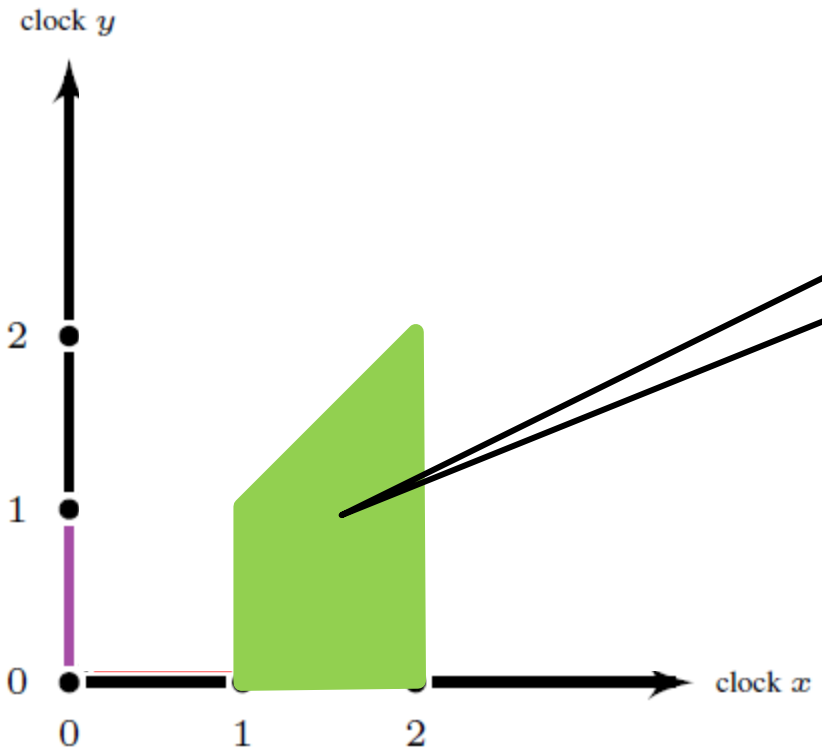
Symbolic Verification

The UPPAAL Verification Engine

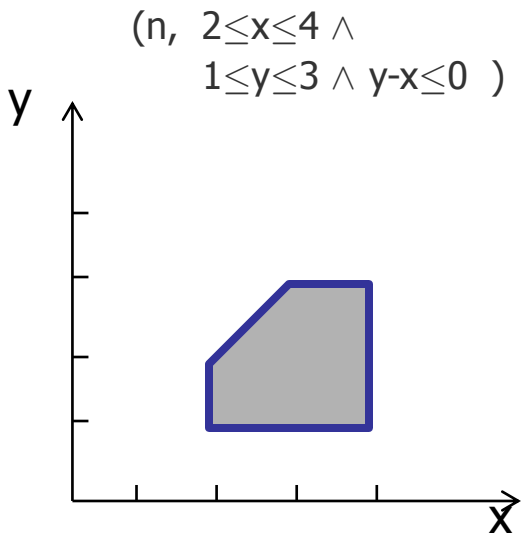
Zones – From Finite to Efficiency

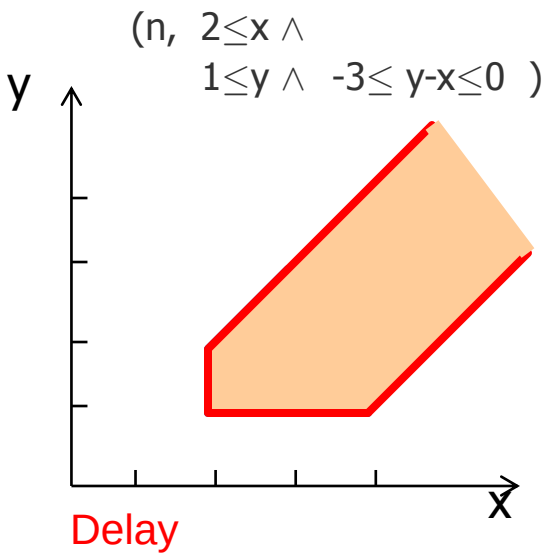


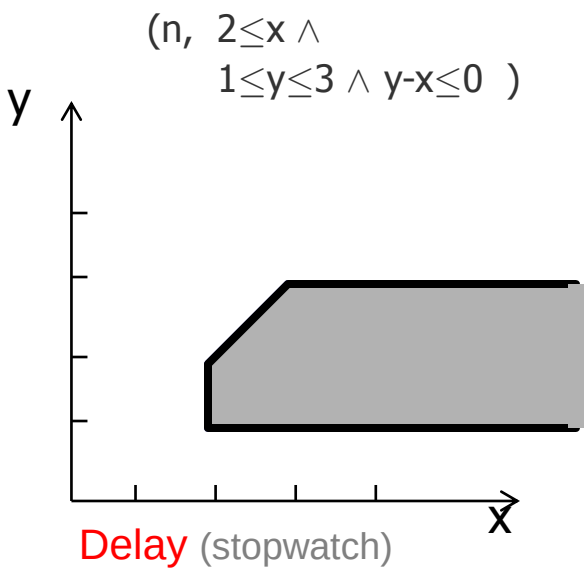
A zone Z:

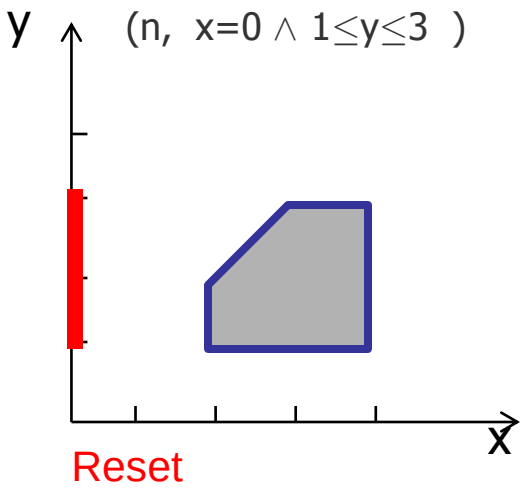
$1 \leq x \leq 2 \ \wedge$
$0 \leq y \leq 2 \ \wedge$
$x - y \geq 0$

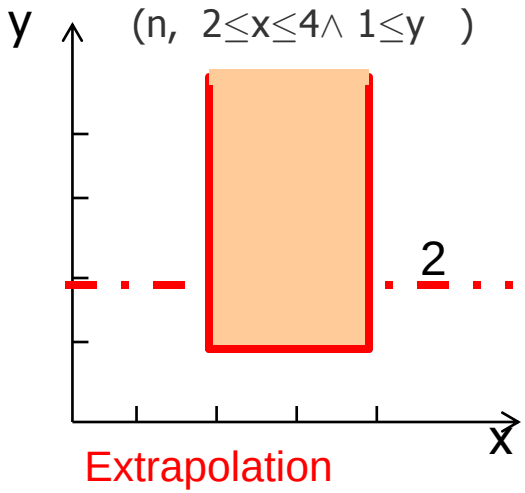
Zones – Operations

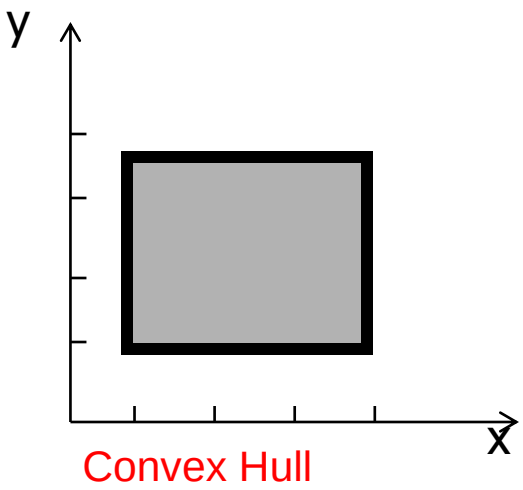




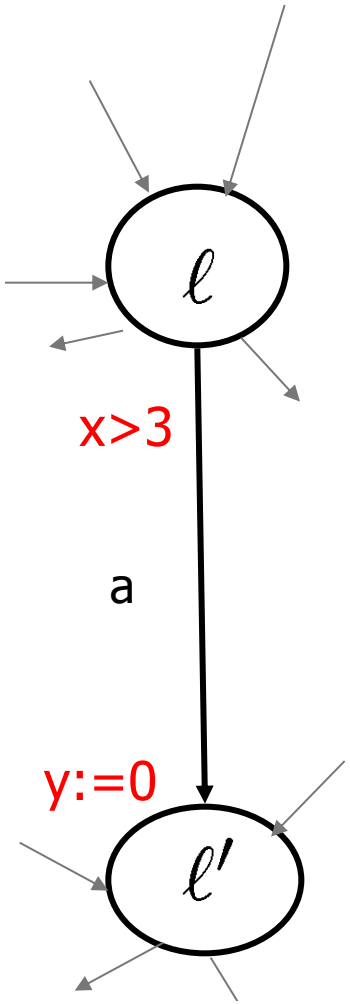


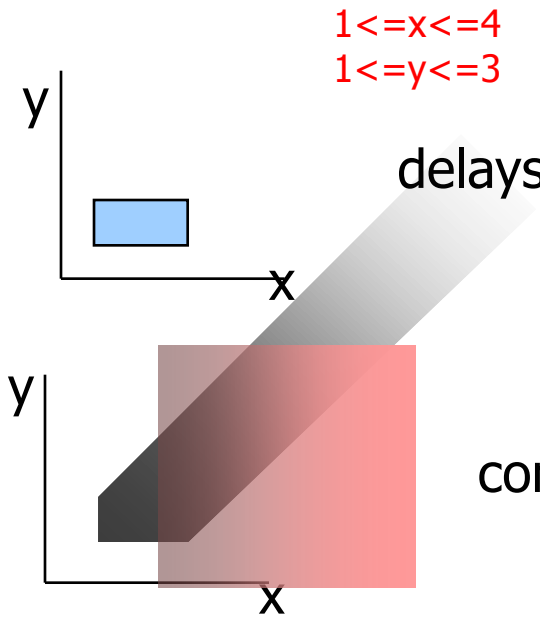


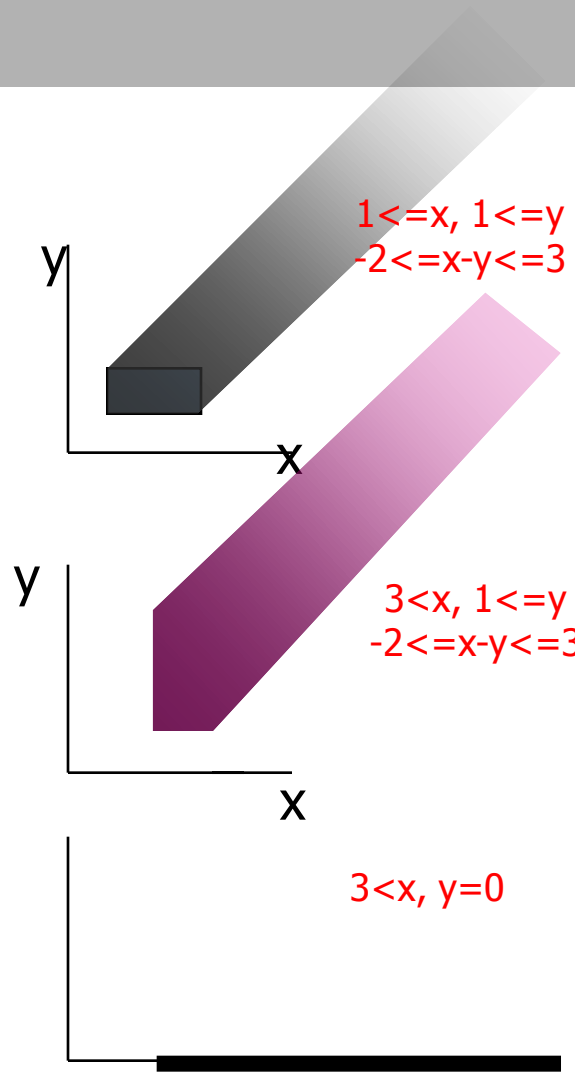




Symbolic Transitions



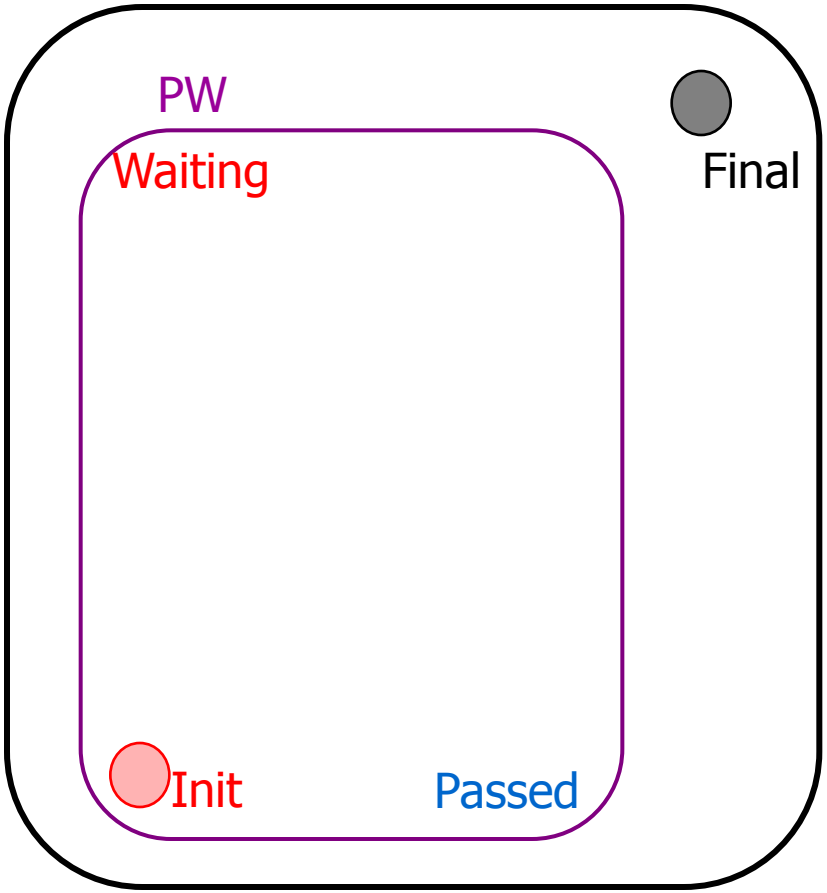




$$(\ell, 1 \leq x \leq 4, 1 \leq y \leq 3) \Rightarrow_a (\ell', 3 < x, y = 0)$$

Forward Reachability

Init -> Final ?



