Probabilistic Timed Automata with Clock-Dependent Probabilities

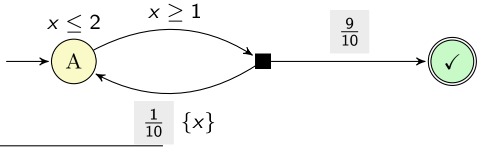
Jeremy Sproston

Dipartimento di Informatica
University of Turin
Italy

RP 2017
9th September 2017

Previous work: probabilistic timed automata

- Probabilistic timed automata (PTA) [GJ95,KNSS02]: timed automata with (discrete) probabilistic choice over edges.
- PTA conservatively extend:
  - (Alur-Dill) timed automata (clock variables, constraints and resets);
  - (Segala) probabilistic automata (presence of *nondeterministic* and *probabilistic* choice over transitions).
- Example of PTA:
  - System repeatedly attempts to complete a task.
  - Each task attempt takes between 1 and 2 time units (*nondeterministic* choice).
  - A task attempt can be successful or unsuccessful (*probabilistic* choice).



[GJ95] H. Gregersen and H. E. Jensen. "Formal Design of Reliable Real Time Systems". MS Thesis. Aalborg Univ., 1995.

[KNSS02] M. Kwiatkowska et al. "Automatic verification of real-time systems with discrete probability distributions". In: *TCS* 286 (2002), pp. 101–150.

Previous work: probabilistic timed automata

- Nondeterminism means that there is no unique probability of a system event: identify *maximum* or *minimum* probability of an event.

Result (maximum probabilistic reachability problem) [KNSS02,LS07]

Given a PTA and a threshold $\lambda \in [0, 1]$, the problem of determining whether the maximum probability that the PTA reaches a set of final locations greater than λ is decidable (EXPTIME-complete).

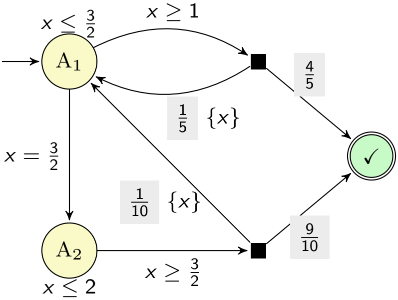
- Region-graph-based construction of a finite-state probabilistic automaton that is equivalent (w.r.t. time-abstract probabilistic bisimulation) to the PTA.
- Extend to minimum probabilistic reachability problem, probabilistic model checking problems ($\mathrm{P}\textsc{ctl}^*$ etc.), max./min. probabilistic/priced properties etc.

[KNSS02] M. Kwiatkowska et al. "Automatic verification of real-time systems with discrete probability distributions". In: *TCS* 286 (2002), pp. 101–150.

[LS07] F. Laroussinie and J. Sproston. "State explosion in almost-sure probabilistic reachability". In: *IPL* 102.6 (2007), pp. 236–241.

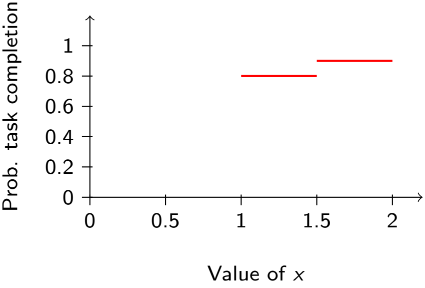
Motivation: probability changing with time

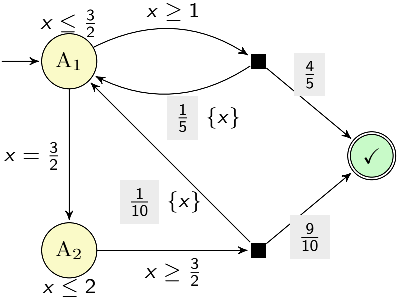
- Probabilities may depend on time: e.g., success of task completion may increase with the amount of time dedicated to the task attempt.
- Expressing a relationship between probabilities and time in PTAs: split guards; duplicate locations; distributions over edges have different probabilities.
- Example:
  - As before, task completion between 1 and 2 time units.
  - Assign a higher probability to task completion when x is in interval $[\frac{3}{2}, 2]$.
  - Note that probabilities remain *constant* as time passes when x is in interval $[1, \frac{3}{2}]$, similarly with $[\frac{3}{2}, 2]$.



