Green's Relations and their Use in Automata Theory

Thomas Colcombet[⋆]
thomas.colcombet@liafa.jussieu.fr

Liafa/Cnrs/Université Paris Diderot–Paris 7

Abstract. The objective of this survey is to present the ideal theory of monoids, the so-called Green's relations, and to illustrate the usefulness of this tool for solving automata related questions.
We use Green's relations for proving four classical results related to automata theory: The result of Schützenberger characterizing star-free languages, the theorem of factorization forests of Simon, the characterization of infinite words of decidable monadic theory due to Semenov, and the result of determinization of automata over infinite words of McNaughton.

Introduction

In this lecture, we will establish several classical results related to automata theory, respectively due to Schützenberger, Simon, Semenov, and McNaughton. These problems are all related in a more or less direct way to language theory and automata. Despite their obvious intrinsic interest, these results will be for us excuses for presenting the approach via monoids and semigroups which allows to uniformly apprehend these, *a priori* unrelated, questions. That is why this lecture is structured as the interleaving of the proofs of the above results with the necessary algebraic material.

We devote a particular attention to the theory of ideals in monoids, the so called *Green's relations*. When working in language theory using automata, several tools comes naturally into play. A typical example is the use of the decomposition of the graph of the automaton into strongly connected components, and the use of the dag of the connected components for driving an induction in a proof. The Green's relations provide the necessary tools for using similar arguments on the monoid rather than on the automaton. Since monoids are more informative than automata, the resulting techniques are more powerful than the corresponding ones on automata (this gain usually comes at the price of a worth complexity in decision procedures and constructions). The Green's relations are well known, and presented in deep detail in several places, see for instance [5, 13]. For this reason we do not establish here the results related to this theory.

[⋆] Supported by the project ANR 2010 BLAN 0202 02 FREC, and the ERC Starting Grant GALE.

We do not try either to be exhaustive in any way. Our goal is different. We are interested in illustrating how to use this tool.

We use four classical results as illustrations. The first one is the theorem of *Schützenberger* [17] characterizing the languages which can be described by star-free expressions. The second one is the theorem of factorization forests of *Simon* [19], which gives a form of generalized Ramsey argument for regular languages. The third one is a theorem of *Semenov* [18] which gives a necessary and sufficient condition for an infinite word to have a decidable monadic second-order theory. The fourth theorem, due to *McNaughton* [9], states that automata over infinite words can be made deterministic.

The lecture is structured as follows. We first briefly recall some basic definitions concerning semigroups and monoids in Section 1. We then present the results of Schützenberger and Simon in Section 2 and 3 respectively. We then introduce the framework of ω-semigroups in Section 4, and use it for establishing the results of semenov and McNaughton in Sections 5 and 6 respectively.

1 Basics on Monoids

A *monoid* $\mathbf{M}$ is a set together with an associative binary operator $\cdot$ which has a *neutral element* denoted 1 (such that $1x = x1 = x$ for all x). An element e such that $ee = e$ is called an *idempotent*. A *monoid morphism* from a monoid $\mathbf{M}$ to another $\mathbf{M}'$ is an application from M to M' such that $\alpha(1) = 1$, and $\alpha(ab) = \alpha(a)\alpha(b)$ for all a, b in $\mathbf{M}$.

A particular example is the *free monoid generated by a set A*, it is the set of words over the alphabet A equipped with the concatenation product. The neutral element is ε. An example of a finite monoid consists of the two elements a, b equipped with the product $aa = ab = ba = a$, and $bb = b$.

A language $L \subseteq A^*$ is recognizable by a monoid $(M, \cdot)$ if there exists a morphism α from A^* to $\mathbf{M}$ and a subset $F \subseteq M$ such that $L = \alpha^{-1}(F)$.