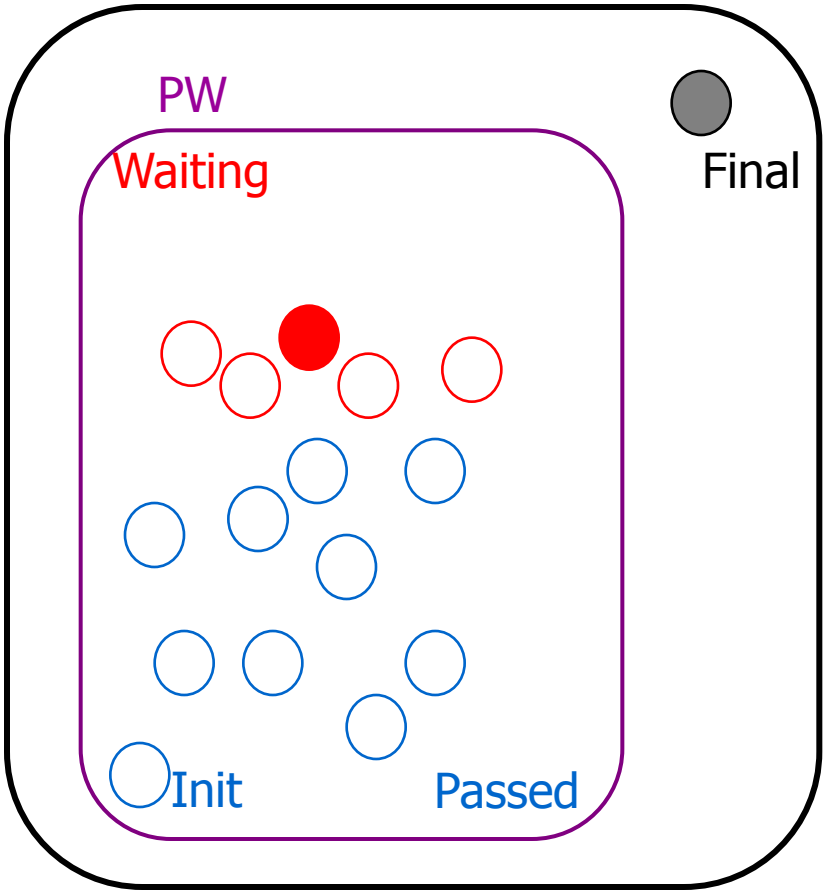
INITIAL Passed := Ø;
Waiting := $\{(n_0,Z_0)\}$

REPEAT
pick (n,Z) in Waiting
if (n,Z) = Final return true
for all $(n,Z) \rightarrow (n',Z')$:
if for some (n',Z'') $Z' \subseteq Z''$ continue
else add (n',Z') to Waiting
move (n,Z) to Passed

UNTIL Waiting = Ø
return false

Forward Reachability

Init -> Final ?



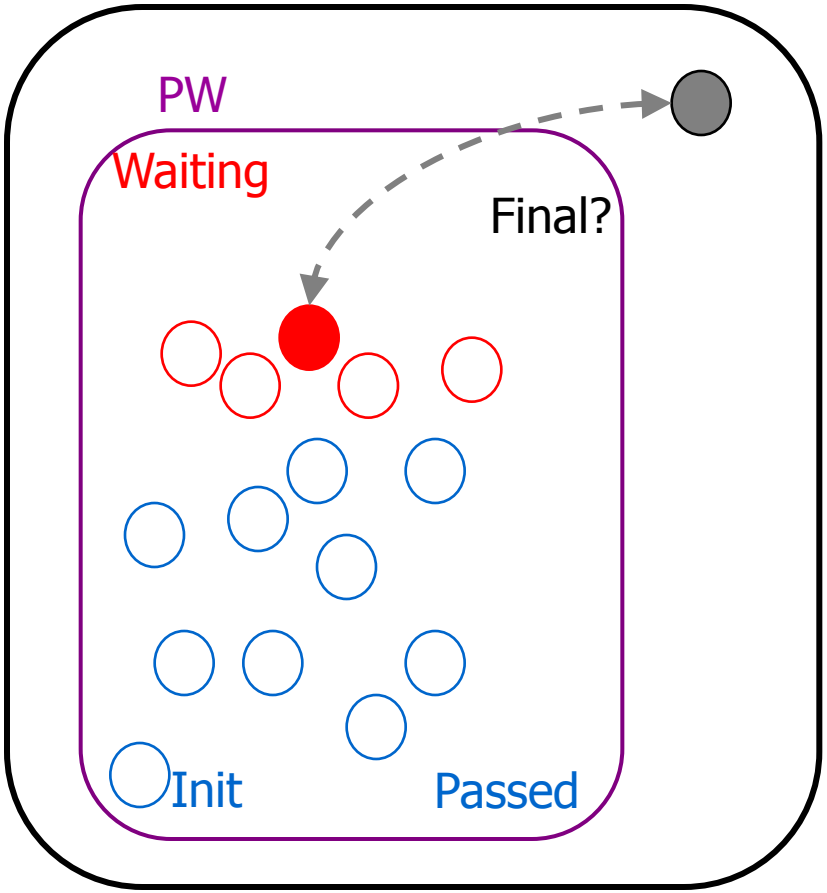
INITIAL Passed := Ø;
Waiting := $\{(n_0,Z_0)\}$

REPEAT
pick (n,Z) in Waiting
if (n,Z) = Final return true
for all $(n,Z) \rightarrow (n',Z')$:
if for some (n',Z'') $Z' \subseteq Z''$ continue
else add (n',Z') to Waiting
move (n,Z) to Passed

UNTIL Waiting = Ø
return false

Forward Reachability

Init -> Final ?



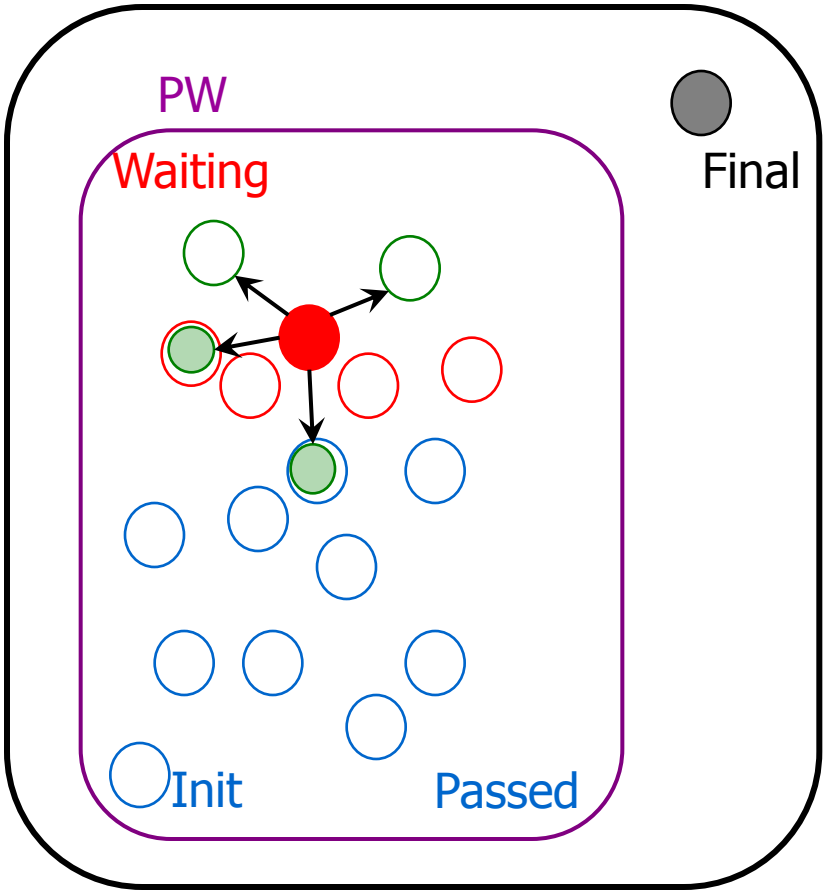
INITIAL Passed := Ø;
Waiting := $\{(n_0,Z_0)\}$

REPEAT
pick (n,Z) in Waiting
if (n,Z) = Final return true
for all $(n,Z) \rightarrow (n',Z')$:
if for some (n',Z'') $Z' \subseteq Z''$ continue
else add (n',Z') to Waiting
move (n,Z) to Passed

UNTIL Waiting = Ø
return false

Forward Reachability

Init -> Final ?



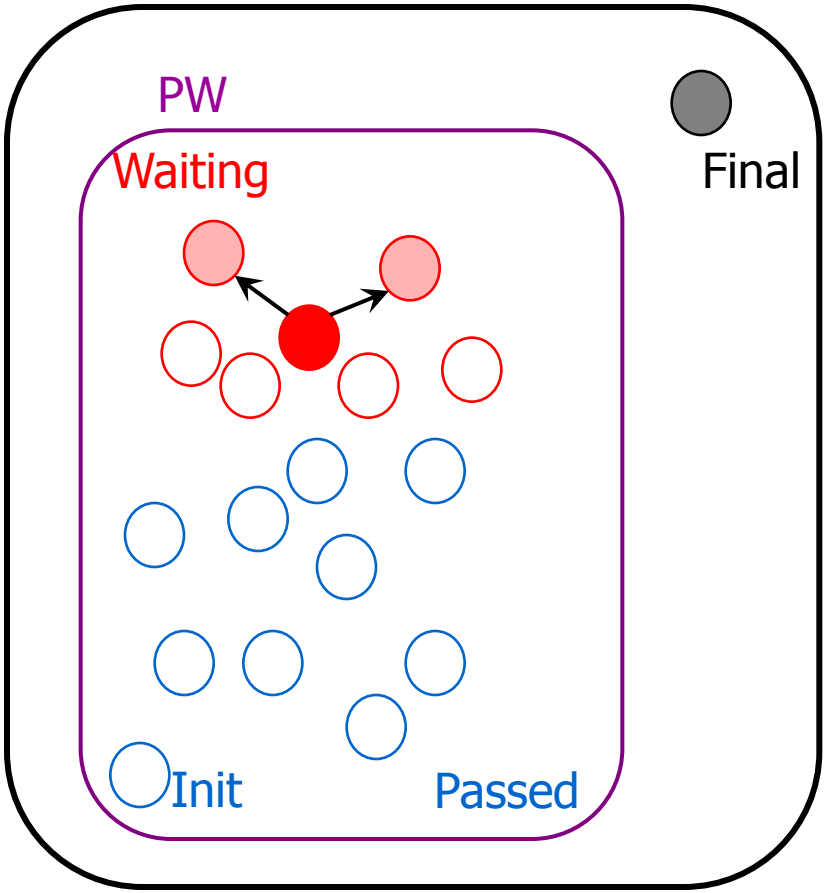
INITIAL Passed := Ø;
Waiting := $\{(n_0,Z_0)\}$

REPEAT
pick (n,Z) in Waiting
if (n,Z) = Final return true
for all (n,Z)→(n',Z'):
if for some (n',Z'') $Z' \subseteq Z''$ continue
else add (n',Z') to Waiting
move (n,Z) to Passed

UNTIL Waiting = Ø
return false

Forward Reachability

Init -> Final ?



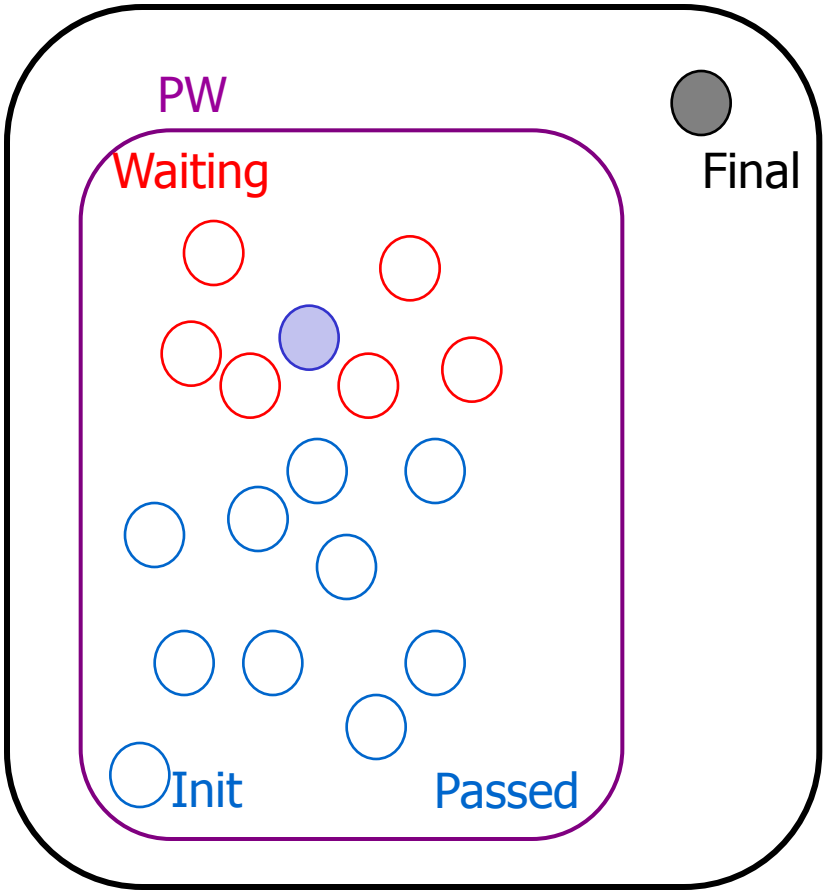
INITIAL Passed := Ø;
Waiting := $\{(n_0,Z_0)\}$

REPEAT
pick (n,Z) in Waiting
if (n,Z) = Final return true
for all (n,Z)→(n',Z'):
if for some (n',Z'') $Z' \subseteq Z''$ continue
else add (n',Z') to Waiting
move (n,Z) to Passed

UNTIL Waiting = Ø
return false

Forward Reachability

Init -> Final ?



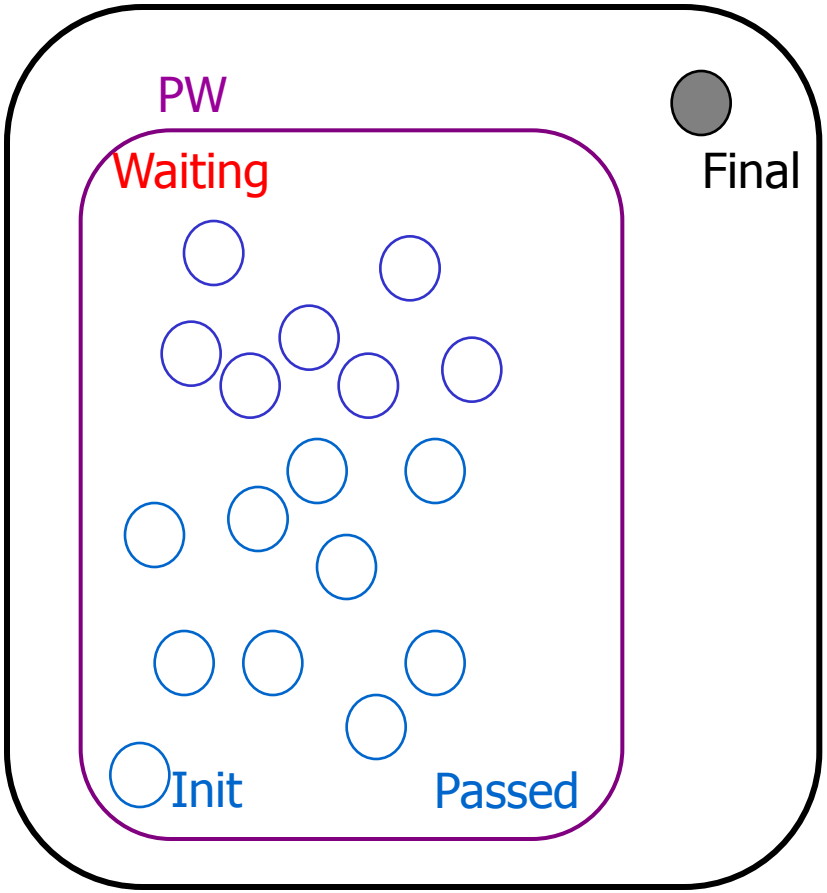
INITIAL Passed := Ø;
Waiting := $\{(n_0,Z_0)\}$

REPEAT
pick (n,Z) in Waiting
if (n,Z) = Final return true
for all (n,Z)→(n',Z'):
if for some (n',Z'') $Z' \subseteq Z''$ continue
else add (n',Z') to Waiting
move (n,Z) to Passed

UNTIL Waiting = Ø
return false

Forward Reachability

Init -> Final ?

