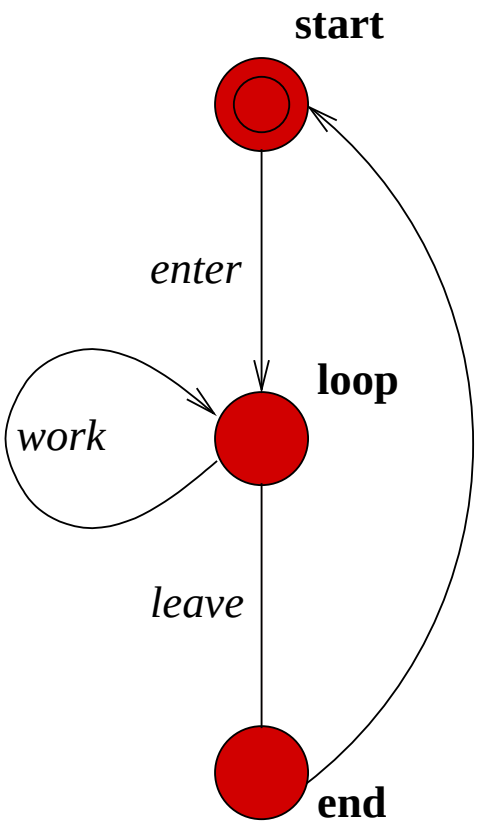
timed safety automata

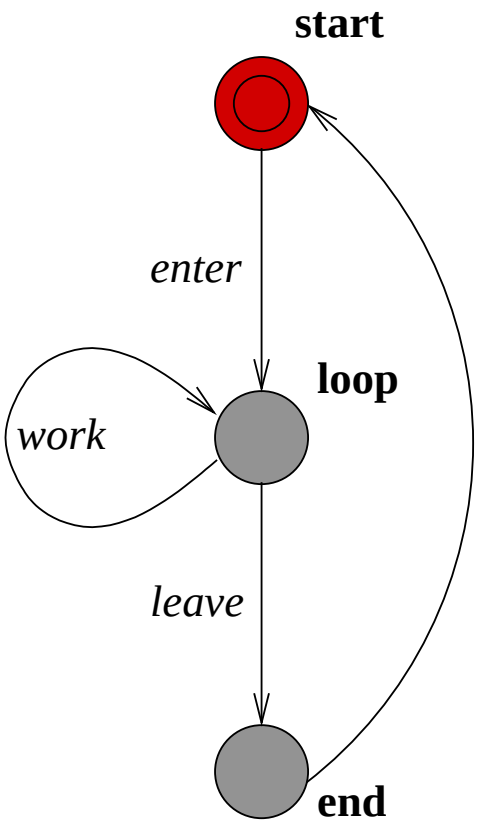
Markus Kuderer

Universität Karlsruhe

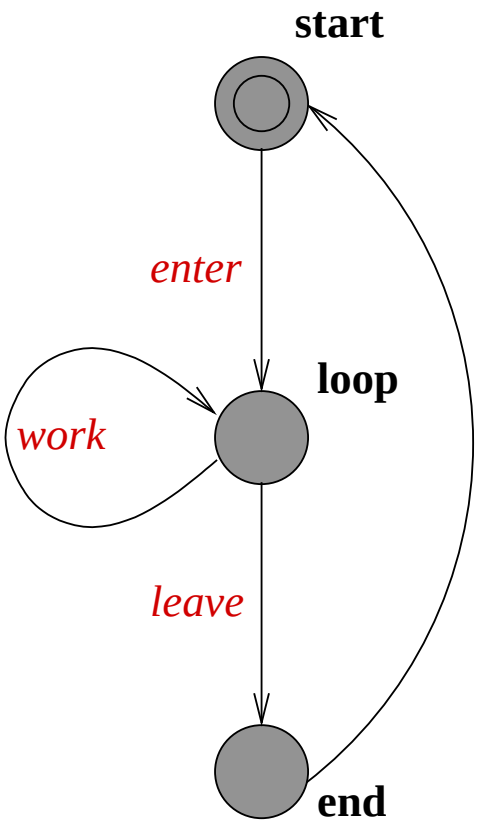
13.Juli 2007



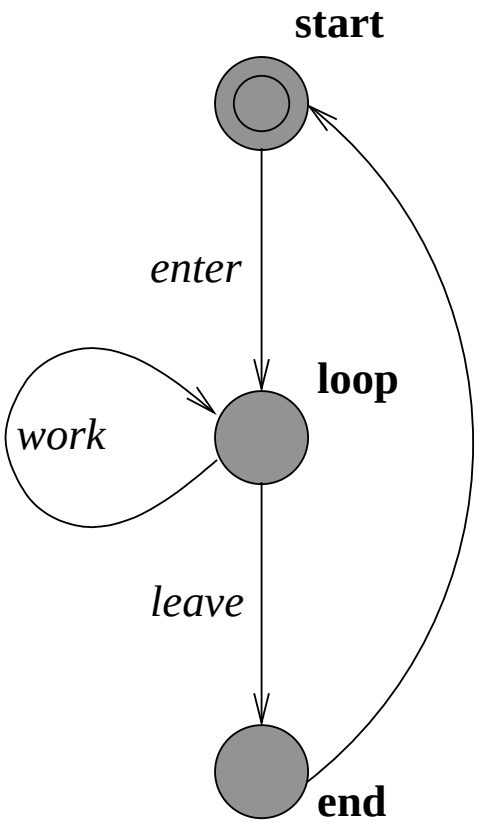
- Knoten



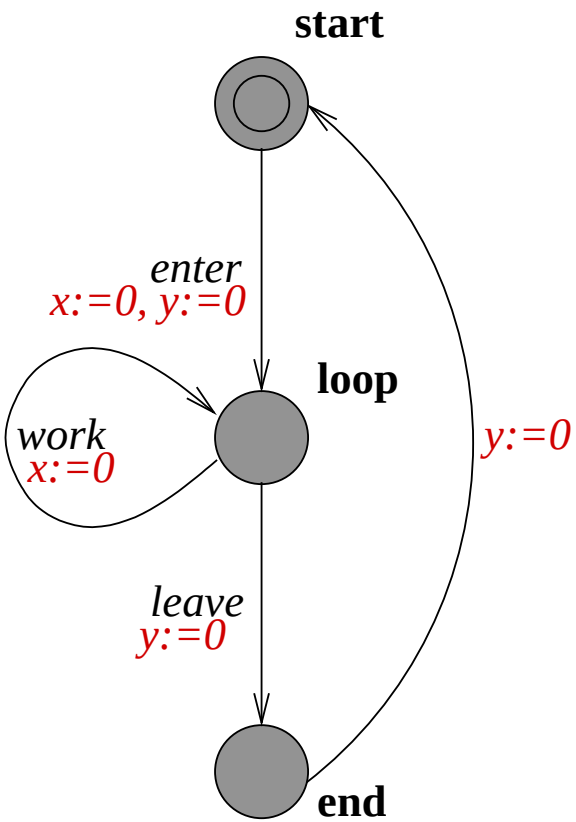
- Knoten
- Startknoten



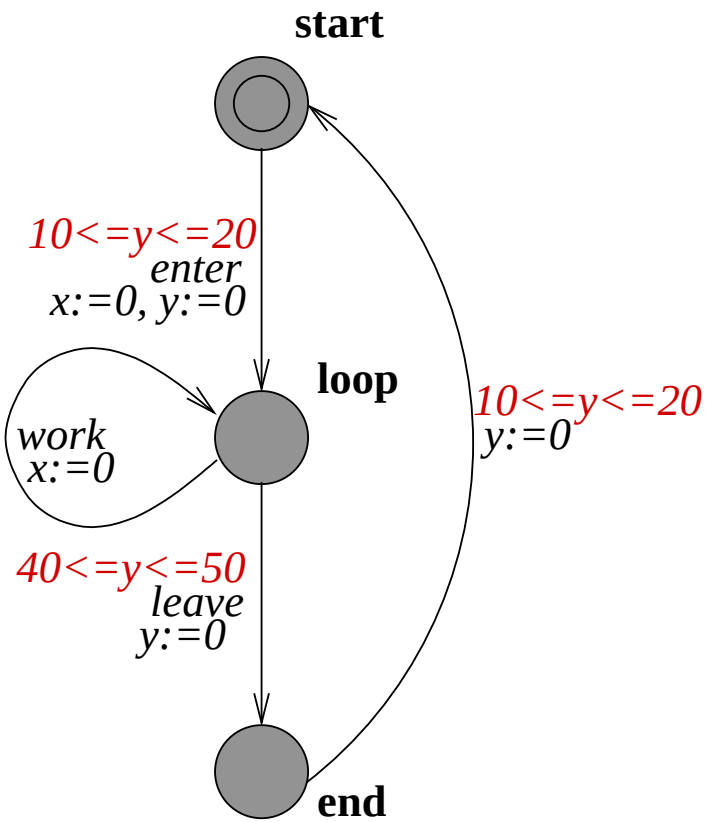
- Knoten
- Startknoten
- Alphabet



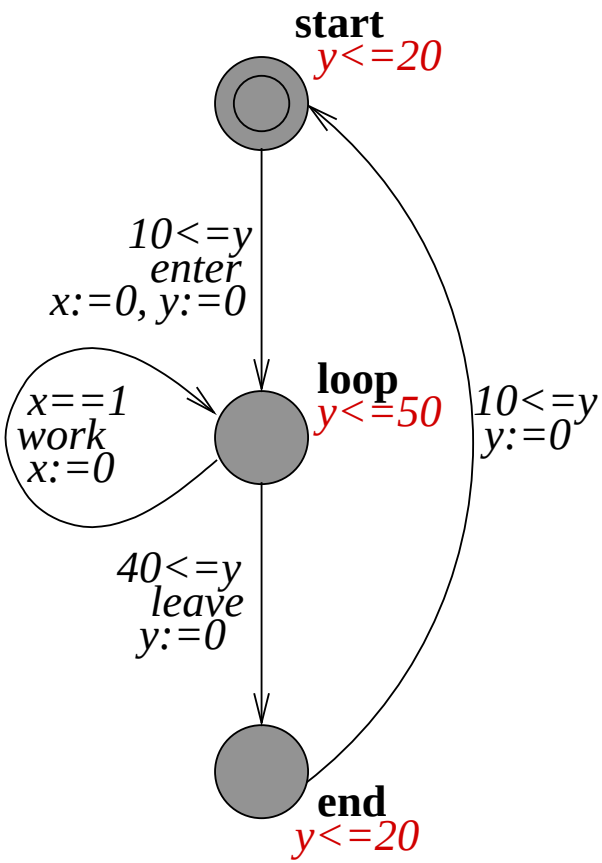
- Knoten
- Startknoten
- Alphabet
- Uhren $\in \{x, y, \dots\}$



- Knoten
- Startknoten
- Alphabet
- Uhren $\in \{x, y, \ldots\}$
  - Reset



- Knoten
- Startknoten
- Alphabet
- Uhren $\in \{x, y, \dots\}$
  - Reset
  - Übergangsbedingungen



- Knoten
- Startknoten
- Alphabet
- Uhren $\in \{x, y, \dots\}$
  - Reset
  - Übergangsbedingungen
  - Invarianten

Syntax

Definitionen

- $N =$ Endliche Menge an Knoten, l_0 Startknoten
- $\Sigma = \{a, b, ...\}$ Alphabet
- $\mathcal{C} = \{x, y, ...\}$ Uhren
- $\mathcal{B}(\mathcal{C}) = \{g, h, ...\}$ Beschränkungen

