Equivalence checking of prefix-free transducers and deterministic two-tape automata

V.A. Zakharov

Lomonosov Moscow State University

December 14, 2018

Preliminaries

A word over alphabet A is any finite sequence $w = a_1 a_2 \ldots a_k$ of letters in A . The empty word is denoted by ε .

Given a pair of words u and v, we write uv for their concatenation.

The set of all words over an alphabet A is denoted by A^* .

A language over A is any subset of A^* .

Concatenation of languages L_1 and L_2 is the language

$$L_1 L_2 = \{uv \ : \ u \in L_1, v \in L_2\}.$$

If $L_1 = \emptyset$ or $L_2 = \emptyset$ then $L_1 L_2 = \emptyset$.

A transduction over alphabets A and B is any subset of $A^* \times B^*$.

Real Time Finite Transducers

A Real Time Finite Transducer over an input alphabet Σ and an output alphabet Δ is a quadruple $\pi = \langle Q, q_0, F, \longrightarrow \rangle$, where

- Q is a finite set of states ,
- q_0 is an initial state ,
- $F \subseteq Q$ is a subset of final states , and
- $\longrightarrow$ is a finite transition relation of the type $Q \times \Sigma \times \Delta^* \times Q$.

We will write $\pi(q_0)$ to emphasize that q_0 is the initial state of π .

Transitions (q, a, u, q') in $\longrightarrow$ are depicted as $q \xrightarrow{a/u} q'$.

A run of π on an input word $w = a_1 a_2 \ldots a_n$ is any finite sequence of transitions $q \xrightarrow{a_1/u_1} q_1 \xrightarrow{a_2/u_2} \cdots \xrightarrow{a_{n-1}/u_{n-1}} q_{n-1} \xrightarrow{a_n/u_n} q'$.
The pair (w, u) , where $u = u_1 u_2 \ldots u_n$, is a label of this run.

We write $q \xrightarrow{w/u}_* q'$ when a transducer π has a run labeled with (w, u) from a state q to a state q' . If $q' \in F$ then a run is final .

A transduction relation realized by a transducer π at its state q is the set of pairs $TR(\pi, q) = \{(w, u) : q \xrightarrow{w/u}_* q', q' \in F\}$.

Real Time Finite Transducers

Transducers $\pi_1(q_1)$ and $\pi_2(q_2)$ are called equivalent ($\pi_1(q_1) \sim \pi_2(q_2)$ in symbols) iff $TR(\pi_1, q_1) = TR(\pi_2, q_2)$.

Equivalence checking problem for transducers is that of checking, given a pair of transducers π_1 and π_2 , whether $\pi_1 \sim \pi_2$ holds.

A transducer π is called

- deterministic if for every letter a and a state q it has at most one transition of the form $q \xrightarrow{a/u} q'$,
- k-ambiguous if for every input word w there is at most k final runs of π on w from the initial state q_0 ,
- k-valued if for every input word w the transduction relation $TR(\pi, q_0)$ contains at most k images of w ,
- of length-degree k if for every input word w , the number of distinct lengths of the images u of w in $Tr(\pi, q_0)$ is at most k .

Real Time Finite Transducers

Equivalence checking problem is undecidable for

- transducers with ε -transitions
  (Fisher P.S., Rozenberg A.L., 1966)
- real time transducers (Griffiths T., 1968)
- transducers over one-letter alphabet (Ibarra O., 1972).

Equivalence checking problem is decidable for

- deterministic transducers
  (Blattner M, Head T., 1979): PTime
- single-valued transducers
  (Schutzenberger M. P., 1977): PSpace
- unambiguous transducers (Gurari E., Ibarra O., 1983): PTime
- k-ambiguous transducers (Gurari E., Ibarra O., 1983)
- k-valued transducers
  (Culik K., Karhumaki J., 1986): Time $2^{O(n^2)}$
- transducers of length-degree k (Weber A., 1992): Time $2^{2^{2^n}}$

Two-tape finite automata

A Two-tape Finite State Automaton (2-FSA) over disjoint alphabets Σ and Δ is a 5-tuple $M = \langle S_1, S_2, s_0, F, \rightarrow \rangle$ such that

- S_1, S_2 is a partitioning of a finite set S of states ,
- $s_0 \in S_1$ is an initial state ,
- $F \subseteq S$ is a subset of final states , and
- $\rightarrow$ is a transition relation of the type $(S_1 \times \Sigma \times S) \cup (S_2 \times \Delta \times S)$.

A run of 2-FSA M is any sequence of transitions

$$s \xrightarrow{z_1} s_1 \xrightarrow{z_2} \cdots \xrightarrow{z_{n-1}} s_{n-1} \xrightarrow{z_n} s'.$$

A run is complete if $s = s_0$ and $s' \in F$.

A 2-FSA M accepts a pair of words $(w, u) \in \Sigma^* \times \Delta^*$ if there is a complete run of M such that w is the projection of the word $z_1 z_2 \ldots z_{n-1} z_n$ on the alphabet Σ and u is the projection of the same word $z_1 z_2 \ldots z_{n-1} z_n$ on the alphabet Δ .

Two-tape finite automata

A transduction relation recognized by a 2-FSA M is the set $TR(M)$ of all pairs of words accepted by M .

