20. Probabilistic Model Checking

Computer-Aided Verification

Dave Parker

University of Birmingham

2017/18

Reminders & updates

- No lectures next week
- Assessment 4 (SPIN)
  - due 12 noon Thur 22 Mar
  - help: Facebook, email, office hours, …
- Exam & revision
  - revision lecture at start of summer term
  - see message next week about content/resources

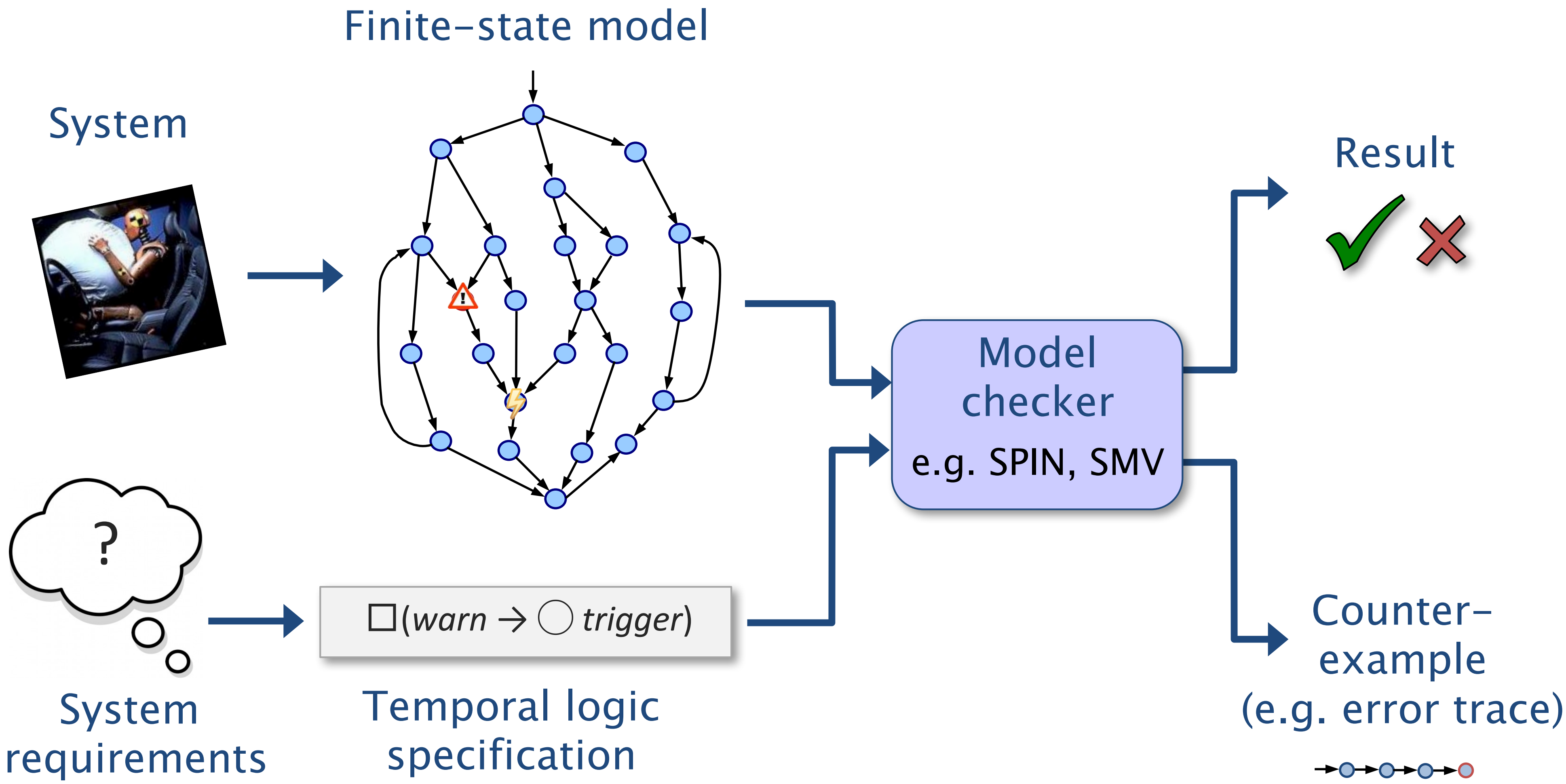
Module syllabus

- Modelling sequential and parallel systems
  - labelled transitions systems, parallel composition
- Temporal logic
  - LTL, CTL and CTL*, etc.
- Model checking
  - CTL model checking algorithms
  - automata-theoretic model checking (LTL)
- Verification tools: SPIN
- Advanced verification techniques
  - bounded model checking via propositional satisfiability
  - symbolic model checking
  - probabilistic model checking