Beschränkungen ($\mathcal{B}(\mathcal{C})$)

konjunktive Formel mit atomaren Bedingungen der Form:
$x \sim n$ oder $x - y \sim n$ wobei $\sim \in \{\leq, <, =, >, \geq\}$ und $n \in \mathbb{N}$

Beispiel

$x - y \leq 3 \wedge y = 4 \wedge y - z > 17$

was nicht geht:

- $x + y < 3$
- ...
- $x - y < 3,5$
- ...

timed based automaton

- $\mathcal{A} =< N, l_0, E, I >$

timed based automaton

- $\mathcal{A} =< N, l_0, E, I >$
- $E \subseteq N \times \mathcal{B}(\mathcal{C}) \times \Sigma \times 2^{\mathcal{C}} \times N$ Übergangsfunktion

timed based automaton

- $\mathcal{A} =< N, l_0, E, I >$
- $E \subseteq N \times \mathcal{B}(\mathcal{C}) \times \Sigma \times 2^{\mathcal{C}} \times N$ Übergangsfunktion
- $I : N \mapsto \mathcal{B}(\mathcal{C})$ Invarianten

timed based automaton

- $\mathcal{A} =< N, l_0, E, I >$
- $E \subseteq N \times \mathcal{B}(\mathcal{C}) \times \Sigma \times 2^{\mathcal{C}} \times N$ Übergangsfunktion
- $I : N \mapsto \mathcal{B}(\mathcal{C})$ Invarianten

