INFORMATIQUE THÉORIQUE ET APPLICATIONS

B. LE SAËC
I. LITOVSKY

Rebootable and suffix-closed ω-power languages

Informatique théorique et applications, tome 26, nº 1 (1992), p. 45-58.
<http://www.numdam.org/item?id=ITA_1992__26_1_45_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 26, n° 1, 1992, p. 45 à 58)

REBOOTABLE AND SUFFIX-CLOSED ω-POWER LANGUAGES (*)

by B. LE SAËC ([1]) and I. LITOVSKY ([1])

Communicated by J. BERSTEL

Abstract. – *The* ω*-languages* R^ω *such that* (1) $\mathrm{Pref}(R^\omega) R^\omega = R^\omega$, (2) $\mathrm{Suf}(R^\omega) = R^\omega$ *or* (3) $\mathrm{Pref}(R^\omega)\ \mathrm{Suf}(R^\omega) = R^\omega$ *are characterized via properties of the language* $\mathrm{Stab}(R^\omega) = \{u \in \Sigma^* : u R^\omega \subset R^\omega\}$ *and via properties of* ω*-generators of* R^ω. *Nicely, each characterization for* (1) *provides one for* (2) *and* (3) *by replacing "prefix" by "suffix" and "factor", respectively. Moreover* (3) *characterizes the* ω*-languages* R^ω *which are left* ω*-ideals in* $\mathrm{Alph}(R^\omega)$.

Résumé. – *Les* ω*-languages* R^ω *tels que* (1) $\mathrm{Pref}(R^\omega) R^\omega = R^\omega$, (2) $\mathrm{Suf}(R^\omega) = R^\omega$ *on* (3) $\mathrm{Pref}(R^\omega)\ \mathrm{Suf}(R^\omega) = R^\omega$ *sont caractérisés au moyen de propriétés du langage* $\mathrm{Stab}(R^\omega) = \{u \in \Sigma^* : u R^\omega \subset R^\omega\}$ *et au moyen de propriétés des* ω*-générateurs de* R^ω. *Toute caractérisation pour* (1) *fournit une caractérisation pour* (2) *et* (3) *en remplaçant « préfixe » pour « suffixe » on « facteur », selon les cas. De plus* (3) *caractérise les* ω*-langages* R^ω *qui sont des* ω*-idéaux à gauche de* $\mathrm{Alph}(R^\omega)$.

0. INTRODUCTION

In this paper, we study properties of ω-languages over a finite alphabet Σ. An intuitive motivation may be found in regarding ω-languages as infinite behaviours of process (*cf.* [2]). In this way, Σ is a set of actions. Moreover the processes are assumed to be controlled by a manager while the users can only observe the sequences of actions. We shall use this interpretation in the sequel.

First we study the behaviour of a process when an interruption arises: could the manager restart the process without "disturbing" the users, that is, without asking the users to forget the sequence already seen? Hence the manager is interested in the *rebooting points*, that is, the points where the process may be restarted as if it was in the initial state, but without cancelling

(*) Received June 1990, revised September 1990.
([1]) Laboratoire Bordelais de Recherche en Informatique, U.R.A. 1304 du C.N.R.S., E.N.S.E.R.B., 351, cours de la Libération, 33405 Talence Cedex, France.

0988-3754/92/01 45 14/$3.40/© AFCET-Gauthier-Villars

the action sequence already performed. In other words, given the ω-language L of acceptable behaviours of P, we find the prefixes x of L such that the ω-language $x L$ is contained in L. That leads us to consider the greatest language X such that $XL = L$. In particular, languages or ω-languages L such that $\mathrm{Pref}(L)L = L$ where $\mathrm{Pref}(L)$ is the set of all prefixes of L are very convenient for the manager. Such languages or ω-languages L are said to be *rebootable*.

Next, we consider the following situation: a process P is active and a new user arrives. Then the manager has to find the *access points*, that is, the points x such that the end of any acceptable behaviour beginning with x remains in L. In other words we are interested in the greatest language X included in $\mathrm{Pref}(L)$ such that $X^{-1}L = L$. So the *accessible* ω-languages are convenient for the manager: they are defined by $\mathrm{Pref}(L)^{-1}L = L$. They are called the *suffix-closed* ω-languages [7].

Finally, we consider the ω-languages having both features, being rebootable and suffix-closed. They are characterized by the following property: one can substitute any prefix of L for any other one without changing the membership to L. Such ω-langages may be called *prefix-switchable*. This notion is an extension of the one of *absolutely closed* ω-languages [7] where the condition $\mathrm{Pref}(L) = \Sigma^*$ is added.

In this paper, the results concern mainly ω-power languages L, that is, ω-languages of the form R^ω for some language R. Counterexamples show that these results do not hold without assuming that L is an ω-power language. The different charactarizations for the ω-languages R^ω are only based on properties of languages. In this way, the stabilizer $\mathrm{Stab}(R^\omega)$ of R^ω introduced in [14] as the set $\{u \in \Sigma^* : u R^\omega \subset \mathrm{R}^\omega\}$ works well. Indeed each property of R^ω is characterized bu a corresponding property of $\mathrm{Stab}(R^\omega)$. So the characterizations state:

R^ω is rebootable iff $\mathrm{Stab}(R^\omega)$ is prefix-closed;

R^ω is suffix-closed iff $\mathrm{Stab}(R^\omega)$ is suffix-closed;

R^ω is a left ω-ideal iff $\mathrm{Stab}(R^\omega)$ is factor-closed.

