Reinforcement Learning of Theorem Proving

November 3, 2019

Overview

How tableau provers work

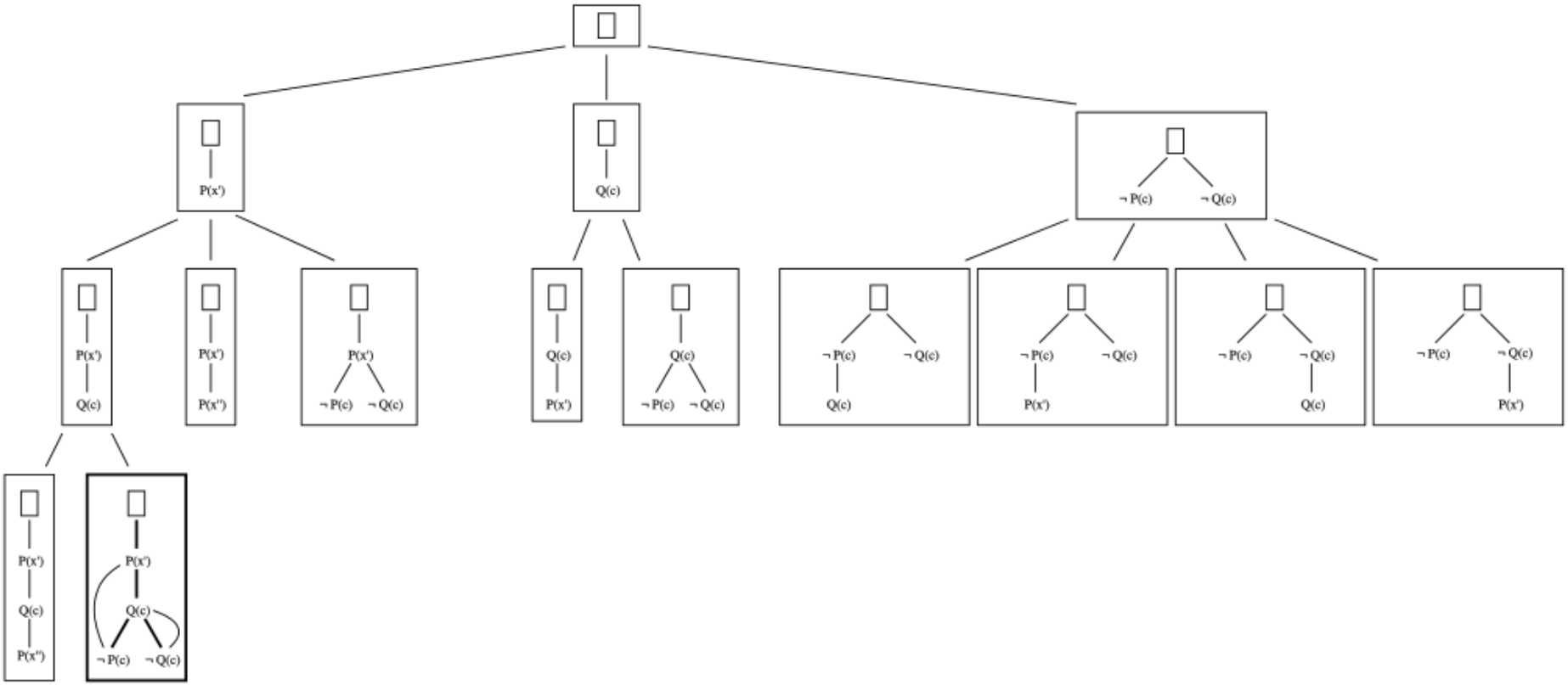


Figure: The search tree of a tableau based ATP[1]