Schreibweise

- $l \xrightarrow{g,a,r} l' :=< l, g, a, r, l' > \in E$

Semantik

Definitionen

- $u : \mathcal{C} \rightarrow \mathbb{R}_+$ Zeitfunktion
- $u \in g :\Leftrightarrow$ Werte erfüllen Beschränkung g $\in \mathcal{B}(\mathcal{C})$

Beispiel

$u(x) = 3, u(y) = 5 \Rightarrow$
$u \in (y - x \leq 3)$ aber $u \notin (x > 7)$

Operationen

- $(u + d)(x) := u(x) + d$
- $[r \to 0]u(x) := \begin{cases} 0, & x \in r \\ u(x), & \text{sonst} \end{cases}$

Zustand

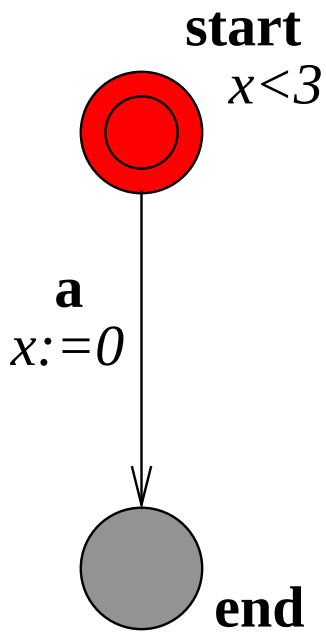
$< l, u >$ l : Aktiver Knoten, u : Zeitfunktion

Zeitübergang

$< l, u > \xrightarrow{d} < l, u + d >$, wenn $u \in I(l), (u + d) \in I(l)$

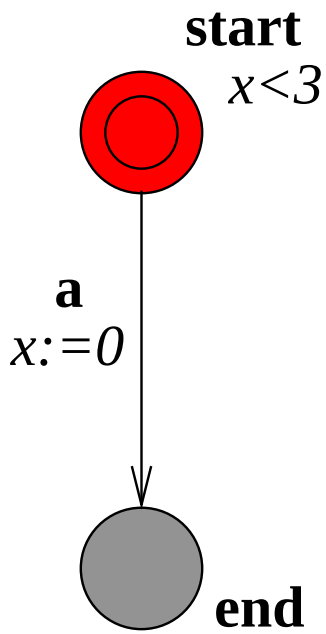
Knotenübergang

$< l, u > \xrightarrow{a} < l', u' >$, wenn
$l \xrightarrow{g,a,r} l', u \in g, u' = [r \rightarrow 0]u, u' \in I(l')$



$\mathcal{A} =< N, l_0, E, I >$

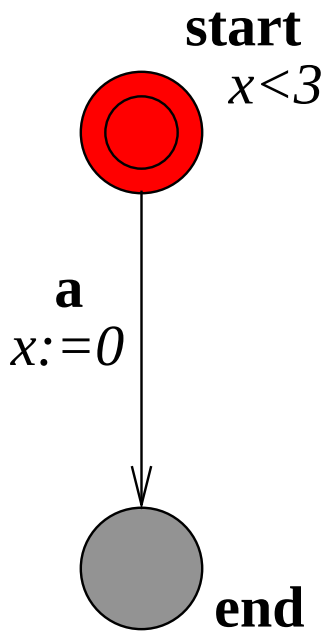
- $N = \{start, end\}$
- $l_0 = start$
- $I(start) = \{x < 3\}$; $I(end) = \{\}$
- $start \xrightarrow{\{\},a,\{x\}} end$



$\mathcal{A} =< N, l_0, E, I >$

- $N = \{start, end\}$
- $l_0 = start$
- $I(start) = \{x < 3\}$; $I(end) = \{\}$
- $start \xrightarrow{\{\},a,\{x\}} end$

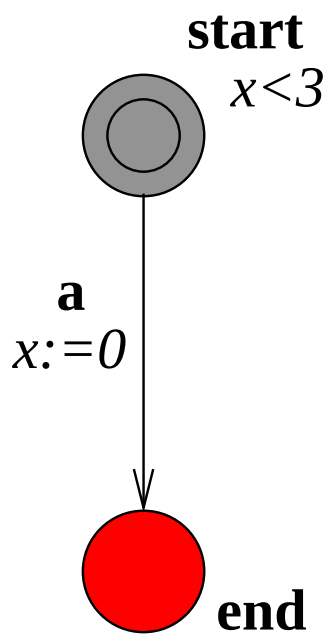
1. $< start, u >,\ u(x) = 0, u(y) = 0$



$\mathcal{A} =< N, l_0, E, I >$

- $N = \{start, end\}$
- $l_0 = start$
- $I(start) = \{x < 3\}$; $I(end) = \{\}$
- $start \xrightarrow{\{\},a,\{x\}} end$

1. $< start, u >,\ u(x) = 0, u(y) = 0$
   $\rightarrow$
2. $< start, u' >,\ u' = u + 2$
   - $u \in I(start)$
   - $u + 2 \in I(start)$



$\mathcal{A} =< N, l_0, E, I >$

- $N = \{start, end\}$
- $l_0 = start$
- $I(start) = \{x < 3\}$; $I(end) = \{\}$
- $start \xrightarrow{\{\},a,\{x\}} end$

1. $< start, u >,\ u(x) = 0, u(y) = 0$

   $\rightarrow$
2. $< start, u' >,\ u' = u + 2$
   - $u \in I(start)$
   - $u + 2 \in I(start)$

