International Journal of Foundations of Computer Science
Vol. 16, No. 5 (2005) 851–865

COVER TRANSDUCERS FOR FUNCTIONS WITH FINITE DOMAIN

JEAN-MARC CHAMPARNAUD[1], FRANCK GUINGNE[2,3] and GEORGES HANSEL[2]

[1] PSI Laboratory (Université de Rouen, CNRS)
76821 Mont-Saint-Aignan — France
Jean-Marc.Champarnaud@univ-rouen.fr – http://www.univ-rouen.fr/psi/

[2] LIFAR Laboratory (Université de Rouen)
76821 Mont-Saint-Aignan — France
{Franck.Guingne, Georges.Hansel}@univ-rouen.fr – http://www.univ-rouen.fr/LIFAR/

[3] Xerox Research Centre Europe – Grenoble Laboratory
6 chemin de Maupertuis – 38240 Meylan — France
Franck.Guingne@xrce.xerox.com – http://www.xrce.xerox.com

Received 20 November 2004
Accepted 14 February 2005
Communicated by L. Ilie and D. Wotschke

Cover automata were introduced a few years ago for designing a compact representation of finite languages. Our aim is to extend this notion to cover transducers for functions with finite domain. Given two alphabets Σ and Ω, and a function $\alpha : \Sigma^* \to \Omega^*$ of order l (the maximal length of a word in the domain of α), a cover transducer for α is any subsequential transducer that realizes the function α when its input is restricted to the set of words of Σ^* having a length not greater than l. We study the problem of reducing the number of states of a cover transducer. We report experimental results, from an implementation using WFSC (Weighted Finite State Compiler), a Xerox tool for handling weighted finite state automata and transducers.

1. Introduction

Cover automata for finite languages were introduced by Câmpeanu *et al.* [4]. A cover automaton for a language L of order l (the maximal length of a word in L) is a deterministic automaton $\mathcal{A}$ such that $L(\mathcal{A}) \cap \Sigma^{\leq l} = L$, where $\Sigma^{\leq l}$ is the subset of Σ^* of words whose length is not greater than l. In this paper, we define the notion of a cover transducer for a function with finite domain as an extension of the notion of a cover automaton for a finite language. Given two alphabets Σ and Ω, and a function $\alpha : \Sigma^* \to \Omega^*$ of order l (the maximal length of a word in the domain of α), a cover transducer for α is any subsequential transducer that realizes the function α when its input is restricted to $\Sigma^{\leq l}$.

Since covering generally reduces the size of an automaton [15], it is of practical interest to be able to compute a minimal cover automaton for L (with respect to the number of states). It is shown in [4] that a minimal cover automaton can be obtained from any cover automaton for L by merging states according to a relation involving their right languages. Minimality (with respect to L) comes from the fact that a similarity relation on

$\Sigma^{\leq l}$ $[10, 9, 6]$ underlies the state relation (see [6] for a general study of similarity relations). Several algorithms were designed for computing a minimal cover automaton $[4, 5, 13]$, either from a deterministic automaton recognizing L, or from an arbitrary cover automaton for L. The best algorithm currently known [13] is $O(n \log n)$ time and $O(n)$ space.

Our solution for cover transducers is less ambitious since it seems quite difficult to give a straightforward characterization of minimal cover transducers. We first discuss the relative merging power of different relations defined on the set of states of the initial cover transducer. We show that the relation on $\Sigma^{\leq l}$ that underlies the coarsest merging relation is not semi-transitive. Computing a minimal partition of the set of states according to this merging relation is therefore more complex than according to a similarity relation.

We also show that the algorithm for minimizing a subsequential transducer [8] or a weighted deterministic automaton [14] can be adapted for reducing cover transducers. Our solution combines the construction of a prefix transducer [1] and the computation of a minimal cover automaton. We discuss the power of this technique and we report experimental results obtained using Xerox tools for creating and manipulating finite state automata and transducers: XFST $[11, 2]$ for the unweighted case, and WFSC [12] for the weighted case.

Our algorithm was run on acyclic transducers with several hundred states and the analysis of the state reduction enlightens the interest of cover transducers for handling dictionaries.

Useful definitions concerning automata, cover automata and transducers are recalled in the following section. Basic tools for the study of cover transducers are introduced in Section 3. Relations for reducing a cover transducer are compared in Section 4 (merging relations) and in Section 5 (similarity relations). Reduction via the minimization of a cover automaton is studied in Section 6. Section 7 reports on implementation aspects and presents an analysis of experimental results.

2. Preliminaries

2.1. *Automata*

The reader is assumed to be familiar with automata theory [17]; here we just introduce some notation. Let $\mathcal{A} = (\Sigma, Q, q_s, Q_+, \cdot)$ be a deterministic automaton on the alphabet Σ, where Q is the finite set of *states*, $q_s \in Q$ is the *initial state*, Q_+ is the set of *final states* and the *transition function*, denoted by $\cdot$, maps $(q, a) \in Q \times \Sigma$ to $q \cdot a \in Q$. The *left language* of a state $q \in Q$ is $\overleftarrow{L(q)} = \{x \in \Sigma^* \mid q_s \cdot x = q\}$. The *right language* of q is $\overrightarrow{L(q)} = \{x \in \Sigma^* \mid q \cdot x \in Q_+\}$. A deterministic automaton is said to be *complete* if its transition function is a total one. A deterministic automaton can be completed by adding a *sink state* to Q. A *semiautomaton* is an automaton without defined final states. The *level* of a state q is the length of a shortest path from the initial state q_s to q: $\forall q \in Q,\ \mathrm{level}(q) = \min\{|x| \mid x \in \Sigma^* \text{ and } q_s \cdot x = q\}$. The subset of words of Σ^* having a length not greater than l is denoted by $\Sigma^{\leq l}$. A language L is said to be *of order* l if the maximal length of a word in L is equal to l.

2.2. *Cover automata*

A relation $\sim$ over $\Sigma^{\leq l}$ is *semi-transitive* if and only if for all x, y, z in $\Sigma^{\leq l}$ such that $|x| \leq |y| \leq |z|$, the following implications hold: $x \sim y$ and $y \sim z \Rightarrow x \sim z$, and $x \sim y$ and $x \sim z \Rightarrow y \sim z$. A reflexive, symmetric and semi-transitive relation is a *similarity relation*. Let L be a language of order l. Let x and y be two words of $\Sigma^{\leq l}$ and $k = l - \max\{|x|, |y|\}$. The words x and y are said to be *similar with respect to* L (we write $x \sim_L y$) if and only if for all t in $\Sigma^{\leq k}$ the equivalence $xt \in L \Leftrightarrow yt \in L$ holds. The relation $\sim_L$ is a similarity relation $[10, 9]$.

A *cover automaton* $[4]$ for a language L of order l is a deterministic automaton $\mathcal{C} = (\Sigma, Q, q_s, Q_+, \cdot)$ such that $L(\mathcal{C}) \cap \Sigma^{\leq l} = L$. A *minimal* cover automaton for L has a minimal number of states among the cover automata for L. Let $\mathcal{C}$ be a cover automaton for the language L of order l and Q be its set of states. The *height* of a state q is $\mathrm{height}(q) = l - \mathrm{level}(q)$. Two states p and q of Q such that $h = \min\{\mathrm{height}(p), \mathrm{height}(q)\}$ can be *merged* according to the relation $\sim_{\mathcal{C}}$ defined on Q by $p \sim_{\mathcal{C}} q \Leftrightarrow \overrightarrow{L(p)} \cap \Sigma^{\leq h} = \overrightarrow{L(q)} \cap \Sigma^{\leq h}$.

Since $\mathcal{A}$ is a cover automaton for L, the relations $\sim_{\mathcal{C}}$ and $\sim_L$ are such that $p \sim_{\mathcal{C}} q \Rightarrow (\forall (x, y) \mid q_s \cdot x = p$ and $q_s \cdot y = q), x \sim_L y$.

