The Coarsest Congruence for Timed Automata with Deadlines Contained in Bisimulation

Pedro R. D'Argenio[1] and **Biniam Gebremichael**[2]

[1] Universidad de Córdoba, Argentina & Univeristy of Twente, The Netherlands

[2] Radboud University Nijmegen, The Netherlands

CONCUR 05

San Francisco, August 23rd, 2005

Contents

Timed Automata models

1. Timed Automata[Alur & Dill, 1994]

- time progress controlled by invariants on locations
- tools UPPAAL, KRONOS
- several advantages in comparison with other TA models

Timed Automata models

1. Timed Automata[Alur & Dill, 1994]

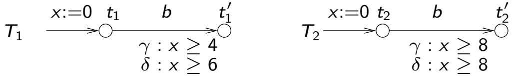
- time progress controlled by invariants on locations
- tools UPPAAL, KRONOS
- several advantages in comparison with other TA models

Limitations:

- only strong synchronization (hard real-time)
  - Why not delayable synchronization
  - Eg. T_1 may wait/ignore/force T_2.
- composition my introduce time deadlock
  - time deadlock is serious problem in TA
  - avoid it by construction (deduce from components)

Timed Automata models

2. Timed Automata with Deadlines [Bornot & Sifakis, 2000]



$tpc(t) = \bigwedge_e(\neg\delta_e)$

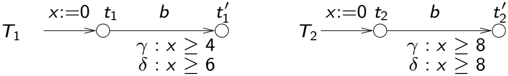
- time progress controlled by deadlines on transitions (deadline implies guard)
- Tools: IF, MoDeST
- strong and delayable synchronization

Gain:

- time deadlock is avoided by construction
- delayable synchronization (several flavors).
- applications: soft real-time, stochastic, performance analysis

Timed Automata models

2. Timed Automata with Deadlines [Bornot & Sifakis, 2000]



$tpc(t) = \bigwedge_e (\neg \delta_e)$

- time progress controlled by deadlines on transitions (deadline implies guard)
- Tools: IF, MoDeST
- strong and delayable synchronization

Gain:

- time deadlock is avoided by construction
- delayable synchronization (several flavors).
- applications: soft real-time, stochastic, performance analysis

Timed Automata models

2. Timed Automata with Deadlines [Bornot & Sifakis, 2000]

$$T_1 \xrightarrow{x:=0} t_1 \xrightarrow[\substack{\gamma\,:\,x \geq 4 \\ \delta\,:\,x \geq 6}]{b} t_1' \qquad T_2 \xrightarrow{x:=0} t_2 \xrightarrow[\substack{\gamma\,:\,x \geq 8 \\ \delta\,:\,x \geq 8}]{b} t_2' \qquad tpc(t) = \bigwedge_e (\neg \delta_e)$$

- time progress controlled by deadlines on transitions (deadline implies guard)
- Tools: IF, MoDeST
- strong and delayable synchronization

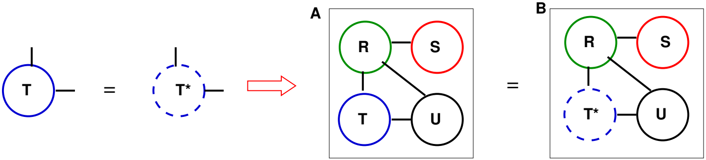
Gain:

- time deadlock is avoided by construction
- delayable synchronization (several flavors).
- applications: soft real-time, stochastic, performance analysis

Lose: strong bisimulation is not congruent

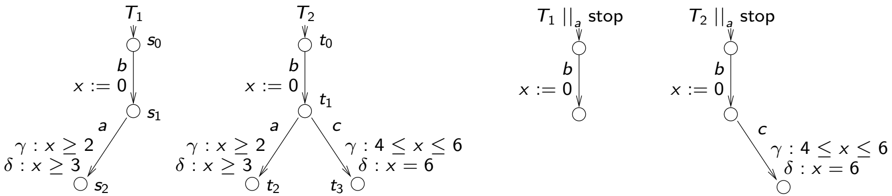
The problem with delayable synchronization

