AN AUTOMATA-THEORETIC APPROACH TO THE WORD PROBLEM FOR ω-TERMS OVER R

JORGE ALMEIDA AND MARC ZEITOUN

ABSTRACT. This paper studies the pseudovariety R of all finite $\mathcal{R}$-trivial semigroups. We give a representation of pseudowords over R by infinite trees, called R-trees. Then we show that a pseudoword is an ω-term if and only if its associated tree is regular (*i.e.*, it can be folded into a finite graph), or equivalently, if the ω-term has a finite number of tails. We give a linear algorithm to compute a compact representation of the R-tree for ω-terms, which yields a linear solution of the word problem for ω-terms over R. We finally exhibit a basis for the ω-variety generated by R and we show that there is no finite basis. Several results can be compared to recent work of Bloom and Choffrut on long words.

1. INTRODUCTION

The main contribution of this paper is the solution of a word problem over R, the pseudovariety of all finite $\mathcal{R}$-trivial semigroups. This pseudovariety corresponds, in Eilenberg's correspondence, to disjoint unions of languages of the form $A_0^*a_1A_1^*a_2\dots a_nA_n^*$, where the a_i's are letters and $a_i \notin A_{i-1}$ for $1 \le i \le n$. Also, finite $\mathcal{R}$-trivial semigroups are the divisors of transition semigroups of the so-called *very weak* automata, that is, automata whose state set is partially ordered and the transition function does not decrease the state. They can even be characterized as the divisors of *extensive* automata, that is, very weak automata where the order on states is total.

Given two terms built from letters of an alphabet A using the concatenation and the ω-power, we show how to decide in linear time whether these terms coincide over all A-generated elements of R, with the usual interpretation of the ω-power in semigroups. We also characterize the set of pseudowords—also known as implicit operations—over R which can be represented by such ω-terms. Since R satisfies the identity in ω-terms $x^{\omega-1} = x^\omega$, all results of this paper can be formulated either for ω-terms, or for κ-terms, where $\kappa = \{\_\cdot\_, \_^{\omega-1}\}$ is the implicit signature consisting of the semigroup multiplication and the unary $(\omega-1)$-power. We shall state most results using the signature $\{\_\cdot\_, \_^{\omega}\}$, but this is mainly a matter of style.

The motivation of this work is the κ-tameness property for R. Historically, the notion of tameness was discovered in attempting to find general decidability properties of pseudovarieties which might be preserved under taking semidirect products [5]. It remains open whether it does indeed play such a role, although under certain finiteness hypotheses it has been shown to do so [2].

Proving the κ-tameness of a pseudovariety V consists in solving two subproblems. The first one is the κ-word problem, for which this paper gives an efficient solution. Informally, the second question is whether equation systems[1] with rational constraints having a solution in any semigroup of V also have a *uniform* solution in κ-terms, satisfying the same constraints. This property has proven to be robust and helpful for the solution of the membership problem (see *e.g.* [4], where the κ-tameness of R is used to decide joins involving R.) Moreover, if V enjoys it, then V has decidable pointlikes, an important property of pseudovarieties [5, 20].

Another motivation for this study comes from the related work of S. Bloom and Ch. Choffrut [11]. Given a finite set A, the collection of all finite or countably infinite A-labeled posets can be endowed with a binary concatenation operation of posets $\_\cdot\_$, and with a unary ω-power $\_^\omega$. Bloom and Choffrut recently proved in [11] that the Birkhoff variety generated by these algebras is not finitely based, and that it is defined by the following

2000 *Mathematics Subject Classification.* 20M05, 20M07, 20M35.

Key words and phrases. Semigroup, pseudovariety, word problem, automata minimization, pseudoword, omega-term, identity basis.

This work was done in part while the authors were visiting the *Centro Internacional de Matemática*, in Coimbra, Portugal. Financial support of *Fundação Calouste Gulbenkian* (FCG), FCT, *Faculdade de Ciências da Universidade de Lisboa* (FCUL) and *Reitoria da Universidade do Porto* is gratefully acknowledged. This work was also partly supported by the PESSOA French-Portuguese project Egide-Grices 11113YM *Automata, profinite semigroups and symbolic dynamics*.

Jorge Almeida: Work supported, in part, by *Fundação para a Ciência e a Tecnologia* (FCT) through the *Centro de Matemática da Universidade do Porto*, by the FCT and POCTI approved project POCTI/ 32817/MAT/2000 which is partly funded by the European Community Fund FEDER, and by the INTAS grant #99-1224.

[1]Strictly speaking, if the equations are given by arbitrary κ-terms, the property is named complete κ-tameness, whereas κ-tameness stands for a restricted class of equations.

set of identities.

$$\begin{cases} (x \cdot y) \cdot z = x \cdot (y \cdot z) \\ (x^r)^\omega = x^\omega, \qquad r \geq 2 \\ (xy)^\omega = x(yx)^\omega \end{cases}$$

They also studied ordinal words, that is, labeled ordinals. Among them, they characterized labeled ordinals built from letters of A using the operations $\_ \cdot \_$ and $\_^\omega$: these are exactly the ordinals of length less than ω^ω and having a finite number of tails (suffixes, in some sense). Finally, they proved that the word problem for two ω-terms u, v can be solved in time $O(|u|^2|v|^2)$, where $|u|$ and $|v|$ denote the lengths of u and v.

Motivated by these results and by the fact that pseudowords over R are labeled ordinals [7], we show that:

– the word problem for ω-terms u, v over R and on an alphabet A can be solved in time $O\big(|A|(|u| + |v|)\big)$, using automata-based techniques. More specifically, we can compute for any ω-term u an automaton $\mathcal{A}(u)$ of size $|A||u|$. Two terms are equal over R if and only if the associated automata recognize the same language. Due to the specific form of these automata, this can again be tested in linear time;
– a pseudoword over R coincides with an ω-term if and only if it has a finite number of distinguished suffixes (resp. factors);
– the variety of ω-semigroups generated by R is not finitely based;
– we exhibit an infinite basis for this variety.

Although these results are very similar to those of [11], the involved word problems are different, and neither set of results seems to directly imply the other one.

The paper is organized as follows. In Section 2, we set up the notation and we recall prerequisites on semigroups and pseudovarieties. In Section 3, we exhibit a sufficient condition for continuity of infinite products in pro-R semigroups and we use it to associate R-trees and R-automata to pseudowords over R. These objects are used in Section 4 to solve the word problem for ω-terms over R and to derive several characterizations of pseudowords having a representation as an ω-term. We then exhibit a canonical form for ω-terms over R, which can be exponentially larger than the original term, in terms of the size of the alphabet, but remains polynomially small, for a fixed alphabet, in terms of the size of the minimal R-automaton of the ω-term. Section 5 presents a linear-time algorithm to compute the canonical R-automaton associated to an ω-term, defined in Section 3, thus proving that the complexity of the word problem for ω-terms over R is linear. We introduce in Section 6 a set of identities in ω-terms. We prove, by a rather involved argument with various levels of nested inductions which uses several key results from previous sections, that this set is a basis for the ω-variety generated by R. We also show that this ω-variety is not finitely based. It should be noted that a recursive basis for R was previously announced without proof in [6]. It included our basis and two extra superfluous identities. Finally, we discuss some open problems in Section 7.

2. Preliminaries

We briefly recall notation in this section. We refer the reader to [2] for the notions of pseudovarieties, pro-V semigroups and implicit signatures. We assume that the reader is acquainted with these notions, and is also familiar with the basics of automata theory. See [16] for instance.

2.1. **Notation. Words.** Throughout this paper, A denotes a finite set. We write $|A|$ for its cardinality. The free semigroup (resp. the free monoid) generated by A is denoted by A^+ (resp. by A^*). As usual, we write x^* instead of $\{x\}^*$. The length of a word $x \in A^*$ is denoted by $|x|$. The empty word is denoted by ε or 1. The number of occurrences of a letter $a \in A$ in x is denoted by $|x|_a$. Finally, the *content* $c(x)$ of x is the smallest subset B of A such that $x \in B^*$. Given a language $L \subseteq A^*$, we denote by L^1 the language $L \cup \{1\}$.

Automata. We denote a (deterministic) automaton over an alphabet A by a tuple $\mathcal{A} = \langle V, \delta, \mathsf{v}_0, F\rangle$, where V is the state set of $\mathcal{A}$, $\mathsf{v}_0 \in V$ is its initial state, $F \subseteq V$ is its set of final states and $\delta : V \times A \to V$ is its transition function. We will often denote by $\mathsf{v}.a$ the state $\delta(\mathsf{v}, a)$ reached from v by reading letter a, when this state exists. We denote by $\mathsf{v}.L$ the set of all states reached from v by reading a word of L.

Functions. In the sequel, functions are assumed to be partial unless otherwise stated. Let X, Y denote sets. If $\mathcal{C}$ is a set of functions from X to Y, and if $X' \subseteq X$, then we set $\mathcal{C}(X') = \{y \in Y \mid \exists f \in \mathcal{C}, \exists x \in X', y = f(x)\}$.

For a function $f : X \to Y$, let $\mathrm{dom}(f) = \{x \in X \mid f(x) \text{ is defined}\}$ denote its domain. If $f, g : X \to Y$ are two functions and if $x \in X$, then we write $f(x) = g(x)$ to mean that x belongs to $\mathrm{dom}(f)$ if and only if it belongs to $\mathrm{dom}(g)$ and if $x \in \mathrm{dom}(f)$, then $f(x) = g(x)$. Finally, let $\mathcal{F}$ be a set of functions from X to itself. Abusing notation, we denote again by $\mathcal{F}^*$ the set $\{\alpha_1 \circ \cdots \circ \alpha_n \mid n \geq 0, \alpha_i \in \mathcal{F}\}$. We will also often write fg instead of $f \circ g$ and $fg(x)$ instead of $f(g(x))$.

Semigroups, Green relation $\mathcal{R}$. Given a semigroup S, we let S^1 be the semigroup S itself if it is a monoid, or the disjoint union $S \uplus \{1\}$ where 1 acts as a neutral element otherwise. Given an element s of a finite semigroup (resp. of a compact topological semigroup), the subsemigroup (resp. the closed subsemigroup) generated by s contains a unique idempotent, denoted by s^ω. The set of idempotents of a semigroup S is denoted by $E(S)$.

For any semigroup S, we denote by $\preceq_{\mathcal{R}} \subseteq S \times S$ the relation such that $s \preceq_{\mathcal{R}} t$ if and only if there exists $u \in S^1$ such that $s = tu$. The equivalence relation $\mathcal{R}$ is defined by $s \mathrel{\mathcal{R}} t \Leftrightarrow s \preceq_{\mathcal{R}} t$ and $t \preceq_{\mathcal{R}} s$. A semigroup S is $\mathcal{R}$-*trivial* if for all $s, t \in S$ we have $s \mathrel{\mathcal{R}} t \Rightarrow s = t$.

2.2. **Background. Pseudovarieties.** A semigroup *pseudovariety* is a class of finite semigroups closed under taking subsemigroups, homomorphic images and finite direct products. In what follows, V denotes a pseudovariety.

We denote by S the pseudovariety of all finite semigroups. Given a semigroup $S \in \mathsf{S}$, an element s of S and an integer $k \in \mathbb{Z}$, the sequence $(s^{n!+k})_n$ is defined for all sufficiently large n and eventually stabilizes, that is it converges in the discrete topology. We denote by $s^{\omega+k}$ its limit.

A semigroup S is *aperiodic* if $s^\omega = s^{\omega+1}$ for all $s \in S$. We denote by A the pseudovariety of all finite aperiodic semigroups. In the present paper, we focus on the pseudovariety R of all finite $\mathcal{R}$-trivial semigroups, which is a subpseudovariety of A. It is classical that a semigroup S is in R if and only if for all $s, t \in S$ we have $(st)^\omega = (st)^\omega s$.

Profinite and pro-V semigroups. In what follows, finite semigroups are all equipped with the discrete topology. We say that a class $\mathcal{H}$ of homomorphisms from a semigroup S into semigroups *separates points* if for all distinct elements s, t of S, there exists $h \in \mathcal{H}$ such that $h(s) \neq h(t)$.

A topological semigroup is a *profinite semigroup* (resp. a *pro-V semigroup*) if it is a projective limit of finite semigroups (resp. of semigroups of V). It is well known that profinite semigroups are 0-dimensional (and hence totally disconnected). More precisely, a pro-V semigroup is a compact semigroup S which is *residually in* V in the sense that the class of all continuous homomorphisms from S into members of V separates points.

Since, in a finite semigroup S, the sequence $(s^{n!+k})_{n>|k|}$ converges for $s \in S$ and $k \in \mathbb{Z}$, the same is true in every profinite semigroup. We denote the limit by $s^{\omega+k}$. This extends the notation introduced above for finite semigroups.

A profinite semigroup S is A-*generated* if there exists a function $\eta : A \to S$ such that the subsemigroup generated by $\eta(A)$ is dense in S. We say that η is the generating function. Let 2^A be the set of all subsets of A. Then, $(2^A, \cup)$ is a finite semigroup. Let S be a profinite A-generated semigroup, and let $\eta : A \to S$ be the generating function. We say that S has a *content function* if there exists a continuous homomorphism $c : S \to 2^A$ such that $c\eta(a) = \{a\}$ for all $a \in A$. If such a continuous homomorphism exists, then it is unique. It may then be defined by the condition that, for $a \in A$ and $s \in S$, $a \in c(s)$ if and only if there is some factorization of s in which $\eta(a)$ is one of the factors.

Given a finite set A and a pseudovariety V, there is a *free pro-V semigroup on A*, that is a pro-V semigroup S endowed with a generating function $\iota : A \to S$ such that, for every function $\varphi : A \to T$ into a pro-V semigroup T, there exists a (unique) continuous homomorphism $\hat{\varphi} : S \to T$ such that $\hat{\varphi} \circ \iota = \varphi$. It is immediate to verify that such a pro-V semigroup is unique, up to isomorphism of topological semigroups respecting the choice of generators; we denote it $\overline{\Omega}_A\mathsf{V}$.

The *canonical projection on* V is the unique continuous homomorphism $p_{\mathsf{V}} : \overline{\Omega}_A\mathsf{S} \to \overline{\Omega}_A\mathsf{V}$ determined by the choice of generators.

Pseudowords and pseudoidentities. The elements of $\overline{\Omega}_A\mathsf{S}$ are named *pseudowords* (sometimes also *implicit operations* or *profinite words*). For example, if $u \in \overline{\Omega}_A\mathsf{S}$, then u^ω is again a pseudoword.

A formal equality of the form $u = v$ with $u, v \in \overline{\Omega}_A\mathsf{S}$ for some finite set A is called a *pseudoidentity*. It is said to be *valid* in a profinite semigroup S and we write $S \models u = v$ if $\varphi(u) = \varphi(v)$ for every continuous homomorphism $\varphi : \overline{\Omega}_A\mathsf{S} \to S$. For instance, the pseudoidentity $x^{\omega+1} = x^\omega$ is valid in any aperiodic semigroup. It is easy to check that the validity of a pseudoidentity in a finite semigroup is preserved under taking homomorphic images, closed subsemigroups and finite direct products. Hence the class of all finite semigroups which verify all members of a given set Σ of pseudoidentities is a pseudovariety, which is said to be *defined* by Σ. Conversely, by Reiterman's Theorem [18] every pseudovariety is defined by some set of pseudoidentities.

In the language of pseudoidentities, earlier definitions of pseudovarieties which are important for this paper may now be formulated as follows: A is defined by $x^{\omega+1} = x^\omega$ and R is defined by $(xy)^\omega x = (xy)^\omega$. Of course, there are many other possible definitions of these pseudovarieties by means of pseudoidentities. For instance, R is also defined by $(xyz)^\omega y = (xyz)^\omega$.

Implicit signatures and ω-terms. An implicit signature is a set of pseudowords over A containing the semigroup multiplication ab, also denoted $\_\cdot\_$. We will mainly work with the signature $\{\_\cdot\_, \_^\omega\}$ consisting of the semigroup multiplication and the unary ω-power. An ω-semigroup is an algebra over the signature $\{\_\cdot\_, \_^\omega\}$. Each finite semigroup has a natural interpretation as an ω-semigroup, by interpreting s^ω as the unique idempotent of the subsemigroup generated by s.

Given an alphabet A, we denote by $\Omega_A^\omega\mathsf{V}$ the V-free ω-semigroup over A, that is, the ω-subsemigroup of $\overline{\Omega}_A\mathsf{S}$ generated by A. An *ω-term over* V is an element of $\Omega_A^\omega\mathsf{V}$. An *$\omega$-identity over* V is a pair of ω-terms over V and an *ω-identity* is an ω-identity over S. We also denote by $u = v$ the ω-identity (u, v).

We call an *ω-term* an element of the free term algebra generated by A over the signature $\{\_\cdot\_, \_^\omega\}$. An ω-term over a pseudovariety V has a (nonunique) representation as an ω-term. Equality of ω-terms is denoted by $\equiv$. Given an ω-term w, its *size* or *length* $|w|$ is defined inductively by $|a| = 1$ for $a \in A$, $|uv| = |u| + |v|$ and $|u^\omega| = |u| + 1$.

All these definitions can be reformulated for the canonical signature $\kappa = \{\_\cdot\_, \_^{\omega-1}\}$ consisting of the semigroup multiplication, and the unary $(\omega-1)$-power. This way, we can define κ-terms and κ-identities (over V), and the V-free κ-semigroup over A, denoted $\Omega_A^\kappa\mathsf{V}$. If V is aperiodic, then any κ-term coincides, in $\overline{\Omega}_A\mathsf{V}$, with the ω-term obtained by replacing all $(\omega-1)$-powers by ω. Since R is aperiodic, our results can also be formulated in terms of the signature κ.

A characterization of equality over R. The following is a simple unique factorization statement for pseudowords which may be considered folklore. A proof is included for the sake of completeness.

Proposition 2.1. *Let $x, y, z, t \in \overline{\Omega}_A\mathsf{S}$ and $a, b \in A$ be such that $xay = zbt$. Suppose that $a \notin c(x)$ and $b \notin c(z)$. If either*

(a) $c(x) = c(z)$, *or*
(b) $c(xa) = c(zb)$,

then $x = z$, $a = b$, and $y = t$.

Proof. Recall that the content $c(x)$ of $x \in \overline{\Omega}_A\mathsf{S}$ is the projection of x into 2^A. By projection into the free left-zero semigroup on 2 letters, we see that an element of $\overline{\Omega}_A\mathsf{S}$ can only have one first letter. If $S \in \mathsf{S}$, then $S^1 \in \mathsf{S}$. In case (a), substituting by 1 all letters of $c(x)$, we obtain $ay' = bt'$ and so $a = b$ by uniqueness of first letters. In case (b), substituting 1 for all letters except a and b, and assuming $a \neq b$, from uniqueness of first letters we conclude that either $a \notin c(z)$ or $b \notin c(x)$, which is in contradiction with (b). Hence in both cases, $a = b$, and (a) holds.

Suppose next that $x \neq z$. Then, there exists a positive integer n and a continuous homomorphism $\varphi : \overline{\Omega}_A\mathsf{S} \to T_n$ into the semigroup of all transformations of $\{1, \ldots, n\}$ (acting on $\{1, \ldots, n\}$ on the right) such that $\varphi(x) \neq \varphi(z)$. Without loss of generality, we may assume that

$$1\varphi(x) = i \neq j = 1\varphi(z) \tag{1}$$

with $\{i, j\} \cap \{2, 3\} = \emptyset$ and that the image under φ of any letter fixes 2 and 3. Since $a \notin c(x) \cup c(z)$, we may redefine $\varphi(a)$ without affecting (1) and we do so by letting $i\varphi(a) = 2$ and $j\varphi(a) = 3$. Then $1\varphi(xay) = 2$ while $1\varphi(zbt) = 3$, in contradiction with the hypothesis that $xay = zbt$. Hence $x = z$.

Finally, suppose that $y \neq t$. Then, for some n, there exists a continuous homomorphism $\varphi : \overline{\Omega}_A\mathsf{S} \to T_n$ such that $1\varphi(y) \neq 1\varphi(t)$ and the image under φ of any letter fixes 2. If we change $\varphi(a)$ so that $2\varphi(a) = 1$, then $2\varphi(xa) = 1$ and so $2\varphi(xay) \neq 2\varphi(xat)$. Hence $y = t$. $\square$

Following Proposition 2.1, we define the *left basic factorization* of $w \in \overline{\Omega}_A\mathsf{S}$ as the unique triple $(w_l, a, w_r) \in \overline{\Omega}_A\mathsf{S}^1 \times A \times \overline{\Omega}_A\mathsf{S}^1$ such that

– $w = w_l a w_r$,
– $c(w_l a) = c(w)$,
– $a \notin c(w_l)$.

We denote by $\mathsf{LBF}(w)$ the left basic factorization of w.

Lemma 2.2. *Let $w \in \Omega_A^\kappa\mathsf{S}$ and let (w_l, a, w_r) be its left basic factorization. Then w_l and w_r are κ-terms (and in particular, they are ω-terms over R).*

Proof. We prove the result by induction on the pair $(c(w), |w|)$ where $2^A \times \mathbb{N}$ is ordered lexicographically. If $w \in A$, the result holds. If $w = x^{\omega-1}$ and if the left basic factorization of x is (x_l, a, x_r) with $x_l, x_r \in \Omega_A^\kappa\mathsf{R}$, then the left basic factorization of w is $(x_l, a, x_r w^2)$, since $w = x^{\omega-1} = x.x^{\omega-2} = x.w^2$.

For $w = xy$, two cases may arise. If $c(x) = c(w)$, let (x_l, a, x_r) be the left basic factorization of x. Then the left basic factorization of w is $(x_l, a, x_r y)$. If $c(x) \neq c(w)$, let (z_ℓ, a_0, y_0) be the left basic factorization of y with $\ell = |c(y)| - 1$. Since $|y| < |w|$, y_0 and z_ℓ are κ-terms. Since $c(z_\ell) \subsetneq c(y)$, one can repeat the argument on z_ℓ to obtain the left basic factorization in κ-terms $z_\ell = (z_{\ell-1}, a_1, y_1)$. An easy decreasing induction gives a factorization $y = a_\ell y_\ell \cdots a_1 y_1 a_0 y_0$, with $y_i \in \Omega_A^\kappa \mathsf{R}$ and where $(a_\ell y_\ell \cdots a_{j+1} y_{j+1}, a_j, y_j)$ is a left basic factorization. Let k be maximal such that $c(w) = c(x \cdot a_\ell y_\ell \cdots a_k y_k a_{k-1})$. Then the left basic factorization of w is $(x \cdot a_\ell y_\ell \cdots a_k y_k, a_{k-1}, y_{k-1} \cdots a_0 y_0)$, which only involves κ-terms. $\square$

Note that this result does adapt for ω-terms. For instance, for $a \in A$, the left basic factorization of $a^\omega \in \Omega_A^\omega \mathsf{S}$ is $(a^{\omega-1}, a, 1)$, and $a^{\omega-1}$ does not belong to $\Omega_A^\omega \mathsf{S}$.

The main argument for the solution of the word problem over R is given in [3] and may be phrased in the form of the following theorem.

Theorem 2.3. *Let $v, w \in \overline{\Omega}_A \mathsf{S}$. Let $v = v_1 a v_2$ and $w = w_1 a w_2$ with $a \notin c(v_1 w_1)$. If $\mathsf{R} \models v = w$, then $\mathsf{R} \models v_1 = w_1$ and $\mathsf{R} \models v_2 = w_2$. Moreover, let (v_l, a, v_r) and (w_l, b, w_r) be the left basic factorizations of v and w, respectively. Then*

$$(\mathsf{R} \models v = w) \iff (\mathsf{R} \models v_l = w_l,\ a = b,\ \text{and}\ \mathsf{R} \models v_r = w_r).$$

By Theorem 2.3, there is a unique factorization of $w \in \overline{\Omega}_A \mathsf{R}$ as a triple $(w_l, a, w_r) \in \overline{\Omega}_A \mathsf{R}^1 \times A \times \overline{\Omega}_A \mathsf{R}^1$ such that $w = w_l a w_r$, $c(w_l a) = c(w)$ and $a \notin c(w_l)$. Further, it follows from Theorem 2.3 that, if $w = p_{\mathsf{R}}(v)$ for a certain $v \in \overline{\Omega}_A \mathsf{S}$ and $\mathsf{LBF}(v) = (v_l, b, v_r)$, then $p_{\mathsf{R}}(v_l) = w_l$, $b = a$, and $p_{\mathsf{R}}(v_r) = w_r$. We will therefore also write $\mathsf{LBF}(w) = (w_l, a, w_r)$ and call this *the left basic factorization of* w.

3. Pseudowords over R and R-automata

A representation of pseudowords of $\overline{\Omega}_A \mathsf{R}$ by trees was given in [7]. Here, we consider an alternative representation by automata over $\{0, 1\}$, whose states are A-labeled. We then prove that two pseudowords over R are equal if and only if their associated automata are equal.

3.1. **Infinite products in pro-R semigroups.** In order to define R-automata, we study infinite products in pro-R semigroups. Given a topological semigroup S and a sequence $(s_n)_{n \geq 0} \in S^{\mathbb{N}}$, we denote by $\prod_{n=0}^{\infty} s_n$ the limit of the sequence $(\prod_{n=0}^{N} s_n)_N$ when N grows to infinity, if this limit exists. In this case, we also say that $\prod_{n=0}^{\infty} s_n$ converges. The following well-known fact follows immediately from [7, Lemma 2.1.1]. We include a proof for the sake of completeness.

Lemma 3.1. *Let S be a pro-R semigroup, and let $(s_n)_{n \geq 0} \in S^{\mathbb{N}}$. Then the infinite product $\prod_{n=0}^{\infty} s_n$ converges.*

Proof. Let $t_k = \prod_{n=0}^{k} s_n$. Since S is pro-R, it suffices to check that for any continuous homomorphism $h : S \to U$ from S into a semigroup $U \in \mathsf{R}$, $h(t_k)$ converges in U. We have $h(t_{k+1}) \preceq_{\mathcal{R}} h(t_k)$, and since U is finite, all $h(t_k)$ except a finite number of them are in the same $\mathcal{R}$-class. Since U is $\mathcal{R}$-trivial, we have $h(t_{k+1}) = h(t_k)$ for k large enough, so the sequence converges in U. $\square$

We will use Lemma 3.1 without reference. We next study the continuity of infinite products in pro-R semigroups.

Remark 3.2. Let S be a pro-R semigroup. Then, the mapping

$$p_S : \quad S^{\mathbb{N}} \longrightarrow S$$
$$(s_n)_{n \geq 0} \longmapsto \prod_{n=0}^{\infty} s_n$$

is not necessarily continuous. For instance, consider $e, t \in S$ such that $e^2 = e$, $t^2 \neq t$, $ete \neq e$, and let $(s_n^{(k)})$ be defined by

$$\begin{cases} s_j^{(k)} = e & \text{if } j \neq k \\ s_k^{(k)} = t. \end{cases}$$

Clearly the sequence $(s_n^{(k)})_k$ converges to $(e, e, e, \ldots)$ but the sequence of products converges to $ete \neq e$.

The following lemma states that infinite products in pro-R semigroups having a content function can sometimes be simplified. We will then exploit this simplification to get the continuity of the infinite product over a restricted set of sequences in such semigroups.

