Orbits of Linear Maps and Regular Languages

S. Tarasov, M. Vyalyi

Dorodnitsyn Computing Center of RAS

CSR 2011

Contents

Chamber hitting problem

Definition

An orbit $\mathrm{Orb}_\Phi\, x$ is $\{\Phi^k x : k \in \mathbb{Z}^+\}$, where $\Phi\colon \mathbb{Q}^d \to \mathbb{Q}^d$ is a linear map and $x \in \mathbb{Q}^d$.

Definition

A chamber $H_S = \{x \in \mathbb{Q}^d : \mathrm{sign}(h_i(x)) = s_i \text{ for } 1 \leqslant i \leqslant m\}$, where h_i are affine functions and $s \in \{\pm 1, 0\}^m$ is a sign pattern.

Chamber hitting problem (CHP)

INPUT: $\Phi, x_0, h_1, \ldots, h_m, s$.

OUTPUT: 'yes' if $\mathrm{Orb}_\Phi\, x_0 \cap H_s \neq \varnothing$ and 'no' otherwise.

Chamber hitting problem

Definition

An orbit $\operatorname{Orb}_\Phi x$ is $\{\Phi^k x : k \in \mathbb{Z}^+\}$, where $\Phi\colon \mathbb{Q}^d \to \mathbb{Q}^d$ is a linear map and $x \in \mathbb{Q}^d$.

Definition

A chamber $H_S = \{x \in \mathbb{Q}^d : \operatorname{sign}(h_i(x)) = s_i \text{ for } 1 \leqslant i \leqslant m\}$, where h_i are affine functions and $s \in \{\pm 1, 0\}^m$ is a sign pattern.

Chamber hitting problem (CHP)

INPUT: $\Phi, x_0, h_1, \dots, h_m, s$.
OUTPUT: 'yes' if $\operatorname{Orb}_\Phi x_0 \cap H_s \neq \varnothing$ and 'no' otherwise.

Chamber hitting problem

Definition

An orbit $\mathrm{Orb}_\Phi\, x$ is $\{\Phi^k x : k \in \mathbb{Z}^+\}$, where $\Phi\colon \mathbb{Q}^d \to \mathbb{Q}^d$ is a linear map and $x \in \mathbb{Q}^d$.

Definition

A chamber $H_S = \{x \in \mathbb{Q}^d : \mathrm{sign}(h_i(x)) = s_i \text{ for } 1 \leqslant i \leqslant m\}$, where h_i are affine functions and $s \in \{\pm 1, 0\}^m$ is a sign pattern.

Chamber hitting problem (CHP)

INPUT: $\Phi, x_0, h_1, \ldots, h_m, s$.

OUTPUT: 'yes' if $\mathrm{Orb}_\Phi\, x_0 \cap H_s \neq \varnothing$ and 'no' otherwise.

Special cases: the orbit problem

Orbit problem

INPUT: Φ, x, y.
OUTPUT: 'yes' if $y \in \mathrm{Orb}_{\Phi}\, x$ and
'no' otherwise.

In this case the chamber is $\{y\}$.

Theorem (Kannan, Lipton, 1986)

There exists a polynomial time algorithm for the orbit problem.

Special cases: the orbit problem

Orbit problem

INPUT: Φ, x, y.
OUTPUT: 'yes' if $y \in \mathrm{Orb}_{\Phi}\, x$ and
'no' otherwise.

In this case the chamber is $\{y\}$.

Theorem (Kannan, Lipton, 1986)

There exists a polynomial time algorithm for the orbit problem.

Problems reducible to CHP

Skolem problem

INPUT: $a_1, \ldots, a_d; b_1, \ldots, b_d$.

x_n — a linear recurrent sequence

$$x_n = \sum_{i=1}^{d} a_i x_{n-i}, \ (n > d), \qquad x_n = b_n \ (1 \leqslant n \leqslant d)$$

OUTPUT: 'yes' if $x_n = 0$ for some n and
'no' otherwise.

Positivity problem

INPUT: $a_1, \ldots, a_d; b_1, \ldots, b_d$; x_n is LRS.

OUTPUT: 'yes' if $x_n > 0$ for all n and
'no' otherwise.

Problems reducible to CHP

Skolem problem

INPUT: $a_1, \dots, a_d; b_1, \dots, b_d$.

x_n — a linear recurrent sequence

$$x_n = \sum_{i=1}^{d} a_i x_{n-i}, \ (n > d), \qquad x_n = b_n \ (1 \leqslant n \leqslant d)$$

OUTPUT: 'yes' if $x_n = 0$ for some n and
'no' otherwise.

