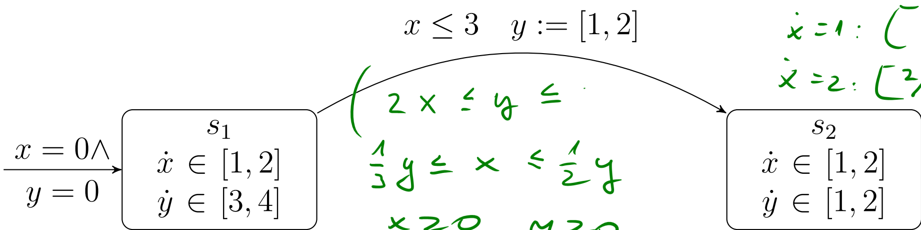
Modeling and Analysis of Hybrid Systems

Series 4

Exercise 1

Consider the following initialized rectangular automaton $\mathcal{A}$:



$(2x \leq y \leq$

$\frac{1}{3}y \leq x \leq \frac{1}{2}y$

$x \geq 0 \quad y \geq 0$

$\dot{x} = 1 : [\frac{1}{3}y, \frac{1}{4}y]$

$\dot{x} = 2 : [2/3\,y, 1/2\,y]$ $[\frac{1}{4}y, 2/3\,y]$

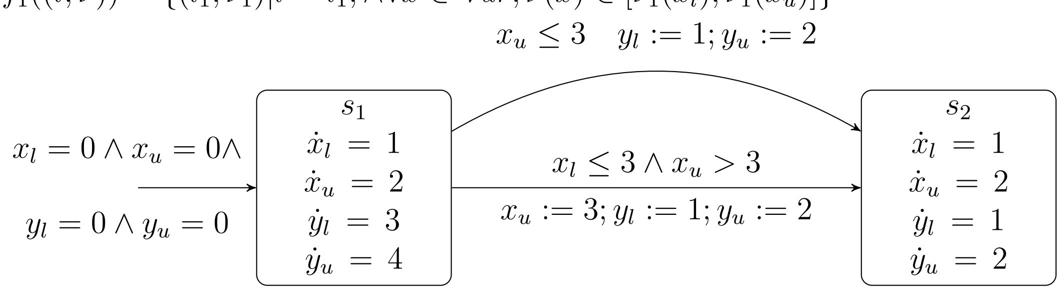
(a) Transform $\mathcal{A}$ into an initialized singular automaton $\mathcal{A}_1$ and specify a function f_1 mapping $\mathcal{A}$-states to $\mathcal{A}_1$-states, such that a state s is reachable in $\mathcal{A}$ iff $f_1(s)$ is reachable in $\mathcal{A}_1$.

(b) Transform $\mathcal{A}_1$ into an initialized stopwatch automaton $\mathcal{A}_2$ and specify a function f_2 mapping $\mathcal{A}_1$-states to $\mathcal{A}_2$-states, such that a state s is reachable in $\mathcal{A}_1$ iff $f_2(s)$ is reachable in $\mathcal{A}_2$. You are allowed to set stopwatch values to any constants.

(c) Transform $\mathcal{A}_2$ into a timed automaton $\mathcal{A}_3$ and specify a function f_3 mapping $\mathcal{A}_2$-states to $\mathcal{A}_3$-states, such that a state s is reachable in $\mathcal{A}_2$ iff $f_3(s)$ is reachable in $\mathcal{A}_3$. You are allowed to set clock values to any constants.

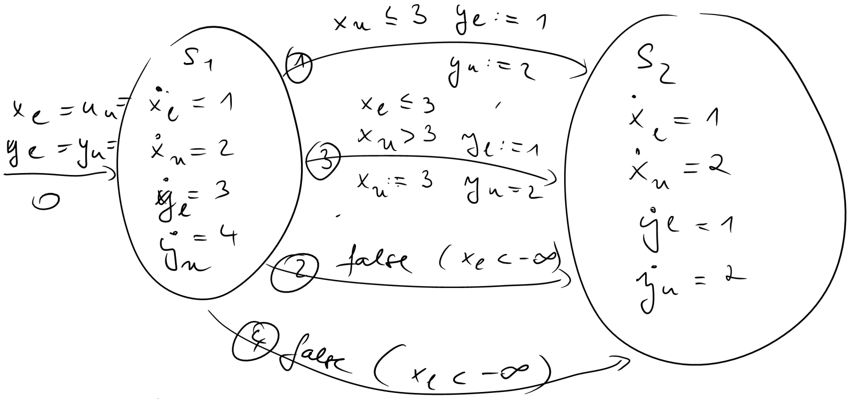
Solution:

