Probabilistic Model Checking of Randomised Distributed Protocols using PRISM

Marta Kwiatkowska

University of Birmingham

VPSM PhD School, Copenhagen, October 2006

Tutorial overview

- Part I - Probabilistic Model Checking
  - Discrete-time Markov chains, Markov decision processes, temporal logic (PCTL), model checking algorithms, probabilistic timed automata
- Part II - Tool Support: PRISM
  - Tools, PRISM: functionality, modelling language, property specifications, tool demo, implementation
- Part III - Case Studies
  - Overview, device discovery in Bluetooth, FireWire root contention, contract signing protocols, Zeroconf protocol

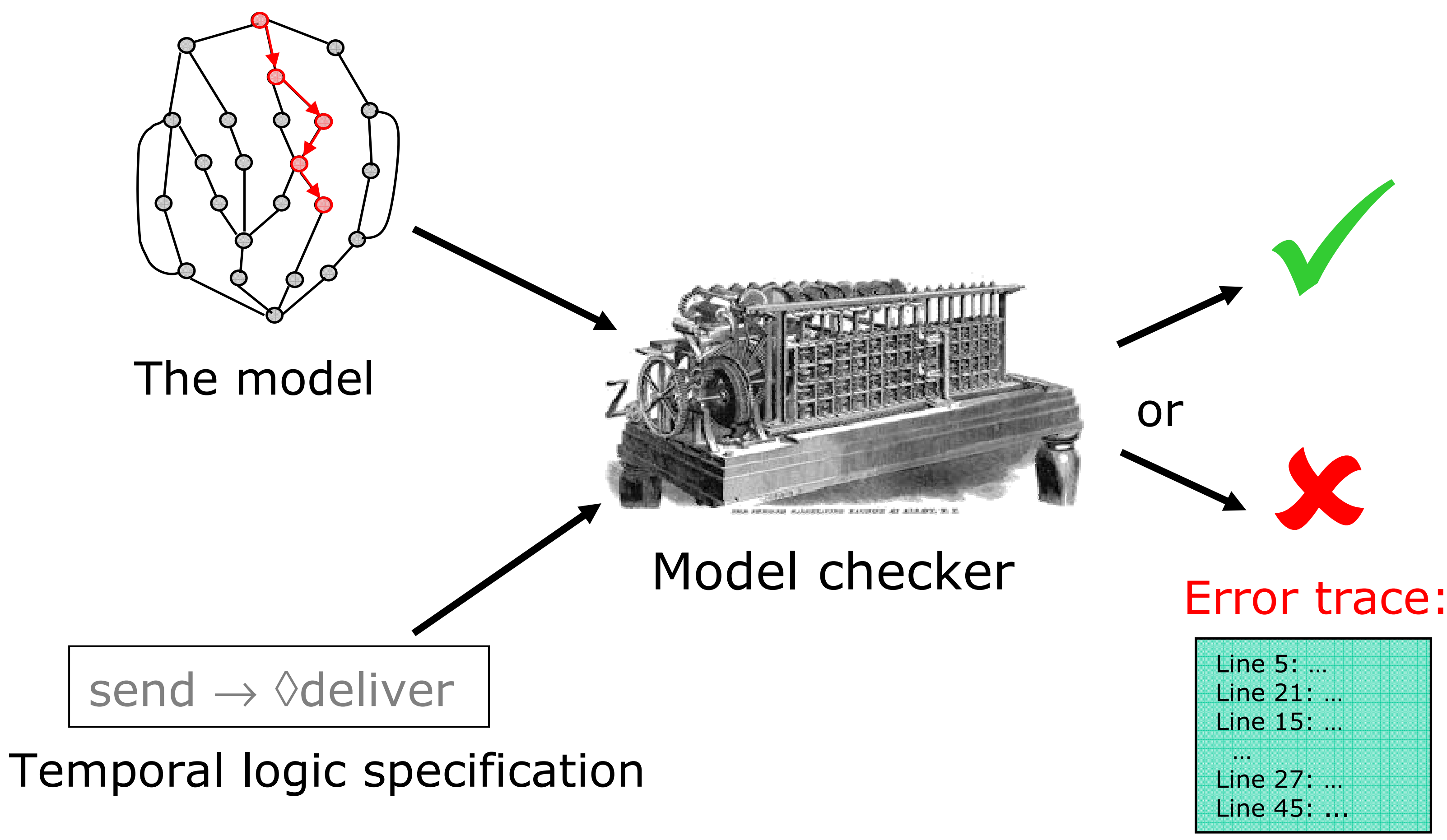
Part I

Probabilistic Model Checking

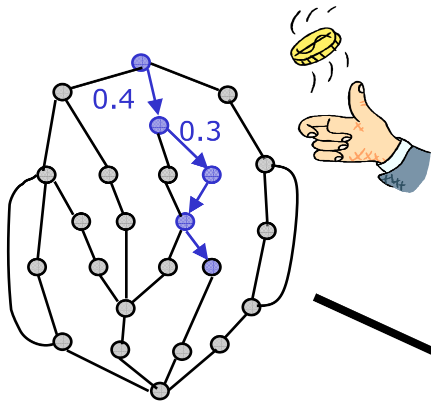
Overview

- What is probabilistic model checking?
- Motivation: Why probability?
- Discrete-time probabilistic models
  - discrete-time Markov chains (DTMCs)
  - Markov decision processes (MDPs)
  - the logic PCTL + costs/rewards
  - model checking for DTMCs, MDPs
- Real-time probabilistic models
  - probabilistic timed automata (PTAs)
  - model checking for PTAs

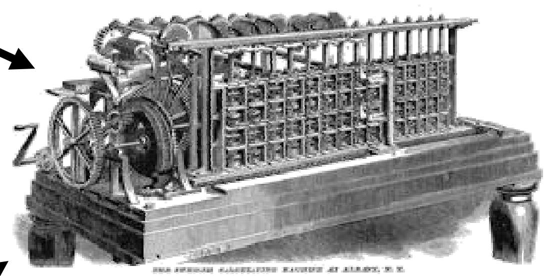
Verification via model checking



Probabilistic model checking

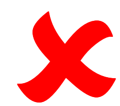


Probabilistic model



Probabilistic
model checker

or



or

The probability

State 5: 0.6789
State 6: 0.9789
State 7: 1.0
…
State 12: 0
State 13:
0.1245

send → P>0.9 [◊deliver]

Probabilistic temporal
logic specification

Motivation - Why probability?

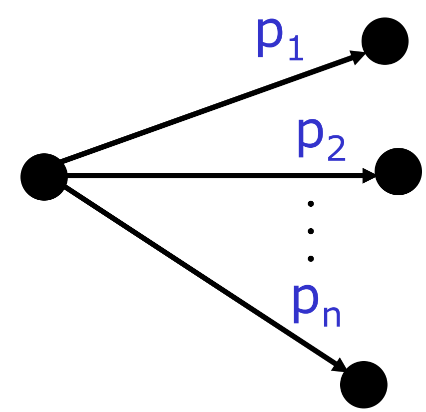
