State Exploration for Real-Time

Martin Fränzle

Carl von Ossietzky Universität

Dpt. of CS

Res. Grp. Hybrid Systems

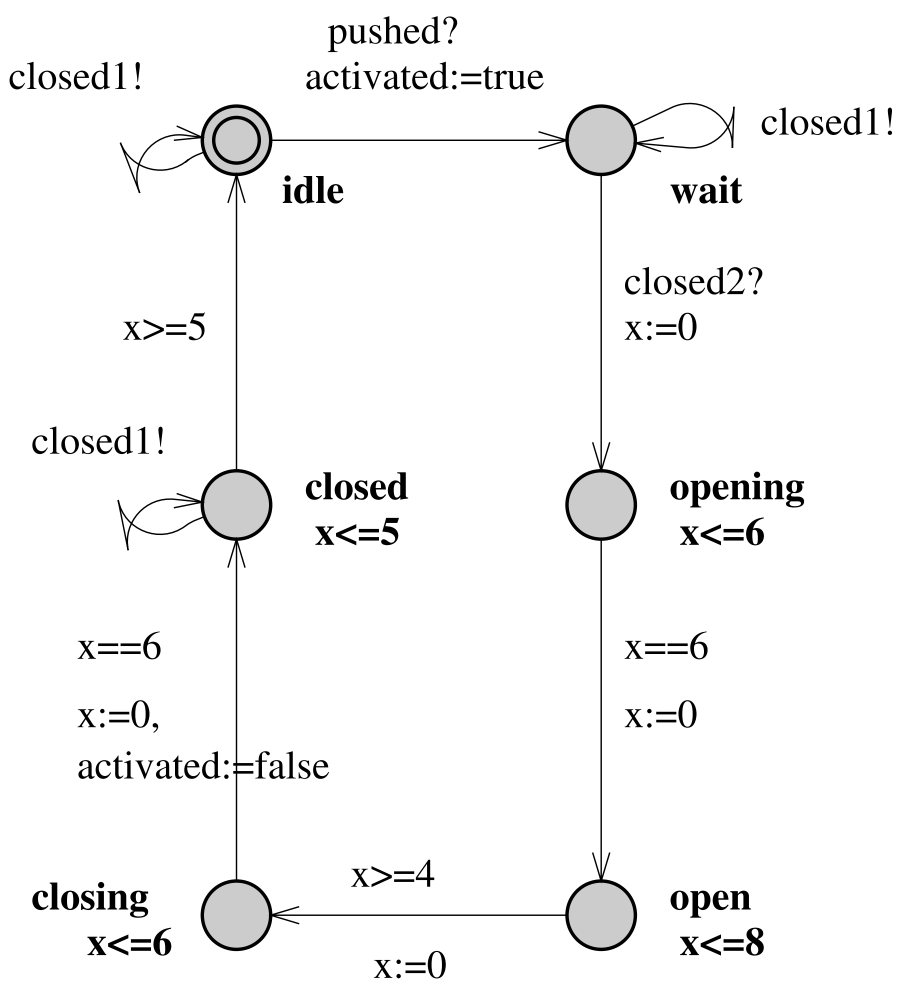
Oldenburg, Germany

What you'll learn

- Alur-Dill timed automata:
  - The model
  - use in verification
  - finitariness: clock regions
- Clock zones as a symbolic representation for TA states:
  - represent (certain) convex unions of clock regions, avoiding the exponential blowup of the region construction
- Difference bound matrices (DBMs)
  - a practical representation of clock zones

Timed Transition Systems

Example



- Stays in state `opening` for exactly 6 time units,
- stays in state `open` between 4 and 8 time units,
- stays in state `closing` for exactly 6 time units,
- stays in state `closed` for exactly 5 time units,
- stays in all other states arbitrarily long.

$\Rightarrow$ Parallel composition is not transition-synchronous!