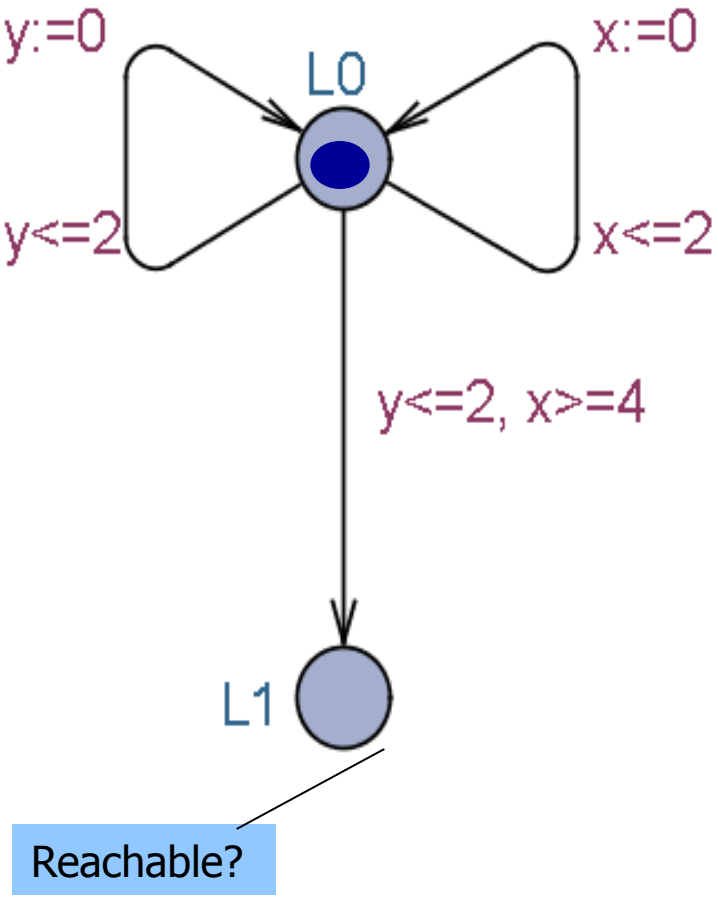


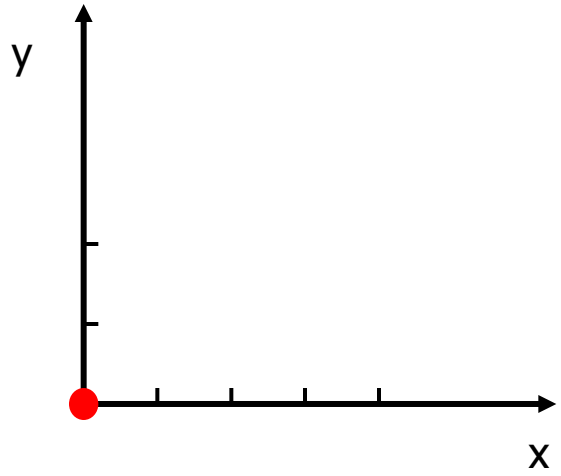
INITIAL Passed := Ø;
Waiting := $\{(n_0,Z_0)\}$

REPEAT
pick (n,Z) in Waiting
if (n,Z) = Final return true
for all (n,Z)→(n′,Z′):
if for some (n′,Z″) $Z' \subseteq Z''$ continue
else add (n′,Z′) to Waiting
move (n,Z) to Passed