- In distributed co-ordination algorithms
  - Elegant and efficient algorithms for symmetry breaking
    - "leader election is eventually resolved with probability 1"
  - In gossip-based routing and multicasting
    - "the message will be delivered to all nodes with high probability"
- When modelling uncertainty in the environment
  - To quantify failures, express soft deadlines, QoS
    - "the chance of shutdown is at most 0.1%"
    - "the probability of a frame being delivered within 5ms is at least 0.95"
  - To quantify environmental factors in decision support
    - "the expected cost of reaching the goal is 100"
- When analyzing system performance
  - To quantify arrivals, service, etc, characteristics
    - "in the long run, mean waiting time in a lift queue is 30 sec"

Application domains

- Communication protocols, ubiquitous computing
  - e.g. Bluetooth, FireWire, WiFi, …
- Security protocols
  - e.g. anonymity, contract signing, PIN cracking, …
- And many others:
  - e.g. computational biology models,
    dynamic power management systems,
    randomized distributed algorithms, …
- More in Part III…

Probabilistic models - Discrete time

- Labelled transition systems
  - discrete time-steps
  - labelling with atomic propositions

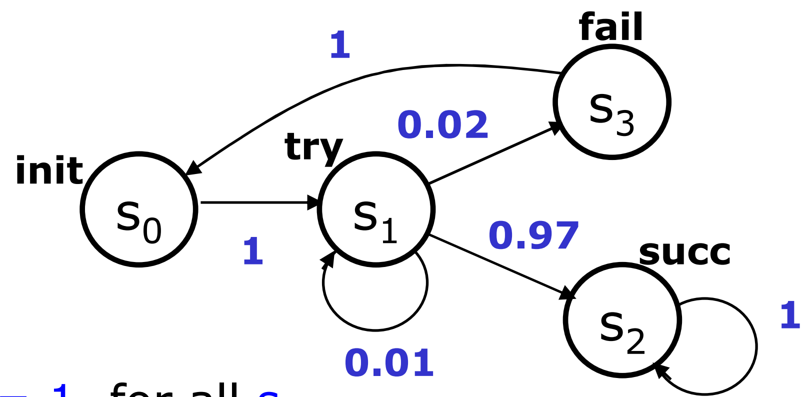


$\sum_i p_i = 1$

- Probabilistic transitions
  - move to state with given probability
  - represented as a discrete probability distribution

- Model types:
  - discrete time Markov chains (DTMCs): probability only
  - Markov decision processes (MDPs): probability + nondeterminism

