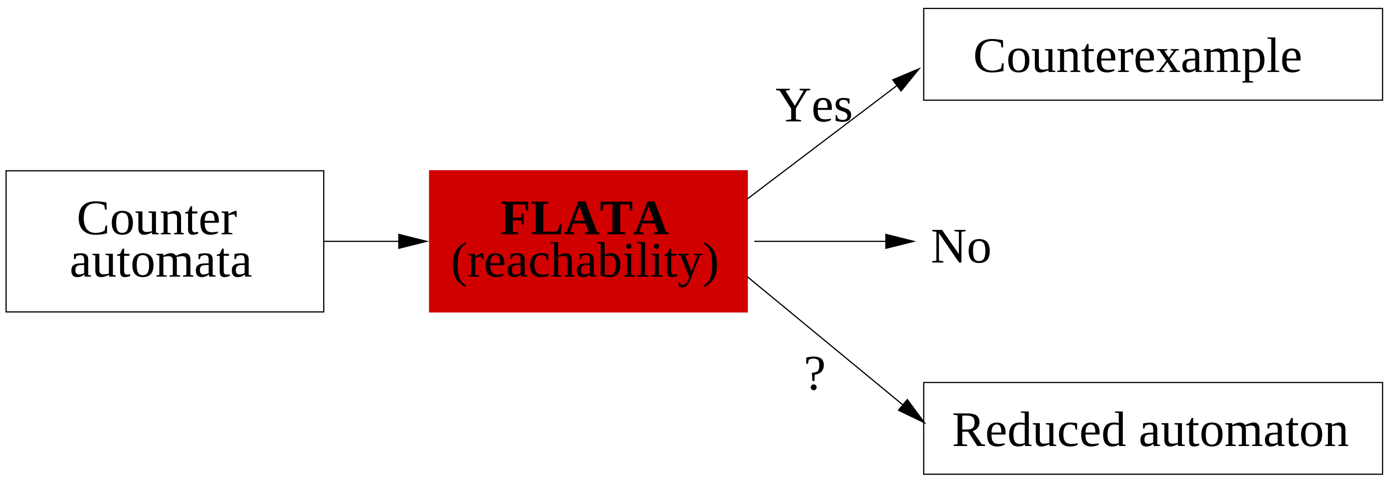
FLATA: A Tool for Manipulation and Analysis of Counter Automata

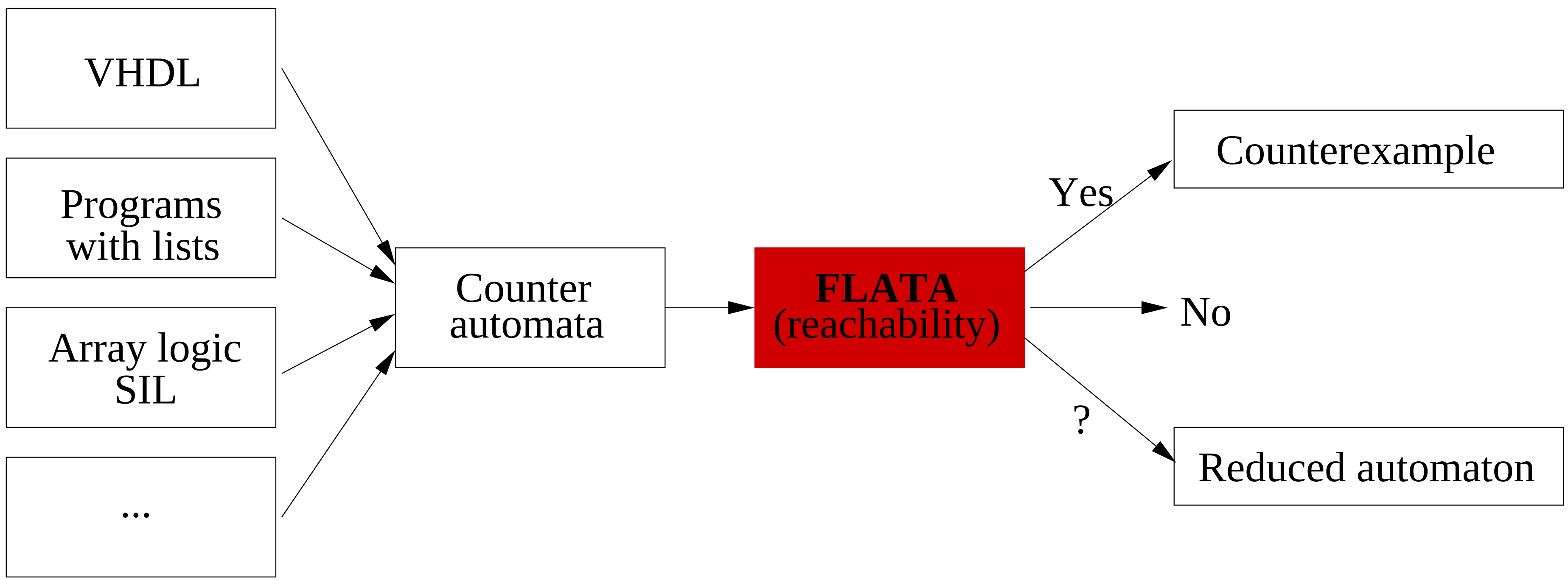
Marius Bozga[1], Radu Iosif[1], **Filip Konečný**[1,2], Tomáš Vojnar[2]

[1] VERIMAG / CNRS / University of Grenoble, France
[2] Brno University of Technology, Czech Republic

FLATA Overview



FLATA Overview



Talk Outline

1. **Counter Automata (CA)**
2. Classes of Transition Labels
3. Running Example
4. Reachability Analysis of CA
5. Simplification of Transitions
6. Future Work
7. Tool Demonstration

Counter Automata

- Sequential non-deterministic programs with unbounded integer variables.