Theorem 1 (Rabin and Scott [16] with credit to Myhill). *A language of finite words over a finite alphabet is regular (i.e., accepted by some standard form of finite state automaton) if and only if it is recognizable by a finite monoid.*

Given a language L there is a particular, minimal, monoid which recognizes it, the syntactic monoid. The *syntactic monoid* of a language L over the alphabet A is an object gathering the minimal amount of information for each word that is relevant for the language. This is obtained by a quotient of words by the so-called *syntactic congruence* $\sim_L$. Two words are equivalent for this relation if they are undistinguishable by the language in any context. Formally, two words u and v are equivalent for a language L is defined as:

$$u \sim_L v \quad \text{if for all } x, y \in A^*,\ xuy \in L \text{ iff } xvy \in L\ .$$

If two words are equivalent for $\sim_L$, this means that in any context, one can safely exchange one for the other. In particular, as its name suggest, $\sim_L$ is a congruence, i.e., $u \sim_L v$ and $u' \sim_L v'$ implies $uu' \sim_L vv'$. This means that the equivalence classes for $\sim_L$ can be equipped with a product. The resulting quotiented monoid $\mathbf{M}_L = A^*/_{\sim_L}$ is called the *syntactic monoid of L*. Furthermore, the application η_L which to a word associates its equivalence class is a morphism, called the *syntactic morphism*.

In particular, setting $F = \eta_L(L)$, we have $u \in L$ if and only if $\eta_L(u) \in F$. In other words, the syntactic monoid $\mathbf{M}_L$ recognizes L using the morphism η_L and the subset $F = \eta_L(L) \subseteq \mathbf{M}_L$.

For instance, consider the language over the alphabet $A = \{a, b, c\}$ consisting of "all words which do not contain two consecutive occurrences of the letter a". The equivalence classes of the syntactic monoid are ε, $a((b+c)^+a)^*$, $(a(b+c)^+)^+$, $((b+c)^+a)^+$, $(b+c)^+(a(b+c)^+)^*$ and A^*aaA^*. We will denote them by 1, a, ab, ba, b and 0 respectively. The notations 1 and 0 are conventional, and correspond to the neutral and the absorbing element (absorbing means $0x = x0 = 0$; such an element is unique, but does not always exist). For the other congruence classes, one fixes a word as representative. The product xy of any two elements x and y of the monoid is given in the following table:

$x \backslash y$	1	a	ab	ba	b	0
1	1	a	ab	ba	b	0
a	a	0	0	a	ab	0
ab	ab	a	ab	a	ab	0
ba	ba	0	0	ba	b	0
b	b	ba	b	ba	b	0
0	0	0	0	0	0	0

The language "no two consecutive occurrences of letter a" is recognized by this monoid, together with the morphism which maps letter a to a and letters b and c to b, and the subset $F = \{1, a, ab, ba, b\}$.

We see on this example that the table of product is not very informative for understanding the structure of the monoid. Natural tools, such as the strongly connected components of the graph of the automaton, are frequently used to design proofs and constructions in automata theory. We do not see immediately anything similar in monoids. The Green's relations that we present below gives such an insight in the structure of the monoid. Even better, since the syntactic monoid is more informative than the minimal automaton (at a price: it is also bigger), the structure of the syntactic monoid reveals even more information than the analysis of the minimal automaton.

Furthermore, the syntactic monoid is something one can work with:

Proposition 1. *The syntactic monoid is finite iff the language is regular. Furthermore, the syntactic monoid of a regular language can be effectively computed from any presentation of the language (by automata, regular expressions, etc...)*

2 Schützenberger's Characterization of Star-Free Languages

Our first result concerns star-free languages. The *star-free languages* are the languages of finite words which can be obtained from finite languages using union, concatenation and complement (of course, intersection and set difference can be derived from the union and complement).

An example is simply A^* (for A the alphabet), which is star-free since it is the complement of the empty language, which is itself finite. More generally, B^* for all subsets B of A is star-free since it can be written as $A^* \setminus (\bigcup_{c \notin B} A^*cA^*)$. Very close is the language of words over $A = \{a, b, c\}$ containing no two consecutive occurrences of the letter a. It is star-free since it can be written as $A^* \setminus (A^*aaA^*)$. However, the language of words of even size is not star-free (we do not prove it here). In general, all star-free languages are regular, but the converse does not hold. This motivates the following question:

> When is a regular language star-free? Is it decidable?

Schützenberger answered the above question as follows:

Theorem 2 (Schützenberger[17]). *A regular language is star-free iff it is recognized by a monoid which has only trivial subgroups*[1]*.*

This famous result is now well understood and has been enriched in many ways. In particular, star-free languages are known to coincide with first-order definable languages as well as with the languages accepted by counter-free automata [10]. This result was the starting point of the very important literature aiming in precisely classifying families of languages. See for instance [14].

