Probabilistic Model Checking and Strategy Synthesis for Robot Navigation

Dave Parker

University of Birmingham

(joint work with Bruno Lacerda, Nick Hawes)

AIMS CDT, Oxford, May 2015

Overview

- Probabilistic model checking
  - verification vs. strategy synthesis
  - Markov decision processes (MDPs)
- Application: Robot navigation
  - probabilistic model checking + MDPs + LTL
- Strategy synthesis techniques
  - multi-objective probabilistic model checking
  - partially satisfiable task specifications
  - uncertainty + stochastic games
  - permissive controller synthesis

Quantitative verification

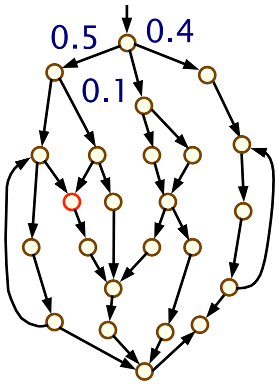
- Formal verification + quantitative aspects
- Probability
  - component failures, lossy communication, unreliable sensors/actuators, randomisation in algorithms/protocols

- Time: delays, time-outs, failure rates, …
- Costs & rewards
  - energy consumption, resource usage, …

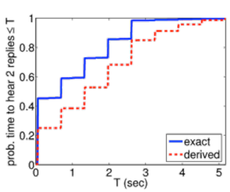
- Not just about correctness…
  - reliability, timeliness, performance, efficiency, …
  - "the probability of an airbag failing to deploy within 0.02 seconds of being triggered is at most 0.001"
  - "the expected energy consumption of the sensor is…"

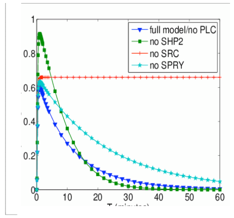
Probabilistic model checking

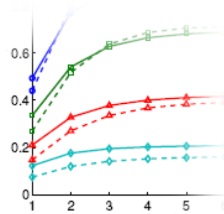
- Construction and analysis of probabilistic models
  - state-transition systems labelled with probabilities (e.g. Markov chains, Markov decision processes)
  - from a description in a high-level modelling language

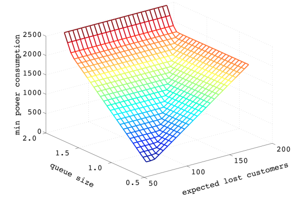


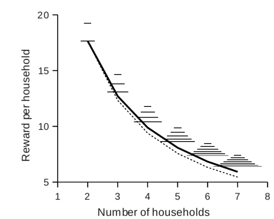
- Properties expressed in temporal logic, e.g. PCTL:
  - trigger $\rightarrow P_{\geq 0.999}$ [$F^{\leq 20}$ deploy]
  - "the probability of the airbag deploying within 20ms of being triggered is at at least 0.999"
  - properties checked against models using exhaustive search and numerical computation





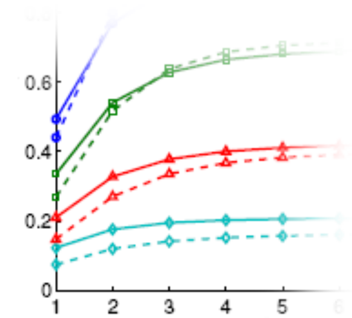






Probabilistic model checking

- Many types of probabilistic models supported
- Wide range of quantitative properties, expressible in temporal logic (probabilities, timing, costs, rewards, …)
- Often focus on numerical results (probabilities etc.)
  - analyse trends, look for system flaws, anomalies

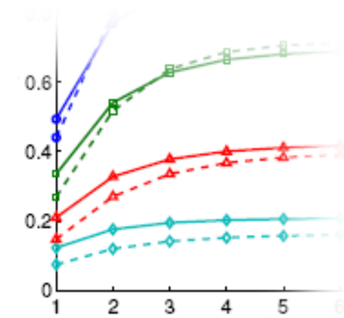


- $P_{\leq 0.1}$ [F *fail*] – “the probability of a failure occurring is at most 0.1”

- $P_{=?}$ [F *fail*] – “what is the probability of a failure occurring?”

Probabilistic model checking

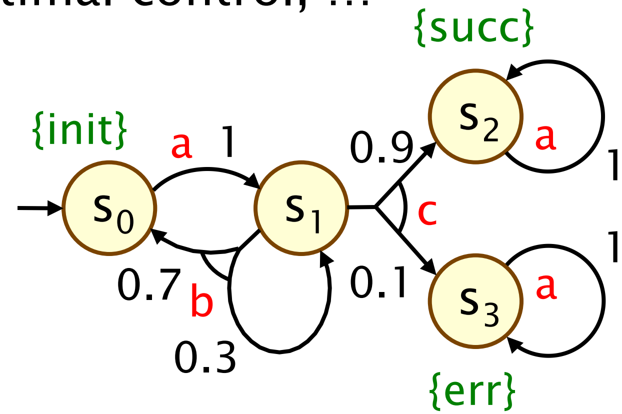
- Many types of probabilistic models supported
- Wide range of quantitative properties, expressible in temporal logic (probabilities, timing, costs, rewards, …)
- Often focus on numerical results (probabilities etc.)
  - analyse trends, look for system flaws, anomalies



- Provides "exact" numerical results/guarantees
  - compared to, for example, simulation/heuristics
  - combines numerical & exhaustive analysis
- Fully automated, tools available, widely applicable
  - network/communication protocols, security, biology, robotics & planning, power management, …
- Key challenge: scalability

Markov decision processes (MDPs)

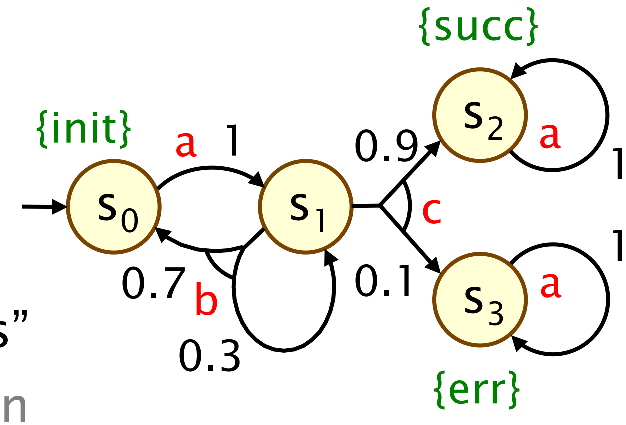
- Markov decision processes (MDPs)
  - also widely used also in: AI, planning, optimal control, …
- A strategy (or “policy” or “adversary”)
  - resolves choices in an MDP based on its history so far



- Used to model:
  - control: decisions made by a controller or scheduler
  - adversarial behaviour of the environment
  - concurrency/scheduling: interleavings of parallel components
- Classes of strategies:
  - memory: memoryless, finite-memory, or infinite-memory
  - randomisation: deterministic or randomised

Verification vs. Strategy synthesis

- 1. Verification
  - quantify over all possible strategies (i.e. best/worst-case)
  - $P_{\leq 0.1}$ [F *err*] : "the probability of an error occurring is $\leq$ 0.1 for all strategies"
  - applications: randomised communication protocols, randomised distributed algorithms, security, …



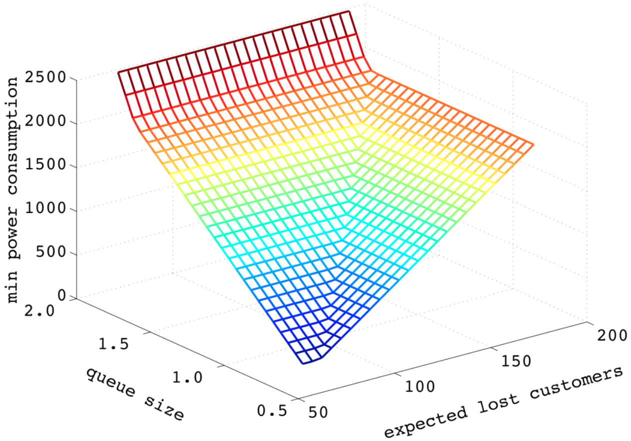
- 2. Strategy synthesis
  - generation of "correct-by-construction" controllers
  - $P_{\leq 0.1}$ [F *err*] : "does there exist a strategy for which the probability of an error occurring is $\leq$ 0.1?"
  - applications: robotics, power management, security, …