Lemma 3.3. *Let S be a pro-R semigroup with a content function c, and $s,t \in S$ such that $c(s) \supseteq c(t)$. Then $s^\omega t = s^\omega$.*

Proof. Let $\eta : A \to S$ be a generating function with respect to which $c : S \to 2^A$ is a content function. Since c is continuous and the subsemigroup of S generated by $\eta(A)$ is dense in S, we may assume that t belongs to the subsemigroup of S generated by $c(s)$. Moreover, since $s^\omega a = s^\omega b = s^\omega$ implies $s^\omega ab = s^\omega$, we only need to consider $t \in c(s)$ so that there exist $s_1, s_2 \in S$ such that $s = s_1 t s_2$. Hence

$$s^\omega t = (s_1 t s_2)^\omega t = (s_1 t s_2)^\omega = s^\omega,$$

where the middle equality is justified since finite $\mathcal{R}$-trivial semigroups satisfy the pseudoidentity $(xyz)^\omega y = (xyz)^\omega$ and pro-R semigroups are residually in R. □

Now, instead of allowing arbitrary infinite products, we constrain the sequences of products to obtain continuity of infinite products. Let S be a profinite semigroup with a content function c. We denote by $\Delta(S)$ the following subset of $S^{\mathbb{N}}$:

$$\Delta(S) = \Big\{(s_n) \in S^{\mathbb{N}} \mid \forall n \geq 0,\ c(s_n) \supseteq c(s_{n+1})\Big\}.$$

We endow $\Delta(S)$ with the induced product topology, and we let p_S be the restriction of the infinite product to $\Delta(S)$.

$$\begin{aligned} p_S : \quad \Delta(S) &\longrightarrow S \\ (s_n)_{n\geq 0} &\longmapsto \prod_{n=0}^{\infty} s_n. \end{aligned}$$

Proposition 3.4. *Let S be an A-generated pro-R semigroup with content function. Then the mapping p_S from $\Delta(S)$ into S is continuous.*

In the following two statements, we first prove Proposition 3.4 when S is finite.

Lemma 3.5. *Let $S \in \mathsf{R}$ be an A-generated semigroup with a content function c, and let $m = |S|$. Let $B \subseteq A$ and $s_0, s_1, \ldots, s_m \in S$ such that $c(s_i) = B$ for all $0 \leq i \leq m$. Then, there exists an idempotent e such that $c(e) = B$ and $s_0 s_1 \cdots s_m \in Se$.*

Proof. Since $m = |S|$, by the pigeonhole principle, there exist i, j such that $0 \leq i < j \leq m$ and $s_0 \cdots s_i = s_0 \cdots s_j = s_0 \cdots s_i (s_{i+1} \cdots s_j)$. Iterating this equality yields $s_0 \cdots s_m = s_0 \cdots s_i (s_{i+1} \cdots s_j)^\omega s_{j+1} \cdots s_m$.

Since $c(s_{i+1} \cdots s_j) = c(s_{j+1} \cdots s_m)$ and since $S \in \mathsf{R}$ is in particular a pro-R semigroup, we get by Lemma 3.3 $(s_{i+1} \cdots s_j)^\omega (s_{j+1} \cdots s_m) = (s_{i+1} \cdots s_j)^\omega$, so $s_0 \cdots s_m = s_0 \cdots s_i (s_{i+1} \cdots s_j)^\omega$. Therefore, the idempotent $e = (s_{i+1} \cdots s_j)^\omega$ satisfies the claim of the lemma. □

Corollary 3.6. *Let $S \in \mathsf{R}$ be an A-generated semigroup with a content function c. Let $(s_n)_{n\geq 0} \in \Delta(S)$. Then $\prod_{n=0}^{\infty} s_i = \prod_{n=0}^{k_S} s_i$, where $k_S = |S||A| + 1$. In particular, the function p_S from $\Delta(S)$ into S is continuous.*

Proof. By definition of $\Delta(S)$, we have $A \supseteq c(s_0) \supseteq c(s_1) \supseteq \cdots \supseteq c(s_{k_S}) \neq \varnothing$. The choice of k_S and the pigeonhole principle imply that there are $|S|+1$ consecutive s_i's among $s_0, s_1, \ldots, s_{k_S}$, say $s_\ell, \ldots, s_{\ell+|S|}$, having the same content, say B. Hence Lemma 3.5 shows that there exist $s \in S$ and $e \in E(S) \cap c^{-1}(B)$ such that

$$\begin{aligned} \prod_{n=0}^{\infty} s_i &= \prod_{n=0}^{\ell+|S|} s_n \prod_{n=\ell+|S|+1}^{\infty} s_n \\ &= se \prod_{n=\ell+|S|+1}^{\infty} s_n \quad \text{by Lemma 3.5} \\ &= se \quad \text{by Lemma 3.3 as } S \text{ is pro-}\mathsf{R} \text{ and } c(e) \supseteq c\Big(\prod_{n=\ell+|S|+1}^{\infty} s_n\Big) \\ &= \prod_{n=0}^{\ell+|S|} s_n. \end{aligned}$$

The second assertion of the statement is now obvious. □

We now know that p_S is continuous when $S \in \mathsf{R}$. To achieve the proof of Proposition 3.4, we show that this property can be transferred to pro-R semigroups. For that purpose, we use the next result whose proof uses well-known techniques and which is included for the sake of completeness.

Lemma 3.7. *Let S be a profinite semigroup and let $\mathcal{H}$ be a family of continuous homomorphisms from S into finite semigroups. Assume that $\mathcal{H}$ separates points, and that if $h_i : S \to T_i$ belongs to $\mathcal{H}$ for $i = 1, \dots, n$, then the homomorphism $h : S \to T_1 \times \cdots \times T_n$ defined by $h(s) = (h_1(s), \dots, h_n(s))$ also belongs to $\mathcal{H}$. Let C be a closed subset of S. Then we have $C = \bigcap_{h \in \mathcal{H}} h^{-1}h(C)$.*

Proof. The inclusion $C \subseteq \bigcap_{h\in\mathcal{H}} h^{-1}h(C)$ clearly holds. Let now $s \in S \setminus C$. Since $\mathcal{H}$ separates points there exists for each $t \in C$ a homomorphism $h_t \in \mathcal{H}$ (which in fact also depends on s) such that $s \notin h_t^{-1}h_t(t)$. Since C is closed and S is profinite, C is compact, and we may extract from the open cover $C \subseteq \bigcup_{t\in C} h_t^{-1}h_t(t)$ a finite cover $C \subseteq \bigcup_{i=1}^{n} h_{t_i}^{-1}h_{t_i}(t_i)$. Consider the continuous homomorphism $h_s : S \to T_1 \times \cdots \times T_n$ defined by $h_s(x) = (h_{t_1}(x), \dots, h_{t_n}(x))$, where h_{t_i} takes its values in T_i. Then $h_s \in \mathcal{H}$ by the hypothesis of the lemma. If some $t \in C$ were such that $h_s(t) = h_s(s)$ then, for t_i such that $h_{t_i}(t) = h_{t_i}(t_i)$, we would have

$$h_{t_i}(s) = h_{t_i}(t) = h_{t_i}(t_i),$$

in contradiction with the choice of h_{t_i}. Hence $s \notin h_s^{-1}h_s(C)$. Therefore $\bigcap\limits_{h\in\mathcal{H}} h^{-1}h(C) \subseteq \bigcap\limits_{s\in S} h_s^{-1}h_s(C) \subseteq C$. □

Proof of Proposition 3.4. Let S be a pro-R semigroup. Let

$$\mathcal{H} = \{h_U : S \to U \mid U \in \mathsf{R} \text{ and } h_U \text{ is a continuous homomorphism}\}.$$

We know that $\mathcal{H}$ separates points (cf. Subsection 2.2). Let $h_U : S \to U$ be a homomorphism of $\mathcal{H}$. We still denote h_U the homomorphism from $\Delta(S)$ into $\Delta(U)$ induced by h_U. Then, since h_U is a continuous homomorphism, the following diagram commutes:

$$\begin{array}{ccc} \Delta(S) & \xrightarrow{\;p_S\;} & S \\ {\scriptstyle h_U}\big\downarrow & & \big\downarrow{\scriptstyle h_U} \\ \Delta(U) & \xrightarrow[\;p_U\;]{} & U \end{array}$$

Let C be a closed subset of S. We have to show that $p_S^{-1}(C)$ is closed. Now, $\mathcal{H}$ satisfies both hypotheses of Lemma 3.7, which yields $C = \bigcap_{h_U\in\mathcal{H}} h_U^{-1}h_U(C)$, so

$$\begin{aligned} p_S^{-1}(C) &= p_S^{-1}\Big(\bigcap_{h_U\in\mathcal{H}} h_U^{-1}h_U(C)\Big) \\ &= \bigcap_{h_U\in\mathcal{H}} p_S^{-1}h_U^{-1}h_U(C) \\ &= \bigcap_{h_U\in\mathcal{H}} h_U^{-1}p_U^{-1}h_U(C), \quad \text{since the above diagram commutes.} \end{aligned}$$

Since C is closed in a profinite semigroup, it is also compact and therefore, so is its image by the continuous homomorphism h_U. By Corollary 3.6, p_U is continuous so each $h_U^{-1}p_U^{-1}h_U(C)$ is closed. Hence $p_S^{-1}(C)$ is closed. This concludes the proof of Proposition 3.4. □

Proposition 3.4 will be used in the proof of Theorem 3.21 below. We can also use it to define the iterated left basic factorization. Let $w \in \overline{\Omega}_A\mathsf{S}$. Let $v_0 = w$ and define sequences v_i, w_i, a_i as follows: if $v_i \neq \varepsilon$, then $(w_i, a_i, v_{i+1}) = \mathsf{LBF}(v_i)$. If for some $p > 0$, $v_p = \varepsilon$, then we set $\llbracket w \rrbracket = p$. Otherwise, we put $\llbracket w \rrbracket = \infty$. By definition of the left basic factorization, we have the following equality if $\llbracket w \rrbracket < \infty$:

$$w = \prod_{i=0}^{\llbracket w \rrbracket - 1} w_i a_i. \tag{2}$$

When $\llbracket w \rrbracket < \infty$, the right hand side of (2) is called the *iterated left basic factorization (on the right)* of $w \in \overline{\Omega}_A\mathsf{S}$. If $\llbracket w \rrbracket = \infty$, then for each $n \geq 0$, we have the factorization

$$w = \Big(\prod_{i=0}^{n} w_i a_i\Big) \cdot v_{n+1} \tag{3}$$

which can be viewed as an infinite product of a sequence in $\Delta(\overline{\Omega}_A\mathsf{S}^1)$ by padding 1's at the right. Applying p_{R}, by Proposition 3.4 we deduce that every $w \in \overline{\Omega}_A\mathsf{R}$ has a factorization as in (2), even when $\llbracket w \rrbracket = \infty$ (where we take $\infty - 1 = \infty$). One can find in [7] an alternative argument to justify the equality (2) when $\llbracket w \rrbracket$ is infinite.

We denote by $\|w\|$ the maximal integer n in (3) such that all $w_i a_i$ in this factorization have the same content as w. If there is no such maximum, then we set $\|w\| = \infty$. We have by definition $\|w\| \leq \llbracket w \rrbracket$ but, for instance, $\|ab^\omega\| = 1$ while $\llbracket ab^\omega \rrbracket = \infty$. Note that

$$\text{if } c(x) = c(y) \text{ then } \|xy\| \geq \|x\| + \|y\|. \tag{4}$$

This inequality may of course be strict, for instance if $x = aba$ and $y = bab$, we have $\|x\| = \|y\| = 1$ and $\|xy\| = 3$.

The *cumulative content* of $w \in \overline{\Omega}_A\mathsf{S}$, denoted $\vec{c}(w)$ is the set of all letters $a \in A$ such that there is a factorization $w = uv$ with $\|v\| = \infty$ and $a \in c(v)$.

If we work instead with $w \in \overline{\Omega}_A\mathsf{R}$, using left basic factorizations within $\overline{\Omega}_A\mathsf{R}$, we obtain similar notions of *iterated left basic factorization*, $\llbracket w \rrbracket$, $\|w\|$, and $\vec{c}(w)$. In particular, from Theorem 2.3 it follows that, if $v \in \overline{\Omega}_A\mathsf{S}$ is such that $p_\mathsf{R}(v) = w$, then $\llbracket w \rrbracket = \llbracket v \rrbracket$, $\|w\| = \|v\|$ and $\vec{c}(w) = \vec{c}(v)$. Furthermore, by the above remark, (2) holds for $w \in \overline{\Omega}_A\mathsf{R}$, even for $\|w\| = \infty$, and we still call its right hand side the iterated left basic factorization of w.

The next statement uses the functions $\|\cdot\|$ and $\vec{c}(\cdot)$ to characterize idempotents over R.

Proposition 3.8. *The following are equivalent for $w \in \overline{\Omega}_A\mathsf{S}$:*

(a) $\mathsf{R} \models v^2 = v$;
(b) $\|v\| = \infty$;
(c) $c(v) = \vec{c}(v)$.

Proof. The equivalence between (b) and (c) follows directly from the definiton of $\vec{c}(v)$. Let us prove $(a) \Leftrightarrow (b)$. Suppose first that $w = p_\mathsf{R}(v)$ is idempotent. By (4), we have $\|w\| = \|w^2\| \geq 2\|w\|$ which implies that $\|w\| = \infty$. Conversely, suppose that $\|w\| = \infty$, say $w = \prod_{n=0}^{\infty} w_n$ with $c(w_n) = c(w)$. Let $\varphi : \overline{\Omega}_A\mathsf{R} \to S$ be a continuous homomorphism into a finite $\mathcal{R}$-trivial semigroup S with a content function. Then $\varphi(w)$ is an idempotent by Lemmas 3.5 and 3.3. Hence $w^2 = w$ since continuous homomorphisms into finite $\mathcal{R}$-trivial semigroups with content functions suffice to separate points of $\overline{\Omega}_A\mathsf{R}$. □

We proceed to examine further features of the function $w \mapsto \|w\|$.

Lemma 3.9. *Let $w = xy \in \overline{\Omega}_A\mathsf{R}$ with $c(x) \subsetneqq c(w)$. Then $\|w\| \leq \|y\| + 1$.*

Proof. The result is trivial if $\|y\|$ is infinite or if x is empty. Otherwise, proceed inductively on $(|c(y)|, \|y\|)$ under the lexicographic ordering. If $|c(y)| = 1$, say $c(y) = \{a\}$, then $a \notin c(x)$ and $c(x)$ contains a letter $b \neq a$ by assumption, so $\|w\| = 1$.

Assume that $|c(y)| > 1$. Let $y = uv$ where u is minimal such that $c(u) = c(y)$, which means that $u = u'a$, where (u', a, v) is the left basic factorization of y. Write $u = zt$ with xz minimal such that $c(xz) = c(w)$. We have $w = xz \cdot tv$.

If $c(tv) \subsetneqq c(w)$, then $\|w\| = 1$, and the result is trivial. So assume that $c(tv) = c(w)$, so that $\|w\| = \|tv\| + 1$.

By definition of u and v, we have $c(v) \subsetneqq c(y)$ or $\|v\| = \|y\| - 1$. In the first case, we have $\|w\| = 2$ and $\|y\| = 1$. In the other case, we have $c(v) = c(y)$. If $c(t) \subsetneqq c(v)$, we have by induction hypothesis $\|tv\| \leq \|v\| + 1 = \|y\|$, so $\|w\| \leq \|y\| + 1$. If on the contrary $c(t) = c(v)$, then using $c(tv) = c(w)$, we obtain $c(t) = c(v) = c(w)$. Since $c(v) = c(y)$, we get $c(t) = c(y) = c(u)$. Since $u = zt$ is the minimal prefix of y such that $c(u) = c(y)$, t is the minimal prefix of tv such that $c(t) = c(y)$. Therefore, $\|w\| = \|v\| + 2 = \|y\| + 1$. □

Corollary 3.10. *Let $x = x_1 \cdots x_r \in \overline{\Omega}_A\mathsf{S}$. Assume that $c(x_i) \subsetneqq c(x)$ for all $i = 1, \ldots, r$. Then $\|x\| < r$. In particular, $\mathsf{R} \not\models x = x^2$.*

Proof. If $r = 2$, then one can easily verify that $\|x\| = 1$. Otherwise, Lemma 3.9 yields $\|x\| \leq \|x_2 \cdots x_r\| + 1$ and $\|x\| < r$ follows by induction on r. □

3.2. **R-automata and R-trees.** In this subsection, we associate with a pseudoword $w \in \overline{\Omega}_A\mathsf{R}$ a (possibly infinite) $A \cup \{\varepsilon\}$-labeled binary tree $\mathcal{T}(w)$ as follows. Let (w_l, m, w_r) be the left basic factorization of w. The root of $\mathcal{T}(w)$ is labeled by m, and the left and right subtrees are obtained by iterating this construction on w_l and w_r, respectively. For instance, for $w = (ab^\omega a)^\omega$, the left basic factorization of w is (a, b, w_1) with $w_1 = b^\omega a(ab^\omega a)^\omega$. Then, the left basic factorization of w_1 is (b^ω, a, w). We obtain the infinite tree shown in Figure 1, called the R-tree of w. Informally, the word problem over $\overline{\Omega}_A\mathsf{R}$ states that two pseudowords have the same R-tree if and only if they are equal. We formalize this result in this subsection. During the analysis of the algorithm for the word problem of ω-terms (Section 5), we will need a more compact representation of these R-trees, where several vertices may have been identified. For that reason, we define R-automata. In the rest of the paper, we denote by $\mathbb{B}$ the alphabet $\{0, 1\}$.

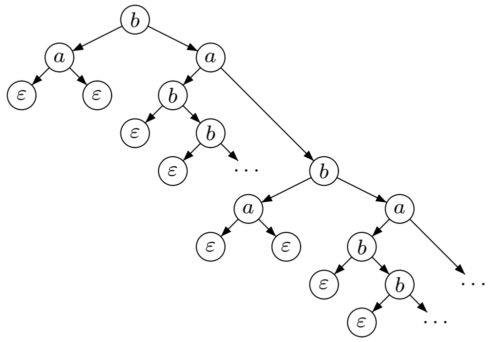


FIGURE 1. The R-tree of the pseudoword $(ab^\omega a)^\omega$

Definition 3.11 (R-automaton)**.** An *A-labeled* R*-automaton* is defined to be a tuple $\mathcal{A} = \langle V, \to, \mathsf{q}, F, \lambda\rangle$ where $\langle V, \to, \mathsf{q}, F\rangle$ is a nonempty (and not necessarily finite) deterministic trim automaton over the alphabet $\mathbb{B} = \{0, 1\}$, and $\lambda : V \to A \cup \{\varepsilon\}$ is a total function. We further require the following conditions.

A.1. The final state set is $F = \lambda^{-1}(\varepsilon)$.
A.2. There is no outgoing transition from F.
A.3. Let $\mathsf{v} \in V \setminus F$. Then both $\mathsf{v}.0$ and $\mathsf{v}.1$ are defined.
A.4. Let $\mathsf{v} \in V \setminus F$. Then

(5) $$\lambda(\mathsf{v}.\mathbb{B}^*) = \lambda(\mathsf{v}.0\mathbb{B}^*) \uplus \{\lambda(\mathsf{v})\}.$$

An R-*tree* is an R-automaton such that every state is reached from the initial state by a unique path.

For an R-tree $\mathcal{A} = \langle V, \to, \mathsf{q}, F, \lambda\rangle$ and $\mathsf{v} \in V$, the *sub-automaton of $\mathcal{A}$ rooted at* v is the R-automaton $\mathcal{A}_\mathsf{v} = \langle \mathsf{v}.\mathbb{B}^*, \to, \mathsf{v}, F \cap \mathsf{v}.\mathbb{B}^*, \lambda\rangle$. If $\mathcal{A}$ is an R-tree, we say *subtree* instead of sub-automaton.

With the convention that 0-transitions go down to the left while 1-transitions go to the right, condition A.4 states that, from any state v, the alphabet labeling the states of the subtree rooted at the left descendant of v is exactly the alphabet labeling the subtree rooted at v minus the label of v. This can be checked on Figure 1, which represents indeed an R-tree.

Definition 3.12. We say that two R-automata $\langle V_i, \to_i, \mathsf{q}_i, F_i, \lambda_i\rangle$ ($i = 0, 1$) are *isomorphic* if there is a bijection $\varphi : V_0 \to V_1$ such that, for all $\mathsf{v} \in V$ and $a \in \mathbb{B}$, $\lambda_1(\varphi(\mathsf{v}).a) = \varphi(\lambda_0(\mathsf{v}.a))$.

We denote by **1** the R-automaton with a single node labeled ε, and by $\mathbb{A}_A$ the set of all A-labeled R-automata except **1**. Observe that (5) implies that if $\mathsf{v}.\alpha$ is defined, then $|\alpha|_0 \leq |A|$: each time we go left, we end up in an R-subtree labeled by a smaller alphabet. Abusing slightly notation, we write $\lambda(\mathcal{A})$ instead of $\lambda(V)$.

Remark 3.13. Let $\mathcal{A}$ be an R-automaton. Consider a loop $\mathsf{p}_0 \xrightarrow{a_1} \mathsf{p}_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} \mathsf{p}_n = \mathsf{p}_0$. Then, $a_i = 1$ for all $i = 1, \ldots, n$. Indeed, if $a_k = 0$ for some k, we would have $\lambda(\mathsf{p}_{k-1}) \in \lambda(\mathsf{p}_{k-1}.0\{0,1\}^*)$, in contradiction with (5).

Definition 3.14 (equivalence of R-automata)**.** Let $k \geq 0$. Two R-automata $\mathcal{A}_i = \langle V_i, \to_i, \mathsf{q}_i, F_i, \lambda_i\rangle$, $i = 0, 1$, are *k-equivalent* if

(6) $$\forall \alpha \in \{0,1\}^*, \quad |\alpha| \leq k \Longrightarrow \lambda_0(\mathsf{q}_0.\alpha) = \lambda_1(\mathsf{q}_1.\alpha).$$

Two R-automata $\mathcal{A}_0$ and $\mathcal{A}_1$ are *equivalent* if they are k-equivalent for all $k \geq 0$. We write $\mathcal{A}_0 \sim_k \mathcal{A}_1$ if $\mathcal{A}_0$ and $\mathcal{A}_1$ are k-equivalent and we let $\sim = \bigcap \sim_k$.

By convention, (6) means that $\lambda_0(\mathsf{q}_0.\alpha)$ and $\lambda_1(\mathsf{q}_1.\alpha)$ are either both defined and equal or both undefined. Figures 1 and 2 give an example of equivalent R-automata (downwards-left edges represent 0-transitions and downwards-right edges indicate 1-transitions).

Fact 3.15. Equivalent R-trees are isomorphic.

Lemma 3.16. *Any* R*-automaton has a unique equivalent* R*-tree.*

Proof. Let $\mathcal{A} = \langle V, \to, \mathsf{q}, F, \lambda\rangle$ be an R-automaton. We define an R-tree $\mathcal{T} = \langle W, \to, \mathsf{p}, G, \nu\rangle$ as follows. Take $W = \{\alpha \in \mathbb{B}^* \mid \mathsf{q}.\alpha \text{ is defined}\}$. The initial state of $\mathcal{T}$ is ε. We set $\nu(\alpha) = \lambda(\mathsf{q}.\alpha)$. The final state set G is $\nu^{-1}(\varepsilon)$. Finally, if $\mathsf{q}.\alpha0$ and $\mathsf{q}.\alpha1$ exist (that is, if $\lambda(\mathsf{q}.\alpha) \neq \varepsilon$), then we define transitions $\alpha \xrightarrow{0} \alpha0$ and $\alpha \xrightarrow{1} \alpha1$

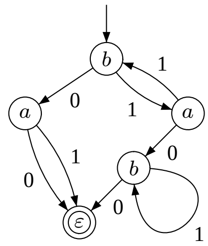


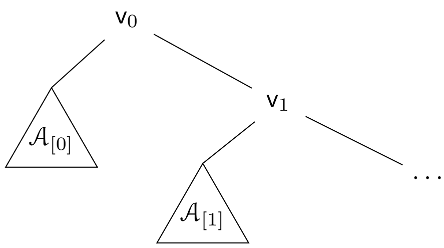
FIGURE 2. The minimal R-automaton of the pseudoword $(ab^\omega a)^\omega$

in $\mathcal{T}$. By definition of ν, properties A.1 to A.4 are transferred from $\mathcal{A}$ to $\mathcal{T}$. The uniqueness is straightforward by Fact 3.15. □

The *unfolding* of $\mathcal{A} \in \mathbb{A}_A$ is the unique R-tree $\vec{\mathcal{A}}$ equivalent to $\mathcal{A}$. If $\mathcal{A} = \langle V, \to, \mathsf{q}, F, \lambda\rangle$ then we write $\vec{\mathcal{A}} = \langle \vec{V}, \to, \vec{\mathsf{q}}, \vec{F}, \lambda\rangle$.

Corollary 3.17. *Let $\mathcal{A}, \mathcal{A}'$ be* R*-automata. Then $\mathcal{A} \sim \mathcal{A}'$ if and only if $\vec{\mathcal{A}} = \vec{\mathcal{A}'}$.*

Let $\|\mathcal{A}\| = \sup\{k \geq 0 \mid \vec{\mathsf{q}}.1^k \text{ is defined}\} \in \mathbb{N} \cup \{\infty\}$. If $\|\mathcal{A}\|$ is finite, then we have $\lambda(\vec{\mathsf{q}}.1^{\|\mathcal{A}\|}) = \varepsilon$. We let $\mathcal{A}_{[i]} = \vec{\mathcal{A}}_{\vec{\mathsf{q}}.1^i0}$ $(0 \leq i \leq \|\mathcal{A}\| - 1)$. The R-tree $\vec{\mathcal{A}}$ is pictured in the following figure.



Definition 3.18 (value of an R-automaton)**.** The *value* $\pi(\mathcal{A}) \in \overline{\Omega}_A\mathsf{R}^1$ of an R-automaton $\mathcal{A}$ is defined inductively on $\lambda(\mathcal{A})$. If $\mathcal{A} = \mathbf{1}$, then $\pi(\mathcal{A}) = 1$. Otherwise,

$$\pi(\mathcal{A}) = \prod_{i=0}^{\|\mathcal{A}\|-1} \pi(\mathcal{A}_{[i]}) \cdot \lambda(\mathsf{q}.1^i). \tag{7}$$

Observe that this correctly defines $\pi(\mathcal{A})$, since by (5), $\lambda(\mathcal{A}_{[i]}) \subsetneqq \lambda(\mathcal{A})$, and since infinite products converge in $\overline{\Omega}_A\mathsf{R}$. Moreover, $c(\pi(\mathcal{A})) = \lambda(\mathcal{A})$. Also note that $\pi(\mathcal{A})$ depends only on $\vec{\mathcal{A}}$, by definition of $\|\mathcal{A}\|$ and $\mathcal{A}_{[i]}$, and since $\lambda(\mathsf{q}.1^i) = \lambda(\vec{\mathsf{q}}.1^i)$.