This result in particular establishes the decidability of being star-free. Indeed, if any monoid recognizing a language has only trivial subgroups, then its syntactic monoid has only trivial subgroups. This yields a decision procedure: construct the syntactic monoid of the language and check that all its subgroups are trivial. The later can be done by an exhaustive check.

We will only prove here the right to left direction of Theorem 2, namely, if a regular language is recognized by some (finite) monoid with only trivial subgroups, then it is star-free. The interested reader can find good expositions of the other directions in many places, see for instance [11]. We assume from now and on that we are given a language L recognised by $\mathbf{M}, \alpha, F$, in which $\mathbf{M}$ is a finite monoid which has only trivial subgroups.

The general approach of the proof is natural. For all elements $a \in \mathbf{M}$, we prove that the language $L_a \overset{\text{def}}{=} \{u \in A \;:\; \alpha(u) = a\}$ is star-free. This concludes the proof of the right to left direction of Theorem 2, since it yields that

$$L = \alpha^{-1}(F) = \bigcup_{a \in F} L_a \quad \text{is star-free.}$$

[1] Here, a subgroup is a subset of the monoid which is equipped of a group structure by the product of the monoid. This terminology is the original one used by Schützenberger. It is natural in the general context of semigroup theory.

However, how do we prove that each L_a is star-free?

Our basic blocks will be languages consisting of words of length one. For each $a \in \mathbf{M}$, set $C_a \overset{\text{def}}{=} \{c \in A \ : \ \alpha(c) = a\}$. This is a finite (hence star-free) language, and $C_a \subseteq L_a$. More precisely, C_a captures all the length-one words in L_a. We could try to get closer to L_a using only concatenations and unions of such elementary languages. However, this would only produce finite languages. This is not sufficient in general. We need another argument. It will take the form of a good induction parameter: we will pre-order the elements of the monoid using one of Green's relations, the $\leq_{\mathcal{J}}$-pre-order, that we introduce now.

In a monoid $\mathbf{M}$, we define the binary relations $\leq_{\mathcal{J}}$ and $\mathcal{J}$ by:

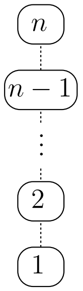
$$a \leq_{\mathcal{J}} b \text{ if } a = xby \text{ for some } x, y \in \mathbf{M}, \qquad a \mathrel{\mathcal{J}} b \text{ if } a \leq_{\mathcal{J}} b \text{ and } b \leq_{\mathcal{J}} a\ .$$

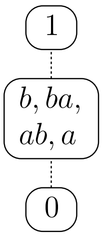
The relation $\leq_{\mathcal{J}}$ is a preorder, and $\mathcal{J}$ is an equivalence relation. In particular, in free monoid of words, $u \geq_{\mathcal{J}} v$ if and only if u appears as a *factor* of v, i.e., v can be written wuw' for some w and w'. We call this the *factor ordering.*

One often describes monoids making explicit the $\mathcal{J}$-pre-order, such as in the following examples.

Our first example is the monoid $(\{1, \ldots, n\}, \min)$. Here, the neutral element is n, and the absorbing element 1. In this case, $\leq_{\mathcal{J}}$ coincide with the usual order $\leq$.

Traditionally, the smaller is an element for the relation $\leq_{\mathcal{J}}$, the lower it appears in the drawing. When one starts from an element and successively performs products to the left or to the right, then the $\mathcal{J}$-class stays the same or goes down. In the later case, one says *falling* in a lower $\mathcal{J}$-class. This supports the intuition behind the relation $\leq_{\mathcal{J}}$ that it captures information that cannot be undone: it is never possible to climb back to 5 from 2 by making any product, to the left or to the right. Informally, "it is impossible to recover from falling in the $\leq_{\mathcal{J}}$ order".





Let us depict now the structure of $\mathcal{J}$-classes of the syntactic monoid of the language "no two consecutive occurrences of the letter a".

Remark the $\mathcal{J}$-equivalence between a, ab, ba, and b (in general, the $\mathcal{J}$-relation is not an order). However, as soon as two consecutive a's are encountered, one falls in the $\mathcal{J}$-class of 0. Once more it is impossible, using any product with a word containing two consecutive a's, to produce one without this pattern.

In general, falling in a $\mathcal{J}$-class can be understood as the discovery of a certain pattern as a factor (here aa, but in general any regular language).