Discrete-time Markov chains (DTMCs)



- Formally, $(S, s_0, \mathbf{P}, L)$:
  - S finite set of states
  - s_0 initial state
  - $\mathbf{P} : S \times S \rightarrow [0,1]$
    probability matrix, s.t. $\sum_{s'} \mathbf{P}(s,s') = 1$, for all s
  - $L : S \rightarrow 2^{AP}$ labelling with atomic propositions

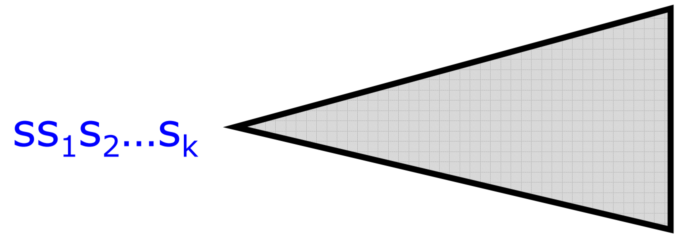
- Unfold into infinite paths $s_0 s_1 s_2 s_3 s_4 \ldots$ s.t. $\mathbf{P}(s_i, s_{i+1}) > 0$, for all i

- Probability for finite paths, multiply along path

  e.g. $P(s_0\, s_1\, s_1\, s_2)$ is $1 \cdot 0.01 \cdot 0.97 = 0.0097$

Probability space

- Intuitively:
  - Sample space = infinite set of paths $Path_s$ from a state s
  - Event = set of paths
  - Basic event = cone



- Formally, $(Path_s,\Omega,Pr_s)$ [KSK76]
  - For finite path $\omega = ss_1...s_n$, define probability: $P(\omega) = ...$
    - 1 if ω has length one
    - $\mathbf{P}(s,s_1) \cdot ... \cdot \mathbf{P}(s_{n-1},s_n)$ otherwise
  - Take Ω as the least σ-algebra containing cones
    - $C(\omega) = \{ \pi \in Path_s \mid \omega \text{ is prefix of } \pi\}$
  - Define $Pr_s(C(\omega)) = P(\omega)$, for all ω
  - Pr_s extends uniquely to measure on $Path_s$

Markov decision processes (MDPs)

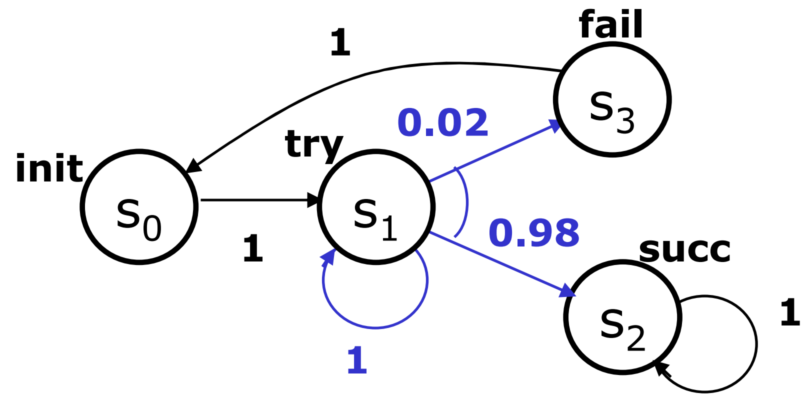
- Generalisation of DTMCs
  - incorporate both probabilistic and nondeterministic choice

- Motivation – many uses in probabilistic modelling
  - Concurrency - parallel composition of DTMCs

    e.g. communication protocols, randomised algorithms, ...
  - Under-specification - some behaviour/parameters unknown
  - Unknown environment - e.g. probabilistic security protocols

Markov decision processes (MDPs)

- Formally, $(S, s_0, Steps, L)$:
  - S finite set of states
  - s_0 initial state
  - Steps maps states s to sets of probability distributions μ over S
  - $L: S \rightarrow 2^{AP}$ atomic propositions



- Unfold into infinite paths $s_0\mu_0 s_1\, \mu_1 s_2 \mu_2 s_3 \ldots$ s.t. $\mu_i(s_i, s_{i+1}) > 0$, all i

- Probability space induced on $Path_s$ by adversary (strategy, policy)
  - resolves all nondeterminism
  - mapping from finite paths $s_0\mu_0 s_1 \mu_1 \ldots s_n$ to a distribution from state s_n

