



Guiding SMT solvers with Monte Carlo Tree Search and neural networks

Stéphane Graham-Lengrand, Michael Färber

► To cite this version:

Stéphane Graham-Lengrand, Michael Färber. Guiding SMT solvers with Monte Carlo Tree Search and neural networks. Third Conference on Artificial Intelligence and Theorem Proving (AITP'2018), Mar 2018, Aussois, France. hal-01935578

HAL Id: hal-01935578

<https://hal.archives-ouvertes.fr/hal-01935578>

Submitted on 26 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Guiding SMT Solvers with Monte Carlo Tree Search and Neural Networks

Stéphane Graham-Lengrand¹, Michael Färber²

¹ CNRS - École Polytechnique, 91120 Palaiseau, France
`graham-lengrand@lix.polytechnique.fr`

² Universität Innsbruck, 6020 Innsbruck, Austria
`michael.farber@gedenkt.at`

Abstract

Monte Carlo Tree Search (MCTS) is a technique to guide search in a large decision space by taking random samples and evaluating their outcome. Frequently, MCTS is employed together with reward heuristics learnt by neural networks. The talk will propose a guidance mechanism for SMT solvers based on a combination of MCTS and neural networks.

Machine learning methods gain importance in automated reasoning. A particularly strong trend are neural networks, having produced state-of-the-art results for premise selection [WTWD17, ISA⁺16]. Outside of automated reasoning, neural networks have been combined with Monte Carlo Tree Search, treating problems as diverse as finding good strategies to play the game of Go [SHM⁺16] and planning of chemical syntheses [SKTW17, SPW17]. In automated reasoning, Monte Carlo Tree Search (MCTS) has been applied to first-order automated theorem proving, using hand-crafted heuristics instead of neural networks [FKU17]. We propose a combination of Monte Carlo Tree Search and neural networks to guide the search performed by an SMT solver.

We are exploring the idea of such guidance in the *Psyche* platform [GL13], which offers a modular architecture for theorem proving. It implements an adaptation, to automated reasoning in general and to SMT solving in particular, of the LCF architecture [Mil79, GMW79].

LCF is mostly used in Interactive Theorem Proving and is particularly widely implemented in the proof assistants of the HOL family, such as the HOL system [HOL], Isabelle [Isa], etc. The LCF architecture allows theorem proving strategies to be programmable, while guaranteeing the correctness of any claim that a formula is provable. The architecture’s *kernel* component offers an API whose primitives implement basic reasoning inferences, and strategies can be programmed on top of the kernel via the API. The claims of provability are then necessarily *correct-by-construction*, assuming the correctness of the kernel, but regardless of any potential defects in the design or in the implementation of strategies (or in the user’s input, for the case of Interactive Theorem Proving).

Psyche embraces this paradigm and, in the case of SMT solving, relates to a position paper by de Moura and Passmore, entitled “The strategy challenge in SMT solving” [dMP13], which promoted the programmability of strategies as compositions of basic reasoning tasks, explicitly referring to the LCF paradigm. This approach opens up the possibilities of extensively experimenting with various strategies, whether they be handcrafted or machine-learned, while never jeopardising the correctness of the solver’s output. Guiding the search by techniques such as MCTS and neural networks can be envisaged more easily in provers whose architecture implements this approach. *Psyche*’s architecture does so at a rather fine-grained level, separating the code that implements reasoning inferences from the code that implements search strategies. More precisely, the CDSAT branch of *Psyche* [CDS] implements the *Conflict-Driven Satisfiability*

framework [BGS17, BGLS18], which lifts from Boolean logic to generic theory combination the conflict-driven clause learning (CDCL) algorithm used in pure SAT-solving:

Given an input SMT problem, the search space explored by *Psyche*/CDSAT consists of *states* that describe specifications for a desired model of the input problem. The *moves* or *actions* that can be made from such a state consist of assigning a value to a term or a literal, thereby specifying the model further, until the existence or non-existence of a model satisfying those specifications is manifest. In the former case, the input problem is concluded to be SAT. The latter case represents a *conflict*, which is analysed so that a lemma can be learnt explaining the reason for the conflict. Some of the assignments are reverted so that another area of the search space can be explored, taking into account the learnt lemmas. If and when these lemmas conclude that no model will be found in the entire search space, the problem is concluded to be UNSAT. *Psyche*/CDSAT is modular in the collection of agents that contribute background knowledge about different theories such as propositional logic and linear arithmetic. These agents offer for each state a range of possible moves, e.g. assigning a truth value to a literal or a rational value to a rational variable, and apply theory-specific inference rules in order to compute consequences of such assignments and detect conflicts.

We propose to apply MCTS guidance for applying moves, which requires a transition probability heuristic for the moves available from a state, and a reward heuristic for states. The former quickly orients the search towards the next states to look at, while the latter, possibly more costly to compute but called less often, contributes to maintaining and updating reward scores for states. These scores are then used to determine whether the search should explore more deeply an area of the search space or whether it should jump to another area. A specificity of satisfiability solving, when expressed as a tree-search problem, is that there are two kinds of conclusions, namely SAT and UNSAT, which may impact what the MCTS heuristics try to achieve, particularly with respect to the exploitation/exploration balance of an MCTS search.