Furthermore, we note that when an ω-language L is not an ω-power language, the stabilizer of L gives no longer reliable characterizations. On the other hand, by considering only regular ω-languages R^ω (and even deterministic regular ω-languages R^ω for the first characterization below), we link properties of R^ω with properties of ω-generators of R^ω in the following way:

R^ω is rebootable iff $R^\omega = G^\omega$ for some language G such that $\mathrm{Pref}(G)G = G$;

R^ω is suffix-closed iff $R^\omega = G^\omega$ for some language G such that $\mathrm{Suf}(G)G = G$;

R^ω is a left ω-ideal iff $R^\omega = G^\omega$ for some language G such that $\mathrm{Fact}(G)G = G$;

or equivalently iff $R^\omega = G^\omega$ for some ideal G.

In the non-regular case, we do not yet have results.

The paper is organized as follows. After recalling definitions and notation (Part 1), we study the rebootable ω-languages (Part 2), next we study the suffix-closed ω-languages (Part 3). In Part 4, left ω-ideals are investigated first as rebootable and suffix-closed ω-languages, then using finitary ideals, and finally via their syntactic monoids.

1. PRELIMINARIES

Let Σ be an alphabet. Σ^* and Σ^ω are the sets of all finite words and of all ω-words over Σ, respectively. Let L be a subset of a set S. The complement of L is denoted by ${}^cL$. The union set $\Sigma^* \cup \Sigma^\omega$ is denoted by Σ^∞. The empty word is denoted by ε and the language $\Sigma^* \setminus \{\varepsilon\}$ is denoted by Σ^+. Subsets of Σ^*, Σ^ω and Σ^∞ are called languages, ω-languages and ∞-languages, respectively. The set of letters which occur in an ∞-language L is denoted by Alph(L). Let u, v be two words $\in \Sigma^\infty$. As usual uv denotes the concatenation of u and v. Let X be a language, and let Y be an ∞-language. XY denotes the set $\{uv \in \Sigma^\infty : u \in X \text{ and } v \in Y\}$ and $X^{-1}Y$ denotes the set $\{w \in \Sigma^\infty : uw \in Y \text{ for some } u \in X\}$. Let L be an ω-language. $UP(L)$ denotes the ω-language of all ultimately periodic ω-words of L, that is, $UP(L) = \{w \in L : w = uv^\omega \text{ for some } u, v \text{ in } \Sigma^+\}$.

Let $u \in \Sigma^\infty$ and $X \subseteq \Sigma^\infty$. A word v is a prefix of u if $u \in v\,\Sigma^\infty$. Let Pref(u) denote the set of all prefixes of u, and let $\text{Pref}(X) = \bigcup_{u \in X} \text{Pref}(u)$. An ∞-word v is a suffix of u if $u \in \Sigma^* v$. Let Suf(u) denote the set of all suffixes of u, and let $\text{Suf}(X) = \bigcup_{u \in X} \text{Suf}(u)$. The language Fact$(X)$ of the factors of X is the language Pref(Suf(X)). X is said to be prefix-closed, suffix-closed or factor-closed if Pref$(X) = X$, Suf$(X) = X$ or Fact$(X) = X$, respectively.

Let $R \subseteq \Sigma^*$. The language X is a left-ideal, a right-ideal or an ideal in R if $RX \subseteq X$, $XR \subseteq X$ or $RXR \subseteq X$, respectively. R is a prefix-free language (or prefix code) if $R\Sigma^+ \cap R = \varnothing$. R is a semaphore code if $R = \Sigma^* S \setminus \Sigma^* S \Sigma^+$ for some nonempty set $S \subseteq \Sigma^+$ [3]. R is an ifl-code if every ω-word has at most one factorization over R [16].

The adherence Adh(R) of R is the ω-language $\{w \in \Sigma^\omega : \text{Pref}(w) \subseteq \text{Pref}(L)\}$ [10, 4]. Recall that every adherence is a closed set for the usual topology in Σ^ω. The limit Lim(R) of R is the ω-language $\{w \in \Sigma^\omega : \text{Pref}(w) \cap R \text{ is infinite}\}$.

For every language $R \subseteq \Sigma^+$, the ω-power R^ω of R is defined by $R^\omega = \{u_1 \ldots u_n \ldots : u_n \in R \text{ for each } n\}$. An ω-generator of R^ω is a language $G \subseteq \Sigma^+$ such that $G^\omega = R^\omega$. An ω-generator G of R^ω is said to be minimal if no proper subset of G is an ω-generator of R^ω. The stabilizer $\mathrm{Stab}(L)$ of an ω-language L is the language $\{u \in \Sigma^* : uL \subseteq L\}$ [14]. Clearly the language $\mathrm{Stab}(L)$ is a submonoid of Σ^*.

A finite automaton over Σ is a quintuple $\mathcal{A} = (\Sigma, Q, \delta, S, F)$ where Q is the (finite) set of states, $S \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of accepting states, and δ is the next state relation, that is, a function from $Q \times \Sigma$ into 2^Q. The automaton $\mathcal{A}$ is said to be deterministic if S is a singleton and δ is a function from $Q \times \Sigma$ into Q. A run of $\mathcal{A}$ on an ω-word $w_1 \ldots w_n \ldots$ is an ω-word $q_0 \ldots q_n \ldots$ in Q^ω such that $q_0 \in S$ and for each n, $q_{n+1} \in \delta(q_n, w_n)$. For any run r, let $\mathrm{Inf}(r)$ be the set $\{q \in Q : q = q_n$ for infinitely many $n\}$. An ω-word w is said to be recognized by $\mathcal{A}$ if $\mathrm{Inf}(r) \cap F \neq \emptyset$ for some run r of $\mathcal{A}$ on w [5]. The ω-language Büchi-recognized by $\mathcal{A}$ is the set of all ω-words recognized by $\mathcal{A}$. Such ω-languages are said to be regular. Recall that the deterministic automata are less powerful than the nondeterministic ones for this recognizing mode. Every ω-language recognized by some deterministic automaton is called a deterministic ω-language. An ω-language is deterministic iff it is the limit of some language [8].

