

# Representation of a class of nondeterministic semiautomata by canonical words<sup>☆</sup>

Janusz. A. Brzozowski\*

*David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ont., Canada N2L 3G1*

---

## Abstract

It has been shown recently that deterministic semiautomata can be represented by canonical words and equivalences; that work was motivated by the trace-assertion method for specifying software modules. Here, we generalize these ideas to a class of nondeterministic semiautomata. A semiautomaton is *settable* if, for every state  $q$ , there exists a word  $w_q$  such that  $q$ , and no other state, can be reached from some initial state by a path spelling  $w_q$ . We extend many results from the deterministic case to *settable* nondeterministic semiautomata. Now each word has a number of canonical representatives. We show that a prefix-rewriting system exists for transforming any word to any of its representatives. If the set of canonical words is *prefix-continuous* (meaning that, if  $w$  and a prefix  $u$  of  $w$  are in the set, then all prefixes of  $w$  longer than  $u$  are also in the set), the rewriting system has no infinite derivations. Examples of specifications of nondeterministic modules are given.

© 2006 Elsevier B.V. All rights reserved.

*Keywords:* Canonical word; Module; Nondeterministic; Prefix-continuous; Prefix-rewriting; Semiautomaton; Settable; Specification; Trace assertion

---

## 1. Introduction

A software or hardware module can often be conveniently described by an automaton. In the *trace-assertion method*, certain important input words (traces), called “canonical”, are first identified and used to represent the states of the automaton. Each of the remaining words is declared equivalent to a canonical word; this equivalence relation in effect specifies the transitions of the automaton. A rewriting system is used to transform any word to its canonical representative. Outputs are first defined for canonical words, and the definition is then extended to arbitrary words.

Trace-assertion specifications of software modules were introduced by Bartussek and Parnas in 1977 [1], and later studied by many authors; see [2] for a recent discussion of the literature on this topic. It turns out that the important

---

<sup>☆</sup> This research was supported by the Natural Sciences and Engineering Research Council of Canada under Grant no. OGP0000871.

\* Tel.: +1 519 8884567; fax: +1 519 8851208.

E-mail address: [brzozo@uwaterloo.ca](mailto:brzozo@uwaterloo.ca).

URL: <http://maveric.uwaterloo.ca>.

issues—of selecting appropriate canonical traces, constructing assertions about equivalence, and finding a suitable rewriting system—can all be treated in the framework of semiautomata (automata without outputs). Relations between trace-assertion specifications and deterministic semiautomata were recently studied in [3]. The additional features associated with outputs, and also applications to practical modules were examined in [2].

Nondeterministic trace-assertion specifications were first considered in [7,8]. The model of a module used in [7], however, is considerably more complex than ours. Among other differences, [7] deals with multi-object modules, whereas we deal exclusively with simple semiautomata. The model used in [8] is also quite different from ours. It admits as traces so-called “step sequences”, which are sets of words, whereas, in the present work, traces are words. Also, we consider only words over the input alphabet of a module, whereas [8] allows also input–output pairs as members of the alphabet. Neither [7] nor [8] deals with rewriting systems, which constitute a major concern of the present paper.

The remainder of the paper is organized as follows. Basic notions about nondeterministic semiautomata are defined in Section 2, whereas Section 3 deals with the class of nondeterministic semiautomata, introduced by Janicki and Sekerinski [8], which we call “settable”. Prefix-rewriting systems are discussed in Section 4, and applied to settable semiautomata in Section 5. Special properties of the rewriting systems, in the case where the set of canonical words is prefix-continuous, are studied in Section 6. Section 7 extends the rewriting system to words that do not have canonical prefixes. In Section 8 we consider complete semiautomata. Several examples of specifications of nondeterministic modules are presented in Section 9.

## 2. Semiautomata

We base our notation for functions and semiautomata on that of Eilenberg [5]. If  $f : X \rightarrow Y$  (also denoted  $X \xrightarrow{f} Y$ ) is a function, we write  $xf$  for the value of  $f$  at  $x$ . If  $g : Y \rightarrow Z$  is another function, then  $xfg$  is unambiguous without parentheses. Also, the element  $x \in X$  can be interpreted as a function  $x : S \rightarrow X$ , where  $S$  is some singleton, and the value of this function is  $x$ . Then,  $xfg$  is the composition of functions  $S \xrightarrow{x} X \xrightarrow{f} Y \xrightarrow{g} Z$ .

If  $\Sigma$  is an alphabet (finite or infinite), then  $\Sigma^+$  and  $\Sigma^*$  denote the free semigroup and the free monoid, respectively, generated by  $\Sigma$ . The empty word is 1. 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 set  $X \subseteq \Sigma^*$  is *prefix-free* if no word of  $X$  is the prefix of any other word of  $X$ . A set  $X$  is *prefix-closed* if, for any  $w \in X$ , every prefix of  $w$  is also in  $X$ . A set  $X$  is *prefix-continuous* [3] if, whenever  $x = uav \in X$  and  $a \in \Sigma$ , then  $u \in X$  implies  $ua \in X$ . Both prefix-free and prefix-closed sets are prefix-continuous.

A *semiautomaton* [6]  $\mathcal{S} = (\Sigma, Q, I, E)$  consists of an *alphabet*  $\Sigma$ , a set  $Q$  of *states*, a set  $I \subseteq Q$  of *initial states*, and a set  $E$  of *edges* of the form  $(p, a, q)$ , where  $p, q \in Q$  and  $a \in \Sigma$ . Sets  $\Sigma, Q, I$ , and  $E$  may be finite or infinite. An edge  $(p, a, q)$  *begins* at  $p$ , *ends* at  $q$ , and has *label*  $a$ . It is also denoted as  $p \xrightarrow{a} q$ .

A *path*  $\pi$  is a finite sequence  $\pi = (q_0, a_1, q_1)(q_1, a_2, q_2) \dots (q_{k-1}, a_k, q_k)$  of consecutive edges,  $k > 0$  being its *length*,  $q_0$ , its *beginning*,  $q_k$ , its *end*, and word  $w = a_1 \dots a_k$ , 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  $P \subseteq Q$  and  $w \in \Sigma^*$ , then  $Pw = \{q \in Q \mid p \xrightarrow{w} q, \text{ for some } p \in P\}$ . Note that, for all  $P \subseteq Q, u, v \in \Sigma^*$ ,  $(Pu)v = P(uv)$ . If  $P = \{p\}$ , we write  $pw$  for  $Pw$ ; if  $Pw = \{q\}$ , we write  $Pw = q$ .

