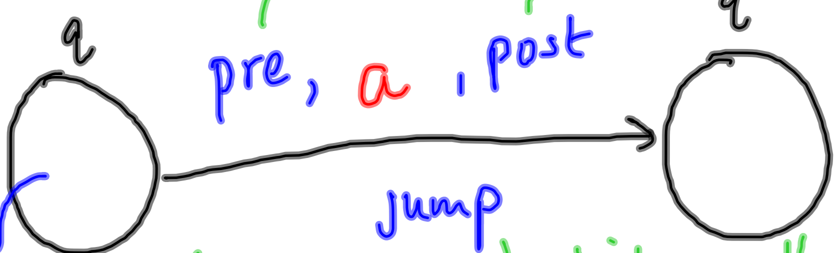
Hybrid Automata

Clocks $\rightarrow$ arbitrary real valued variables

$\dot{x}$ depends on state

guard   reset - can be nondet



which variables are reset $\in 2^X$

init(q) - initial values of each var if q is initial state

inv(q) - invariant conditions for values

flow(q) - specifies $\dot{x}$ for each x

What's decidable about hybrid automata?

Henzinger et al, JCSS

$X = \{\underbrace{x_1, x_2, \ldots, x_n}_{\subseteq \mathbb{R}^n}\}$ $V : X \to \mathbb{R}$

Rectangular hybrid automata

$R \subseteq \mathbb{R}^n$ is a rectangle if it is a product of intervals

Rectangular hybrid automaton

flow(q) pre(t)
inv(q) post(t)
init(q)

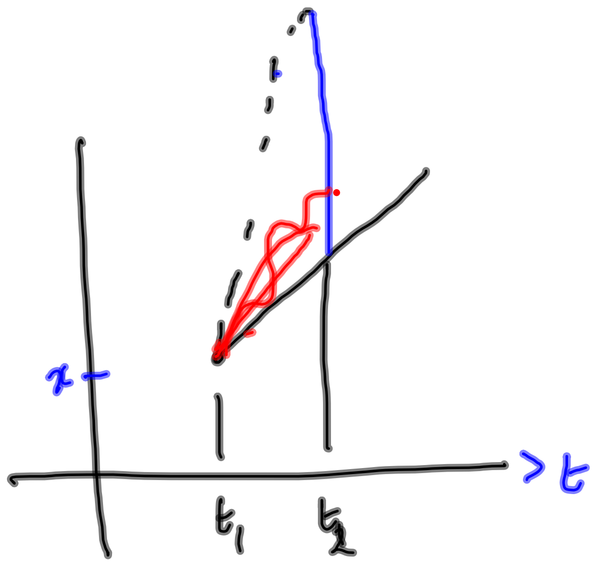
a conjunction of one rectangular constraint per variable

post(t) : $x = 3y + 7z + 8y^3$ ✗

$x \in [l, u)$

$flow(x) \in [1,5)$

effectively assume evolution is piecewise linear



Initialized automaton

$q \xrightarrow{a} q'$ if $flow(q)(x) \neq flow(q')(x)$

then $x \in jump(t)$

Whenever $\dot{x}$ changes, x is reset

Initialized + rectangular $\Leftrightarrow$ positive

Deterministic reset

Reachability is decidable for initialized rectangular automata

reachability of (q, Z)

where Z is a "zone" or rectangle

Initialized Rectangular

$\downarrow$

Initialized Singular – each $\dot{x}$ is a constant
deterministic resets.