Let L be any ω-language. We use the syntactic congruence of L in Σ^* defined in [1] by $u \approx u'$ iff for every v, w_1, w_2 in Σ^*, we have (1) $w_1 u w_2 v^\omega \in L$ iff $w_1 u' w_2 v^\omega \in L$ and (2) $v(uw_2)^\omega \in L$ iff $v(u' w_2)^\omega \in L$. The set $\mathcal{SM}(L)$ of $\approx$-classes is a monoid, called the syntactic monoid of L, which is finite if L is regular [1]. We denote by π the morphism which associates each word with its $\approx$-class. Note that this notion of syntactic monoid for ω-languages is different from the one considered in [7].

2. REBOOTING

Let L be an ω-language. The language $\mathrm{Stab}(L)$ is the greatest solution of the equation $XL = L$ since $\mathrm{Stab}(L) = \{u \in \Sigma^* : uL \subseteq L\}$. In this part, the goal is to characterize the ω-languages such that $\mathrm{Pref}(L)$ is the greatest solution of this equation, that is, such that $\mathrm{Stab}(L) = \mathrm{Pref}(L)$.

DEFINITION 2.0: Let $Y \subseteq \Sigma^\infty$. Y is said to be rebootable if $\mathrm{Pref}(Y)\,Y = Y$.

If L is regular, then $\mathrm{Stab}(L)$ is a regular and constructible language. That is, given an automaton which recognizes L, one can construct an automaton recognizing $\mathrm{Stab}(L)$ [12]. Hence, one can decide whether L is rebootable.

From now on, we consider only ω-power languages. We try to characterize those ω-power languages R^ω which are rebootable via properties of the stabilizer of R^ω and via properties of ω-generators of R^ω. We need the following lemmas.

LEMMA 2.1: *Let $R \subseteq \Sigma^+$ and let $L \subseteq \Sigma^\infty$. Then $L \subseteq RL$ implies $L \subseteq R^\omega$.*

Proof: Let $w \in L$. Then $w = r_1 w_1$ for some $r_1 \in R$ and $w_1 \in L$. In this way, one can constuct a sequence of words $r_i \in R$ such that $r_1 \ldots r_i w_i = w$ for every i. Hence $\mathrm{Pref}(w) = \mathrm{Pref}(r_1 \ldots r_i \ldots)$, that is $w = r_1 \ldots r_i \ldots$ ■

LEMMA 2.2.: *Let R^ω be an ω-power language, and let G be any ω-generator of R^ω. Then the language $G \setminus G(\mathrm{Stab}(R^\omega) \setminus \{\varepsilon\})$ is also an ω-generator of R^ω.*

Proof: Let us denote G' the language $G \setminus G(\mathrm{Stab}(R^\omega) \setminus \{\varepsilon\})$. As $G' \subseteq G$, $G'^\omega \subseteq G^\omega$. Now as $G \subseteq G' \cup G' \mathrm{Stab}(R^\omega)$, $GG^\omega \subseteq (G' \cup G' \mathrm{Stab}(R^\omega)) G^\omega$. Hence $G^\omega \subseteq G' G^\omega$ since $\mathrm{Stab}(R^\omega)) G^\omega \subseteq G^\omega$. Thus $G^\omega \subseteq G'^\omega$ by the previous lemma. ■

In the general case, the languages $G \setminus G(\mathrm{Stab}(R^\omega) \setminus \{\varepsilon\})$ are not minimal ω-generators of R^ω. However, whenever R^ω is rebootable, they are ifl-codes and therefore minimal ω-generators of R^ω. Hence one can states the following result.

PROPOSITION 2.3: *Let* R^ω *be a rebootable ω-language. Then each ω-generator of R^ω contains an ω-generator of R^ω which is an ifl-code.*

In other words, whenever R^ω is rebootable, all minimal ω-generators of R^ω are ifl-codes. Of course, this condition is necessary but not sufficient. The set $R = ab$ is a counterexample. A first characterization of the rebootable ω-languages is given below.

PROPOSITION 2.4: *Let R be a language in Σ^+. The following conditions are equivalent:*

(i) *R^ω is a rebootable ω-language.*

(ii) $\mathrm{Stab}(R^\omega)$ *is a prefix-closed language.*

Proof: The implication (i) ⇒ (ii) is immediate since $\mathrm{Stab}(R^\omega) = \mathrm{Pref}(R^\omega)$. Conversely, we have $R^+ \subseteq \mathrm{Stab}(R^\omega)$ and $\mathrm{Pref}(R^\omega) = \mathrm{Pref}(R^+)$. Hence $\mathrm{Pref}(R^\omega) \subseteq \mathrm{Pref}(\mathrm{Stab}(R^\omega))$. And since $\mathrm{Stab}(R^\omega)$ is prefix-closed, $\mathrm{Pref}(R^\omega) \subseteq \mathrm{Stab}(R^\omega)$. As $\mathrm{Stab}(R^\omega) \subseteq \mathrm{Pref}(R^\omega)$, R^ω is rebootable. ■

Remarks: (1) For any ω-language L, the fact that L is rebootable implies that $\mathrm{Stab}(L)$ is prefix-closed. However, the converse does not hold. As an example, let L be the ω-language $a^* b^\omega$. Then $\mathrm{Stab}(L) = a^*$ which is a prefix-closed language. While L is not rebootable.

(2) Of course, if R is a prefix-closed language, R^ω is a rebootable ω-language. While R^ω may be rebootable without any ω-generator being rebootable. Indeed, let R be the language $a^* b$. Then R^ω is rebootable. However, every prefix-closed ω-generator of R^ω would contain the letter a, this is a contradiction!