A semiautomaton  $\mathcal{S}$  is *accessible* if, for every state  $q$ , there exists  $i \in I, w \in \Sigma^*$  such that there is a path  $i \xrightarrow{w} q$ .

The *language*  $|\mathcal{S}|$  of a semiautomaton  $\mathcal{S} = (\Sigma, Q, I, E)$  is the set of all words that are labels of paths starting in initial states in  $\mathcal{S}$ , that is  $|\mathcal{S}| = \{w \in \Sigma^* \mid Iw \neq \emptyset\}$ . Note that  $|\mathcal{S}|$  is prefix-closed; in particular, if  $|\mathcal{S}| \neq \emptyset$ , then  $1 \in |\mathcal{S}|$ .

A semiautomaton is *complete* if  $I \neq \emptyset$  and, for every  $q \in Q$  and  $a \in \Sigma$ , there is an edge  $(q, a, p) \in E$ , for some  $p \in Q$ . In a complete semiautomaton,  $qw \neq \emptyset$ , for all  $q \in Q, w \in \Sigma^*$ . The language of a complete semiautomaton is  $\Sigma^*$ .

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  $(q, a, p)$ . In case  $\mathcal{S}$  is deterministic and has initial state  $i$ , we write  $\mathcal{S} = (\Sigma, Q, i, E)$ .

In  $\mathcal{S} = (\Sigma, Q, I, E)$ , we define the *language*  $L_q$  of state  $q \in Q$ :  $L_q = \{w \in \Sigma^* \mid q \in Iw\}$ . If  $\mathcal{S}$  is complete, each  $w \in \Sigma^*$  belongs to at least one language  $L_q$ .

### 3. Settable semiautomata

A semiautomaton is *settable to state*  $q$  if there exists a word  $w \in \Sigma^*$  such that  $Iw = q$ , and  $S$  is *settable*,<sup>1</sup> if it is settable to  $q$  for every  $q \in Q$ . Clearly, every settable semiautomaton is accessible. Settability can be tested by examining the accessible deterministic semiautomaton  $S\mathcal{A}$  obtained from  $S$  by the well known subset construction [5]. It is clear that  $S$  is settable to  $q$  if and only if  $\{q\}$  is a state accessible from the initial state  $I$  in  $S\mathcal{A}$ .

From now on we consider only settable semiautomata. Let  $S = (\Sigma, Q, I, E)$  be a settable semiautomaton, and let  $R_q = \{w \in \Sigma^* \mid Iw = q\}$  be the set of all words that set  $S$  to  $q$ . Note that  $R_q \subseteq L_q$ . Also, let  $R_Q = \cup_{q \in Q} R_q$  be the set of *setting* words of  $S$ . Note that, if  $S$  is settable, then, for all  $p, q \in Q$ ,  $L_p = L_q$  implies  $p = q$ .

Let  $\chi : Q \rightarrow R_Q$  be an arbitrary mapping assigning to  $q$  a word  $w_q \in R_q$ . Note that  $\chi$  is injective. If  $P$  is a subset of  $Q$ , then  $P\chi = \{p\chi \mid p \in P\}$ . The set  $X = Q\chi$  of words assigned to  $Q$  is the set of *canonical words* of  $S$ .

Unless stated otherwise, we assume that  $\chi$  has been selected; we call the word  $q\chi$  the *canonical word of state*  $q$ . Furthermore, if  $w \in \Sigma^*$  is such that  $q \in Iw$ , then  $q\chi$  is a *canonical representative* of  $w$ . The set of all canonical representatives of  $w$  is denoted by  $C_w$ . Any word  $w$  may have more than one canonical representative, or none at all; if  $w$  is canonical, however, then  $C_w = \{w\}$ . Note also that if  $1$  is a setting word, then  $I$  is a singleton.

Using  $X$  and  $E$ , define a binary relation  $G \subseteq X\Sigma \times X$  as follows:

$$G = \{(ua, v) \mid a \in \Sigma, u, v \in X, (u\chi^{-1}, a, v\chi^{-1}) \in E, ua \notin X\}. \quad (1)$$

Using  $G$ , define a ternary relation  $H \subseteq X \times \Sigma \times X$ :

$$H = \{(u, a, v) \mid (ua, v) \in G\} \quad (2)$$

$$= \{(u, a, v) \mid a \in \Sigma, u, v \in X, (u\chi^{-1}, a, v\chi^{-1}) \in E, ua \notin X\}. \quad (3)$$

Using  $X$ , define a ternary relation  $K \subseteq X \times \Sigma \times X$ :

$$K = \{(u, a, ua) \mid a \in \Sigma, u, ua \in X\}. \quad (4)$$

Note that, if  $X$  is prefix-free, then  $K = \emptyset$ .

**Proposition 1.** *If  $a \in \Sigma$  and  $u, ua \in X$ , then there is exactly one edge in  $E$  leaving  $u\chi^{-1}$  and labeled  $a$ , namely,  $(u\chi^{-1}, a, (ua)\chi^{-1})$ .*

**Proof.** Suppose  $u, ua \in X$  and  $Iu = u\chi^{-1} = p$ ,  $I(ua) = (ua)\chi^{-1} = q$ , for  $p, q \in Q$ . Then, for some  $i \in I$ , there exists a path  $i \xrightarrow{ua} q$ , and hence a path  $i \xrightarrow{u} r$ , for some  $r \in Q$ , and an edge  $(r, a, q) \in E$ . But  $Iu = p$ , by assumption; hence we must have  $r = p$ . If there is another edge  $(r, a, s)$ , then  $\{q, s\} \subseteq Iua$ , contradicting that  $ua$  is a setting word.  $\square$

**Proposition 2.**  $K = K'$ , where

$$K' = \{(u, a, v) \mid a \in \Sigma, u, v \in X, (u\chi^{-1}, a, v\chi^{-1}) \in E, ua \in X\}. \quad (5)$$