Formal setup

A timed transition system $TTS = (V, E, L, T, \alpha, G, R, Inv, I)$ over a set C of clocks and alphabet Σ has

- a set V of vertices (interpreted as discrete system states, a.k.a. locations),
- a set E of edges (interpreted as possible transitions),
- $L \in V \to \mathcal{P}(AP)$ labels the vertices with atomic propositions that apply in the individual vertices,
- $I \subseteq V$ is a set of initial states,
- $T : E \to (V \times V)$ maps edges to location changes,
- $\alpha : E \to \Sigma$ assigns a communication to transitions,
- $G : E \to \mathcal{P}(ClockVal)$ gives conditions for a transition to be taken,
- $R : E \to \mathcal{P}(C)$ states the clocks to be reset upon a transition,
- $Inv : V \to \mathcal{P}(ClockVal)$ yields state invariants denoting when a state may be held,

where $ClockVal = C \to \mathbb{R}_{\geq 0}$.

Runs of TTS

Given a TTS $(V, E, L, T, \alpha, G, R, Inv, I)$, a run r of the TTS is

- an alternating sequence $(v_0, c_0) \stackrel{(e_0,t_0)}{\longrightarrow} (v_1, c_1) \stackrel{(e_1,t_1)}{\longrightarrow} \ldots$ of
  1. state/clock-valuation pairs $(v_i, c_i) \in V \times ClockVal$,
  2. transition/time pairs $(e_i, t_i) \in E \times \mathbb{R}_{\geq 0}$
- with non-decreasing time stamps: $t_i \leq t_{i+1}$ for each i
- that starts in an initial state: $v_0 \in I$ and $c_0 \equiv 0$
- and is state-transition-consistent: $T(e_i) = (v_i, v_{i+1})$ for each i
- and satisfies the transition guards: $c_i + (t_i - t_{i-1}) \in G(e_i)$ for each i, where $c + t(x) = c(x) + t$ for each clock x and $t_{-1} = 0$,
- and invariably satisfies the state invariants: $c_i + t \in Inv(v_i)$ for each i and each t with $0 \leq t \leq t_i - t_{i-1}$
- and obeys clock resets: $c_{i+1}(x) = \begin{cases} c_i(x) + (t_i - t_{i-1}) & \text{iff } x \notin R(e_i) \\ 0 & \text{iff } x \in R(e_i) \end{cases}$

  for each i and each clock x.

The quest

- The set of states of a TTS is $V \times ClockVal$.
- It is infinite, as $ClockVal = C \to \mathbb{R}_{\geq 0}$.
- Naive forward or backward (on the fly or symbolic) state coloring algorithms need not terminate.

Is reachability analysis etc. nevertheless mechanizable?

