

VERIFICATION OF ONE INTEGER PARAMETER
RECURSIVE SEQUENTIAL PROCEDURES

Ahmed BOUAJJANI

LIAFA - University of Paris 7

joint work with

Peter Habermehl and Richard Mayr

Verification of Boolean Recursive Procedures

Boolean Recursive Procedures \longrightarrow Context-Free Processes

Interprocedural data flow analysis and verification problems (safety properties) of recursive programs can be formulated as

reachability analysis problems for context-free (or pushdown) processes:

\implies Computing sets of successors / predecessors of given sets of configurations.

e.g., [Steffen and al., 96], [Esparza and Knop, 99]

Verification of Boolean Recursive Procedures

Boolean Recursive Procedures \longrightarrow Context-Free Processes

Interprocedural data flow analysis and verification problems (safety properties) of recursive programs can be formulated as

reachability analysis problems for context-free (or pushdown) processes:

\implies Computing sets of successors / predecessors of given sets of configurations.

e.g., [Steffen and al., 96], [Esparza and Knop, 99]

Symbolic Reachability Analysis of Context-Free Processes

Algorithms for symbolic reachability analysis and model-checking of pushdown systems

- Sets of stack configurations are represented by means of finite-state automata.
- Polynomial constructions of the post^* and pre^* images of given regular sets of configurations.
e.g., [Bouajjani, Esparza, Maler, 97], [Finkel, Willems, Wolper, 97], [Esparza, Schwoon, 01]
- Efficient tools have been developed based on these techniques (e.g., Edinburgh, Microsoft).

Recursive Procedures with Integer Parameters

Example: Fibonacci function

$$F(v) = \text{if } n \leq 1 \text{ then return } 1 \\ \text{else return } F(v-1) + F(v-2)$$

Reachable configurations (stack contents) from $F(5)$:

$$\begin{array}{c} F(5) \\ F(4)F(3) \\ F(3)F(2)F(3) \\ F(2)F(1)F(2)F(3) \\ F(1)F(0)F(1)F(2)F(3) \\ F(0)F(1)F(2)F(3) \\ F(1)F(2)F(3) \\ F(2)F(3) \\ F(1)F(0)F(3) \\ F(0)F(3) \\ F(3) \\ F(2)F(1) \\ F(1)F(0)F(1) \\ F(0)F(1) \\ F(1) \\ \epsilon \end{array}$$

Parametrized Context-Free Processes

Integer Symbol Sequences (ISS)

Finite sequences of the form:

$$X_1(k_1)X_2(k_2)\dots X_n(k_n)$$

where $X_i \in \Gamma$ and $k_i \in \mathbb{Z}$

BPA(\mathbb{Z})

- Set Δ of **rewriting rules** of the form:

$$X(v) \rightarrow X_1(e_1)X_2(e_2)\dots X_n(e_n), \quad P(v)$$

where

- e_i is either k_i or $v + k_i$ ($k_i \in \mathbb{Z}$),
 - $P(v)$ is a Presburger predicate.
- **Prefix rewriting:** Defines a transition relation \Longrightarrow_{Δ} on ISS.
 - $\text{post}_{\Delta}^*(C) = \{\alpha \mid \exists \beta \in C. \beta \xrightarrow{*}_{\Delta} \alpha\}$, $\text{pre}_{\Delta}^*(C) = \{\alpha \mid \exists \beta \in C. \alpha \xrightarrow{*}_{\Delta} \beta\}$.

Example

BPA(\mathbb{Z}) system for the Fibonacci function:

$$\begin{array}{lll} F(v) & \rightarrow & \epsilon & v \leq 1 \\ F(v) & \rightarrow & F(v-1)F(v-2) & v > 1 \end{array}$$

Post*({ $F(k) \mid k \geq 0$ }):

$$\begin{array}{l} F(k) \\ F(k-1)F(k-2) \\ F(k-2)F(k-3)F(k-2) \\ F(k-3)F(k-4)F(k-3)F(k-2) \\ F(k-4)F(k-5)F(k-4)F(k-3)F(k-2) \\ \dots \\ F(k-3)F(k-2) \\ F(k-4)F(k-5)F(k-2) \\ \dots \\ F(k-5)F(k-2) \\ F(k-6)F(k-7)F(k-2) \\ F(k-7)F(k-8)F(k-7)F(k-2) \\ \dots \end{array}$$

\mathbb{Z} -input 1-Counter Automata

- **Input** = Integer Symbol Sequence
- **Equality tests** between the integer input and the counter value