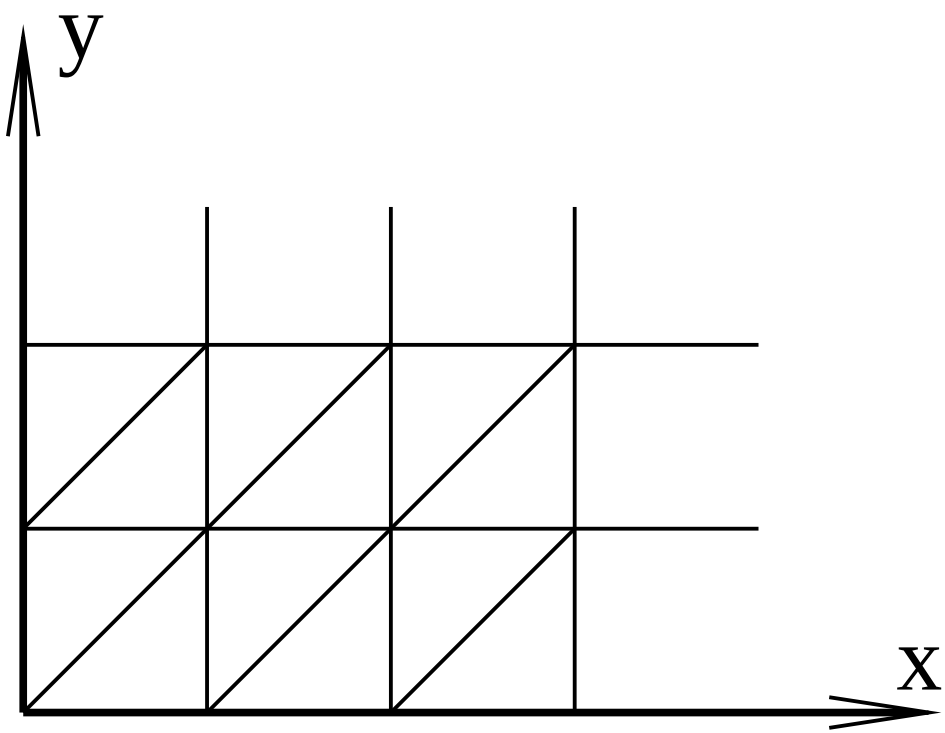
   $\rightarrow$
3. $< end, u'' >$
   - $u' \in \{\}$
   - $u'' = [\{x\} \rightarrow 0](u + 2)$
   - $u'' \in I(end)$

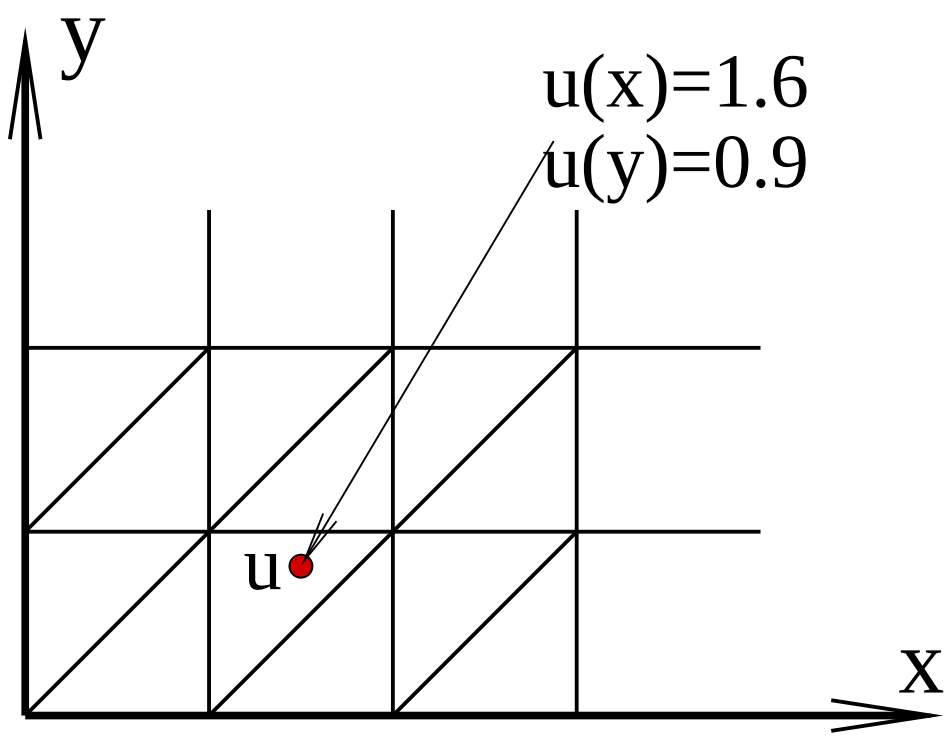
- timed trace $\xi := (t_1, a_1)(t_2, a_2)\ldots$
- run $:=< l_0, u_0 > \xrightarrow{d_1} \xrightarrow{a_1} < l_1, u_1 > \xrightarrow{d_2} \xrightarrow{a_2} \cdots, t_i = t_{i+1} + d_i$
- $L(\mathcal{A}) := \{\xi |$ es gibt run über $\xi\}$