2-FSAs M' and M'' are equivalent if $TR(M') = TR(M'')$.

A 2-FSA M is called deterministic (2-DFSA) if for every letter a and a state s it has at most one transition of the form $s \xrightarrow{a} s'$.

Equivalence checking problem is undecidable for 2-FSAs (Fisher P.S., Rozenberg A.L., 1966)

Equivalence checking problem is decidable for

- 2-DFSA (Bird M., 1973; Valiant L.G., 1974)
- 2-DFSA in polynomial time (Friedman E.P., Greibach S.A., 1982)
- deterministic multi-tape automata (Harju T., Karhumaki J., 1991)

Prefix-free transducers: preliminaries

A word u is a prefix of a word w if $w = uv$ holds for some word v .

In this case w is called an extension of u and $v = u \setminus w$ a left quotient of u with w .

Two words u_1 and u_2 are compatible if one of them is a prefix of the other.

A language L is called prefix-free if all its words are pairwise incompatible.

Two languages L' and L'' are compatible if every word in any of these languages is compatible with some word in the other.

Given a word u and a language L , we denote by

> $Pref(L)$ the set of all prefixes of the words in L ,
>
> $u \setminus L$ a left quotient $\{v : uv \in L\}$ of u with L .

Notice, that if $u \notin Pref(L)$ then $u \setminus L = \emptyset$.

Prefix-free transducers: preliminaries

Proposition 1.

Let L' and L'' be finite prefix-free compatible languages.

Then there exists the unique partitions $L' = \bigcup_{i=1}^{n} L'_i$ and $L'' = \bigcup_{i=1}^{n} L''_i$ such that for every $i, 1 \leq i \leq n$, one of the subsets L'_i or L''_i is a singleton $\{u\}$ and all words from the other are extensions of u .

Such partitioning of a compatible pair of prefix-free languages L' and L'' will be called its splitting . The pairs of corresponding subsets L'_i and L''_i , $1 \leq i \leq n$, will be called its fractions .

Prefix-free transducers: preliminaries

Proposition 1.

Let L' and L'' be finite prefix-free compatible languages.

Then there exists the unique partitions $L' = \bigcup_{i=1}^{n} L'_i$ and $L'' = \bigcup_{i=1}^{n} L''_i$ such that for every $i, 1 \leq i \leq n$, one of the subsets L'_i or L''_i is a singleton $\{u\}$ and all words from the other are extensions of u .

Such partitioning of a compatible pair of prefix-free languages L' and L'' will be called its splitting . The pairs of corresponding subsets L'_i and L''_i , $1 \leq i \leq n$, will be called its fractions .

Example.

$$L' = \{aaabb, bcc, aaabab, bcaca\}, \quad L'' = \{bca, bccaa, aaab, bccc\}$$

Prefix-free transducers: preliminaries

Proposition 1.

Let L' and L'' be finite prefix-free compatible languages.

Then there exists the unique partitions $L' = \bigcup_{i=1}^{n} L'_i$ and $L'' = \bigcup_{i=1}^{n} L''_i$ such that for every $i, 1 \leq i \leq n$, one of the subsets L'_i or L''_i is a singleton $\{u\}$ and all words from the other are extensions of u .

Such partitioning of a compatible pair of prefix-free languages L' and L'' will be called its splitting . The pairs of corresponding subsets L'_i and L''_i , $1 \leq i \leq n$, will be called its fractions .

Example.

$$
\begin{aligned}
L' &= \{aaabb, bcc, aaabab, bcaca\}, & L'' &= \{bca, bccaa, aaab, bccc\} \\
L'_1 &= \{aaabb, aaabab\}, & L''_1 &= \{aaab\}; \\
L'_2 &= \{bcc\}, & L''_2 &= \{bccaa, bccc\}; \\
L'_3 &= \{bcaca\}, & L''_3 &= \{bca\}.
\end{aligned}
$$

Prefix-free transducers: preliminaries

Given a transducer $\pi = \langle Q, q, F, \longrightarrow \rangle$ over languages Σ and Δ , a state $q \in Q$ and a letter $x \in \Sigma$, we denote by

$$Out_{\pi}(q, x) = \{(u, q') : q \xrightarrow{x/u} q'\} \ .$$

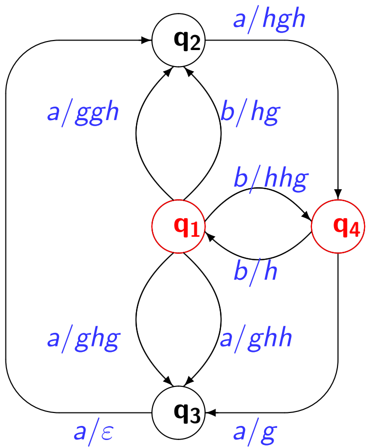
A transducer π is called prefix-free if for every $q \in Q$ and $x \in \Sigma$ the language

$$L_{\pi}(q, x) = \{u : \exists p \ (u, p) \in Out_{\pi}(q, x)\}$$

is prefix-free.

Prefix-free transducers have certain "deterministic" property: for every state q of a prefix-free transducer π and for every pair $(w, u) \in Tr(\pi, q)$ there is the only run of π from the state q labeled with (u, w) .