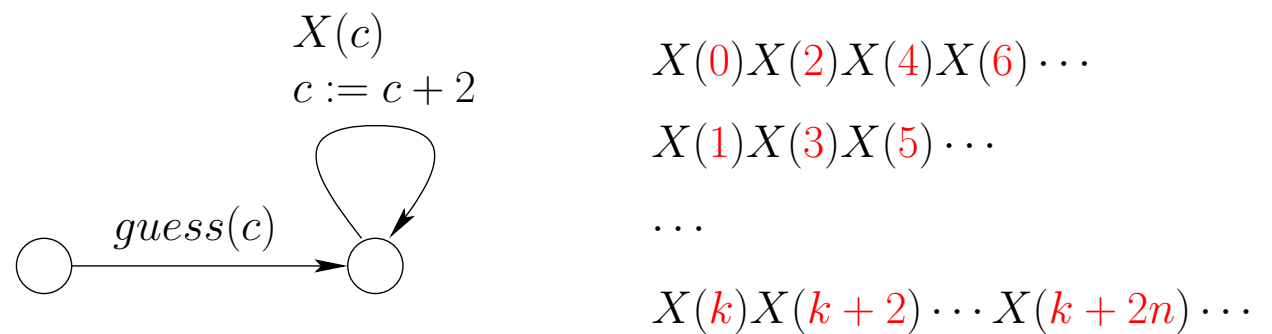


Figure 1: Example

Recognizing Fibonacci Configurations

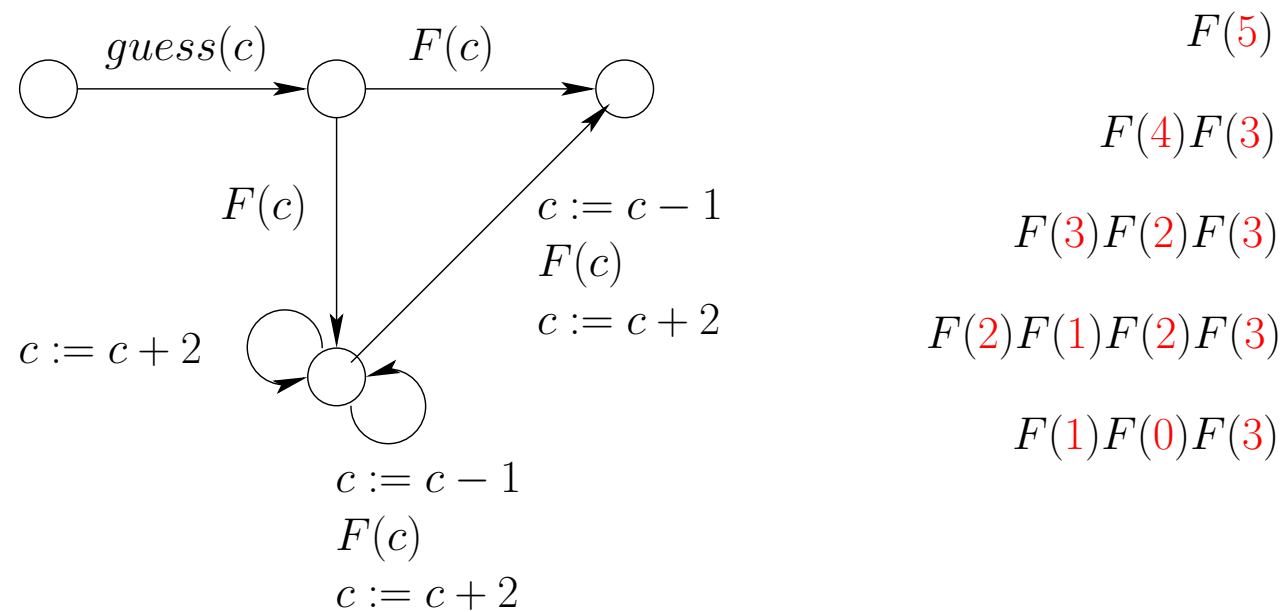


Figure 2: $Post^*(\{F(k) \mid k \geq 0\})$

Main Results (1)

Forward Reachability Analysis

Let Δ be a BPA(\mathbb{Z}) system, and let \mathcal{A} be a \mathbb{Z} -input 1-counter automaton.

Then, a \mathbb{Z} -input 1-counter automaton \mathcal{A}' with $L(\mathcal{A}') = \text{post}_{\Delta}^*(L(\mathcal{A}))$ can be **effectively constructed**.

Main Results (1)

Forward Reachability Analysis

Let Δ be a BPA(\mathbb{Z}) system, and let \mathcal{A} be a \mathbb{Z} -input 1-counter automaton.

Then, a \mathbb{Z} -input 1-counter automaton \mathcal{A}' with $L(\mathcal{A}') = \text{post}_{\Delta}^*(L(\mathcal{A}))$ can be **effectively constructed**.

Backward Reachability Analysis

- The **membership problem** (of an ISS) in $\text{pre}_{\Delta}^*(L(\mathcal{A}))$, where \mathcal{A} is a \mathbb{Z} -input 1-counter automaton, is **undecidable**.

Main Results (1)

Forward Reachability Analysis

Let Δ be a BPA(\mathbb{Z}) system, and let \mathcal{A} be a \mathbb{Z} -input 1-counter automaton.

Then, a \mathbb{Z} -input 1-counter automaton \mathcal{A}' with $L(\mathcal{A}') = \text{post}_{\Delta}^*(L(\mathcal{A}))$ can be **effectively constructed**.

Backward Reachability Analysis

- The **membership problem** (of an ISS) in $\text{pre}_{\Delta}^*(L(\mathcal{A}))$, where \mathcal{A} is a \mathbb{Z} -input 1-counter automaton, is **undecidable**.
- The set $\text{pre}_{\Delta}^*(L(\mathcal{A}))$, where \mathcal{A} is a \mathbb{Z} -input 1-counter automaton, is **not recognizable** by \mathbb{Z} -input 1-counter automata.

Main Results (1)

Forward Reachability Analysis

Let Δ be a BPA(\mathbb{Z}) system, and let \mathcal{A} be a \mathbb{Z} -input 1-counter automaton.

Then, a \mathbb{Z} -input 1-counter automaton \mathcal{A}' with $L(\mathcal{A}') = \text{post}_\Delta^*(L(\mathcal{A}))$ can be **effectively constructed**.

Backward Reachability Analysis

- The **membership problem** (of an ISS) in $\text{pre}_\Delta^*(L(\mathcal{A}))$, where \mathcal{A} is a \mathbb{Z} -input 1-counter automaton, is **undecidable**.
- The set $\text{pre}_\Delta^*(L(\mathcal{A}))$, where \mathcal{A} is a \mathbb{Z} -input 1-counter automaton, is **not recognizable** by \mathbb{Z} -input 1-counter automata.
- Let Δ be a BPA(\mathbb{Z}) system, and let \mathcal{R} be a finite-state automaton.

