Theoretical Computer Science ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Predictable semiautomata[☆]

Janusz Brzozowski [a,*], Nicolae Santean [b]

[a] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1

[b] Computer and Information Sciences, Indiana University South Bend, South Bend, IN 46634, USA

ARTICLE INFO

Keywords:
Automaton
Bound
Delegator
Look-ahead
Nondeterminism
Predictor
Semiautomaton
Simulation

ABSTRACT

We introduce a new class of nondeterministic semiautomata: A nondeterministic semiautomaton $\mathcal{S}$ is predictable if there exists $k \geq 0$ such that, if $\mathcal{S}$ knows the current input a and the next k inputs, the transition under a can be made deterministically. Nondeterminism may occur only when the length of the unread input is $\leq k$. We develop a theory of predictable semiautomata. We show that, if a semiautomaton with n states is k-predictable, but not $(k-1)$-predictable, then $k \leq (n^2-n)/2$, and this bound can be reached for a suitable input alphabet. We characterize k-predictable semiautomata, and introduce the predictor semiautomaton, based on a look-ahead semiautomaton. The predictor is essentially deterministic and simulates a nondeterministic semiautomaton by finding the set of states reachable by a word w, if it belongs to the language L of the semiautomaton (*i.e.*, if it defines a path from an initial state to some state), or by stopping as soon as it infers that $w \notin L$. Membership in L can be decided deterministically.

1. Introduction

Nondeterministic finite automata (NFAs) are ubiquitous. They serve as models for nondeterministic processes, constitute design tools (arguably more convenient than deterministic ones), and are often inevitable. They also have drawbacks, such as increased simulation time and space, and inefficient minimization algorithms.

Several attempts have been made to overcome the disadvantages of nondeterminism. NFAs have been used as models for service-oriented computing [1], and as tools for automated web service composition [4,5]. In both cases, it was imperative to enforce deterministic behavior without changing the structure of the model, *i.e.*, without determinization. For this purpose, "delegators" of NFAs were informally introduced in [4,5]. A delegator is a deterministic finite automaton (DFA) based on the transition graph of the NFA. It has a look-ahead buffer of a fixed length that permits it to determine which of several possible nondeterministic steps should be taken in such a way that, for each word accepted by the NFA, the delegator has a deterministic computation. Look-ahead delegation was studied systematically and in a more abstract framework in [9].

We do not solve the delegator problem, nor determinize NFAs. Instead, we provide a method in which a nondeterministic system can be used essentially deterministically. We use semiautomata (automata without accepting states), because nondeterminism involves transitions, rather than accepting states. We introduce "predictable" semiautomata, in which it is possible to replace a nondeterministic step by a deterministic one, with the aid of a bounded number of input letters from a look-ahead buffer. We compute the set of states reachable by input words from the initial states of a semiautomaton with as little nondeterminism as possible. We also treat nondeterminism as a local, rather than global, phenomenon. Thus our theory is substantially different from the work in [4,5,9].

[☆] This research was supported by the Natural Sciences and Engineering Research Council of Canada, Grant OGP0000871 and Fellowship PDF-32888-2006.

[*] Corresponding address: David R. Cheriton School of Computer Science, University of Waterloo, 200 University Ave. W., Waterloo, ON, Canada N2L 3G1. Tel.: +1 519 885 5015; fax: +1 519 885 1208.

E-mail addresses: brzozo@uwaterloo.ca (J. Brzozowski), nsantean@iusb.edu (N. Santean).

0304-3975/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2009.05.010

Please cite this article in press as: J. Brzozowski, N. Santean, Predictable semiautomata, Theoretical Computer Science (2009), doi:10.1016/j.tcs.2009.05.010

The remainder of the paper is organized as follows. Section 2 defines the notation and terminology related to semiautomata in general, introduces predictable semiautomata, and gives a polynomial-time algorithm for testing for predictability. In Section 3 we show that the upper bound for an n-state semiautomaton over a 1-letter alphabet is $n-1$, and this bound can be met. We then show that the general upper bound is $(n^2-n)/2$, and this bound can be met with a suitable input alphabet. The properties of certain types of words, called "minimal selectors" and "maximal nonselectors", and their relation to predictability are studied in Section 4. Section 5 describes the construction of auxiliary "product semiautomata", which provide a test for predictability and a method of finding minimal selectors and maximal nonselectors. In Section 6 we show that only a part of the product semiautomaton, called the "core", is necessary for our purposes. Semiautomata with look-ahead are defined in Section 7; these "predictors" permit us to simulate a nondeterministic semiautomaton almost deterministically. Section 8 concludes the paper.

2. Predictable semiautomata

For a set X, we denote its cardinality by $X\#$. If Σ is an alphabet, then Σ^+ and Σ^* denote the free semigroup and the free monoid, respectively, generated by Σ. The empty word is 1. For $k \geq 1$, let $\Sigma^{\leq k} = 1 \cup \Sigma \cup \cdots \cup \Sigma^k$. For $w \in \Sigma^*$, $|w|$ denotes the length of w. If $w = uv$, for some $u, v \in \Sigma^*$, then u is a *prefix* of w. A language L is *prefix-closed* if $uv \in L$ implies $u \in L$. If $u \in \Sigma^*$, $v \in \Sigma^+$, then uv is an *extension* of u.

A *semiautomaton* [6] $\mathcal{S} = (\Sigma, Q, P, E)$ consists of an *alphabet* Σ, a set Q of *states*, a set $P \subseteq Q$ of *initial states*, and a set E of *edges* of the form (q, a, r), where $q, r \in Q$ and $a \in \Sigma$. An edge (q, a, r) *begins* at q, *ends* at r, and has *label* a. It is also denoted as $q \xrightarrow{a} r$. A *path* π is a finite sequence $\pi = (q_0, a_1, q_1)(q_1, a_2, q_2) \cdots (q_{k-1}, a_k, q_k)$ of consecutive edges, $k > 0$ being its *length*, q_0, its *beginning*, q_k, its *end*, and word $w = a_1 \cdots a_k$, its *label*. We also write $q_0 \xrightarrow{w} q_k$ for π. Each state q has a *null path* 1_q from q to q with label 1.

If $T \subseteq Q$ and $w \in \Sigma^*$, then $Tw = \{q \in Q \mid t \xrightarrow{w} q,\ \text{for some } t \in T\}$ is the set of states reachable by w from states in T. If $T = \{t\}$, we write tw for Tw; if $Tw = \{q\}$, we write $Tw = q$. A state q of $\mathcal{S}$ is *accessible* if there exists $p \in P$ and $w \in \Sigma^*$ such that $q \in pw$, *i.e.*, such that there is a path $p \xrightarrow{w} q$. A semiautomaton is *accessible* if all of its states are accessible. For any semiautomaton $\mathcal{S}$, the *accessible semiautomaton of* $\mathcal{S}$ is $\mathcal{S}^A$ and is obtained by deleting every state q that is not reachable from any initial state, as well as all transitions leading to and coming from q. Formally, if $\mathcal{S} = (\Sigma, Q, P, E)$ is a semiautomaton, then $\mathcal{S}^A = (\Sigma, Q^A, P, E^A)$ is defined as follows. The set Q^A consists of all states q for which there exists a path $\pi = (q_0, a_1, q_1) \cdots (q_{k-1}, a_k, q_k)$, with $q_0 \in P$ and $q_k = q$. An edge e belongs to E^A if and only if e appears in such a path from some initial state.

For $T \subseteq Q$, the *language of* T is denoted by R_T and it is the set of all labels of paths from states in T, *i.e.*, $R_T = \{w \in \Sigma^* \mid Tw \neq \emptyset\}$. Note that R_T is prefix-closed. The *language* $|\mathcal{S}|$ *of a semiautomaton* $\mathcal{S} = (\Sigma, Q, P, E)$ is $|\mathcal{S}| = R_P$. If $q \in Q$, the *language of* q is $R_q = R_{\{q\}}$.

Semiautomaton $\mathcal{S}$ is *complete* if $P \neq \emptyset$ and, for every $q \in Q$ and $a \in \Sigma$, there is an edge $(q, a, r) \in E$, for some $r \in Q$. If $\mathcal{S}$ is complete, then $R_q = \Sigma^*$ for all $q \in Q$; hence $|\mathcal{S}| = \Sigma^*$.

A semiautomaton $\mathcal{S}$ is *deterministic* if it has at most one initial state, and for every $q \in Q$, $a \in \Sigma$, there is at most one edge (q, a, r). If $\mathcal{S}$ is deterministic and has initial state p, we write $\mathcal{S} = (\Sigma, Q, p, E)$.

Occasionally we find it useful to allow empty-word transitions in semiautomata; these are edges in the graph of a semiautomaton of the form $(q, 1, r)$. We refer to such semiautomata as *1-semiautomata*. Such notions as paths, path labels, and accessibility are extended to 1-semiautomata in the natural way.

We now introduce nondeterministic semiautomata, called "predictable", in which the knowledge of a bounded number of symbols read ahead from the input tape removes nondeterminism. We restrict our attention to finite semiautomata, that is, to semiautomata in which Σ and Q are finite.

Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a semiautomaton. If $q \in Q$, $a \in \Sigma$, then a *fork* (with origin q and input a) is the set $\langle q, a \rangle = \{(q, a, r_1), \ldots, (q, a, r_h)\}$ of all the edges from q labeled a. The *fork set* of $\langle q, a \rangle$ is $\langle\langle q, a \rangle\rangle = \{r_1, \ldots, r_h\}$. We assume that $h > 0$. Note, however, that forks with single edges are permitted; they are called *deterministic transitions*. Allowing such forks has the advantage that a semiautomaton consists only of a set of initial states and forks. A set $T \subseteq Q$ is *critical* if either $T = P$ or $T = \langle\langle q, a \rangle\rangle$, for a fork $\langle q, a \rangle$ in $\mathcal{S}$.

Definition 1. Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a semiautomaton, and $k \geq 0$, an integer. A set $T \subseteq Q$ is *k-predictable* if any two distinct states $s, t \in T$ satisfy $R_s \cap R_t \cap \Sigma^k = \emptyset$. Also $\mathcal{S}$ is *k-predictable* if every critical set of $\mathcal{S}$ is k-predictable, and $\mathcal{S}$ is *predictable* if it is k-predictable for some k. The *index* of a predictable semiautomaton $\mathcal{S}$ is the smallest k such that $\mathcal{S}$ is k-predictable.

