How to Avoid Nondeterminism with a Little Foresight

J. A. Brzozowski and N. Santean

David R. Cheriton School of Computer Science,
University of Waterloo, Waterloo, ON,
Canada N2L 3G1
brzozo@uwaterloo.ca, nsantean@uwaterloo.ca

Abstract. A nondeterministic semiautomaton \( S \) is predictable if there exists an integer \( k \geq 0 \) such that, if \( S \) knows the present input \( a \) and the next \( k \) inputs, then the transition under \( a \) is deterministic. Nondeterminism may occur only when the length of the unread input is less than \( k + 1 \). We develop a theory of predictable semiautomata. We present a test for predictability and introduce the predictor semiautomaton, based on a look-ahead semiautomaton, that is essentially deterministic. The predictor is used in two ways to simulate a nondeterministic semiautomaton. The first simulator finds the set of states reachable by the longest prefix of the input which belongs to the language of the semiautomaton. The second stops as soon as it infers that the input word is not in the language. Moreover, the membership of a word in the language of a semiautomaton can be decided deterministically. We also give sharp bounds on the size of the look-ahead buffer, as a function of the number of states.

1 Introduction

Nondeterministic automata are ubiquitous in computer science. They serve as models for various nondeterministic processes, constitute valuable design tools (often more convenient than their deterministic counterparts), and are inevitable in many applications. They also have some drawbacks, such as increased simulation time and space, and inefficient minimization algorithms. Several attempts have been made recently to overcome the disadvantages of nondeterminism. Nondeterministic finite automata (NFA) have been used as formal models for service-oriented computing [1], and as tools for automated web service composition [3]. In both of these applications, it was imperative to deal with the problems introduced by nondeterminism. For this purpose, “delegators” of NFAs were informally introduced in [3]. A delegator is an equivalent 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. Look-ahead delegation was studied systematically and in a more abstract framework in [5].

We address a problem similar to delegation, but we formulate it in the more general model of semiautomata. We introduce semiautomata, called “predictable”, 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. Our goal is to compute the set of states reachable from the initial set of states of a semiautomaton by a given input word with as little nondeterminism as possible. Our theory is substantially different from the work in [3, 5]. For a detailed comparison of the two approaches, see [2]. For proofs of the results in this paper see the appendix.
2 Predictable Semiautomata

For a set $X$, we denote its cardinality by $X\#$. If $\Sigma$ is an alphabet, then $\Sigma^+$ and $\Sigma^*$ denote the free semigroup and the free monoid, respectively, generated by $\Sigma$. The empty word is 1. For $k \geq 1$, let $\Sigma^\leq k = 1 \cup \Sigma \cup \ldots \cup \Sigma^k$. 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 $w$ and $v$ is a suffix of $w$. A language $L$ is prefix-free if no word of $L$ is a prefix of another word of $L$. It 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 $[4] \mathcal{S} = (\Sigma, Q, P, E)$ consists of an alphabet $\Sigma$, a set $Q$ of states, a set $P \subseteq \Sigma$ of initial states, and an alphabet $E$ 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 $\pi$ is a finite sequence $\pi = (q_0, a_1, q_1)(q_1, a_2, q_2)\ldots(q_{k-1}, a_k, q_k)$ of consecutive edges, $k > 0$ being its length, $q_0$, its beginning, $q_k$, its end, and word $w = a_1\ldots a_k$, its label. We also write $q_0 \xrightarrow{w} q_k$ for $\pi$. 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 $T_w = \{q \in Q \mid t \xrightarrow{w} q,$ for some $t \in T\}$. If $T = \{t\}$, we write $T_w = \{q\}$, we write $T_w = q$. A state $q$ of a semiautomaton $\mathcal{S}$ is accessible if there exists $p \in P, w \in \Sigma^*$ such that there is a path $p \xrightarrow{w} q$. A semiautomaton is accessible if all of its states are accessible.

The language $|\mathcal{S}|$ of a semiautomaton $\mathcal{S}$ is the set of all labels of paths from initial states of $\mathcal{S}$, that is, $|\mathcal{S}| = \{w \in \Sigma^* \mid Pw \neq \emptyset\}$, and $|\mathcal{S}|$ is prefix-closed. 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 $r \in Q$. In a complete semiautomaton $\mathcal{S}$, $qw \neq \emptyset$, for all $q \in Q, w \in \Sigma^*$, and $|\mathcal{S}| = \Sigma^*$. 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 $\mathcal{S} = (\Sigma, Q, p, E)$. If $q$ is a state of $\mathcal{S} = (\Sigma, Q, P, E)$, the language of $q$ is $R_q = \{w \in \Sigma^* \mid qw \neq \emptyset\}$. If $\mathcal{S}$ is complete, $R_q = \Sigma^*$, for all $q \in Q$. The language of a set $T \subseteq Q$ is $R_T = \bigcup_{t \in T} R_t$; thus $R_P = |\mathcal{S}|$.

We restrict our attention to finite semiautomata. 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 $R_s \cap R_t \cap \Sigma^k = \emptyset$. A semiautomaton $\mathcal{S}$ 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 $k$.

A set is 0-predictable if and only if it consists of a single state. Thus, a semiautomaton is 0-predictable if and only if it is deterministic, i.e., its critical sets are singletons. A predictable semiautomaton is either deterministic or incomplete, because in a complete semiautomaton, $R_s = R_t = \Sigma^*$ and $R_s \cap R_t = \Sigma^*$, for all $s, t \in Q$. If a set is $k$-predictable, then it is $k'$-predictable for all $k' > k$, since $R_s$ and $R_t$ are prefix-closed.

**Example 1.** Fork $\langle p, a \rangle = \{(p, a, q)\}$ in Fig. 1 (a) is a deterministic transition, and the fork set of $\langle q, a \rangle = \{(q, a, q), (q, a, r)\}$ is $\langle \langle q, a \rangle \rangle = \{q, r\}$. This set is 1-predictable, since a word of length 1 (here, only $a$) belongs only to $R_q$, and not to $R_t$. The fork set $\{q, r\}$ in Fig. 1 (b) is 1-predictable, because there are no
words of length 1 in \( R_q \) or \( R_r \). Thus the semiautomata of Fig. 1 (a) and (b) are 1-predictable. For 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 \( k \).

![Diagram](image)

**Fig. 1.** Illustrating predictability.

**Definition 2.** If \( \mathcal{S} = (\Sigma, Q, P, E) \) is semiautomaton, and \( T = \{ t_1, \ldots, t_h \} \subseteq Q \), then a word \( w \in \Sigma^* \) is a \( t_i \)-selector in \( T \) if \( w \in \sigma(t_i, T) = \left( R_t \setminus \bigcup_{j \in \{1, \ldots, h\}, j \neq i} R_t \right) \). Word \( w \) is a selector in \( T \) if it is a \( t_i \)-selector in \( T \) for some \( t_i \), that is, if \( w \in \sigma(T) = \bigcup_{i=1}^h \sigma(t_i, T) \). A selector \( w \) is minimal if no prefix of \( w \) is a selector.