**Proof.** If  $(u, a, v) \in K$ , then  $a \in \Sigma$ ,  $u \in X$ , and  $v = ua \in X$ . By Proposition 1, there is an edge  $(u\chi^{-1}, a, (ua)\chi^{-1})$ . Hence  $(u, a, v) \in K'$ . Conversely, suppose  $(u, a, v) \in K'$ . Then  $a \in \Sigma$ ,  $u, ua \in X$ , and  $(u, a, ua) \in K$ . By Proposition 1, there is exactly one edge leaving  $u\chi^{-1}$  and labeled  $a$ . By assumption,  $(u\chi^{-1}, a, v\chi^{-1})$  is such an edge. Hence  $v\chi^{-1} = (ua)\chi^{-1}$ ,  $v = ua$ , and  $(u, a, v) \in K$ .  $\square$

Let  $S\chi = (\Sigma, X, I', E')$  be the semiautomaton in which  $I' = \{i\chi \mid i \in I\}$  is the set of canonical words of the initial states and  $E' = H \cup K' = H \cup K = \{(u, a, v) \mid a \in \Sigma, u, v \in X, (u\chi^{-1}, a, v\chi^{-1}) \in E\}$ .

**Proposition 3.** *Semiautomata  $S$  and  $S\chi$  are isomorphic.*

<sup>1</sup> This notion was introduced by Janicki and Sekerinski [8] under the name “canonical trace property” for nondeterministic automata with one initial state.

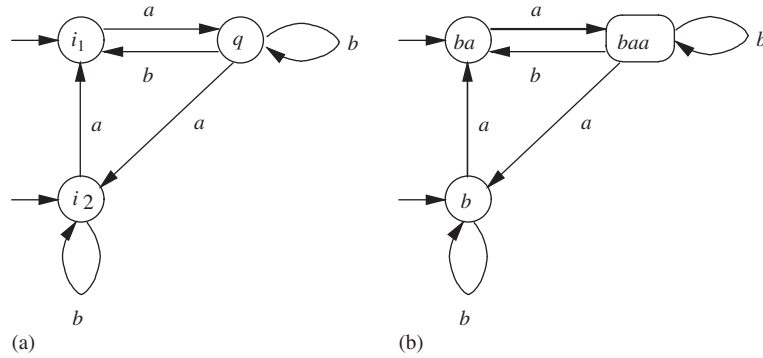


Fig. 1. Semiautomaton  $S_1$ .

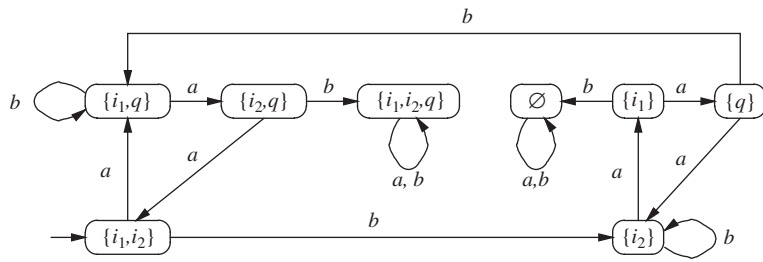


Fig. 2. Semiautomaton  $S_1\Delta$ .

**Proof.** The mapping  $\chi : Q \rightarrow X = Q\chi$  is bijective, and there is a one-to-one correspondence between the states in  $I$  and those in  $I'$ . Since there is a one-to-one correspondence between  $E$  and  $E'$ ,  $\chi$  is an isomorphism.  $\square$

**Example 1.** Semiautomaton  $S_1$  of Fig. 1(a) is settable; suppose  $i_2\chi = b$ ,  $i_1\chi = ba$ , and  $q\chi = baa$ . Relations  $G$ ,  $H$ , and  $K$  are:

$$G = \{(bb, b), (baaa, b), (baab, ba), (baab, baa)\},$$

$$H = \{(b, b, b), (baa, a, b), (baa, b, ba), (baa, b, baa)\},$$

$$K = \{(b, a, ba), (ba, a, baa)\}.$$

Semiautomaton  $S_1\chi$  is shown in Fig. 1(b), where the two edges in  $K$  are shown by thicker lines. The deterministic semiautomaton  $S_1\Delta$  obtained by the subset construction is shown in Fig. 2. We have  $I1 = \{i_1, i_2\}$  and  $C_1 = \{b, ba\}$ ;  $Ib = i_2$  and  $C_b = \{b\}$ ;  $Ibab = \emptyset$  and  $C_{bab} = \emptyset$ ;  $Iaaba = \{i_1, i_2, q\}$  and  $C_{aaba} = \{b, ba, baa\}$ ; etc. Also,  $baab \in L_q$ , but  $baab \notin R_q$ , since  $Ibaab = \{i_1, q\}$ .

#### 4. Prefix-rewriting systems

Let  $\Sigma$  be an alphabet (finite or infinite). Let  $R \subseteq \Sigma^* \times \Sigma^*$  be a binary relation on  $\Sigma^*$ . The pairs in  $R$  are called *rewriting rules* and  $R$  is a *prefix-rewriting system* [9]. Prefix-rewriting systems can also be viewed as ground-term-rewriting systems, and we use some terminology from [10]. Given any  $w, w' \in \Sigma^*$ , we say that  $w$  *rewrites* to  $w'$ , written  $w \vDash w'$ , if there is some  $(y, v) \in R$  such that  $w = yx$  and  $w' = vx$ . We say then that rule  $(y, v)$  *applies* to  $w$ .

The reflexive and transitive closure of  $\vDash$  is denoted by  $\vDash^*$ . Thus,  $w \vDash^* w'$  if and only if  $w = w_0 \vDash w_1 \vDash w_2 \vDash \dots \vDash w_n = w'$  for some  $n$ , and  $n$  is the length of this derivation of  $w'$  from  $w$ . In case  $w$  derives  $w'$  in  $n$  steps, we also write  $w \vDash^n w'$ ; note that  $w \vDash^0 w'$  if and only if  $w = w'$ .