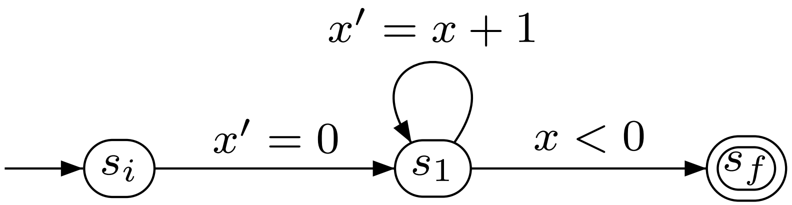
Example

begin
    $x := 0;$
    while $(*)$ **do**
        $x := x + 1;$
    if $(x < 0)$ **then**
        $error();$
end

- Is the error state reachable?
- Reachability of a control state is undecidable [Minsky'67]

Counter Automata

- Finite automata with integer counters $\mathbf{x} = \{x, y, z, \dots\}$



- Transitions labeled with arithmetic relations – formulae over $\mathbf{x}$ and $\mathbf{x}'$
  - $\mathbf{x}$ – current step
  - $\mathbf{x}' = \{x' \mid x \in \mathbf{x}\}$ – next step

- Non-deterministic assignments ($x' \geq 0$)
  - essential for building abstractions

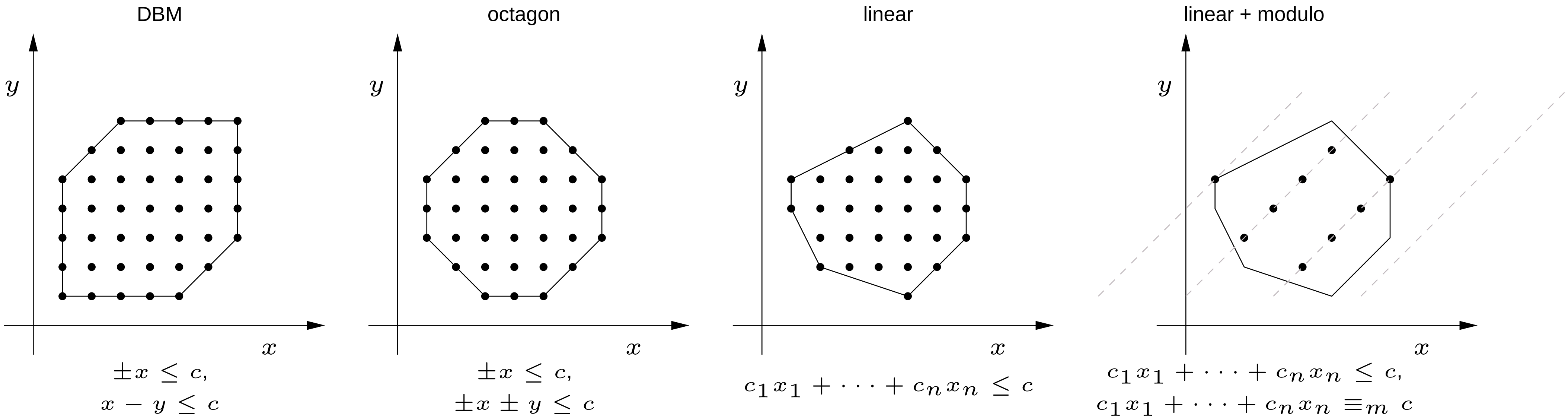
Talk Outline

1. Counter Automata (CA)
2. **Classes of Transition Labels**
3. Running Example
4. Reachability Analysis of CA
5. Simplification of Transitions
6. Future Work
7. Tool Demonstration

Transition Labels

❖ We consider the following classes of relations:

- DBM
- octagons
- linear relations
- linear relations + modulo constraints



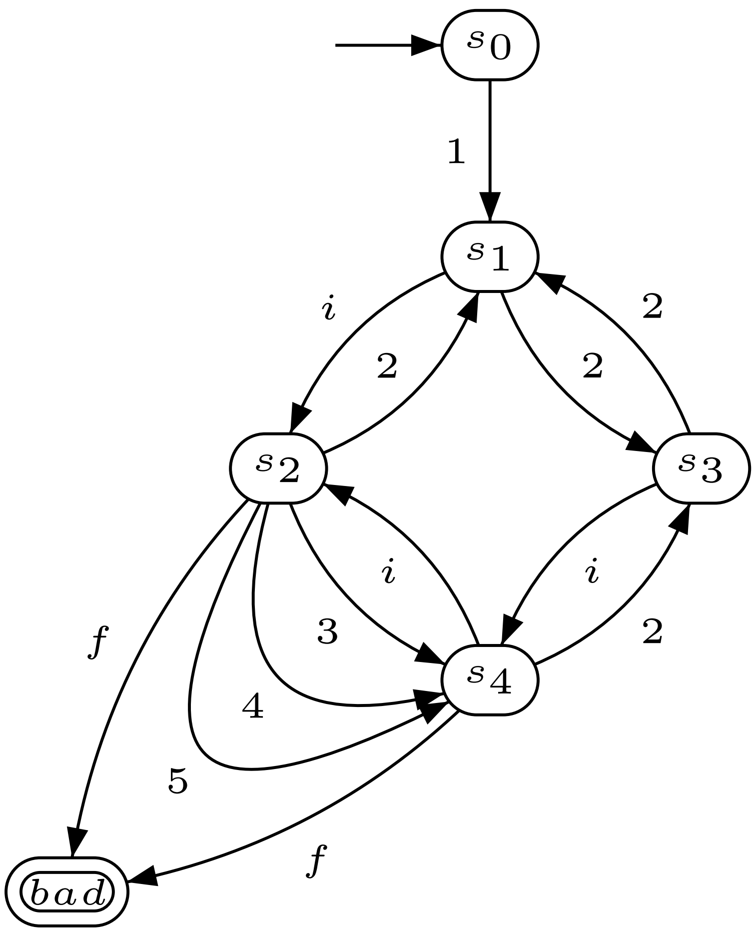
❖ Disjunctions of relations $\equiv$ Presburger arithmetic.

Talk Outline

1. Counter Automata (CA)
2. Classes of Transition Labels
3. **Running Example**
4. Reachability Analysis of CA
5. Simplification of Transitions
6. Future Work
7. Tool Demonstration

Example

❖ CA of a hardware component and a property checking if a exceedes a parametric bound m.



$$
\begin{array}{llll}
i: & & \{a' = a, & m' = m\} \\
1: & & \{a' = 0, & m' \geq 2\} \\
2: & & \{a' = 0, & m' = m\} \\
3: & [m \geq a+2] & \{a' = a+1, & m' = m\} \\
4: & [m \leq a] & \{a' = a+1, & m' = m\} \\
5: & [m = a+1] & \{a' = 0, & m' = m\} \\
f: & [a = m] & \{a' = a, & m' = m\}
\end{array}
$$

Talk Outline

1. Counter Automata (CA)
2. Classes of Transition Labels
3. Running Example
4. **Reachability Analysis of CA**
5. Simplification of Transitions
6. Future Work
7. Tool Demonstration

Reachability Problem

❖ Given a set of initial and final control states, is there an execution from some initial to some final state?