We also define the complementary set of \( t_i \)-nonselectors in \( T \): \( \overline{\sigma}(t_i, T) = R_t \setminus \sigma(t_i, T) \). The set of all nonselectors in \( T \) is \( \overline{\sigma}(T) = \bigcup_{i=1}^h \overline{\sigma}(t_i, T) = R_T \setminus \sigma(T) = \bigcup_{1 \leq i \neq j \leq h} (R_t \cap R_j) \). A \( t_i \)-nonselector \( u \) is maximal if no extension of \( u \) is in \( R_t \).

**Theorem 1.** Let \( \mathcal{S} = (\Sigma, Q, P, E) \) be a semiautomaton and \( T \subseteq Q \). The following are equivalent: (1) \( T \) is predictable. (2) There exists \( k \geq 0 \) such that \( \sigma(t, T) \supseteq R_t \cap \Sigma^k \), for all \( t \in T \). (3) There exists \( k \geq 0 \) such that \( \sigma(T) \supseteq R_T \cap \Sigma^k \). (4) \( \overline{\sigma}(T) \) is finite.

**Theorem 2.** Let \( \mathcal{S} = (\Sigma, Q, P, E) \) be a semiautomaton, \( n = Q^\# \), and \( k \geq 0 \), the smallest integer for which \( \mathcal{S} \) is \( k \)-predictable. If \( Q^\# = 1 \), then \( k \leq n - 1 \), and the bound is reachable. If \( Q^\# > 1 \), then \( k \leq (n^2 - n)/2 \), and the bound is reachable for a suitable \( \Sigma \).

### 3 Product semiautomata

For \( T = \{ t_1, \ldots, t_h \} \subseteq Q \) in \( \mathcal{S} = (\Sigma, Q, P, E) \), we need to find the intersections of the languages \( R_t \) to determine predictability. We could make the semiautomata \( \mathcal{S}_i = (\Sigma, Q, \{ t_i \}, E) \), \( t_i \in T \), deterministic and find their direct product. Instead, we obtain a deterministic direct product by using the subset construction in each step of the direct product construction. For a set \( Q \), let \( 2^Q \) be the set of all subsets of \( Q \). The direct product of \( h \) copies of \( 2^Q \) is denoted \( (2^Q)^h \).

**Definition 3.** Suppose \( \mathcal{S} = (\Sigma, Q, P, E) \) is a semiautomaton and \( T = \{ t_1, \ldots, t_h \} \subseteq Q \). Define the deterministic semiautomaton \( \mathcal{D}(T) = (\Sigma, (2^Q)^h, \gamma_0, E_{\mathcal{D}}) \), where \( \gamma_0 = (\{ t_1 \}, \ldots, \{ t_h \}) \), and, for every \( h \)-tuple \( (S_1, \ldots, S_h) \) of sets of states of \( \mathcal{S} \) and every \( a \in \Sigma \), there is an edge \( (\gamma_0, a, (S_1, \ldots, S_h)) \in E_{\mathcal{D}} \), where \( S_i \) is the set of successor states of the set \( S_i \) under input \( a \) in the semiautomaton \( \mathcal{S} \). The product semiautomaton for \( T \) is the accessible subsemiautomaton of \( \mathcal{D}(T) \) denoted by \( \mathcal{D}(T) = (\Sigma, \Gamma, \gamma_0, E_{\mathcal{D}}) \).
Note that $\mathcal{D}(T)$ is complete. We distinguish several types of states in $\Gamma$: The state $\gamma_0 = (\emptyset, \ldots, \emptyset) \in (2^Q)^h$ is null. A state with only the $i$th component nonempty is $t_i$-singular. It is singular if it is $t_i$-singular for some $i$. Any state in which at least two components are nonempty is plural. A state $\gamma$ is $t_i$-ultimate if it is plural, its $i$th component is nonempty, and the $i$th component of its successor $\gamma a$ is empty, for each $a \in \Sigma$. State $\gamma$ is ultimate if it is $t_i$-ultimate for some $i$. Let $\Gamma_{pl}$ (respectively, $\Gamma_{pr}$) be the set of all plural (respectively, primary) states of $\Gamma$.

A word $w = a_1 \ldots a_i$ defining a path $(\gamma_0, a_1, \gamma_1) \ldots (\gamma_{i-1}, a_i, \gamma_i)$, where $\gamma_0, \ldots, \gamma_{i-1}$ are plural and $\gamma_i = \gamma_0$, is nullary. A word $w$ defining a path $(\gamma_0, a_1, \gamma_1) \ldots (\gamma_{i-1}, a_i, \gamma_i)$, where $\gamma_0, \ldots, \gamma_{i-1}$ are plural and $\gamma_i$ is singular, is primary. If such a word exists, state $\gamma_i$ is also called primary. A state is cyclic if it appears in a cycle.

**Proposition 1.** Let $\mathcal{D}(T)$ be the product semiautomaton of a set $T$ in $\mathcal{I}$. We have: (1) If $\gamma = (S_1, \ldots, S_h)$ and $\gamma' = (S'_1, \ldots, S'_h)$ are two states in $\Gamma$ such that $\gamma' = \gamma w$, for some $w \in \Sigma^*$, and $S_i = \emptyset$, for some $i \in \{1, \ldots, h\}$, then also $S'_i = \emptyset$. (2) A word $w$ is in the language $R_T$ if and only if $\gamma_0w \neq \emptyset$. (3) A word $w$ is a $t_i$-selector in $T$ if and only if $\gamma_0w$ is $t_i$-singular. (4) A word is a minimal selector if and only if it is primary. (5) A word $w$ is a maximal $t_i$-nonselector if and only if $\gamma_0w$ is $t_i$-ultimate.

Proposition 1 allows us to prove the following result:

**Proposition 2.** (1) The set of all minimal selectors in $T$ is prefix-free. (2) No selector is a prefix of a nonselector. (3) For any $t_i \in T$, no maximal $t_i$-nonselector is a prefix of a $t_i$-selector. (4) For any $t_i \in T$, the set of all maximal $t_i$-nonselectors is prefix-free.

**Theorem 3.** A set $T = \{t_1, \ldots, t_h\} \subseteq Q$ of a semiautomaton $\mathcal{I}$ is predictable if and only if the product semiautomaton $\mathcal{D}(T)$ does not have cyclic plural states. Moreover, if $\mathcal{D}(T)$ does not have cyclic plural states, and the length of a longest primary or nullary word in product semiautomata $\mathcal{D}(T)$ over all critical sets $T$ is $k$, then $\mathcal{I}$ is $k$-predictable, but not $(k-1)$-predictable.

**Corollary 1.** (1) If $T$ is a predictable set of a semiautomaton $\mathcal{I}$, then the set of minimal selectors in $T$ is finite. (2) If $T$ is $k$-predictable, then every word $w$ in $R_T \cap \Sigma^k \Sigma^*$ has a prefix which is a minimal selector.

**Proposition 3.** Let $\mathcal{I} = (\Sigma, Q, P, E)$, and $T = \{t_1, \ldots, t_h\} \subseteq Q$. If $T$ is $k$-predictable, then the following hold: (1) If $t_i$ has no selectors, then $R_{t_i} \subseteq \Sigma^{<k-1}$. (2) Every $t_i \in T$ has either a minimal $t_i$-selector or a maximal $t_i$-nonselector. (3) Every nonselector is of length less than $k$.

As we shall see, for a predictable semiautomaton $\mathcal{I}$, a part of the product semiautomaton suffices for finding selectors and nonselectors.

