Analysis of Arithmetic Prolog Programs using Abstract Interpretation

Alexander Weinert

Master Thesis Presentation

June 22, 2015

Roadmap

Program

YES/NO/
MAYBE

Roadmap

Program $\longrightarrow$ YES/NO/MAYBE

Roadmap

Program $\longrightarrow$ YES/NO/MAYBE

Question: Do all evaluations of the program terminate?

```
fac(X)  =
```

Prolog - Introduction

```
fac(X)  =
        if X > 0 then Y1 = fac(X-1), return Y1 * X
```

Prolog - Introduction

```
fac(X)  =
        if X > 0 then Y1 = fac(X-1), return Y1 * X
        if X == 0 then return 1
```

Prolog - Introduction

```
fac(X)  =
        if X > 0 then Y1 = fac(X-1), return Y1 * X
        if X == 0 then return 1


fac(X, Y)  :-
```

PROLOG - Introduction

```
fac(X)  =
        if X > 0 then Y1 = fac(X-1), return Y1 * X
        if X == 0 then return 1
```

```
fac(X, Y)  :- X > 0,
```

PROLOG - Introduction

```
fac(X)  =
        if X > 0 then Y1 = fac(X-1), return Y1 * X
        if X == 0 then return 1
```

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1),
```

Prolog - Introduction

```
fac(X)  =
        if X > 0 then Y1 = fac(X-1), return Y1 * X
        if X == 0 then return 1
```

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
```

Prolog - Introduction

```
fac(X)  =
        if X > 0 then Y1 = fac(X-1), return Y1 * X
        if X == 0 then return 1
```

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :-
```

```
fac(X)  =
        if X > 0 then Y1 = fac(X-1), return Y1 * X
        if X == 0 then return 1
```

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0,
```

```
fac(X)   =
         if X > 0 then Y1 = fac(X-1), return Y1 * X
         if X == 0 then return 1
```

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```

Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```

$\boxed{fac(1, Res)}$

Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```

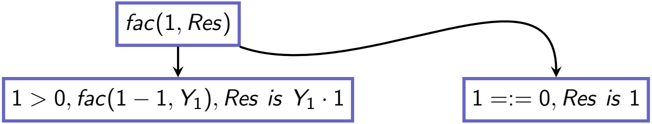
$$\boxed{\mathit{fac}(1, \mathit{Res})}$$

$$\downarrow$$

$$\boxed{1 > 0, \mathit{fac}(1 - 1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1}$$

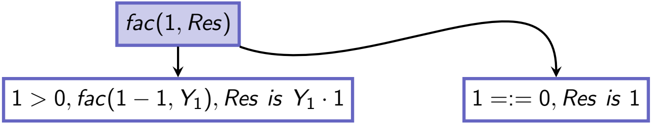
Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```



Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```



Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```

$fac(1, Res)$

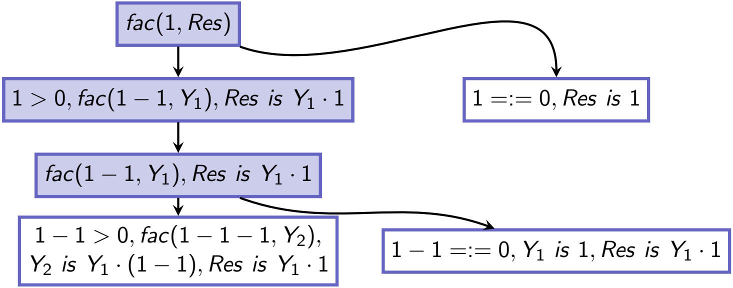
$1 > 0, fac(1-1, Y_1), Res\ is\ Y_1 \cdot 1$

$1 =:= 0, Res\ is\ 1$

$fac(1-1, Y_1), Res\ is\ Y_1 \cdot 1$

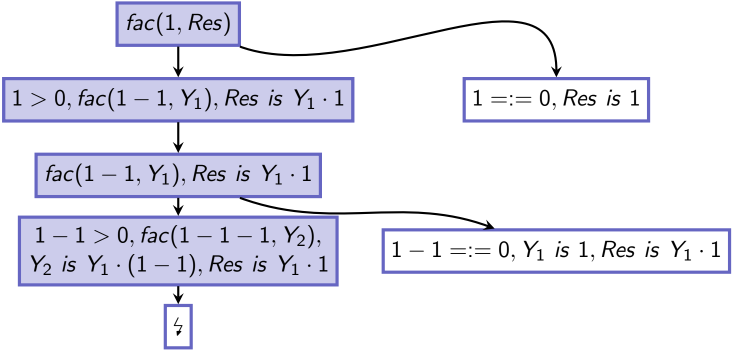
Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```



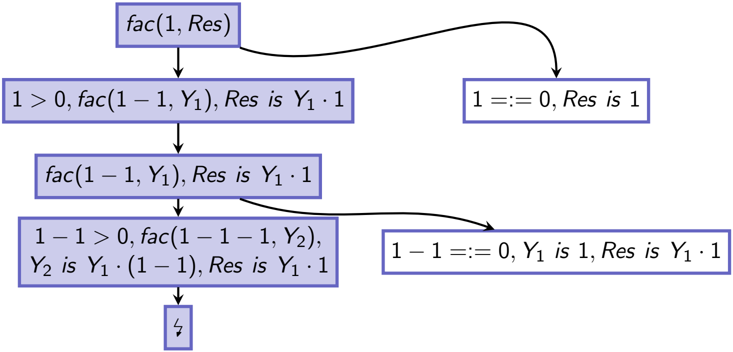
Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```



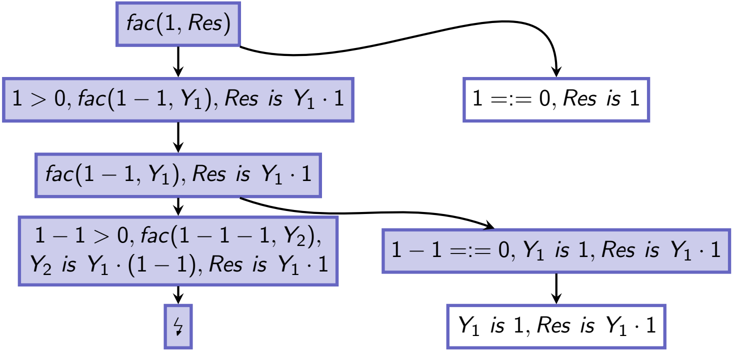
Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```



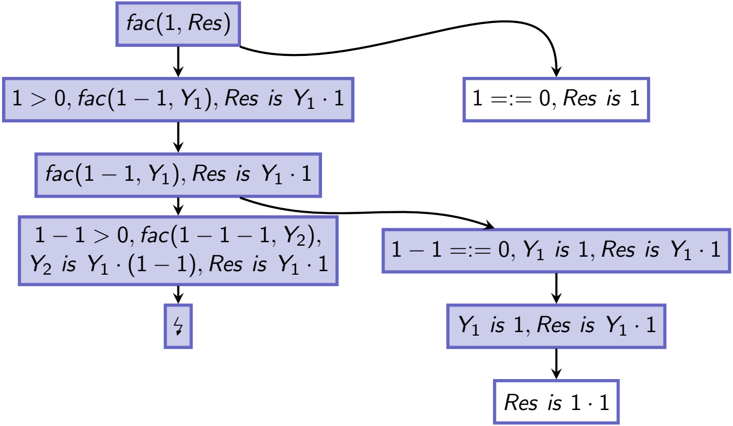
Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```



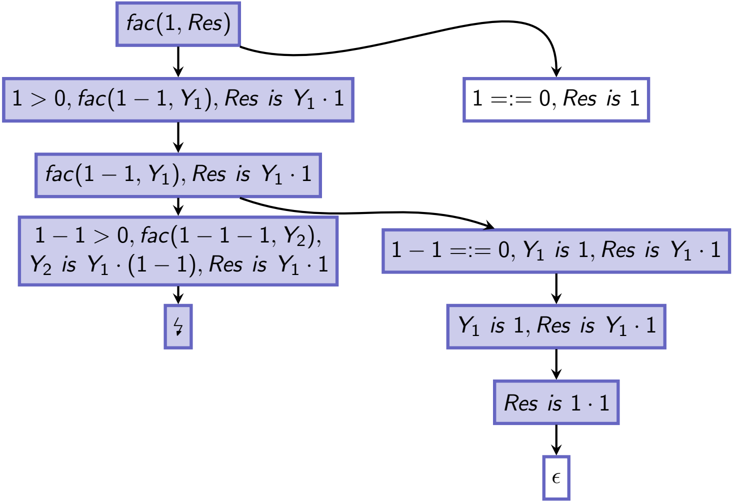
Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```



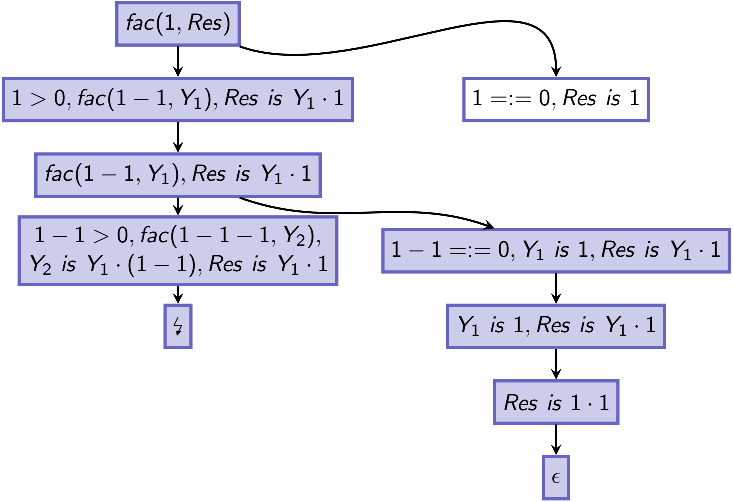
Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```



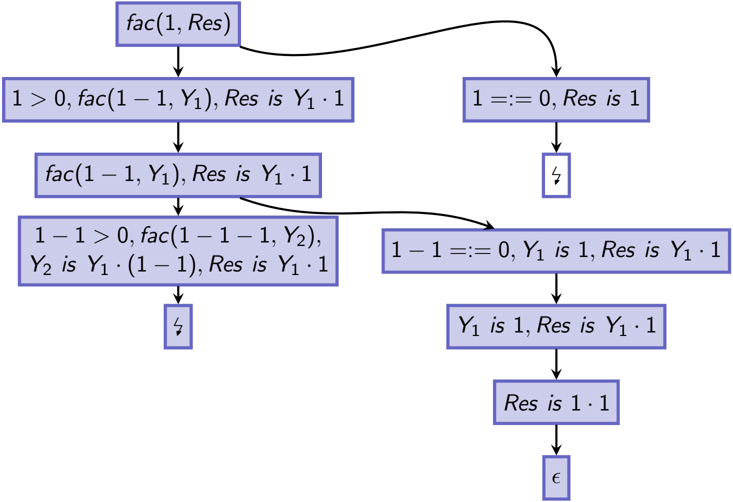
PROLOG - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```



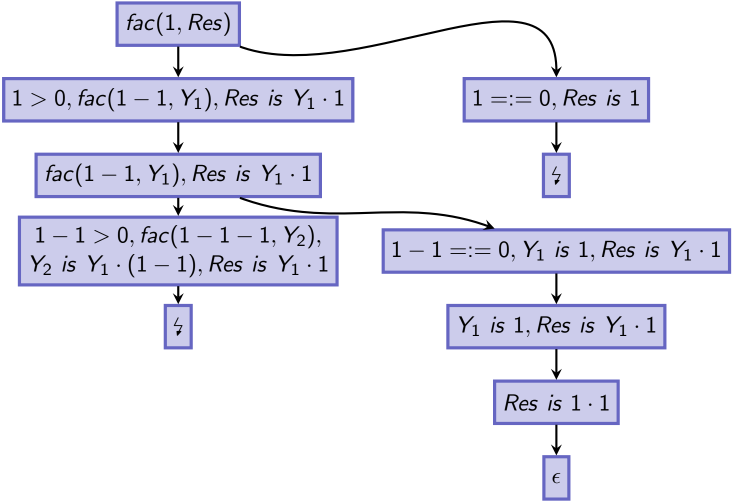
Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```



Prolog - Evaluation

```
fac(X, Y)  :- X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```



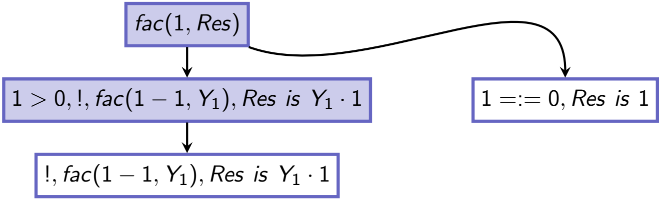
Prolog - Cut

```
fac(X, Y)  :- X > 0,    fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```