Motivation: probability changing with time

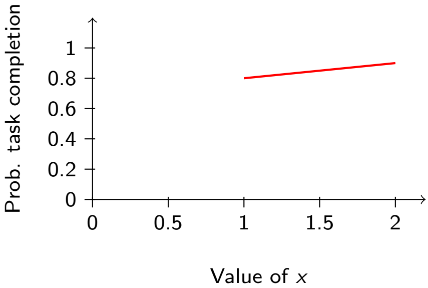
- Probabilities may depend on time: e.g., success of task completion may increase with the amount of time dedicated to the task attempt.
- Expressing a relationship between probabilities and time in PTAs: split guards; duplicate locations; distributions over edges have different probabilities.

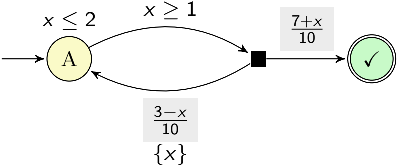




Motivation: probability changing with time

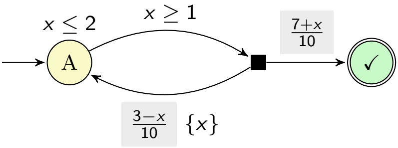
- Alternative to piecewise constant functions: *piecewise linear* functions.





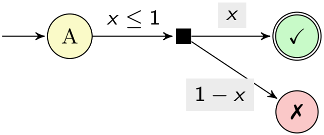
Clock-dependent probabilistic timed automata

- A *clock-dependent probabilistic timed automaton* (cdPTA) comprises:
  - Standard PTA components:
    - Locations (+ initial location); clocks; invariant conditions.
    - "Nails" (the black squares, each with a source location and a guard condition).
    - Edges from nails to locations (with clock resets).
  - *Distribution templates*: functions $\mathfrak{d}$: ClockVals $\rightarrow$ Dist(Edges) associated with each nail, describing which distribution over edges to use given the current clock valuation.
- Piecewise linear clock dependencies: distribution templates described by (sums of) piecewise linear functions (defined with respect to intervals with endpoints in $\mathbb{Q}$), one for each clock.
  - E.g., for clock valuation $v \in$ ClockVals, if $v(x)$ is in interval $[1, 2]$, edge to ✓ location has probability $\frac{7+v(x)}{10}$.



Clock-dependent probabilistic timed automata

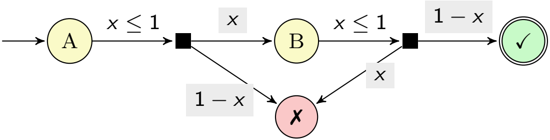
- Example 1.



- Maximum probability strategy to reach location ✓:
  - Leave location A when x is equal to 1.
  - Probability of reaching location ✓ for this strategy is 1.

Clock-dependent probabilistic timed automata

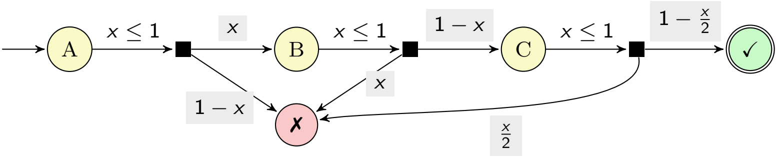
- Example 2.



- Maximum probability strategy to reach location ✓:
  - Leave location A when x is equal to $\frac{1}{2}$, then leave location B instantly.
  - Probability of reaching location ✓ for this strategy is $\frac{1}{4}$.

Clock-dependent probabilistic timed automata

- Example 3.



- Maximum probability strategy to reach location ✓:
  - Leave location A when x is equal to $1-\frac{\sqrt{3}}{3}$, then leave locations B and C instantly.
  - Probability of reaching location ✓ for this strategy is ≈ 0.19245.