Properties of DTMCs and MDPs: PCTL

- PCTL: Probabilistic Computation Tree Logic [HJ94,BdA95]
  - extension of (non-probabilistic) temporal logic CTL
  - new probabilistic operator, e.g. send $\rightarrow$ P>0.9 [F deliver]
  - "if a message is sent, probability eventually delivered is >0.9"
- Syntax:
  - ϕ ::= true | a | $\phi \wedge \phi$ | $\neg\phi$ | P~p [α] (state formulas)
  - α ::= X ϕ | ϕ U ϕ (path formulas)
  - where a is an atomic proposition, $p \in [0,1]$, $\sim \in \{<,>,\leq,\geq\}$
- Also:
  - "bounded until" (ϕ U≤k ϕ), "eventually" (F ϕ = true U ϕ)
  - "quantitative form" P=? [α] (more in Part II)

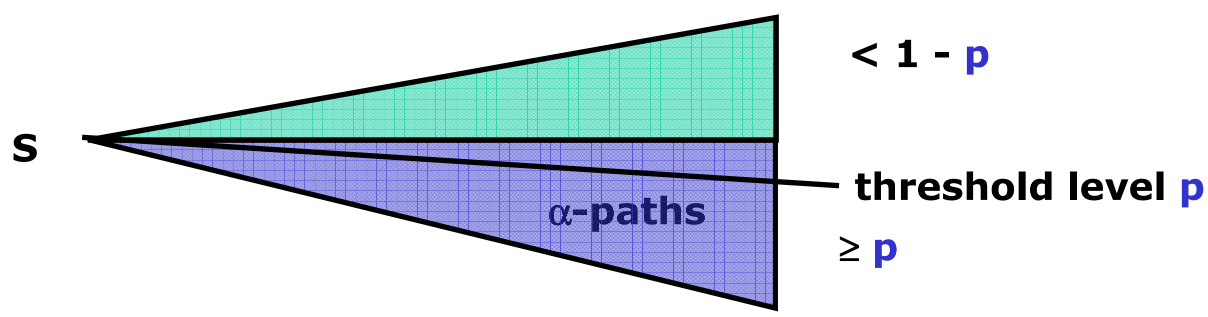
PCTL - Semantics for DTMCs

- Semantics of (non-probabilistic) state formulas:
  - for a state s of the DTMC:
  - $s \models a \quad \Leftrightarrow \quad a \in L(s)$
  - $s \models \phi_1 \wedge \phi_2 \quad \Leftrightarrow \quad s \models \phi_1$ and $s \models \phi_2$
  - $s \models \neg\phi \quad \Leftrightarrow \quad s \models \phi$ is false

- Semantics of path formulas:
  - for a path $\pi = s_0 s_1 s_2 \cdots$ in the DTMC
  - $\pi \models X\ \phi \quad \Leftrightarrow \quad s_1 \models \phi$ (“next”)
  - $\pi \models \phi_1\ U\ \phi_2 \quad \Leftrightarrow \quad \exists k$ s.t. $s_k \models \phi_2$ and $s_j \models \phi_1$ for all $j<k$ (“until”)

PCTL – Semantics for DTMCs

- Semantics of the probabilistic operator P
  - quantitative analogue of $\forall, \exists$
  - $s \models P_{\sim p}\,[\alpha] \;\Leftrightarrow\; Pr_s\{\, \pi \in Path_s \mid \pi \models \alpha \,\} \sim p$



  - subsumes the qualitative variants $P_{=1}\,[\alpha]$, $P_{>0}\,[\alpha]$

PCTL – Semantics for MDPs

- Semantics is parameterised by a class of adversaries Adv
  - e.g. Adv is "all adversaries" or "all fair adversaries"
  - reasoning about worst-case/best-case scenario

- Non-probabilistic state formulas, path formulas – as before

- The probabilistic operator:
  - $s \vDash_{Adv} P_{\sim p} [\alpha] \Leftrightarrow Pr_s^A \{ \pi \in Path_s \mid \pi \vDash_{Adv} \alpha \} \sim p \ \forall A \in Adv$
  - "probability meets the bound $\sim p$ for all adversaries in Adv"
  - Pr^A_s = probability measure for adversary A over paths $Path_s$

Costs and Rewards

- Augment DTMC/MDP with reward structure: (**r**,**R**)
  - vector **r** of state rewards, matrix **R** matrix of transition rewards
- Analysis of reward-based properties
  - instantaneous, e.g. "queue size", "number of active hosts", …
  - cumulative, e.g. "power consumed", "number of messages lost", …
- Extend PCTL with rewards:
  - R~r [I=T] : expected reward at time T is ~r
  - R~r [F ϕ] : expected reward to reach a state satisfying ϕ is ~r
  - R~r [C≤T] : expected reward accumulated by time T is ~ r

PCTL model checking for DTMCs

- Compute $Sat(\phi)$, i.e. set of states satisfying formula ϕ, by induction on structure of ϕ (like for CTL)
- For the non-probabilistic operators:

  $Sat(a) = L(a)$, $Sat(\neg\phi) = S \backslash Sat(\phi)$, $Sat(\phi_1 \wedge \phi_2) = Sat(\phi_1) \cap Sat(\phi_2)$

