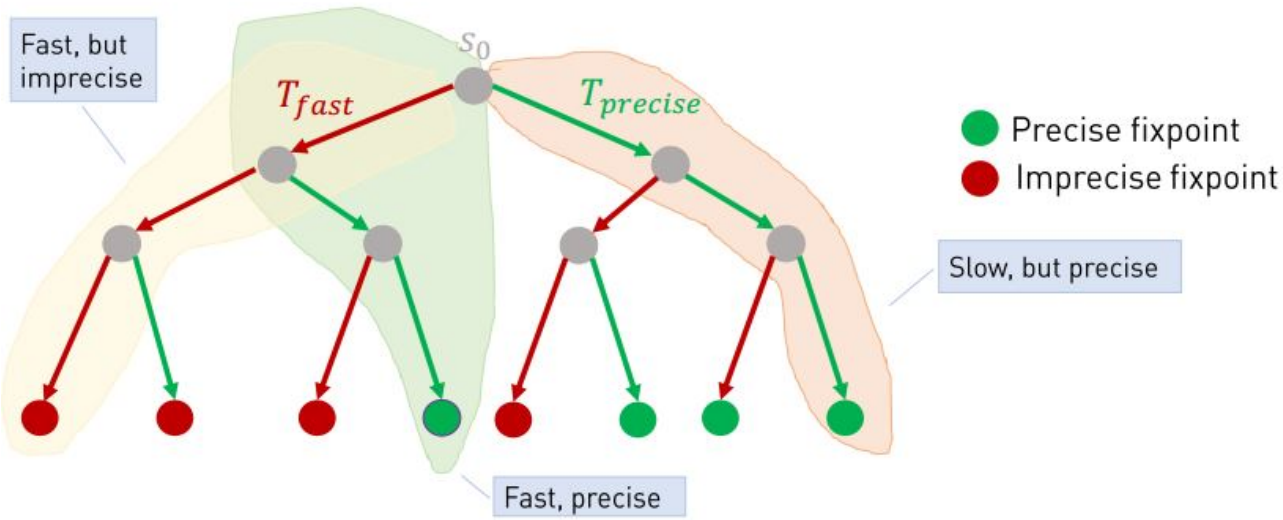
《Fast Numerical Program Analysis with Reinforcement Learning》Review

Paper Info

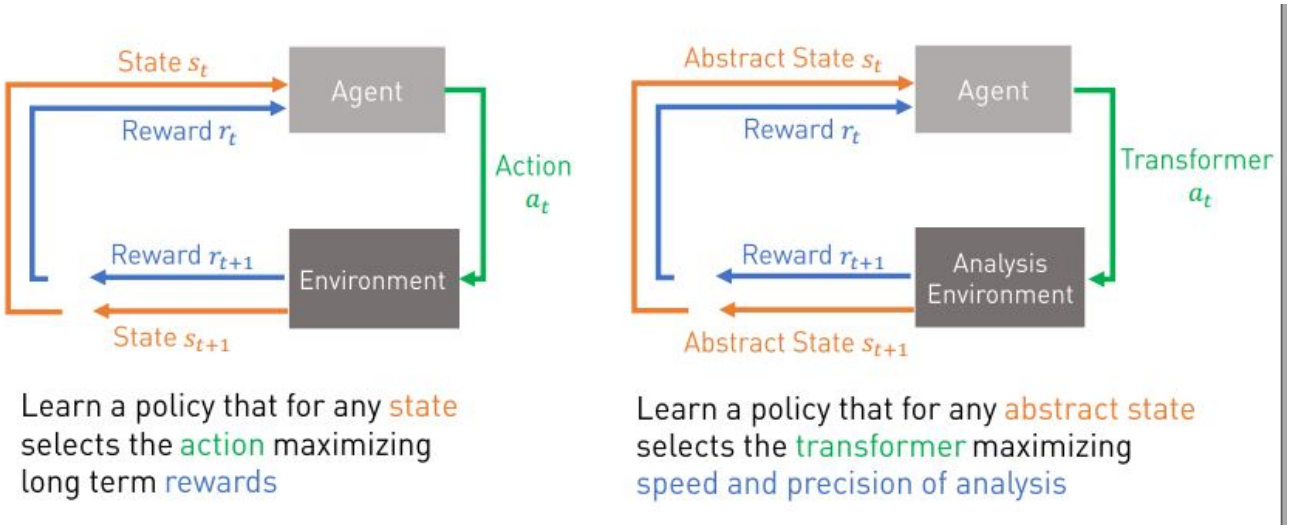
Fast Numerical Program Analysis with Reinforcement Learning Gaandeep Singh, Markus Puschel and Martin Vechev

Main Work

The work focus on the following problem: find a precise and fast numerical program analyzer to solve numercial abstract domain. The key work of this paper is instantiation of RL to Static Analysis. The mapping relation is following. RL concept Static analysis concept Angent Static Analyzer State Features of abstract state Action Abstract transformer Reward function Transformer precision and runtime Feature Value associated with abstract state features and transformer



In this task, we have two different choice: fast but imprecise transformer, and slow but precise transformer. In some situations, fast but imprecise transformer is enough to generate a fair precise abstract domain. Hence, we can leverage the RL method to do this work.



The reward can be set to the following way.

$$r(s_t, a_t, s_{t+1}) = 3n_s + 2n_b + n_{hb} - log_{10}(cyc)$$

The first three parameters reflect the precision of the numerical domain. The last parameter cyc reflects the time cost of the action. Hence, it can trade-off the precision and the efficiency very well.

In the end, Agents obtain the function Q:

$$Q(s, a) = \sum_{j=1}^{l} \alpha_j \phi_j(s, a)$$

We choose the action as:

$$a_t = argmax_{a \in \mathcal{A}} Q(s_t, a)$$

Future Work

This paper mainly focused on the polyhedra domain, and proposed a RL framework to make the analyzer work fast and precisely. In the future, some other numerical domains can be equipped to its own RL framework respectively.