A general study of similarity relations over the set $\Sigma^{\leq l}$ can be found in $[6]$. Let $\sim$ be a similarity relation on $\Sigma^{\leq l}$. An element x of $\Sigma^{\leq l}$ is said to be *minimal* if for all $y \in \Sigma^{\leq l}$, $y \sim x \Rightarrow |y| \geq |x|$. The set of all minimal elements of $\Sigma^{\leq l}$ is denoted M. A deterministic semiautomaton is a *similarity semiautomaton* for the relation $\sim$ if for all $q \in Q$, $\overleftarrow{L}(q)$ is a similarity set. Such a semiautomaton is said to *recognize* the relation $\sim$. The main results are the following:

Theorem 1. $[6]$ 1) *The relation* $\sim$ *is an equivalence relation on* M.
2) *Let* π_M *be the partition of* M *into equivalence classes. Then any minimal similarity partition of* $\Sigma^{\leq l}$ *(according to* $\sim$*) has* $|\pi_M|$ *elements and there exists such a partition.*
3) *Let* $\sim$ *be a right-invariant similarity relation on* $\Sigma^{\leq l}$*. Then any similarity semiautomaton recognizing* $\sim_L$ *has at least* $|\pi_M|$ *states and there exists a semiautomaton with* $|\pi_M|$ *states that recognizes* $\sim_L$.

A straightforward application to the relation $\sim_L$ yields the following results.

Theorem 2. $[6]$ 1) *A semiautomaton recognizing the relation* $\sim_L$*, when equipped with a convenient set of final states, is a cover automaton for* L.
2) *Conversely, given an arbitrary cover automaton* $\mathcal{C}$ *for* L*, the underlying semiautomaton of* $\mathcal{C}$ *recognizes the relation* $\sim_L$.
3) *Any cover automaton for the language* L *has at least* $|\pi_M|$ *states and there exists a cover automaton with* $|\pi_M|$ *states for* L.

The relation $\sim_L$ being a similarity one, there exists a (not necessarily unique) minimal cover automaton for L. It should be noted that minimality is defined with respect to the language L. On the other hand, given a cover automaton $\mathcal{C}$, a minimal one, denoted by $C(\mathcal{C})$ in the sequel, can be computed by merging states, according to the following theorem.

Theorem 3. [4] *A cover automaton $\mathcal{C}$ for a finite language L is minimal if and only if no two different states of $\mathcal{C}$ can be merged according to the relation $\sim_{\mathcal{C}}$.*

2.3. *Subsequential transducers*

Subsequential transducers $[16, 7, 3]$ are a relevant model for studying functions with finite domain. A *subsequential transducer* is a tuple $\mathcal{S} = (\Sigma, \Omega, Q, q_-, \mathtt{i}, \mathtt{t}, \cdot, *)$ where:
– Σ (resp. Ω) is the *input* (resp. *output*) *alphabet*,
– Q is the finite set of *states* and $q_- \in Q$ is the *initial state*,
– $\mathtt{i} \in \Omega^*$ is the *initialization value* and $\mathtt{t}$*:* $Q \to \Omega^*$ is the *termination function*,
– the *transition function*, denoted by $\cdot$, maps $(q, a) \in Q \times \Sigma$ to $q \cdot a \in Q$,
– the *output function*, denoted by $*$, maps $(q, a) \in Q \times \Sigma$ to $q * a \in \Omega^*$.

The transition (resp. output) function is extended to map $Q \times \Sigma^*$ into Q (resp. Ω^*). The set of *final states* of $\mathcal{S}$ is equal to the domain $\mathrm{dom}(\mathtt{t})$ of $\mathtt{t}$. Therefore, by sake of simplicity, we do not include $\mathrm{dom}(\mathtt{t})$ in the tuple that defines $\mathcal{S}$. A *path* is a finite sequence $((q_i, a_i, b_i, q_{i+1}))_{i=0,\ldots,n-1}$ of tuples in $Q \times \Sigma \times \Omega^* \times Q$ with $q_i \cdot a_i = q_{i+1}$ and $q_i * a_i = b_i$. A *final* path ends in $q_n \in \mathrm{dom}(\mathtt{t})$. A *successful* path is a final path starting in $q_0 = q_-$. The word $a_0 \cdots a_{n-1} \in \Sigma^*$ (resp. $b_0 \cdots b_{n-1} \in \Omega^*$, $b_0 \cdots b_{n-1}\mathtt{t}(q_n) \in \Omega^*$) is the *input* (resp. *output, final*) label of the path. A transducer is said to be *trim* if each state $q \in Q$ lies on a successful path.

A subsequential transducer $\mathcal{S}$ realizes a *subsequential* function $S : \Sigma^* \to \Omega^*$ such that $\forall x \in \mathrm{dom}(S)$, $S(x) = \mathtt{i}(q_- * x)\mathtt{t}(q_- \cdot x)$. The *order* of a function $\alpha : \Sigma^* \to \Omega^*$ is the maximal length of a word in $\mathrm{dom}(\alpha)$, the domain of α. The subsequential transducer $\mathcal{S}_p$ is deduced from $\mathcal{S}$ by letting p be the new initial state and ε be the initialization value. The function S_p realized by $\mathcal{S}_p$ is such that $\forall x \in \mathrm{dom}(S_p)$, $S_p(x) = (p * x)\mathtt{t}(p \cdot x)$. Two subsequential transducers $\mathcal{S}$ and $\mathcal{S}'$ are said to be *equivalent* if they realize the same function.

3. Cover transducers: basic properties

In this section, we state the definition of a cover transducer as well as additional definitions and propositions that are particularly useful in the sequel.

Let $\mathcal{S} = (\Sigma, \Omega, Q, q_-, \mathtt{i}, \mathtt{t}, \cdot, *)$ be a subsequential transducer. The *underlying automaton* of $\mathcal{S}$ is the automaton $A(\mathcal{S}) = (\Theta_{\mathcal{A}}, Q \cup \{q_s, q_t\}, q_s, \{q_t\}, \cdot_{\mathcal{A}})$ such that:
– $\Theta_{\mathcal{A}} = \{(a, b) \in \Sigma \times \Omega^* \mid \exists q \in Q \text{ s. t. } q * a = b\} \cup \{(\varepsilon, \mathtt{i})\} \cup \{(\varepsilon, \mathtt{t}(q)) \mid q \in \mathrm{dom}(\mathtt{t})\}$,
– $\forall q \in Q, \forall \theta = (a, b) \in \Theta_{\mathcal{A}}, q \cdot_{\mathcal{A}} \theta = q \cdot a$,
– $q_s \cdot_{\mathcal{A}} (\varepsilon, \mathtt{i}) = q_-$ and $\forall q \in \mathrm{dom}(\mathtt{t})$, $q \cdot_{\mathcal{A}} (\varepsilon, \mathtt{t}(q)) = q_t$.

The *underlying language* of $\mathcal{S}$ is the language $L(\mathcal{A})$ recognized by $A(\mathcal{S})$. Given an automaton $\mathcal{A}$ and a transducer $\mathcal{S}$ such as $\mathcal{A} = A(\mathcal{S})$, we say that is $\mathcal{S}$ is the *overlying transducer* of $\mathcal{A}$ and we write $\mathcal{S} = T(\mathcal{A})$.

Definition 4. *Let α be a function of order l. A subsequential transducer $\mathcal{S}$ is a cover transducer for α if for all $x \in \Sigma^{\leq l}$, $S(x) = \alpha(x)$.*

In the sequel, the restriction of the function S_p to $\Sigma^{\leq h}$ is denoted by S_p^h. Note that, by construction, the underlying automaton $\mathcal{A}$ of a cover transducer for a function α of order l is a cover automaton for the language $L'_{\mathcal{A}} = L(\mathcal{A}) \cap \Theta_{\mathcal{A}}^{\leq l+2}$.

