III. Timed Automata.

Notiztitel 09.03.2005

III. 1. Syntax

Kripke-Structure:	States	Locations	L
Timed automaton:	Start states	Start locations	L_0
	Labelling function	✓	I
	Transition relation.	✓	$\longrightarrow$
		Clocks	C
		constraints	CC
		resets.	

TA: $(L, L_0, I, \rightarrow, C, i_{tr}, r)$

$I: L \rightarrow 2^{AP}$ $\quad \rightarrow \subseteq L \times L$

$r: (\rightarrow) \rightarrow 2^C$

$i_{tr}: (\rightarrow) \rightarrow CC$

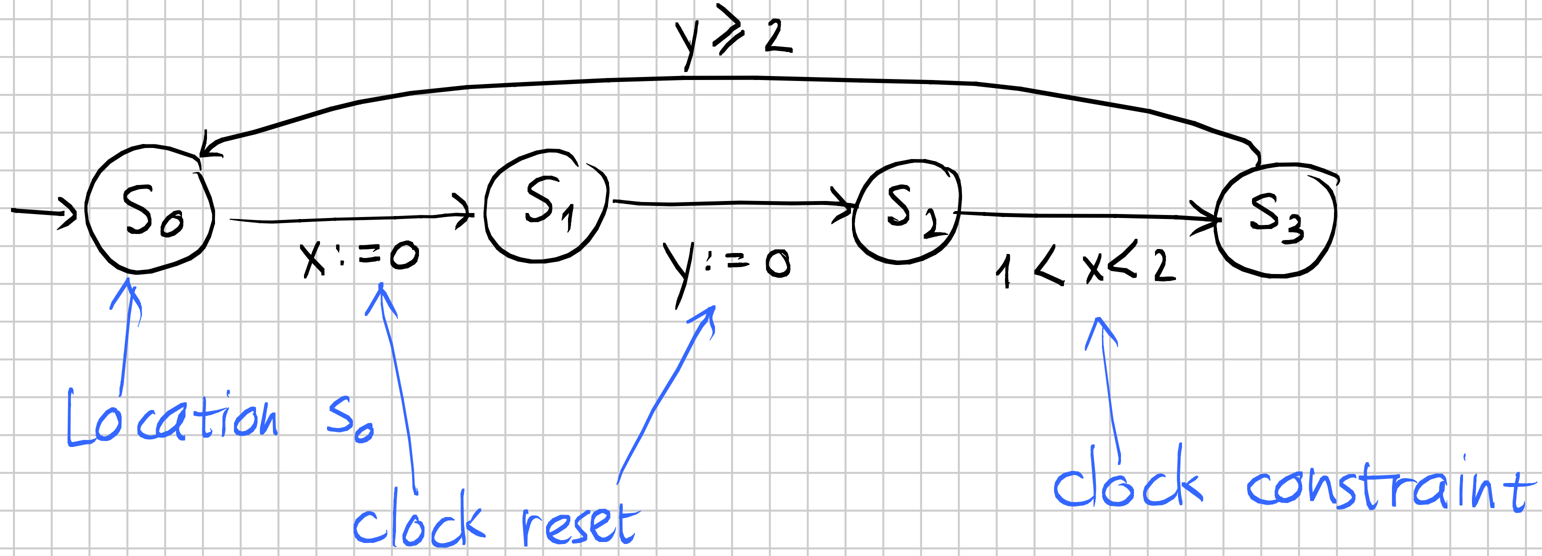
Clock:

1. Start with all clocks $= 0$
2. Clocks run with the time
3. Transition $t \in \rightarrow$ is only enabled if $i_{tr}(t)$ is true.
   If t is taken, $r(t)$ are reset.

Clock constraints: conjunctions of conditions like

$$x \geq c, \quad x > c, \quad x \leq c, \quad x < c \qquad (c \in \mathbb{N}) \; (x \in C)$$

Clock reset: a set of clocks, with the interpretation: reset these clocks to 0.