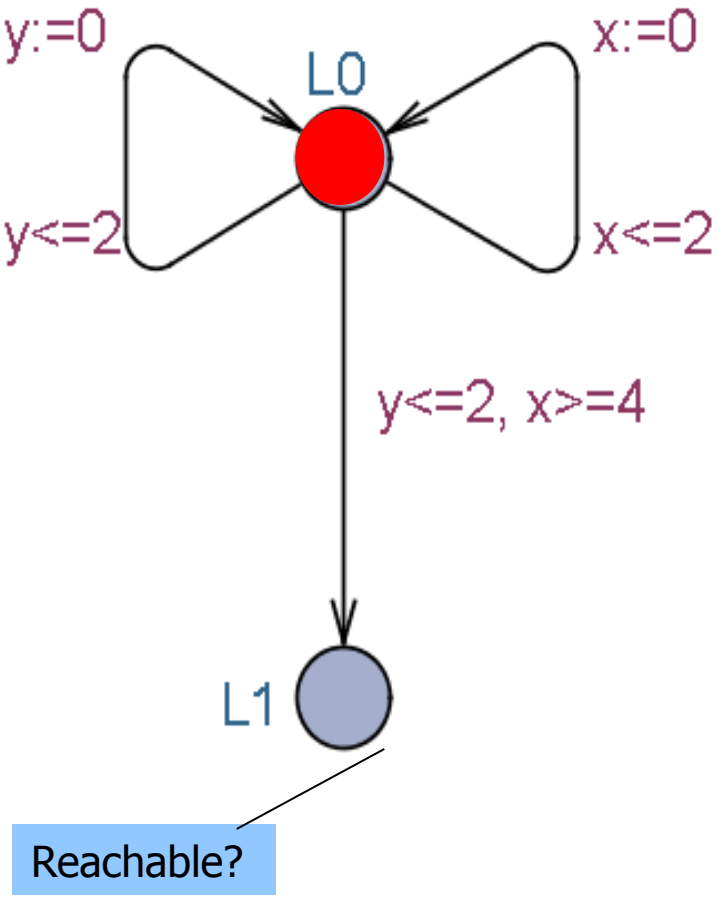
UNTIL Waiting = Ø
return false

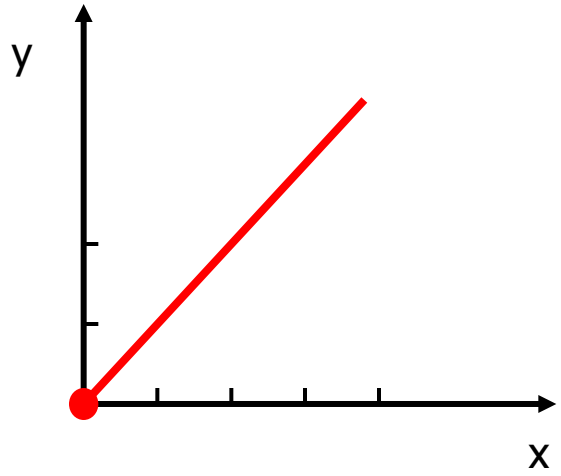
Symbolic Exploration





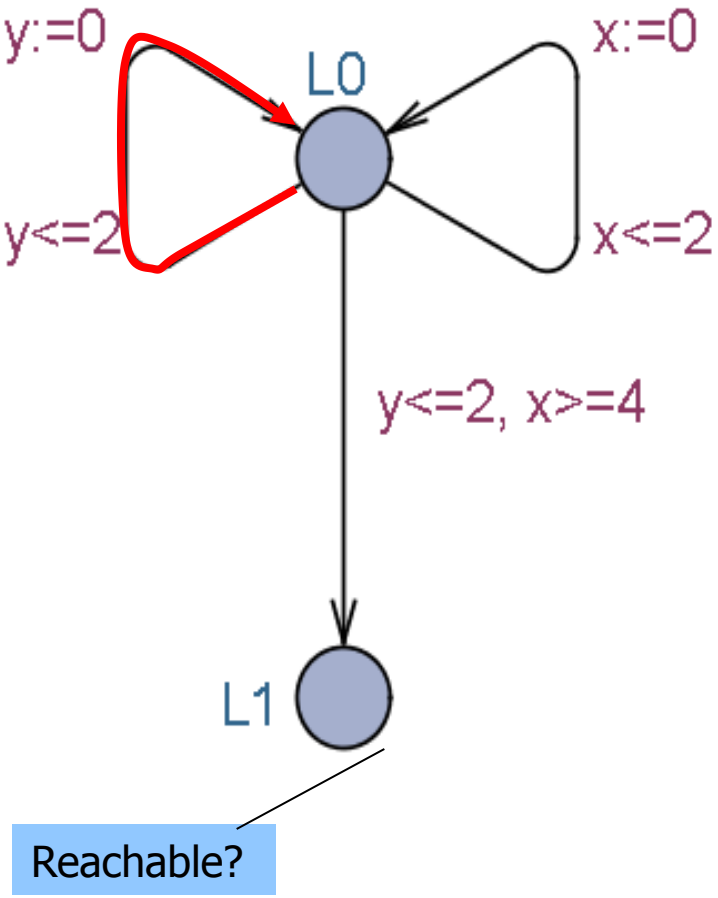
Symbolic Exploration

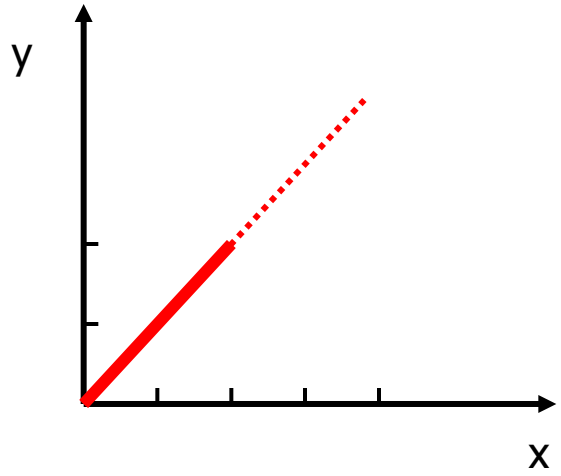




Delay

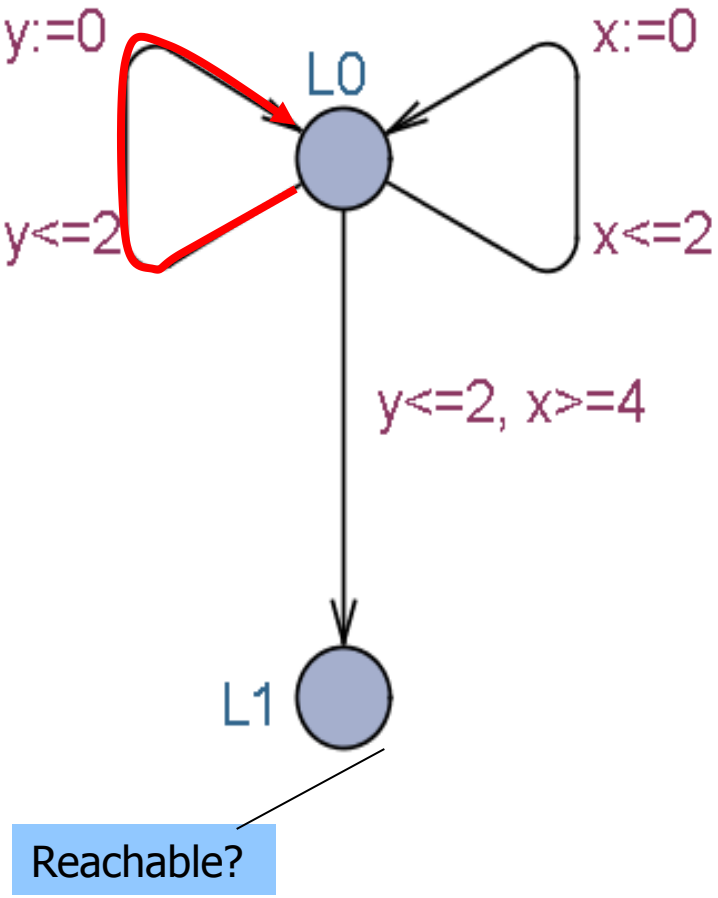
Symbolic Exploration

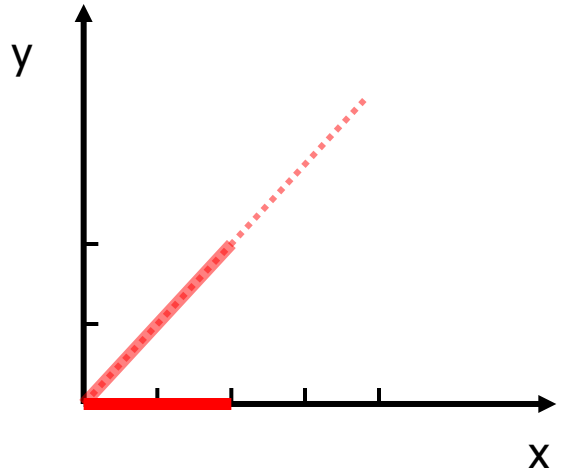




Left

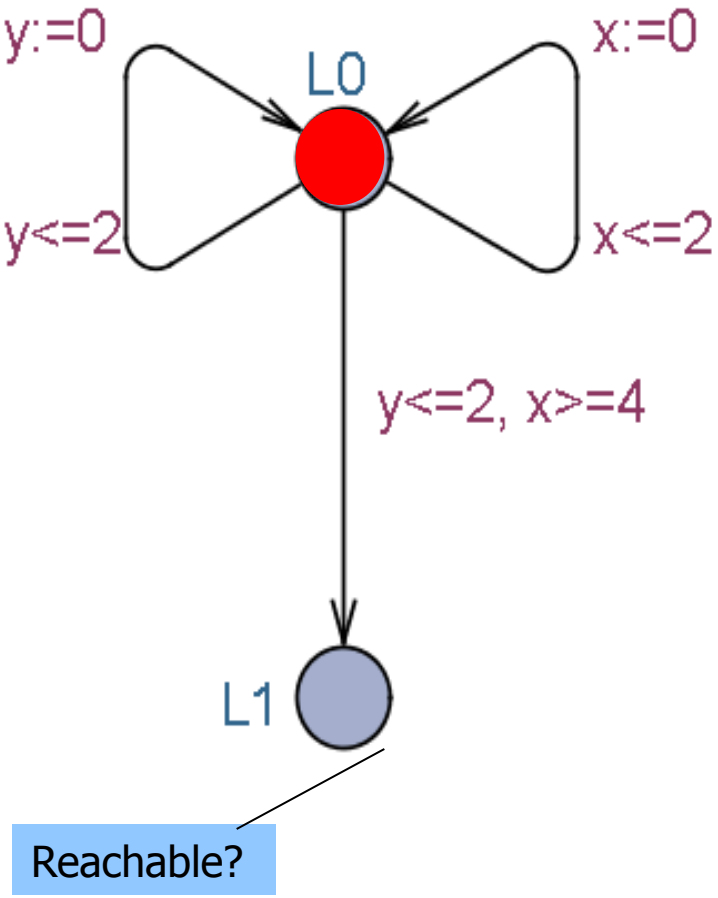
Symbolic Exploration

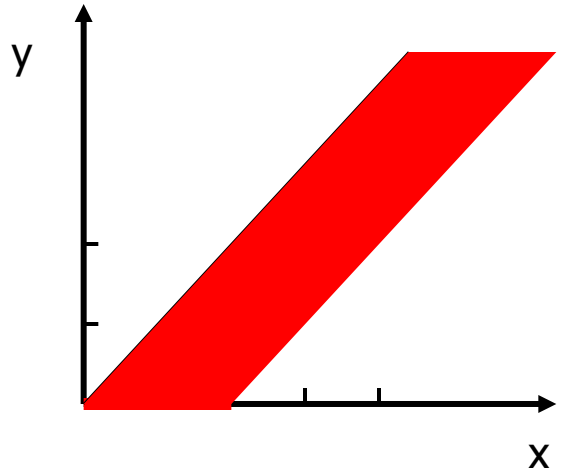




Left

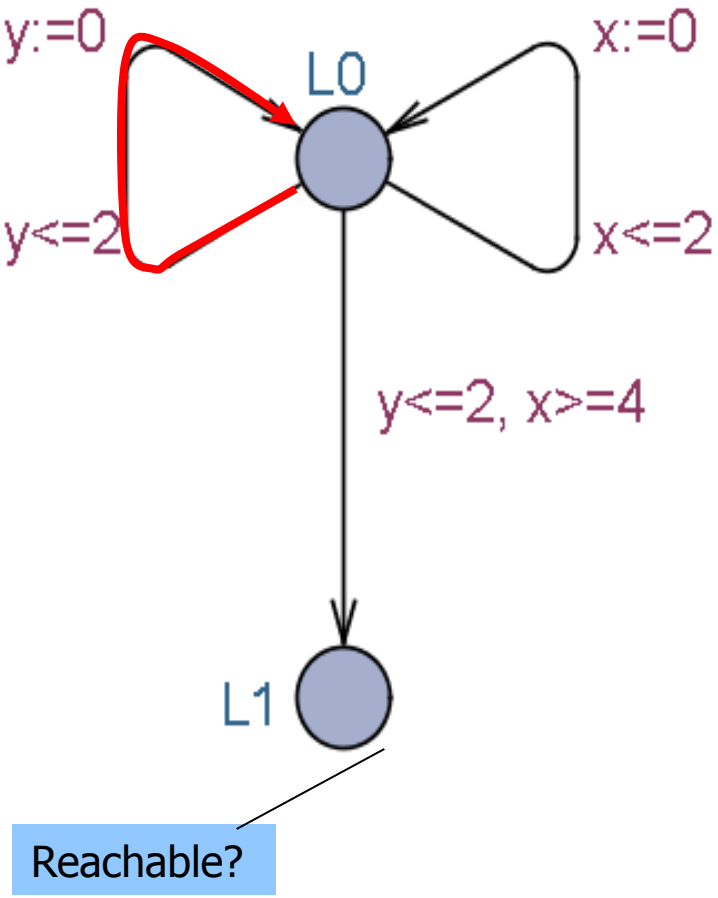
Symbolic Exploration

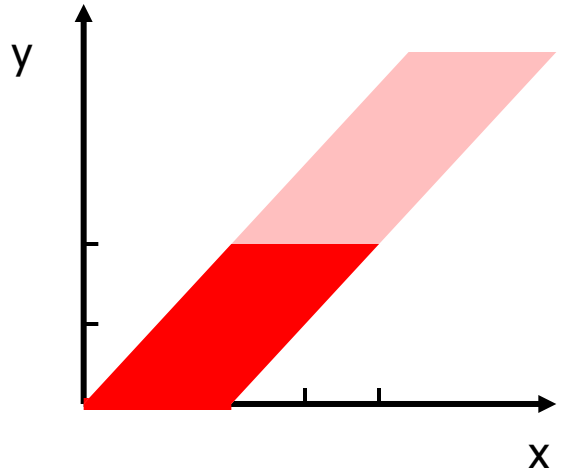




Delay

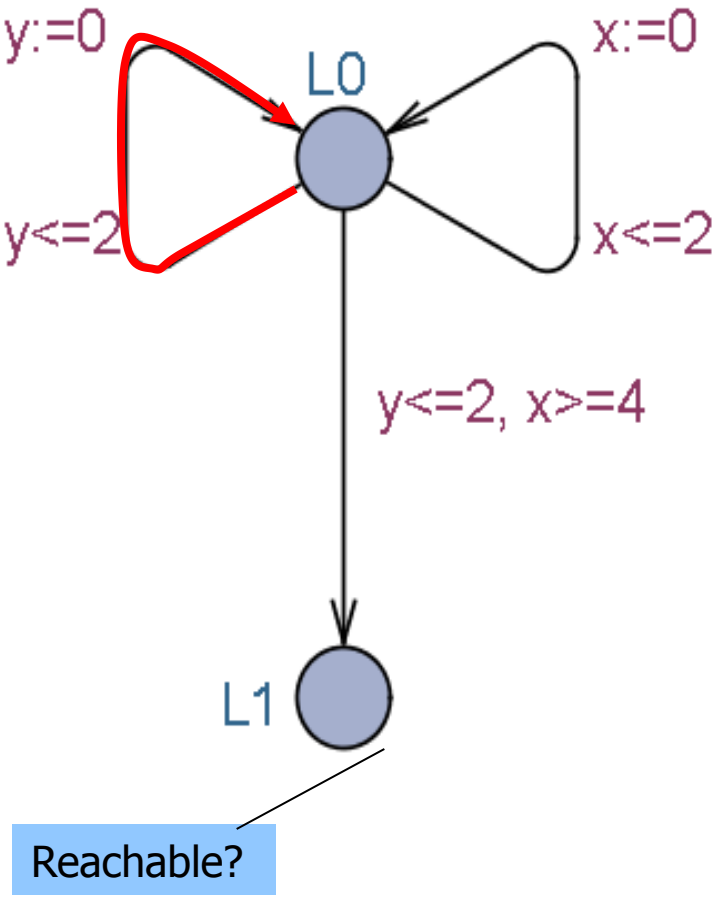
Symbolic Exploration

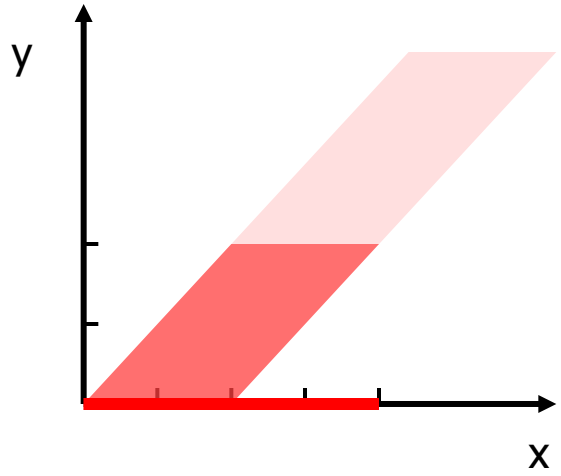




Left

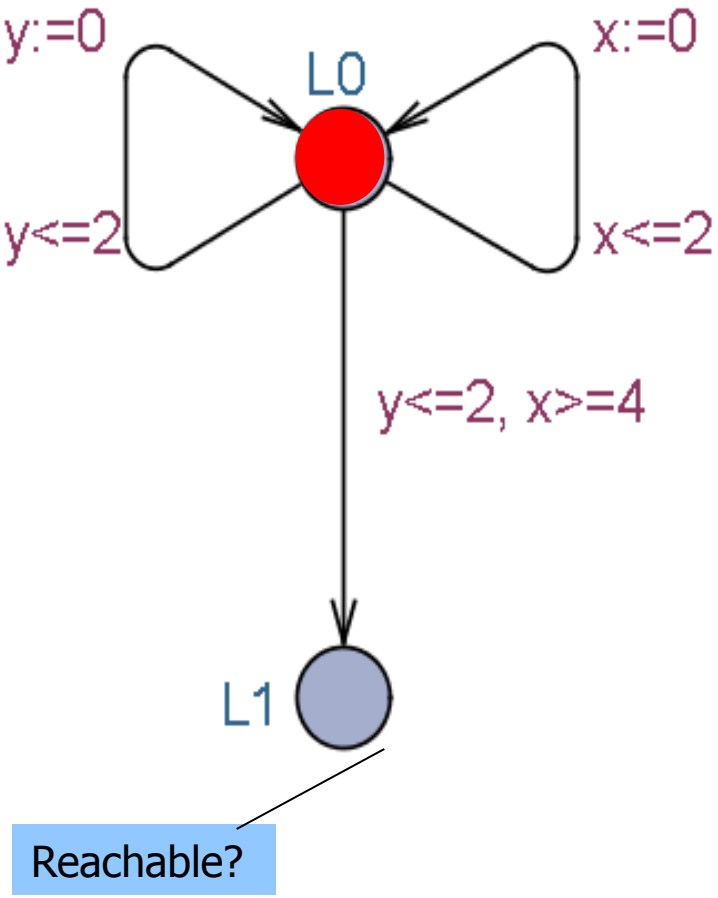
Symbolic Exploration

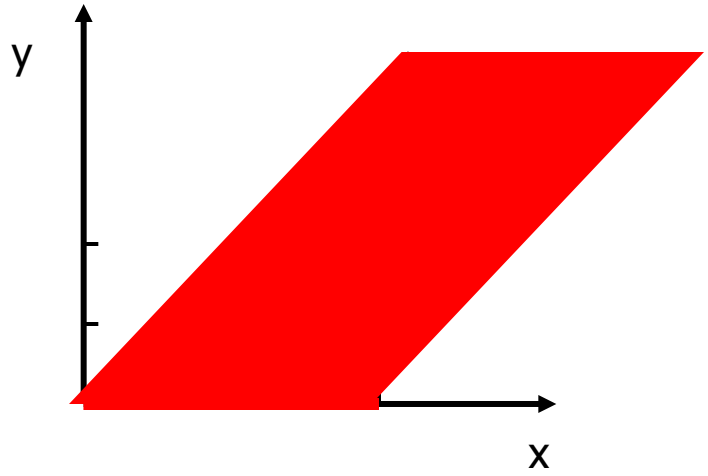




Left

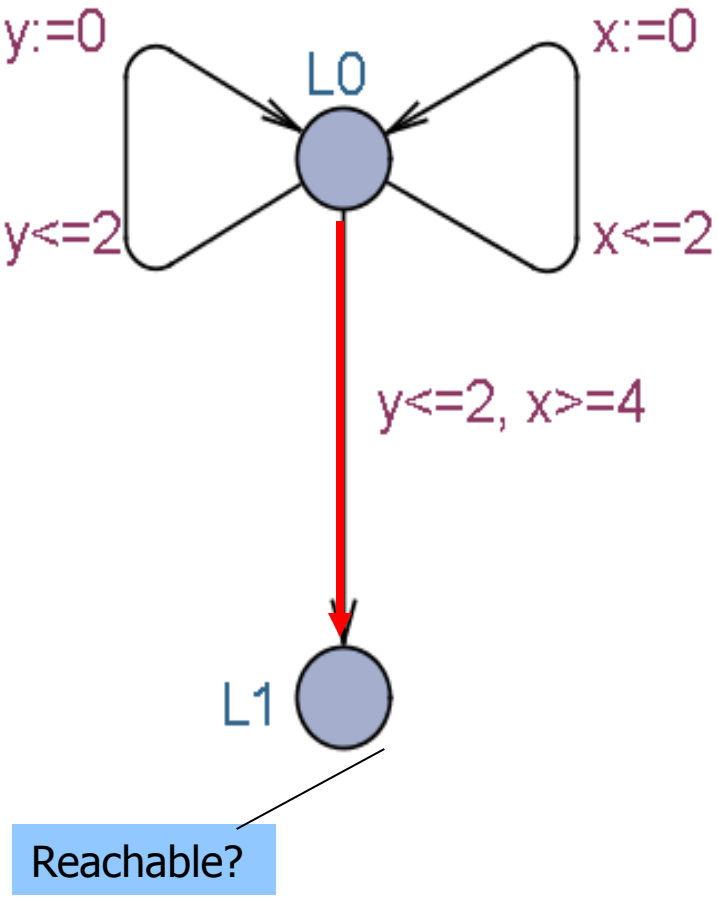
Symbolic Exploration

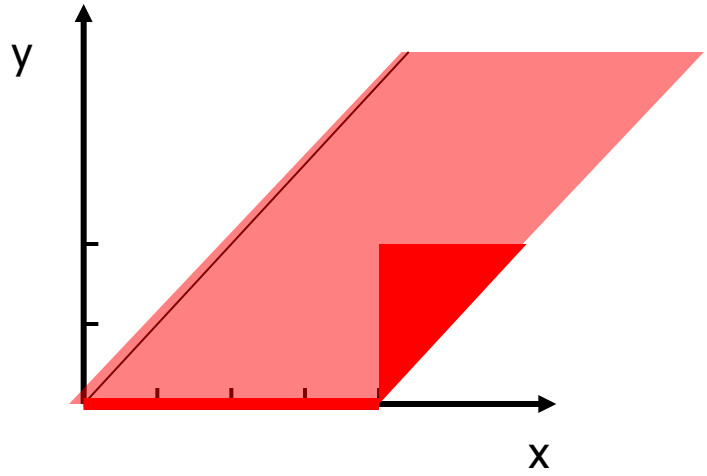




Delay

Symbolic Exploration

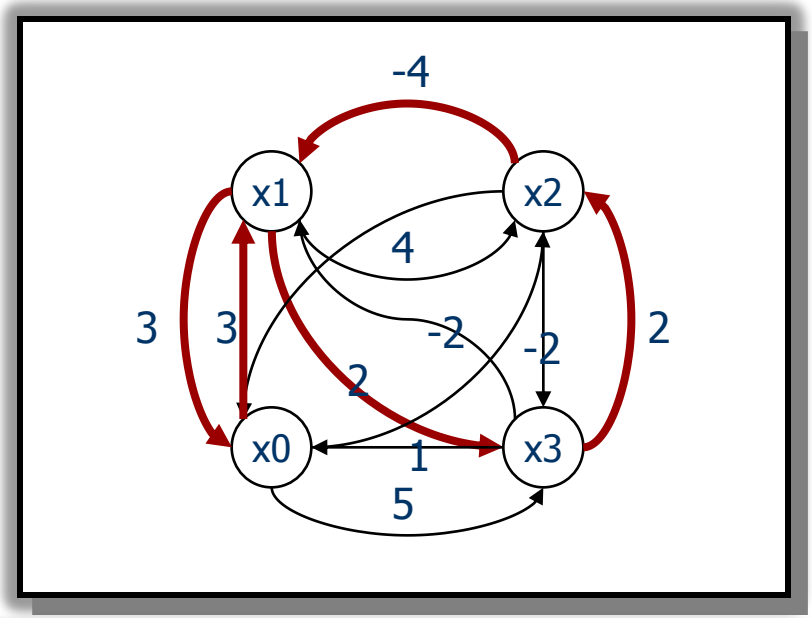


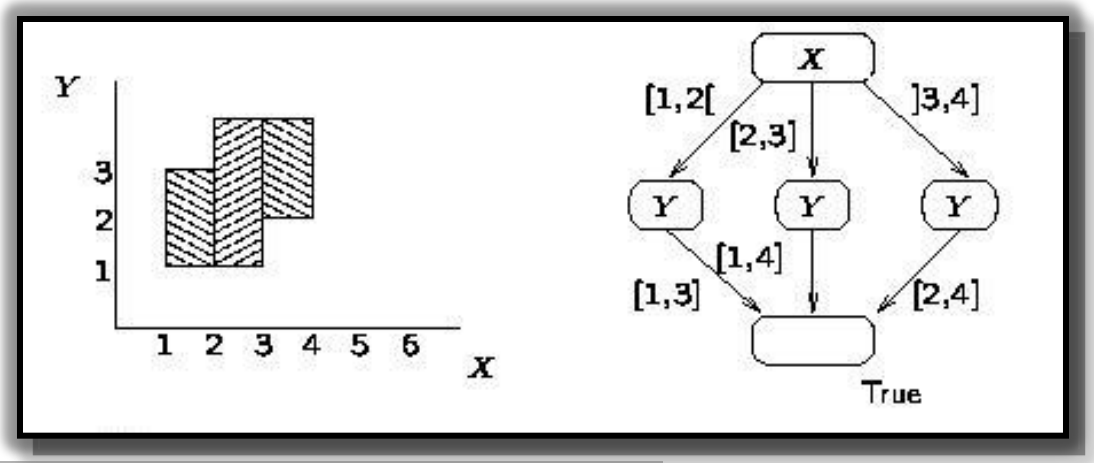


Down

Datastructures for Zones