Let y and z be two elements of Ω^*. The element z is said to be a *prefix* of y ($z \preceq y$) if there exists an element t of Ω^* such that $y = zt$. The element t is denoted by $z^{-1}y$. Let $E \subset \Omega^*$. We denote by $\bigwedge_{u \in E} u$ the *longest common prefix* (lcp for short) of the elements in E.

Definition 5. *Let $\mathcal{S}$ be a cover transducer for a function α of order l. Let p be a state of $\mathcal{S}$. We define the following longest common prefixes:*
– $\lambda_{\mathcal{S}}(p) = \bigwedge_{x \in \Sigma^*} S_p(x)$,
– $\nu_{\mathcal{S}}(p, h) = \bigwedge_{x \in \Sigma^{\leq h}} S_p(x)$, *with* $0 \leq h \leq \mathrm{height}(p)$,
– $\mu_{\mathcal{S}}(p) = \nu_{\mathcal{S}}(p, \mathrm{height}(p)) = \bigwedge_{x \in \Sigma^{\leq \mathrm{height}(p)}} S_p(x)$.

Lemma 6. *Let $\mathcal{S}$ be a cover transducer and p be a state of $\mathcal{S}$. The following relation holds:* $\lambda_{\mathcal{S}}(p) \preceq \mu_{\mathcal{S}}(p) = \nu_{\mathcal{S}}(p, \mathrm{height}(p)) \preceq \nu_{\mathcal{S}}(p, \mathrm{height}(p) - 1) \preceq \ldots \preceq \nu_{\mathcal{S}}(p, 0)$.

Definition 7. *Let $\mathcal{S} = (\Sigma, \Omega, Q, q_-, \mathtt{i}, \mathtt{t}, \cdot, *)$ be a subsequential transducer. The prefix transducer of $\mathcal{S}$ is the transducer $\mathcal{P} = (\Sigma, \Omega, Q, q_-, \mathtt{i}_{\mathcal{P}}, \mathtt{t}_{\mathcal{P}}, \cdot, *_{\mathcal{P}})$ such that:*
– $\mathtt{i}_{\mathcal{P}} = \mathtt{i}\lambda_{\mathcal{S}}(q_-)$ *and,* $\forall p \in \mathrm{dom}(\mathtt{t})$, $\mathtt{t}_{\mathcal{P}}(p) = \lambda_{\mathcal{S}}(p)^{-1}\mathtt{t}(\mathtt{p})$,
– $p *_{\mathcal{P}} a = \lambda_{\mathcal{S}}(p)^{-1}(p * a)\lambda_{\mathcal{S}}(p \cdot a)$, $\forall p \in Q$, $\forall a \in \Sigma$.

Proposition 8. [7] *The following properties hold:*
– *the transducers $\mathcal{P}$ and $\mathcal{S}$ are equivalent,*
– *the underlying automata of $\mathcal{P}$ and $\mathcal{S}$ are identical,*
– $\forall p \in Q$, $\forall x \in \Sigma^*$, $P_p(x) = \lambda_{\mathcal{S}}(p)^{-1}S_p(x)$, *and* $\lambda_{\mathcal{P}}(p) = \varepsilon$.

The prefix transducer of $\mathcal{S}$ is denoted by $P(\mathcal{S})$. In the following, we address both the general case when $\mathcal{S}$ is an arbitrary cover transducer, and the acyclic case when $\mathcal{S}$ realizes the function α. In the acyclic case, the following relation holds: $\forall p \in Q$, $\lambda_{\mathcal{S}}(p) = \mu_{\mathcal{S}}(p)$, and $\mathcal{P}$ enjoys specific properties due to the specific properties of $\mu_{\mathcal{S}}$. Therefore $\lambda_{\mathcal{S}}$ (resp. $\mathcal{P}$) is rather denoted $\mu_{\mathcal{S}}$ (resp. $\mathcal{M}$) in the acyclic case.

4. Merging relations

Let $\mathcal{S}$ be a cover transducer for a function α of order l. Our aim is to compute a reduced cover transducer. We first give a precise meaning to the notion of merging two states in a cover transducer.

Definition 9. *Let $\mathcal{S} = (\Sigma, \Omega, Q, q_-, \mathtt{i}, \mathtt{t}, \cdot, *)$ be a cover transducer for the function α of order l. Let $p, q \in Q$, $p \neq q$ and $q \neq q_-$ We consider the subsequential transducer $\mathcal{F}(\mathcal{S}, p, q) = (\Sigma, \Omega, Q_{\mathcal{F}}, q_-, \mathtt{i}, \mathtt{f}, \bullet, \star)$ such that:*
– $Q_{\mathcal{F}} = Q \setminus \{q\}$,
– $\forall r \in Q_{\mathcal{F}}$, $\mathtt{f}(r) = \mathtt{t}(r)$,
– $\forall r \in Q_{\mathcal{F}}$, $\forall a \in \Sigma$, $r \bullet a =$ ***if*** $r \cdot a \neq q$ ***then*** $r \cdot a$ ***else*** p.
– $\forall r \in Q_{\mathcal{F}}$, $\forall a \in \Sigma$, ***if*** $r \cdot a \neq q$ ***then*** $r \star a = r * a$.

We write $\mathcal{F}$ for $\mathcal{F}(\mathcal{S}, p, q)$ if there is no ambiguity. By construction the state q is removed; every in-going transition $(r, a, r{*}a, q)$ of q in $\mathcal{S}$ is replaced by an in-going transition $(r, a, r \star a, p)$ of p in $\mathcal{F}$, where $r \star a \in \Omega^*$ is a parameter that will be fixed later.

Definition 10. *Let $\mathcal{S}$ be a cover transducer for a function α of order l and Q be its set of states. A relation R on Q is said to be* a merging relation in $\mathcal{S}$ *if and only if for every pair $(p, q) \in Q \times Q$ such that pRq, the output function $\star$ of $\mathcal{F}(\mathcal{S}, p, q)$ can be fixed so that $\mathcal{F}(\mathcal{S}, p, q)$ be a cover transducer for α.*

4.1. *The merging relation $\approx_\mathcal{S}$*

For minimizing a subsequential transducer, it is natural to compare the functions S_p and S_q. For the covering problem, given two states p and q with $\mathrm{height}(p) \geq \mathrm{height}(q) = h$, it is natural to compare the functions S_p^h and S_q^h. The most general relation we can think of in a cover transducer $\mathcal{S}$ seems to be the following.

Definition 11. *The relation $\approx_1$ on Q is such that $p \approx_1 q$ is equivalent to the two following conditions:*
(i) $\mathrm{height}(p) \geq \mathrm{height}(q) = h$*,*
(ii) there exist $\beta, \gamma \in \Omega^*$ such that $\beta^{-1} S_p^h = \gamma^{-1} S_q^h$.

Notice that β depends on p and h and γ depends on q and h. Condition (ii) amounts to say that there exists a function $G : \Sigma^{\leq h} \to \Omega^*$ such that $S_p^h = \beta G$ and $S_q^h = \gamma G$. It implies that β is a prefix of $\nu_p = \nu_\mathcal{S}(p, h)$ and γ is a prefix of $\nu_q = \nu_\mathcal{S}(q, h)$.

Definition 12. *Let $\approx_0$ be the relation on Q such that $p \approx_0 q$ is equivalent to the two following conditions:*
(i) $\mathrm{height}(p) \geq \mathrm{height}(q) = h$*,*
(ii) $\nu_p^{-1} S_p^h = \nu_q^{-1} S_q^h$*.*

Lemma 13. *The relation $\approx_0$ is coarser than the relation $\approx_1$.*

Proof. We prove that $p \approx_1 q \Rightarrow p \approx_0 q$. We suppose that $p \approx_1 q$. We set $\varphi = \bigwedge_{x \in \Sigma^{\leq h}} G(x)$ and $H = \varphi G$. We have $\nu_p = \beta\varphi$ and $\nu_q = \gamma\varphi$. Consequently, $S_p^h = \beta\varphi H$ and $S_q^h = \gamma\varphi H$. Finally, $\mathcal{S}_p^h = \nu_p H$ and $S_q^h = \nu_q H$. Hence $p \approx_0 q$.