PROPOSITION 2.5: *Let R be a rebootable language in Σ^+. Then R^ω is a rebootable ω-language.*

Proof: if R is a rebootable language, R is a semigroup and thus $\mathrm{Pref}(R^\omega)=\mathrm{Pref}(R)$. Hence R^ω is rebootable. ■

For the converse, we consider only the regular ω-power languages. Note that regular rebootable ω-power languages may be nondeterministic, as shown by the following example.

Example 2.6: Let R be the regular language $ac(a^* b)^* + a$. As $\mathrm{Pref}(R) \subseteq R^+$, $\mathrm{Pref}(R^+) R^\omega = R^\omega$, that is, R^ω is rebootable. On the other hand, it is easy to verify that R^ω is not a deterministic regular ω-language.

LEMMA 2.7: *Let R^ω be a deterministic regular ω-language. There exists an integer n such that for each ω-generator G of R^ω,* $\mathrm{Stab}(R^\omega) G^n$ *is an ω-generator of R^ω. Moreover, if $\mathscr{A}$ is a deterministic automaton recognizing R^ω then n can be chosen such that $n-1$ is the number of states of $\mathscr{A}$.*

Proof: For each integer $n>0$, $G^n \subseteq \mathrm{Stab}(R^\omega) G^n$. Hence $G^\omega \subseteq (\mathrm{Stab}(R^\omega) G)^\omega$. Now, let $\mathscr{A}=(\Sigma, Q, \{s\}, T, \delta)$ be a deterministic automaton Büchi-recognizing R^ω, we denote $\mathrm{Card}(Q)+1$ by n. Given $w \in (\mathrm{Stab}(R^\omega) G^\omega)^\omega$, we can write $w=u_1 v_1 \ldots u_i v_i \ldots$ where for each i, $u_i \in \mathrm{Pref}(R^\omega)$ and $v_i \in G^n$. As $u_1 v_1 \ldots u_i v_i^\omega \in R^\omega$, for each i, the set

$$Ex(\delta(\delta(s, u_1 v_1 \ \ldots \ v_{i-1} u_i), v_i)) \cap T \neq \varnothing$$

where $Ex(\delta(q, x_1 \ldots x_n))$ denotes the set $\{q' \in Q : q'=\delta(q, x_1 \ldots x_i)$ for some i in $\{1, \ldots, n\}\}$. Hence $w \in R^\omega$. ■

Thus for the deterministic regular ω-power languages, we obtain the following characterization:

PROPOSITION 2.8: *Let R^ω be a deterministic regular ω-language. The following properties are equivalent:*

(i) *R^ω is a rebootable ω-language.*

(ii) *R^ω has a rebootable ω-generator.*

Moreover, if R^ω is rebootable and recognized by a given deterministic finite automaton $\mathcal{A}$, then from $\mathcal{A}$ one can construct a finite automaton recognizing a rebootable ω-generator of R^ω.

Proof : The implication (ii) ⇒ (i) is stated in Proposition 2.5. It remains to prove the implication (i) ⇒ (ii). In view of Lemma 2.7, for any ω-generator G of R^ω, $\mathrm{Pref}(R^\omega)G^n$ is an ω-generator of R^ω for some n. Furthermore, $\mathrm{Pref}(R^\omega)G^n$ is rebootable. Indeed, we have the equality $\mathrm{Pref}(\mathrm{Pref}(R^\omega)G^n)=\mathrm{Pref}(R^\omega)$ and thus the equalities

$$\begin{aligned}\mathrm{Pref}(\mathrm{Pref}(R^\omega)G^n)(\mathrm{Pref}(R^\omega)G^n) &= \mathrm{Pref}(R^\omega)(\mathrm{Pref}(R^\omega)G^n)\\ &=(\mathrm{Pref}(R^\omega)\,\mathrm{Pref}(R^\omega))G^n=\mathrm{Pref}(R^\omega)G^n\end{aligned}$$

since $\mathrm{Pref}(R^\omega)$ is equal to the monoid $\mathrm{Stab}(R^\omega)$. Furthermore, we can construct regular ω-generators of R^ω [12]. Hence we can construct regular rebootable ω-generators of R^ω. ■

3. SUFFIX-CLOSED ω-LANGUAGES R^ω

Given an ω-language L, we consider the points of L where one can access while remaining in L, that is, we find the prefixes x of L such that $x^{-1}L\subseteq L$. This set of *cancellable* prefixes is $\{x\in\mathrm{Pref}(L): x^{-1}L\subseteq L\}$ and it is easy to verify that it is equal to $\mathrm{Stab}({}^cL)\cap\mathrm{Pref}(L)$. We are interested in ω-languages in which every prefix is an access point. Therefore, we investigate the ω-languages such that $\mathrm{Pref}(L)\subseteq\mathrm{Stab}({}^cL)$.

DEFINITION 3.1: Let L be an ω-language in Σ^ω. L is said to be suffix-closed if $(\Sigma^*)^{-1}L=L$, that is, if $\mathrm{Suf}(L)=L$.

Let us note that $(\Sigma^*)^{-1}L=L$ is equivalent to $(\mathrm{Pref}(L))^{-1}L=L$ and that the suffix-closed languages are characterized by the fact that $\mathrm{Pref}(L)\subseteq\mathrm{Stab}({}^cL)$.

Since $\mathrm{Suf}(R^\omega)=\mathrm{Suf}(R)R^\omega$, it is immediate that:

LEMMA 3.2: *Let R be a suffix-closed language, then R^ω is a suffix-closed ω-language.*

However, it may happen for some suffix-closed and deterministic regular ω-languages R^ω that R^ω has no suffix-closed ω-generator, as shown by the following example.