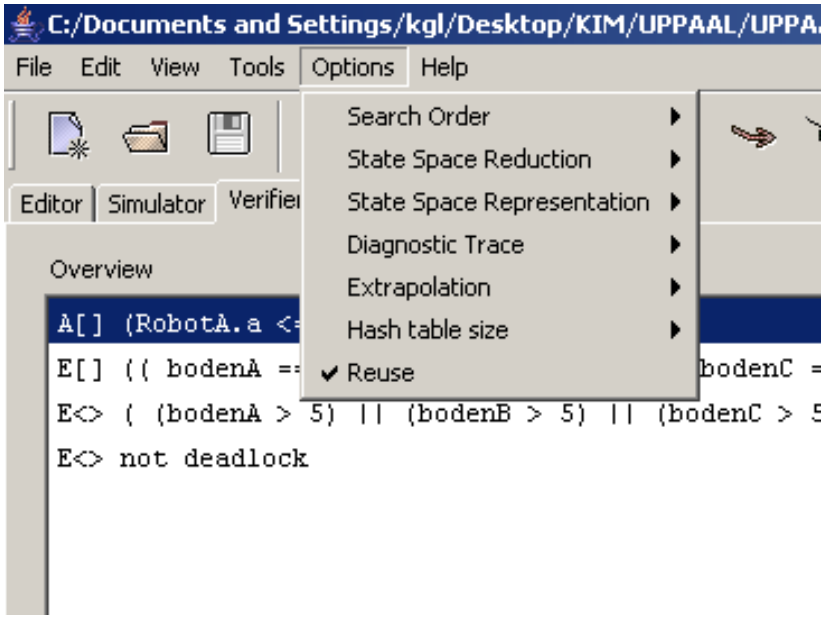
- Difference Bounded Matrices (DBMs)
- Minimal Constraint Form
  [RTSS97]
- Clock Difference Diagrams
  [CAV99]





Verification Options

Verification Options



Search Order
- Depth First
- Breadth First

State Space Reduction
- None
- Conservative
- Aggressive

State Space Representation
- DBM
- Compact Form
- Under Approximation
- Over Approximation

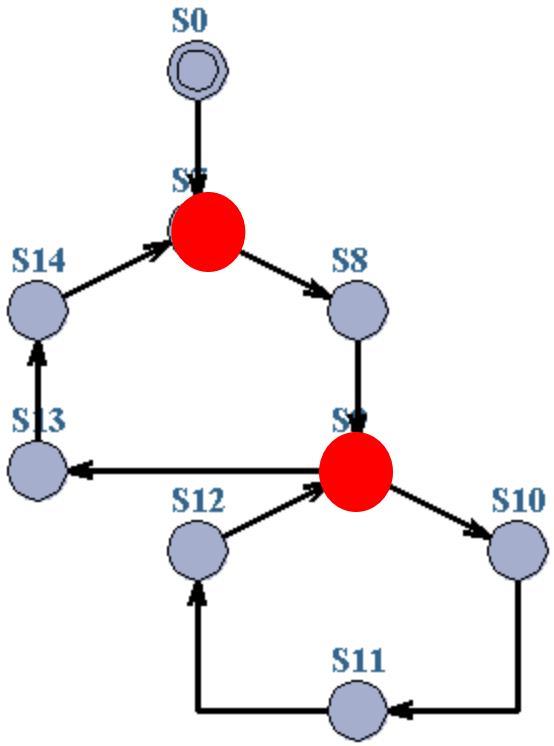
Diagnostic Trace
- Some
- Shortest
- Fastest

Extrapolation

Hash Table size

Reuse

State Space Reduction



Cycles:

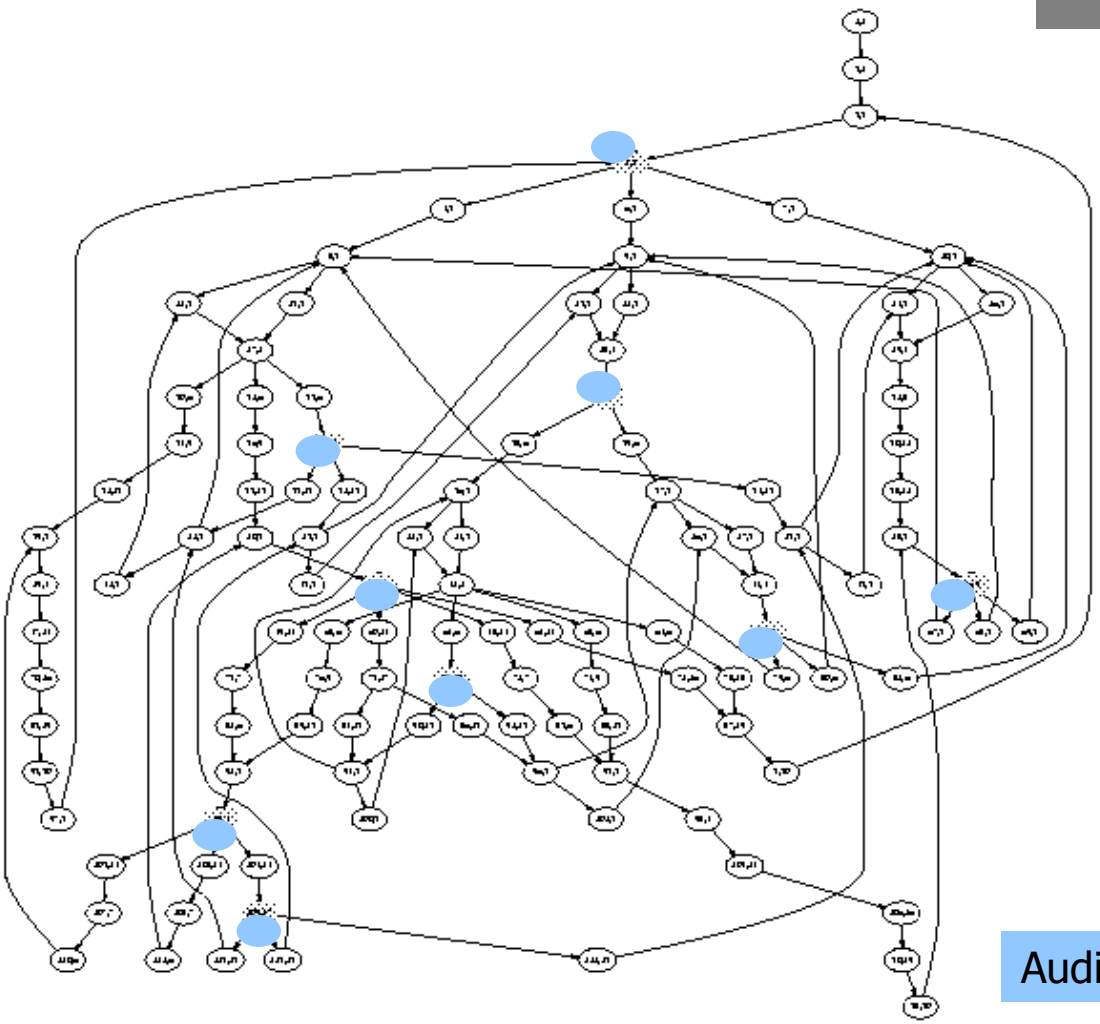
Only symbolic states involving loop-entry points need to be saved on Passed list

To Store or Not To Store

Behrmann, Larsen, Pelanek 2003

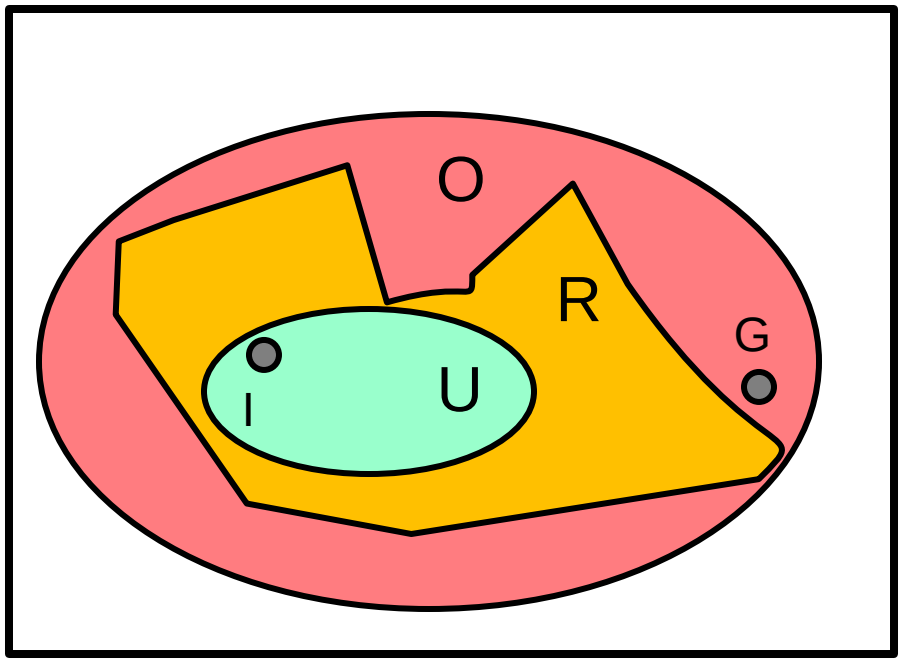
117 states$_{total}$
→
81 states$_{entrypoint}$
→
9 states

Time OH
less than 10%



Audio Protocol

Over/Under Approximation



Declared State Space

Question:

$G \in R$?

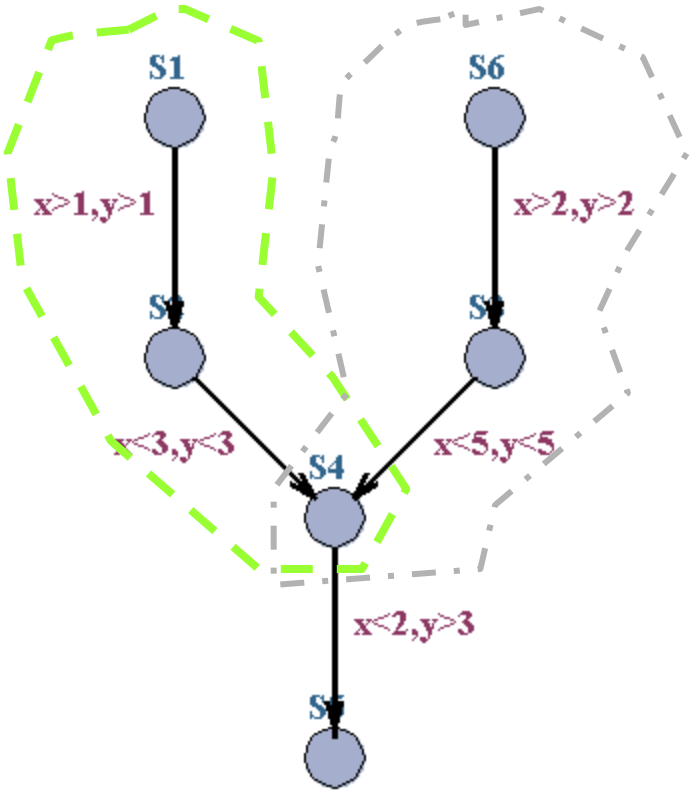
How to use:

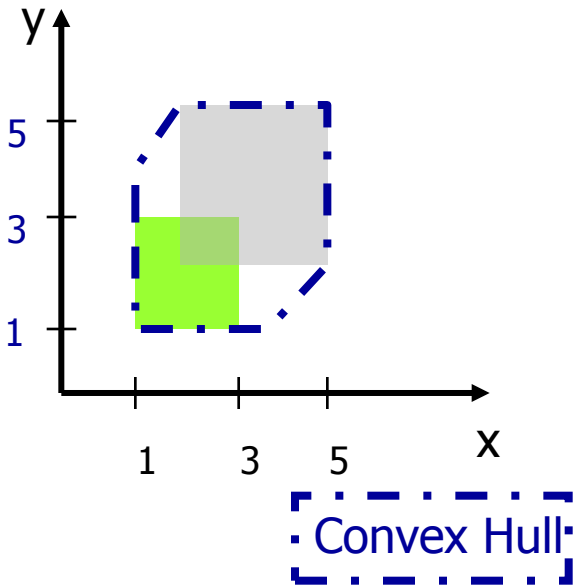
$G \in O$?