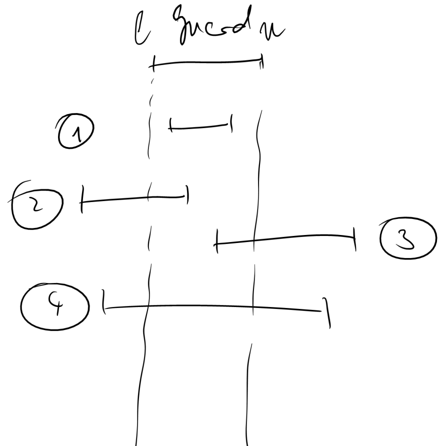
(a) $f_1((l,\nu)) = \{(l_1,\nu_1) | l = l_1, \wedge \forall x \in \mathit{Var}, \nu(x) \in [\nu_1(x_l), \nu_1(x_u)]\}$



(b) $f_2((l_1,\nu_1)) = (l_2,\nu_2)$ such that $l_1 = l_2 \wedge$

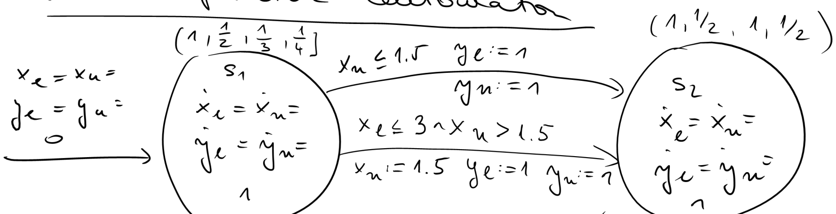
init. simple automaton

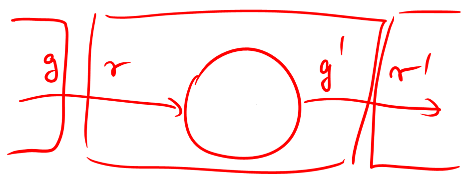




$$\alpha((s_i,(x_l,x_u,y_l,y_u))) = \{(s_i,(x,y)) \mid x_l \le x \le x_u \wedge y_l \le y \le y_u\}$$

init. stopwatch automaton





$$p(s_i,(x_l,x_u,y_l,y_u)) = \begin{cases} (s_1,(x_l,2x_u,3y_l,4y_u)) & \text{if } s_i = s_1 \\ (s_2,(x_l,2x_u,y_l,2y_u)) & \text{if } s_i = s_2 \end{cases}$$

timed automaton

no change (identity transf.)

$\gamma \equiv id$

Reachability in s_1

TA : $\{(s_1,(x_l,x_u,y_l,y_u)) \mid x_l = x_u = y_l = y_u\}$

i.stw : —||—

i.sing. : $\{(s_1,(x_l,x_u,y_l,y_u)) \mid x_l = \frac{1}{2}x_u \wedge x_l = \frac{1}{4}y_u \wedge x_l = \frac{1}{3}y_l\}$

$\{(s_1,(x_l,x_u,y_l,y_u)) \mid \cancel{\exists x_l', x_u', y_l', y_u'}.\ x_l = \frac{1}{2}x_u \wedge x_l = \frac{1}{3}y_l \wedge x_l = \frac{1}{4}y_u \wedge \cancel{x_l = x_l'} \wedge \cancel{x_u = 2x_u'} \wedge y_l = 3y_l' \wedge y_u = 4y_u'\}$

$x_u' = \frac{1}{2}x_u$ $y_l' = \frac{1}{3}y_l$ $y_u' = \frac{1}{4}y_u$

i.RA: $\{(s_1,(x,y)) \mid \exists x_l, x_u, y_l, y_u.\ x_l = \frac{1}{2}x_u \wedge x_l = \frac{1}{4}y_u \wedge x_l = \frac{1}{3}y_l \wedge x_l \le x \le x_u \wedge y_l \le y \le y_u\}$

