few facts proved about them. Storing and maintaining proofs has so far been a secondary aspect. Their primary emphasis was not (so far) on building up libraries of more and more advanced proved theorems about the world, but rather on covering as many concepts as possible by suitable definitions.

On the other hand, the mathematical theories have a much larger number of nontrivial mathematical theorems in them, and their formal content typically follows some established informal theory developments based on well-known and fixed mathematical foundations. There is more concept/fact re-use in mathematics, and nontrivial proofs of many facts exist and (at least in theory) can be made available in common formats and for large-theory techniques based on inspection of previous proofs and theory developments.

3 Automated Methods for Reasoning in Large Theories

The existing large-theory reasoning methods can be divided into several groups, using various criteria. One criterion is the method used for knowledge selection. The methods developed so far include syntactic heuristics, heuristics using semantic information, methods that look at previous solutions, and combinations thereof. Systems and methods that make use mainly of syntactic criteria for premise selection include:

- The SInE (SUMO Inference Engine) algorithm by Kryštof Hoder [6], and its E implementation by Stephan Schulz.[1] The basic idea is to use global frequencies of symbols to define their global *generality*, and build a relation linking each symbol S with all formulas F in which S is has the lowest global generality among the symbols of F. In common-sense ontologies, such formulas typically *define* the symbols linked to them, which is the reason for calling this relation a *D-relation*. Premise selection for a conjecture is then done by recursively following the D-relation, starting with the conjecture's symbols. Various parameters can be used, e.g., limiting the recursion depth significantly helps for the Mizar library [26], and preliminary experiments show that also for the Isabelle/HOL library.

- The default premise selection heuristic used by the Isabelle/Sledgehammer export [11] seems to be quite similar to SInE, however it works internally in Isabelle, and uses additional mechanisms like blacklisting. D-relation is not used there, the formulas are linked to all symbols they contain.

- The *Conjecture Symbol Weight* clause selection heuristics in E prover [18] give lower weights to symbols contained in the conjecture, thus preferring during the inference steps the clauses that have common symbols with the conjecture. This is remotely similar to general *goal-oriented* ATP techniques, as for example the Set of Support (SoS) strategy in resolution/superposition provers,[2]. Note that also the majority of tableau calculi are in practice goal-oriented, and the leanCoP [12] prover in particular performs surprisingly well on the MPTP Challenge large-theory benchmark.

A method which is purely signature-based, however the word *semantics* appears in it, is *latent semantics*. Latent semantics is a machine learning method that has been successfully used for example in the Netflix Challenge, and in web search. Its principle is to automatically derive "semantic" equivalence classes of words (like *car, vehicle, automobile*) from their co-occurrences in documents, and to use such equivalence classes (also called *synsets* in the WordNet ontology) instead of the original words for searching and related tasks. This technique has been so far used in:

- Paul Cairns' Alcor system [1] for searching and advice over the Mizar library.

- Yuri Puzis' initial relevance ordering of premises used in the SRASS ATP metasystem [22].

[1] `http://www.mpi-inf.mpg.de/departments/rg1/conferences/deduction10/slides/stephan-schulz.pdf`

[2] In particular, SPASS [30] has been used successfully on the Isabelle data.

Semantics (in the original logical sense) has been used for a relatively long time in various ways for guiding the ATP inference processes. An older system that is worth mentioning with respect to the current efforts is John Slaney's SCOTT system [20] constraining Otter inferences by validity in models. A similar idea has been recently revived by Jiří Vyskočil at the Prague ATP seminar: His observation was that mathematicians have very fast conjecture-rejection methods based on a (relatively small) pool of (often imprecise) models in their heads, similar to some fast heuristic software testing methods. This motivated Petr Pudlák's semantic axiom selection system for large theories [15], implemented later also by Geoff Sutcliffe in SRASS. The basic idea is to use finite model finders like MACE [9] and Paradox [2] to find counter-models of the conjecture, and gradually select axioms that exclude such counter-models. The models can differentiate between a formula and its negation, which is typically beyond the heuristic symbolic means. This idea has been also used later in the MaLARea system [27], however in the context of many problems solved simultaneously and many models kept in the pool, and using the models found also as classification features for machine learning.