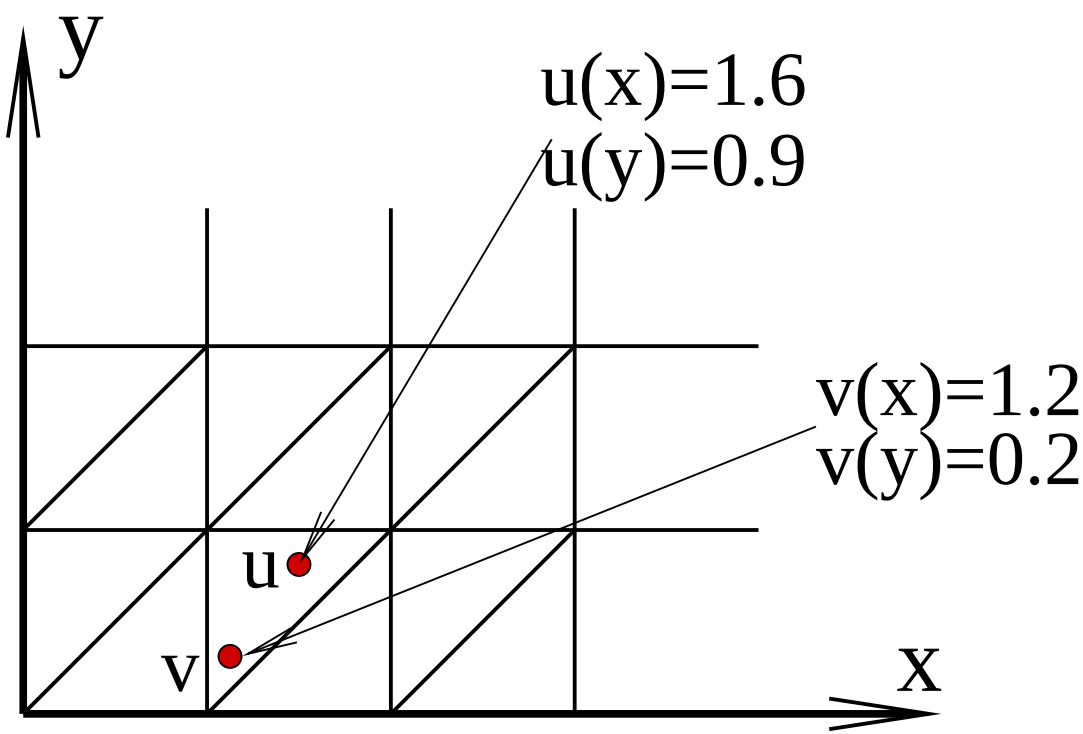
Regionen

Definitionen

- $k : \mathcal{C} \to \mathbb{N}$ maximale Konstante für entsprechende Uhr
- $d = \lfloor d \rfloor + \{d\}$
- u, v äquivalent $(u \dot{\sim}_k v) \Leftrightarrow \forall x, y :$
  1. $u(x), v(x) > k(x)$ oder $\lfloor u(x) \rfloor = \lfloor v(x) \rfloor$
  2. $u(x) > k(x)$ oder $\{u(x)\} = 0 \Leftrightarrow \{v(x)\} = 0$
  3. $u(x) > k(x), u(y) > k(y)$ oder
     $\{u(x)\} \leq \{u(y)\} \Leftrightarrow \{v(x)\} \leq \{v(y)\}$
- $[u]$ über $\dot{\sim}$ wird Region genannt $\Rightarrow$ Regionen sind endlich


y
x


y
u(x)=1.6
u(y)=0.9
u
x



$$\Rightarrow \forall g : u \in g \Leftrightarrow v \in g$$

Zonen

Definition

Eine Zone ist die Lösung einer Menge von Beschränkungen (aus $\mathcal{B}(\mathcal{C})$!), d.h. die maximale Menge an Zeitfunktionen, die die Beschränkungen erfüllen

Beispiel

$D = x < 1 \quad u(x) = 3.1 \Rightarrow u \notin D$

$v(x) = 0.5 \Rightarrow v \in D$

Zonen werden durch Beschränkungen angegeben, bestehen aber aus Zeitfunktionen!

Speicherung

DMB(Different Bound Matrices)

- Referenz-Uhr 0 wird eingeführt. $\mathcal{C}_0 = \mathcal{C} \cup \{0\}$
- Bedingungen können so in der Form $x_i - x_j \sim n$ angegeben werden. $(\sim \in \{<, \leq\})$
- $(n, \sim)$ wird in D_{ij} gespeichert $(D = |\mathcal{C}_0| \times |\mathcal{C}_0|$ -Matrix)

