Probabilistic Model Checking for Safety and Performance Guarantees

Dave Parker

University of Birmingham

Dagstuhl seminar "Analysis of Autonomous Mobile Collectives in Complex Physical Environments", October 2019
Probabilistic model checking

- Probabilistic model checking
  - formal construction/analysis of probabilistic models
  - “correctness” properties expressed in temporal logic
  - e.g. trigger → P_{\geq 0.999} [ F_{\leq 20} \text{ deploy} ]
  - mix of exhaustive & numerical/quantitative reasoning

- Typically focus on numerical/quantitative results
  - analyse trends, look for system flaws, anomalies

- Wide range of quantitative properties expressible
  - probabilities, timing, energy, costs, rewards, ...
  - reason about safety, reliability, performance, timeliness, ...
PRISM (and extensions)

- **PRISM model checker**: [www.prismmodelchecker.org](http://www.prismmodelchecker.org)
- **Wide range of probabilistic models**
  - Discrete states & probabilities: *Markov chains*
  - + Nondeterminism: *Markov decision processes* (MDPs)
  - + Real-time clocks: *probabilistic timed automata* (PTAs)
  - + Partial observability: *POMDPs* and *POPTAs*
  - + Multiple players: *(turn-based)* stochastic games
  - + Concurrency: *concurrent stochastic games*
- **Unified modelling language/approach**
- **Various verification engines**: symbolic, explicit-state, exact, parametric, statistical model checking, abstraction, …
- **Many application domains**: network/comm. protocols, security, biology, robotics & planning, power management, scheduling, …
Probabilistic models

• Discrete–time Markov chains (DTMCs)
  – e.g. what is the probability of reaching state $t$?
  – e.g. $P_{<0.0001} [ F t ]$

• Markov decision processes (MDPs)
  – mix nondeterministic and probabilistic choice
  – strategies (or policies) resolve actions based on history
  – e.g. what is the maximum probability of reaching $t$ achievable by any strategy?

• Either:
  – adversarial view, i.e. verify against any possible strategy
  – or control view, i.e. synthesise a safe/optimal strategy
Application: Mobile robot navigation

- **Robot navigation planning**: [IROS'14, IJCAI’15, ICAPS’17, IJRR’19]
  - synthesis of plans for tasks with **probabilistic guarantees**
  - **MDP** models navigation through uncertain environment
  - stochastic time delays due to obstacles (typically human traffic)
  - MDP parameters/distributions learnt from logs of previous exploration
Application: Mobile robot navigation

- **Formal task description using co-safe LTL**
  - flexible, unambiguous specification
  - e.g. $\neg \text{zone}_3 \ U (\text{zone}_1 \land (F \text{zone}_4))$ – “patrol zones 1 then 4, without passing through zone 3”

- **Meaningful guarantees on performance**
  - probability of successful task completion (within deadline)
  - optimal strategies for timely task completion
  - c.f. ad-hoc reward structures, e.g. with discounting
  - QoS guarantees fed into task planning

- **Implementation and evaluation**
  - finite-memory MDP strategies converted to navigation controllers
  - ROS module based on PRISM
  - 100s of hrs of autonomous deployment
Application: UUV mission plans

- **PRINCESS**: Developing verified adaptive software systems
  - for operation in dynamic and uncertain environments
  - focus: autonomous underwater vehicle navigation
  - DARPA-funded project, under the BRASS program

- Adaptations are verified at runtime
  - produce probabilistic guarantees of correctness/safety
  - mission (path) plans for ocean search operations

- Verification tasks
  - ensure low probability of mission failure
    - (vehicle loss due to excessive power consumption)
  - inputs: battery usage + failure models, ocean/tide models
    - Markov chain models constructed
Application challenge: Smart farm

- **What level of abstraction?**
  - “farm-level”: navigation grid (+ robot state: cargo, failures, ...)

- **Uncertainty/probability**
  - stochastic travel delays due to humans/vehicles
  - failures of individual robots

- **Verification/guarantees**
  - robot task sequence completed within time $T$ with probability $p$?
  - how does this vary as the underlying failures change?
  - can we synthesise a time-optimal plan?
  - how do we ensure a repair robot is always available?
More probabilistic model checking…

- **Multi-objective model checking** [TACAS’11], [ICAPS’17]
  - investigate trade-offs between conflicting objectives
  - e.g., strategy to minimises expected task time,
    while ensuring probability of task success is > p
  - ...and while ensuring location can always be reached within time T with probability q
  - multi-objective analysis via Pareto curves

- **Partially observable MDPs (POMDPs)** [RTS’17]
  - strategy sees only observations, not full state
  - strategy maintains belief state about the true state of the MDP
  - e.g. localisation error, sensor noise; uncertainty about state of robot 2
  - verification tool support in e.g. PRISM-pomdps
More probabilistic model checking...

- **Stochastic game model checking**
  - multiple agents/components with differing objectives
  - e.g., controller vs. environment; system vs. attacker
  - control + adversarial aspects combined

- **PRISM-games model checker**
  - probabilistic model checking of rPATL
  - “can robots 1,2 collaborate so that the probability of task completion within T is at least 0.95, whatever robots 3,4 do?”
  - turn-based and concurrent stochastic games [QEST’19]
  - Nash equilibria based properties [FM’19]

- **Multi-robot systems [IROS’18]**
  - combined task allocation and planning
  - performed on a sequential abstraction; probabilistic guarantees then computed on a product model fragment
Challenges

- **Scalability**
  - how to tackle state-space blow-up, especially for multi-robot

- **Further models/properties**
  - e.g. partial observability + stochastic games

- **Uncertainty**
  - how to represent/reason about model imprecision?
  - accuracy vs efficiency trade-offs

- **Machine learning**
  - how to reason about the integration of learning?