Then, a \mathbb{Z} -input 1-counter automaton \mathcal{A} with $L(\mathcal{A}) = \text{pre}_\Delta^*(L(\mathcal{R})\uparrow)$ can be **effectively constructed**.

where, for any regular language L over Γ ,

$$L\uparrow = \{X_1(k_1)X_2(k_2)\cdots X_n(k_n) \mid X_1X_2\cdots X_n \in L, \text{ and } k_1, \dots, k_n \in \mathbb{Z}\}$$

Configuration Properties

Pattern Constraints

$$\varphi = \langle A_1, \dots, A_n, P \rangle$$

where A_1, \dots, A_n are finite automata over Γ , and P is an n -ary Presburger predicate.

Semantics

Let w be an ISS. Then, $w \models \langle A_1, \dots, A_n, P \rangle$ iff

$\exists w_1, \dots, w_n \in \text{ISS}, \exists X_1, \dots, X_n \in \Gamma, \exists k_1, \dots, k_n \in \mathbb{Z}$, such that

$$w = w_1 \cdot X_1(k_1) \cdot w_2 \cdot X_2(k_2) \cdots w_n \cdot X_n(k_n)$$

and

- $\forall i \in \{1, \dots, n\}, w_i|_{\Gamma} \cdot X_i \in L(A_i)$,
- $P(k_1, \dots, k_n)$ is true.

Reachability/Safety Properties

Decide whether

$$w \models \text{EF } \varphi$$

i.e., $\exists w'. w' \in \text{post}_{\Delta}^*(w)$ and $w' \models \varphi$.

Reachability/Safety Properties

Decide whether

$$w \models \text{EF } \varphi$$

i.e., $\exists w'. w' \in \text{post}_{\Delta}^*(w)$ and $w' \models \varphi$.

Examples

- Can the procedure X be called with some parameter **greater than 5** ?

$$\text{EF}\langle X, \Gamma^*, v_1 \geq 5 \rangle$$

- Can the execution stack contain two instances of the procedures X with **same parameter** ?

$$\text{EF}\langle \Gamma^* X, \Gamma^* X, \Gamma^*, v_1 = v_2 \rangle$$

- The stack **always** contains an **increasing sequences** of X -parameters

$$\neg \text{EF}\langle \Gamma^* X, \Gamma^* X, \Gamma^*, v_1 \geq v_2 \rangle$$

Main Results (2)

Pattern Constraints Reachability Properties

Theorem

The problem $w \models \text{EF } \varphi$ is **decidable**.

Main Results (2)

Pattern Constraints Reachability Properties

Theorem

The problem $w \models \text{EF } \varphi$ is **decidable**.

Reachable Parameter n -vectors

What is the set of all possible parameter values for which X can be called ?

Main Results (2)

Pattern Constraints Reachability Properties

Theorem

The problem $w \models \text{EF } \varphi$ is **decidable**.

Reachable Parameter n -vectors

What is the set of all possible parameter values for which X can be called ?

$$\{k \mid X(k) \cdot w' \in \text{post}_{\Delta}^*(w)\}$$

Main Results (2)

Pattern Constraints Reachability Properties

Theorem

The problem $w \models \text{EF } \varphi$ is **decidable**.

Reachable Parameter n -vectors

What is the set of all possible parameter values for which X can be called ?

$$\{k \mid X(k) \cdot w' \in \text{post}_{\Delta}^*(w)\}$$

Theorem

Let Δ be a BPA(\mathbb{Z}) system, let w be an initial configuration (ISS), and let φ be a pattern constraint.

Then, the set

$$\{(k_1, \dots, k_n) \in \mathbb{Z}^n \mid \exists w' = w_1 \cdot X_1(k_1) \cdot w_2 \cdot X_2(k_2) \cdots w_n \cdot X_n(k_n) \in \text{post}_{\Delta}^*(w). w' \models \varphi\}$$

is **semilinear** and **effectively constructible**.

Outline

- \mathbb{Z} -input 1-Counter Automata,
- Construction of the post^* image,
- Reachability properties,
- Conclusion.

\mathbb{Z} -input 1-Counter Automata

Definition

- Control states Q (including q_0 , accept, fail)
- Counter c (with initial value 0)
- Instructions
 - $(q : c := c + 1; \text{goto } q')$
 - $(q : c := c - 1; \text{goto } q')$
 - $(q : \text{If } c \geq 0 \text{ then goto } q' \text{ else goto } q'')$.
 - $(q : \text{If } c = 0 \text{ then goto } q' \text{ else goto } q'')$.

\mathbb{Z} -input 1-Counter Automata

Definition

- Control states Q (including q_0 , accept, fail)
- Counter c (with initial value)
- Instructions
 - $(q : c := c + 1; \text{goto } q')$
 - $(q : c := c - 1; \text{goto } q')$
 - $(q : \text{If } c \geq 0 \text{ then goto } q' \text{ else goto } q'')$.
 - $(q : \text{If } c = 0 \text{ then goto } q' \text{ else goto } q'')$.
 - $(q : \text{Read input } S(i). \text{ If } S = X \text{ and } i = K \text{ then goto } q' \text{ else goto } q'')$.
 - $(q : \text{Read input } S(i). \text{ If } S = X \text{ and } i = c \text{ then goto } q' \text{ else goto } q'')$.

