

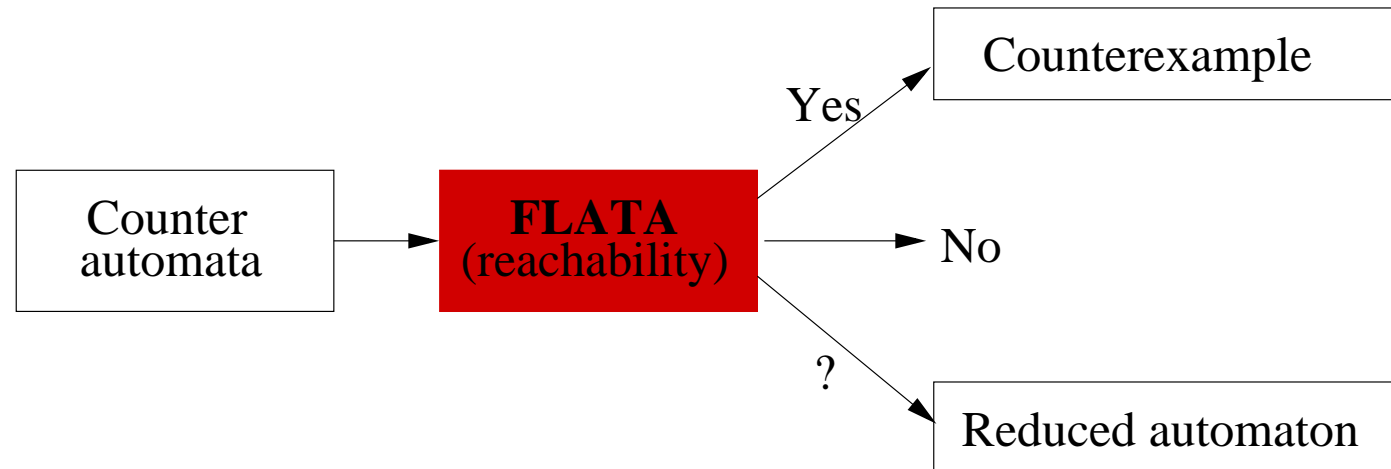
FLATA: A Tool for Manipulation and Analysis of Counter Automata

Marius Bozga¹, Radu Iosif¹, Filip Konečný^{1,2}, Tomáš Vojnar²

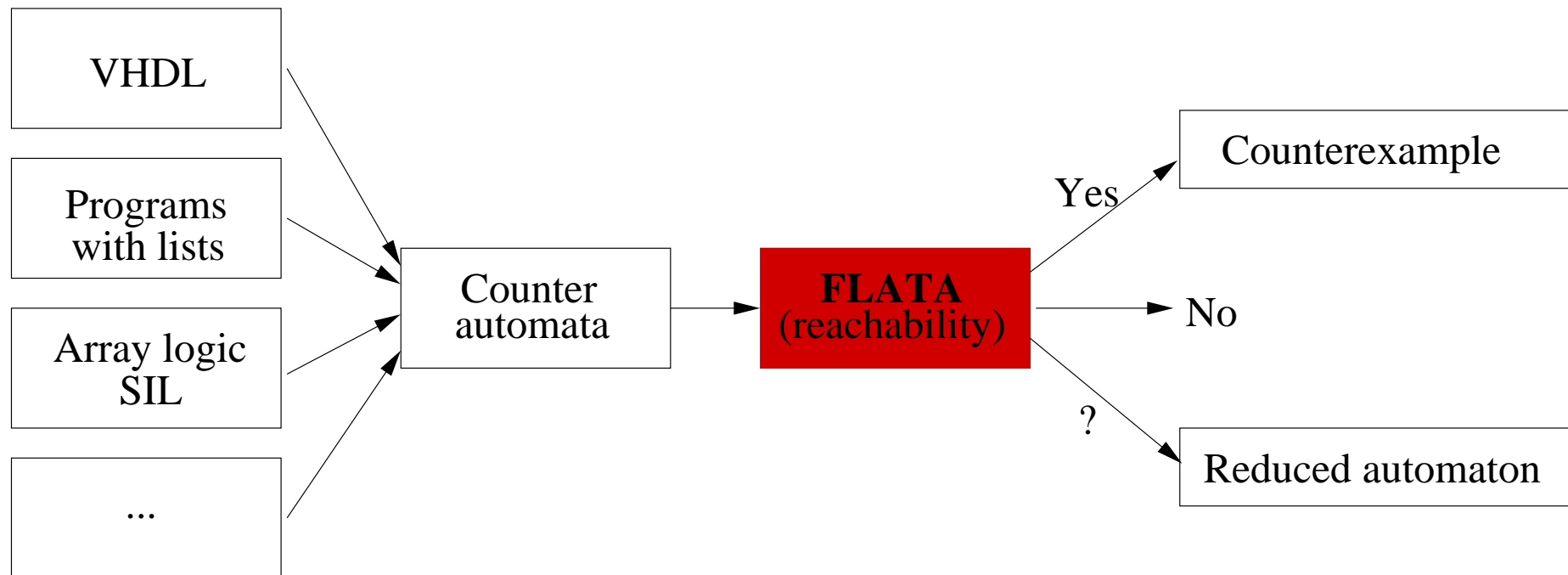
¹ VERIMAG / CNRS / University of Grenoble, France

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**.

