Saw

ANONYMOUS AUTHOR(S)

1 COST FUNCTION

A cost function $cost : ProofScript \longrightarrow \mathbb{R}$ evaluate the goodness of a given proof script (i.g. a sequence of rewriting rules that have no adjacent duplicate rules (e.g. $[a; a; b; a]$ should be reduced to $[a; b; a]$)), the better proof scripts should have lower costs. If a proof script works dependent on the proof goal (i.g. a proof state, represented as an AST) we want to prove, thus we should consider the proof goal when we build the cost function. For a given proof goal a and a size bound k, the ideal cost function $cost^*$ is

$$cost^*(p : ProofScript) \equiv -\mathrm{Avg}_{|p'| \leq k} \frac{|PPrefix(a, p \oplus p')| - |PPrefix(a, p)|}{|p \oplus p'| - |p|}$$

where $\oplus$ is the merge operation on two proof scripts and

$$PPrefix(a, p_0) = p \iff \begin{cases} p = p_0 & \text{when } a \overset{p_0}{\not\longrightarrow} Qed \\ p \text{ is the shortest prefix of } p_0 \text{ such that } a \xrightarrow{p} Qed \end{cases}$$

Intuitively, this cost function attempts to append all possible proof scripts within the size bound k to p to count how far p is from proving the given proof goal a. If p can also prove a, then $PPrefix(a, p \oplus p') = PPrefix(a, p)$ thus the cost is 0; otherwise, if there is no p' that can make $a \xrightarrow{p \oplus p'} Qed$, p will have cost -1.

However, this cost function is too complicated and only makes sense with a very large k ($k \longrightarrow \infty$). The approximate version of the cost function can use features to abstract the proof states to a vector of real numbers. Thus, we decompose the new cost function into two parts: features over proof states and weighted function:

(1) The feature space should be complete, that is a function $\alpha : State \times State \longrightarrow \mathbb{R}^n$ such that if two proof scripts have the same features in terms of the proof goal, we believe they should have the same "goodness", that is the same ability to prove the proof goal:

$$\begin{aligned} &\forall (p_1\ p_2 : ProofScript)(a\ b_1\ b_2 : State), \\ &\quad a \xrightarrow{p_1} b_1 \land a \xrightarrow{p_2} b_2 \land (\alpha(a, b_1) = \alpha(a, b_2)) \land (\exists (p_1' : ProofScript), a \xrightarrow{p_1 \oplus p_1'} Qed) \implies \\ &\quad (\exists (p_2' : ProofScript), a \xrightarrow{p_2 \oplus p_2'} Qed) \end{aligned}$$

(2) The weighted function should separate the good proof scripts from the bad scripts under the approximation of features. As a supervised learning, we should label some proof scripts as positive, some as negative. On the other hand, we can use the ideal cost function $cost^*$ to help us find the bad scripts.

For the given proof goal a, perhaps a better abstraction of a proof script p should consider all intermediate proof states during the rewriting of all rules in p from a.

2022. XXXX-XXXX/2022/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 EGRAPH

The egraph is used in the SMT solving and automated proving, perhaps there is a connection between SAW and egraph. One difference is that we do not know the meaning of the operators (nodes) in the graph...

ACKNOWLEDGEMENTS

REFERENCES