**Definition 4.** The core semiautomaton 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

$$\Omega = \begin{cases} \Gamma_{pl} \cup \Gamma_{pr} \cup \{\gamma_0\} & \text{if there is an edge from a plural state to } \gamma_0, \\
\Gamma_{pl} \cup \Gamma_{pr} & \text{otherwise,} \end{cases}$$

and $E_{\mathcal{C}}$ consists of edges of $\mathcal{D}(T)$ that join a plural state to a plural state, a primary state, or $\gamma_0$. 

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 2. The semiautomaton $S$ of Fig. 2 (a) has only one critical set that is not a singleton, namely, $T = \{p, q\}$, corresponding to the fork $\langle p, a \rangle$. The product semiautomaton $D(T)$ is shown in Fig. 2 (b), where we write $p$ for $\{p\}$, $pq$ for $\{p, q\}$, etc. Since no plural state is cyclic, $S$ is predictable. The core semiautomaton $C(T)$ is shown in Fig. 3. The length of a longest primary or nullary word is 3, and the set $\{p, q\}$ and $S$ are 3-predictable by Theorem 3. The minimal $p$-selectors are $a$, $c$, $ba$, $bb$ and $bcb$, and the only minimal $q$-selector is $bcc$. The nonselectors are 1, $b$, and $bc$ and none is maximal. There is one nullary word $bca$. In each deterministic transition in Fig. 2 (a), 1 is the minimal selector.
Example 3. In the semiautomaton of Fig. 4 there are two initial states $q_1$ and $q_6$ and two forks. Minimal selectors are shown in square brackets on the arrows leading to the selected states, for example $[ba]$. Maximal nonselectors are shown in “floor” brackets, for example $[b]$. The core semiautomata of the critical sets are shown in Fig. 5. The critical set $\{q_1, q_6\}$ has minimal $q_1$-selectors $a$, $ba$, and $bb$, and no $q_6$-selectors. The critical set $\{q_2, q_3\}$ has minimal $q_2$-selectors $a$ and $bb$, and $q_3$-selector $ba$. The critical set $\{q_4, q_5, q_6\}$ has minimal $q_4$-selector $a$, $q_6$-selector $b$, and there are no $q_5$-selectors. The empty word $1$ is a selector in each deterministic transition. There is a maximal $q_6$-nonselector $b$ in the set of initial states, and a maximal $q_5$-nonselector $1$ in the fork $\{q_2, a\}$. The semiautomaton is 2-predictable.

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 with 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.

4 Predictors

The concepts of the previous sections are now used to simulate a predictable semiautomaton almost deterministically. Starting with a semiautomaton $\mathcal{S}$, we define a semiautomaton $\mathcal{P}$ that has $\Sigma \times \Sigma^k$ as input alphabet; the new input consists of the current input letter $a$ and up to $k$ letters of look-ahead information.
Definition 5. Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a $k$-predictable semiautomaton, $k \geq 0$. The predictor of $\mathcal{S}$ is a semiautomaton $\mathcal{P} = (\Sigma \times \Sigma^k, Q, P, E_\mathcal{P})$, 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 $(q, a)$ is a fork, and $(q, a, r)$, an edge in $\mathcal{S}$, then $(q, (a, [u]), r) \in E_\mathcal{P}$, if $u$ is a minimal $r$-selector, and $(q, (a, [u]), r) \in E_\mathcal{P}$, if $u$ is a maximal $r$-nonselector.

By Proposition 3 (2), each state in $P$ and in $\langle q, a \rangle$, 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.

The objective of a predictor is to find the set of all states reachable from the set of initial states, and to do this with as little nondeterminism as possible. We now describe two ways of using a predictor. The purpose of the first simulation is to compute the set of states that can be reached by any prefix $w'$ of $w$, if a prefix $w'$ is not in $|\mathcal{S}|$, then the set of states reached is empty. If a predictor operates in this fashion, it continues looking for the next state, or set of states, until it reaches the longest prefix of $w$ that is in $|\mathcal{S}|$. This is done even though in some cases the predictor is able to decide that the input word is not in the language of the semiautomaton; we call this maximal simulation.

Definition 6. In a predictor $\mathcal{P}$, for a word $w \in \Sigma^*$ and $T \subseteq Q$, the longest prefix $x$ of $w$ which is also a prefix of a minimal selector or a maximal nonselector of a state in $T$ is the key of $w$ in $T$. The key applies to a state $t \in T$ if it is a prefix of a minimal $t$-selector or a maximal $t$-nonselector.

The key always exists, since 1 is a prefix of every word. For every $T \subseteq Q$ and $w \in \Sigma^*$, there is at least one state $t \in T$ to which the key applies. If $T$ is $k$-predictable and $w \in \Sigma^*$, then the key of $w$ in $T$ must belong to $R_T$. Thus, if $w'$ is the longest prefix of $w$ that is in $R_T$, then the keys of $w$ and $w'$ in $T$ coincide.

Lemma 1. 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.

Lemma 2. Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a semiautomaton, let $T \subseteq Q$ be $k$-predictable and let $w \in \Sigma^*$ be an input word. Then the following hold: (1) If $w'$ is the longest prefix of $w$ that is in $R_T$, then the key of $w$ in $T$ is either a minimal selector in $T$ or it is $w'$ itself. (2) If $w'$ is an arbitrary prefix of $w$ that is in $R_T$, and $t \in T$, then $w' \in R_t$ if and only if the key of $w'$ in $T$ applies to $t$.

Definition 7. Maximal simulation: Given a predictor $\mathcal{P}(\mathcal{S}) = \mathcal{P} = (\Sigma \times \Sigma^k, Q, P, E_\mathcal{P})$ of a $k$-predictable $\mathcal{S}$ and an input word $w$, a prefix $y$ of $w$ derives a state $s \in Q$, written $y \Rightarrow s$, as follows:

1. Basis Step (Step 0):
   $1 \Rightarrow s$ if $s \in P$ and the key of $w$ in $P$ applies to $s$.
2. Induction Step (Step $m + 1$, $m \geq 0$):
   The induction is on the number $m \geq 0$ of derivation steps. Assume that $w = ya$, for some $a \in \Sigma$, $y, z \in \Sigma^*$; then $ya \Rightarrow s$ if (a) $y \Rightarrow r$, for some $r \in Q$, (b) $s \in \langle \langle r, a \rangle \rangle$, and (c) the key of $z$ in $\langle \langle r, a \rangle \rangle$ applies to $s$. 

Theorem 4. Let \( \mathcal{I} \) be \( k \)-predictable and let \( \mathcal{P} = (\Sigma \times \Sigma^k, Q, P, E, \rho) \) be its predictor. Let \( w' = yv \in \Sigma^* \) be the longest prefix of the input word \( w \) that is in \(|\mathcal{I}| = R_P \), and let \( y \) be any prefix of \( w' \). Then

1. The predictor operation is correct in the sense that \( y \Rightarrow q \) in predictor \( \mathcal{P} \) if and only if \( q \in Py \) and \( v \in R_q \).
2. The simulation stops with the remaining input \( v \) if and only if \( y \Rightarrow q \), for some \( q \in Q \), and one of the following holds: (a) \( v = 1 \); this implies that \( w \in |\mathcal{I}| \), or (b) \( v = az \), for some \( a \in \Sigma \), \( z \in \Sigma^* \) and there is no fork \( \langle q, a \rangle \) in \( \mathcal{I} \); this implies that \( w \notin |\mathcal{I}| \).