The $\leq_{\mathcal{J}}$-pre-order is a good idea as an induction parameter. Indeed, unless $b \leq_{\mathcal{J}} a$, one does not see how knowing that L_a is star-free can help proving that L_b is also star-free. This is why our proof proceeds by establishing the following induction step.

Induction step: Assuming L_b is star-free for all $b >_{\mathcal{J}} a$, then L_a is star-free.

Assuming this induction step granted for all $a \in \mathbf{M}$, we obtain immediately that L_a is star-free for all $a \in \mathbf{M}$, and hence L itself is star-free. The theorem is established. Let us establish now this induction step. We assume from now and on a fixed, and the hypothesis of the induction step fulfilled. The key lemma is:

Lemma 1. *The language* $L_{\not\geq_{\mathcal{J}} a} \overset{\mathit{def}}{=} \{u \;:\; \alpha(u) \not\geq_{\mathcal{J}} a\}$ *is star-free.*

Proof. It follows from the following equation which witnesses the star-freeness:

$$L_{\not\geq_{\mathcal{J}} a} = A^* K_a A^* \;, \qquad \text{where} \quad K_a \overset{\text{def}}{=} \bigcup_{b \not\geq_{\mathcal{J}} a} C_b \cup \bigcup_{\substack{bcd \not\geq_{\mathcal{J}} a \\ c >_{\mathcal{J}} a}} C_b L_c C_d \;.$$

We prove this equality by establishing a double inclusion. The easiest direction is the inclusion from right to left. Indeed, we have $K_a \subseteq L_{\not\geq_{\mathcal{J}} a}$ by construction. Furthermore, by definition of the $\mathcal{J}$-pre-order, belonging to $L_{\not\geq_{\mathcal{J}} a}$ is preserved under any product to the left or to the right. Hence $A^* K_a A^* \subseteq L_{\not\geq_{\mathcal{J}} a}$.

For the opposite inclusion, we prove that every word in $L_{\not\geq_{\mathcal{J}} a}$ which is minimal (i.e., such that no strict factor belongs to $L_{\not\geq_{\mathcal{J}} a}$) belongs to K_a. Let u be such a minimal word. Clearly u cannot be of length 0, since this would mean $\alpha(u) = 1_{\mathbf{M}} \geq_{\mathcal{J}} a$, and hence $u \notin L_{\not\geq_{\mathcal{J}} a}$. If u has length 1 then it directly falls in $\bigcup_{b \not\geq_{\mathcal{J}} a} C_b$, which is the first part in the definition of K_a. The interesting case is when u has length at least 2. In this case, u can be written as $u = xvy$ for some letters $x, y \in A$.

Let $b = \alpha(x)$, $c = \alpha(v)$, and $d = \alpha(y)$. We claim that $c >_{\mathcal{J}} a$, and for proving this, we use the following lemma:

Lemma 2. *In a finite monoid, if $c \,\mathcal{J}\, bc \,\mathcal{J}\, cd$, then $c \,\mathcal{J}\, bcd$.*

This results is in fact a direct consequence of more elementary results presented below. Let us show this simplicity by giving a proof, though the necessary material has not been yet given.
Assume $c \,\mathcal{J}\, bc \,\mathcal{J}\, cd$, then by Lemma 7, $c \,\mathcal{R}\, cd$. Hence by Lemma 3, $bcd \,\mathcal{R}\, bc$. Thus $bcd \,\mathcal{J}\, bc$. □

For the sake of contradiction, assume $c \not>_{\mathcal{J}} a$. We have $a \leq_{\mathcal{J}} bc$ (by minimality in the choice of u), hence $a \leq_{\mathcal{J}} c$. Combined with $a \not<_{\mathcal{J}} c$, we obtain $a \,\mathcal{J}\, c$. Furthermore we have $c \,\mathcal{J}\, a \leq_{\mathcal{J}} bc \leq_{\mathcal{J}} c$, hence $bc \,\mathcal{J}\, c$ and similarly $cd \,\mathcal{J}\, c$. Using

Lemma 2, we get $bcd\,\mathcal{J}\,c\,\mathcal{J}\,a$. This contradicts $u \in L_{\not\geq_{\mathcal{J}} a}$ since $\alpha(u) = bcd$. Hence $c >_{\mathcal{J}} a$.