- **Compositionality:** A component can be replaced with behaviorally equivalent component without affecting the big system

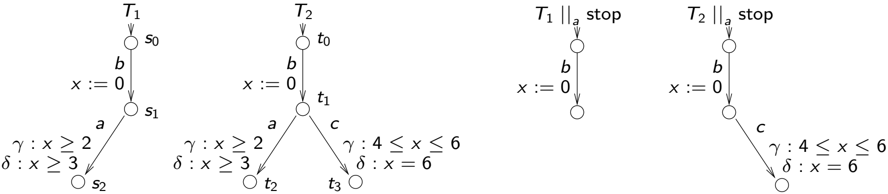


- this does not hold for delayable synchronization in TADs
- even if $T = T*$, A and B may not be equivalent

Example 1

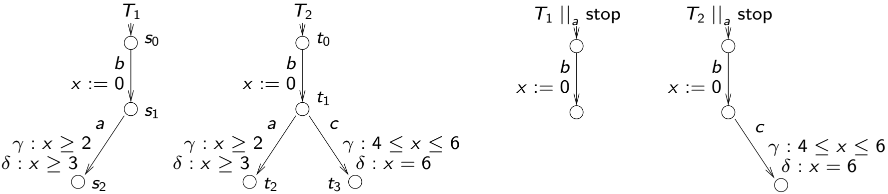

T_1
s_0
b
$x := 0$
s_1
a
$\gamma : x \geq 2$
$\delta : x \geq 3$
s_2
T_2
t_0
b
$x := 0$
t_1
a
c
$\gamma : x \geq 2$
$\delta : x \geq 3$
$\gamma : 4 \leq x \leq 6$
$\delta : x = 6$
t_2
t_3
$T_1 \,||_a$ stop
b
$x := 0$
$T_2 \,||_a$ stop
b
$x := 0$
c
$\gamma : 4 \leq x \leq 6$
$\delta : x = 6$

Example 1



$$T_1 \sim T_2 \quad \text{but} \quad T_1 \,||_a^{\otimes} \text{ stop} \nsim T_2 \,||_a^{\otimes} \text{ stop}$$

Example 1



$$T_1 \sim T_2 \quad \textcolor{red}{\text{but}} \quad T_1 \,||_a^{\otimes} \text{ stop} \not\sim T_2 \,||_a^{\otimes} \text{ stop}$$

Problem: Delayable synchronization reveals hidden behaviors

- but $\sim$ can not see hidden behaviors
- we need a different $\sim$ (a congruent and coarsest $\sim$)

problem already known [Bornot & Sifakis, 2000] but unsolved

The Goal of This Work

Find an equivalence relation R for TADs such that:

1. it is bisimulation $\quad (\subseteq \ \sim)$
2. it is congruent (for parallel composition)
3. it is the coarsest

Timed Bisimulation

Two states are timed bisimilar ($\sim$) if

for any discrete transition or time passage (α)

$$\begin{array}{ccc} s & \xrightarrow{\alpha} & s' \\ \wr & & \\ t & & \end{array} \quad \text{implies } \exists t' \text{ such that} \quad \begin{array}{ccc} s & \xrightarrow{\alpha} & s' \\ \wr & & \wr \\ t & \xrightarrow{\alpha} & t' \end{array}$$

and $\sim$ is symmetric.