□

Let us examine conditions for the relation $\approx_0$ to be a merging relation in $\mathcal{S}$. Clearly, merging two states p and q such that $\mathrm{height}(p) \geq \mathrm{height}(q)$ implies that each transition $(r, a, r * a, q)$ in $\mathcal{S}$ be replaced by a transition $(r, a, r \star a, p)$ in $\mathcal{F}(\mathcal{S}, p, q)$, with $r \star a = (r{*}a)\nu_q\nu_p^{-1}$. Indeed, it would be convenient to be able to extract the function H by dividing S_p^h by $\nu(p, h)$, but it is generally not possible since it would lead to divide S_p by $\nu(p, h)$. Consequently it is necessary that ν_p be a suffix of $(r * a)\nu_q$, for all u such that (r, a, u, q) is a transition in $\mathcal{S}$. This condition is satisfied in particular when ν_p is a suffix of ν_q. Hence the definition:

Definition 14. *The relation $\approx_\mathcal{S}$ over Q is such that $p \approx_\mathcal{S} q$ is equivalent to the three following conditions:*

(i) $\mathrm{height}(p) \geq \mathrm{height}(q) = h$,
(ii) $\nu_{\mathcal{S}}(p,h)^{-1}S_p^h = \nu_{\mathcal{S}}(q,h)^{-1}S_q^h$,
(iii) there exists $\delta \in \Omega^*$ *such that* $\nu_{\mathcal{S}}(q,h) = \delta\nu_{\mathcal{S}}(p,h)$.

We now prove that the relation $\approx_{\mathcal{S}}$ is a merging relation in $\mathcal{S}$. The next proposition is a generalization of Lemma 17 in [4] that addresses the case of cover automata for a finite language.

Proposition 15. *The relation* $\approx_{\mathcal{S}}$ *is a merging relation in* $\mathcal{S}$.

Proof. The reasoning is similar to the one in the proof of Lemma 17 in [4]. As a consequence of Definition 14 the definition of the output function $\star$ of the transducer $\mathcal{F}(\mathcal{S},p,q)$ can be fixed by setting $r \star a = (r * a)\nu_q\nu_p^{-1} = (r * a)\delta$, for all pairs (r,a) in $Q_{\mathcal{F}} \times \Sigma$ such that $r \cdot a = q$. Moreover, the condition (ii) can be rewritten $p \approx_{\mathcal{S}} q \Rightarrow S_p^h = \delta^{-1}S_q^h$. We now prove that $p \approx_{\mathcal{S}} q \Rightarrow \mathcal{F}(\mathcal{S},p,q)$ is a cover transducer for α.

Let us first notice that since $\mathrm{height}(r) = l \Leftrightarrow r = q_-$, we have $q \neq q_-$ and thus q_- is a valid initial state for $\mathcal{F}$. Given the initial path with input label x in $\mathcal{S}$, we show that the corresponding path in $\mathcal{F}$ (that does not contain the state q) is such that $S(x) = F(x)$. Let $x \in \Sigma^{\leq l}$. We have to prove that $|\mathcal{F}|(x) = \alpha(x)$, which is equivalent to prove that $|\mathcal{F}|(x) = |\mathcal{S}|(x)$. If there is no prefix x_1 of x such that $q_- \cdot x_1 = q$, then the path from q_- to $q_- \cdot x_1$ in $\mathcal{S}$ is also a path in $\mathcal{F}$. Thus $|\mathcal{F}|(x) = |\mathcal{S}|(x)$. Otherwise, let $x = x_1'x_2$ where x_1' is the shortest prefix of x such that $q_- \cdot x_1' = q$. Since $q \neq q_-$, x_1 is not empty. We set $x_1' = x_1 a$, with $a \in \Sigma$. The initial path with input label x_1 in $\mathcal{S}$ has an output label equal to $(q_- * x_1) \cdot (r * a)$. By definition of the output function $\star$, the initial path with input label x_1 in $\mathcal{F}$ has an output label equal to $(q_- * x_1) \cdot (r * a)\delta$. Therefore, it suffices to prove that $F_p(x_2) = \delta^{-1}S_q(x_2)$.

First, consider the case $|x_2| = 0$. Since $\mathtt{f}(p) = \mathtt{t}(p)$, we have $F_p(\varepsilon) = S_p(\varepsilon)$. Since $p \approx_{\mathcal{S}} q$, we have $S_p(\varepsilon) = \delta^{-1}S_q(\varepsilon)$. Hence $F_p(\varepsilon) = \delta^{-1}S_q(\varepsilon)$. Suppose that the statement holds for $|x_2| < l'$, with $0 < l' \leq l - |x_1|$, which implies $l' \leq h$. Consider the case $|x_2| = l'$. If there is no nonempty prefix y of x_2 such that $p \cdot y = q$, then $F_p(x_2) = S_p(x_2)$. Since $p \approx_{\mathcal{S}} q$ and $|x_2| = l' \leq h$, we have $S_p(x_2) = \delta^{-1}S_q(x_2)$ and thus we get $F_p(x_2) = \delta^{-1}S_q(x_2)$. Otherwise, let $x_2 = yz$ where y is the shortest nonempty prefix of x_2 such that $p \cdot y = q$ (and $p \bullet y = p$). Then $|z| < l'$. By induction hypothesis, $F_p(z) = \delta^{-1}S_q(z)$. Therefore $F_p(yz) = (p \star y)F_p(z) = (p * y)S_q(z) = S_p(yz)$. Since $p \approx_{\mathcal{S}} q$ and $|yz| = l' \leq h$, we have $S_p(yz) = \delta^{-1}S_q(yz)$. Therefore $F_p(yz) = \delta^{-1}S_q(yz)$.

□

4.2. *Partitioning the set of states w.r.t. the relation* $\approx_{\mathcal{S}}$

We now show that the relation $\approx$ on $\Sigma^{\leq l}$ that underlies the relation $\approx_{\mathcal{S}}$ is not a similarity relation. Let $x \in \Sigma^{\leq l}$. For all $0 \leq h \leq l - |x|$ we set $N(x,h) = \bigwedge_{u \in \Sigma^{\leq h}} \alpha(xu)$.

Definition 16. *Let* α *be a function of order* l*. Let* x *and* y *be two words of* $\Sigma^{\leq l}$ *and* $h = l - \max\{|x|,|y|\}$*. The relation* $\approx$ *on* $\Sigma^{\leq l}$ *is such that* $x \approx y$ *is equivalent to the two following conditions:*

(i) $\forall u \in \Sigma^{\leq h}$, $N(x,h)^{-1}\alpha(xu) = N(y,h)^{-1}\alpha(yu)$,
(ii) there exists $D \in \Omega^*$ *such that* $N(y,h) = DN(x,h)$.

Since $\mathcal{S}$ is a cover transducer for α, the relations $\approx_{\mathcal{S}}$ and $\approx$ are such that $p \approx_{\mathcal{S}} q \Rightarrow (\forall (x,y) \mid q_{-} \cdot x = p$ and $q_{-} \cdot y = q), x \approx y$.

Lemma 17. *The relation* $\approx$ *on* $\Sigma^{\leq l}$ *is not a similarity relation.*

The merging relation $\approx_{\mathcal{S}}$ is based on a relation on $\Sigma^{\leq l}$ that enjoys no nice transitivity property. Consequently, finding a minimal partition of the set of states according to $\approx$ is *a priori* a more difficult problem than finding one according to a similarity relation. At the present time we do not know whether there exists an appropriate algorithm or not for computing a minimal partition according to the relation $\approx$.

5. Similarity relations

We now consider a very simple merging relation.

Definition 18. *Two different states* p *and* q *are said to be similar* ($p \sim_{\mathcal{S}} q$) *if the two following conditions are satisfied:*
(i) $\mathrm{height}(p) \geq \mathrm{height}(q) = h$,
(ii) $S_p^h = S_q^h$.

