Specifying Urgency in Timed I/O Automata

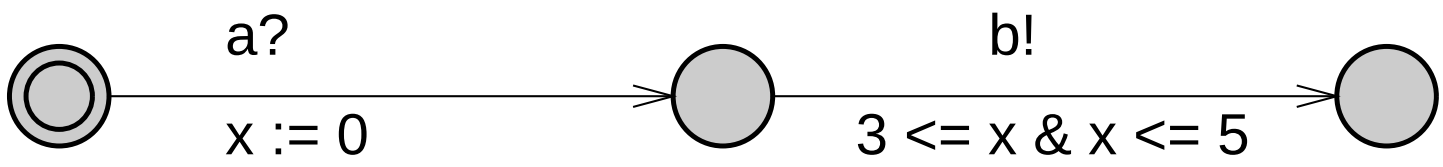
Biniam Gebremichael Frits Vaandrager

Radboud University Nijmegen, The Netherlands

SEFM, Koblenz, Germany, September 7, 2005

Motivation

Alur-Dill style timed automata popular model for real-time systems

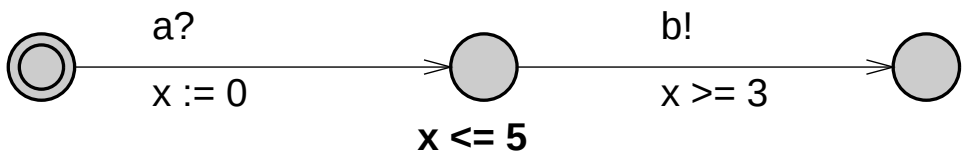


- Finite automata enriched with clock variables $x, y, \ldots$
- All clocks proceed with rate $\mathbf{d}(x) = 1$
- Clocks can be reset and tested

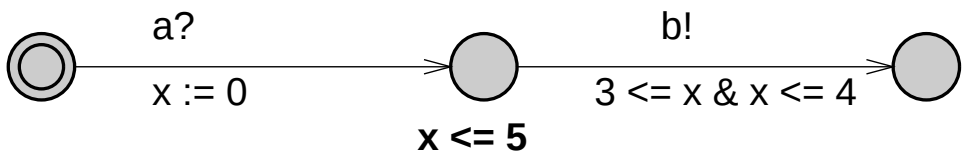
How to specify urgency?

UPPAAL Tool:

Use location invariants to specify urgency



Problem: Global time deadlocks!



How to specify urgency?

Several proposals in the literature

- Progress/urgency
  - Büchi acceptance criterion [Alur and Dill 1994]
  - Location invariant [Alur and Henzinger 1994] as in UPPAAL
  - Deadline on transition [Bornot and Sifakis 1998]
  - Stopping condition [Kaynar et al. 2003]

Goal of this work

➜ Study urgency in setting of TIOA model/language of Lynch et al.

Main source of inspiration: Bornot and Sifakis

Content

Timed I/O Automata (TIOA)

A mathematical framework to model and analyze timed systems [Kaynar, Lynch, Segala and Vaandrager 2003]

- Special class of HIOA without continuous interaction
- Input/Output distinction
- Precondition-effect style
- More expressive than Timed Automata [Alur and Dill 94]
  - No finiteness restrictions
  - More general dynamic behavior

Timed Automata a la Lynch et al

Tuple $\mathcal{A} = (X, Q, \Theta, E, H, \mathcal{D}, \mathcal{T})$

X: internal variables

Q: states, a set of valuations of X

Θ: start states a non-empty subset of Q

E, H: external and internal actions, $A \triangleq E \cup H$

$D \subseteq Q \times A \times Q$: discrete transitions

$\mathcal{T}$: a set of *trajectories* over Q, satisfying certain axioms

Adding Urgency

- $\mathcal{A} = (X, Q, \Theta, E, H, \mathcal{D}, \mathcal{T})$
- $U : Q \times (E \cup H) \rightarrow \{\text{true}, \text{false}\}$ – urgency predicate
- TAU = $(\mathcal{A}, U)$

a is enabled in **x** $\Leftrightarrow$ $\exists \mathbf{x}' : (\mathbf{x}, a, \mathbf{x}') \in D$

a is urgent in **x** $\Leftrightarrow$ a enabled in **x** $\wedge$ $U(\mathbf{x}, a) =$ true

