Presburger Arithmetic And Verification of Infinite State Systems

Jérôme Leroux

LaBRI, University of Bordeaux, Talence, France.

MOVEP'10

An Infinite State System

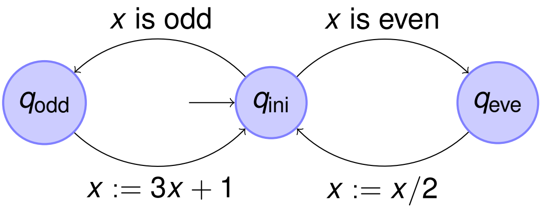


Figure: Syracuse

In this presentation, we consider counter systems:

- A finite set of counter variables.
- A finite control structure (a finite graph).
- Labelled with actions manipulating the variable contents.

Good model:

- C programs, expect system calls, heap manipulations, recursive calls, floating point arithmetic operations...
- C programs manipulating linked data structures [BBH$^+$06].
- Abstraction of communicating processes [BCR01], [BMWK09].

The safety verification problem

Input : An initial and a final configuration.
Decide : if the final configuration is reachable from the initial one.

Some remarks:

- The problem is recursively-countable : we prove the reachability with a path.
- The problem is not recursive : we prove the non-reachability with an inductive invariant that contains the initial configuration but not the final configuration.

The big problem:

- Find out a “good” logic (expressive, decidable) to express invariants.
- Find out a way for computing an invariant in this logic.

Outline

1. Presburger Arithmetic
2. Formulas to Automata
3. Automata to Formulas
4. Presburger Counter Systems Reachability Problem
5. Conclusion

Outline

Grammar

$\mathrm{FO}(\mathbb{N}, +, 0, 1)$

Let X be a countable set of variables.

Definition (Presburger Formulas)

$$
\begin{aligned}
t &:= 0 \mid 1 \mid x \mid t_1 + t_2\\
p &:= t_1 = t_2 \mid \top \mid \perp\\
\phi &:= p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \exists x\, \phi \mid \forall x\, \phi
\end{aligned}
$$

with $x \in X$

Examples:

- Even numbers : $\exists y\; x = y + y$
- Odd numbers : $\exists y\; x + 1 = y + y$

Term Variables

Definition

$\mathrm{var}(t) \subseteq X$ is the set of variables of a term t.

$\mathrm{var}(0) = \emptyset$
$\mathrm{var}(1) = \emptyset$
$\mathrm{var}(x) = \{x\}$
$\mathrm{var}(t_1 + t_2) = \mathrm{var}(t_1) \cup \mathrm{var}(t_2)$

$\mathrm{var}(x + y) = \{x, y\}$.

Free Variables

Definition

$\text{var}(\phi) \subseteq X$ is the set of variables of a Presburger formula ϕ.

$\text{var}(t_1 = t_2) = \text{var}(t_1) \cup \text{var}(t_2)$
$\text{var}(\top) = \emptyset$
$\text{var}(\bot) = \emptyset$
$\text{var}(\neg\phi) = \text{var}(\phi)$
$\text{var}(\phi_1 \vee \phi_2) = \text{var}(\phi_1) \cup \text{var}(\phi_2)$
$\text{var}(\phi_1 \wedge \phi_2) = \text{var}(\phi_1) \cup \text{var}(\phi_2)$
$\text{var}(\exists x\, \phi) = \text{var}(\phi) \backslash \{x\}$
$\text{var}(\forall x\, \phi) = \text{var}(\phi) \backslash \{x\}$

$\text{var}(x = y + y) = \{x, y\}$

$\text{var}(x = x) = \{x\}$

$\text{var}(\exists y\, x = y + y) = \{x\}$

Valuations

Definition

A valuation v is a total function $v : X \mapsto \mathbb{N}$.
$v(t)$ is the valuation of a term t.

$v(0) = 0$
$v(1) = 1$
$v(t_1 + t_2) = v(t_1) + v(t_2)$

For instance if $t = 1 + (x + (x + y))$ then:

$$v(t) = 1 + 2v(x) + v(y)$$

Models

$v \models \phi$ is defined by induction.

$v \models t_1 = t_2$ iff $v(t_1) = v(t_2)$
$v \models \top$
$v \not\models \bot$
$v \models \neg\phi$ iff $v \not\models \phi$
$v \models \phi_1 \vee \phi_2$ iff $v \models \phi_1$ or $v \models \phi_2$
$v \models \phi_1 \wedge \phi_2$ iff $v \models \phi_1$ and $v \models \phi_2$
$v \models \exists x\, \phi$ iff $\exists n \in \mathbb{N}$ such that $v[x \mapsto n] \models \phi$
$v \models \forall x\, \phi$ iff $\forall n \in \mathbb{N}$ we have $v[x \mapsto n] \models \phi$

Presburger Sets

Let $\vec{x} = (x_1, \ldots, x_d)$ be a vector of distinct variables.
Let $v(\vec{x}) = (v(x_1), \ldots, v(x_d))$.