Thus, $u \in C_b L_c C_d$, $bcd \not\geq_{\mathcal{J}} a$, and $c >_{\mathcal{J}} a$, i.e., $u \in K_a$. Consider now a word $u \in L_{\not\geq_{\mathcal{J}} a}$. It has a minimal factor $u' \in L_{\not\geq_{\mathcal{J}} a}$. We have seen that $u' \in K_a$. Hence $u \in A^* K_a A^*$. □

We immediately deduce:

Corollary 1. *The language* $L_{\mathcal{J}a} = \{u \ :\ \alpha(u)\ \mathcal{J}\ a\}$ *is star-free.*

Proof. This follows from the equation $L_{\mathcal{J}a} = \bigcap_{b >_{\mathcal{J}} a} L_{\not\geq_{\mathcal{J}} b} \setminus L_{\not\geq_{\mathcal{J}} a}$. □

At this point, we are able to define the $\mathcal{J}$-class of a. However, we need to be even more precise, and define precisely a. We will need some more of Green's relations.

The order $\leq_{\mathcal{J}}$ makes no distinction on whether the products are performed on the left or on the right. The relations $\leq_{\mathcal{L}}$ and $\leq_{\mathcal{R}}$ refine the $\leq_{\mathcal{J}}$ order as follows:

$$\begin{array}{llll} a \leq_{\mathcal{L}} b & \text{if } a = xb \text{ for some } x \in \mathbf{M}, & & a\,\mathcal{L}\,b \ \text{ if } a \leq_{\mathcal{L}} b \text{ and } b \leq_{\mathcal{L}} a\ , \\ a \leq_{\mathcal{R}} b & \text{if } a = bx \text{ for some } x \in \mathbf{M}, & \text{and} & a\,\mathcal{R}\,b \ \text{ if } a \leq_{\mathcal{R}} b \text{ and } b \leq_{\mathcal{R}} a\ . \end{array}$$

An elementary, but important, property of these relations is:

Lemma 3. *If* $a \leq_{\mathcal{L}} b$ *then* $ac \leq_{\mathcal{L}} bc$. *If* $a\,\mathcal{L}\,b$ *then* $ac\,\mathcal{L}\,bc$.
If $a \leq_{\mathcal{R}} b$ *then* $ca \leq_{\mathcal{R}} cb$. *If* $a\,\mathcal{R}\,b$ *then* $ca\,\mathcal{R}\,cb$.

The relation $\mathcal{L}$ considers equivalent elements which are convertible one into the other by product on the left. One can think of the piece of informations preserved by such conversions as located "close to the right" of the element. Considering that $\mathcal{L}$ identifies information concerning the "right" of the element, and $\mathcal{R}$ information which concerns the "left" of the element, the two relations $\mathcal{L}$ and $\mathcal{R}$ may seem rather independent. This intuitive idea is captured by the following key lemma.

Lemma 4. $\mathcal{L} \circ \mathcal{R} = \mathcal{R} \circ \mathcal{L}$.

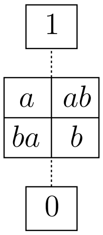
Thus, we define $\mathcal{D} \stackrel{\text{def}}{=} \mathcal{L} \circ \mathcal{R}$. In general, we have $\mathcal{D} \subseteq \mathcal{J}$, but:

Lemma 5. *In a finite monoid,* $\mathcal{D} = \mathcal{J}$.

This result is the central one in the theory of finite monoids. All results presented in this lecture using the finiteness assumption are more or less directly derived from this lemma.

A consequence for us is that we can depict monoids in a refined way.

The presentation of the syntactic monoid of the language "no two consecutive occurrences of a" can be refined as shown. The $\mathcal{J}$-class $\{a, ab, ba, b\}$ has been subdivided according to the $\mathcal{L}$-classes and the $\mathcal{R}$-classes, yielding an "egg-box" presentation of each class. The $\mathcal{R}$-classes are drawn as '$\mathcal{R}$'ows (here $\{1\}$, $\{a, ab\}$, $\{b, ba\}$ and $\{0\}$), and the $\mathcal{L}$-classes as columns (here $\{1\}$, $\{a, ba\}$, $\{b, ab\}$ and $\{0\}$). The last of Green's relations is $\mathcal{H}$, defined by:



$$\mathcal{H} \overset{\text{def}}{=} \mathcal{L} \cap \mathcal{R} \ .$$

Quite naturally $\mathcal{H}$-classes correspond to the atomic boxes at the intersection of rows and columns. In our example, all $\mathcal{H}$-classes are singletons.

