Extensions of Automata: Concurrent, Timed, Hybrid

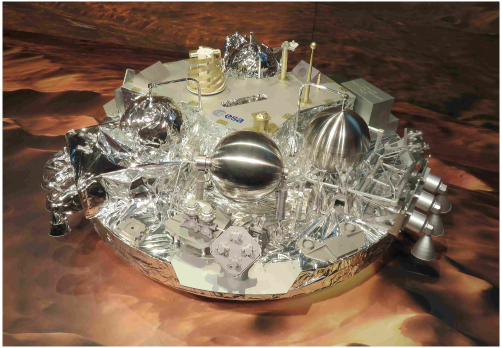
Uli Fahrenberg

EPITA Rennes

Foundations of Security and Concurrency
July 2024

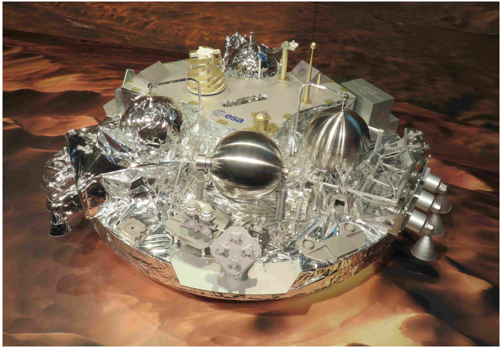
Schiaparelli

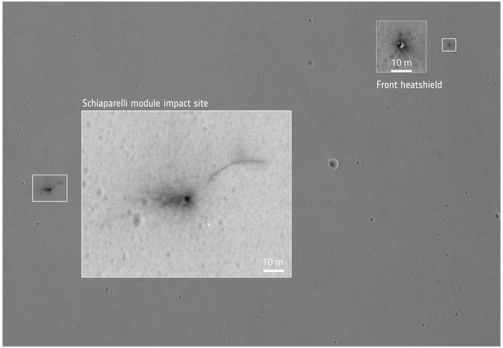
Experimental Mars lander, ESA / Roscosmos



Schiaparelli

Experimental Mars lander, ESA / Roscosmos

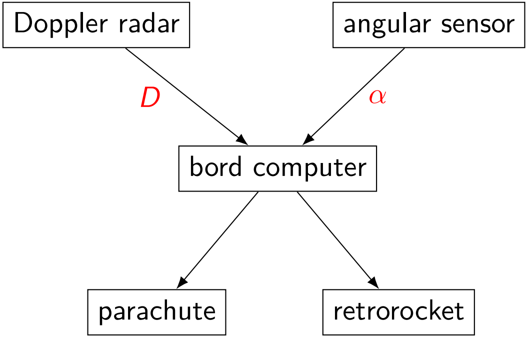


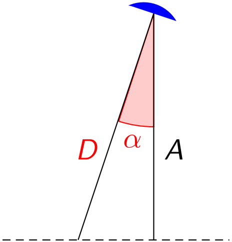


- an example of a cyber-physical system

Schiaparelli

Schematics (simplified)

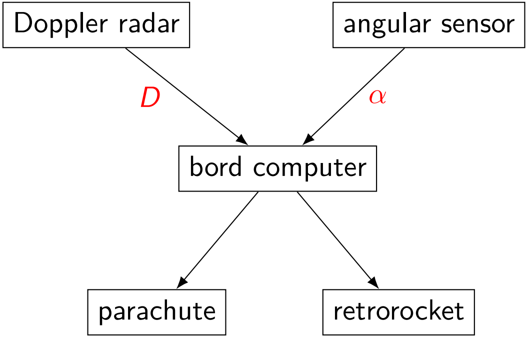


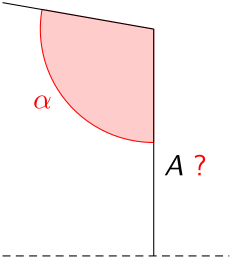


$$A = D \cdot \cos(\alpha)$$

Schiaparelli

Schematics (simplified)