Prefix-free transducers: equivalence checking



Prefix-free transducers: equivalence checking

Idea

The equivalence checking technique for prefix-free transducers is based on manipulations with regular expressions.

1. We introduce for every state q of a transducer π a variable X_q .

2. We associate with a transducer π a system of linear regular expression equations $\mathcal{E}(\pi)$ over variables $X_q,\ q \in Q$, which specifies the behaviour of π .

3. To check the equivalence $\pi(q') \sim \pi(q'')$ we add to the set of equations $\mathcal{E}(\pi)$ the equivalence requirement which is an equation of the form $X_{q'} = X_{q''}$

4. Then we verify whether the resulting system of equations has a solution.

Equivalence checking: assumptions

For the sake of clarity we will assume that:

- the input alphabet $\Sigma = \{a_1, \ldots, a_k\}$ and $\Gamma \cap \Delta = \emptyset$; symbols x , y , z will denote arbitrary letters from Σ , and symbols u , v , w will denote words from Δ^* .
- $\pi' = \pi(q')$ and $\pi'' = \pi(q'')$,
- the transducer π is trim , i.e. a final state is reachable from each state of π .

Equivalence checking: regexes

Regular expressions (regexes) are built of

variables $X_1, X_2, \ldots$,
constants $0, 1$,
and letters from Σ and Δ

by means of concatenation $\cdot$ and alternation $+$.

Regexes are interpreted on the semiring of transductions over Σ and Δ.

0 is interpreted as the transduction $\emptyset$
1 as $\{(\varepsilon, \varepsilon)\}$,
every letter x as $\{(x, \varepsilon)\}$,
every word u as $\{(\varepsilon, u)\}$.

Concatenation of transductions T_1 and T_2 is defined as:

$$T_1 T_2 = \{(h_1 h_2, u_1 u_2) : (h_1, u_1) \in T_1, (h_2, u_2) \in T_2\}.$$

Equivalence checking: linear regexes

We will focus on linear regexes of two types.

A Δ -regex is any expression of the form

$$E = u_1 \cdot X_1 + u_2 \cdot X_2 + \cdots + u_n \cdot X_n.$$

When a set of words $\{u_1, u_2, \ldots, u_n\}$ is prefix-free then such a Δ -regex will be also called prefix-free.

A Σ -regex is any expression of the form

$$G = a_1 \cdot E_1 + a_2 \cdot E_2 + \cdots + a_k \cdot E_k,$$

where $E_i, 1 \leq i \leq k,$ are Δ -regexes.

Equivalence checking: systems of equations

With each state q of a transducer π we associate a variable X_q , and for every pair $q \in Q$ and $x \in \Sigma$ we build a Δ -regex

$$E_{q,x} = \sum_{(u,p) \in Out_\pi(q,x)} u \cdot X_p.$$

Then the transducer π is specified by the system of equations $\mathcal{E}_\pi$:

$$\{X_q = \sum_{x \in \Sigma} x \cdot E_{q,x} + c_q : q \in Q\},$$

where $c_q = 1$ if $q \in F$, or $c_q = 0$ otherwise.

Equivalence checking: systems of equations

With each state q of a transducer π we associate a variable X_q , and for every pair $q \in Q$ and $x \in \Sigma$ we build a Δ -regex

$$E_{q,x} = \sum_{(u,p) \in Out_\pi(q,x)} u \cdot X_p.$$

Then the transducer π is specified by the system of equations $\mathcal{E}_\pi$:

$$\{X_q = \sum_{x \in \Sigma} x \cdot E_{q,x} + c_q : q \in Q\},$$

where $c_q = 1$ if $q \in F$, or $c_q = 0$ otherwise.

Proposition 2.

For every finite transducer π the system of equation $\mathcal{E}_\pi$ has the unique solution $\{X_q = Tr(\pi, q) : q \in Q\}$.

Equivalence checking: systems of equations

With each state q of a transducer π we associate a variable X_q , and for every pair $q \in Q$ and $x \in \Sigma$ we build a Δ -regex

$$E_{q,x} = \sum_{(u,p) \in Out_\pi(q,x)} u \cdot X_p.$$

Then the transducer π is specified by the system of equations $\mathcal{E}_\pi$:

$$\{X_q = \sum_{x \in \Sigma} x \cdot E_{q,x} + c_q : q \in Q\},$$

where $c_q = 1$ if $q \in F$, or $c_q = 0$ otherwise.

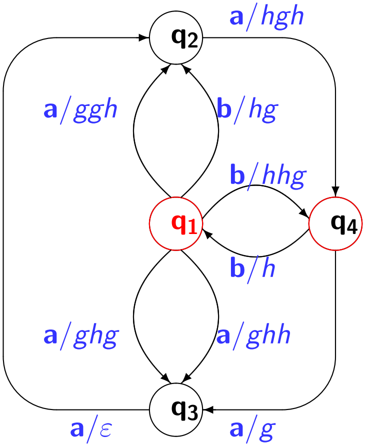
Proposition 2.

For every finite transducer π the system of equation $\mathcal{E}_\pi$ has the unique solution $\{X_q = Tr(\pi, q) : q \in Q\}$.

