Resynchronizing Classes of Word Relations

María Emilia Descotte [1]
LaBRI

17e Journées Montoises d'informatique théorique
September 12th 2018

[1]Joint work with D. Figueira and G. Puppis

Languages vs. relations

Languages **Relations**

Languages vs. relations

Languages		**Relations**
• Finite monoids	$\rightsquigarrow$	• REC

Languages vs. relations

Languages		**Relations**
• Finite monoids	$\rightsquigarrow$	• REC
• NFA's	$\rightsquigarrow$	• REG

Languages vs. relations

Languages		**Relations**
• Finite monoids	$\rightsquigarrow$	• REC
• NFA's	$\rightsquigarrow$	• REG
• Regular expressions	$\rightsquigarrow$	• RAT

Languages vs. relations

Languages		**Relations**
• Finite monoids	$\rightsquigarrow$	• REC
$\parallel$		$\subsetneq$
• NFA's	$\rightsquigarrow$	• REG
$\parallel$		$\subsetneq$
• Regular expressions	$\rightsquigarrow$	• RAT

Synchronized pairs of words (over a fixed alphabet $\mathbb{A}$)

Synchronizing pairs of words

A synchronization of (w_1, w_2) is a word over $\mathbf{2} \times \mathbb{A}$ so that the projection on $\mathbb{A}$ of positions labeled i is exactly w_i for $i = 1, 2$.

Synchronized pairs of words (over a fixed alphabet $\mathbb{A}$)

Synchronizing pairs of words

A synchronization of (w_1, w_2) is a word over $\mathbf{2} \times \mathbb{A}$ so that the projection on $\mathbb{A}$ of positions labeled i is exactly w_i for $i = 1, 2$.

Example

$(1, a)(1, b)(2, a)$ and $(1, a)(2, a)(1, b)$ synchronize (ab, a).

Synchronized pairs of words (over a fixed alphabet $\mathbb{A}$)

Synchronizing pairs of words

A synchronization of (w_1, w_2) is a word over $\mathbf{2} \times \mathbb{A}$ so that the projection on $\mathbb{A}$ of positions labeled i is exactly w_i for $i = 1, 2$.

Example

$(1, a)(1, b)(2, a)$ and $(1, a)(2, a)(1, b)$ synchronize (ab, a).

Every word $w \in (\mathbf{2} \times \mathbb{A})^*$ is a synchronization of a unique pair (w_1, w_2) that we denote $[\![w]\!]$.

$$[\![(1, a)(1, b)(2, a)]\!] = [\![(1, a)(2, a)(1, b)]\!] = (ab, a).$$

Synchronized relations

Synchronizing relations

We lift this notion to languages $L \subseteq (\mathbf{2} \times \mathbb{A})^*$

$$[\![L]\!] = \{[\![w]\!] \mid w \in L\}$$

Synchronized relations

Synchronizing relations

We lift this notion to languages $L \subseteq (\mathbf{2} \times \mathbb{A})^*$

$$[\![L]\!] = \{[\![w]\!] \mid w \in L\}$$

Example

$\mathbb{A} = \{a, b\}$, $L = ((1, a)(2, a) \cup (1, a)(2, b) \cup (1, b)(2, a) \cup (1, b)(2, b))^*$,

$$[\![L]\!] = \{(w_1, w_2) \mid |w_1| = |w_2|\}.$$

C-controlled relations

Restrictions on the shape of the projection over **2**

$\rightsquigarrow$

Infinitely many different classes of relations.

C-controlled relations

Restrictions on the shape of the projection over $\mathbf{2}$

$\rightsquigarrow$

Infinitely many different classes of relations.

C-controlled words and languages

$C \subseteq \mathbf{2}^*$ regular

-$w \in (\mathbf{2} \times \mathbb{A})^*$ is C-controlled if its projection over $\mathbf{2}$ belongs to C.

-$L \subseteq (\mathbf{2} \times \mathbb{A})^*$ is C-controlled if all its words are.