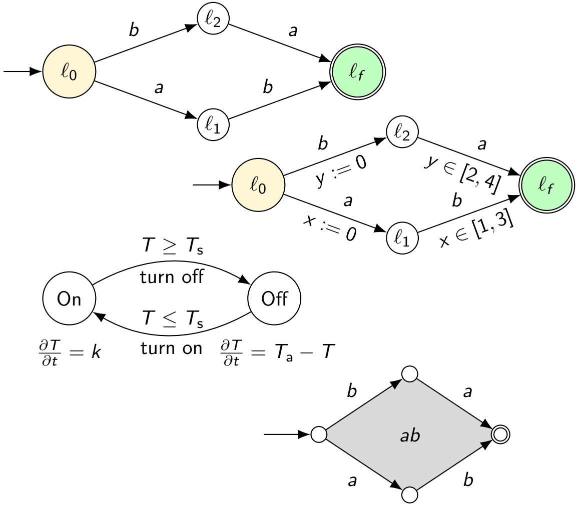
$$A = D \cdot \cos(\alpha)$$

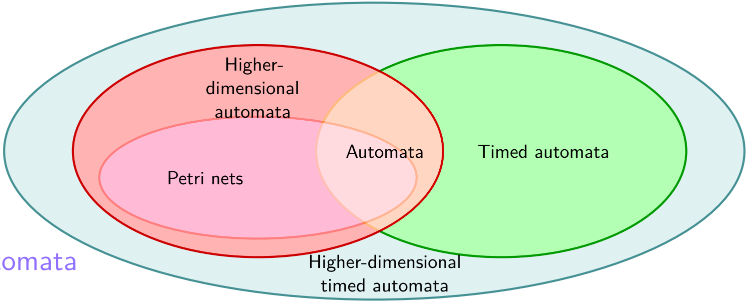
Automata

- cyber-physical systems must often be modeled and verified
- that may be done using automata
- but these need to capture timing constraints, physical information, and concurrency

⇒ timed automata; hybrid automata; higher-dimensional automata

- but how to combine them?





Higher-dimensional automata
Automata
Timed automata
Petri nets
Higher-dimensional timed automata

Who am I

Uli Fahrenberg

- University studies in mathematics and computer science
- PhD in mathematics
- Worked at Aalborg University (DK), University of Rennes, École polytechnique (Paris)
- Interested in category theory, algebraic topology, automata theory, concurrency theory, verification
- Professor at EPITA since 2021

EPITA

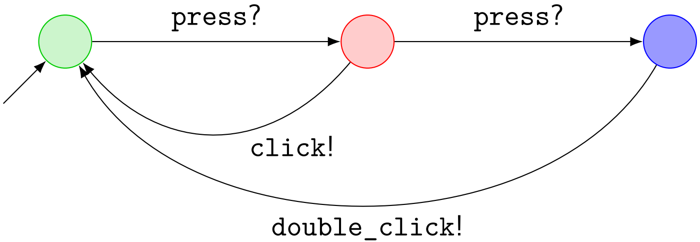
- École Pour l'Informatique et les Techniques Avancées
- private engineering school specialized in software engineering
- in Paris, Lyon, Rennes, Strasbourg, and Toulouse
- 700 students $\times$ 5 years
- accredited engineering diploma

Who are we

Joint work with

- Amazigh Amrane, EPITA Paris
- Hugo Bazille, EPITA Rennes
- Emily Clement, U Paris Cité
- Marie Fortin, U Paris Cité
- Christian Johansen, NTNU Gjøvik
- Georg Struth, U of Sheffield
- Krzysztof Ziemiański, Warsaw U

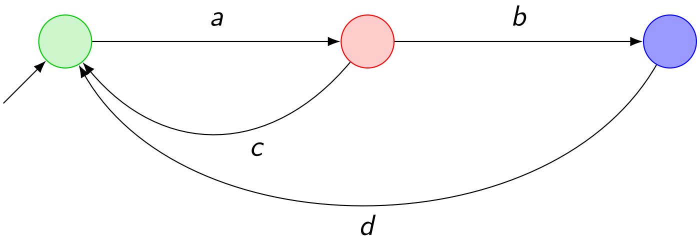
Automata



- states
- transitions labeled with actions

- operational semantics: machine which changes state depending on inputs and emits outputs
- denotational semantics: what are executions?

