Schema for parallel insertion and deletion

Lila Kari, Shinnosuke Seki

Department of Computer Science, University of Western Ontario

Developments in Language Theory
August 19th, 2010

Notation

Σ	alphabet
Σ^*	the set of all words over Σ
u, v, w	words
L, L_1, L_2, L_3	given languages
R, R_1, R_2, R_3	given regular languages
X, Y	unknown variables
$+$	union of sets
L^c	complement of L, i.e., $L^c = \Sigma^* \setminus L$
2^L	power set of L

Parallel operations

Example ([Kari91])

Parallel insertion $\Leftarrow$ is defined as follows: for a word $u = a_1 a_2 \cdots a_n (a_i \in \Sigma)$ and a language L,

$$u \Leftarrow L = La_1 La_2 L \cdots La_{n-1} La_n L.$$

Question

How to control parallel insertion (where to insert L)?

p-schema-based insertion

Let $\mathfrak{F} = \{(u_1, u_2, \ldots, u_{n-1}, u_n) \mid n \geq 1, u_1, \ldots, u_n \in \Sigma^*\}$.

Definition

For $f = (u_1, u_2, \ldots, u_n) \in \mathfrak{F}$, insertion $\longleftarrow_f$ based on f is defined as:

$$u \longleftarrow_f L = \begin{cases} u_1 L u_2 L \cdots u_{n-1} L u_n & \text{if } u = u_1 u_2 \cdots u_n \\ \emptyset & \text{otherwise.} \end{cases}$$

We call $F \subseteq \mathfrak{F}$ a *p-schema* because it can specify how to parallel-insert a language L into a word u.
We can extend $\longleftarrow_F$ naturally into an operation between languages as:

$$L_1 \longleftarrow_F L_2 = \bigcup_{u \in L_1, f \in F} u \longleftarrow_f L_2.$$

Various instances of *p*-schema-based insertion

Syntactic and Semantic instances of *p*-schema-based insertion $L_1 \leftarrow_F L_2$ include

operation	*p*-schema
catenation $L_1 L_2$	$\Sigma^* \times \{\lambda\}$
reverse catenation $L_2 L_1$	$\{\lambda\} \times \Sigma^*$
insertion $\{xL_2y \mid xy \in L_1\}$	$\Sigma^* \times \Sigma^*$
parallel insertion $L_1 \Leftarrow L_2$	$\bigcup_{n \geq 0} \left(\{\lambda\} \times \underbrace{\Sigma \times \cdots \times \Sigma}_{n \text{ times}} \times \{\lambda\} \right)$
inserting exactly 2 L's	$\Sigma^* \times \Sigma^* \times \Sigma^*$
(x, y)-contextual insertion	$\Sigma^* x \times y \Sigma^*$
Parallel insertion next to $b \in \Sigma$	$\{(u_1, \ldots, u_n) \mid n \geq 1, u_1, \ldots, u_n \in (\Sigma \setminus \{b\})^* b\}$.

p-schema-based deletion

Definition

For $f = (u_1, u_2, \ldots, u_n) \in \mathfrak{F}$, deletion $\rightarrowtail_f$ based on f is defined as:

$$w \rightarrowtail_f L = \begin{cases} \{u_1 u_2 \cdots u_n\} & \text{if } w \in u_1 L u_2 L \cdots u_{n-1} L u_n \\ \emptyset & \text{otherwise} \end{cases}$$

$\rightarrowtail_f$ is also extended to an operation between languages as follows:

$$L_1 \rightarrowtail_F L_2 = \bigcup_{w \in L_1, f \in F} w \rightarrowtail_f L_2.$$

Classes of p-schemata

Definition

For a p-schema F, its schema language $\psi(F)$ is defined over $\Sigma \cup \{\#\}$ as:

$$\psi(F) = \{u_1\#u_2\#\cdots u_{n-1}\#u_n \mid (u_1, u_2, \ldots, u_{n-1}, u_n) \in F\}.$$

Let $\mathcal{C}$ be a class of languages over $\Sigma \cup \{\#\}$. We say that a p-schema F is in $\mathcal{C}$ if $\psi(F) \in \mathcal{C}$.

regular p-schema

A p-schema F is regular if $\psi(F)$ is regular.

Objectives

Question

Is it decidable whether language equations of the following forms:

1. $X \leftarrowtail_F L_2 = L_3$ and $X \rightarrowtail_F L_2 = L_3$
2. $L_1 \leftarrowtail_X L_2 = L_3$ and $L_1 \rightarrowtail_X L_2 = L_3$
3. $L_1 \leftarrowtail_F X = L_3$ and $L_1 \rightarrowtail_F X = L_3$

have a solution or not?

Existence of maximum solution and decision algorithm I

Do language equations of the previous forms have the (unique) maximum solution if they have a solution?

No for $L_1 \longleftarrow_F X = L_3$ and $L_1 \rightarrowtail_F X = L_3$

Yes for the others

algorithm [Kari91]

1. Construct the candidate of maximum solution,
2. Substitute it into the equation,
3. Test whether both sides become equal.