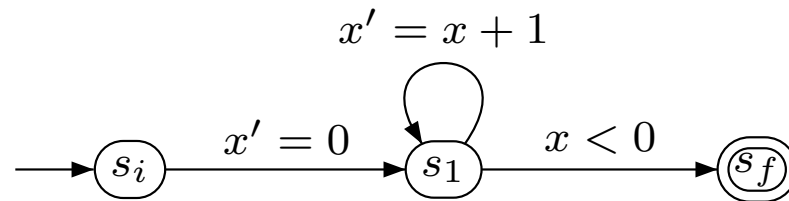
Example

```
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

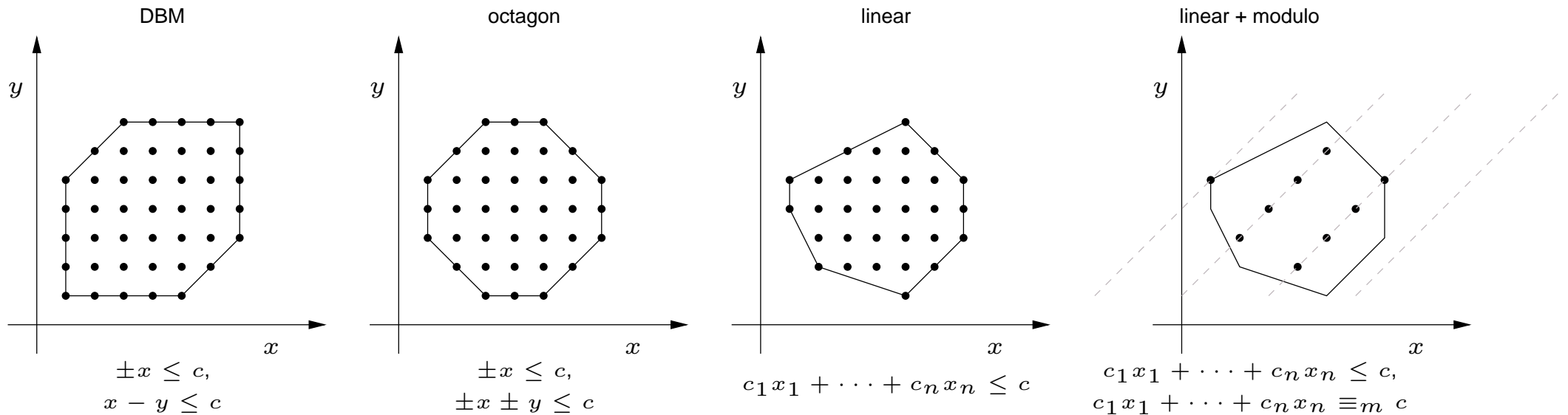
Talk Outline

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



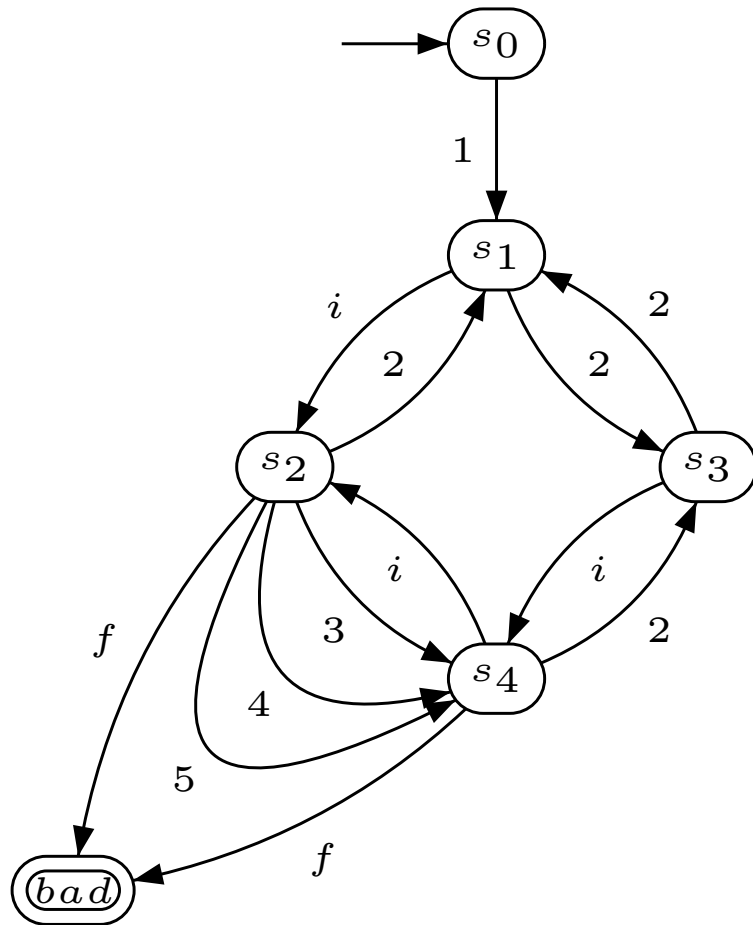
❖ Disjunctions of relations \equiv Presburger arithmetic.

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 exceeds a parametric bound m .