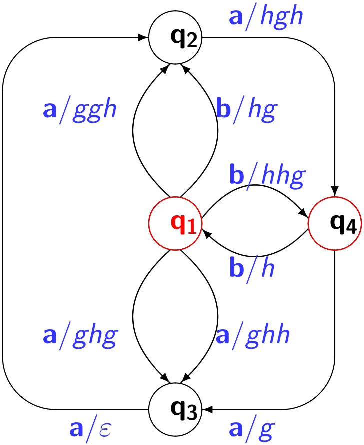
Corollary.

$\pi(p) \sim \pi(q) \iff \mathcal{E}_\pi \cup \{X_p = X_q\}$ has a solution.

Equivalence checking: systems of equations



Equivalence checking: systems of equations



The system of equations $\mathcal{E}_\pi$:

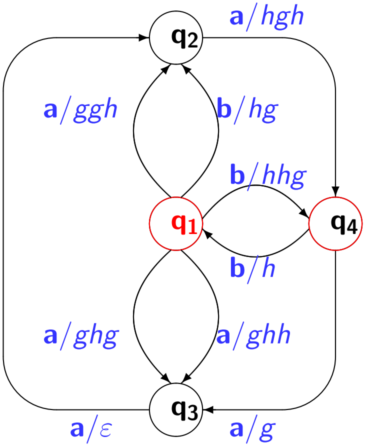
$$X_1 = \mathbf{a} \cdot (ggh \cdot X_2 + ghg \cdot X_3 + ghh \cdot X_3) + \mathbf{b} \cdot (hg \cdot X_2 + hhg \cdot X_4) + 1$$

$$X_2 = \mathbf{a} \cdot hgh \cdot X_4$$

$$X_3 = \mathbf{a} \cdot X_2$$

$$X_4 = \mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_1 + 1$$

Equivalence checking: systems of equations



The system of equations $\mathcal{E}_\pi$:

$$X_1 = \mathbf{a} \cdot (ggh \cdot X_2 + ghg \cdot X_3 + ghh \cdot X_3) + \mathbf{b} \cdot (hg \cdot X_2 + hhg \cdot X_4) + 1$$

$$X_2 = \mathbf{a} \cdot hgh \cdot X_4$$

$$X_3 = \mathbf{a} \cdot X_2$$

$$X_4 = \mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_1 + 1$$

Equivalence checking problem $\pi(q_1) \sim \pi(q_4)$:

$$X_1 = X_4$$

Equivalence checking: systems of equations

We say that a system of linear equations

$$\mathcal{E} = \mathcal{E}_\pi(X_1, \ldots, X_n) \cup \{X'_j = E_j(X_1, \ldots, X_n) : 1 \leq j \leq m\},$$

is reduced if $\{X_1, \ldots, X_n\}$ and $\{X'_1, \ldots, X'_m\}$ are disjoint sets of variables and all right-hand sides E_j are Δ -regexes.

Equivalence checking: systems of equations

We say that a system of linear equations

$$\mathcal{E} = \mathcal{E}_\pi(X_1, \ldots, X_n) \cup \{X'_j = E_j(X_1, \ldots, X_n) : 1 \leq j \leq m\},$$

is reduced if $\{X_1, \ldots, X_n\}$ and $\{X'_1, \ldots, X'_m\}$ are disjoint sets of variables and all right-hand sides E_j are Δ -regexes.

Proposition 3.

Every reduced system of equations $\mathcal{E}$ has the unique solution.

Equivalence checking: systems of equations

Some other extensions of the systems $\mathcal{E}_\pi$ have no solutions.

Proposition 4.

If languages $L_1 = \{u_1, \ldots, u_\ell\}$ and $L_2 = \{v_1, \ldots, v_m\}$ are incompatible then a system of equations

$$\mathcal{E}_\pi(X_1, \ldots, X_n) \cup \{\sum_{i=1}^{\ell} u_i \cdot X_i = \sum_{j=1}^{m} v_j \cdot X_j\}$$

has no solutions.

Equivalence checking: systems of equations

Some other extensions of the systems $\mathcal{E}_\pi$ have no solutions.

Proposition 4.

If languages $L_1 = \{u_1, \ldots, u_\ell\}$ and $L_2 = \{v_1, \ldots, v_m\}$ are incompatible then a system of equations

$$\mathcal{E}_\pi(X_1, \ldots, X_n) \cup \{\sum_{i=1}^{\ell} u_i \cdot X_i = \sum_{j=1}^{m} v_j \cdot X_j\}$$

has no solutions.

Proposition 5.

If a set of words $\{u_1, \ldots, u_\ell\}$ is prefix-free and a system

$$\mathcal{E}_\pi(X_1, \ldots, X_n) \cup \{X_1 = \sum_{i=1}^{\ell} u_i \cdot X_i\}$$

has a solution then $\ell = 1$ and $u_1 = \varepsilon$.

Systems of equations: solution technique

An iterative procedure checks the solvability of the system of equations $\mathcal{E}_1 = \mathcal{E}_\pi \cup \{X_p = X_q\}$ for prefix-free transducer π by bringing this system to an equivalent reduced form.