❖ Reduce the given CA to one with less control locations and equivalent reachability problem.

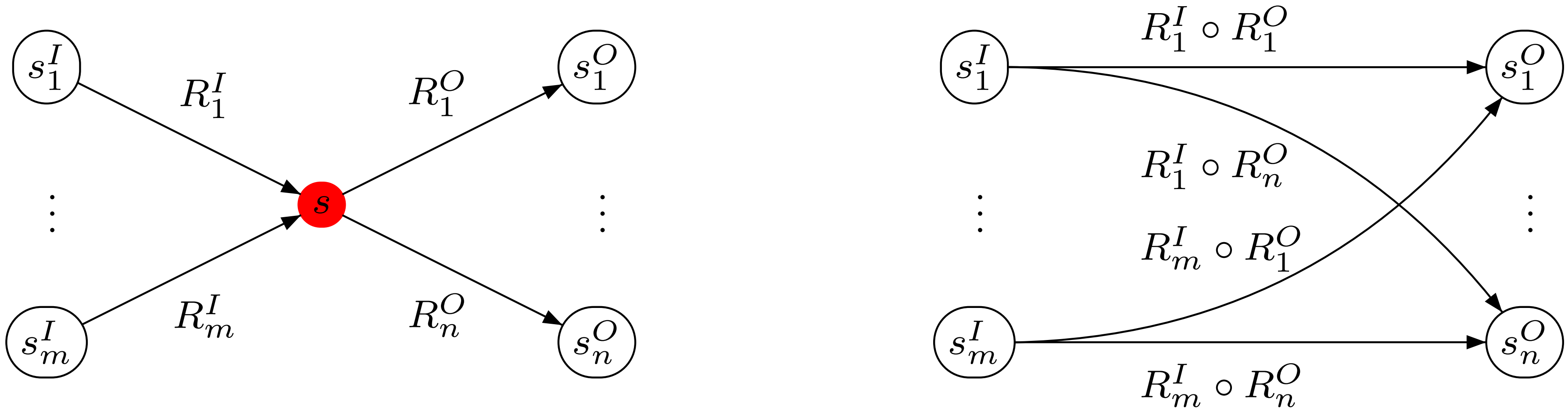
❖ Reduction techniques based on:

- composition of transitions
- acceleration

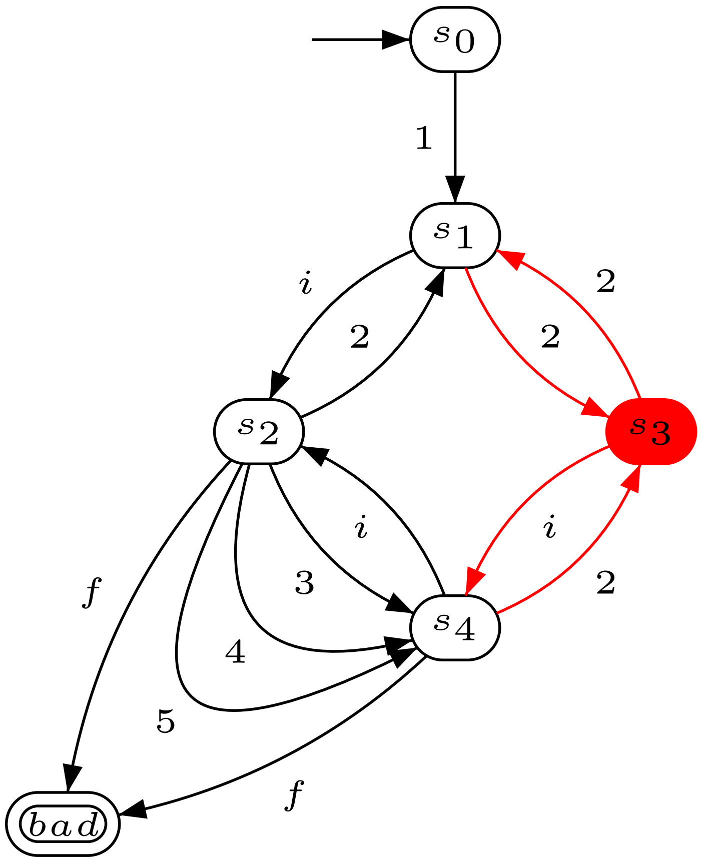
States without Loops

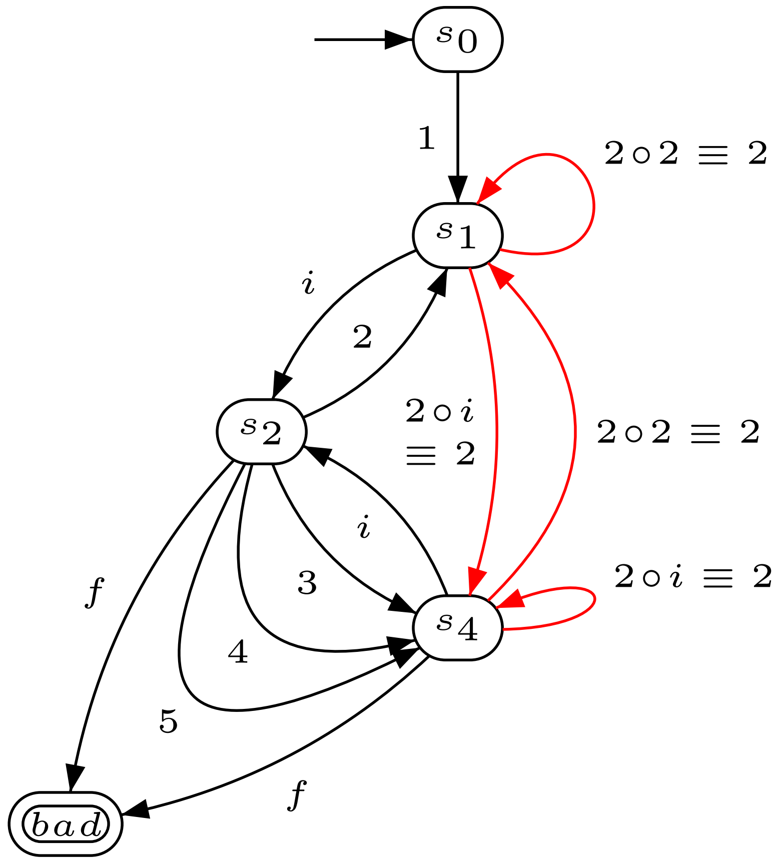
- Reduction
  1. consider path segments of the form $s_i \xrightarrow{R^I} s \xrightarrow{R^O} s_o$
  2. collapse each of them to a single transition $s_i \xrightarrow{R^I \circ R^O} s_o$