\mathbb{Z} -input 1-Counter Automata

Definition

- Control states Q (including q_0 , accept, fail)
- Counter c (with initial value)
- Instructions
 - $(q : c := c + 1; \text{goto } q')$
 - $(q : c := c - 1; \text{goto } q')$
 - $(q : \text{If } c \geq 0 \text{ then goto } q' \text{ else goto } q'')$.
 - $(q : \text{If } c = 0 \text{ then goto } q' \text{ else goto } q'')$.
 - $(q : \text{Read input } S(i). \text{ If } S = X \text{ and } i = K \text{ then goto } q' \text{ else goto } q'')$.
 - $(q : \text{Read input } S(i). \text{ If } S = X \text{ and } i = c \text{ then goto } q' \text{ else goto } q'')$.
 - $(q : \text{If } P(c) \text{ then goto } q' \text{ else goto } q'')$, where P is a unary Presburger predicate.

\mathbb{Z} -input 1-Counter Automata

Definition

- Control states Q (including q_0 , accept, fail)
- Counter c (with initial value)
- Instructions
 - $(q : c := c + 1; \text{goto } q')$
 - $(q : c := c - 1; \text{goto } q')$
 - $(q : \text{If } c \geq 0 \text{ then goto } q' \text{ else goto } q'')$.
 - $(q : \text{If } c = 0 \text{ then goto } q' \text{ else goto } q'')$.
 - $(q : \text{Read input } S(i). \text{ If } S = X \text{ and } i = K \text{ then goto } q' \text{ else goto } q'')$.
 - $(q : \text{Read input } S(i). \text{ If } S = X \text{ and } i = c \text{ then goto } q' \text{ else goto } q'')$.
 - $(q : \text{If } P(c) \text{ then goto } q' \text{ else goto } q'')$, where P is a unary Presburger predicate.

Properties

- Presburger tests can be eliminated,
- Membership problem is **decidable**,
- Emptiness problem is **decidable**.

Construction of the post^* image

Theorem

Let Δ be a BPA(\mathbb{Z}) system, and let \mathcal{A} be a \mathbb{Z} -input 1-counter automaton.

Then, a \mathbb{Z} -input 1-counter automaton \mathcal{A}' with $L(\mathcal{A}') = \text{post}_\Delta^*(L(\mathcal{A}))$ can be effectively constructed.

Steps of the Construction

- Normal Form for BPA(\mathbb{Z}) systems:

– Right hand sides of lengths at most 2,

$$\begin{aligned} X(v) &\rightarrow Y(e_1)Z(e_2) P(v) \\ X(v) &\rightarrow Y(e_1) P(v) \\ X(v) &\rightarrow \epsilon P(v) \end{aligned}$$

– Elimination of ϵ -rules (pop operations)

\Rightarrow Characterization of the symbols which can be rewritten to ϵ

- Special form of \mathbb{Z} -input 1-counter automata
- Saturation construction

Characterization of ϵ -Reducible Terms

Let Δ be a set of BPA(\mathbb{Z}) rules and X a process symbol.

A Presburger formula P_X such that

$$\{k \in \mathbb{Z} \mid P_X(k) \text{ is true}\} = \{k \in \mathbb{Z} \mid X(k) \xrightarrow{*}_{\Delta} \epsilon\}$$

can be *effectively constructed*.

Characterization of ϵ -Reducible Terms

Let Δ be a set of BPA(\mathbb{Z}) rules and X a process symbol.

A Presburger formula P_X such that

$$\{k \in \mathbb{Z} \mid P_X(k) \text{ is true}\} = \{k \in \mathbb{Z} \mid X(k) \xRightarrow{*}_{\Delta} \epsilon\}$$

can be **effectively constructed**.

Reduction to reachability analysis in Alternating 1-Counter Automata

- Construction of an Alternating 1-Counter Automaton (with Presburger tests):

– We associate with a the rule

$$X(v) \rightarrow X_1(v + k_1) \cdots X_n(v + k_n), \quad P(v)$$

the \wedge -transition

$$q_X \rightarrow \{(q_{X_1}, k_1), \dots, (q_{X_n}, k_n)\} \text{ if } P(c)$$

– We associate with a the rule

$$X(v) \rightarrow \epsilon, \quad P(v)$$

the transition

$$q_X \rightarrow \{(accept, 0)\} \text{ if } P(c)$$

- $\{k \in \mathbb{Z} \mid X(k) \xRightarrow{*}_{\Delta} \epsilon\} = pre^*(\{\langle accept, n \rangle \mid n \geq 0\})$

Characterization of ϵ -Reducible Terms

Let Δ be a set of BPA(\mathbb{Z}) rules and X a process symbol.

A Presburger formula P_X such that

$$\{k \in \mathbb{Z} \mid P_X(k) \text{ is true}\} = \{k \in \mathbb{Z} \mid X(k) \xRightarrow{*}_{\Delta} \epsilon\}$$

can be **effectively constructed**.

Reduction to reachability analysis in Alternating 1-Counter Automata

- Construction of an Alternating 1-Counter Automaton (with Presburger tests):

– We associate with a the rule

$$X(v) \rightarrow X_1(v + k_1) \cdots X_n(v + k_n), \quad P(v)$$

the \wedge -transition

$$q_X \rightarrow \{(q_{X_1}, k_1), \dots, (q_{X_n}, k_n)\} \text{ if } P(c)$$

– We associate with a the rule

$$X(v) \rightarrow \epsilon, \quad P(v)$$

the transition

$$q_X \rightarrow \{(accept, 0)\} \text{ if } P(c)$$

- $\{k \in \mathbb{Z} \mid X(k) \xRightarrow{*}_{\Delta} \epsilon\} = pre^*(\{\langle accept, n \rangle \mid n \geq 0\})$ **Constructible [Bouajjani, Esparza, Maler 97]**

Elimination of the ϵ -Rules

Let \mathcal{A} be a \mathbb{Z} -input 1-Counter Automaton, and let Δ be a BPA(\mathbb{Z}) system.