The predictor is optimal in the sense that there is no unnecessary nondeterminism, that is, no prefix \( y \) of \( w' = yaz \) derives a state from which it is impossible to continue the derivation. Moreover, if \( |z| \geq k \) in a \( k \)-predictable semiautomaton \( \mathcal{I} \), the induction step of the predictor is deterministic, because \( z \) is guaranteed to have a prefix \( u \) which is a minimal selector, by Corollary 1. Thus nondeterminism occurs only for words of length less than or equal to \( k \). As the next result shows, even that nondeterminism can be avoided, if one is interested only in determining whether \( w \in |\mathcal{I}| \), rather than in finding all the states reached by \( w \). Thus, for the membership problem, one can arbitrarily select any possible next state in any nondeterministic step, and always reach the same conclusion.

Corollary 2. In a predictor \( \mathcal{P} \), \( w \in |\mathcal{I}| \) if and only if \( w \Rightarrow q \), for some \( q \in Q \).

Example 4. The predictor \( \mathcal{P} \) of the semiautomaton \( \mathcal{I} \) of Fig. 4 is illustrated in Fig. 4 as well, if we interpret the edges appropriately. Thus, the edge labeled \( a[a, bb] \) is interpreted as two edges \( (a, [a]) \) and \( (a, [bb]) \).

Suppose the input tape has \( w = aaababaab \). The following deterministic computation takes place for the prefix \( aaababa \) of \( w \):

\[
\begin{align*}
[a] & \rightarrow q_1 \\
  a[a] & \rightarrow q_2 \quad a[a] \rightarrow q_4 \quad a[a] \rightarrow q_1 \quad b[1] \rightarrow q_3 \\
  a[ba] & \rightarrow q_3 \quad b[1] \rightarrow q_7 \\
  a[1] & \rightarrow q_1
\end{align*}
\]

First, the initial state is selected because the key of \( w \in \{q_1, q_6\} \) is \( a \) and it applies to \( q_1 \) only. Next, the key of \( aaababaab \) in \( \{q_2, q_3\} \) is \( a \) and it applies only to \( q_2 \). The rest of the computation is similar. When \( q_1 \) is reached and the remaining input is \( ab \), the key of \( b \) in \( \{q_2, q_3\} \) is \( b \), and it applies to both \( q_2 \) and \( q_3 \). Therefore, we have the following endings to the computation: \( q_1 \rightarrow q_2 \rightarrow q_6 \) and \( q_1 \rightarrow q_3 \rightarrow q_7 \). Thus the set of states reached by \( w \) is \( \{q_6, q_7\} \).

The mandate of maximal simulation was to exhibit the longest computation of the semiautomaton, regardless of the ultimate acceptance or rejection of the input word. In contrast to this, the second simulation decides as soon as possible whether the input is accepted or rejected; for this and other reasons it is more efficient than maximal simulation.

Definition 8. In a predictor \( \mathcal{P} \), for a word \( w \in \Sigma^* \) and \( T \subseteq Q \), we define the handle of \( w \) in \( T \) as follows. 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 either the handle is a minimal \( t \)-selector, or it is a prefix of a minimal \( t \)-selector or of a maximal \( t \)-nonselector.

In contrast to the computation of keys in maximal simulation, finding a handle does not involve looking for common prefixes. Also, a word has at most one handle in a set \( T \).
Lemma 3. Absence of a handle stops a minimal derivation before maximal derivation reaches the longest prex in converse is false in general. Using handles usually shortens the time for the membership decision, since the tor, and w

Example 5. On the other hand, for w = yv

Theorem 5. Let \( \mathcal{S} = (\Sigma, Q, P, E, \mathcal{P}) \) be a k-predictable semiautomaton, \( \mathcal{P} = (\Sigma \times \Sigma^{\leq k}, Q, P, E, \mathcal{P}) \), its predictor, and w = yv \in \Sigma^*. An input word of \( \mathcal{S} \). Then the predictor operation is correct in the following sense:

Example 5. The semiautomaton of Fig. 6 is 4-predictable. The word w = abab has no handle in P. Hence minimal simulation immediately yields the empty set of states, rejecting w. In contrast to this, maximal simulation has the following three derivations:

\[ \begin{align*}
(a) & \quad |ab \rightarrow p_1 \rightarrow q_1 \rightarrow r_1 \rightarrow s_1, \\
(b) & \quad |ab \rightarrow p_2 \rightarrow q_2 \rightarrow r_1 \rightarrow s_2, \\
(c) & \quad |ab \rightarrow p_3 \rightarrow q_3 \rightarrow r_1 \rightarrow s_3. \end{align*} \]

It stops after consuming aba, because there is no fork of the form \( \langle s_i, b \rangle \), for any i \in \{1, 2, 3\}. Thus, for w = abab, maximal simulation derives the empty set of states, rejecting w as well. On the other hand, for w = aba, both simulations have the same three derivations/yields above.

\[ \begin{align*}
(a) & \quad |ab \rightarrow p_1 \rightarrow q_1 \rightarrow r_1 \rightarrow s_1, \\
(b) & \quad |ab \rightarrow p_2 \rightarrow q_2 \rightarrow r_2 \rightarrow s_2, \\
(c) & \quad |ab \rightarrow p_3 \rightarrow q_3 \rightarrow r_3 \rightarrow s_3. \end{align*} \]
Example 6. In the predictor of Fig. 4, for $w = abba$, both simulations have only one derivation/yield corresponding to the path: $[a] \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{b[1]} q_6 \xrightarrow{b[1]} q_5$. Then both stop.

We can also use a simulation that takes advantage of both the efficiency of minimal simulation and the information provided by maximal simulation.

**Optimal simulation:** Start a minimal simulation and let it run as long as it finds handles. If $w$ is consumed, then it is accepted. Otherwise, when minimal simulation stops, maximal simulation takes over. Here, $w$ is rejected, but maximal simulation runs as far as the longest prefix of $w$ in $|\mathcal{F}|$.

Both maximal and minimal simulations operate “almost deterministically” in finding next states. However, by Corollary 3, we can achieve total determinism concerning acceptance. Run minimal simulation until more than one choice appears, and then choose an arbitrary branch in every step. Then minimal simulation is completely deterministic.

If the size of the alphabet $\Sigma$, 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.

5 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 $\mathcal{F}$. As long as the input word has length greater than $k$, 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.

**Acknowledgement** This research was supported by the Natural Sciences and Engineering Research Council of Canada under grant No. OGP0000871 and fellowship No. PDF-328881-2006.