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 :	$[a = m]$	$\{a' = a,$	$m' = m\}$

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

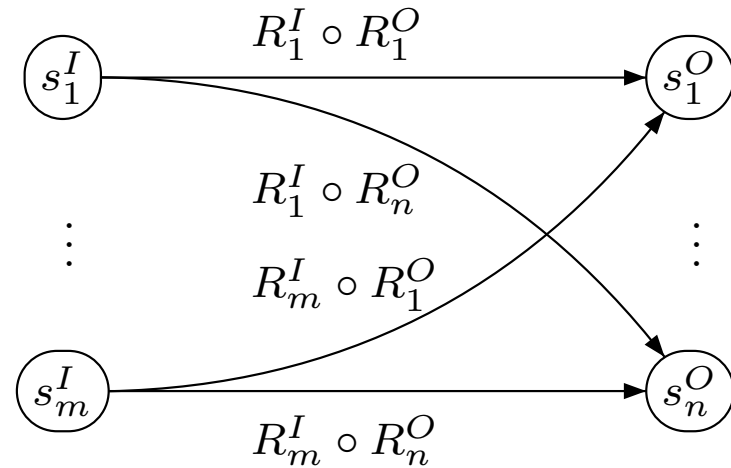
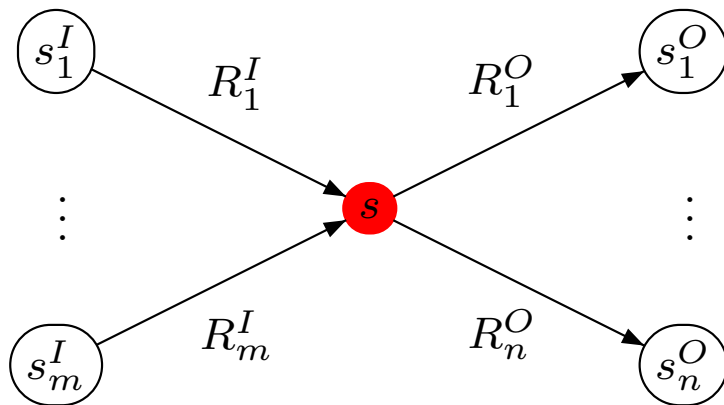
States without Loops

❖ Reduction

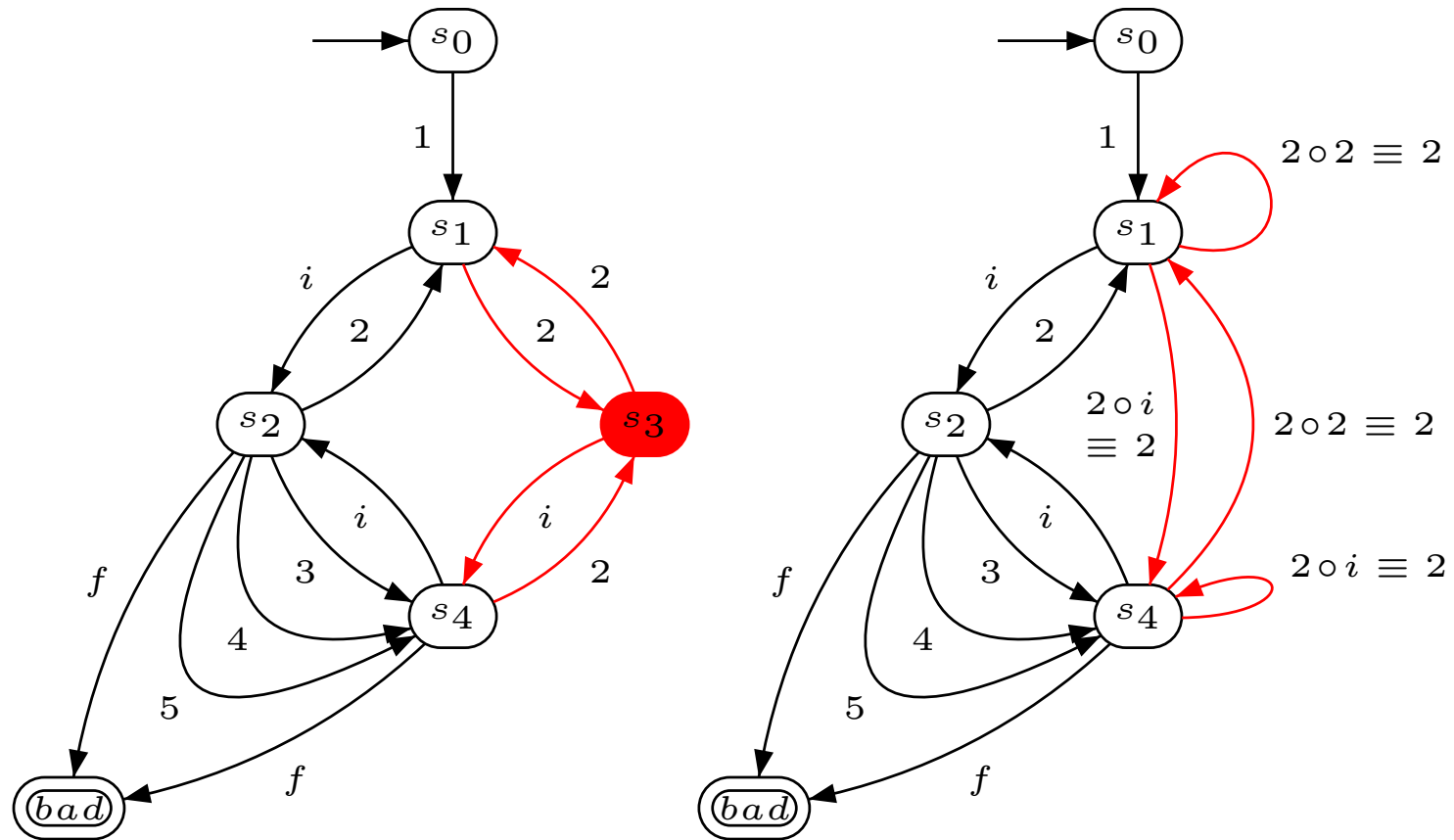
1. consider path segments of the form
2. collapse each of them to a single transition
(operation \circ is relational composition)

$$s_i \xrightarrow{R^I} s \xrightarrow{R^O} s_o$$

$$s_i \xrightarrow{R^I \circ R^O} s_o$$



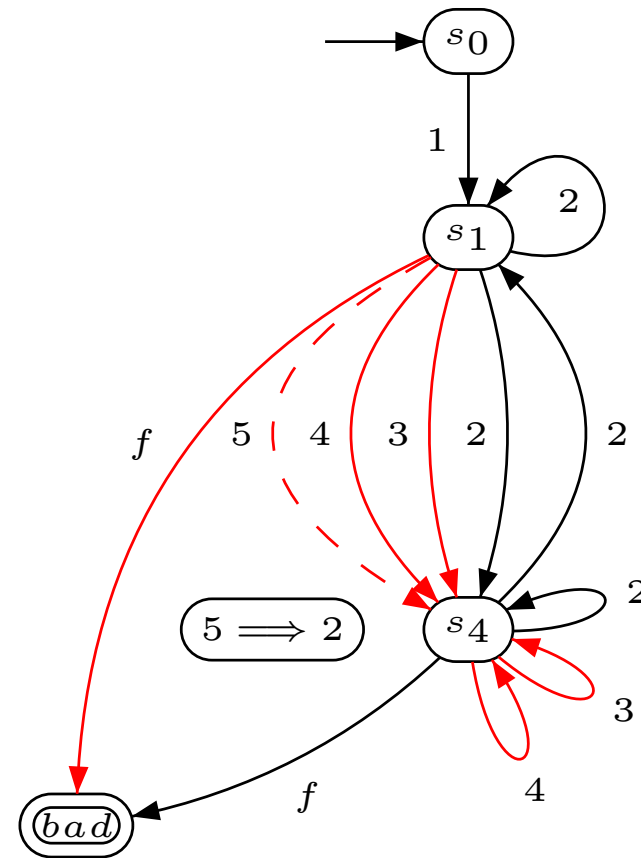
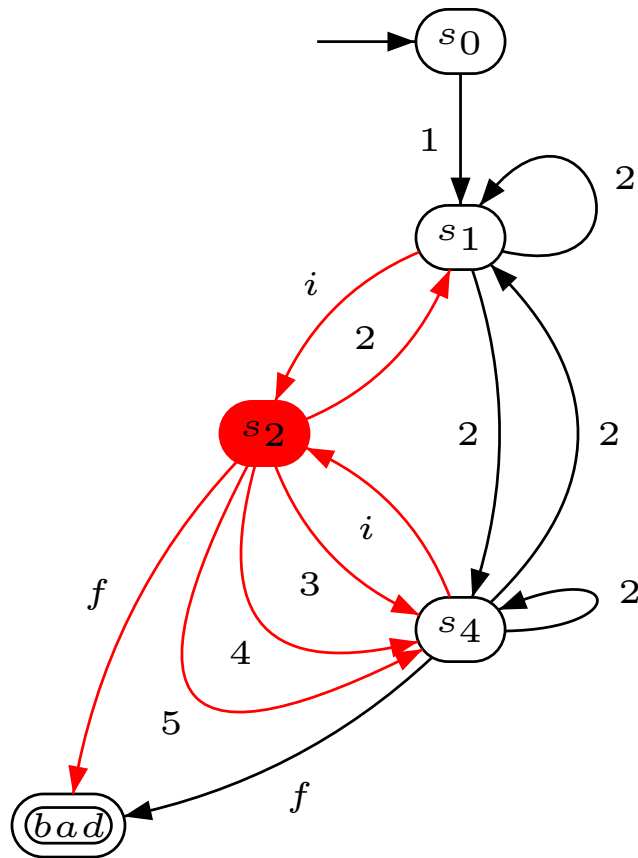
States without Loops – Example



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

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

States without Loops – Example



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

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

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

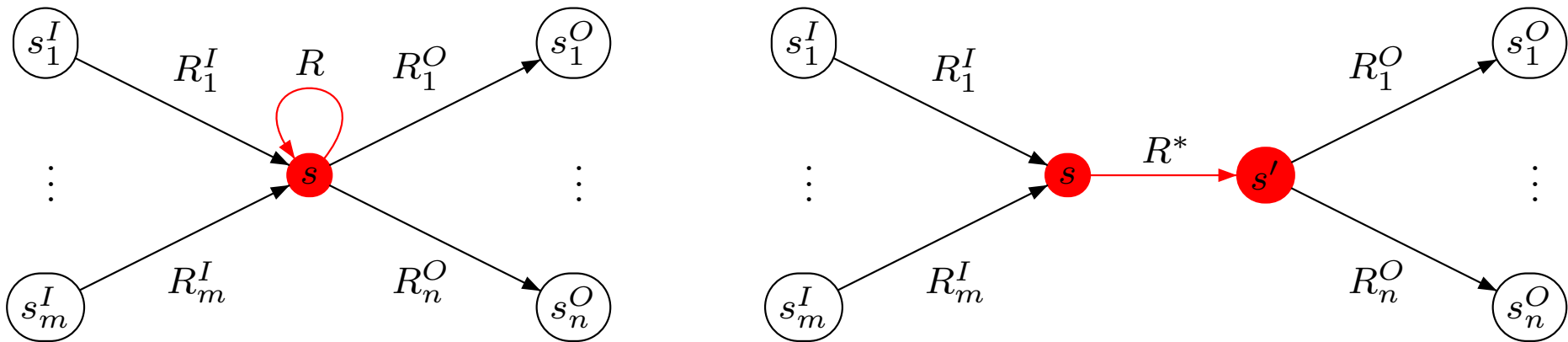
- $(x' = x + 1)^+ \equiv (x' = x + 1) \vee (x' = x + 2) \vee \dots$
 $(x' = x + 1)^+ \equiv (x' \geq x + 1)$