A word  $w \in \Sigma^*$  is *irreducible* by  $R$  (or simply *irreducible*, if  $R$  is understood), if there is no  $w' \in \Sigma^*$ , such that  $w \vDash w'$ , that is, if no rule applies to  $w$ . System  $R$  is *right-reduced* if, for every pair  $(y, v)$  in  $R$ ,  $v$  is irreducible by  $R$ .

A prefix-rewriting system is *Noetherian* if there is no word  $w$  from which a derivation of infinite length exists.

If  $w = yx \in \Sigma^*$ , and  $(y, v) \in R$ , for some  $v \in \Sigma^*$ , we call  $x$  a *key suffix* of  $w$ .

The following result, proved in [10] for ground-term-rewriting systems, applies also to prefix-rewriting systems. For completeness, we provide a proof of this theorem modified to prefix-rewriting systems.

**Theorem 1.** *If  $R$  is right-reduced, then it is Noetherian.*

**Proof.** Suppose  $R$  is right-reduced. If  $w \vDash w'$ , then  $w = yx$  and  $w' = vx$ , for some  $y, v, x \in \Sigma^*$ , where  $(y, v) \in R$ . Suppose next that  $w' = vx = y'x' \vDash v'x' = w''$ . Since  $R$  is right-reduced, no rule applies to  $v$ ; hence  $v$  must be a proper prefix of  $y'$ , and the key suffix  $x'$  is shorter than the key suffix  $x$ . If  $x' = 1$ , the derivation stops. Since the key suffix decreases with each step,  $R$  must be Noetherian.  $\square$

Let  $L = \{y \mid (y, v) \in R\}$  be the set of all left-hand sides of the pairs in  $R$ .

**Proposition 4.** *Every word  $w \in \Sigma^*$  has at most one key suffix if and only if  $L$  is prefix-free.*

**Proof.** Suppose  $L$  is prefix-free, and  $w \in \Sigma^*$  has two key suffixes,  $x$  and  $x'$ , that is,  $w = yx = y'x'$ , where  $x \neq x'$ . Then either  $y$  is a prefix of  $y'$  or vice versa. This contradicts the fact that  $L$  is prefix-free. Hence  $w$  has at most one key suffix. Conversely, suppose every word has at most one key suffix and  $L$  is not prefix-free. Then there exist  $y$  and  $y'$  in  $L$  such that  $y' = yx$ , for some  $y, y' \in \Sigma^*$ ,  $x \in \Sigma^+$ . Then  $y'$  has key suffixes  $x$  and  $1$ , which is a contradiction. Thus,  $L$  must be prefix-free.  $\square$

Proposition 4 states that, if  $L$  is prefix-free and several rules apply to a word  $w$ , they all apply to the same prefix of  $w$ .

## 5. Prefix-rewriting in settable semiautomata

Our objective is to define a rewriting system that allows us to transform any word to any one of its canonical representatives. Let  $\mathcal{S} = (\Sigma, Q, I, E)$  be a settable semiautomaton, and let  $X$  be a set of canonical words for  $\mathcal{S}$ . We use the set  $G$  defined by (1) as a prefix rewriting system. Thus, if  $(y, v)$  is a pair in  $G$ , then  $yx \vDash vx$  for all  $x \in \Sigma^*$ .

**Proposition 5.** *If  $w \vDash^* w'$ , then  $Iw' \subseteq Iw$ .*

**Proof.** First, if  $(ua, v) \in G$ , then  $Iu = p$ ,  $Iv = q$ , for some  $p, q \in Q$ , and  $(p, a, q) \in E$ . Thus  $q \in Iua$ , and  $Iv \subseteq Iua$ . Second, if  $w = uax \vDash vx = w'$ , then  $(ua, v) \in G$ , and  $Iv \subseteq Iua$ . Consequently,  $Iw' = Ivx \subseteq Iuax = Iw$ . Finally, if  $w \vDash^0 w'$ , then  $w = w'$ , and the proposition holds trivially. Now suppose that  $w \vDash^n w'$  implies  $Iw' \subseteq Iw$ , and consider  $w''$  such that  $w' \vDash w''$ . Then  $Iw'' \subseteq Iw'$ , the induction step goes through, and the claim holds.  $\square$

The following result is a generalization of Lemma 3 of [3].

**Lemma 1.** *For  $w \in \Sigma^*$ , the following hold:*

- (i) *If no prefix of  $w$  is canonical, then  $w \vDash^* w'$  implies  $w' = w$ .*
- (ii) *If  $w$  has a canonical prefix and  $w \vDash^* w'$ , then  $w'$  has a canonical prefix.*
- (iii) *Let  $w'$  be any canonical representative of  $w$ . Then  $w \vDash^* w'$  if and only if  $w$  has a canonical prefix.*

**Proof.** Suppose no prefix of  $w$  is canonical. Then no rule applies to  $w$ , because all the rules are of the form  $(ua, v)$ , where  $u$  is canonical. Consequently,  $w$  can only derive itself, and it can do so, because  $\vDash^*$  is reflexive.

For the second claim, suppose  $w$  has a canonical prefix and  $w \models^* w'$ . If  $w \models^0 w'$ , then  $w = w'$ , and the claim holds. Assume now that  $w \models w'$ . Then  $w$  has the form  $w = uax$ , where  $u, x \in \Sigma^*$ ,  $a \in \Sigma$ ,  $u$  is canonical and  $ua$  is not. Also  $w' = vx$ , where  $v$  is canonical, and so  $w'$  has a canonical prefix. The claim now follows by transitivity.

For the third claim, suppose that  $w$  has a canonical prefix. We first show by induction on the length of  $w$  that  $w \models^* w'$  for all  $w' \in C_w$ .

If  $w = 1$ , then  $w$  can have only one canonical prefix, namely itself, and  $I$  is a singleton, say,  $I = \{i\}$ . Thus  $I1 = \{i\}1 = i$ , and  $1$  has only one canonical representative, namely, itself. Since  $1 \models^0 1$ , the claim holds for the basis case.

Now suppose that every word of length less than or equal to  $n$  that has a canonical prefix satisfies the claim. Consider  $w = ua$  with  $|u| = n$  and  $a \in \Sigma$ , where  $w$  has a canonical prefix. If  $w$  itself is canonical, then it has only one canonical representative, namely itself, and  $w \models^0 w$ . If  $w$  is not canonical, then  $u$  has a canonical prefix, and the induction assumption applies to  $u$ . Consider a canonical representative  $w' \in C_w$  of  $w$ . We want to show that  $w \models^* w'$ .