Automata



- states
- transitions labeled with actions

- operational semantics: machine which changes state depending on inputs and emits outputs
- denotational semantics: what are executions? $(ac + abd)^\omega$

Automata

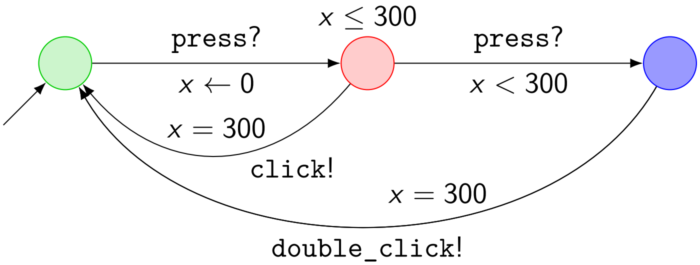
Definition

An automaton is a tuple $\mathcal{A} = (L, \bot, \top, \Sigma, E)$ consisting of a set L of states, a set of initial states $\bot \subseteq L$, a set of accepting states $\top \subseteq L$, a set Σ of labels, and a set $E \subseteq L \times \Sigma \times L$ of edges.

- A path is a finite sequence $\pi = \ell_1 \xrightarrow{a_1} \ell_2 \xrightarrow{a_2} \cdots \xrightarrow{a_{n-1}} \ell_n$ of connected transitions.
- Its label is $\lambda(\pi) = a_1 a_2 \dots a_{n-1}$.
- It is accepting if $\ell_1 \in \bot$ and $\ell_n \in \top$.
- The language of $\mathcal{A}$ is $\{\lambda(\pi) \mid \pi \text{ accepting path in } \mathcal{A}\}$.

- We only consider finite executions here.
- Hence languages are sets of (finite) words $a_1 a_2 \dots a_k \in \Sigma^*$.
- (Usually, L and Σ are also to be finite.)

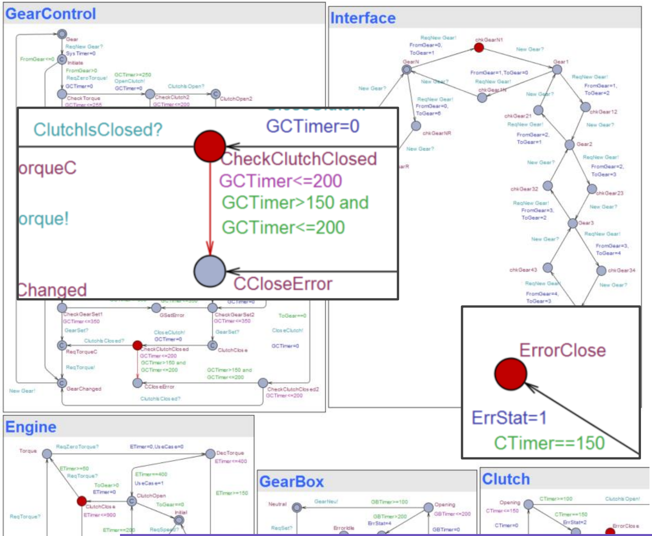
Timed automata



- states and labeled transitions
- states and transitions conditioned on values of clocks
- transitions may reset clocks
- modeling and analysis of real-time systems

UppAal

- <Global variables>
  - UseCase = 0
  - FromGear = 0
  - ToGear = 0
  - ErrStat = 0
- <Constraints>
  - CTimer ≥ 0
  - ETimer ≥ 0
  - GBTimer ≥ 0
  - GCTimer ≥ 0
  - SysTimer ≥ 0
  - GearControl.GCTimer ≥ 0
  - CTimer = ETimer
  - ETimer = GBTimer
  - GBTimer = GCTimer
  - GCTimer = SysTimer
  - SysTimer = GearControl.GCTimer
  - GearControl.GCTimer = CTimer