Prolog - Cut

```
fac(X, Y)  :- X > 0, !, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```

Prolog - Cut


$fac(1, Res)$
$1 > 0, !, fac(1-1, Y_1), Res \text{ is } Y_1 \cdot 1$
$1 =:= 0, Res \text{ is } 1$
$!, fac(1-1, Y_1), Res \text{ is } Y_1 \cdot 1$

Prolog - Cut

$$\begin{tikzcd}
\boxed{fac(1, Res)} \arrow[d] \arrow[dr, bend left] & \\
\boxed{1 > 0, !, fac(1-1, Y_1), Res\ is\ Y_1 \cdot 1} \arrow[d] & \boxed{1 =:= 0, Res\ is\ 1} \\
\boxed{!, fac(1-1, Y_1), Res\ is\ Y_1 \cdot 1} \arrow[d] & \\
\boxed{fac(1-1, Y_1), Res\ is\ Y_1 \cdot 1} &
\end{tikzcd}$$

PROLOG - Cut

$$fac(1, Res)$$

$$1 > 0, !, fac(1-1, Y_1), Res \; is \; Y_1 \cdot 1$$

$$1 =:= 0, Res \; is \; 1$$

$$!, fac(1-1, Y_1), Res \; is \; Y_1 \cdot 1$$

$$fac(1-1, Y_1), Res \; is \; Y_1 \cdot 1$$

Prolog - Cut

$$fac(1, Res)$$

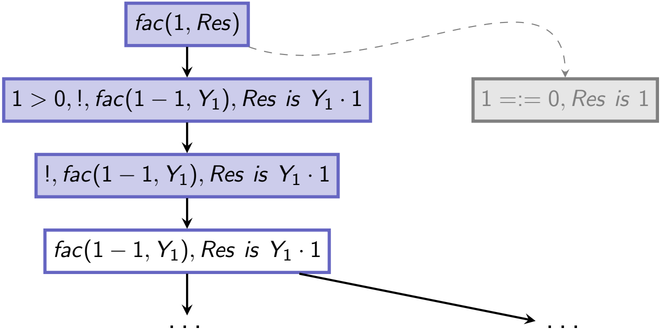
$$1 > 0, !, fac(1-1, Y_1), Res \ is \ Y_1 \cdot 1$$

$$1 =:= 0, Res \ is \ 1$$

$$!, fac(1-1, Y_1), Res \ is \ Y_1 \cdot 1$$

$$fac(1-1, Y_1), Res \ is \ Y_1 \cdot 1$$

Prolog - Cut


$fac(1, Res)$
$1 > 0, !, fac(1-1, Y_1), Res\ is\ Y_1 \cdot 1$
$1 =:= 0, Res\ is\ 1$
$!, fac(1-1, Y_1), Res\ is\ Y_1 \cdot 1$
$fac(1-1, Y_1), Res\ is\ Y_1 \cdot 1$
...
...

Termination

Given: Prolog Program, some query template

Question:

Termination

Given: Prolog Program, some query template

Question: For all queries matching the template:

Termination

Given: Prolog Program, some query template

Question: For all queries matching the template: Does the inference of the query on the program eventually terminate?

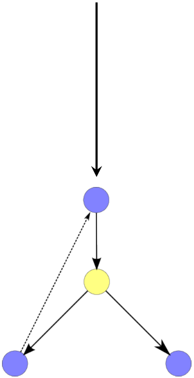
Roadmap

Program

YES/NO/
MAYBE

Roadmap

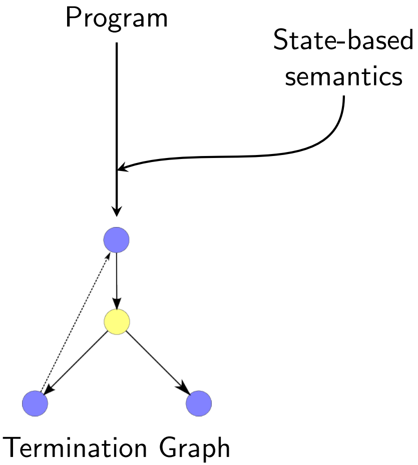
Program



Termination Graph

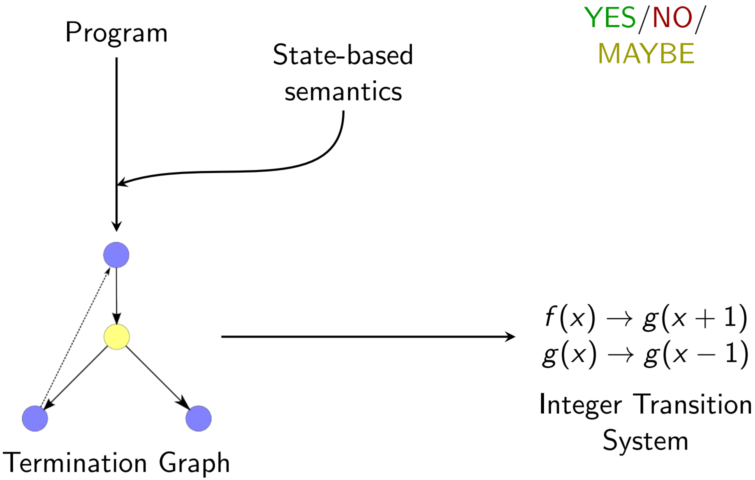
YES/NO/
MAYBE

Roadmap



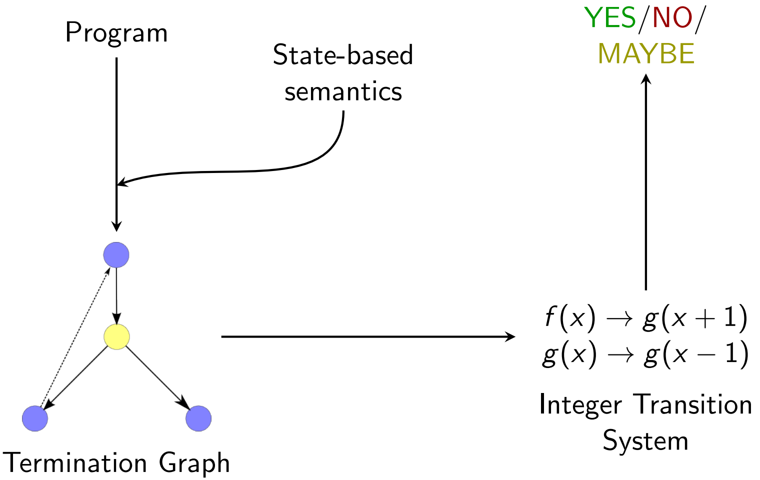
YES/NO/
MAYBE

Roadmap



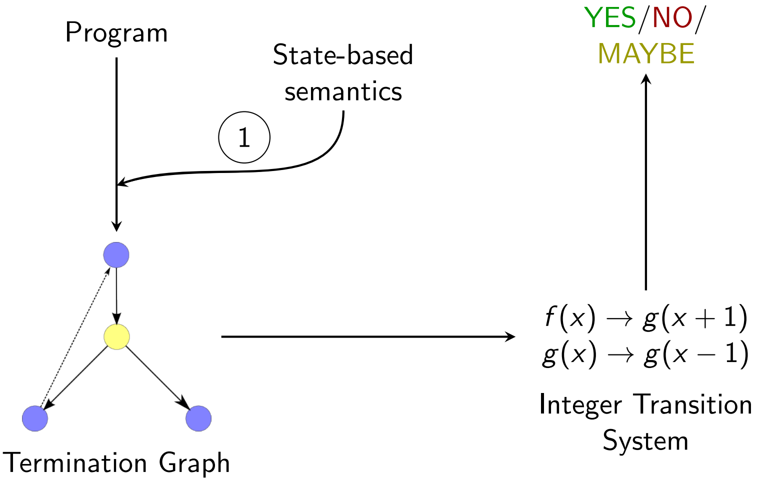
Program
State-based semantics
YES/NO/MAYBE
Termination Graph
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Integer Transition System

Roadmap



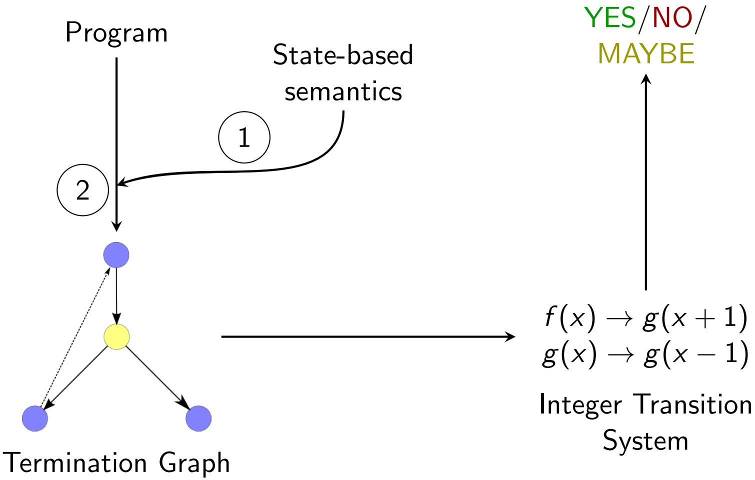
Program
State-based semantics
YES/NO/MAYBE
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Integer Transition System
Termination Graph

Roadmap

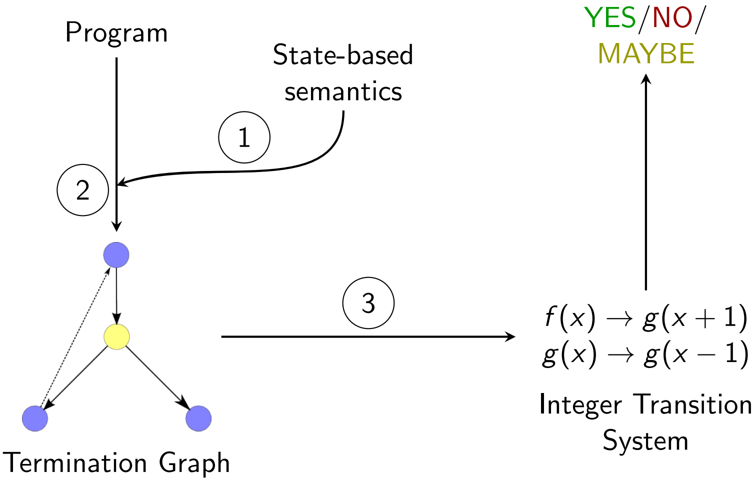


Program
State-based semantics
1
YES/NO/MAYBE
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Integer Transition System
Termination Graph

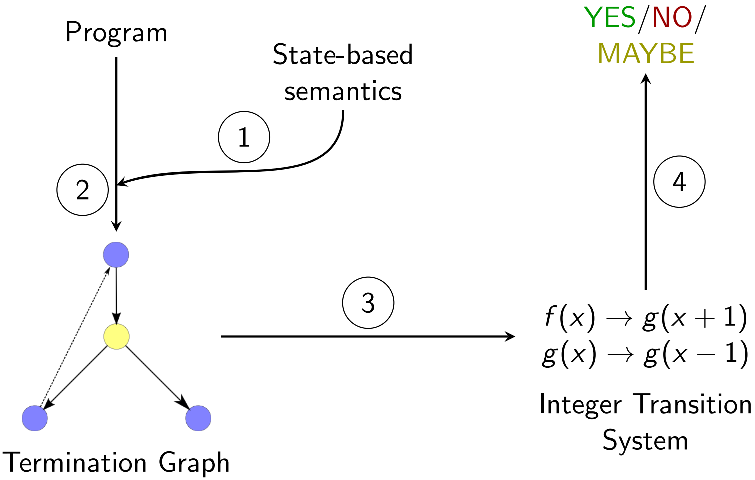
Roadmap


Program
State-based
semantics
1
2
YES/NO/
MAYBE
$f(x) \rightarrow g(x+1)$
$g(x) \rightarrow g(x-1)$
Integer Transition
System
Termination Graph

Roadmap


YES/NO/
MAYBE
Program
State-based
semantics
1
2
3
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Integer Transition
System
Termination Graph

Roadmap



YES/NO/
MAYBE
Program
State-based
semantics
1
2
4
3
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Integer Transition
System
Termination Graph

From Trees to States

- Prolog: Tree-based semantics
- Well-known techniques: State-based semantics

From Trees to States

- Prolog: Tree-based semantics
- Well-known techniques: State-based semantics

Solution: State-based semantics for Prolog
(Linear Operational Semantics, Ströder et al., 2012)

From Trees to States

- Prolog: Tree-based semantics
- Well-known techniques: State-based semantics

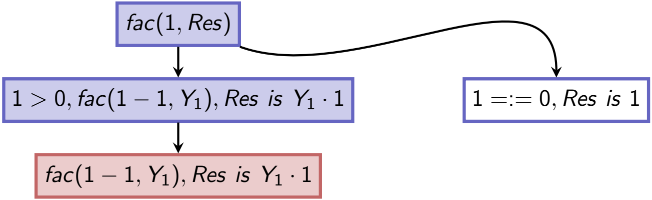
Solution: State-based semantics for Prolog
(Linear Operational Semantics, Ströder et al., 2012)

- Basic Idea: Leaves of tree describe state of inference
- Use front of tree as state

From Trees to States

$$\begin{tikzcd}
\boxed{\mathit{fac}(1, \mathit{Res})} \arrow[d] \arrow[rr, bend left, shift left] & & \\
\boxed{1 > 0, \mathit{fac}(1 - 1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1} \arrow[d] & & \boxed{1 =:= 0, \mathit{Res}\ \mathit{is}\ 1} \\
\boxed{\mathit{fac}(1 - 1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1} & &
\end{tikzcd}$$

From Trees to States



$$\boxed{fac(1-1, Y1), Res\ is\ Y_1 \cdot 1}$$

From Trees to States

$$
\begin{tikzcd}
\boxed{\mathit{fac}(1, \mathit{Res})} \arrow[d] \arrow[rr, bend left] & & \\
\boxed{1 > 0, \mathit{fac}(1 - 1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1} \arrow[d] & & \boxed{1 =:= 0, \mathit{Res}\ \mathit{is}\ 1} \\
\boxed{\mathit{fac}(1 - 1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1} & &
\end{tikzcd}
$$

$$\mathit{fac}(1 - 1, Y1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1 \quad \mid \boxed{1 =:= 0, \mathit{Res}\ \mathit{is}\ 1}$$

From Trees to States