Axioms on trajectories

T0 *Existence of point trajectories*

T1 *Prefix closure*

T2 *Suffix closure*

T3 *Concatenation closure*

T4 *Urgency*: For every $\tau \in \mathcal{T}$, $t \in dom(\tau)$ and $a \in A$:
if a is urgent in $\tau(t)$ then $t = \tau.ltime$

T5 *Maximality*: For every $\tau \in \mathcal{T}$, if τ is maximal and finite then τ is right-closed and some $a \in A$ is urgent in $\tau.lval$

Timed Automata Syntax Example

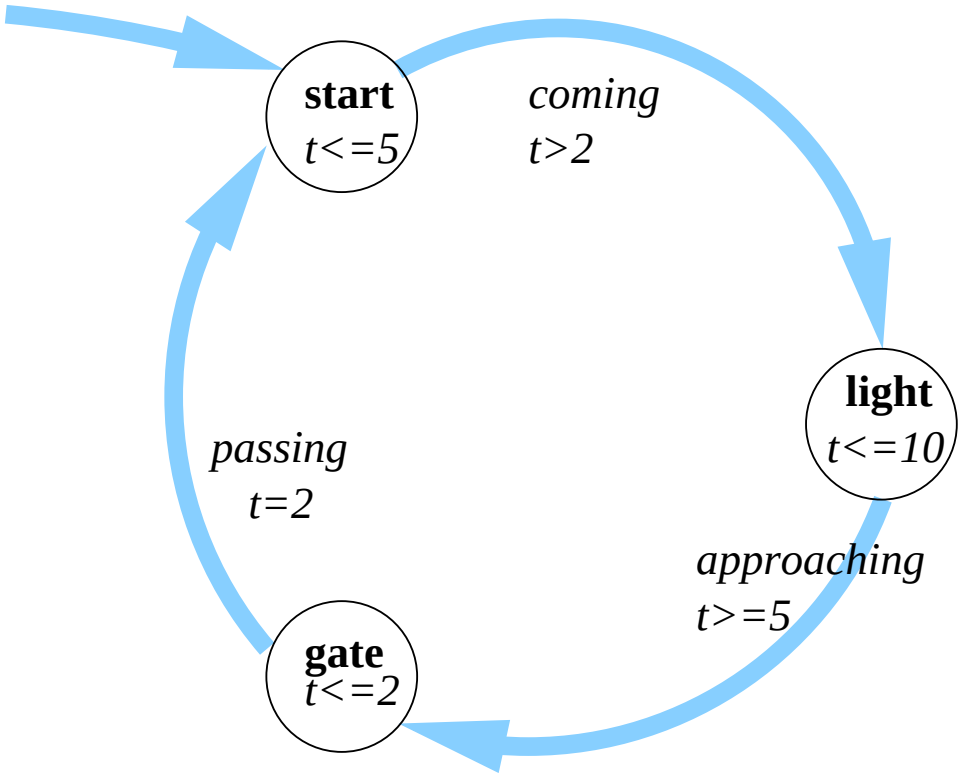
Automaton $Train$

states
- **discrete** $control \in \{start, light, gate\}$ **initially** $start$
- **clock** $t \in \mathsf{R}$ **initially** 0

signature
- **external** $coming, approaching, passing$

transitions
- **external** $coming$
  - **pre** $t > 2 \wedge control = start$
  - **urgent when** $t \geq 5$
  - **eff** $control := light, t := 0$
- **external** $approaching$
  - **pre** $t > 5 \wedge control = light$
  - **urgent when** $t \geq 10$
  - **eff** $control := gate, t := 0$
- **external** $passing$
  - **pre** $t = 2 \wedge control = gate$
  - **urgent when** $true$
  - **eff** $control := start, t := 0$

Timed Automata Syntax Example



Timed Automata Syntax Example

Automaton $TimedChannel(b, M)$

states
- **discrete** $queue$, finite sequence over $M \times R$
- **initially** `empty`
- **analog** $now \in$ R **initially** 0

signature
- **external** $send(m), receive(m)$

transitions
- **external** $send(m)$
  - **eff** $add(m, now + b)$ to the end of the $queue$
- **external** $receive(m, \mathbf{local}\ u)$
  - **pre** (m, u) is first element of $queue$
  - **urgent when** $(now \geq u)$
  - **eff** remove first element of $queue$