Simple clock constraints

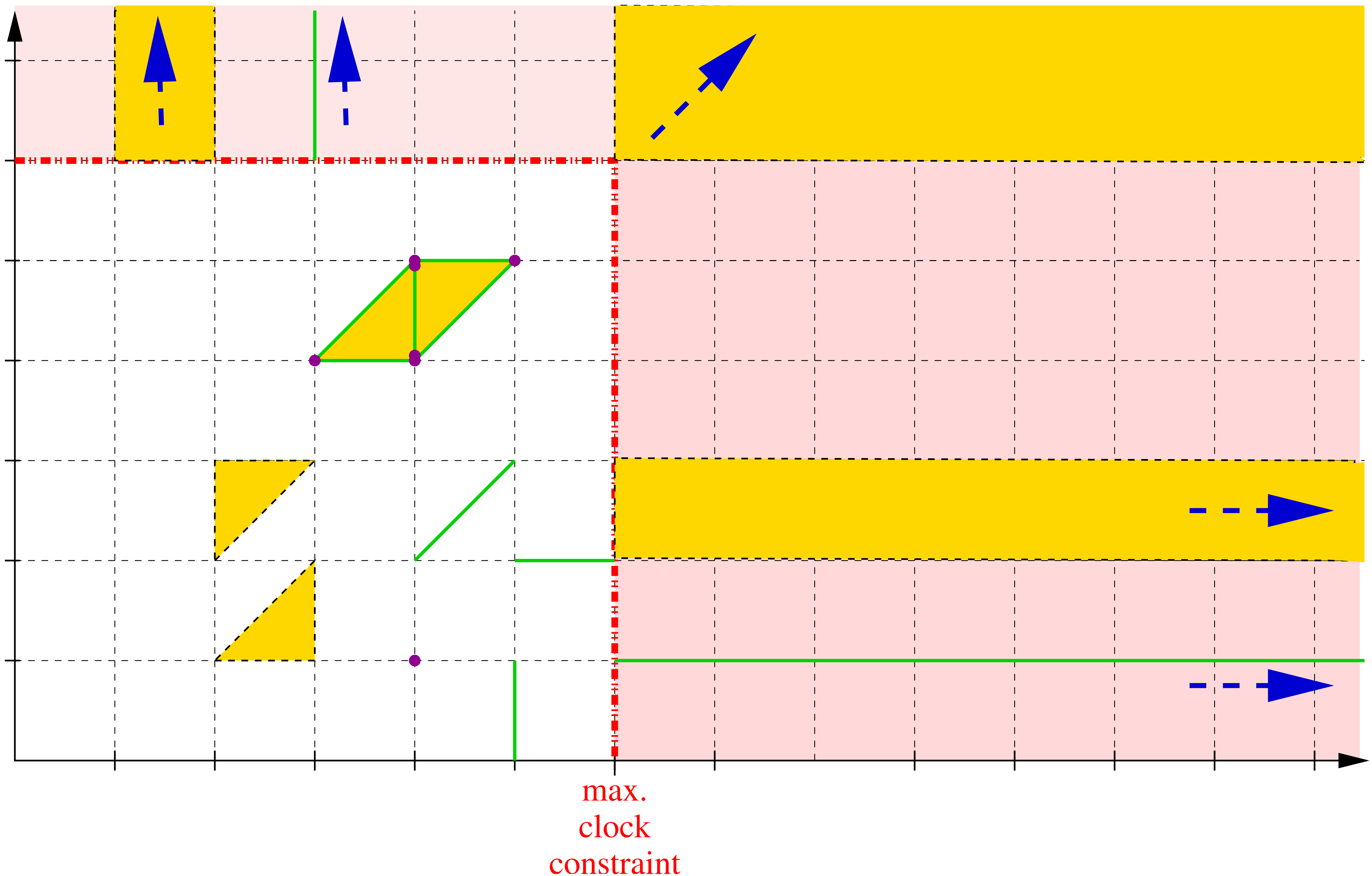
A clock constraint is simple iff

- it is of the form $x \sim k$, where x is a clock, k an integer constant, and $\sim$ one of $<, \leq, =, \geq, >$
- a conjunction of such simple constraints.

From now on, we will concentrate on TTS where

- all guards are simple,
- all invariants are simple.

Clock regions



Time-abstract bisimulation

A time-abstract bisimulation between two TTS is a relation

$$\sim \subset (V \times \mathrm{ClockVal}) \times (V' \times \mathrm{ClockVal}')$$

s.t. for each $(v, c) \sim (v', c')$:

1. if there is $(v_1, c_1) \in V \times \mathrm{ClockVal}$ and $(e, t) \in E \times \mathbb{R}_{\geq 0}$ s.t.

$$(v, c) \stackrel{(e,t)}{\longrightarrow} (v_1, c_1)$$

then there is $(v_1', c_1') \in V' \times \mathrm{ClockVal}'$ and $(e', t') \in E' \times \mathbb{R}_{\geq 0}$ s.t.

$$(v', c') \stackrel{(e',t')}{\longrightarrow} (v_1', c_1') \quad \text{and} \quad \alpha(e) = \alpha(e') \quad \text{and} \quad (v_1, c_1) \sim (v_1', c_1')$$

N.B.: t and t' are not related! $\rightsquigarrow$ time abstraction.