At the beginning of each iteration t the procedure gets at the input a system of equations of the form

$$\mathcal{E}_t = \mathcal{E}_{\pi_t} \cup \{X_i = E_i : 1 \leq i \leq N_t\},$$

where π_t is some prefix-free transducer and all Δ -regexes E_i are prefix-free.

If a variable X occurs more than once in $\mathcal{E}_t$ then we call it active .

At the t -th iteration equivalent transformations are applied to $\mathcal{E}_t$.

Systems of equations: solution technique

$$\mathcal{E}_t = \mathcal{E}_{\pi_t} \cup \{X_i = E_i : 1 \leq i \leq N_t\}$$

1) Removing of identities.
Equations of the form $X = X$ are removed from $\mathcal{E}_t$.

Systems of equations: solution technique

$$\mathcal{E}_t = \mathcal{E}_{\pi_t} \cup \{X_i = E_i : 1 \leq i \leq N_t\}$$

1) Removing of identities.
Equations of the form $X = X$ are removed from $\mathcal{E}_t$.

2) Checking if $\mathcal{E}_t$ is reduced system
If none of the variables $X_1, \ldots, X_{N_t}$ from left-hand sides of equations $X_i = E_i$ occurs elsewhere then terminate and announce the **solvability**.

Systems of equations: solution technique

$$\mathcal{E}_t = \mathcal{E}_{\pi_t} \cup \{X_i = E_i : 1 \leq i \leq N_t\}$$

1) Removing of identities.
Equations of the form $X = X$ are removed from $\mathcal{E}_t$.

2) Checking if $\mathcal{E}_t$ is reduced system
If none of the variables $X_1, \ldots, X_{N_t}$ from left-hand sides of equations $X_i = E_i$ occurs elsewhere then terminate and announce the **solvability**.

3) Elimination of variables.
For every equation of the form $X_i = E_i$ in $\mathcal{E}_t$

- if X_i is in Δ -regex E_i then terminate and announce the **unsolvability**;
- otherwise in all other equations of $\mathcal{E}$ replace all the occurrences of X_i with E_i .

Systems of equations: solution technique

The system of equations:

$$
\begin{aligned}
X_1 &= \mathbf{a} \cdot (ggh \cdot X_2 + ghg \cdot X_3 + ghh \cdot X_3) + \\
&\quad \mathbf{b} \cdot (hg \cdot X_2 + hhg \cdot X_4) + 1 \\
X_2 &= \mathbf{a} \cdot hgh \cdot X_4 \\
X_3 &= \mathbf{a} \cdot X_2 \\
X_4 &= \mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_1 + 1 \\
X_1 &= X_4
\end{aligned}
$$

Systems of equations: solution technique

The system of equations:

$$\begin{aligned}
X_4 &= \mathbf{a} \cdot (ggh \cdot X_2 + ghg \cdot X_3 + ghh \cdot X_3) + \\
&\quad \mathbf{b} \cdot (hg \cdot X_2 + hhg \cdot X_4) + 1 \\
X_2 &= \mathbf{a} \cdot hgh \cdot X_4 \\
X_3 &= \mathbf{a} \cdot X_2 \\
X_4 &= \mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_4 + 1 \\
X_1 &= X_4
\end{aligned}$$

Systems of equations: solution technique

The system of equations:

$$
\begin{aligned}
X_4 &= \mathbf{a} \cdot (ggh \cdot X_2 + ghg \cdot X_3 + ghh \cdot X_3) + \\
&\quad \mathbf{b} \cdot (hg \cdot X_2 + hhg \cdot X_4) + 1 \\
X_2 &= \mathbf{a} \cdot hgh \cdot X_4 \\
X_3 &= \mathbf{a} \cdot X_2 \\
X_4 &= \mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_4 + 1 \\
X_1 &= X_4
\end{aligned}
$$

Systems of equations: solution technique

The number of active variables in $\mathcal{E}_t$ decreases. But this substitution has a side effect: non-standard equations of the form

A) $E' = E''$, where E', E'' are non-variable Δ -regexes, and

B) $E = G$, where E is a Δ -regex and G is a Σ -regex,

may appear in $\mathcal{E}_t$. It may also happen that

C) several equations of the form $X = G$ with the same variable X appear in $\mathcal{E}_t$.

Systems of equations: solution technique

4) Elimination of non-standard equations $E = G$.
Equations which spoil the system are removed from $\mathcal{E}_t$.

- for every equation of the form $E(X_1, \ldots, X_\ell) = G$ replace all the occurrences of variables $X_i, 1 \leq i \leq \ell$, in Δ -regex E with Σ -regexes G_i that correspond to these variables in the equations $X_i = G_i$ from the subsystem $\mathcal{E}_{\pi_t}$: as the result we obtain an equation of the form
$$E(G_1, \ldots, G_\ell) = G \ ;$$
- for every pair of equations $X = G'$ and $X = G''$ with the same left-hand side but different Σ -regexes G' and G'' replace one of these equations with the equation $G' = G''$.

All equations of the form $E = G$ disappear and all equations of the form $X = G$ will have pairwise different left-hand side variables.
But this is achieved by inserting to the system non-standard equations of the form $G' = G''$ where G', G'' are Σ -regexes.