Overview

- Quantitative verification
  - motivation
  - application areas
- Probabilistic model checking
  - discrete-time Markov chains (DTMCs)
  - probabilistic temporal logic (PCTL)
- Background reading:
  - "Quantitative Verification: Formal Guarantees for Timeliness, Reliability and Performance"
  - PRISM: http://www.prismmodelchecker.org/
  - [BK08] Chapter 10

Verification via model checking



Motivation

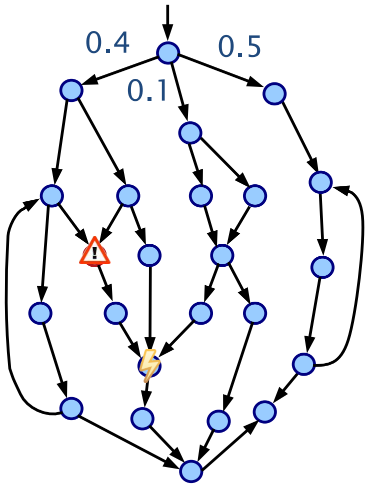
- Verifying probabilistic systems…
  - unreliable or unpredictable behaviour
    - failures of physical components
    - unreliable sensors/actuators
    - message loss in wireless communication
  - randomisation in algorithms/protocols
    - random back-off in communication protocols
    - random routing to reduce flooding or provide anonymity

- We need to verify quantitative system properties
  - "the probability of the airbag failing to deploy within 0.02 seconds of being triggered is at most 0.001"
  - "with probability 0.99, the packet arrives within 10 ms"

Probabilistic model checking

System

Probabilistic model
e.g. Markov chain



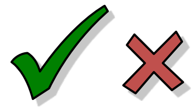
System requirements

$warn \rightarrow P_{>0.9} [\bigcirc\ trigger]$

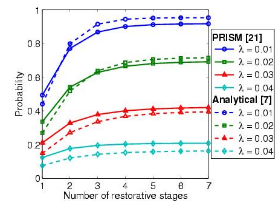
Probabilistic temporal logic specification
e.g. PCTL, CSL, LTL

Probabilistic model checker
e.g. PRISM

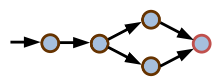
Result



Quantitative results



Counter-example



Probabilistic model checking

- Construction and analysis of finite probabilistic models
  - e.g. Markov chains, Markov decision processes, …
  - specified in high-level modelling formalisms
  - exhaustive model exploration (all possible states/executions)

- Automated analysis of wide range of quantitative properties
  - properties specified using temporal logic
  - “exact” results obtained via numerical computation
  - linear equation systems, iterative methods, uniformisation, …
  - as opposed to, for example, Monte Carlo simulations
  - efficient techniques from verification + performance analysis
  - mature tool support available, e.g. PRISM

Case studies

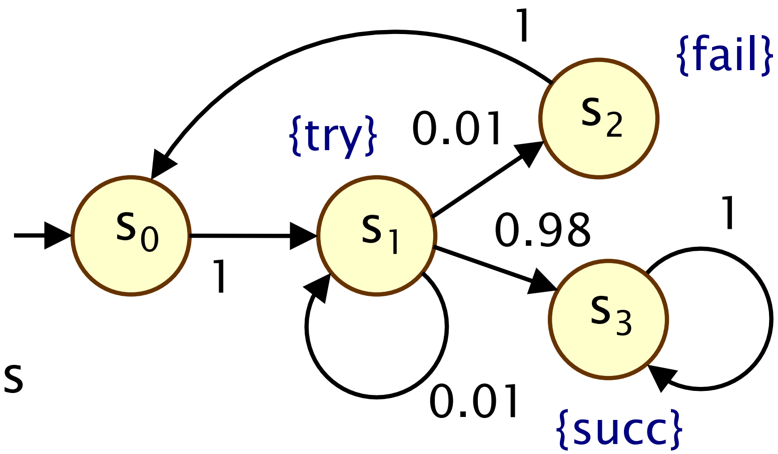
- Randomised communication protocols
  - Bluetooth, FireWire, Zeroconf, 802.11, Zigbee, gossiping, …
- Security protocols/systems
  - pin cracking, anonymity, quantum crypto, contract signing, …
- Performance & reliability
  - airbag controller, nanotechnology, cloud computing, …
- Planning & controller synthesis
  - robotics, autonomous driving, dynamic power management, …

- And many more
  - cell signalling pathways, DNA computing, randomised algorithms
  - see: www.prismmodelchecker.org/casestudies