The merging relation $\sim_{\mathcal{S}}$ is underlied by a similarity relation $\sim$ on $\Sigma^{\leq l}$ and thus it can be computed efficiently.

Definition 19. *Let* α *be a function of order* l. *Let* x *and* y *be two words of* $\Sigma^{\leq l}$ *and* $h = l - \max\{|x|, |y|\}$. *The relation* $\sim$ *on* $\Sigma^{\leq l}$ *is defined by:*

$$x \sim y \Leftrightarrow (\forall u \in \Sigma^{\leq h}, \alpha(xu) = \alpha(yu))$$

Lemma 20. *The relation* $\sim$ *is a similarity relation on* $\Sigma^{\leq l}$.

Proof. The relation $\sim$ is reflexive and symmetric. Let us show that it is semi-transitive. Let x, y, z be words of $\Sigma^{\leq l}$ such that $|x| \leq |y| \leq |z|$. We first check that $x \sim y$ and $y \sim z \Rightarrow x \sim z$. Let $u \in \Sigma^{\leq l}$ such that $|u| \leq l - |z|$. Since $y \sim z$, we have $\alpha(yu) = \alpha(zu)$. Since $|y| \leq |z|$ and $x \sim y$, we have $\alpha(xu) = \alpha(yu)$. Consequently, $\alpha(xu) = \alpha(zu)$. Hence $x \sim z$. The proof of the second relation ($x \sim y$ and $x \sim z \Rightarrow y \sim z$) is similar.

□

Since $\mathcal{S}$ is a cover transducer for α, the relations $\sim_{\mathcal{S}}$ and $\sim$ are such that $p \sim_{\mathcal{S}} q \Rightarrow (\forall (x,y) \mid q_{-} \cdot x = p$ and $q_{-} \cdot y = q), x \sim y$. Consequently finding a minimal partition of Q according to the relation $\sim_{\mathcal{S}}$ can be achieved by computing a minimal partition of the relation $\sim$ on $\Sigma^{\leq l}$.

According to Proposition 8, the transducers $\mathcal{P}$ and $\mathcal{S}$ realize the same function. Thus $\mathcal{P}$ is a cover transducer for α and a relation $\sim_{\mathcal{P}}$ can be defined. The condition (ii) of Definition 18 is replaced by the condition $P_p^h = P_q^h$, that is equivalent to $\lambda_{\mathcal{S}}(p)^{-1}S_p^h = \lambda_{\mathcal{S}}(q)^{-1}S_q^h$. The relation $\sim_{\mathcal{P}}$ is a merging relation in $\mathcal{P}$ and it is underlied by a similarity relation on $\Sigma^{\leq l}$. In the acyclic case, we consider $\mathcal{M}$ instead of $\mathcal{P}$ and the relation $\sim_{\mathcal{M}}$ instead of $\sim_{\mathcal{P}}$.

Our aim is to compare the relations $\sim_{\mathcal{S}}$, $\sim_{\mathcal{P}}$ and $\sim_{\mathcal{M}}$. We write $\sim_1 \geq \sim_2$ if the relation $\sim_1$ is coarser than the relation $\sim_2$ and $\sim_1 \neq \sim_2$ if the relations $\sim_1$ and $\sim_2$ are incomparable.

5.1. *Relative merging power of the relations $\sim_{\mathcal{S}}$, $\sim_{\mathcal{P}}$ and $\sim_{\mathcal{M}}$*

We compare the power of the relations $\sim_{\mathcal{S}}$ and $\sim_{\mathcal{P}}$ in the general case, and the power of the relations $\sim_{\mathcal{S}}$ and $\sim_{\mathcal{M}}$ in the acyclic case. We also consider the restriction $\hat{\sim}_{\mathcal{S}}$ (resp. $\hat{\sim}_{\mathcal{P}}, \hat{\sim}_{\mathcal{M}}$) of the relation $\sim_{\mathcal{S}}$ (resp. $\sim_{\mathcal{P}}, \sim_{\mathcal{M}}$) to the set Q_Δ defined as follows. For all $0 \leq h \leq l$ we set $Q_h = \{p \in Q \mid \text{height}(p) = h\}$. We set $Q_\Delta = \cup_{0 \leq h \leq l} Q_h \times Q_h$.

The connection with the equivalence relation $\equiv_{\mathcal{S}}$ used for minimizing a subsequential transducer [8] is the following. By definition, we have $p \equiv_{\mathcal{S}} q \Leftrightarrow \forall x \in \Sigma^*, S_p(x) = S_q(x)$. It is easy to see that $p \equiv_{\mathcal{S}} q \Rightarrow \lambda_{\mathcal{S}}(p) = \lambda_{\mathcal{S}}(q)$ and that the relation $\equiv_{\mathcal{P}}$ is thus coarser than the relation $\equiv_{\mathcal{S}}$. On the opposite, we show that the relations $\sim_{\mathcal{S}}$ and $\sim_{\mathcal{P}}$ are incomparable. The reason is that the prefix $\lambda_{\mathcal{S}}(p)$ is computed on the set Σ^*, whereas similarity of p and q is checked on the set $\Sigma^{\leq h}$. Therefore there may exist two states p and q in Q, such that $p \sim_{\mathcal{S}} q$ and $\lambda_{\mathcal{S}}(p) \neq \lambda_{\mathcal{S}}(q)$. This explanation is made more precise by the following lemmas and proposition.

Lemma 21. *The following implication holds:*
$p \sim_{\mathcal{S}} q \Rightarrow \forall\, 0 \leq k \leq h, \nu_{\mathcal{S}}(p,k) = \nu_{\mathcal{S}}(q,k)$.

Lemma 22. *The following assertions hold:*
1) $p \sim_{\mathcal{S}} q \Rightarrow \lambda_{\mathcal{S}}(p) \preceq \lambda_{\mathcal{S}}(q)$ *or* $\lambda_{\mathcal{S}}(q) \preceq \lambda_{\mathcal{S}}(p)$,
2) $p \sim_{\mathcal{S}} q \Rightarrow \mu_{\mathcal{S}}(p) \preceq \mu_{\mathcal{S}}(q)$,
3) $(p \sim_{\mathcal{P}} q \Rightarrow p \sim_{\mathcal{S}} q) \Leftrightarrow \lambda_{\mathcal{S}}(p) = \lambda_{\mathcal{S}}(q)$,
4) $(p \sim_{\mathcal{M}} q \Rightarrow p \sim_{\mathcal{S}} q) \Leftrightarrow \mu_{\mathcal{S}}(p) = \mu_{\mathcal{S}}(q)$.

Proposition 23. *The following properties hold:*
1) *The relations $\sim_{\mathcal{S}}$ and $\sim_{\mathcal{P}}$ are incomparable.*
2) *The restrictions $\hat{\sim}_{\mathcal{S}}$ and $\hat{\sim}_{\mathcal{P}}$ are incomparable.*
3) *The relations $\sim_{\mathcal{S}}$ and $\sim_{\mathcal{M}}$ are incomparable.*
4) *The relation $\hat{\sim}_{\mathcal{M}}$ is coarser than $\hat{\sim}_{\mathcal{S}}$.*

Proof. 1) We first show that $(p \sim_{\mathcal{S}} q \not\Rightarrow p \sim_{\mathcal{P}} q)$. Obviously, $(p \sim_{\mathcal{S}} q \Rightarrow p \sim_{\mathcal{P}} q) \Leftrightarrow \lambda_{\mathcal{S}}(p) = \lambda_{\mathcal{S}}(q)$. By Lemma 22–1, we know that $p \sim_{\mathcal{S}} q \Rightarrow \lambda_{\mathcal{S}}(p) \preceq \lambda_{\mathcal{S}}(q)$ or $\lambda_{\mathcal{S}}(q) \preceq \lambda_{\mathcal{S}}(p)$. Therefore, $(p \sim_{\mathcal{S}} q \Rightarrow p \sim_{\mathcal{P}} q)$ is not necessarily true for all $q \in Q$.