Parallel Composition for TADs

$$\frac{s_1 \xrightarrow{a,\gamma,\delta,\mathbf{x}}_1 s_1',\ a \notin B}{(s_1, s_2) \xrightarrow{a,\gamma,\delta,\mathbf{x}} (s_1', s_2) \quad (s_2, s_1) \xrightarrow{a,\gamma,\delta,\mathbf{x}} (s_2, s_1')}$$

$$\frac{s_1 \xrightarrow{a,\gamma_1,\delta_2,\mathbf{x}_1}_1 s_1',\ s_2 \xrightarrow{a,\gamma_2,\delta_2,\mathbf{x}_2}_2 s_2',\ a \in B}{(s_1, s_2) \xrightarrow{a,\gamma_1 \wedge \gamma_2,(\delta_1,\gamma_1)\otimes(\delta_2,\gamma_2),\mathbf{x}_1 \cup \mathbf{x}_2} (s_1', s_2')}$$

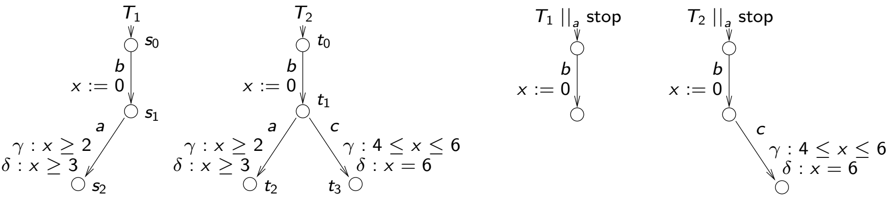
- Synchronization MAY take place when both guards are true
- Synchronization MUST take place when some function ($\otimes$) of the deadlines and the guards is true.
- $\otimes$ – distributive wrt $\vee$, preserves $\delta \Rightarrow \gamma$, preserves left closure.

Parallel Composition for TADs

$$\frac{s_1 \xrightarrow{a,\gamma,\delta,\mathbf{x}}_1 s_1',\ a \notin B}{\begin{array}{c}(s_1, s_2) \xrightarrow{a,\gamma,\delta,\mathbf{x}} (s_1', s_2) \\ (s_2, s_1) \xrightarrow{a,\gamma,\delta,\mathbf{x}} (s_2, s_1')\end{array}} \qquad \frac{s_1 \xrightarrow{a,\gamma_1,\delta_2,\mathbf{x}_1}_1 s_1',\ s_2 \xrightarrow{a,\gamma_2,\delta_2,\mathbf{x}_2}_2 s_2',\ a \in B}{(s_1, s_2) \xrightarrow{a,\gamma_1 \wedge \gamma_2,(\delta_1,\gamma_1)\otimes(\delta_2,\gamma_2),\mathbf{x}_1 \cup \mathbf{x}_2} (s_1', s_2')}$$

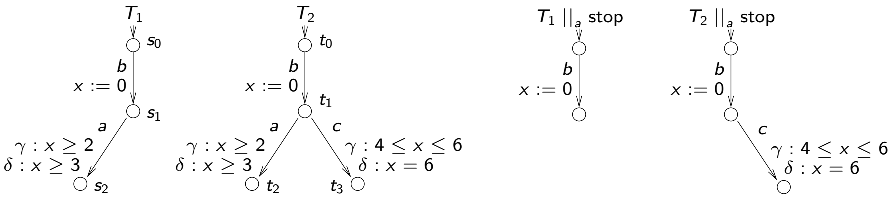
- Synchronization MAY take place when both guards are true
- Synchronization MUST take place when some function ($\otimes$) of the deadlines and the guards is true.
- $\otimes$ – distributive wrt $\vee$, preserves $\delta \Rightarrow \gamma$, preserves left closure.
  - Patient synchronization: ($\delta_1 \wedge \delta_2$)
  - Impatient synchronization ($(\delta_1 \vee \delta_2) \wedge (\gamma_1 \wedge \gamma_2)$)
  - Other guard synchronizations: MAX, MIN, OR.