Definition

A set $S \subseteq \mathbb{N}^d$ is said to be denoted by $\phi(\vec{x})$ where ϕ is a Presburger formula with $\text{var}(\phi) \subseteq \{x_1, \ldots, x_d\}$ if:

$$S \ = \ \{v(\vec{x}) \mid v \models \phi\}$$

In this case S is called a Presburger set.

Presburger Sets : Linear Constraints

$\{\vec{n} \in \mathbb{N}^2 \mid n_1 \leq n_2\}$ is denoted by $\phi(x_1, x_2)$ with:
$\phi \ = \ \exists y\, x_1 + y = x_2$

Let $\alpha_1, \ldots, \alpha_d \in \mathbb{Z}$ and $\beta \in \mathbb{Z}$.
$\{\vec{n} \in \mathbb{N}^d \mid \alpha_1 n_1 + \cdots + \alpha_d n_d \leq \beta\}$ is a Presburger set.

Presburger Sets : Divisibility Constraints

$z_1 \sim_m z_2$ if m divides $z_1 - z_2$.

$\{n \in \mathbb{N} \mid n \sim_2 0\}$ is denoted by $\phi(x)$ with:
$\phi \ = \ \exists y\, x = y + y$

Let $\alpha_1, \ldots, \alpha_d \in \mathbb{Z}$, $\beta \in \mathbb{Z}$ and $m \in \mathbb{N}_{>0}$.
$\{\vec{n} \in \mathbb{N}^d \mid \alpha_1 n_1 + \cdots + \alpha_d n_d \sim_m \beta\}$ is a Presburger set.

Quantifier Elimination

$\mathrm{FO}\left(\mathbb{N}, +, 0, 1, \leq, (\sim_m)_{m \in \mathbb{N}_{>0}}\right)$

$t \ := \ 0 \mid 1 \mid x \mid t_1 + t_2$

$p \ := \ t_1 = t_2 \mid t_1 \leq t_2 \mid t_1 \sim_m t_2 \mid \top \mid \bot$

$\phi \ := \ p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \exists x\, \phi \mid \forall x\, \phi$

Definition (Equivalence)

$\phi_1 \equiv \phi_2$ iff for every valuation v we have $v \models \phi_1$ iff $v \models \phi_2$.

Theorem ([Pre29])

Any Presburger formula is effectively equivalent to a quantifier-free formula in $\mathrm{FO}\left(\mathbb{N}, +, 0, 1, \leq, (\sim_m)_{m \in \mathbb{N}_{>0}}\right)$

Satisfiability Problem

Definition (Satisfiability)

ϕ *satisfiable* if $v \models \phi$ for at least one valuation v

Theorem ([Ber77])

The satisfiability problem for the Presburger arithmetic is complete for the class of problems decidable by alternating Turing machines working in 2-EXPTIME with at most n alternations.

The satisfiability problem is decidable in 2-EXPSPACE.

Semilinear Sets

Definition (Linear sets)

$\{\vec{v_0} + n_1\vec{v_1} + \cdots + n_k\vec{v_k} \mid n_1, \ldots, n_k \in \mathbb{N}\}$
with $\vec{v_0}, \ldots, \vec{v_k} \in \mathbb{N}^d$.

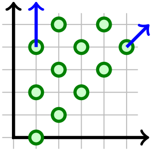


Figure: $\vec{v_0} = (1, 0)$, $\vec{v_1} = (1, 1)$, $\vec{v_2} = (0, 2)$

Definition (Semilinear sets)

Finite union of linear sets.

Theorem ([GS66])

A set is Presburger iff it is semilinear.

Outline

Presburger Formulas vs Automata

Presburger formulas (for denoting Presburger sets):

- Lack canonical representations.
- Simplification procedure difficult.

Deterministic automata (for recognizing regular languages):

- Unique minimal deterministic automaton.
- $n\log(n)$ minimization procedure.