III.2 Semantics of Timed Automata

Clock assignments: functions $C \longrightarrow \mathbb{R}_0^+$

$\mu, \nu, \ldots$

States: pairs (Location, Clock assignment)

State changes:
- time passes b)
- a transition is taken a)

Initial state: (Initial location, all clocks are $=0$)

a) $(\ell, v) \xrightarrow{\text{discrete}} (\ell', v')$ if $t = (\ell, \ell') \in \longrightarrow$

$i_{tr}(t)$ is satisfied by v

$$v'(x) = \begin{cases} 0 & \text{if } x \in r(t) \\ v(x) & \text{otherwise} \end{cases}$$

b) $(\ell, v) \xrightarrow{\text{time}(\delta)} (\ell', v')$ if $\ell' = \ell$

$v'(x) = v(x) + \delta$

Runs: $\underbrace{(\ell_0, v_0)}^{\text{initial state.}} \longrightarrow (\ell_1, v_1) \longrightarrow (\ell_2, v_2) \longrightarrow \cdots$

where each arrow is a discrete or a time transition.

"Wrong" runs: – only discrete steps : time stops

– $(\ell_0, v_0) \xrightarrow{t(1)} (\ell_1, v_1) \xrightarrow{t(\frac{1}{2})} (\ell_2, v_2) \xrightarrow{t(\frac{1}{4})} (\ell_3, v_3) \xrightarrow{t(\frac{1}{8})} \cdots$: convergent time

Zenoness := time does not diverge.

In most semantics, only non-Zeno runs count.

III.3. TCTL : a logic for timed automata.

A) Syntax

Syntax of CTL:
$$\Phi ::= a \mid \neg\Phi \mid \Phi \wedge \Phi \mid A\varphi \mid E\varphi$$
$$\varphi ::= X\,\Phi \mid \Phi\,\mathcal{U}\,\Phi$$

Syntax of TCTL:
$$\Phi ::= \text{same}$$
$$\varphi ::= \Phi\,\mathcal{U}_{\sim c}\,\Phi \qquad \sim\,\in \{<, \leq, >, \geq\},\ c \in \mathbb{N}$$

e.g. $E(\text{red}\ \mathcal{U}_{\leq 4}\ \text{blue})$ it is possible to reach a blue state within 4 time units, only touching re

B] Semantics

Given a TA $\mathfrak{M} = (L, L_0, I, \to, C, i_{tr}, r)$, a state (ℓ, v) satisfies ϕ if:

$(\mathfrak{M}, \ell, v) \models a$ if $I(\ell) \ni a$

$\left.\begin{array}{l} \models \neg \phi \\ \models \phi \wedge \psi \end{array}\right\}$ same as in CTL

$\models A\varphi$ if all runs starting in (ℓ, v) satisfy φ

$\models E\varphi$ if some run etc.

sloppy!

$(\mathfrak{M}, (\ell_0, v_0) \to (\ell_1, v_1) \to \cdots) \models \phi \, \mathcal{U}_{\sim c} \, \psi$ if there is an i such that $(\mathfrak{M}, \ell_i, v_i) \models \psi$; for every $j < i$, we have $(\mathfrak{M}, \ell_j, v_j) \models \phi$, and $\mathrm{time}(i) \sim c$

A position in a run is a pair (i, δ) where

$i \in \mathbb{N}$, and $\delta = 0$ if $(\ell_i, v_i) \xrightarrow{\text{discrete}}$

$\delta \leq \varepsilon$ if $(\ell_i, v_i) \xrightarrow{\text{time}(\varepsilon)}$