**References**

6 Appendix

6.1 Proof of Theorem 1

(1) $\Rightarrow$ (2): If $T = \{t_1, \ldots, t_h\}$ is predictable, then it is $k$-predictable for some $k$. Let $w \in R_h \cap \Sigma^k$. If $w \not\in \sigma(t_i, T)$, then there exists $j \in \{1, \ldots, h\}$, $j \neq i$, such that $w \not\in R_j$. Then $w \in R_i \cap R_j \cap \Sigma^k$, contradicting the $k$-predictability of $T$. (2) $\Rightarrow$ (3): Obvious. (3) $\Rightarrow$ (4): If $\overline{\sigma}(T)$ is infinite, there exist $i$ and $j$ in $\{1, \ldots, h\}$, $i \neq j$, and $w \in \Sigma^*$, such that $w \in R_i \cap R_j$ and $|w| > k$. Let $w = uv$, where $|u| = k$; then also $u \in R_i \cap R_j$, since $R_i$ and $R_j$ are prefix-closed. Now $u \in R_i \cap \Sigma^k$, and, by (3), $u$ is a selector. But this cannot be, since also $u \in R_j$. Therefore $\overline{\sigma}(T)$ must be finite. (4) $\Rightarrow$ (1): If $\overline{\sigma}(T)$ is finite, let a longest word in $\overline{\sigma}(T) = \bigcup_{1 \leq i \neq j \leq h} (R_i \cap R_j)$ be of length $k - 1$. Then $R_i \cap R_j \cap \Sigma^k = \emptyset$, for all $i, j \in \{1, \ldots, h\}, i \neq j$, and $T$ is predictable. \qed

6.2 Proof of Theorem 2

If $\mathcal{S}$ is deterministic, then $k = 0$, and the bound is trivially satisfied. If $\mathcal{S}$ is not deterministic, there must be at least one critical set $T = \{t_1, \ldots, t_h\}$, $h \geq 2$, which is $k$-predictable.

Case 1: $\Sigma# = 1$

We claim that $T$ is predictable if and only if at most one of the languages in $\{R_i\}_{1 \leq i \leq h}$ 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 $R_i$ and $R_j$ are infinite then $R_i \cap R_j = a^*$, since $R_i$ and $R_j$ are prefix-closed. Hence $T$ is not predictable by Theorem 1, and $R_i$ and $R_j$ cannot both be infinite.

Without loss of generality, assume now that $R_{i_1}, \ldots, R_{i_{n-1}}$ are finite. We distinguish two cases:

1. $R_h$ is infinite. Then $R_h = a^*$, since $R_h$ is prefix-closed. Let $w$ be a longest word in $\bigcup_{1 \leq i < h} R_i$, and assume that $w \in R_j$, $j \neq h$. Since $R_j$ 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 $t_j u = t_j v$. Then also $ux^2v \in R_j$, contradicting that $w$ is a longest word of $R_j$. Also, a path $\pi$ from $t_j$ spelling $w$ cannot visit $t_h$, otherwise $R_j$ would be infinite, since $R_h$ is infinite. Thus $\pi$ has at most $n - 1$ states, and $|w| \leq n - 2$. Now $T$ cannot be $|w|$-predictable, because $w \in R_j \cap R_h$ (since $R_h = a^*$), but it is $(|w| + 1)$-predictable. Thus we must have $k = |w| + 1 \leq n - 1$.

2. $R_h$ is finite. Let $w$ be a longest word in $\bigcup_{1 \leq i < h} R_i$, and assume $w \in R_j$. As above, a path originating in $t_j$ and spelling $w$ can involve at most $n$ states; thus $|w| \leq n - 1$. If $|w| < n - 1$ then clearly $k \leq |w| + 1 \leq n - 1$. When $|w| = n - 1$, a path $\pi$ 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 $R_j$. Thus, $k = |w| = n - 1$ in this case.

The semiautomaton in Figure 7 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 Figure 7 and make states 1 and 2 initial, we have an example in which the bound can be reached when $|\mathcal{S}|$ is finite. \qed

Case 2: $\Sigma# > 1$

First, we develop necessary conditions on paths from critical sets in predictable semiautomata.
Lemma 4. Let $\mathcal{S} = (\Sigma, Q, P, E)$ be a predictable semiautomaton, with $Q = \{1, \ldots, n\}$, and let $r_1, s_1$ be two distinct states of a critical set in $\mathcal{S}$. If $w = a_1 \ldots a_{m-1}$ is a longest word in $R_{r_1} \cap R_{s_1}$, let $\pi_1 = (r_1, a_1, r_2) \ldots (r_{m-1}, a_{m-1}, r_m)$ and $\pi_2 = (s_1, a_1, s_2) \ldots (s_{m-1}, a_{m-1}, s_m)$ be two paths spelling $w$, originating from $r_1$ and $s_1$, respectively. Then the sequence $\mathcal{L} = (r_1, s_1) \ldots (r_m, s_m)$ of ordered pairs of states encountered by $\pi_1$ and $\pi_2$ must satisfy the following conditions: First, $r_1 \neq s_1$, and, for all $i, j \in \{1, \ldots, m\}, i \neq j$, (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 $s_j \neq r_i$.

Proof. Let $w = uxv$, where $u = a_1 \ldots a_{i-1}$, $x = a_i \ldots a_{j-1}$, and $v = a_j \ldots a_{m-1}$ are the labels of the paths $r_1 \xrightarrow{u} r_i \xrightarrow{x} r_j \xrightarrow{v} r_{m-1}$, respectively, in $\pi_1$, and hence also the labels of the paths $s_1 \xrightarrow{u} s_i \xrightarrow{x} s_j \xrightarrow{v} s_{m-1}$ in $\pi_2$. By our hypothesis, $r_1 \neq s_1$. For the remaining conditions we have: (1): If there exist $i, j$, $1 \leq i < j \leq t$, with $r_i = r_j$ and $s_i = s_j$, then $ux^jv \in R_{r_i} \cap R_{s_i}$, contradicting the maximality of $|w|$. (2): If there exist $i, j$, $1 \leq i < j \leq t$, such that $r_i = s_j$ and $r_j = s_i$, then one verifies that $ux^jv \in R_{r_i} \cap R_{s_i}$, contradicting the maximality of $|w|$. (3): If there exist $i, j$, $1 \leq i < j \leq t$, such that $r_i = s_i$ and $r_j = r_j$, then $ux^jv \in R_{r_i} \cap R_{s_i}$, contradicting the maximality of $|w|$. Similarly, if $s_j = r_i$, since $s_i = s_j = r_i$, there is a loop labeled $x$ on $r_i$ and again $ux^jv \in R_{r_i} \cap R_{s_i}$.

Next we prove a combinatorial result about sequences $\mathcal{L}$ satisfying the conditions of Lemma 4.

Lemma 5. 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 the conditions of Lemma 4, then $m \leq (n^2 - n)/2$ and the bound is sharp.

Proof. We first show that the bound can be achieved. Consider the sequence $\mathcal{L} = (1, 1), \ldots, (1, n), (2, 1), \ldots, (2, n), \ldots, (n, 1), \ldots, (n, n)$, which has $n^2$ elements and satisfies Condition (1). If we remove the pairs $(i, i)$, for all $1 \leq i \leq n$, we have a sequence of $n^2 - n$ pairs, in which $r_i \neq s_1$ and which satisfies Condition (3) as well, since $r_i$ is never equal to $s_i$. Finally, for all $i \neq j$, remove either $(i, j)$ or $(j, i)$. Now the sequence also satisfies Condition (2). Since there are $(n^2 - n)/2$ pairs removed in the last step, the final sequence has $(n^2 - n)/2$ elements. Thus the bound can be reached.