Binary Decomposition

Definition

A word $\sigma = a_1 \ldots a_k$ with $a_j \in \{0, 1\}$ such that:

$$n = a_1 2^0 + \cdots + a_k 2^{k-1}$$

Extension : basis

A *base of decomposition* $r \in \mathbb{N}_{>1}$
The *digit alphabet* $\Sigma_r = \{0, \ldots, r-1\}$

Definition (Encodings for $\mathbb{N}$ in basis r)

A word $\sigma = a_1 \ldots a_k$ with $a_j \in \Sigma_r$ such that:

$$n = a_1 r^0 + \cdots + a_k r^{k-1}$$

Extension : vectors

A dimension $d \in \mathbb{N}$.
The *digit-vector alphabet* $\Sigma_{r,d} = \Sigma_r^d$

Definition (Encodings for $\mathbb{N}^d$)

A word $\sigma = \vec{a_1} \ldots \vec{a_k}$ with $\vec{a_j} \in \Sigma_{r,d}$ such that:

$$n = \vec{a_1} r^0 + \cdots + \vec{a_k} r^{k-1}$$

Sets Recognizable in Base r

Definition

A set $S \subseteq \mathbb{N}^d$ is encoded by a language $L \subseteq \Sigma_{r,d}^*$ if S is the set of vectors $\vec{n} \in \mathbb{N}^d$ with at least one encoding $\sigma \in L$.

If L is regular, S is said to be recognizable in base r.

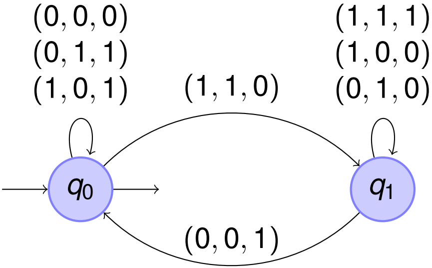


Figure: $\{\vec{n} \in \mathbb{N}^3 \mid n_1 + n_2 = n_3\}$

Saturated Languages

If S_1, S_2 are encoded by languages L_1, L_2
Then $S_1 \cup S_2$ is encoded by $L_1 \cup L_2$
But $S_1 \cap S_2$ is not encoded by $L_1 \cap L_2$

For instance $L_1 = \{0\}$ and $L_2 = \{0.0\}$.
Then $S_1 = S_2 = \{0\} = S_1 \cap S_2$
but $L_1 \cap L_2 = \emptyset$.

Definition (Saturated)

$L \subseteq \Sigma_{r,d}^*$ such that for every encodings σ_1, σ_2 of the same vector $\vec{n}$:

$$\sigma_1 \in L \iff \sigma_2 \in L$$

Lemma

*Every set $S \subseteq \mathbb{N}^d$ is encoded by a **unique** saturated language $L \subseteq \Sigma_{r,d}^*$.*

Saturation Procedure

Lemma

if X is encoded by L then X is encoded by the saturated language:

$$\bigcup_{i\in\mathbb{N}} L \cdot (0^i)^{-1} \cdot 0^*$$

In particular this language is regular if L is regular.

Boolean Operations

Let S_1, S_2, S be encoded by the saturated languages L_1, L_2, L

$S_1 \cup S_2$ is encoded by the saturated language $L_1 \cup L_2$
$S_1 \cap S_2$ is encoded by the saturated language $L_1 \cap L_2$
$\mathbb{N}^d \backslash S$ is encoded by the saturated language $\Sigma_{r,d}^* \backslash L$

Lemma

The class of subsets of $\mathbb{N}^d$ recognizable in base r is stable by union, intersection, and complement.

Projection

$$\begin{array}{rcccl} \pi_i & : & \mathbb{N}^d & \longrightarrow & \mathbb{N}^{d-1} \\ & & \vec{n} & \longmapsto & (n_1, \ldots, n_{i-1}, n_{i+1}, \ldots, n_d) \end{array}$$

Lemma

Let $X \subseteq \mathbb{N}^d$ encoded by $L \subseteq \Sigma_{r,d}^*$.
Then $\pi_i(X)$ is encoded by:

$$\{\pi_i(a_1) \cdots \pi_i(a_k) \mid a_1 \ldots a_k \in L\}$$

In particular the language is regular if L is regular.

Presburger Automata

Theorem ([Cob69], [Kla04],[DGH10])

Every Presburger set is recognizable in base r.
Moreover, the set denoted by $\phi(\vec{x})$ can be encoded by a deterministic automaton in 3-EXPTIME.

Efficient algorithms for:

- linear constraints.
- divisibility constraints.

can be found in [WB95].

What Are The Recognizable Sets ?

$r^* = \{r^n \mid n \in \mathbb{N}\}$ is encoded by $L = 0^*.1$.

Thus any set in $\mathrm{FO}(\mathbb{N}, +, 0, 1, r^*)$ is recognizable in base r.