Counterexample for T5

automaton $\mathcal{A}$

states	$b : Bool$ **initially** false
	clock x **initially** 0
signature	**external** a
transitions	**external** a
	pre $x > 4 \wedge b =$ false
	urgent when true
	eff $b :=$ true

- Time stops at $x = 4$
- No transition is enabled at $x = 4$

$\Rightarrow$ axiom **T5** does not hold

Proving Axiom T5

For each transition definition tr, let

$b(\vec{h})$

pre $pre(\vec{v}, \vec{h})$

urgent when $urg(\vec{v}, \vec{h})$

eff $eff(\vec{v}, \vec{h}, \vec{v}')$

$$Urg(tr)(\vec{v}, \vec{h}) \quad \triangleq \quad \exists \vec{h} : pre(\vec{v}, \vec{h}) \wedge urg(\vec{v}, \vec{h})$$

Theorem 1 *If predicate* $\bigvee_{tr} Urg(tr)$ *is left-closed then* **T5** *holds*

Counterexample for T5 . . . continues

automaton $\mathcal{A}$

states	$b : Bool$ **initially** false	
	clock x **initially** 0	
signature	**external** $a1, a2$	
transitions	**external** $a1$	**external** $a2$
	pre $x > 4 \wedge b =$ false	**pre** $x = 4 \wedge b =$ false
	urgent when true	**urgent when** true
	eff $b :=$ true	**eff** $b :=$ true

- Time stops at $x = 4$ and $a2$ is enabled

$\Rightarrow$ axiom **T5** holds!

Adding I/O Distinction

A *Timed I/O Automaton* is a Timed Automaton such that:

- External actions partitioned into inputs and outputs
- The following axioms are satisfied

  E0 Input actions not urgent

  E1 Input actions always enabled

  E2 *Time reactivity:*

  Every state $\mathbf{x}$ has outgoing trajectory τ such that either

  - Time can grow to infinity, or
  - A locally controlled action is enabled at the end of τ

Time Reactivity

E0 and **E1** hold if, at the syntactic level, input actions have

- precondition true, and
- urgency predicate false

Theorem 2 *TIOAs with urgency time reactive by construction:*
Each timed I/O automaton with urgency satisfies axiom **E2**

Composition

Timed automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are *compatible* if $H_1 \cap A_2 = H_2 \cap A_1 = \emptyset$ and $X_1 \cap X_2 = \emptyset$

Composition $\mathcal{A}_1 || \mathcal{A}_2$

- enabled: if $a \in A_i$ then $\mathbf{x} \lceil X_i \xrightarrow{a}_i \mathbf{x}' \lceil X_i$
- urgent: $U((\mathbf{x}_1, \mathbf{x}_2), a) = U_1(\mathbf{x}_1, a) \vee U_2(\mathbf{x}_2, a)$

Theorem 3 *If $\mathcal{A}_1$ and $\mathcal{A}_2$ are compatible timed I/O automata then $\mathcal{A}_1 || \mathcal{A}_2$ is a timed I/O automaton*

I/O distinction essential for this result.

Urgency v. Deadline Predicate [Sifakis et al.]

external $a(\vec{h})$	**external** $a(\vec{h})$
pre $pre(\vec{v}, \vec{h})$	**pre** $pre(\vec{v}, \vec{h})$
urgent when $urg(\vec{v}, \vec{h})$	**deadline** $pre(\vec{v}, \vec{h}) \wedge urg(\vec{v}, \vec{h})$
eff $eff(\vec{v}, \vec{h}, \vec{v}')$	**eff** $eff(\vec{v}, \vec{h}, \vec{v}')$

- Urgency predicates are often shorter than deadline predicates
- Deadline predicate is urgency predicate
- Replace $urg(\vec{v}, \vec{h})$ by $pre(\vec{v}, \vec{h}) \wedge urg(\vec{v}, \vec{h})$ then it is a deadline predicate

Urgency v. Stopping Conditions [Kaynar et al.]