Next, we prove that $(n^2 - n)/2$ is an upper bound by induction on $n$. If $n = 1$, then only the empty sequence satisfies all the conditions. Hence $m = 0 = (n^2 - n)/2$. If $n = 2$, then the empty sequence, $(1, 2)$ and $(2, 1)$ are the only sequences satisfying the conditions. Here $m \leq 1 = (n^2 - n)/2$.

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 of elements from $\{1, \ldots, n\}$ satisfying all the conditions, and assume for the sake of contradiction 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 the element $n$, 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 \(m'\) elements 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, \(m' \leq M(n-1) \leq [(n-1)^2 - (n-1)]/2 = (n^2 - n)/2 - (n-1)/2\).

In addition to the elements of \(\mathcal{L}'\), \(\mathcal{L}\) contains elements 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, in this case, we have \(M(n) = m' + 1 \leq (n^2 - n)/2 - (n-2) < (n^2 - n)/2\), which contradicts our assumption.

If \(\mathcal{L}\) does not contain \((n,n)\), it contains at most \((n-1)\) pairs involving \(n\). In this case \(M(n) \leq m' + (n-1) \leq (n^2 - n)/2\), which is again a contradiction.

Consequently, \(M(n) \leq (n^2 - n)/2\) and the induction step goes through. \(\square\)

**Proof of Theorem 2 for Case 2**

Let \(r_1, s_1\) be two distinct states of \(T\). By Lemma 4, every sequence \(\mathcal{L} = (r_1, s_1), \ldots, (r_n, s_n)\) must satisfy all the conditions of the lemma, and by Lemma 5, the length of a longest word \(w \in R_{r_1} \cap R_{s_1}\) is \(|w| = m - 1 \leq (n^2 - n)/2\). This implies that, if \(T\) is \(k\)-predictable, then necessarily \(k \leq |w| + 1 \leq (n^2 - n)/2\). Since this holds for all critical sets, it holds for \(\mathcal{S}\).

Next we prove that this bound is achievable for a suitable alphabet. Let \(n \geq 1\), and let \(\mathcal{L} = (r_1, s_1), \ldots, (r_n, s_n)\) be a sequence satisfying the conditions of Lemma 5, 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 \(E_1 = \{r_1, a_1\}\) is a fork in \(\mathcal{S}\) and its fork set \(\langle r_1, a_1 \rangle = \{r_1, s_1\}\) is a critical set in \(\mathcal{S}\). Consider the word \(w = a_2 \ldots a_k\); clearly, \(w \in R_{r_1} \cap R_{s_1}\), implying that the semiautomaton is not \((k-1)\)-predictable, since \(|w| = k - 1\). Observe now that \(R_{r_1} \cap R_{s_k} = \emptyset\); 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 Condition (1) of Lemma 4. Thus \(w\) is a longest word in \(R_{r_1} \cap R_{s_1}\), implying that \(\langle r_1, a_1 \rangle\) 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 \((r_1, s_1)\) in \(\mathcal{L}\) with \(r_1 = s_1\), since only such states have outgoing edges with a same label, by the construction of \(\mathcal{S}\). By the argument used in the proof of Lemma 5, the sequence \(\mathcal{L}\) cannot be of maximal length. Hence there are no other forks, and \(\mathcal{S}\) is \(k\)-predictable, where \(k = (n^2 - n)/2\) is the smallest such integer. \(\square\)

**6.3 Proof of Proposition 1**

Properties (1)–(3) and (5) follow from the definition of \(\mathcal{S}(T)\).

For (4), if \(w\) is a minimal \(t_r\)-selector, then \(y_0w\) is \(t_r\)-singular by (3). By definition, \(w\) has no prefix \(u\) which is a selector; hence no state \(y_0u\) is singular, again by (3), and no state \(y_0u\) is null, by (1). Hence all the states
of the form $\gamma_0u$ are plural, and $w$ is primary. Conversely, if $w$ is primary, then $\gamma_0, \ldots, \gamma_{n-1}$ are plural, no prefix of $w$ is a selector, and $w$ is minimal.

For (6), if $\gamma = \gamma_0w$ is $t_i$-final, then $\gamma$ is plural and its $i$th component is nonempty. Thus $w$ is not a $t_i$-selector and $w \in R_{t_i}$. Hence $w$ must be a $t_i$-nullary selector. Because every extension $wa$ leads to a state with an empty $i$th 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 $t_i$-nullary selector, then $w \in R_{t_i}$ and $w \notin R_{t_j}$ for some $j \neq i$. Hence $\gamma_0w$ is a state with nonempty $i$th and $j$th components. If $\gamma_0w$ is not $t_{i}$-final, then there exists $a \in \Sigma$ such that $\gamma_0wa$ has a nonempty $i$th component. But then $wa \in R_{t_i}$ and $w$ is not maximal. Therefore $\gamma_0w$ is $t_i$-final.

6.4 Proof of Proposition 2

1. Suppose that $w = uv$, and $w$ and its prefix $u$ are both selectors, say $w$ is a $t_i$-selector and $u$ is a $t_j$-selector. Then, by Proposition 1 (3), $u$ leads $\gamma_0$ to a $t_j$-singular state $\gamma$, and $w$ leads $\gamma_0$ to a $t_i$-singular state $\gamma'$. Since $\gamma' = \gamma v$, this contradicts Proposition 1 (1), for the empty $i$th component in $\gamma$ cannot become nonempty in $\gamma'$. Thus $a t_j$-selector can be a prefix of a $t_i$-selector only if $i = j$. However, if $w$ is a minimal selector, it cannot have a selector as a prefix, and the claim holds.

2. Suppose that selector $u$ is a prefix of a nullary selector $uv$. Now $u$ leads to a singular state, whereas $uv$ requires that the state reached by it be plural. This contradicts Proposition 1 (1).

3. If $u$ is a maximal $t_i$-nullary selector, then the $i$th component of $\gamma_0ua$ is empty for all $a \in \Sigma$. Hence no extension $uav$ of $u$ is in $R_{t_i}$, and hence cannot be a $t_i$-selector.

4. If $w$ is a maximal $t_i$-nullary selector then the $i$th component of $\gamma_0wa$ is empty for all $a \in \Sigma$. It follows that no extension $wv$ of $w$ may be a maximal $t_i$-nullary selector. Thus the claim holds.

6.5 Proof of Theorem 3

If $\mathcal{D}(T)$ has a cyclic plural state $\gamma$, then $\gamma$ has two nonempty components, say $i$ and $j$. Since $\mathcal{D}$ is accessible, $\gamma$ is reachable by some word $u \in \Sigma^*$ from the initial state $\gamma_0$. Since $\gamma$ is cyclic, there is a word $v \in \Sigma^*$ such that $uv = \gamma$. This implies that $uv^n \in R_{t_i} \cap R_{t_j}$ for all $n$. Since $n$ can be arbitrarily large, $\mathcal{D}(T)$ is not finite and $T$ is not predictable, by Theorem 1.

Conversely, suppose that $\mathcal{D}(T)$ does not have any cyclic plural states for any critical set $T$. Then there are no arbitrarily long words from $\gamma_0$ to a plural state, since no plural state can be repeated, for it would then be cyclic. Let $k - 1$ be the length of a longest such word, taken over all critical sets $T$. Since $\mathcal{D}(T)$ is complete, there is a path from $\gamma_0$ for every word in $\Sigma^*$. Thus every word of length $k$ is either singular or nullary. Then the length of a longest word among the primary and nullary words over all $T$ is $k$.