$G \in U$?

$$G \in U \Rightarrow G \in R$$
$$\neg(G \in O) \Rightarrow \neg(G \in R)$$

Over-approximation Convex Hull

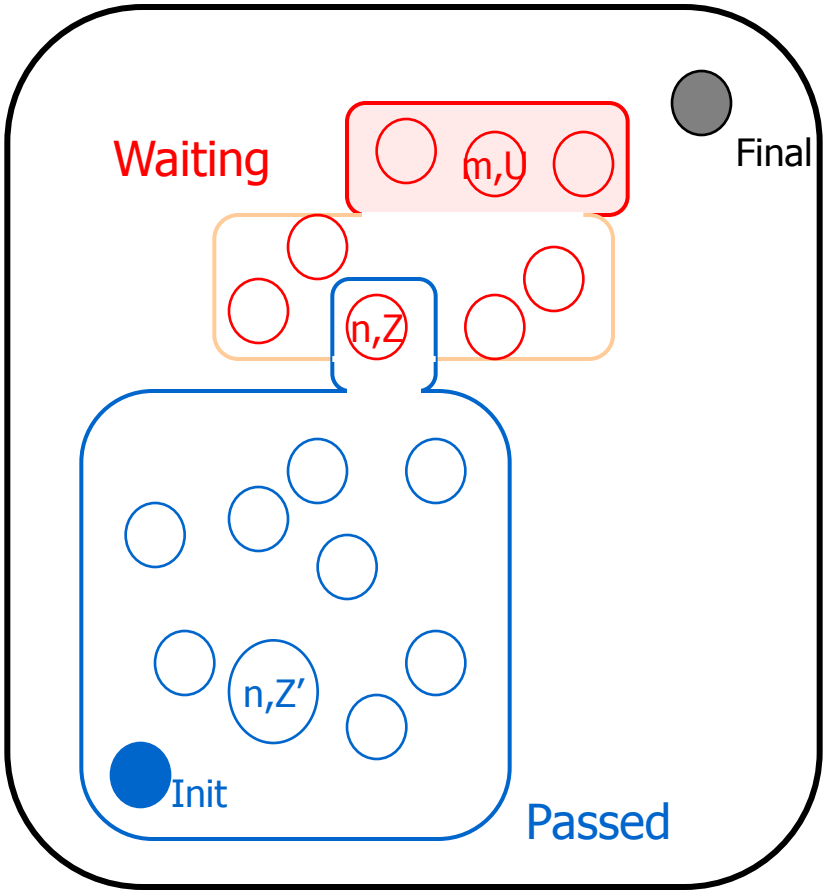




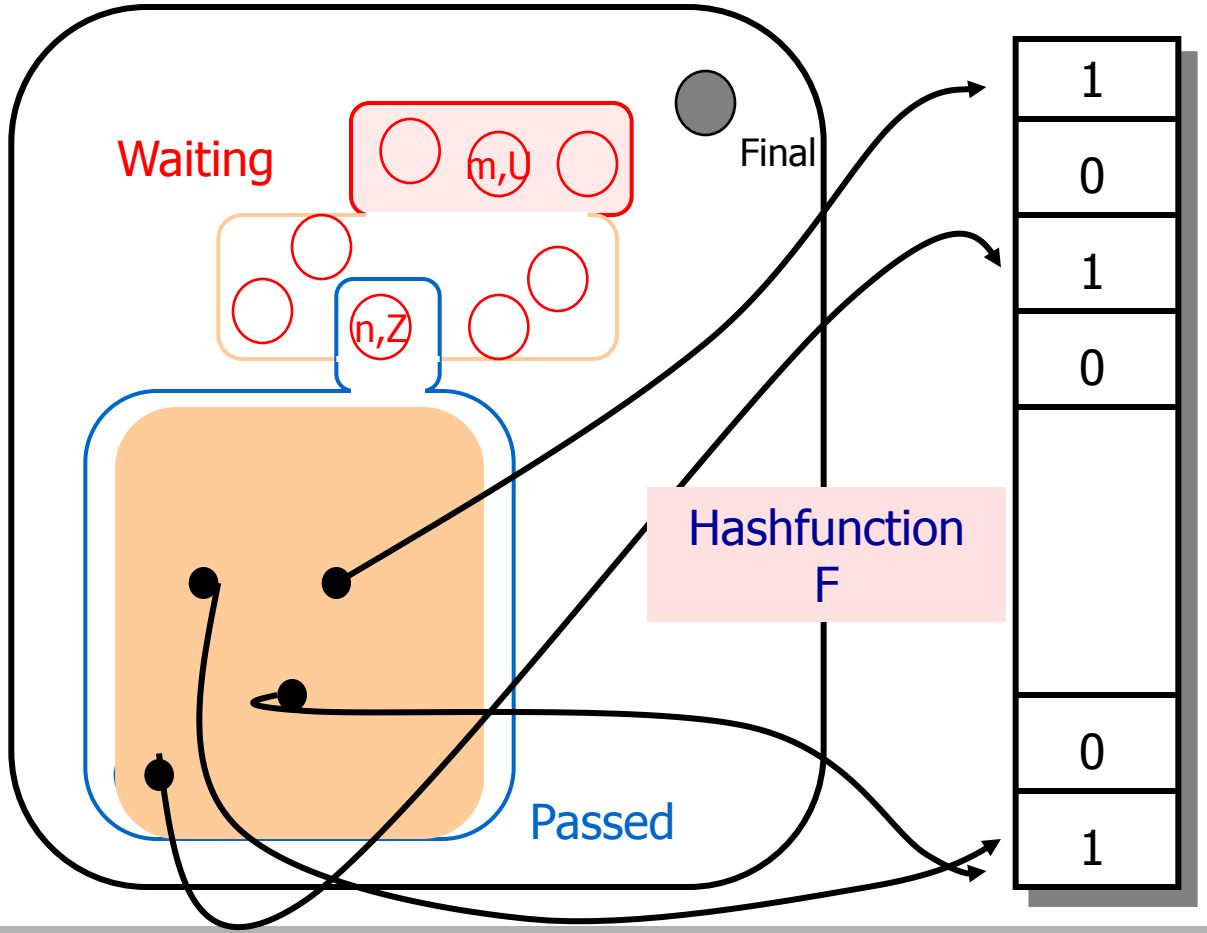
TACAS04: An EXACT method performing as well as Convex Hull has been developed based on abstractions taking max constants into account distinguishing between clocks, locations and $\leq$ & $\geq$

A. David [39]

Under-approximation Bitstate Hashing



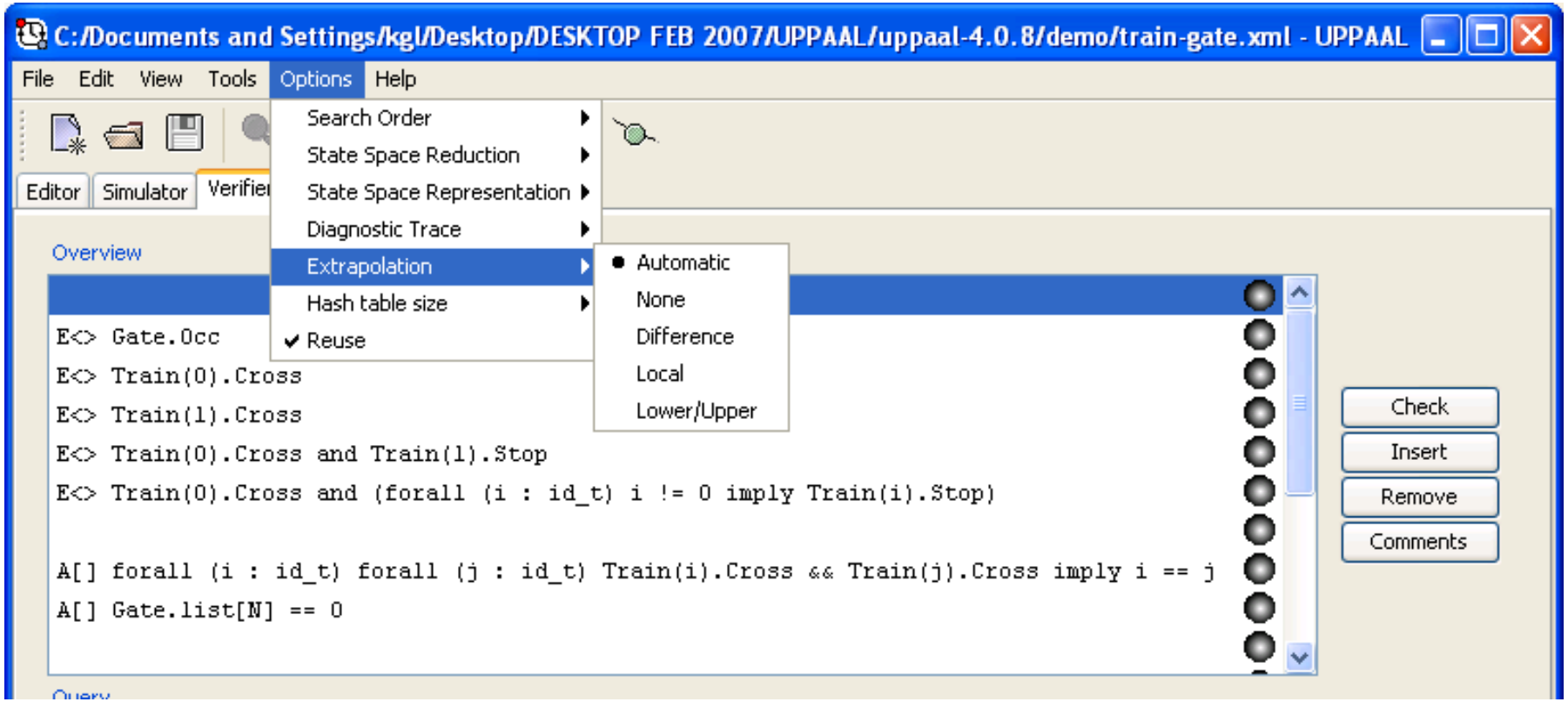
Under-approximation Bitstate Hashing



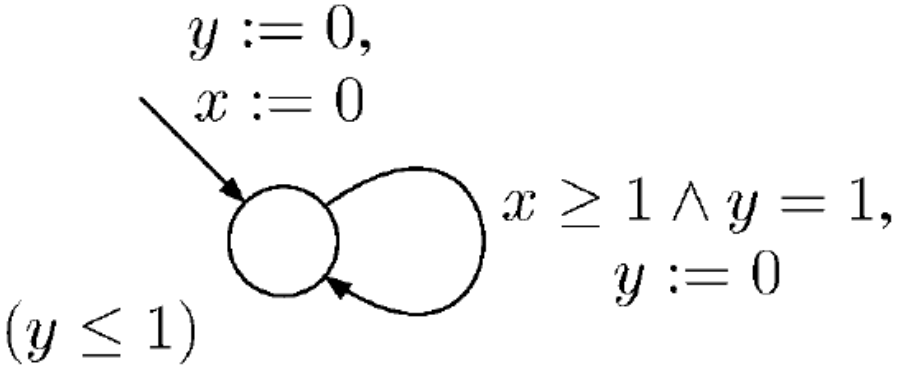
Passed=
Bitarray

UPPAAL
4 - 512 Mbits

Extrapolation

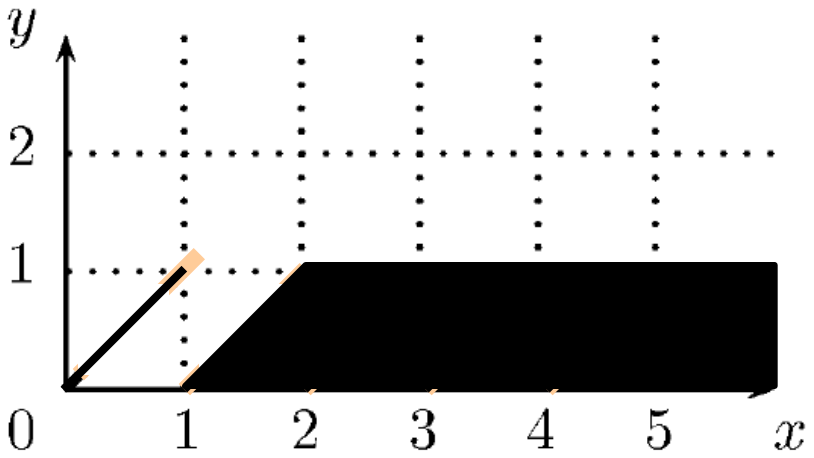


Forward Symbolic Exploration



TERMINATION not garanteed

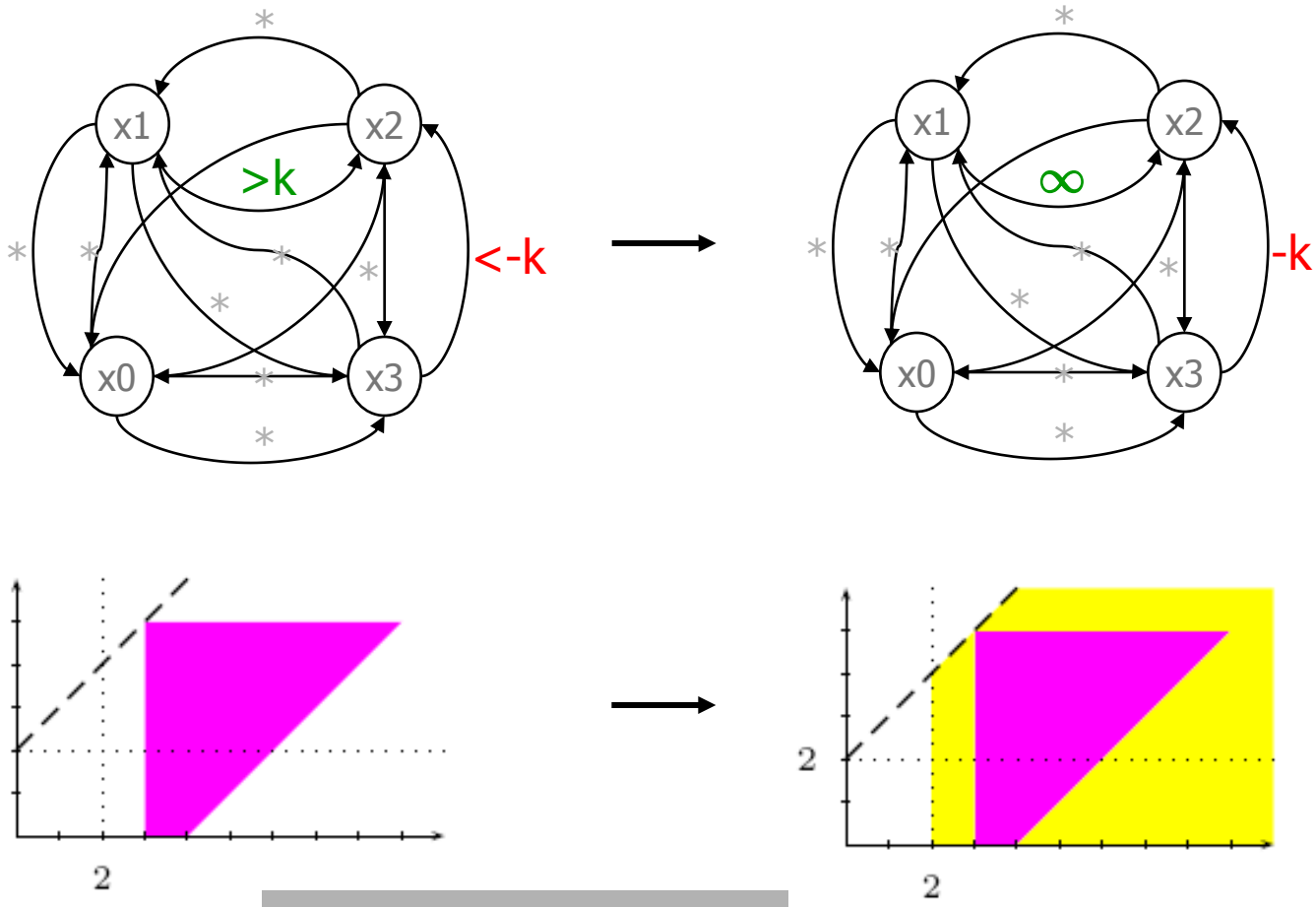
Need for Finite Abstractions



Abstraction by Extrapolation

[Daws,Tripakis 98]

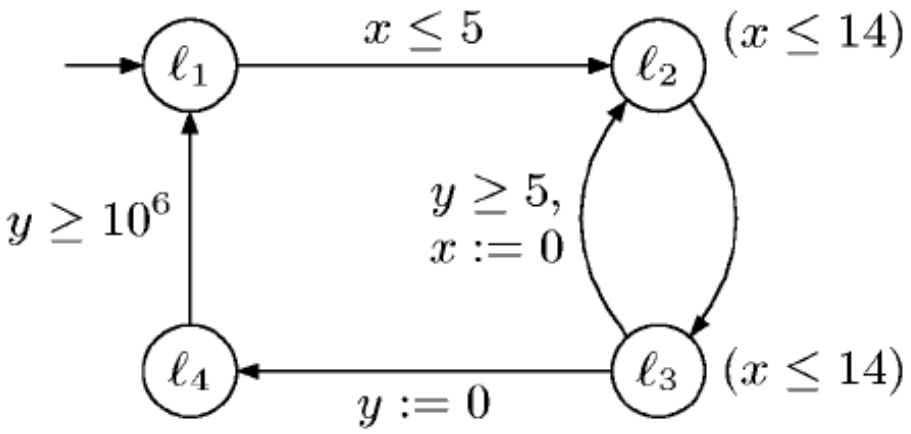
Let *k* be the largest constant appearing in the TA



Sound & Complete
Ensures Termination

Location Dependency

[Behrmann, Bouyer, Fleury, Larsen 03]



$k_x = 5$ $k_y = 10^6$

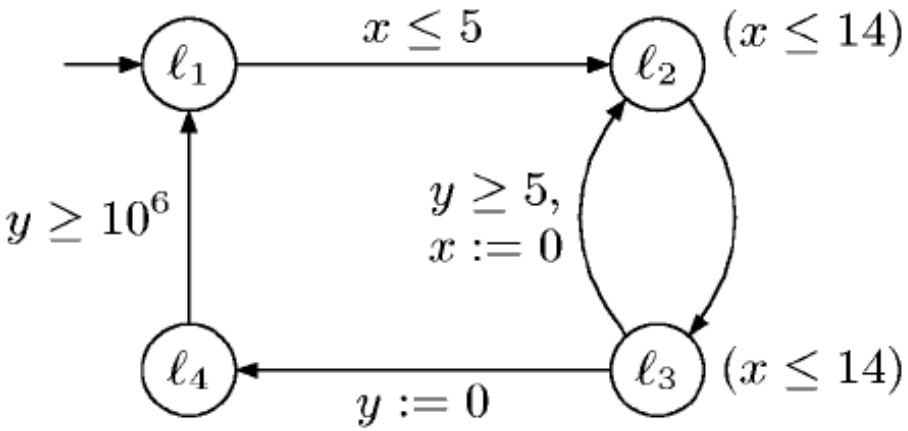
Will generate all symbolic states of the form

$(l_2, x \in [0,14], y \in [5,14n], y-x \in [5,14n-14])$

for $n \leq 10^6/14$!!