3.3. **Topology of** R**-automata.** Let $d : \mathbb{A}_A \times \mathbb{A}_A \to \mathbb{R}_+$ be defined by

$$d(\mathcal{A}_1, \mathcal{A}_2) = \begin{cases} 0 & \text{if } \mathcal{A}_1 \sim \mathcal{A}_2 \\ 2^{-r(\mathcal{A}_1, \mathcal{A}_2)} & \text{if } \mathcal{A}_1 \not\sim \mathcal{A}_2, \text{ with } r(\mathcal{A}_1, \mathcal{A}_2) = \min\{k \geq 0 \mid \mathcal{A}_1 \not\sim_k \mathcal{A}_2\}. \end{cases}$$

It is a routine exercise to establish the following observation.

Fact 3.19. The function d is a pseudo-metric such that $d(\mathcal{A}_1, \mathcal{A}_2) = 0$ if and only if $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent.

Hence, d induces a distance over $\mathbb{A}_A/{\sim}$. Abusing notation, we still denote this distance d. Thus, $(\mathbb{A}_A/{\sim}, d)$ is a metric space.

Remark 3.20. (a) Using the finiteness of A, one shows by a standard extraction argument that $(\mathbb{A}_A/{\sim}, d)$ is compact.

(b) The function $\mathbb{A}_A \to \mathbb{A}_A \times A \times \mathbb{A}_A$ which sends the R-automaton $\mathcal{A}$, with root q, to $(\mathcal{A}_{\mathsf{q}.0}, \lambda(\mathsf{q}), \mathcal{A}_{\mathsf{q}.1})$ is continuous.

Since $\pi(\mathcal{A})$ only depends on $\vec{\mathcal{A}}$, we may define $\pi(\mathcal{A}/{\sim}) = \pi(\vec{\mathcal{A}})$. This leads to the following topological representation of $\overline{\Omega}_A\mathsf{R}$.

Theorem 3.21. *The mapping* $\pi : \mathbb{A}_A/\!\sim \longrightarrow \overline{\Omega}_A\mathsf{R}$ *is a homeomorphism.*

Proof. We prove that π is continuous by induction on the size of A. If A is empty, then there is nothing to show. We denote by $\mathbb{A}_A$ the set of R-automata over alphabets of size less than or equal to n. Then π can be factorized as

$$\mathbb{A}_A/\!\sim \xrightarrow{\psi_1} \Big[(\mathbb{A}_A/\!\sim \times A) \cup \{1\}\Big]^{\mathbb{N}} \xrightarrow{\psi_2} \Delta\Big(\overline{\Omega}_A\mathsf{R}^1\Big) \xrightarrow{\psi_3} \overline{\Omega}_A\mathsf{R}^1$$

where, letting $\mathcal{A} = \langle V, \to, \mathsf{q}, F, \lambda\rangle$, the (partial) functions ψ_1, ψ_2 and ψ_3 are defined by

– $\big(\psi_1(\mathcal{A})\big)_i = \begin{cases} \Big(\mathcal{A}_{[i]},\ \lambda(\mathsf{q}.1^i)\Big) & \text{if } 0 \le i \le \|\mathcal{A}\| - 1 \\ 1 & \text{otherwise;} \end{cases}$

– $\psi_2\big((r_i)_{i\ge 0}\big) = (s_i)_{i\ge 0}$, where $s_i = \pi(\mathcal{A}_i)a_i$ if $r_i = (\mathcal{A}_i, a_i)$, and $s_i = 1$ if $r_i = 1$;

– $\psi_3\big((s_i)_{i\ge 0}\big) = \displaystyle\prod_{i=0}^{\infty} s_i$.

We endow $\Big[(\mathbb{A}_A/\!\sim \times A) \cup \{1\}\Big]^{\mathbb{N}}$ and $\Delta\Big(\overline{\Omega}_A\mathsf{R}^1\Big)$ with the product topology. It then suffices to show that ψ_1, ψ_2 and ψ_3 are continuous. Now, by Remark 3.20(b), each component of ψ_1 is continuous. Continuity of ψ_2 follows directly from the induction hypothesis. Finally, the continuity of ψ_3 is given by Proposition 3.4.

We now prove that the function π is injective. Let $\mathcal{A} = \langle V, \to, \mathsf{q}, F, \lambda\rangle$ and $\mathcal{A}' = \langle V', \to', \mathsf{q}', F', \lambda'\rangle$ be such that $\pi(\mathcal{A}) = \pi(\mathcal{A}')$. Since there is a unique R-tree in each equivalence class, we can assume that $\mathcal{A}$ and $\mathcal{A}'$ are R-trees. By (7), we have

$$\prod_{i=0}^{\|\mathcal{A}\|-1} \pi(\mathcal{A}_{[i]}) \cdot \lambda(\mathsf{q}.1^i) = \prod_{i=0}^{\|\mathcal{A}'\|-1} \pi(\mathcal{A}'_{[i]}) \cdot \lambda'(\mathsf{q}'.1^i). \tag{8}$$

Observe that both sides of (8) are precisely iterated left basic factorizations. By Theorem 2.3, this factorization is unique, so $\|\mathcal{A}\| = \|\mathcal{A}'\|$, and for $0 \le i < \|\mathcal{A}\|$, $\pi(\mathcal{A}_{[i]}) = \pi(\mathcal{A}'_{[i]})$ and $\lambda(\mathsf{q}.1^i) = \lambda'(\mathsf{q}'.1^i)$. Since all $\mathcal{A}_{[i]}$ and $\mathcal{A}'_{[i]}$ are R-trees over smaller alphabets, the induction hypothesis gives $\mathcal{A}_{[i]} = \mathcal{A}'_{[i]}$. Hence $\mathcal{A} = \mathcal{A}'$.

We prove that π is surjective. Let $w \in \overline{\Omega}_A\mathsf{R}$. We construct an R-tree $\mathcal{A}$ such that $w = \pi(\mathcal{A})$. We argue by induction on $c(w)$. If $c(w) = \{a\}$, then w is entirely determined by $\llbracket w\rrbracket$, and we take for $\mathcal{A}$ the unique R-tree such that $\lambda(\mathcal{A}) = \{a\}$ and $\|\mathcal{A}\| = \llbracket w\rrbracket$. Otherwise, let $w = \prod_{i=0}^{\llbracket w\rrbracket-1} w_i a_i$ be the iterated left basic factorization of w. By definition, we have $c(w_i) \subsetneq c(w)$ and the induction hypothesis gives R-trees $\mathcal{A}_i = (V_i, \to_i, \mathsf{q}_i, F_i, \lambda_i)$ such that $w_i = \pi(\mathcal{A}_i)$. We construct $\mathcal{A} = \langle V, \to, \mathsf{q}, F, \lambda\rangle$ as follows.

$$V = \begin{cases} \biguplus V_i \uplus \{\mathsf{v}_i \mid i \ge 0\} & \text{if } \llbracket w\rrbracket = \infty \\ \biguplus V_i \uplus \{\mathsf{v}_i \mid 0 \le i \le \llbracket w\rrbracket - 1\} \uplus \{\mathsf{v}_\varepsilon\} & \text{if } \llbracket w\rrbracket \text{ is finite} \end{cases}$$
$$\lambda(\mathsf{v}_i) = a_i, \text{ and } \lambda(\mathsf{v}_\varepsilon) = \varepsilon.$$
$$\mathsf{v}_i.0 = \mathsf{q}_i,$$
$$\mathsf{v}_i.1 = \mathsf{v}_{i+1} \text{ if } i < \llbracket w\rrbracket \text{ and } \mathsf{v}_{\llbracket w\rrbracket-1}.1 = \mathsf{v}_\varepsilon.$$

The labeling and the transitions on V_i are given by those of $\mathcal{A}_i$. It is then straightforward to check that $\mathcal{A}$ is an R-tree such that $\pi(\mathcal{A}) = w$.

To conclude the proof, it remains to observe that the continuity of π^{-1} follows from the compactness of $(\mathbb{A}_A/\!\sim, d)$. □

Let $\mathcal{A}_i = \langle V_i, \to_i, \mathsf{q}_i, F_i, \lambda_i\rangle$, $i = 0, 1$, be R-automata and let $a \in A$ be such that $\lambda(V_1) \subseteq \lambda(V_0) \uplus \{a\}$. We denote by $(\mathcal{A}_0, a, \mathcal{A}_1)$ the R-automaton $\mathcal{A} = \langle V, \to, \mathsf{q}, F, \lambda\rangle$ where $V = V_0 \uplus V_1 \uplus \{\mathsf{q}\}$, with $\lambda(\mathsf{q}) = a$, $\mathsf{q}.0 = \mathsf{q}_0$, $\mathsf{q}.1 = \mathsf{q}_1$, and where the other transitions and labels are given by those of $\mathcal{A}_0$ and $\mathcal{A}_1$.

If $w \in \overline{\Omega}_A\mathsf{S}$, let $\mathcal{T}(w)$ be the R-tree representing $\pi^{-1}(\mathsf{p}_\mathsf{R}(w))$. The proof of Theorem 3.21 shows that, if $\mathsf{LBF}(w) = (w_l, m, w_r)$, then we have $\mathcal{T}(w) = (\mathcal{T}(w_l), m, \mathcal{T}(w_r))$.

By Theorem 3.21, the R-automata $\mathcal{A}$ equivalent to $\mathcal{T}(w)$ are exactly those satisfying $\pi(\mathcal{A}) = w$. If $\pi(\mathcal{A}) = w$, then we say that $\mathcal{A}$ is an R-automaton of w.

3.4. **Wrappings of R-automata.** For an R-automaton $\mathcal{A} = \langle V, \to, \mathsf{q}, F, \lambda\rangle$ and $\mathsf{v} \in V$, we let $[\mathsf{v}] = \pi(\mathcal{A}_\mathsf{v})$.

Lemma 3.22. *Let $\mathcal{A} = \langle V, \to, \mathsf{q}, F, \lambda\rangle$ be an* R*-automaton and let $\mathsf{v} \in V \setminus F$. Then, the left basic factorization of $[\mathsf{v}]$ is $[\mathsf{v}.0] \cdot \lambda(\mathsf{v}) \cdot [\mathsf{v}.1]$. Therefore, by uniqueness of the left basic factorization, we have*

$$[\mathsf{v}_1] = [\mathsf{v}_2] \Longrightarrow \begin{cases} \lambda(\mathsf{v}_1) = \lambda(\mathsf{v}_2) \\ [\mathsf{v}_1.0] = [\mathsf{v}_2.0] \\ [\mathsf{v}_1.1] = [\mathsf{v}_2.1]. \end{cases}$$

Proof. If $\mathcal{A} = \mathbf{1}$, then the result is true. Otherwise, we have by definition $[\mathsf{v}] = [\mathsf{v}.0]\lambda(\mathsf{v})[\mathsf{v}.1]$. Hence the result follows from $c([\mathsf{v}]) = c([\mathsf{v}.0]) \uplus \{\lambda(\mathsf{v})\}$ (by (5)). □

Lemma 3.22 justifies the following definition. The *wrapping* of an R-automaton $\mathcal{A} = \langle V, \to, \mathsf{q}, F, \lambda\rangle$ is the R-automaton $[\mathcal{A}] = \langle [V], \to, [\mathsf{q}], [F], \nu\rangle$ defined by

- $[V] = \{[\mathsf{v}] \mid \mathsf{v} \in V\} \subseteq \overline{\Omega}_A\mathsf{R}$.
- $[\mathsf{v}].0 = [\mathsf{v}.0]$ and $[\mathsf{v}].1 = [\mathsf{v}.1]$.
- Finally, $\nu([\mathsf{v}]) = \lambda(\mathsf{v})$.

Thus, the wrapping of $\mathcal{A}$ is obtained by merging states representing the same pseudoword. For $w \in \overline{\Omega}_A\mathsf{S}$, we define its *wrapped* R*-automaton* as $\mathcal{A}(w) = [\mathcal{T}(w)]$. For instance, the R-automaton of Figure 2 is the wrapped R-automaton of $(ab^\omega a)^\omega$, as we have identified all states representing the same pseudoword.

We define the *value* of a path $\mathsf{q}_0 \xrightarrow{\alpha_0} \mathsf{q}_1 \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_n} \mathsf{q}_{n+1}$ in an R-automaton $\mathcal{A}$ as $\prod_{i=0}^{n}(\lambda(\mathsf{q}_i), \alpha_i) \in (A \times \{0,1\})^*$. The *language* $\mathcal{L}(\mathsf{v}) \subseteq (A \times \{0,1\})^*$ associated with a state v of $\mathcal{A}$ is the set of all values of paths from v to ε, that is, the set of all values of successful paths in $\mathcal{A}_\mathsf{v}$. The language $\mathcal{L}(\mathcal{A})$ associated with $\mathcal{A}$ is the language associated with its root. Finally, the language $\mathcal{L}(w)$ associated with $w \in \overline{\Omega}_A\mathsf{S}$ is $\mathcal{L}(w) = \mathcal{L}(\mathcal{A}(w))$.

Lemma 3.23. *Let $\mathcal{A}_1, \mathcal{A}_2$ be* R*-automata. If $\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)$, then $\vec{\mathcal{A}_1} = \vec{\mathcal{A}_2}$.*

Proof. It suffices to note that for an R-automaton $\mathcal{A}$, $\mathcal{L}(\mathcal{A})$ uniquely determines the set of maximal paths in $\mathcal{A}$, which in turn uniquely determines $\vec{\mathcal{A}}$. □

Proposition 3.24. *Let $v, w \in \overline{\Omega}_A\mathsf{S}$. Then $\mathsf{R} \models v = w \Longleftrightarrow \mathcal{L}(v) = \mathcal{L}(w)$.*

Proof. Assume that $\mathsf{R} \models v = w$. By Theorem 3.21, we have $\mathcal{T}(v) = \mathcal{T}(w)$, hence $\mathcal{A}(v) = \mathcal{A}(w)$ and $\mathcal{L}(v) = \mathcal{L}(w)$. Conversely, if $\mathcal{L}(v) = \mathcal{L}(w)$, then by Lemma 3.23, we have $\mathcal{T}(v) = \mathcal{T}(w)$, and by Theorem 3.21, $\mathsf{R} \models v = w$. □

4. THE WORD PROBLEM FOR ω-TERMS OVER R

4.1. **Tails of pseudowords.** We define in this subsection several types of factors of pseudowords $w \in \overline{\Omega}_A\mathsf{S}$. Let $f_\alpha(w)$ and $m_\alpha(w)$ be defined inductively on the length of $\alpha \in \{0,1\}^*$ as follows.

$$f_\varepsilon(w) = w$$
$$\big(f_{\alpha 0}(w), m_\alpha(w), f_{\alpha 1}(w)\big) \stackrel{\text{def}}{=} \mathsf{LBF}(f_\alpha(w)).$$

The set of R-*factors* of w is

$$\mathcal{F}(w) = \big\{ f_\alpha(w) \mid \alpha \in \{0,1\}^* \big\}.$$

Note that $\mathcal{F}(\mathcal{F}(w)) = \mathcal{F}(w)$, since by definition $f_{\alpha 0} = f_0 \circ f_\alpha$ and $f_{\alpha 1} = f_1 \circ f_\alpha$.

The set of *relative tails* of w is defined by

$$\mathcal{R}(w) = \big\{ f_\alpha(w) \mid \alpha \in \{0,1\}^* 1 \big\} = f_1(\mathcal{F}(w)).$$

Obviously, a relative tail is also an R-factor. Let now $s_\alpha(w)$ be defined inductively on the length of $\alpha \in \{0,1\}^*$ as follows:

$$s_\varepsilon(w) = w \tag{9}$$
$$s_{0\alpha}(w) = s_\alpha\big(f_0(w)\big) \cdot m_\varepsilon(w) \cdot f_1(w) \tag{10}$$
$$s_{1\alpha}(w) = s_\alpha\big(f_1(w)\big). \tag{11}$$

Note that $c(s_\alpha(w)) \subseteq c(w)$. The set of *absolute tails*, or R-*suffixes* of w is defined by

$$\mathcal{S}(w) = \big\{ s_\alpha(w) \mid \alpha \in \{0,1\}^* \big\}.$$

We will need the following technical results further relating f_α with s_β.

Lemma 4.1. *Let w be a pseudoword and let k be a positive integer. Then we have the following equalities of pseudowords:*

$$w = f_{0^k}(w) \cdot m_{0^{k-1}}(w) \cdot s_{0^{k-1}1}(w). \tag{12}$$

$$s_{0^k1}(w) = f_{0^k1}(w) \cdot m_{0^{k-1}}(w) \cdot s_{0^{k-1}1}(w). \tag{13}$$

Proof. Note that, by definition, $s_1 = f_1$. Hence the equality (12) holds for $k = 1$. Similarly, using (10) and (11), a simple calculation shows that (13) holds for $k = 1$. Assume inductively that, for a given $k \geq 1$ and every pseudoword w, the equalities (12) and (13) both hold. By the induction hypothesis (12) and applying left basic factorization to $f_{0^k}(w)$, we deduce that

$$\begin{aligned} w &= f_{0^{k+1}}(w) m_{0^k}(w) f_{0^k1}(w) \cdot m_{0^{k-1}}(w) \cdot s_{0^{k-1}1}(w) \\ &= f_{0^{k+1}}(w) m_{0^k}(w) s_{0^k1}(w), \end{aligned}$$

in view of (13), which establishes (12) for $k+1$. It remains to show (13) for $k+1$. Applying (10) with $\alpha = 0^k1$ and (13) to the pseudoword $f_0(w)$, we obtain

$$\begin{aligned} s_{0^{k+1}1}(w) &= s_{0^k1}(f_0(w)) \cdot m_\varepsilon(w) \cdot f_1(w) \\ &= f_{0^k1}(f_0(w)) m_{0^{k-1}}(f_0(w)) s_{0^{k-1}1}(f_0(w)) \cdot m_\varepsilon(w) f_1(w) \\ &= f_{0^{k+1}1}(w) m_{0^k}(w) \cdot s_{0^{k-1}1}(f_0(w)) m_\varepsilon(w) f_1(w) \\ &= f_{0^{k+1}1}(w) \cdot m_{0^k}(w) \cdot s_{0^k1}(w) \end{aligned}$$

which completes the induction step. □

Lemma 4.2. *Let w be any pseudoword and $\alpha \in \{0,1\}^*$. Then $f_1(s_\alpha(w)) = s_\beta(w)$ for some $\beta \in \{0,1\}^*$.*

Proof. Clearly $f_1(s_\varepsilon(w)) = f_1(w) = s_1(w)$. Proceeding by induction on α, assume that $f_1(s_\alpha(w)) = s_\beta(w)$. Then we have

$$f_1(s_{0\alpha}(w)) = f_1[s_\alpha(f_0(w)) m_\varepsilon(w) f_1(w)]. \tag{14}$$

The expression on the right side of (14) reduces to $f_1(w) = s_1(w)$ in case $c(s_\alpha(f_0(w))) = c(f_0(w))$. Otherwise, we use Lemma 4.1 to pull out from $f_1(w)$ the shortest factor to complete $f_0(s_{0\alpha}(w))$:

$$f_1(w) = f_{10^k}(w) m_{10^{k-1}}(w) s_{0^{k-1}1}(f_1(w)) = f_{10^k}(w) m_{10^{k-1}}(w) s_{10^{k-1}1}(w),$$

so that, for a suitable k, $f_1(s_{0\alpha}(w)) = s_{10^{k-1}1}(w)$. Finally, we have

$$f_1(s_{1\alpha}(w)) = f_1(s_\alpha(f_1(w))) = s_\beta(f_1(w)) = s_{1\beta}(w),$$

which establishes the induction step. □

Again, since the projection in $\overline{\Omega}_A\mathsf{R}$ of the left basic factorization of $w \in \overline{\Omega}_A\mathsf{S}$ gives the left basic factorization of $p_\mathsf{R}(w)$, all constructions and previous factorizations which we derived in this subsection may be applied to pseudowords over S. The following result however does assume aperiodicity.

Corollary 4.3. *Let $x, y \in \overline{\Omega}_A\mathsf{R}$ be such that xy^ω is an idempotent. Then $f_{1^*}(xy^\omega) \subseteq f_{1^*}(x)y^\omega \cup \mathcal{S}(y)y^\omega$.*

Proof. It is sufficient to show that the set $f_{1^*}(x)y^\omega \cup \mathcal{S}(y)y^\omega$ is closed under f_1, since this set contains xy^ω. So pick z in $f_{1^*}(x) \cup \mathcal{S}(y)$.

If $c(z) \subsetneqq c(zy^\omega)$, then there is $k \geq 1$ such that $c(zf_{0^k}(y)m_{0^{k-1}}(y)) = c(zy^\omega)$, and $m_{0^{k-1}}(y) \notin c(z)$. By equality (12) of Lemma 4.1 applied to y, we deduce $\mathsf{LBF}(zy.y^\omega) = (zf_{0^k}(y), m_{0^{k-1}}(y), s_{0^{k-1}1}(y)y^\omega)$. By aperiodicity of R and uniqueness of left basic factorization, we have $\mathsf{LBF}(zy^\omega) = \mathsf{LBF}(zy.y^\omega)$, hence $f_1(zy^\omega) = s_{0^{k-1}1}(y)y^\omega \in \mathcal{S}(y)y^\omega$.

Suppose next that $c(z) = c(y)$ so that $f_1(zy^\omega) = f_1(z)y^\omega$. Then, in case $z \in f_{1^*}(x)$, we have again $f_1(z) \in f_{1^*}(x)$ while, in case $z = s_\alpha(y)$, Lemma 4.2 guarantees that $f_1(z) = f_1(s_\alpha(y)) = s_\beta(y)$ for some $\beta \in \{0,1\}^*$. Hence $f_1(z) \in f_{1^*}(x) \cup \mathcal{S}(y)$. □

4.2. **Several characterizations of ω-terms.** This subsection is devoted to the proof of the following theorem, which gives several characterizations of ω-terms over R and which may be regarded as a sort of periodicity result. It should be compared with [11, Theorem 5.1], which shows similar characterizations for an ordinal word to be represented by an ω-term.

Theorem 4.4. *Let $w \in \overline{\Omega}_A\mathsf{R}$. The following conditions are equivalent:*

(a) *$\mathcal{L}(w)$ is rational.*
(b) *$\mathcal{A}(w)$ is finite.*

(c) *The set* $\{\pi(\mathcal{T}(w)_{\mathsf{v}}) \mid \mathsf{v} \in V\}$ *is finite, where* $\mathcal{T}(w) = \langle V, \to, q, F, \lambda\rangle$.
(d) $\mathcal{F}(w)$ *is finite.*
(e) $\mathcal{R}(w)$ *is finite.*
(f) $\mathcal{S}(w)$ *is finite.*
(g) $w \in \Omega^{\omega}_{A}\mathsf{R}$.
Moreover, if $w \in \Omega^{\omega}_{A}\mathsf{R}$*, then* $|\mathcal{F}(w)| = |\mathcal{A}(w)|$.

We say that an ω-term w is *reduced* if there is no subterm of w of the form $y^{\omega}z$, with $c(z) \subseteq c(y)$, and there is no subterm of the form $(xy^{\omega})^{\omega}$, where x may be empty, and with $c(x) \subseteq c(y)$.

Lemma 4.5. *Let* w *be an* ω*-term which defines an idempotent in* $\overline{\Omega}_A\mathsf{R}$*. Then there exist* ω*-terms* x, y *such that* $w = xy^{\omega}$, $|x| + |y| < |w|$ *and* xy^{ω} *is reduced.*

Proof. The rewriting rules $y^{\omega}z \to y^{\omega}$ if $c(z) \subseteq c(y)$, and $(xy^{\omega}z)^{\omega} \to xy^{\omega}$ if $c(xz) \subseteq c(y)$ do not change the value of an ω-term over R. Moreover, since they decrease the length, they form a Noetherian system. Let v be a reduced ω-term obtained from w by applying rules of this system. Since w is idempotent, so is v. Moreover, $|v| \leq |w|$.

Let $v = x_1 \cdots x_r$ where x_i is either a letter or a term of the form y_i^{ω}. By Corollary 3.10, there exists i such that $x_i = y_i^{\omega}$ and $c(y_i) = c(v)$. Since v is reduced, we have $i = r$. Therefore, v is of the form xy^{ω} (with $x_1 \cdots x_{r-1} = x$ and $x_r = y^{\omega}$). Finally $|x| + |y| < |v| \leq |w|$. □

Proof of Theorem 4.4. $(a) \Leftrightarrow (b)$ From $\mathcal{A}(w)$, one constructs a finite automaton recognizing $\mathcal{L}(w)$ by adding as a first component of any edge label the label of its origin. Conversely, one can transform the minimal automaton of $\mathcal{L} \subseteq (A \times \{0,1\})^*$ into a state-labeled automaton whose associated language is $\mathcal{L}$, by removing the first component from every edge label and labeling the origin state with it. These transformations obviously preserve finiteness.

$(b) \Leftrightarrow (c)$ comes from the definition of $\mathcal{A}(w)$, whose states are the pseudowords $\pi(\mathcal{T}(w)_{\mathsf{v}})$.

$(c) \Leftrightarrow (d)$ follows directly from Lemma 3.22 applied to $\mathcal{T}(w)$.

$(d) \Rightarrow (e)$ is obvious since $\mathcal{R}(w) \subseteq \mathcal{F}(w)$.

$(e) \Rightarrow (f)$ Assume that $\mathcal{R}(w)$ is finite. We prove that $\mathcal{S}(w)$ is also finite by induction on $|A|$. The result is trivial if $|A| = 0$. Otherwise, let $\mathcal{S}_n(w) = \{s_{\alpha}(w) \mid \alpha \in \{0,1\}^n\}$. The inductive definition (9)–(11) of s_{α} gives

$$\begin{aligned}\mathcal{S}_{n+1}(\mathcal{R}(w)) &\subseteq \mathcal{S}_n\big[f_0(\mathcal{R}(w))\big] \cdot A \cdot f_1(\mathcal{R}(w)) \cup \mathcal{S}_n\big[f_1(\mathcal{R}(w))\big] \\ &\subseteq \mathcal{S}\big[f_0(\mathcal{R}(w))\big] \cdot A \cdot \mathcal{R}(w) \cup \mathcal{S}_n\big(\mathcal{R}(w)\big).\end{aligned}$$

Hence, proceeding by induction on n, $\mathcal{S}_n(\mathcal{R}(w))$ is contained in $\mathcal{S}\big[f_0(\mathcal{R}(w))\big] \cdot A \cdot \mathcal{R}(w) \cup \mathcal{R}(w)$ for every n, and therefore so is $\mathcal{S}(\mathcal{R}(w))$. Therefore,

$$\begin{aligned}\mathcal{S}(w) &\subseteq \{w\} \cup \mathcal{S}(f_0(w)) \cdot A \cdot \mathcal{R}(w) \cup \mathcal{S}(\mathcal{R}(w)) \\ &\subseteq \{w\} \cup \Big[\mathcal{S}\big[\{f_0(w)\} \cup f_0(\mathcal{R}(w))\big]\Big] \cdot A \cdot \mathcal{R}(w) \cup \mathcal{R}(w).\end{aligned}$$

To prove the finiteness of $\mathcal{S}(w)$, it remains to show that $\mathcal{S}\big[f_0(w) \cup f_0(\mathcal{R}(w))\big]$ is finite. Let $u \in \{f_0(w)\} \cup f_0(\mathcal{R}(w))$. Since $c(f_0(x)) \subsetneqq c(x)$ for all x, we have $c(u) \subsetneqq c(w)$. Moreover,

$$\mathcal{R}(\mathcal{F}(w)) = f_1(\mathcal{F}(\mathcal{F}(w))) = f_1(\mathcal{F}(w)) = \mathcal{R}(w).$$

In particular, $\mathcal{R}(u) \subseteq \mathcal{R}(w)$, so $\mathcal{R}(u)$ is finite. Hence we can apply the induction hypothesis to u, so $\mathcal{S}(u)$ is finite. Therefore, $\mathcal{S}(w)$ is finite.