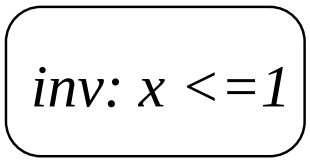
external $a(\vec{h})$
  pre $pre(\vec{v}, \vec{h})$
  urgent when $urg(\vec{v}, \vec{h})$
  eff $eff(\vec{v}, \vec{h}, \vec{v}')$

external $a(\vec{h})$
  pre $pre(\vec{v}, \vec{h})$
  eff $eff(\vec{v}, \vec{h}, \vec{v}')$
  stop when $\bigvee_{tr}(pre(\vec{v}, \vec{h}) \wedge urg(\vec{v}, \vec{h}))$

- Urgency predicates often shorter and more natural than stopping conditions
- Stopping conditions do not ensure time reactivity
- **stop when** = disjunction of all $pre(\vec{v}, \vec{h}) \wedge urg(\vec{v}, \vec{h})$ for all transitions
- **urgent when** = **stop when** if **E2** holds, and simplify formula

Urgency v. Location Invariant (as in UPPAAL)

- Location invariants does not ensure time reactivity

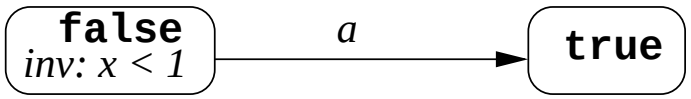


universe stops at $x = 1$

- Urgency predicates stop time for a good reason

Urgency v. Location Invariant

- Location invariants can specify strict upper bound



states $b : Bool$ **initially** false
t : R **initially** $0 \leq t < 1$

transitions **external** a
pre $b =$ false
urgent when $x = t$
eff $b :=$ true

not an elegant solution

- Do we need strict upper bounds in practice?

Urgency Predicate and Model Checking

➜ If a is `urgent channel` in UPPAAL then location [**S3,T3**] has a non-convex zone $(x \geq 10 \wedge y = 5) \vee (y \geq 5 \wedge x = 10)$

solution 1: Remove clock guards

solution 2: Only output actions can be urgent. Input are always enabled and not urgent.

Urgency and UPPAAL

- To avoid non-convexity of zones, UPPAAL imposes rather strong restriction in the syntax of invariants:
  (only conjunction of upper bound on clocks are allowed)

$$inv = \bigwedge_i x_i \leq l_i$$

- Adopt similar restriction on urgency, apply I/O distinction then non-convexity will go away!

$$urg = \bigvee_i x_i \geq l_i$$

Translating Urgency Predicate to Invariant

- Example. **urgent when** $x \geq 5$ translates to
  $inv = \neg(x \geq 5) \vee (x = 5) = x \leq 5$
- In general, if $Urg(tr)$ is stable and left closed.

$$inv \quad = \quad \neg \left(\bigvee_{tr} Urg(tr) \right) \vee LH \left(\bigvee_{tr} Urg(tr) \right),$$

  - inv should hold initially and,
  - after a discrete transition
    $pre(\vec{v}, \vec{h}) \wedge \mathit{eff}\,(\vec{v}, \vec{h}, \vec{v}') \implies inv(\vec{v}')$ holds.

Related Work

- TIOA and deadlines
  - I/O automata with upper and lower bounds associated with tasks [Merritt et al. ’91]
  - deadlines in invariant proofs [Attiya and Lynch 1992]
- TAD used in
  - MoDeST: A Modeling language for Stochastic Timed systems [D’Argenio et al. 2001]
  - Modeling timeouts without time locks [Bowman ARTS’99].
  - Real-time Profile for UML [S. Graf and I. Ober 2003]
- tool: IF - TAD based validation tool for timed systems

Conclusion and Future work

- **Improved TIOA**
  - ➜ Shorter and more natural specs
  - ➜ Ensures time reactivity by construction
  - ➜ Easier to prove state invariants
- **Comparison and translation between different urgency specification styles**
- **Future work**
  - ➜ Receptiveness
  - ➜ Urgency predicates for HIOA
  - ➜ Proof rules for simulation and liveness proofs
  - ➜ Implementing urgency effi ciently in UPPAAL