One verifies that a semiautomaton $\mathcal{S}$ is k-predictable if and only if it is "deterministic $(k+1)$-look-ahead" [8]. A set is 0-predictable if and only if it consists of a single state. Thus, $\mathcal{S}$ is 0-predictable if and only if it is deterministic, *i.e.*, its critical sets are singletons.

Note that, for $k > 0$, a k-predictable semiautomaton $\mathcal{S}$ must be incomplete. If it were complete, then we would have $R_s = R_t = \Sigma^*$, and $R_s \cap R_t \cap \Sigma^k = \Sigma^k$ would be nonempty, for all $s, t \in Q$, $s \neq t$, and for all $k \geq 0$. Thus the condition for k-predictability could not be satisfied for any k. Since a 0-predictable semiautomaton is deterministic, it follows that a k-predictable semiautomaton is either deterministic or incomplete.

If a set is k-predictable, then it is k'-predictable for all $k' > k$, since R_s and R_t are prefix-closed.

Please cite this article in press as: J. Brzozowski, N. Santean, Predictable semiautomata, Theoretical Computer Science (2009), doi:10.1016/j.tcs.2009.05.010

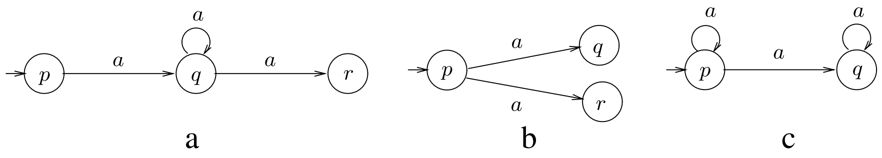

Fig. 1. Illustrating predictability.

Example 2. In Fig. 1(a), fork $\langle p, a\rangle$ is a deterministic transition, and fork set $\langle\langle q, a\rangle\rangle = \{q, r\}$ is 1-predictable. Fork set $\{q, r\}$ in Fig. 1(b) is 1-predictable, because there are no words of length 1 in R_q or R_r. Fork set $\{p, q\}$ in Fig. 1(c) is not k-predictable for any $k \geq 0$, because $a^k \in R_p \cap R_q \cap \Sigma^k$ for all k. ■

We now describe a method for testing whether a set T of states of a semiautomaton $\mathcal{S} = (\Sigma, Q, P, E)$ is predictable. Suppose Q has n states and Σ has m letters; then $\mathcal{S}$ has at most mn^2 edges. For $t_i, t_j \in T$, $t_i \neq t_j$, let $\mathcal{S}_i = (\Sigma, Q, \{t_i\}, E)$ and $\mathcal{S}_j = (\Sigma, Q, \{t_j\}, E)$ be copies of $\mathcal{S}$ with initial states t_i and t_j, respectively.

The *square*[1] of $\mathcal{S}$ [2] is the semiautomaton

$$\mathcal{S}^{\square} = \mathcal{S} \times \mathcal{S} = (\Sigma, Q \times Q, P \times P, E^{\square}),$$

where $((q_1, q_2), a, (r_1, r_2)) \in E^{\square}$ if and only if $(q_1, a, r_1), \in E$ and $(q_2, a, r_2) \in E$. Note that $\mathcal{S}^{\square}$ has n^2 states and at most mn^4 edges. Let $\mathcal{S}^{\square}_{i,j} = (\Sigma, Q, \{t_i, t_j\}, E^{\square})$ be $\mathcal{S}^{\square}$ with initial state changed to $\{t_i, t_j\}$.

For $i < j$, the *nondeterministic product* of $\mathcal{S}_i$ and $\mathcal{S}_j$ is the accessible semiautomaton $\mathcal{S}^{A}_{i,j} = (\Sigma, Q_{i,j}, \{(t_i, t_j)\}, E_{i,j})$ of $\mathcal{S}^{\square}_{i,j}$. Here $Q_{i,j}$ is the set of all states from $Q \times Q$ that are reachable from the initial state (t_i, t_j) in $\mathcal{S}^{\square}$, and $E_{i,j}$ is the set of edges from $E^{\square}$ that appear on some path from (t_i, t_j). Let $L_{i,j} = R_{(t_i,t_j)}$ be the language of $\mathcal{S}_{i,j}$.

For the next theorem it is convenient to use a 1-semiautomaton, that is, a semiautomaton with empty-word transitions. We define a *(nondeterministic) pair 1-semiautomaton*

$$\mathcal{N}_T = (\Sigma, Q_{\mathcal{N}}, q_0, E_{\mathcal{N}}),$$

where

$$\begin{aligned} Q_{\mathcal{N}} &= \{q_0\} \cup Q \times Q, \quad \text{with } q_0 \notin Q \times Q,\\ P_{\mathcal{N}} &= \{(t_i, t_j) \mid t_i, t_j \in T, i < j\},\\ E_{\mathcal{N}} &= \{(q_0, 1, (t_i, t_j)) \mid (t_i, t_j) \in P_{\mathcal{N}}\} \cup \bigcup_{\{t_i, t_j \in T, i \neq j\}} E_{i,j}. \end{aligned}$$

Theorem 3. *A set T is predictable if and only if $\mathcal{N}_T$ has no cycles. Moreover, T is of index k if and only if the length of a longest path from q_0 in $\mathcal{N}_T$ is k. If $\mathcal{S}$ has n states and is predictable with index k, then $k \leq n^2$. Predictability and k-predictability can be tested in time polynomial in the size of $\mathcal{N}_T$.*

Proof. Suppose $\mathcal{N}_T$ has a cycle. Then there exists a path

$$((q_1, q'_1), a_1, (q_2, q'_2)), \ldots, ((q_{\alpha-1}, q'_{\alpha-1}), a_{\alpha-1}, (q_\alpha, q'_\alpha)), \ldots, \ldots, ((q_{\beta-1}, q'_{\beta-1}), a_{\beta-1}, (q_\beta, q'_\beta)),$$

where $(q_1, q'_1) = (t_i, t_j) \in P_{\mathcal{N}}$, and $(q_\beta, q'_\beta) = (q_\alpha, q'_\alpha)$. Thus $w_r = uv^r = a_1 \cdots a_{\alpha-1}(a_\alpha \cdots a_{\beta-1})^r$ is in $L_{i,j}$ for every r. By the construction of $\mathcal{S}_{i,j}$, we have paths

$$\begin{aligned} &(q_1, a_1, q_2), \ldots, (q_{\alpha-1}, a_{\alpha-1}, q_\alpha), \ldots, (q_{\beta-1}, a_{\beta-1}, q_\beta),\\ &(q'_1, a_1, q'_2), \ldots, (q'_{\alpha-1}, a_{\alpha-1}, q'_\alpha), \ldots, (q'_{\beta-1}, a_{\beta-1}, q'_\beta) \end{aligned}$$

in $\mathcal{S}_i$ and $\mathcal{S}_j$, showing that $w_r \in R_{t_i} \cap R_{t_j}$. Since r can be arbitrarily large, T cannot be predictable. Thus, if T is predictable, then $\mathcal{N}_T$ has no cycles.

Conversely, suppose $\mathcal{N}_T$ has no cycles. Let the length of a longest path from q_0 be k. Since the label of the first edge in any path from q_0 is the empty word, the length of a longest word in $L_{i,j}$ is $k - 1$, since such a word starts in (t_i, t_j). Then, over all i, j such that $(t_i, t_j) \in P_{\mathcal{N}}$, we have $R_{t_i} \cap R_{t_j} \cap \Sigma^k = \emptyset$ for all i, j, and T is k-predictable, but not $(k - 1)$-predictable.

Since there are $n^2 + 1$ states in $\mathcal{N}_T$, the length of a longest path in $\mathcal{N}_T$ is $\leq n^2$ and the length of a longest word in the language of $\mathcal{N}_T$ is $\leq n^2 - 1$; hence we have $k \leq n^2$. If we repeat this for all critical sets T, and if the length of a longest path in $\mathcal{N}_T$ is k, then $\mathcal{S}$ is predictable with index k, where $k \leq n^2$.

Semiautomaton $\mathcal{N}_T$ has at most $1+n^2$ states and at most $n(n-1)/2+mn^4$ edges. Since testing whether $\mathcal{N}_T$ has a cycle and finding a longest path in $\mathcal{N}_N$ can be done in time linear in the size of the semiautomaton, predictability and k-predictability can be tested in polynomial time. □

[1] A similar construction was used in [7] under the name *state-pair graph*.

Please cite this article in press as: J. Brzozowski, N. Santean, Predictable semiautomata, Theoretical Computer Science (2009), doi:10.1016/j.tcs.2009.05.010

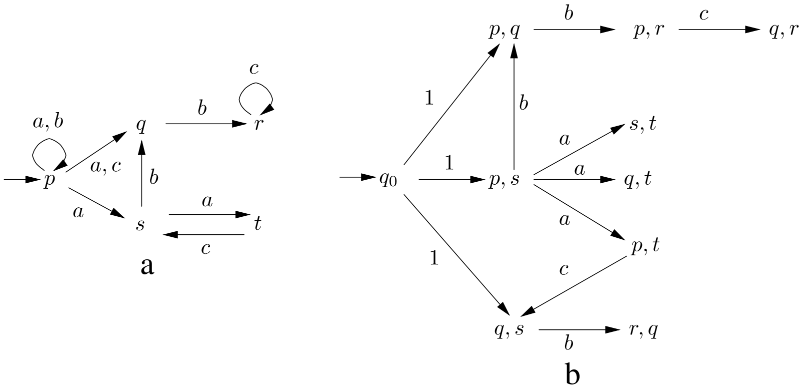

Fig. 2. A semiautomaton and its nondeterministic pair semiautomaton.

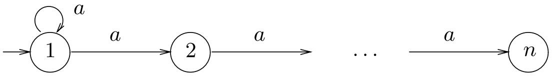

Fig. 3. An n-state unary semiautomaton which is $(n-1)$-predictable.

The upper bound of n^2 for k will be improved in the next section.