Discrete-time Markov chains

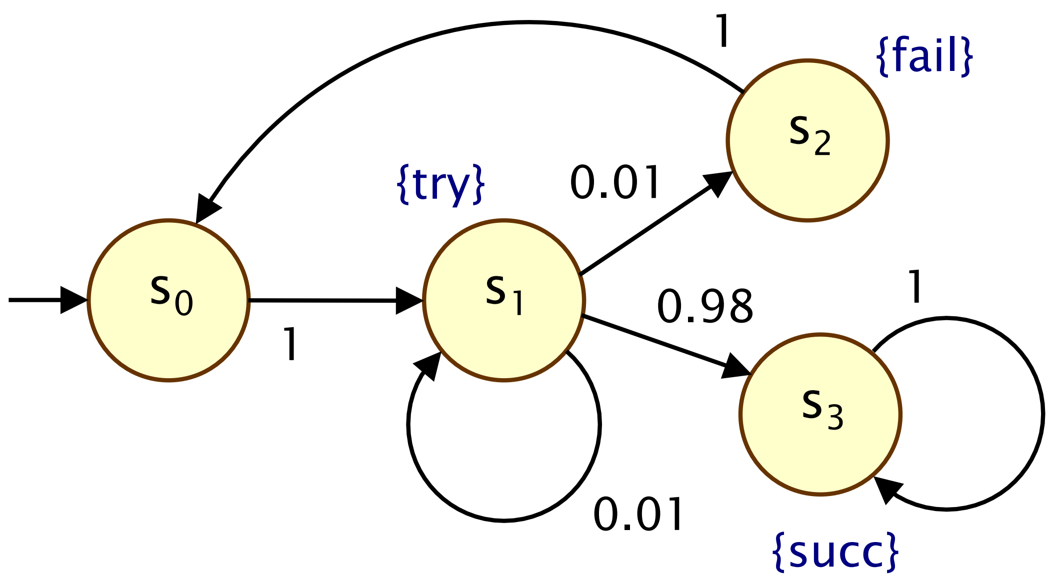
- Discrete-time Markov chains (DTMCs)
  - labelled transition systems augmented with probabilities

- States
  - set of states representing possible configurations of the system being modelled
- Transitions
  - transitions between states model evolution of systems state; occur in discrete time-steps
- Probabilities
  - probabilities of making transitions between states are given by discrete probability distributions

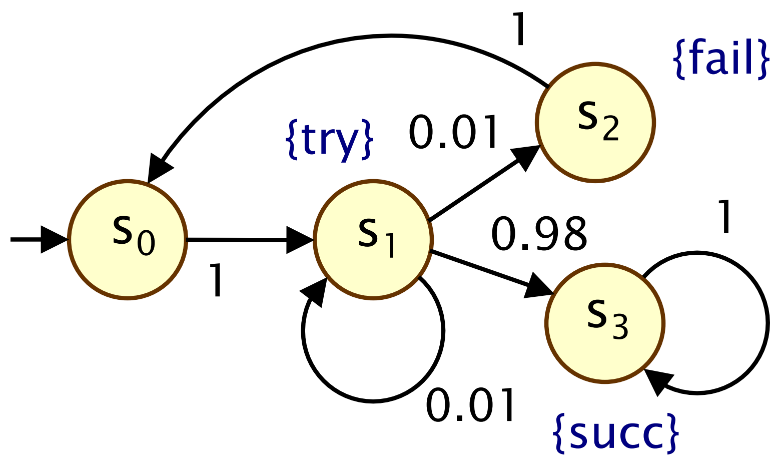


Simple DTMC example

- Modelling a very simple communication protocol
  - after one step, process starts trying to send a message
  - with probability 0.01, channel unready so wait a step
  - with probability 0.98, send message successfully and stop
  - with probability 0.01, message sending fails, restart



Discrete-time Markov chains



- Formally, a DTMC D is
  - a tuple (S, s_{init}, P, L)
- where:
  - S is a set of states ("state space")
  - $s_{init} \in S$ is the initial state
  - $P : S \times S \rightarrow [0,1]$ is the transition probability matrix
    - where $\Sigma_{s' \in S} P(s,s') = 1$ for all $s \in S$
  - AP is a set of atomic propositions
  - $L : S \rightarrow 2^{AP}$ is a labelling function
- Transition probabilities
  - $P(s,s')$ gives the probability of moving from s to s'