Clock-dependent probabilistic timed automata

- Region graph [AD94] and corner-point abstraction [BBL08]: finite-state transition systems that can be used for solving reachability/model checking/optimality etc. problems on (P)TA.
- Obtained by a finite partitioning of the state space, using a time granularity such that each constant used in the guard/invariant constraints are multiples of the time granularity.
  - E.g., for a (P)TA with guards $x \geq \frac{3}{2}$, $y \geq 1$ and invariants $x \geq 2$, $y < \frac{5}{2}$, the coarsest granularity is $\frac{1}{2}$.
- Rely on fact that choices (of time delays) witnessing the solution of a problem (w.r.t. reachability/model checking/optimality...) are made *at* or *arbitrarily close to* multiples of the time granularity.
  - Difficulty: in cdPTA (even with one clock) this does not occur (previous example with maximum probability of reaching ✓ location has probability $1 - \frac{\sqrt{3}}{3}$).

[AD94] R. Alur and D. L. Dill. "A theory of timed automata". In: *TCS* 126.2 (1994), pp. 183–235.

[BBL08] P. Bouyer, E. Brinksma, and K. G. Larsen. "Optimal Infinite Scheduling for Multi-Priced Timed Automata". In: *FMSD* 32.1 (2008), pp. 2–23.

Undecidability

Result

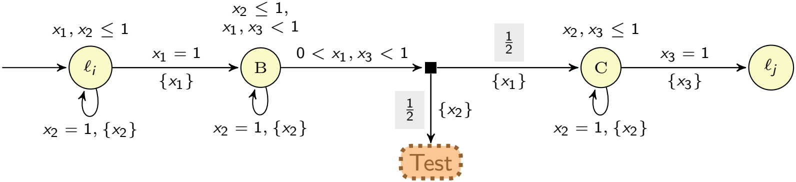
The maximal reachability problem is undecidable for cdPTAs with at least 3 clocks.

- Simulate a two-counter machine:
  - Encode value of a counter c_i using a clock x_i:
    $$x_1 = \frac{1}{2^{c_1}} \text{ and } x_2 = \frac{1}{2^{c_2}} .$$
  - Represent each instruction using a cdPTA module that maintains the counter encoding: based on [ABKMT16].
- The two-counter machine does not halt if the maximum probability of reaching target locations in the cdPTA is at least $\frac{1}{4}$.
  - Correct simulation of the two-counter machine corresponds to reaching target locations with probability $\frac{1}{4}$ in each module.
  - Hence halting corresponds to reaching target locations with probability less than $\frac{1}{4}$.

[ABKMT16] S. Akshay et al. "Stochastic Timed Games Revisited". In: *MFCS'16*. Vol. 58. LIPIcs. Leibniz-Zentrum für Informatik, 2016, 8:1–8:14.

Undecidability

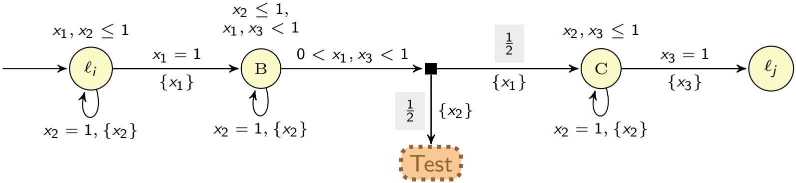
- Encoding increment instruction for c_1.



- Let δ be the amount of time that elapses in location B.

Undecidability

- Encoding increment instruction for c_1.

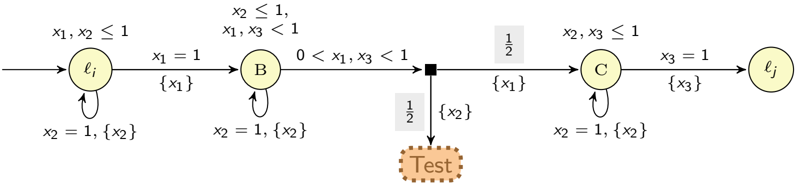


$$\begin{pmatrix} x_1 = \frac{1}{2^{c_1}} \\ x_2 = \frac{1}{2^{c_2}} \\ x_3 = 0 \end{pmatrix}$$

- Let δ be the amount of time that elapses in location B.