Example 4. The semiautomaton of Fig. 2 (a) has 5 states and one nontrivial fork $\langle p, a\rangle$. The nondeterministic pair semiautomaton $\mathcal{N}_T$ for the fork set $T = \{p, q, s\}$ is shown in Fig. 2 (b). There are three semiautomata $\mathcal{S}_{p,q}$, $\mathcal{S}_{p,s}$, and $\mathcal{S}_{q,s}$. Since there are no cycles, T, and hence also $\mathcal{S}$, are predictable. The length of a longest path

$$(q_0, 1, (p, s)), ((p, s), a, (p, t)), ((p, t), c, (q, s)), ((q, s), b, (r, q))$$

is 4; hence $\mathcal{S}$ has predictability index 4. ■

3. Predictability bounds

The case of a one-letter alphabet is special.

Proposition 5. *Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a semiautomaton over a one-letter alphabet. If $\mathcal{S}$ has n states and $k \geq 0$ is its predictability index, then $k \leq n - 1$.*

Proof. If $k = 0$, the bound is trivially satisfied. Hence assume that there is at least one critical set $T = \{t_1, \ldots, t_h\}$, $h \geq 2$, which is k-predictable.

We claim that T is predictable if and only if at most one of the languages $\{R_{t_i}\}_{1\leq i\leq h}$ is infinite. Note first that, if a language L over one letter a is prefix-closed, then L is infinite if and only if $L = a^*$. For $1 \leq i \neq j \leq h$, if R_{t_i} and R_{t_j} are infinite then $R_{t_i} \cap R_{t_j} = a^*$, since R_{t_i} and R_{t_j} are prefix-closed. Hence T is not predictable, and R_{t_i} and R_{t_j} cannot both be infinite.

Without loss of generality, assume now that $R_{t_1}, \ldots, R_{t_{h-1}}$ are finite. We distinguish two cases:

(1) R_{t_h} is infinite. Let w be a longest word in $\bigcup_{1\leq i<h} R_{t_i}$, and assume that $w \in R_{t_j}, j \neq h$. Since R_{t_j} is finite, no path originating in t_j and spelling w can have a state repeated. For suppose that $w = uxv$, for some $x \in \Sigma^+$, $u, v \in \Sigma^*$, and $t_j u = t_j ux$. Then also $ux^2v \in R_{t_j}$, contradicting that w is a longest word of R_{t_j}. We also observe that a path π from t_j spelling w cannot visit t_h; otherwise R_{t_j} would be infinite, since R_{t_h} is infinite. Thus π has at most $n-1$ states, and $|w| \leq n - 2$. Now T cannot be $|w|$-predictable, because $w \in R_{t_j} \cap R_{t_h}$, but it is $(|w|+1)$-predictable. Thus we must have $k = |w|+1 \leq n-1$.

(2) R_{t_h} is finite. Let w be a longest word in $\bigcup_{1\leq i\leq h} R_{t_i}$, and assume $w \in R_{t_j}$. A path originating in t_j and spelling w can involve at most n states (it can possibly visit t_h); thus $|w| \leq n - 1$. If $|w| < n - 1$ then clearly $k \leq |w| + 1 \leq n - 1$. When $|w| = n - 1$, a path π originating in t_j and spelling w uses all the states of $\mathcal{S}$. There cannot be another path originating in t_i, $i \neq j$, spelling w; for then there would be a loop, contradicting the finiteness of R_{t_j}. Thus, $k = |w| = n - 1$ in this case.

The semiautomaton in Fig. 3 has n states and is $(n-1)$-predictable; thus the bound can be reached when $|\mathcal{S}|$ is infinite. If we remove the loop in Fig. 3 and make states 1 and 2 initial, we reach the bound when $|\mathcal{S}|$ is finite. □

Before addressing the case of a general alphabet, we develop some necessary conditions on paths originating from critical sets in predictable semiautomata.

Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a predictable semiautomaton, with state set $Q = \{1, \ldots, n\}$, and let r_1, s_1 be two distinct states of a critical set in $\mathcal{S}$. If $w = a_1 \cdots a_{m-1}$ is a longest word in $R_{r_1} \cap R_{s_1}$, let $\pi_1 = (r_1, a_1, r_2) \cdots (r_{m-1}, a_{m-1}, r_m)$ and $\pi_2 = (s_1, a_1, s_2) \cdots (s_{m-1}, a_{m-1}, s_m)$ be two paths spelling w, originating from r_1 and s_1, respectively. Let $1 \leq i < j \leq m$. We define three *path conditions* on the states that appear in such paths:

(1) Either $r_i \neq r_j$ or $s_i \neq s_j$.
(2) Either $r_i \neq s_j$ or $r_j \neq s_i$.
(3) If $r_i = s_i$, then $r_j \neq r_i$ and $s_j \neq r_i$.

We now show that these path conditions must be always satisfied; otherwise, the word w cannot be longest.

Lemma 6. *Let π_1 and π_2 be two paths as defined above. Then the sequence $\mathcal{L} = (r_1, s_1), \ldots, (r_m, s_m)$ of ordered pairs of states encountered by π_1 and π_2 must satisfy the path conditions.*

Proof. By our hypothesis, $r_1 \neq s_1$. For the remaining conditions we have:

(1) If $r_i = r_j$ and $s_i = s_j$, let x be the label of the path $r_i, \ldots, r_j$ in π_1. Then $w = uxv$ for some $u, v \in \Sigma^*$, and $ux^2v \in R_{r_1} \cap R_{s_1}$, contradicting the maximality of $|w|$.
(2) If $r_i = s_j$ and $r_j = s_i$, let x be the label of the path $r_i, \ldots, r_j$ in π_1, and hence also the label of the path $s_i, \ldots, s_j$ in π_2, and let $w = uxv$. Then one verifies that $ux^3v \in R_{r_1} \cap R_{s_1}$, contradicting the maximality of $|w|$.
(3) If $r_i = s_i$ and $r_j = r_i$, let x be the label of the path $r_i, \ldots, r_j = r_i$ in π_1, and let $w = uxv$. Then $ux^2v \in R_{r_1} \cap R_{s_1}$, contradicting the maximality of $|w|$. Similarly, if $s_j = r_i$, let x be the label of the path $s_i, \ldots, s_j$ in π_2, and let $w = uxv$. Since $s_i = s_j = r_i$, there is a loop labeled x on r_i, and again $ux^2v \in R_{r_1} \cap R_{s_1}$. □

Lemma 7. *Let $n > 0$, and let $\mathcal{L} = (r_1, s_1), \ldots, (r_m, s_m)$ be a sequence of ordered pairs of elements from $\{1, \ldots, n\}$. If $\mathcal{L}$ satisfies $r_1 \neq s_1$ and the path conditions, then $m \leq (n^2 - n)/2$ and the bound can be reached.*

Proof. We first show that the bound can be reached. Condition (1) is satisfied by the sequence $\mathcal{L} = (1, 1), \ldots, (1, n), (2, 1), \ldots, (2, n), \ldots, (n, 1), \ldots, (n, n)$, which has length n^2. If we remove pairs (i, i), for all $1 \leq i \leq n$, we have a sequence of length $n^2 - n$, in which $r_1 \neq s_1$, and which satisfies Condition (3) as well. Finally, for all $i \neq j$, remove either (i, j) or (j, i), but not both. Now the sequence also satisfies Condition (2) and has length $(n^2 - n)/2$.

Next, we proceed by induction on n. If $n = 1$, then only the empty sequence satisfies all the conditions. Here $m = 0 = (n^2 - n)/2$. If $n = 2$, only the empty sequence, $(1, 2)$ and $(2, 1)$ satisfy the conditions. Thus $m \leq 1 = (n^2 - n)/2$.

For any $n > 0$, let $M(n)$ be the length of a longest sequence of pairs of elements from $\{1, \ldots, n\}$ satisfying all the conditions. Assume that $M(n-1) \leq ((n-1)^2 - (n-1))/2$, for some $(n-1) \geq 2$. Let $\mathcal{L}$ be a sequence with $M(n)$ pairs satisfying all the conditions, and assume that $M(n) > (n^2 - n)/2$. If $M(n) > (n^2 - n)/2$ and $n \geq 3$, then $M(n) > n$. Thus $\mathcal{L}$ contains at least $n + 1$ pairs. There are at most $2n - 1$ pairs involving n, namely the pairs of the form (n, i) and (i, n). However, if both (n, i) and (i, n) appear in $\mathcal{L}$, then Condition (2) is violated. Hence there are at most n pairs involving n, and at least one pair (i, j) not involving n. Without loss of generality we may assume that the first pair of $\mathcal{L}$ does not contain n, for if it did, we could interchange it with the pair (i, j).

Let $\mathcal{L}'$ be the sequence with m' pairs obtained from $\mathcal{L}$ by removing all the pairs containing n. Then $\mathcal{L}'$ satisfies all the conditions as well, and its elements are from the set $\{1, \ldots, n-1\}$. By the induction hypothesis,

$$m' \leq M(n-1) \leq ((n-1)^2 - (n-1))/2 = (n^2 - n)/2 - (n-1).$$

In addition to the pairs of $\mathcal{L}'$, $\mathcal{L}$ contains pairs from the set

$$\{(1, n), (2, n), \ldots, (n, n), (n, 1), (n, 2), \ldots, (n, n-1)\}.$$

If $\mathcal{L}$ contains (n, n), then it cannot contain any other pair involving n, for this would violate Condition (3). Hence,

$$M(n) = m' + 1 \leq (n^2 - n)/2 - (n-2) < (n^2 - n)/2,$$

which contradicts our assumption. If $\mathcal{L}$ does not contain (n, n), it contains at most $(n-1)$ pairs involving n. Now $M(n) \leq m' + (n-1) \leq (n^2 - n)/2$, which is again a contradiction. Thus, $M(n) \leq (n^2 - n)/2$ and the induction step goes through. □

Theorem 8. *Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a semiautomaton, $n = Q\#$, and $k \geq 0$, the predictability index of $\mathcal{S}$. If $\Sigma\# > 1$, then $k \leq (n^2 - n)/2$, and this bound is reachable for a suitable Σ.*

Proof. First, we prove that the bound is reachable for a suitable alphabet. Let $n \geq 1$, and let $\mathcal{L} = (r_1, s_1), \ldots, (r_k, s_k)$ be a sequence satisfying the conditions of Lemma 7, with $k = (n^2 - n)/2$. Consider the semiautomaton $\mathcal{S} = (\Sigma, Q, P, E)$, with $\Sigma = \{a_1, \ldots, a_k\}$, $Q = \{1, \ldots, n\}$, $P = \{r_1\}$ and $E = E_1 \cup E_r \cup E_s$, where $E_1 = \{(r_1, a_1, r_1), (r_1, a_1, s_1)\}$, $E_r = \{(r_i, a_{i+1}, r_{i+1}) \mid 1 \leq i < k\}$, and $E_s = \{(s_i, a_{i+1}, s_{i+1}) \mid 1 \leq i < k\}$. We show that $\mathcal{S}$ is k-predictable, but not $(k-1)$-predictable.