Let Δ_ϵ be the set of ϵ -rules in Δ .

- Construct \mathcal{A}' , the closure of \mathcal{A} under ϵ -rules,

$$L(\mathcal{A}') = \text{post}_{\Delta_\epsilon}^*(L(\mathcal{A}))$$

- Construct Δ' , the smallest set of rules such that,

- $\Delta \setminus \Delta_\epsilon \subseteq \Delta'$,

- For each rule of Δ

$$X(v) \rightarrow X_1(v + k_1)X_2(v + k_2), \quad P(v)$$

Δ' contains the rule

$$X(v) \rightarrow X_2(v + k_2), \quad P(v) \wedge P_{X_1}(v + k_1)$$

- $\implies \text{post}_{\Delta}^*(L(\mathcal{A})) = \text{post}_{\Delta'}^*(L(\mathcal{A}'))$

Special form for \mathbb{Z} -input 1-Counter Automata

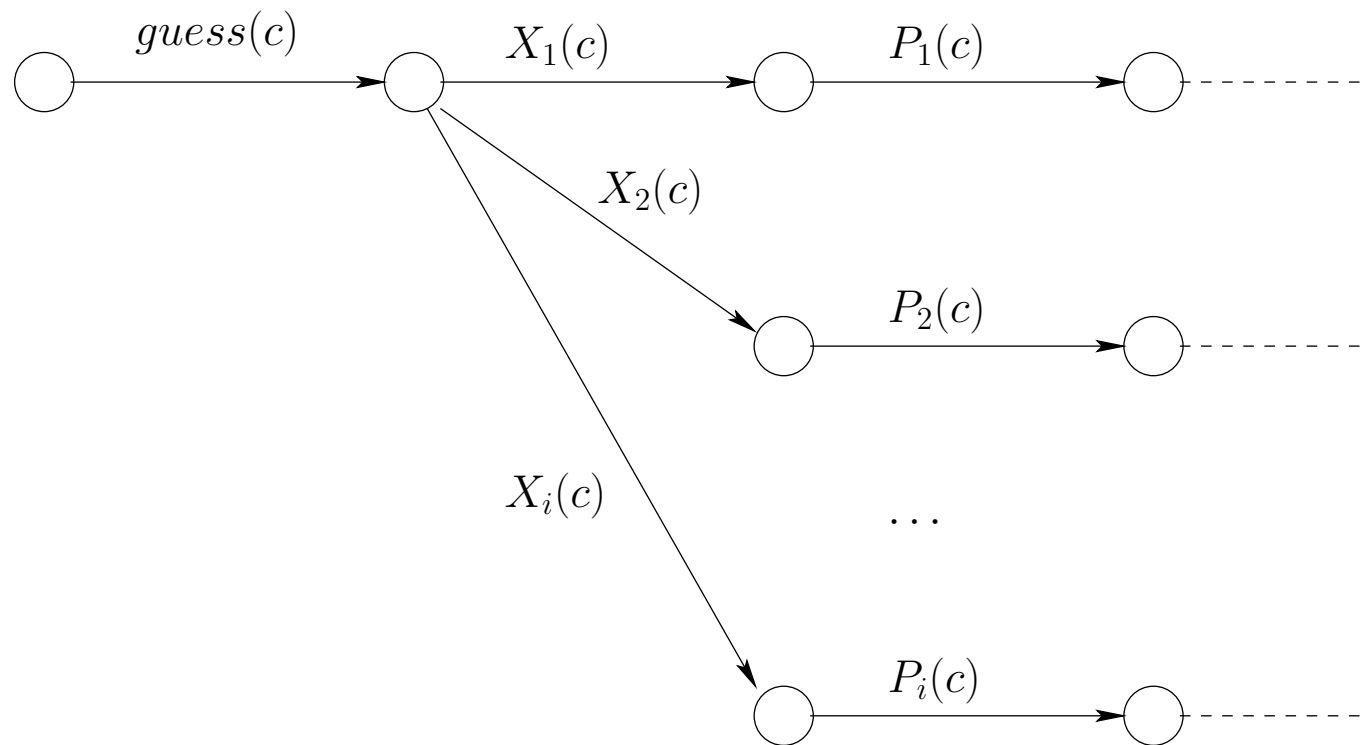


Figure 3: No Tests on the Counter Before an Input

Saturation Construction

$$X(v) \rightarrow Y(v + 3)Z(v - 2), \quad P(v)$$

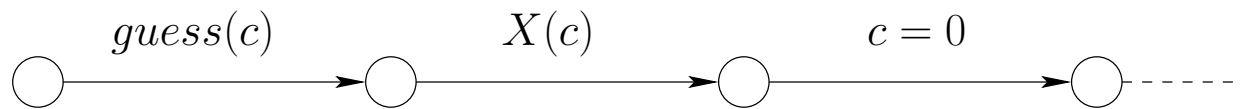


Figure 4: Example

Saturation Construction

$$X(v) \rightarrow Y(v + 3)Z(v - 2), P(v)$$

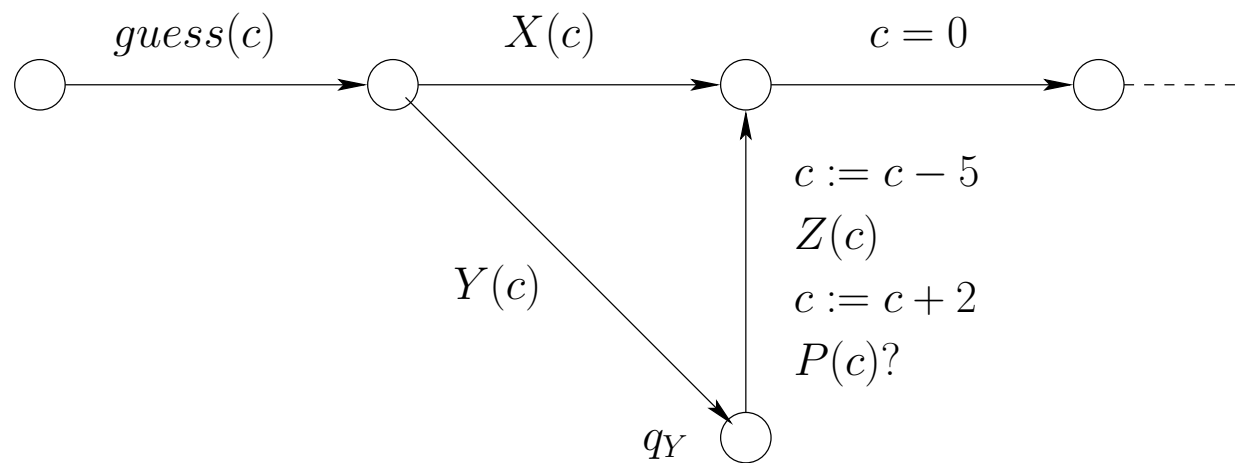


Figure 5: Example

Saturation Construction

$$X(v) \rightarrow Y(v + 3)Z(v - 2), P(v)$$

$$Y(v) \rightarrow Y(v - 4)$$

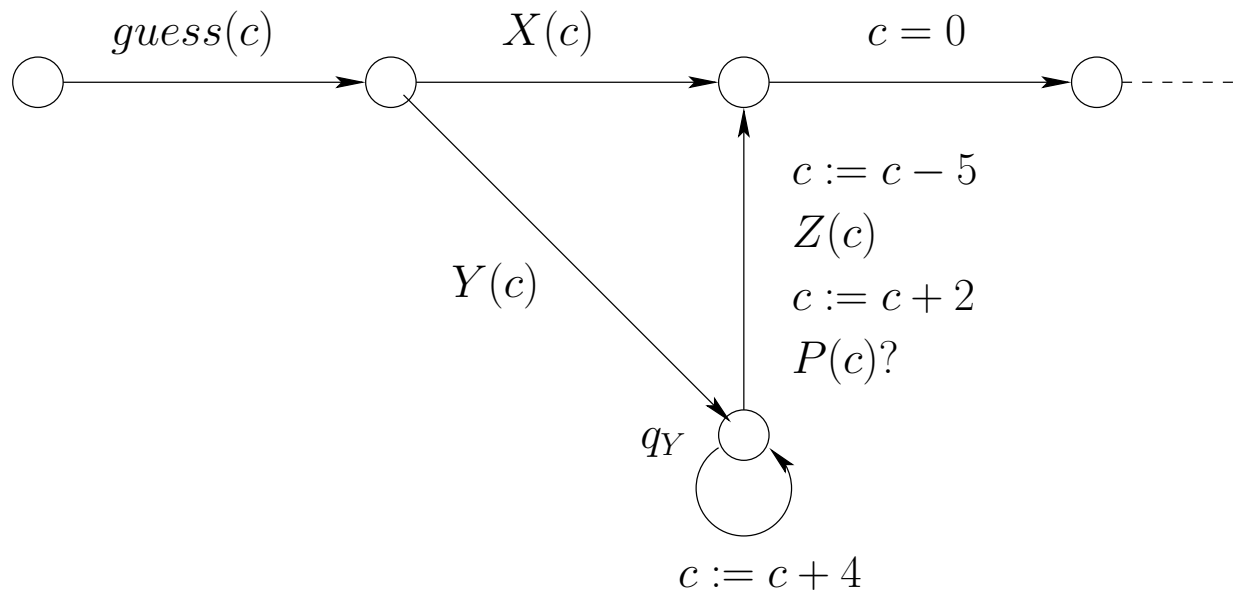
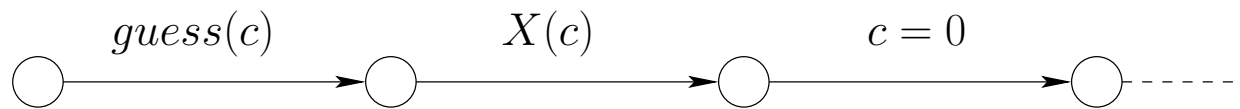