We now show that $(p \sim_{\mathcal{P}} q \not\Rightarrow p \sim_{\mathcal{S}} q)$. By Lemma 22–3, $(p \sim_{\mathcal{P}} q \Rightarrow p \sim_{\mathcal{S}} q) \Leftrightarrow \lambda_{\mathcal{S}}(p) = \lambda_{\mathcal{S}}(q)$. Since it is possible to have simultaneously $p \sim_{\mathcal{P}} q$ and $\lambda_{\mathcal{S}}(p) \neq \lambda_{\mathcal{S}}(q)$, $(p \sim_{\mathcal{P}} q \Rightarrow p \sim_{\mathcal{S}} q)$ is not necessarily true for all $q \in Q$. We conclude that $\sim_{\mathcal{S}} \neq \sim_{\mathcal{P}}$.

2) By Lemma 21, $\lambda_{\mathcal{S}}(p) \preceq \nu_{\mathcal{S}}(p,h)$ and $\lambda_{\mathcal{S}}(q) \preceq \nu_{\mathcal{S}}(q,h)$. The comparison of the relations $\sim_{\mathcal{S}}$ and $\sim_{\mathcal{P}}$ does not depend on whether $\text{height}(p)$ and $\text{height}(q)$ are equal or not. Consequently, we have $\hat{\sim}_{\mathcal{S}} \neq \hat{\sim}_{\mathcal{P}}$.

3) Let us check that ($p \sim_{\mathcal{S}} q \not\Rightarrow p \sim_{\mathcal{M}} q$). It is clear that ($p \sim_{\mathcal{P}} q \Rightarrow p \sim_{\mathcal{M}} q$) $\Leftrightarrow$ $\mu_{\mathcal{S}}(p) = \mu_{\mathcal{S}}(q)$. By Lemma 22–2, we know that $p \sim_{\mathcal{S}} q \Rightarrow \mu_{\mathcal{S}}(p) \preceq \mu_{\mathcal{S}}(q)$. Therefore, ($p \sim_{\mathcal{S}} q \Rightarrow p \sim_{\mathcal{M}} q$) is not necessarily true for all $q \in Q$. On the other hand, proof of ($p \sim_{\mathcal{M}} q \not\Rightarrow p \sim_{\mathcal{S}} q$) is similar to case (2), using Lemma 22–4. We thus conclude that $\sim_{\mathcal{S}} \neq \sim_{\mathcal{M}}$.

4) Let $(p, q) \in Q_h \times Q_h$ and $p \sim_{\mathcal{S}} q$. Then by Lemma 21, $\mu_{\mathcal{S}}(p) = \nu_{\mathcal{S}}(p, h) = \nu_{\mathcal{S}}(q, h) = \mu_{\mathcal{S}}(q)$. Consequently, we have $p \sim_{\mathcal{S}} q \Rightarrow p \sim_{\mathcal{M}} q$. We conclude that $\hat{\sim}_{\mathcal{M}} \geq \hat{\sim}_{\mathcal{S}}$.

□

6. Reduction via the minimization of a cover automaton

Let $\mathcal{S}$ be a cover transducer for a function α of order l. We are concerned here by computing a reduced cover transducer from $\mathcal{S}$, through the minimization of the cover automaton associated either to the underlying automaton of $\mathcal{S}$ or to the underlying automaton of the prefix transducer of $\mathcal{S}$ (that is $\mathcal{P}$ for the general case or $\mathcal{M}$ for the acyclic case). More precisely, given a cover transducer $\mathcal{S}$, we consider the transducer $\mathcal{R}$ such that either $\mathcal{R} = \mathcal{S}$, $\mathcal{R} = \mathcal{P}$ or $\mathcal{R} = \mathcal{M}$ and we compute a reduced cover transducer $\mathcal{U}_{\mathcal{R}}$ according to the following scheme.

Proposition 24. *Let $\mathcal{U}_{\mathcal{R}}$ be computed from $\mathcal{R}$ as follows:*
1) *Consider the underlying automaton $\mathcal{A} = A(\mathcal{R})$ The automaton $\mathcal{A}$ is a cover automaton for the language $L'_{\mathcal{A}} = L(\mathcal{A}) \cap \Theta_{\mathcal{A}}^{\leq l+2}$. Compute a minimal cover automaton $\mathcal{C} = C(A(\mathcal{R}))$ from $A(\mathcal{R})$.*
2) *Let $\mathcal{U}_{\mathcal{R}} = T(C(A(\mathcal{R})))$ be the overlying transducer of $\mathcal{C}$.*
Then $\mathcal{U}_{\mathcal{R}}$ is a cover transducer for α and it has fewer states than $\mathcal{S}$.

The proof of Proposition 24 is based on the two following lemmas.

Lemma 25. *Let $\mathcal{S}$ be a subsequential transducer and $\mathcal{A} = A(\mathcal{S})$ be its underlying automaton. Let $\Theta = \Theta_{\mathcal{A}}$ be the alphabet of $\mathcal{A}$. The set of the successful paths of $\mathcal{S}$ (resp. $\mathcal{A}$) is denoted by $\Pi_{\mathcal{S}}$ (resp. $\Pi_{\mathcal{A}}$). Let $q_0 = q_{\_}$ The following properties are equivalent:*

(1) There exists a path $\pi_{\mathcal{S}} \in \Pi_{\mathcal{S}}$ such that $\pi_{\mathcal{S}} = ((q_i, x_i, u_i, q_{i+1}))_{0 \leq i < m}$, with $x_i \in \Sigma$ and $\forall\, 0 \leq i < m$, $u_i \in \Omega^*$.
(2) Let $x = x_0 \ldots x_{m-1}$ (resp. $u = u_1 \ldots u_{m-1}$) be the input (resp. output) label of the path $\pi_{\mathcal{S}}$. The function realized by $\mathcal{S}$ is such that: $S(x) = \mathtt{i}\ u\ \mathtt{t}(q_{\_} \cdot x)$.
(3) Let $\pi_{\mathcal{A}} = ((q_s, (\varepsilon, \mathtt{i}), q_0), ((q_i, (x_i, u_i), q_{i+1}))_{0 \leq i < m}, (q_m, (\varepsilon, \mathtt{t}(q_m)), q_t))$, with $\forall$ $0 \leq i < m$, $(x_i, u_i) \in \Sigma \times \Omega^*$, Then the path $\pi_{\mathcal{A}}$ belongs to $\Pi_{\mathcal{A}}$.
(4) Let $a = (\varepsilon, \mathtt{i})(x_0, u_0) \ldots (x_{m-1}, u_{m-1})(\varepsilon, \mathtt{t}(q_m))$ be the label of $\pi_{\mathcal{A}}$. Then the word $a \in \Theta^*$ belongs to $L(\mathcal{A})$.

Lemma 26. *Let $\mathcal{S}$ be a cover transducer for a function α of order l and $\mathcal{A} = A(\mathcal{S})$ be its underlying automaton on the alphabet Θ. Let $L'_{\mathcal{A}} = L(\mathcal{A})) \cap \Theta^{\leq l+2}$. By construction, the automaton $\mathcal{A}$ is a cover automaton for the language $L'_{\mathcal{A}}$.*

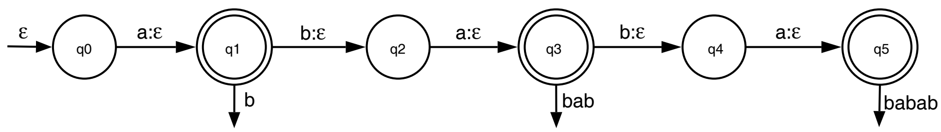


Fig. 1. The prefix-tree transducer $\mathcal{S}$ realizing α.