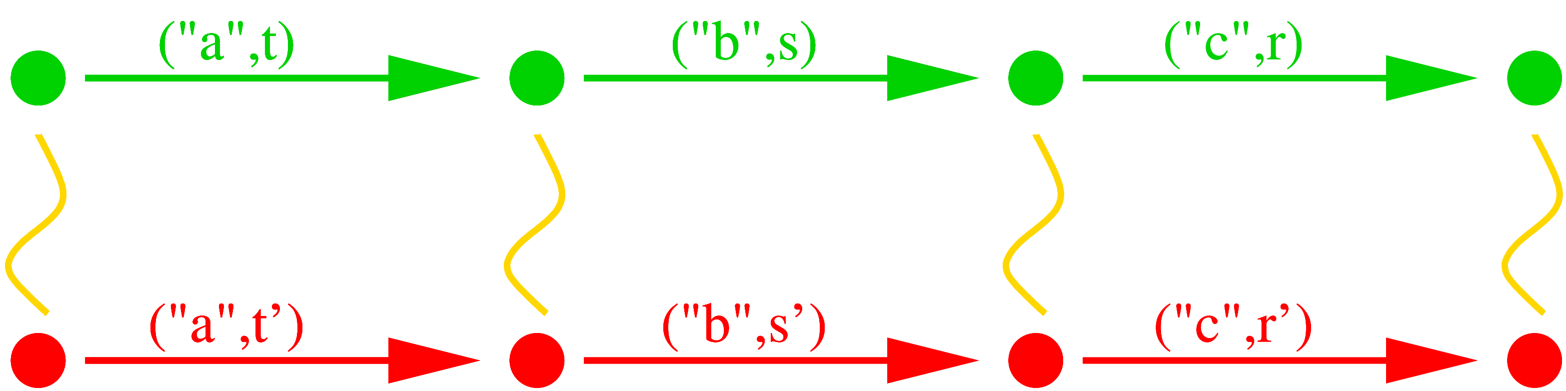
Time-abstract bisimulation (cntd.)

2. if there is $(v'_1, c'_1) \in V' \times \mathrm{ClockVal}'$ and $(e', t') \in E' \times \mathbb{R}_{\geq 0}$ s.t.

$$(v', c') \stackrel{(e',t')}{\longrightarrow} (v'_1, c'_1)$$

then there is $(v_1, c_1) \in V \times \mathrm{ClockVal}$ and $(e, t) \in E' \times \mathbb{R}_{\geq 0}$ s.t.

$$(v, c) \stackrel{(e,t)}{\longrightarrow} (v_1, c_1) \quad \text{and} \quad \alpha(e) = \alpha(e') \quad \text{and} \quad (v_1, c_1) \sim (v'_1, c'_1)$$



States in the $\sim$ relation follow similar (same labels, different timing) traces.

Clock regions vs. time-abstract bisimulation

Thm.: If $\sim$ is a time-abstract bisimulation *on a TTS* s.t. $\sim$ does only relate identical vertices (yet with potentially different clock val.s) and if $(v, c) \sim (v', c')$ then a vertice $w \in V$ is reachable from (v, c) iff w is reachable from (v', c').

Thm.: For any TTS, the relation $\sim \subset (V \times \mathrm{ClockVal}) \times (V \times \mathrm{ClockVal})$ defined by $(v, c) \sim (v', c')$ iff