```
fac(X, Y)  :-  X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :-  X =:= 0, Y is 1.
```

$$fac(1, Res)$$

$$\downarrow$$

From Trees to States

```
fac(X, Y)  :-  X > 0, !, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :-  X =:= 0, Y is 1.
```

$$fac(1, Res)$$

$$\downarrow$$

$$\boxed{1 > 0, !, fac(1-1, Y_1), Res\ is\ Y_1 \cdot 1} \quad |$$

From Trees to States

```
fac(X, Y)  :-  X > 0, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :-  X =:= 0, Y is 1.
```

$$fac(1, Res)$$

$$\downarrow$$

$$1 > 0, !, fac(1 - 1, Y_1), Res \ is \ Y_1 \cdot 1 \quad | \quad \boxed{1 =:= 0, Res \ is \ 1}$$

From Trees to States

$$1 > 0\ , !, \mathit{fac}(1-1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1 \mid 1 =:= 0, \mathit{Res}\ \mathit{is}\ 1$$

$$\downarrow$$

From Trees to States

$$\boxed{1 > 0}, !, \mathit{fac}(1-1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1 \mid 1 =:= 0, \mathit{Res}\ \mathit{is}\ 1$$

$$\downarrow$$

From Trees to States

$$1 > 0\ , !, \mathit{fac}(1 - 1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1 \mid 1 =:= 0, \mathit{Res}\ \mathit{is}\ 1$$

$$\downarrow$$

$$!\ , \mathit{fac}(1 - 1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1 \mid\ 1 =:= 0,\ \mathrm{Res}\ \ \mathit{is}\ 1$$

From Trees to States

$$1 > 0\ , !, \mathit{fac}(1-1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1 \mid 1 =:= 0, \mathit{Res}\ \mathit{is}\ 1$$

$$\downarrow$$

$$\boxed{!}, \mathit{fac}(1-1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1 \mid \boxed{1 =:= 0,\ \mathrm{Res}\ \ \mathit{is}\ 1}$$

$$\downarrow$$

From Trees to States

$$1 > 0\ , !, \mathit{fac}(1-1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1 \mid 1 =:= 0, \mathit{Res}\ \mathit{is}\ 1$$

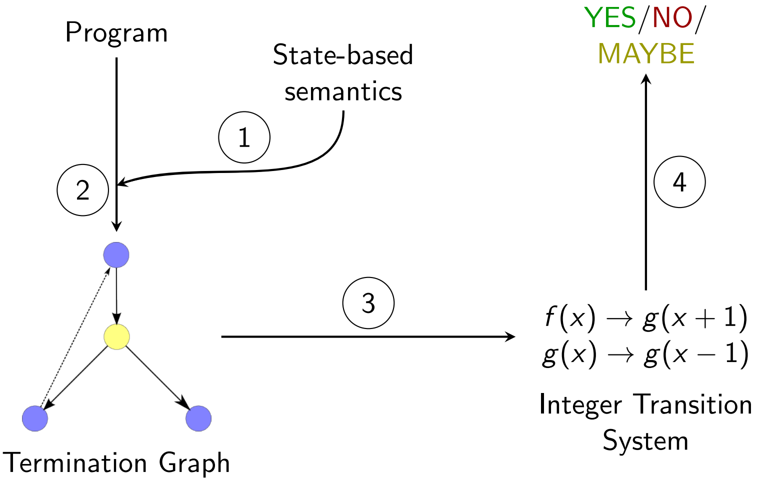
$$\downarrow$$

$$!\ , \mathit{fac}(1-1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1 \mid\ 1 =:= 0,\ \mathrm{Res}\ \ \mathit{is}\ 1$$

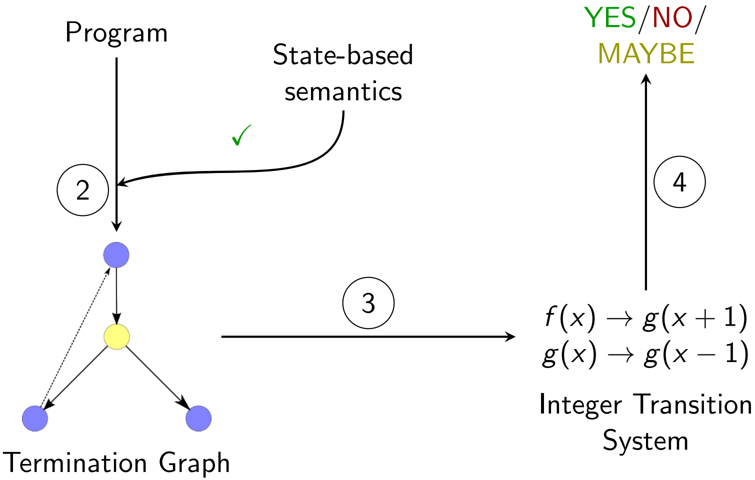
$$\downarrow$$

$$\mathit{fac}(1-1, Y_1), \mathit{Res}\ \mathit{is}\ Y_1 \cdot 1$$

Roadmap


Program
State-based semantics
YES/NO/MAYBE
1
2
3
4
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Integer Transition System
Termination Graph

Roadmap



Program
State-based
semantics
✓
2
Termination Graph
3
4
YES/NO/
MAYBE
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Integer Transition
System

From Programs to Graphs

Given: Some Program, some query template

Goal: Finite representation of all possible inferences

From Programs to Graphs

Given: Some Program, some query template

Goal: Finite representation of all possible inferences

Idea: Represent set of runs as graph

From Programs To Graphs - Starting State

```
fac(X, Y)  :- X > 0, !, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```

$\boxed{fac(X_1, X_2)}$

From Programs To Graphs - Starting State

```
fac(X, Y)  :- X > 0, !, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```

$$\boxed{\mathit{fac}(X_1, X_2)}$$

$$\downarrow$$

$$\boxed{X_1 > 0, !, \mathit{fac}(X_1 - 1, Y_1), X_2 \ \mathit{is}\ Y_1 \cdot X_1 \mid X_1 =:= 0, X_2 \ \mathit{is}\ 1}$$

From Programs To Graphs - Starting State

```
fac(X, Y)  :- X > 0, !, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```

$$\boxed{fac(X_1, X_2)}$$

$$\downarrow$$

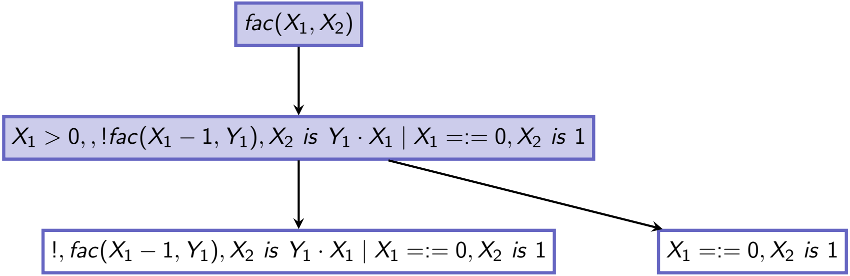
$$\boxed{X_1 > 0, !, fac(X_1 - 1, Y_1), X_2 \text{ } is \text{ } Y_1 \cdot X_1 \mid X_1 =:= 0, X_2 \text{ } is \text{ } 1}$$

$$\downarrow$$

$$\boxed{!, fac(X_1 - 1, Y_1), X_2 \text{ } is \text{ } Y_1 \cdot X_1 \mid X_1 =:= 0, X_2 \text{ } is \text{ } 1}$$