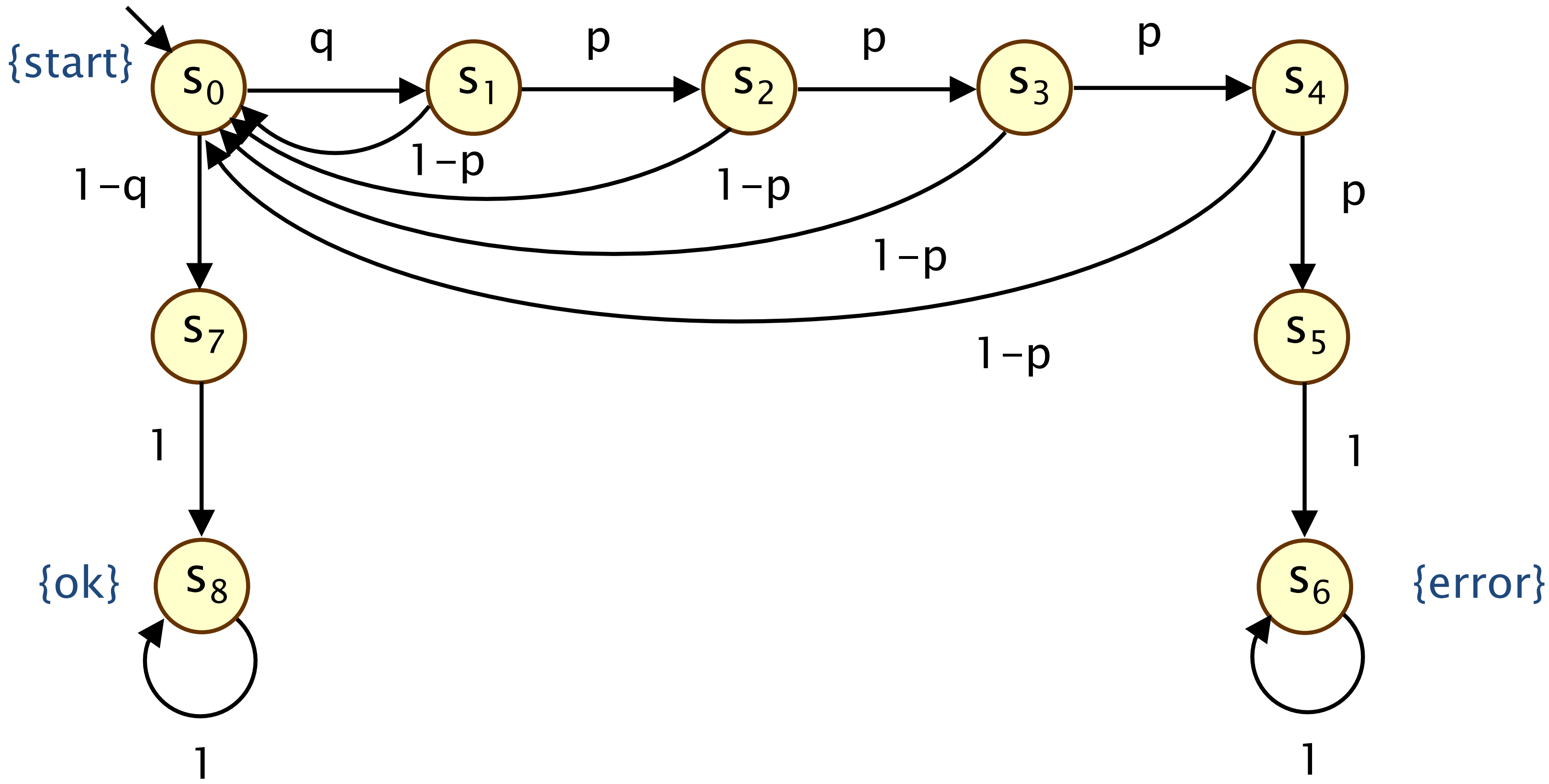
DTMC example – Zeroconf

- Zeroconf = “Zero configuration networking”
  - self-configuration for local, ad-hoc networks
  - automatic configuration of unique IP for new devices
  - simple; no DHCP, DNS, …

- Basic idea:
  - 65,024 available IP addresses (IANA-specified range)
  - new node picks address U at random
  - broadcasts “probe” messages: “Who is using U?”
  - any node already using U replies; protocol restarts
  - messages may not get sent (transmission fails, host busy, …)
  - so: nodes send multiple (n) probes, waiting after each one

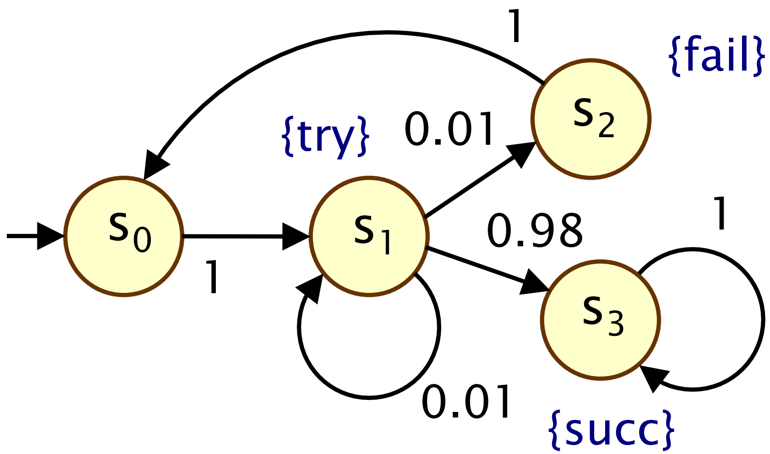
DTMC for Zeroconf

- $n=4$ probes, m existing nodes in network
- probability of message loss: p
- probability that new address is in use: $q = m/65024$