$(f) \Rightarrow (g)$ Assume that $\mathcal{S}(w)$ is finite. We prove by induction on $|c(w)|$ that w is an ω-term. For $c(w) = \{a\}$, either $w = a^{\omega}$ or w is a word. Otherwise, let $w = \prod_{i=0}^{[\![w]\!]-1} w_i a_i$ be the iterated left basic factorization of w.

Let $0 \leq j < [\![w]\!]$. We put $v_j = \prod_{i=j}^{[\![w]\!]-1} w_i a_i$. Let $u \in \mathcal{S}(w_j)$. We claim that $ua_jv_{j+1} \in \mathcal{S}(w)$. We have $w_j = f_{1^j0}(w)$, $a_j = m_{1^j}(w) = m_{\varepsilon}(f_{1^j}(w))$ and $v_{j+1} = f_{1^{j+1}}(w)$. Let $u = s_{\alpha}(w_j)$. Then

$$\begin{aligned}ua_jv_{j+1} &= s_{\alpha}(f_{1^j0}(w)) \cdot m_{1^j}(w) \cdot f_{1^{j+1}}(w) \\ &= s_{\alpha}(f_0(f_{1^j}(w))) \cdot m_{\varepsilon}(f_{1^j}(w)) \cdot f_1(f_{1^j}(w)) \\ &= s_{0\alpha}(f_{1^j}(w)) \\ &= s_{1^j0\alpha}(w) \in \mathcal{S}(w).\end{aligned}$$

Let $u, u' \in \mathcal{S}(w_j)$. We have $a_j \notin c(uu')$, so by Theorem 2.3, if $ua_jv_{j+1} = u'a_jv_{j+1}$, then $u = u'$. Hence $u \mapsto ua_jv_{j+1}$ is an injection from $\mathcal{S}(w_j)$ to $\mathcal{S}(w)$. Since $\mathcal{S}(w)$ is finite by assumption, so is $\mathcal{S}(w_j)$.

We have $c(w_j) \subsetneq c(w)$. By the induction hypothesis, all w_j's are ω-terms. This concludes the proof when $[\![w]\!]$ is finite. Assume now that $[\![w]\!]$ is infinite. Let $u_{\ell,k} = \prod_{i=\ell}^{\ell+k-1} w_i a_i$ and $u_i = u_{0,i}$, so that $w = u_i \cdot w_i a_i \cdot v_{i+1}$. By definition, $v_i = f_{1^i}(w) \in \mathcal{S}(w)$, so there exist $\ell \geq 0$ and $k > 0$ such that $v_{\ell+k} = v_\ell = u_{\ell,k} v_{\ell+k}$, so $v_\ell = u_{\ell,k}^\omega v_\ell$. Since $c(v_\ell) = c(v_{\ell+k}) \subseteq c(u_{\ell,k})$, we have $v_\ell = u_{\ell,k}^\omega$. Therefore, $w = u_\ell v_\ell = u_\ell u_{\ell,k}^\omega$, which is an ω-term.

$(g) \Rightarrow (d)$ Let $w \in \Omega_A^\omega \mathsf{R}$. We proceed by induction on $(|c(w)|, |w|)$ under the lexicographic ordering. If $c(w) = \{a\}$, then $\mathcal{F}(w)$ is finite if w is a word or is the set $\{1, a^\omega\}$ if $w \notin A^+$.

Otherwise, we first claim that the set $f_{1^*}(w)$ is finite. Let $w = \prod_{i=0}^{[\![w]\!]-1} w_i a_i$ be the iterated left basic factorization of w. If $\|w\|$ is finite, then one can write $w = w_0 a_0 \cdots w_k a_k v$ with $a_i \in A$, $c(w_i) = c(w) \setminus \{a_i\}$, $c(v) \subsetneq c(w)$. By Lemma 2.2, v is an ω-term. By induction hypothesis, $f_{1^*}(v)$ is finite. Hence, so is $f_{1^*}(w) = f_{1^*}(v) \cup \{w_i a_i \cdots w_k a_k v \mid i \leq k\}$. If on the contrary, $\|w\|$ is infinite, then w is idempotent. By Lemma 4.5 one can write $w = xy^\omega$ with xy^ω reduced and $|x| + |y| < |w|$. Since $|y| < |w|$ and $c(y) \subseteq c(w)$, by the induction hypothesis applied to y we deduce that $\mathcal{F}(y)$ is finite. Since we have already shown that $(d) \Rightarrow (f)$, we conclude that $\mathcal{S}(y)$ is also finite. By Corollary 4.3, we have $f_{1^*}(w) = f_{1^*}(xy^\omega) \subseteq f_{1^*}(x)y^\omega \cup \mathcal{S}(y)y^\omega$ which is finite by the above and the induction hypothesis applied to x. This proves the claim.

Let $\ell \geq 0$ and $k > 0$ be such that $f_{1^{\ell+k}}(w) = f_{1^\ell}(w)$. Then we have the following equalities of pseudowords over R:

$$\begin{aligned} f_{1^\ell}(w) &= w_\ell a_\ell \cdots w_{\ell+k-1} a_{\ell+k-1} f_{1^{\ell+k}}(w) \\ &= w_\ell a_\ell \cdots w_{\ell+k-1} a_{\ell+k-1} f_{1^\ell}(w) \\ &= (w_\ell a_\ell \cdots w_{\ell+k-1} a_{\ell+k-1})^\omega f_{1^\ell}(w) \\ &= (w_\ell a_\ell \cdots w_{\ell+k-1} a_{\ell+k-1})^\omega. \end{aligned}$$

Therefore $w = w_0 a_0 \cdots w_{\ell-1} a_{\ell-1} (w_\ell a_\ell \cdots w_{\ell+k-1} a_{\ell+k-1})^\omega$.

By the expression of w, we see that the set $W = \{w_0, \ldots, w_{\ell+k-1}\}$ contains $f_{1^*0}(w)$. Now, $\mathcal{F}(w) = f_{1^*}(w) \cup \mathcal{F}(f_{1^*0}(w)) \subseteq f_{1^*}(w) \cup \mathcal{F}(W)$. Moreover, W is a finite set of ω-terms, each over a smaller alphabet than w. By the induction hypothesis, $\mathcal{F}(W)$ is finite. Since we already know that $f_{1^*}(w)$ is finite, so is $\mathcal{F}(w)$. □

4.3. **Canonical forms.** Throughout this subsection, we use freely the fact that the left basic factorization of an ω-term produces factors which are κ-terms, hence ω-terms over R, as given by Lemma 2.2.

Consider a finite R-automaton $\mathcal{A} = \langle Q, \to, \mathsf{q}_\varepsilon, F, \lambda \rangle$. For $\alpha \in \{0,1\}^*$, let $\mathsf{q}_\alpha = \mathsf{q}_\varepsilon.\alpha$, when it is defined, and let $Q_\alpha = \mathsf{q}_\alpha.\{0,1\}^*$, $F_\alpha = F \cap Q_\alpha$ and $\mathcal{A}_\alpha = \langle Q_\alpha, \to, \mathsf{q}_\alpha, F_\alpha, \lambda \rangle$.

We associate to $\mathcal{A}$ a possibly empty ω-term $\omega(\mathcal{A})$ by induction on $|Q|$. If $Q = \{\mathsf{q}_\varepsilon\}$, then q_ε is labeled ε and there are no transitions, so we set $\omega(\mathcal{A}) = 1$. Otherwise, we distinguish two cases. If there is no edge $\mathsf{p} \to \mathsf{q}_\varepsilon$, then $\mathcal{A}_0$ and $\mathcal{A}_1$ have fewer states than $\mathcal{A}$. We set $\omega(\mathcal{A}) = \omega(\mathcal{A}_0) \cdot \lambda(\mathsf{q}_\varepsilon) \cdot \omega(\mathcal{A}_1)$. Otherwise, consider a loop $\mathsf{q}_\varepsilon = \mathsf{p}_0 \xrightarrow{a_1} \mathsf{p}_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} \mathsf{p}_n \xrightarrow{a_{n+1}} \mathsf{q}_\varepsilon$. By Remark 3.13, $a_i = 1$ for all $i = 1, \ldots, n+1$. Moreover, $\mathcal{A}_{1^i0}$ has fewer states than $\mathcal{A}$, by (5). We set $\omega(\mathcal{A}) = \left[\prod_{i=0}^n \omega(\mathcal{A}_{1^i0}) \lambda(\mathsf{q}_{1^i})\right]^\omega$.

The *canonical form* $\mathsf{cf}(w)$ of a possibly empty ω-term w is defined to be $\omega(\mathcal{A}(w))$. We say that w *is in canonical form* if $w \equiv \mathsf{cf}(w)$. Observe that $\mathsf{cf}(w)$ is well defined since, by Theorem 4.4, $\mathcal{A}(w)$ is finite. Note also that, like $\mathcal{A}(w)$, $\mathsf{cf}(w)$ only depends on the interpretation of w in $\overline{\Omega}_A \mathsf{R}^1$.

As an example, the R-automaton $\mathcal{A}$ of $w = (ab^\omega a)^\omega$ is given in Figure 2. There is a loop with two edges around q_ε, so $\mathsf{cf}(w) = [\omega(\mathcal{A}_0) \cdot \lambda(\mathsf{q}_\varepsilon) \cdot \omega(\mathcal{A}_{10}) \cdot \lambda(\mathsf{q}_1)]^\omega = [a \cdot b \cdot \omega(\mathcal{A}_{10}) \cdot a]^\omega$. Similarly, there is a loop with a single edge around q_{10}, so $\omega(\mathcal{A}_{10}) = b^\omega$, and finally, $\mathsf{cf}(w) = (abb^\omega a)^\omega$ (hence w is not in canonical form).

We call a factor (in $\overline{\Omega}_A \mathsf{R}$) of the form ua of a pseudoword w *fringy* if $c(ua) = c(w)$ and $a \notin c(u)$. Let w be an ω-term in canonical form. We define recursively an associated ω-term w' by letting:

- $w' = w_1' a w_2$ if $w \equiv w_1 a w_2$ and $w_1 a$ is a fringy factor of w with $a \in A$;
- $w' = v_1' a (v_2 v_1 a)^\omega$ if $w \equiv v^\omega$ and $v \equiv v_1 a v_2$ where $v_1 a$ is a fringy factor of v.

We will need the following technical result.

Lemma 4.6. *Let w be an ω-term in canonical form. Then $\mathsf{R} \models w' = w$ and w' admits a unique factorization of the form*

$$w' \equiv a_1 u_1 a_2 u_2 \cdots a_n u_n \text{ with } c(u_i) \subseteq \{a_1, \ldots, a_i\} \tag{15}$$

where the a_i are the distinct letters that appear in w. Moreover, in this factorization, each u_i is in canonical form.

Proof. Each of the recursion steps in the definition of w' uses the previous recursion steps and perhaps the pseudoidentity $(xy)^\omega = x(yx)^\omega$, which is valid in R. Hence $\mathsf{R} \models w' = w$. Each of those steps also brings out a fringy factor of a left factor of the previous step, which guarantees that in w' all first occurrences of letters are found outside ω-powers. The uniqueness of the factorization follows from the uniqueness of left basic factorizations.

It remains to show that each u_i is in canonical form. Proceeding by induction on $c(w)$, we distinguish the two cases in the definition of w'. In case $w \equiv w_1 a w_2$ and $w_1 a$ is a fringy factor of w with $a \in A$, then both w_1 and w_2 are in canonical form by the definition of canonical form and $w' \equiv w_1' a w_2$. By uniqueness of the left basic factorization, we have $w_1' \equiv a_1 u_1 \cdots a_{n-1} u_{n-1}$, $a = a_n$, and $w_2 \equiv u_n$. Now, it suffices to apply the induction hypothesis to w_1 to conclude that the u_i ($i = 1, \ldots, n-1$) are in canonical form.

In case $w \equiv v^\omega$ with $v \equiv v_1 a v_2$, $a \in A$, and $v_1 a$ a fringy factor of v, we have $w' \equiv v_1' a (v_2 v_1 a)^\omega$, so that $v_1' \equiv a_1 u_1 \cdots a_{n-1} u_{n-1}$, $a = a_n$, and $(v_2 v_1 a)^\omega \equiv u_n$. By definition of canonical form, since we assume w is in canonical form, v_1 is in canonical form and v_2 must admit a factorization $v_2 \equiv z_1 b_1 \cdots z_r b_r$ in fringy factors $z_i b_i$ of v, such that each of the z_i is in canonical form. This implies that $u_n \equiv (v_2 v_1 a)^\omega$ is also in canonical form. The result now follows as in the previous case by applying the induction hypothesis to v_1. □

We call (15) the *left expanded canonical form of* w and denote it by $\mathsf{cf}'(w)$. For instance, we have $((abb^\omega a)^\omega)' = ab(b^\omega aab)^\omega$, and $\mathsf{cf}'((ab^\omega a)^\omega) = \mathsf{cf}'((abb^\omega a)^\omega) = a \cdot 1 \cdot b \cdot (b^\omega aab)^\omega$.

Proposition 4.7. *Let $u_1, \ldots, u_n, v, w$ be ω-terms. Then*

(a) $\mathsf{R} \models w = \mathsf{cf}(w)$.

(b) $\mathsf{R} \models v = w$ *if and only if* $\mathsf{cf}(v) \equiv \mathsf{cf}(w)$.

(c) *If* $c(v) \cap c(w) = \emptyset$ *then* $\mathsf{cf}(vw) \equiv \mathsf{cf}(v)\,\mathsf{cf}'(w)$.

(d) *If w admits a factorization* $w \equiv u_1 \cdots u_m (u_{m+1} \cdots u_n)^\omega$, *where*

- (i) *each of the ω-terms u_i, $1 \leq i \leq n$, is a fringy factor of the product $u_1 \cdots u_n$,*
- (ii) *there exist no integers $k \geq 0$ and ℓ, $1 \leq \ell < n-m$, such that* $\mathsf{R} \models u_{m+1} \cdots u_n = (u_{m+1} \cdots u_{m+\ell})^k$,
- (iii) $\mathsf{R} \not\models u_m = u_n$,

then

$$\mathsf{cf}(w) \equiv \mathsf{cf}(u_1) \cdots \mathsf{cf}(u_m)(\mathsf{cf}(u_{m+1}) \cdots \mathsf{cf}(u_n))^\omega.$$

Proof. Statements (a) and (b) are a direct consequence of the very definition of $\omega(\mathcal{A})$ and $\pi(\mathcal{A})$, and of Theorem 3.21.

Let us show (d). Since u_i is a fringy factor of $u_1 \cdots u_n$, the root r_i of the R-automaton $\mathcal{A}(u_i)$ is not the end of any edge, and the edge labeled 1 from r_i leads to the final state. Consider the R-automaton $\mathcal{B}$ which is obtained from the $\mathcal{A}(u_i)$ by changing the edge labeled 1 from r_i to make it end at r_{i+1}, for $i = 1, \ldots, n-1$, and at r_{m+1} for $i = n$. Then $\mathcal{B}$ is equivalent to the R-automaton $\mathcal{A}(w)$. Moreover, the minimization of $\mathcal{B}$ to obtain the $\mathcal{A}(w)$ is done by identifying only states from different $\mathcal{A}(u_i)$. It does not change the path starting from the root following edges labeled 1, since the hypotheses (ii) and (iii) ensure that the states r_i cannot be identified. The formula for the canonical form $\mathsf{cf}(w)$ now follows directly from the definition.

It remains to prove (c). By (a), (b), and Lemma 4.6, we have

$$\mathsf{R} \models \mathsf{cf}(vw) = \mathsf{cf}(v)\,\mathsf{cf}(w) = \mathsf{cf}(v)\,\mathsf{cf}'(w).$$

Hence it suffices to show that $\mathsf{cf}(v)\,\mathsf{cf}'(w)$ is in canonical form. Let $w' = \mathsf{cf}'(w)$ and consider its factorization of the form (15). Then, by definition of canonical form and since u_n is in canonical form by Lemma 4.6, we have

$$\mathsf{cf}(v a_1 u_1 a_2 u_2 \cdots a_n u_n) \equiv \mathsf{cf}(v a_1 u_1 a_2 u_2 \cdots u_{n-1}) a_n u_n.$$

Now the result follows by induction on n. □

We shall prove in Section 5 that the size of $\mathcal{A}(w)$ is linear in that of w. For our canonical forms, the situation is not so favorable.

Proposition 4.8. *Let w_n be the sequence of ω-terms defined by*

$$w_1 = (a_1^2 b_1^2)^\omega$$
$$w_{n+1} = (w_n a_{n+1}^2 b_{n+1}^2)^\omega.$$

Then w_n has length $5n$ while its canonical form has length $\geq 3^n$, for $n \geq 1$.

Proof. We start by introducing some auxiliary sequences of ω-terms:

(16) $$r_0 = t_0 = 1$$

(17) $$t_1 = (b_1 a_1^2 b_1)^\omega$$

(18) $$r_{n+1} = r_n a_{n+1}^2 b_{n+1} t_{n+1}$$

(19) $$t_{n+1} = (b_{n+1} r_n a_{n+1} \cdot a_{n+1} b_{n+1}^2 r_{n-1} a_n^2 b_n \cdot t_n a_{n+1}^2 b_{n+1})^\omega.$$

For convenience, also let $w_0 = 1$. Let $A_n = \{a_1, b_1, \dots, a_n, b_n\}$. By induction on n, one can easily verify that $c(r_n) = c(t_n) = A_n$. From this observation it follows that each of the ω-terms

$$r_n a_{n+1}^2 b_{n+1},\ b_{n+1} r_n a_{n+1},\ a_{n+1} b_{n+1}^2 r_{n-1} a_n^2 b_n,\ t_n a_{n+1}^2 b_{n+1}$$

has content A_{n+1} but, in each case, dropping the last letter produces an ω-term with smaller content. Combining formulas (18) and (19), we obtain

$$r_{n+1} = r_n a_{n+1}^2 b_{n+1} (b_{n+1} r_n a_{n+1}^2 b_{n+1} \cdot b_{n+1} r_n a_{n+1}^2 b_{n+1})^\omega \ (n \geq 1). \tag{20}$$

We next claim that

$$\mathsf{R} \models w_n = r_n \tag{21}$$

for all $n \geq 0$. This is obvious for $n = 0$. For $n = 1$, using the fact that $\mathsf{R} \models (xy)^\omega = x(yx)^\omega$, we have

$$\mathsf{R} \models w_1 = (a_1^2 b_1^2)^\omega = (a_1^2 b_1 \cdot b_1)^\omega = a_1^2 b_1 (b_1 a_1^2 b_1)^\omega = r_1.$$

Assuming the claim true for a given $n \geq 1$, and using also the fact that $\mathsf{R} \models (x^2)^\omega = x^\omega$, we obtain

$$\begin{aligned}\mathsf{R} \models w_{n+1} &= (w_n a_{n+1}^2 b_{n+1}^2)^\omega = (w_n a_{n+1}^2 b_{n+1} \cdot b_{n+1})^\omega \\ &= w_n a_{n+1}^2 b_{n+1} (b_{n+1} w_n a_{n+1}^2 b_{n+1})^\omega \\ &= w_n a_{n+1}^2 b_{n+1} (b_{n+1} w_n a_{n+1}^2 b_{n+1} \cdot b_{n+1} w_n a_{n+1}^2 b_{n+1})^\omega \\ &= r_{n+1}\end{aligned}$$

in view of (20), which establishes the claim.

The next step consists in proving by induction on n that

$$\mathsf{cf}(r_n) \equiv \mathsf{cf}'(r_n) \equiv r_n \text{ and } \mathsf{cf}(t_n) \equiv t_n, \tag{22}$$

The cases $n \leq 1$ are immediate. One then checks that the hypotheses of Proposition 4.7(d) hold for $m = 0$, $n = 3$, $u_1 = b_{n+1} r_n a_{n+1}$, $u_2 = a_{n+1} b_{n+1}^2 r_{n-1} a_n^2 b_n$ and $u_3 = t_n a_{n+1}^2 b_{n+1}$. Therefore, from the factorization (19), we obtain

$$\mathsf{cf}(t_{n+1}) \equiv (\mathsf{cf}(b_{n+1} r_n a_{n+1}) \cdot \mathsf{cf}(a_{n+1} b_{n+1}^2 r_{n-1} a_n^2 b_n) \cdot \mathsf{cf}(t_n a_{n+1}^2 b_{n+1}))^\omega$$

Assuming (22) for $n - 1$ and n, and using Proposition 4.7 (c), we deduce that

$$\begin{aligned}\mathsf{cf}(t_{n+1}) &\equiv (b_{n+1}\, \mathsf{cf}'(r_n) a_{n+1} \cdot a_{n+1} b_{n+1}^2\, \mathsf{cf}'(r_{n-1}) a_n^2 b_n \cdot \mathsf{cf}(t_n) a_{n+1}^2 b_{n+1})^\omega \\ &\equiv (b_{n+1} r_n a_{n+1} \cdot a_{n+1} b_{n+1}^2 r_{n-1} a_n^2 b_n \cdot t_n a_{n+1}^2 b_{n+1})^\omega \\ &\equiv t_{n+1}.\end{aligned}$$

Similarly, using the factorizations (18) and (19), we obtain

$$\begin{aligned}\mathsf{cf}(r_{n+1}) &\equiv \mathsf{cf}(r_n) a_{n+1}^2 b_{n+1} (b_{n+1} r_n a_{n+1} \cdot a_{n+1} b_{n+1}^2 r_{n-1} a_n^2 b_n \cdot t_n a_{n+1}^2 b_{n+1})^\omega \\ &\equiv r_n a_{n+1}^2 b_{n+1} t_{n+1} \equiv r_{n+1}\end{aligned}$$

and

$$\begin{aligned}\mathsf{cf}'(r_{n+1}) &\equiv \mathsf{cf}'(r_n a_{n+1}^2 b_{n+1} t_{n+1}) \equiv \mathsf{cf}'(r_n a_{n+1}^2) b_{n+1} t_{n+1} \\ &\equiv \mathsf{cf}'(r_n) a_{n+1}^2 b_{n+1} t_{n+1} \equiv r_{n+1}.\end{aligned}$$

This concludes the induction proof of (22). Combining with (21) and Proposition 4.7(b), we obtain the formula $\mathsf{cf}(w_n) \equiv r_n$.

To finish the proof, it remains to compute $|r_n|$. From formulas (16)–(18) and (20), we obtain $|r_1| = 8$ and the recurrence relation $|r_{n+1}| = 3|r_n| + 12$ $(n \geq 1)$, which yields immediately $|r_n| = 14.3^{n-1} - 6$. □

We may also have an exponential decrease in length in the canonical form, even for a reduced ω-term.

Proposition 4.9. *Define a sequence z_n by $z_0 = 1$, $z_{n+1} = (z_n a_n z_n)^\omega$. Then each z_n is a reduced ω-term of length $2^{n+1} - 2$ while its canonical form has length $2n$.*

Proof. Let x_n be the sequence defined by $x_0 = 1$, $x_{n+1} = (x_n a_n)^\omega$. Note that R verifies the following identities:

$$z_{n+1} = (z_n \cdot a_n z_n)^\omega = z_n (a_n z_n z_n)^\omega = z_n (a_n z_n)^\omega = (z_n a_n)^\omega$$

where we use the fact that z_n is an idempotent over R. By Proposition 4.7(d) and (c), we get $\mathsf{cf}(x_{n+1}) = (\mathsf{cf}(x_n) a_n)^\omega$ since $a_n \notin c(x_n)$. By induction on n one now immediately deduces that R satisfies $z_n = x_n$ and that $x_n \equiv \mathsf{cf}(x_n) \equiv \mathsf{cf}(z_n)$. The calculation of the lengths is straightforward. □

One should stress that, although we have defined the canonical form for an ω-term w, the canonical form is by definition determined by the associated wrapped R-automaton $\mathcal{A}(w)$. In the following result, we establish an upper bound for the size of $\mathsf{cf}(w)$ in terms of the size of $\mathcal{A}(w)$. Denote by $|\mathcal{A}|$ the number of states of the R-automaton $\mathcal{A}$.

Proposition 4.10. *Let w be an ω-term over an alphabet A. Then the length of* $\mathsf{cf}(w)$ *is* $O(|\mathcal{A}(w)|^{|A|})$.

Proof. Consider the following number in $[0, +\infty]$:

$$u_n = \sup\left\{\frac{|\mathsf{cf}(w)|}{|\mathcal{A}(w)|^n} : w \text{ is an } \omega\text{-term and } |c(w)| = n\right\}.$$

We show that the sequence $(u_n)_n$ is bounded by 2, which suffices to establish the proposition.

We first note that $u_1 = 1$ by just considering the possibilities for ω-terms of content $\{a\}$: if $w = a^m$ then $|\mathsf{cf}(w)| = |w| = m$ and $|\mathcal{A}(w)| = m + 1$; if w is not a word, then $|\mathsf{cf}(w)| = |a^\omega| = 2$ and $|\mathcal{A}(w)| = 2$.

Suppose that w is an ω-term with $n = |c(w)| > 1$. Let $w = w_0 a_0 \cdots w_k a_k w_{k+1}$ where the w_i are ω-terms and the a_i are letters such that $c(w_i a_i) = c(w)$, and k is as large as possible so that there is a simple path in $\mathcal{A}(w)$ labeled 1^k from the root q. Note that, by definition of the canonical form, in the case where $\|w\|$ is finite, then its value is $k + 1$ and

$$\mathsf{cf}(w) = \mathsf{cf}(w_0)a_0 \cdots \mathsf{cf}(w_k)a_k\,\mathsf{cf}(w_{k+1}); \tag{23}$$

otherwise,

$$\mathsf{cf}(w) = \mathsf{cf}(w_0)a_0 \cdots \mathsf{cf}(w_{i-1})a_{i-1}\big(\mathsf{cf}(w_i)a_i \cdots \mathsf{cf}(w_k)a_k\big)^\omega \tag{24}$$

for some $i \geq 0$. Note also that $\mathcal{A}_{\mathsf{q}1^j0} = \mathcal{A}(w_j)$ for $j = 0, \ldots, k$ and, in the case where $\|w\|$ is finite, $\mathcal{A}_{\mathsf{q}1^k} = \mathcal{A}(w_{k+1})$. By definition of u_{n-1}, we have

$$|\mathsf{cf}(w_j)| \leq u_{n-1}|\mathcal{A}(w_j)|^{n-1} \leq u_{n-1}|\mathcal{A}(w)|^{n-1}$$

for $0 \leq j \leq k$ and also for $j = k + 1$ in case $\|w\|$ is finite. By (23) and (24) and since $k + 2 \leq |\mathcal{A}(w)|$, it follows that, in both cases,

$$\begin{aligned}|\mathsf{cf}(w)| &\leq (k+2)u_{n-1}|\mathcal{A}(w)|^{n-1} + k + 1\\ &\leq u_{n-1}|\mathcal{A}(w)|^n + |\mathcal{A}(w)|.\end{aligned}$$

Hence $u_n \leq u_{n-1} + |\mathcal{A}(w)|^{1-n} \leq u_{n-1} + \frac{1}{2^{n-1}}$. Combining with the fact that $u_1 = 1$, we conclude that $u_n \leq 2$ for all n. □

5. A LINEAR-TIME ALGORITHM COMPUTING WRAPPED R-AUTOMATA

In this section, we solve the word problem for ω-terms over R. Let v and w be two ω-terms, and let (v_l, m_v, v_r) and (w_l, m_w, w_r) be their left basic factorizations, respectively. Since v and w are ω-terms, so are v_l, v_r, w_l, w_r, and they are easy to compute, as well as the letters m_v and m_w. From Theorem 2.3, we know that $\mathsf{R} \models v = w$ is equivalent to $m_v = m_w$, $\mathsf{R} \models v_l = w_l$, and $\mathsf{R} \models v_r = w_r$. To check the last two identities, we could repeat this process inductively, but there is *a priori* no guarantee for it to terminate. Hence, even if the left basic factorization for ω-terms is computable, it does not yield immediately an algorithm checking equality between ω-terms over R.