- $(x' = x + 1)^* \equiv (x' = x) \vee (x' = x + 1) \vee (x' = x + 2) \vee \dots$
 $(x' = x + 1)^* \equiv (x' \geq x)$

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

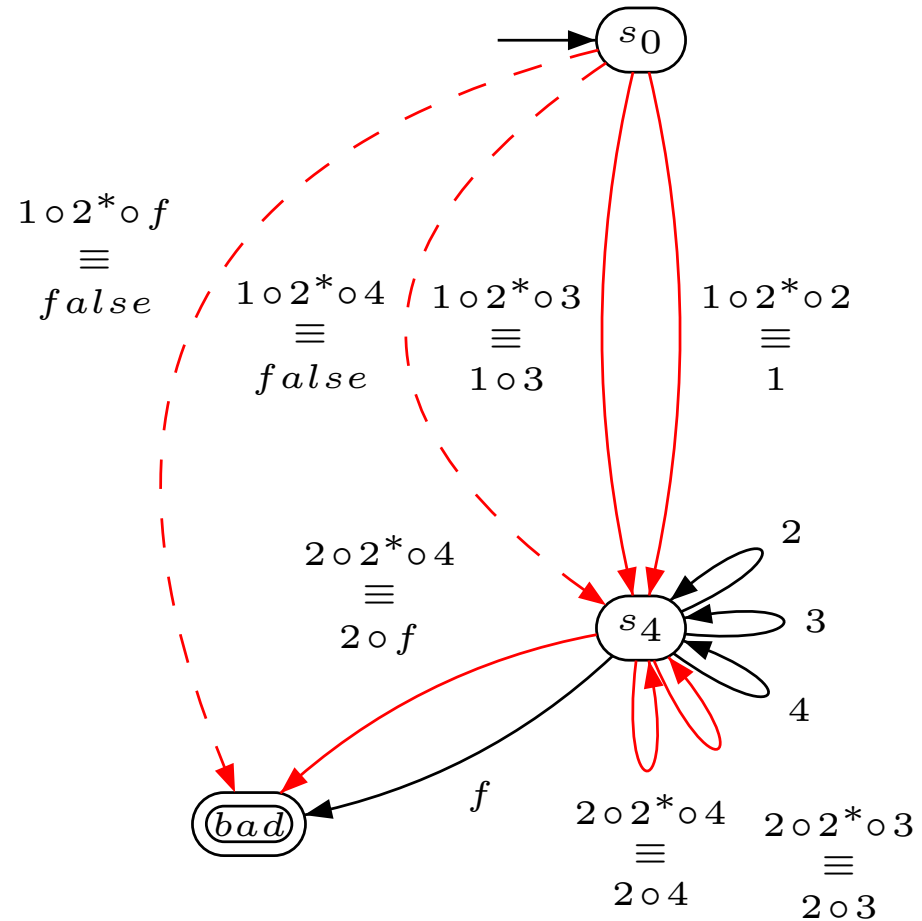
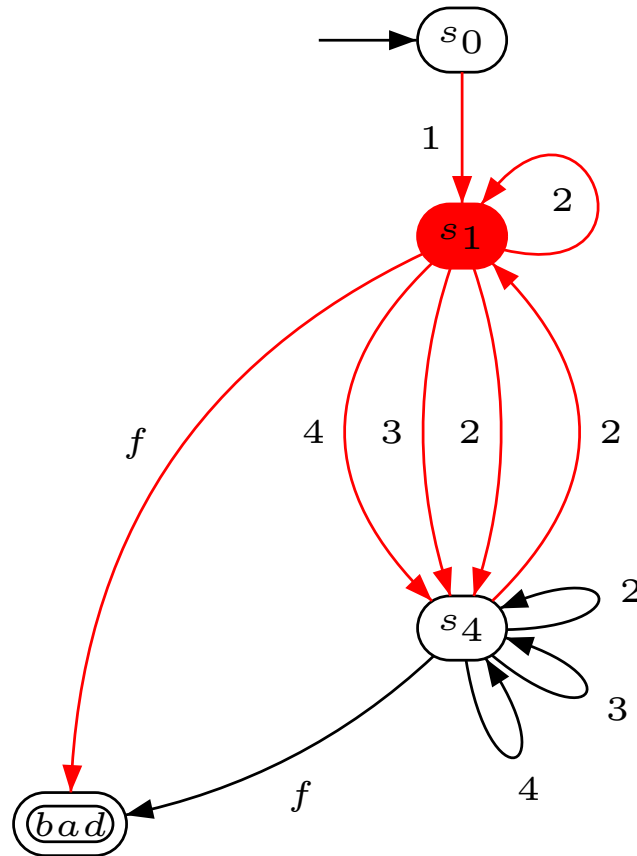
Acceleration of One Self-loop

❖ Reduction

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



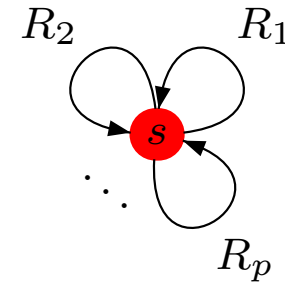
Acceleration of One Self-loop – Example



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

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

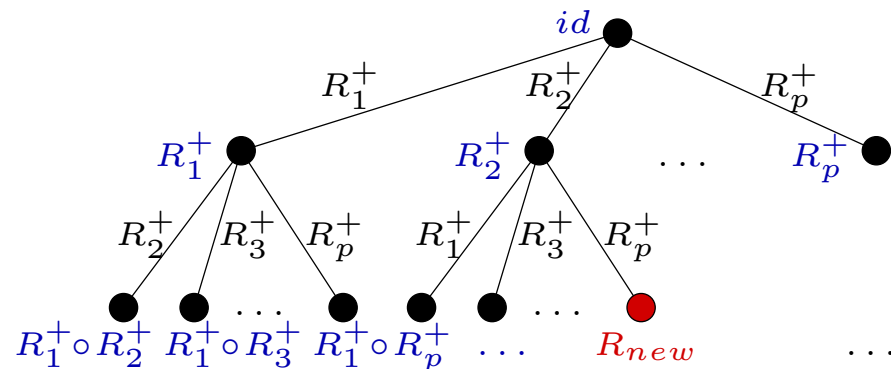
Control states with ≥ 2 loops



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

❖ Semi-algorithm

- build a labeled tree in BFS manner
- edge labels $L = \{R_1^+, R_2^+, \dots, R_p^+\}$
- for $l \in L$, define $\text{succ}(l) = \{l' \in L \mid l' \neq l \wedge l \circ l' \neq \text{false}\}$



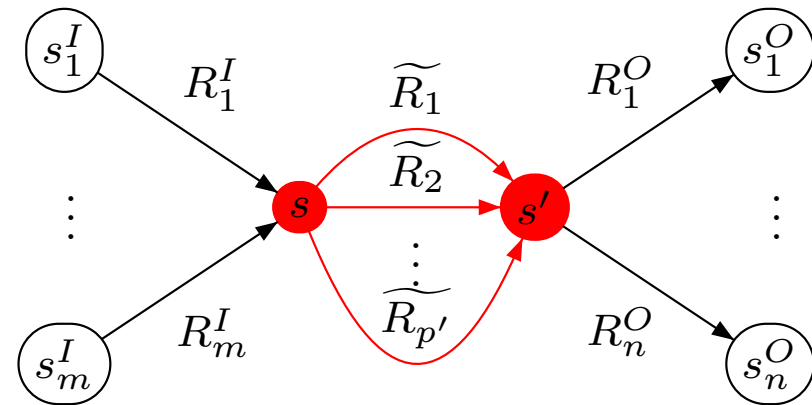
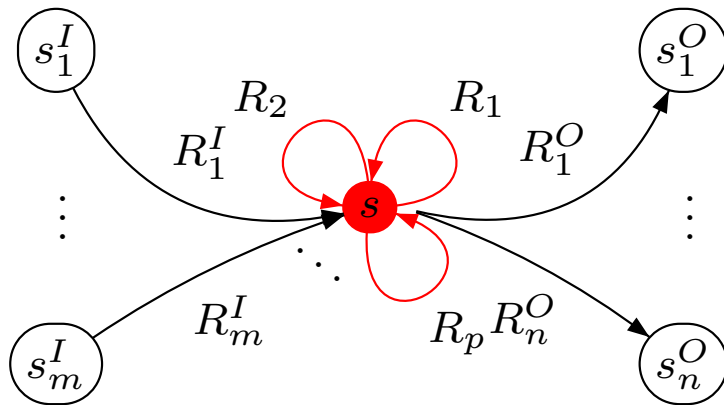
Cl – set of node labels

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

- continue until working list of nodes is empty