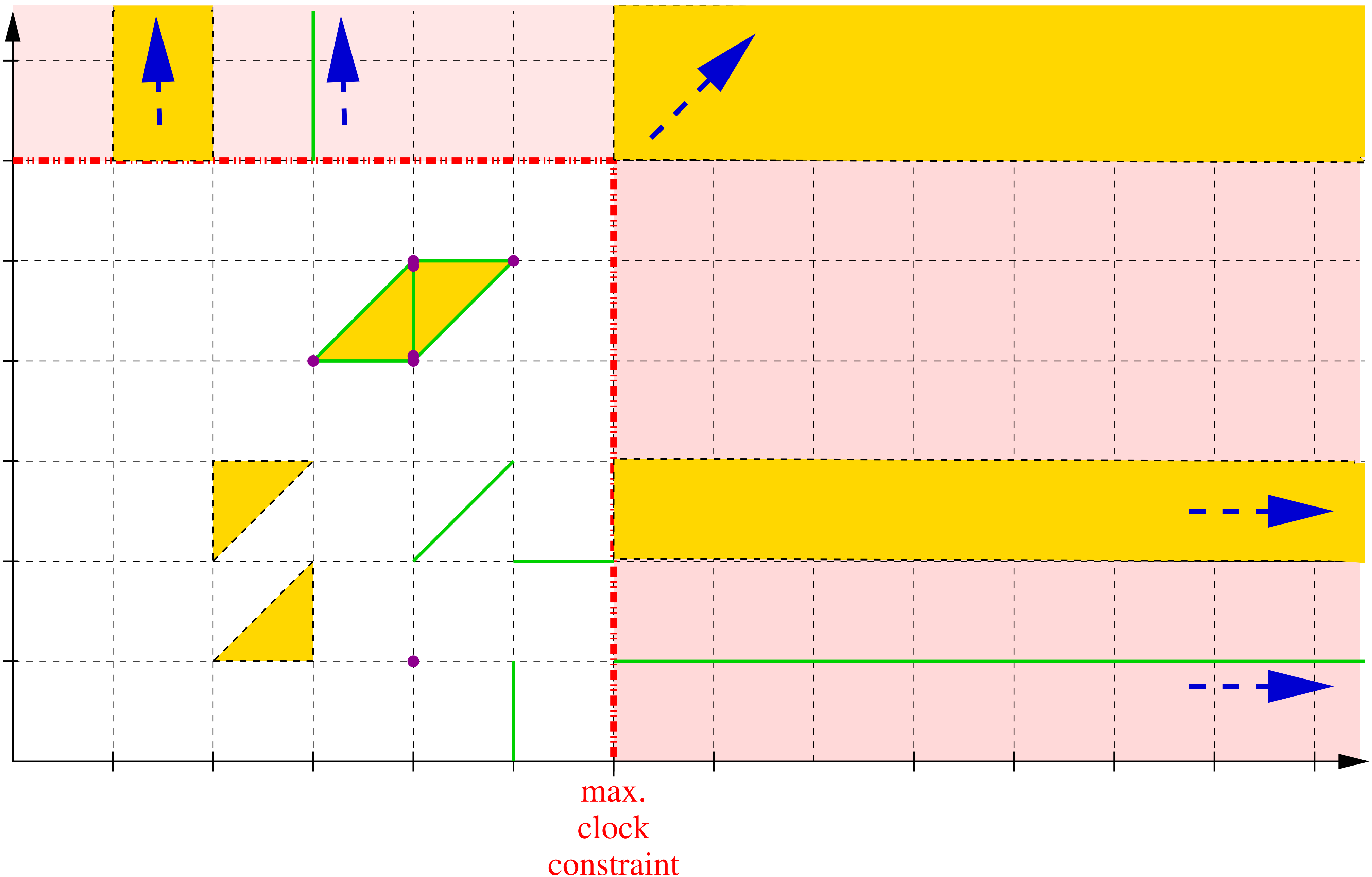
1. $v = v'$,
2. F.e. clock x, $\lfloor c(x) \rfloor = \lfloor c'(x) \rfloor$ or $c(x) > mc < c'(x)$,
3. F.e. clock x, $\mathrm{fract}(c(x)) = 0 \iff \mathrm{fract}(c'(x)) = 0$ or $c(x) > mc < c'(x)$,
4. F.e. clock x, y, $\mathrm{fract}(c(x)) \leq \mathrm{fract}(c(y)) \iff \mathrm{fract}(c'(x)) \leq \mathrm{fract}(c'(y))$ or $\ldots$

is a time-abstract bisimulation on the TTS (i.e., between the states of just that one TTS).
(mc is the maximum time constant in the TTS.)

Cor.: Wrt. vertex reachability (and other time-abstract notions like existence of time-abstract traces), states in the above $\sim$ relation are indistinguishable.

Obs.: For any TTS, there are only finitely many equivalence classes wrt. $\sim$.

Equivalence classes of $\sim$



The region automaton

Given the TTS $= (V, E, L, T, \alpha, G, R, Inv, I)$, we define its region "automaton" (like the TTS, it actually lacks an acceptance condition) to be the finite Kripke structure

$\mathcal{A}_{\mathsf{TTS}} = ([V \times ClockVal]_{\sim}, \rightarrow, L', [I \times \{x \mapsto 0\}]_{\sim})$ with

- $x \rightarrow y$ iff there is $(v, c) \in x$, $(v', c') \in y$, $t \geq 0$, and $e \in E$ s.t. $(v, c) \stackrel{(e,t)}{\rightarrow} (v', c')$
- $L'([v, c]) = L(v)$.

- This is a finite Kripke structure that can be subjected to CTL model-checking etc.
- but its size is exponential in the number of clocks: #regions $= |C|! \cdot 2^{|C|} \cdot \prod_{c \in C}(2\max(c) + 2)$
- Can we do the state-space traversal more symbolicly, representing *sets* of regions by predicates?

Clock zones

Clock zones

A clock zone is the set of satisfying assignments in $\mathbb{R}^n_{\geq 0}$ of a conjunction of

- inequations that compare a clock to an integer constant and
- inequations that compare the difference of two clocks to an integer constant.