Undecidability

- Encoding increment instruction for c_1.

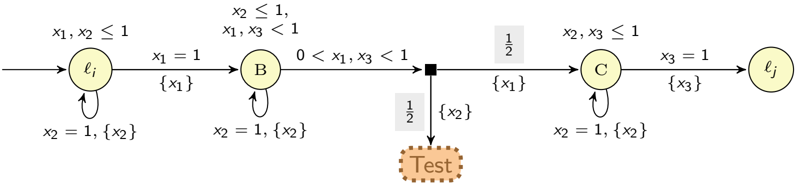


$$\begin{pmatrix} x_1 = \frac{1}{2^{c_1}} \\ x_2 = \frac{1}{2^{c_2}} \\ x_3 = 0 \end{pmatrix} \begin{pmatrix} x_1 = 0 \\ x_2 = \frac{1}{2^{c_2}} + 1 - \frac{1}{2^{c_1}} \mod 1 \\ x_3 = 1 - \frac{1}{2^{c_1}} \end{pmatrix}$$

- Let δ be the amount of time that elapses in location B.

Undecidability

- Encoding increment instruction for c_1.

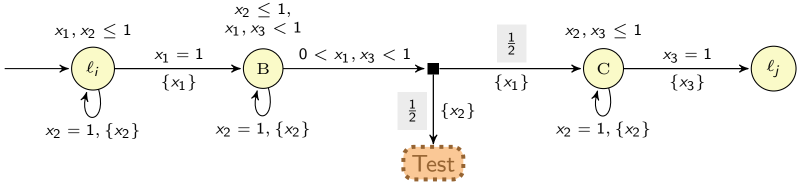


$$\begin{pmatrix} x_1 = \frac{1}{2^{c_1}} \\ x_2 = \frac{1}{2^{c_2}} \\ x_3 = 0 \end{pmatrix} \begin{pmatrix} x_1 = 0 \\ x_2 = \frac{1}{2^{c_2}} + 1 - \frac{1}{2^{c_1}} \mod 1 \\ x_3 = 1 - \frac{1}{2^{c_1}} \end{pmatrix} \qquad \begin{pmatrix} x_1 = 0 \\ x_2 = \frac{1}{2^{c_2}} + 1 - \frac{1}{2^{c_1}} + \delta \mod 1 \\ x_3 = 1 - \frac{1}{2^{c_1}} + \delta \end{pmatrix}$$

- Let δ be the amount of time that elapses in location B.

Undecidability

- Encoding increment instruction for c_1.

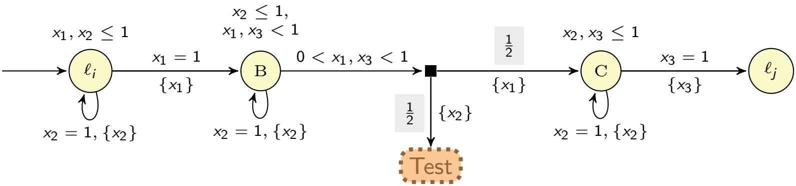


$$\begin{pmatrix} x_1 = \frac{1}{2^{c_1}} \\ x_2 = \frac{1}{2^{c_2}} \\ x_3 = 0 \end{pmatrix} \begin{pmatrix} x_1 = 0 \\ x_2 = \frac{1}{2^{c_2}} + 1 - \frac{1}{2^{c_1}} \mod 1 \\ x_3 = 1 - \frac{1}{2^{c_1}} \end{pmatrix} \qquad \begin{pmatrix} x_1 = 0 \\ x_2 = \frac{1}{2^{c_2}} + 1 - \frac{1}{2^{c_1}} + \delta \mod 1 \\ x_3 = 1 - \frac{1}{2^{c_1}} + \delta \end{pmatrix} \begin{pmatrix} x_1 = \frac{1}{2^{c_1}} - \delta \\ x_2 = \frac{1}{2^{c_2}} \\ x_3 = 0 \end{pmatrix}$$

- Let δ be the amount of time that elapses in location B.

Undecidability

- Encoding increment instruction for c_1.