- Two dual problems; same underlying computation:
  - compute optimal (minimum or maximum) values

Applications

- Examples of PRISM-based strategy synthesis

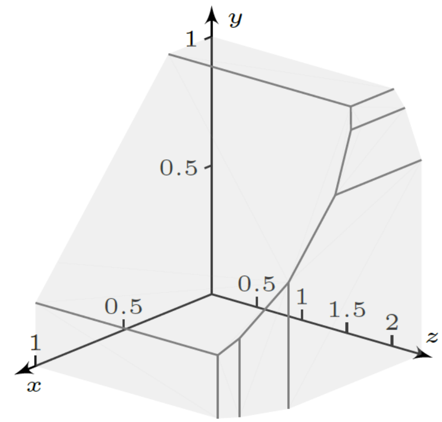
Synthesis of dynamic power management controllers [TACAS'11]



Minimise disk drive energy consumption, subject to constraints on:
(i) expected job queue size;
(ii) expected number of lost jobs

Motion planning for a service robot using LTL [IROS'14]

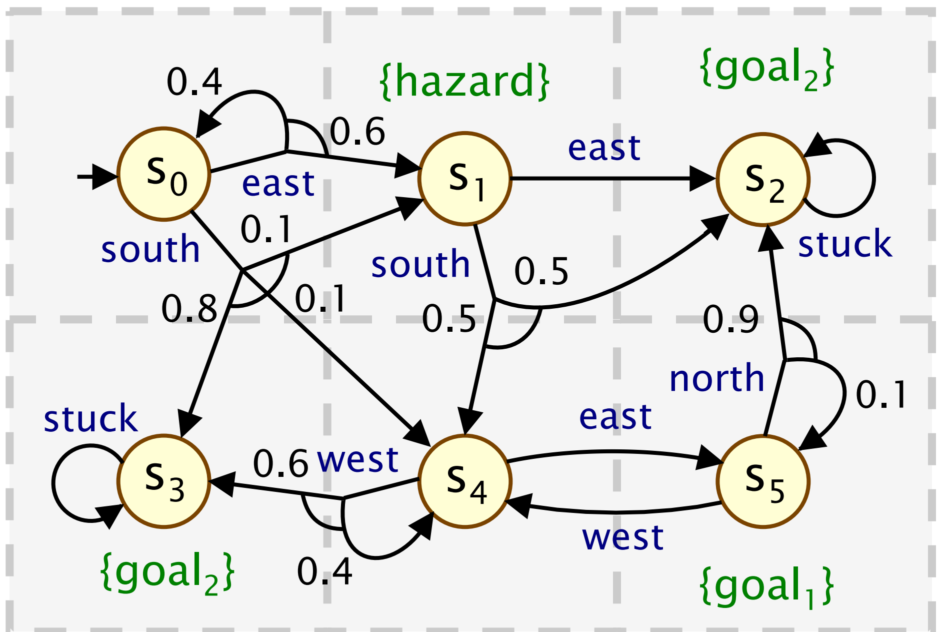
Team formation strategy synthesis [CLIMA'11, ATVA'12]



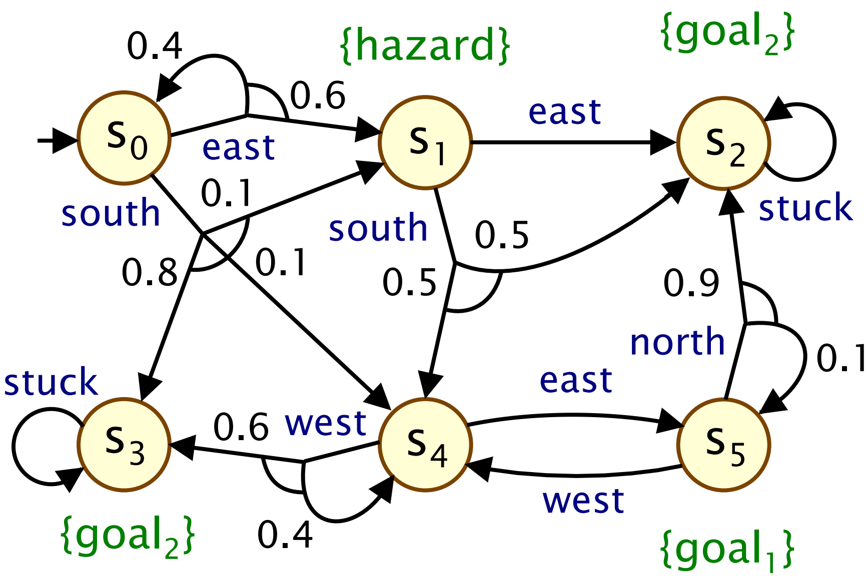
Pareto curve:
x="probability of completing task 1";
y="probability of completing task 2";
z="expected size of successful team"

Example

- Example MDP
  - robot moving through terrain divided in to 3 x 2 grid



Example – Reachability



Verify: $P_{\leq 0.6}$ [F goal$_1$]

or

Synthesise for: $P_{\geq 0.4}$ [F goal$_1$]

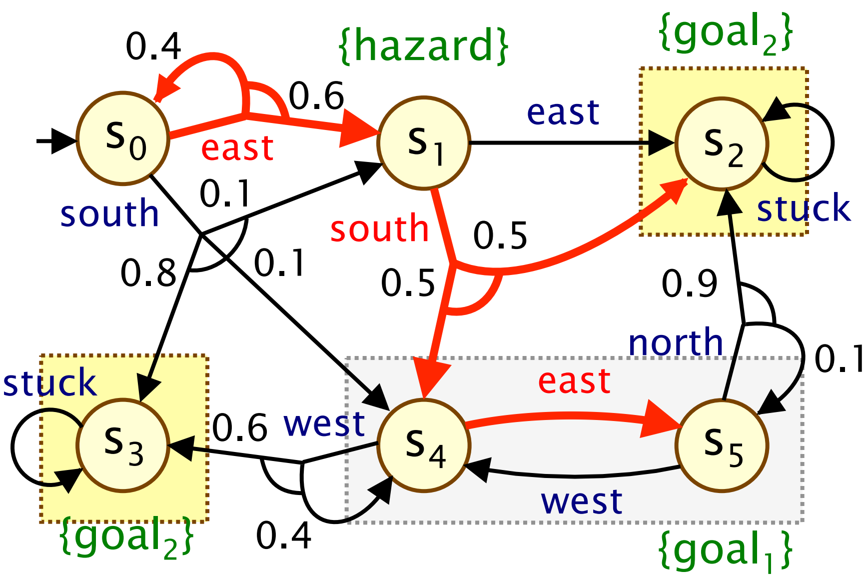
$\Downarrow$

Compute: $P_{max=?}$ [F goal$_1$]

Optimal strategies:
memoryless and deterministic

Computation:
graph analysis + numerical soln. (linear programming, value iteration, policy iteration)

Example – Reachability



Optimal strategy:
s_0 : east
s_1 : south
s_2 : –
s_3 : –
s_4 : east
s_5 : –

Verify: $P_{\leq 0.6}$ [F goal$_1$]

or

Synthesise for: $P_{\geq 0.4}$ [F goal$_1$]

$\Downarrow$

Compute: $P_{max=?}$ [F goal$_1$] = 0.5

Optimal strategies:
memoryless and deterministic

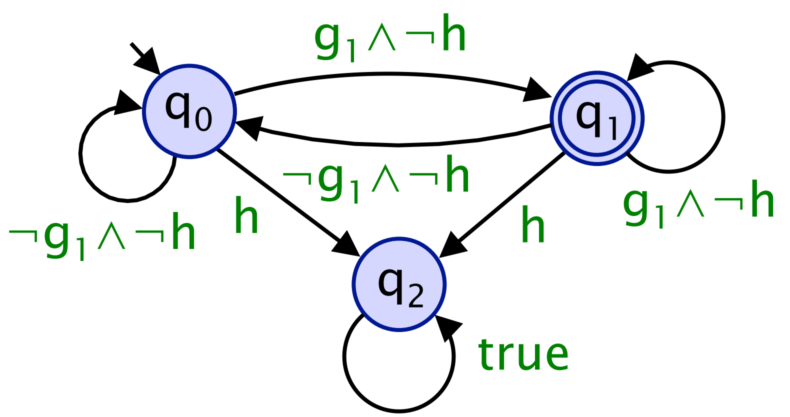
Computation:
graph analysis + numerical soln. (linear programming, value iteration, policy iteration)