C-controlled relations

Restrictions on the shape of the projection over $\mathbf{2}$

$\rightsquigarrow$

Infinitely many different classes of relations.

C-controlled words and languages

$C \subseteq \mathbf{2}^*$ regular
-$w \in (\mathbf{2} \times \mathbb{A})^*$ is C-controlled if its projection over $\mathbf{2}$ belongs to C.
-$L \subseteq (\mathbf{2} \times \mathbb{A})^*$ is C-controlled if all its words are.

Examples

-Every $w \in (\mathbf{2} \times \mathbb{A})^*$ is $\mathbf{2}^*$-controlled,
-$(1, a)(1, b)(2, a)$ is 1^*2^*-controlled,
-$(1, a)(2, a)(1, b)$ isn't 1^*2^*-controlled,
-L (previous slide) is $(12)^*$-controlled.

C-controlled relations

Restrictions on the shape of the projection over $\mathbf{2}$

$\rightsquigarrow$

Infinitely many different classes of relations.

C-controlled words and languages

$C \subseteq \mathbf{2}^*$ regular
-$w \in (\mathbf{2} \times \mathbb{A})^*$ is C-controlled if its projection over $\mathbf{2}$ belongs to C.
-$L \subseteq (\mathbf{2} \times \mathbb{A})^*$ is C-controlled if all its words are.

Examples

-Every $w \in (\mathbf{2} \times \mathbb{A})^*$ is $\mathbf{2}^*$-controlled,
-$(1, a)(1, b)(2, a)$ is 1^*2^*-controlled,
-$(1, a)(2, a)(1, b)$ isn't 1^*2^*-controlled,
-L (previous slide) is $(12)^*$-controlled.

C-controlled relations

Given a regular language $C \subseteq \mathbf{2}^*$

$$\textsc{Rel}(C) = \{[\![L]\!] \mid L \text{ is reg. and } C\text{-controlled}\}$$

C-controlled relations

Restrictions on the shape of the projection over $\mathbf{2}$

$\rightsquigarrow$

Infinitely many different classes of relations.

C-controlled words and languages

$C \subseteq \mathbf{2}^*$ regular
-$w \in (\mathbf{2} \times \mathbb{A})^*$ is C-controlled if its projection over $\mathbf{2}$ belongs to C.
-$L \subseteq (\mathbf{2} \times \mathbb{A})^*$ is C-controlled if all its words are.

Examples

-Every $w \in (\mathbf{2} \times \mathbb{A})^*$ is $\mathbf{2}^*$-controlled,
-$(1, a)(1, b)(2, a)$ is 1^*2^*-controlled,
-$(1, a)(2, a)(1, b)$ isn't 1^*2^*-controlled,
-L (previous slide) is $(12)^*$-controlled.

C-controlled relations

Given a regular language $C \subseteq \mathbf{2}^*$

$$\textsc{Rel}(C) = \{[\![L]\!] \mid L \text{ is reg. and } C\text{-controlled}\}$$

Examples

-$\textsc{Rel}(1^*2^*)$ =REC,
-$\textsc{Rel}((12)^*(1^* \cup 2^*))$ =REG,
-$\textsc{Rel}(\mathbf{2}^*)$ =RAT.

Class Containment Problem

Class Containment Problem	
Input:	Two regular languages $C, D \subseteq \mathbf{2}^*$
Output:	Is $\textsc{Rel}(C) \subseteq \textsc{Rel}(D)$?

Class Containment Problem

Class Containment Problem	
Input:	Two regular languages $C, D \subseteq \mathbf{2}^*$
Output:	Is $\textsc{Rel}(C) \subseteq \textsc{Rel}(D)$?

Examples

-If $C \subseteq D$, then $\textsc{Rel}(C) \subseteq \textsc{Rel}(D)$,
-$\textsc{Rel}(1^*2^*) \subseteq \textsc{Rel}((12)^*(1^* \cup 2^*))$,
-$\textsc{Rel}((12)^*(1^* \cup 2^*)) \not\subseteq \textsc{Rel}(1^*2^*)$,
-$\textsc{Rel}(1^*2^*) = \textsc{Rel}(2^*1^*)$,
-$\textsc{Rel}((12)^*) = \textsc{Rel}((21)^*)$.