$\exists x_l, x_u, y_l, y_u.\ x_l = \frac{1}{2}x_u \wedge \boxed{x_l = \frac{1}{4}y_u} \wedge x_l = \frac{1}{3}y_l \wedge x_l \le x \le x_u \wedge y_l \le y \le y_u \equiv$

$\exists x_u, y_l, y_u.\ \underbrace{\frac{1}{4}y_u = \frac{1}{2}x_u}_{x_u = \frac{1}{2}y_u} \wedge \frac{1}{4}y_u = \frac{1}{3}y_l \wedge \frac{1}{4}y_u \le x \le x_u \wedge y_l \le y \le y_u \equiv$

$\exists y_l, y_u.\ \underbrace{3y_u = 4y_l}_{y_l = \frac{3}{4}y_u} \wedge y_u \le 4x \le 2y_u \wedge y_l \le y \le y_u$

$\exists y_u.\ y_u \le 4x \le 2y_u \wedge \frac{3}{4}y_u \le y \le y_u \equiv$

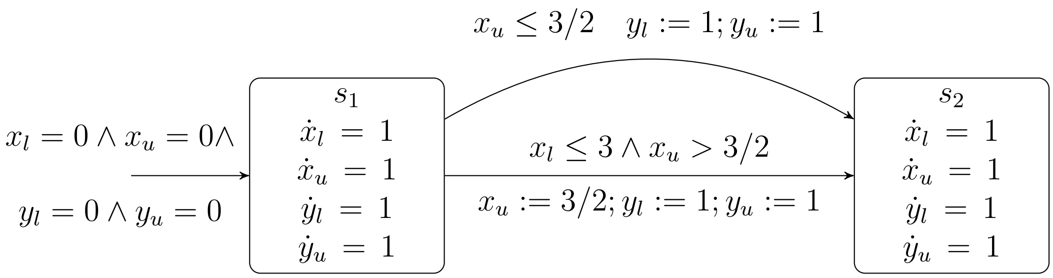
$\exists y_u.\ \begin{matrix} 2x \\ y \end{matrix} \le y_u \le \begin{matrix} 4x \\ 4/3\, y \end{matrix} \equiv \begin{matrix} 2x \le 4x \\ 0 \le x \end{matrix} \wedge \begin{matrix} 2x \le 4/3\, y \\ x \le 2/3\, y \end{matrix} \wedge \begin{matrix} y \le 4x \\ 1/4\, y \le x \end{matrix} \wedge \begin{matrix} y \le 4/3\, y \\ 0 \le y \end{matrix} \equiv$

(1) $\nu_2(x_l) = \nu_1(x_l)$

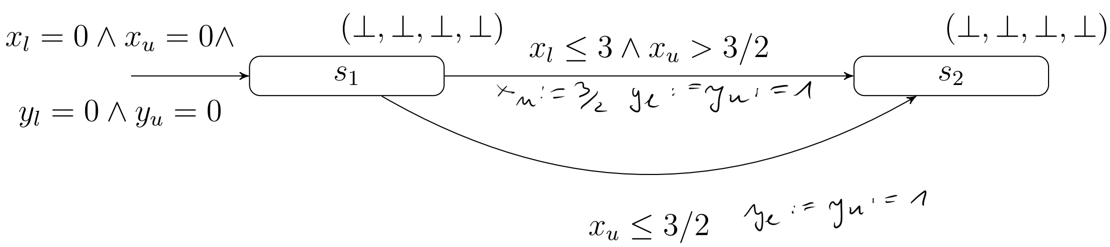
(2) $\nu_2(x_u) = \frac{1}{2}\nu_1(x_u)$

(3) $\nu_2(y_l) = \begin{cases} \frac{1}{3}\nu_1(y_l) & \textit{if } l_2 = s_1 \\ \nu_1(y_l) & \textit{else} \end{cases}$

(4) $\nu_2(y_u) = \begin{cases} \frac{1}{4}\nu_1(y_u) & \textit{if } l_2 = s_1 \\ \frac{1}{2}\nu_1(y_l) & \textit{else} \end{cases}$