Observe that $E_1 = \langle r_1, a_1 \rangle$ is a fork in $\mathcal{S}$ and its fork set $\langle\langle r_1, a_1 \rangle\rangle = \{r_1, s_1\}$ is critical. Consider $w = a_2 \cdots a_k$; clearly, $w \in R_{r_1} \cap R_{s_1}$, implying that $\mathcal{S}$ is not $(k-1)$-predictable, since $|w| = k-1$. Also $R_{r_k} \cap R_{s_k} = \emptyset$; for if both (r_k, a_j, r) and (s_k, a_j, s) were in E for some $r, s \in Q$ and $j \in \{1, \ldots, m\}$, then either $(r_k, s_k) = (r_{j-1}, s_{j-1})$ for $j > 1$, or $(r_k, s_k) = (r_1, s_1)$ for $j = 1$. In both cases, this would violate path Condition (1). Thus w is a longest word in $R_{r_1} \cap R_{s_1}$, implying that $\langle\langle r_1, a_1 \rangle\rangle$ is k-predictable.

Since $P\# = 1$, P is 0-predictable. If there exists a fork other than $\langle r_1, a_1 \rangle$ in $\mathcal{S}$, then there must be a pair (r_i, s_i) in $\mathcal{L}$ with $r_i = s_i$, since only such states have outgoing edges with a same label, by the construction of $\mathcal{S}$. By the argument in the proof of Lemma 7, $\mathcal{L}$ cannot be of maximal length. Hence there are no other forks, and $\mathcal{S}$ has index k.

In summary, the minimal predictability bound of $(n^2 - n)/2$ can be reached, possibly at the cost of a large alphabet. □

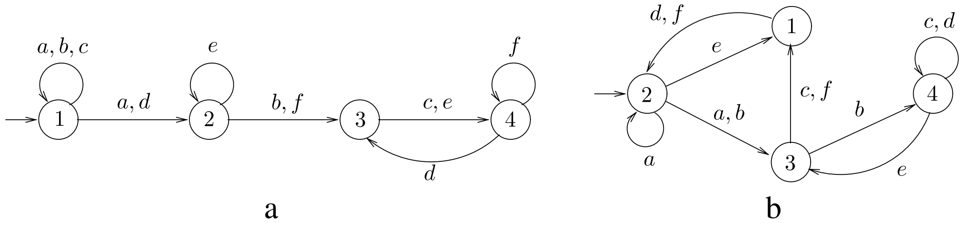

Fig. 4. Semiautomata reaching the predictability bound.

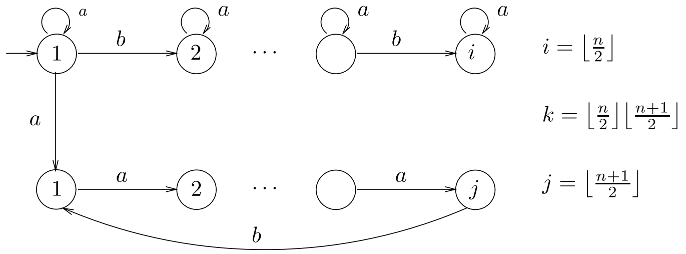
a
a
a
a
1
b
2
. . .
b
i
$i = \lfloor \frac{n}{2} \rfloor$
a
$k = \lfloor \frac{n}{2} \rfloor \lfloor \frac{n+1}{2} \rfloor$
1
a
2
. . .
a
j
$j = \lfloor \frac{n+1}{2} \rfloor$
b

Fig. 5. A k-predictable semiautomaton over a two-letter alphabet.

Example 9. For $n = 4$, the semiautomata in Fig. 4(a) and 4(b) correspond to the sequences

$$\mathcal{L}_1 = \{(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)\}$$
$$\text{and } \mathcal{L}_2 = \{(2,3), (3,4), (1,4), (2,4), (1,3), (2,1)\},$$

respectively. Both sequences obey the conditions of Lemma 7 and are of maximal length. Therefore, both semiautomata are 6-predictable, and reach the upper bound for 4 states. ■

Remark 10. For fixed alphabets, the bounds may turn out to be smaller. The semiautomaton in Fig. 5 over a two-letter alphabet has n states and is k-predictable, where

$$k = \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n+1}{2} \right\rfloor.$$

When n is even, then $k = n^2/4$, and when n is odd, $k = (n^2 - 1)/4$. Thus, in both cases we have $k \le n^2/4$—a value strictly smaller than the upper bound of $n(n-1)/2$ when $n > 2$; when $n = 2$ the bound is 1 and is reached. In general, we do not know whether the bound in Theorem 8 can be reached for a binary alphabet or a larger fixed alphabet.

4. Selectors and nonselectors

We classify a word w in the language R_T of a set $T \subseteq Q$ as a "t-selector in T", if it originates in $t \in T$ (is the label of a path from t) and in no other state of T, or as a "t-nonselector in T", if it originates in t and in at least one other state of T. These words play a key role in predictability. Selectors are look-ahead words that permit us to choose only one state from a set T, whereas nonselectors limit the choice to a subset of T that has at least two states.

We use the symbol $\stackrel{\triangle}{=}$ to mean "is by definition".

Definition 11. If $\mathcal{S} = (\Sigma, Q, P, E)$ is a semiautomaton, $T = \{t_1, \ldots, t_h\} \subseteq Q$, then a word $w \in \Sigma^*$ is a t_i-*selector in* T if

$$w \in \sigma(t_i, T) \stackrel{\triangle}{=} \Big(R_{t_i} \setminus \bigcup_{j \in \{1,\ldots,h\}, j \neq i} R_{t_j} \Big).$$

Word w is a t_i-*nonselector in* T if

$$w \in \overline{\sigma}(t_i, T) \stackrel{\triangle}{=} R_{t_i} \setminus \sigma(t_i, T) = \bigcup_{1 \le j \le h, j \neq i} (R_{t_i} \cap R_{t_j}).$$

Word w is a *selector in* T if it is a t_i-selector in T for some t_i, *i.e.*, if

$$w \in \sigma(T) \stackrel{\triangle}{=} \bigcup_{i=1}^{h} \sigma(t_i, T).$$

Please cite this article in press as: J. Brzozowski, N. Santean, Predictable semiautomata, Theoretical Computer Science (2009), doi:10.1016/j.tcs.2009.05.010

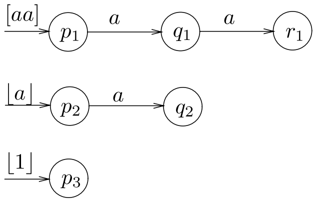

Fig. 6. Selectors and nonselectors.

Word w is a *nonselector in* T if it is a t_i-nonselector in T for some t_i, *i.e.*, if

$$w \in \overline{\sigma}(T) \overset{\triangle}{=} \bigcup_{i=1}^{h} \overline{\sigma}(t_i, T) = R_T \setminus \sigma(T) = \bigcup_{1\le i\ne j\le h} (R_{t_i} \cap R_{t_j}).$$

A selector w is *minimal* if no prefix of w is a selector. A t_i-nonselector u is *maximal* if no extension of u is in R_{t_i}.

Example 12. In Fig. 1(a), the set of q-selectors in the fork set $\langle\langle p, a\rangle\rangle = \{q\}$ is a^*, and 1 is the only minimal q-selector in $\{q\}$. There are no q-nonselectors in $\{q\}$. Fork $\langle q, a\rangle$ has critical set $T = \{q, r\}$. The set of q-selectors in T is a^+, and a is a minimal q-selector in T. The empty word 1 is the only q-nonselector in T, and it is not maximal because $a = 1a \in R_q$. There are no r-selectors in T, and 1 is the only r-nonselector in T; it is maximal because no extension of 1 is in R_r.

In Fig. 1(b), the fork set $\langle\langle p, a\rangle\rangle$ is $T = \{q, r\}$. Here $R_T = \{1\}$, there are no selectors, and 1 is a maximal q-nonselector in T and a maximal r-nonselector in T. Thus, there exist sets that are predictable and yet have no selectors. Also, a set that is not predictable may have selectors, as we shall see in Example 24.

In Fig. 1(c), there is a fork $\langle p, a\rangle$ with fork set $T = \{p, q\}$. There are no selectors, since $R_p \cap R_q = a^*$. Every word in a^* is a p-nonselector and a q-nonselector, and there are no maximal nonselectors. ■

When necessary, we denote minimal selectors in square brackets and maximal nonselectors in "floor" brackets, as in the next example.

Example 13. The semiautomaton $\mathcal{S}$ of Fig. 6 illustrates the usefulness of minimal selectors and maximal nonselectors. The only critical set with more than one element is $P = \{p_1, p_2, p_3\}$. One verifies that $\mathcal{S}$ is 2-predictable. There is a minimal p_1-selector aa, and maximal nonselectors a for p_2, and 1 for p_3. In the figure, $[aa]$ is a minimal selector and $\lfloor a\rfloor$ is a maximal nonselector.

