Formally Verified Solution Methods for Markov Decision Processes

Maximilian Schäffeler
maximilian.schaeffeler@tum.de

Supervisors: Tobias Nipkow & Jan Křetínský
Collaborators: Mohammad Abdulaziz

Motivation

• Markov decision processes (MDPs) model systems where agent maximizes rewards under uncertainty
• Occur in safety-critical cyber-physical systems in planning, reinforcement learning, model checking

⇒ Use Interactive Theorem Provers to develop trustworthy software

Our Tool: Isabelle/HOL

• Isabelle/HOL is an interactive theorem prover with an extensive library (AFP), and powerful automation (sledgehammer)
• Supports code extraction, small & highly trustworthy kernel

Markov Decision Processes

An MDP consists of
• states $S$, enabled actions $A_s$ per state
• transition probabilities $K : S \times A \rightarrow P(S)$
• bounded rewards $r : S \times A \rightarrow \mathbb{R}$, with discount factor $\lambda$
• Goal of the agent: maximize rewards by optimizing a policy $\pi$ selecting actions based on past observations

Stochastic process

• Law of the stochastic process $\kappa$: extend run by one time step

$$\kappa : (S \times A)^{n+1} \rightarrow P(S \times A)$$

⇒ Used to define the probability space of sequences $T_x$

$\kappa(s_0, a_0, \ldots, s_n, a_n) =$ do{
  $s_{n+1} \leftarrow K(s_n, a_n)$; \text{ transition to next state} \\
  $a_{n+1} \leftarrow \pi(s_n, \ldots, a_n, s_{n+1})$; \text{ use policy to select action} \\
  return $(s_{n+1}, a_{n+1})$ \text{ return state-action pair}

Expected Total Reward

• $v_\pi$ is the cumulative, discounted reward collected during a run
• The discount factor makes the agent short-sighted

$$v_\pi = \sum_{t \geq 0} \lambda^t P_t r_t$$

Verified Algorithms

• We formalize algorithms to find $v^*$
  – (Gauss-Seidel) Value iteration
  – (Modified) Policy Iteration

• Refine algorithms to be executable, use efficient data structures
• Improvement: execute all but the last iteration on floats

• Benchmarks vs. Prism, Storm on problems from IPC 2018 show a slowdown of 1-2 orders of magnitude. Precise arithmetic is expensive.

Proof Example

lemma contraction_L: \(\dist(L \cdot v) (L \cdot u) \leq L \cdot \dist(v \cdot u)\)

proof -
  have aux: "L \cdot v \cdot s - L \cdot u \cdot s \leq \| \cdot \dist(v \cdot u)"
  if less: "(\cdot v \cdot s) \geq (\cdot u \cdot s)" for $v \cdot s$
  proof -
  have "L \cdot v \cdot s \cdot L \cdot u \cdot s \leq \| \cdot \dist(v \cdot u)"
  by (auto simp L_def algebra simp)
  also have "\cdot \leq \| \cdot dist(v \cdot u)"
  by (auto simp linf_unit)
  using abs le norm from by fastforce
  also have "\cdot \leq \| \cdot dist(v \cdot u)"
  by (auto simp linf_unit simp_norm)
  finally show thesis
  by auto
  qed

thus "\dist(L \cdot v) (L \cdot u) \leq \| \cdot \dist(v \cdot u)\ for s using aux auxf(v \cdot u) by (cases \cdot v \cdot s \geq \cdot L \cdot u \cdot s)
  (auto simp: dist_real_def dist_commute)

Conclusion & Future Work

• ITPs with powerful automation and a large library of formal proofs enable developing executable algorithms for reasoning under uncertainty
• Certification of unverified solvers can handle larger state spaces
• Future directions: floating-point guarantees, Monte-Carlo algorithms, factored MDPs

References