Example 3.3: Let R be the regular prefix-free language a^*ba. As $\mathrm{Suf}(R)=R+a+\varepsilon$, $\mathrm{Suf}(R)R\subseteq R^+$. Hence, R^ω is suffix-closed. R^ω is obviously regular. Furthermore, $R^\omega=\mathrm{Lim}(R^+)$, that is, R^ω is deterministic [8]. However, no ω-generator of R^ω is suffix-closed. Indeed every ω-generator would contain a or b. Thus a^ω or b^ω would belong to R^ω, a contradiction!

In other words, the suffix-closed ω-generators do not characterize the regular suffix-closed ω-languages R^ω. Instead, they are characterized via suffix-closed languages by the following proposition.

PROPOSITION 3.4: *Let R be a language in Σ^+. The following properties are equivalent:*

(i) *R^ω is suffix-closed.*

(ii) $\mathrm{Stab}(R^\omega)$ *is suffix-closed.*

Proof: Assume that R^ω is suffix-closed. Let $u \in \mathrm{Stab}(R^\omega)$. We have $u R^\omega \subseteq R^\omega$ and for any suffix u' of u, also $u' R^\omega \subseteq R^\omega$. Hence $u' \in \mathrm{Stab}(R^\omega)$. Conversely, as $R \subseteq \mathrm{Stab}(R^\omega)$, $\mathrm{Suf}(R) \subseteq \mathrm{Stab}(R^\omega)$. On the other hand $\mathrm{Suf}(R^\omega) = \mathrm{Suf}(R) R^\omega$, hence $\mathrm{Suf}(R^\omega) \subseteq R^\omega$. ■

Remark: If L is not an ω-power language, the fact that $\mathrm{Stab}(L)$ is suffix-closed does not imply that L is suffix-closed. Consider $L = a^+ b^\omega$ for example.

On the other hand, by definition, the fact that R^ω is suffix-closed implies that $\mathrm{Stab}({}^c(R^\omega)) \cap \mathrm{Pref}(R^\omega)$ is prefix-closed. Unfortunately this last condition is not sufficient. Consider for example $R = ba^*$, where $\mathrm{Stab}({}^c(R^\omega)) \cap \mathrm{Pref}(R^\omega)$ is reduced to the set $\{\varepsilon\}$. Nevertheless, we shall see that it can be completed to a sufficient condition.

LEMMA 3.5: *Let R be a language in Σ^+. If R^ω is suffix-closed then each ω-generator of R^ω contains a prefix-free ω-generator of R^ω. Furthermore each prefix-free ω-generator of R^ω is contained in* $\mathrm{Stab}({}^c(R^\omega)) \cap \mathrm{Pref}(R^\omega)$.

Proof: Let G be an ω-generator of R^ω. By Lemma 2.2 the language $P = G \setminus G(\mathrm{Stab}(R^\omega) \setminus \{\varepsilon\})$ is an ω-generator of R^ω. We prove that P is a prefix-free language. Assume that there exist u and $v \in P$ such that $uu' = v$. As $u' R^\omega \subseteq u^{-1}(R^\omega)$, we have $u' R^\omega \subseteq R^\omega$, that is, $u' \in \mathrm{Stab}(R^\omega)$. Now, the definition of P implies that $u' = \varepsilon$. Hence P is prefix-free. Now, for each $u \in P$, $R^\omega \subseteq u^{-1}(R^\omega) \subseteq \mathrm{Suf}(R^\omega) = R^\omega$. Hence $u^{-1}(R^\omega) = R^\omega$, thus

$$P \subseteq \mathrm{Stab}({}^c(R^\omega)) \cap \mathrm{Pref}(R^\omega). \quad ■$$

In other words, if R^ω is suffix-closed, then all minimal ω-generators of R^ω are prefix-free languages. This condition is necessary, but not sufficient, consider $R = ab$ for example.

PROPOSITION 3.6: *Let R be a language in Σ^+. The following properties are equivalent:*

(i) *R^ω is suffix-closed.*

(ii) $\mathrm{Stab}({}^c(R^\omega)) \cap \mathrm{Pref}(R^\omega)$ *is prefix-closed and contains an ω-generator of R^ω.*

Proof: If R^ω is suffix-closed, by Lemma 3.5 $\mathrm{Stab}({}^c(R^\omega)) \cap \mathrm{Pref}(R^\omega)$ contains an ω-generator of R^ω. Furthermore, let $u \in \mathrm{Stab}({}^c(R^\omega)) \cap \mathrm{Pref}(R^\omega)$ and let $u' \in \mathrm{Pref}(u)$. If $u' \notin \mathrm{Stab}({}^c(R^\omega))$, $u' w \in R^\omega$ for some $w \in {}^c(R^\omega)$. Since R^ω is suffix-closed, this is a contradiction! Hence $\mathrm{Stab}({}^c(R^\omega)) \cap \mathrm{Pref}(R^\omega)$ is prefix-closed. Conversely, let G be an ω-generator of R^ω, such that $G \subseteqq \mathrm{Stab}({}^c(R^\omega)) \cap \mathrm{Pref}(R^\omega)$. We have $\mathrm{Suf}(G^\omega) = \mathrm{Suf}(G)\, G^\omega \subseteqq (\mathrm{Pref}(G))^{-1} G^\omega$. Since $\mathrm{Stab}({}^c(R^\omega)) \cap \mathrm{Pref}(R^\omega)$ is prefix-closed, we obtain the inclusion $(\mathrm{Pref}(G))^{-1} G^\omega \subseteqq G^\omega$. Thus $R^\omega = G^\omega$ is suffix-closed. ■

Example 2.6 shows that regular ω-power languages may be nondeterministic. In contrast to this, for the regular suffix-closed ω-power languages we have the following result.