Figure 6: Example

Saturation Construction

$$X(v) \rightarrow X(v - 2)Y(v + 3), \quad P(v)$$

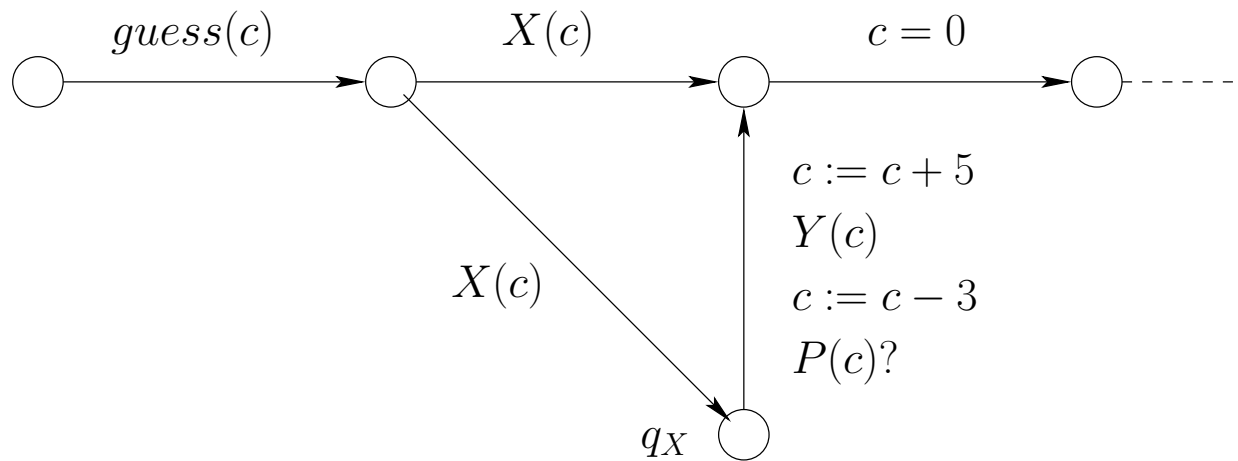


$$X(12) \Longrightarrow X(10)Y(15) \xRightarrow{*} X(4)Y(9)Y(11)Y(13)Y(15)$$

Figure 7: Example

Saturation Construction

$$X(v) \rightarrow X(v - 2)Y(v + 3), \quad P(v)$$

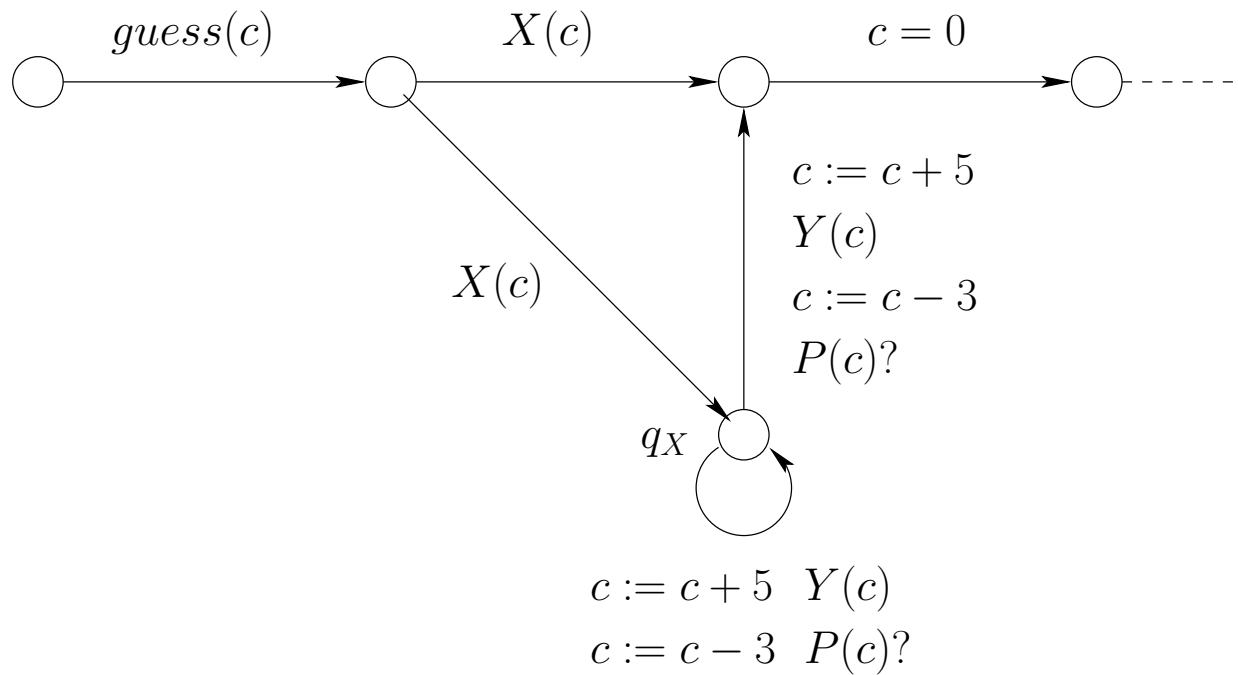


$$X(12) \implies X(10)Y(15) \xRightarrow{*} X(4)Y(9)Y(11)Y(13)Y(15)$$

Figure 8: Example

Saturation Construction

$$X(v) \rightarrow X(v - 2)Y(v + 3), \quad P(v)$$



$$X(12) \Longrightarrow X(10)Y(15) \xRightarrow{*} X(4)Y(9)Y(11)Y(13)Y(15)$$

Figure 9: Example

Recognizing Fibonacci Configurations

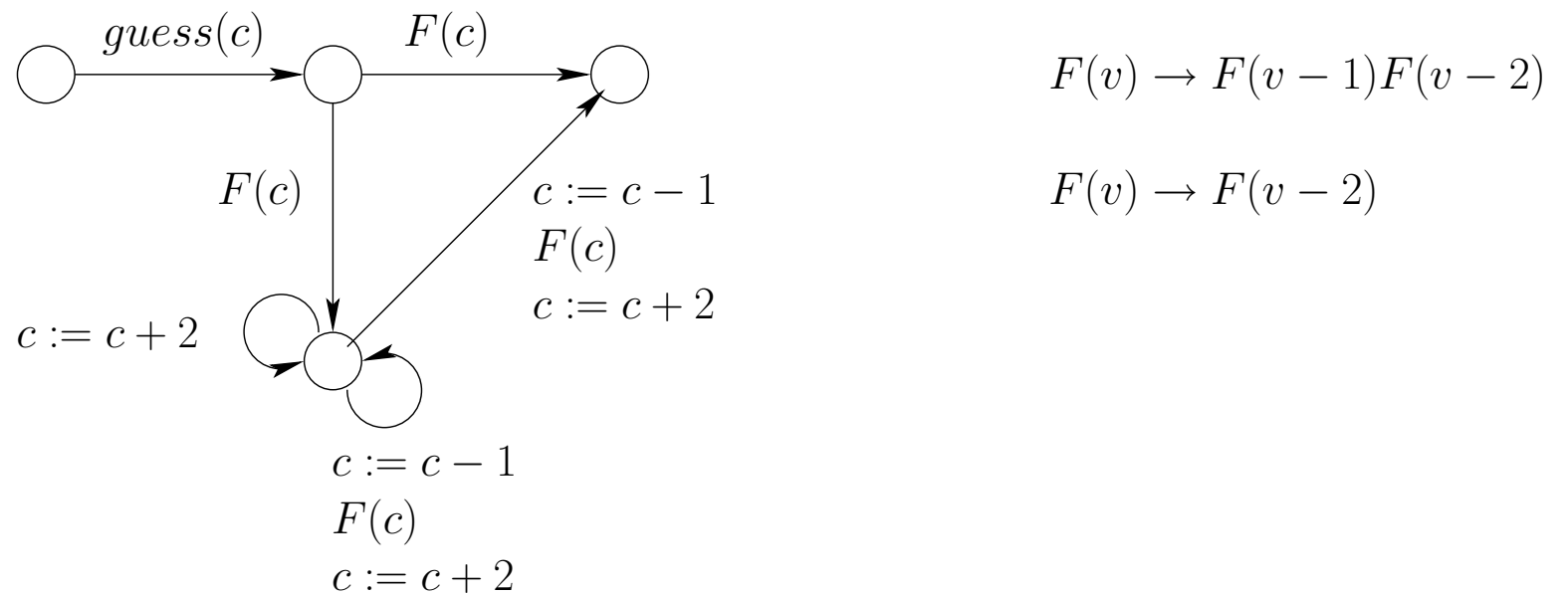


Figure 10: $\text{Post}^*(\{F(k) \mid k \geq 0\})$

Reachability Properties (1)

Theorem

The problem $w \models \text{EF } \varphi$ is **decidable**,

for any BPA(\mathbb{Z}) system Δ , and pattern constraint $\varphi = \langle A_1, \dots, A_n, P \rangle$.

Reachability Properties (1)

Theorem

The problem $w \models \text{EF } \varphi$ is **decidable**,

for any BPA(\mathbb{Z}) system Δ , and pattern constraint $\varphi = \langle A_1, \dots, A_n, P \rangle$.

Construction of a Pushdown Automaton with Reversal Bounded Counters

- The automaton recognizes the set of sequences:

$$\sigma_1 X_1(k_1) \sigma_2 X_2(k_2) \cdots \sigma_n X_n(k_n)$$

such that, there exists

$$w_1 X_1(k_1) w_2 X_2(k_2) \cdots w_n X_n(k_n) \in \text{post}^*(w)$$

where $\forall i \in \{1, \dots, n\}$. $\sigma_i = w_i|_{\Gamma}$

- Integers in the input are incoded in 1-ary,
- Comparisons with the counter are done using reversal bounded counters,
- Presburger tests can also be done in a reversal bounded way,
- **Emptiness of pushdown reversal bounded counter automata is decidable [Ibarra 78].**

Reachability Properties (2)

Theorem