MaLARea is also an example of a system that uses learning from previous proofs for guiding premise-selection for new conjectures. The idea of this approach is to define suitable features characterizing conjectures (symbolic, semantic, structural, etc.), and to use machine learning methods on available proofs to learn the function that associates the conjecture features with the relevant premises. A sophisticated learning approach has been suggested and implemented in E prover by Stephan Schulz for his PhD work [17], which unfortunately preceded the appearance of large theories by several years.[3] In this approach, proofs are abstracted into proof traces, consisting of clause patterns in which symbol names are abstracted into higher-order variables. Such proof traces from many proofs are collected into a common knowledge base, which is loaded when a new problem is solved, and used for guiding clause selection. This is probably quite similar to the *hints* technique in Prover9 [8], which however seems to be used more in a single-problem proof-shortening scenario.

Note that such techniques already move the large-theory techniques towards smart general-purpose ATP techniques for proof guidance. A recent attempt in this direction is the MaLeCoP system [28]. There, the clause relevance is learned from all closed tableau branches, and the tableau extension steps are guided by a trained machine learner that takes as input features a suitable encoding of the literals on the current tableau branch. In some sense this tries to transfer the promising premise selection techniques deeper into the core of ATP systems. Unlike the above mentioned technique used in E prover, the advising is however left to external systems, which communicate with the prover over a sufficiently fast link.

4 More Systems and Metasystems

Not all systems do premise selection, however they may be still worth of mentioning.

One way how to reason with full large theories is to significantly limit the reasoning power. At the extreme, such methods become the many search methods available for the corpora mentioned above. A somewhat more involved memorization/reasoning technique is *subsumption* implemented in various ATP systems. A type-aware extension of subsumption is implemented for the Mizar library in the MoMM system [24]. Extending such limited systems further in a controlled and restricted way might be quite rewarding.

[3] The author and Stephan Schulz have shortly tried to revive this old E code and test it on the MPTP Challenge benchmark in 2007, however without any significant results. So this advanced code is still waiting to be properly revived and tested.

Another interesting large-theory techniques is lemmatization and concept creation. An example lemmatization system has been implemented by Petr Pudlák in his PhD thesis [14]: The system uses lemmas found in successful proofs to enrich the whole theory, find new proofs, and shorten existing ones. Concept creation is a long-time AI research, going back to Lenat's seminal work on AM [7]. Recently, concept creation has been tried to shorten long, automatically produced proofs in [29]. Refactoring of proofs into human-digestible form seems to be a very interesting task that we are facing more and more as the automated methods are getting more and more usable. As computers are getting better in solving hard and large problems, we should also make them better in explaining their solutions to us.

References

[1] P. A. Cairns. Informalising formal mathematics: Searching the Mizar library with latent semantics. In A. Asperti, G. Bancerek, and A. Trybulec, editors, *MKM*, volume 3119 of *Lecture Notes in Computer Science*, pages 58–72. Springer, 2004.

[2] K. Claessen and N. Sorensson. New Techniques that Improve MACE-style Finite Model Finding. In P. Baumgartner and C. Fermueller, editors, *Proceedings of the CADE-19 Workshop: Model Computation - Principles, Algorithms, Applications*, 2003.

[3] U. Furbach, I. Glöckner, and B. Pelzer. An application of automated reasoning in natural language question answering. *AI Commun.*, 23(2-3):241–265, 2010.

[4] T. C. Hales. Introduction to the flyspeck project. In Thierry Coquand, Henri Lombardi, and Marie-Françoise Roy, editors, *Mathematics, Algorithms, Proofs*, volume 05021 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2005.

[5] T. C. Hales, J. Harrison, S. McLaughlin, T. Nipkow, S. Obua, and R. Zumkeller. A revision of the proof of the kepler conjecture. *Discrete & Computational Geometry*, 44(1):1–34, 2010.

[6] K. Hoder and A. Voronkov. Sine qua non for large theory reasoning. In *CADE 11*, 2011. To appear.

[7] D. Lenat. *An Artificial Intelligence Approach to Discovery in Mathematics.* PhD thesis, Stanford University, Stanford, USA, 1976.

[8] W.W. McCune. Prover9. http://www.mcs.anl.gov/ mccune/prover9/.

[9] W.W. McCune. Mace4 Reference Manual and Guide. Technical Report ANL/MCS-TM-264, Argonne National Laboratory, Argonne, USA, 2003.

[10] J. Meng and L. C. Paulson. Translating higher-order clauses to first-order clauses. *J. Automated Reasoning*, 40(1):35–60, 2008.