COROLLARY 3.7: *Let R be a regular language in Σ^+. If R^ω is suffix-closed then R^ω is a deterministic regular ω-language.*

Proof: By Lemma 3.5, $R^\omega = P^\omega$ for some prefix-free language P. Now since P is prefix-free, $P^\omega = \mathrm{Lim}(P^*)$. Hence R^ω is regular and deterministic. ■

Remark: If L is not an ω-power language L may be suffix-closed, regular and nondeterministic. Consider $(a+b)^* a^\omega$ for example.

Now we are able to characterize the regular suffix-closed ω-languages R^ω via their ω-generators.

PROPOSITION 3.8: *Let R be a regular language in Σ^+. The following properties are equivalent.*

(i) *R^ω is suffix-closed.*

(ii) *$R^\omega = G^\omega$ for some language G such that* $\mathrm{Suf}(G)\, G = G$.

Moreover, if R^ω is suffix-closed and recognized by a given deterministic finite automaton $\mathscr{A}$, then from $\mathscr{A}$ one can construct a finite automaton recognizing a suffix-closed ω-generator of R^ω.

Proof: If R^ω is suffix-closed, by Lemma 2.7 the language $G = \mathrm{Stab}(R^\omega) R^n$ is an ω-generator of R^ω for some computable integer $n > 0$. Now G satisfies $\mathrm{Suf}(G)\, G = G$. Indeed $G \subseteqq \mathrm{Stab}(R^\omega)$. Hence, in view of Proposition 3.4, $\mathrm{Suf}(G) \subseteqq \mathrm{Stab}(R^\omega)$. Thus $\mathrm{Suf}(G)\, G \subseteqq \mathrm{Stab}(R^\omega)\, G = G$ and so $\mathrm{Suf}(G)\, G = G$. Furthermore an automaton recognizing G can be constructed. If $R^\omega = G^\omega$ for some language G such that $\mathrm{Suf}(G)\, G = G$, R^ω is suffix- closed $\mathrm{Suf}(G^\omega) = \mathrm{Suf}(G)\, G^\omega$. ■

4. LEFT ω-IDEALS R^{ω}

Now we consider the ω-languages which are both rebootable and suffix-closed. They are characterized by $\mathrm{Pref}(L)\,\mathrm{Suf}(L)=\mathrm{L}$. In fact, we prove that they are nothing but the absolutely closed ω-languages studied in [7]. Moreover in the case when $L=R^{\omega}$, they are exactly the left ω-ideals [7] of the form R^{ω}. Then these ω-languages R^{ω} are characterized, first by using the properties of being rebootable and suffix-closed, then via ideals of Σ^*, finally using the syntactic monoid of R^{ω} in the sense of [1].

DEFINITION 4.1: [7] An ω-language L is said to be a left ω-ideal in Σ^* if $\Sigma^* L=L$.

That is the equality $\mathrm{Stab}(L)=\Sigma^*$ characterizes the left ω-ideals. Since $\mathrm{Stab}(L)$ is a monoid, one can also note that L is a left ω-ideal iff $\mathrm{Stab}(L)$ is a left-ideal. Moreover, as in the case of languages, L is a left ω-ideal in $\mathrm{Alph}(L)$ iff ${}^{c}L$ is a suffix-closed ω-language.

DEFINITION 4.2: An ω-language L is said to be absolutely closed in Σ^* if L is both a left ω-ideal in Σ^* and a suffix-closed ω-language.

The following proposition characterizes the ω-languages which are both rebootable and suffix-closed.

PROPOSITION 4.3: *Let L be an ω-language such that $\Sigma=\mathrm{Alph}(L)$. The following properties are equivalent:*

(i) $\mathrm{Pref}(L)\,\mathrm{Suf}(L)=L$.

(ii) *L is absolutely closed in Σ^*.*

Proof: Assume that $\mathrm{Pref}(L)\,\mathrm{Suf}(L)=L$. As $\varepsilon\in\mathrm{Pref}(L)$, L is suffix-closed. Now, given a letter x in Σ, since L is suffix-closed, $xw\in L$ for some $w\in\Sigma^{\omega}$. And since L is rebootable, $x\in\mathrm{Stab}(L)$. Now as $\mathrm{Stab}(L)$ is a monoid, we obtain $\mathrm{Stab}(L)=\Sigma^*$. That is L is a left-ideal in Σ^* and therefore L is absolutely closed in Σ^*. The converse is obvious. ■

Hence every ω-language L which is both rebootable and suffix-closed, is a left ω-ideal in $(\mathrm{Alph}(L))^*$. Conversely, all left ω-ideals L are rebootable since $\mathrm{Stab}(L)=\Sigma^*$ and $\Sigma^* L=L$. However, they are not suffix-closed in general. For example, $L=(a+b)^* ba^{\omega}$ is a left ω-ideal with $a^{\omega}\in\mathrm{Suf}(L)\,L$. In contrast to this, the left ω-ideals of the form R^{ω} are always suffix-closed as stated in the following lemma.