- For the probabilistic operator:

  $Sat(P{\sim}p[\alpha]) = \{s \in S \mid Pr_s(\alpha) \sim p\}$

  where $Pr_s(\alpha) = Pr_s\{\pi \in Path_s \mid \pi \vDash \alpha\}$

- Computation of probabilities $Pr_s(\alpha)$
  - next operator: $Pr_s(X\ \phi) = \sum_{s' \in \mathbf{Sat}(\phi)} \mathbf{P}(s,s')$
  - until operator: $Pr_s(\phi_1\ U\ \phi_2)$ from solution of linear equation system
- (computation of costs/rewards for $R{\sim}r[F\ \phi]$ similar to until)

PCTL until for DTMCs

- Let $x_s = Pr_s(\phi_1 \text{ U } \phi_2)$ be probabilities for until operator
- $(x_s)_{s \in S}$ can be obtained from the recursive linear equation:
  - $x_s = 0$ if $s \in S^{no}$
  - $x_s = 1$ if $s \in S^{yes}$
  - $x_s = \sum_{s' \in S} \mathbf{P}(s,s') \cdot x_{s'}$ if $s \in S^?$

  where:
  - S^{yes} = states that satisfy $\phi_1 \text{ U } \phi_2$ with probability exactly 1
  - S^{no} = states that satisfy $\phi_1 \text{ U } \phi_2$ with probability exactly 0
  - $S^? = S \backslash (S^{no} \cup S^{yes})$
- S^{yes}, S^{no} can be computed by graph traversal algorithms
  - for qualitative PCTL (e.g. $P{>}0[\phi_1 \text{U} \phi_2]$) no computation needed
- Linear equation systems typically solved with
  - iterative numerical solution algorithms, e.g. Gauss-Seidel

PCTL model checking for MDPs

- As for DTMCs, proceed by induction on structure of formula ϕ
  - and non-probabilistic operators are trivial
- For probabilistic operator, compute min or max values, e.g.:

  $\mathrm{Sat}(P{>}p[\alpha]) = \{ s \in S \mid Pr_s^{min}(\alpha) > p \}$

  where $Pr_s^{min}(\alpha) = \min \{ Pr_s^{A}(\alpha) : A \in Adv \}$
- Probabilities for until: $Pr_s^{min}(\phi_1 U \phi_2)$ or $Pr_s^{max}(\phi_1 U \phi_2)$:
  - (as for DTMCs, combination of graph traversal algorithms and numerical computation algorithms)
  - iterative solution technique, form of Bellman equation
    - also known as “value iteration” (from dynamic programming)
  - or: linear optimisation problems
    - direct solution via e.g. Simplex, Ellipsoid method

PCTL until for MDPs (iterative)

- Iterative solution for min until probabilities (max similar):
- $Pr_s(\phi_1 \text{ U } \phi_2) = \lim n \to \infty\ x_s^{(n)}$ where:
  - $x_s^{(n)} = 0$ if $s \in S^{no}$
  - $x_s^{(n)} = 1$ if $s \in S^{yes}$
  - $x_s^{(n)} = 0$ if $s \in S^{?}$ and $n=0$
  - $x_s^{(n)} = \min_{\mu \in Steps(s)} \sum_{s' \in S} \mu(s') \cdot x_{s'}^{(n-1)}$ if $s \in S^{?}$ and $n>0$

  where:
  - S^{yes} = states satisfying $\phi_1 U \phi_2$ with prob. 1 for all adversaries
  - S^{no} = states satisfying $\phi_1 U \phi_2$ with prob. 0 for some adversary
  - $S^{?} = S \backslash (S^{no} \cup S^{yes})$
- S^{yes}, S^{no} can again be computed by graph traversal algorithms
- (similar formulation to compute costs/rewards for $R{\sim}r[F\ \phi]$)

PCTL until for MDPs (linear optimisation)

- Solution for min/max until probabilities via linear programming
- $x_s = 0$ for $s \in S^{no}$, $x_s = 1$ for $s \in S^{yes}$
- For $s \in S^?$, solve linear optimisation problem:

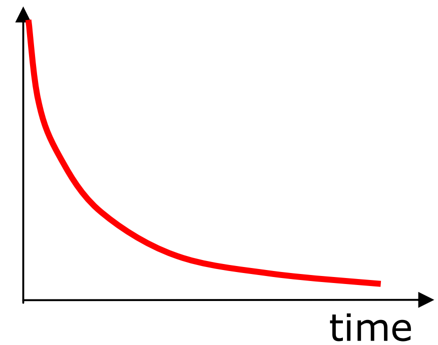
- Minimise $\sum_{s \in S?} x_s$ subject to the constraints:
  - $x_s \leq \sum_{s' \in S?} \mu(s') \cdot x_s + \sum_{s' \in Syes} \mu(s')$
    for all $s \in S^?$ and all $\mu \in Steps(s)$