- or until $\text{depth} = MAX$

Control states with ≥ 2 loops

If semi-algorithm succeeded, let $Cl = \{\widetilde{R}_1, \widetilde{R}_2, \dots, \widetilde{R}_{p'}\}$. Use meta-transitions in Cl to reduce CA.



Flat Counter Automata

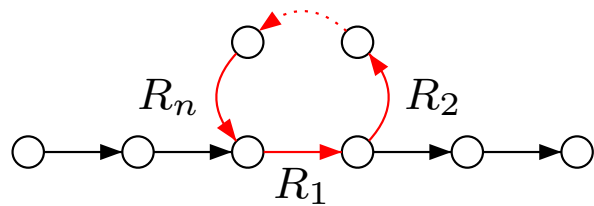
- ❖ Flat CA – automata restricted in terms of:
 1. control structure – no nested cycles
 2. labeling of transitions inside cycles – DBM, Octagon

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



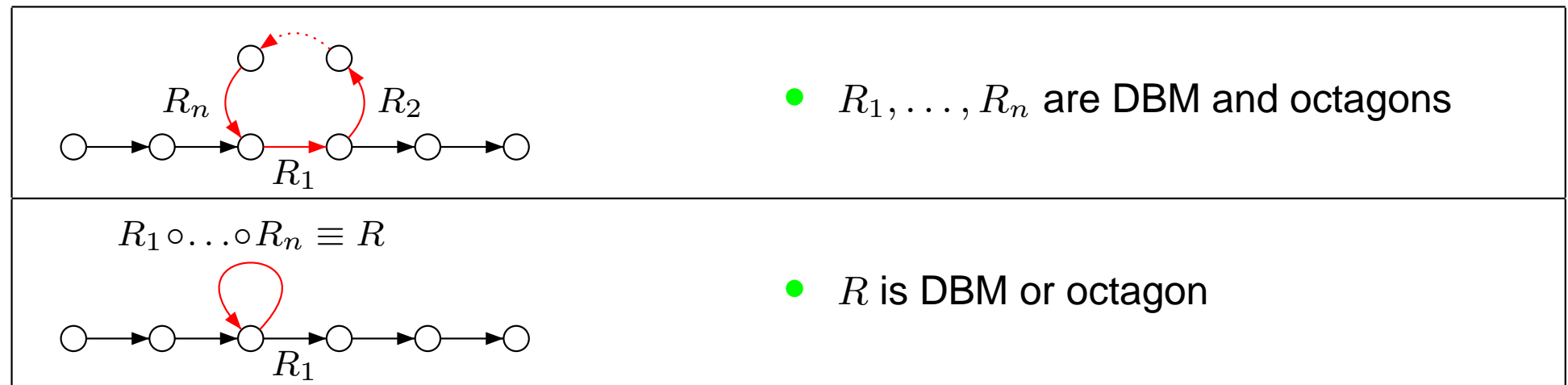
- R_1, \dots, R_n are DBM and octagons

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

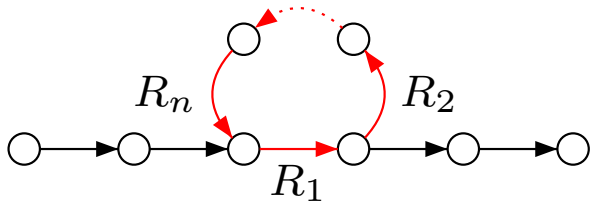
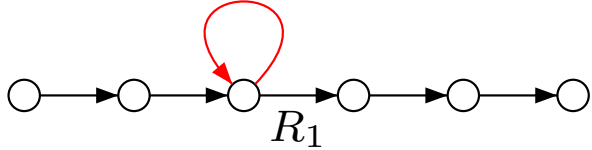
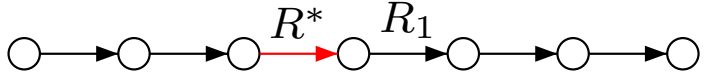


Flat Counter Automata

❖ 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

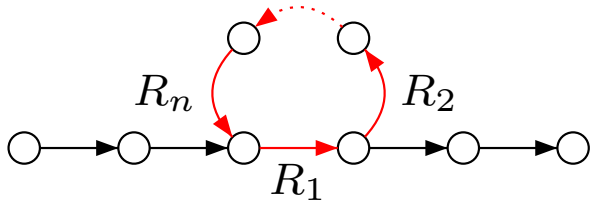
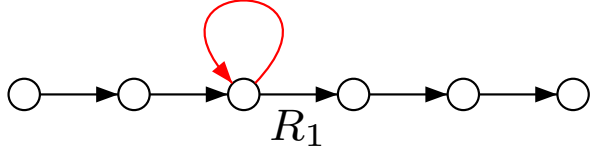
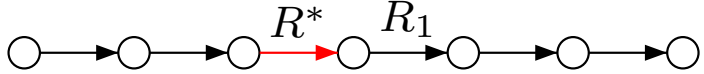
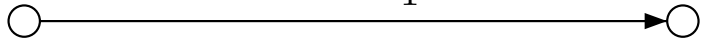