Since  $w'$  is a canonical representative of  $w = ua$ , there exist  $i, i' \in I$ ,  $p, q \in Q$ , paths  $i \xrightarrow{u} p$ ,  $i' \xrightarrow{w'} q$ , and edge  $(p, a, q)$ , such that  $Iw' = q$  and  $q \in Iw$ . By the induction assumption,  $u$  derives every one of its canonical representatives. In particular,  $u \models^* u'$  where  $Iu' = p$ . Then also  $ua \models^* u'a$ . If  $u'a$  is canonical, then  $u'a = w'$ ,  $w = ua \models^* u'a = w'$ , and we are done. Otherwise, since  $(p, a, q)$  is an edge of  $\mathcal{S}$ , there is a rule  $(u'a, w')$  in  $G$ , and  $w = ua \models^* u'a \models w'$ , as required. Thus, the induction step goes through, showing that every word having a canonical prefix derives all of its canonical representatives.

Conversely, if  $w$  does not have a canonical prefix, then it is not canonical, and can only derive itself. Hence  $w$  cannot derive any canonical word.  $\square$

**Example 2.** For the semiautomaton  $\mathcal{S}_1$  of Fig. 1(a), suppose the canonical words are  $i_1\chi = baaaa$ ,  $i_2\chi = b$ , and  $q\chi = baa$ . Then  $K = \emptyset$  and  $G = \{(ba, baaaa)_1, (bb, b)_2, (baaa, b)_3, (baab, baa)_4, (baab, baaaa)_5, (baaaaa, baa)_6\}$ , where the pairs of  $G$  are numbered by subscripts for convenience.

Since  $Ibaabab = Q$ , the word  $baabab$  has three canonical representatives, derived as follows:  $baabab \stackrel{4}{\models} baaab \stackrel{3}{\models} bb \stackrel{2}{\models} b$ ,  $baabab \stackrel{5}{\models} baaaaab \stackrel{3}{\models} baab \stackrel{4}{\models} baa$ , and  $baabab \stackrel{5}{\models} baaaaab \stackrel{3}{\models} baab \stackrel{5}{\models} baaaa$ . Repeated use of Rule 1 leads to an infinite derivation. Hence this system is not Noetherian. Note also that canonical words may be reducible. For example,  $baa \stackrel{1}{\models} baaaa \stackrel{6}{\models} baa$ .

## 6. Prefix-continuous canonical sets

If a semiautomaton has a prefix-continuous canonical set, the rewriting system  $G$  is better behaved, as we shall see, but not all semiautomata have such sets. For example, in the settable semiautomaton  $\mathcal{S}_2$  of Fig. 3, the canonical word of state  $i$  must be 1, and the canonical words of states  $p$  and  $q$  must be of length at least 2. Hence, there is no prefix-continuous canonical set.

The following result is Lemma 4 of [3].

**Lemma 2.** *If  $X$  is prefix-continuous, then  $L$  is prefix-free. If  $X$  (and therefore also the semiautomaton) is finite, the converse also holds.*

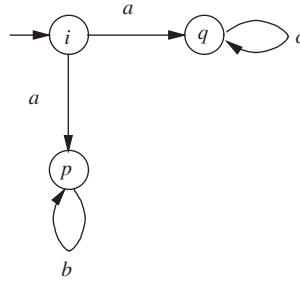
It is shown in [3] that the converse of Lemma 2 may not hold if  $X$  is infinite.

From Lemma 2 and Proposition 4 we have:

**Corollary 1.** *If  $X$  is prefix-continuous, then every word  $w \in \Sigma^*$  has at most one key suffix.*

**Definition 1.** Given a set  $X$  of canonical words, define the following subsets:

- $W = \Sigma^* \setminus X\Sigma^*$  is the set of *acannotical* words.
- $X_0 = X \setminus X\Sigma^+$  is the set of *minimal canonical* words.
- $Y = X_0\Sigma^+$  is the set of *post-canonical* words.

Fig. 3. Semiautomaton  $S_2$ .

Note that  $(W, X_0, Y)$  is a partition of  $\Sigma^*$ .

The following result is implied by Lemma 6 of [3].

**Lemma 3.** *If  $X$  is prefix-continuous and  $w \in X$ , then  $w$  is irreducible by  $G$ .*

The following result is a generalization of Theorem 4 of [3].

**Theorem 2.** *The rewriting system  $G$  is Noetherian if and only if the set  $X$  of canonical words is prefix-continuous.*

**Proof.** If  $X$  is prefix-continuous, and  $w \in X$ , then  $w$  is irreducible by  $G$ , by Lemma 3. Since the right member of every pair in  $G$  is in  $X$ ,  $G$  is right-reduced, and therefore Noetherian, by Theorem 1. Conversely, suppose that  $X$  is not prefix-continuous. Then there exists  $w = uax \in X$  such that  $u \in X$ , but  $ua \notin X$ . Since  $w$  is canonical, there exists some  $i \in I$  and a path  $i \xrightarrow{w} q$ , where  $q\chi = w$ . Since  $w = uax$ , this path consists of path  $i \xrightarrow{u} r$ , where  $r\chi = u$ , edge  $(r, a, p)$ , for some  $p \in Q$  and a path  $p \xrightarrow{x} q$ . Since  $ua$  is not canonical and  $p \in Iua$ , there is a canonical word  $v$  such that  $Iv = p$ , and  $(ua, v)$  is a rule in  $G$ . Thus  $w = uax \vDash vx$ . Since  $q \in px$ , we also have  $q \in Ivx$ . Thus  $Ivx \neq \emptyset$ . By Proposition 5,  $Ivx \subseteq Iuax = Iw$ . Since  $Ivx \subseteq Iw$ , and  $Iw = q$ , then also  $Ivx = q$ , and  $w$  is the canonical representative of  $vx$ . By Lemma 1 (3),  $vx$  derives all of its canonical representatives. Hence  $vx \vDash^* w$ . Altogether,  $w \vDash vx \vDash^* w$ , and we have an infinite derivation.  $\square$