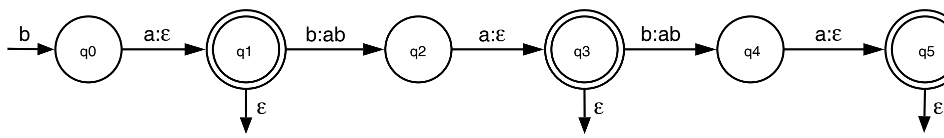


Fig. 2. The prefix transducer $P(\mathcal{S})$.

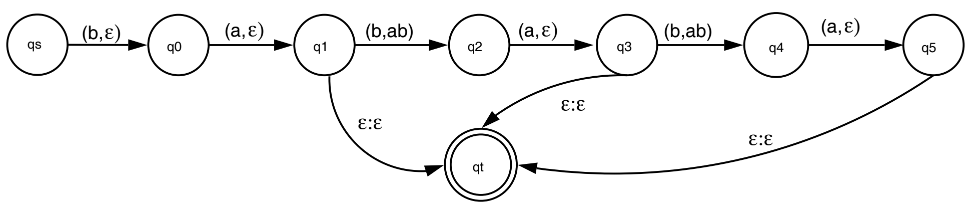


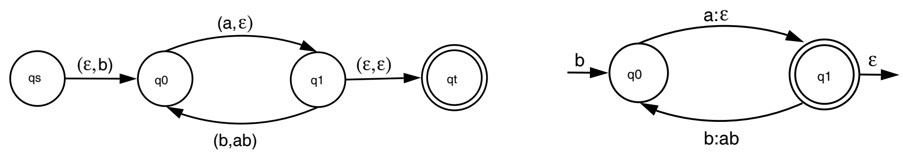
Fig. 3. The underlying automaton $A(P(\mathcal{S}))$.

Let $\mathcal{C} = C(A(\mathcal{S}))$ be a minimal cover automaton for the language $L'_{\mathcal{A}}$. Then the overlying transducer $\mathcal{U}_{\mathcal{S}} = T(C(A(\mathcal{S})))$ of $\mathcal{C}$ is a cover transducer for α.

Proof. (of Proposition 24) By Lemma 26, the transducer $\mathcal{U}_{\mathcal{R}} = T(C(A(\mathcal{R})))$ is a cover transducer for α in each of the three cases $\mathcal{R} = \mathcal{S}$, $\mathcal{R} = P(\mathcal{S})$ and $\mathcal{R} = M(\mathcal{S})$. By construction, we have $|T(C(A(\mathcal{R})))| = |C(A(\mathcal{R}))| - 2 \leq |A(\mathcal{R})| - 2 = |\mathcal{R}|$. Since $|\mathcal{S}| = |\mathcal{R}| = |\mathcal{M}|$, we have $|\mathcal{U}_{\mathcal{R}}| \leq |\mathcal{S}|$.

□

Following Körner's example [13] for cover automata, we consider the function $\alpha : \{a,b\}^* \to \{a,b\}^*$ such that $\mathrm{dom}(\alpha) = \{a, aba, ababa\}$ and $\alpha(a) = b$, $\alpha(aba) = bab$, $\alpha(ababa) = babab$. Let $\mathcal{S}$ be the prefix-tree transducer $\mathcal{S}$ that realizes α (Figure 1). The construction of the reduced cover transducer $T(C(A(P(\mathcal{S}))))$ is illustrated by Figures 2, 3, 4(a) and 4(b).



(a) $C(A(P(\mathcal{S})))$ (b) $T(C(A(P(\mathcal{S}))))$

Fig. 4. A minimal cover automaton $C(A(P(\mathcal{S})))$ (a) and its overlying transducer $T(C(A(P(\mathcal{S}))))$ (b).

6.1. *Power of reductions based on a minimal cover automaton*

Given a cover transducer $\mathcal{S}$ and the relation $\sim_{\mathcal{S}}$, we define the associated relation $\sim_{A(\mathcal{S})}$ on the set of states of the underlying automaton $A(\mathcal{S})$. Let Q be the set of states of $\mathcal{S}$ and $\mathrm{height}(q)$ be the height of q in $\mathcal{S}$. By convention, we set $\mathrm{height}_{A(\mathcal{S})}(q_s) = l + 1$ and $\mathrm{height}_{A(\mathcal{S})}(q_t) = -1$ so that, for all $q \in Q$, $\mathrm{height}_{A(\mathcal{S})}(q) = \mathrm{height}(q)$. Let p and q be two states of $A(\mathcal{S})$ such that $h = \mathrm{height}_{A(\mathcal{S})}(q) \leq \mathrm{height}_{A(\mathcal{S})}(p)$.

Definition 27. *The relation $\sim_{A(\mathcal{S})}$ on $Q \cup \{q_s, q_t\}$ is defined by*

$$p \sim_{A(\mathcal{S})} q \Leftrightarrow \overrightarrow{L_p^{A(\mathcal{S})}} \cap \Theta_{\mathcal{A}}^{\leq h+1} = \overrightarrow{L_q^{A(\mathcal{S})}} \cap \Theta_{\mathcal{A}}^{\leq h+1}$$

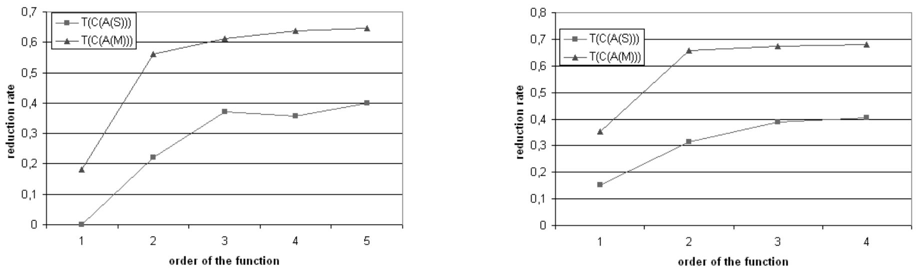
Since q_t is the unique final state, for all $q \neq q_t$, we have $q_t \not\sim_{A(\mathcal{S})} q$. On the other hand, the state $q_s \cdot (\varepsilon, \mathtt{i}) = q_-$ is not final whereas for all $q \neq q_s$, the state $q \cdot (\varepsilon, \mathtt{i})$ is either not defined or final. Hence, for all $q \neq q_s$, we have $q_s \not\sim_{A(\mathcal{S})} q$. Consequently, $\sim_{A(\mathcal{S})}$ can be considered as a relation on Q.

We consider the transducer $\mathcal{R}$ with either $\mathcal{R} = \mathcal{S}$, $\mathcal{R} = \mathcal{P}$ (general case) or $\mathcal{R} = \mathcal{M}$ (acyclic case). The relations $\sim_{A(\mathcal{P})}$ and $\sim_{A(\mathcal{M})}$ are defined in the same way as the relation $\sim_{A(\mathcal{S})}$. We first compare the relations $\sim_{A(\mathcal{R})}$ and $\sim_{\mathcal{R}}$, for $\mathcal{R} = \mathcal{S}$, $\mathcal{R} = \mathcal{P}$ and $\mathcal{R} = \mathcal{M}$. Then we compare the relations $\sim_{A(\mathcal{S})}$ and $\sim_{A(\mathcal{P})}$ (general case) and $\sim_{A(\mathcal{S})}$ and $\sim_{A(\mathcal{M})}$ (acyclic case) as well as their restrictions to Q_Δ.

Proposition 28. 1) *For $\mathcal{R} = \mathcal{S}$, $\mathcal{R} = \mathcal{P}$ and $\mathcal{R} = \mathcal{M}$, the relation $\sim_{\mathcal{R}}$ is coarser than the relation $\sim_{A(\mathcal{R})}$.*
2) *The relations $\hat{\sim}_{\mathcal{M}}$ and $\hat{\sim}_{A(\mathcal{M})}$ are equivalent.*

Proposition 30 is similar to Proposition 23. Its proof is based on the following lemma.