The above inductive approach consists in fact in computing the R-trees of v and w. It clearly gives a semi-algorithm for deciding whether $v \neq w$ over R. When constructing the R-trees, if we could test whether the value of a subtree has already been produced during the computation, then we would end up with a finite wrapped R-automaton.

To construct the wrapped R-automata of v and w, we will in fact compute intermediate equivalent R-automata, which are not completely wrapped. We call them the R-graphs of v and w. We will then show how to minimize R-graphs in linear time, as already sketched in [8], to obtain the wrapped R-automata of v and w, which we finally compare. The overall complexity of the algorithm is $O(|A|(|v| + |w|))$.

Informal presentation of the algorithm. As explained above, each node v of the R-tree of a pseudoword w can be associated with a pseudoword $[\mathsf{v}]$ over R: if (w_l, m, w_r) is the left basic factorization of w, then the root of $\mathcal{T}(w)$ is associated with w, its left child with w_l and its right child with w_r. If two nodes are associated with the same pseudoword over R, then we obtain the wrapped R-automaton by identifying all subtrees corresponding to the same value, and we know that its finiteness characterizes ω-terms over R (see Theorem 4.4). Given ω-terms v, w, we proceed as follows.

(a) We compute R-automata $\mathcal{G}(v)$ and $\mathcal{G}(w)$ equivalent to $\mathcal{T}(v)$ and $\mathcal{T}(w)$, respectively, which, like $\mathcal{A}(v)$ and $\mathcal{A}(w)$, are finite. These R-automata are called R-graphs. We prove that one can compute them in time $O(|A| \cdot (|v| + |w|))$.

Note that the R-graph $\mathcal{G}(w)$ we shall obtain will not necessarily identify *all* subtrees labeled with a common value. This explains that the R-graphs are not canonical: even if two ω-terms are equal over R, their R-graphs are not necessarily equal. Still, there are enough identifications of isomorphic subtrees to end up with a finite object.

(b) The R-graph $\mathcal{G}(w)$ of w can be transformed in a finite automaton $\mathcal{A}'(w)$ over $A\times\{0,1\}$ such that $\mathsf{R}\models v=w$ if and only if $\mathcal{A}'(v)=\mathcal{A}'(w)$. In fact, $\mathcal{A}'(w)$ is obtained from $\mathcal{A}(w)$ just by assimilating the labels of the states by the labels of the edges.

(c) The automaton $\mathcal{A}'(w)$ can be constructed from w in time $O(|w||c(w)|)$.

(d) From (c) and (b), we deduce that the word problem for two ω-terms v,w of $\Omega_A^\omega\mathsf{R}$ can be solved in time $O(|A|\cdot(|v|+|w|))$.

5.1. **Notation and definitions.** In this subsection, we set up simple but useful notation. Let A be a finite alphabet and let $\mathbb{N}_+=\mathbb{N}\setminus\{0\}$. In order to distinguish occurrences of letters in a word of A^+, we associate to each $x\in A^+$ a word $x_\mathbb{N}\in(A\times\mathbb{N}_+)^+$ containing all original positions of letters of x. To this aim, we define a family of functions $p_k: A^+\to(A\times\mathbb{N}_+)^+$ as follows.

$$\begin{aligned} p_k(a)&=(a,k+1) && \text{for } a\in A,\\ p_k(ay)&=p_k(a)p_{k+1}(y) && \text{for } a\in A \text{ and } y\in A^+. \end{aligned}$$

We let $x_\mathbb{N}=p_0(x)\in(A\times\mathbb{N}_+)^+$. For instance, $aba_\mathbb{N}=(a,1)(b,2)(a,3)$. Abusing notation, we sometimes denote the pair $(a,i)\in A\times\mathbb{N}_+$ by a_i when this will not cause any confusion. Thus, we will also write $aba_\mathbb{N}=a_1b_2a_3$. Finally, we denote by π_A and $\pi_\mathbb{N}$ the projections from $(A\times\mathbb{N})^*$ to A^* and $\mathbb{N}^*$, respectively (here, $\mathbb{N}^*$ means the set of finite sequences of integers, *i.e.*, the free monoid over $\mathbb{N}$). If $B\subseteq A$, we denote by π_B the projection from $(A\times\mathbb{N})^*$ to B^* which acts as π_A on $B\times\mathbb{N}$ and erases letters of $B\setminus A\times\mathbb{N}$. Finally, we let $c_B=c\circ\pi_B$ and $c_\mathbb{N}=c\circ\pi_\mathbb{N}$.

Consider two symbols] and [not belonging to A and let $A_{[]}=A\uplus\{],[\}$. A *well-parenthesized* word over $A_{[]}$ is a word which does not contain [] as a factor and which can be reduced to the empty word ε by the rewriting rules $[\,]\to\varepsilon$ and $a\to\varepsilon$ for $a\in A$. In other terms, the language of well-parenthesized words over $A_{[]}$ is generated by the (non-ambiguous) context-free grammar $S\to[S]S\mid[S]\mid aS\mid a$ ($a\in A$). We say that $x\in(A_{[]}\times\mathbb{N})^+$ is well parenthesized if so is $\pi_{A_{[]}}(x)$. We denote by $\mathsf{Dyck}(A)$ (resp. by $\mathsf{Dyck}(A\times\mathbb{N}_+)$) the language of well-parenthesized words over $A_{[]}$ (resp. over $A_{[]}\times\mathbb{N}_+$).

We define inductively the *tail* $\mathsf{t}_i(x)$ from position $i\in\mathbb{N}$ of a well-parenthesized word x. Let $(x,y)\in\mathsf{Dyck}(A\times\mathbb{N}_+)\times\mathsf{Dyck}(A\times\mathbb{N}_+)^1$ and $i>0$. Then we set

$$\begin{aligned} \mathsf{t}_i(\varepsilon)&=\varepsilon\\ \mathsf{t}_i(xy)&=\mathsf{t}_i(y) && \text{for } i\notin c_\mathbb{N}(x)\\ \mathsf{t}_i(a_iy)&=y && \text{for } a\in A, i\in\mathbb{N}\\ \mathsf{t}_i([_ix]_ly)&=[_ix]_ly\\ \mathsf{t}_i([_kx]_ly)&=\mathsf{t}_i(x)[_kx]_ly && \text{if } k\neq i \text{ and } i\in c_\mathbb{N}(x)\\ \mathsf{t}_i([_kx]_iy)&=y && \text{if } k\neq i \text{ and } i\notin c_\mathbb{N}(x). \end{aligned}$$

The case $i=0$ is special, we set $\mathsf{t}_0(x)=x$. Observe that we do not restrict this definition to words in which a position, like i, occurs at most once in the word. That is, x, for instance, may contain several letters of the form (a,i) for the same i.

We define as well the *prefix up to letter* $a\in A$, $\mathsf{p}_a(x)$, of a well-parenthesized word by setting, for $x,y\in\mathsf{Dyck}(A\times\mathbb{N}_+)$:

$$\begin{aligned} \mathsf{p}_a(\varepsilon)&=\varepsilon\\ \mathsf{p}_a(xy)&=x\mathsf{p}_a(y) && \text{if } a\notin c_A(x)\\ \mathsf{p}_a(a_iy)&=\varepsilon && \text{for } a\in A, i\in\mathbb{N}_+\\ \mathsf{p}_a([_kx]_ly)&=\mathsf{p}_a(x) && \text{if } a\in c_A(x). \end{aligned}$$

The inductive definition immediately yields the following statement.

Fact 5.1. Let $x\in\mathsf{Dyck}(A\times\mathbb{N}_+)^1$, let $a\in A$, and let $i\geq 0$. Then

(a) $\mathsf{t}_i(x)\in\mathsf{Dyck}(A\times\mathbb{N}_+)^1$.

(b) $\mathsf{p}_a(x)\in\mathsf{Dyck}((A\setminus\{a\})\times\mathbb{N}_+)^1$.

For a well-parenthesized word $x \in \mathsf{Dyck}(A \times \mathbb{N}_+)$, a letter $a \in A$, and an integer $i \geq 0$, we let

$$x(i,a) = \mathsf{p}_a(\mathsf{t}_i(x)). \tag{25}$$

For the description of the algorithm, we represent ω-terms by well-parenthesized words by replacing ω-powers by pairs of brackets. To each ω-term $w \in \Omega_A^\omega \mathsf{S}$, we associate $\mathsf{word}(w) \in \mathsf{Dyck}(A)$. Conversely, we associate to $x \in \mathsf{Dyck}(A)$ an ω-term $\mathsf{om}(x)$ such that $\mathsf{om}(\mathsf{word}(w)) = w$. Formally, let $u, v \in \Omega_A^\omega \mathsf{S}$, $x, y \in \mathsf{Dyck}(A)$ and $a \in A$ and put:

$$\begin{array}{rl|rl} \mathsf{word}(a) &= a & \mathsf{om}(a) &= a \\ \mathsf{word}(u \cdot v) &= \mathsf{word}(u)\ \mathsf{word}(v) & \mathsf{om}(xy) &= \mathsf{om}(x)\ \mathsf{om}(y) \\ \mathsf{word}(u^\omega) &= [\mathsf{word}(u)] & \mathsf{om}([x]) &= (\mathsf{om}(x))^\omega. \end{array}$$

It will be convenient to use an end marker $\# \notin A_{[\,]}$. We let $A_\# = A \uplus \{\#\}$, $A_{[\#]} = A_{[\,]} \uplus \{\#\}$, and for an ω-term w on A, we define

$$\overline{w} = (\mathsf{word}(w\#))_{\mathbb{N}} \in \mathsf{Dyck}(A_\# \times \mathbb{N}_+).$$

For instance, $\overline{a(ab)^\omega c} = a_1[_2a_3b_4]_5c_6\#_7$ and $\pi_{A_{[\,]}}(\overline{a(ab)^\omega c}) = a[ab]c$. Finally, let

$$\eta = \mathsf{om} \circ \pi_{\mathsf{A}_{[\,]}} : \mathsf{Dyck}(\mathsf{A}_\# \times \mathbb{N}_+) \to \Omega_{\mathsf{A}}^\omega \mathsf{S}.$$

From the very definitions, we have:

Fact 5.2. Let w be an ω-term, and $x, y \in \mathsf{Dyck}(A \times \mathbb{N}_+)$. Then we have

(a) $\eta(\overline{w}) = w$.
(b) $\eta(xy) = \eta(x)\eta(y)$.
(c) $\eta([_k x]_\ell) = (\eta(x))^\omega$. □

For an ω-term w, we let

$$w(i,a) = \eta(\overline{w}(i,a)).$$

Note that by definition, $w(i,a)$ is an ω-term and $a \notin c(w(i,a))$.

A *marker* of a well-parenthesized word $x \in \mathsf{Dyck}(A \times \mathbb{N}_+)$ is a letter $a_i \in c(x)$ with $a \notin \{\,],[\,\}$ such that x has a factorization $x = ya_iz$, with $a \notin c_A(y)$, and where y and z are (not necessarily well parenthesized) words over $(A_{[\,]} \times \mathbb{N}_+)^*$. For instance a_1 and b_2 are markers of $a_1[_4b_2]_2a_3a_1$ but a_3 is not. Note that there are $|c_A(x)|$ markers in x and that the first occurrence of a marker a_i in x uniquely determines the factorization $x = ya_iz$. The *principal marker* of x is the unique marker a_i of x such that this factorization satisfies $c_A(x) = c_A(ya_i)$.

5.2. The R-graph associated to an ω-term.

In this subsection we define the R-graph $\mathcal{G}(w)$ of an ω-term w. We first need several technical but easy lemmas.

Lemma 5.3. *Let $x \in \mathsf{Dyck}(A \times \mathbb{N}_+)$, and let $a, b \in A$. Then,*

$$b \in c_A(\mathsf{p}_a(x)) \Longrightarrow \mathsf{p}_b(\mathsf{p}_a(x)) = \mathsf{p}_b(x).$$

Proof. Assume that $b \in c_A(\mathsf{p}_a(x))$. Then $a \neq b$ by Fact 5.1 (2). Proceed by induction on $|x|$:

- if $|x| = 1$, then the hypothesis $b \in c_A(\mathsf{p}_a(x))$ cannot hold.
- If $x = yz$ with $y \in \mathsf{Dyck}(A \times \mathbb{N}_+)$ and $a, b \notin c_A(y)$, then we get $\mathsf{p}_b(\mathsf{p}_a(x)) = y \cdot \mathsf{p}_b(\mathsf{p}_a(z))$ and $\mathsf{p}_b(x) = y \cdot \mathsf{p}_b(z)$. Since $b \in c_A(\mathsf{p}_a(x)) \setminus c_A(y) = c_A(y) \cup c_A(\mathsf{p}_a(z)) \setminus c_A(y) \subseteq c_A(\mathsf{p}_a(z))$, the result follows from the induction hypothesis applied to z.
- If $x = a_ix'$, then $\mathsf{p}_a(x) = \varepsilon$, which contradicts $b \in c_A(\mathsf{p}_a(x))$.
- If $x = b_ix'$, then $\mathsf{p}_b(\mathsf{p}_a(x)) = \mathsf{p}_b(x) = \varepsilon$.
- Finally, assume that $x = [_ky]_lz$ with $y \in \mathsf{Dyck}(A \times \mathbb{N}_+)$, and $a \in c_A(y)$ or $b \in c_A(y)$.
  - If $a \in c_A(y)$, then $\mathsf{p}_a(x) = \mathsf{p}_a(y)$, hence $b \in c_A(\mathsf{p}_a(y))$. By induction, $\mathsf{p}_b(\mathsf{p}_a(y)) = \mathsf{p}_b(y)$. Moreover, $b \in c_A(\mathsf{p}_a(y))$, so in particular $b \in c_A(y)$. Therefore, $\mathsf{p}_b(x) = \mathsf{p}_b(y)$. Hence, $\mathsf{p}_b(\mathsf{p}_a(x)) = \mathsf{p}_b(\mathsf{p}_a(y)) = \mathsf{p}_b(y) = \mathsf{p}_b(x)$.
  - If $a \notin c_A(y)$ and $b \in c_A(y)$, then $\mathsf{p}_a(x) = [_ky]_l\mathsf{p}_a(z)$, and $\mathsf{p}_b(x) = \mathsf{p}_b(y)$. Hence, $\mathsf{p}_b(\mathsf{p}_a(x)) = \mathsf{p}_b([_ky]_l\mathsf{p}_a(z)) = \mathsf{p}_b(y) = \mathsf{p}_b(x)$. □

Lemma 5.4. *Let $x \in \mathsf{Dyck}(A \times \mathbb{N}_+)$. If $k \in c_{\mathbb{N}}(\mathsf{p}_a(x))$, then $a \in c_A(\mathsf{t}_k(x))$.*

Proof. If x is a letter, the result is obvious. Otherwise we proceed by induction and distinguish the following cases.

- $x = yz$, $|y|, |z| \geq 1$, and $a \in c_A(y)$. In this case, $\mathsf{p}_a(x) = \mathsf{p}_a(y)$, so $k \in c_{\mathbb{N}}(\mathsf{p}_a(y))$. Since $|y| < |x|$, the induction hypothesis applies to y so $a \in c_A(t_k(y))$. Since $x = yz$ and $k \in c_{\mathbb{N}}(y)$, $t_k(x) = t_k(y)z$, we get $a \in c_A(t_k(x))$.

– If $x = yz$ $|y|, |z| \geq 1$, and $a \in c_A(z) \setminus c_A(y)$. In this case, $\mathsf{p}_a(x) = y\mathsf{p}_a(z)$. If $k \in c_{\mathbb{N}}(y)$, then $\mathsf{t}_k(x) = \mathsf{t}_k(y)z$, and $a \in c_A(z) \subseteq c_A(\mathsf{t}_k(y)z) = c_A(\mathsf{t}_k(x))$. Assume on the contrary that $k \notin c_{\mathbb{N}}(y)$. Since $k \in c_{\mathbb{N}}(\mathsf{p}_a(x)) = c_{\mathbb{N}}(y\mathsf{p}_a(z))$, we get $k \in c_{\mathbb{N}}(\mathsf{p}_a(z))$. The induction hypothesis applied to z yields $a \in c_A(\mathsf{t}_k(z)) = c_A(\mathsf{t}_k(x))$.
– $x = [_i y]_j$. We have $k \in c_{\mathbb{N}}(\mathsf{p}_a(x)) = c_{\mathbb{N}}(\mathsf{p}_a(y))$. Since $|y| < |x|$, the induction hypothesis yields $a \in c_A(\mathsf{t}_k(y))$. Now, $\mathsf{t}_k(x) = \mathsf{t}_k(y)x$, hence $a \in c_A(\mathsf{t}_k(x))$. □

Lemma 5.5. *Let $x \in \mathsf{Dyck}(A \times \mathbb{N}_+)$ and let $k \in c_{\mathbb{N}}(\mathsf{p}_a(x))$. Then we have*

$$\mathsf{t}_k(\mathsf{p}_a(x)) = \mathsf{p}_a(\mathsf{t}_k(x)).$$

Proof. We proceed by induction on $|x|$. Again, we observe that the result holds if x is a letter, and we distinguish the following cases.

– $x = yz$, $a \in c_A(y)$. We then have $\mathsf{p}_a(x) = \mathsf{p}_a(y)$, so $\mathsf{t}_k(\mathsf{p}_a(x)) = \mathsf{t}_k(\mathsf{p}_a(y))$. On the other hand, $k \in c_{\mathbb{N}}(\mathsf{p}_a(x)) = c_{\mathbb{N}}(\mathsf{p}_a(y))$ by hypothesis. Applying the induction hypothesis to y, we get $\mathsf{t}_k(\mathsf{p}_a(y)) = \mathsf{p}_a(\mathsf{t}_k(y))$. Finally, by Lemma 5.4, $a \in c_A(\mathsf{t}_k(y))$ since $k \in c_{\mathbb{N}}(\mathsf{p}_a(x)) = c_{\mathbb{N}}(\mathsf{p}_a(y))$. This justifies the last equality in $\mathsf{p}_a(\mathsf{t}_k(x)) = \mathsf{p}_a(\mathsf{t}_k(yz)) = \mathsf{p}_a(\mathsf{t}_k(y)z) = \mathsf{p}_a(\mathsf{t}_k(y))$. Hence $\mathsf{t}_k(\mathsf{p}_a(x)) = \mathsf{p}_a(\mathsf{t}_k(x))$.
– $x = yz$, $a \in c_A(z) \setminus c_A(y)$. In this case, $\mathsf{p}_a(x) = y\mathsf{p}_a(z)$. If $k \in c_{\mathbb{N}}(y)$, then $\mathsf{t}_k(\mathsf{p}_a(x)) = \mathsf{t}_k(y)\mathsf{p}_a(z)$ and $\mathsf{p}_a(\mathsf{t}_k(x)) = \mathsf{p}_a(\mathsf{t}_k(y)z) = \mathsf{t}_k(y)\mathsf{p}_a(z)$.
  If on the contrary $k \in c_{\mathbb{N}}(\mathsf{p}_a(z)) \setminus c_{\mathbb{N}}(y)$, then $a \in c_A(\mathsf{t}_k(z))$ by Lemma 5.4, $\mathsf{t}_k(\mathsf{p}_a(x)) = \mathsf{t}_k(y\mathsf{p}_a(z)) = \mathsf{t}_k(\mathsf{p}_a(z))$, and $\mathsf{p}_a(\mathsf{t}_k(x)) = \mathsf{p}_a(\mathsf{t}_k(z))$. The induction hypothesis applied to z gives the result.
– $x = [_i y]_j$, $a \in c_A(y)$. Here, $k \in c_{\mathbb{N}}(\mathsf{p}_a(x)) = c_{\mathbb{N}}(\mathsf{p}_a(y))$. By induction hypothesis, we have $\mathsf{t}_k(\mathsf{p}_a(y)) = \mathsf{p}_a(\mathsf{t}_k(y))$. Moreover, $a \in c_A(\mathsf{t}_k(y))$ by Lemma 5.4. Therefore, $\mathsf{t}_k(\mathsf{p}_a(x)) = \mathsf{t}_k(\mathsf{p}_a(y)) = \mathsf{p}_a(\mathsf{t}_k(y)) = \mathsf{p}_a(\mathsf{t}_k(y)[_i y]_j) = \mathsf{p}_a(\mathsf{t}_k(x))$. □

We can apply Lemma 5.5 to a word of the form $\overline{w}$.

Corollary 5.6. *Let w be an ω-term and let $k \in c_{\mathbb{N}}(\mathsf{p}_a(\overline{w}))$. Then we have*

$$\mathsf{t}_k(\mathsf{p}_a(\overline{w})) = \mathsf{p}_a(\mathsf{t}_k(\overline{w})).$$

Lemma 5.7. *Let $x \in \mathsf{Dyck}(A)$ and let $k \in c_{\mathbb{N}}(\mathsf{t}_i(x_{\mathbb{N}}))$. Then $\mathsf{t}_k(\mathsf{t}_i(x_{\mathbb{N}})) = \mathsf{t}_k(x_{\mathbb{N}})$.*

Proof. We proceed by induction on $|x|$. If $x \in A^+$, then the result is trivial.

– If $x_{\mathbb{N}} = yz$, with $y, z \in \mathsf{Dyck}(A \times \mathbb{N}_+)$, then i (resp. k) cannot be in both $c_{\mathbb{N}}(y)$ and $c_{\mathbb{N}}(z)$. Assume that the statement is true for y and z.
  ◦ If $i \in c_{\mathbb{N}}(z)$, then $k \in c_{\mathbb{N}}(\mathsf{t}_i(x_{\mathbb{N}})) = c_{\mathbb{N}}(\mathsf{t}_i(z))$ and by induction hypothesis, $\mathsf{t}_k(\mathsf{t}_i(x_{\mathbb{N}})) = \mathsf{t}_k(\mathsf{t}_i(z)) = \mathsf{t}_k(z) = \mathsf{t}_k(x_{\mathbb{N}})$.
  ◦ If $i, k \in c_{\mathbb{N}}(y)$ then $\mathsf{t}_i(x_{\mathbb{N}}) = \mathsf{t}_i(y)z$, and since $k \notin c_{\mathbb{N}}(z)$, we have $k \in c_{\mathbb{N}}(\mathsf{t}_i(y))$. By induction hypothesis $\mathsf{t}_k(\mathsf{t}_i(y)) = \mathsf{t}_k(y)$. Hence $\mathsf{t}_k(\mathsf{t}_i(x_{\mathbb{N}})) = \mathsf{t}_k(\mathsf{t}_i(yz)) = \mathsf{t}_k(\mathsf{t}_i(y)z) = \mathsf{t}_k(\mathsf{t}_i(y))z = \mathsf{t}_k(y)z = \mathsf{t}_k(yz) = \mathsf{t}_k(x_{\mathbb{N}})$.
  ◦ If $i \in c_{\mathbb{N}}(y)$ and $k \in c_{\mathbb{N}}(z)$, we have $\mathsf{t}_i(x_{\mathbb{N}}) = \mathsf{t}_i(y)z$, and $\mathsf{t}_k(\mathsf{t}_i(x_{\mathbb{N}})) = \mathsf{t}_k(\mathsf{t}_i(y)z) = \mathsf{t}_k(z) = \mathsf{t}_k(x_{\mathbb{N}})$.
– If $x_{\mathbb{N}} = [_1 y]_n$, then if $i = 1$, we have $\mathsf{t}_i(x_{\mathbb{N}}) = x_{\mathbb{N}}$ and so $\mathsf{t}_k(\mathsf{t}_i(x_{\mathbb{N}})) = \mathsf{t}_k(x_{\mathbb{N}})$. Otherwise, we have by definition $\mathsf{t}_i(x_{\mathbb{N}}) = \mathsf{t}_i(y)x_{\mathbb{N}}$ and so $\mathsf{t}_k(\mathsf{t}_i(x_{\mathbb{N}})) = \mathsf{t}_k(\mathsf{t}_i(y)x_{\mathbb{N}})$. Therefore, using the definitions and the induction hypothesis,
  ◦ if $k \in c_{\mathbb{N}}(\mathsf{t}_i(y))$, then $\mathsf{t}_k(\mathsf{t}_i(x_{\mathbb{N}})) = \mathsf{t}_k(\mathsf{t}_i(y))x_{\mathbb{N}} = \mathsf{t}_k(y)x_{\mathbb{N}} = \mathsf{t}_k(x_{\mathbb{N}})$;
  ◦ if $k \notin c_{\mathbb{N}}(\mathsf{t}_i(y))$, then $\mathsf{t}_k(\mathsf{t}_i(x_{\mathbb{N}})) = \mathsf{t}_k(x_{\mathbb{N}})$. □

Lemma 5.8. *Let w be an ω-term, let $i \geq 0$ and $a \in A$. Assume that b_k is a marker of $\overline{w}(i, a)$. Then*
(a) $\mathsf{p}_b(\overline{w}(i, a)) = \overline{w}(i, b)$;
(b) $\mathsf{t}_k(\overline{w}(i, a)) = \overline{w}(k, a)$.

Proof. (a). Let $x = \overline{w}(i, \#)$, so that, by (25), $\overline{w}(i, a) = \mathsf{p}_a(x)$ and $\overline{w}(i, b) = \mathsf{p}_b(x)$. Since b_k is a marker of $\overline{w}(i, a)$, $b \in c_A(\mathsf{p}_a(x))$. By Lemma 5.3, we have $\mathsf{p}_b(\mathsf{p}_a(x)) = \mathsf{p}_b(x)$, that is $\mathsf{p}_b(\overline{w}(i, a)) = \overline{w}(i, b)$.