Linear temporal logic (LTL)

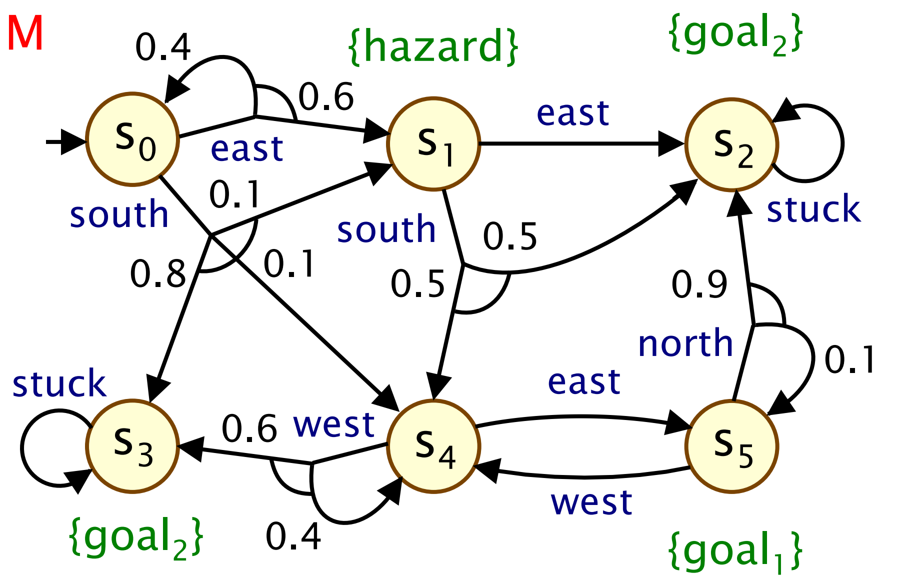
- Probabilistic LTL (multiple temporal operators)
  - e.g. $P_{max=?}$ [(G¬hazard) ∧ (GF $goal_1$)] – "maximum probability of avoiding hazard and visiting $goal_1$ infinitely often?"
  - e.g. $P_{max=?}$ [¬$zone_3$ U ($zone_1$ ∧ (F $zone_4$)] – "max. probability of patrolling zones 1 then 4, without passing through 3".

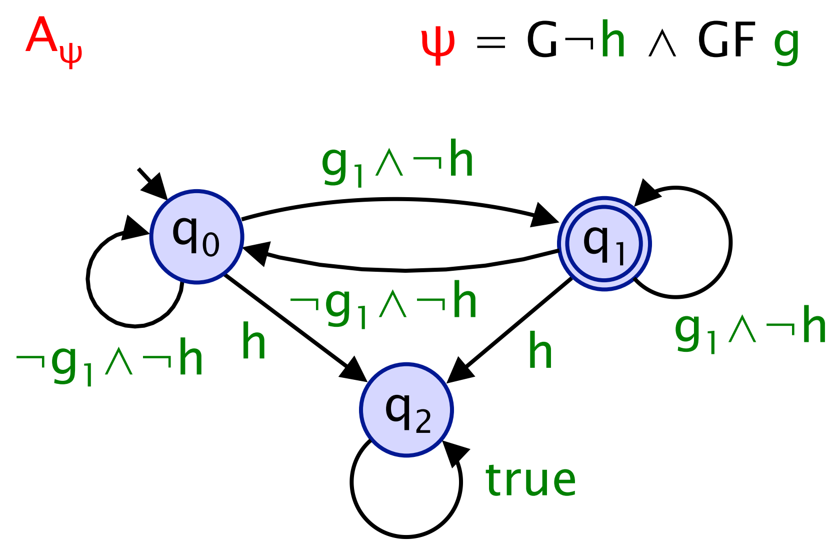
- Probabilistic model checking
  - convert LTL formula ψ to deterministic automaton A_ψ (Buchi, Rabin, finite, …)
  - build/solve product MDP $M \otimes A_\psi$
  - reduction to simpler problem
  - optimal strategies are:
    - deterministic
    - finite-memory

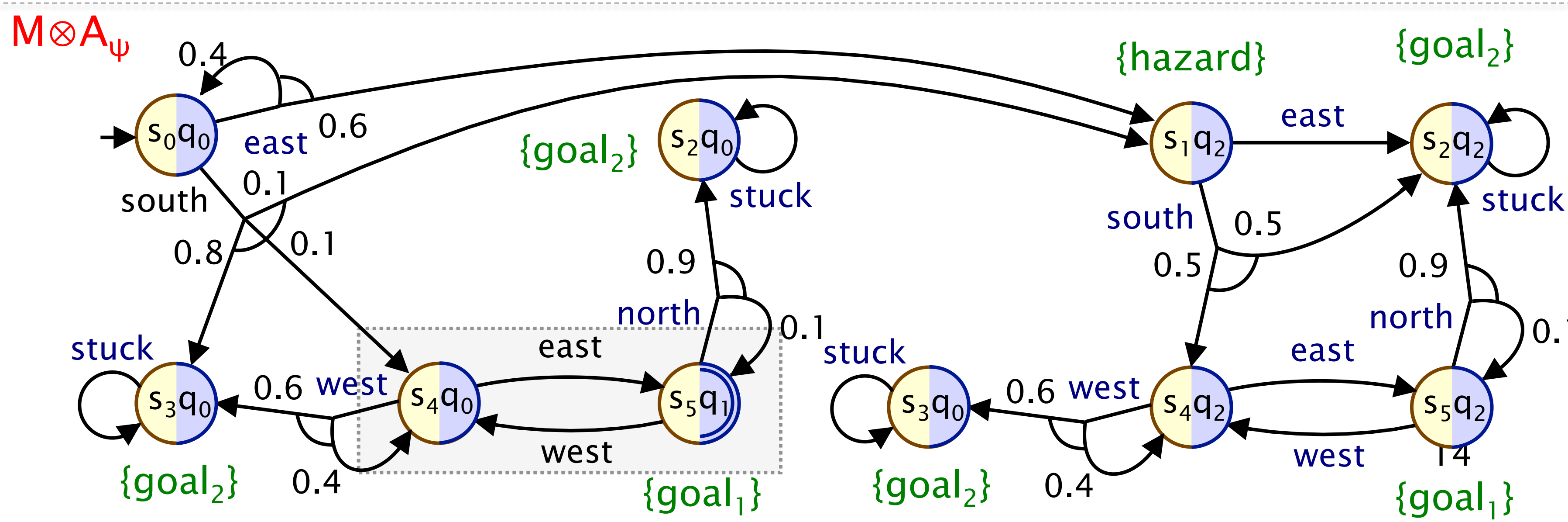
Det. Buchi automaton A_ψ
for ψ = G¬h ∧ GF g_1