$\downarrow$

Initialized Stopwatch $\dot{x} \in \{0,1\}$
deterministic resets

$\downarrow$

Timed Automata with constant resets

Initialized Stopwatch

$\downarrow$

Timed Automata

$(q, x_1, x_2, \ldots, x_n, \text{some } \dot{x}_i = 0)$

if $\dot{x}_i = 0$, x_i is post(t)
for last transition when
$\dot{x}_i$ went from 1 to 0

$(q, x_1, \ldots, x_n, f : X \to \mathbb{R} \cup \{\bot\})$

If $\dot{x}_i = 1$, same value as stopwatch and $f(x) = \bot$

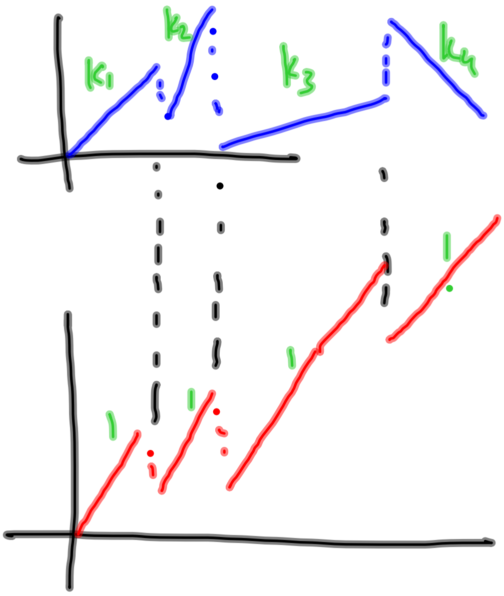
If $\dot{x}_i = 0$, $f(x)$ = stopped value of x

Initialized singular

Initialized stopwatch

$\forall q \; \exists k \quad \dot{x} \in [k,k]$ in q.

det. jumps



Simple scaling relates configurations

Initialized Rectangular

↓

Initialized Singular

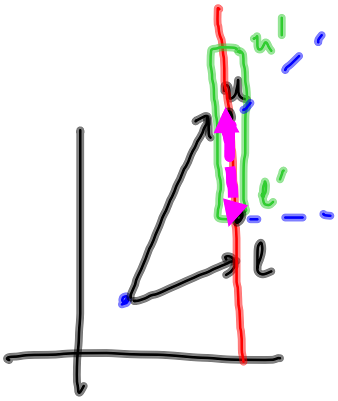
$\dot{x} \in [l, u]$

$\dot{x}_l = l$ $\dot{x}_u = u$

"Compact" rectangles

"

bounded & closed



$x_l \le u'$

$x_u \ge l'$

Relaxing conditions yields undecidability

Timed automata are decidable – one-slope variables "initialized" by default, det jumps

2 slope variables $\dot{x} = k_1$ or $\dot{x} = k_2$
but not initialized

"Simple" rectangular automaton

- exactly one x is not a clock
- initially all variables are 0, all resets to 0
- all rectangles are compact (bounded & closed)

Thm: Reachability is undecidable for simple automata with one two-slope variable.

$\exists y$ s.t. $\dot{y} \in \{k_1, k_2\}$ in each state

2 counter machines

Instructions

1:

2: ← $c_1{++}$, $c_1{--}$

$c_2{++}$, $c_2{--}$

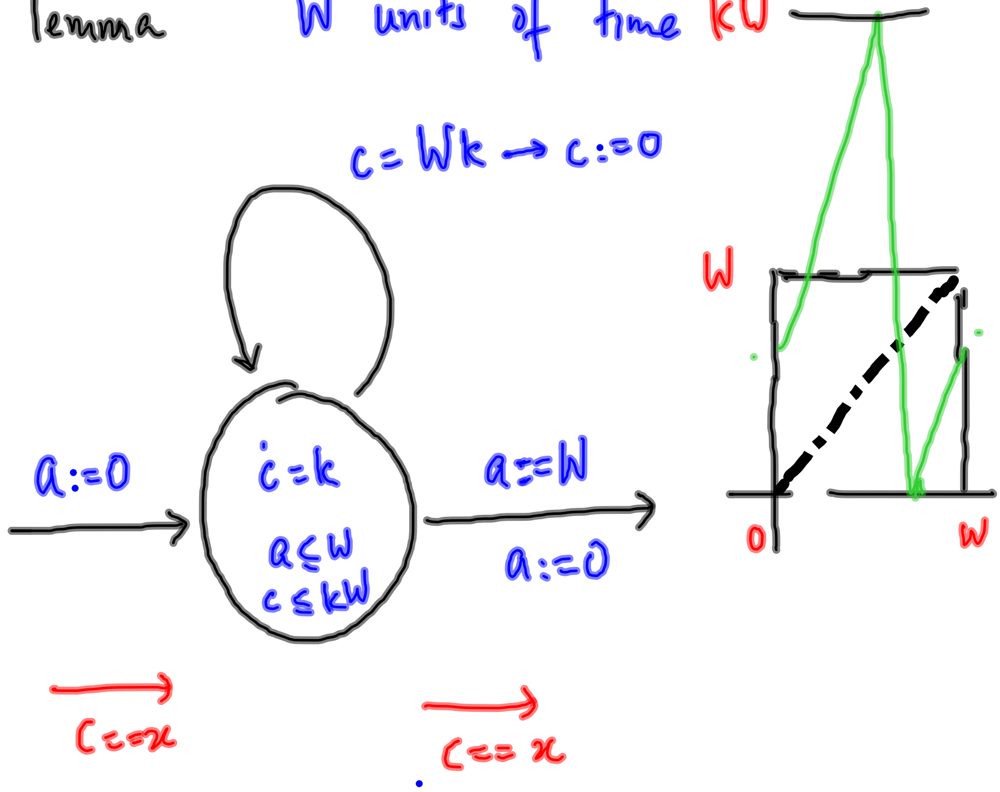
⋮

if $c_i == 0$ goto ℓ

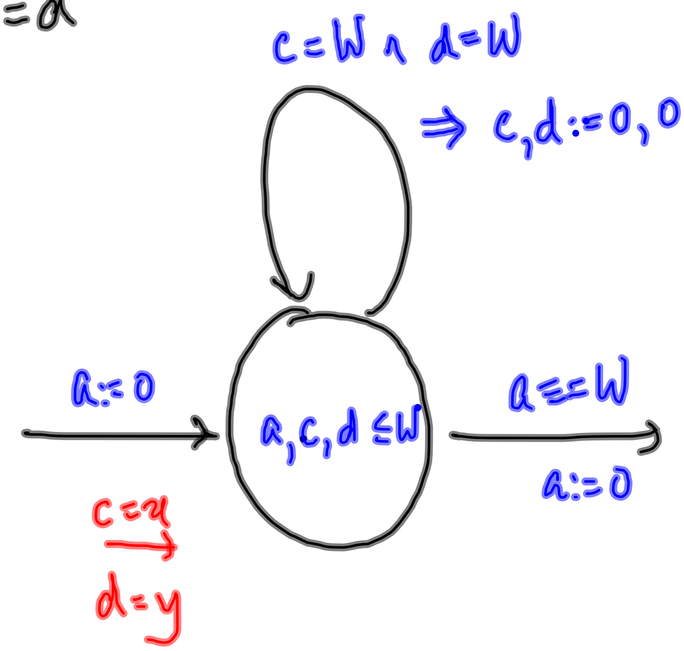
n: ← halt

"Wrapping" lemma

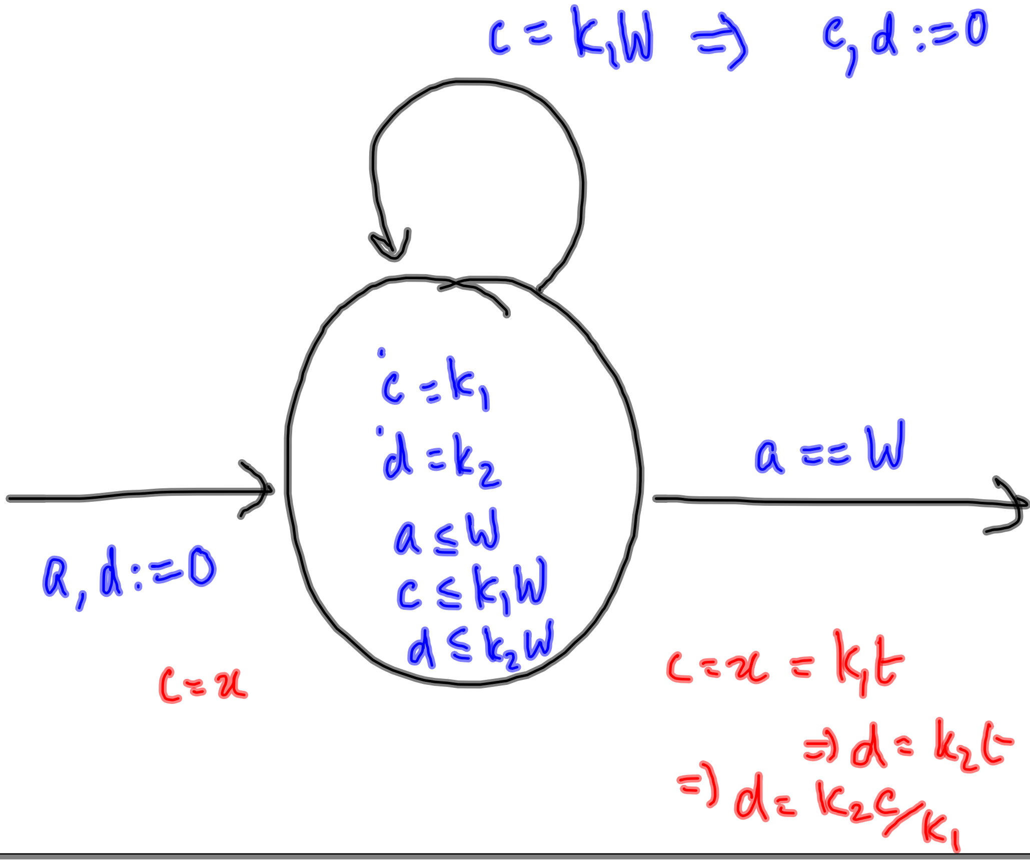
W units of time kW



Testing c=d



Two non clocks



Simulation of 2 counter machine

Counters C — clock c

D — clock d

3 more clocks a, b, b'

One non-clock z – slopes. k_1, k_2

$k_1 > k_2 > 0$

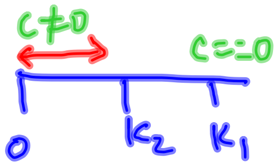
Value of $C = u$ is encoded $c = k_1 \cdot \left(\frac{k_2}{k_1}\right)^u$