 <p>A sequence of six nodes connected by black arrows. The third node has a red arrow labeled R_n pointing to the fourth node. The fourth node has a red arrow labeled R_2 pointing to the fifth node. The fifth node has a red arrow labeled R_1 pointing back to the third node. A dotted red arrow also connects the fourth node to the fifth node.</p>	<ul style="list-style-type: none"> • R_1, \dots, R_n are DBM and octagons
<p>$R_1 \circ \dots \circ R_n \equiv R$</p>  <p>A sequence of six nodes connected by black arrows. The third node has a red self-loop arrow labeled R_1.</p>	<ul style="list-style-type: none"> • R is DBM or octagon
 <p>A sequence of six nodes connected by black arrows. The third node has a red arrow labeled R^* pointing to the fourth node. The fourth node has a red arrow labeled R_1 pointing to the fifth node.</p>	<ul style="list-style-type: none"> • 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

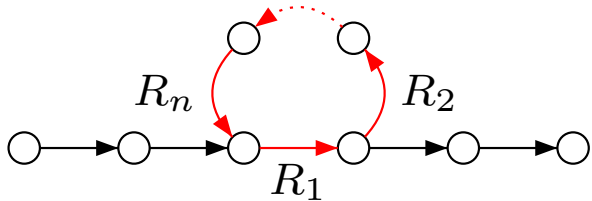
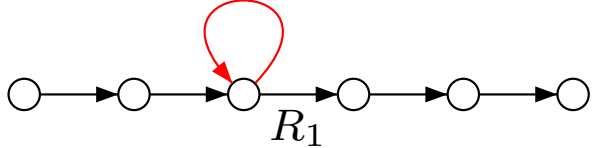
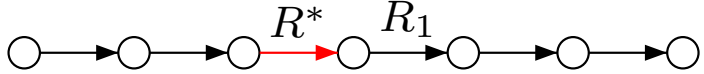
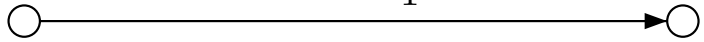