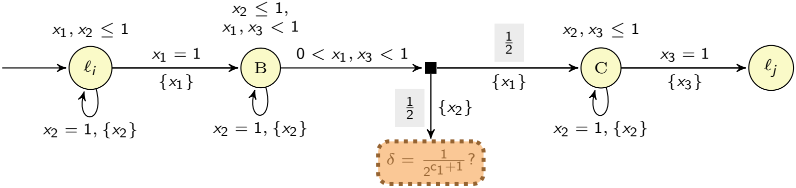


$$\begin{pmatrix} x_1 = \frac{1}{2^{c_1}} \\ x_2 = \frac{1}{2^{c_2}} \\ x_3 = 0 \end{pmatrix} \begin{pmatrix} x_1 = 0 \\ x_2 = \frac{1}{2^{c_2}} + 1 - \frac{1}{2^{c_1}} \mod 1 \\ x_3 = 1 - \frac{1}{2^{c_1}} \end{pmatrix} \qquad \begin{pmatrix} x_1 = 0 \\ x_2 = \frac{1}{2^{c_2}} + 1 - \frac{1}{2^{c_1}} + \delta \mod 1 \\ x_3 = 1 - \frac{1}{2^{c_1}} + \delta \end{pmatrix} \begin{pmatrix} x_1 = \frac{1}{2^{c_1}} - \delta \\ x_2 = \frac{1}{2^{c_2}} \\ x_3 = 0 \end{pmatrix}$$

- For x_1 to be equal to $\frac{1}{2^{c_1+1}}$ on entry to ℓ_j, must have $\delta = \frac{1}{2^{c_1+1}}$.

Undecidability

- Encoding increment instruction for c_1.

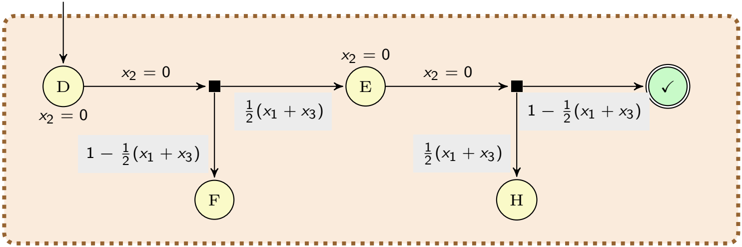


$$\begin{pmatrix} x_1 = \frac{1}{2^{c_1}} \\ x_2 = \frac{1}{2^{c_2}} \\ x_3 = 0 \end{pmatrix} \begin{pmatrix} x_1 = 0 \\ x_2 = \frac{1}{2^{c_2}} + 1 - \frac{1}{2^{c_1}} \mod 1 \\ x_3 = 1 - \frac{1}{2^{c_1}} \end{pmatrix} \qquad \begin{pmatrix} x_1 = 0 \\ x_2 = \frac{1}{2^{c_2}} + 1 - \frac{1}{2^{c_1}} + \delta \mod 1 \\ x_3 = 1 - \frac{1}{2^{c_1}} + \delta \end{pmatrix} \begin{pmatrix} x_1 = \frac{1}{2^{c_1}} - \delta \\ x_2 = \frac{1}{2^{c_2}} \\ x_3 = 0 \end{pmatrix}$$

- For x_1 to be equal to $\frac{1}{2^{c_1+1}}$ on entry to ℓ_j, must have $\delta = \frac{1}{2^{c_1+1}}$.