     (operation $\circ$ is relational composition)



States without Loops – Example

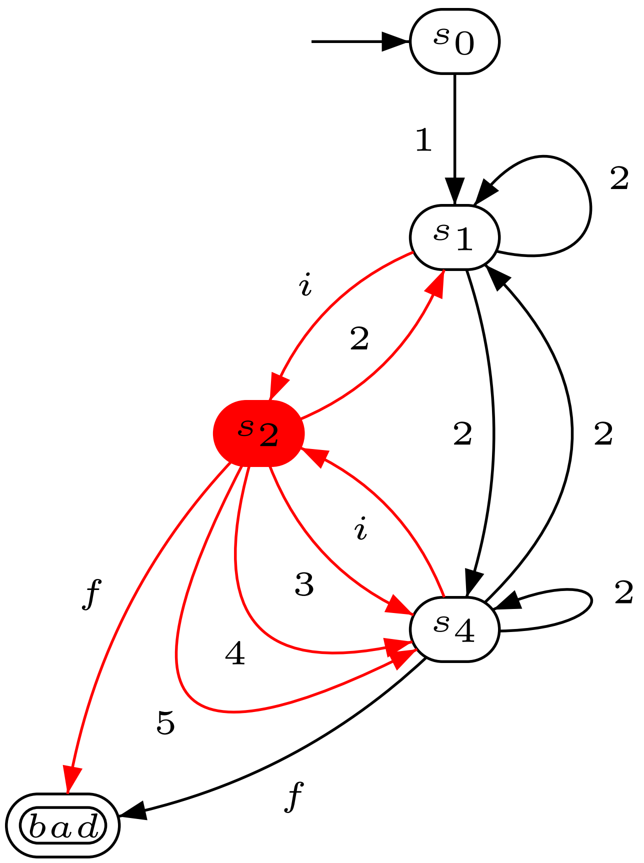


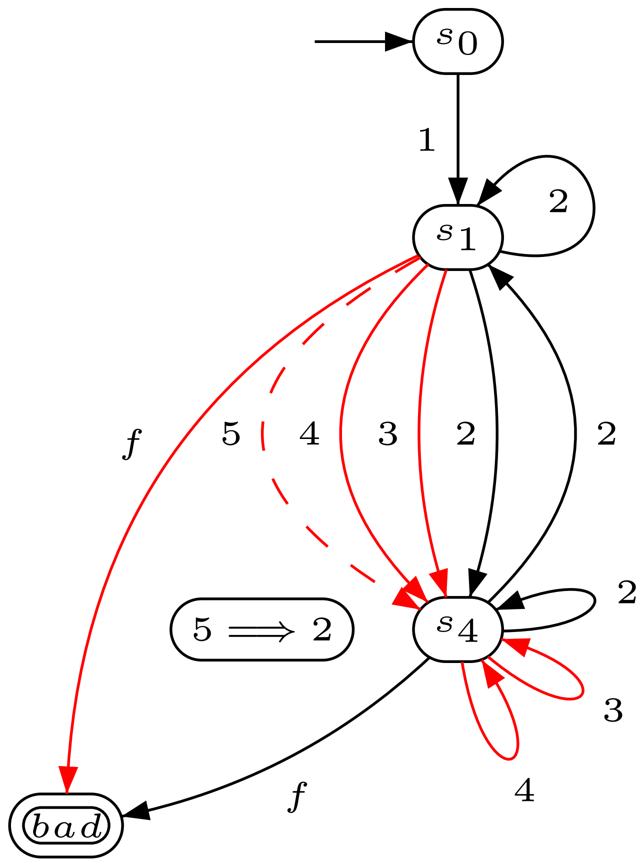


$$
\begin{array}{lll}
i: & \{a' = a, & m' = m\} \\
1: & \{a' = 0, & m' \geq 2\} \\
2: & \{a' = 0, & m' = m\}
\end{array}
$$

$$
\begin{array}{llll}
3: & [m \geq a + 2] & \{a' = a + 1, & m' = m\} \\
4: & [m \leq a] & \{a' = a + 1, & m' = m\} \\
5: & [m = a + 1] & \{a' = 0, & m' = m\} \\
f: & [a = m] & \{a' = a, & m' = m\}
\end{array}
$$

States without Loops – Example





$$\begin{array}{lll} i: & \{a' = a, & m' = m\} \\ 1: & \{a' = 0, & m' \geq 2\} \\ 2: & \{a' = 0, & m' = m\} \end{array}$$

$$\begin{array}{llll} 3: & [m \geq a+2] & \{a' = a+1, & m' = m\} \\ 4: & [m \leq a] & \{a' = a+1, & m' = m\} \\ 5: & [m = a+1] & \{a' = 0, & m' = m\} \\ f: & [a = m] & \{a' = a, & m' = m\} \end{array}$$

Transitive Closure

- Need to compute effects of loops

- Transitive closure of $R(\mathbf{x}, \mathbf{x}')$
  - $R^+ \equiv \bigvee_{i=1}^{\infty} R^i$, where $R^{i+1} \equiv R^i \circ R$
  - $R^* \equiv \mathcal{I} \vee R^+$, where $\mathcal{I} \equiv \bigwedge_{x \in \mathbf{x}} x' = x$

- Example
  - $(x' = x + 1)^+ \equiv (x' = x + 1) \vee (x' = x + 2) \vee \ldots$

    $(x' = x + 1)^+ \equiv (x' \geq x + 1)$
  - $(x' = x + 1)^* \equiv (x' = x) \vee (x' = x + 1) \vee (x' = x + 2) \vee \ldots$