Valuation Function

Definition

The valuation function $V_r : \mathbb{N} \mapsto r^*$ is defined over any $n \in \mathbb{N}_{>0}$ by:
$V_r(n)$ is the greatest integer in r^* that divides n

$\{(n, V_r(n)) \mid n \in \mathbb{N}\backslash\{0\}\}$ is encoded by:

$$L = (0,0)^*.\bigcup_{b \neq 0}(b,1).\Sigma_{r,2}^*$$

Characterization

Theorem ([BHMV94])

A set is recognizable in base r if and only if it is definable in $\mathrm{FO}(\mathbb{N}, +, 0, 1, V_r)$.

Moreover, the set denoted by $\phi(\vec{x})$ can be encoded by a deterministic automaton in time:

$2^{2^{\cdot^{\cdot^{\cdot^{2^n}}}}}$ a tower of height n.

Sets Recognizable in Multiple Basis

Observe that if $S \subseteq \mathbb{N}^d$ is recognizable in basis r then it is recognizable in basis r^n for every $n \in \mathbb{N}_{>0}$.

Definition (Multiplicatively dependant)

$r_1, r_2 \in \mathbb{N}_{>1}$ are multiplicatively dependant if there exists $n_1, n_2 \in \mathbb{N}_{>0}$ such that $r_1^{n_1} = r_2^{n_2}$

Lemma

Let $r_1, r_2 \in \mathbb{N}_{>1}$ be multiplicatively dependant basis of decomposition. A set is recognizable in base r_1 iff it is recognizable in base r_2.

Theorem ([Cob69],[Sem77])

Let $r_1, r_2 \in \mathbb{N}_{>1}$ be multiplicatively independant basis of decomposition. A set is recognizable in basis r_1 and r_2 iff it is Presburger.

Outline

Why ?

- Understand the complexity gap.
- Extract geometrical properties.
- Used the automata minimization procedure as a formula simplification and normalization procedure.

Ultimately Periodic Sets

Definition

A set $S \subseteq \mathbb{N}$ is ultimately periodic if

$$\exists k \, \exists m \geq 1 \ \wedge \ (\, \forall n \geq k \ \Rightarrow \ (\, n \in S \ \Leftrightarrow \ n + m \in S \,) \,)$$

is equivalent to $\top$.

A set $S \subseteq \mathbb{N}$ is Presburger iff it is ultimately periodic (based on quantifier elimination).

Muchnik Criterion

Theorem ([Muc03])

Let $d \in \mathbb{N}$ and let P_d be an uninterpreted symbol of dimension d.

We can effectively compute a formula ψ_d in $\mathrm{FO}\,(\mathbb{N}, +, 0, 1, P_d)$ such that the formula ψ_d is equivalent to $\top$ when P_d is interpreted as a set $S \subseteq \mathbb{N}^d$ iff S is Presburger.

Related Works

Presburger Synthesis:

- An EXPTIME algorithm for Boolean combinations of linear equalities [Ler03].
- An EXPTIME algorithm for conjunctions of inequalities [Lat04].
- An 2-EXPTIME algorithm for sets $B + P^*$ where B, P finite [Lug04].
- A PTIME algorithm for the full Presburger arithmetic [Ler05].

Affine Spaces

Definition (Affine spaces)

A non-empty subset A of $\mathbb{Q}^d$ satisfying a conjunction of linear equalities. Its direction $\vec{A}$ is the subset of $\mathbb{Q}^n$ satisfying the *homogeneous* conjunction.

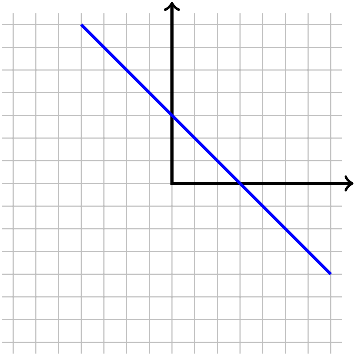
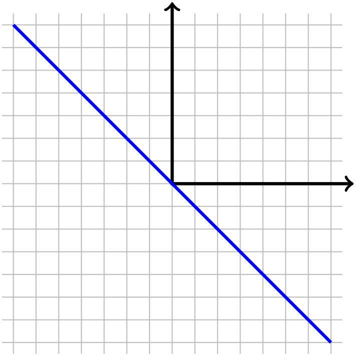
 

Figure: An affine space and its direction

Semi-Affine spaces

Definition (Semi-affine spaces[Ler04])

A finite union $S = \bigcup_{i=1}^{k} A_i$ of non-empty affine spaces A_i.
Its direction $\vec{S} = \bigcup_{i=1}^{k} \vec{A_i}$.