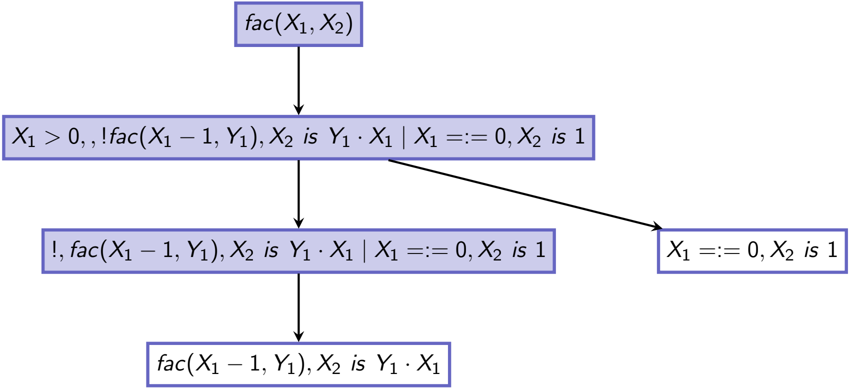
From Programs To Graphs - Starting State

```
fac(X, Y)  :- X > 0, !, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```

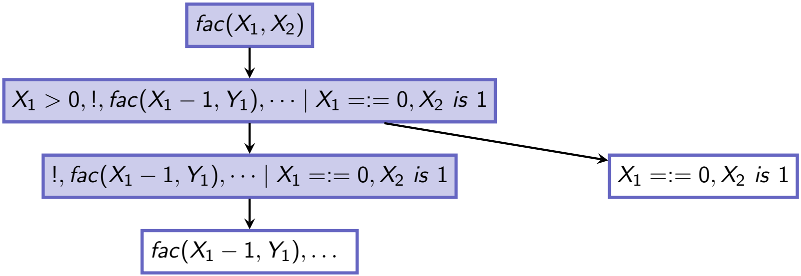


From Programs To Graphs - Starting State

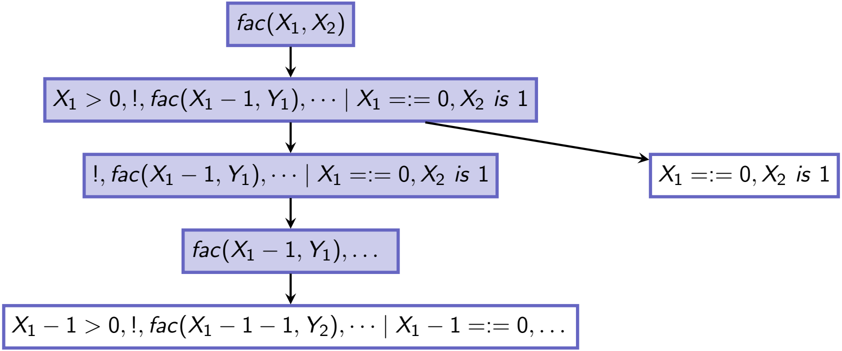
```
fac(X, Y)  :- X > 0, !, fac(X - 1, Y1), Y is Y1 * X.
fac(X, Y)  :- X =:= 0, Y is 1.
```



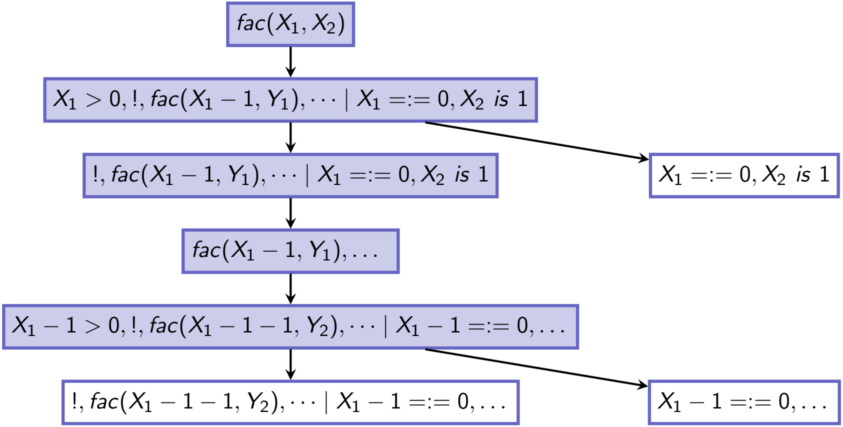
From Programs To Graphs - Nonterminating Construction



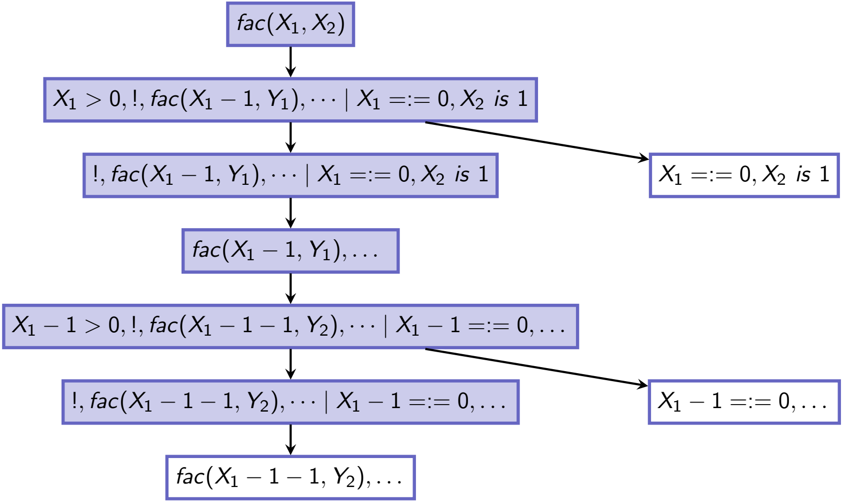
From Programs To Graphs - Nonterminating Construction



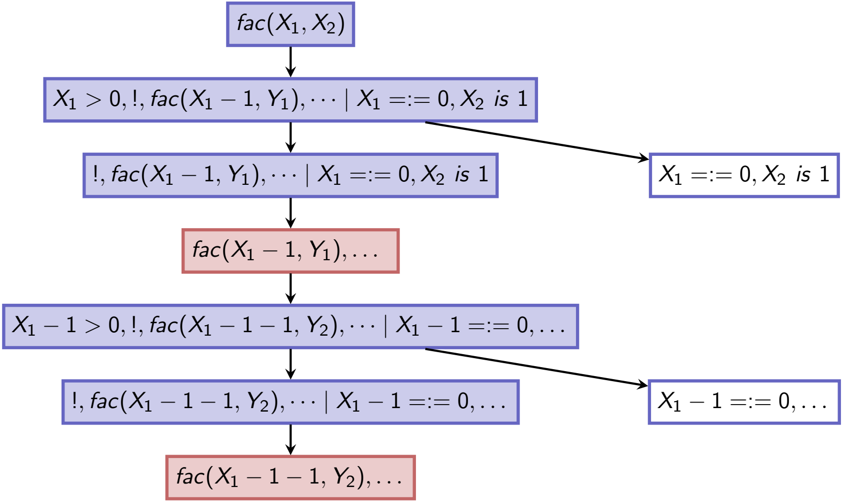
From Programs To Graphs - Nonterminating Construction



From Programs To Graphs - Nonterminating Construction



From Programs To Graphs - Nonterminating Construction



From Programs To Graphs - Split rule

$$\boxed{fac(X_1 - 1, Y_1) \quad , \quad X_2 \ is \ Y_1 \cdot X_1}$$

From Programs To Graphs - Split rule

$$\boxed{\boxed{fac(X_1 - 1, Y_1)}\ ,\quad X_2\ is\ Y_1 \cdot X_1}$$

$$\boxed{fac(X_1 - 1, Y_1)}$$

From Programs To Graphs - Split rule

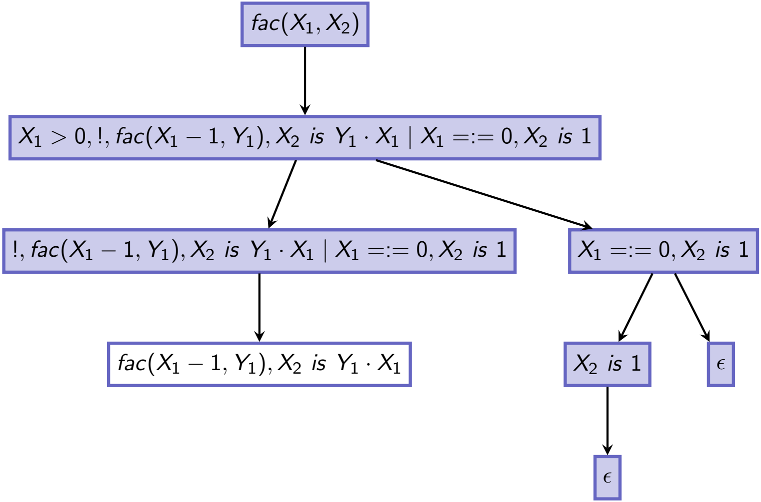
$\boxed{fac(X_1 - 1, Y_1) \;\; , \;\; \boxed{X_2 \; is \; Y_1 \cdot X_1}}$

$\boxed{fac(X_1 - 1, Y_1)}$ $\boxed{X_2 \; is \; Y_1 \cdot X_1}$

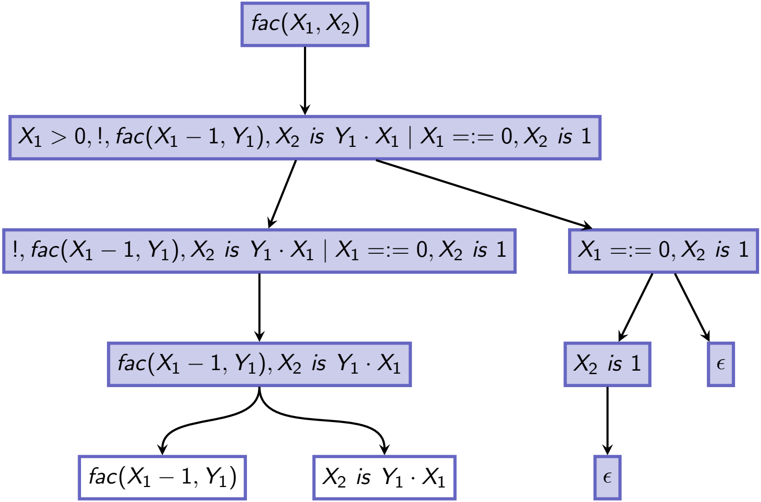
From Programs To Graphs - Split rule

$$\begin{tikzcd} & \boxed{fac(X_1 - 1, Y_1) \quad , \quad X_2 \ is \ Y_1 \cdot X_1} \arrow[dl] \arrow[dr] & \\ \boxed{fac(X_1 - 1, Y_1)} & & \boxed{X_2 \ is \ Y_1 \cdot X_1} \end{tikzcd}$$

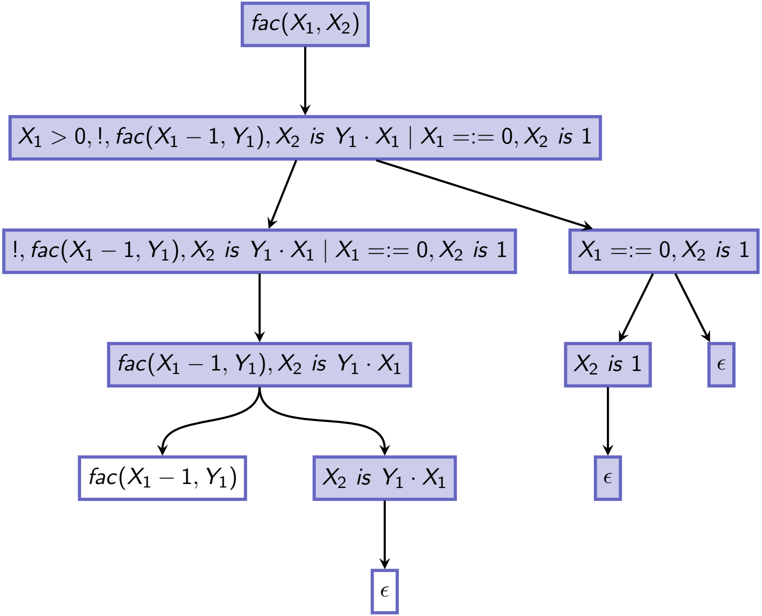
From Programs To Graphs - Split rule



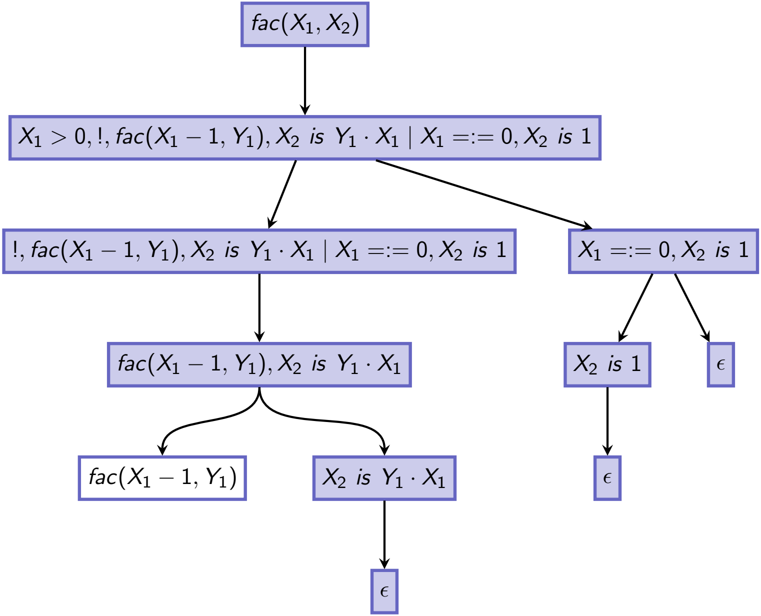
From Programs To Graphs - Split rule



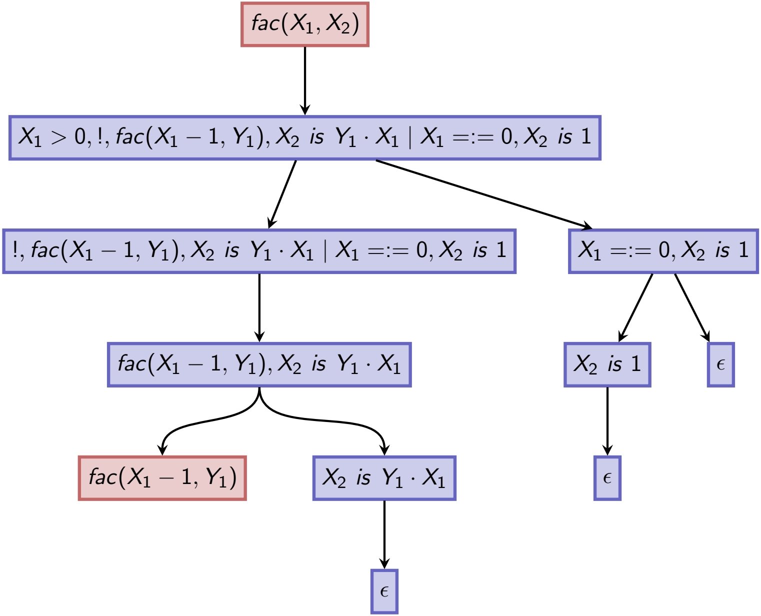
From Programs To Graphs - Split rule



From Programs To Graphs - Split rule



From Programs To Graphs - Split rule



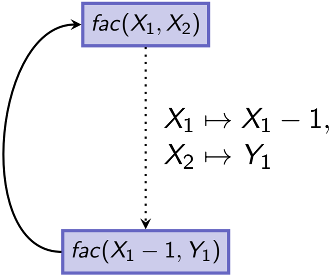
From Programs To Graphs - Instance Rule

$$\begin{tikzcd} \boxed{fac(X_1, X_2)} \arrow[d, dotted] \\ \boxed{fac(X_1 - 1, Y_1)} \end{tikzcd}$$

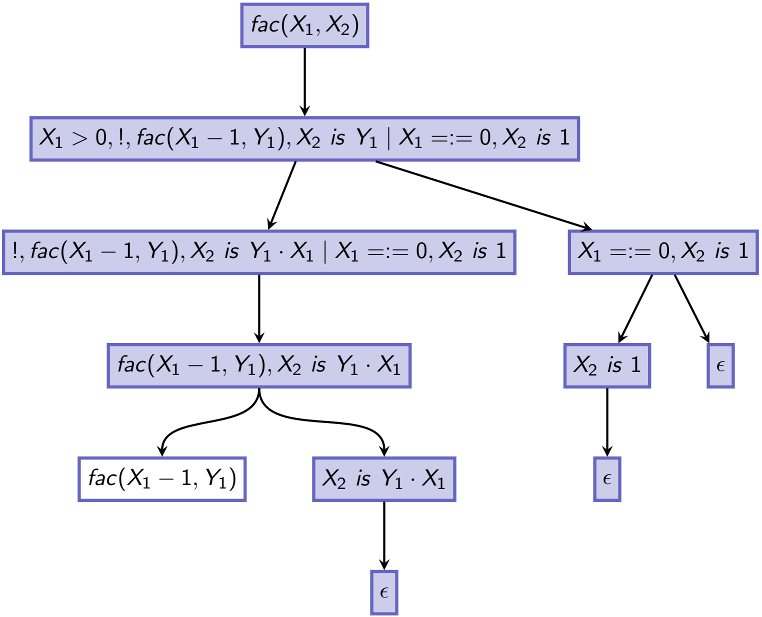
From Programs To Graphs - Instance Rule

$$\begin{tikzcd} \boxed{fac(X_1, X_2)} \arrow[d, dotted, "{\begin{array}{l} X_1 \mapsto X_1 - 1, \\ X_2 \mapsto Y_1 \end{array}}"] \\ \boxed{fac(X_1 - 1, Y_1)} \end{tikzcd}$$

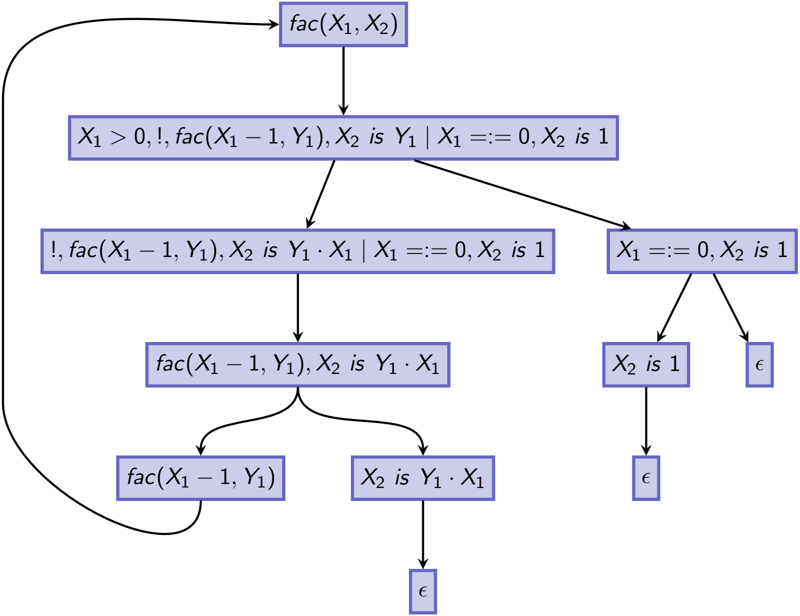
From Programs To Graphs - Instance Rule



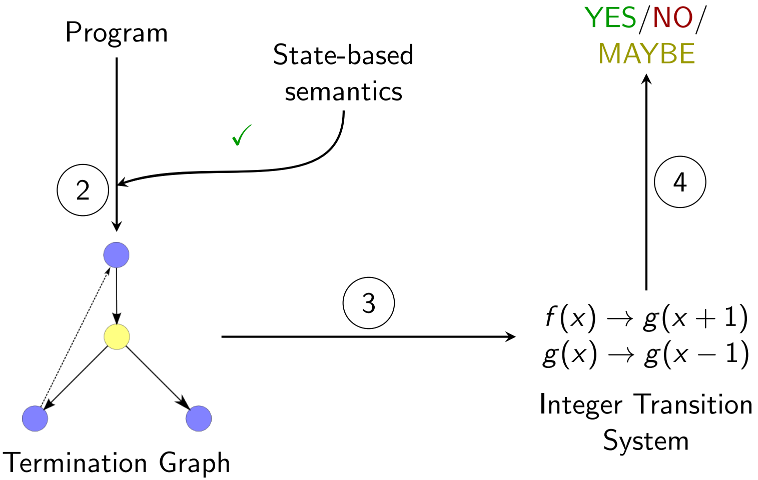
From Programs To Graphs - Final Result



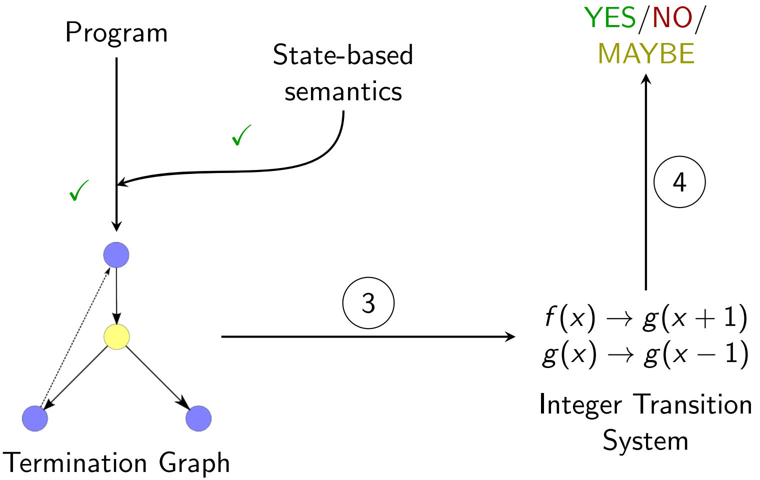
From Programs To Graphs - Final Result



Roadmap


Program
State-based semantics
✓
2
3
4
YES/NO/MAYBE
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Integer Transition System
Termination Graph

Roadmap



YES/NO/
MAYBE
Program
State-based
semantics
✓
✓
4
3
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Termination Graph
Integer Transition
System

Integer Transition Systems

Integer Transition System:

$$f(x) \to f(x+1)$$

Integer Transition Systems

Integer Transition System:

$$f(x) \to f(x+1) \qquad \mid x < 0$$

From Graphs to Transition Systems

Given: Some Termination Graph

Goal: Integer Transition System that terminates if

From Graphs to Transition Systems

Given: Some Termination Graph

Goal: Integer Transition System that terminates if all runs described by the Termination Graph terminate

From Graphs to Transition Systems

Given: Some Termination Graph

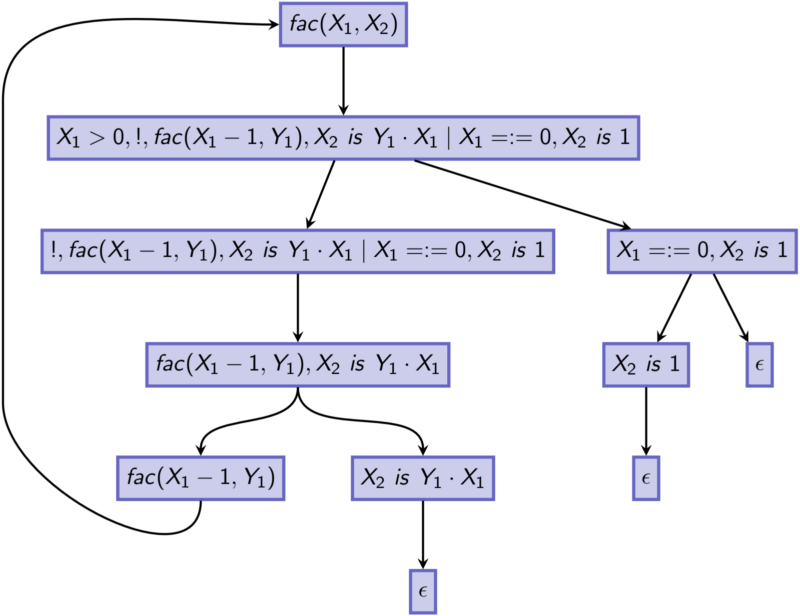
Goal: Integer Transition System that terminates if all runs described by the Termination Graph terminate

Idea: Encode graph locally, node by node

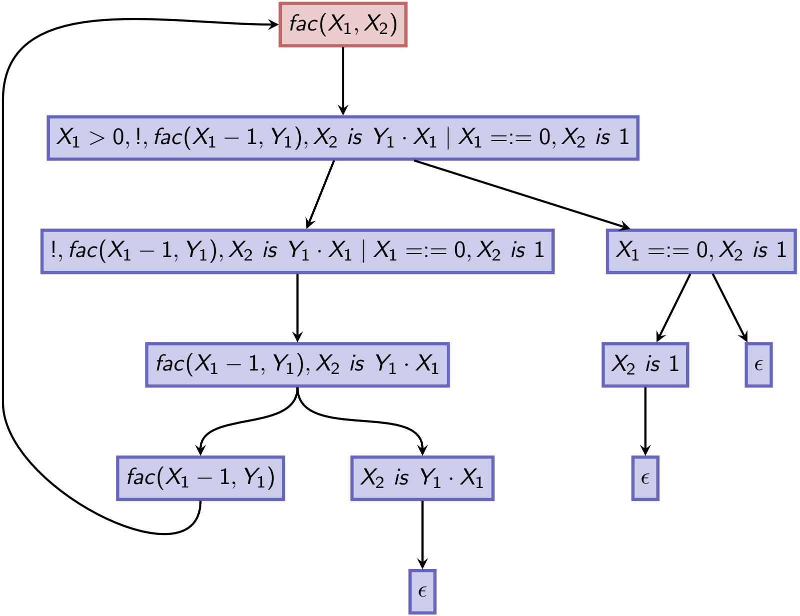
From Graphs to Transition Systems

Paths in Graph $\approx$ Evaluations

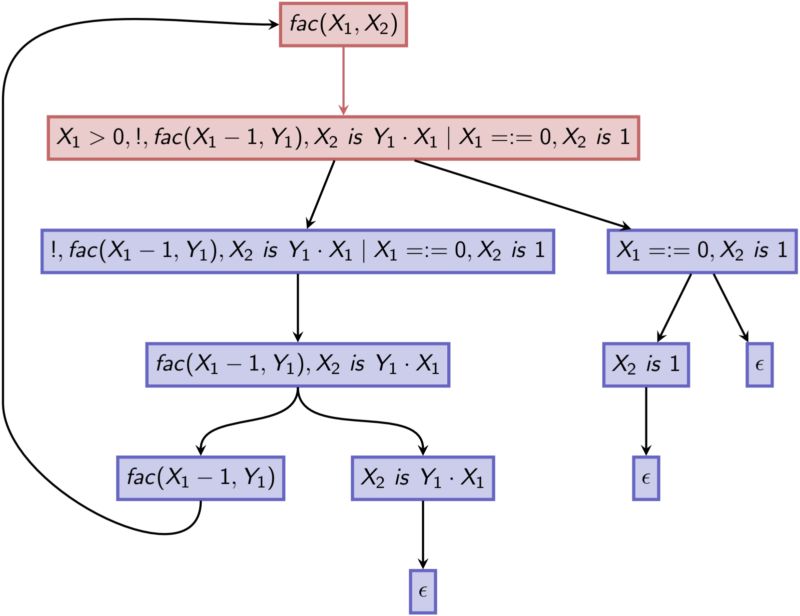
From Graphs to Transition Systems



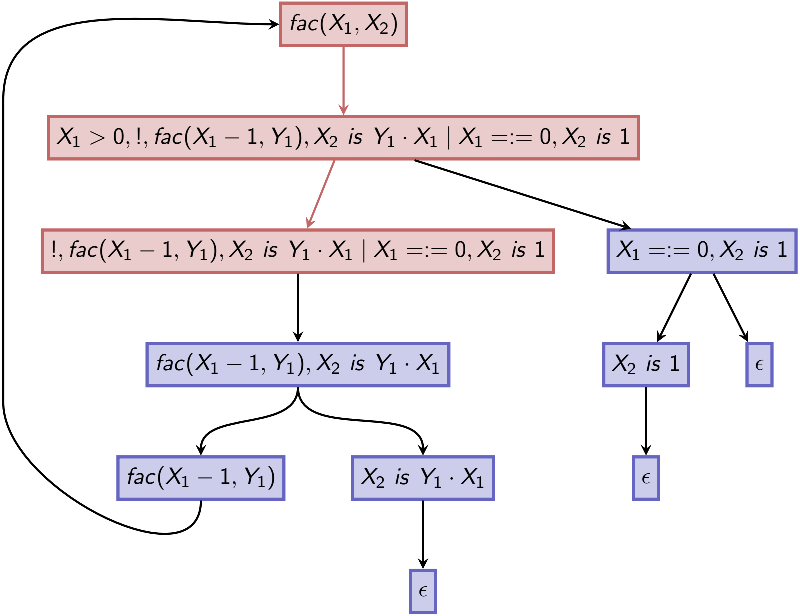
From Graphs to Transition Systems



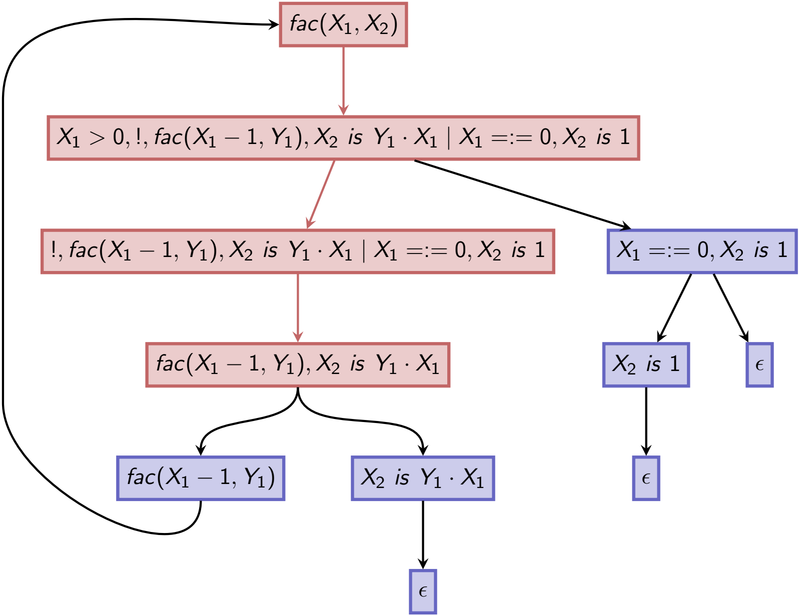
From Graphs to Transition Systems



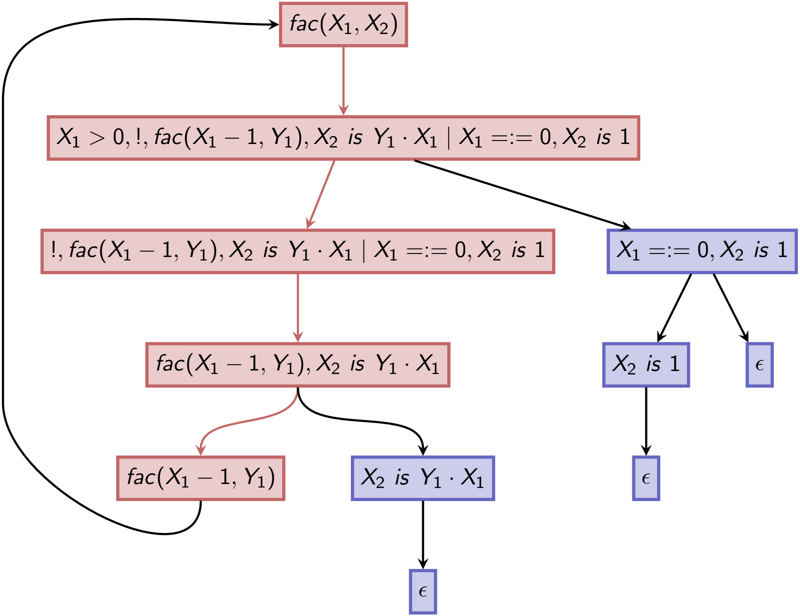
From Graphs to Transition Systems



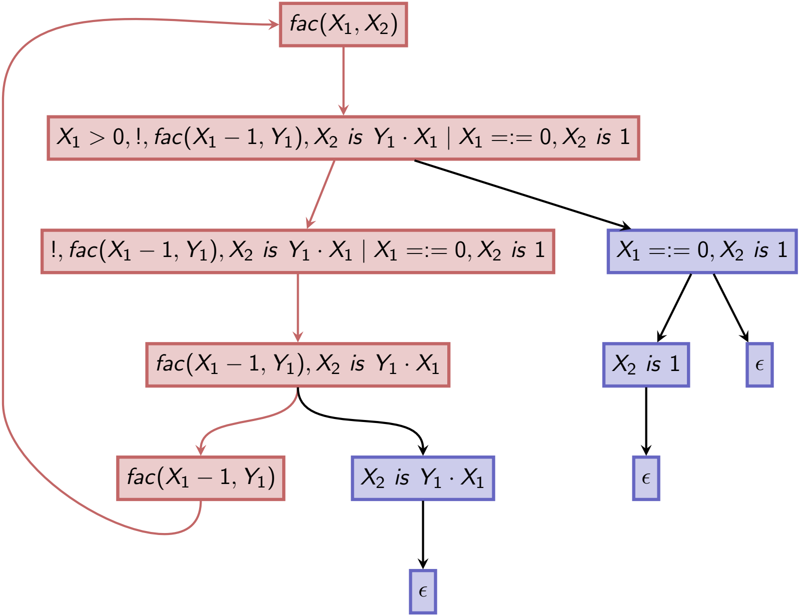
From Graphs to Transition Systems



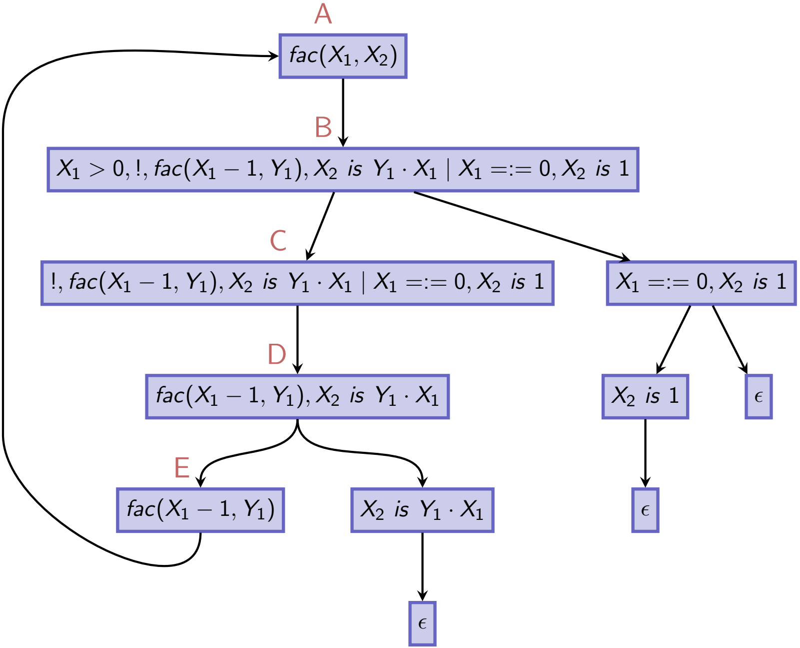
From Graphs to Transition Systems



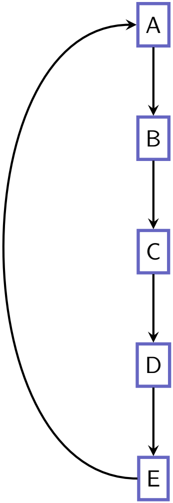
From Graphs to Transition Systems



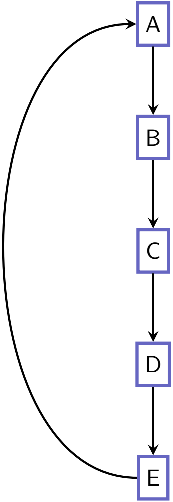
From Graphs to Transition Systems



From Graphs to Transition Systems

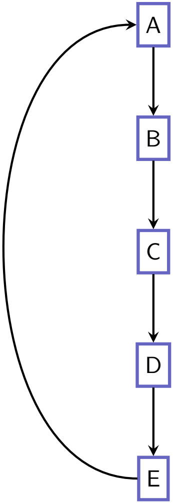


From Graphs to Transition Systems



$$\begin{array}{ccc} A & \to & B \\ B & \to & C \\ C & \to & D \\ D & \to & E \\ E & \to & A \end{array}$$

From Graphs to Transition Systems



$$\begin{array}{ccc} A & \rightarrow & B \\ B & \rightarrow & C \\ C & \rightarrow & D \\ D & \rightarrow & E \\ E & \rightarrow & A \end{array}$$

$$\downarrow$$

$$A \rightarrow A$$

From Graphs to Transition Systems

A

$$\begin{tikzcd} \mathit{fac}(\ X_1\ ,\ X_2\ ) \arrow[d, "B"] \\ X_1\ > 0, !, \mathit{fac}(\ X_1\ - 1,\ Y_1\ ),\ X_2\ \ \mathit{is}\ \ Y_1\ \cdot\ X_1\ \mid\ X_1\ =:= 0,\ X_2\ \ \mathit{is}\ 1 \end{tikzcd}$$

From Graphs to Transition Systems

$$\begin{array}{c} A \\ \boxed{fac(\boxed{X_1}, \boxed{X_2})} \\ \big\downarrow \, B \\ \boxed{\boxed{X_1} > 0, !, fac(\boxed{X_1} - 1, \boxed{Y_1}), \boxed{X_2} \ is \ \boxed{Y_1} \cdot \boxed{X_1} \mid \boxed{X_1} =:= 0, \boxed{X_2} \ is \ 1} \end{array}$$

From Graphs to Transition Systems

A

$$\begin{tikzcd}
fac(\boxed{X_1}, \boxed{X_2}) \arrow[d, "B"] \\
\boxed{X_1} > 0, !, fac(\boxed{X_1} - 1, \boxed{Y_1}), \boxed{X_2} \; is \; \boxed{Y_1} \cdot \boxed{X_1} \mid \boxed{X_1} =:= 0, \boxed{X_2} \; is \; 1
\end{tikzcd}$$

$$A(\qquad) \rightarrow B(\qquad)$$

From Graphs to Transition Systems

A

$$\boxed{fac(\boxed{X_1}, \boxed{X_2})}$$

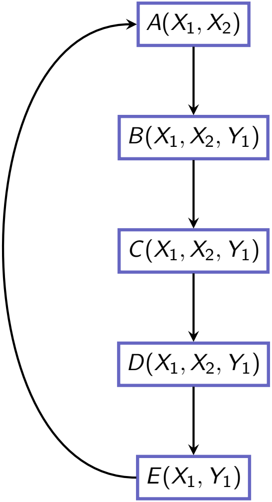
$$\downarrow$$

B

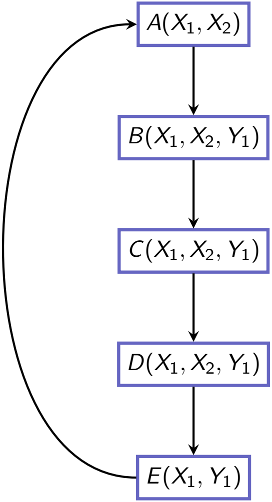
