The Simplest Language Where Equivalence of Finite Substitutions Is Undecidable

Michal Kunc

Masaryk University Brno

Outline

Our result — infinite systems of equalities between finite languages.

Interpretations of the result:

- equivalence of finite substitutions on a regular language
- equivalence of finite transducers
- systems of language equations

Main ideas of the proof.

The Result

Basic notions:

finite alphabet $A = \{a, b, \dots\}$ of letters

word over A ... finite sequence of letters from A

A^* ... the monoid of words over A with the operation of concatenation

$K, L \subseteq A^*$ languages over A

$K \cdot L = \{\, uv \mid u \in K,\ v \in L \,\}$

$K^n = \underbrace{K \cdots K}_{n}$

$\wp_{\mathrm{f}}(A^*)$ ... the monoid of all finite languages over A

Theorem:

There exists no algorithm deciding whether given finite languages K, L and M satisfy

$K^n \cdot M = L^n \cdot M$ for every non-negative integer n.

Equivalence of Finite Substitutions on a Regular Language

Finite substitution:

A, B ... finite alphabets

Any mapping $\varphi\colon A \longrightarrow \wp_{\mathrm{f}}(B^*)$ uniquely extends to a homomorphism $\varphi\colon A^* \longrightarrow \wp_{\mathrm{f}}(B^*)$.

$R \subseteq A^*$ fixed regular language (defined by a finite automaton)

Instance:

$\varphi, \psi\colon A^* \longrightarrow \wp_{\mathrm{f}}(B^*)$ finite substitutions

Question:

Does $\varphi(u) = \psi(u)$ hold for all $u \in R$?

Trivially decidable if R is finite.

Equivalence of Finite Substitutions on a Regular Language R

Easily decidable if R employs only one letter a:

R eventually periodic, p period of R

$a^n \in R$ arbitrary

$\varphi(a^n) = \psi(a^n) \implies \varphi(a^{np}) = (\varphi(a^n))^p = (\psi(a^n))^p = \psi(a^{np})$

sufficient to test for the first n periods of R

Our result:

Equivalence of finite substitutions on a^*b is undecidable.

($\varphi(a) = K$, $\psi(a) = L$, $\varphi(b) = \psi(b) = M$)

Previous results:

Lisovik 1997: undecidable on $a\{b, c\}^*d$

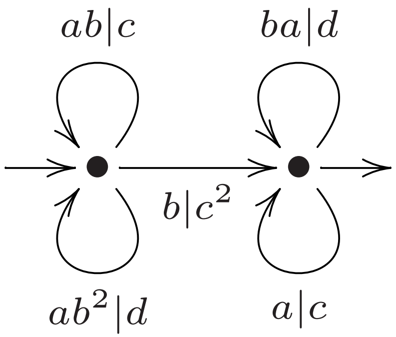
Karhumäki & Lisovik 2003: undecidable on ab^*c

Finite Transducer

A input alphabet, B output alphabet

- non-deterministic finite automaton with each transition labelled by both input and output word
- defines a binary relation between A^* and B^*

Example: $A = \{a, b\}, B = \{c, d\}$



the relation contains, e.g., (ab^3a, c^3d) and (ab^3a, dc^3)

Equivalence Problem for Finite Transducers

Instance:

finite transducers $\mathcal{A}$ and $\mathcal{B}$

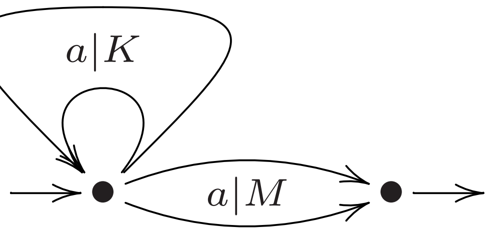
Question:

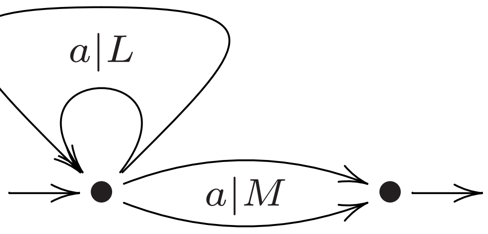
Do $\mathcal{A}$ and $\mathcal{B}$ define the same relation?

