# 6. Probabilities: Markov chains and statistical model checking

José Proença

System Verification (CC4084) 2024/2025

CISTER – U.Porto, Porto, Portugal

https://fm-dcc.github.io/sv2425

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

CISTER - Research Centre in
Real-Time & Embedded
Computing Systems

# Where we are

# Syllabus

- Introduction to model-checking
- CCS: a simple language for concurrency
  - Syntax
  - Semantics
  - Equivalence
  - mCRL2: modelling
- Dynamic logic
  - Syntax
  - Semantics
  - Relation with equivalence
  - mCRL2: verification
- Timed Automata
  - Syntax
  - Semantics (composition, Zeno)
  - Equivalence
  - UPPAAL: modelling
- Temporal logics (LTL/CTL)
  - Syntax
  - Semantics
  - UPPAAL: verification
- Probabilistic and stochastic systems
  - Going probabilistic
  - UPPAAL: monte-carlo

# Going probabilistic

# Motivation

## Systems can get very complex

- E.g., 5 components, 3 possible traces each
- No communication (pure interleaving)
- Many permutations

# Motivation

## Systems can get very complex

- E.g., 5 components, 3 possible traces each
- No communication (pure interleaving)
- Many permutations
- More components, more traces – untreatable

# Motivation

**Systems can get very complex**

- E.g., 5 components, 3 possible traces each
- No communication (pure interleaving)
- Many permutations
- More components, more traces – untreatable

- Verifying deadlock freedom (and others) requires traversing all states
- Approximation:
  - traverse only part of the states
  - give more priority to some actions
  - return (statistically) likelihood of a given property

# Recall: A taxonomy of transition systems

- $\alpha : S \to N \times S$ Moore machine
- $\alpha : S \to \texttt{Bool} \times S^N$ deterministic automata
- $\alpha : S \to \texttt{Bool} \times \mathrm{P}(S)^N$ non-deterministic automata (reactive)

- $\alpha : S \to \mathrm{P}(N \times S)$ non deterministic LTS (generative)
- $\alpha : S \to (S + 1)^N$ partial deterministic LTS
- $\alpha : S \to \mathrm{P}(S)$ unlabelled TS

- $\alpha : S \to \mathrm{D}(S)$ Markov chain

# Bringing probabilities to transition systems

## Markov chains

$$\alpha : S \to \mathrm{D}(S)$$

where $D(S)$ is the set of all discrete probability distributions on set $S$

A Markov chain goes from a state $s$ to a state $s'$ with probability $p$ if

$$\alpha(s) = \mu \quad \text{with} \quad \mu(s') = p > 0$$

# Recall discrete distributions

**Recall**

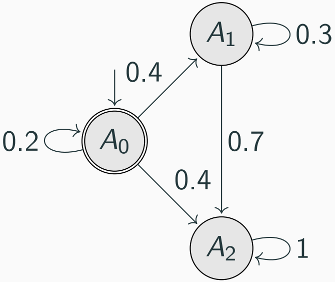
$\mu : S \to [0,1]$ is a discrete probability distribution if

- $\{s \in S \mid \mu(s) > 0\}$, is finite (called the support of $\mu$), and
- $\sum_{s\in S} \mu(s) = 1$

**Examples**

- Dirac distribution: $\mu_s^1 = \{s \to 1\}$
- Product distribution: $(\mu_1 \times \mu_2)\langle s, t\rangle = \mu_1(s) \times \mu 2(t)$