Paths in DTMCs

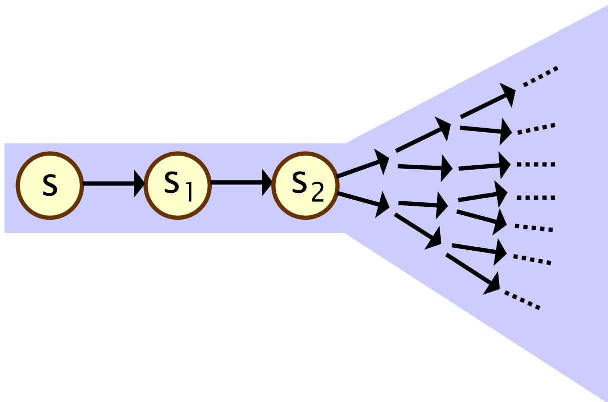
- A (finite or infinite) path through a DTMC
  - is a sequence of states $s_0s_1s_2s_3\ldots$ such that $\mathbf{P}(s_i,s_{i+1}) > 0\ \forall i$
  - represents an execution (i.e. one possible behaviour) of the system which the DTMC is modelling
  - Paths(s) is the set of all (infinite) paths starting in s



- Examples:
  - never succeeds: $(s_0s_1s_2)^\omega$
  - tries, waits, fails, retries, succeeds: $s_0s_1s_1s_2s_0s_1(s_3)^\omega$

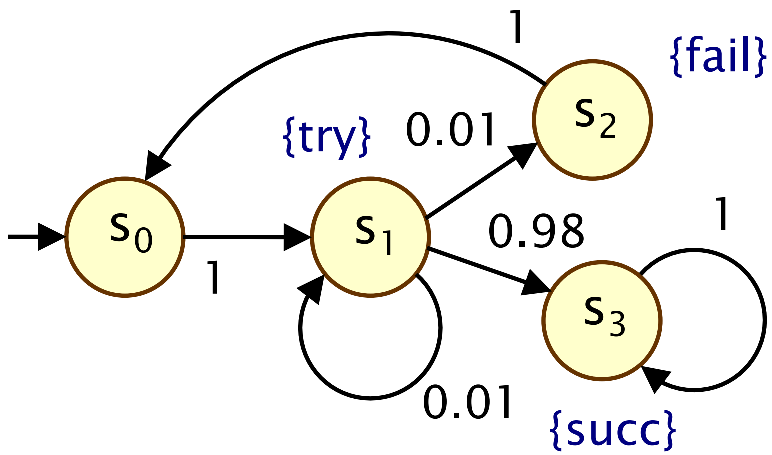
Paths and probabilities

- To reason (quantitatively) about this system
  - need to define a probability measure over paths
- More precisely:
  - probability measure Pr_s over Paths(s)
  - basic idea: defined on finite paths, extended to infinite paths
  - $\mathbf{P}(ss_1s_2) = \mathbf{P}(s,s_1)\mathbf{P}(s_1,s_2)$



Paths and probabilities

- Examples



- "try and fail immediately"
  - paths starting with prefix $s_0 s_1 s_2$
  - probability : $\mathbf{P}(s_0 s_1 s_2)$

    $= \mathbf{P}(s_0,s_1)\mathbf{P}(s_1,s_2) = 1 \cdot 0.01 = 0.01$

- "eventually successful and with no failures"
  - paths $s_0 s_1 s_3 \ldots$, $s_0 s_1 s_1 s_3 \ldots$, $s_0 s_1 s_1 s_1 s_3 \ldots$, …
  - probability:

    $= \mathbf{P}_{s0}(s_0 s_1 s_3) + \mathbf{P}_{s0}(s_0 s_1 s_1 s_3) + \mathbf{P}_{s0}(s_0 s_1 s_1 s_1 s_3) + \ldots$

    $= 1 \cdot 0.98 + 1 \cdot 0.01 \cdot 0.98 + 1 \cdot 0.01 \cdot 0.01 \cdot 0.98 + \ldots$