For groups, on the contrary, there is only one $\mathcal{H}$-class.

Our next objective is to identify the $\mathcal{R}$-class and $\mathcal{L}$-class of a.

Lemma 6. *The language* $L_{\mathcal{R}a} \overset{\text{def}}{=} \{u \ : \ \alpha(u) \mathrel{\mathcal{R}} a\}$ *is star-free.*

Proof. Assume first (the interesting case), that a is not $\mathcal{R}$-equivalent to $1_{\mathbf{M}}$. In this case, the star-freeness of $L_{\mathcal{R}a}$ is established by the following equation:

$$L_{\mathcal{R}a} = L_{\mathcal{J}a} \cap \left(\bigcup_{bc \leq_{\mathcal{R}} a,\ b >_{\mathcal{J}} a} L_b C_c A^* \right) \ .$$

Clearly, if a word u belongs to the right-hand side of the equation, this means it has a prefix v belonging to $L_b C_c$. We derive $\alpha(u) \leq_{\mathcal{R}} \alpha(v) = bc \leq_{\mathcal{R}} a$. Since furthermore $\alpha(u) \mathrel{\mathcal{J}} a$, one can use the following important lemma:

Lemma 7. *In a finite monoid,* $b \leq_{\mathcal{R}} a$ *and* $a \mathrel{\mathcal{J}} b$ *implies* $a \mathrel{\mathcal{R}} b$.

(Said differently, if $ab \mathrel{\mathcal{J}} a$ then $ab \mathrel{\mathcal{R}} a$.)

from which we immediately get that $\alpha(u) \mathrel{\mathcal{R}} a$, i.e., $u \in L_{\mathcal{R}a}$.

Conversely, consider some $u \in L_{\mathcal{R}a}$. First of all, clearly $u \in L_{\mathcal{J}a}$. Let v be a minimal prefix of u such that $\alpha(v) \leq_{\mathcal{R}} a$. Since $1_{\mathbf{M}} \not\leq_{\mathcal{R}} a$ we have $\alpha(v) \neq 1_{\mathbf{M}}$, and hence $v \neq \varepsilon$. Thus we can write v as wx for some letter x. Setting $b = \alpha(w)$ and $c = \alpha(x)$, we have that u belongs to $L_b C_c A^*$. Furthermore $bc = \alpha(v) \geq_{\mathcal{R}} \alpha(u) \geq_{\mathcal{R}} a$. Finally, by minimality of v, $b = \alpha(w) \not\leq_{\mathcal{R}} a$. Since furthermore $b = \alpha(w) \geq_{\mathcal{R}} \alpha(u) \geq_{\mathcal{R}} a$, we obtain $b >_{\mathcal{R}} a$. This means by Lemma 7 that $b >_{\mathcal{J}} a$.

It remains the case $1_{\mathbf{M}} \in L_{\mathcal{R}a}$. We use the following lemma.

Lemma 8. *In a finite monoid* $\mathbf{M}$*, the* $\mathcal{J}$*-class of* $1_{\mathbf{M}}$ *coincides with its* $\mathcal{H}$*-class.*

Proof. Assume $1_{\mathbf{M}} \mathrel{\mathcal{J}} a$. Since furthermore $a \leq_{\mathcal{R}} 1_{\mathbf{M}}$ (by $a = 1_{\mathbf{M}}a$), we have $a \mathrel{\mathcal{R}} 1_{\mathbf{M}}$ using Lemma 7. Similarly, $a \mathrel{\mathcal{L}} 1_{\mathbf{M}}$. Thus $a \mathrel{\mathcal{H}} 1_{\mathbf{M}}$. □

Hence $L_{\mathcal{R}a} = L_{\mathcal{J}a}$ is star-free by Corollary 1. □

By symmetry, $L_{\mathcal{L}a} \overset{\text{def}}{=} \{u \;:\; \alpha(u) \mathrel{\mathcal{L}} a\}$ is also star-free. From which we get.