**Proposition 6.** *If  $X$  is prefix-continuous,  $(\Sigma^*, \vDash^*)$  is a partially ordered set.*

**Proof.** By definition,  $\vDash^*$  is reflexive and transitive. If  $w \vDash^* w'$ ,  $w' \vDash^* w$ , and  $w \neq w'$ , then  $G$  is not Noetherian, contradicting Theorem 2. Hence  $\vDash^*$  is antisymmetric, and hence a partial order.  $\square$

---

**Algorithm 1.** DERIVE ( $w \in X_0\Sigma^*$ )

---

```

1:  $D \leftarrow \{w\}$ 
2:  $u \leftarrow$  longest canonical prefix of  $w$ 
3: if  $u \neq w$  then
4:    $\{w$  has the form  $uax$  where  $a \in \Sigma, x \in \Sigma^*\}$ 
5:    $p \leftarrow Iu$ 
6:   for all  $q \in Q$  such that  $(p, a, q) \in E$  do
7:      $v \leftarrow q\chi$ 
8:      $D \leftarrow D \cup \text{DERIVE}(vx)$ 
9:   end for
10: end if
11: return  $D$ 
  
```

---



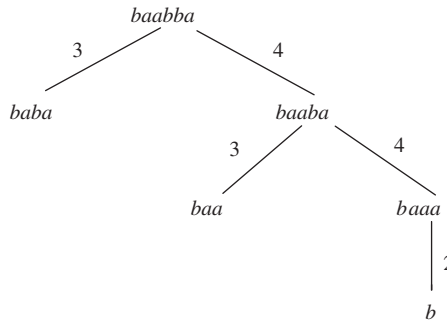


Fig. 4. DERIVE(*baabba*).

We use the convention that  $w'$  is “below”  $w$ , if  $w \vDash^* w'$ . In the partially ordered set  $(\Sigma^*, \vDash^*)$  the irreducible words are minimal. By Lemma 1(1), all acanonical words are irreducible. In the prefix-continuous case, all the words that can be derived from a word that is not acanonical can be found using Algorithm DERIVE.

**Example 3.** Return to the semiautomaton of Example 1(a) with  $i_2\chi = b$ ,  $i_1\chi = ba$ , and  $q\chi = baa$ . The set  $\{b, ba, baa\}$  is prefix-continuous. The rewriting rules are  $(bb, b)_1$ ,  $(baaa, b)_2$ ,  $(baab, ba)_3$ ,  $(baab, baa)_4$ . The set of acanonical words is  $1 + a\Sigma^*$ , word  $b$  is the only minimal canonical word, and the set of post-canonical words is  $b\Sigma^+$ . We now evaluate DERIVE(*baabba*). The longest canonical prefix of *baabba* is *baa*, and  $p = q$ . There are two edges:  $(q, b, i_1)$  and  $(q, b, q)$ . We use  $(q, b, i_1)$  first, that is, apply Rule 3; then  $v = i_1\chi$ ,  $baabba \vDash baba$ , and  $vx = baba$  is irreducible, since there are no edges from  $(ba)\chi^{-1} = i_1$  labeled  $b$ . Thus DERIVE(*baba*) = {*baba*}, and  $D = \{baabba, baba\}$ . Use  $(q, b, q)$  next, that is, apply Rule 4; then  $v = q\chi$ ,  $baabba \vDash baaba$ ,  $vx = baaba$  and  $D = \{baabba, baba\} \cup \text{DERIVE}(baaba)$ . To find DERIVE(*baaba*), Rules 3 and 4 are applicable, yielding  $baaba \vDash baa$ , where *baa* is irreducible, and  $baaba \vDash baaa$ , which leads to  $baaa \vDash b$ , by Rule 2. Altogether, DERIVE(*baabba*) = {*baabba, baba, baaba, baa, baaa, b*}. The derivations are shown in Fig. 4. The irreducible words are the two canonical words  $b$  and *baa*, and word *baba* which is not in the language of the semiautomaton.

### 7. Rewriting systems for all words

As in [3], we want to derive the canonical representatives of acanonical words. For this we define the following acanonical rewriting rules:

$$A = \{(1, i\chi) \mid i \in I\}.$$

These rules are used differently than the rules of  $G$ . A rule of  $A$  is used as a pre-processing step for an acanonical word  $w$ . By applying such a rule, we rewrite  $w$  as  $i\chi w$ , and now the new word  $i\chi w$  is not acanonical. We then use only the prefix-rewriting rules of  $G$  to transform  $i\chi w$  to any one of its canonical representatives, thus obtaining all canonical representatives of  $w$ .

Let the rewriting system  $\hat{G}$  be defined as  $\hat{G} = G \cup A$ , with the restriction that an acanonical rule can be applied only once to an acanonical word, and then the rules of  $G$  are used. In this section,  $w \vDash^* w'$  means that  $w'$  is derivable from  $w$  in the rewriting system  $\hat{G}$ . The next theorem summarizes the properties of  $\hat{G}$ ; these claims are easily verified.

**Theorem 3.** Let  $S$  be a settable semiautomaton with canonical set  $X$ . Then

- (i) Every word derives in  $\hat{G}$  all of its canonical representatives.
- (ii)  $\hat{G}$  is Noetherian if and only if  $X$  is prefix-continuous.
- (iii) The acanonical words are maximal in the partial order  $(\Sigma^*, \vDash^*)$ .



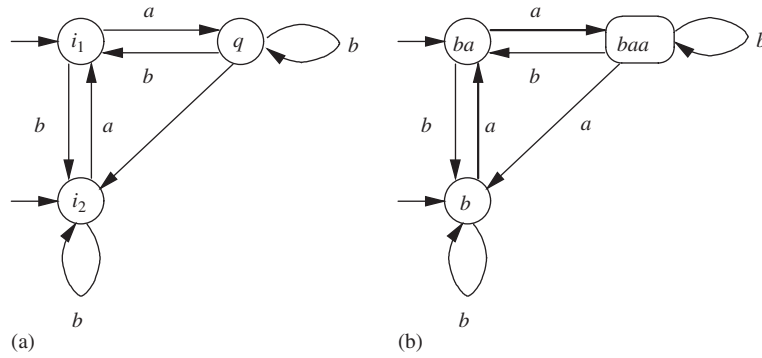


Fig. 5. Semiautomaton  $\mathcal{S}_3$ .