LEMMA 4.4: *Let R be a language in Σ^+. If R^{ω} is a left ω-ideal then R^{ω} is suffix-closed.*

Proof: For every ω-word w in $(\Sigma^*)^{-1} R^\omega$, there exists a word $u \in \Sigma^*$ such that $uw \in R^\omega$. Hence there exist a word $v \in \Sigma^*$ and an ω-word $w' \in \Sigma^\omega$ such that $w = vw'$, $uv \in R^+$ and $w' \in R^\omega$. As R^ω is a left ω-ideal, one has $vw' \in R^\omega$. ■

PROPOSITION 4.5: *Let R be a language in* Σ^+. *The following properties are equivalent:*

(i) R^ω *is a left* ω-*ideal.*

(ii) R^ω *is rebootable and suffix-closed.*

(iii) Stab(R^ω) *is factor-closed.*

Proof: If R^ω is a left ω-ideal, R^ω is rebootable. Moreover R^ω is suffix-closed by Lemma 4.4. On the other hand, R^ω is rebootable and suffix-closed iff Stab(R^ω) is factor-closed by Proposition 2.4 and Proposition 3.4. Finally, If R^ω is rebootable and suffix-closed, R^ω is a left ω-ideal by Proposition 4.3. ■

Remarks: (1) If R^ω is a left ω-ideal, then ${}^c(R^\omega)$ is also a left ω-ideal. The converse does not hold. Consider $R = a + ba$ for example.

(2) If L is not an ω-power language, (iii) does not imply (i). Consider $L = a^* b^\omega$ for example.

When R^ω is suffix-closed, for each ω-generator G of R^ω, suf(G^+) is contained in Pref(G^+). Hence, we have the following lemma.

LEMMA 4.6: *Let* R^ω *be a suffix-closed* ω-*language. Then for every* ω-*generator* G *of* R^ω, *we have* Fact(G^+) = Pref(G^+).

LEMMA 4.7: *Let R be a regular language in* Σ^+. *If* R^ω *is a left* ω-*ideal then* R^ω *is a deterministic regular* ω-*language.*

Proof: Since the left ω-ideals R^ω are suffix-closed ω-power languages, Corollary 3.7 gives the results. ■

Remark: Some regular left ω-ideals may be nondeterministic. For example, consider $\Sigma^* a^\omega$.

Now, we can state characterizations for the regular left ω-ideals R^ω using their ω-generators.

PROPOSITION 4.8: *Let R be a regular language in* Σ^+. *Then the following properties are equivalent:*

(i) R^ω *is a left* ω-*ideal.*

(ii) $R^\omega = G^\omega$ *for some language G such that* Fact$(G)\,G = G$.

(iii) R^ω *as a left ideal for* ω-*generator.*

(iv) *R^ω as an ideal for ω-generator.*

Moreover, if R^ω is a left ω-ideal and recognized by a given deterministic finite autamaton $\mathcal{A}$, then from $\mathcal{A}$ one can construct a finite automaton recognizing an ω-generator G of R^ω such that $\text{Fact}(G)G=G$, *G is a left ideal or G is an ideal.*

Proof: (i) $\Rightarrow$ (ii) By Proposition 4.5, R^ω is rebootable and suffix-closed. Then Corollary 3.7 implies that R^ω is a deterministic regular ω-language. Hence $R^\omega = G^\omega$ for some language G such that $\text{Pref}(G)G=G$ by Proposition 2.8. Now Lemma 4.6 gives the implication.

(ii) $\Rightarrow$ (i) $\text{Fact}(G)G=G$ implies $\text{Suf}(G)G=G$. Hence R^ω is a suffix-closed ω-power language, and thus it is a regular deterministic ω-language. Then the equality $\text{Pref}(G)G=G$ implies that the ω-language R^ω is rebootable.

(i) $\Rightarrow$ (iii) By Proposition 2.7, each left ω-ideal R^ω has a left-ideal $\Sigma^* I$ for ω-generator.

(iii) $\Rightarrow$ (iv) This implication comes from the equality $(\Sigma^* I)^\omega = (\Sigma^* I \Sigma^*)^\omega$.

(iv) $\Rightarrow$ (i) If $R^\omega = I^\omega$ for some ideal I, then R^ω is a left ω-ideal. ■

Let us now consider the minimal ω-generators of the left ω-ideals R^ω. Since $\text{Stab}(R^\omega)=\Sigma^*$, every minimal ω-generator of R^ω is a prefix code. More precisely, in the case when R^ω is the whole left-ideal Σ^ω, the minimal ω-generators of R^ω are exactly the finite maximal prefix codes of Σ^*, otherwise we have:

PROPOSITION 4.9: *Let R^ω be a left ω-ideal such that $R^\omega \neq \Sigma^\omega$. The minimal ω-generators of R^ω are exactly the infinite maximal prefix codes ω-generating R^ω.*

Proof: Since $\text{Stab}(R^\omega)=\Sigma^*$, every minimal ω-generator G of R^ω is a prefix code. It remains to prove that G is maximal and infinite. Assume that G is not maximal, that is, $G+u$ is a prefix code for some $u \notin G$. As $uv^\omega \in R^\omega$ for any v in R, u is a prefix of g or g is a prefix of u for some g in G, this is a contradiction! Furthermore G is infinite otherwise R^ω is closed [8] and then it is the whole ω-language Σ^ω. ■

Conservely the fact that C is a maximal prefix code, does not imply that C^ω is a left ω-ideal. For example, $C=b+a^*a$ is an infinite maximal prefix code. However C^ω is not a left ω-ideal, indeed $b^\omega \in C^\omega$ and $ab^\omega \notin C^\omega$. For the semaphore codes [3], which are particular maximal prefix codes, we have the following characterization.

PROPOSITION 4.10: *Let R be a language in Σ^+. The following properties are equivalent:*

(i) *R^ω is a left ω-ideal.*

(ii) *$R^\omega = C^\omega$ for some semaphore code C.*

Proof: The implication (i) ⇒ (ii) proceeds from Proposition 4.9. Conversely, let C be a semaphore code. $C\Sigma^*$ is a left ω-ideal and $(C\Sigma^*)^\omega = C(\Sigma^* C\Sigma^*)(C\Sigma^*)^\omega$ which is contained in $C(C\Sigma^*)(C\Sigma^*)^\omega$. Hence $(C\Sigma^*)^\omega \subseteq C^\omega$, thus $(C\Sigma^*)^\omega = C^\omega$. ■

Remark: It may happen that some minimal ω-generators of an ω-ideal R^ω are not semaphore codes.