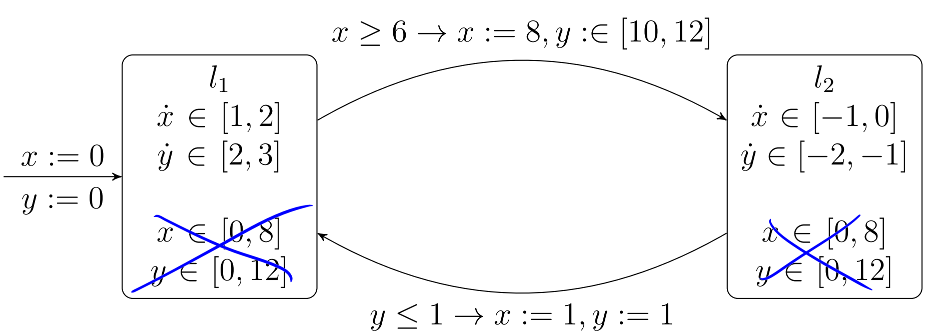


(c) $f_3((l_2, \nu_2)) = (l_2, \nu_2)$
The automaton $\mathcal{A}_2$ is already a timed automaton:



Exercise 2

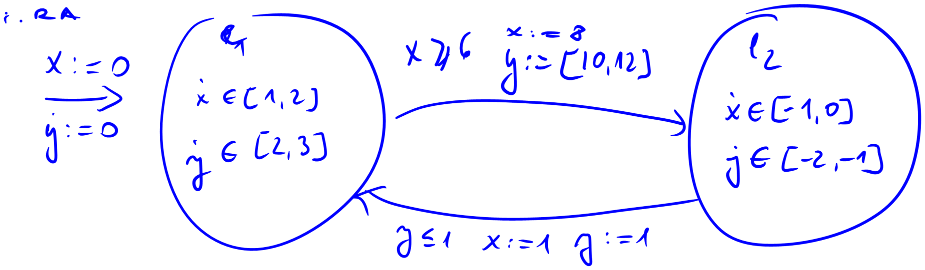
Consider the following initialized rectangular automaton $\mathcal{A}$:



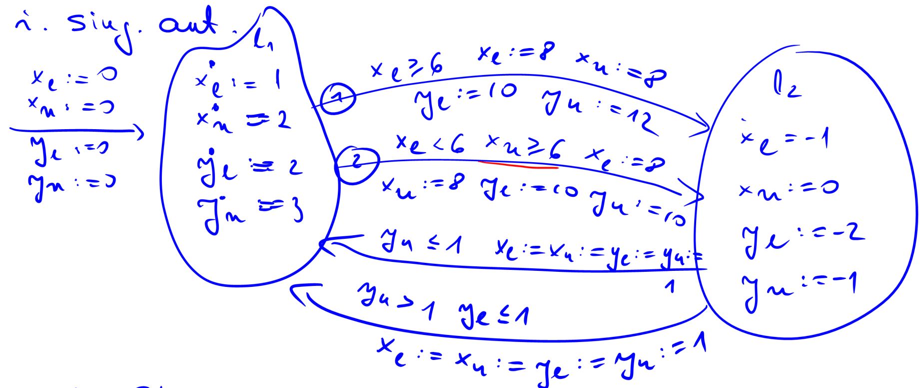
(a) Transform $\mathcal{A}$ into an equivalent singular automaton $\mathcal{A}_1$.

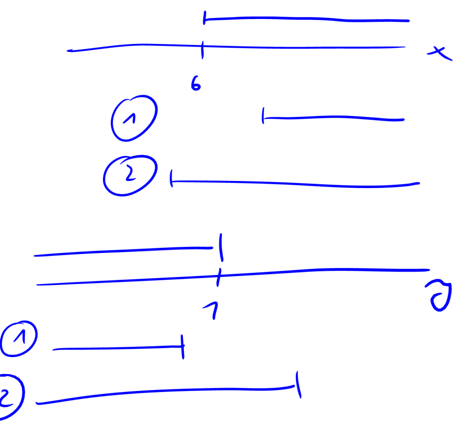
(b) Transform $\mathcal{A}_1$ into an equivalent stopwatch automaton $\mathcal{A}_2$.