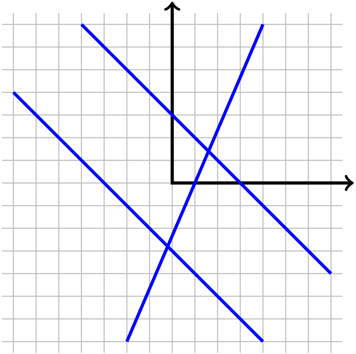


Figure: A semi-affine space and its direction

Semi-Affine Encapsulation

Definition ([Ler04])

For any set $R \subseteq \mathbb{Q}^d$ there exists a minimal for $\subseteq$ semi-affine space that contains R called the **semi-affine encapsulation** of R.

Theorem ([Ler05])

Directions of semi-affine encapsulations of subsets of $\mathbb{N}^d$ recognized by automata are computable in PTIME.

Boundaries

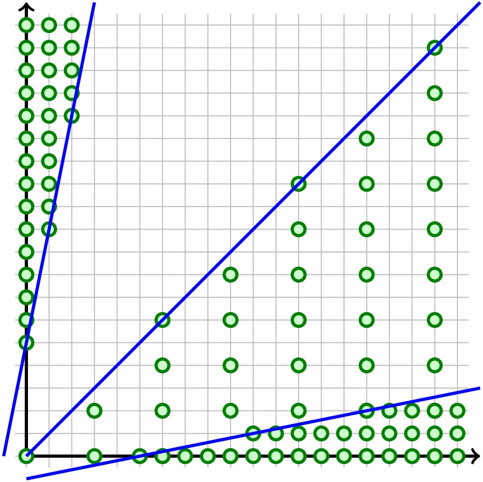
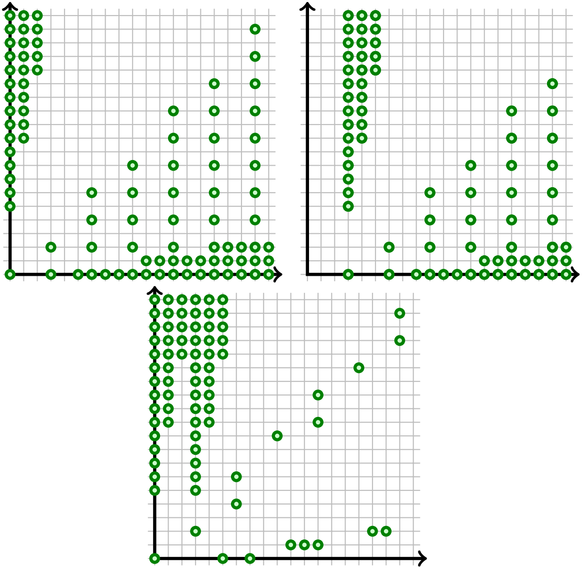
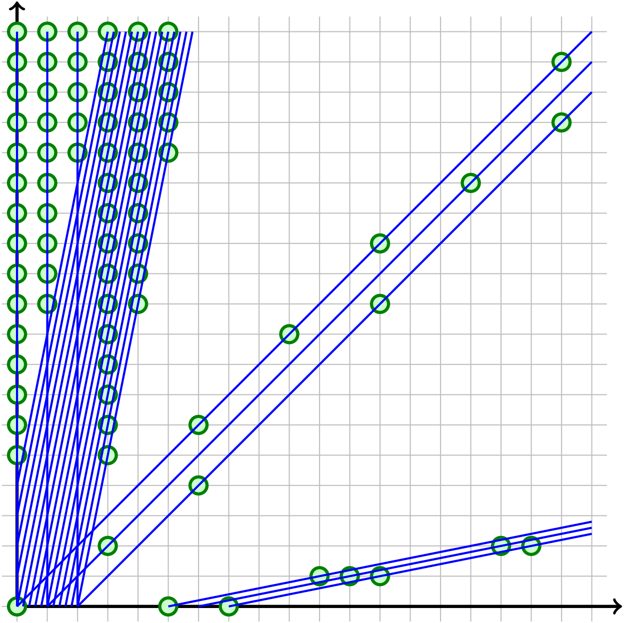


Figure: A Presburger set

Key Idea





Theorem ([Ler05])

We can decide in PTIME if an automaton recognizes a Presburger set S. Moreover, in this case, we can compute in PTIME a Presburger formula $\phi(\vec{x})$ that denotes S.

This algorithm is implemented in TAPAS [LP09].

Outline

Presburger Relations

Let $(x_1, x'_1, \ldots, x_d, x'_d)$ be a sequence of distinct variables.
$\vec{x} = (x_1, \ldots, x_d)$
$\vec{x'} = (x'_1, \ldots, x'_d)$

Definition

A relation $R \subseteq \mathbb{N}^d \times \mathbb{N}^d$ is said to be denoted by $\phi(\vec{x}, \vec{x'})$ where ϕ is a Presburger formula with var(ϕ) $\subseteq \{x_1, x'_1, \ldots, x_d, x'_d\}$ if:

$$R = \{(v(\vec{x}), v(\vec{x'})) \mid v \models \phi\}$$

In this case R is called a Presburger relation.