If the input word to $\mathcal{S}$ is 1, then any state in P can be the initial state, and there is no further computation. If the input word is a, then the initial state could not be p_3, but is limited to $\{p_1, p_2\}$. Finally, if the input word begins with aa, then the initial state is necessarily p_1. ■

Selectors and nonselectors have the following prefix properties:

Proposition 14. *Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a semiautomaton and $T \subseteq Q$.*

(1) *Every extension of a t-selector is a t-selector.*
(2) *The set of all nonselectors in T is prefix-closed.*
(3) *If an s-selector u is a prefix of any word w in R_t, then $s = t$.*
(4) *The set of all minimal selectors in T is prefix-free.*
(5) *No selector is a prefix of a nonselector.*
(6) *For any $t \in T$, no maximal t-nonselector is a prefix of a t-selector.*
(7) *For any $t \in T$, the set of all maximal t-nonselectors is prefix-free.*

Proof. (1) If u is a t-selector and uv is a t-nonselector, then $uv \in R_s$ for some $s \in T$, $s \ne t$. Since R_s is prefix-closed, also $u \in R_s$, which is a contradiction.
(2) Word w is a nonselector if and only if there exist $s, t \in T$, such that $w \in R_s \cap R_t$. Since R_s and R_t are prefix-closed, we have $u \in R_s \cap R_t$, for every prefix u of w.
(3) This follows from (1).
(4) This follows from the definition of minimal selector.
(5) This follows from (2).
(6) If u is a maximal t-nonselector, then $ua \notin R_t$, for all $a \in \Sigma$. Hence no extension of a maximal t-nonselector is in R_t.
(7) This follows by the same reasoning as (6). □

Theorem 15. *Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a semiautomaton and $T \subseteq Q$. The following are equivalent: (1) T is k-predictable. (2) Every word of length k in R_T is a selector in T. (3) Every word of length $\ge k$ in R_T has a minimal selector in T as a prefix. (4) Every nonselector is of length $< k$.*

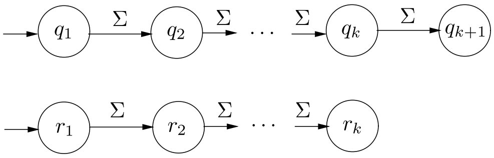

Fig. 7. A semiautomaton with many minimal selectors and maximal nonselectors.

Proof. (1) ⇒ (2) Suppose $w \in \Sigma^k$. If w is nonselector in T, then $w \in R_s \cap R_t \cap \Sigma^k$, for some $s, t \in T$, contradicting (1). Hence w must be a selector in T.

(2) ⇒ (3) Every word w of length $\geq k$ in R_T has a prefix u of length k, and u is a selector in T by (2). By definition of minimal selector, u (and hence also w) must have a minimal selector as a prefix.

(3) ⇒ (4) If $|w| \geq k$, then w has a minimal selector as a prefix by (3). Then w is an extension of a selector, and must itself be a selector. Hence, if w is a nonselector in T, then $|w| < k$.

(4) ⇒ (1) If a longest nonselector in T is of length $< k$, then $R_s \cap R_t \cap \Sigma^k = \emptyset$, for all $s, t \in T$, $s \neq t$, and T is k-predictable. □

Corollary 16. *If T is a k-predictable set of a semiautomaton $\mathcal{S}$, then every minimal selector in T is of length $\leq k$.*

Proposition 17. *Let $\mathcal{S} = (\Sigma, Q, P, E)$, let $T \subseteq Q$, and let $t \in T$. If T is k-predictable, then there exists either a minimal t-selector in T or a maximal t-nonselector in T.*

Proof. Note first that R_t is not empty, since $1 \in R_t$. If t has a selector in T, then it has a minimal selector in T. Assume now that t has no selectors in T. If R_t is finite, let w be a longest word in R_t, necessarily a nonselector. Then $wa \notin R_t$ for all $a \in \Sigma$, and w is a maximal t-nonselector in T. By Theorem 15(4), the case where R_t is infinite is impossible. □

We now show that the number of minimal selectors and maximal nonselectors of a state of a k-predictable semiautomaton may be exponential in k.

Proposition 18. *Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a semiautomaton with an alphabet of m letters, let $T \subseteq Q$ be a k-predictable set, and let $t \in T$. Then the number S_t of minimal t-selectors (N_t of maximal t-nonselectors) is bounded from above by m^k (m^{k-1}), and this bound is reachable.*

Proof. Suppose that T is a k-predictable set. Each $w \in R_T \cap \Sigma^k$ has a prefix which is a minimal selector. Since the set of all minimal selectors is prefix-free, w has exactly one such prefix. Since there are at most m^k such words w, we have $S_t \leq m^k$. Similarly, every maximal nonselector is of length $\leq k - 1$; thus it is a prefix of a word of length $k - 1$. Since the set of all maximal t-nonselectors is prefix-free, we have $N_t \leq m^{k-1}$.

Now consider the semiautomaton of Fig. 7. There are $2k + 1$ states, and only one nontrivial critical set, $\{q_1, r_1\}$, the set of initial states. Since $R_{q_1} \cap R_{r_1} \cap \Sigma^k = \emptyset$, the semiautomaton is k-predictable. Every word of length k is a minimal q_1-selector; thus there are m^k minimal selectors. Also, every word of length $k - 1$ is a maximal q_1-nonselector (and a maximal r_1-nonselector). Thus there are m^{k-1} maximal nonselectors. □

By Theorem 8, $k \leq (n^2 - n)/2$; hence $S_t \leq m^{(n^2-n)/2}$ and $N_t \leq m^{(n^2-n)/2-1}$. The semiautomaton in Fig. 7 has $n = 2k + 1$ states. For that automaton $S_t = m^{(n-1)/2}$ and $N_t = m^{(n-1)/2-1}$, which shows that S_t and N_t can be exponential in the number of states. We do not know whether the bounds $m^{(n^2-n)/2}$ and $m^{(n^2-n)/2-1}$ can be met.

5. Product semiautomata

We now introduce deterministic "product semiautomata", whose definition involves subset constructions. They explain the nature and the role of selectors and nonselectors, and provide another test for predictability.

Recall that, to test for predictability of a set $T = \{t_1, \dots, t_h\} \subseteq Q$ of states of $\mathcal{S} = (\Sigma, Q, P, E)$, we need to show that there exists a k such that $R_{t_i} \cap R_{t_j} \cap \Sigma^k = \emptyset$ for all i, j such that $i \neq j$ and $1 \leq i, j \leq h$. The language R_{t_i} is defined by the nondeterministic semiautomaton $\mathcal{S}_i = (\Sigma, Q, \{t_i\}, E)$. We first determinize each such semiautomaton using the subset construction and obtain a deterministic semiautomaton $\mathcal{D}_i = (\Sigma, 2^Q, t_i, E_i)$, where 2^Q is the set of all subsets of Q. Then we construct the direct product $\mathcal{D}_1 \times \cdots \times \mathcal{D}_h$ of these h deterministic semiautomata in order to study intersections of their languages. The direct product of h copies of 2^Q is denoted by $(2^Q)^h$.

Definition 19. If $\mathcal{S} = (\Sigma, Q, P, E)$ is a semiautomaton and $T = \{t_1, \dots, t_h\} \subseteq Q$, define the deterministic semiautomaton

$$\mathcal{D}_1 \times \cdots \times \mathcal{D}_h = (\Sigma, (2^Q)^h, \gamma_0, \widehat{E}_{\mathcal{D}}),$$

where $\gamma_0 = (\{t_1\}, \dots, \{t_h\})$, and, for every state $(S_1, \dots, S_h)$ of $\mathcal{D}_1 \times \dots \times \mathcal{D}_h$ and every $a \in \Sigma$, there is an edge $((S_1, \dots, S_h), a, (S'_1, \dots, S'_h)) \in \widehat{E}_{\mathcal{D}}$, if (S_i, a, S'_i) is an edge in the semiautomaton $\mathcal{D}_i$, for $i = 1, \dots, h$. Then the *product semiautomaton* for T is the accessible semiautomaton of $\mathcal{D}_1 \times \cdots \times \mathcal{D}_h$, and it is denoted by

$$\mathcal{D}(T) = (\Sigma, \Gamma, \gamma_0, E_{\mathcal{D}}).$$

Note that $\mathcal{D}(T)$ is complete.

Please cite this article in press as: J. Brzozowski, N. Santean, Predictable semiautomata, Theoretical Computer Science (2009), doi:10.1016/j.tcs.2009.05.010

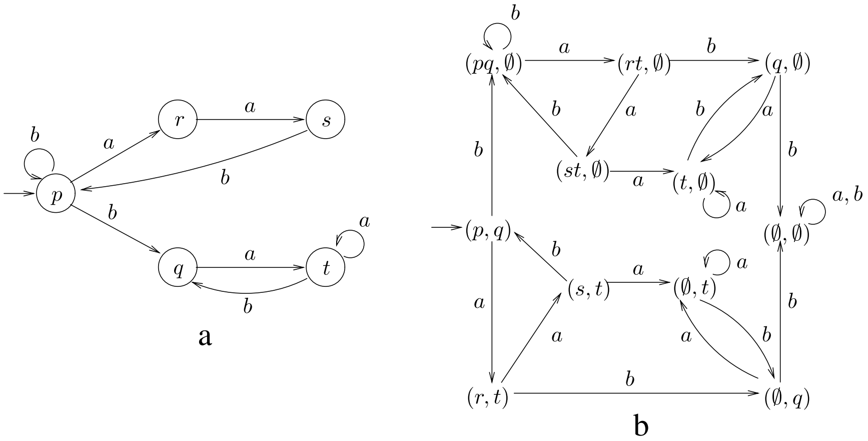

Fig. 8. An unpredictable semiautomaton and its product semiautomaton.

Next we identify several types of states in the product semiautomaton; these states will help us to find minimal selectors and maximal nonselectors.

Let $\gamma_{empty} = (\emptyset, \dots, \emptyset) \in (2^Q)^h$.

- A state is t_i-*singular* if only its ith component is nonempty; it is t_i-*plural* if at least two of its components, including the ith, are nonempty.
- A state γ_i is t_i-*primary* if there exists a path $(\gamma_0, a_1, \gamma_1) \cdots (\gamma_{j-1}, a_j, \gamma_j)$, where $\gamma_0, \dots, \gamma_{j-1}$ are t_i-plural and γ_j is t_i-singular.
- A state γ is t_i-*ultimate* if it is t_i-plural and the ith component of γa is empty, for each $a \in \Sigma$.
- A state is *singular*, *plural*, *primary*, or *ultimate*, if it is t_i-singular, t_i-plural, t_i-primary, or t_i-ultimate for some $t_i \in T$, respectively.
- Let Γ_{pl} (Γ_{pr}) be the set of all plural (primary) states of Γ.
- A state is *cyclic* if it appears in a cycle.