    $= 0.9898989898\ldots$

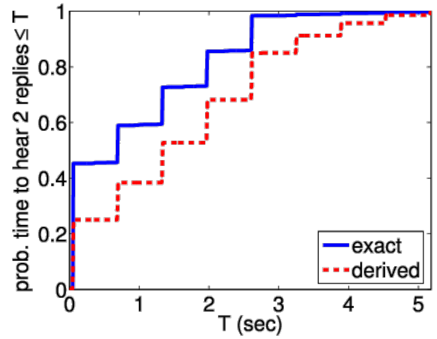
    $= 98/99$

In practice, computed by solving linear equation systems

Case study: Bluetooth

- Device discovery between a pair of Bluetooth devices
  - performance essential for this phase

- Complex discovery process
  - two asynchronous 28-bit clocks
  - pseudo-random hopping between 32 frequencies
  - random waiting scheme to avoid collisions
  - 17,179,869,184 initial configurations



- Probabilistic model checking (PRISM)
  - "probability discovery time exceeds 6s is always < 0.001"
  - "worst-case expected discovery time is at most 5.17s"

PCTL

- Temporal logic for describing properties of DTMCs
  - PCTL = Probabilistic Computation Tree Logic

- Extension of (non-probabilistic) temporal logic CTL
  - key addition is probabilistic operator P
  - quantitative extension of CTL's ∀ and ∃ operators

- Example
  - $\text{send} \rightarrow P_{\geq 0.95} [\ \Diamond^{\leq 10}\ \text{deliver}\ ]$
  - "if a message is sent, then the probability of it being delivered within 10 steps is at least 0.95"

CTL syntax

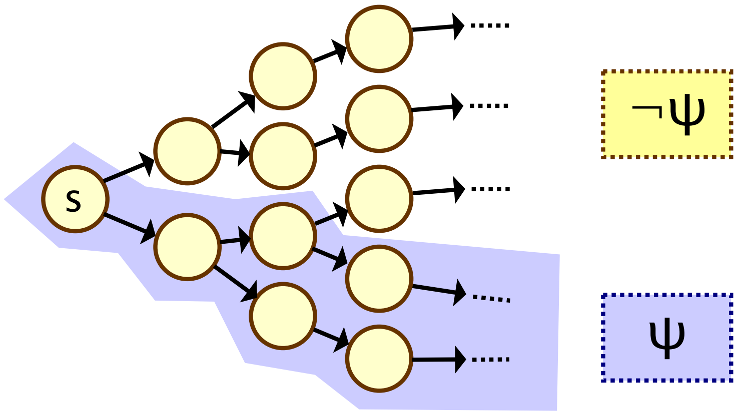
- Syntax split into state and path formulae
  - specify properties of states/paths, respectively
  - a CTL formula is a state formula ϕ
- State formulae:
  - $\phi \ ::= \ \text{true} \mid a \mid \phi \wedge \phi \mid \neg\phi \mid \forall\, \psi \mid \exists\, \psi$
  - where $a \in AP$ and ψ is a path formula
- Path formulae
  - $\psi \ ::= \ \bigcirc\, \phi \mid \phi \ U \ \phi \mid \ldots$
  - where ϕ is a state formula

PCTL syntax

- Syntax split into state and path formulae
  - specify properties of states/paths, respectively
  - a PCTL formula is a state formula ϕ
- State formulae:
  - $\phi ::= \text{true} \mid a \mid \phi \wedge \phi \mid \neg\phi \mid P_{\sim p} [\psi]$
  - where $a \in AP$ and ψ is a path formula, $p \in [0,1]$ is a probability bound, $\sim \in \{<,>,\leq,\geq\}$
- Path formulae
  - $\psi ::= \bigcirc \phi \mid \phi \ U \ \phi \mid \phi \ U^{\leq k} \ \phi \mid \ldots$
  - where ϕ is a state formula, $k \in \mathbb{N}$

PCTL semantics for DTMCs

- Semantics of the probabilistic operator P
  - example: $s \models P_{<0.25}$ [◯ fail] $\Leftrightarrow$ "the probability of atomic proposition fail being true in the next state of outgoing paths from s is less than 0.25"
  - informal definition: $s \models P_{\sim p}$ [ψ] means that "the probability, from state s, that ψ is true for an outgoing path satisfies $\sim p$"
  - formally: $s \models P_{\sim p} [\psi] \Leftrightarrow Pr_s \{\pi \in Path(s) \mid \pi \models \psi \} \sim p$



$Pr_s(\psi) \sim p$?