Presburger Counter Systems

Definition

A Presburger counter system is a tuple (Q, d, Δ) where:

- Q is a finite set of *control states*.
- Δ is a finite set of triples (q, ϕ, q') where $q, q' \in Q$ and ϕ is a Presburger formula satisfying $\mathrm{var}(\phi) \subseteq \{x_1, x_1', \ldots, x_d, x_d'\}$.

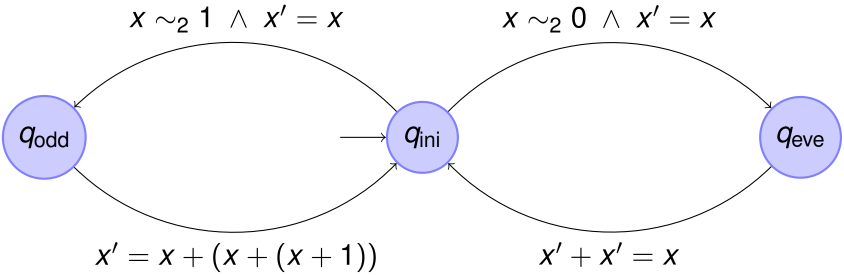


Figure: A Presburger counter system.

Semantics

A configuration c is a couple $(q, \vec{n}) \in Q \times \mathbb{N}^d$.

The semantics is given by $(q, \vec{n}) \xrightarrow{\phi} (q', \vec{n'})$ if $(q, \phi, q') \in \Delta$ and $(\vec{n}, \vec{n'}) \in R_\phi$ where R_ϕ is the Presburger relation denoted by $\phi(\vec{x}, \vec{x'})$.

The Reachability Problem

We introduce $\xrightarrow{*}$ defined by $c \xrightarrow{*} c'$ if there exists:

$$c = c_0 \xrightarrow{\phi_1} c_1 \cdots \xrightarrow{\phi_k} c_k = c'$$

In this case c' is said to be *reachable* from c

With the Minsky machines:

Lemma

The reachability problem is undecidable for the class of Presburger Counter Systems.

Inductive Invariant

Definition

A set $C \subseteq Q \times \mathbb{N}^d$ is called an inductive invariant if for every $c \xrightarrow{\phi} c'$ with $c \in C$, we have $c' \in C$.

Definition

A set $C \subseteq Q \times \mathbb{N}^d$ is said Presburger if $C = \bigcup_{q \in Q} N_q$ where $N_q \subseteq \mathbb{N}^d$ is Presburger for every q.

Symbolic Computation

Semi-algorithm deciding the reachability problem $c_\bullet \xrightarrow{*} c_!$.

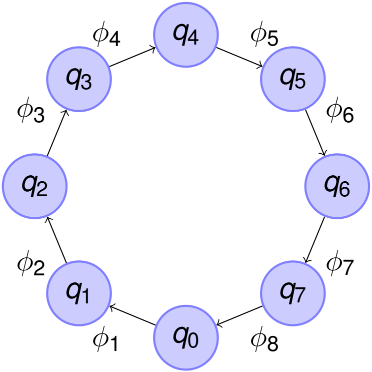
Initially $k = 0$ and $C_0 = \{c_\bullet\}$.
We repeat forever the following loop:
If $c_! \in C_k$ return “reachable”.
Compute:

$$C_{k+1} = C_k \cup \{c_{k+1} \mid \text{there exists } c_k \xrightarrow{\phi} c_{k+1} \text{ with } c_k \in C_k\}$$

If $C_{k+1} \subseteq C_k$ return “unreachable”.
Otherwise increment k.

Acceleration Techniques

The iterative effect of cycles.



$R_1 \cdots R_8$ is a Presburger relation denoted by $\phi(\vec{x}, \vec{x'})$.

In general even if R is a Presburger relation, R^* is not a Presburger binary relation.
We cannot even decide if it is definable in the Presburger arithmetic.