sometimes $<$

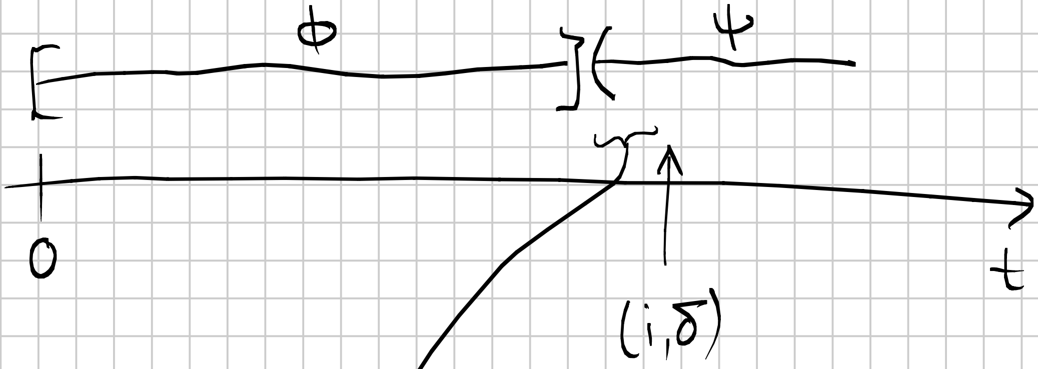
$(\mathfrak{M}, (\ell_0, v_0) \to (\ell_1, v_1) \to \dots) \models \phi \, \mathcal{U}_{\sim c} \, \psi$ if

1. there is a position (i, δ) such that $(\mathfrak{M}, \ell_i, v_i + \delta) \models \psi$

2. for every position $(j, \varepsilon) < (i, \delta)$, we have $(\mathfrak{M}, \ell_j, v_j + \varepsilon) \models \phi$

3. $\underbrace{\text{time}(i, \delta)}_{\delta + \sum_{j=0}^{i-1} \delta_j} \sim c$ where $(\ell_j, v_j) \xrightarrow{\text{time}(\delta_j)} (\ell_{j+1}, v_{j+1})$

Still a problem.



should satisfy ϕ

We accept, for now, that this situation is not handled intuitionally,

III. 4. TCTL Model Checking.

Problem: infinitely many states

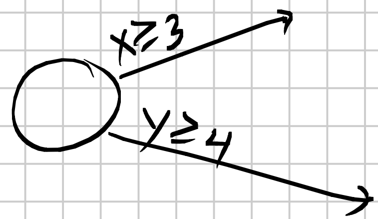
Solution: identify some states via an equivalence relation that respects TCTL-equivalence, similar to yesterday's bisimulations.

States can be distinguished by a suitable TCTL formula if:

- clock assignments like $x = 3{,}1$ / $x = 4{,}2$ with different integer values (a formula like $E\Diamond_{\leq 4}\, a$)
- fractional value $= 0$ / $\neq 0$ ($E\Diamond_{\leq 4}\, a$ vs. $E\Diamond_{< 4}\, a$)
- different orderings of fractional values.

  $x = 2{,}1$; $y = 3{,}7$ / $x = 2{,}5$; $y = 3{,}3$

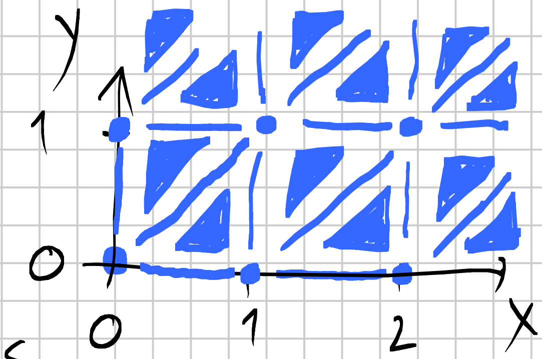
  $0{,}1 < 0{,}7$ $\qquad$ $0{,}5 > 0{,}3$



If two clock assignments cannot be distinguished by the above clauses, they are equivalent.

Equivalence classes are called <u>regions</u>.

Example: $C = \{x, y\}$



no longer $\mathbb{R}_0^+ \times \mathbb{R}_0^+$ cases,
but "only" countably many.

We extend this relation to states: $(\ell, v) \sim (\ell', v')$
if $\ell = \ell'$ and $v \sim v'$.