(b) Since $b_k \in c(\overline{w}(i, a))$, $b \neq a$. We proceed by induction on the construction of w. Also, b_k is the only letter of $c_{\mathbb{N}}^{-1}(k)$ in $\overline{w}$. Hence if $w \in A^*$, both sides of (b) are the factor of $\overline{w}$ starting after b_k and ending before the next letter of $c_A^{-1}(a)$. We have to show $\mathsf{t}_k(\mathsf{p}_a(\mathsf{t}_i(\overline{w}))) = \mathsf{p}_a(\mathsf{t}_k(\bar{w}))$. Since $\overline{w}(i, a)$ contains at least one letter, $i \in c_{\mathbb{N}}(\mathsf{p}_a(\overline{w}))$ and in view of Corollary 5.6, this is equivalent to $\mathsf{t}_k(\mathsf{t}_i(\mathsf{p}_a(\overline{w}))) = \mathsf{t}_k(\mathsf{p}_a(\overline{w}))$. Now, $\mathsf{p}_a(\overline{w})$ is well parenthesized by Fact 5.1 (2), hence the result follows from Lemma 5.7. □

Any word x of the form $\overline{w}(i, a)$ satisfies the following condition:

$$(H(x)) \qquad \forall b, b' \in A, \quad \forall j \in \mathbb{N}_+, \quad (b_j, b'_j \in c(x) \Longrightarrow b = b').$$

Indeed, we have $c(\overline{w}(i,a)) \subseteq c(\overline{w})$, and for each $1 \leq j \leq |\overline{w}|$, there is exactly one letter of $\overline{w}$ belonging to $c_{\mathbb{N}}^{-1}(j)$.

Let Σ be a set of ω-term identities. Recall that an identity $u = v$ is a *consequence* of Σ if it belongs to the fully invariant congruence on the algebra of all ω-terms generated by Σ. This congruence may be described as the equivalence relation generated by all pairs of the form $(s\,\ell\,t, s\,r\,t)$, where s, t are ω-terms and $\ell = r$ is obtained from an identity of Σ by substituting the variables x and y by appropriate ω-terms. We also say that Σ *deduces* $u = v$ and we write $\Sigma \vdash u = v$. The next two statements derive some consequences of $\{t^\omega = t^{\omega+1}\}$.

Lemma 5.9. *Let $x \in \mathsf{Dyck}(A \times \mathbb{N}_+)$ satisfying $(H(x))$ and suppose that a_i is a marker of x. Then*

$$\{t^\omega = t^{\omega+1}\} \vdash \eta(x) = \eta(\mathsf{p}_a(x) \cdot a_i \cdot \mathsf{t}_i(x)). \tag{26}$$

Proof. We proceed by induction on $|x|$. If $|x| = 1$, then $x = a_i$, $\eta(x) = a$, $\mathsf{p}_a(x) = \varepsilon = \mathsf{t}_i(x)$, hence (26) holds.

Otherwise, $x = ya_iz$ with $c_A(y) \subseteq c_A(x) \setminus \{a\}$. If y is well parenthesized, then so is z. By definition of y, $a \notin c_A(y)$ so in this case $\mathsf{p}_a(x) = y$. Furthermore, assume that $i \in c_{\mathbb{N}}(y)$. In this case, there is some letter $b_i \in c(y)$. By $(H(x))$ we would have $a = b$, in contradiction with $a \notin c_A(y)$. Hence, $\mathsf{t}_i(x) = z$. Therefore, (26) can be written $\{t^\omega = t^{\omega+1}\} \vdash \eta(x) = \eta(ya_iz)$, which holds trivially.

Assume now that y is not well parenthesized. One can write $y = y''[_k y'$ and $z = z']_l z''$ such that $y'', z'' \in \mathsf{Dyck}(A \times \mathbb{N}_+)^1$ and $y'a_iz' \in \mathsf{Dyck}(A \times \mathbb{N}_+)$. Let $w = y'a_iz'$. We have $|w| \leq |x| - 2$. Since w is a factor of x, $H(w)$ holds. Hence, we can apply the induction hypothesis to w. Since a_i is a marker of x, we have $a \notin c_A(y)$ hence $a \notin c_A(y')$. Hence a_i is a marker of $w = y'a_iz'$, and by induction hypothesis

$$\{t^\omega = t^{\omega+1}\} \vdash \eta(w) = \eta(\mathsf{p}_a(w) \cdot a_i \cdot \mathsf{t}_i(w)). \tag{27}$$

Since y'' is well parenthesized and $a \in c_A(w)$, $a \notin c_A(y'')$, we also have

$$\mathsf{p}_a(x) = \mathsf{p}_a(y''[_k w]_l z'') = y'' \cdot \mathsf{p}_a(w). \tag{28}$$

In the same way, using $i \notin c_{\mathbb{N}}(y)$ and $i \in c_{\mathbb{N}}(w)$

$$\mathsf{t}_i(x) = \mathsf{t}_i(y''[_k w]_l z'') = \mathsf{t}_i(w)[_k w]_l z''. \tag{29}$$

We now deduce the following sequence of ω-identities from $\{t^\omega = t^{\omega+1}\}$:

$$\begin{aligned}
\{t^\omega = t^{\omega+1}\} \vdash \eta(x) &= \eta(y''[_k w]_l z'') \\
&= \eta(y'') \cdot \eta(w)^\omega \cdot \eta(z'') && \text{as } y'', z'' \in \mathsf{Dyck}(A \times \mathbb{N}_+)^1 \\
&= \eta(y'') \cdot \eta(w) \cdot \eta(w)^\omega \cdot \eta(z'') && \text{using } t^\omega = t^{\omega+1} \\
&= \eta(y'') \cdot \eta(\mathsf{p}_a(w) \cdot a_i \cdot \mathsf{t}_i(w)) \cdot \eta(w)^\omega \cdot \eta(z'') && \text{by (27)} \\
&= \eta(y'') \cdot \eta(\mathsf{p}_a(w)) \cdot a \cdot \eta(\mathsf{t}_i(w)) \cdot \eta(w)^\omega \eta(z'') && \text{by Facts 5.1 and 5.2} \\
&= \eta(y''\mathsf{p}_a(w)) \cdot a \cdot \eta(\mathsf{t}_i(w)[_k w]_l z'') && \text{idem} \\
&= \eta(\mathsf{p}_a(x)) \cdot a \cdot \eta(\mathsf{t}_i(x)) && \text{by (28) and (29)} \\
&= \eta(\mathsf{p}_a(x) \cdot a_i \cdot \mathsf{t}_i(x)).
\end{aligned}$$

□

When applying Lemma 5.9 to words of the form $\overline{w}(i,a)$, we obtain the following formulation.

Corollary 5.10. *Let w be an ω-term. Then for every $i \in c_{\mathbb{N}}(\overline{w})$ and every $a \in c_A(\overline{w})$, we have $\{t^\omega = t^{\omega+1}\} \vdash w(i,a) = w(i,b) \cdot b \cdot w(k,a)$ where b_k is an arbitrary marker of $\overline{w}(i,a)$.*

Proof. Let $x = \overline{w}(i,a)$. Then we know by Lemma 5.8 that $\mathsf{p}_b(x) = \overline{w}(i,b)$ and $\mathsf{t}_k(x) = \overline{w}(k,a)$. We thus have to show that $\{t^\omega = t^{\omega+1}\} \vdash \eta(x) = \eta(\mathsf{p}_b(x)) \cdot b \cdot \eta(\mathsf{t}_k(x))$, that is $\{t^\omega = t^{\omega+1}\} \vdash \eta(x) = \eta(\mathsf{p}_b(x) \cdot b_k \cdot \mathsf{t}_k(x))$. Since any x of the form $\overline{w}(i,a)$ satisfies $(H(x))$, the result follows directly from Lemma 5.9. □

The next variation is the basis to build up the R-graph $\mathcal{G}(w)$.

Corollary 5.11. *Let w be an ω-term. Let $i \in \mathbb{N}$ and $a \in A_\#$. Let b_k be the principal marker of $\overline{w}(i,a)$. Then, the left basic factorization of $w(i,a)$ is $(w(i,b), b, w(k,a))$.*

Proof. Let $x = \overline{w}(i,a)$, so that $x = yb_kz$, with $c_A(y) = c_A(x) \setminus \{b\}$. Since $(H(x))$ holds, by Lemma 5.9 the equation $\eta(x) = \eta(\mathsf{p}_b(x) \cdot b_k \cdot \mathsf{t}_k(x))$ is a consequence of $\{t^\omega = t^{\omega+1}\}$, hence it is valid in R. Since $b \notin c_A(\mathsf{p}_b(x))$, this proves that $(\eta(\mathsf{p}_b(x)), b, \eta(\mathsf{t}_k(x)))$ is the left basic factorization of $\eta(x) = w(i,a)$. It remains to show that

(a) $\mathsf{R} \models \eta(\mathsf{p}_b(\overline{w}(i,a))) = w(i,b)$;
(b) $\mathsf{R} \models \eta(\mathsf{t}_k(\overline{w}(i,a))) = w(k,a)$.

Now, both properties follow from Lemma 5.8. □

In particular, we re-obtain, with the above alternative proof, that there is a finite number of relative/absolute tails for an ω-term over R (which is part of Theorem 4.4):

Corollary 5.12. *Let $w \in \Omega_A^\omega \mathsf{R}$. Then*

(a) *each absolute tail of w is of the form $w(i, \#)$;*

(b) *each relative tail of w is of the form $w(i, a)$, where $a \in A_\#$.*

In particular, there are at most $|\overline{w}||c(w)|$ different tails.

Corollary 5.11 now makes it possible to construct the finite R-graph $\mathcal{G}(w) = (V(w), E(w))$ of w as follows.

- There is one state $\mathsf{q}(i, a)$ for each $(i, a) \in [0, |\mathsf{word}(w)|] \times (A \cup \{\#\})$. The R-subautomaton from $\mathcal{G}(w)$ rooted at state $\mathsf{q}(i, a)$ will be an R-automaton of the ω-term $w(i, a)$.
- The root of $\mathcal{G}(w)$ is $\mathsf{q}(0, \#)$. In the sequel, we will not consider states which cannot be reached from the root.
- Edges of $\mathcal{G}(w)$ are labeled by 0 or 1: $E(w) \subseteq V(w) \times \{0, 1\} \times V(w)$. Let $\mathsf{q}(i, a)$ be a state of $\mathcal{G}(w)$ and let (u, m, v) be the left basic factorization of $w(i, a)$. Let b_j be the principal marker of $\overline{w}(i, a)$. By Corollary 5.11, u is equal to $w(i, b)$ over R, and v is equal to $w(j, a)$ over R. The two outgoing edges from $\mathsf{q}(i, a)$ are $\mathsf{q}(i, a) \xrightarrow{0} \mathsf{q}(i, b)$ and $\mathsf{q}(i, a) \xrightarrow{1} \mathsf{q}(j, a)$.
- Finally, the labeling of states is defined by $\lambda(\mathsf{q}(i, a)) = b$, where the principal marker of $\overline{w}(i, a)$ is of the form b_k, or $\lambda(\mathsf{q}(i, a)) = \varepsilon$ if $w(i, a)$ is empty.

The R-graph of $w = (ab^\omega a)^\omega$ is pictured in Figure 3. We have $\mathsf{word}(w) = [_1 a_2 [_3 b_4]_5 a_6]_7$. The principal marker of $w(0, \#)$ is b_4, so the left son of the root corresponds to $\mathsf{q}(0, b)$ and its right child to $\mathsf{q}(4, \#)$. Inside each state, we have indicated, in addition to the labeling by $\{a, b, \varepsilon\}$, the pair (i, a) corresponding to the ω-term that the state represents. For instance, the root, labeled b, represents $w(0, \#)$.

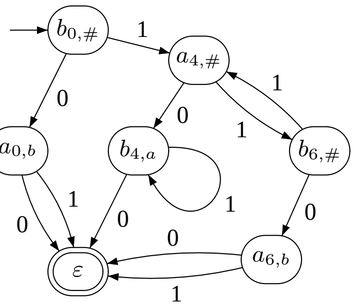


FIGURE 3. The R-graph of the pseudoword $(ab^\omega a)^\omega$

Using Corollary 5.11, we obtain:

Proposition 5.13. *For every ω-term w, $\mathcal{G}(w)$ is an* R*-automaton and is equivalent to $\mathcal{T}(w)$. Moreover, $\mathcal{G}(w)$ is finite, of size $O(|c(w)||w|)$.*

In Figure 3, note that two pairs of states can be identified since $w(0, b) = w(6, b)$ and $w(0, \#) = w(6, \#)$. Merging the states in both pairs produces exactly the wrapped R-automaton (which was shown on Figure 2).

One equivalent way to determine which states have to be merged is to push the labels of states, which are the markers, on edges. We get a graph that we consider as a (usual) automaton $\mathcal{G}'(w)$ on the alphabet $\{0, 1\} \times A$: the state set of $\mathcal{G}'(w)$ is the set of states of $\mathcal{G}(w)$, the initial state of $\mathcal{G}'(w)$ is the root of $\mathcal{G}(w)$, and the transitions are defined as follows. If b_k is the marker of $\overline{w}(i, a)$, then we have two transitions from $\mathsf{q}(i, a)$: $\mathsf{q}(i, a) \xrightarrow{(0,b)} \mathsf{q}(i, b)$ and $\mathsf{q}(i, a) \xrightarrow{(1,b)} \mathsf{q}(k, a)$. If $\mathsf{q}(i, a)$ is labeled by ε, then there is no outgoing transition from that state.

For instance, the automaton $\mathcal{G}'((ab^\omega a)^\omega)$ is shown on Figure 4.

By its definition, the wrapped R-automaton $\mathcal{A}(w)$ of w is obtained from $\mathcal{G}(w) \sim \mathcal{T}(w)$ by identifying states from which one can read the same languages of labeled paths. On the other hand, the minimal automaton $\mathcal{A}'(w)$ of $\mathcal{G}'(w)$ is obtained from $\mathcal{G}'(w)$ by identifying states from which the same language can be read. From that observation, obtaining $\mathcal{A}(w)$ from $\mathcal{A}'(w)$ is again just a matter of transferring letters appearing as the second component of transitions in $\mathcal{A}'(w)$ back to states.

Proposition 5.14. *The wrapped* R*-automaton $\mathcal{A}(w)$ of w is obtained from the minimal automaton $\mathcal{A}'(w)$ of $\mathcal{G}'(w)$ as follows:*

- *$\mathcal{A}(w)$ and $\mathcal{A}'(w)$ only differ by the labeling of the transitions (and the fact that the states of $\mathcal{A}(w)$ are labeled). That is, the state set of $\mathcal{A}'(w)$ is (in bijection with) the set of states of the* R*-automaton $\mathcal{A}(w)$, its initial state is the root of $\mathcal{A}(w)$, its final states are those labeled by ε in $\mathcal{A}(w)$;*

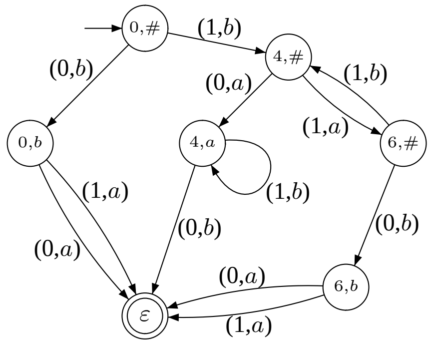


FIGURE 4. The R-graph, viewed as an automaton on $\{0,1\} \times A$, for $(ab^\omega a)^\omega$

– *transitions of* $\mathcal{A}'(w)$ *are obtained as follows from transitions of* $\mathcal{A}(w)$*: for each transition* $\mathsf{v} \xrightarrow{\alpha} \mathsf{w}$ *of* $\mathcal{A}(w)$*, with* $\alpha \in \{0,1\}$*, there is a transition* $\mathsf{v} \xrightarrow{(\alpha,\lambda(\mathsf{v}))} \mathsf{w}$ *in* $\mathcal{A}'(w)$*.*

It is obvious that one can obtain $\mathcal{G}'(w)$ from $\mathcal{G}(w)$ and $\mathcal{A}(w)$ back from $\mathcal{A}'(w)$ in linear time. Therefore, in order to solve the ω-word problem over R in linear time, it remains:

– to compute $\mathcal{G}(w)$ in linear time. This is the purpose of Subsection 5.3;
– to show that $\mathcal{G}'(w)$ can be minimized in linear time. The reason why it works relatively easily is that automata $\mathcal{G}'(w)$ have a special form. For instance, we deduce from Remark 3.13 that all loops are labeled by letters of the form $(1,a)$. The linear-time minimization procedure is the topic of [9], and has been sketched in [8]. For the sake of completeness, we recall briefly the algorithm in Subsection 5.4.

5.3. **Efficient computation of** R**-graphs.** Computing the R-graph of an ω-term w amounts to computing, for each pair (i,a) the principal marker b_k of $w(i,a)$. By definition of $\mathcal{G}(w)$, we know from Corollary 5.11 that the two edges labeled by 0 and 1 from $\mathsf{q}(i,a)$ lead to $\mathsf{q}(i,b)$ and $\mathsf{q}(k,a)$, respectively. In this subsection, we assume w is given and show that one can compute this information in time $O(|w| \cdot |c(w)|)$.

The complication comes from nesting of ω-powers. For instance, let $w = (ae(ba(cacb)^\omega dab)^\omega)^\omega e$ and $i = 9$ (the position of the third a). Then, the principal marker of $w(i,\#)$ is the first occurrence of e, since

$$w(i,\#) = cb(cacb)^\omega(ba(cacb)^\omega dab)^\omega(ae(ba(cacb)^\omega dab)^\omega)^\omega e.$$

Since from a tree representation of w, one can compute $\mathsf{word}(w)$ in time $O(|w|)$, we can assume that the ω-term is readily given by $\mathsf{word}(w)$. We assume that letters of $\mathsf{word}(w)$ are stored in a random access array of size $|\mathsf{word}(w)|$. The i^{th} cell of this array stores an a if and only if the i^{th} letter of $\mathsf{word}(w)$ is a_i.

We assume that letters of A are integers. Even if A is not known, one can rename all letters other than the brackets with names in $\{1,2,\ldots,|c(w)|\}$ in time $O(|\mathsf{word}(w)| \cdot \log|c(w)|) = O(|w| \cdot \log|c(w)|)$, scanning the word once. The factor $O(\log|c(w)|)$ comes from the fact that we must determine for each scanned letter whether it has already been given a new name or not. So we assume that we know $c(w)$ and that we can allocate $c(w)$-indexed arrays.

We define for $x \in (A_{[\#]} \times \mathbb{N})^+$ the sequence $\mathsf{first}(x) \in (A \times \mathbb{N})^*$ of first occurrences of letters in x. Let $x = ya_kz$ with $c_A(y) \cup \{a\} = c_A(x)$ and $a \notin c_A(y)$. Then, $\mathsf{first}(x) = \mathsf{first}(y)a_k$. *E.g.*, $\mathsf{first}(a_1[_2a_3b_4]_5c_6\#_7) = a_1b_4c_6$. Using Algorithm 1, one can compute $\mathsf{first}(x(i,\#))$ for every position $i > 0$ carrying a letter from A (in fact, from $A \cup \{[\,\}$) in $O(|w||c(w)|)$-time.

We do not give a formal proof of the algorithm, which would be very tedious. Instead, we explain in detail how it works. This should convince the reader of its correctness.

We use a standard pseudocode syntax. The argument x of the procedure is assumed to be of the form $\mathsf{word}(w)$. Note that we do not compute $\mathsf{first}(x(0,\#))$, but it is easy to compute afterward in $O(|w||c(w)|)$-time. We did not declare some variables, namely i, j, k, ℓ, `row` and `line`. The variable `row` denotes a list of positions in the interval $[1,|x|]$, and the variable `line` denotes a list of pairs of the form $(i,a) \in [1,|x|] \times A$. The relevant variables are the following:

– i (undeclared) represents the current position,
– `S` is a stack storing the pending opening brackets.
– `wait` is an $|A|$-indexed array, and `wait(`a`)` represents previous positions j (less than the current value of i) for which we did not find the first occurrence of letter a in $x(j,\#)$ yet. Such a position j is still "waiting" for an a.

Algorithm 1 Computes first$(x(i, \#))$ for all $i > 0$ carrying a letter of A

```
    procedure Table_first(x: Word)
    local S:  Stack
    local wait: Array [1..|A|] of lists of positions
    local res: Array [1..|x|] of lists of pairs (i, a)
1:  for i ← 1 to |x| do
2:    if x[i] = '[' then
3:      push(S, i)
4:      for all a ∈ A do
5:        prepend(wait[a], i)                      ▷ [_i is "waiting" for 'a'
6:      end for
7:    else if x[i] = ']' then
8:      matchingOpen ← pop(S)
9:      for all a ∈ A do
10:       if wait[a] ≠ Nil and first(wait[a]) = matchingOpen then
11:         removeFirst(wait[a])
12:       end if
13:     end for
14:     line ← res[matchingOpen]
15:     for k ← 1 to |line| do
16:       row ← wait[letter(line[k])]
17:       wait[letter(line[k])] ← Nil
18:       for ℓ ← 1 to |row| do
19:         append(res[row[ℓ]], line[k])
20:       end for
21:     end for
22:   else                                         ▷ We read a letter from A
23:     row ← wait[x[i]]                           ▷ positions waiting for x[i]
24:     for j ← 1 to |row| do
25:       append (res[row[j]], (i, x[i]))
26:     end for
27:     wait[x[i]] ← Nil
28:     for all a ∈ A do
29:       append (wait[a], i)
30:     end for
31:   end if
32: end for
    end procedure
```

– `res` is the result we should return at the end of the function. It is an array indexed by the positions of x, from 1 (first letter) to $|x|$. At the end of the algorithm, `res[`j`]` contains the list of letters of first$(x(j, \#))$. Letter a_i is represented by the pair (i, a).

The function `letter`, used at lines 16 and 17, extracts from a pair (i, a) the letter a. We also used auxiliary functions on stacks (`push`, `pop`) or lists, like `append`, `prepend`, or `first`, `removeFirst`, and $|.|$ (for the length), whose names are self-explanatory. We denote by `Nil` the empty list.

The algorithm scans x from left to right. Depending on the current letter, it distinguishes 3 cases:

– If the current letter is an opening bracket, the algorithm remembers it by pushing it on the stack `S` (line 3). It puts further at the beginning of each list `wait[`a`]` the position of that opening bracket, to indicate that this position is now "waiting" for an a (lines 4–6).

– If the current letter belongs to A (lines 23–30), it recovers in `wait` the positions which were waiting for the current letter x`[`i`]`, and appends the current letter with its position, $(i, x[i])$, to all those waiting positions in the result `res`. It then resets `wait[`x`[`i`]]` to the empty list (line 27). Finally, it appends the current position i to all lists `wait[`a`]`, since position i, which we have just treated, now waits for all letters to be collected in first$(x(i, \#))$.

– Finally, assume that the current letter is a closing bracket. We first recover the matching opening bracket by popping it off the stack `S` (line 8), and removing it from all lists `wait[a]` (lines 9–13). Due to the fact that letters of A are appended to these lists (lines 28–30) while opening brackets are prepended to it (line 5), we know that if the matching opening bracket occurs in a list `wait[`a`]`, it must in fact be the first element. This is why we can use an $O(1)$ call `removeFirst`, which removes the first element of the list.

  In the case of a closing bracket, it remains to treat the underlying ω-power. We have to take into account that a position inside an ω-power can view, as a first occurrence, a letter which precedes it in x, due to the ω-power. For instance, if $x = [_1 a_2 b_3]_4$, then the first a seen by position 3 is a_2. We recover this information when closing a bracket, here $]_4$. In the example, the first a seen in $x(3, \#)$ is also the first a in $x(1, \#)$. This is general: if a position ℓ inside the ω-power still waits for letter a, the appropriate a is precisely that of $\mathsf{first}(x(\mathtt{matchingOpen}, \#))$, if it exists. Hence, to extend the sequence of first occurrences of letters seen from a position ℓ inside the ω-power, one just needs to add, in order, all letters already appearing in $\mathsf{first}(x(\mathtt{matchingOpen}, \#))$ but not yet appearing in $\mathsf{first}(x(\ell, \#))$. After this operation, one also needs to reset `wait[`a`]` to `Nil`, for all a occurring in $\mathsf{first}(x(\mathtt{matchingOpen}, \#))$. This is exactly what the algorithm does at lines 14–21.

  For instance with $x = [_1 a_2 b_3]_4$, one checks that, when reading $]_4$, positions 2 and 3 are still waiting for a, and position 3 is waiting for a b. The word $\mathsf{first}(x(1, \#))$ seen from the matching opening bracket computed when scanning $]_4$ is $a_2 b_3$. Therefore, we first add a_2 to positions still waiting for an a, that is, 2 and 3: we add a_2 (named $(2, a)$ in the algorithm) to `res[2]` and `res[3]`. Then we reset `wait[`a`]` to `Nil`. Finally we add b_3 to positions waiting for a b similarly and reset `wait[`b`]` to `Nil`.

The algorithm is easily seen to run in $O(|w||c(w)|)$. Therefore:

Lemma 5.15. *Let $w \in \Omega_A^\omega \mathsf{R}$. Algorithm 1 computes in time $O(|w||c(w)|)$ a table giving, for each i such that there exists $a_i \in c(\overline{w}) \cap A \times \mathbb{N}$, the word $\mathsf{first}(w(i, \#))$.*

The $O(|w||c(w)|)$ precomputation of Lemma 5.15 yields an $O(1)$ algorithm for computing both $\xrightarrow{0}$ and $\xrightarrow{1}$ edges from a state of $V(w)$. Indeed, from the word $\mathsf{first}(w(i, \#))$, one can immediately deduce the word $\mathsf{first}(w(i, a))$, which is the largest prefix of $\mathsf{first}(w(i, \#))$ not containing a. Then, if the last letter of $\mathsf{first}(w(i, a))$ is b_k, then, b_k is the principal marker of $\mathsf{first}(w(i, a))$ and we have in $\mathcal{G}(w)$ edges $\mathsf{q}(i, a) \xrightarrow{0} \mathsf{q}(i, b)$ and $\mathsf{q}(i, a) \xrightarrow{1} \mathsf{q}(k, a)$. Since the size of $\mathcal{G}(w)$ is in $O(|w|)$, we obtain the following theorem.

Theorem 5.16. *One can construct $\mathcal{G}(w)$ in time $O(|w|\,|c(w)|)$.*

5.4. **Wrapping and minimization.** The purpose of this subsection is to describe an efficient algorithm to wrap a finite R-automaton. As explained in Subsection 5.2, given an R-automaton $\mathcal{A}$, one can construct a finite automaton recognizing $\mathcal{L}(\mathcal{A})$ by simply adding as a first component of any edge label the label of its origin. By definition of the wrapping, $\mathcal{A}$ is wrapped if and only if this automaton is minimal. Conversely, one can transform the minimal automaton of $\mathcal{L} \subseteq (\{0, 1\} \times A)^*$ into a wrapped state-labeled automaton whose associated language is $\mathcal{L}$ by removing the first component from every edge label and labeling the origin state with it. Through this straightforward translation, finding the wrapping of $\mathcal{A}$ is equivalent to minimizing its associated automaton.

The standard algorithms to minimize a deterministic automaton, such as Hopcroft's one [15] have time complexity $O(mn \log n)$, where $m = |A|$ and n is the number of states. (See [17, 10] for recent presentations and complexity analyses.) For deterministic acyclic automata, Revuz [19] has described an algorithm working in time $O(m + d)$, where d is the number of transitions. It was originally designed to compress dictionaries. A finite R-automaton $\mathcal{A}$ is acyclic if and only if the ω-term it describes does not involve the ω-power, in which case Revuz's algorithm would directly apply to produce the desired wrapping.