Systems of equations: solution technique

The system of equations:

$$
\begin{aligned}
X_4 &= \mathbf{a} \cdot (ggh \cdot X_2 + ghg \cdot X_3 + ghh \cdot X_3) + \\
&\quad \mathbf{b} \cdot (hg \cdot X_2 + hhg \cdot X_4) + 1 \\
X_2 &= \mathbf{a} \cdot hgh \cdot X_4 \\
X_3 &= \mathbf{a} \cdot X_2 \\
X_4 &= \mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_4 + 1 \\
X_1 &= X_4
\end{aligned}
$$

Systems of equations: solution technique

The system of equations:

$$\underline{\mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_4 + 1} = \mathbf{a} \cdot (ggh \cdot X_2 + ghg \cdot X_3 + ghh \cdot X_3) + \mathbf{b} \cdot (hg \cdot X_2 + hhg \cdot X_4) + 1$$

$$X_2 = \mathbf{a} \cdot hgh \cdot X_4$$

$$X_3 = \mathbf{a} \cdot X_2$$

$$X_4 = \underline{\mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_4}$$

$$X_1 = X_4$$

Systems of equations: solution technique

5) Elimination of nonstandard equations $G' = G''$.
Remove from the system every equation of the form

$$\sum_{i=1}^{k} \mathbf{a_i} \cdot E_i' = \sum_{i=1}^{k} \mathbf{a_i} \cdot E_i''$$

and inserts instead of it k equations $E_i' = E_i'', 1 \leq i \leq k$.
Thus, all equations of the form $G' = G''$ disappear from the system due to the introduction of new equations of the form $E' = E''$.

After this step equations of this form are the only non-standard equations that remain in the system.

Systems of equations: solution technique

The system of equations:

$$
\begin{aligned}
\mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_4 + 1 = \mathbf{a} \cdot (ggh \cdot X_2 + ghg \cdot X_3 + ghh \cdot X_3) + \\
\mathbf{b} \cdot (hg \cdot X_2 + hhg \cdot X_4) + 1
\end{aligned}
$$

$$X_2 = \mathbf{a} \cdot hgh \cdot X_4$$

$$X_3 = \mathbf{a} \cdot X_2$$

$$X_4 = \mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_4$$

$$X_1 = X_4$$

Systems of equations: solution technique

The system of equations:

$$g \cdot X_3 = ggh \cdot X_2 + ghg \cdot X_3 + ghh \cdot X_3$$

$$h \cdot X_4 = hg \cdot X_2 + hhg \cdot X_4$$

$$X_2 = \mathbf{a} \cdot hgh \cdot X_4$$

$$X_3 = \mathbf{a} \cdot X_2$$

$$X_4 = \mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_4$$

$$X_1 = X_4$$

Systems of equations: solution technique

6) Elimination of nonstandard equations $E' = E''$.

For every equation $\sum_{i=1}^{\ell} u_i \cdot X'_i = \sum_{j=1}^{m} v_j \cdot X''_j$

check the compatiblity of $L' = \{u_1, \ldots, u_\ell\}$ and $L'' = \{v_1, \ldots, v_m\}$

.

- if L' and L'' are incompatible then terminate and announce the **unsolvability** of the system;
- otherwise make a splitting $L' = \bigcup_{i=1}^{n} L'_i$ and $L'' = \bigcup_{i=1}^{n} L''_i$, remove the equation from $\mathcal{E}_t$, and insert for every fraction $L'_i = \{u_{i_0}\}$ and $L''_i = \{v_{i_1}, \ldots, v_{i_r}\}$ an equation

$$X'_{i_0} = (u_{i_0} \setminus v_{i_1}) \cdot X''_{i_1} + \cdots + (u_{i_0} \setminus v_{i_r}) \cdot X''_{i_r}.$$

We obtain the system of equations $\mathcal{E}_{t+1}$ which is equivalent to $\mathcal{E}_t$ but has a smaller number of active variables than $\mathcal{E}_t$.

Systems of equations: solution technique

The system of equations:

$$g \cdot X_3 = ggh \cdot X_2 + ghg \cdot X_3 + ghh \cdot X_3$$

$$h \cdot X_4 = hg \cdot X_2 + hhg \cdot X_4$$

$$X_2 = \mathbf{a} \cdot hgh \cdot X_4$$

$$X_3 = \mathbf{a} \cdot X_2$$

$$X_4 = \mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_4$$

$$X_1 = X_4$$

Systems of equations: solution technique

The system of equations:

$$X_3 = gh \cdot X_2 + hg \cdot X_3 + hh \cdot X_3$$

$$X_4 = g \cdot X_2 + hg \cdot X_4$$

$$X_2 = \mathbf{a} \cdot hgh \cdot X_4$$

$$X_3 = \mathbf{a} \cdot X_2$$

$$X_4 = \mathbf{a} \cdot g \cdot X_3 + \mathbf{b} \cdot h \cdot X_4$$

$$X_1 = X_4$$

Prefix-free transducers: equivalence checking

Proposition 6.

For every prefix-free transducer π and a pair of its states p, q the procedure above when being applied to the system of equations $\mathcal{E}_1 = \mathcal{E}_\pi \cup \{X_p = X_q\}$ terminates and correctly detects the solvability of $\mathcal{E}_1$.