We are investigating the use, for transition probabilities, of existing theory-agnostic heuristics for choosing assignments, usually based on the *activity score* of terms and literals. Those that have often participated to recent conflicts have a high activity [MMZ⁺01] and will be picked with higher probability. This encourages exploitation, triggering the use of recently used lemmas and possibly combining them into a proof of UNSAT.

We propose on the other hand to use, for the reward heuristic, an estimation of proximity between the state to be evaluated and a SAT state / model. This estimation lends itself to being learnt by a neural network, trained on previously completed runs. We propose to trigger this evaluation for conflict states, which comprise a trail of assignments, the lemma it generates, and previous lemmas present at the time of conflict. All of these need to be embedded to a feature vector that is tractable by a neural network, using similar methods as [WTWD17] or [JU17]. Training data for the neural network can be generated by feeding actual SAT states to it, labelling them with maximal reward, as well as feeding it conflict states, labelling them e.g. with the Hamming distance between the conflict state and the actual SAT state.

One of the reasons why we believe that *Psyche* lends itself to this approach is that the basic inferences and the search space are well-identified. Moreover, the prover’s states are persistent data-structures, inherited from the functional programming nature of the LCF approach, which should simplify the recording of the states’ rewards and allow quick state switches during exploration.

At the moment, two components of the proposed approach have been integrated to *Psyche*/CDSAT, which is written in OCaml. First, the OCaml code for MCTS, which was originally developed for connection tableaux [FKU17], but which is sufficiently modular to be applicable to other tree search problems. Second, the OCaml bindings for TensorFlow, which can train

and apply a neural net directly in *Psyche*. What is left to do before evaluating the approach with benchmarks is to encode the feature extraction and organise the training on a suitable set of examples.

References

- [BGLS18] Maria Paola Bonacina, Stéphane Graham-Lengrand, and Natarajan Shankar. Proofs in conflict-driven theory combination. In June Andronick and Amy Felty, editors, *Proc. of the 7th Int. Conf. on Certified Programs and Proofs (CPP'18)*. ACM Press, January 2018.
- [BGS17] Maria Paola Bonacina, Stéphane Graham-Lengrand, and Natarajan Shankar. Satisfiability modulo theories and assignments. In de Moura [dM17], pages 42–59.
- [CDS] The CDSAT system. Available at <https://github.com/disteph/cdsat>.
- [dM17] Leonardo de Moura, editor. *CADE-26*, volume 10395 of *LNCS*. Springer, 2017.
- [dMP13] Leonardo Mendonça de Moura and Grant Olney Passmore. The strategy challenge in SMT solving. In Maria Paola Bonacina and Mark E. Stickel, editors, *Automated Reasoning and Mathematics - Essays in Memory of William W. McCune*, volume 7788, pages 15–44, 2013.
- [FKU17] Michael Färber, Cezary Kaliszyk, and Josef Urban. Monte Carlo tableau proof search. In de Moura [dM17], pages 563–579.
- [GL13] Stéphane Graham-Lengrand. Psyche: a proof-search engine based on sequent calculus with an LCF-style architecture. In Didier Galmiche and Dominique Larchey-Wendling, editors, *Proc. of the 22nd Int. Conf. on Automated Reasoning with Analytic Tableaux and Related Methods (Tableaux'13)*, volume 8123 of *LNCS*, pages 149–156. Springer-Verlag, September 2013.
- [GMW79] Michael Gordon, Robin Milner, and Christopher Wadsworth. *Edinburgh LCF: a mechanized logic of computation*, volume 78. 1979.
- [HOL] The HOL system.
- [Isa] The Isabelle theorem prover.
- [ISA⁺16] Geoffrey Irving, Christian Szegedy, Alexander A. Alemi, Niklas Eén, François Chollet, and Josef Urban. DeepMath - deep sequence models for premise selection. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *NIPS*, pages 2235–2243, 2016.
- [JU17] Jan Jakubuv and Josef Urban. ENIGMA: efficient learning-based inference guiding machine. In Herman Geuvers, Matthew England, Osman Hasan, Florian Rabe, and Olaf Teschke, editors, *CICM*, volume 10383 of *LNCS*, pages 292–302. Springer, 2017.
- [Mil79] Robin Milner. LCF: A way of doing proofs with a machine. In Jirí Běčvář, editor, *Proc. of the 8th Int. Symp. on Mathematical Foundations of Computer Science*, volume 74 of *LNCS*, pages 146–159. Springer-Verlag, 1979.

- [MMZ⁺01] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *DAC*, pages 530–535, New York, NY, USA, 2001.
- [SHM⁺16] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.
- [SKTW17] Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focussed molecule libraries for drug discovery with recurrent neural networks. *CoRR*, abs/1701.01329, 2017.
- [SPW17] Marwin H. S. Segler, Mike Preuss, and Mark P. Waller. Learning to plan chemical syntheses. *CoRR*, abs/1708.04202, 2017.
- [WTWD17] Mingzhe Wang, Yihe Tang, Jian Wang, and Jia Deng. Premise selection for theorem proving by deep graph embedding. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 2783–2793, 2017.