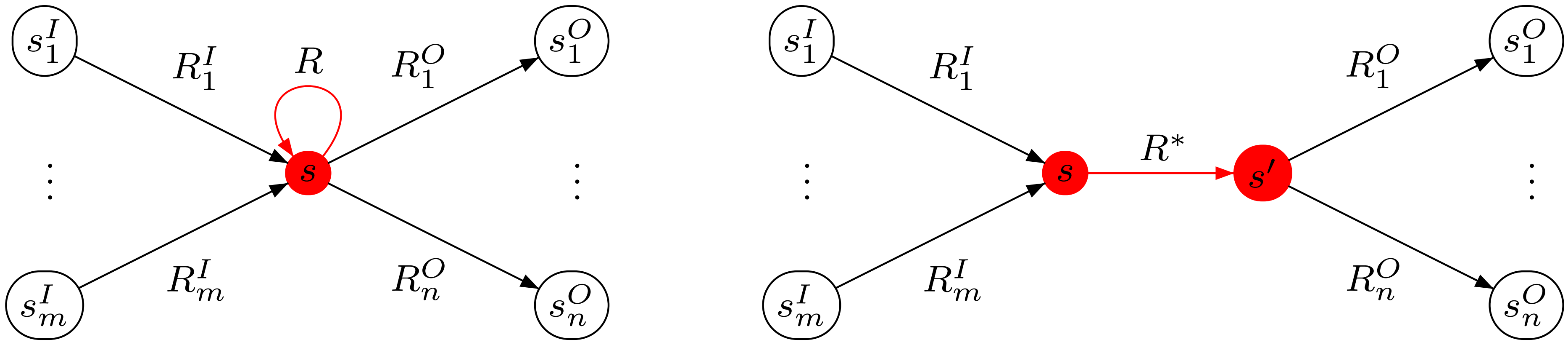
    $(x' = x + 1)^* \equiv (x' \geq x)$

- Bozga, Iosif, Konečný. *Fast Acceleration of Ultimately Periodic Relations.* CAV'10

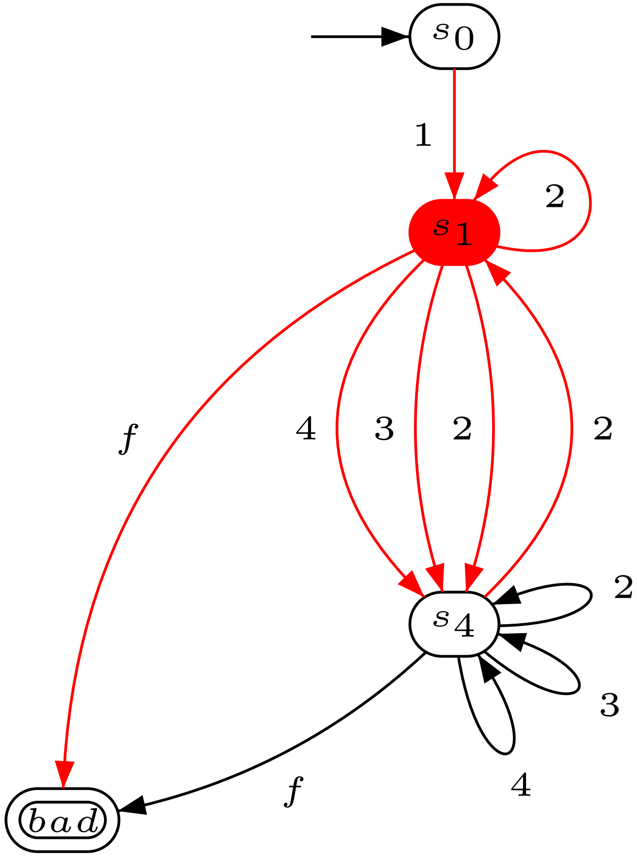
Acceleration of One Self-loop

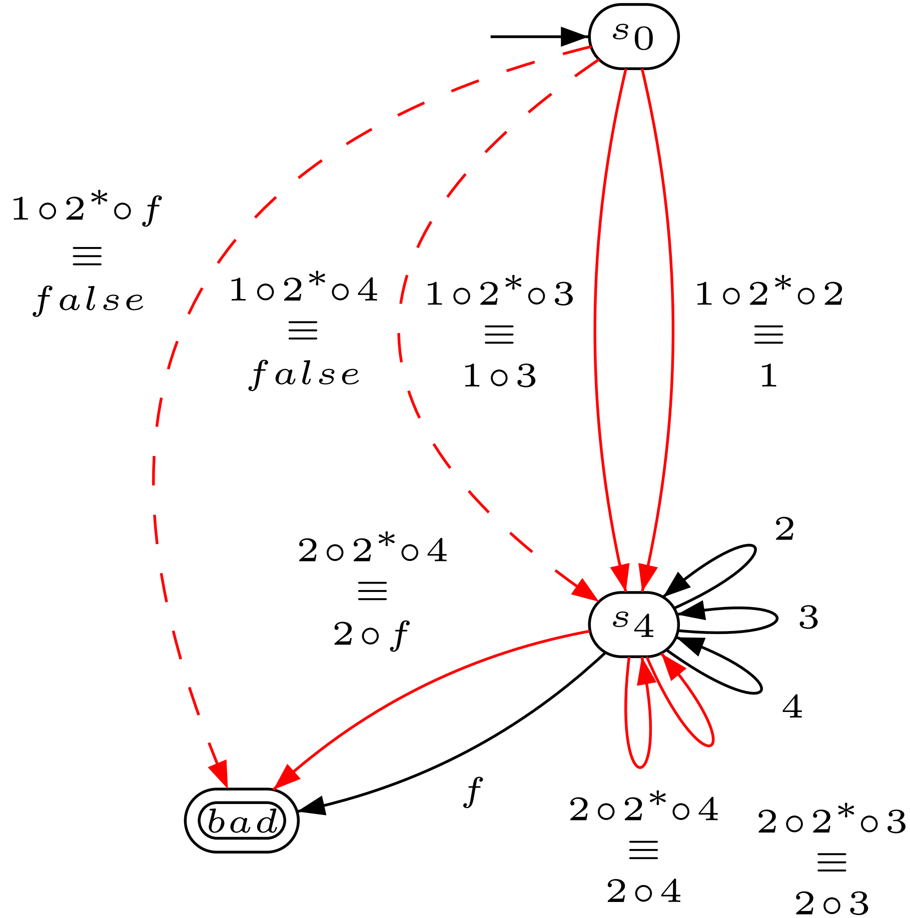
Reduction

1. accelerate the loop (compute R^*) $\quad s \xrightarrow{R} s$
2. replace the loop with a meta-transition $\quad s \xrightarrow{R^*} s'$