An important property of our automata is that their strongly connected components are cycles, that is, any two distinct loops are disjoint. The reason is again that any loop is labeled only by letters of the form $(1, a)$ and that from any state, there is at most one such transition. It is shown in [9] how to minimize in $O(m + d)$-time automata whose strongly connected components are cycles. For the rest of this presentation, we explain the algorithm on R-automata.

Compared with the acyclic case, there is an additional difficulty: in the acyclic version, a height function measuring the longest path starting from each state is computed at the beginning of the algorithm. The situation is then simple, in that the minimization can only identify states having the same height. If we do have cycles, such paths can be infinite. However, since all cycles are disjoint, we can, after a preprocessing phase, treat separately the states belonging to cycles and the other states. A natural analog of Revuz's height function is obtained by letting edges in cycles have weight zero.

The algorithm involves a loop. At each iteration, the first processing stage rolls paths coming to a cycle if this does not change the language. Consider for example a usual automaton with a single initial state q_0, one simple path from q_0 to q_1 labeled v and one cycle around q_1 labeled u, as pictured in Figure 5. If $v = u'u^r$ with $r \geq 0$ and u' a suffix of u, then we do not change the language by rolling the simple path around the cycle, that is, by only retaining the cycle and choosing as the new initial state the unique state q_2 of the cycle such that $\mathsf{q}_2 \cdot v = \mathsf{q}_1$.

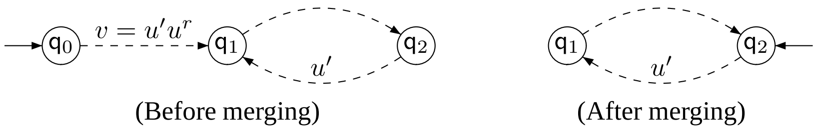


FIGURE 5. Merging a path ending in a cycle

Because of this phenomenon, one cannot compute once for all a height function which would assign weight 0 to edges of cycles and 1 to other edges: an edge which is not in a cycle in the original R-automaton could well be rolled and its weight change from 1 to 0. This is the reason why our height (called *level* in the sequel) is not precomputed. Rather, we compute on the fly the next slice of states we need to treat. In other words, since rolling paths around cycles may change the level of states that lie above them, we have to recompute this level. We do this only locally: we just update correctly levels of states we are about to treat, to remain linear.

The second step in the iteration of the main loop of the algorithm is to minimize cycles one by one. The important point here is that cycles can be represented by (circular) words which take into account the labels (of the states, if we work with R-automata) and the fact that a state is final or not. Minimizing a cycle is then exactly finding the primitive root of this word, which can be performed in linear time with classical pattern-matching algorithms.

The third and last step is to identify, at the current level, all equal cycles and all states not belonging to a cycle. This can be done in linear time (with respect to the size of all cycles and isolates states to be treated) using bucket sort, exactly as in Revuz's algorithm. Here is a more detailed sketch of the algorithm:

(a) Given a finite R-automaton $\mathcal{A}$, compute its strongly connected components with Tarjan's algorithm [22, 13, 10].

(b) Compute an initial *level* function that measures, for each state, the maximum weight of a path to the terminal state, assigning weight 0 to edges in cycles and weight 1 to all other edges. This can be done efficiently by a simple traversal of the graph that is further used to assign a level value to each edge that is not in a cycle, a value which is initialized to the level of the end state plus 1. Both these level functions will be updated in the main loop of the algorithm as a result of rolling paths with all edges labeled 1 around cycles to which they lead. The level of edges serves as a mechanism to propagate to higher levels changes coming from identifications done at lower levels.

(c) From this point on, we construct successive equivalence relations on sets of states which are approximations to the congruence on $\mathcal{A}$ whose quotient determines the minimized R-automaton. We do so level by level, at each stage suitably joining elements into equivalence classes. The first step consists in putting the final state into its own class.

(d) This is the main cycle in the algorithm. Proceed by increasing level $n \geq 1$, as in the following loop. At the end of level n, all states processed in it will have level-value n and they will all be assigned to an equivalence class, which remains unaltered at higher levels.

For each nonterminal state v, denote by $0_\mathsf{v}, 1_\mathsf{v}$ the edges starting from v labeled $0, 1$, respectively. If $\mathsf{level}(0_\mathsf{v}) \leq n$, then let $\zeta(\mathsf{v})$ denote the pair consisting of label $\lambda(\mathsf{v})$ of the state v and the class $[\mathsf{v}0]$ containing the state at the end of the edge 0_v.

(i) Call subroutine $\mathsf{Level}(n)$ which returns the list S of states whose current level-value is n.

(ii) For each state in S which lies in a cycle, put it in its own singleton class.

(iii) Roll 1-labeled paths leading to cycles in S around the corresponding cycles by testing for each successive state v which is not in the cycle whether $\zeta(\mathsf{v})$ is defined and whether it coincides with $\zeta(\mathsf{w})$, where w is the unique state in the cycle such that for all sufficiently large k, $\mathsf{v}1^k = \mathsf{w}1^k$. In the negative case, do not proceed with the test for states u such that $\mathsf{v} \in \mathsf{u}1^*$. In the affirmative case, add v to the class of w, as a result of which the edge 1_v becomes a cycle-edge and thus no longer contributes to the level function; this leads us to reduce $\mathsf{level}(\mathsf{v})$ to n and $\mathsf{level}(e)$ to $n+1$ for every edge e which ends at state v.

(iv) Since the previous step may change the level functions, lowering to level n states that were previously considered at higher levels, we call subroutine $\mathsf{Level}(n)$ again. This will return an updated value for S

which contains the previous value since the previous step only affects the level-values of states at higher levels.

(v) For each cycle C in S, do the following steps which suitably merge all equivalence classes of states in the cycle according to their identification in the minimized R-automaton:

- compute the (circular) word W_C whose letters are the successive $\zeta(\mathsf{w})$ with w in C;
- compute the primitive root W'_C of W_C; this can be done by computing the shortest border u of W_C (*i.e.*, the shortest nonempty word which is both a prefix and a suffix of W_C), such that $u^{-1}W_C$ is also a border; that this computation can be performed in linear time in terms of the length of W_C follows from the fact that the list of all borders can be computed within this time-complexity [14];
- compute the minimal conjugate V_C of W'_C; this can be done in linear time in terms of the length W'_C [12, 21];
- merge classes of states in C according to the periodic repetition of V_C in W_C.

(vi) To merge classes of states in different cycles C of S, start by lexicographically sorting the words V_C using bucket sort [13]. This determines in particular which cycles have the same words $V = V_C$ and their classes associated with corresponding positions in V are merged.

(vii) To merge the remaining states v in S into classes, start by lexicographically sorting (using a bucket sort) their associated triples $(\lambda(\mathsf{v}), [\mathsf{v}_1], [\mathsf{v}_2])$, where $\mathsf{v}_1, \mathsf{v}_2$ denote the ends of the edges $0_\mathsf{v}, 1_\mathsf{v}$, respectively. As in the previous step, this determines in particular which states have the same associated triples, and those that do are merged into the same class.

$(viii)$ Increment n by 1 and proceed until a subroutine call returns the empty list.

To complete the description of the algorithm, it remains to indicate what the subroutine $\mathsf{Level}(n)$ does. It starts by updating the level-value of the beginning state v of each edge e such that $\mathsf{level}(e) = n$ according to the formula

$$\mathsf{level}(\mathsf{v}) = \begin{cases} \max\{\mathsf{level}(e), \mathsf{level}(f)\} & \text{if } e \text{ is not in a cycle and } \{e, f\} = \{0_\mathsf{v}, 1_\mathsf{v}\} \\ \max_x \mathsf{level}(x) & \text{otherwise} \end{cases}$$

where the second maximum runs over all edges x with label 0 which start in the cycle that contains v. Then return all states for which the new level-value is n.

Theorem 5.17. *The above algorithm minimizes a given* R*-automaton with s states in time $O(s)$.*

Since the R-graph $\mathcal{G}(w)$ of an ω-term w can be computed in linear time (Theorem 5.16) and computing the wrapped R-automaton just involves this minimization procedure, we have shown our main result.

Theorem 5.18. *The word problem for ω-terms over* R *can be solved in time $O(mn)$, where m is the number of letters involved and n is the maximum of the lengths of the ω-terms to be tested.*

6. THE EQUATIONAL THEORY OF THE ω-VARIETY GENERATED BY R

Let R^ω be the ω-variety generated by R, that is, the Birkhoff variety generated by all ω-semigroups $(S, \_\cdot\_, \_^\omega)$, where $(S, \_\cdot\_)$ is a finite $\mathcal{R}$-trivial semigroup. By Birkhoff's theorem, R^ω is defined by a set of ω-identities. Let Σ be the following set of ω-identities.

$$(\Sigma) \quad \begin{cases} (xy)^\omega = (xy)^\omega x = (xy)^\omega x^\omega = x(yx)^\omega & (30) \\ (x^\omega)^\omega = x^\omega, & (31) \\ (x^r)^\omega = x^\omega, \qquad r \geq 2. & (32) \end{cases}$$

This section is devoted to the proof of the following theorem.

Theorem 6.1. (a) *The set Σ is a basis for $\mathcal{R}$.*
(b) *The ω-variety $\mathcal{R}$ has no finite basis of identities.*

The rest of this section is devoted to the proof of Theorem 6.1. Assuming (a), we will first prove (b) which is easier. First note that one can deduce aperiodicity from Σ.

Fact 6.2. By (30) and (32), one obtains $\Sigma \vdash x^\omega = (xx)^\omega = (xx)^\omega x = x^{\omega+1}$.

Combining Corollary 5.10, fact 6.2 we obtain the following statement.

Corollary 6.3. *Let w be an ω-term, let $i \in c_\mathbb{N}(\overline{w})$, $a \in c_A(\overline{w})$ and let b_k be a marker of $\overline{w}(i, a)$. Then*

$$\Sigma \vdash w(i, a) = w(i, b) \cdot b \cdot w(k, a). \quad \square$$

Proof of Theorem 6.1 part (b), assuming part (a). By equational completeness, to prove that R^ω is not finitely based it suffices to show that no finite subset of Σ defines the variety R^ω. For this purpose, consider the semigroups presented by

$$S_p = \langle a, e, f : a^p = 1,\ ea = ef = e^2 = e,\ fa = fe = f^2 = f,\\ ae = e,\ af = f \rangle$$

where p is a positive integer. This semigroup is realized for instance as the semigroup of transformations of the set $\{1, \dots, p, p+1, p+2\}$, where a acts on $\{1, \dots, p\}$ as the cycle $(1, \dots, p)$ and fixes the other two points, and e and f are constant maps, respectively with values $p+1$ and $p+2$. In particular, S_p has $p+2$ elements. On S_p, we define a unary operation τ by taking

$$\tau(e) = e,\ \tau(f) = f,\ \tau(1) = e,\ \tau(a^k) = f\ (k \in \mathbb{Z} \setminus p\mathbb{Z}),$$

which determines a unary semigroup $\mathcal{S}_p = (S_p, \cdot, \tau)$. Note that $\tau(a^p) = \tau(1) = e \neq f = \tau(a)$ and so $\mathcal{S}_p$ fails the identity $(x^p)^\omega = x^\omega$. It is pure routine to verify that $\mathcal{S}_p$ satisfies the identities in (30), (31), and (32) for r relatively prime with p, which completes the proof of statement (b). □

The proof of Theorem 6.1(a) will involve several technical lemmas establishing a number of formal consequences of the set Σ of identities introduced in Section 6. The first result is an improvement of Lemma 3.3 for the case of ω-terms but neither result seems to directly imply the other.

Lemma 6.4. *Let u, v be ω-terms such that $c(v) \subseteq c(u)$. Then $\Sigma \vdash u^\omega v = u^\omega$.*

Proof. We start by considering the case in which v is a variable $x \in c(u)$. If there is a factorization of the form $u \equiv u'xu''$ (where u' and u'' may be empty), then $\Sigma \vdash u^\omega = (u'xu'')^\omega = u'(xu''u')^\omega = u'(xu''u')^\omega x = u^\omega x$. Otherwise, there is a factorization of the form $u = u'w^\omega u''$ such that $x \in c(w)$. Then, by induction on the construction of the ω-term u, we have $\Sigma \vdash w^\omega = w^\omega x$, which reduces the problem to the above case.

We then proceed by induction on the construction of the ω-term v. Note that

$$\Sigma \vdash (xy)^\omega = x(yx)^\omega = x(yx)^\omega y^\omega = (xy)^\omega y^\omega.$$

Hence, assuming inductively that we may deduce from Σ the identities $u^\omega = u^\omega v_i$ ($i = 1, 2$), we may also deduce the identities

$$u^\omega = u^\omega v_2 = u^\omega v_1 v_2, \quad \text{and}$$
$$u^\omega = (u^\omega)^\omega = (u^\omega v_1)^\omega = (u^\omega v_1)^\omega v_1^\omega = u^\omega v_1^\omega,$$

which completes the induction step and the proof. □

Lemma 6.5. *For every ω-term u there is an ω-term v in reduced form such that $\Sigma \vdash u = v$.*

Proof. We proceed by induction on the construction of ω-terms. First, it is easy to see that every ω-term u is of the form $u_1 v_1^\omega \cdots u_n v_n^\omega u_{n+1}$ where each u_i is a (possibly empty) word and each v_i is an ω-term where the maximum number of nested ω-powers is smaller than in the original ω-term. By the induction hypothesis, we may assume that each ω-term u_i, v_i is in reduced form.

Suppose that some v_i admits a factorization of the form $v_i = xy^\omega$ with $c(x) \subseteq c(y)$. By Lemma 6.4, Σ implies the identities $v_i^\omega = (xy^\omega)^\omega = xy^\omega(xy^\omega)^\omega = xy^\omega$ and so we may replace v_i^ω by xy^ω in the above expression, where xy^ω is already in reduced form by the assumption that v_i is. Therefore, one may assume that each v_i^ω is in reduced form.

Finally, applying again Lemma 6.4, we may further assume that no u_i admits a factorization $u_i = u_i' u_i''$ with u_i' nonempty (and u_i'' possibly empty) such that $c(u_i') \subseteq c(v_{i-1})$; in case $u_i = \varepsilon$ and $i > 1$, we also assume that $c(v_i) \nsubseteq c(v_{i-1})$. In this way, we obtain the desired reduced ω-term v such that $\Sigma \vdash u = v$. □

The following is a partial cancellation law for the variety defined by Σ.

Lemma 6.6. *Let u, v be ω-terms such that Σ implies the identity $u = v$ and let a be a letter such that $a \in c(u)$. Let a_i be a marker in $\overline{u}$ and a_j be a marker in $\overline{v}$. Then Σ implies the identities $u(0, a) = v(0, a)$ and $u(i, \#) = v(j, \#)$.*

Proof. By definition of a consequence of a set of identities, it suffices to assume that $(u, v) = (s\,\ell\,t, s\,r\,t)$, where s, t are ω-terms and $\ell = r$ is obtained from an identity of Σ by substituting the variables x and y by appropriate ω-terms.

If the letter a appears in s, then it appears in $\overline{s}$ as a marker a_k, $u(0, a) \equiv s(0, a) \equiv v(0, a)$, $u(i, \#) \equiv s(k, \#)\,\ell\,t$, and $v(j, \#) \equiv s(k, \#)\,r\,t$, from which it follows that the identity $u(0, a) = v(0, a)$ is trivial and $\Sigma \vdash u(i, \#) = v(j, \#)$. At the other end, if the letter does not occur in s, ℓ or r, then a is a marker a_k in $\overline{t}$, $u(0, a) \equiv s\,\ell\,(t(0, a))$, $v(0, a) \equiv s\,r\,(t(0, a))$, and $u(i, \#) \equiv t(k, \#) \equiv v(j, \#)$ and the result follows similarly.

It remains to treat the case where the letter a does not occur in s but it occurs in ℓ, and so also in r. We may then as well assume that s and t are empty terms, that is $u \equiv \ell$ and $v \equiv r$. So, we take each of the identities from Σ, consider the letter x, and y if present, as ω-terms, which produces an identity $u = v$, and compute in each case the terms $u(0,a)$, $v(0,a)$, $u(i,\#)$, and $v(j,\#)$. This is a routine calculation which is included for the sake of completeness.

For the identities in (30), suppose first that $a \in c(x)$. If u and v both belong to $\{(xy)^\omega, (xy)^\omega x, (xy)^\omega x^\omega\}$, then $i = j$ and $u(0,a) \equiv x(0,a) \equiv v(0,a)$, while for $z = u, v$, we have $z(i,\#) \equiv x(i-1,\#)\, y\, z$ (where the -1 accounts for the opening parenthesis in z). Since $u = v$ is an identity of Σ and $i = j$, we have indeed $\Sigma \vdash u(i,\#) = v(j,\#)$ in this case. In the other case, if one of the terms u or v, say u, is equal to $x(yx)^\omega$ and v belongs to $\{(xy)^\omega, (xy)^\omega x, (xy)^\omega x^\omega\}$, then $i = j-1$, $u(0,a) \equiv x(0,a) \equiv v(0,a)$, while $u(i,\#) \equiv x(i,\#)(yx)^\omega$ and again, $v(j,\#) \equiv x(j-1,\#)\, y\, v \equiv x(i,\#)\, y\, v$. Therefore, we have $\Sigma \vdash u(i,\#) = x(i,\#)(yx)^\omega = x(i,\#)yx(yx)^\omega = x(i,\#)yu = x(i,\#)yv = v(j,\#)$, so that the conclusion of the lemma is also verified.

Suppose next that $a \notin c(x)$. Then in all cases $i = j$, $u(0,a) \equiv x(y(0,a)) \equiv v(0,a)$, and for $z = u, v$, we have $z(i,\#) = y(i - |\overline{x}| - 1, \#)\, z$ so that the conclusion of the lemma is trivial in this case.

Finally, for the one-variable identities (31) and (32), assume for instance that $v = x^\omega$. In both cases, we have $u(0,a) \equiv x(0,a) \equiv v(0,a)$ and $v(j,\#) \equiv x(j-1,\#)\, x^\omega$. For (31), we obtain $i = j+1$, $u(i,\#) \equiv x(j-1,\#)\, x^\omega (x^\omega)^\omega$ while, for the identity (32), $i = j$ and $u(i,\#) = x(j-1,\#)\, x^{r-1}(x^r)^\omega$. Thus we require the identities $x^\omega = x^\omega (x^\omega)^\omega = x^{r-1}(x^r)^\omega$, which are easily shown to be consequences of Σ. $\square$

We say that $u_1 \cdots u_k$ is a *Σ-fringy decomposition* of an ω-term u if each u_i is a fringy factor of $u_1 \cdots u_k$ and $\Sigma \vdash u = u_1 \cdots u_k$.

We will show that if an ω-term u^ω is in reduced form, then one can deduce from Σ a factorization $u_1 u_2$ of u, such that for some $r \geq 0$ and $s \geq 1$, $u^r u_1$ and $(u_2 u_1)^s$ have Σ-fringy decompositions. Consider for instance $u = (a^2 b^2)^\omega c^2 d^2$ (which is the ω-term w_2 from Proposition 4.8, up to a renaming of the letters). Obviously, $\Sigma \vdash u = u_1 u_2$ with $u_1 = (a^2 b^2)^\omega c^2 d$, and $u_2 = d$. Furthermore, both $u_1 = u^0 u_1$ and $(u_2 u_1)^2$ admit Σ-fringy decompositions.

Lemma 6.7. *Let u be an ω-term such that u^ω is in reduced form. Then there are ω-terms u_1, u_2, where u_1 is not empty (and u_2 may be empty) and integers $r \geq 0$, $s \geq 1$ such that $\Sigma \vdash u = u_1 u_2$ and the ω-terms $u^r u_1$ and $(u_2 u_1)^s$ admit Σ-fringy decompositions. Moreover, u_1 and u_2 may be chosen so that the maximum number of nested ω-powers in each of them does not exceed that maximum for u.*

Proof. By Corollary 5.11, there are sequences $(i_n)_n$, of positions in u^ω, and $(a_n)_n$, of letters in u^ω, such that $i_0 = 1$, $v_n = u^\omega(i_n, a_n)a_n$ is a fringy factor of u^ω and $(a_n, i_{n+1}) \in A \times \mathbb{N}_+$ is a marker of $\overline{u^\omega}(i_n, \#)$. Note that, since $\overline{u^\omega}$ starts with an opening parenthesis, $i_n > 1$ for all $n > 0$. Since the sequence $(i_n)_n$ takes its values in a finite set, there are positive integers n, m such that $n < m$ and $i_n = i_m$ (and therefore $a_n = a_m$, since i_n uniquely determines a_n). Since u^ω is reduced, u is not an idempotent in $\overline{\Omega}_A \mathsf{R}$ and we may assume, without loss of generality (increasing m if necessary), that $i_{n+1} - 1$ is the position of the first occurrence of a_n in u, where the -1 accounts for the opening parenthesis in u^ω. We let $u_1 = u(0, a_n)a_n$ and $u_2 = u(i_{n+1} - 1, \#)$. Note that u_1 is not empty. Then, we have $\Sigma \vdash u = u_1 u_2$ by Corollary 6.3, and the last sentence in the statement of the lemma is also guaranteed.

Let r be the number of indices $j \in \{0, \ldots, n-1\}$ and s be the number of indices $j \in \{n, \ldots, m-1\}$ in both cases such that $c(u(i_j - 1, \#)) \neq c(u)$. These numbers count how many times we have to wrap around the ω-power to get the next fringy factor, respectively before we get to the index n and from then on until we get to the index m. By Corollary 6.3, we may deduce from Σ the equalities $u^r u_1 = v_0 \cdots v_n$ and $(u_2 u_1)^s = v_{n+1} \cdots v_m$. Observe that the latter equality implies that $s \neq 0$, since $n + 1 \leq m$ and none of the v_i's is empty. $\square$

In the former example, $\overline{u^\omega} = [_1[_2 a_3 a_4 b_5 b_6]_7 c_8 c_9 d_{10} d_{11}]_{12} \#_{13}$ yields

$$
\begin{aligned}
&v_0 = (a^2b^2)^\omega c^2 d, && a_0 = d, && i_1 = 10, && c(u(9,\#)) = \{d\},\\
&v_1 = d(a^2b^2)^\omega c, && a_1 = c, && i_2 = 8, && c(u(7,\#)) = \{c,d\},\\
&v_2 = cd^2a^2b, && a_2 = b, && i_3 = 5, && c(u(4,\#)) = \{a,b,c,d\},\\
&v_3 = b(a^2b^2)^\omega c^2 d, && a_3 = d, && i_4 = i_1,
\end{aligned}
$$

and since the 10-th letter is the first occurrence of d, there is no need to continue. So $n = 1$, $m = 4$, $u_1 = (a^2b^2)^\omega c^2 d$, $u_2 = d$, $r = 0$, $s = 2$. Note that the terms v_1, v_2, v_3 and v_4 are exactly those appearing in the canonical form of w_2 obtained in the proof of Proposition 4.8 (namely $r_1 a_2^2 b_2$, $b_2 r_1 a_2$, $a_2 b_2^2 r_0 a_1^2 b_1$ and $t_1 a_2^2 b_2$, respectively. See equations (18) and (19)).

It is immediately verified that $\mathsf{R} \models \Sigma$. Conversely, if $u = v$ is an identity which is valid in R^ω, then the pseudoidentity $u = v$ is valid in R. Therefore, establishing Theorem 6.1 (a) amounts to proving the following theorem:

Theorem 6.8. *Let u and v be two ω-terms. Then*

$$\mathsf{R} \models u = v \implies \Sigma \vdash u = v. \tag{33}$$

Proof. We proceed by induction on the common content of u and v. In case $c(u) = c(v) = \emptyset$, the result is obvious. We now assume that it holds for all ω-terms u, v whose content has less than p elements. The proof will be broken into several intermediate results which in turn may involve other induction schemes, so we will refer to this induction hypothesis as (IH).

Note that (IH) implies that if w is an ω-term with $|c(w)| < p$ then $\Sigma \vdash w = \mathsf{cf}(w)$. Indeed, this follows from (33) using Proposition 4.7(a). We will show this property remains valid for ω-terms which involve p letters.

Proposition 6.9. *Let u be an ω-term with $|c(u)| = p$. Assuming (IH), then Σ implies the identity $u = \mathsf{cf}(u)$.*

Proof. Let $\xi(u)$ be the sequence of integers whose nth term counts, in a factorization of u into ω-powers and letters, the number of factors which are ω-powers with the maximum number n of nested ω-powers. For instance, for the ω-term $u = ((xy)^\omega z)^\omega t(x^\omega)^\omega xy(zt)^\omega$, we have $\xi(u) = (3, 1, 2, 0, 0, \ldots)$. Given two distinct sequences $(m_i)_i$ and $(n_i)_i$ of nonnegative integers with only finitely many nonzero entries, we write $(m_i)_i < (n_i)_i$ if, for the largest i such that $m_i \neq n_i$, we have $m_i < n_i$. Note that this defines a well-ordering of the set of all such sequences. Indeed, this is clearly a total ordering and, by dropping all null components, the set of elements below one given sequence is identified with the set of elements below an element of a lexicographic product of finitely many copies of $\mathbb{N}$, which is well known to be well ordered.

The proof proceeds by induction on $\xi(u)$. If u is a word, then $u \equiv \mathsf{cf}(u)$ and so the trivial identity $u = \mathsf{cf}(u)$ is a consequence of Σ. Suppose next that $\Sigma \vdash v = \mathsf{cf}(v)$ for every ω-term v such that $\xi(v) < \xi(u)$. We need another embedded intermediate result, namely the following complement of Proposition 4.7 about canonical forms of ω-terms.

The *cumulative content* $\vec{c}(w)$ of an ω-term w is the set of all letters a such there is some factorization $w = w_1 w_2^\omega w_3$ with $a \in c(w_2)$ and $c(w_3) \subseteq c(w_2)$. Note that the cumulative content of an ω-term w coincides with the cumulative content of the pseudoword defined by w.

Proposition 6.10. *Let v, w be ω-terms and let a be a letter.*

(a) *If v is a fringy factor of vw, then $\Sigma \vdash \mathsf{cf}(vw) = \mathsf{cf}(v)\,\mathsf{cf}(w)$.*
(b) *If $u = v^\omega$ for some ω-term v, with v^ω reduced, and $\Sigma \vdash z = \mathsf{cf}(z)$ for every ω-term z with $\xi(z) < \xi(u)$, then $\Sigma \vdash \mathsf{cf}(u) = u$.*
(c) *If $a \notin c(v)$ and $\Sigma \vdash z = \mathsf{cf}(z)$ for every ω-term z with $\xi(z) < \xi(w)$, then $\Sigma \vdash \mathsf{cf}(vaw) = \mathsf{cf}(v) \cdot a \cdot \mathsf{cf}(w)$.*
(d) *If $a \notin \vec{c}(v)$ and $\Sigma \vdash z = \mathsf{cf}(z)$ for every ω-term z with $\xi(z) < \xi(w)$, then $\Sigma \vdash \mathsf{cf}(vaw) = \mathsf{cf}(v) \cdot a \cdot \mathsf{cf}(w)$.*

Proof. For the proof of (a), we consider two cases, namely whether the initial state of $\mathcal{A}(vw)$ is the end of an edge or not. By definition of the canonical form, in the negative case we have the equality of ω-terms $\mathsf{cf}(vw) \equiv \mathsf{cf}(v)\,\mathsf{cf}(w)$. Otherwise, again by definition of the canonical form, vw is an idempotent over R. Since v is a fringy factor of vw, we have $\vec{c}(vw) = \vec{c}(w)$ and $c(v) = c(vw) = c(w)$, hence w is an idempotent over R. Let q be the initial state of the R-automaton $\mathcal{A}(vw)$. Then the automaton obtained from $\mathcal{A}(vw)$ by replacing the root with $\mathsf{q}.1$ is the R-automaton $\mathcal{A}(w)$. Hence $\mathsf{cf}(vw) \equiv (\mathsf{cf}(v)u)^\omega$ for some ω-term u in canonical form such that $\mathsf{cf}(w) \equiv (u\,\mathsf{cf}(v))^\omega$ and so we have

$$\Sigma \vdash \mathsf{cf}(vw) = (\mathsf{cf}(v)u)^\omega = \mathsf{cf}(v)(u\,\mathsf{cf}(v))^\omega = \mathsf{cf}(v)\,\mathsf{cf}(w),$$

which proves (a).