Corollary 2. *The language* $L_{\mathcal{H}a} \overset{\text{def}}{=} \{u \;:\; \alpha(u) \mathrel{\mathcal{H}} a\}$ *is star-free.*

Proof. Indeed $L_{\mathcal{H}a} = L_{\mathcal{R}a} \cap L_{\mathcal{L}a}$, and both $L_{\mathcal{L}a}$ and $L_{\mathcal{R}a}$ are star-free. □

Here the poof is concluded using the next lemma.

A monoid is called *$\mathcal{H}$-trivial* if all its $\mathcal{H}$-classes are singltons.

Lemma 9. *A monoid has only trivial subgroups if and only if it is* $\mathcal{H}$*-trivial.*

One direction is simple. Indeed, if you consider any subgroup of the monoid, then all its elements are $\mathcal{H}$-equivalent in the monoid. Thus $\mathcal{H}$-trivial implies that all subgroups are trivial. The converse requires more work.

Such monoids are also called *aperiodic*, which signifies that there exists some n such that $a^n = a^{n+1}$ for all $a \in \mathbf{M}$.

Hence, we deduce that $L_a = L_{\mathcal{H}a}$ which is star-free by Corollary 2. This completes the proof of the induction step, and hence the proof of the theorem of Schützenberger.

3 The Theorem of Factorization Forests of Simon

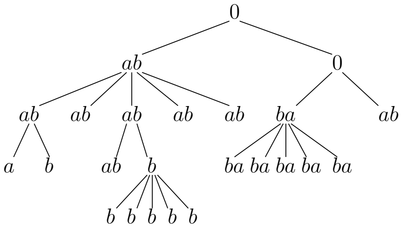
The theorem of forest factorizations is due to Simon [19]. It is at first glance, a purely algebraic result. It takes a finite monoid as input, as well as a morphism from words to this monoid, and shows the existence of factorizations with special properties for every word. However, this result has consequences which are purely automata related. Some of them are surveyed in [1]. A remarkable application is its use for proving the decidability of the limitedness problem for distance automata. Distance automata [6] are non-deterministic finite state automata over finite words, equipped with a set of special transitions. The function computed by such an automaton associates to each input word the minimal number of special transitions contained in some accepting run (∞ if there are no accepting runs). The limitedness problem consists in deciding whether the function computed by a distance automaton is bounded (in fact, over its domain, i.e., the set of words which are not given value ∞). This problem is decidable [6], and the optimal algorithm for it is due to Leung [8]. A simplified proof of validity of this theorem is due to Simon using the factorization forest theorem [20]. The theorem of factorization forests has other interesting applications. It is used for instance for characterizing the polynomial closure of classes of regular languages [15].

We fix from now and on a finite monoid $\mathbf{M}$.We will work with *sequences* of elements in the monoid. We denote them separated by commas for avoiding confusion with the product. Given a sequence $v = a_1, \dots, a_n$, $\pi(v)$ denotes the value $a_1 a_2 \cdots a_n$.

A *factorization (tree)* (over the monoid $\mathbf{M}$) is a finite unranked ordered tree T whose leaves are labelled by elements of $\mathbf{M}$ (the label of node x is denoted $T(x)$) such that for every non-leaf node x of children $y_1, \dots, y_k$ (read from left to right), $T(x) = \pi(T(y_1), \dots, T(y_k))$. A *factorization of a sequence* $a_1, \dots, a_n$ (of element in $\mathbf{M}$) is a factorization tree such that the labels of leaves read from left to right are $a_1, \dots, a_n$. Traditionally, the height of a factorization is computed with leaves excluded, i.e., the height of a single leaf factorization is by convention 0.

A factorization is *Ramsey* if for all nodes x of rank[2] three or more, its children $y_1, \dots, y_k$ are such that $T(y_1) = \dots = T(y_n) = e$ where e is an idempotent (in particular, we also have $T(x) = e$).

Here is for instance an example of a Ramsey factorization of height 4, in the context of the syntactic monoid of our running example: the language of words "without two consecutive occurrences of letter a".



The theorem of factorization forests is then the following.

Theorem 3 (Simon [19]). *Every sequence of elements from a finite monoid M admits a Ramsey factorization of height at most* $3|M| - 1$.