Acceleration of One Self-loop – Example

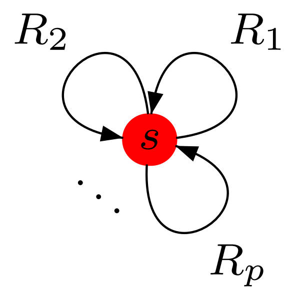




$i:$	$\{a' = a, \quad m' = m\}$
$1:$	$\{a' = 0, \quad m' \geq 2\}$
$2:$	$\{a' = 0, \quad m' = m\}$

$3:$	$[m \geq a+2]$	$\{a' = a+1,$	$m' = m\}$
$4:$	$[m \leq a]$	$\{a' = a+1,$	$m' = m\}$
$5:$	$[m = a+1]$	$\{a' = 0,$	$m' = m\}$
$f:$	$[a = m]$	$\{a' = a,$	$m' = m\}$

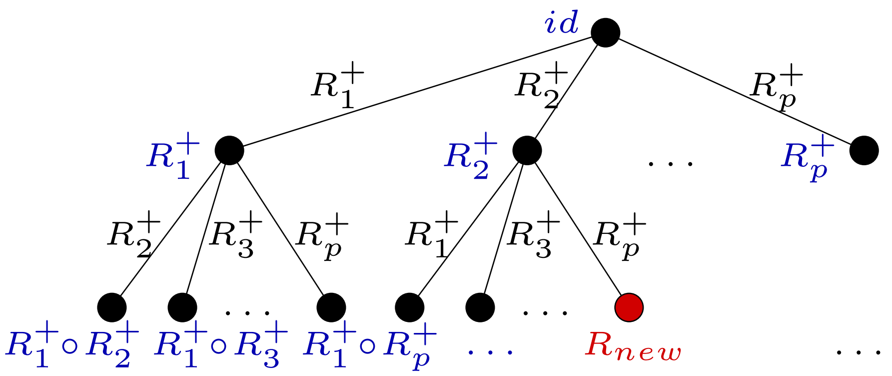
Control states with ≥ 2 loops



❖ State with p loops. Compute $(R_1 \vee R_2 \vee \cdots \vee R_p)^*$

❖ Semi-algorithm

- build a labeled tree in BFS manner
- edge labels $L = \{R_1^+, R_2^+, \ldots, R_p^+\}$
- for $l \in L$, define $succ(l) = \{l' \in L \mid l' \not\equiv l \;\wedge\; l \circ l' \not\equiv false\}$



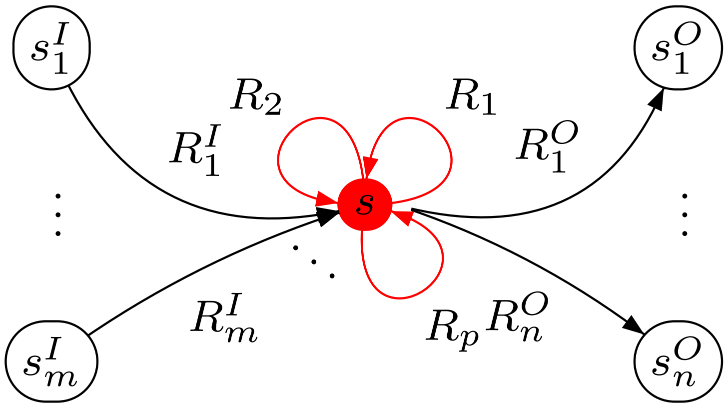
Cl – set of node labels

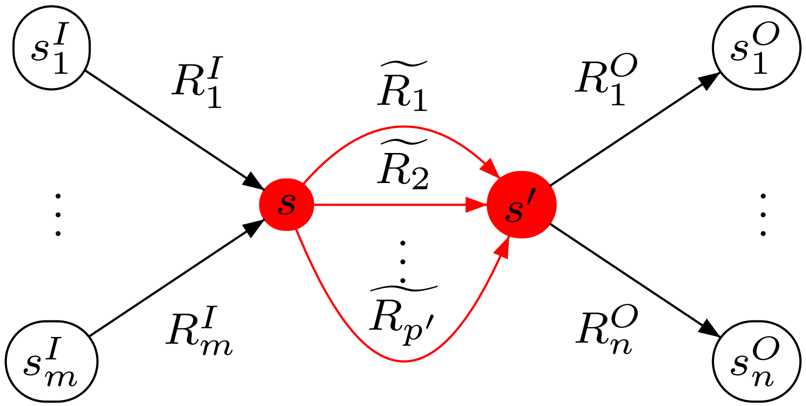
$R_{new} \Longrightarrow R$ for some $R \in Cl$

- continue until working list of nodes is empty
- or until $depth = MAX$

Control states with ≥ 2 loops

If semi-algorithm succeeded, let $Cl = \{\widetilde{R_1}, \widetilde{R_2}, \ldots, \widetilde{R_{p'}}\}$. Use meta-transitions in Cl to reduce CA.





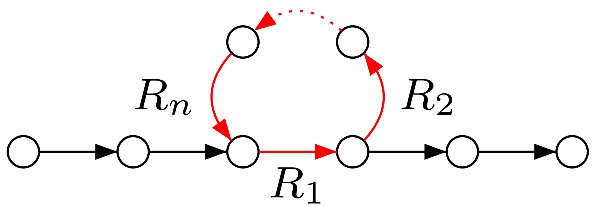
Flat Counter Automata

- Flat CA – automata restricted in terms of:
  1. control structure – no nested cycles
  2. labeling of transitions inside cycles – DBM, Octagon

Flat Counter Automata

- Flat CA – automata restricted in terms of:
  1. control structure – no nested cycles
  2. labeling of transitions inside cycles – DBM, Octagon

- Reachability decidable for flat CA

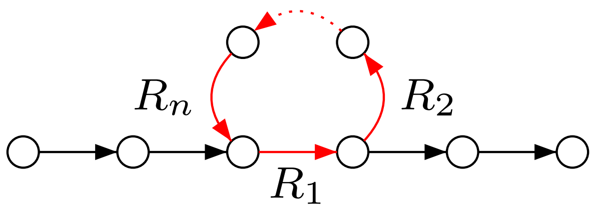


- $R_1, \ldots, R_n$ are DBM and octagons

Flat Counter Automata

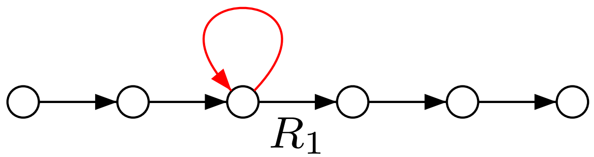
- Flat CA – automata restricted in terms of:
  1. control structure – no nested cycles
  2. labeling of transitions inside cycles – DBM, Octagon