- Griffiths 1968: undecidable in general
- many positive and negative results for special cases
- Ibarra 1978, Lisovik 1979: undecidable for one-letter input alphabet

Our result:

Equivalence problem for two-state finite transducers with unary input alphabet and all transitions starting in the initial state is undecidable.





Equations over Words

- concatenation as operation, letters as constants
- finitely many variables
- for variables words are substituted
- for instance, solutions of equation $xba = abx$ are exactly $x = a(ba)^n$, where $n \in \mathbb{N}_0$
- PSPACE algorithm deciding satisfiability, $\mathrm{EXPTIME}$ algorithm finding all solutions
  (Makanin 1977, Plandowski 2006)
- Conjecture: Satisfiability problem is NP-complete.

Every (infinite) rational system of word equations (defined by a finite transducer) is algorithmically equivalent to some of its finite subsystems.

(Culik II & Karhumäki 1983, Albert & Lawrence 1985, Guba 1986)

Equations over Finite Languages

- concatenation of languages as operation, finite languages as constants
- finitely many variables X, Y, Z, ...
- for variables either finite or arbitrary languages are substituted

Singleton constants:

- satisfiability-equivalent to equations over words

  ($\mathrm{shortlex}$-minimal words of an arbitrary language solution form a word solution)

Satisfiability of language equations by arbitrary languages is undecidable for

- equations with finite constants, union and concatenation
- systems of equations with regular constants and concatenation (MK 2007)

  ($K \cdot X = X \cdot L$, $A^* \cdot X = A^*$)

Open questions:

- satisfiability of equations over finite languages with concatenation (and union)
- satisfiability of equations with finite constants and concatenation

What about infinite systems?

Rational Systems of Equations over Finite Languages

Our result:

There exists no algorithm deciding whether given finite languages form a solution of the rational system $\{\, X^n Z = Y^n Z \mid n \in \mathbb{N} \,\}$.

Consequences:

- Satisfiability of rational systems of equations over finite languages is undecidable.
  (even without variables: $\{\, K^n M = L^n M \mid n \in \mathbb{N} \,\}$)
- Rational systems of equations over finite languages need not be equivalent to finite systems.

Proof

- the same general idea as for ab^*c by Karhumäki and Lisovik
- reduction of the universality problem for blind counter automata with all states final

Blind counter automata:

- introduced by Greibach 1978
- non-deterministic finite automaton over $\{a, b\}$ + one counter assuming arbitrary integer values
- transitions read letters and possibly modify the counter value by one
- no information about the current value of the counter available to the automaton
- acceptance by zero-valued counter

Universality problem:

Does a given blind counter automaton accept all words?

undecidable even for automata with all states final (Lisovik 1991)

Proof

For each blind counter automaton we construct finite languages K, L and M such that

$$K^n M = L^n M \iff \text{the automaton accepts all words of length } n.$$

- alphabet . . . letters a, b and many additional auxiliary letters
- all computations of the automaton encoded into both $K^n M$ and $L^n M$

Representation of transitions:

- w . . . certain fixed word containing only auxiliary letters
- computations of the automaton encoded as words of the form $\{(wa)^3, (wb)^3\}^*$
- one copy of $(wa)^3$ stands for one transition reading a
- states and transitions represented by cutting w at certain points
- M serves for completing the final unfinished copy of w

Representation of the counter:

- number of copies of $(wa)^3$ and $(wb)^3$ in a word from $K^n M$ depends on:

  the number of transitions

  the final value of the counter
- one transition represented by:

  one word from K . . . counter not modified

  one half of a word from K . . . decrementation

  one and a half word from K . . . incrementation
- computation of length n returns counter to zero $\iff$

  $\iff$ the corresponding word from $\{(wa)^3, (wb)^3\}^n$ belongs to $K^n M$
- $L \supseteq K$ such that $L^n M$ contains in addition all words from $\{(wa)^3, (wb)^3\}^n$

How is this achieved?

- L contains initial words for building words from $\{(wa)^3, (wb)^3\}^n$
- in the case of the language ab^*c, these words can be put into the image of a
- in our case, these additional initial words can occur inside of $L^n M$
- many auxiliary words for compensating undesired occurrences (shifting by several letters)