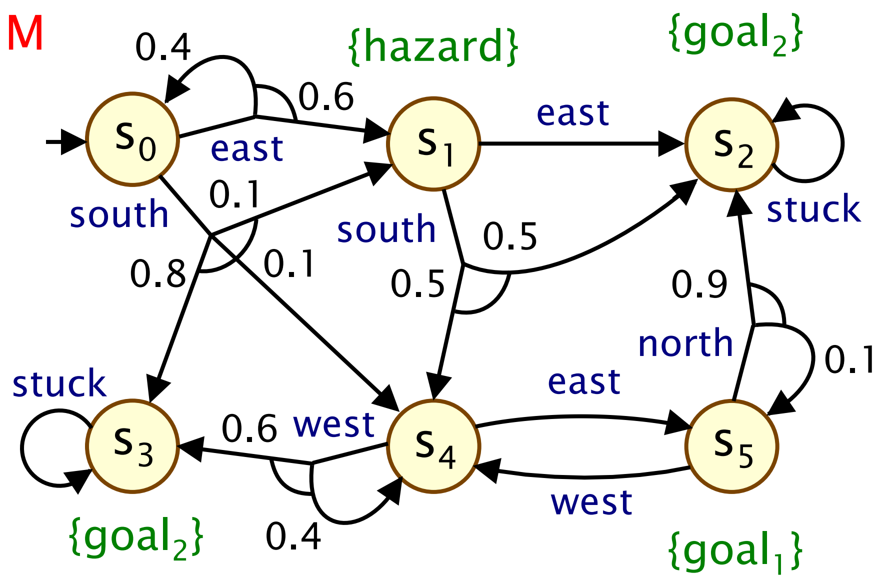
Example: Product MDP construction

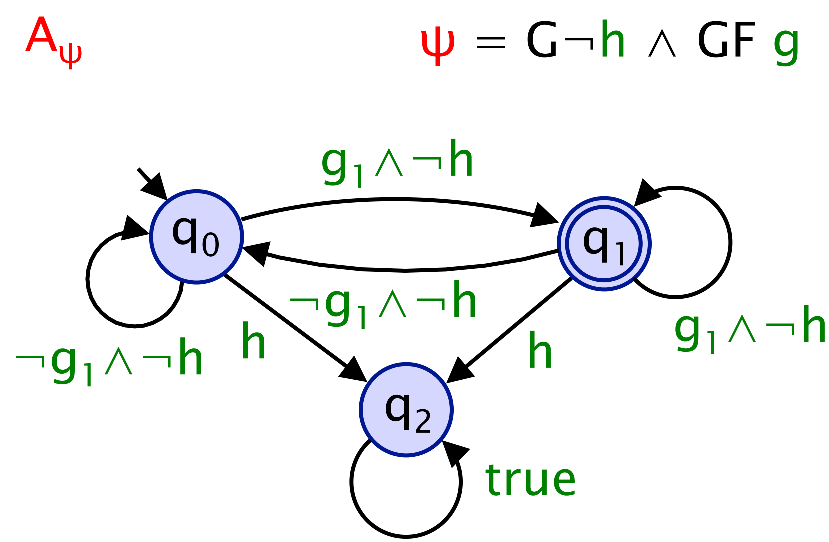


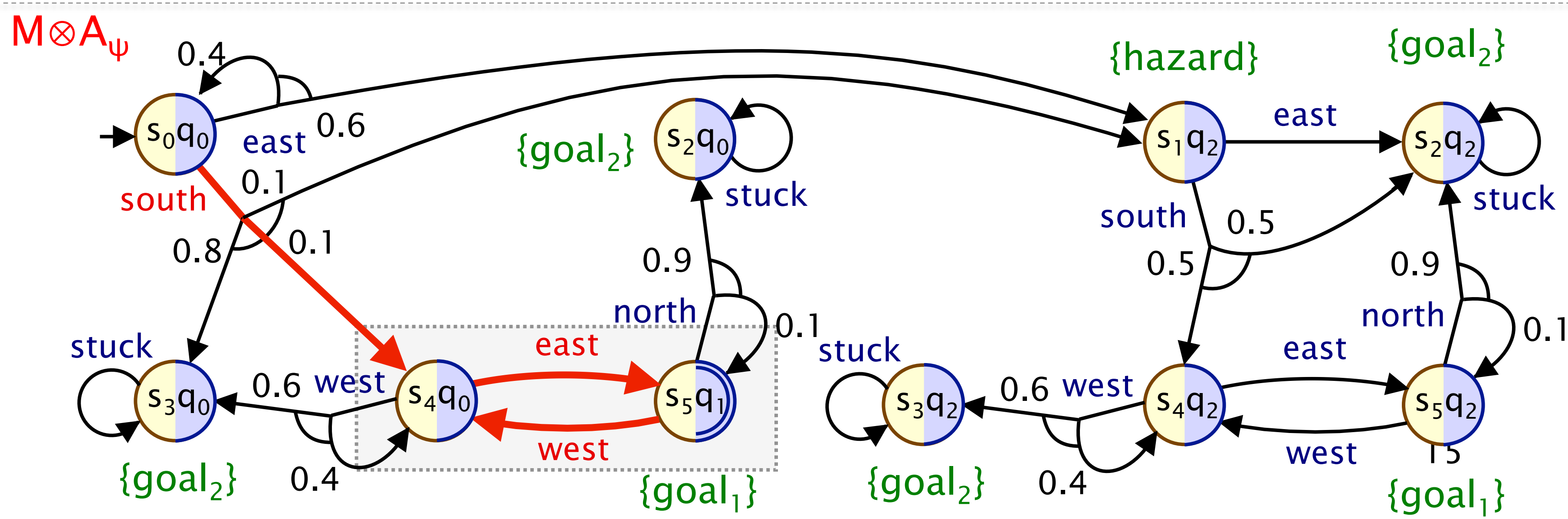




Example: Product MDP construction





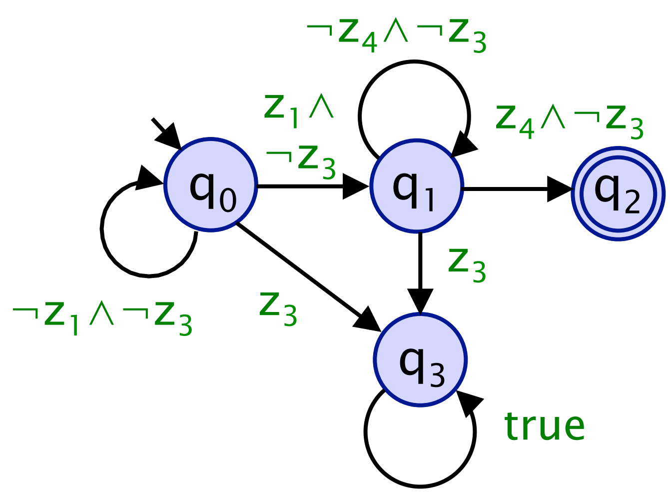


Co-safe LTL (and expected cost)

- Often focus on tasks completed in finite time
  - can restrict to co-safe fragment(s) of LTL
  - (any satisfying execution has a "good prefix")
  - e.g. $P_{max=?}$ [$\neg zone_3$ U ($zone_1 \wedge$ (F $zone_4$)]
  - for simplicity, can restrict to syntactically co-safe LTL

- Expected cost/reward to satisfy (co-safe) LTL formula
  - e.g. $R_{min=?}$ [$\neg zone_3$ U ($zone_1 \wedge$ (F $zone_4$)] – "minimise exp. time to patrol zones 1 then 4, without passing through 3".

- Solution:
  - product of MDP and DFA
  - expected cost to reach accepting states in product