(above is for min, the max prob.s computed similarly)

(similar formulation to compute costs/rewards for $R{\sim}r[F\ \phi]$)

Probabilistic models – Continuous time

- Assumptions on time and probability
  - Continuous passage of time
  - Continuous randomly distributed delays



$$\int f(x)\,dx = 1$$

- Model types
  - Probabilistic timed automata (PTAs): dense time, (usually) discrete probability, admit nondeterminism
  - Continuous time Markov chains (CTMCs): exponentially distributed delays, discrete space, no nondeterminism

Time, clocks and zones

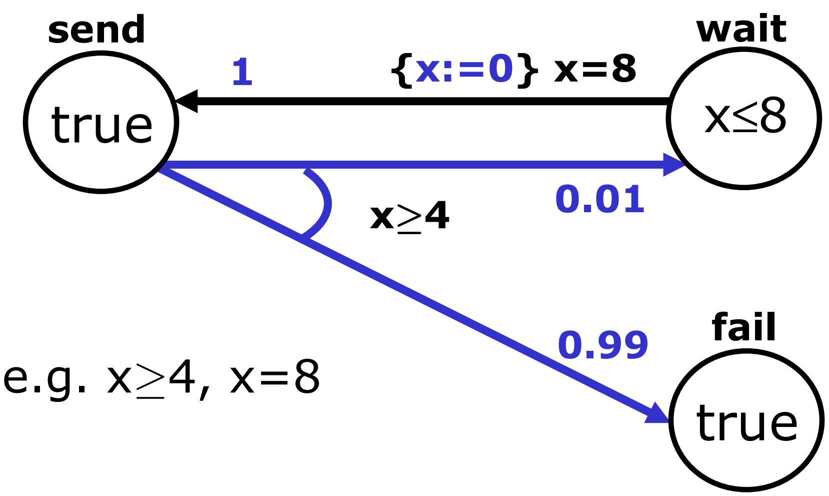
- Dense real-time, $t \in \mathbb{R}_{\geq 0}$
- Finite set $\mathcal{X}$ of clocks take values from time domain $\mathbb{R}_{\geq 0}$
  - clocks increase at the same rate as real time
  - $\mathbf{v} : \mathcal{X} \to \mathbb{R}_{\geq 0}$ is called a clock valuation
    - $\mathbf{v}+t$ is clock valuation where all clocks incremented by t
    - $\mathbf{v}[X:=0]$ is the clock valuation where all clock in X are reset

- Clock Constraints, for $x,y \in \mathcal{X}$, $c \in \mathbb{N}$, $\sim \in \{<,>,\leq,\geq\}$

$$\zeta ::= x \sim c \mid x\text{-}y \sim c \mid \zeta \wedge \zeta \mid \zeta \vee \zeta \mid \neg \zeta$$

  - closed, diagonal-free if do not feature $x < c$, $x > c$, $x\text{-}y \sim c$
  - $CC(\mathcal{X})$ set of clock constraints over $\mathcal{X}$
  - $\mathbf{v} \models \zeta$ if substituting the values of the clocks from $\mathbf{v}$ in ζ yields true