A word w is *singular* (*primary, plural*) if $\gamma_0 w$ is singular (primary, plural); $w = a_1 \cdots a_i$ is *nullary* if its path is $(\gamma_0, a_1, \gamma_1) \cdots (\gamma_{i-1}, a_i, \gamma_i)$, where $\gamma_0, \dots, \gamma_{i-1}$ are plural and $\gamma_i = \gamma_{empty}$.

Example 20. Fig. 8(a) shows a semiautomaton with one initial state and one fork $\langle p, b\rangle$, with fork set $T = \{p, q\}$. The product semiautomaton $\mathcal{D}(T)$ is given in Fig. 8(b), where, for simplicity, we represent sets of states as words; for example, we write pq for $\{p, q\}$.

Here $\gamma_{empty} = (\emptyset, \emptyset)$. There are five p-singular states: $(pq, \emptyset)$, $(rt, \emptyset)$, $(st, \emptyset)$, $(q, \emptyset)$, and $(t, \emptyset)$, and two q-singular states: $(\emptyset, t)$ and $(\emptyset, q)$. States (p, q), (r, t), and (s, t) are both p-plural and q-plural. State $(pq, \emptyset)$ is p-primary, while $(\emptyset, t)$ and $(\emptyset, q)$ are q-primary. There are no p-ultimate or q-ultimate states. All the plural states are cyclic.

Examples of singular words are *bbaa* and *aabaaa*. All words in $(aab)^*(1 + a + aa)$ are plural. There is an infinite number of p-primary words; the set of all such words is denoted by the regular expression $(aab)^*b$. There is an infinite number of q-primary words: $(aab)^*(ab + aaa)$. Note that a primary state may be also reached by words that are not primary. For example, $(\emptyset, t)$ can be reached by *aba*. There are no nullary words. ■

Our next proposition characterizes the selectors and nonselectors in a set T of states of a semiautomaton $\mathcal{S}$ in terms of the types of states reached by these words in the product automaton $\mathcal{D}(T)$.

Proposition 21. *Let $\mathcal{D}(T)$ be the product semiautomaton of a set T in $\mathcal{S}$.* (1) *Word w is a t_i-selector in T if and only if $\gamma_0 w$ is t_i-singular.* (2) *Word w is a minimal t_i-selector if and only if $\gamma_0 w$ is t_i-primary.* (3) *Word w is a t_i-nonselector if and only if $\gamma_0 w$ is t_i-plural.* (4) *Word w is a maximal t_i-nonselector if and only if $\gamma_0 w$ is t_i-ultimate.*

Proof. Properties (1) and (3) follow easily from the definitions.

For (2), if w is a minimal t_i-selector, then $\gamma_0 w$ is t_i-singular by (1). If w has a proper prefix u, then u must be a nonselector. Thus every state of the form $\gamma_0 u$ is plural, and hence w is primary. Conversely, if w is primary, then it defines a path $(\gamma_0, a_1, \gamma_1) \cdots (\gamma_{m-1}, a_m, \gamma_m)$, where $\gamma_0, \dots, \gamma_{m-1}$ are plural and γ_m is singular. Therefore no proper prefix of w is a selector, and w is minimal.

For (4), if $\gamma = \gamma_0 w$ is t_i-ultimate, then γ is t_i-plural. Since w is not a t_i-selector and $w \in R_{t_i}$, then w is a t_i-nonselector. Because every extension wa leads to a state with an empty ith component, $wa \notin R_{t_i}$, and no extension wau is in R_{t_i}, since R_{t_i} is prefix-closed. Hence w is maximal. Conversely, if w is a maximal t_i-nonselector, then $w \in R_{t_i}$ and $w \in R_{t_j}$ for some $j \neq i$. Hence $\gamma_0 w$ is a state with nonempty ith and jth components. If $\gamma_0 w$ is not t_i-ultimate, then there exists $a \in \Sigma$ such that $\gamma_0 wa$ has a nonempty ith component. But then $wa \in R_{t_i}$ and w is not maximal. Therefore $\gamma_0 w$ is t_i-ultimate. □

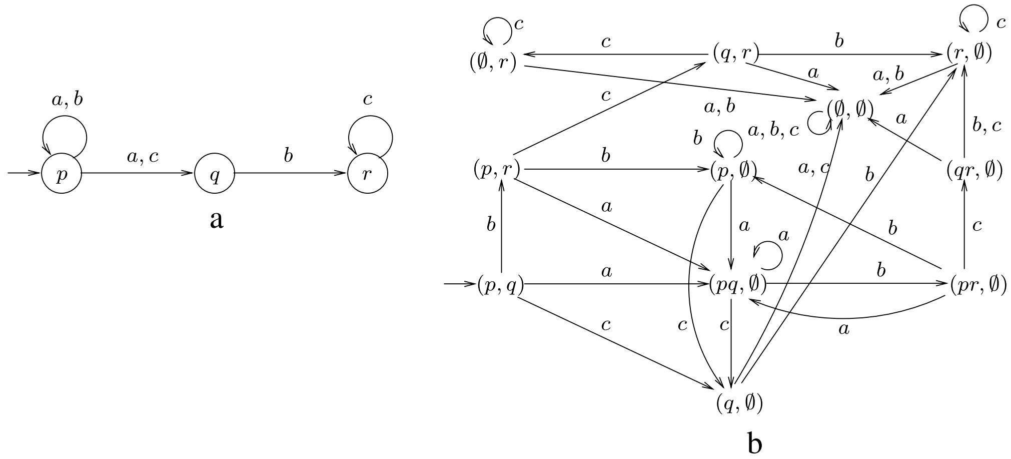

Fig. 9. A predictable semiautomaton and its product semiautomaton.

Theorem 22. *The length of a longest primary or nullary word in product semiautomata $\mathcal{D}(T)$ over all critical sets T is k if and only if $\mathcal{S}$ is predictable with index k.*

Proof. Let $k-1$ be the length of a longest plural word w in $\mathcal{D}(T)$. Then w is a nonselector in T, by Proposition 21(3). By Theorem 15, T is not $(k-1)$-predictable. Since there are no plural words of length k in $\mathcal{D}(T)$, every word u of length k is either nullary or singular. In the first case, $u \notin R_T$. In the second case, u is a selector by Proposition 21(1). Thus every word of length k in R_T is a selector, and T is k-predictable by Theorem 15.

Conversely, if T is k-predictable, then all nonselectors in T and (by Proposition 21(3)) all plural words in $\mathcal{D}(T)$ are of length $< k$, by Theorem 15. If T is not $(k-1)$-predictable, then R_T must have a nonselector of length $k-1$, and hence there is a plural word of that length in $\mathcal{D}(T)$. □

Corollary 23. *A set $T = \{t_1, \ldots, t_h\} \subseteq Q$ of $\mathcal{S}$ is predictable if and only if the product semiautomaton $\mathcal{D}(T)$ does not have cyclic plural states.*

Example 24. The semiautomaton of Fig. 8 is not predictable because its product semiautomaton has cyclic plural states. This is in spite of the fact that the set $\{p, q\}$ has an infinite number of minimal selectors denoted by the regular expression $(aab)^*(b + ab + aaa)$.

Semiautomaton $\mathcal{S}$ of Fig. 9 (a) has one nontrivial critical set $T = \{p, q\}$, of fork $\langle p, a\rangle$. The product semiautomaton $\mathcal{D}(T)$ is shown in Fig. 9(b). Since no plural state is cyclic, $\mathcal{S}$ is predictable.

The plural words, and hence nonselectors, are 1, b, and bc, and none is maximal. There is one nullary word bca. In every deterministic transition in Fig. 9 (a), 1 is the minimal selector. The primary words a, c, ba, bb and bcb are minimal p-selectors, and the only minimal q-selector is bcc. The length of a longest primary or nullary word is 3; hence the set $\{p, q\}$ and $\mathcal{S}$ are 3-predictable by Theorem 22. ■

6. Core semiautomata

We now show that, for a predictable semiautomaton $\mathcal{S}$, a part of the product semiautomaton suffices for finding selectors and nonselectors.

Definition 25. The *core* of a product semiautomaton $\mathcal{D}(T) = (\Sigma, \Gamma, \gamma_0, E_{\mathcal{D}})$ is an incomplete deterministic semiautomaton $\mathcal{C}(T) = (\Sigma, \Omega, \gamma_0, E_{\mathcal{C}})$, where

$$\Omega = \begin{cases} \Gamma_{pl} \cup \Gamma_{pr} \cup \{\gamma_{empty}\} & \text{if there is an edge from a plural state to } \gamma_{empty}, \\ \Gamma_{pl} \cup \Gamma_{pr} & \text{otherwise,} \end{cases}$$

and $E_{\mathcal{C}}$ consists of edges that join a plural state to a plural state, primary state, or γ_{empty}.

Since minimal selectors are primary words and maximal nonselectors lead to ultimate states, both can be found from the core semiautomaton. Thus, it is not necessary to construct the full product semiautomaton.

Example 26. The core semiautomaton $\mathcal{C}(T)$ of product semiautomaton $\mathcal{D}(T)$ of Fig. 9(b) is in Fig. 10. The minimal selectors and maximal nonselectors are now easier to find by inspection. ■

Example 27. In Fig. 11 the symbols in [ ] and ⌊ ⌋ show the minimal selectors and maximal nonselectors. There are two initial states and two forks. The core semiautomata are shown in Fig. 12. The semiautomaton is 2-predictable. ■

Please cite this article in press as: J. Brzozowski, N. Santean, Predictable semiautomata, Theoretical Computer Science (2009), doi:10.1016/j.tcs.2009.05.010

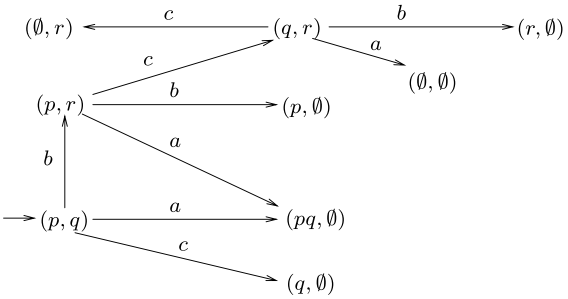

Fig. 10. The core of the product semiautomaton in Fig. 9(b).