Suppose next that u is an ω-term as in (b). Applying Lemma 6.7 to v^ω, we obtain two ω-terms v_1, v_2 and two integers $r \geq 0, s \geq 1$ such that $\Sigma \vdash v = v_1 v_2$ and the ω-terms $v^r v_1$ and $(v_2 v_1)^s$ admit Σ-fringy decompositions:

$$\Sigma \vdash v^r v_1 = u_1 \cdots u_m$$
$$\Sigma \vdash (v_2 v_1)^s = u_{m+1} \cdots u_n$$

By simple applications of identities deduced from Σ, we obtain

$$\Sigma \vdash u = v^\omega = v^r v^\omega = v^r (v_1 v_2)^\omega = v^r v_1 (v_2 v_1)^\omega = v^r v_1 \big((v_2 v_1)^s\big)^\omega, \tag{34}$$

where the last equality is justified since $s \geq 1$, by Lemma 6.7. Let I be the set all integers k such that $0 \leq k < \min(m, n-m)$ and $\mathsf{R} \not\models u_{m-k} = u_{n-k}$; let $p = \min I$ if $I \neq \varnothing$, and $p = 0$ otherwise.

We claim that $\Sigma \vdash u = u_1 \cdots u_{m-p}(u_{m-p+1} \cdots u_{n-p})^\omega$. Indeed, for all $i = 1, \ldots, n$, we have $\xi(u_i) < \xi(u)$, and therefore, by the hypothesis of (b), $\Sigma \vdash \mathsf{cf}(u_i) = u_i$. By definition of p, we have further $\mathsf{R} \models u_{m-k} = u_{n-k}$ for $k > p$, hence $\mathsf{cf}(u_{m-k}) \equiv \mathsf{cf}(u_{n-k})$ by Proposition 4.7(b). Hence, for $k > p$, $\Sigma \vdash u_{m-k} = \mathsf{cf}(u_{m-k}) \equiv \mathsf{cf}(u_{n-k}) = u_{n-k}$, and

$$\begin{aligned}
\Sigma \vdash u &= u_1 \cdots u_m(u_{m+1} \cdots u_n)^\omega \\
&= u_1 \cdots u_{m-p}(u_{m-p+1} \cdots u_{n-p})^\omega u_{n-p+1} \cdots u_n && \text{applying } p \text{ times (30)} \\
&= u_1 \cdots u_{m-p}(u_{m-p+1} \cdots u_{n-p})^\omega && \text{using Lemma 6.4, since} \\
& && c(u_i) = c(u_j) \text{ for } 1 \le i, j \le n.
\end{aligned}$$

This proves the claim. Let

$$\ell = \min\{i \ge m-p+1 \mid \exists k \ge 1, \mathsf{R} \models u_{m-p+1} \cdots u_{n-p} = (u_{m-p+1} \cdots u_i)^k\}.$$

Since $\xi(u_{m-p+1} \cdots u_{n-p}), \xi((u_{m-p+1} \cdots u_\ell)^k) < \xi(u)$, we obtain, using again the hypothesis of (b), that Σ deduces $u_{m-p+1} \cdots u_{n-p} = (u_{m-p+1} \cdots u_\ell)^k$. Therefore, using (32), Σ deduces $u = u_1 \cdots u_{m-p}(u_{m-p+1} \cdots u_\ell)^\omega$. In particular, since R satisfies Σ, both sides of this identity have the same canonical form. One can apply Proposition 4.7(d) to obtain $\mathsf{cf}(u) \equiv \mathsf{cf}(u_1) \cdots \mathsf{cf}(u_{m-p})(\mathsf{cf}(u_{m-p+1}) \cdots \mathsf{cf}(u_\ell))^\omega$ (note that hypotheses (ii) and (iii) of Proposition 4.7(d) hold by the choice of p and ℓ, respectively). Finally, since $\Sigma \vdash u_i = \mathsf{cf}(u_i)$ for $i = 1, \ldots, n$, we conclude that $\Sigma \vdash \mathsf{cf}(u) = u_1 \cdots u_{m-p}(u_{m-p+1} \cdots u_\ell)^\omega = u$, which proves (b).

The proof of (c) is rather more complicated. It proceeds by induction on $\xi(w)$. The case when $\xi(w)$ is the constant zero sequence occurs when $w = \varepsilon$ and thus it follows directly from the definition of canonical form as $\mathsf{cf}(va) \equiv \mathsf{cf}(v)a$. Thus, we assume that $w \neq \varepsilon$. Let i be such that a_i is the first occurrence of a in $\overline{vaw}$, b_k be the principal marker of $\overline{vaw}$, and c_ℓ be the principal marker of $\overline{vaw}(i, \#)$ (which exists since $vaw(i, \#) =_{\mathsf{R}} w \neq \varepsilon$). If $i = k$, then the desired result, namely that $\Sigma \vdash \mathsf{cf}(vaw) = \mathsf{cf}(v)\, a\, \mathsf{cf}(w)$, follows directly from (a). So we may assume that $i < k$ and therefore we have $i < k \le \ell$. Let $w_1 = (vaw)(i, b)$, $w_2 = (vaw)(k, c)$ in case $k < \ell$, and $w_3 = (vaw)(\ell, \#)$.

By Lemma 6.5, without loss of generality, we may assume that the factorization $w = x_0 y_1^\omega x_1 \cdots y_m^\omega x_m$ is in reduced form, where the x_j are possibly empty words and the y_j are ω-terms. For a marker d_q in $\overline{w}$, consider the first factor, x_j or y_j^ω, from left to right, that involves the letter d. If this first factor is x_j and the factorization $x_j = x'_j d x''_j$ is such that $d \notin c(x'_j)$, then clearly $\xi(x_0 y_1^\omega x_1 \cdots y_j^\omega x'_j)$ and $\xi(x''_j y_{j+1}^\omega x_{j+1} \cdots y_m^\omega x_m)$ are both smaller than $\xi(w)$ and so we may apply the hypothesis of (c) to both. On the other hand, if the first factor containing d is y_j, then by Corollary 6.3 there are ω-terms y'_j and y''_j, whose maximum number of nested ω-powers does not exceed that of y_j, such that $\Sigma \vdash y_j = y'_j d y''_j$ and $d \notin y'_j$. In this case, we obtain

$$\begin{aligned}
\Sigma \vdash w &= x_0 y_1^\omega x_1 \cdots y_{j-1}^\omega x_{j-1} y'_j \cdot d \cdot y''_j y_j^\omega x_j \cdots y_m^\omega x_m \\
&= x_0 y_1^\omega x_1 \cdots y_{j-1}^\omega x_{j-1} y'_j \cdot d \cdot (y''_j y'_j d)^\omega x_j \cdots y_m^\omega x_m
\end{aligned}$$

using the identity $x(yx)^\omega = (xy)^\omega$, where $\xi(x_0 y_1^\omega x_1 \cdots y_{j-1}^\omega x_{j-1} y'_j) < \xi(w)$ and $\xi((y''_j y'_j d)^\omega x_j \cdots y_m^\omega x_m) \le \xi(w)$. Moreover, equality occurs in this latter inequality if and only if $j = 1$ and $x_0 = \varepsilon$. We will apply these observations to the markers b_k and c_ℓ.

The preceding paragraph guarantees in particular that $\xi(w_1) < \xi(w)$ so that we may apply the hypothesis of (c) to obtain

$$\Sigma \vdash \mathsf{cf}(vaw_1) = \mathsf{cf}(v)\, a\, \mathsf{cf}(w_1). \tag{35}$$

In case $k = \ell$, we have $\mathsf{R} \models w = w_1 b w_3$, therefore $\mathsf{cf}(vaw) = \mathsf{cf}(vaw_1 b w_3)$ and

$$\begin{aligned}
\Sigma \vdash \mathsf{cf}(vaw) = \mathsf{cf}(vaw_1 b w_3) &= \mathsf{cf}(vaw_1)\, b\, \mathsf{cf}(w_3) && \text{by } (a) \\
&= \mathsf{cf}(v)\, a\, \mathsf{cf}(w_1)\, b\, \mathsf{cf}(w_3) && \text{by (35)} \\
&= \mathsf{cf}(v)\, a\, \mathsf{cf}(w_1 b w_3) && \text{by } (a) \\
&= \mathsf{cf}(v)\, a\, \mathsf{cf}(w).
\end{aligned}$$

Suppose next that $k < \ell$. In this case, we have $\mathsf{R} \models w = w_1 b w_2 c w_3$, so $\mathsf{cf}(vaw) = \mathsf{cf}(vaw_1 b w_2 c w_3)$ and

$$\begin{aligned}
\Sigma \vdash \mathsf{cf}(vaw) = \mathsf{cf}(vaw_1 b w_2 c w_3) &= \mathsf{cf}(vaw_1)\, b\ \mathsf{cf}(w_2 c w_3) && \text{by } (a) \\
&= \mathsf{cf}(v)\, a\, \mathsf{cf}(w_1)\, b\ \mathsf{cf}(w_2 c w_3) && \text{by (35).}
\end{aligned}$$

Hence, to conclude the proof that $\Sigma \vdash \mathsf{cf}(vaw) = \mathsf{cf}(v)\, a\ \mathsf{cf}(w)$, it suffices to show that

$$\Sigma \vdash \mathsf{cf}(w_1)\, b\ \mathsf{cf}(w_2 c w_3) = \mathsf{cf}(w). \tag{36}$$

In case $\xi(w_2cw_3) < \xi(w)$, the induction hypothesis yields (36) directly. On the other hand, by the above observations with $d = b$, otherwise we may assume that $\xi(w_2cw_3) = \xi(w)$ and that $x_0 = \varepsilon$ and $b \in c(y_1)$.

We now distinguish two cases according to whether or not $c \in c(y_1)$. Since c_ℓ is the principal marker of $\overline{vaw}(i, \#)$ and $w = x_0 y_1^\omega x_1 \cdots y_m^\omega x_m$ is in reduced form, in case $c \in c(y_1)$ it follows that $w = y_1^\omega$. Hence, in the notation introduced for the above observations, with $d = b$, and using Lemma 6.6, the set Σ deduces the identities $w_1 = y_1'$ and $w_2cw_3 = y_1'' y_1^\omega$. Since $\xi(y_1') < \xi(w)$, which yields $\Sigma \vdash \mathsf{cf}(y_1') = y_1'$ using this time the induction hypothesis, this allows us to obtain the following consequences of Σ:

$$\begin{aligned}
\Sigma \vdash \mathsf{cf}(w_1)\, b\ \mathsf{cf}(w_2cw_3) &= \mathsf{cf}(y_1')\, b\ \mathsf{cf}(y_1'' y_1^\omega) \\
&= \mathsf{cf}(y_1')\, b\ \mathsf{cf}((y_1'' y_1' b)^\omega) && \text{using } x(yx)^\omega = (xy)^\omega \\
&= y_1' b (y_1'' y_1' b)^\omega && \text{by induction hypothesis, and by } (b) \text{ applied to } (y_1'' y_1' b)^\omega \\
&= y_1^\omega && \text{using } x(yx)^\omega = (xy)^\omega \text{ and } \Sigma \vdash y_1' b y_1'' = y_1 \\
&= \mathsf{cf}(y_1^\omega) && \text{by } (b) \\
&= \mathsf{cf}(w),
\end{aligned}$$

which establishes (36) in this case (observe, for the third equality, that we may assume that $(y_1'' y_1' b)^\omega$ is reduced).

To conclude the proof of (c), it remains to consider the case $x_0 = \varepsilon$, $b \in c(y_1)$, $c \notin c(y_1)$. In this case, we deduce using Lemma 6.6 that $\Sigma \vdash w_1 = y_1'$ and $\Sigma \vdash w_2 = y_1'' y_1^\omega w_2' = (y_1'' y_1' b)^\omega w_2'$ for some ω-term w_2'. The above observations applied to $d = c$ show that $\xi((y_1'' y_1' b)^\omega w_2') < \xi(w)$. Since $c(y_1) = c(y_1'' y_1' b)$, observe that the principal marker of $\overline{(y_1'' y_1' b)^\omega w_2'\, c\, w_3}$ also corresponds to the displayed occurrence of the letter c. We now obtain the following consequences of Σ:

$$\begin{aligned}
\Sigma \vdash \mathsf{cf}(w_1)\, b\ \mathsf{cf}(w_2cw_3) &= \mathsf{cf}(y_1')\, b\ \mathsf{cf}((y_1'' y_1' b)^\omega w_2'\, c\, w_3) \\
&= \mathsf{cf}(y_1')\, b\ \mathsf{cf}((y_1'' y_1' b)^\omega w_2')\, c\ \mathsf{cf}(w_3) && \text{by } (a) \\
&= \mathsf{cf}(y_1'\, b\, (y_1'' y_1' b)^\omega w_2')\, c\ \mathsf{cf}(w_3) && \text{by the hypothesis of } (c) \\
& && \text{since } \xi((y_1'' y_1' b)^\omega w_2') < \xi(w) \\
&= \mathsf{cf}(y_1^\omega w_2')\, c\ \mathsf{cf}(w_3) && \text{using } x(yx)^\omega = (xy)^\omega \\
&= \mathsf{cf}(y_1^\omega w_2'\, c\, w_3) && \text{by } (a) \\
&= \mathsf{cf}(w),
\end{aligned}$$

which proves (36) and completes the induction step for the proof of (c).

To prove (d), we let $\chi(v, a, w) = (|c(vaw)|, |c(v)|, \|v\|)$ and we order such triples lexicographically. We proceed by induction on $\chi(v, a, w)$, assuming that property (d) holds for all triples (v', a', w') with $\chi(v', a', w') < \chi(v, a, w)$.

We first assume that $c(va) \subsetneqq c(vaw)$. Then, by Corollary 6.3 there exist ω-terms w_1, w_2 and a letter b such that vaw_1b is a fringy factor of vaw_1bw_2 and $\Sigma \vdash w_1bw_2 = w$. Then $\chi(v, a, w_1) < \chi(v, a, w)$ which yields

$$\begin{aligned}
\Sigma \vdash \mathsf{cf}(vaw) &= \mathsf{cf}(vaw_1)\, b\, \mathsf{cf}(w_2) && \text{by } (a) \\
&= \mathsf{cf}(v)\, a\, \mathsf{cf}(w_1)\, b\, \mathsf{cf}(w_2) && \text{by the induction hypothesis} \\
&= \mathsf{cf}(v)\, a\, \mathsf{cf}(w) && \text{by } (c).
\end{aligned}$$

Hence we may assume that $c(va) = c(vaw)$. In case $a \notin c(v)$, we may apply (c) directly to obtain the desired result. Hence we will assume that $a \in c(v)$, in which case the principal marker of vaw is found within v. By Corollary 6.3 there exist ω-terms v_1, v_2 and a letter b such that v_1b is a fringy factor of v_1bv_2 and $\Sigma \vdash v_1bv_2 = v$. Since $a \in c(v) \setminus \vec{c}(v)$, we have $\|v\| < \infty$ by Proposition 3.8 and either $c(v_2) \subsetneqq c(v)$ or $\|v_2\| < \|v\|$. In either case, we find that $\chi(v_2, a, w) < \chi(v, a, w)$ while $\vec{c}(v_2) \subseteq \vec{c}(v)$, so that $a \notin \vec{c}(v_2)$. This allows us to show that

$$\begin{aligned}
\Sigma \vdash \mathsf{cf}(vaw) &= \mathsf{cf}(v_1) b\, \mathsf{cf}(v_2aw) && \text{by } (a) \\
&= \mathsf{cf}(v_1) b\, \mathsf{cf}(v_2) a\, \mathsf{cf}(w) && \text{by the induction hypothesis} \\
&= \mathsf{cf}(v) a\, \mathsf{cf}(w) && \text{by } (a).
\end{aligned}$$

This completes the induction step and the proof of Proposition 6.10. □

Back to the proof of Proposition 6.9, without loss of generality, we may assume that u is reduced, noting that the reduction, which is performed using identities from Σ by Lemma 6.5, does not affect the canonical form by Proposition 4.7(b) nor does it increase the value of $\xi(u)$. Then u is of the form $u = u_0 v_1^\omega u_1 \cdots v_k^\omega u_k$ where the u_i are words, which may be empty, and the v_i are ω-terms. Since u is reduced, each nonempty u_i with $i > 0$ must start with a letter a_i which does not belong to $\vec{c}(v_i)$. If there is any $i \geq 1$ such that $u_i \neq \varepsilon$ then, applying the induction hypothesis on the parameter ξ, we obtain that Σ implies the identities $u_0v_1^\omega u_1\cdots v_i^\omega = \mathsf{cf}(u_0v_1^\omega u_1\cdots v_i^\omega)$ and $u_iv_{i+1}^\omega u_{i+1}\cdots v_k^\omega u_k = \mathsf{cf}(u_iv_{i+1}^\omega u_{i+1}\cdots v_k^\omega u_k)$ and so also the identity $u = \mathsf{cf}(u)$ by Proposition 6.10(d). In case $u_0 \neq \varepsilon$, the induction hypothesis similarly implies that Σ allows us to deduce the identity $v_1^\omega u_1\cdots v_k^\omega u_k = \mathsf{cf}(v_1^\omega u_1\cdots v_k^\omega u_k)$, from which the identity $u = \mathsf{cf}(u)$ follows. It remains to treat the case in which $u_i = \varepsilon$ for all i, that is $u = v_1^\omega\cdots v_k^\omega$ is a product of ω-powers.

The case when u is a single ω-power is given by Proposition 6.10(b). We proceed by considering the case $k \geq 2$. Then we apply Lemma 6.7 to v_2^ω (which is reduced) to obtain ω-terms w_1, w_2 and positive integers r, s such that $\Sigma \vdash v_2 = w_1w_2$ and there are Σ-fringy decompositions of $v_2^rw_1$ and $(w_2w_1)^s$. Then, as in (34) we obtain

$$\Sigma \vdash u = v_1^\omega\cdots v_k^\omega = v_1^\omega v_2^r w_1 \cdot (w_2w_1)^\omega v_3^\omega\cdots v_k^\omega.$$

Since each of the factors $x = v_1^\omega v_2^r w_1$ and $y = (w_2w_1)^\omega v_3^\omega\cdots v_k^\omega$ has a smaller ξ-value than u by Lemma 6.7, we may apply the induction hypothesis to deduce that Σ implies the identities $x = \mathsf{cf}(x)$ and $y = \mathsf{cf}(y)$. Now $v_2^rw_1$ is a product of fringy factors of v_2^ω and u is reduced. Hence x is of the form $x = za$ for some letter a such that $a \notin \vec{c}(z)$: indeed $\vec{c}(x) = \vec{c}(v_2^rw_1)$ since u is reduced and so $\vec{c}(x) = \emptyset$ since $\|v_2^rw_1\| < \infty$; now, if $a \in \vec{c}(z)$ then $a \in \vec{c}(x)$, in contradiction with what we have just shown. By Proposition 6.10(d) and Proposition 4.7(b), it follows that

$$\Sigma \vdash \mathsf{cf}(u) = \mathsf{cf}(xy) = \mathsf{cf}(x)\,\mathsf{cf}(y) = xy = u,$$

which completes the induction step and the proof of Proposition 6.9. □

Back to the proof of Theorem 6.8, note that, by Proposition 4.7(b), two ω-terms coincide in R if and only if their canonical forms are equal. Hence, if $\mathsf{R} \models u = v$ for two ω-terms u, v involving p letters, then $\Sigma \vdash u = \mathsf{cf}(u) = \mathsf{cf}(v) = v$ by Proposition 6.9. This proves the induction step for (IH) and concludes the proof of the theorem. □

7. Open problems

We have exhibited a very efficient algorithm to solve the word problem for ω-terms over R. The algorithm has essentially three stages: (1) to construct an R-automaton for each of the ω-terms; (2) to wrap these automata; and (3) to compare the wrapped automata. We observe that we have obtained algorithms with optimal asymptotic complexity for each of these stages. But, we have not shown that there is no other, asymptotically more efficient, algorithm to solve the problem and we do not know if there is one.

There are several related algorithmic questions on ω-terms and their wrapped R-automata representations. Of course, if we work with ω-terms, it is trivial to compute products and ω-powers since we can just do it graphically. However, if instead we are given their wrapped R-automata representations, then it is not at all obvious how to efficiently obtain the wrapped R-automata for the product or the ω-power since it appears that, in general these operations may completely change the structure of the given R-automata.

The difficulty here is that the only natural way we have presented to build an ω-term from a wrapped R-automaton, whose wrapped R-automaton is the given one, is through the construction of the canonical form, which we have shown can have exponential length in terms of the size of the alphabet (Proposition 4.8). On the other hand, if we fix the alphabet then, by Proposition 4.10, the size of the canonical form is bounded by a polynomial function of the size of the R-automaton. By definition of the canonical form, it may be computed within the same time bound. So, if we are given wrapped R-automata over a fixed alphabet, we can compute efficiently ω-terms of which they are the wrapped R-automata and concatenate them or take their ω-powers. Then, by applying the algorithms of Subsections 5.3 and 5.4, we may compute the wrapped R-automaton of the thus computed ω-terms. By Theorems 5.16 and 5.17, the overall cost of this algorithm is polynomial in the size of the given R-automata. However, the above bound for the complexity of the stage in algorithm computing representative ω-terms becomes exponential if we do not bound the alphabet. We do not know whether the upper bound of Proposition 4.10 is optimal. Here are some related questions:

- (a) is there a polynomial asymptotic upper bound for the size of an ω-term whose wrapped R-automaton is given?
- (b) in the affirmative case, can we compute it efficiently?
- (c) find tight lower and upper bounds for the number of states of the wrapped R-automata representing the "product" or the "ω-power" of given wrapped R-automata.

Another direction which seems to be worth investigating is the following. There is a pseudovariety which is closely related with R to which it should be possible to extend the considerations in this paper. That is the pseudovariety DA, which consists of all finite semigroups whose regular elements are idempotents. For this pseudovariety, there is a tool corresponding to the left basic factorization which was introduced in [1], namely what in that paper is called a "basic boundary factorization". This consists in locating, from both sides, the last letter to occur for the first time, with possible coincidence or cross-over. The similarity between the nature of the two factorizations suggests that indeed the same techniques could work in that case. We have not attempted to carry out this program.

References

[1] J. Almeida, A syntactical proof of locality of DA, Int. J. Algebra Comput. 6 (1996) 165–177.

[2] J. Almeida, Finite semigroups: An introduction to an unified theory of pseudovarieties, in: G. Gomes, J.-E. Pin, P. Silva (Eds.), Proceedings of Semigroups, Algorithms, Automata and Languages, World Scientific, 2002, pp. 3–64.

[3] J. Almeida, A. Azevedo, The join of the pseudovarieties of $\mathcal{R}$–trivial and $\mathcal{L}$–trivial monoids, J. Pure Appl. Algebra 60 (1989) 129–137.

[4] J. Almeida, J. C. Costa, M. Zeitoun, Tameness of pseudovariety joins involving R, Monatshefte für Mathematik 146 (2) (2005) 89–111.

[5] J. Almeida, B. Steinberg, Syntactic and global semigroup theory: a synthesis approach, in: Algorithmic problems in groups and semigroups (Lincoln, NE, 1998), Trends Math., Birkhäuser Boston, Boston, MA, 2000, pp. 1–23.

[6] J. Almeida, B. Steinberg, On the decidability of iterated semidirect products with applications to complexity, Proc. London Math. Soc. (2) 80 (2000) 50–74.

[7] J. Almeida, P. Weil, Free profinite $\mathcal{R}$-trivial monoids, Int. J. Algebra Comput. 7 (5) (1997) 625–671.

[8] J. Almeida, M. Zeitoun, The equational theory of ω-terms for finite $\mathcal{R}$-trivial semigroups, in: M. Branco, G. Gomes (Eds.), Proceedings of Semigroups and Languages, World Scientific, Singapore, 2004, pp. 1–23.

[9] J. Almeida, M. Zeitoun, A linear time algorithm for minimizing disjoint-loop automata, submitted.

[10] D. Beauquier, J. Berstel, Ph. Chrétienne, Éléments d'Algorithmique, Masson, 1992, in French. `http://www-igm.univ-mlv.fr/~berstel/Elements/Elements.pdf`.

[11] S. Bloom, Christian. Choffrut, Long words, the theory of concatenation and omega power, Theoret. Comput. Sci. 259 (2001) 533–548.

[12] K. S. Booth, Lexicographically least circular substrings, Inform. Process. Lett. 10 (1980) 240–242.

[13] T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, McGraw-Hill Higher Education, 2001.

[14] M. Crochemore, W. Rytter, Text Algorithms, The Clarendon Press Oxford University Press, New York, 1994, with a preface by Zvi Galil.

[15] J. E. Hopcroft, An $n \log n$ algorithm for minimizing states in a finite automaton, in: Z. Kohavi (Ed.), Theory of machines and computations (Proc. Internat. Sympos., Technion, Haifa, 1971), Academic Press, New York, 1971, pp. 189–196.

[16] J. E. Hopcroft, J. D. Ullman, Introduction to automata theory, languages, and computation, Addison-Wesley, Reading, Mass., 1979.

[17] T. Knuutila, Re-describing an algorithm by Hopcroft, Theoret. Comput. Sci. 250 (2001) 333–363.

[18] J. Reiterman, The Birkhoff theorem for finite algebras, Algebra Universalis 14 (1982) 1–10.

[19] D. Revuz, Minimisation of acyclic deterministic automata in linear time, Theoret. Comput. Sci. 92 (1992) 181–189.

[20] J. Rhodes, B. Steinberg, Pointlike sets, hyperdecidability and the identity problem for finite semigroups, Int. J. Algebra Comput. 9 (3-4) (1999) 475–481, dedicated to the memory of Marcel-Paul Schützenberger.

[21] Y. Shiloach, Fast canonization of circular strings, J. Algorithms 2 (1981) 107–121.

[22] R. E. Tarjan, Depth first search and linear graph algorithms, SIAM Journal on Computing 1 (2) (1972) 146–160.

Centro de Matemática e Dep. Mat. Pura, Faculdade de Ciências,, Universidade do Porto, Rua do Campo Alegre, 687, 4169-007 Porto, Portugal.

LIAFA, Université Paris 7 & CNRS UMR 7089., Current affiliation: LaBRI, Université Bordeaux 1 & CNRS UMR 5800., 351 cours de la Libération, 33405 Talence Cedex, France.