Theorem 1.

Equivalence problem for finite prefix-free transducers is decidable in time $O(n^2)$.

2-tape automata and generalized transducers

Let $M = \langle S_1, S_2, s_0, F, \rightarrow \rangle$ be a 2-DFSA over alphabets Σ and Δ .
Without loss of generality we will assume that $F \subseteq S_1$.

For every $\widehat{s} \in S_1$ and $x \in \Sigma$ define a set $Out_M(\widehat{s}, x) \subseteq \Delta^* \times S_1$.

Consider the transition $\widehat{s} \xrightarrow{x} s$ of M .

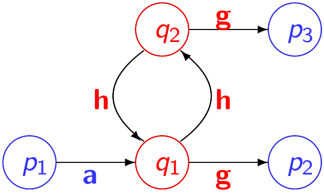
1) If $s \in S_1$ then $Out_M(\widehat{s}, x) = \{(\varepsilon, s)\}$.

2) If $s \in S_2$ then $Out_M(\widehat{s}, x)$ is a set of all pairs $(z_1 z_2 \ldots z_{n-1} z_n, s')$ such that there exists a run of M

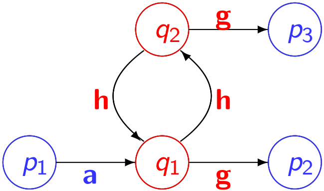
$$s \xrightarrow{z_1} s_1 \xrightarrow{z_2} \cdots \xrightarrow{z_{n-1}} s_{n-1} \xrightarrow{z_n} s'.$$

which passes only via states s_i of the set S_2 and ends at $s' \in S_1$.

2-tape automata and generalized transducers

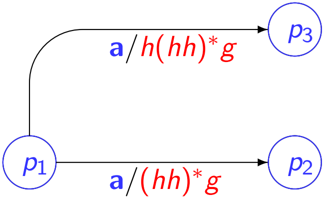


2-tape automata and generalized transducers



$$Out_M(p_1, \mathbf{a}) = \{(g, p_2), (hhg, p_2), \ldots, (h^{2k}g, p_2), \ldots$$
$$(hg, p_3), (hhhg, p_3), \ldots, (h^{2k+1}g, p_3), \ldots\}$$

2-tape automata and generalized transducers



2-tape automata and generalized transducers

Proposition 7.

For every 2-DFSA $M = \langle S_1, S_2, s_0, F, \to \rangle$ over Σ and Δ , a pair of states $\widehat{s}, s \in S_1$, and a letter $x \in \Sigma$ the set of words

$$L_M(\widehat{s}, s, x) = \{w : (w, s) \in Out_M(\widehat{s}, x)\}$$

is a regular prefix-free language. Moreover, the union

$$L_M(\widehat{s}, x) = \bigcup_{s \in S_1} L_M(\widehat{s}, s, x)$$

is also a regular prefix-free language.

We associate with every 2-DFSA M a transducer $\pi_M = \langle S_1, s_0, F, \longrightarrow \rangle$ over Σ and Δ such that the transition relation $\longrightarrow$ meets the requirement $s \xrightarrow{x/w} s' \Leftrightarrow (w, s') \in Out_M(\widehat{s}, x)$ for every quadruple $(s, x, w, s') \in S_1 \times \Sigma \times \Delta^* \times S_1$.

Proposition 8.

The equality $TR(M) = TR(\pi_M, s_0)$ holds for every 2-DFSA M .

2-tape automata and generalized transducers

A generalized prefix-free finite transducer over languages Σ and Δ is a quadruple $\Pi = \langle Q, q_0, F, \longrightarrow \rangle$, where

Q is a finite set of states,

q_0 is an initial state,

F is a subset of final states, and

$\psi : Q \times \Sigma \times Q \to PFReg(\Delta)$ is a transition function such that for every q and x the language $\bigcup_{q' \in Q} \psi(q, x, q')$ is prefix-free.

As usual, we write $q \xrightarrow{x/L} q'$ whenever $\psi(q, x, q') = L$.
A run of Π is any finite sequence of transitions

$$q \xrightarrow{a_1/L_1} q_1 \xrightarrow{a_2/L_2} \dots \xrightarrow{a_{n-1}/L_{n-1}} q_{n-1} \xrightarrow{a_n/L_n} q'.$$

When writing $q \xrightarrow{w/L}_* q'$ we mean that Π has a run such that $w = a_1 a_2 \dots a_n$ and $L = L_1 L_2 \dots L_n$.
A transduction relation realized by Π in its state q is the set of pairs $TR(\Pi, q) = \{(w, u) \ : \ q \xrightarrow{w/L}_* q', u \in L, q' \in F\}$.

Generalized prefix-free transducers: equivalence checking

For every 2-DFSA M there exists a generalized prefix-free finite transducer Π_M such that $TR(M) = TR(\Pi_M, s_0)$.

To check the equivalence of generalized prefix-free finite transducers we adapt the approach developed for the analysis of ordinary prefix-free finite transducers.

Regexes are built of variables $X_1, X_2, \dots$, constants $0, 1$, and letters from Σ , but instead of Δ we will use prefix-free regular languages from $PFReg$ as constants.