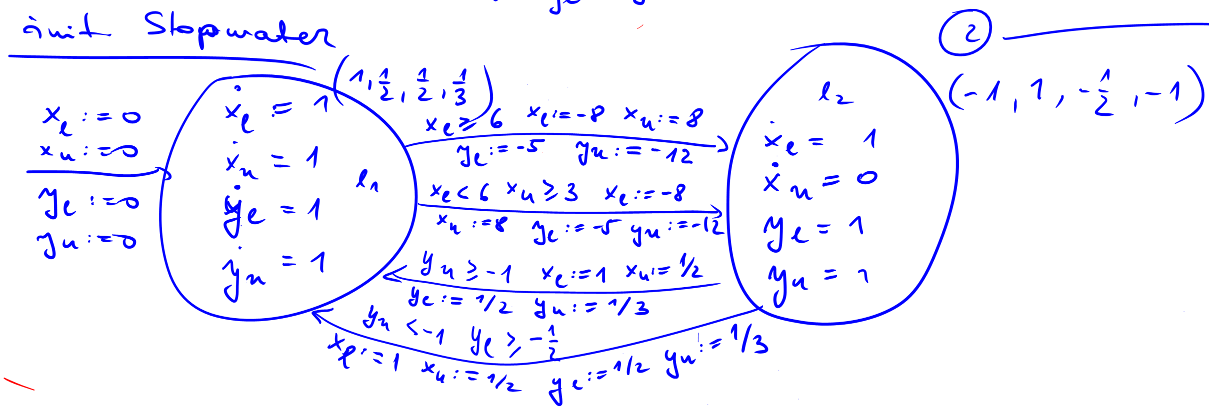
(c) Transform $\mathcal{A}_2$ into an equivalent timed automaton $\mathcal{A}_3$.



$\alpha\big(\ell_i,(x_l,x_u,y_l,y_u)\big) = \{(\ell_i,(x,y)) \mid x_l \leq x \leq x_u \wedge y_l \leq y \leq y_u\}$



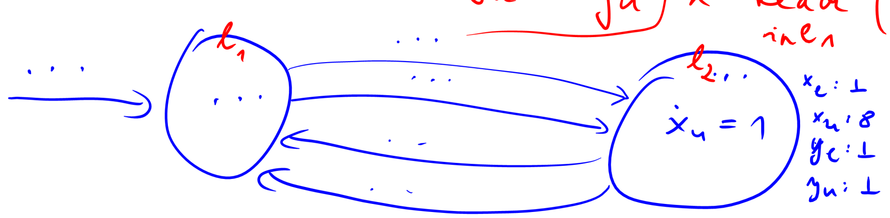




$\beta\big((\ell_1,(x'_l,x'_u,y'_l,y'_u))\big) = (\ell_1,(x_l,x_u,y_l,y_u))$ s.t. $x_l = x'_l \wedge x_u = 2x'_u \wedge y_l = 2y'_l \wedge y_u = 3y'_u$

$(x_l,x_u,y_l,y_u) = \beta(x'_l,x'_u,y'_l,y'_u)$

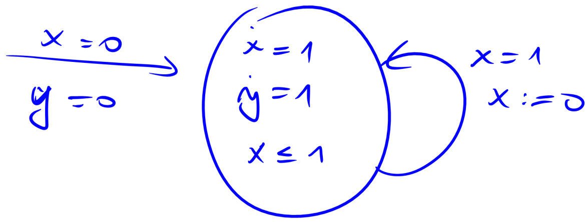
Reach in ℓ_1 (init. sing.) $= \{(x_l,x_u,y_l,y_u) \mid \exists\, x'_l, x'_u, y'_l, y'_u .\ x_l = x'_l \wedge x_u = 2x'_u \wedge y_l = 2y'_l \wedge y_u = 3y'_u \wedge$ Reach in ℓ_1 (init. stopwatch) $\in (x'_l,x'_u,y'_l,y'_u)\}$

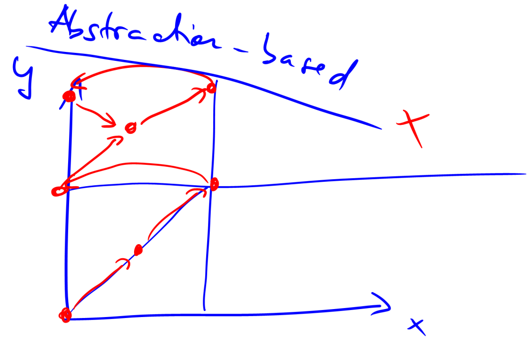


Reach (init. Stopw) in ℓ_1 = Reach (TA) in ℓ_1

—‖— ℓ_2 $= \{(x_l,x_u,y_l,y_u) \mid \exists\, x'_u .\ (x_l,x'_u,y_l,y_u) \in$ Reach (TA) in ℓ_2 and $x_u = 8\}$

Disadvantage of (forward) fixedpoint computations.





