

Reinforcement Learning of Theorem Proving

October 31, 2019

Overview

1 How tableau provers work

2 Reinforcement Learning

- Basics
- Application to ATP
- Results

3 Summary

How tableau provers work

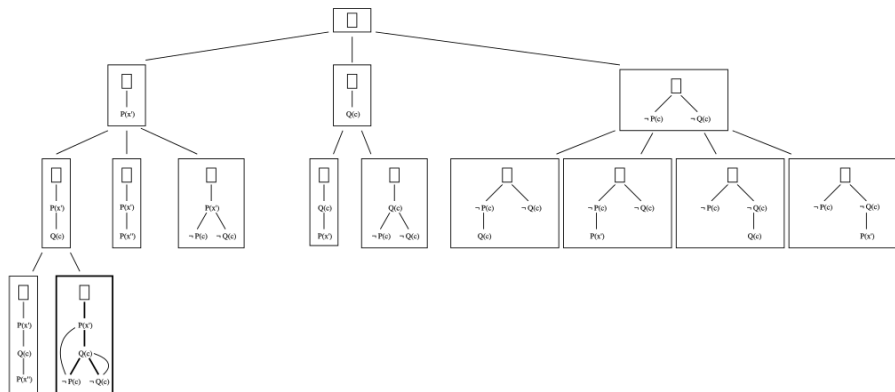


Figure: The search tree of a tableau based ATP¹

¹Source: Wikipedia

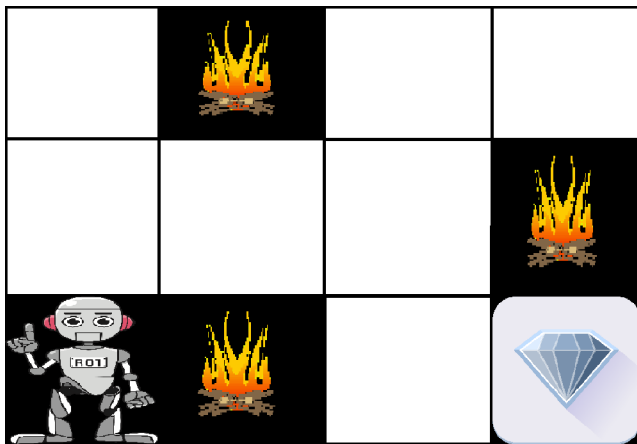


Figure: An agent has to reach a reward without burning²

²Source: GeeksForGeeks

Application to ATP

RL to ATP mapping:

- agent \leftrightarrow ATP
- environment \leftrightarrow search tree
- actions \leftrightarrow extending search tree
- reward \leftrightarrow finding a closed tableau

Application to ATP – the UCT formula

For each node i , we calculate the Upper Confidence bound applied to Trees, i.e. UCT(Kocsis and Szepesvári, 2006):

$$\frac{w_i}{n_i} + c \cdot p_i \cdot \sqrt{\frac{\ln N_i}{n_i}}$$

- w_i – total reward
- n_i – number of visits
- c – hyperparameter
- p_i – prior probability
- N_i – total parent visits

Application to ATP – Learning the parameters

Step 1: Extracting features

For each Literal L , e.g. $f(X, Y) = g(sk_1, sk_2(x))$ ³

- Build its **feature tree**
- Count the **term walks** of length 3
 - E.g. $(=, f, *)$ occurs twice
- Count the occurrences of each triple, create literal and clause feature vectors
- Feature vector for each state contains:
 - the triplet occurrence counts for each of its clauses and goals
 - additional metadata, e.g. number of goals, most common symbols, etc.

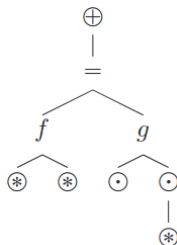


Figure: Feature Tree

³Example from Jakubuv and Urban (2017)

Application to ATP – Learning the parameters

Steps 2&3: Data Extraction and Learning

- Unrestricted ATP runs to accumulate (some) UCT proof data
- Associate node state and action features with action 'relevance'
 - $\frac{\text{total frequency of action}}{\text{action frequency at node}}$
- Associate node state feature with value
 - 0 if node not a proof
 - $0.99^{\text{proof depth}}$ otherwise
- Apply regression on the logits to learn 'relevance' and value

Results

Results from 2003 problems of the Mizar Mathematical Library (Grabowski et al., 2010) with limit of 2×10^6 inferences.

Iteration	1	5	10	15	20
Proved	1037	1182	1210	1223	1235

Table: Proved problems per iterations of learning

Methodology	Proved	IPS
Heuristics	876	64K
RL	1235	16K

Table: Using RL gives 40% more proves but slows down the inference speed

- Reinforcement Learning can be applied to tableau based provers
- Many new problems solved
- RL (and ML methods in general) slow down provers

Questions?

- Grabowski, A., Kornilowicz, A., and Naumowicz, A. (2010). Mizar in a nutshell. *Journal of Formalized Reasoning*, 3(2):153–245.
- Jakubuv, J. and Urban, J. (2017). ENIGMA: efficient learning-based inference guiding machine. In *Intelligent Computer Mathematics - 10th International Conference, CICM 2017, Edinburgh, UK, July 17-21, 2017, Proceedings*, pages 292–302.
- Kocsis, L. and Szepesvári, C. (2006). Bandit based monte-carlo planning. In *Proceedings of the 17th European Conference on Machine Learning, ECML'06*, pages 282–293, Berlin, Heidelberg. Springer-Verlag.