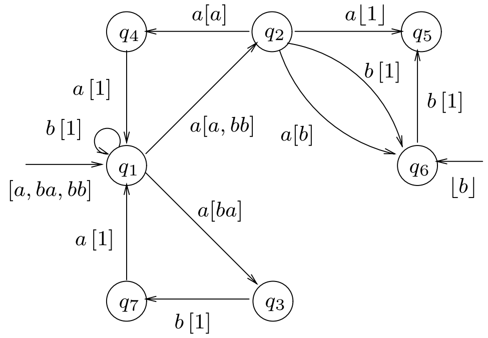

Fig. 11. Illustrating selectors and nonselectors.

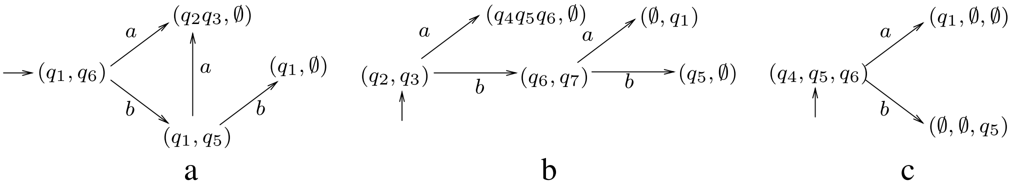

Fig. 12. Core semiautomata for Example 27.

The number of states of the core semiautomaton is $\leq (n+1)(2^n)^n$, since there are $n+1$ critical sets, each of which may have n states, and each state is a subset of the set of all n states. More research is needed to determine how complex the core semiautomata are in typical cases. When the semiautomaton has a small predictability bound, or only a few forks, or small critical sets, it may be practical to construct the predictor. In any case, the minimal selectors and maximal nonselectors need only be computed once, and can be used in many applications, as explained in the next section.

Minimal selectors permit us to determine precisely which state t must be chosen from a set T in a computation step. Sometimes it is also possible to reduce nondeterminism further by using maximal nonselectors. A maximal nonselector restricts the choice of states from T to a subset $S \subseteq T$ containing at least two states. In the worst case, when $S = T$, no reduction is possible. These ideas are applied in the next section.

7. Predictors

The concepts of the previous sections are now used to simulate a predictable semiautomaton almost deterministically. Let

$$\Sigma_s^{\leq k} = \{[w] \mid w \in \Sigma^{\leq k}\}$$

be the set of all possible minimal selectors, let

$$\Sigma_n^{\leq k-1} = \{\lfloor w \rfloor \mid w \in \Sigma^{\leq k-1}\},$$

be the set of all possible maximal nonselectors, and let

$$\Sigma_{s,n} = \Sigma_s^{\leq k} \cup \Sigma_n^{\leq k-1}.$$

Starting with a semiautomaton $\mathcal{S}$, we define a semiautomaton $\mathcal{P}$ that has $\Sigma \times \Sigma_{s,n}$ as input alphabet; the new input consists of the current input letter a and up to k letters of look-ahead information.

Definition 28. Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a k-predictable semiautomaton, $k \geq 0$. The *predictor* of $\mathcal{S}$ is a semiautomaton $\mathcal{P} = (\Sigma \times \Sigma_{s,n}, Q, P, E_{\mathcal{P}})$, where:

(1) The set of initial states is P. The sets of minimal p-selectors and maximal p-nonselectors in P are associated with each state $p \in P$.

(2) If $\langle q, a\rangle$ is a fork, and (q, a, r), an edge in $\mathcal{S}$, then $(q, (a, \lceil u \rceil), r) \in E_{\mathcal{P}}$, if u is a minimal r-selector, and $(q, (a, \lfloor u \rfloor), r) \in E_{\mathcal{P}}$, if u is a maximal r-nonselector.

By Proposition 17, each state in P and in $\langle\langle q, a\rangle\rangle$, for each $q \in Q, a \in \Sigma$, has a minimal selector or a maximal nonselector. Thus it is easily verified that there is a one-to-one correspondence between predictable semiautomata and predictors.

Example 29. Fig. 11 without the minimal selectors and maximal nonselectors is a semiautomaton $\mathcal{S}$. With the minimal selectors and maximal nonselectors, it becomes a predictor. The incoming edge of q_1 is labeled with minimal selectors $\lceil a \rceil$, $\lceil ba \rceil$, and $\lceil bb \rceil$, and that of q_6, with maximal nonselector $\lfloor b \rfloor$. The edge (q_1, a, q_2) is replaced by edges $(q_1, (a, \lceil a \rceil), q_2)$ and $(q_1, (a, \lceil bb \rceil), q_2)$, *etc.* Note that the two objects have the same set of states and initial states. The alphabet of the predictor has been expanded to include minimal selectors and maximal nonselectors, and the transitions have been modified accordingly. ■

We now describe how a predictor can be used to simulate a nondeterministic semiautomaton. The objective is to compute the set of states that can be reached by any $w \in |\mathcal{S}|$; if w is not in $|\mathcal{S}|$, then the set of states reached is empty. The simulation decides as soon as possible whether the input is accepted or rejected.

Lemma 30. *Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a semiautomaton, and let $T \subseteq Q$ be k-predictable. If $w \in R_t$, for some state $t \in T$, then one of the following conditions holds:* (1) *A prefix u of w is a minimal t-selector.* (2) *$|w| < k$, and w is a prefix of a minimal t-selector.* (3) *$|w| < k$, and w is a prefix of a maximal t-nonselector.*

Proof. If $|w| \geq k$, then w has a prefix which is a minimal selector, by Theorem 15(3). If $|w| < k$, and w is a selector, then it has a prefix which is a minimal selector, by the definition of the latter. Assume now that w is a t-nonselector and $|w| < k$. Consider any extension wx of w. This extension can be a t-selector, a t-nonselector, or not in R_t. If wx is a t-selector, then it has a prefix u which is a minimal t-selector. Now u cannot be a prefix of w, since no selector is a prefix of a nonselector, by Proposition 14(5). Hence u is an extension of w, and (2) holds. If neither (1) nor (2) holds, then all extensions of w are either t-nonselectors or are not in R_t. If, for all $a \in \Sigma$, the extension wa is not in R_t, then w is a maximal t-nonselector. Otherwise, there is an a such that wa is a t-nonselector. Continuing with this argument we obtain longer and longer t-nonselectors. By Theorem 15(4), every nonselector is of length less than k. Therefore we must eventually reach a maximal t-nonselector, and (3) holds. □

Definition 31. In a predictor $\mathcal{P}$, for a word $w \in \Sigma^*$ and $T \subseteq Q$, we define the *handle* of w in T. If a minimal selector x in T is a prefix of w, then x is the handle. If w is a prefix of a minimal selector or a maximal nonselector in T, then w itself is the handle. Otherwise, w does not have a handle. If w has a handle in T, the handle *applies* to a state $t \in T$ if $w \in R_t$.

Definition 32. Given a predictor $\mathcal{P}(\mathcal{S}) = \mathcal{P} = (\Sigma \times \Sigma_{s,n}, Q, P, E_{\mathcal{P}})$ of a k-predictable semiautomaton $\mathcal{S}$ and an input word w, a prefix y of w *yields* a state $s \in Q$, written $y \Rightarrow s$, as follows:

- *Basis Step* (Step 0): $1 \Rightarrow s$ if $s \in P$ and the handle of w in P applies to s.
- *Induction Step* (Step $m + 1$, $m \geq 0$): Assume now that $w = yaz$, for some $a \in \Sigma$, $y, z \in \Sigma^*$. Then $ya \Rightarrow s$ if $y \Rightarrow r$, for some $r \in Q$, $s \in \langle\langle r, a\rangle\rangle$ and the handle of z in $\langle\langle r, a\rangle\rangle$ applies to s.

Example 33. Consider the semiautomaton of Fig. 11, the fork $\langle q_1, a\rangle$ and the fork set $\langle\langle q_1, a\rangle\rangle = \{q_2, q_3\}$. Words a and bb are minimal q_2-selectors, and ba is a minimal q_3-selector. Words $1, a, b, ba$ and bb are all prefixes of minimal selectors. Hence they are their own handles. Word abb has handle a, and word bbb has handle bb.

Suppose $w = aaab$. The handle of w in $P = \{q_1, q_6\}$ is a. Since a applies only to q_1, we have $1 \Rightarrow q_1$.

Now w has the form $w = aaab = (1)(a)(aab) = yaz$. Since $1 \Rightarrow q_1$, the current input is a, the fork set to consider is $\langle\langle q_1, a\rangle\rangle = \{q_2, q_3\}$. Since the handle of $z = aab$ in $\{q_2, q_3\}$ is a, and the handle applies to q_2, we have $a \Rightarrow q_2$.

Now w has the form $w = aaab = (a)(a)(ab) = yaz$. Since $a \Rightarrow q_2$, the current input is a, the fork set is $\langle\langle q_2, a\rangle\rangle = \{q_4, q_5, q_6\}$. Since the handle of $z = ab$ in $\{q_4, q_5, q_6\}$ is a, and the handle applies to q_4, we have $aa \Rightarrow q_4$.

Now w has the form $w = aaab = (aa)(a)(b) = yaz$. Since $aa \Rightarrow q_4$, the current input is a, the fork set is $\langle\langle q_4, a\rangle\rangle = \{q_1\}$. Since the handle of $z = b$ in $\{q_1\}$ is 1, and the handle applies to q_1, we have $aaa \Rightarrow q_1$.

Now w has the form $w = aaab = (aaa)(b)(1) = yaz$. Since $aaa \Rightarrow q_1$, the current input is b, the fork set is $\langle\langle q_1, b\rangle\rangle = \{q_1\}$. Since the handle of $z = 1$ in $\{q_1\}$ is 1, and the handle applies to q_1, we have $aaab \Rightarrow q_1$. ■

The next result proves the correctness of the predictor's simulation, and defines termination conditions.