Positivity problem

INPUT: $a_1, \dots, a_d; b_1, \dots, b_d$; x_n is LRS.

OUTPUT: 'yes' if $x_n > 0$ for all n and
'no' otherwise.

State of the art

Open questions

Is CHP decidable? Is Skolem problem decidable? Is positivity problem decidable?

Known decidability results for small d

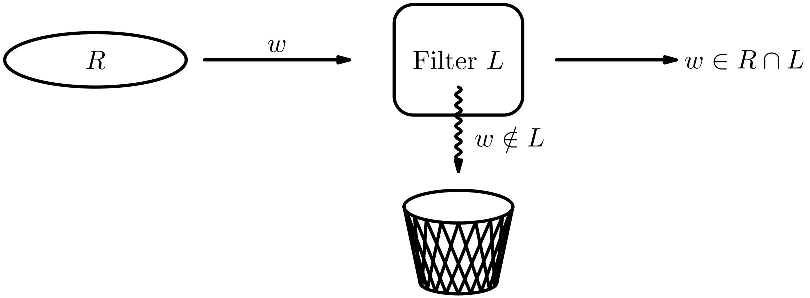
	$d = 2$	$d = 3$	$d = 4$	$d = 5$
Skolem	folklore	Vereshchagin, 1985		Halava et al., 2005
Pos. pr.	Halava et al., 2006	Laohakosol, Tangsupphathawat, 2009		
CHP	Sechin, 2011			

Regular realizability problems (RR)

A set $L \subset \Sigma^*$ is called a filter. Each filter determines a specific regular realizability problem:

L-realizability problem

INPUT: a description of a regular language R.
OUTPUT: 'yes' if $R \cap L \neq \varnothing$ and
'no' otherwise.



Permutation filter

Definition

$P_{\mathbb{B}} \subset \{\#, 0, 1\}^*$ consists of permutation words, i.e., words of the form

$$\#w_1\#w_2\# \dots w_N\#,$$

where

- $w_i \in \{0, 1\}^*$ are blocks,
- $|w_i| = n$, $i = 1, 2, \dots, N$ (n is the block rank),
- $N = 2^n$, $n \geqslant 1$,
- each binary word of length n is a block.

Examples

$$\#00\#11\#10\#01\# \in P_{\mathbb{B}}$$

$$\#10\#11\#00\#01\# \in P_{\mathbb{B}}$$

$$\#10\#01\#00\#11\# \in P_{\mathbb{B}}$$

Orbits vs Regular realizability

Theorem (Tarasov, Vyalyi, 2010)

CHP and $P_{\mathbb{B}}$-realizability problem are Turing equivalent.

From $P_{\mathbb{B}}$-realizability to CHP

Reduction starts from a $\mathbb{Q}$-linear extension of the transition monoid.

Orbits vs Regular realizability

Theorem (Tarasov, Vyalyi, 2010)

CHP and $P_{\mathbb{B}}$-realizability problem are Turing equivalent.

From $P_{\mathbb{B}}$-realizability to CHP

Reduction starts from a $\mathbb{Q}$-linear extension of the transition monoid.

From CHP to $P_{\mathbb{B}}$-realizability

The idea is to represent an arithmetic computation in a 'natural' form.

The main construction

- R_1, R_2 — regular languages.
- How to check that there exists an integer n such that

$$\mathrm{Card}(\{w : |w| = n \ \wedge \ w \in R_1\}) = \mathrm{Card}(\{w : |w| = n \ \wedge \ w \in R_2\})? \quad (\clubsuit)$$

- Regular expression

$$E = \#((R_1 \cap R_2)\#)^*((R_1 \setminus R_2)\#(R_2 \setminus R_1)\#)^*((\overline{R_1} \cap \overline{R_2})\#)^*$$

- $(\clubsuit)$ is equivalent to $E \cap P_{\mathbb{B}} \neq \varnothing$.

From CHP to $P_{\mathbb{B}}$-realizability

The idea is to represent an arithmetic computation in a ‘natural’ form.

The main construction

- R_1, R_2 — regular languages.
- How to check that there exists an integer n such that

$$\mathrm{Card}(\{w : |w| = n \ \wedge \ w \in R_1\}) = \mathrm{Card}(\{w : |w| = n \ \wedge \ w \in R_2\})? \quad (\clubsuit)$$

- Regular expression

$$E = \#((R_1 \cap R_2)\#)^*((R_1 \setminus R_2)\#(R_2 \setminus R_1)\#)^*((\overline{R_1} \cap \overline{R_2})\#)^*$$

- $(\clubsuit)$ is equivalent to $E \cap P_{\mathbb{B}} \neq \varnothing$.