Modified Δ -regexes: $L_1 \cdot X_1 + L_2 \cdot X_2 + \cdots + L_n \cdot X_n$, where $L_i \in FPReg$ for every $i, 1 \leq i \leq n$.

The system of equations $\mathcal{E}_1 = \mathcal{E}_\Pi \cup \{X_{q'} = X_{q''}\}$ for Π is constructed in the same way as for ordinary transducers.

Propositions 2–5 hold for equations with modified Δ -regexes.

Rules 1)–5) of the solvability checking procedure remain the same.

Generalized prefix-free transducers: equivalence checking

6′) Elimination of nonstandard equations $E' = E''$.

For every equation (*) $\sum_{i=1}^{\ell} L'_i \cdot X'_i = \sum_{j=1}^{m} L''_j \cdot X''_j$ check the compatiblity of $L' = \bigcup_{i=1}^{\ell} L'_i$ and $L'' = \bigcup_{j=1}^{m} L''_j$.

If the languages are incompatible then terminate and announce the **unsolvability** of the system. Otherwise,

6.1 For every $i, 1 \leq i \leq \ell$, such that $L'_i \cap Pref(L'') \neq \emptyset$ find any word $w \in L'_i \cap Pref(L'')$, and add an equation $X'_i = \sum_{j=1}^{m} (w \setminus L''_j) \cdot X''_j$ to the system, and replace all other occurrences of X'_i with the right-hand side of this equation.

6.2 Do the same for every $j, 1 \leq j \leq m$.

6.3 If the equation (*) does not become an identity, then terminate and announce the **unsolvability** of the system.

Generalized prefix-free transducers: equivalence checking

Proposition 9.

If an equation $L_0 \cdot X_0 = \sum_{i=1}^{n} L_i \cdot X_i$ with a prefix-free Δ -regex at the right-hand side has a prefix-free solution, and $w \in L_0 \cap Pref(\bigcup_{i=1}^{n} L_i)$, then the equation $X_0 = \sum_{i=1}^{n} (w \setminus L_i) \cdot X_i$ has the same solution.

Theorem 2.

The equivalence problem for generalized prefix-free finite transducers is decidable in time $O(n^3)$.

Corollary

The equivalence problem for deterministic two-tape finite state automata is decidable in time $O(n^3)$.

Conclusions

Topics for future research.

Conclusions

Topics for future research.

1. Prove the complexity estimates in Theorems 1 and 2.

Conclusions

Topics for future research.

1. Prove the complexity estimates in Theorems 1 and 2.

2. Find and prove an analogue of Proposition 1 for regular languages instead of words. Modify correspondingly the Rule $6'$).

Conclusions

Topics for future research.

1. Prove the complexity estimates in Theorems 1 and 2.

2. Find and prove an analogue of Proposition 1 for regular languages instead of words. Modify correspondingly the Rule 6′).

3. Build a minimization procedure for prefix-free finite and generalized transducers.

Conclusions

Topics for future research.

1. Prove the complexity estimates in Theorems 1 and 2.

2. Find and prove an analogue of Proposition 1 for regular languages instead of words. Modify correspondingly the Rule $6'$).

3. Build a minimization procedure for prefix-free finite and generalized transducers.

4. Find solutions to equivalence and minimization problems for generalized (non prefix-free) transducers.

Conclusions

Topics for future research.

1. Prove the complexity estimates in Theorems 1 and 2.

2. Find and prove an analogue of Proposition 1 for regular languages instead of words. Modify correspondingly the Rule 6′).

3. Build a minimization procedure for prefix-free finite and generalized transducers.

4. Find solutions to equivalence and minimization problems for generalized (non prefix-free) transducers.

5. How to modify the presented technique to cope with equivalence problem for compatibility-free finite transducers operating over $\Delta_1 \times \Delta_2$ instead of Δ ?

Conclusions

Topics for future research.

1. Prove the complexity estimates in Theorems 1 and 2.

2. Find and prove an analogue of Proposition 1 for regular languages instead of words. Modify correspondingly the Rule 6′).

3. Build a minimization procedure for prefix-free finite and generalized transducers.

4. Find solutions to equivalence and minimization problems for generalized (non prefix-free) transducers.

5. How to modify the presented technique to cope with equivalence problem for compatibility-free finite transducers operating over $\Delta_1 \times \Delta_2$ instead of Δ ?

6. How to solve equivalence problem for compatibility-free generalized transducers operating over $\Delta_1 \times \Delta_2$

Conclusions

Topics for future research.

1. Prove the complexity estimates in Theorems 1 and 2.

2. Find and prove an analogue of Proposition 1 for regular languages instead of words. Modify correspondingly the Rule 6′).

3. Build a minimization procedure for prefix-free finite and generalized transducers.

4. Find solutions to equivalence and minimization problems for generalized (non prefix-free) transducers.

5. How to modify the presented technique to cope with equivalence problem for compatibility-free finite transducers operating over $\Delta_1 \times \Delta_2$ instead of Δ ?

6. How to solve equivalence problem for compatibility-free generalized transducers operating over $\Delta_1 \times \Delta_2$

$$\equiv$$

i.e. **equivalence problem for deterministic 3-tape automata** ?