Example 1 – Revised



$$T_1 \sim T_2 \quad \text{but} \quad T_1 \,||_a^{\otimes}\, \text{stop} \nsim T_2 \,||_a^{\otimes}\, \text{stop}$$

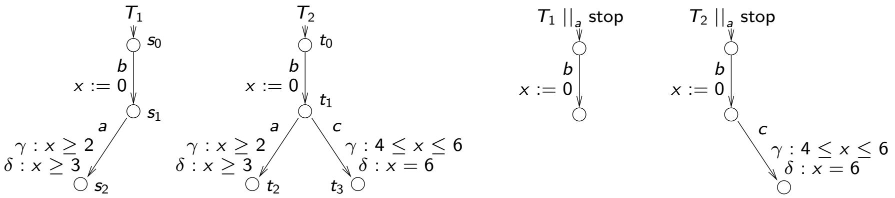
Example 1 – Revised



$$T_1 \sim T_2 \quad \text{but} \quad T_1 \,||_a^{\otimes}\, \text{stop} \nsim T_2 \,||_a^{\otimes}\, \text{stop}$$

Goal: Distinguish T_1 and T_2 – Ask what is after $x = 3$?

Example 1 – Revised



$$T_1 \sim T_2 \quad \text{but} \quad T_1 \,||_a^{\otimes}\, \text{stop} \nsim T_2 \,||_a^{\otimes}\, \text{stop}$$

Goal: Distinguish T_1 and T_2 – Ask what is after $x = 3$?

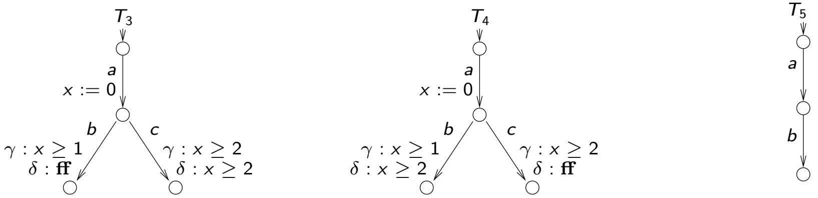
Solution: Allow time to progress beyond *tpc*

$$\text{potential time delay } s\rho \xrightarrow{[d]} s(\rho + d)$$

$T_1 \nsim T_2$ achieved $T_2 = b.3.[1].c$

Example 2

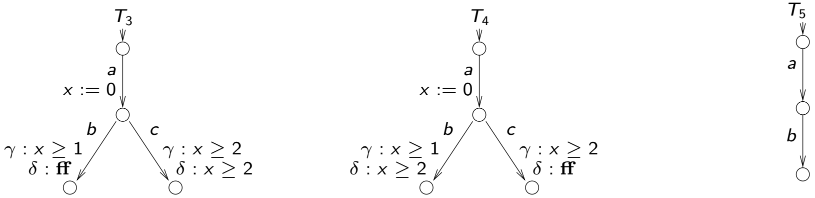
Potential time delay is not enough!



$$T_3 \sim T_4 \quad \text{but} \quad T_3 \mid\mid^{\otimes} T_5 \nsim T_4 \mid\mid^{\otimes} T_5 \qquad (T_3 \mid\mid^{\otimes} T_5 = a.3)$$

Example 2

Potential time delay is not enough!



$$T_3 \sim T_4 \quad \text{but} \quad T_3 \,||^{\otimes}\, T_5 \not\sim T_4 \,||^{\otimes}\, T_5 \qquad (T_3 \,||^{\otimes}\, T_5 = a.3)$$

Problem: When time progressed beyond *tpc*, it is relevant to know whose deadline is dropped (b's or c's).

Solution:

- parametrize potential time delay by a set of actions (D) whose deadlines will have no effect on *tpc*.
- drop transition (∇_D) instead of potential time delay $[d]$.
- $T_3 \not\sim T_3$ achieved $T_3 = a.2.\nabla_{\{c\}}.5$

Semantics of TADs extended with Drop-transitions

- State - was $s\rho$ is $(s, D)\rho$
  - D set of dropped actions
- drop transition: ∇_E – drop the actions in E.