Previous work[2]

Decidability and complexity

The problem is decidable for $\textsc{Rel}(D)$ =REC, REG or Length-pres.

[2] D. Figueira and L. Libkin. Synchronizing relations on words. *ACM Transactions on Computer Systems, 2015.*

Previous work[2]

Decidability and complexity

The problem is decidable for $\textsc{Rel}(D)$ =REC, REG or Length-pres.

Resynchronization

The proof is constructive in terms of the automaton:

> *Given a C-controlled language L, one can effectively construct a D-controlled language L' such that $[\![L]\!] = [\![L']\!]$.*

[2] D. Figueira and L. Libkin. Synchronizing relations on words. *ACM Transactions on Computer Systems, 2015.*

Our contribution

We prove that the Class Containment Problem is decidable for arbitrary C and D and, in case of positive answer, we give an effective method for resynchronizing relations.

Our contribution

We prove that the Class Containment Problem is decidable for arbitrary C and D and, in case of positive answer, we give an effective method for resynchronizing relations.

Proof idea

Step 1: Rewrite C and D as finite unions of *simple languages*.

Step 2: Characterization for simple languages.

Step 3: Induction on the amount of disjuncts in the unions.

Step 1: Decomposition into simple languages

Concat-star languages

$$C_1^* u_1 \cdots C_n^* u_n$$

with $C_1, \ldots, C_n$ regular languages, $u_1, \ldots, u_n$ words.

Step 1: Decomposition into simple languages

Concat-star languages

$$C_1^* u_1 \cdots C_n^* u_n$$

with $C_1, \ldots, C_n$ regular languages, $u_1, \ldots, u_n$ words.

Simple languages

Concat-star languages of star-height 1 + extra restrictions.

Step 1: Decomposition into simple languages

Concat-star languages

$$C_1^* u_1 \cdots C_n^* u_n$$

with $C_1, \ldots, C_n$ regular languages, $u_1, \ldots, u_n$ words.

Simple languages

Concat-star languages of star-height 1 + extra restrictions.

Examples

- $1^*(12)^*2^*12$ ✓
- $1^*(12 \cup 1)^*(112)^*1$ ✓
- $(1^*2)^*2^*11$ ✗
- $(12)^*1^* \cup (12)^*2^*$ ✗

Step 1: Decomposition into simple languages

Every regular language is a finite union of concat-star languages.

Step 1: Decomposition into simple languages

Every regular language is a finite union of concat-star languages.

Every concat-star language is *Rel-equivalent* to a finite union of concat-star languages of star-height 1.

Step 1: Decomposition into simple languages

Every regular language is a finite union of concat-star languages.

Every concat-star language is *Rel-equivalent* to a finite union of concat-star languages of star-height 1.

Every concat-star language of star-height 1 is *Rel-equivalent* to a finite union of simple languages.

Step 2: Characterization for simple languages

Parikh ratio

-$w \in \mathbf{2}^* \setminus \{\varepsilon\}$, $\rho(w) = \frac{|w|_1}{|w|}$.
-$C \subseteq \mathbf{2}^*$, $\rho(C) = \{\rho(w) \mid w \in C \setminus \{\varepsilon\}\} \subseteq [0,1]_{\mathbb{Q}}$.

Step 2: Characterization for simple languages

Parikh ratio

-$w \in \mathbf{2}^* \setminus \{\varepsilon\}$, $\rho(w) = \frac{|w|_1}{|w|}$.
-$C \subseteq \mathbf{2}^*$, $\rho(C) = \{\rho(w) \mid w \in C \setminus \{\varepsilon\}\} \subseteq [0,1]_{\mathbb{Q}}$.