We end this part with a characterization of the regular left ω-ideals R^ω via the syntactic monoid [1] of R^ω. Note that the syntactic monoid of a left ω-ideal R^ω, taken in the sense of [7] is trivial.

LEMMA 4.11: *Let I be a regular ideal in Σ^*. Then I is contained in a class of $\mathcal{SM}(I^\omega)$.*

Proof: Let v and v' be two words $\in I$. For every u, $u' \in \Sigma^*$ and $w \in \Sigma^\omega$, $uvw \in I^\omega$ iff $uv'w \in I^\omega$ and $u(u'v)^\omega$ and $u(u'v')^\omega \in I^\omega$. Thus v and v' are syntacticly equivalent. ■

Now we have the following result which emphasizes that there exists always one greatest ideal ω-generating I^ω, while I^ω has not necessarily one greatest ω-generator [12].

PROPOSITION 4.12: *Let I be a regular ideal in Σ^*. Then $\pi(I)$ is the zero in $\mathcal{SM}(I^\omega)$ and $\pi^{-1}(\pi(I))$ is the greatest ideal ω-generating I^ω.*

Proof: By definition, $\pi(I)$ is the zero in $\mathcal{SM}(I^\omega)$. Moreover $\pi^{-1}(\pi(I))$ is an ideal and as $I \subseteq \pi^{-1}(\pi(I))$, $I^\omega \subseteq (\pi^{-1}(\pi(I)))^\omega$. On the other hand for each $w \in UP[(\pi^{-1}(\pi(I)))^\omega]$, $w = uv^\omega$ for some u and $v \in \pi^{-1}(\pi(I))$, since $\pi^{-1}(\pi(I))$ is an ideal. Then u and v are syntacticly equivalent with any word in I. Hence $uv^\omega \in I^\omega$. Now as I^ω and $(\pi^{-1}(\pi(I)))^\omega$ are regular ω-languages, we have the equality $I^\omega = (\pi^{-1}(\pi(I)))^\omega$ [5]. ■

PROPOSITION 4.13: *Let R be a regular language. The following properties are equivalent:*

(i) *R^ω is a left ω-ideal.*

(ii) *$\mathcal{SM}(R^\omega)$ have a zero f and f is such that $\pi^{-1}(f)$ is an ω-generator of R^ω.*

Proof: If R^ω is a left ω-ideal, $R^\omega = I^\omega$ for some regular ideal I. Hence, $\pi(I)$ is a zero in $\mathcal{SM}(R^\omega)$ and $\pi^{-1}(\pi(I))$ is an ω-generator of R^ω. Conversely, if f is the zero of $\mathcal{SM}(R^\omega)$, R^ω is left ω-ideal since $\pi^{-1}(f)$ is a left ω-ideal. ∎

ACKNOWLEDGEMENTS

The authors are very grateful to two referees. Their suggestions and corrections have greatly improved the presentation of the results.

REFERENCES

1. A. Arnold, A Syntactic Congruence for Rational ω-languages, *Theoret Comput. Sci.*, 1985, *39*, pp. 333-335.
2. A. Arnold and M. Nivat, Comportements de processus, *Rapport* interne, L.I.T.P., 1982, pp. 82-12.
3. J. Bertsel and D. Perrin, Theory of Codes, *Academic Press*, New York, 1985.
4. L. Boasson and M. Nivat, Adherences of Languages, *J. Comput. System Sci.*, 1980, *20*, pp. 285-309.
5. J. R. Büchi, On Decision Method in Restricted Second-Order Arithmetics, *Proc. Congr. Logic, Method. and Phulos. Sci.*, Stanford Univ. Press, 1962, p. 1-11.
6. S. Eilenberg, Automata, Languages and Machines, *A*, *Academic Press*, New York, 1974.
7. H. Jürgensen and G. Thierrin, On ω-languages Whose Syntactic Monoid is Trivial, *J. Comput. Inform. Sci.*, 1983, *12*, pp. 359-365.
8. L. H. Landweber, Decision Problems for ω-Automata, *Math. Syst. Theory*, 1969, *3*, pp. 376-384.
9. M. Latteux and E. Timmeran, Finitely Generated ω-Languages, *Inform Process. Lett.*, 1986, *23*, pp. 171-175.
10. R. Linder and L. Staiger, Algebraische Codierungstheorie-Theorie der sequentiellen Codierungen, *Akademie-Verlag*, Berlin, 1977.
11. I. Litovsky, Générateurs des langages rationnels de mots infinis, *Thèse*, Univ. Lille-I, 1988.
12. I. Litovsky and E. Timmerman, On Generators of Rational ω-Power Languages, *Theoret. Comput. Sci.*, 1987, *53*, pp. 187-200.
13. R. MacNaughton, Testing and Generating Infinite Sequences by a Finite Automaton, *Inform. Control*, 1966, *9*, pp. 521-530.
14. L. Staiger, A Note on Connected ω-languages, *Elektron. Inform. Kybernetik*, 1980, *16*, 5/6, pp. 245-251.
15. L. Staiger, Finite State ω-Languages, *J. Comput. System Sci.*, 1983, *27*, pp. 434-448.
16. L. Saiger, On Infinitary Finite Length Codes, *Theore. Inform. Appli.* 1986, *20,* 4, pp. 483-494.
17. L. Staiger, Research in the Theory of ω-Languages, *Elektron. Inform. Kybernetik*, 1987, *23*, 8/9, pp. 415-439.