Formal-Assisted Reinforcement Learning

Thuan Cao and Stefan Mitsch[0000-0002-3194-9759]

School of Computing
DePaul University
Chicago IL 60604, USA
{tcao8,smitsch}@depaul.edu

Keywords: Differential dynamic logic · Reinforcement Learning

Abstract

Reinforcement learning (RL) performs well for tasks in complex environments when clearly specified desired outcomes are available, but it comes at the cost of a lack of formal safety guarantees. This drawback is in part due to the "black-box" nature of RL and the finiteness of training runs, and in part due to semantic differences between formal safety models (nondeterministic physics models and nondeterministic action safety envelopes) and the RL training environment (simulated environment with stochastic actions). This makes translating the safety logic to the RL environment a safety-critical, laborious, and error-prone process.

We propose a formal specification language that ties formal models to RL environments through automated synthesis. Our method bases on differential dynamic logic to specify safety models, extended with annotations to aid translation to the Gymnasium environment. Structured syntactic annotations in the form of comments in the formal model allow the user to link the formal specification to the reward/penalty signals and the gradients more precisely.

```
// Rewarding speed (reward=isEfficient <=> reward=v-0)
// @progress: reward=isEfficient, on_success=continue, on_failure=continue
Bool isEfficient(Real v) <-> ( v > 0 );

// Limit speed and enforce safety distance (shape=isSafe <=> shape=(abs(x-xo) - stopDist(v))
// @safety: penalty=-1000, shape=isSafe, on_failure=truncate:status=failure, on_success=continue
Bool isSafe(Real x, Real xo, Real v) <-> (abs(x-xo) > stopDist(v));
```

The framework then automatically generates a RL environment (simulator) and reward signals. This allows the RL agent to be trained on the same safety specifications that were formally proven correct in differential dynamic logic and greatly reduces the effort of reward engineering.