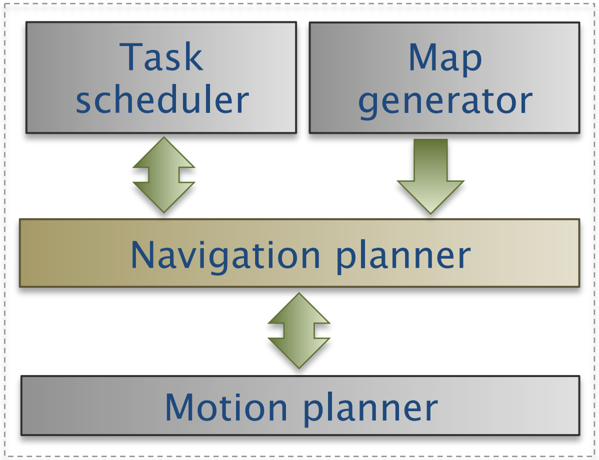


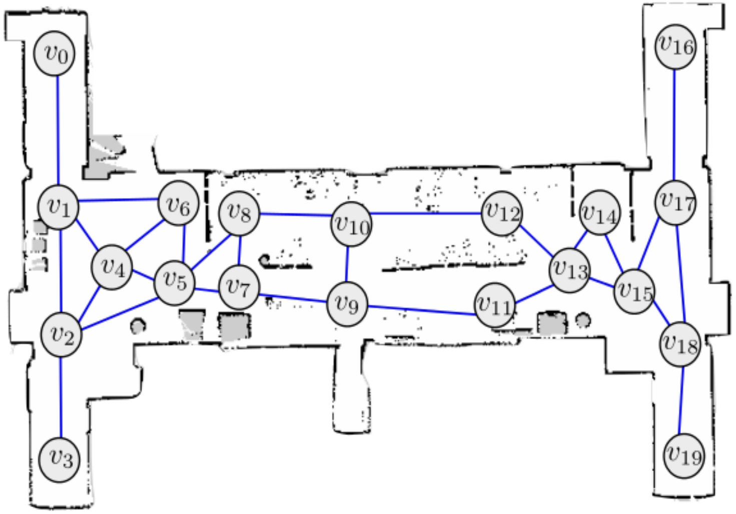
Overview

- Probabilistic model checking
  - verification vs. strategy synthesis
  - Markov decision processes (MDPs)
- Application: Robot navigation
  - probabilistic model checking + MDPs + LTL
- Strategy synthesis techniques
  - multi-objective probabilistic model checking
  - partially satisfiable task specifications
  - uncertainty + stochastic games
  - permissive controller synthesis

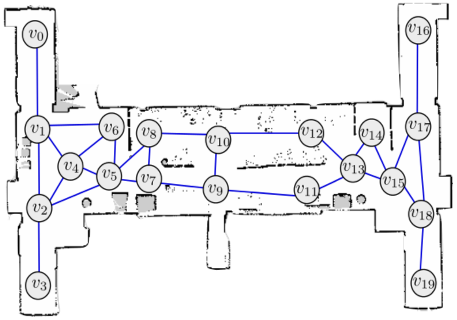
Application: Robot navigation

- Navigation planning:
  - MDP models navigation through an uncertain environment
  - LTL used to formally specify tasks to be executed
  - synthesise finite-memory strategies to construct plans/controllers





Application: Robot navigation



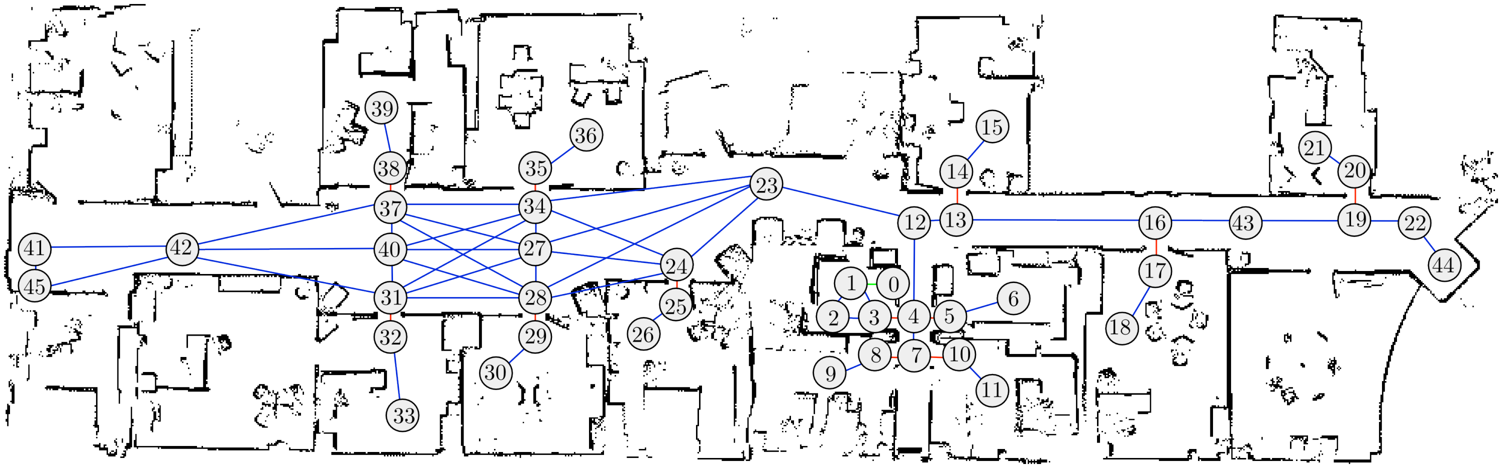
- Navigation planning MDPs
  - expected timed on edges + probabilities
  - learnt using data from previous explorations
- LTL-based task specification
  - expected time to satisfy (one or more) co-safe LTL formulas
- Benefits of the approach
  - LTL: flexible, unambiguous property specification
  - efficient, fully-automated techniques
    - LTL-to-automaton conversion, MDP solution
  - c.f. ad-hoc reward structures, e.g. with discounting
  - meaningful properties: probabilities, time, energy,…
  - guarantees on performance ("correct by construction")

Implementation & deployment

- Implementation
  - MetraLabs Scitos A5 robot
  - ROS module based on PRISM
  - with extensions:
    - co-safe LTL expectation
    - efficient re-planning [IROS'14]

G4S Technology, Tewkesbury (STRANDS)

- Example deployment:



Probabilistic model checking

- Further use of probabilistic model checking…
  - (various probabilistic models, query languages)

- Nested queries
  - e.g. $R_{min=?}$ [safe U (zone$_1$ $\wedge$ (F zone$_4$)] – "minimise exp. time to patrol zones 1 then 4, passing only through safe".
  - where safe denotes states satisfying $\langle\langle$ctrl$\rangle\rangle$ $R_{<2}$ [F base] – "there is a strategy to return to base with expected time $<$ 2"

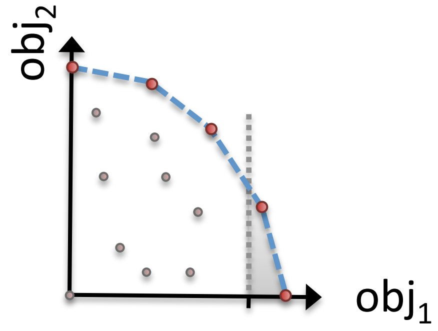
- Analysis of generated controllers
  - expected power consumption to complete tasks?
  - conditional expectation, e.g. expected time to complete task, assuming it is completed successfully?
  - more detailed timing information (not just mean time)

Overview

- Probabilistic model checking
  - verification vs. strategy synthesis
  - Markov decision processes (MDPs)
- Application: Robot navigation
  - probabilistic model checking + MDPs + LTL
- Strategy synthesis techniques
  - multi-objective probabilistic model checking
  - partially satisfiable task specifications
  - uncertainty + stochastic games
  - permissive controller synthesis