$$(s, D)\rho \xrightarrow{\nabla_E} (s, D \cup E)\rho$$

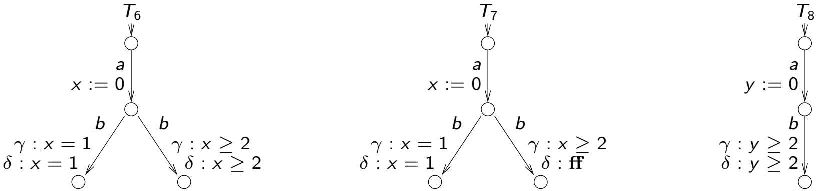
- delay transition: The deadlines associated with the dropped actions have no influence over the *tpc*.

$$tpc(s, D) = \bigwedge\{\neg\delta \mid s \xrightarrow{a,\gamma,\delta,\mathbf{x}} s' \text{ and } a \notin D\}$$

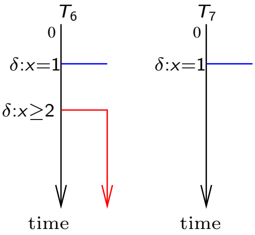
$$\text{delay transition} \quad \frac{\forall d' < d : \rho + d' \models tpc(s, \mathcal{A} - D)}{(s, D)\rho \xrightarrow{d} (s, D)(\rho + d)}$$

Example 3

Once a deadline is dropped it can not be observed again

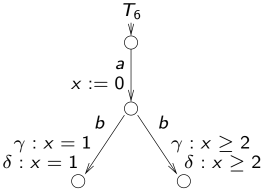


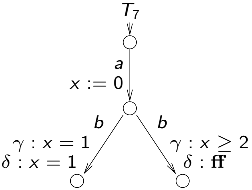
$$T_6 \sim T_7 \quad \text{but} \quad T_6 \mid\mid^{\otimes}_{\mathcal{A}} T_8 \not\sim T_7 \mid\mid^{\otimes}_{\mathcal{A}} T_8$$

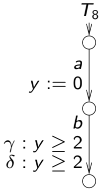


Example 3

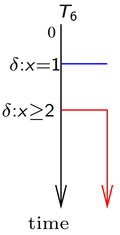
Once a deadline is dropped it can not be observed again

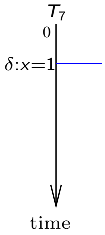






$$T_6 \sim T_7 \quad \text{but} \quad T_6 \,||^{\otimes}_{\mathcal{A}}\, T_8 \nsim T_7 \,||^{\otimes}_{\mathcal{A}}\, T_8$$





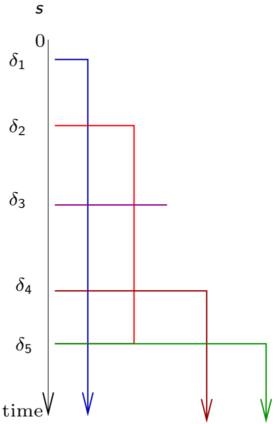
- Solution
  - make dropped deadlines observable again
  - extra undrop action
- $\delta = x \geq 2 \otimes y \geq 2$ versus
- $\delta = \mathbf{ff} \otimes y \geq 2$ – No problem for $\otimes = \vee$

Undrop transition

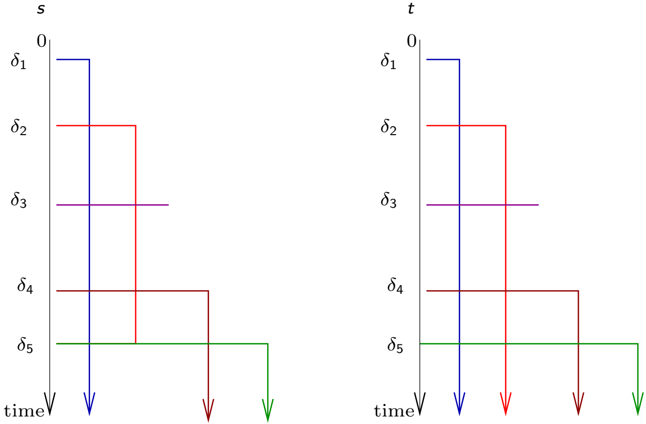
undrop transition: In the future all disregarded deadline will be considered again