- Reachability decidable for flat CA



- $R_1, \ldots, R_n$ are DBM and octagons

$R_1 \circ \ldots \circ R_n \equiv R$

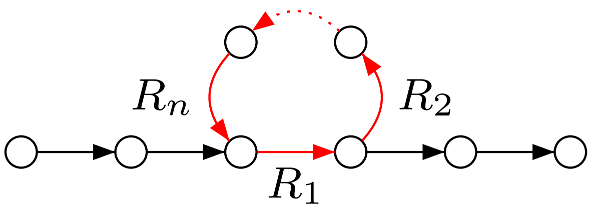


- R is DBM or octagon

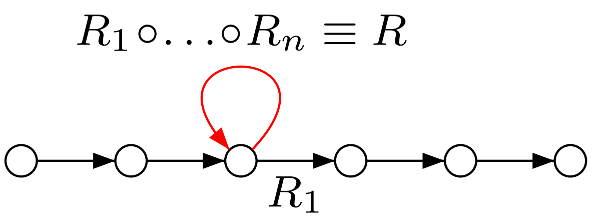
Flat Counter Automata

- Flat CA – automata restricted in terms of:
  1. control structure – no nested cycles
  2. labeling of transitions inside cycles – DBM, Octagon

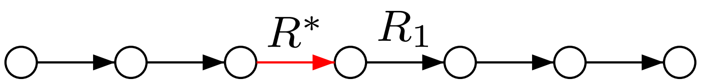
- Reachability decidable for flat CA



- $R_1, \ldots, R_n$ are DBM and octagons



- R is DBM or octagon



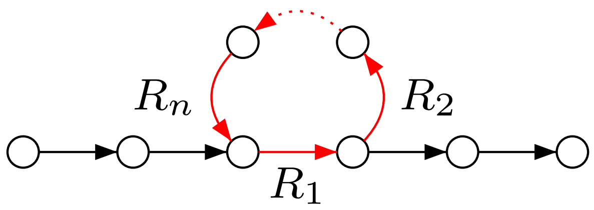
- R^* is Presburger definable

Flat Counter Automata

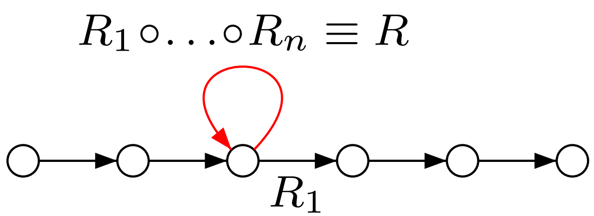
❖ Flat CA – automata restricted in terms of:

1. control structure – no nested cycles
2. labeling of transitions inside cycles – DBM, Octagon

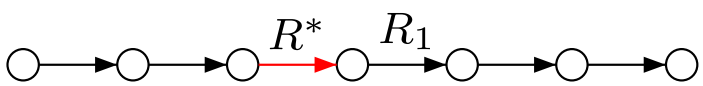
❖ Reachability decidable for flat CA



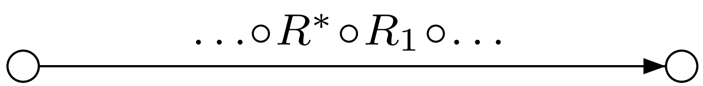
- $R_1, \ldots, R_n$ are DBM and octagons



- R is DBM or octagon



- R^* is Presburger definable

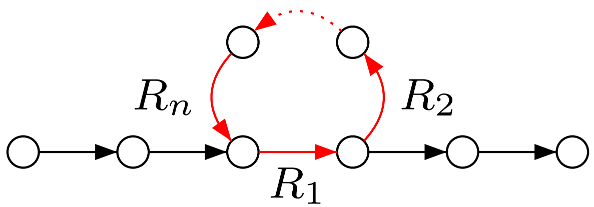


- Presburger formula

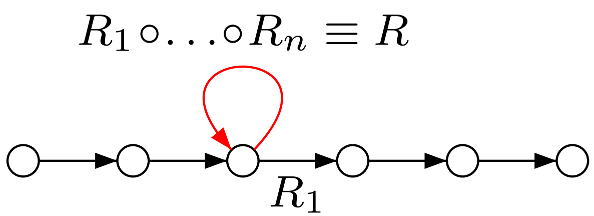
Flat Counter Automata

- Flat CA – automata restricted in terms of:
  1. control structure – no nested cycles
  2. labeling of transitions inside cycles – DBM, Octagon

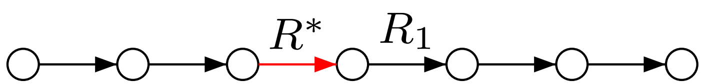
- Reachability decidable for flat CA



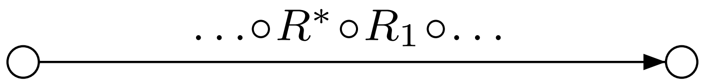
- $R_1, \ldots, R_n$ are DBM and octagons



- R is DBM or octagon



- R^* is Presburger definable



- Presburger formula

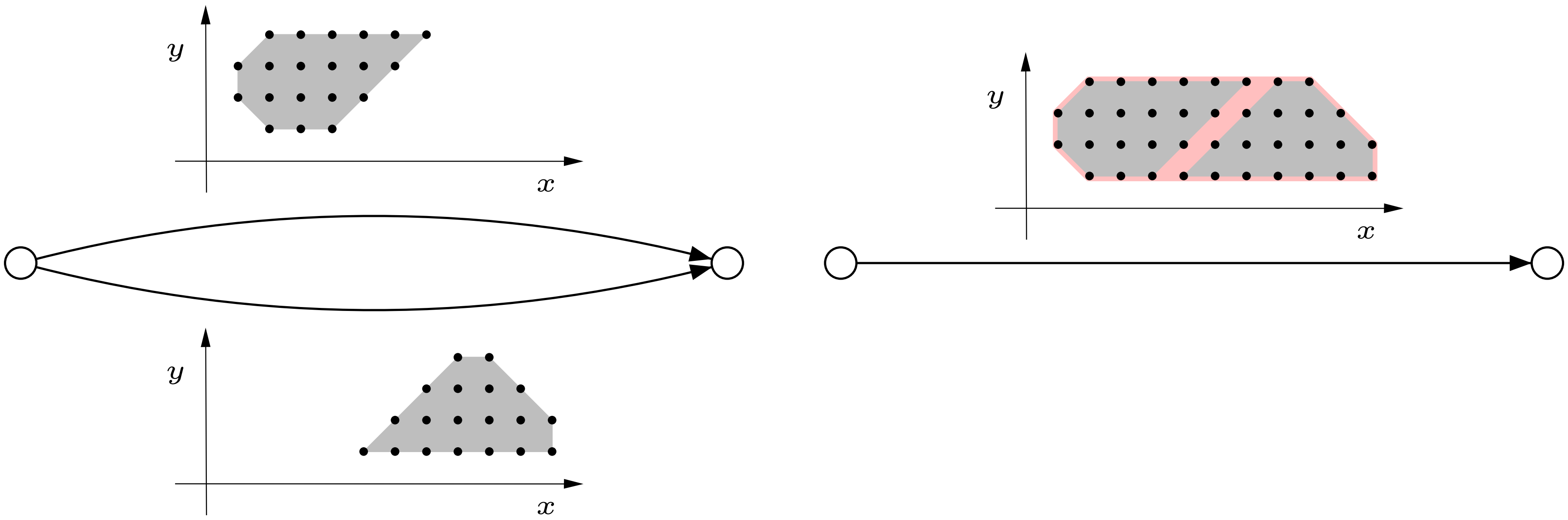
- FLATA guarantees termination for flat models (e.g. automata from SIL logic).