Forward fixedpoint: ✗

$(x=0 \wedge y=0) \rightarrow (x=y \wedge 0 \leq x \leq 1) \rightarrow (x=0 \wedge y=1) \rightarrow$

$(0 \leq x \leq 1 \wedge y=1+x) \rightarrow (x=0 \wedge y=2) \rightarrow \ldots$

Solution:

A singular automaton $\mathcal{A}_1$ of $\mathcal{A}$ can be obtained by introduction of new clocks as presented in the lecture. Having invariants on the original clocks, it is required to handle those separately. Invariant constraints need to hold for all new clocks as well, which introduces new invariant constraints, here: $x_l \in [0,8], x_u \in [0,8], y_l \in [0,12], y_u \in [0,12]$. Ordering the constraints ascending by the time they will be violated allows to create an ordered set of new locations, where the clocks for that respective constraint are stopped. Thereby we introduce new locations $l_{12}, l_{13}, l_{22}, l_{23}$ for each invariant constraint that can be violated and connect them with transitions. The original transitions are copied for those locations.

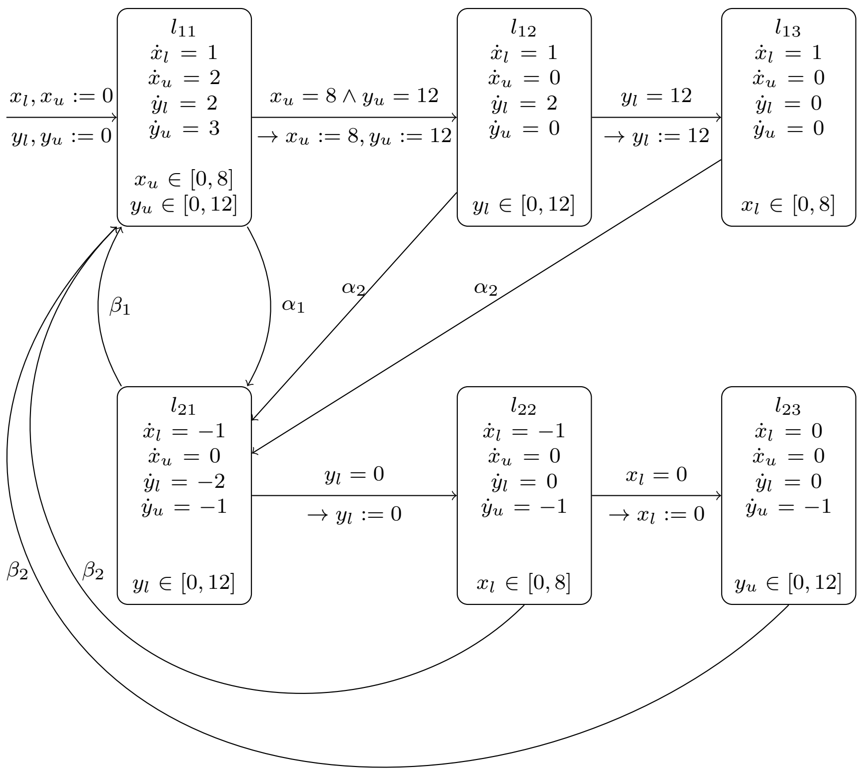


Figure 1: The initialized singular automaton $\mathcal{A}_1$

notations	guards	resets
α_1	$x_u \geq 6$	$x_l := 8, x_u := 8, y_l := 10, y_u := 12$
α_2	true	$x_l := 8, x_u := 8, y_l := 10, y_u := 12$
β_1	$y_l \leq 1$	$x_l := 1, x_u := 1, y_l := 1, y_u := 1$
β_2	true	$x_l := 1, x_u := 1, y_l := 1, y_u := 1$

Table 1: Notations in Figure 1

The initialized stopwatch automaton $\mathcal{A}_2$ can be obtained from $\mathcal{A}_1$ by scaling the clocks as presented in the lecture. However special attention is required for scaling clocks with negative derivatives: All constraints on these clocks e.g. guards and invariants have to be "mirrored on the zero-axis", i.e. the scaled absolute value of the satisfying interval is considered. As an example consider a clock $\dot{x} = -2$ constrained by the invariant $x \in [-5, -3]$. After transformation to $\dot{x} = 1$ the invariant constraint is $x \in [\frac{3}{2}, \frac{5}{2}]$.