$$(s, D)\rho \xrightarrow{\Delta} (s, \varnothing)\rho$$

Example 4


s
0
δ_1
δ_2
δ_3
δ_4
δ_5
time
t
0
δ_1
δ_2
δ_3
δ_4
δ_5
time

Example 4



s can do $s \xrightarrow{\nabla_{\mathcal{A}}}, \xrightarrow{d_5}, \xrightarrow{\Delta}, \xrightarrow{\nabla_{bbg}}, \xrightarrow{d}$ but not t.

Extended Semantics of TAD

Let $\Sigma = \mathcal{A} \cup 2^{\mathcal{A}} \cup \{\Delta\} \cup \mathbb{R}_{\geq 0}$ be the set of actions then $\longrightarrow$ is the smallest relation satisfying

A1: discrete transition

$s \xrightarrow{a,\gamma,\delta,\mathbf{x}} s'$ and $\rho \models \gamma$ implies $(s, D)\rho \xrightarrow{a} (s', \varnothing)\rho\{\mathbf{x}_i{:=}0\}$

A2: delay transition

$\forall d' < d : \ \rho + d' \models tpc(s, \mathcal{A} - D)$ implies

$(s, D)\rho \xrightarrow{d} (s, D)\rho + d$

A3: drop transition – no precondition

$(s, D)\rho \xrightarrow{\nabla_E} (s, D \cup E)\rho$

A4: undrop transition – no precondition

$(s, D)\rho \xrightarrow{\Delta} (s', \varnothing)\rho$

Drop-bisimulation ($\sim^{\nabla}$)

The same as the standard bisimulation except both TADs have to match on drop and undrop actions besides the delay and discrete actions.

$$\mathcal{A} \cup \mathbb{R}_{\geq 0} \quad \mapsto \quad \underbrace{\mathcal{A} \cup \mathcal{A}_{\nabla} \cup \{\Delta\}}_{\textit{discrete action}} \cup \mathbb{R}_{\geq 0}$$

$\sim^{\nabla}$ in terms of $\sim$

$$T_1 \sim^{\nabla} T_2 \Leftrightarrow TS_{\nabla}(T_1) \sim TS_{\nabla}(T_2)$$

What is Drop-bisimulation Good for?

What is Drop-bisimulation Good for?

Results – $\sim^{\nabla}$ is a:

1. bisimulation ($\sim^{\nabla} \subset \sim$)
2. congruent ($T_1 \sim^{\nabla} T_2 \Rightarrow T_1 \,||^{\otimes}\, T_0 \sim^{\nabla} T_2 \,||^{\otimes}\, T_0$)
3. coarsest ($\forall T_0$ if $T_1 \,||^{\otimes}\, T_0 \sim^{\nabla} T_2 \,||^{\otimes}\, T_0$ then $T_1 \sim^{\nabla} T_2$)
4. decidable
   - there is an equivalent symbolic bisimulation which is decidable

What is Drop-bisimulation Good for?

Results – $\sim^{\nabla}$ is a:

1. bisimulation ($\sim^{\nabla} \subset \sim$)
2. congruent ($T_1 \sim^{\nabla} T_2 \Rightarrow T_1 \,||^{\otimes}\, T_0 \sim^{\nabla} T_2 \,||^{\otimes}\, T_0$)
3. coarsest ($\forall T_0$ if $T_1 \,||^{\otimes}\, T_0 \sim^{\nabla} T_2 \,||^{\otimes}\, T_0$ then $T_1 \sim^{\nabla} T_2$)
4. decidable
   - there is an equivalent symbolic bisimulation which is decidable