[11] J. Meng and L. C. Paulson. Lightweight relevance filtering for machine-generated resolution problems. *J. Applied Logic*, 7(1):41–57, 2009.

[12] J. Otten and W. Bibel. leanCoP: Lean Connection-Based Theorem Proving. *Journal of Symbolic Computation*, 36(1-2):139–161, 2003.

[13] A. Pease and G. Sutcliffe. First order reasoning on a large ontology. In G. Sutcliffe et al. [23].

[14] P. Pudlak. Search for Faster and Shorter Proofs using Machine Generated lemmas. 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 34–52, 2006.

[15] P. Pudlak. Semantic selection of premisses for automated theorem proving. In G. Sutcliffe et al. [23].

[16] D. Ramachandran, Reagan P., and K. Goolsbey. First-orderized ResearchCyc: Expressiveness and Efficiency in a Common Sense Knowledge Base. In P. Shvaik , editor, *Proceedings of the Workshop on Contexts and Ontologies: Theory, Practice and Applications*, 2005.

[17] S. Schulz. *Learning Search Control Knowledge for Equational Deduction.* PhD thesis, Technische Universität München, Munich, Germany, 2000.

[18] S. Schulz. E: A Brainiac Theorem Prover. *AI Communications*, 15(2-3):111–126, 2002.

[19] J. Shrager, R. Waldinger, M. Stickel, and J. P. Massar. Deductive biocomputing. *PLoS ONE*, 2(4):e339, Apr 2007.

[20] J. K. Slaney, E. Lusk, and W. W. McCune. SCOTT: Semantically Constrained Otter, System Description. In A. Bundy, editor, *Proceedings of the 12th International Conference on Automated Deduction*, number 814 in Lecture Notes in Artificial Intelligence, pages 764–768. Springer-Verlag, 1994.

[21] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: A large ontology from Wikipedia and WordNet. *J. Web Semantics*, 6(3):203–217, 2008.

[22] G. Sutcliffe and Y. Puzis. SRASS — A semantic relevance axiom selection system. In F. Pfenning, editor, *CADE*, volume 4603 of *Lecture Notes in Computer Science*, pages 295–310. Springer, 2007.

[23] G. Sutcliffe, J. Urban, and S. Schulz, editors. *Proceedings of the CADE-21 Workshop on Empirically Successful Automated Reasoning in Large Theories*, volume 257 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[24] J. Urban. MoMM - fast interreduction and retrieval in large libraries of formalized mathematics. *International Journal on Artificial Intelligence Tools*, 15(1):109–130, 2006.

[25] J. Urban. Mptp 0.2: Design, implementation, and initial experiments. *J. Automated Reasoning*, 37(1-2):21–43, 2006.

[26] J. Urban, K. Hoder, and A. Voronkov. Evaluation of automated theorem proving on the Mizar mathematical library. In K. Fukuda, J. van der Hoeven, M. Joswig, and N. Takayama, editors, *ICMS*, volume 6327 of *Lecture Notes in Computer Science*, pages 155–166. Springer, 2010.

[27] J. Urban, G. Sutcliffe, P. Pudlák, and J. Vyskocil. MaLARea SG1–machine learner for automated reasoning with semantic guidance. In A. Armando, P. Baumgartner, and G. Dowek, editors, *IJCAR*, volume 5195 of *Lecture Notes in Computer Science*, pages 441–456. Springer, 2008.

[28] J. Urban, Jirí Vyskocil, and Petr Stepánek. MaLeCoP: Machine learning connection prover. In K. Brünnler and G. Metcalfe, editors, *TABLEAUX*, volume 6793 of *Lecture Notes in Computer Science*, pages 263–277. Springer, 2011.

[29] J. Vyskocil, D. Stanovský, and J. Urban. Automated proof compression by invention of new definitions. In E. M. Clarke and A. Voronkov, editors, *LPAR (Dakar)*, volume 6355 of *Lecture Notes in Computer Science*, pages 447–462. Springer, 2010.

[30] C. Weidenbach, D. Dimova, A. Fietzke, R. Kumar, M. Suda, and P. Wischnewski. Spass version 3.5. In R. A. Schmidt, editor, *CADE*, volume 5663 of *Lecture Notes in Computer Science*, pages 140–145. Springer, 2009.