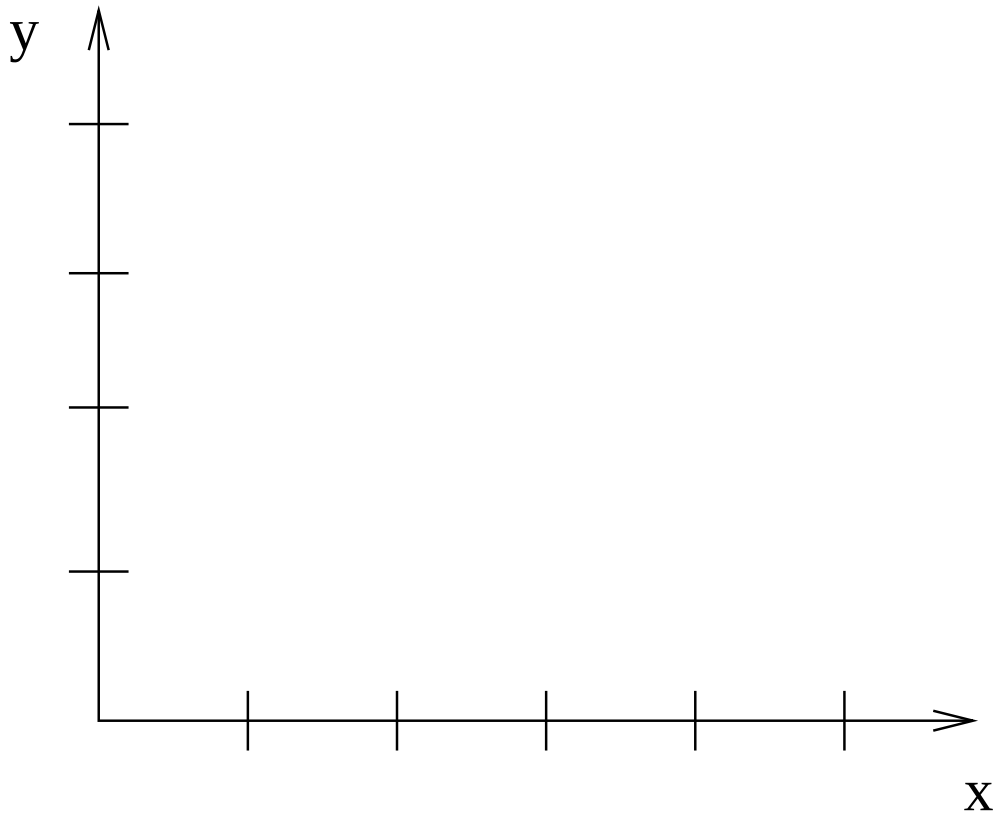
Beispiel

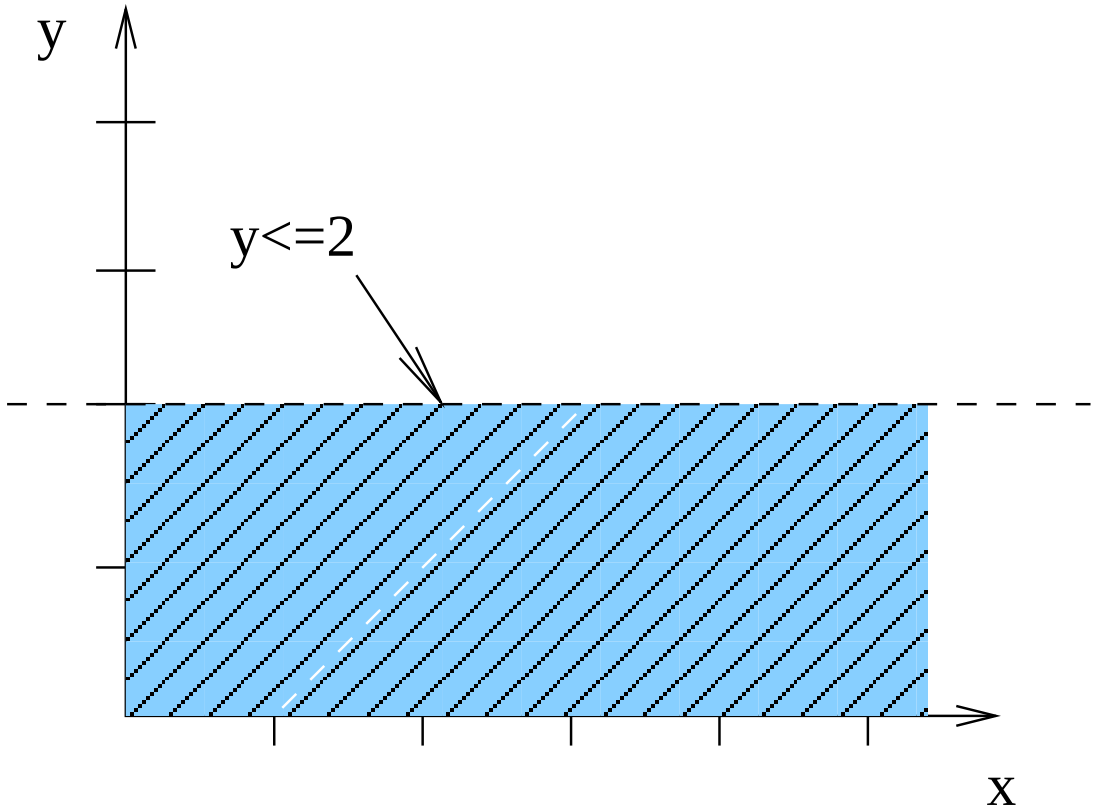
$$D = (x - y \geq -10) \wedge (x - y < 10) \wedge (y \geq 20)$$

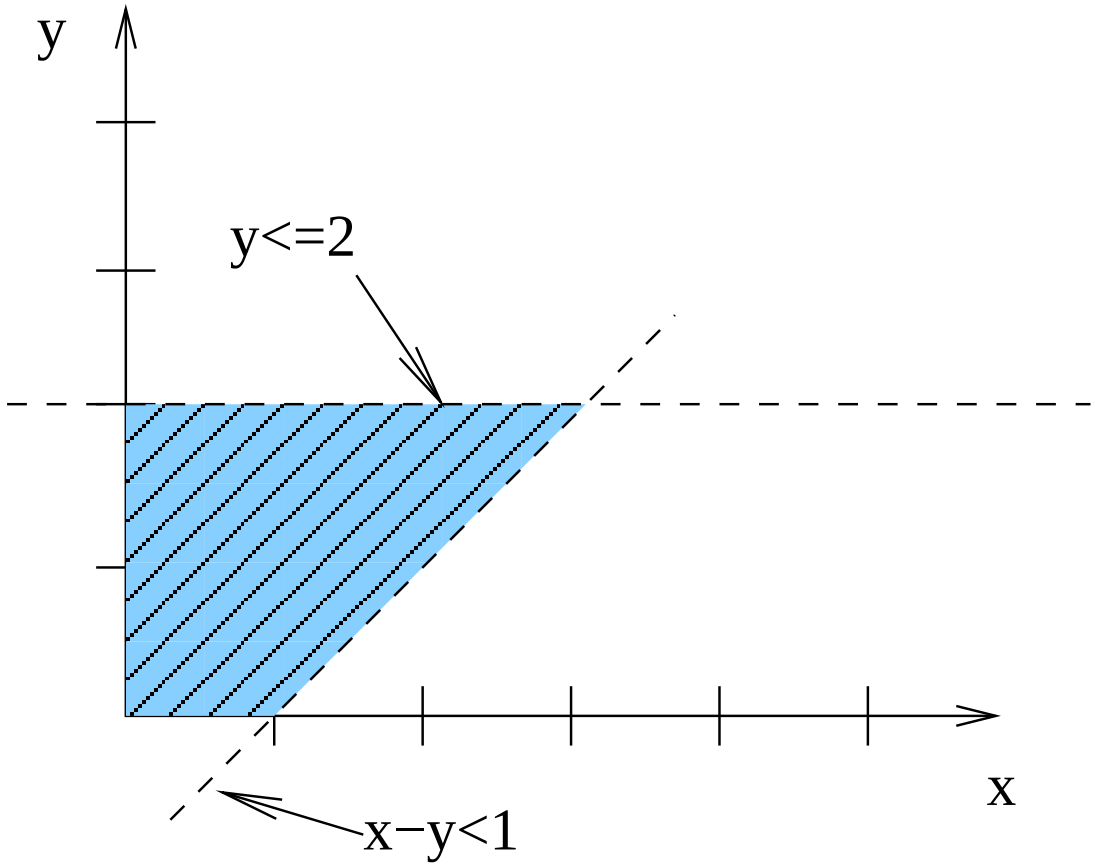
$$\Rightarrow D = (y - x \leq +10) \wedge (x - y < 10) \wedge (0 - y \leq -20)$$