We now claim that every critical set $T$ is $k$-predictable. By construction of $\mathcal{D}(T)$, any word $w \in \Sigma^k$ leads either to $\gamma_0$ or to a singular state. In the first case, $w \notin R_T$ by Proposition 1 (2). In the second case, $w$ is a selector by Proposition 1 (3). In either case the condition for $k$-predictability holds. Finally, $\mathcal{D}$ is not $(k - 1)$-predictable, since it has a plural state reached by a word of length $k - 1$. □
6.6 Proof of Corollary 1

Each minimal selector is a primary word by Proposition 1 (4). By Theorem 3, \( \emptyset(T) \) has no cyclic plural states, since \( T \) is predictable. Hence all primary words are of length \( \leq k \), for some \( k \), and the first claim follows. For the second claim, note that a word of length \( k \) is either singular or nullary. Since the prefix of length \( k \) of a word in \( R_T \) cannot be nullary, it must be singular. By Proposition 1 (3), the prefix must be a selector, and hence must have a prefix that is a minimal selector.

6.7 Proof of Proposition 3

1. Suppose \( T \) is \( k \)-predictable and \( t_i \in T \) has no selectors. If there exists \( w \in R_{t_i} \), \( |w| \geq k \), and \( w \) is not a selector, then \( w \in R_{t_j} \) for some \( t_j \in T, i \neq j \). If \( u \) is a prefix of \( w \) of length \( k \), then \( u \in R_{t_i} \cap R_{t_j} \), meaning that \( R_{t_i} \cap R_{t_j} \cap \Sigma^k \neq \emptyset \), contradicting the \( k \)-predictability of \( T \).

2. If \( h = 1 \), then \( T = \{ t_1 \} \), and \( 1 \) is a minimal \( t_1 \)-selector. Now assume that \( h > 1 \). If \( t_i \) has a selector, then it has a minimal selector. Therefore assume that state \( t_i \) has no selector; we show that it then has a maximal nonselector. First, if \( R_{t_i} = \{ \} \), then \( 1 \) is a maximal \( t_i \)-nonselector. Next suppose that \( R_{t_i} \) is finite, and let \( w \) be a longest word in \( R_{t_i} \). Since \( w \) is not a selector, there must exist a \( j \neq i \) such that \( w \in R_{t_j} \). Hence \( \gamma_0 w \) is plural. Also, for any \( a \in \Sigma \), the \( i \)th component of \( \gamma_0 wa \) is empty, since \( w \) is a longest word. Thus \( \gamma_0 w \) is \( t_i \)-ultimate, and \( w \) is a maximal \( t_i \)-nonselector. By (1), the case where \( R_{t_i} \) is infinite is impossible.

3. If \( T \) is \( k \)-predictable, and \( w \) is a nonselector with \( |w| \geq k \), then \( w \in R_T \) and \( w \) has a selector as a prefix, by Corollary 1 (2). This contradicts Proposition 2 (2).

6.8 Proof of Lemma 1

Since \( w \in R_{t_i} \), \( w \) is either a \( t \)-selector or a \( t \)-nonselector. If it is a \( t \)-selector, then it has a prefix that is a minimal \( t \)-selector, and (1) holds.

If (1) does not hold, then necessarily \( |w| < k \). Otherwise \( w \in R_T \cap \Sigma^k \Sigma^* \) and, by Corollary 1, \( w \) has a prefix which is a minimal selector, contradicting the assumption that (1) does not hold.

Assume now that (1) does not hold, and implicitly, 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 2 (2). 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 Proposition 3 (3), every non-selector is of length less than \( k \). Therefore we must eventually reach a maximal \( t \)-nonselector, and (3) holds.

6.9 Proof of Lemma 2

Let \( w' \) be the longest prefix of \( w \) that is in \( R_T \). Then there exists \( t \in T \) such that \( w' \in R_t \), and Lemma 1 applies; thus one of the three cases occurs. If a prefix \( u \) of \( w' \) is a minimal selector, then \( u \) is the key of \( w' \),
and also of $w$, in $T$. This follows from the fact that no minimal selector or maximal nonselector can be an extension of a minimal selector, by Proposition 2 (1) and (2), respectively. If one of the other two conditions of Lemma 1 holds, $w'$ is a prefix of a minimal $t$-selector or of a maximal $t$-nonselector. Then clearly $w'$ is the key of $w$ in $T$, since no prefix of $w$ longer than $w'$ is in $R_T$. This proves (1).

For the second claim, suppose $w'$ is an arbitrary prefix of $w$ that is in $R_T$. First suppose that $w'$ has a prefix $u$ which is a minimal selector in $T$. By a reasoning similar to that in the proof of (1) above, $u$ is the key of $w'$ in $T$. Now, if $w' \in R_t$, for some $t \in T$, then $u$ is a minimal $t$-selector and $u$ applies to $t$ in $T$, $u$ being its own prefix. Conversely, if the key $u$ of $w'$ in $T$ applies to $t$, then $w'$ can only be in $R_t$, since $u$ is then a minimal $t$-selector, being a minimal selector in $T$.

Now suppose that $w'$ does not have a prefix which is a minimal selector. By Corollary 1, we must have $|w'| < k$.

If $w' \in R_t$ for some $t \in T$, either (2) or (3) holds, by Lemma 1. Hence $w'$ is a prefix of a minimal $t$-selector or a maximal $t$-nonselector $u$ in $T$. Since $w'$ is its own longest prefix, the key in this case is $w'$ itself, and $w'$ applies to $t$.

Conversely, assume that the key $x$ of $w'$ in $T$ applies to $t \in T$; then $x$ is a prefix of a minimal $t$-selector $u$ or of a maximal $t$-nonselector $u'$. In either case, $u$ and $u'$ are in $R_t$, and so is the prefix $x$. We prove next that $x = w'$. From this it follows that $w' \in R_t$.

To prove that $x = w'$, note that, if $w' \nsubseteq R_t$, then $w' \in R_s$ for some $s \in T$, since $w' \in R_T$. By Lemma 1, since $w' \in R_s$, either (2) or (3) holds, and $w'$ is a prefix of a minimal $s$-selector or a maximal $s$-nonselector in $T$. It is its own key in $T$, since it is its own longest prefix. Thus our claim that $x = w'$ holds. $\Box$

### 6.10 Proof of Theorem 4

First, we show that, if $y \Rightarrow q$, then $q \in Py$ and $v \in R_q$. We proceed by induction on the length of the prefix $y$. If $1 \Rightarrow s$, then $s \in P$ by Definition 7 (1). Thus $s \in P1 = P$. Since $w'$ is in $R_P$ by assumption, and the key of $w'$ in $P$ applies to $s$, we have $w' \in R_s$, by Lemma 2 (2). Therefore the claim holds for the basis. Now assume that, for an arbitrary prefix $y$ of $w' = yz$, if $y \Rightarrow r$, then $r \in Py$ and $v \in R_r$. Suppose that $ya \Rightarrow s$. Then $y \Rightarrow r$, for some $r \in Q$, $s \in \langle (r, a) \rangle$, and the key of $z$ in $\langle (r, a) \rangle$ applies to $s$. By the induction hypothesis, $r \in Py$ and $az \in R_r$. Since $s \in \langle (r, a) \rangle$, there is an edge $(r, a, s) \in E$; hence $s \in Pya$. Since $z \in R_{\langle (r, a) \rangle}$ because $az \in R_r$, and the key of $z$ in $\langle (r, a) \rangle$ applies to $s$, by Lemma 2 (2) we have $z \in R_s$. Thus the induction goes through, and the claim holds.