From CHP to $P_{\mathbb{B}}$-realizability

The idea is to represent an arithmetic computation in a 'natural' form.

The main construction

- R_1, R_2 — regular languages.
- How to check that there exists an integer n such that

$$\mathrm{Card}(\{w : |w| = n \ \wedge \ w \in R_1\}) = \mathrm{Card}(\{w : |w| = n \ \wedge \ w \in R_2\})? \qquad (\clubsuit)$$

- Regular expression

$$E = \#((R_1 \cap R_2)\#)^*((R_1 \setminus R_2)\#(R_2 \setminus R_1)\#)^*((\overline{R_1} \cap \overline{R_2})\#)^*$$

- $(\clubsuit)$ is equivalent to $E \cap P_{\mathbb{B}} \neq \varnothing$.

From CHP to $P_{\mathbb{B}}$-realizability

The idea is to represent an arithmetic computation in a ‘natural’ form.

The main construction

- R_1, R_2 — regular languages.
- How to check that there exists an integer n such that

$$\mathrm{Card}(\{w : |w| = n \ \wedge \ w \in R_1\}) = \mathrm{Card}(\{w : |w| = n \ \wedge \ w \in R_2\})? \quad (\clubsuit)$$

- Regular expression

$$E = \#((R_1 \cap R_2)\#)^*((R_1 \setminus R_2)\#(R_2 \setminus R_1)\#)^*((\overline{R_1} \cap \overline{R_2})\#)^*$$

- $(\clubsuit)$ is equivalent to $E \cap P_{\mathbb{B}} \neq \varnothing$.

More examples of relation between RR and linear algebra

undecidable	track product of the periodic and permutation filter	
unknown	permutation filter	
decidable	surjective filter	injective filter

Surjective filter

Definition

$S_{\mathbb{B}}$ consists of words of the form

$$\#w_1\#w_2\#\dots w_N\#,$$

where

- $w_i \in \{0,1\}^*$ are blocks,
- $|w_i| = n$, $i = 1, 2, \dots, N$, n is the block rank,
- each binary word of length n is a block.

Examples

$$\#00\#00\#11\#10\#01\# \in S_{\mathbb{B}}$$
$$\#10\#11\#10\#00\#01\# \in S_{\mathbb{B}}$$
$$\#10\#01\#00\#01\#11\# \in S_{\mathbb{B}}$$

Injective filter

Definition

$I_{\mathbb{B}}$ consists of words of the form

$$\#w_1\#w_2\#\dots w_N\#,$$

where

- $w_i \in \{0,1\}^*$ are blocks,
- $|w_i| = n$, $i = 1, 2, \dots, N$, n is the block rank,
- $w_i \neq w_j$ for $i \neq j$.

Examples

$$\#00\#10\#01\# \in I_{\mathbb{B}}$$

$$\#101\#111\#001\#010\# \in I_{\mathbb{B}}$$

$$\#1000\#0110\#0000\#1111\# \in I_{\mathbb{B}}$$

Decidability results

Theorem

$I_{\mathbb{B}}$-realizability problem is decidable.
$S_{\mathbb{B}}$-realizability problem is decidable.

Proofs are based on converting $I_{\mathbb{B}}$-realizability problem (resp., $S_{\mathbb{B}}$-realizability problem) to a problem about orbits.

Decidability results

Theorem

$I_{\mathbb{B}}$-realizability problem is decidable.
$S_{\mathbb{B}}$-realizability problem is decidable.

Proofs are based on converting $I_{\mathbb{B}}$-realizability problem (resp., $S_{\mathbb{B}}$-realizability problem) to a problem about orbits.

An undecidable problem

Zero in the Upper Right Corner Problem (ZURC)

INPUT: $A_1, \ldots, A_N$ are $D \times D$ integer matrices.
OUTPUT: 'yes' if there exists a sequence $j_1, \ldots, j_\ell$ such that

$$(A_{j_1} A_{j_2} \ldots A_{j_\ell})_{1D} = 0 \quad \text{and}$$

'no' otherwise.

Theorem (Bell, Potapov, 2006)

The ZURC problem is undecidable for $N = 2$ and $D = 18$.

The ZURC problem is reduced to the regular realizability problem for the track product of periodic and permutation filters.