 <p>A sequence of five nodes connected by black arrows. A cycle of three nodes is highlighted with red arrows: the second node points to the third (labeled R_n), the third points to the fourth (labeled R_1), and the fourth points back to the second (labeled R_2). A dotted red arrow also points from the second node to the third.</p>	<ul style="list-style-type: none"> • R_1, \dots, R_n are DBM and octagons
<p>$R_1 \circ \dots \circ R_n \equiv R$</p>  <p>A sequence of five nodes connected by black arrows. The third node has a red self-loop arrow labeled R_1.</p>	<ul style="list-style-type: none"> • R is DBM or octagon
 <p>A sequence of five nodes connected by black arrows. A red arrow labeled R^* points from the second node to the third, and a black arrow labeled R_1 points from the third to the fourth.</p>	<ul style="list-style-type: none"> • R^* is Presburger definable
<p>$\dots \circ R^* \circ R_1 \circ \dots$</p>  <p>A single black arrow labeled $\dots \circ R^* \circ R_1 \circ \dots$ connecting two nodes.</p>	<ul style="list-style-type: none"> • 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

 <p>A sequence of five nodes connected by black arrows. A cycle of three nodes is highlighted with red arrows and labels: R_n (top), R_2 (right), and R_1 (bottom). A dotted red arrow indicates a continuation of the cycle.</p>	<ul style="list-style-type: none"> • R_1, \dots, R_n are DBM and octagons
<p>$R_1 \circ \dots \circ R_n \equiv R$</p>  <p>A sequence of five nodes connected by black arrows. A self-loop on the third node is highlighted with a red arrow and labeled R_1.</p>	<ul style="list-style-type: none"> • R is DBM or octagon
 <p>A sequence of five nodes connected by black arrows. A red arrow between the third and fourth nodes is labeled R^*, and the next black arrow is labeled R_1.</p>	<ul style="list-style-type: none"> • R^* is Presburger definable
<p>$\dots \circ R^* \circ R_1 \circ \dots$</p>  <p>A single long black arrow between two nodes.</p>	<ul style="list-style-type: none"> • Presburger formula

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

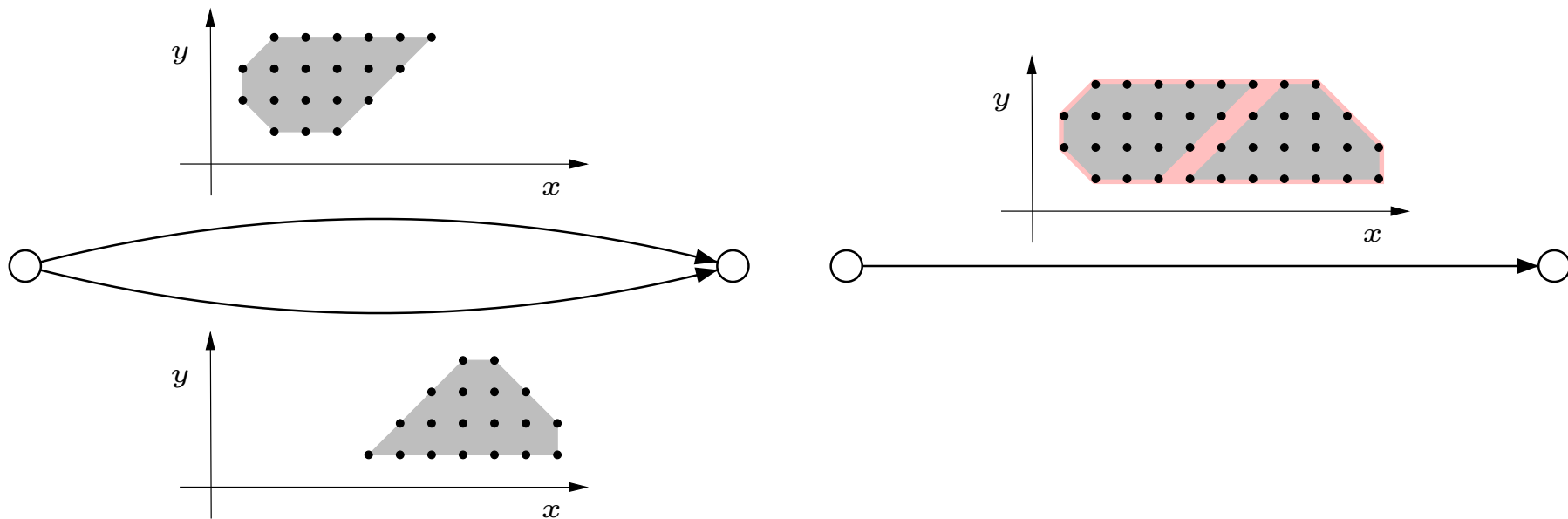
Talk Outline

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



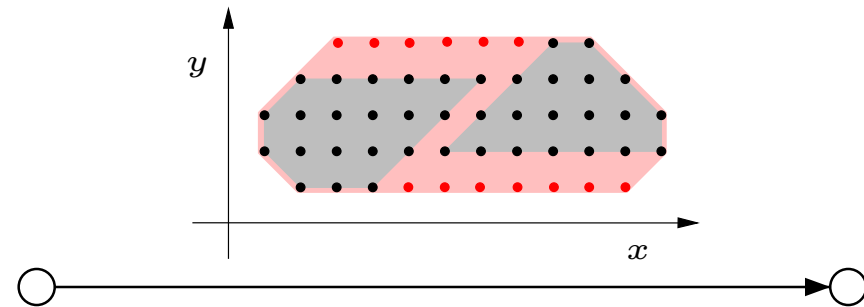
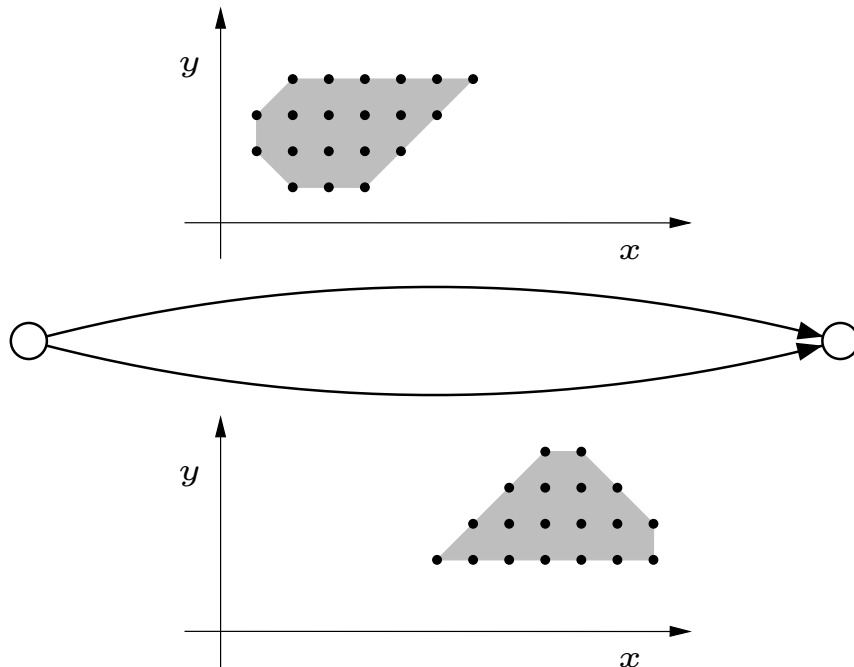
Talk Outline

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>