But $y \geq 10^6$ is not RELEVANT in l_2

Location Dependent Constants



$k_x = 5 \quad k_y = 10^6$

$k_x^i = 14$ for $i \in \{1,2,3,4\}$

$k_y^i = 5$ for $i \in \{1,2,3\}$

$k_y^4 = 10^6$

k_j^i may be found as solution to simple linear constraints!

Active Clock Reduction:

$k_j^i = -\infty$

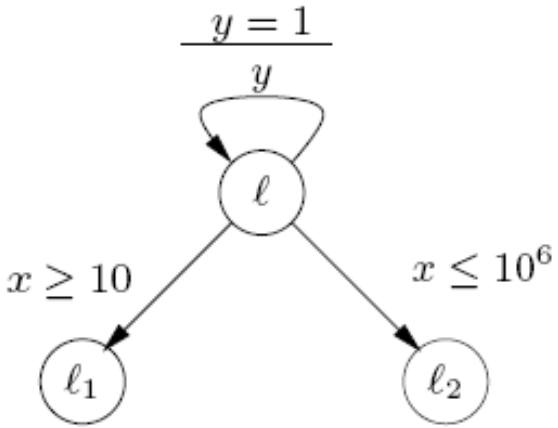
Experiments

Active by default

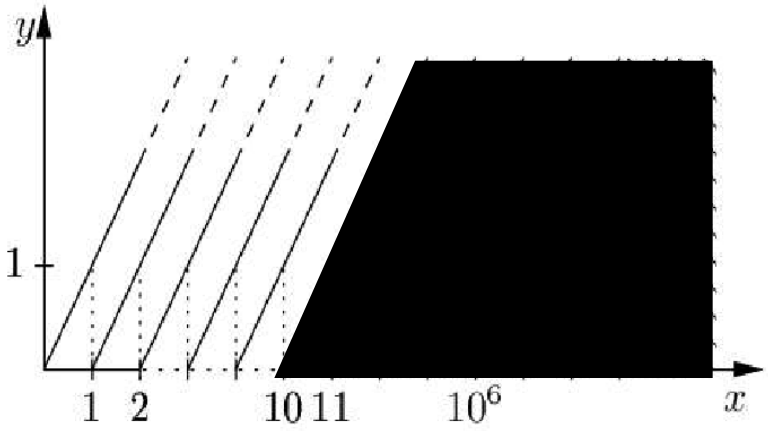
	Constant BIG	*Global Method*	*Active-clock Reduction*	*Local Constants*
Naive Example	10^3	0.05s/1MB	0.05s/1MB	0.00s/1MB
	10^4	4.78s/3MB	4.83s/3MB	0.00s/1MB
	10^5	484s/13MB	480s/13MB	0.00s/1MB
	10^6	stopped	stopped	0.00s/1MB
Two Processes	10^3	3.24s/3MB	3.26s/3MB	0.01s/1MB
	10^4	5981s/9MB	5978s/9MB	0.37s/2MB
	10^5	stopped	stopped	72s/5MB
Asymmetric Fischer	10^3	0.01s/1MB	0.01s/1MB	0.01s/1MB
	10^4	2.20s/3MB	2.20s/3MB	0.85s/2MB
	10^5	333s/19MB	333s/19MB	160s/13MB
	10^6	33307s/122MB	33238s/122MB	16330s/65MB
Bang & Olufsen	25000	stopped	159s/243MB	123s/204MB

Lower and Upper Bounds

[Behrmann, Bouyer, Larsen, Pelanek 04]



$k_x^l = 10^6$



Given that $x \leq 10^6$ is an *upper* bound implies that

(l, v_x, v_y) *simulates* (l, v'_x, v_y)

whenever $v'_x \geq v_x \geq 10$.

For reachability downward closure wrt simulation suffices!

Additional “secrets”

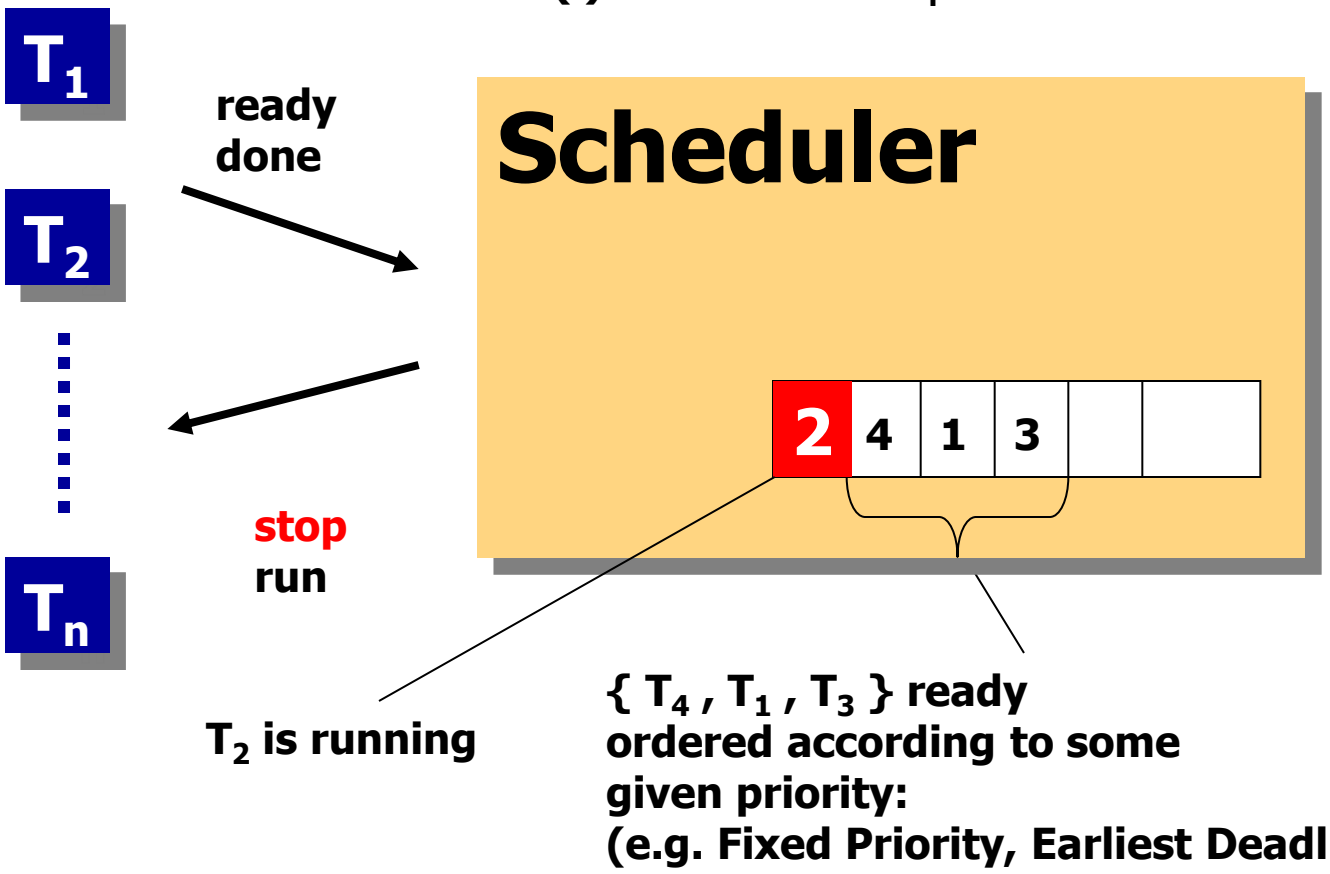
- Sharing among symbolic states
  - location vector / discrete values / zones
- Symmetry Reduction
- Sweep Line Method
- Guiding wrt Heuristic Value (CORA)
  - User-supplied / Auto-generated
- “Manual” tricks:
  - active variable reduction
  - Value passing using arrays of channels

Application: Schedulability Analysis

Task Scheduling

utilization of CPU

P(i), [E(i), L(i)], .. : period or
earliest/latest arrival or .. for T_i
C(i): execution time for T_i
D(i): deadline for T_i



Classical Scheduling Theory

Utilisation-Based Analysis

- A simple sufficient but not necessary schedulability test exists

$$U \equiv \sum_{i=1}^{N} \frac{C_i}{T_i} \leq N\,(2^{1/N} - 1)$$

$$U \leq 0.69 \quad \text{as} \quad N \to \infty$$

Where C is WCET and T is period

41

Response Time Equation

$$R_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

Where $hp(i)$ is the set of tasks with priority higher than task i

Solve by forming a recurrence relationship:

$$w_i^{n+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i^n}{T_j} \right\rceil C_j$$

The set of values $w_i^0, w_i^1, w_i^2, \ldots, w_i^n, \ldots$ is monotonically non decreasing
When $w_i^n = w_i^{n+1}$ the solution to the equation has been found, w_i^0 must not be greater that R_i (e.g. 0 or C_i)

42

Classical WCRT Analysis

Quasimodo

- "Classical" scheduling analysis technique
- *For all tasks i: WCRT$_i$ ≤ Deadline$_i$*

$$R_i = B_i + C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

Blocking times for priority inheritance protocol (BSW):

$$Blocking(i) = \sum_{r=1}^{R} usage(r,i)\,WCET_{CriticalSection}(r)$$

Blocking times for priority ceiling protocol (ASW):

$$Blocking(i) = \max_{r=1}^{R} usage(r,i)\,WCET_{CriticalSection}(r)$$

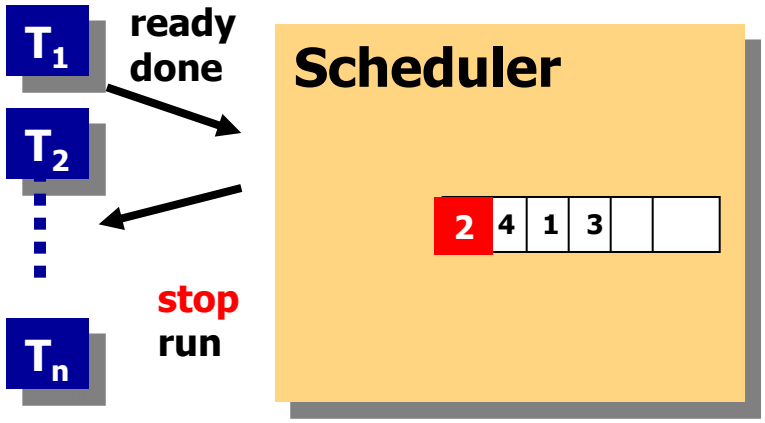
Quasimodo Workshop, Eindhoven, Nov 6, 2009 Page 21

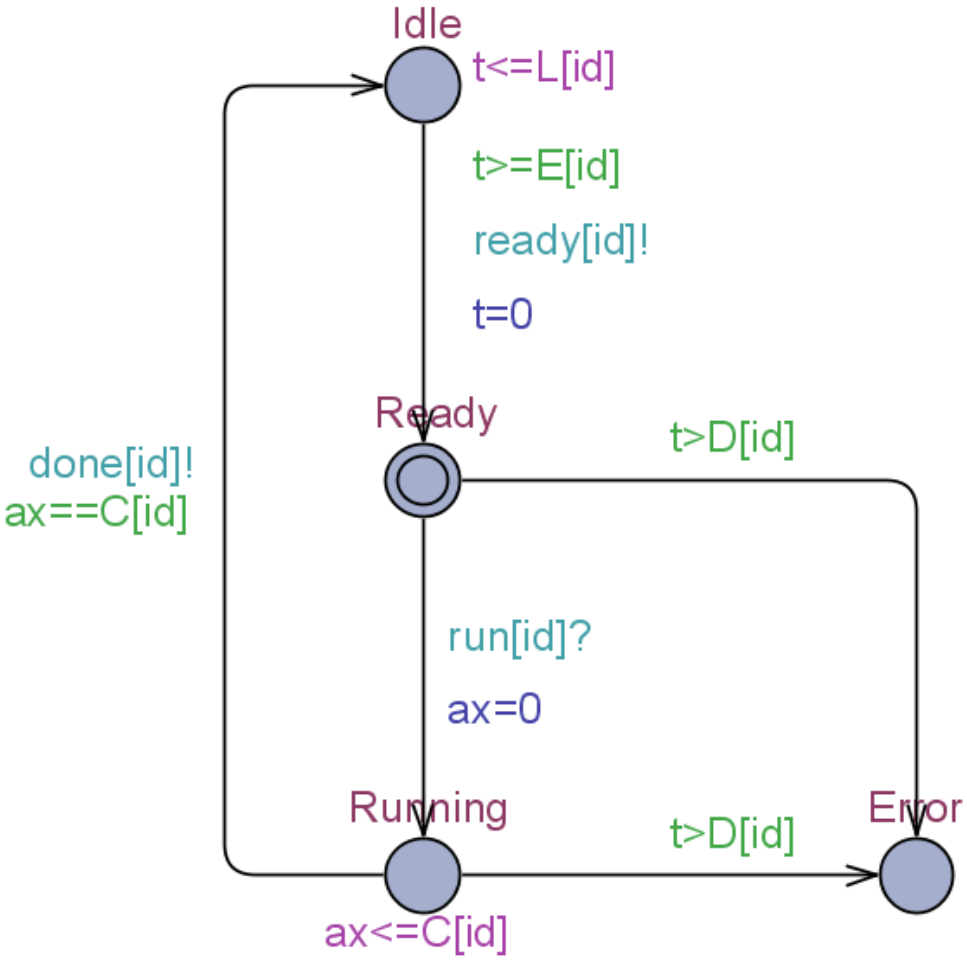
✓ Simple to perform

– Overly conservative
– Limited settings
– Single-processor

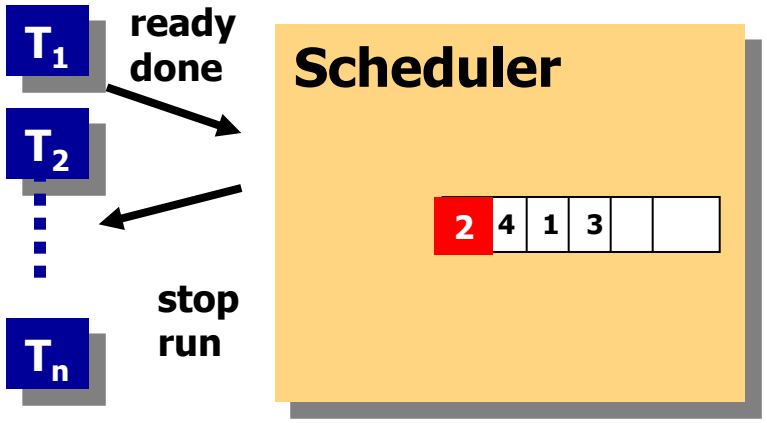
⇒ Do it in UPPAAL!