By introduction of a dedicated clock x_0 representing the value 0, difference logic formulae of the specific conjunctive form

$$\begin{aligned}
\phi &::= \bigwedge_{x \in C} (x_0 - x \leq 0) \;\wedge\; \bigwedge_{i=1}^{n} \psi_i \\
\psi_i &::= c_{i1} - c_{i2} \sim_i k_i \\
\sim_i &::= < \mid \leq \\
k_i &::\in \mathbb{Z}
\end{aligned}$$

form an appropriate symbolic representation of clock zones.

Closure properties of clock zones

If ϕ and ψ are symbolic representations of clock zones and $d \in \mathbb{N}$ then symbolic representations

- $\phi \wedge \psi$ for zone intersection: $[\![\phi \wedge \psi]\!] \stackrel{\text{def}}{=} \{\vec{x} \in \mathbb{R}^n_{\geq 0} \mid \vec{x} \models \phi \text{ and } \vec{x} \models \psi\}$
- $\exists x_i.\phi$ for clock hiding:

$$[\![\exists x_i.\phi]\!] \stackrel{\text{def}}{=} \left\{ (x_1, \ldots, x_n) \;\middle|\; \begin{array}{l} \text{there is } y \in \mathbb{R}_{\geq 0} \text{ s.t.} \\ (x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n) \models \phi \end{array} \right\}$$

- $\phi[x_i := 0]$ for clock reset: $[\![\phi[x_i := 0]]\!] \stackrel{\text{def}}{=} [\![x_i = 0 \wedge \exists x_i.\phi]\!]$
- $\phi \uparrow$ for elapse of time:

$$[\![\phi \uparrow]\!] \stackrel{\text{def}}{=} \{(x_1 + \delta, \ldots, x_n + \delta) \mid (x_1, \ldots, x_n) \models \phi, \delta \in \mathbb{R}_{\geq 0}\}$$

can be obtained effectively.

TA reachability using zones: the idea

1. Represent reachable state sets by lists of pairs of locations and clock zones $\langle(l_1, z_1), \ldots, (l_m, z_m)\rangle$,
2. for such a pair, compute the set $Post_t(l, z)$ of successors under a transition t with $T(t) = (l, l')$ by
   - let time elapse starting from z: $\phi_1 = z\uparrow$ represents states reachable under arbitrary passage of time
   - intersect ϕ_1 with $Inv(l)$: $\phi_2 = \phi_1 \wedge Inv(l)$ reflects states reachable through time passage consistent with the location invariant (N.B.: invariant is convex due to simplicity)
   - intersect ϕ_2 with guard $G(t)$: $\phi_3 = \phi_2 \wedge G(t)$ reflects states reachable through time passage which enable the transition t
   - reset the clocks in $R(t)$: $\phi_4 = \phi_3[r_1 := 0] \ldots [r_j := 0]$, where $\{r_1, \ldots, r_j\} = R(t)$, reflects the clock readings after performing t's resets
   - intersect with the target loc.'s invariant: $\phi_5 = \phi_4 \wedge Inv(l')$
   - do the location change: $Post_t(l, z) = (l', \phi_5)$.

The state-space exploration

1. Start with the state list $R_0 = I \times \{\text{``}\bigwedge_{x \in C}(x_0 - x \leq 0) \wedge \bigwedge_{x \in C}(x - x_0 \leq 0)\text{''}\}$.
2. Repeat
   (a) select $(l_i, z_i) \in R_k$ and $t \in E$ with source l_i s.t. $Post_t(l_i, z_i)$ is not already subsumed by R_k,
   (b) build $R_{k+1} = R_k \cdot \langle Post_t(l_i, z_i) \rangle$

   until no such $(l_i, z_i) \in R_k$ and $t \in E$ is found.

N.B. Subsumption test can be performed at various levels of detail.

The problem

- Iterating $\mathrm{Post}_t(l, z)$ for all pairs (l, z) in the list of reachable states and all transitions need not terminate:

- In the region graph, we solved the problem by not distinguishing clock readings above the max. clock constant.
- We can achieve a similar effect by widening zones that extend beyond the max. clock constant:
  - Any constraint of the form $x_i - x_j \sim l$ with $l > maxconstant$ is removed from the symbolic representation when it arises.