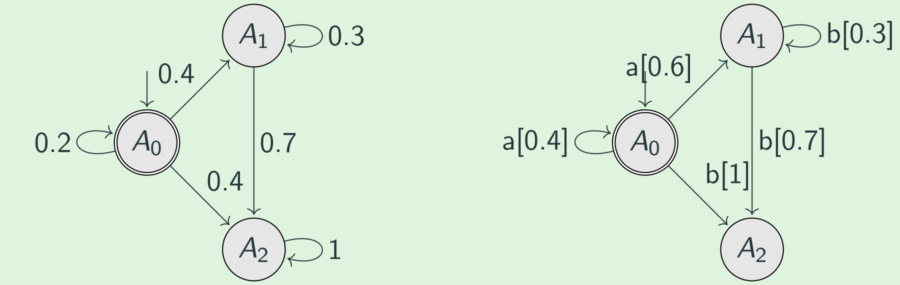
## Example



## Reactive PTS

$$\alpha : S \longrightarrow (\mathrm{D}(S) + 1)^N$$

**Ex. 6.1: Formalise the systems below as functions**

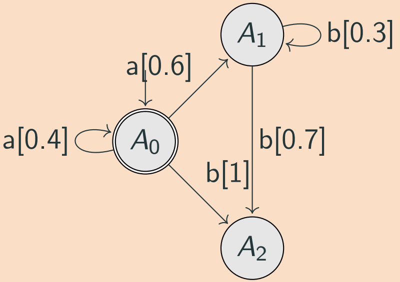



Notions of bisimulation arise naturally.

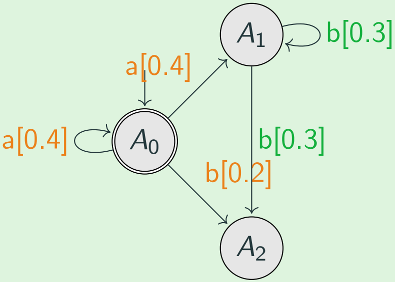
# Generative PTS

$$\alpha : S \to \mathrm{D}((S \times N) + 1)$$

**Before (reactive)**

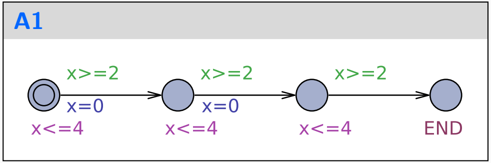


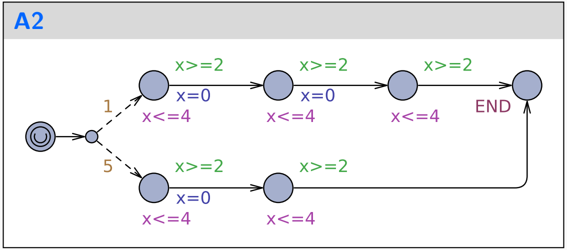

**Ex. 6.2: Now (generative) – formalise it**

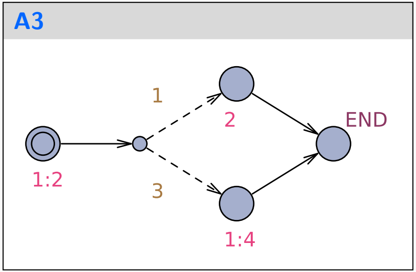


# Probabilities in Uppaal

# Stochastic Timed Automata – examples



# Stochastic Timed automata Definition

$$\langle \mathbf{L}, \mathbf{L_0}, \mathbf{Act}, \mathbf{C}, \mathbf{Tr}, \mathbf{Inv} \rangle$$

where

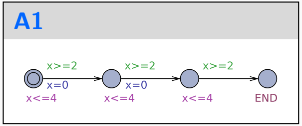
- $L$ is a set of locations, and $L_0 \subseteq L$ the set of initial locations
- $Act$ is a set of actions and $C$ a set of clocks
- $Tr \subseteq L \times \mathcal{C}(C) \times Act \times \mathcal{P}(C) \times \mathbb{N} \times L$ is the transition relation

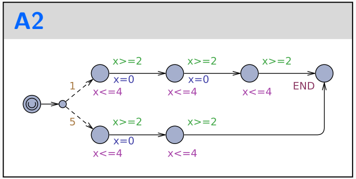
  $$\ell_1 \xrightarrow{g,a,U,w} \ell_2$$