Theorem 1 $\sim^{\nabla} = \sim^{\phi}$ **(symbolic bisimulation)**

Theorem 2 $\sim^{\phi}$ **is decidable**

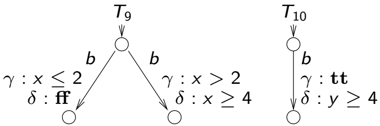
Theorem 3 $\sim^{\phi}$ **is congruent for** $||$

Theorem 4 $\sim^{\nabla}$ **is the coarsest** **[12 pages of proof]**

Symbolic Characterization of Drop-bisimulation

Symbolic Bisimulation ($\sim^{\phi}$) – $s \sim^{\phi} t$ iff

1. $\sim^{\phi}$ is symmetric.
2. ϕ is open ended clock constraint ($\Uparrow$-closed).
3. Every action in $\mathcal{A}$ is simulated by one or more edges labeled with the same action, and the destination locations are bisimilar.



4. Time progress conditions if t and s are equivalent $\forall a \in \mathcal{A}$.
   $\phi \Rightarrow (tpc(t, a) \Leftrightarrow tpc(u, a))$

Drop Bisimulation is Equivalent to Symbolic Bisimulation

Theorem: For an initial clock constraint $\phi_0 \equiv \bigwedge_{x,y \in \mathcal{C}_1 \cup \mathcal{C}_2} (0 \leq x = y)$

$$T_1 \sim^{\phi_0} T_2 \text{ if and only if } T_1 \sim^{\nabla} T_2$$

Theorem: $\sim^{\phi}$ is decidable, so is $\sim^{\nabla}$ **Proof hint:**

- follows from [Lin & Yi 2000 and Čerāns 1992]
- There are only finite regions, and finite $a \in \mathcal{A}$

Proving Congruence of Drop Bisimulation

Theorem: $\sim^{\nabla}$ is congruent for parallel composition

Proof hint:

- first prove congruence on symbolic semantics, then apply $\sim^{\nabla}$ iff $\sim^{\phi}$ (non conventional approach)
- Why not directly prove on the transition system?
  - Defining parallel composition on the transition system is very complex
    - Needs complex bookkeeping to know which deadline is blocking time progress
    - Commit to one instance of $\otimes$

Theorem: $\sim^{\phi}$ is congruent for parallel composition

$$T_1 \sim^{\phi} T_2 \; and \; T_3 \sim^{\phi} T_4 \; implies \; T_1 \, ||^{\otimes} \, T_3 \sim^{\phi} T_2 \, ||^{\otimes} \, T_4$$

The same holds for $\sim^{\nabla}$.

Proving Coarsest Congruence

Theorem: $\sim^{\nabla}$ is the coarsest congruent for parallel composition

$$\forall T_0 : \; \textit{if} \;\; T_1 \,||_B^{\otimes}\, T_0 \sim^{\nabla} T_2 \,||_B^{\otimes}\, T_0 \;\; \textit{then} \;\; T_1 \sim^{\nabla} T_2$$

proof hint: by contradiction. Construct a test automaton T_t that distinguishes $T_1 and T_2$.

The test automaton has transitions, similar to the drop and undrop actions of the extended semantics

$$s_D \xrightarrow{a,\mathbf{tt},\mathbf{0}_\delta,\varnothing} s_\varnothing \qquad s_D \xrightarrow{\nabla_{D'},\mathbf{tt},\mathbf{ff},\varnothing} s_{D\cup D'} \qquad s_D \xrightarrow{\Delta,\mathbf{tt},\mathbf{ff},\varnothing} s_\varnothing$$