$$\boxed{\boxed{X_1} > 0, !, fac(\boxed{X_1} - 1, \boxed{Y_1}), \boxed{X_2}\ is\ \boxed{Y_1} \cdot \boxed{X_1} \mid \boxed{X_1} =:= 0, \boxed{X_2}\ is\ 1}$$

$$A(X_1, X_2) \quad \rightarrow \quad B(X_1, X_2, Y_1)$$

From Graphs to Transition Systems

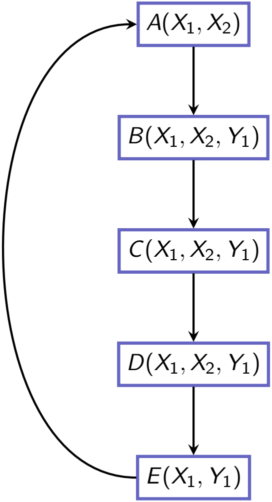


From Graphs to Transition Systems



$$
\begin{array}{lcl}
A(X_1, X_2) & \to & B(X_1, X_2, Y_1) \\
B(X_1, X_2, Y_1) & \to & C(X_1, X_2, Y_1) \\
C(X_1, X_2, Y_1) & \to & D(X_1, X_2, Y_1) \\
D(X_1, X_2, Y_1) & \to & E(X_1, Y_1) \\
E(X_1, Y_1) & \to & A(X_1, X_2)
\end{array}
$$

From Graphs to Transition Systems

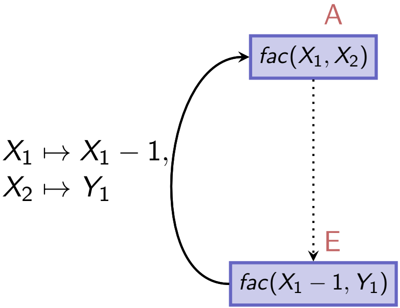


$$\begin{array}{lcl}
A(X_1, X_2) & \to & B(X_1, X_2, Y_1) \\
B(X_1, X_2, Y_1) & \to & C(X_1, X_2, Y_1) \\
C(X_1, X_2, Y_1) & \to & D(X_1, X_2, Y_1) \\
D(X_1, X_2, Y_1) & \to & E(X_1, Y_1) \\
E(X_1, Y_1) & \to & A(X_1, X_2)
\end{array}$$

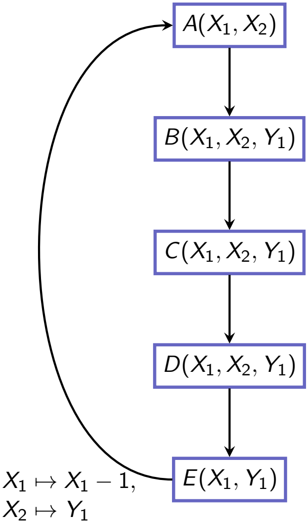
$$\downarrow$$

$$A(X_1, X_2) \quad \to \quad A(X_1, Y_1)$$

From Programs To Graphs - Instance Rule

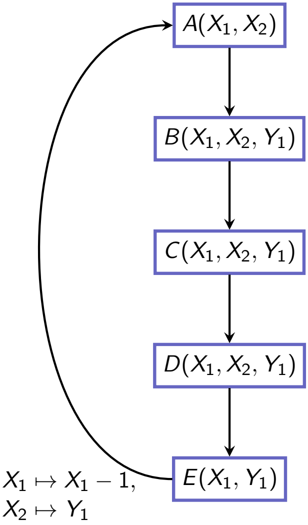


From Graphs to Transition Systems



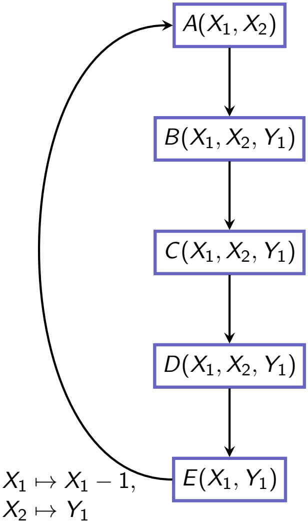
$$
\begin{array}{lcl}
A(X_1, X_2) & \to & B(X_1, X_2, Y_1) \\
B(X_1, X_2, Y_1) & \to & C(X_1, X_2, Y_1) \\
C(X_1, X_2, Y_1) & \to & D(X_1, X_2, Y_1) \\
D(X_1, X_2, Y_1) & \to & E(X_1, Y_1) \\
E(X_1, Y_1) & \to & A(\qquad\qquad)
\end{array}
$$

From Graphs to Transition Systems



$$
\begin{array}{lcl}
A(X_1, X_2) & \rightarrow & B(X_1, X_2, Y_1) \\
B(X_1, X_2, Y_1) & \rightarrow & C(X_1, X_2, Y_1) \\
C(X_1, X_2, Y_1) & \rightarrow & D(X_1, X_2, Y_1) \\
D(X_1, X_2, Y_1) & \rightarrow & E(X_1, Y_1) \\
E(X_1, Y_1) & \rightarrow & A(X_1 - 1, Y_1)
\end{array}
$$

From Graphs to Transition Systems

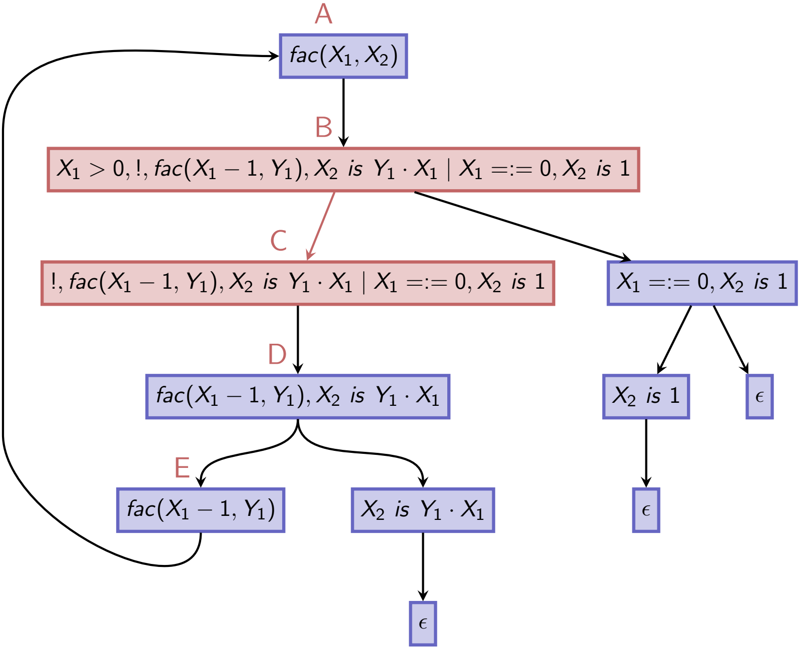


$$
\begin{array}{lcl}
A(X_1, X_2) & \to & B(X_1, X_2, Y_1) \\
B(X_1, X_2, Y_1) & \to & C(X_1, X_2, Y_1) \\
C(X_1, X_2, Y_1) & \to & D(X_1, X_2, Y_1) \\
D(X_1, X_2, Y_1) & \to & E(X_1, Y_1) \\
E(X_1, Y_1) & \to & A(X_1 - 1, Y_1)
\end{array}
$$

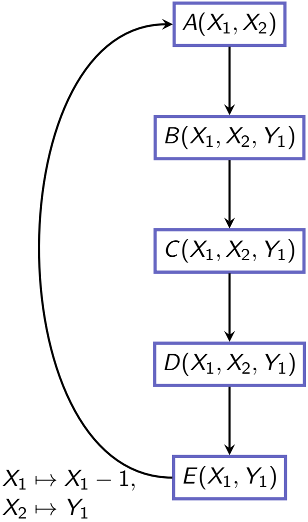
$$\downarrow$$

$$A(X_1, X_2) \to A(X_1 - 1, Y_1)$$

From Graphs to Transition Systems

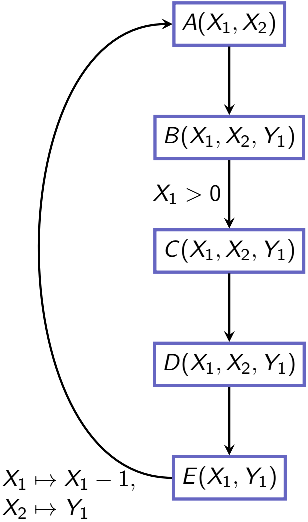


From Graphs to Transition Systems



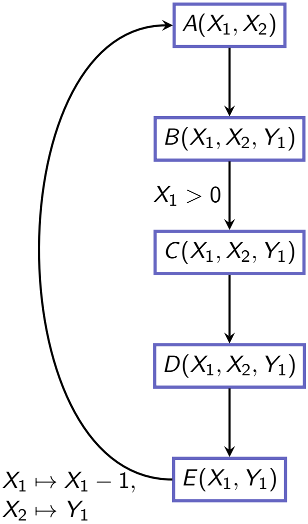
$$
\begin{aligned}
A(X_1, X_2) &\quad\rightarrow\quad B(X_1, X_2, Y_1)\\
B(X_1, X_2, Y_1) &\quad\rightarrow\quad C(X_1, X_2, Y_1)\\
C(X_1, X_2, Y_1) &\quad\rightarrow\quad D(X_1, X_2, Y_1)\\
D(X_1, X_2, Y_1) &\quad\rightarrow\quad E(X_1, Y_1)\\
E(X_1, Y_1) &\quad\rightarrow\quad A(X_1 - 1, Y_1)
\end{aligned}
$$

From Graphs to Transition Systems



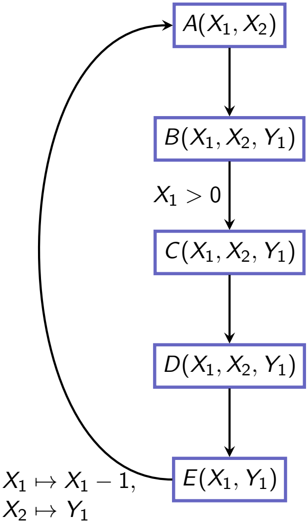
$$
\begin{array}{lcl}
A(X_1, X_2) & \rightarrow & B(X_1, X_2, Y_1) \\
B(X_1, X_2, Y_1) & \rightarrow & C(X_1, X_2, Y_1) \\
C(X_1, X_2, Y_1) & \rightarrow & D(X_1, X_2, Y_1) \\
D(X_1, X_2, Y_1) & \rightarrow & E(X_1, Y_1) \\
E(X_1, Y_1) & \rightarrow & A(X_1 - 1, Y_1)
\end{array}
$$

From Graphs to Transition Systems