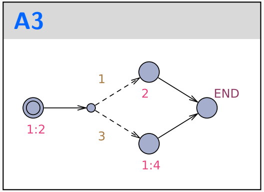
  denotes a transition from location $\ell_1$ to $\ell_2$, labelled by $a$, enabled if guard $g$ is valid, which, when performed, resets the set $U$ of clocks, with a probability given by the weight $w$
- $Inv : L \longrightarrow \mathcal{C}(C) + \mathbb{Q}$ is the assignment of invariants or rates (of an exponential distribution) to locations

where $\mathcal{C}(C)$ denotes the set of clock constraints over a set $C$ of clock variables

# Again A1,A2,A3: Timed PTS

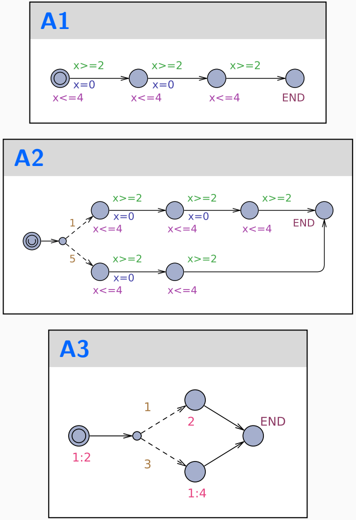


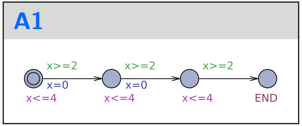

- Probability of $\langle A1_0, \overline{0}\rangle \xrightarrow{0.5} \langle A1_0, \overline{0.5}\rangle$?
- Probability of $\langle A2_0, \overline{0}\rangle \xrightarrow{0.5} \langle A2_0, \overline{0.5}\rangle$?

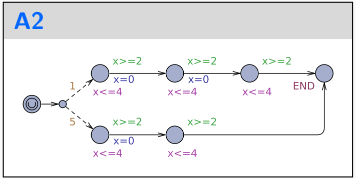
# Again A1,A2,A3: Timed PTS

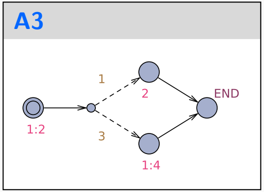



- Probability of $\langle A1_0, \overline{0} \rangle \xrightarrow{0.5} \langle A1_0, \overline{0.5} \rangle$?
- Probability of $\langle A2_0, \overline{0} \rangle \xrightarrow{0.5} \langle A2_0, \overline{0.5} \rangle$?
- Probability of $\langle A3_0, \overline{0} \rangle \xrightarrow{0.5} \langle A3_0, \overline{0.5} \rangle$?

# Again A1,A2,A3: Timed PTS

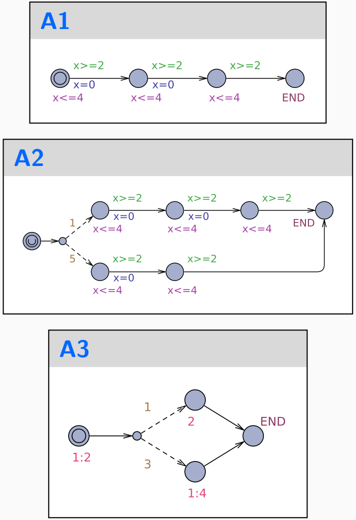



- Probability of $\langle A1_0, \overline{0}\rangle \xrightarrow{0.5} \langle A1_0, \overline{0.5}\rangle$?
- Probability of $\langle A2_0, \overline{0}\rangle \xrightarrow{0.5} \langle A2_0, \overline{0.5}\rangle$?
- Probability of $\langle A3_0, \overline{0}\rangle \xrightarrow{0.5} \langle A3_0, \overline{0.5}\rangle$?
- Probability of reaching $A1_1$?
- Probability of reaching $A2_1$?

# Again A1,A2,A3: Timed PTS

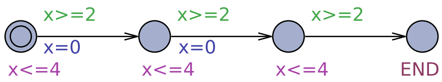



- Probability of $\langle A1_0, \overline{0} \rangle \xrightarrow{0.5} \langle A1_0, \overline{0.5} \rangle$?
- Probability of $\langle A2_0, \overline{0} \rangle \xrightarrow{0.5} \langle A2_0, \overline{0.5} \rangle$?
- Probability of $\langle A3_0, \overline{0} \rangle \xrightarrow{0.5} \langle A3_0, \overline{0.5} \rangle$?
- Probability of reaching $A1_1$?
- Probability of reaching $A2_1$?
- Probability of reaching $A3_{END}$ in less than 4.3?