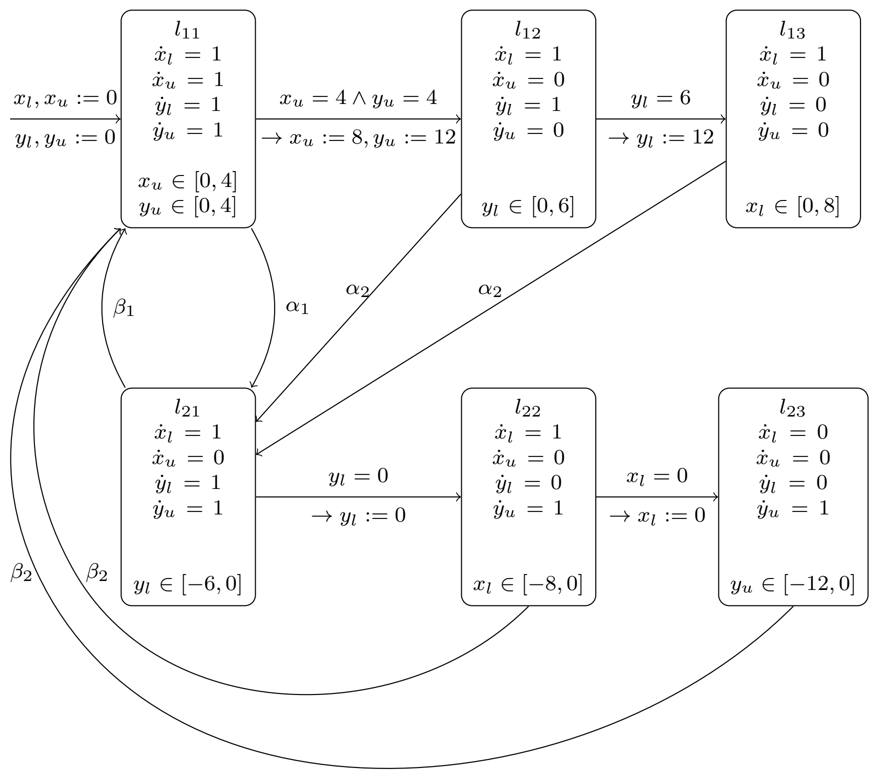


Figure 2: The initialized stopwatch automaton $\mathcal{A}_2$

notations	guards	resets
α_1	$x_u \geq 3$	$x_l := -8, x_u := 8, y_l := -5, y_u := -12$
α_2	true	$x_l := -8, x_u := 8, y_l := -5, y_u := -12$
β_1	$y_l \geq -\frac{1}{2}$	$x_l := 1, x_u := \frac{1}{2}, y_l := \frac{1}{2}, y_u := \frac{1}{3}$
β_2	true	$x_l := 1, x_u := \frac{1}{2}, y_l := \frac{1}{2}, y_u := \frac{1}{3}$

Table 2: Notations in Figure 2

The timed automaton $\mathcal{A}_3$ can be obtained from the initialized singular automaton $\mathcal{A}_2$ as presented in the lecture. For further details we also refer to the provided overview of the transformation handed out. This solution differs from our usual approach in that we do not allow to set variables to specific values. In case a variable is set to some value upon taking a transition while the flow does not change, we still need to track this value in the annotations and consider it for constraint adaptions (guards and invariants).