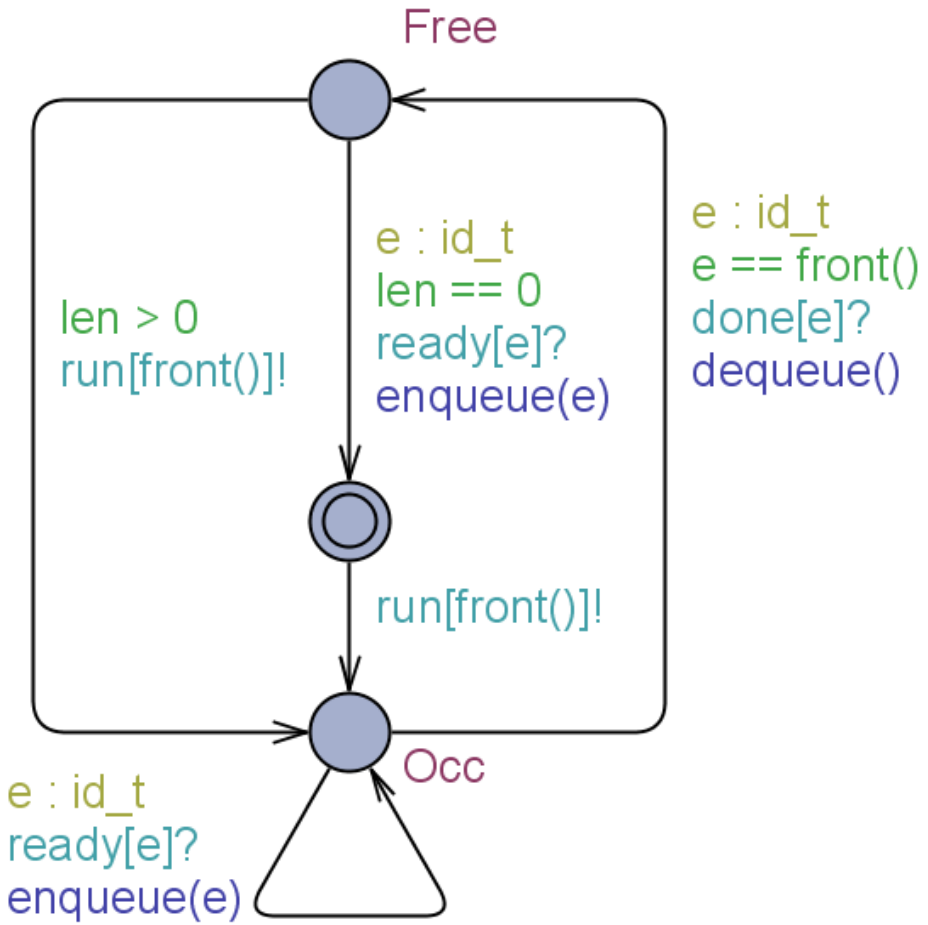
Modeling Task





Modeling Scheduler

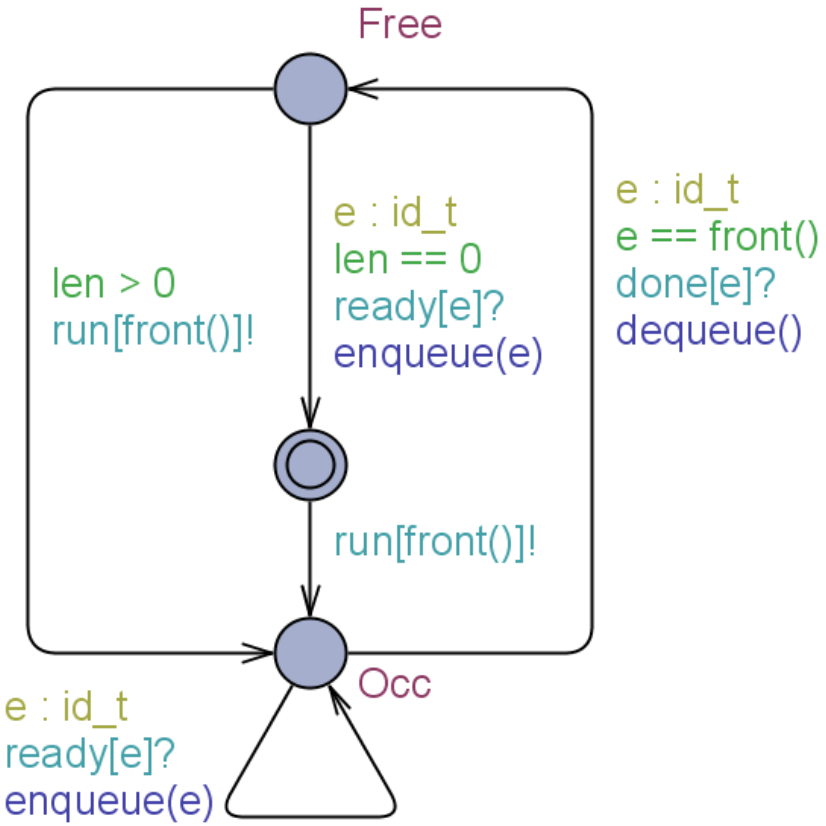




Implementation of enqueue/dequeue
⇒ scheduling policy

Modeling Queue

In UPPAAL 4.0
User Defined Function

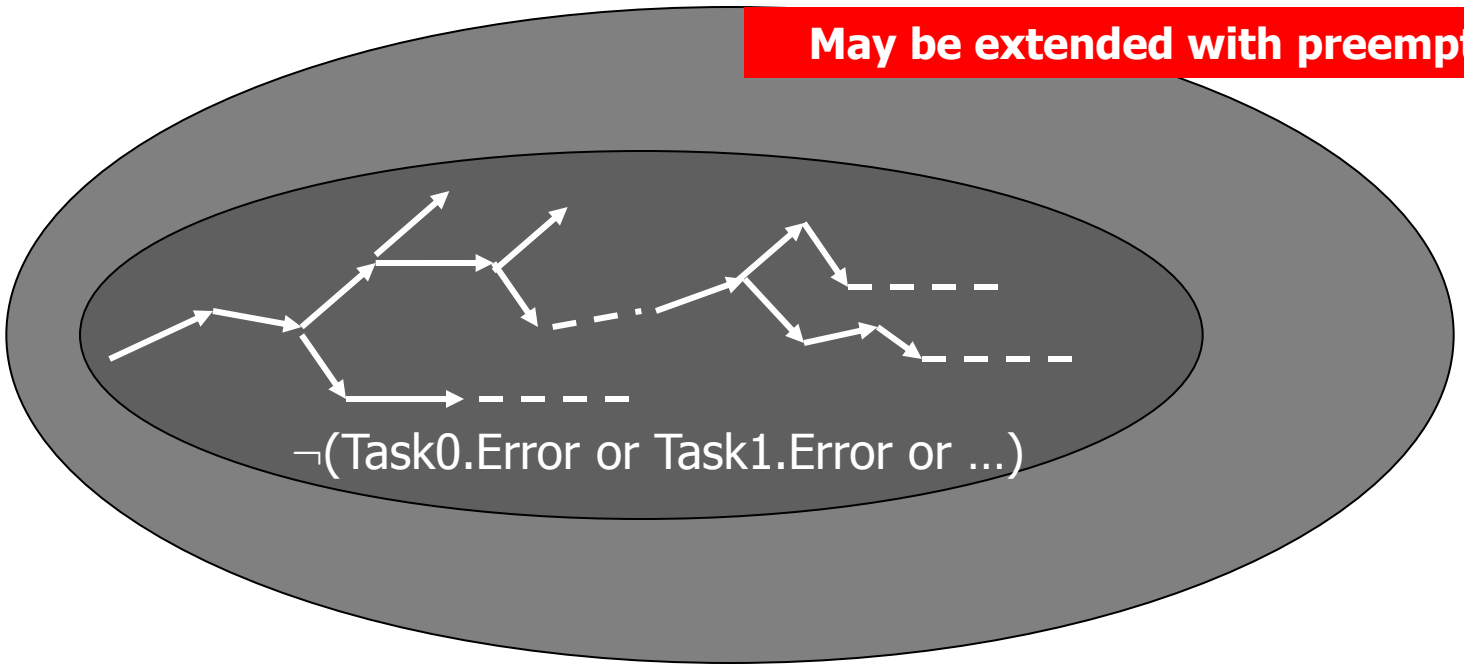



```
// Put an element at the end of the queue
void enqueue(id_t element)
{
int tmp=0;
list[len++] = element;
if (len>0)
{
        int i=len-1;
        while (i>1 && P[list[i]]>P[list[i-1]])
        {
                tmp = list[i-1];
                list[i-1] = list[i];
                list[i] = tmp;
                i--;
        }
}
}

// Remove the front element of the queue
void dequeue()
{
  ......
```

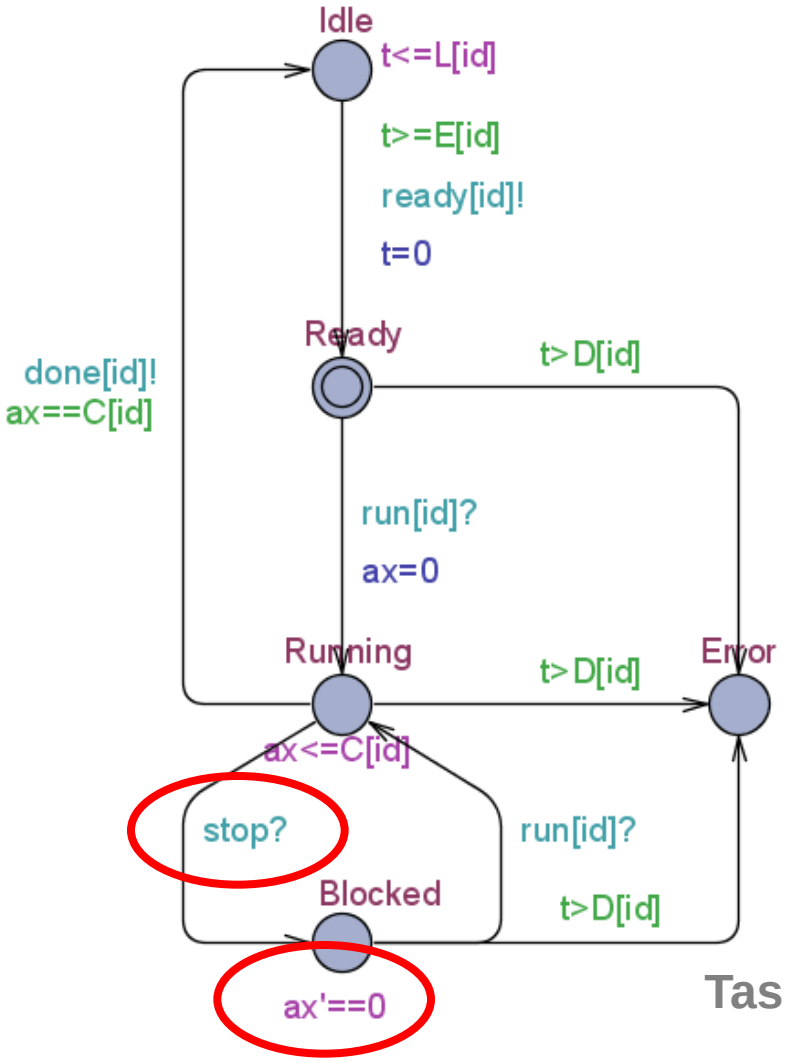
Sort by priority

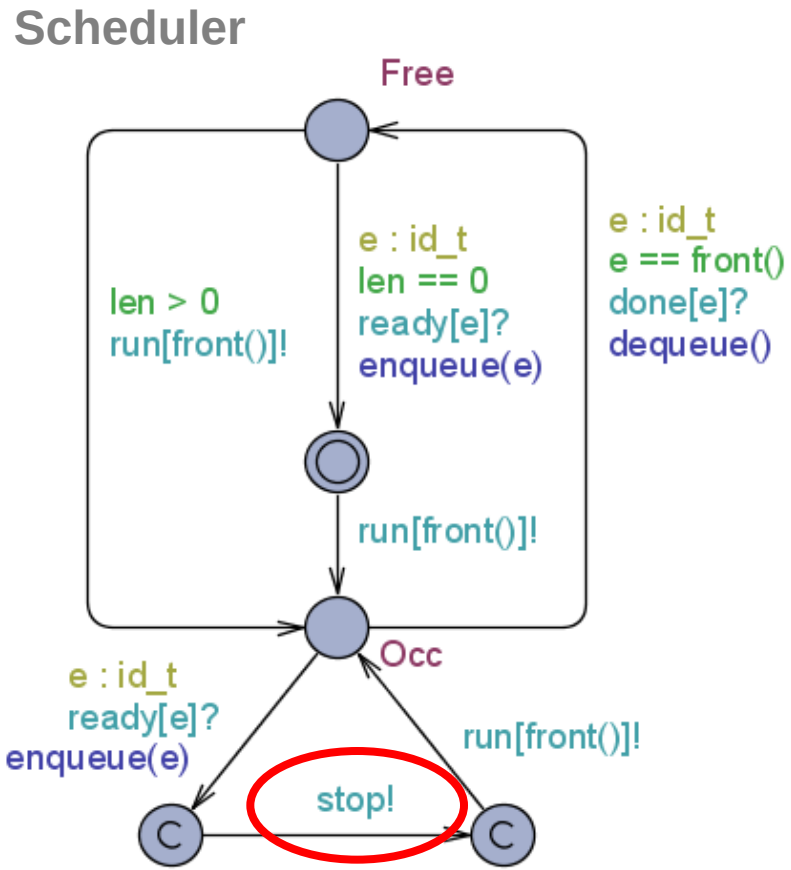
Schedulability = Safety Property



A□ ¬(Task0.Error or Task1.Error or …)

Preemption – Stopwatches!





Defeating undecidability ☺

Stop-Watches

- Make reachability undecidable.
- Over-approximation used in UPPAAL
  - $\Rightarrow$ Safe for positive schedulability results!

- What to do if you violate deadlines?
  - Try to validate the trace using other techniques, e.g., polyhedra.
  - Use SMC!