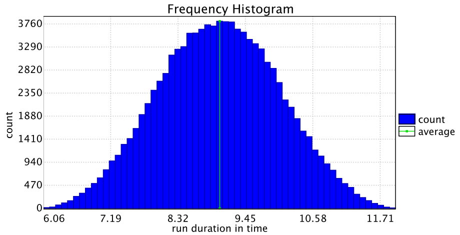
  $= \ldots$

# A1: When does it end?

## A1



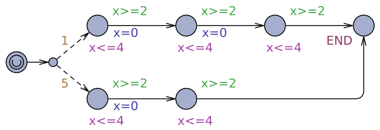

## A1's histogram



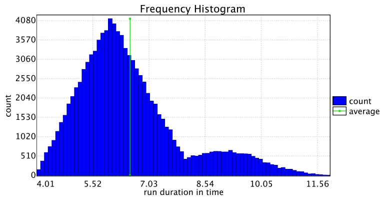

Parameters: α=0.05, ε=0.05, bucket width=0.10002, bucket count=59
Runs: 102000 in total, 102000 (100%) displayed, 0 (0%) remaining
Span of displayed sample: [6.06618, 11.9675]
Mean estimate of displayed sample: 9.00732 ± 0.00614869 (95% CI)

- Run 102000 times
- Histogram: how many times it took [9..9.1] seconds?
- ...

## A2




## A2's histogram

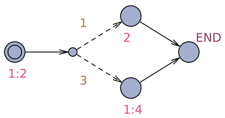



Parameters: α=0.05, ε=0.05, bucket width=0.10102, bucket count=78
Runs: 100000 in total, 100000 (100%) displayed, 0 (0%) remaining
Span of displayed sample: [4.0185, 11.8982]
Mean estimate of displayed sample: 6.5045 ± 0.00872567 (95% CI)

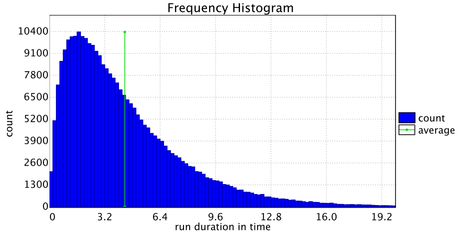
- Run 100000 times
- Histogram: how many times it took [9..9.1] seconds?
- ...

# A3: When does it end?

### A3




### A3's histogram




Parameters: α=0.05, ε=0.05, bucket width=0.19991, bucket count=100
Runs: 300000 in total, 299181 (99.727%) displayed, 819 (0.273%) remaining
Span of displayed sample: [0.0045298, 19.9958]
Mean estimate of displayed sample: 4.31514 ± 0.0118734 (95% CI)

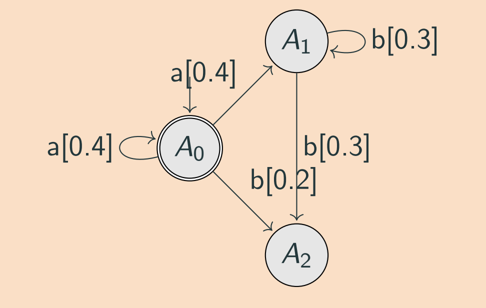
- Run 300000 times
- Histogram: how many times it took [9..9.1] seconds?
- ...