$$
\begin{aligned}
A(X_1, X_2) &\quad\to\quad B(X_1, X_2, Y_1) \\
B(X_1, X_2, Y_1) &\quad\to\quad C(X_1, X_2, Y_1) \quad \mid X_1 > 0 \\
C(X_1, X_2, Y_1) &\quad\to\quad D(X_1, X_2, Y_1) \\
D(X_1, X_2, Y_1) &\quad\to\quad E(X_1, Y_1) \\
E(X_1, Y_1) &\quad\to\quad A(X_1 - 1, Y_1)
\end{aligned}
$$

From Graphs to Transition Systems

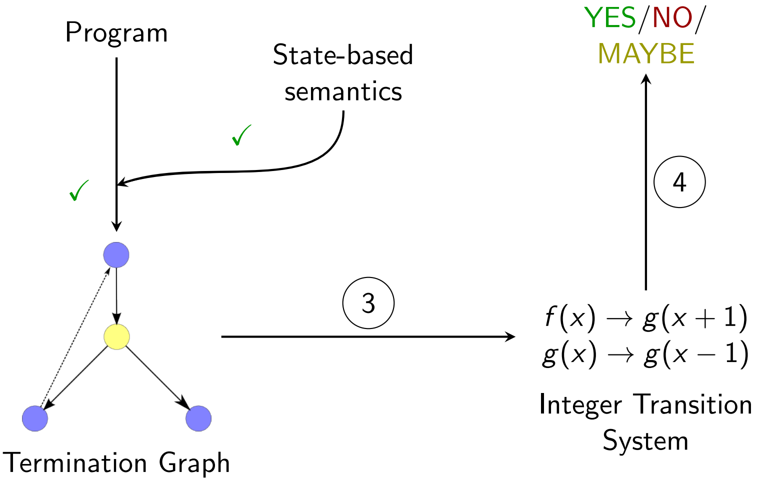


$$
\begin{array}{lcll}
A(X_1, X_2) & \to & B(X_1, X_2, Y_1) & \\
B(X_1, X_2, Y_1) & \to & C(X_1, X_2, Y_1) & \mid X_1 > 0 \\
C(X_1, X_2, Y_1) & \to & D(X_1, X_2, Y_1) & \\
D(X_1, X_2, Y_1) & \to & E(X_1, Y_1) & \\
E(X_1, Y_1) & \to & A(X_1 - 1, Y_1) &
\end{array}
$$

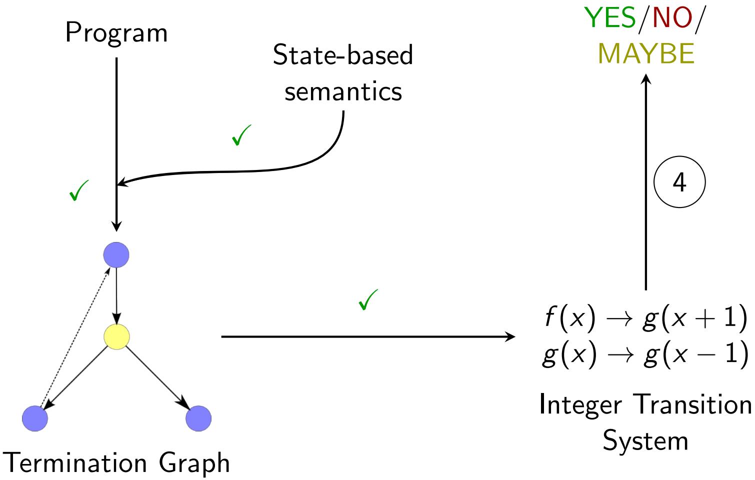
$$\downarrow$$

$$A(X) \quad \to \quad A(X - 1) \quad \mid X > 0$$

Roadmap


Program
State-based semantics
YES/NO/MAYBE
4
3
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Integer Transition System
Termination Graph

Roadmap


Program
State-based
semantics
✓
✓
YES/NO/
MAYBE
4
✓
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Integer Transition
System
Termination Graph

Termination of Integer Transition Systems

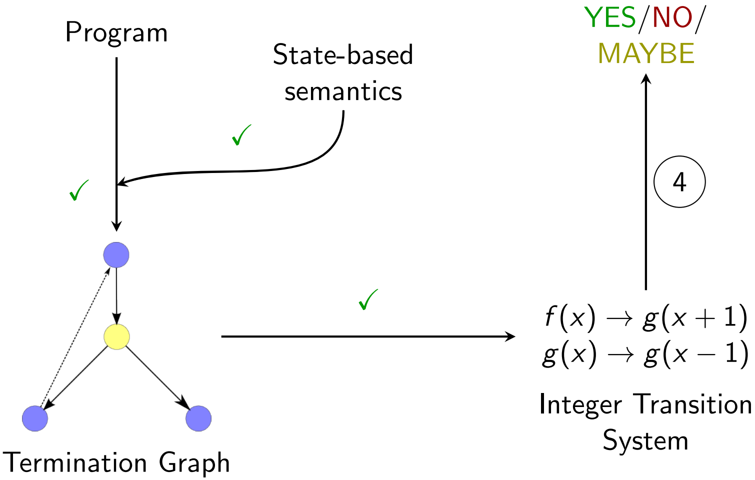
Well-studied problem

Termination of Integer Transition Systems

Well-studied problem

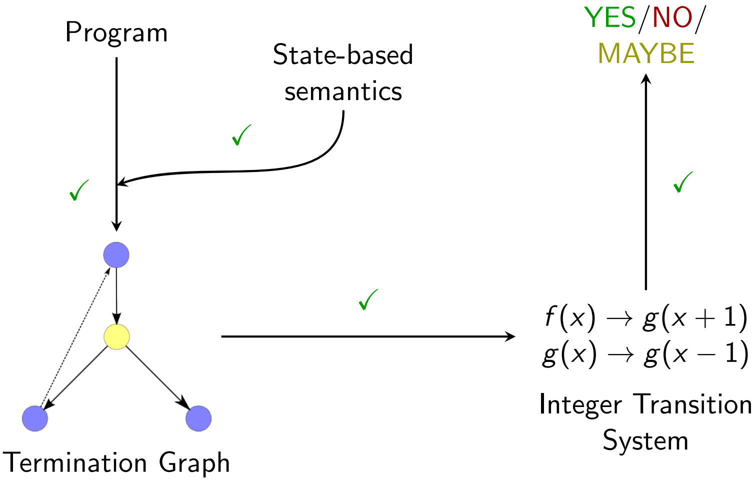
Use known techniques to show termination
(Transition Invariants, Podelski and Rybalchenko, 2004)

Roadmap

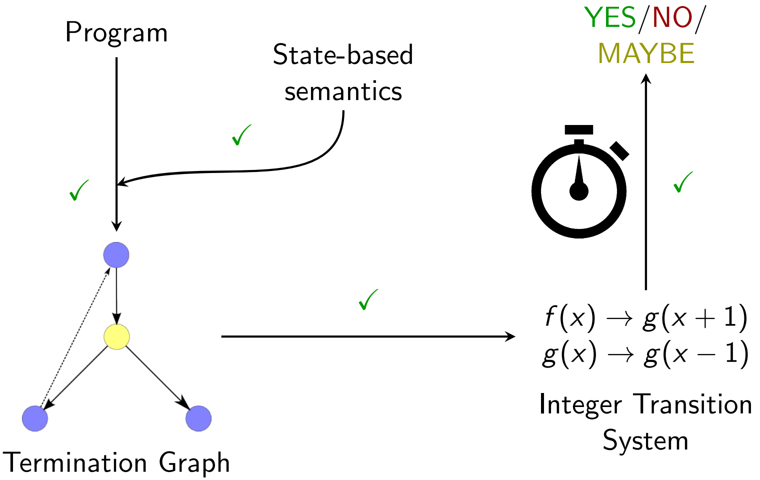


YES/NO/
MAYBE
Program
State-based
semantics
✓
✓
4
✓
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Integer Transition
System
Termination Graph

Roadmap


YES/NO/
MAYBE
Program
State-based
semantics
✓
✓
✓
✓
$f(x) \to g(x+1)$
$g(x) \to g(x-1)$
Integer Transition
System
Termination Graph

Roadmap



Stopwatch: `http://icons8.com/`

Results - Compared Approaches

Term Rewriting with Argument Filter Directly from Program

(Termination Proofs, Schneider-Kamp et al., 2009)

Results - Compared Approaches

Term Rewriting with Argument Filter Directly from Program

(Termination Proofs, Schneider-Kamp et al., 2009)

Dependency Triples Graph Construction

(Dependency Triples, Ströder et al., 2011)

Results - Compared Approaches

Term Rewriting with Argument Filter Directly from Program

(Termination Proofs, Schneider-Kamp et al., 2009)

Dependency Triples Graph Construction

(Dependency Triples, Ströder et al., 2011)

Term Rewriting Graph Construction

(Symbolic Evaluation Graphs, Giesl et al., 2012)

Results - Compared Approaches

Term Rewriting with Argument Filter Directly from Program
(Termination Proofs, Schneider-Kamp et al., 2009)

Dependency Triples Graph Construction
(Dependency Triples, Ströder et al., 2011)

Term Rewriting Graph Construction
(Symbolic Evaluation Graphs, Giesl et al., 2012)

Integer Transition Systems This approach
(Analysis of Arithmetic Prolog Programs, Weinert, 2015)

Results - Compared Approaches

Term Rewriting with Argument Filter Directly from Program