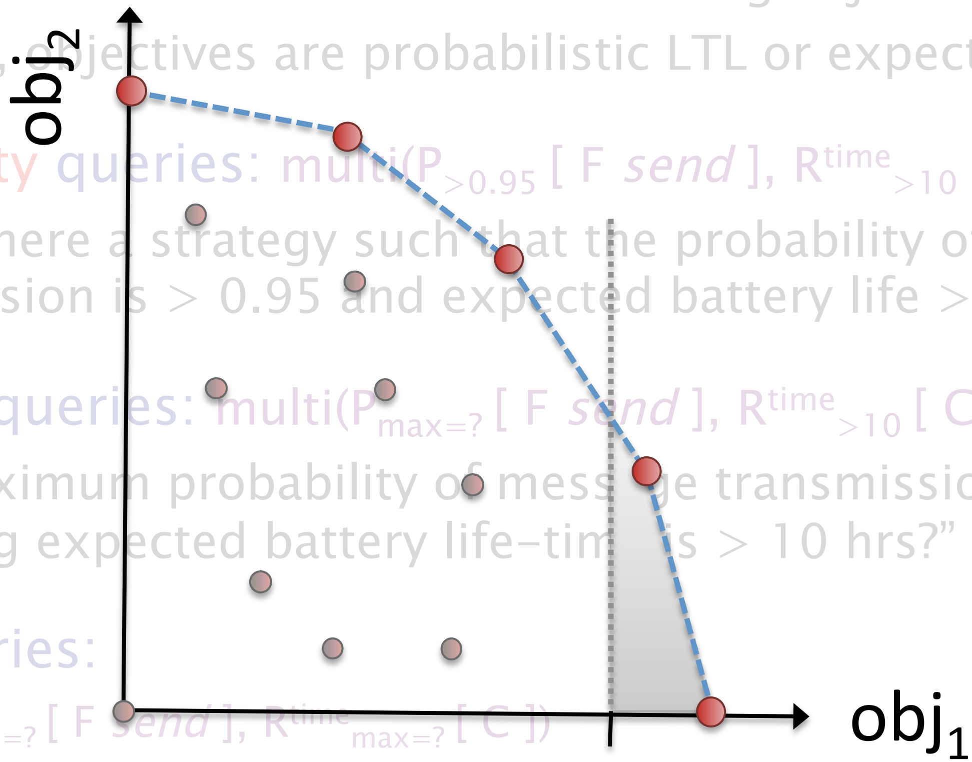
Multi-objective model checking

- Multi-objective probabilistic model checking
  - investigate trade-offs between conflicting objectives
  - in PRISM, objectives are probabilistic LTL or expected costs
- Achievability queries: multi($P_{>0.95}$ [F *send*], $R^{time}_{>10}$ [C])
  - e.g. "is there a strategy such that the probability of message transmission is > 0.95 and expected battery life > 10 hrs?"
- Numerical queries: multi($P_{max=?}$ [F *send*], $R^{time}_{>10}$ [C])
  - e.g. "maximum probability of message transmission, assuming expected battery life-time is > 10 hrs?"
- Pareto queries:
  - multi($P_{max=?}$ [F *send*], $R^{time}_{max=?}$ [C])
  - e.g. "Pareto curve for maximising probability of transmission and expected battery life-time"



Multi-objective model checking

- Multi-objective probabilistic model checking
  - investigate trade-offs between conflicting objectives
  - in PRISM, objectives are probabilistic LTL or expected rewards
- Achievability queries: multi($P_{>0.95}$ [F *send*], $R^{time}_{>10}$ [C])
  - e.g. “is there a strategy such that the probability of message transmission is > 0.95 and expected battery life > 10 hrs?”
- Numerical queries: multi($P_{max=?}$ [F *send*], $R^{time}_{>10}$ [C])
  - e.g. “maximum probability of message transmission, assuming expected battery life-time is > 10 hrs?”
- Pareto queries:
  - multi($P_{max=?}$ [F *send*], $R^{time}_{max=?}$ [C])
  - e.g. "Pareto curve for maximising probability of transmission and expected battery life-time”



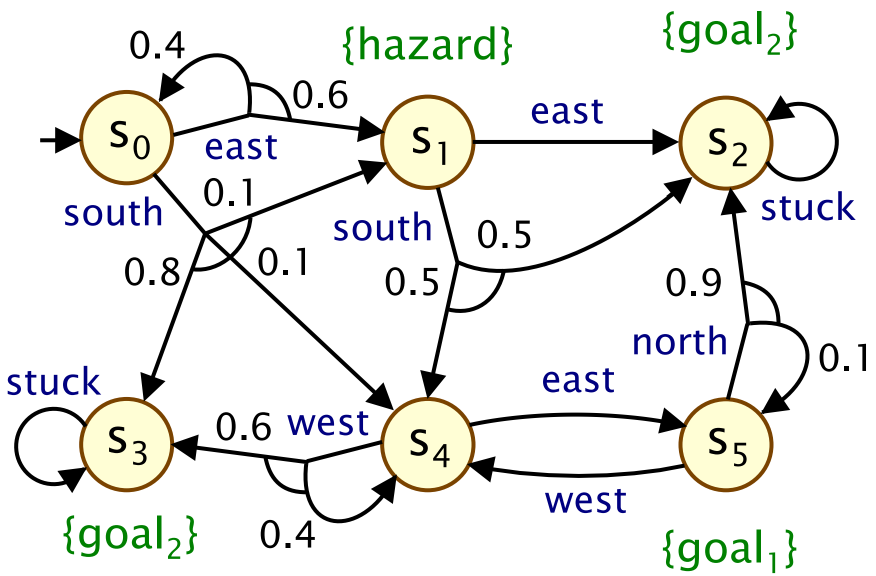
Multi-objective model checking

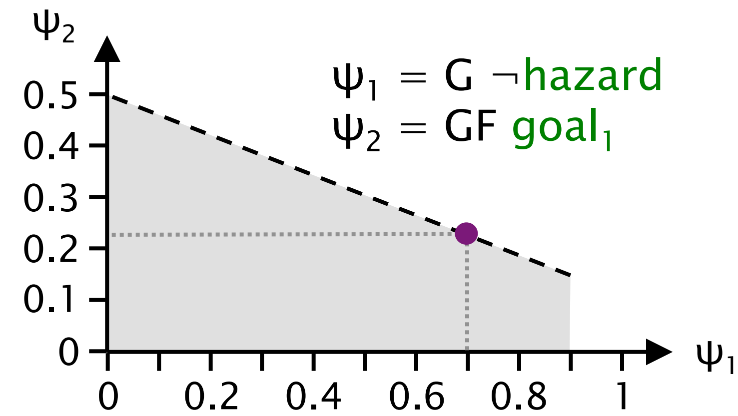
- Optimal strategies:
  - usually finite-memory (e.g. when using LTL formulae)
  - may also need to be randomised

- Computation:
  - construct a product MDP (with several automata), then reduces to linear programming [TACAS'07,TACAS'11]
  - can be approximated using iterative numerical methods, via approximation of the Pareto curve [ATVA'12]

- Extensions [ATVA'12]
  - arbitrary Boolean combinations of objectives
    - e.g. $\psi_1 \Rightarrow \psi_2$ (all strategies satisfying ψ_1 also satisfy ψ_2)
    - (e.g. for assume-guarantee reasoning)
  - time-bounded (finite-horizon) properties

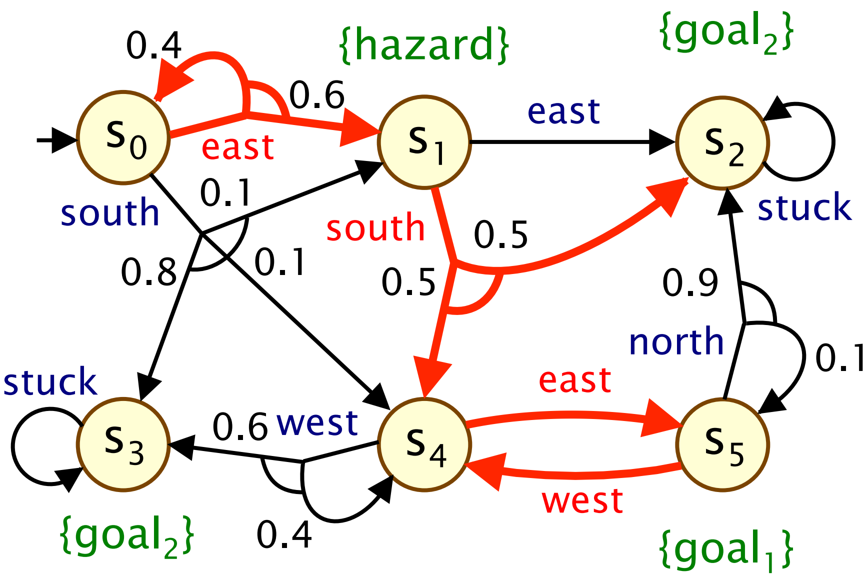
Example - Multi-objective





- Achievability query
  - $P_{\geq 0.7}$ [G ¬hazard] ∧ $P_{\geq 0.2}$ [GF goal$_1$] ? True (achievable)
- Numerical query
  - $P_{max=?}$ [GF goal$_1$] such that $P_{\geq 0.7}$ [G ¬hazard] ? ~0.2278
- Pareto query
  - for $P_{max=?}$ [G ¬hazard] ∧ $P_{max=?}$ [GF goal$_1$] ?