Timed automata

Definition

The set $\Phi(C)$ of clock constraints ϕ over a finite set C is defined by the grammar

$$\phi ::= x \bowtie k \mid \phi_1 \wedge \phi_2 \qquad (x, y \in C, k \in \mathbb{Z}, \bowtie \in \{\leq, <, \geq, >\}).$$

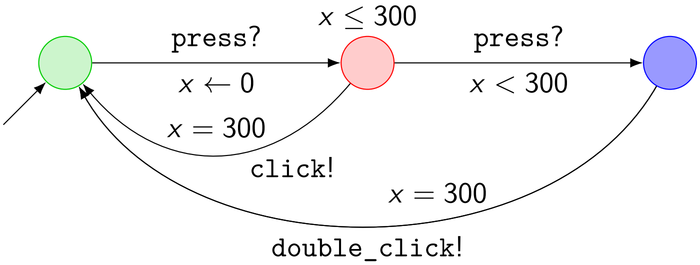
Definition

A timed automaton is a tuple $\mathcal{A} = (L, \bot, \top, C, \Sigma, I, E)$ consisting of a set L of locations, initial and accepting locations $\bot, \top \subseteq L$, a finite set C of clocks, a set Σ of labels, an invariants mapping $I : L \to \Phi(C)$, and a set $E \subseteq L \times \Phi(C) \times \Sigma \times 2^C \times L$ of edges.

- (Usually, L and Σ are to be finite.)
- The operational semantics of $\mathcal{A}$ is the infinite automaton with states $L \times \mathbb{R}_{\geq 0}^C$, alphabet $\Sigma \cup \mathbb{R}_{\geq 0}$, and transitions

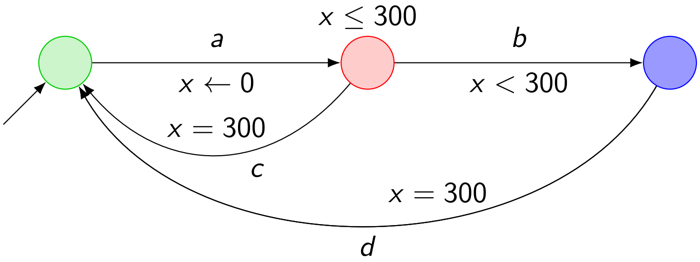
$$\begin{aligned} E = &\{(\ell, v) \xrightarrow{\delta} (\ell, v + \delta) \mid \forall t \in [0, \delta] : v + t \models I(\ell)\} \\ &\cup \{(\ell, v) \xrightarrow{a} (\ell', v') \mid \exists (\ell, \phi, a, r, \ell') \in E : v \models \phi, v' = v[r \leftarrow 0]\}. \end{aligned}$$

Timed automata



- if two press? within 300 time units, then double_click!, else click!

Timed automata

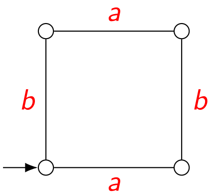


- if two `press?` within 300 time units, then `double_click!`, else `click!`
- language (one cycle only):

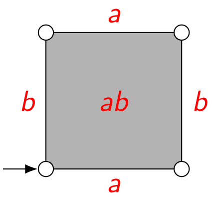
$$L = \{\delta_0 a\, \delta_1 c \mid \delta_1 = 300\} \cup \{\delta_0 a\, \delta_1 b\, \delta_2 d \mid \delta_1 < 300, \delta_1 + \delta_2 = 300\}$$