Difference Bound Matrices

Difference Bound Matrices

Difference bound matrices (DBMs) are a canonizable representation for *conjunctive* formulae in difference logic

$$\begin{aligned} \phi &::= \bigwedge_{i=1}^{n} \psi_i \\ \psi_i &::= c_{i1} - c_{i2} \sim_i k_i \\ \sim_i &::= < \mid \leq \\ k_i &::\in \mathbb{Z} \end{aligned}$$

Given a finite clock set C (in practice containing the pseudo-clock x_0), a DBM M over C is a mapping

$$\underbrace{(C \times C)}_{\text{clock pairs}} \rightarrow \left(\underbrace{\{<, \leq\} \times \mathbb{Z}}_{\text{constraint on diff.}} \cup \underbrace{\{(<, \infty)\}}_{\text{unconstrained}} \right) .$$

Encoding: $M(x, y) = (\sim, k) \;\triangleq\; x - y \sim k$

Implied constraints and tightening

Observation: $x - y \sim_1 k_1$ and $y - z \sim_2 k_2$ implies $x - z \sim k_1 + k_2$, where

$$\sim = \begin{cases} \sim_1 & \text{iff } \sim_1 = \sim_2 \\ < & \text{otherwise.} \end{cases}$$

Consequence: A DBM may contain constraint pairs which imply constraints that are tighter than the recorded constraints: $M(x, y) = (\sim_1, k_1) \wedge M(y, z) = (\sim_2, k_2) \wedge M(x, z) = (\sim, k)$ and

1. $k > k_1 + k_2$ or
2. $k = k_1 + k_2$ but $\sim = \leq$, yet $\sim_1 = <$ or $\sim_2 = <$.

Solution: *Tighten the DBM* by replacing the constraint by the stronger implied constraint.
Repeat this until no implied constraint stronger than a recorded constraint remains. This brings the DBM into a *canonical form*.
Such canonization of DBMs can be done in cubic time using the *Floyd-Warshall algorithm*.

Properties of canonical DBMs

Thm: A *canonical* DBM is unsatisfiable iff there is some $x \in C$ such that $M(x, x) = (<, 0)$ or $M(x, x) = (\sim, k)$ with $k < 0$.

Cor: Satisfiability test of *canonical* DBMs runs in $O(|C|)$ time.

Operations on clock zones using ca. DBMs

Intersection:

$$M \wedge N(x,y) = \begin{cases} M(x,y) & \text{if } M(x,y) \text{ is tighter than } N(x,y) \\ N(x,y) & \text{otherwise} \end{cases}$$

Clock reset: When the dedicated clock variable x_0 is used,

$$M[z := 0](x,y) = \begin{cases} M(x,y) & \text{if } x \neq z \text{ and } y \neq z \\ M(x,x_0) & \text{if } x \neq z \text{ and } y = z \\ M(x_0,y) & \text{if } x = z \text{ and } y \neq z \\ (\leq, 0) & \text{if } x = y = z \end{cases}$$

Note that canonicity saves an explicit quantifier elimination as the implied constraints are already in place!

These operations do not preserve canonicity!

Operations on clock zones using can. DBMs

Elapse of time: When the dedicated clock variable x_0 is used,

$$M \uparrow (x, y) = \begin{cases} M(x, y) & \text{if } x = x_0 \text{ or } \quad y \neq x_0 \\ (<, \infty) & \text{if } x \neq x_0 \text{ and } y = x_0 \end{cases}$$

Widening: When the maximum clock constant is k,

$$\widetilde{M}(x, y) = \begin{cases} M(x, y) & \text{if } M(x, y) = (\sim, l) \text{ with } |l| \leq |k| \\ (<, \infty) & \text{otherwise} \end{cases}$$

Pros and cons

- Zone-based reachability analysis usually is explicit wrt. discrete locations:
  - maintains a list of location-zone pairs or
  - maintains a list of location-DBM pairs
  - ☹ confined wrt. size of discrete state space
  - ☺ avoids blowup by number of clocks and size of clock constraints through symbolic representation of clocks
- Region-based analysis provides a finite-state abstraction, amenable to finite-state symbolic MC
  - ☺ less dependent on size of discrete state space
  - ☹ exponential in number of clocks