Undecidability

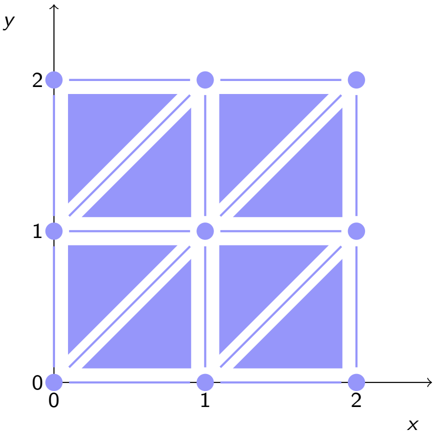
- Widget for testing whether $\delta = \frac{1}{2^{c_1+1}}$.
  - Rewrite to $\delta = \frac{1}{2^{c_1+1}} + \epsilon$, for $\epsilon \in (-\frac{1}{2^{c_1+1}}, \frac{1}{2^{c_1+1}})$.
  - Therefore widget tests whether $\epsilon = 0$.
  - On entry to location D: $\begin{pmatrix} x_1 = \delta \\ x_2 = 0 \\ x_3 = 1 - \frac{1}{2^{c_1}} + \delta \end{pmatrix}$, i.e., $\begin{pmatrix} x_1 = \frac{1}{2^{c_1+1}} + \epsilon \\ x_2 = 0 \\ x_3 = 1 - \frac{1}{2^{c_1+1}} + \epsilon \end{pmatrix}$



  - Only path to reach ✓ location: has probability $\frac{1}{2}(x_1 + x_3)(1 - \frac{1}{2}(x_1 + x_3))$.
  - $\frac{1}{2}(x_1 + x_3)$ equals $\frac{1}{2} + \epsilon$, $1 - \frac{1}{2}(x_1 + x_3)$ equals $\frac{1}{2} - \epsilon$.
  - Multiplying these together obtains $\frac{1}{4} - \epsilon^2$, which is maximised (and equals $\frac{1}{4}$) when $\epsilon = 0$.

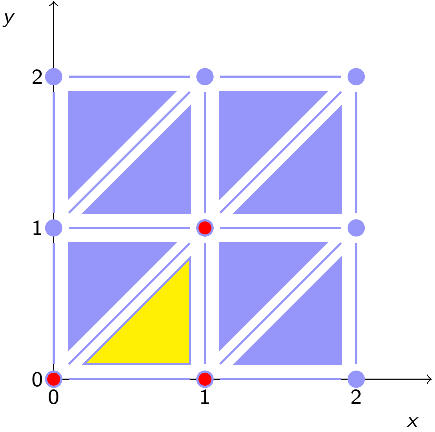
Approximation using the region graph

- Finite-state probabilistic automaton to *approximate* maximum/minimum probabilities of reaching final locations?
- Use the region graph, plus the concept of *corner*.



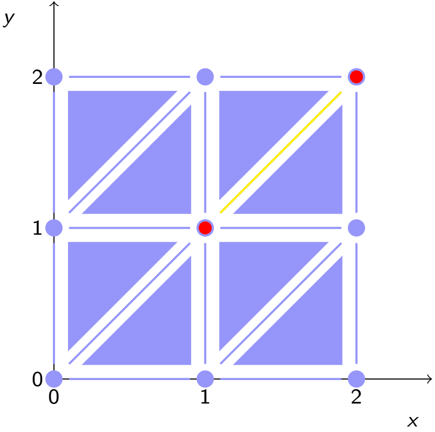
Approximation using the region graph

- Finite-state probabilistic automaton to *approximate* maximum/minimum probabilities of reaching final locations?
- Use the region graph, plus the concept of *corner*.



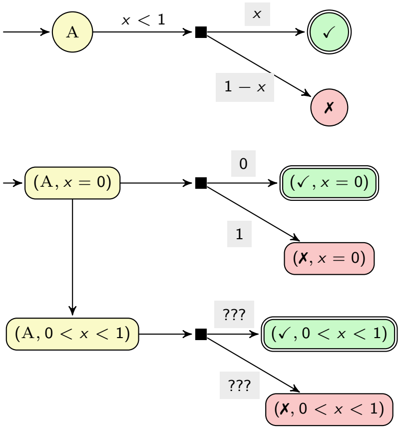
Approximation using the region graph

- Finite-state probabilistic automaton to *approximate* maximum/minimum probabilities of reaching final locations?
- Use the region graph, plus the concept of *corner*.