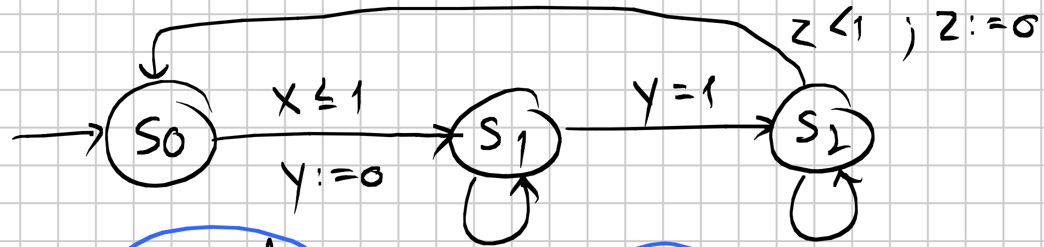
The region automaton:

- states are regions of the original TA.
- transitions are $\xrightarrow{\text{discrete}} \cup \xrightarrow{\text{time}(\delta)}$ for small δ
- Initial states: regions of the original states.
- Labeling: as a single region only contains one location, it is uniquely defined.

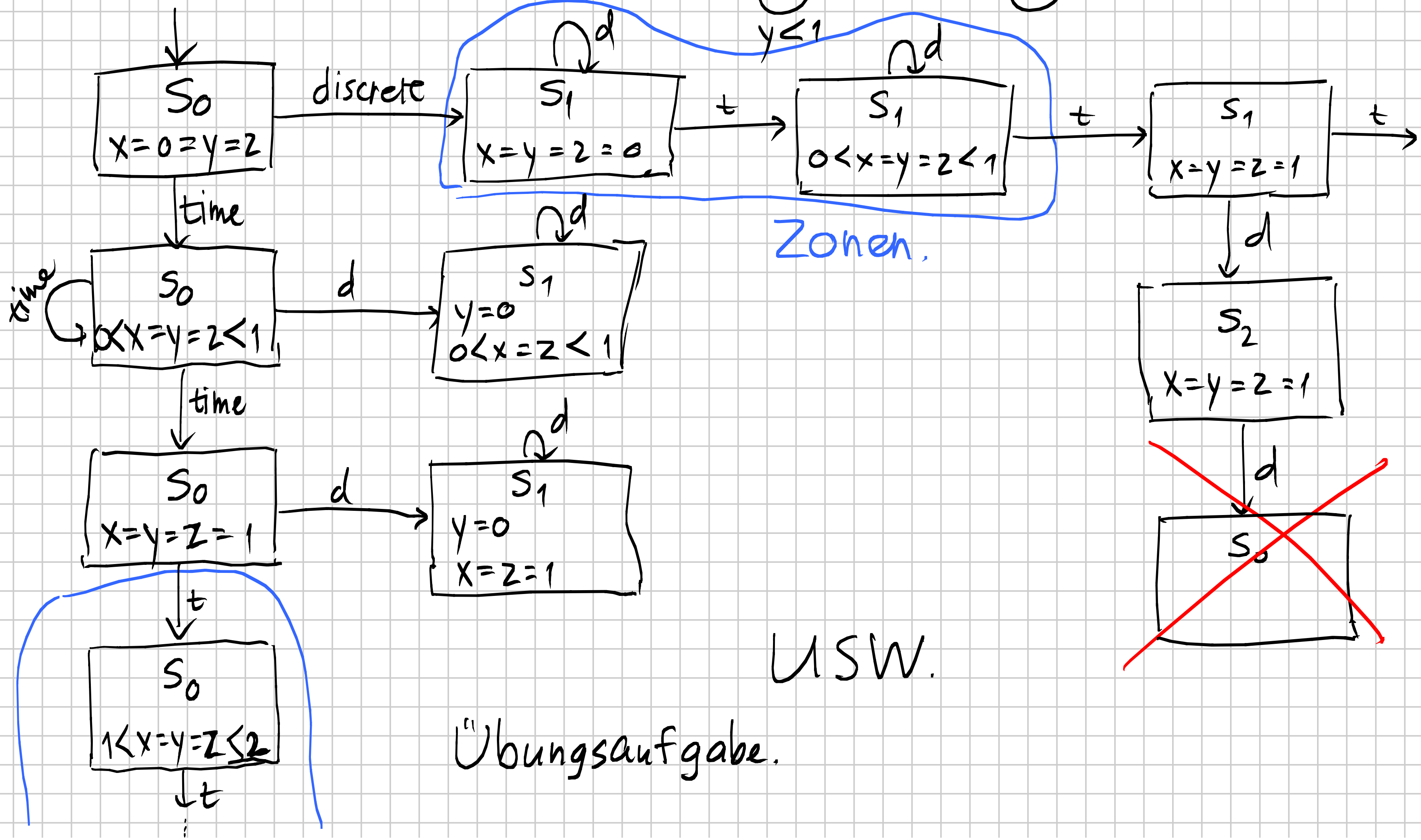
$\rightarrow$ This is a Kripke structure, no TA any more.