An undecidable problem

Zero in the Upper Right Corner Problem (ZURC)

INPUT: $A_1, \ldots, A_N$ are $D \times D$ integer matrices.
OUTPUT: 'yes' if there exists a sequence $j_1, \ldots, j_\ell$ such that

$$(A_{j_1} A_{j_2} \ldots A_{j_\ell})_{1D} = 0 \quad \text{and}$$

'no' otherwise.

Theorem (Bell, Potapov, 2006)

The ZURC problem is undecidable for $N = 2$ and $D = 18$.

The ZURC problem is reduced to the regular realizability problem for the track product of periodic and permutation filters.

An undecidable problem

Zero in the Upper Right Corner Problem (ZURC)

INPUT: $A_1, \dots, A_N$ are $D \times D$ integer matrices.
OUTPUT: 'yes' if there exists a sequence $j_1, \dots, j_\ell$ such that

$$(A_{j_1} A_{j_2} \dots A_{j_\ell})_{1D} = 0 \quad \text{and}$$

'no' otherwise.

Theorem (Bell, Potapov, 2006)

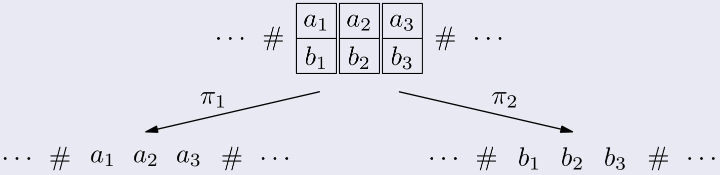
The ZURC problem is undecidable for $N = 2$ and $D = 18$.

The ZURC problem is reduced to the regular realizability problem for the track product of periodic and permutation filters.

Track product

For languages $L_1 \subset (\{\#\} \cup \Sigma_1)^*$, $L_2 \subset (\{\#\} \cup \Sigma_2)^*$ the track product $L_1 \| L_2 \subset (\{\#\} \cup \Sigma_1 \times \Sigma_2)^*$.

Projections



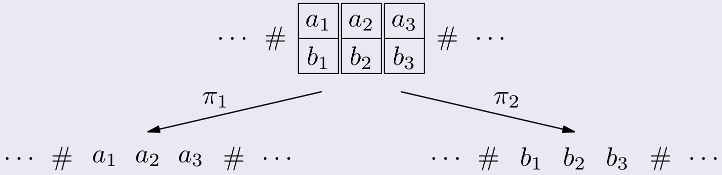
Definition of $L_1 \| L_2$

$$L_1 \| L_2 = \{ w \in (\{\#\} \cup \Sigma_1 \times \Sigma_2)^* \mid \pi_1 w \in L_1;\ \pi_2 w \in L_2 \}$$

Track product

For languages $L_1 \subset (\{\#\} \cup \Sigma_1)^*$, $L_2 \subset (\{\#\} \cup \Sigma_2)^*$ the track product $L_1 \| L_2 \subset (\{\#\} \cup \Sigma_1 \times \Sigma_2)^*$.

Projections



Definition of $L_1 \| L_2$

$$L_1 \| L_2 = \{w \in (\{\#\} \cup \Sigma_1 \times \Sigma_2)^* \mid \pi_1 w \in L_1;\ \pi_2 w \in L_2\}$$

Track product of periodic and permutation filters

Definitions

- Periodic filter $\mathrm{Per}_\Sigma \subset (\{\#\} \cup \Sigma)^*$ consists of words of the form
$$\#w\#w\#\ldots w\#,$$
where $w \in \{0,1\}^*$.
- Definition of the permutation filter P_Σ over the alphabet $\{\#\} \cup \Sigma$ is similar to the binary case.

Theorem

ZURC $\leq_m$ $(\mathrm{Per}_{\Sigma_1} \| P_{\Sigma_2})$*-regular realizability for* $|\Sigma_1| = 2$, $|\Sigma_2| = 648$.

Informally, the periodic part is to represent a sequence of matrices and the permutation part is to encode the condition that the URC entry is 0.

Track product of periodic and permutation filters

Definitions

- Periodic filter $\mathrm{Per}_\Sigma \subset (\{\#\} \cup \Sigma)^*$ consists of words of the form
$$\#w\#w\#\ldots w\#,$$
where $w \in \{0,1\}^*$.
- Definition of the permutation filter P_Σ over the alphabet $\{\#\} \cup \Sigma$ is similar to the binary case.

Theorem

ZURC $\leq_m (\mathrm{Per}_{\Sigma_1} \| P_{\Sigma_2})$*-regular realizability for* $|\Sigma_1| = 2$, $|\Sigma_2| = 648$.

Informally, the periodic part is to represent a sequence of matrices and the permutation part is to encode the condition that the URC entry is 0.