Existence of maximum solution and decision algorithm II

Corollary

For regular languages R_1, R_2, R_3 and a regular p-schema F, it is decidable whether

- $X \leftarrowtail_F R_2 = R_3$
- $X \rightarrowtail_F R_2 = R_3$
- $R_1 \leftarrowtail_X R_2 = R_3$
- $R_1 \rightarrowtail_X R_2 = R_3$

has a solution or not.

Different approach to language equations

In contrast, $L_1 \leftarrowtail_F X = L_3$ and $L_1 \rightarrowtail_F X = L_3$ may not have the unique maximum solution but multiple maximal solutions.

Example

Let $L_{\text{even}} = \{a^{2m} \mid m \geq 0\}$, $L_{\text{odd}} = \{a^{2n+1} \mid n \geq 0\}$, and $F = \Sigma^* + (\Sigma^* \times \Sigma^* \times \Sigma^*)$.

$$\begin{aligned} L_{\text{even}} \leftarrowtail_F L_{\text{even}} &= L_{\text{even}} \leftarrowtail_F L_{\text{odd}} = L_{\text{even}}; \\ L_{\text{even}} \rightarrowtail_F L_{\text{even}} &= L_{\text{even}} \rightarrowtail_F L_{\text{odd}} = L_{\text{even}}. \end{aligned}$$

Actually, both L_{even} and L_{odd} are maximal solutions to $L_{\text{even}} \leftarrowtail_F X = L_{\text{even}}$ and $L_{\text{even}} \rightarrowtail_F X = L_{\text{even}}$.

We propose another approach to solving these equations based on the notion of syntactic congruence.

Syntactic congruence

Definition

For a language L, the syntactic congruence $\equiv_L$ is an equivalence relation defined as: for $u, v \in \Sigma^*$,

$$u \equiv_L v \stackrel{\text{def}}{\iff} \forall x, y \in \Sigma^*, xuy \in L \iff xvy \in L$$

Theorem ([Rabin and Scott, 1959])

The index of $\equiv_L$ is finite iff L is regular.

Theorem

For a regular language R, each equivalence class in $\Sigma^*/\equiv_R$ is a regular language.

Solving $L_1 \leftarrow\!\!\prec_F X = L_3$ I

Lemma

Let $L_1, L_3 \subseteq \Sigma^*$. Then for any $w \in \Sigma^*$ and $L_2 \subseteq \Sigma^*$,

$$(L_1 \leftarrow\!\!\prec_F (\{w\}+L_2)) \cap L_3^c \neq \emptyset \iff (L_1 \leftarrow\!\!\prec_F ([w]_{\equiv_{L_3^c}}+L_2)) \cap L_3^c \neq \emptyset.$$

Assume that $u = u_1u_2u_3u_4 \in L_1$, $(u_1, u_2, u_3, u_4) \in F$, $w_1, w_2 \in [w]_{\equiv_L}$, and $v \in L_2$ s.t. $u_1 w_1 u_2 v u_3 w_2 u_4 \in L_3^c$. Then,

$$\begin{aligned} u_1 w_1 u_2 v u_3 w_2 u_4 \in L_3^c &\iff u_1 w u_2 v u_3 w_2 u_4 \in L_3^c \\ &\iff u_1 w u_2 v u_3 w u_4 \in L_3^c. \end{aligned}$$

Observe that this word is in $L_1 \leftarrow\!\!\prec_F (\{w\} + L_2)$.

Solving $L_1 \longleftarrow_F X = L_3$ II

Syntactic solution

For a language L, a solution to a given equation is syntactic w.r.t. L if it is a union of equivalence classes in $\Sigma^* / \equiv_L$.

Proposition

For languages L_1, L_3, $L_1 \longleftarrow_F X = L_3$ has a solution iff it has a syntactic solution w.r.t. L_3.

To decide whether $L_1 \longleftarrow_F X = L_3$, therefore, it suffices to check whether or not it has a syntactic solution w.r.t. L_3. Recall that if L_3 is regular, then

- there exist at most finite numbers of syntactic solutions, (the index of $\equiv_{L_3}$ is finite)

Solving $L_1 \leftarrowtail_F X = L_3$ III

- such syntactic solutions are regular, and
- solely determined by L_3

Proposition

For regular languages R_1, R_3 and a regular p-schema F, it is decidable whether $R_1 \leftarrowtail_F X = R_3$ has a solution.

Note that all maximal solutions to $L_1 \leftarrowtail_F X = L_3$ are syntactic w.r.t. L_3.

Theorem

For regular languages R_1, R_3 and a regular p-schema F, the set of all maximal solutions to $R_1 \leftarrowtail_F X = R_3$ is effectively constructible.

Solving the inequality $L_1 \longleftarrow_F X \subseteq L_3$

Theorem

For regular languages R_1, R_3 and a regular p-schema F, the set of all maximal solutions to $R_1 \longleftarrow_F X \subseteq R_3$ is effectively constructible.