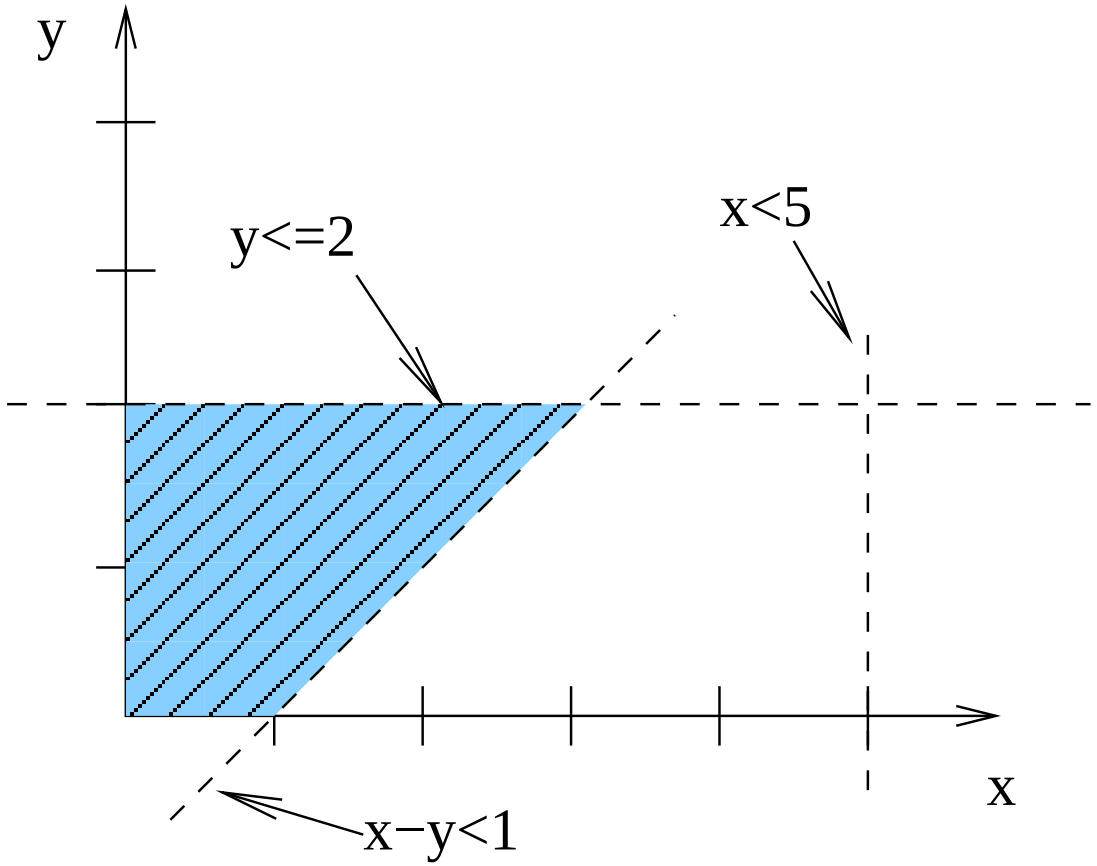
$$\begin{pmatrix} (0, \leq) & (0, \leq) & (-20, \leq) \\ \infty & (0, \leq) & (10, <) \\ \infty & (10, \leq) & (0, \leq) \end{pmatrix}$$

- Es gibt i.A. unendlich viele Zonen, die den gleichen Lösungsraum haben
- Diese lassen sich aber auf eine eindeutig bestimmte Form bringen: Die *kanonische Form*


y
x

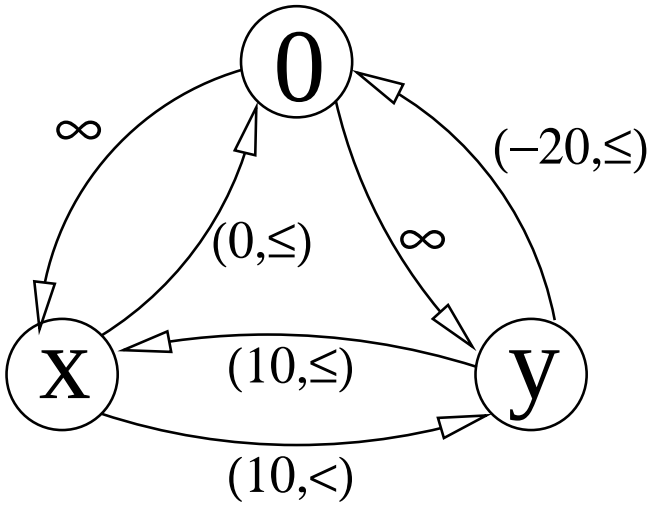

y
y<=2
x


y
y<=2
x
x−y<1

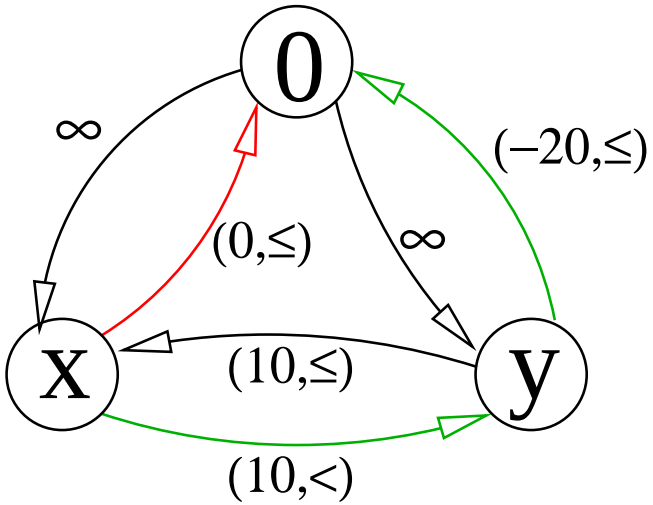

y
y<=2
x<5
x
x−y<1

Berechnung der kanonischen Form
kürzeste Wege im Graphen

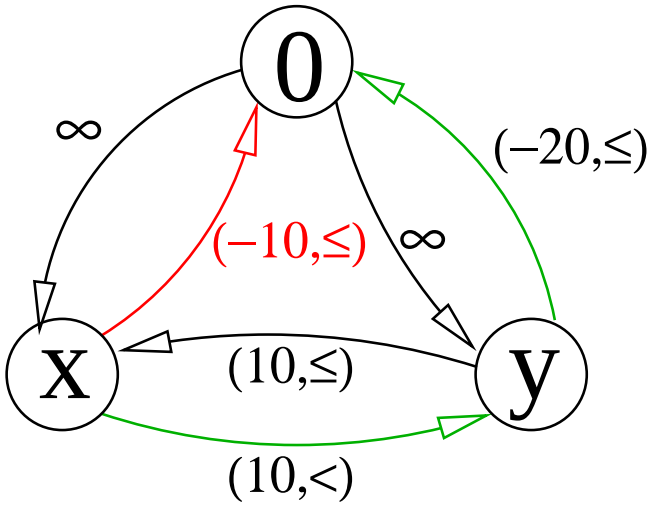
$$\begin{pmatrix} (0,\leq) & (0,<) & (-20,\leq) \\ \infty & (0,\leq) & (10,\leq) \\ \infty & (10,<) & (0,\leq) \end{pmatrix}$$



$$\begin{pmatrix} (0,\leq) & (0,<) & (-20,\leq) \\ \infty & (0,\leq) & (10,\leq) \\ \infty & (10,<) & (0,\leq) \end{pmatrix}$$



$$\begin{pmatrix} (0,\leq) & (-10,<) & (-20,\leq) \\ \infty & (0,\leq) & (10,\leq) \\ \infty & (10,<) & (0,\leq) \end{pmatrix}$$



Operationen

- $D^{\uparrow} := \{u + d | u \in D, d \in \mathbb{R}_+\}$
- $r(D) := \{[r \mapsto 0]u | u \in D\}$