**Example 4.** For the canonical words of Example 3, the set of acanonical rules is  $A = \{(1, b), (1, ba)\}$ . To derive the canonical representatives of the acanonical word 1, it suffices to use the two acanonical rules  $1 \vDash b$  and  $1 \vDash ba$ . Similarly, for  $a$ , we have  $a \vDash ba$  and  $a \vDash baa$ . For  $aa$  we have  $aa \vDash baa$ , where  $baa$  is canonical, and  $aa \vDash baaa$ . In the second case, we then use Rule 2 of  $G$  to obtain  $baaa \vDash b$ , thus finding the second canonical representative of  $aa$ . In the case of  $ab$ , we have  $ab \vDash bab$ , from which no further derivation is possible; note that  $bab$  is not in the language of the semiautomaton  $\mathcal{S}_1$ . We also have  $ab \vDash baab$ , and we can then derive the two canonical representatives of  $ab$  by using the rules  $baab \vDash ba$  and  $baab \vDash baa$ .

**8. Complete semiautomata**

In the case of complete semiautomata, we have the following result:

**Theorem 4.** Let  $\mathcal{S} = (\Sigma, Q, I, E)$  be a complete settable semiautomaton with  $X$  as the set of canonical words. If  $X$  is prefix-continuous, a word is irreducible in  $\hat{G}$  if and only if it is canonical.

**Proof.** By Lemma 3, if  $X$  is prefix-continuous and  $w$  is canonical, then it is irreducible. Conversely, if  $\mathcal{S}$  is complete, then  $Iw \neq \emptyset$  for every word  $w$ , and every  $w$  has at least one canonical representative. If  $w$  is acanonical, it is reducible by its acanonical rules. If  $w$  is post-canonical but not minimal canonical, then it derives its canonical representatives by Lemma 1(3), and is reducible. Thus the only irreducible words are the minimal canonical words, which are obviously canonical.

**Example 5.** The semiautomaton of Fig. 5 is complete. Suppose  $i_2\chi = b$ ,  $i_1\chi = ba$ , and  $q\chi = baa$ . The set  $\{b, ba, baa\}$  is prefix-continuous. The rewriting rules are  $G = \{(bb, b)_1, (bab, b)_2, (baaa, b)_3, (baab, ba)_4, (baab, baa)_5\}$  and  $A = \{(1, b)_6, (1, ba)_7\}$ . The set of acanonical words is  $1 + a\Sigma^*$ , word  $b$  is the only minimal canonical word, and the set of post-canonical words is  $b\Sigma^+$ . The derivations of the canonical words from  $w = baabba$  are: (1)  $baabba \xrightarrow{4} baba \xrightarrow{2} ba$ , (2)  $baabba \xrightarrow{5} baaba \xrightarrow{4} baa$ , and (3)  $baabba \xrightarrow{5} baaba \xrightarrow{5} baaa \xrightarrow{3} b$ .

**9. Examples of nondeterministic modules**

A trace-assertion specification [3] of a complete deterministic semiautomaton  $\mathcal{S} = (\Sigma, Q, i, E)$  consists of a set  $X \subseteq \Sigma^*$  of canonical words, an initial canonical word  $x_0 \in X$ , and a relation  $\hat{G} \subseteq \Sigma^* \times \Sigma^*$ , which permits us to reconstruct the edges of the semiautomaton, and also defines a prefix-rewriting system allowing us to rewrite any word

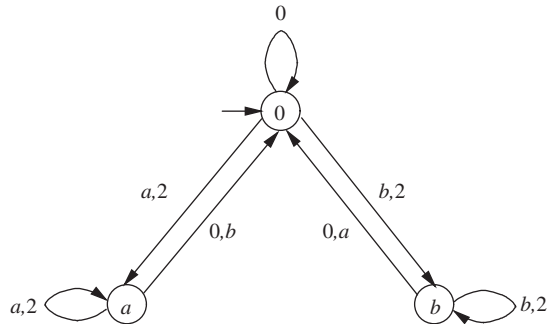


Fig. 6. Simple arbiter.

as its canonical representative. In the deterministic case, a word  $y$  can appear as the left-hand side of a pair  $(y, v)$  in  $\hat{G}$  at most once. The smallest right congruence containing  $\hat{G}$  is precisely the state-equivalence relation  $\equiv$ , where  $w \equiv w'$  if and only if  $iw = iw'$ .

For of a nondeterministic settable semiautomaton, we have a set  $X \subseteq \Sigma^*$  of canonical words, a set  $X_0 \subseteq X$  of initial canonical words, and a relation  $\hat{G} \subseteq \Sigma^* \times \Sigma^*$ . The smallest right congruence containing  $\hat{G}$  is no longer an equivalence relation, but it is a *compatibility*, meaning that it is reflexive and symmetric. In general,  $\hat{G}$  allows us to derive from any word all of its canonical representatives. Moreover, if  $X$  is prefix-continuous, then the rewriting system has no infinite derivations.

### 9.1. A primitive arbiter

The semiautomaton of a primitive arbiter [4] is shown in Fig. 6. The input alphabet is  $\Sigma = \{0, a, b, 2\}$ . If the input is 0, there are no requests. If the input is  $a$  (respectively  $b$ ), user  $a$  (respectively  $b$ ) is requesting service, whereas both users are asking for service when the input is 2. In state 0 no requests are being served, whereas in state  $a$  (respectively  $b$ ), user  $a$  (respectively  $b$ ) is being served. If there are two requests in state 0, either user  $a$  or user  $b$  is selected nondeterministically. If  $a$  is picked, then  $a$  continues to be served if the request continues, or if both users ask for service. If there are no requests in state  $a$ , the arbiter returns to state 0. If user  $a$  now removes its request and user  $b$  puts in a request at the same time, the arbiter first resets to state 0, and then offers service to user  $b$ . The transitions from state  $b$  are symmetric.

