Notes on Reinforcement Learning from Formal Specifications

Ramneet Singh

Advised by Suguman Bansal

April 2023

Contents

Paper : A Framework for Transforming Specifications in Reinforcement Learning

Overview

- Reactive Synthesis: Compute policies to control a **known MDP** satisfying a temporal logic specification.
  - Maximise the probability that an infinite execution of the system under the policy satisfies the specification.
  - Well-developed theory and tools
- Reinforcement Learning: Compute policies to control an **unknown MDP**, maximising some notion of aggregate reward (where each local transition is associated with a reward).
  - Maximise the expected aggregated reward of an infinite execution of the system under the policy.
  - **RL algorithms with convergence and efficient PAC guarantees are known for discounted-sum rewards.**
- New Research Area: **Develop RL algorithms for synthesis of policies from specifications**.
- Key Contribution: **A formal framework for reasoning about these techniques and their theoretical guarantees.**
  - Sampling-based reduction
  - Preservation of optimal policies, convergence, robustness
  - **Impossibility Result**: No RL algorithms with PAC-MDP guarantees for safety specifications.

Motivation

- **Problem with Rewards**: Too low-level, manually encoding desired behaviour is tough.
- **Why Logical Specifications?**:
  - More natural to specify higher-level objectives like "reach these targets in this order while avoiding obstacles".
  - Verifiable - can check if the policy satisfies the specification.
  - Can design specification-aware learning algorithms since it is known in advance.

Existing Techniques

- **Typical RL from Specifications Algorithm:**
    1. Translate the logical specification to an automaton that accepts executions that satisfy the specification.
    2. Define an MDP that is the product of the MDP being controlled and the specification automaton
    3. Associate rewards with the transitions of the product MDP so that either discounted-sum or limit-average aggregation (roughly) captures acceptance by the automaton.
    4. Apply an off-the-shelf RL algorithm such as Q-learning to synthesize the optimal policy.
- All existing algorithms have *conditions*, e.g.:
    - Convergence when the optimal policy satisfies the specification almost surely.
    - Parameterised reduction, with the discount factor being the parameter.

Plan

1. Define an RL task (M, ϕ) consisting of an MDP M and a specification ϕ. M is defined by its states, actions, reset and step functions. ϕ can be transition-based rewards, reward machines, safety specifications, reachability specifications, LTL formulas.

2. **It is not possible to reduce all LTL specifications to (discounted-sum) reward machines (which are reward functions with an internal state) when the underlying MDP M is kept fixed.**

3. Define *sampling-based reduction* from (M, ϕ) to $(\overline{M}, \phi')$, preservation of optimal policies, convergence, and robustness (that is, policies close to optimal in one get mapped to ones close to optimal in the other).

4. **Robust sampling-based reductions do not exist for transforming safety (as well as reachability) specifications to discounted rewards.**

5. **RL algorithms with PAC-MDP guarantees do not exist for safety (and reachability) specifications.**