Übergänge

- $< l, D > \rightsquigarrow < l, D^{\uparrow} \wedge I(l) >$ (alle in Zukunft liegenden Zuweisungen, die Bedingungen noch erfüllen)
- $< l, D > \rightsquigarrow < l', r(D \wedge g) \wedge I(l') >$ (Übergänge, die Resetbedingungen und neue Bedingungen erfüllen)

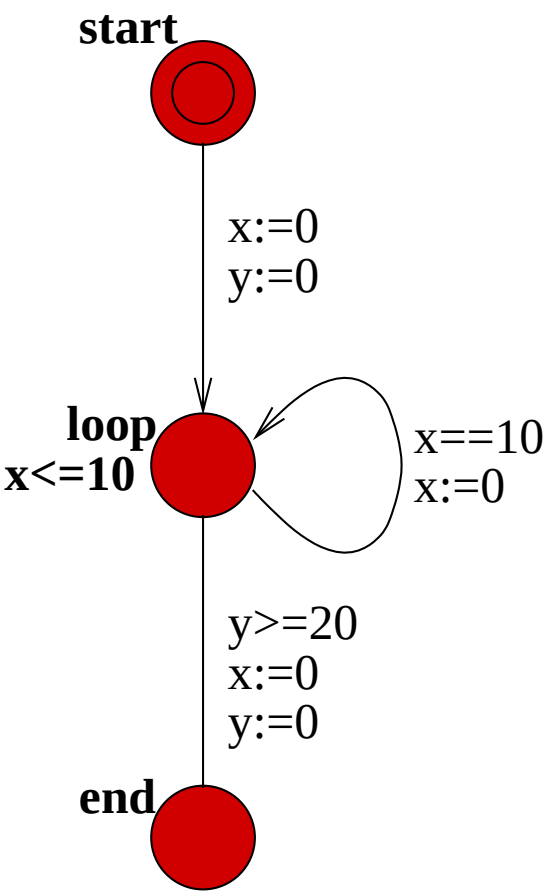
korrekt und vollständig

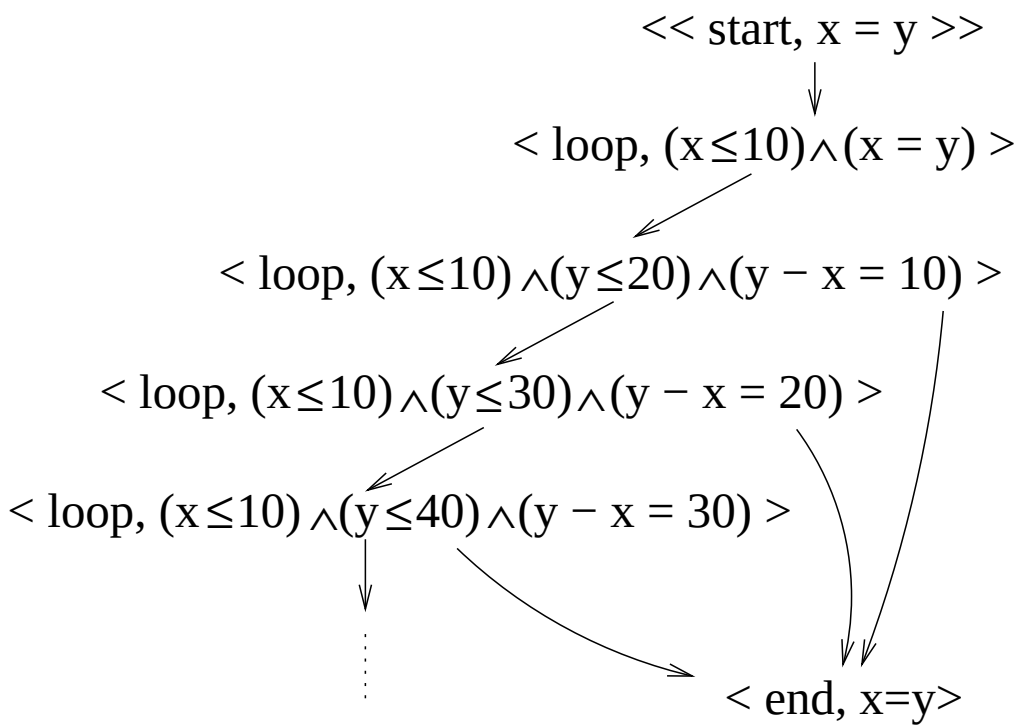
- $< l_0, \{u_0\} > \rightsquigarrow^* < l_f, D_f > \Rightarrow < l_0, u_0 > \rightarrow^* < l_f, u_f >$ für alle $u_f \in D_f$
- $< l_0, u_0 > \rightarrow^* < l_f, u_f > \Rightarrow < l_0, \{u_0\} > \rightsquigarrow^* < l_f, D_f >$ für ein $D_f, u_f \in D_f$

korrekt und vollständig

- $< l_0, \{u_0\} > \rightsquigarrow^* < l_f, D_f > \Rightarrow < l_0, u_0 > \rightarrow^* < l_f, u_f >$ für alle $u_f \in D_f$
- $< l_0, u_0 > \rightarrow^* < l_f, u_f > \Rightarrow < l_0, \{u_0\} > \rightsquigarrow^* < l_f, D_f >$ für ein $D_f, u_f \in D_f$

Aber: $\rightsquigarrow$ ist immer noch unendlich


start
x:=0
y:=0
loop
x<=10
x==10
x:=0
y>=20
x:=0
y:=0
end


<< start, x = y >>
< loop, (x≤10)∧(x = y) >
< loop, (x≤10)∧(y≤20)∧(y − x = 10) >
< loop, (x≤10)∧(y≤30)∧(y − x = 20) >
< loop, (x≤10)∧(y≤40)∧(y − x = 30) >
< end, x=y>

Normalisierung

Normalisierung

Definition

$norm_{k,\mathcal{G}}(D) = \{u | u \dot{\sim}_{k,\mathcal{G}} v, v \in D\}$

$u \sim_{k,\mathcal{G}} v$, wenn:

- $u \sim_k v$, d.h. in der selben Region, und
- für alle $g \in \mathcal{G}, u \in g \Leftrightarrow v \in g$

$\rightsquigarrow_{k,\mathcal{G}}$ ist korrekt, vollständig und endlich

Entscheidbarkeit

- Inklusion: i.A. unentscheidbar
- Bisimulation: entscheidbar
- Erreichbarkeit: entscheidbar

Fragen?