Some Results

- Let $f : \mathbb{Z}^d \mapsto \mathbb{Z}^d$ be a total linear function $f(\vec{x}) = M\vec{x} + \vec{v}$. Then f^* is definable in $\mathrm{FO}\,(\mathbb{Z}, +, 0, 1, \leq)$ if and only if M^* is finite [Boi03].
- Let $f : \mathbb{Z}^d \mapsto \mathbb{Z}^d$ be a linear function partially defined over a Presburger set. Then f^* is definable in $\mathrm{FO}\,(\mathbb{Z}, +, 0, 1, \leq)$ if M^* is finite [FL02].
- Let ϕ be a conjunction of difference constraints $z_i - z_j \leq c$ with $z_i \in \{x_i, x_i'\}$ and $z_j \in \{x_j, x_j'\}$. Then ϕ^* is effectively definable in $\mathrm{FO}\,(\mathbb{Z}, +, 0, 1, \leq)$ [CJ98].
- Let ϕ be a conjunction of octagonal constraints $z_i - z_j \leq c$ or $z_i + z_j \leq c$ with $z_i \in \{x_i, x_i'\}$ and $z_j \in \{x_j, x_j'\}$. Then ϕ^* is effectively definable in $\mathrm{FO}\,(\mathbb{Z}, +, 0, 1, \leq)$ [BGI09].

Sometimes Approximation Is Mandatory

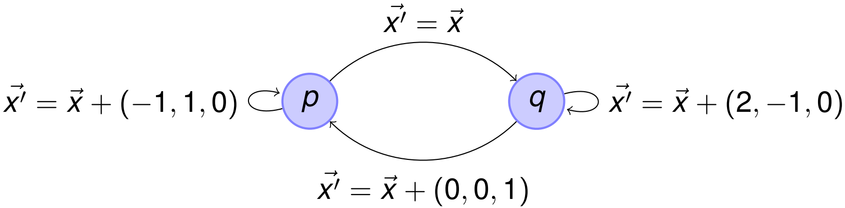


Figure: A Vector Addition System With States [HP76]

Reachability set from $(p, (1, 0, 0))$

$$\{p\} \times \{\vec{n} \in \mathbb{N}^3 \mid n_1 + n_2 \leq 2^{n_3}\}$$
$$\cup \{q\} \times \{\vec{n} \in \mathbb{N}^3 \mid n_1 + 2n_2 \leq 2^{n_3+1}\}$$

Vector Addition Systems With States

Definition

A Vector Addition System with States (VASS) is a Presburger counter systems with formulas ϕ of the form $\vec{x'} = \vec{x} + \vec{v}$ with $\vec{v} \in \mathbb{Z}^d$.

Theorem ([Ler09])

If $c_{!}$ is not reachable from $c_{\bullet}$ then there exists a Presburger Inductive Invariant that contains $c_{\bullet}$ but not $c_{!}$.

Outline

Conclusion

We have seen:

- Some links between automata and Presburger arithmetic.
- Applications of the Presburger arithmetic on the verification of counter systems.

Many open problems:

- Find a decision procedure for the Presburger arithmetic combining efficiently formulas and automata.
- Improve acceleration techniques to be “complete” for the VASS reachability problem.
- Find interesting classes of Presburger formulas that can be iterated.
- And many other problems based on results not presented here.

References I

Ahmed Bouajjani, Marius Bozga, Peter Habermehl, Radu Iosif, Pierre Moro, and Tomás Vojnar.
Programs with lists are counter automata.
In *Computer Aided Verification, 18th International Conference, CAV 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, volume 4144 of *Lecture Notes in Computer Science*, pages 517–531. Springer, 2006.

Thomas Ball, Sagar Chaki, and Sriram K. Rajamani.
Parameterized verification of multithreaded software libraries.
In *Tools and Algorithms for the Construction and Analysis of Systems, 7th International Conference, TACAS 2001 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2001 Genova, Italy, April 2-6, 2001, Proceedings*, volume 2031 of *Lecture Notes in Computer Science*, pages 158–173. Springer, 2001.

References II

Leonard Berman.
Precise bounds for Presburger arithmetic and the reals with addition: Preliminary report.
In *Proc. 18th IEEE Symp. Foundations of Computer Science (FOCS'77), Providence, RI, USA, Oct.-Nov. 1977*, pages 95–99, Providence, Rhode Island, 31 October–2 November 1977. IEEE.

Marius Bozga, Codruta Gîrlea, and Radu Iosif.
Iterating octagons.
In *Tools and Algorithms for the Construction and Analysis of Systems, 15th International Conference, TACAS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings*, volume 5505 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 2009.

References III

Véronique Bruyère, Georges Hansel, Christian Michaux, and Roger Villemaire.
Logic and *p*-recognizable sets of integers.
Bull. Belg. Math. Soc., 1(2):191–238, March 1994.

Gérard Basler, Michele Mazzucchi, Thomas Wahl, and Daniel Kroening.
Symbolic counter abstraction for concurrent software.
In *Computer Aided Verification, 21st International Conference, CAV 2009, Grenoble, France, June 26 - July 2, 2009. Proceedings*, volume 5643 of *Lecture Notes in Computer Science*, pages 64–78. Springer, 2009.