Higher-dimensional automata

$a|b$



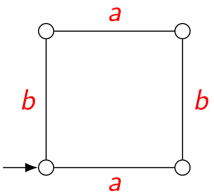
$a.b + b.a$

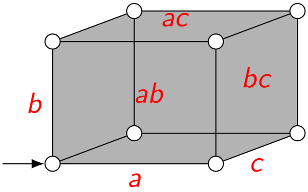


a and b are independent

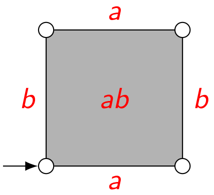
Higher-dimensional automata

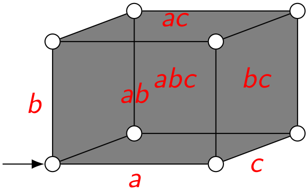
$a|b$ $a|b|c$





$a|b + a|c + b|c$





$\{a, b, c\}$ independent

Higher-dimensional automata

A conclist is a finite, ordered and Σ-labelled set. (a list of events)

A precubical set X consists of:

- A set of cells X (cubes)
- Every cell $x \in X$ has a conclist $\text{ev}(x)$ (list of events active in x)
- We write $X[U] = \{x \in X \mid \text{ev}(x) = U\}$ for a conclist U (cells of type U)
- For every conclist U and $A \subseteq U$ there are:
  upper face map $\delta_A^1 : X[U] \to X[U - A]$ (terminating events A)
  lower face map $\delta_A^0 : X[U] \to X[U - A]$ (unstarting events A)
- Precube identities: $\delta_A^\mu \delta_B^\nu = \delta_B^\nu \delta_A^\mu$ for $A \cap B = \emptyset$ and $\mu, \nu \in \{0, 1\}$

A higher dimensional automaton (HDA) is a precubical set X with start cells $\bot \subseteq X$ and accept cells $\top \subseteq X$ (not necessarily vertices)

Higher-dimensional automata

HDAs as a model for concurrency:

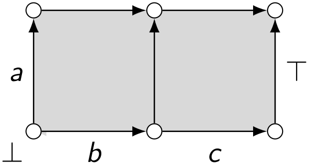
- vertices $x \in X[\emptyset]$: states
- edges $a \in X[\{a\}]$: labeled transitions
- n-squares $\alpha \in X[\{a_1, \dots, a_n\}]$ ($n \geq 2$): independency relations / concurrently executing events

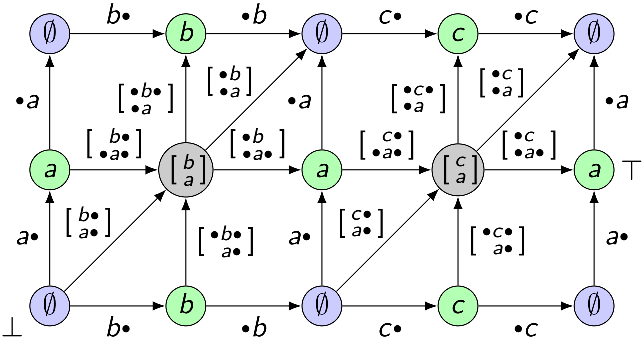
van Glabbeek (TCS 2006): Up to history-preserving bisimilarity, HDAs generalize "the main models of concurrency proposed in the literature"

Lots of recent activity on languages of HDAs:

- Kleene theorem
- Myhill-Nerode theorem
- Büchi-Elgot-Trakhtenbrot theorem
- ...

Higher-dimensional automata





- The operational semantics of an HDA $(X, \bot, \top, \Sigma)$ is the automaton with states X, alphabet $\mathsf{St}_\Sigma \cup \mathsf{Te}_\Sigma$, and transitions

$$E = \{\delta_A^0(\ell) \xrightarrow{A\uparrow \mathrm{ev}(\ell)} \ell \mid A \subseteq \mathrm{ev}(\ell)\} \cup \{\ell \xrightarrow{\mathrm{ev}(\ell)\downarrow A} \delta_A^1(\ell) \mid A \subseteq \mathrm{ev}(\ell)\}.$$

- Here, the language is $\{ \left[\begin{smallmatrix} b\bullet \\ a\bullet \end{smallmatrix}\right] \left[\begin{smallmatrix} \bullet b \\ \bullet a\bullet \end{smallmatrix}\right] \left[\begin{smallmatrix} c\bullet \\ \bullet a\bullet \end{smallmatrix}\right] \left[\begin{smallmatrix} \bullet c \\ \bullet a\bullet \end{smallmatrix}\right] \}{\downarrow}$.