Theorem 34. *Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a k-predictable semiautomaton, $\mathcal{P} = (\Sigma \times \Sigma_{s,n}, Q, P, E_{\mathcal{P}})$, its predictor, and $w = yv \in \Sigma^*$, an input word of $\mathcal{S}$. Then the predictor operation is correct in the following sense:*

1. *If $w \in |\mathcal{S}|$, then $y \Rightarrow q$ in $\mathcal{P}$ if and only if $q \in Py$ and $v \in R_q$ in $\mathcal{S}$.*

2. *The simulation stops with the remaining input v if and only if one of the following holds: (a) It is step 0 and w has no handle in P; this implies $w \notin |\mathscr{S}|$. (b) It is a step > 0 and $v = 1$; this implies that $w \in |\mathscr{S}|$. (c) It is a step > 0 and $v = az$, for some $a \in \Sigma, z \in \Sigma^*$ and there is no fork $\langle q, a\rangle$ in $\mathscr{S}$; then $w \notin |\mathscr{S}|$. (d) It is a step > 0 and $v = az$, for some $a \in \Sigma, z \in \Sigma^*$ there is a fork $\langle q, a\rangle$ in $\mathscr{S}$, but z has no handle in $\langle\langle q, a\rangle\rangle$; this implies $w \notin |\mathscr{S}|$.*

Proof. First, we show that, if $y \Rightarrow q$, then $q \in Py$ and $v \in R_q$, by induction on the length of the prefix y. If $1 \Rightarrow s$, then $s \in P$ by Definition 32. Thus $s \in P1 = P$. Since $w \in R_P$ by assumption, and the handle of w in P applies to s, we have $w \in R_s$, and the basis holds. Now assume that, for an arbitrary prefix y of $w = yaz$, if $y \Rightarrow r$, then $r \in Py$ and $v \in R_r$. Suppose that $ya \Rightarrow s$. Then $y \Rightarrow r$, for some $r \in Q$, $s \in \langle\langle r, a\rangle\rangle$, and the handle x of z in $\langle\langle r, a\rangle\rangle$ applies to s. By the induction hypothesis, $r \in Py$ and $az \in R_r$. Since $s \in \langle\langle r, a\rangle\rangle$, there is an edge $(r, a, s) \in E$; hence $s \in Pya$. Since $z \in R_{\langle\langle r,a\rangle\rangle}$ because $az \in R_r$, and the handle of z in $\langle\langle r, a\rangle\rangle$ applies to s, we have $z \in R_s$. Thus the induction goes through.

Second, assume that $q \in Py$ and $v \in R_q$; we show that $y \Rightarrow q$, by induction on the length of y. Consider first the factorization $w = yv = 1w$. If $p \in P$ and $w \in R_p$, then the handle of w in P applies to p. By Definition 32, the empty prefix 1 of w derives p. Now assume that, for an arbitrary prefix y of $w = yv$, if $r \in Py$ and $v \in R_r$, then $y \Rightarrow r$. Suppose now that $w = yaz$. Since $w \in R_P$, there exists $r \in Py$ such that $az \in R_r$, and hence there is a fork $\langle r, a\rangle$. Let s be any state in $T = \langle\langle r, a\rangle\rangle$ such that $z \in R_s$. By the induction hypothesis, $y \Rightarrow r$. Since $s \in T$ and $z \in R_s$, the handle of z in T applies to s. Therefore $ya \Rightarrow s$, and the claim holds.

For (a), if w has no handle in P, then $1 \Rightarrow p$ is false for all $p \in P$, by Definition 32, and the simulation stops. Now if $w \in |\mathscr{S}|$, then $w \in R_P$, and hence $w \in R_p$, for some $p \in P$. Since also $p \in P1$, the first part of the theorem applies, and $1 \Rightarrow p$, which is a contradiction; thus $w \notin |\mathscr{S}|$.

For (b), if the simulation has consumed y, $y \Rightarrow q$, and $v = 1$, then $w = y$, and the entire input has been processed. Since v does not have the form az, the simulation stops. Since $w = y \Rightarrow q$, we have $w \in |\mathscr{S}|$.

For (c), assume that $w = yv = yaz$, the simulation has consumed y, $y \Rightarrow q$, but there is no fork $\langle q, a\rangle$ in $\mathscr{S}$. The induction step cannot be carried out, and $az \notin R_q$. Suppose now that $w \in |\mathscr{S}|$. Since $y \Rightarrow q$, we have $q \in Py$ and $az \in R_q$, by the first part of the theorem. This is a contradiction; thus $w \notin |\mathscr{S}|$.

For (d), assume that $w = yv = yaz$, the simulation has consumed y, $y \Rightarrow q$, and z has no handle in $\langle\langle q, a\rangle\rangle$. The derivation stops because the induction step cannot be carried out. If $w \in |\mathscr{S}|$, since $y \Rightarrow q$, we know by the first part of the theorem that $q \in Py$ and $v \in R_q$. Also, there exists $r \in \langle\langle q, a\rangle\rangle$ such that (q, a, r) is an edge in $\mathscr{S}$ and $z \in R_r$. But now $r \in Pya$ and $z \in R_r$ implies that $ya \Rightarrow r$, again by the first part. Thus z must have a handle in $\langle\langle q, a\rangle\rangle$, which is a contradiction, and $w \notin |\mathscr{S}|$.

One verifies that, if none of the conditions (a)–(d) holds, then the derivation continues. This concludes the proof of the second claim. □

Corollary 35. *In a predictor $\mathscr{P}$, $w \in |\mathscr{S}|$ if and only if $w \Rightarrow q$, for some $q \in Q$.*

Example 36. We continue with the semiautomaton of Fig. 11, and expand Example 33. Suppose $w = aaababaab$. We write $a(z)$ for az to make it easier to identify the handle of z. Let q be the current state, and v, the remaining input. In Step 0, $v = w = aaababaab$, and the handle of w in $\{q_1, q_6\}$ is a. We have $1 \Rightarrow q_1$, since a applies only to q_1. With the look-ahead information, next seven steps are deterministic:

$$\overset{[a]}{\Rightarrow} q_1 \overset{a[a]}{\Rightarrow} q_2 \overset{a[a]}{\Rightarrow} q_4 \overset{a[1]}{\Rightarrow} q_1 \overset{b[1]}{\Rightarrow} q_1 \overset{a[ba]}{\Rightarrow} q_3 \overset{b[1]}{\Rightarrow} q_7 \overset{a[1]}{\Rightarrow} q_1.$$

Here, the transition $q_1 \overset{a[a]}{\Rightarrow} q_2$ means that the current state is q_1, and the current input is a. Since $q = q_1$, $v = az = a(aababaab)$, the handle of z in $\{q_2, q_3\}$ is a. Hence $a \Rightarrow q_2$, since a applies only to q_2, and a is consumed from the input.

Next $q = q_2$, $v = az = a(ababaab)$, and the handle of z in $\{q_4, q_5, q_6\}$ is a. Then $aa \Rightarrow q_4$, since a applies only to q_4, and a is consumed, *etc.*

This deterministic computation continues until $q = q_1$ and $v = a(b)$. Now the handle of b in $\{q_2, q_3\}$ is b itself. Since the handle is a prefix of minimal selectors ba and bb, it applies to both q_2 and q_3.

Finally, for $v = b(1)$, q_2 moves to q_6, and q_3 moves to q_7. Thus w yields $\{q_6, q_7\}$.

As another example, consider $w = abba$. Here we have:

$$\overset{[a]}{\Rightarrow} q_1 \overset{a[a]}{\Rightarrow} q_2 \overset{b[1]}{\Rightarrow} q_6 \overset{b[1]}{\Rightarrow} q_5.$$

Then the computation stops, yielding the empty set of reachable states. ■

The simulation operates "almost deterministically" in finding next states. By Corollary 35, we can achieve total determinism concerning the membership of a word in the language of the semiautomaton: Run the simulation until more than one choice appears, and then choose an arbitrary branch in every step (since all choices lead to the same conclusion). If the size of the alphabet Σ, the number of nontrivial forks and the integer k are not prohibitively large, one can pre-compute the set of states reachable from any fork by each word of length $\leq k$, and use a look-up table for the last part of the computation, thus making the entire computation completely deterministic.

Another type of simulation, which predicts the next state as long as a prefix of the input word is in the language of the semiautomaton, is described in [3].

8. Conclusions

We have introduced a class of semiautomata in which it is possible to remove most of the nondeterminism by using a finite amount of look-ahead information from the input tape. In the worst case, the length k of the look-ahead buffer is quadratic in the number n of states of $\mathcal{S}$. As long as the input word has length $> k$, the computation is deterministic, and nondeterminism is limited to the last k letters of the input word. The application to nondeterministic automata (semiautomata with accepting states) is straightforward. To determine whether a word w is accepted by an automaton, find the set of states derived by w and check whether any of these states are accepting.

References

[1] D. Berardi, D. Calvanese, G.D. Giacomo, M. Lenzerini, M. Mecella, Automatic composition of e-services that export their behavior, in: E. Orłowska, M. Papazoglou, S. Weerawarana, J. Yang (Eds.), Proc. ICSOC 2003, in: LNCS, vol. 2910, 2003, pp. 43–58.

[2] J. Berstel, D. Perrin, Theory of Codes, Academic Press, New York, 1985.

[3] J. Brzozowski, N. Santean, Predictable semiautomata. Technical Report CS-2007-03, School of Computer Science, University of Waterloo, Waterloo, ON, Canada, pp. 30, 2007. http://www.cs.uwaterloo.ca/research/tr/2007/.

[4] Z. Dang, O.H. Ibarra, J. Su, Composability of infinite-state activity automata, in: R. Fleischer, G. Trippen (Eds.), Proc. ISAAC 2004, in: LNCS, vol. 3341, 2004, pp. 377–388.

[5] C.G. Gerede, R. Hull, O.H. Ibarra, J. Su, Automated composition of e-services: Lookaheads, in: Proc. 2nd International Conference on Service Oriented Computing, 2004 pp. 252–262.

[6] A. Ginzburg, Algebraic Theory of Automata, Academic Press, New York, 1968.

[7] Y.-S. Han, Y. Wang, D. Wood, Infix-free regular expressions and languages, International Journal of Foundations of Computer Science 17 (2) (2006) 379–393.

[8] Y.-S. Han, D. Wood, Generalizations of 1-deterministic regular languages, Information and Computation 206 (9–10) (2008) 1117–1125.

[9] B. Ravikumar, N. Santean, Deterministic simulation of NFA with k-symbol lookahead, International Journal of Foundations of Computer Science 18 (5) (2007) 949–973.

Please cite this article in press as: J. Brzozowski, N. Santean, Predictable semiautomata, Theoretical Computer Science (2009), doi:10.1016/j.tcs.2009.05.010