Bernard Boigelot.
On iterating linear transformations over recognizable sets of integers.
Theor. Comput. Sci., 309(1-3):413–468, 2003.

References IV

Hubert Comon and Yan Jurski.
Multiple counters automata, safety analysis and presburger arithmetic.
In *Computer Aided Verification, 10th International Conference, CAV '98, Vancouver, BC, Canada, June 28 - July 2, 1998, Proceedings*, volume 1427 of *Lecture Notes in Computer Science*, pages 268–279. Springer, 1998.

A. Cobham.
On the base-dependance of sets of numbers recognizable by finite automata.
Mathematical Systems Theory, 3:186–192, 1969.

References V

Antoine Durand-Gasselin and Peter Habermehl.
On the use of non-deterministic automata for presburger arithmetic.
In *CONCUR 2010*, 2010.
to appear.

Alain Finkel and Jérôme Leroux.
How to compose presburger-accelerations: Applications to broadcast protocols.
In *FST TCS 2002: Foundations of Software Technology and Theoretical Computer Science, 22nd Conference Kanpur, India, December 12-14, 2002, Proceedings*, volume 2556 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2002.

Seymour Ginsburg and Edwin H. Spanier.
Semigroups, Presburger formulas and languages.
Pacific J. Math., 16(2):285–296, 1966.

References VI

J. E. Hopcroft and J. Pansiot.
On the reachability problem for 5-dimensional vector addition systems.
Technical Report TR76-280, Cornell University, Computer Science Department, June 1976.

Felix Klaedtke.
On the automata size for presburger arithmetic.
In *Proc. 19th Annual IEEE Symposium on Logic in Computer Science (LICS'04), Turku, Finland July 2004*, pages 110–119. IEEE Comp. Soc. Press, 2004.

Louis Latour.
From automata to formulas: Convex integer polyhedra.
In *19th IEEE Symposium on Logic in Computer Science (LICS 2004), 14-17 July 2004, Turku, Finland, Proceedings*, pages 120–129. IEEE Computer Society, 2004.

References VII

Jérôme Leroux.
Algorithmique de la vérification des systèmes à compteurs. Approximation et accélération. Implémentation de l'outil Fast.
PhD thesis, Ecole Normale Supérieure de Cachan, Laboratoire Spécification et Vérification. CNRS UMR 8643, décembre 2003.

Jérôme Leroux.
Disjunctive invariants for numerical systems.
In *Automated Technology for Verification and Analysis: Second International Conference, ATVA 2004, Taipei, Taiwan, ROC, October 31-November 3, 2004. Proceedings*, volume 3299 of *Lecture Notes in Computer Science*, pages 93–107. Springer, 2004.

References VIII

Jérôme Leroux.
A polynomial time presburger criterion and synthesis for number decision diagrams.
In *20th IEEE Symposium on Logic in Computer Science (LICS 2005), 26-29 June 2005, Chicago, IL, USA, Proceedings*, pages 147–156. IEEE Computer Society, 2005.

Jérôme Leroux.
The general vector addition system reachability problem by presburger inductive invariants.
In *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009, 11-14 August 2009, Los Angeles, CA, USA*, pages 4–13. IEEE Computer Society, 2009.

References IX

Jérôme Leroux and Gérald Point.
Tapas: The talence presburger arithmetic suite.
In *Tools and Algorithms for the Construction and Analysis of Systems, 15th International Conference, TACAS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings*, volume 5505 of *Lecture Notes in Computer Science*, pages 182–185. Springer, 2009.

Denis Lugiez.
From automata to semilinear sets: A logical solution for sets l(c, p).
In *Implementation and Application of Automata, 9th International Conference, CIAA 2004, Kingston, Canada, July 22-24, 2004, Revised Selected Papers*, volume 3317 of *Lecture Notes in Computer Science*, pages 321–322. Springer, 2004.

References X

A. Muchnik.
The definable criterion for definability in presburger arithmetic and its applications.
Theoretical Computer Science, 290:1433–1444, 2003.

M. Presburger.
Uber die volstandigkeit eines gewissen systems der arithmetik ganzer zahlen, in welchem die addition als einzige operation hervortritt.
In *C. R. 1er congres des Mathematiciens des pays slaves, Varsovie*, pages 92–101, 1929.

A.L. Semenov.
Presburgerness of predicates regular in two number systems.
Siberian Mathematical Journal, 18:289–299, 1977.

References XI

Pierre Wolper and Bernard Boigelot.
An automata-theoretic approach to Presburger arithmetic constraints.
In *Proc. 2nd Int. Symp. Static Analysis (SAS'95), Glasgow, UK, Sep. 1995*, volume 983 of *Lecture Notes in Computer Science*, pages 21–32. Springer, 1995.