PCTL examples

- $P_{\leq 0.05}$ [$\Diamond$ err/total>0.1]
  - "with probability at most 0.05, more than 10% of the NAND gate outputs are erroneous"
- $P_{\geq 0.8}$ [$\Diamond^{\leq k}$ reply_count=n]
  - "the probability that the sender has received n acknowledgements within k clock-ticks is at least 0.8"
- $P_{<0.4}$ [$\neg fail_A$ U $fail_B$]
  - "the probability that component B fails before component A is less than 0.4"
- $\neg$oper $\rightarrow$ $P_{\geq 1}$ [$\Diamond$ ($P_{>0.99}$ [$\Box^{\leq 100}$ oper])]
  - "if the system is not operational, it almost surely reaches a state from which it has a greater than 0.99 chance of staying operational for 100 time units"

Qualitative vs. quantitative properties

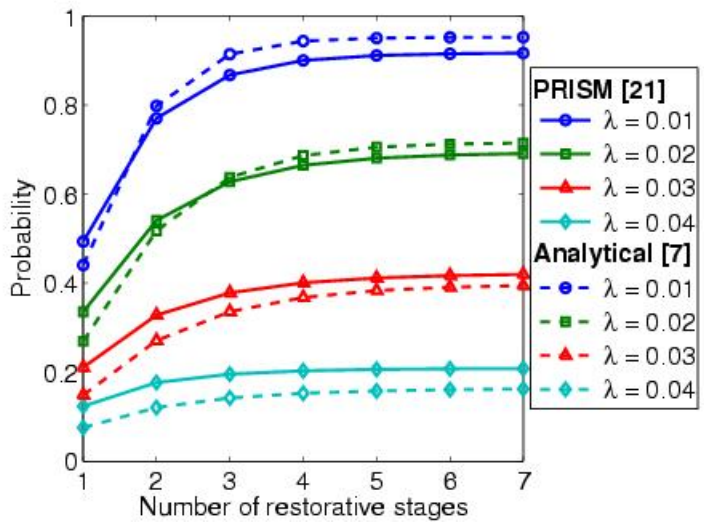
- P operator of PCTL can be seen as a quantitative analogue of the CTL operators $\forall$ (for all) and $\exists$ (there exists)

- Qualitative PCTL properties
  - $P_{\sim p}$ [ψ] where p is either 0 or 1
- Quantitative PCTL properties
  - $P_{\sim p}$ [ψ] where p is in the range (0,1)

- $P_{>0}$ [$\Diamond\phi$] is identical to $\exists\Diamond\phi$
  - there exists a finite path to a ϕ-state
- $P_{\geq 1}$ [$\Diamond\phi$] is (similar to but) weaker than $\forall\Diamond\ \phi$
  - a ϕ-state is reached "almost surely"

Numerical properties

- Consider a PCTL formula $P_{\sim p}$ [ψ]
  - if the probability is unknown, how to choose the bound p?
- When the outermost operator of a PTCL formula is P
  - PRISM allows formulae of the form $P_{=?}$ [ψ]
  - "what is the probability that path formula ψ is true?"
- Model checking is no harder: compute the values anyway
- Useful to spot patterns, trends
- Example
  - $P_{=?}$ [◇ err/total>0.1]
  - "what is the probability that 10% of the NAND gate outputs are erroneous?"