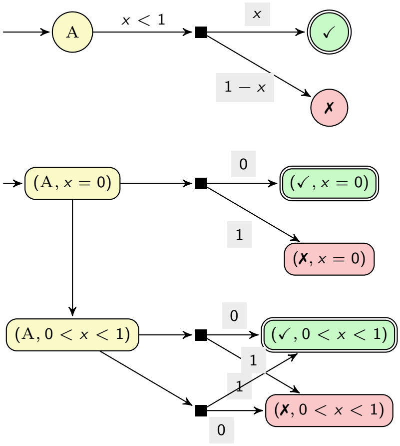
Approximation using the region graph

- Region graph (+ corner-points) $\rightsquigarrow$ finite-state probabilistic automaton.
  - States: classically-defined regions.
  - Transitions: use valuations corresponding to corner points of regions.
- Simple example:



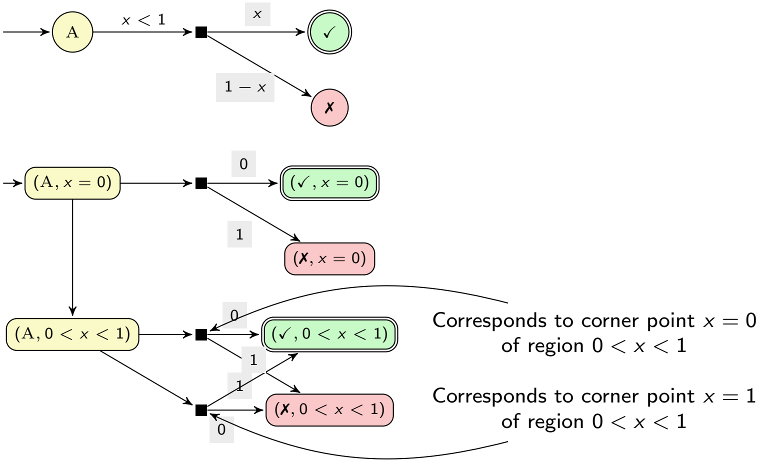
Approximation using the region graph

- Region graph (+ corner-points) $\rightsquigarrow$ finite-state probabilistic automaton.
  - States: classically-defined regions.
  - Transitions: use valuations corresponding to corner points of regions.
- Simple example:



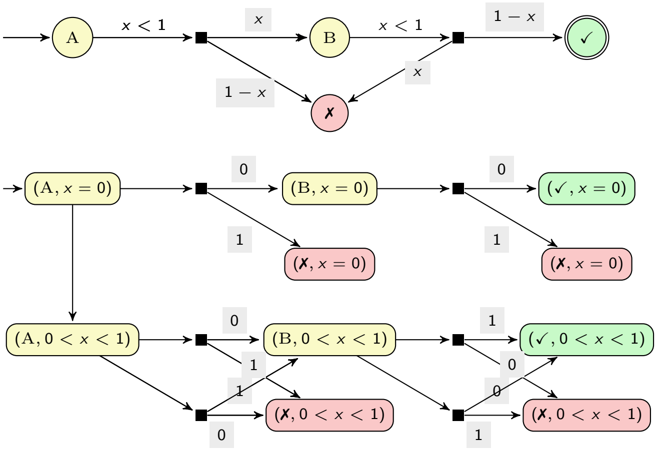
Approximation using the region graph

- Region graph (+ corner-points) $\rightsquigarrow$ finite-state probabilistic automaton.
  - States: classically-defined regions.
  - Transitions: use valuations corresponding to corner points of regions.
- Simple example:



Approximation using the region graph

- Region graph (+ corner-points) $\rightsquigarrow$ finite-state probabilistic automaton.
  - States: classically-defined regions.
  - Transitions: use valuations corresponding to corner points of regions.
- Simple example:



Approximation using the region graph

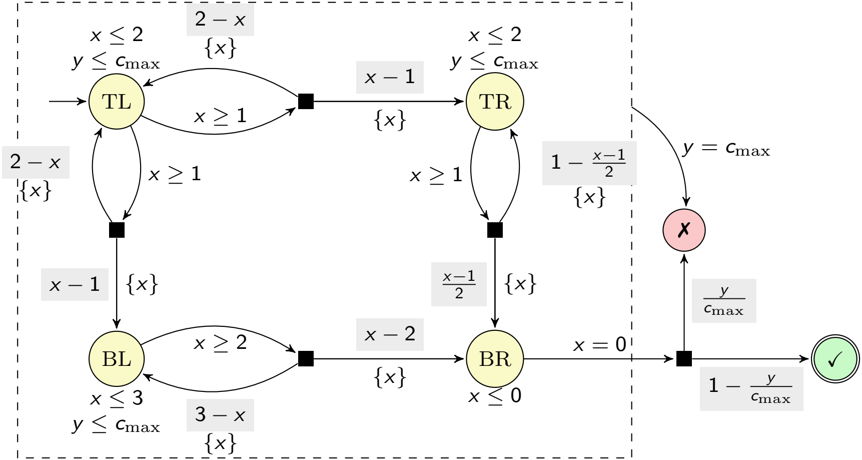
- Given cdPTA $\mathcal{P}$ with final locations F, the clock-dependent region graph $\mathcal{A}_k$ with time granularity $\frac{1}{k}$ for $k \in \mathbb{N}$:
  - $\mathbb{P}^{\max}_{\mathcal{P}}(F)$ denotes the maximum probability of reaching F in $\mathcal{P}$;
  - $\mathbb{P}^{\max}_{\mathcal{A}_k}(F)$ denotes the maximum probability of reaching F in $\mathcal{A}_k$.

Result (conservative approximation)

$$\mathbb{P}^{\max}_{\mathcal{P}}(F) \leq \mathbb{P}^{\max}_{\mathcal{A}_k}(F) \qquad \mathbb{P}^{\max}_{\mathcal{A}_{2k}}(F) \leq \mathbb{P}^{\max}_{\mathcal{A}_k}(F)\ .$$

- Hence, if the answer to the maximum probabilistic reachability problem is No for $\mathcal{A}_k$, then it is also No for $\mathcal{P}$.
- Analogous results can be obtained also for minimum probability of reaching final locations.

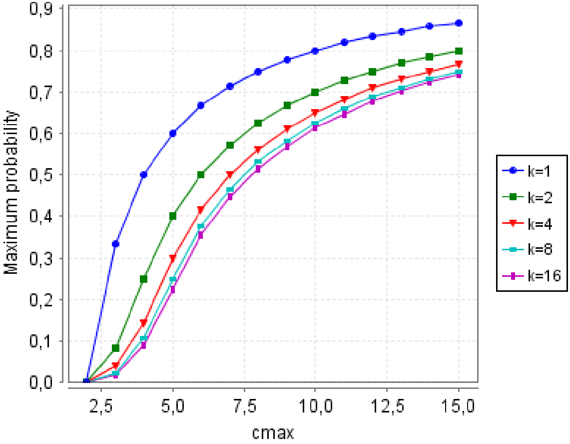
Example



$x \le 2$
$y \le c_{\max}$
TL
$2 - x$
$\{x\}$
$x \ge 1$
$x - 1$
$\{x\}$
$x \le 2$
$y \le c_{\max}$
TR
$2 - x$
$\{x\}$
$x \ge 1$
$x \ge 1$
$1 - \frac{x-1}{2}$
$\{x\}$
$y = c_{\max}$
✗
$x - 1$
$\{x\}$
$\frac{x-1}{2}$
$\{x\}$
$\frac{y}{c_{\max}}$
BL
$x \ge 2$
$x - 2$
$\{x\}$
BR
$x = 0$
✓
$1 - \frac{y}{c_{\max}}$
$x \le 3$
$y \le c_{\max}$
$3 - x$
$\{x\}$
$x \le 0$

Example

- Maximum probability of reaching location ✓ (obtained by encoding the clock-dependent region graph in the probabilistic model checking tool PRISM).



Conclusions

- Basic (quantitative) probabilistic verification problems for cdPTA are undecidable.
- ... but approximation of reachability probabilities is possible with the clock-dependent region graph.
- Future work:
  - Monotone functions.
  - Qualitative problems.
  - Game-based approximations.
  - Approximation up to ϵ given clock-dependencies of certain forms (e.g., piecewise-linear).
  - (Simple classes of) hybrid systems (e.g., a robot has a greater chance of detecting a person in need of rescue the closer it is to the person; already present in some stochastic hybrid system formalisms).