↳ We almost know how to check TCTL formulas now.

Example:



$C = \{x, y, z\}$



usw.

Übungsaufgabe.

Tool : UPPAAL

Uppsala + Aalborg

Sweden Denmark

Sweden Denmark

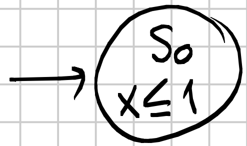
extends TA by: - Location invariants.

a function $i_{loc} : L \longrightarrow CC$

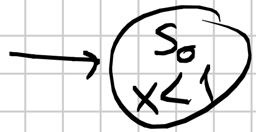
it is not allowed to stay in ℓ

so lang that $i_{loc}(\ell)$ would become

false.

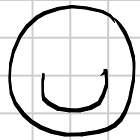


New problem:
timelock.



further UPPAAL extensions:

- urgent locations: it is forbidden to wait.



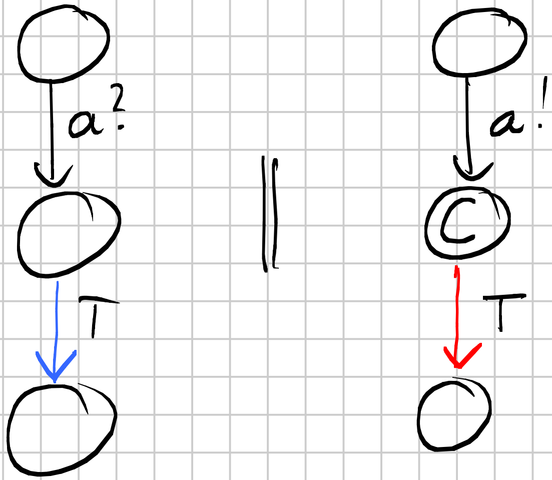
- UPPAAL models may contain multiple TA with synchronisation.

  + add action labels $\rightarrow \subseteq L \times A \times L$

synchronisation only between <name>! and <name>?, exactly two parties.

$$\frac{E \xrightarrow{a!} E' \qquad F \xrightarrow{a?} F'}{E \| F \xrightarrow{\tau} E' \| F'}$$

- Committed locations: more strict than urgent locations. Must be left in the next transition — no interleaving allowed.



The red arrow must precede the blue arrow.

Uppaal is available at www.uppaal.com.

Nachtrag: Checking $\Phi \,\mathcal{U}_{\sim c}\, \Psi$.

Main idea: add one more clock, called "formula clock", fc.

Augmented clock assignments: $C \cup \{fc\} \longrightarrow \mathbb{R}_0^+$.

Augmented region: same as a region, but with an augmented clock assignment.

In the augmented region automaton, add one extra atomic proposition: $a_{\sim c}$ holds in all states that satisfy:

$$fc \sim c.$$

Check for $\Phi \,\mathcal{U}\, (\Psi \wedge a_{\sim c})$ in each state of the augmented region automaton where $fc = 0$.

Finally, map back from the augmented region automaton to the region automaton.

$$(RA, (\ell, [v])) \models \phi \, \mathcal{U}_{\sim c} \, \psi$$

$$\text{iff} \quad (\text{augm. } RA, (\ell, [v] \oplus f_c \mapsto 0)) \models \phi \, \mathcal{U} (\psi \land a_{\sim c})$$

We need a fairness condition to avoid Zeno runs.

A run in the region automaton can correspond to a non-Zeno run in the TA, if for all $x \in C$

- either the clock x is reset infinitely often;
- or the clock value of x is unbounded.

and the run contains infinitely many time steps.