$D = u$ is encoded $d = k_1 \cdot \left(\frac{k_2}{k_1}\right)^u$

$C = u \qquad c = k_1 \cdot \left(\frac{k_2}{k_1}\right)^u$

$C = 0 \qquad c = k_1$

$C \geq 1 \qquad c \in (0, k_2]$



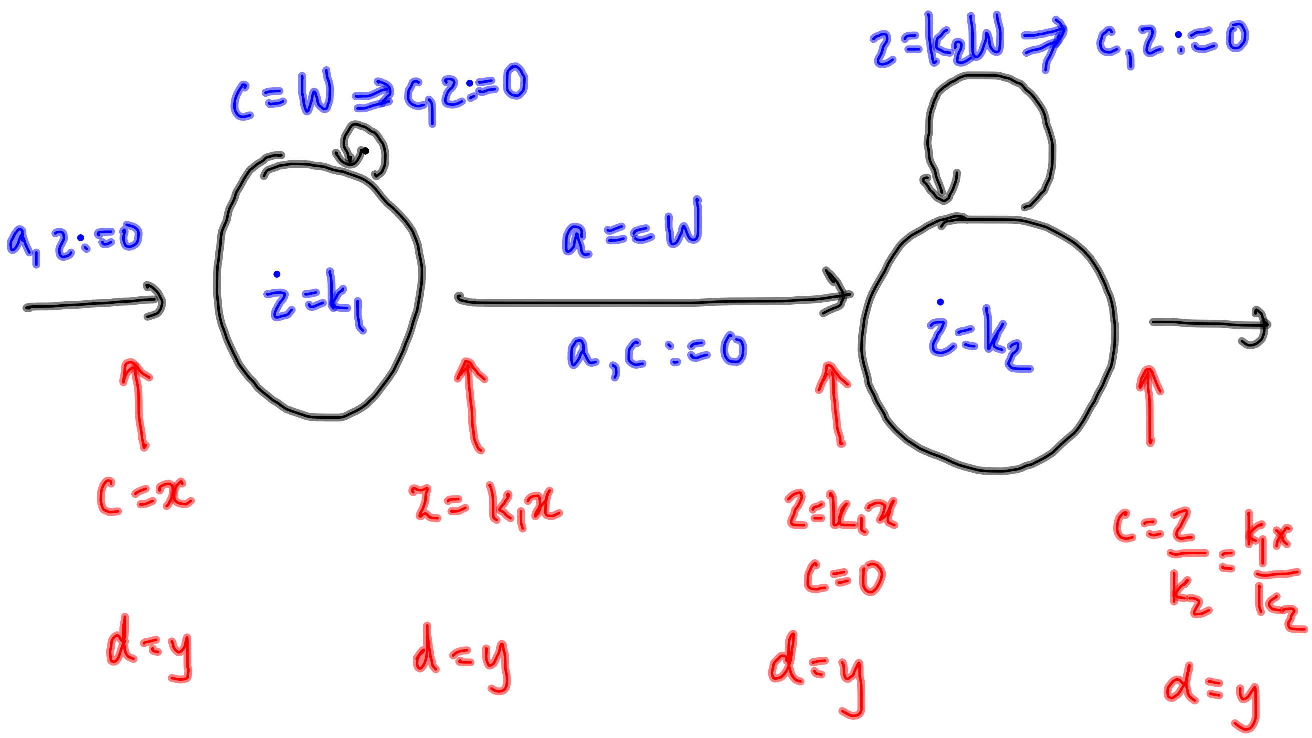
if $C == 0 \qquad c \in [k_1, k_1]$

$C \,!= 0 \qquad c \in (0, k_2]$

decrement / increment — multiply by $\frac{k_2}{k_1}$

divide by $\frac{k_2}{k_1}$ = multiply by $\frac{k_1}{k_2}$

Multiply by k_1/k_2



$k_1 > 0 > k_2$

$k_1 > 0$, $k_2 = 0$

$\vdots$

Different encodings of C, D

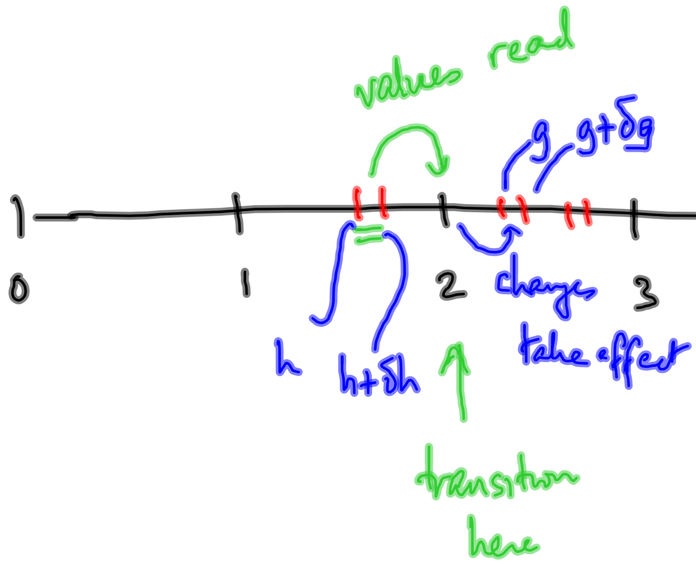
Non exactness does not help

Alternative "practical" approach

- Discretize time
- Inaccuracy of time of measurement
- Delay in effecting changes

"lazy" hybrid automata

Agrawal & Thiagarajan



$$g < g+\delta g < h < h+\delta h$$