Synchronizing morphisms

$C = C_1^* u_1 \cdots C_n^* u_n$, $D = D_1^* v_1 \cdots D_m^* v_m$. $C \xrightarrow{s.m.} D$ is

$$f : [1, \ldots, n] \to [1, \ldots, m] \text{ s.t.}$$

i) f is monotonic and
ii) $\rho(C_i^*) \subseteq \rho(D_{f(i)}^*)$ for all $i = 1, \ldots, n$.

Step 2: Characterization for simple languages

Parikh ratio

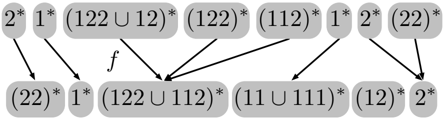
-$w \in \mathbf{2}^* \setminus \{\varepsilon\}$, $\rho(w) = \frac{|w|_1}{|w|}$.
-$C \subseteq \mathbf{2}^*$, $\rho(C) = \{\rho(w) \mid w \in C \setminus \{\varepsilon\}\} \subseteq [0,1]_{\mathbb{Q}}$.

Synchronizing morphisms

$C = C_1^* u_1 \cdots C_n^* u_n$, $D = D_1^* v_1 \cdots D_m^* v_m$. $C \xrightarrow{s.m.} D$ is

$$f : [1,\ldots,n] \to [1,\ldots,m] \text{ s.t.}$$

i) f is monotonic and
ii) $\rho(C_i^*) \subseteq \rho(D_{f(i)}^*)$ for all $i = 1,\ldots,n$.



Step 2: Characterization for simple languages

Proposition

For all simple languages $C, D \subseteq \mathbf{2}^*$,

$$\textsc{Rel}(C) \subseteq \textsc{Rel}(D) \text{ iff } \pi(C) \subseteq \pi(D) \text{ and } C \xrightarrow{s.m.} D.$$

Step 2: Characterization for simple languages

Proposition

For all simple languages $C, D \subseteq \mathbf{2}^*$,

$$\textsc{Rel}(C) \subseteq \textsc{Rel}(D) \text{ iff } \pi(C) \subseteq \pi(D) \text{ and } C \xrightarrow{s.m.} D.$$

Examples

-$\textsc{Rel}((12)^*(112)^*) \subseteq \textsc{Rel}((12 \cup 11122)^*(121)^*1^*2^*)$,
-$\textsc{Rel}((112)^*(12)^*) \not\subseteq \textsc{Rel}((12 \cup 11122)^*(121)^*1^*2^*)$.

Step 3: Dealing with unions

Unions on the left

$$\textsc{Rel}(C_1 \cup C_2) \subseteq \textsc{Rel}(D) \text{ iff}$$
$$\textsc{Rel}(C_1) \subseteq \textsc{Rel}(D) \text{ and } \textsc{Rel}(C_2) \subseteq \textsc{Rel}(D).$$

Step 3: Dealing with unions

Unions on the left

$$\text{Rel}(C_1 \cup C_2) \subseteq \text{Rel}(D) \text{ iff}$$
$$\text{Rel}(C_1) \subseteq \text{Rel}(D) \text{ and } \text{Rel}(C_2) \subseteq \text{Rel}(D).$$

Unions on the right

For C simple and $D = \bigcup_j D_j$ a finite union of simple languages, the following are equivalent:

i) $\text{Rel}(C) \subseteq \text{Rel}(D)$,

ii) $\pi(C) \subseteq \pi(D)$, $\exists j$ with $C \xrightarrow{s.m.} D_j$ and in addition, if C is heterogeneous, then $\text{Rel}(C \setminus [D_j]_\pi) \subseteq \text{Rel}(\bigcup_{j' \neq j} D_{j'})$.

$$[D_j]_\pi = \pi^{-1}(\pi(D_j)) = \{w \mid \pi(w) \in \pi(D_j)\}.$$

Future work

- Our proof gives an effective algorithm to resynchronize relations. We would like to determine the exact complexity.

Future work

- Our proof gives an effective algorithm to resynchronize relations. We would like to determine the exact complexity.
- What about k-ary relations? Step 1 relies on geometric arguments that only hold in dimension 2.

Thanks for your attention!