Lemma 29. *The following assertions hold:*
1) $p \sim_{A(\mathcal{S})} q \Rightarrow \lambda_{\mathcal{S}}(p) \preceq \lambda_{\mathcal{S}}(q)$ *or* $\lambda_{\mathcal{S}}(q) \preceq \lambda_{\mathcal{S}}(p)$,
2) $p \sim_{A(\mathcal{P})} q \Rightarrow \mu_{\mathcal{S}}(p) \preceq \mu_{\mathcal{S}}(q)$,
3) $(p \sim_{A(\mathcal{P})} q \Rightarrow p \sim_{A(\mathcal{S})} q) \Rightarrow \lambda_{\mathcal{S}}(p) = \lambda_{\mathcal{S}}(q)$,
4) $(p \sim_{A(\mathcal{M})} q \Rightarrow p \sim_{A(\mathcal{S})} q) \Rightarrow \mu_{\mathcal{S}}(p) = \mu_{\mathcal{S}}(q)$.



(a) Size of the alphabet = 4

(b) Size of the alphabet = 5

Fig. 5. Acyclic case: relative size of $T(C(A(\mathcal{S})))$ and $T(C(A(\mathcal{M})))$ w.r.t. the order l of the function. The size of the input and output alphabets is 4 (a) or 5 (b).

Proposition 30. *The following properties hold:*
1) *The relations* $\sim_{A(\mathcal{S})}$ *and* $\sim_{A(\mathcal{P})}$ *are incomparable.*
2) *The restrictions* $\hat{\sim}_{A(\mathcal{S})}$ *and* $\hat{\sim}_{A(\mathcal{P})}$ *are incomparable.*
3) *The relations* $\sim_{A(\mathcal{S})}$ *and* $\sim_{A(\mathcal{M})}$ *are incomparable.*
4) *The relation* $\hat{\sim}_{A(\mathcal{M})}$ *is coarser than* $\hat{\sim}_{A(\mathcal{S})}$.

According to Proposition 28-1, a relation defined on a transducer is coarser than the associated relation on the underlying automaton; thus the technique based on the minimization of a cover automaton is only an approximation for the problem of reducing a cover transducer.

However, as a corollary of Proposition 28-2 and Proposition 30-4 it turns out that it is relevant to use the prefix transducer of $\mathcal{S}$ for solving the acyclic case. Indeed, the acyclic case is the only one where the restriction to Q_Δ of the transducer relation is not coarser than the restriction of the corresponding automaton relation. Moreover, the relations $\sim_{A(\mathcal{S})}$ and $\sim_{A(\mathcal{M})}$ being incomparable, and the restriction $\hat{\sim}_{A(\mathcal{M})}$ being coarser than the restriction $\hat{\sim}_{A(\mathcal{S})}$, it can be expected that $\sim_{A(\mathcal{M})}$ has a better average reduction ratio than $\sim_{A(\mathcal{S})}$.

7. Experimental results

The scheme described by Proposition 24 was implemented using the Xerox tools for creating and manipulating finite state automata and transducers: XFST $[11, 2]$ (unweighted case), and WFSC $[12]$ (weighted case). A command was developed that combines the construction of a prefix transducer according to the algorithm of Béal and Carton $[1]$, and the computation of a minimal cover automaton according to the algorithm of Körner $[13]$. Given an acyclic cover transducer $\mathcal{S}$ for the function α, this command computes the cover transducers $T(C(A(\mathcal{S})))$ and $T(C(A(P(\mathcal{S}))))$.

Numerous tests were carried out in the acyclic case, with the following features: the size of the input and ouput alphabets are identical and they rank from 2 to 5; functions are randomly generated; more than fifty percent of the states are final states. The respective reduction ratios of the cover transducers $T(C(A(\mathcal{S})))$ and $T(C(A(\mathcal{M})))$ (w.r.t. the order l of the function) are given by Figure 5(a) and Figure 5(b).

Two main observations can be made from these experimental results: first, the reduction ratio increases with the order of the function for both relations $\sim_{A(\mathcal{S})}$ and $\sim_{A(\mathcal{M})}$; secondly, the reduction ratio is significantly greater for the transducer $T(C(A(\mathcal{M})))$ than for the transducer $T(C(A(\mathcal{S})))$: hence we check that the relation $\sim_{A(\mathcal{M})}$ has a better average reduction ratio than the relation $\sim_{A(\mathcal{S})}$.

8. Conclusion

The notion of a cover transducer for a function with finite domain can be defined following the notion of a cover automaton for a finite language. The problem of reducing a cover transducer is, however, not as simple. On the one hand, merging relations have no nice transitivity property and the computation of a minimal partition is thus difficult. On the other hand, using finer similarity relations and a technique based on minimizing the underlying cover automaton yields an approximation that is only relevant for the acyclic case.

References

[1] M.-P. Béal and O. Carton, *Computing the Prefix of an Automaton*, RAIRO Theoret. Comput. Sci. **34:6** (2001), 503–515.

[2] K. R. Beesley and L. Karttunen, *Finite State Morphology*, CLSI Publications, Palo Alto, CA, USA, 2003.

[3] J. Berstel, *Transductions and Context-Free Languages*, Teubner Sttutgart, 1979.

[4] C. Câmpeanu, N. Sântean, and S. Yu, *Minimal Cover-automata for Finite Languages*, Theoret. Comput. Sci. **267** (2001), 3–16.

[5] C. Câmpeanu, A. Păun and S. Yu, *An Efficient Algorithm for Constructing Minimal Cover Automata for Finite Languages*, Intern. J. of Foundations of Comput. Sc., **13-1**(2002), 99–113.

[6] J.-M. Champarnaud, F. Guingne and G. Hansel, *Similarity Relations and Cover Automata*, RAIRO-Info. Theor. Appl., **39**(2005), 115–123.

[7] C. Choffrut, *Contribution à l'étude de quelques familles remarquables de fonctions rationnelles*, Thèse d'Etat, Université Paris VII, 1978.

[8] C. Choffrut, *Minimizing Subsequential Transducers: A Survey*, Theoret. Comput. Sci. **292** (2003), no. 1, 131–143.

[9] C. Dwork and L. Stockmeyer, *A Time Complexity Gap for Two-Way Probabilistic Finite-State Automata*, SIAM J. on Computing, **19** (1990), 1011–1023.

[10] J. Kaneps and R. Friedvalds, *Running Time to Recognize Non-Regular Languages by 2-Way Probabilistic Automata*, in ICALP'91, Lecture Notes in Computer Science, Springer-Verlag, 510(1991), 174–185.

[11] L. Karttunen, T. Gaál, R. M. Kaplan, A. Kempe, P. Tapanainen and T. Yampol, *Xerox Finite-State Home Page*, 1996–2004, Xerox Research Centre Europe, Grenoble, France. URL: http://www.xrce.xerox.com/competencies/content-analysis/fst/.

[12] A. Kempe, C. Baeijs, T. Gaál, F. Guingne and F. Nicart, WFSC – A New Weighted Finite State Compiler, in CIAA'2003, Lecture Notes in Computer Science, O. Ibarra and Z. Dang eds., Springer-Verlag, 2759(2003), 108–119.

[13] H. Körner, *A Time and Space Efficient Algorithm for Minimizing Cover Automata for Finite Languages*, Int. J. of Foundations of Comput. Sci. **14** (2003), 1071–1086.

[14] M. Mohri, *Minimization Algorithms for Sequential Transducers*, Theoret. Comput. Sci. **234** (2000), no. 1–2, 177–201.

[15] N. Sântean, *Towards a Minimal Representation for Finite Languages: Theory and Practice*, Master's Thesis, The University of Western Ontario, 2000.

[16] M. P. Schützenberger, *Sur une variante des fonctions séquentielles*, Theoret. Comput. Sci. **11** (1977), 47–57.

[17] S. Yu, *Regular Languages*, in G. Rozenberg and A. Salomaa eds., Handbook of Formal Languages, Springer, Berlin, 1997.