Talk Outline

1. Counter Automata (CA)
2. Classes of Transition Labels
3. Running Example
4. Reachability Analysis of CA
5. **Simplification of Transitions**
6. Future Work
7. Tool Demonstration

Simplification of Disjunctive Transitions

- Precise simplification
  - Bagnara, Hill, Zaffanella. *Exact Join Detection for Convex Polyhedra and Other Numerical Abstractions.* Computational Geometry, 2010

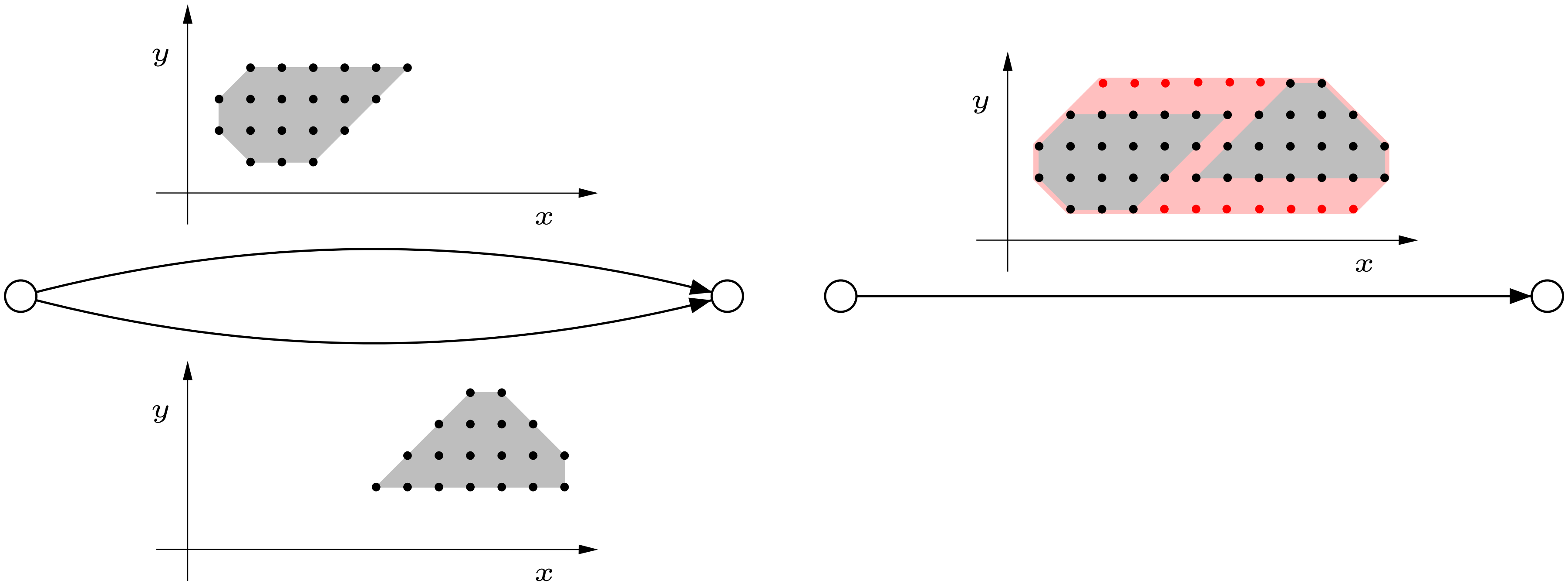


Talk Outline

1. Counter Automata (CA)
2. Classes of Transition Labels
3. Running Example
4. Reachability Analysis of CA
5. Simplification of Transitions
6. **Future Work**
7. Tool Demonstration

Future Work

- Imprecise simplification (over-approximation)
  - octagonal hull



- Abstraction-refinement
- Procedures

Talk Outline

1. Counter Automata (CA)
2. Classes of Transition Labels
3. Running Example
4. Reachability Analysis of CA
5. Simplification of Transitions
6. Future Work
7. **Tool Demonstration**

Experiments with FLATA

model type	model	output	exec. time [s]
VHDL	counter	empty	0.36
	counter+bug	CE	0.59
	register	empty	0.52
	register+bug	CE	0.53
	synlifo	empty	25.19
	synlifo+bug	CE	22.62
Lists	insdel	CE	0.49
	listreversal	empty	9.42
	listcounter	empty	0.79
SIL	simple-a	valid	0.85
	simple-b	falsifiable	1.47
	rotationVC-valid	valid	4.8
	rotationVC-not-valid	falsifiable	2.8
	splitVC-valid	valid	9.55
	splitVC-not-valid	falsifiable	5.41

- Comparison with other tools is to be done.

Tool Demonstration

- CA models
  - VHDL

    Smrčka, Vojnar. *Verifying Parameterised Hardware Designs via Counter Automata.* HVC'07.
  - L2CA

    Bouajjani, Bozga, Habermehl, Iosif, Moro, Vojnar. *Programs with Lists are Counter Automata.* CAV'06
  - SIL

    Habermehl, Iosif, Vojnar. *A Logic of Singly Indexed Arrays.* LPAR'08

`http://www-verimag.imag.fr/FLATA.html`