Second, assume that $q \in Py$ and $v \in R_q$; we show that $y \Rightarrow q$. Again, we proceed by induction on the length of the prefix. Consider first the factorization $w' = yv = 1w'$. By Lemma 2 (2), if $p \in P$ and $w' \in R_p$, then the key of $w'$ in $P$ applies to $p$. By Definition 7 (1), 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 a fork $\langle r, a \rangle$. Let $s$ be any state in $T = \langle (r, a) \rangle$ such that $z \in R_s$. By the induction hypothesis, $y \Rightarrow r$. Since $s \in T$ and $z \in R_s$, the key of $z$ in $T$ applies to $s$ by Lemma 2 (2). Therefore $ya \Rightarrow s$, and the claim holds.

Now consider termination. Since 1 is a prefix of all words, the input word always has a key in $P$, and Step 1 is always executed. Consequently, the derivation can stop only if $w = yv$ and some state $q$ has been derived by $y$. 
If $v = 1$, then $v$ does not begin with a letter, the induction step cannot be carried out, and the derivation stops. Here $w = y$ is clearly in $|\mathcal{S}|$, since $q \in Py$ by Theorem 4 (1), and that implies $y \in R_P = |\mathcal{S}|$.

If $v = az$, for some $a \in \Sigma$, $z \in \Sigma^*$, and there is a fork $\langle q, a \rangle$ in $\mathcal{S}$, the derivation continues, since $z$ always has a key in $\langle \langle q, a \rangle \rangle$. Thus the derivation can stop only if $v = az$, but there is no fork $\langle q, a \rangle$ in $\mathcal{S}$. Clearly, $az \notin R_g$. Suppose now that $w = yaz \in |\mathcal{S}|$. Since $y \Rightarrow q$, we have $q \in Py$ and $az \in R_q$, by Theorem 4 (1). This is a contradiction, and $w \notin |\mathcal{S}|$.

6.11 Proof of Corollary 2

We claim that, if $w \Rightarrow q$, then $w \in |\mathcal{S}|$. If $w = 1$, then $1 \Rightarrow s$ implies $s \in P$, showing that $P$ is not empty. But then $1 \in R_g \subseteq |\mathcal{S}|$. Now assume that $y \Rightarrow r$ implies $y \in |\mathcal{S}|$, and consider $ya$, for some $a \in \Sigma$. If $ya$ derives $s$, then $s \in \langle \langle r, a \rangle \rangle$, and there is an edge $(r, a, s)$ in $\mathcal{S}$. Hence $ya \in |\mathcal{S}|$, and our claim follows.

Conversely, if $w \in |\mathcal{S}|$, then $w' = w$; also there exists $q \in Pw$, by definition of $|\mathcal{S}|$. Since $1 \in R_q$, by Theorem 4, applied with $y = w'$ and $v = 1$, we have $w = w' \Rightarrow q$.

6.12 Proof of Lemma 3

If $y \Rightarrow q$, then $y \Rightarrow q$, since a handle in $T$ is also a key in $T$. For the converse, we proceed by induction on the length of a prefix of $w$.

If $y = 1$ and $y \Rightarrow q$, then, by definition, $q \in P$ and the key of $w$ in $P$ applies to $q$. By Lemma 2 (1), the key of $w$ in $P$ is either a minimal selector or $w$ itself, since $w$ is the longest prefix of $w$ which is in $R_P = |\mathcal{S}|$.

If the key is a minimal selector, then it is also a handle of $w$ in $P$, by definition. If the key is $w$ itself, then, by the definition of a key, $w$ is a prefix of a minimal selector or of a maximal nonselector in $P$. In either case, $w$ is a handle as well, by definition. Thus, $y \Rightarrow q$, since the handle of $w$ in $P$ applies to $q$.

Assume now that $w = yaz$ and the implication holds for $y$. We prove that, if $ya \Rightarrow q$, then $ya \Rightarrow q$. Let $r$ be a state such that $q \in \langle \langle r, a \rangle \rangle = T$ and $y \Rightarrow r$. Then the key of $z$ in $T$ applies to $q$. Since $ya \Rightarrow q$, we have $z \in R_T$ by Theorem 4. By Lemma 2 (1), the key is either $z$ itself or a minimal selector; in either case, the key is the handle of $z$ in $T$, which applies to $q$. By definition, $ya \Rightarrow q$, and the induction is complete.

6.13 Proof of Theorem 5

If $w \in |\mathcal{S}|$ then $y \Rightarrow q$ if and only if $y \Rightarrow q$, by Lemma 3. By Theorem 4, $y \Rightarrow q$ if and only if $q \in Py$ and $v \in R_q$. Hence (1) holds.

For (a), if $w$ has no handle in $P$, then $1 \rightarrow p$ is false for all $p \in P$, by Definition 9, and the simulation stops. Now if $w \in |\mathcal{S}|$, then $w \in R_P$, and hence $w \in R_p$, for some $p \in P$. Since also $p \in P1$, Part (1) of the theorem applies, and $1 \rightarrow p$, which is a contradiction; thus $w \notin |\mathcal{S}|$.

For (b), if the minimal 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 $w \in |\mathcal{S}|$.

For (c), assume that $w = yv = yaz$, the minimal simulation has consumed $y$, $y \Rightarrow q$, but there is no fork $\langle q, a \rangle$ in $\mathcal{S}$. Clearly, the induction step cannot be carried out, and $az \notin R_q$. Suppose now that $w \in |\mathcal{S}|$. Since $y \Rightarrow q$, we have $q \in Py$ and $az \in R_q$, by Theorem 5 (1). This is a contradiction; thus $w \notin |\mathcal{S}|$. 

Predictable Semiautomata
For (d), assume that $w = yv = yaz$, the minimal simulation has consumed $y$, $y \to q$, and $z$ has no handle in $(q,a))$. Then the derivation stops because the induction step cannot be carried out. If $w \in |\mathcal{S}|$, since $y \to q$, we know by Part (1) of the theorem that $q \in P_y$ and $v \in R_q$. Also, there exists $r \in (q,a)$ such that $(q,a,r)$ is an edge in $\mathcal{S}$ and $z \in R_y$. But now $r \in Py$ and $z \in R_y$ implies that $ya \to r$, again by Part (1). Thus $z$ must have a handle in $(q,a))$, which is a contradiction. We conclude that $w \not\in |\mathcal{S}|$. \qed

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

6.14 Proof of Corollary 3

One verifies that $w \to q$ implies $w \in |\mathcal{S}|$. Conversely, if $w \in |\mathcal{S}|$, then there exists $q \in Pw$, by definition of $|\mathcal{S}|$. Since also $1 \in R_y$, by Theorem 5 (1) applied with $y = w$ and $v = 1$, we have $w \to q$. \qed