(Termination Proofs, Schneider-Kamp et al., 2009)

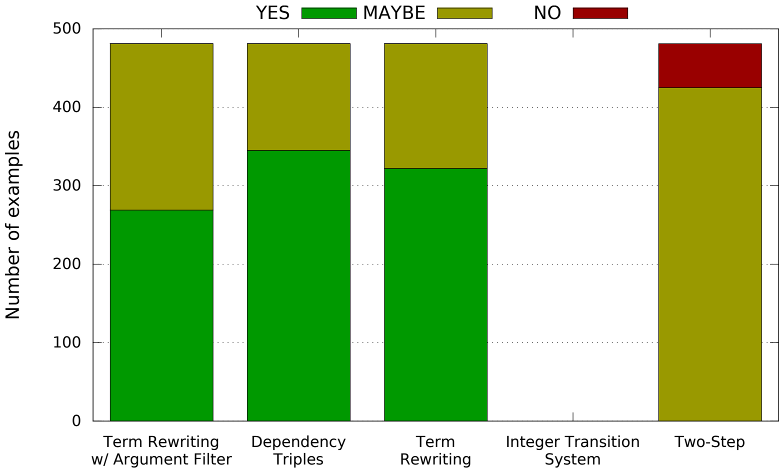
Dependency Triples Graph Construction
(Dependency Triples, Ströder et al., 2011)

Term Rewriting Graph Construction
(Symbolic Evaluation Graphs, Giesl et al., 2012)

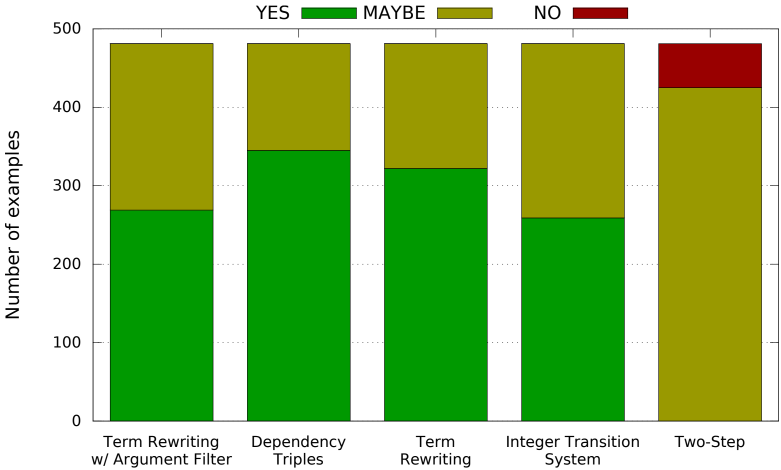
Integer Transition Systems This approach
(Analysis of Arithmetic Prolog Programs, Weinert, 2015)

Two-Step Constraint Satisfaction Problem
(Non-termination Analysis, Voets et al., 2011)

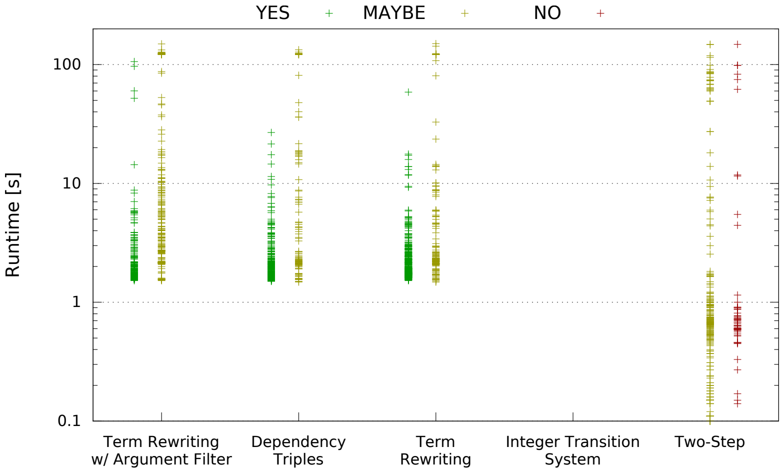
Results - Logic Benchmarks - Power



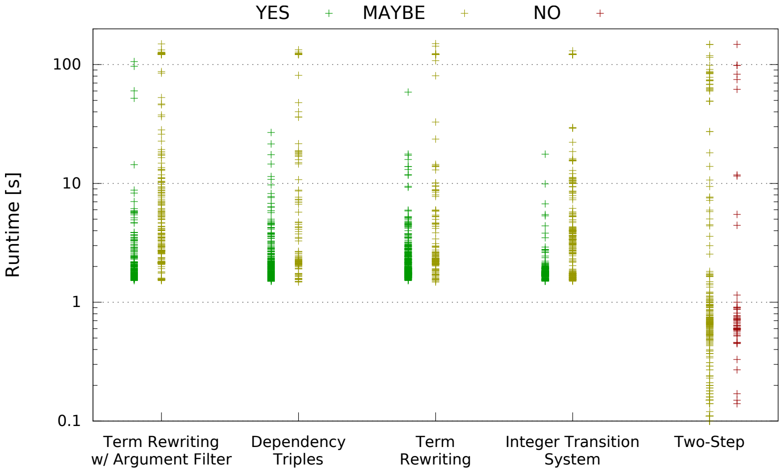
Results - Logic Benchmarks - Power



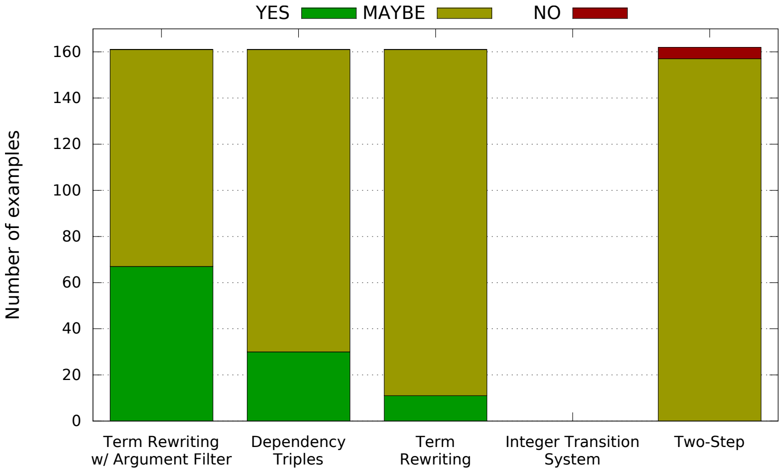
Results - Logic Benchmarks - Runtime



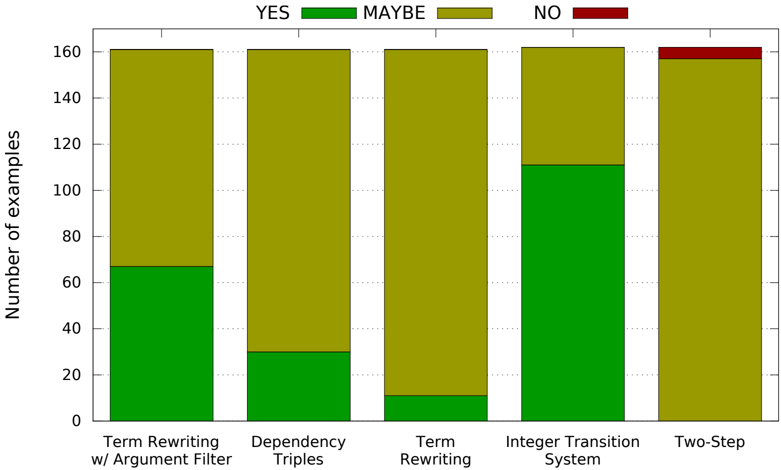
Results - Logic Benchmarks - Runtime



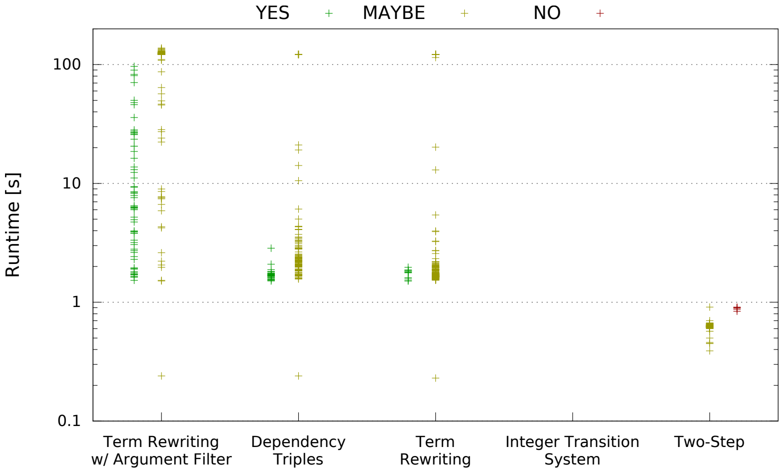
Results - Numerical Benchmarks - Power



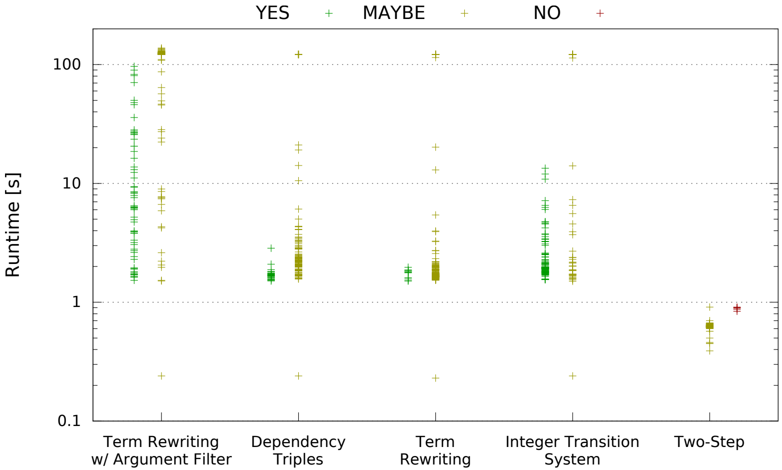
Results - Numerical Benchmarks - Power



Results - Numerical Benchmarks - Runtime



Results - Numerical Benchmarks - Runtime



Contributions - Theoretical

- Extended construction of Termination Graphs, taking arithmetic comparisons and evaluations into account
- Developed new construction of Integer Transition System from Termination Graphs

Contributions - Theoretical

- Extended construction of Termination Graphs, taking arithmetic comparisons and evaluations into account
- Developed new construction of Integer Transition System from Termination Graphs
- Extended abstract state to store arithmetic knowledge

Contributions - Theoretical

- Extended construction of Termination Graphs, taking arithmetic comparisons and evaluations into account
- Developed new construction of Integer Transition System from Termination Graphs
- Extended abstract state to store arithmetic knowledge
- Separated abstract semantics and termination analysis

Contributions - Theoretical

- Extended construction of Termination Graphs, taking arithmetic comparisons and evaluations into account
- Developed new construction of Integer Transition System from Termination Graphs
- Extended abstract state to store arithmetic knowledge
- Separated abstract semantics and termination analysis
- Proved soundness of all steps of the construction

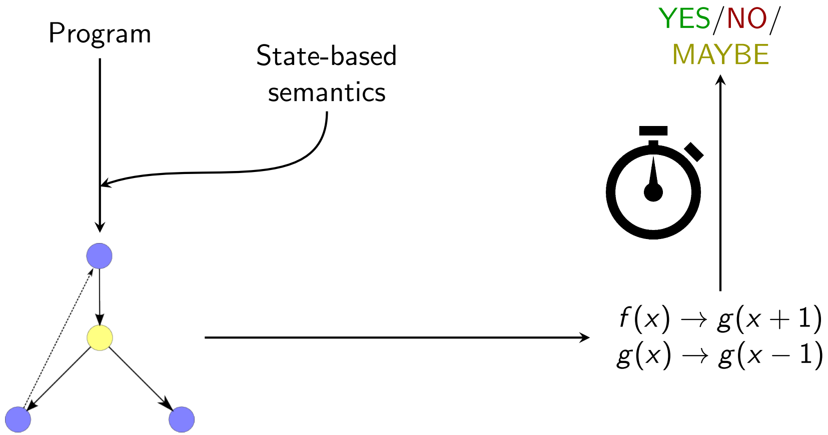
Contributions - Practical

- Implemented extension of construction of Termination Graphs, optimization through SMT solver
- Implemented construction of Integer Transition Systems from Termination Graphs

Contributions - Practical

- Implemented extension of construction of Termination Graphs, optimization through SMT solver
- Implemented construction of Integer Transition Systems from Termination Graphs
- Added 162 numerical benchmarks to benchmark suite
- Performed experiments comparing this approach to existing ones

`http://alexanderweinert.net/talks`
`alexander.weinert@rwth-aachen.de`



Stopwatch: `http://icons8.com/`