[1] Source: Wikipedia

RL Basics

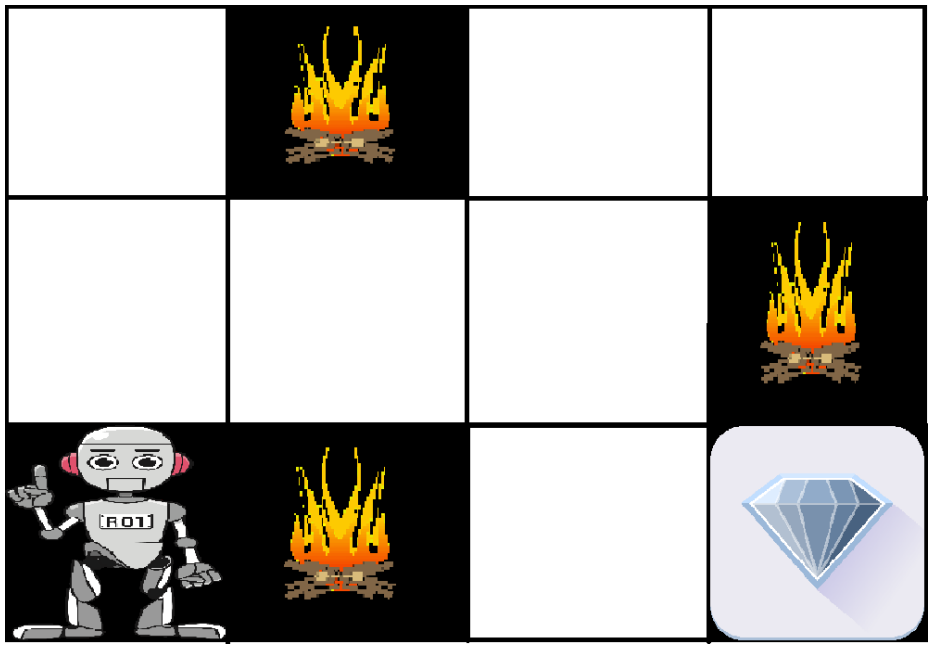


Figure: An agent has to reach a reward without burning[2]

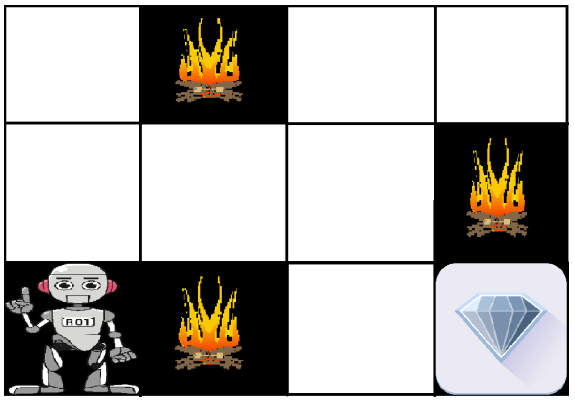
- policy learning
- value learning

[2]Source: Geeks for Geeks

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 formula

Let's combine policy and value learning into a formula:

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

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

On every step, take the node with highest f_i.

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 it's feature tree
- Count the term walks of length 3
  - E.g. $(=, f, \circledast)$ occurs twice
- Count the occurrences of each triple, create literal and clause feature vectors
- E.g. for L we get $\{(+, =, f) : 1, (=, f, \circledast) : 2\}$

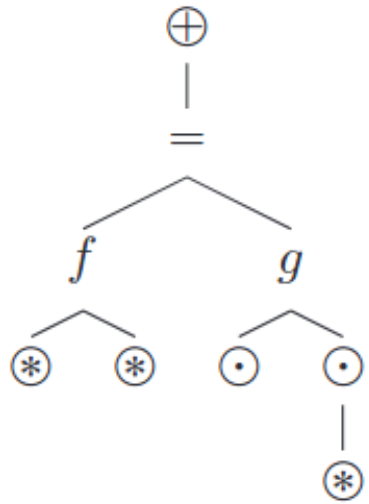


Figure: Feature Tree

[3] Example and picture from Jakubuv and Urban (2017)

Application to ATP – Learning the parameters

Step 1: Extracting features

Feature vector for each state contains:

- the triplet occurrence counts for each of its clauses and goals
- additional metadata, e.g. number of open goals, most common symbols, etc.

Application to ATP – Learning the parameters

Steps 2&3: Data Extraction and Lerning

- Unrestricted ATP runs to accumulate (some) proof data

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

- Associate node state and action features with action 'relevance'
  - $\frac{\textit{total frequency of action A}}{\textit{action A frequency at node}}$
- Associate node state feature with value
  - 0 if node not a proof
  - $0.99^{\textit{proof depth}}$ 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

Summary

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

Questions?

Bibliography

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.