Higher-dimensional timed automata

- In real-time formalisms, everything is synchronous
  - timed automata, timed Petri nets, hybrid automata, etc.
- and concurrency is interleaving

- In formalisms for (non-interleaving) concurrency, no real time
  - same for distributed computing theory
  - (Petri nets have a concurrent semantics; timed Petri nets don't)

- Our goal: formalisms for real-time concurrent systems

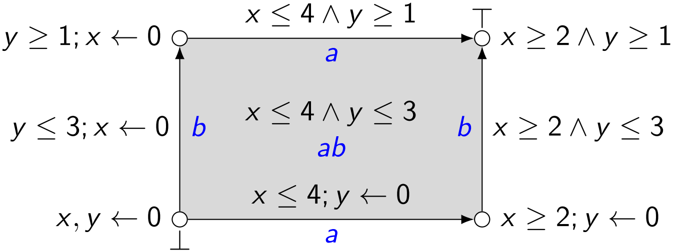
- Here: the marriage between timed and higher-dimensional automata

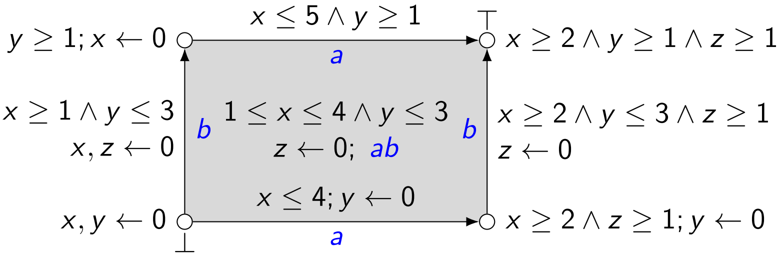
Higher-dimensional timed automata

Definition

An HDTA is a structure $(L, \bot, \top, \Sigma, C, \text{inv}, \text{exit})$, where $(L, \bot, \top, \Sigma)$ is an HDA, C is a finite set of clocks, and $\text{inv} : L \to \Phi(C)$, $\text{exit} : L \to 2^C$ give invariant and exit conditions for each cell.

Intuition:
- $\text{inv}(\ell)$: conditions on the clock values while delaying in ℓ
- $\text{exit}(\ell)$: clocks which are reset to 0 when leaving ℓ.





- The operational semantics of an HDTA X is the infinite automaton with states $X \times \mathbb{R}_{\geq 0}^{C}$, alphabet $\mathsf{St}_\Sigma \cup \mathsf{Te}_\Sigma \cup \mathbb{R}_{\geq 0}$, and transitions

$$E = \{(\ell, v) \xrightarrow{\delta} (\ell, v + \delta) \mid \forall t \in [0, \delta] : v + t \models \mathrm{inv}(\ell)\}$$

$$\cup \{(\delta_A^0(\ell), v) \xrightarrow{A\uparrow \mathrm{ev}(\ell)} (\ell, v') \mid A \subseteq \mathrm{ev}(\ell), v' = v[\mathrm{exit}(\delta_A^0(\ell)) \leftarrow 0]\}$$

$$\cup \{(\ell, v) \xrightarrow{\mathrm{ev}(\ell)\downarrow A} (\delta_A^1(\ell), v') \mid A \subseteq \mathrm{ev}(\ell), v' = v[\mathrm{exit}(\ell) \leftarrow 0]\}.$$

Conclusion

Automata, automata, automata

- useful to provide operational semantics to other models
- well-developed language theory

Timed automata

- useful for modeling and verifying real-time systems
- badly behaved language theory

Higher-dimensional automata

- nice for modeling (and verifying?) concurrent systems
- nice language theory

Higher-dimensional timed automata

- for modeling (and verifying?) real-time concurrent systems