The arbiter semiautomaton is settable and complete. Suppose  $0\chi = 1$ ,  $a\chi = a$ , and  $b\chi = b$ ; this is a prefix-closed set, and there are no acanonical rules. Here,  $X = \{1, a, b\}$ ,  $X_0 = \{1\}$ , and  $G = \{(0, 1), (2, a), (2, b), (a0, 1), (aa, a), (ab, 1), (a2, a), (b0, 1), (ba, 1), (bb, b), (b2, b)\}$ . Word  $02a20a$  has the following derivations: (1)  $02a20a \vDash 2a20a \vDash aa20a \vDash a20a \vDash a0a \vDash a$ , (2)  $02a20a \vDash 2a20a \vDash ba20a \vDash 20a \vDash a0a \vDash a$ , and (3)  $02a20a \vDash 2a20a \vDash ba20a \vDash 20a \vDash b0a \vDash a$ .

### 9.2. An urn

An urn, called “unique integer module” in [8], contains two balls labeled 1 and 2. The operation  $g$  (get) randomly selects one of the balls and removes it from the urn. The second get operation removes the second ball.

The semiautomaton of the urn is shown in Fig. 7, where, for now, all the edge labels are considered to be  $g$ . This semiautomaton is settable to state  $\{1, 2\}$  by 1, and to state  $\emptyset$ , by  $gg$ . However, it is not settable to state  $\{1\}$  or  $\{2\}$ , and our theory is not applicable.

If we consider the semiautomaton alphabet to be the set of pairs (input,output), that is, pairs  $(g, j)$ , where  $j \in \{1, 2\}$ , then the resulting semiautomaton is deterministic, and our theory applies. Let  $(g, j)$  be represented by  $g_j$ , for  $j = 1, 2$ . Then we can use the canonical set  $\{1, g_1, g_2, g_1g_2\}$ , initial canonical set  $\{1\}$ ,  $G = \{(g_2g_1, g_1g_2)\}$ , and  $K = \{(1, g_1, g_1), (1, g_2, g_2), (g_1, g_2, g_1g_2)\}$ .

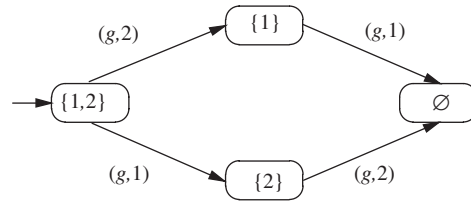


Fig. 7. An urn.

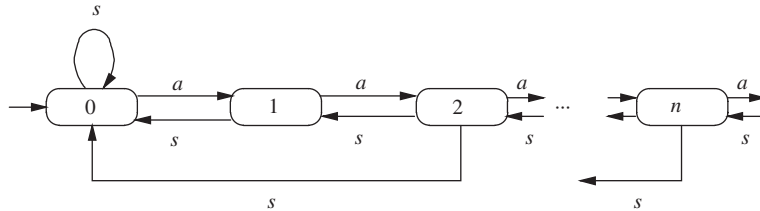


Fig. 8. A drunk counter.

### 9.3. A drunk counter

This example is a simplified version of the “drunk stack” module described in [8]; see also [3]. The counter is initially 0. It has two operations:  $a$  (add), which adds 1 to the present count, and  $s$  (subtract), which, if the count is  $\geq 2$ , nondeterministically subtracts either 1 or 2 from the present count. If the present count is 1, then  $s$  subtracts 1, and if the count is 0,  $s$  does not change the count.

The counter semiautomaton of Fig. 8 is complete and settable. An obvious set of canonical words for this counter is  $X = \{1, a, aa, aaa, \dots\}$ , with  $X_0 = \{1\}$ . Here  $X$  is prefix closed, and there are no acanonical words. Relation  $G$  is infinite of course, but it is finitely representable as follows:  $G = \{(s, 1), (as, 1)\} \cup \{(a^n s, a^{n-1}), (a^n s, a^{n-2}) \mid n \geq 2\}$ .

The counter modeled on the “very drunk stack” of [8] would have an add operation which would nondeterministically choose to add either 1 or 2 to the counter contents. One verifies that this semiautomaton is not settable.

### Acknowledgment

I am very grateful to Elad Lahav for suggesting several important improvements to this paper.

### References

- [1] W. Bartussek, D. Parnas, Using assertions about traces to write abstract specifications for software modules, Report TR77-012, University of North Carolina, Chapel Hill, (1977), Also in: D.M. Hoffman, D.M. Weiss (Eds.), *Software Fundamentals*, Addison-Wesley, 2001, pp. 9–28.
- [2] J.A. Brzozowski, H. Jürgensen, Theory of deterministic trace-assertion specifications, Technical Report CS-2004-30, School of Computer Science, University of Waterloo, 2004, (<http://www.cs.uwaterloo.ca/cs-archive/CS-2004/CS-2004.shtml>).
- [3] J.A. Brzozowski, H. Jürgensen, Representation of semiautomata by canonical words and equivalences, *Internat. J. Found. Comput. Sci.* 16 (5) (2005) 831–850.
- [4] J.A. Brzozowski, H. Zhang, Delay-insensitivity and semi-modularity, *Formal Methods System Design* 16 (2000) 191–218.
- [5] S. Eilenberg, *Automata, Languages, and Machines*, Academic Press, New York, 1974.
- [6] A. Ginzburg, *Algebraic Theory of Automata*, Academic Press, New York, 1968.
- [7] M. Iglewski, M. Kubica, J. Madey, Trace specifications of nondeterministic multi-object modules, Technical Report TR 95-05 (205), Institute of Informatics, Warsaw University, Poland, 1995.

- [8] R. Janicki, E. Sekerinski, Foundations of the trace assertion method of module interface specifications, *IEEE Trans. Software Eng.* 27 (7) (2001) 577–598.
- [9] N. Kuhn, K. Madlener, A method for enumerating cosets of a group presented by a canonical system, in: *Proc. ACM-SIGSAM Internat. Symp. on Symbolic and Algebraic Computation*, 1989, pp. 338–350.
- [10] W. Snyder, Efficient ground completion: an  $O(n \log n)$  algorithm for generating reduced sets of ground rewrite rules equivalent to a set of ground equations  $E$ , in: N. Dershowitz, (Ed.), *Rewriting Techniques and Applications, Proc. RTA-89, Lecture Notes in Computer Science*, Vol. 355, 1989, pp. 419–433.