Example – Multi-objective



Strategy 1
(deterministic)

s_0 : east

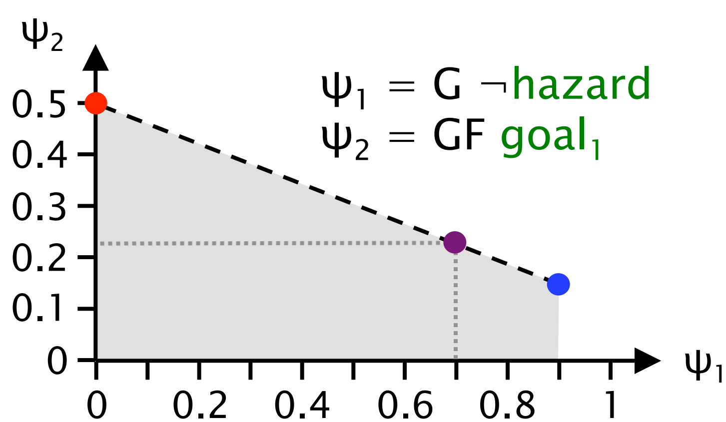
s_1 : south

s_2 : –

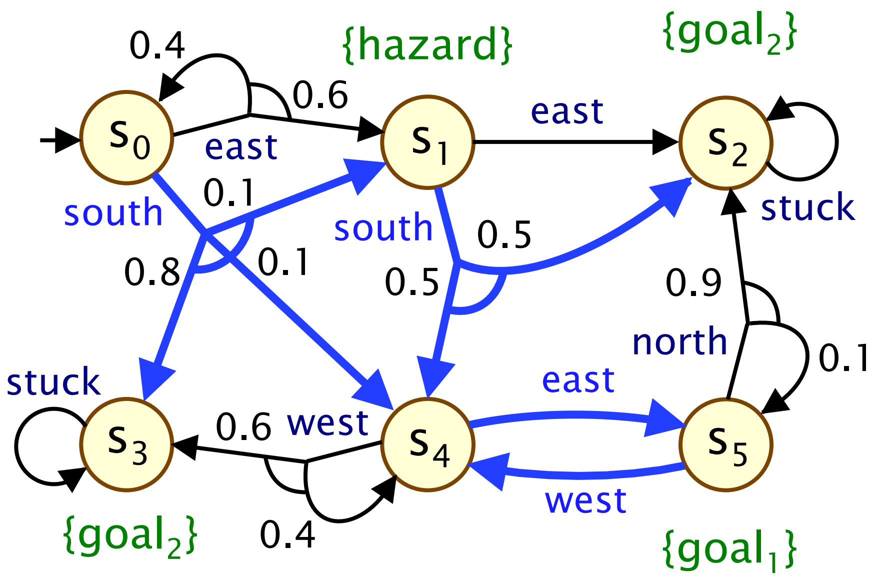
s_3 : –

s_4 : east

s_5 : west



Example - Multi-objective



Strategy 2
(deterministic)

s_0 : south

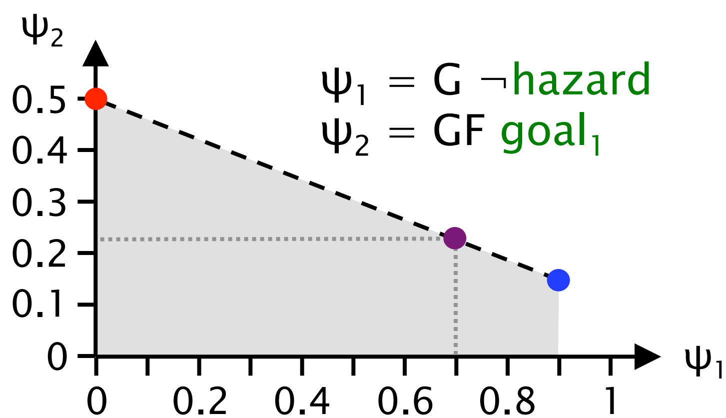
s_1 : south

s_2 : –

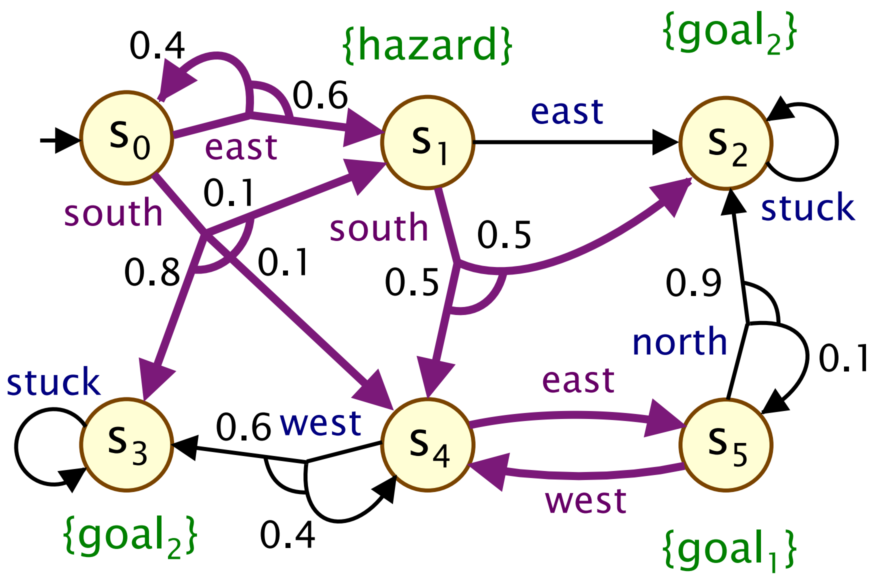
s_3 : –

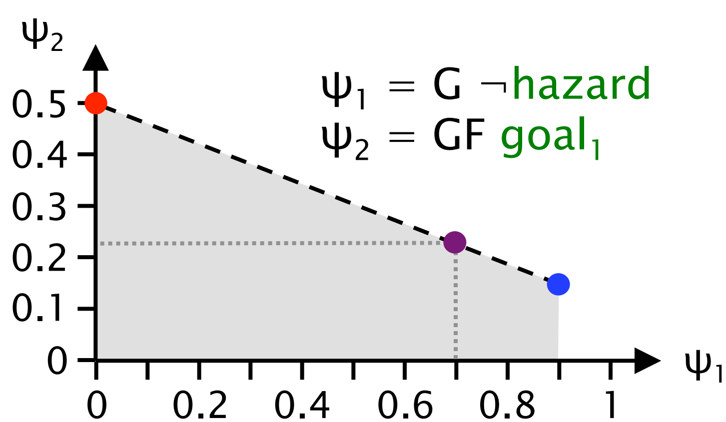
s_4 : east

s_5 : west



Example - Multi-objective





Optimal strategy:
(randomised)