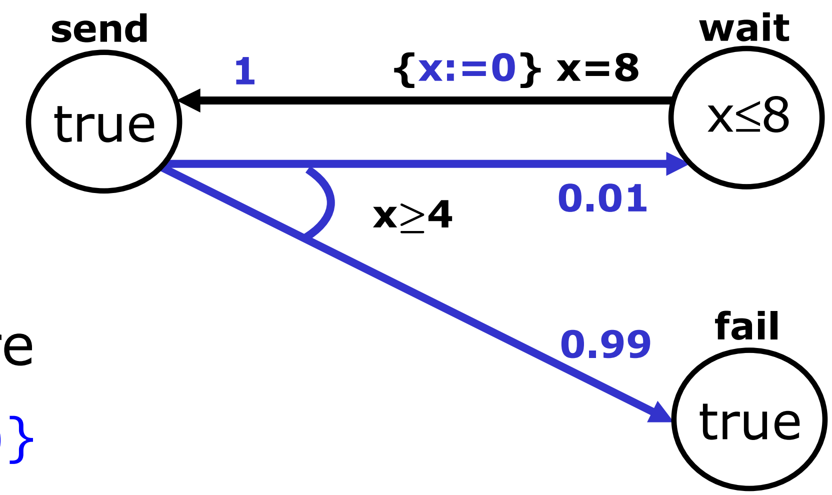
Probabilistic timed automata - Syntax



- Features:
  - Clocks, x, real-valued
  - Can be reset, e.g. {x:=0}
  - Invariants, e.g. $x \leq 8$
  - Probabilistic transitions, guarded e.g. $x \geq 4$, $x=8$
- Formally, $PTA=(Loc,l_0,inv,prob,L)$
  - Loc finite set of locations, l_0 initial location
  - $inv : Loc \rightarrow CC(\mathcal{X})$ maps locations to invariant clock constraints
  - $(l,g,p) \in prob \subseteq Loc \times CC(\mathcal{X}) \times Dist(2^{\mathcal{X}} \times Loc)$ probabilistic edge relation
    - l is the source location
    - g is the guard
    - $p(l',X)$ is the probability of moving to location l' and resetting the clocks X
  - $L: S \rightarrow 2^{AP}$ atomic propositions

Probabilistic timed automata - Semantics

- PTA=(Loc,l_0,inv,prob,L)



- MDP_{PTA}=(S,s_0,Steps,L') where
  - S={(l,**v**) | l∈ Loc ∧ **v** ⊨ inv(l)}
  - s_0=(l_o,**0**), L'(l,**v**)=L(l)
  - μ ∈ Steps(l,**v**) if one of the following conditions is satisfied:
    - time transition: ∃t ∈ $\mathbb{R}_{\geq 0}$ such that μ(l,**v**+t)=1 and inv(l) satisfied by **v**+t' for all 0 ≤ t' ≤ t
    - discrete transition: ∃(l,g,p)∈prob such that **v** ⊨ g and for any (l',**v**') ∈ S: μ (l',**v**') = ∑{ p(l',X) | X ⊆ $\mathcal{X}$ ∧ **v**[X:=0]=**v**' }

Probabilistic timed automata - Properties

- Probabilistic reachability
  - What is the maximum probability a data packet lost in the first 5 seconds of operation?
  - What is minimum probability that a message is sent with at most 4 retransmissions?
- Expected reachability
  - What is the maximum expected time until a data packet is delivered?
  - What is the minimum number of packets sent before a failure occurs?
- Probabilistic Timed CTL based on TCTL [AD94]
  - example: $z.[P_{\geq 0.98} (\Diamond \text{ delivered} \wedge z < 5)]$

    "under any scheduling, with probability ≥ 0.98 the message is correctly delivered within 5 ms"

PTA model checking - Digital clocks

- Time domain restricted to $\mathbb{N}$
  - based on digitisation of timed automata [HMP92]
  - restricted to closed, diagonal-free PTAs
    - not important for many case studies
  - integer-valued clocks and only integer-valued time elapse allowed
  - $t \in \mathbb{N}$ clock x increment by $\min\{\mathbf{v}(x)+t,\ k_{max}+1)$
    - k_{max}: largest constant in the clock constraints of the PTA
  - finiteness of state space immediate
  - preserves a subset of properties [KNPS06]:
    - Probabilistic reachability and expected reachability
  - Does not preserve PTCTL
  - Inefficiency: large constants yield very large state spaces

PTA model checking – Zone based

- Symbolic (zone based) approaches
  - Based on the notation of symbolic states (l, ζ)
    - l is location and ζ is a clock constraint
    - Encodes the set of states $\{ (l,\mathbf{v}) \mid \mathbf{v} \models \zeta \}$
  - Region graph approach [KNSS02,ACD93]
    - Allows verification of full PTCTL
    - Prohibitively large state spaces for realistic systems
  - Forward exploration [KNSS02]
    - Approximate results: upper bound on maximum reachability probabilities
    - Efficient operations on symbolic states
  - Backwards exploration [KNSW04]
    - Allows for the verification of full PTCTL
    - Requires complex operations on symbolic states

Other research topics

- More expressive logics: LTL, PCTL*, … see e.g. [CY95]
- Fairness considerations for MDP verification [BK98,Bai98]
- Long run average properties for MDPs [dAl97]
- Probabilistic process algebras, e.g. [Han94,Hil96]
- Probabilistic verification for other model types:
  - continuous-time Markov chains (CTMCs) [BHHK03]
  - continuous-time MDPs (CTMDPs) [BHKH06]
  - labelled Markov processes (LMPs) [DEP02]
  - interactive Markov chains [Her02]

References (DTMCs and MDPs)

- **[Bai98]** C. Baier. On algorithmic verification methods for probabilistic systems. Habilitation thesis, Fakultat fur Mathematik & Informatik, Universitat Mannheim, 1998.
- **[BK98]** C. Baier and M. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11(3):125-155, 1998.
- **[BdA95]** A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proc. Foundations of Software Technology and Theoretical Computer Science*, pp. 499-513, 1995.
- **[CY95]** C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM,* 42(4):857-907, 1995.