An application

Note that $L^* = \{\lambda\} \longleftarrow_{\mathfrak{F}} L$. Due to the above theorem, for a given regular language R, we can construct all the maximal languages X such that $X^* \subseteq R$.

Solving multiple-variables equations with p-schema based insertion

Remember that syntactic solutions of $L_1 \leftarrow\!\!\prec_F Y = L_3$ are solely determined by L_3.

Theorem

For a regular language R_3 and p-schema F, it is decidable whether $X \leftarrow\!\!\prec_F Y = R_3$ has a solution.

Proof.

N.B. $|\Sigma^* / \equiv_{R_3}|$ is finite. So for each candidate R_c of syntactic solutions, let us check whether $X \leftarrow\!\!\prec_F R_c = R_3$ has a solution. □

Theorem

For regular languages R_1, R_3, it is decidable whether $R_1 \leftarrow\!\!\prec_X Y = R_3$ has a solution.

Solving $L_1 \rightarrowtail_F X = L_3$ I

Lemma

Let $L_1 \subseteq \Sigma^*$. For any word $w \in \Sigma^*$ and a language $L_2 \subseteq \Sigma^*$,

$$L_1 \rightarrowtail_F (\{w\} + L_2) = L_1 \rightarrowtail_F ([w]_{\equiv_{L_1}} + L_2).$$

Corollary

$$(L_1 \rightarrowtail_F (\{w\}+L_2)) \cap L_3^c \neq \emptyset \iff (L_1 \rightarrowtail_F ([w]_{\equiv_{L_1}}+L_2)) \cap L_3^c \neq \emptyset.$$

Proposition

For languages L_1, L_3, the equation $L_1 \rightarrowtail_F X = L_3$ has a solution iff it has a syntactic solution w.r.t. L_1.

Solving $L_1 \rightarrowtail_F X = L_3$ II

Lemma

For an arbitrary a complete set $\mathfrak{R}(L_1)$ *of representatives of* $\Sigma^* / \equiv_{L_1}$,

$$L_1 \leftarrowtail_F L_2 = L_1 \leftarrowtail_F \{w \in \mathfrak{R}(L_1) \mid w \in L_2\}.$$

Theorem

For regular languages R_1, R_3, a regular p-schema F, a complete system $\mathfrak{R}(R_1)$ of representatives of $\Sigma^* / \equiv_{R_1}$, the set of all solutions to $R_1 \rightarrowtail_F X = R_3$ which are a subset of $\mathfrak{R}(R_1)$ is effectively constructible.

Solving $L_1 \rightarrowtail_F X = L_3$ III

Corollary

For regular languages R_1, R_3, and a regular p-schema F, the set of all syntactic solutions to $R_1 \rightarrowtail_F X = R_3$ is effectively constructible, and hence, so is the set of its maximal solutions.

Corollary

For regular languages R_1, R_3, and a regular p-schema F, the set of all minimal solutions to $R_1 \rightarrowtail_F X = R_3$ modulo $\equiv_{R_1}$ is effectively constructible.

Solving multiple-variable equations with *p*-schema based deletion

Recall that syntactic solutions to $L_1 \rightarrowtail_F Y = L_3$ is determined by L_1 (not L_3).

Theorem

For regular languages R_1, R_3, it is decidable whether $R_1 \rightarrowtail_X Y = R_3$ has a solution.

Open problem

Is it decidable whether $X \rightarrowtail_F Y = R_3$ for a regular language R_3 and *p*-schema F?

Undecidability

Let $\mathrm{NCM}(1)$ be the class of languages accepted by a finite automaton augmented with 1 one-reversal counter.

Proposition

If one of L_1, L_3, F is in $\mathrm{NCM}(1)$, then it is undecidable whether $L_1 \leftarrowtail_F X = L_3$ ($L_1 \rightarrowtail_F X = L_3$) has a solution or not.

Conclusion

Contributions

1. *p*-schema-based insertion and deletion
2. algorithms to solve $L_1 \leftarrowtail_F X = L_3$ and $L_1 \rightarrowtail_F X = L_3$

Future works

1. Once we weaken the regularity condition on L_3, our algorithm does not work any more to solve $L_1 \leftarrowtail_F X = L_3$. For instance, if $L_3 \in \mathrm{DCM}(1)$, can we solve this equation?
2. Can we solve $X \rightarrowtail_F Y = R_3$ for a regular language R_3 and a regular *p*-schema F?

Apology

I sincerely apologize for the following 2 errors and any of your inconveniences caused by these.

1. Proposition 1 requires $k_2 = 0$
2. In Theorem 11, DPCM should be replaced with REG.

Thank you very much for listening so attentively.

References I

[Kari91] L. Kari,
On insertion and deletion in formal languages,
Ph.D. Thesis, University of Turku, 1991.

[Kari94] L. Kari,
On language equations with invertible operations,
Theoretical Computer Science, 132 (1994) 129-150.

[Rabin and Scott, 1959] M. Rabin and D. Scott,
Finite automata and their decision problems,
em IBM Journal of Research and Development, 3 (1959) 114-125.