s_0 : 0.3226 : east
0.6774 : south

s_1 : 1.0 : south

s_2 : –

s_3 : –

s_4 : 1.0 : east

s_5 : 1.0 : west

Application: Partially satisfiable tasks

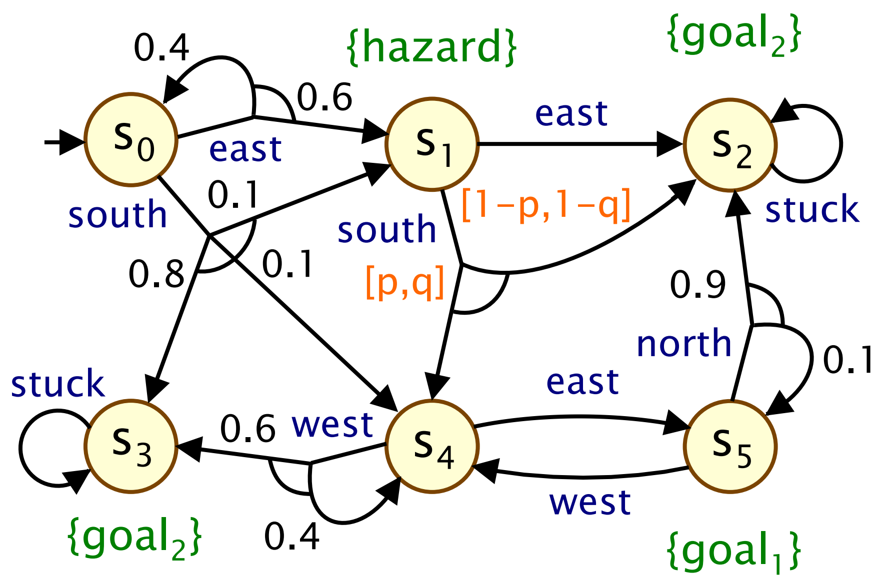
- Partially satisfiable task specifications
  - via multi-objective probabilistic model checking [IJCAI'15]
  - e.g. $P_{max=?}$ [$\neg zone_3$ U ($room_1 \wedge$ (F $room_4 \wedge$ F $room_5$)] < 1
- Synthesise strategies that, in decreasing order of priority:
  - maximise the probability of finishing the task;
  - maximise progress towards completion, if this is not possible;
  - minimise the expected time (or cost) required
- Progress metric constructed from DFA
  - (distance to accepting states, reward for decreasing distance)
- Encode prioritisation using multi-objective queries:
  - $p = P_{max=?}$ [task]
  - r = multi($R^{prog}_{max=?}$ [C], $P_{>=p}$ [task])
  - multi($R^{time}_{min=?}$ [C], $P_{>=p}$ [task] $\wedge$ $R^{prog}_{>=r}$ [C])

Overview

- Probabilistic model checking
  - verification vs. strategy synthesis
  - Markov decision processes (MDPs)
- Application: Robot navigation
  - probabilistic model checking + MDPs + LTL
- Strategy synthesis techniques
  - multi-objective probabilistic model checking
  - partially satisfiable task specifications
  - uncertainty + stochastic games
  - permissive controller synthesis

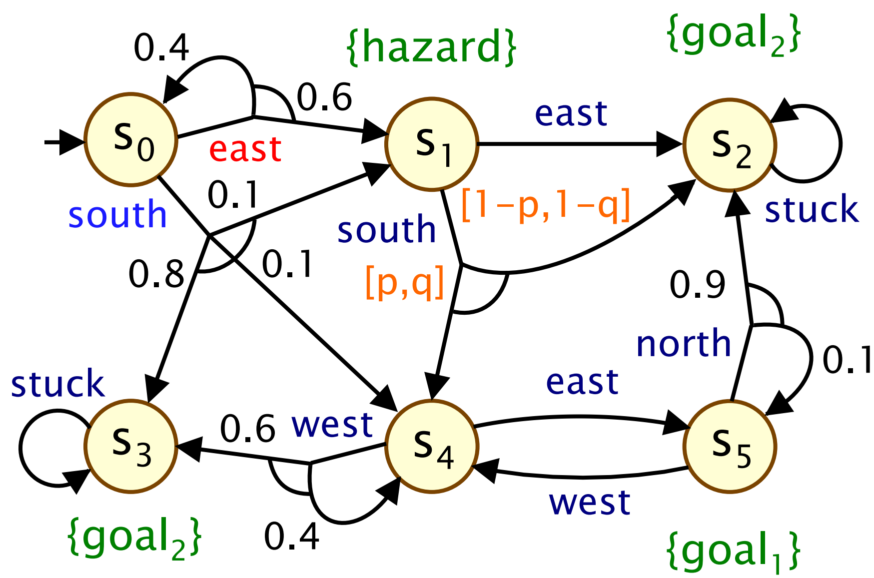
MDPs + uncertainty

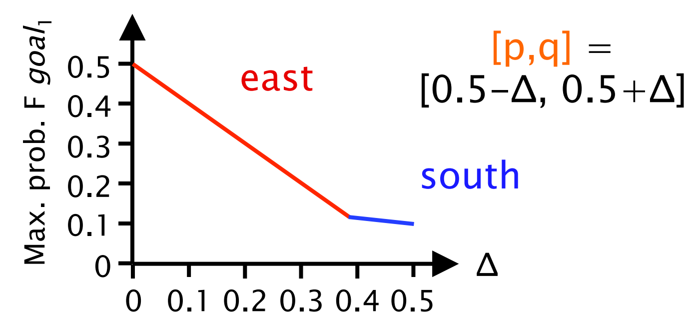
- Modelling uncertainty
  - e.g., transitions probabilities (or costs) specified as intervals
- Worst-case analysis
  - i.e. adversarial choice of probability values
  - stochastic game: controller vs. environment
  - "min-max" analysis



MDPs + uncertainty

- Modelling uncertainty
  - e.g., transitions probabilities (or costs) specified as intervals
- Worst-case analysis
  - i.e. adversarial choice of probability values
  - stochastic game: controller vs. environment
  - "min-max" analysis
- PRISM-games [FMSD'13]
  - stochastic multi-player games
  - temporal logic queries (rPATL)
  - e.g. $\langle\langle ctrl \rangle\rangle\ P_{max=?}\ [\ F\ goal_1\ ]$
  - reduces to solving 2-player game

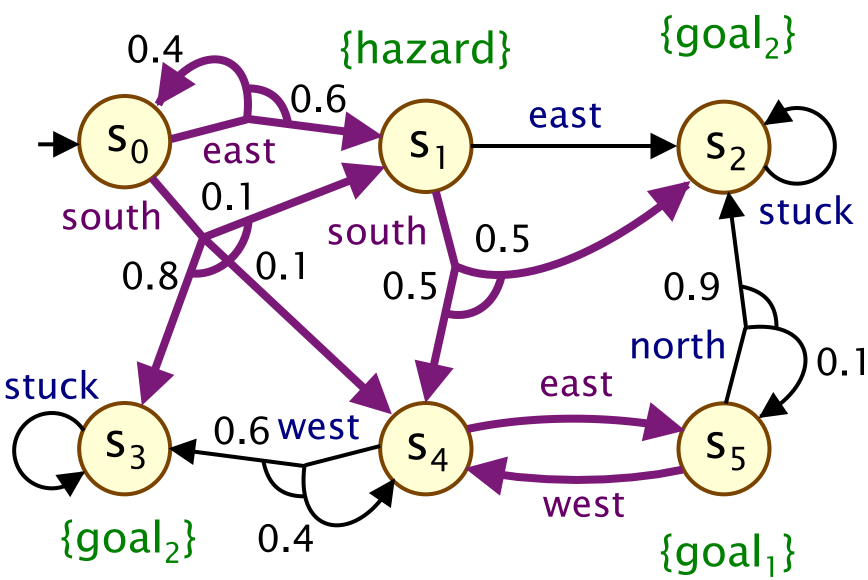




Permissive controller synthesis

- Multi-strategy synthesis [TACAS'14]
  - for Markov decision processes and stochastic games
  - choose sets of actions to take in each state
  - controller is free to choose any action at runtime
  - flexible/robust (e.g. actions become unavailable or goals change)

- Example



Multi-strategy:

s_0 : east or south

s_1 : south

s_2 : –

s_3 : –

s_4 : east

s_5 : west

Permissive controller synthesis

- Multi-strategies and temporal logic
  - multi-strategy Θ satisfies a property $P_{>p}$ [F goal] iff any strategy σ that adheres to Θ satisfies $P_{>p}$ [F goal]
- We quantify the permissivity of multi-strategies
  - by assigning penalties to each action in each state
  - a multi-strategy is penalised for every action it blocks
  - static and dynamic (expected) penalty schemes
- Permissive controller synthesis
  - ∃ a multi-strategy satisfying $P_{\leq 0.6}$ [F $goal_1$] with penalty $< c$?
  - what is the multi-strategy with optimum permissivity?
  - reduction to mixed-integer LP problems
  - other applications: energy management, cloud scheduling, …

Conclusion

- Probabilistic model checking & strategy synthesis
  - Markov decision processes, temporal logic, PRISM
- Robot navigation using MDPs & LTL
  - PRISM extension embedded in ROS navigation stack
- Recent extensions
  - multi-objective probabilistic model checking
  - uncertainty & stochastic games, permissive controller synthesis
- Challenges & directions
  - partial information/observability, e.g. POMDPs
  - probabilistic models with continuous time (or space)
  - scalability, e.g. symbolic methods, abstraction

www.prismmodelchecker.org