References (DTMCs and MDPs)

- **[dAl97]** L. de Alfaro. Formal verification of probabilistic systems, Ph.D. thesis, Stanford University, 1997.
- **[HJ94]** H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512-535, 1994.
- **[KSK76]** J. Kemeny, J. Snell, A. Knapp. Denumerable Markov Chains, 2nd Edition, Springer, 1976.

References (PTAs)

- **[ACD93]** R. Alur, C. Courcoubetis and D. Dill. Model-Checking in Dense Real-time. *Information and Computation*, 104(1):2-34, 1993.
- **[AD94]** R. Alur and D. Dill. A theory of timed automata**.** *Theoretical Computer Science* 126:183-235, 1994
- **[HMP92]** T. Henzinger, Z. Manna, and A. Pnueli. What good are digital clocks? *Proc. 19th Int. Coll. Automata, Languages, and Programming* (ICALP), volume 623 of LNCS 623, pages 545-558, Springer, 1992.

References (PTAs)

- **[KNPS06]** M. Kwiatkowska, G. Norman, D. Parker and J. Sproston. Performance Analysis of Probabilistic Timed Automata using Digital Clocks. *Formal Methods in System Design*, 29, pages 33-78, Springer, 2006.
- **[KNSW04]** M. Kwiatkowska, G. Norman, J. Sproston and F. Wang. Symbolic Model Checking for Probabilistic Timed Automata. In *Proc. FORMATS & FTRTFT'04*, volume 3253 of LNCS, pages 293-308, Springer, 2004.
- **[KNSS02]** M. Kwiatkowska, G. Norman, R. Segala and J. Sproston. Automatic Verification of Real-time Systems with Discrete Probability Distributions. *Theoretical Computer Science*, 282:101-150, 2002.

References

- **[Hil96]** J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.
- **[Han94]** H. Hansson. *Time and Probability in Formal Design of Distributed Systems*. Elsevier, 1994.
- **[BHKH06]** C. Baier, H. Hermanns, J.-P. Katoen, B. Haverkort. Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *Theoretical Computer Science*, 345(1):2-26, 2006.
- **[BHHK03]** C. Baier, B. Haverkort, H. Hermanns, J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains, *IEEE Trans. Software Eng*. 29(6):524-541, 2003.
- **[Her02]** H. Hermanns. Interactive Markov Chains and the Quest for Quantified Quality, Lecture Notes in Computer Science, Vol. 2428, Springer, Berlin, 2002.

Further reading

- **[VOSS]** *Validation of Stochastic Systems: A Guide to Current Research*. LNCS Volume 2925 (Tutorial Volume). Springer, 2004.
- **[dAl99]** L. de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In Proc. Int. Conf. Concurrency Theory (CONCUR'99), volume 1664 of LNCS, pages 66–81. Springer, 1999.
- **[RKNP04]** J. Rutten, M. Kwiatkowska, G. Norman and D. Parker. Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems. Volume 23 of CRM Monograph Series. American Mathematical Society. P. Panangaden and F. van Breugel (editors). 2004.
- **[Seg96]** R. Segala. Modelling and verification of randomized distributed real time systems, Ph.D. thesis, MIT, 1995.

Further reading (PTAs)

- **[ACD91]** R. Alur, C. Courcoubetis, and D. Dill. Model-checking for probabilistic realtime systems. In Proc. 18th ICALP, volume 510 of LNCS, pages 115–126, 1991
- **[Beu03]** D. Beauquier. Probabilistic timed automata, *Theoretical Computer Science*, 292(1):65–84, 2003.
- **[DKN04]** C. Daws, M. Kwiatkowska, G. Norman. Automatic verification of the IEEE1394 root contention protocol with Kronos and PRISM, *Int. Journal on Software Tools for Technology Transfer*, 5(2–3):221–236, 2004.

Further reading (general distributions)

- **[BD04]** M. Bravetti and P. D’Argenio. Tutte le algebre insieme: Concepts, discussions and relations of stochastic process algebras with general distributions. In *Validation of Stochastic Systems: A Guide to Current Research*, volume 2925 of LNCS (Tutorial Volume). Springer, 2004.
- **[CSKN05]** S. Cattani, R. Segala, M. Kwiatkowska, G. Norman. Stochastic transition systems for continuous state spaces and non-determinism. In *Proc. FOSSACS'05*, volume 3441 of LNCS, pages 125-139, Springer Verlag, 2005.
- **[DEP02]** J. Desharnais, A. Edalat, P. Panangaden. Bisimulation for labelled Markov processes. Information and Computation, 179(2):163–193, 2002.
- **[KNSS00a]** M. Kwiatkowska, G. Norman, R. Segala, J. Sproston. Verifying Quantitative Properties of Continuous Probabilistic Timed Automata. In *Proc. Concurrency Theory*, volume 1877 of LNCS, pages 123-137, Springer, 2000.