# Generative Timed PTS

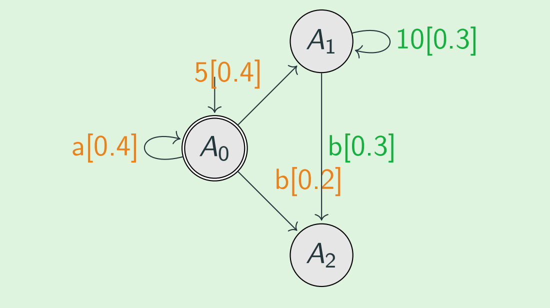
$$\alpha : S \rightarrow \mathrm{D}_{disc}((S \times N) + 1)$$

$$\alpha : S \rightarrow \mathrm{D}_{cont}((S \times (N + \mathcal{R}_0^+) + 1))$$

**Before (PTS)**




**Ex. 6.3: Now (Timed PTS) – formalise it**

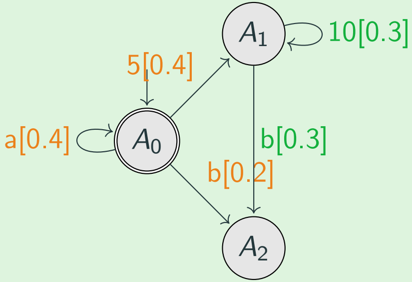


$$\alpha : S \to \mathrm{D}_{disc}((S \times N) + 1)$$
$$\alpha : S \to \mathrm{D}_{cont}((S \times (N + \mathcal{R}_0^+) + 1))$$

**Notes**

- Continuous time: continuous distribution
- Probabilities both at continuous delays and discrete transitions.

**Ex. 6.3: Now (Timed PTS) – formalise it**



# Probabilistic queries in Uppaal

# Probabilistic queries

- `Pr[c<=10; 100]([] safe)` – runs 100 stochastic simulations and estimates the probability of `safe` remaining true within 10 cost units, based on 100 runs.
- `Pr[<=10](<> good)` – runs a number of stochastic simulations and estimates the probability of `good` eventually becoming true within 10 time units. The number of runs is decided based on the probability interval precision ($\pm\varepsilon$) and confidence level (level of significance $\alpha$).
- `Pr[<=10](<> good) >= 0.5` – checks if the probability of reaching good within 10 time units is greater than 50% (less runs than calculating the probability, using "Walds's algorithm")
- `E[<=10; 100](max: cost)` runs 100 stochastic simulations and estimates the maximal value of cost expression over 10 time units of stochastic simulation.

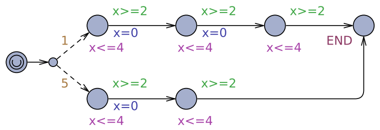
More at `https://docs.uppaal.org/language-reference/query-syntax/statistical_queries/`

# Running a single simulation

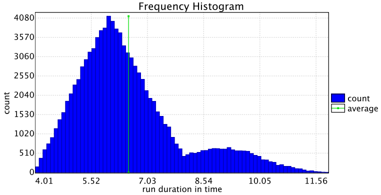
- `simulate[<=10] { x, y }` creates one stochastic simulation run of up to 10 time units in length and plot the values of `x` and `y` expressions over time (after checking, right-click the query and choose a plot).
- Variations: `[c<=10] / [#<=10]` – based on clock *c* or based on the number of transitions.

# Replicate the histograms

## A2



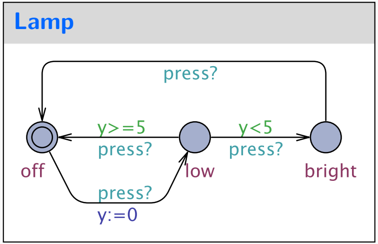

## A2's histogram




Parameters: α=0.05, ε=0.05, bucket width=0.10102, bucket count=78
Runs: 100000 in total, 100000 (100%) displayed, 0 (0%) remaining
Span of displayed sample: [4.0185, 11.8982]
Mean estimate of displayed sample: 6.5045 ± 0.00872567 (95% CI)

**Ex. 6.4:** Replicate the visualisation

**Ex. 6.5:** Replicate the visualisation also for A1 and A3




**Ex. 6.6:** Adapt the model to make it stochastic

**Ex. 6.7: Adapt requirements to make them probabilistic**

1. The lamp can become bright;
2. The lamp will eventually become bright;
3. The lamp can never be on for more than 3600s;
4. It is possible to never turn on the lamp;
5. Whenever the light is bright, the clock $y$ is non-zero;
6. Whenever the light is bright, it will eventually become off.