Let Δ be a BPA(\mathbf{Z}) system, let w be an initial configuration (ISS), and let φ be a pattern constraint.

Then, $\{(k_1, \dots, k_n) \in \mathbf{Z}^n \mid \exists w' = w_1 \cdot X_1(k_1) \cdot w_2 \cdot X_2(k_2) \cdots w_n \cdot X_n(k_n) \in \text{post}_\Delta^*(w). w' \models \varphi\}$

is **semilinear** and **effectively constructible**.

Reachability Properties (2)

Theorem

Let Δ be a BPA(\mathbb{Z}) system, let w be an initial configuration (ISS), and let φ be a pattern constraint.

Then, $\{(k_1, \dots, k_n) \in \mathbb{Z}^n \mid \exists w' = w_1 \cdot X_1(k_1) \cdot w_2 \cdot X_2(k_2) \cdots w_n \cdot X_n(k_n) \in \text{post}^*_\Delta(w). w' \models \varphi\}$ is **semilinear** and **effectively constructible**.

Construction of a Pushdown Automaton with Reversal Bounded Counters

- The automaton recognizes the set of sequences:

$$\sigma_1 X_1(k_1) \sigma_2 X_2(k_2) \cdots \sigma_n X_n(k_n)$$

such that, there exists

$$w_1 X_1(k_1) w_2 X_2(k_2) \cdots w_n X_n(k_n) \in \text{post}^*(w)$$

where $\forall i \in \{1, \dots, n\}. \sigma_i = w_i|_\Gamma$

- Integers in the input are incoded in 1-ary,
- Comparisons with the counter are done using reversal bounded counters,
- Presburger tests can also be done in a reversal bounded way,
- **The Parikh image of the language of a pushdown reversal bounded counter automaton is semilinear [Ibarra 78].**

Conclusion

- **Parametrized** prefix rewrite rules \longrightarrow Recursive procedures with **parameters**,
- Symbolic representation recognizing **languages over infinite alphabets**,
- The presented results can be extended to procedures with **string parameters** (stack operations),

$$X(v) \rightarrow Y(av)Z(b^{-1}v), \quad v \in L \quad (L \text{ is a regular language})$$

- Very close to the undecidability border,
- Accurate approximate analysis techniques ?