Which Synchronization Operations are Supported by $\sim^{\nabla}$

$$\frac{s_1 \xrightarrow{a,\gamma,\delta,\mathbf{x}}_1 s_1',\ a \notin B}{(s_1, s_2) \xrightarrow{a,\gamma,\delta,\mathbf{x}} (s_1', s_2) \quad (s_2, s_1) \xrightarrow{a,\gamma,\delta,\mathbf{x}} (s_2, s_1')} \qquad \frac{s_1 \xrightarrow{a,\gamma_1,\delta_2,\mathbf{x}_1}_1 s_1',\ s_2 \xrightarrow{a,\gamma_2,\delta_2,\mathbf{x}_2}_2 s_2',\ a \in B}{(s_1, s_2) \xrightarrow{a,\gamma_1 \oplus \gamma_2, (\delta_1,\gamma_1) \otimes (\delta_2,\gamma_2), \mathbf{x}_1 \cup \mathbf{x}_2} (s_1', s_2')}$$

- **Synchronizing guards** $\gamma_1 \oplus \gamma_2$

AND: both guards true ($\gamma_1 \wedge \gamma_2$). supported by $\sim^{\nabla}$

OR: one guard true ($\gamma_1 \vee \gamma_2$).

MIN: one guard true, the second guard will be true in the future (the faster forces the slower)

MAX: one guard true, the second guard was true in the past. (the faster waits the slower). Can be expressed in terms of AND.

- **Synchronizing deadlines**

Which Synchronization Operations are Supported by $\sim^{\nabla}$

- **Synchronizing deadlines** $(\delta_1, \gamma_1) \otimes (\delta_2, \gamma_2)$
  - any $\otimes$ that,
    - is distributive wrt $\vee$,
    - preserves $\delta \Rightarrow \gamma$,
    - preserves left closure,
    - has identity deadline

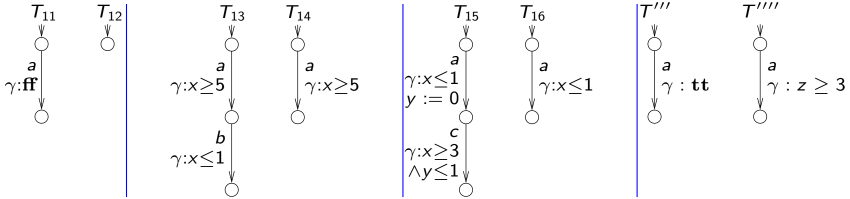
Patient: both deadlines true $\delta_1 \wedge \delta_2$,
Impatient: one deadline true and both guards true $(\delta_1 \vee \delta_2) \wedge (\gamma_1 \wedge \gamma_2)$.
Strong: one deadline true $(\delta_1 \vee \delta_2)$ (does not preserve $\delta \Rightarrow \gamma$)

Conclusion

- **Summary:**
  We have characterized the coarsest congruence relation that is included in the bisimulation relation for Timed Automata with Deadlines. An equivalent symbolic bisimulation is also characterized and proved to be decidable.
- **Related work:**
  - Huimin Lin & Wang Yi (2002) have done similar symbolic characterization for Timed Automata with Invariants.
  - Timed IO Automata with Urgency [Gebremichael & Vaandrager, 2004] solves the problem of delayable synchronization and parallel composition by IO distinction.
- **Future work:**
  - Axiomatization of Timed Automata with Deadlines.

APPENDIX: Examples on synchronizing Guards

- **OR:** $T_{11} \,||_a^{\otimes}\, T'''$ can do a but not $\not\sim^{\nabla}$ $T_{12} \,||_a^{\otimes}\, T'''$
- **MIN:** $(\gamma_1 \wedge \gamma_2 \Downarrow) \vee (\gamma_2 \wedge \gamma_1 \Downarrow)$. in $T_{13} \,||_a^{\otimes}\, T'''$ action b is possible but not in $T_{14} \,||_a^{\otimes}\, T'''$
- **MAX:** $(\gamma_1 \wedge \gamma_2 \Uparrow) \vee (\gamma_2 \wedge \gamma_1 \Uparrow)$. in $T_{15} \,||_a^{\otimes}\, T''''$, a can be delayed until $z > 3$ and c will be possible. remove $\gamma : x < 1$ from T_{15} to express MAX in AND.



$T_{11} \sim^{\nabla} T_{12}$ and $T_{13} \sim^{\nabla} T_{14}$