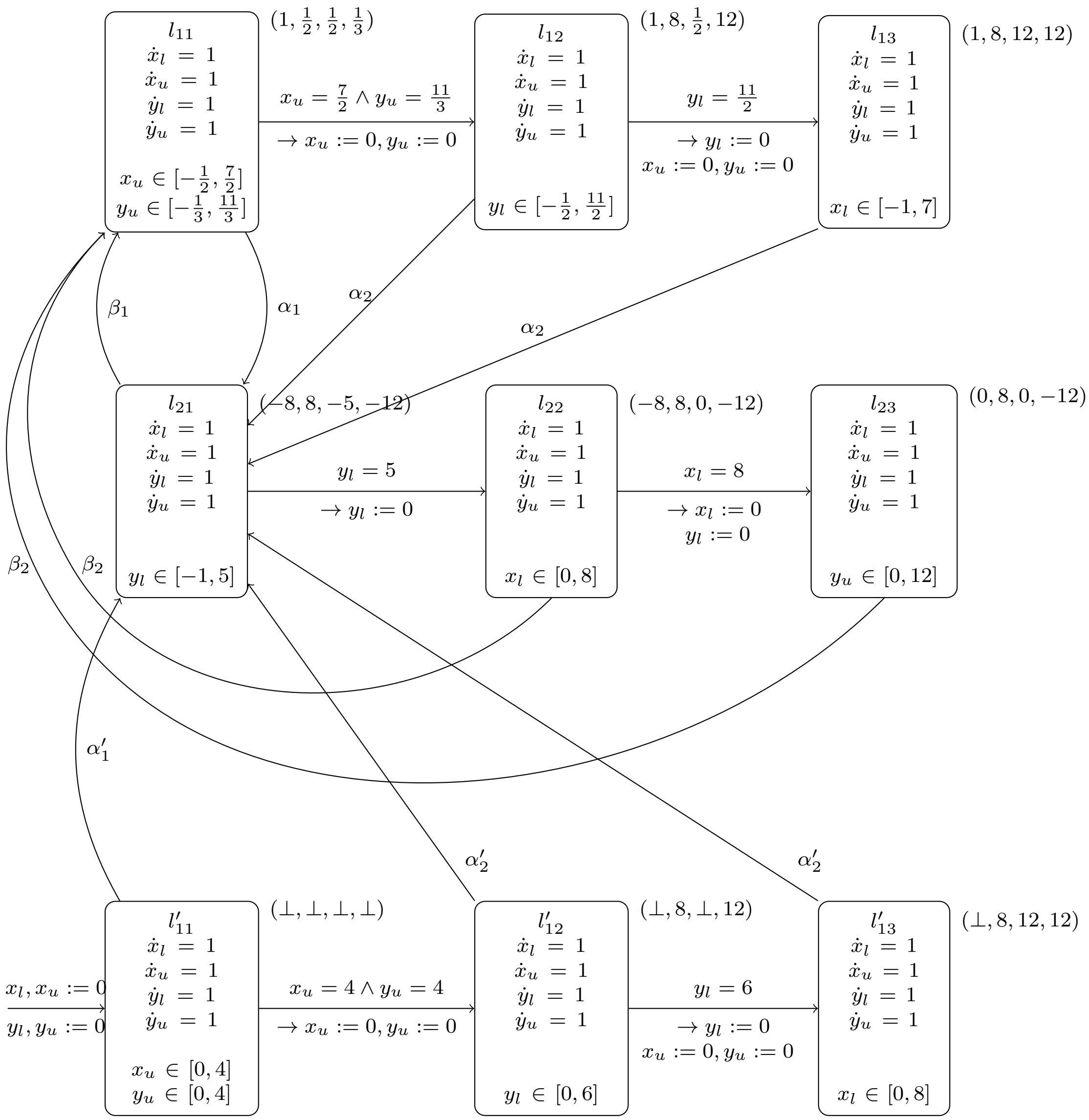


Figure 3: The timed automaton $\mathcal{A}_3$

notations	guards	resets
α_1	$x_u \geq \frac{5}{2}$	$x_l := 0, x_u := 0, y_l := 0, y_u := 0$
α_2	true	$x_l := 0, x_u := 0, y_l := 0, y_u := 0$
β_1	$y_l \geq \frac{9}{2}$	$x_l := 0, x_u := 0, y_l := 0, y_u := 0$
β_2	true	$x_l := 0, x_u := 0, y_l := 0, y_u := 0$
α'_1	$x_u \geq 3$	$x_l := 0, x_u := 0, y_l := 0, y_u := 0$
α'_2	true	$x_l := 0, x_u := 0, y_l := 0, y_u := 0$

Table 3: Notations in Figure 3