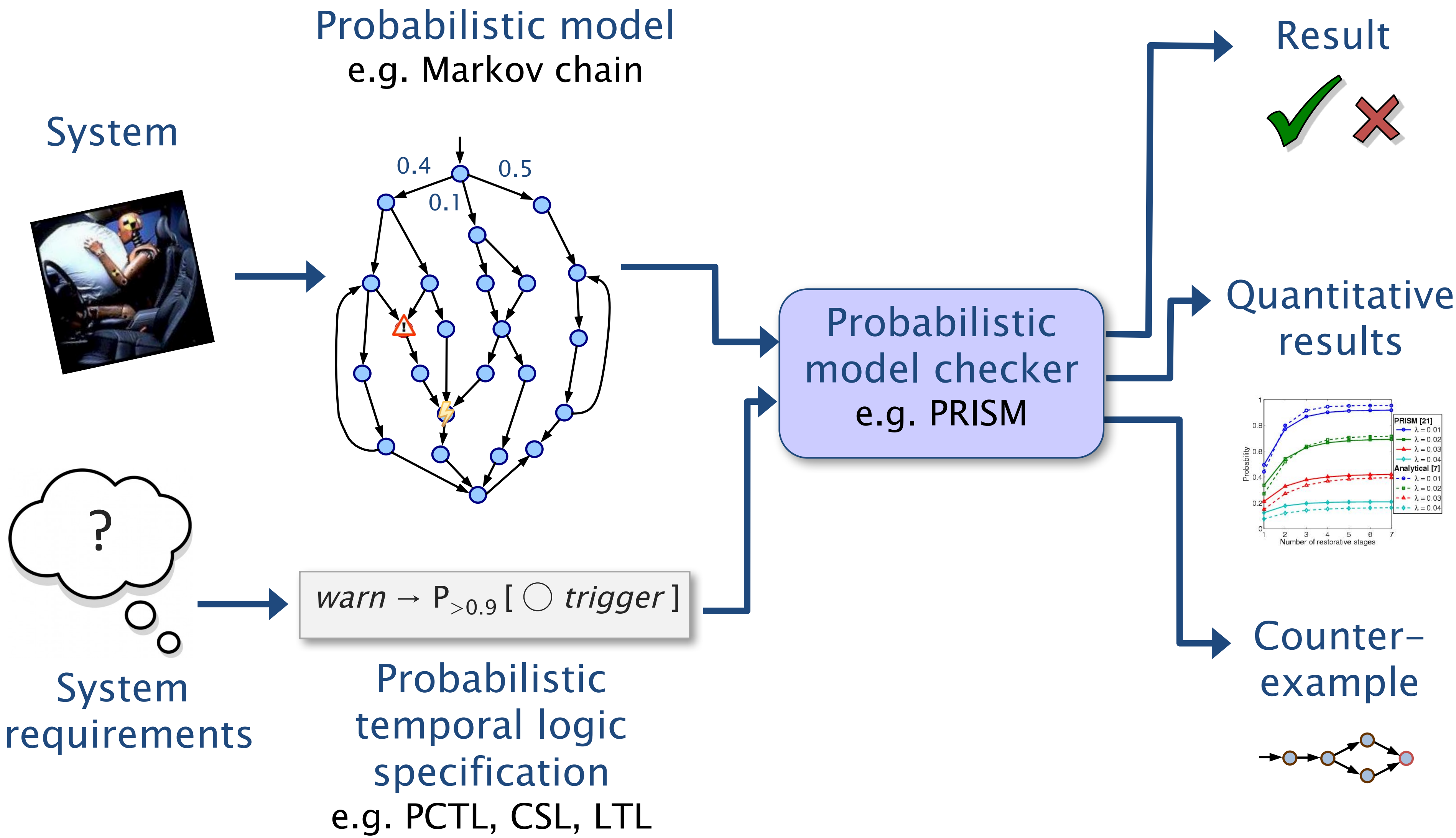


Probabilistic model checking

- More specification formalisms
  - probabilistic LTL
  - e.g. $P_{=?}$ ($\Box\Diamond$ send): "what is the probability that the protocol successfully sends a message infinitely often?"
  - e.g. $P_{=?}$ ($\neg zone_3$ U ($zone_1 \wedge (\Diamond\ zone_4$))): " what is the probability of visiting zone 1, without passing through zone 3, and then going to zone 4?"
  - PCTL* (subsumes PCTL and probabilistic LTL)
  - costs, rewards, …

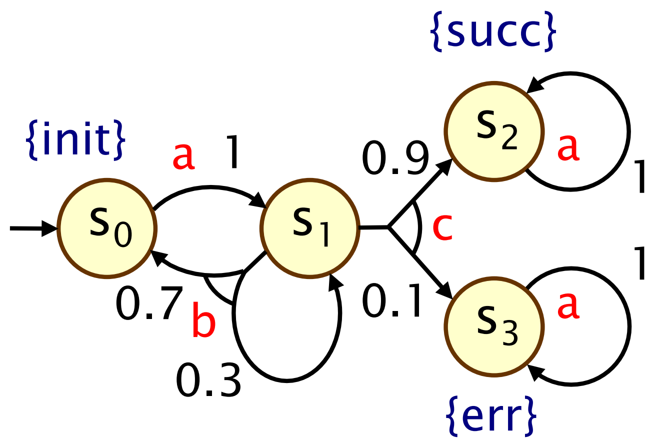
- More probabilistic models
  - continuous-time Markov chains
    - adds a notion of real (not discrete) time
  - Markov decision processes…
    - adds nondeterminism

Probabilistic model checking



Markov decision processes (MDPs)

- Markov decision processes (MDPs)
  - model nondeterministic as well as probabilistic behaviour
  - widely used also in: AI, planning, optimal control, …



- Nondeterminism for:
  - control: decisions made by a controller or scheduler
  - adversarial behaviour of the environment
  - concurrency/scheduling: interleavings of parallel components
  - abstraction, or under-specification, of unknown behaviour

Summary

- Quantitative verification
  - reasoning about probability, time, …
  - unreliable or unpredictable behaviour, randomisation
  - quantitative "correctness": reliability, timeliness, performance, …
- Probabilistic model checking
  - discrete-time Markov chains (DTMCs)
  - paths, probability measures
  - probabilistic temporal logic (PCTL)
- PRISM
  - http://www.prismmodelchecker.org/