We follow here the similar proofs from [4, 7]. The original bound given by Simon is $9|M|$ instead of $3|M|$. The bound of $3|M|-1$ has been announced by Kufleitner, but is proved here. The proof completely follows the descriptions of the monoid in terms of the Green's relations.

A sequence $a_1, \dots, a_n$ over M will be called *X-smooth* for some $X \subseteq \mathbf{M}$ if $a_i \cdots a_j \in X$ for all $1 \leq i \leq j \leq n$ (in particular each $a_i \in X$ and $a_1 \cdots a_n \in X$).

Lemma 10. *Let H be an $\mathcal{H}$-class of a finite monoid M, then every H-smooth sequence admits a Ramsey factorization of height at most* $3|H| - 1$.

Proof. We need a better understanding of the internal structure of $\mathcal{H}$-classes:

[2] The rank of a node is the number of its children.

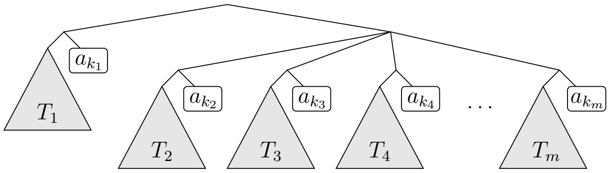
Lemma 11. *If an $\mathcal{H}$-class H contains an idempotent e, it is a group of neutral element e. Otherwise $ab <_{\mathcal{J}} a$ for all $a, b \in H$*

If H contains no idempotent element, then an H-smooth sequence has length at most 1. In this case, the single node factorization is Ramsey. It has height 1.

The interesting case is when H contains an idempotent e. Given an H-smooth sequence $u = a_1, \dots, a_n$, call its *width* the value $|S(a_1, \dots, a_n)|$ where the set $S(a_1, \dots, a_n)$ abbreviates $\{a_1 \dots a_l \ : \ 1 \leq l \leq n\}$ (remark that $S(a_1, \dots, a_n)$ is empty iff $n = 0$). The construction is by induction on the width of the sequence. The base case is for $n = 0$. In this case, there exists a factorization of height at most 0 (this is a somewhat distorted case).

Let $v = a_1, \dots, a_n$ for some $n > 0$. Define a to be $a_1 \cdots a_n$ and let a' be the inverse of a in the group H. Set $0 < k_1 < \cdots < k_m = n$ to be the list of indexes such that $a_1 \cdots a_{k_i} = a$ for all i. Let $v_1 = a_1, \dots, a_{k_1 - 1}$, $v_2 = a_{k_1+1}, \dots, a_{k_2-1}$, etc... Remark that $S(v_1) \subseteq S(v)$ and $a \notin S(v_1)$. Hence the width of v_1 is less than the one of v. One can apply the induction hypothesis, and get a Ramsey factorization T_1 for v_1. Similarly for all $i > 1$, $aS(v_i) \subseteq S(v)$, and $a \notin S(v_i)$. Furthermore $S(v_i) = a'aS(v_i)$, hence $|S(v_i)| \leq |aS(v_i)| < |S(v)|$. Thus, we can also apply the induction hypothesis, and get a Ramsey factorization T_i for v_i. Remark here that some of the v_i's may be empty. We do not pay attention to this harmless detail.

We now construct the following factorization:



In this construction, the empty trees are removed, and the labels of nodes are completed (in a unique way). We have to prove that the resulting factorization is indeed Ramsey. For this, it is sufficient to prove it for the only new node of rank possibly greater or equal to 3. Its i'th children has value $a_{k_i+1} \cdots a_{k_{i+1}}$. We compute:

$$\begin{aligned} a_{k_i+1} a_{k_2} &= a' a a_{k_i+1} \cdots a_{k_2} \\ &= a' a_1 \cdots a_{k_i} a_{k_i+1} \cdots a_{k_2} \\ &= a' a = e \ . \end{aligned}$$

Hence the new node is Ramsey.

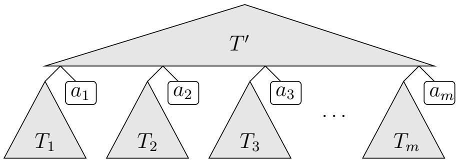
For the height, remark that for the width 1, all the T_i's are empty, and hence the construction can be simplified, and the resulting height is 2 (recall that leaves do not count in the height). At each induction step, the height of the factorization increases by at most 3. Hence, in the end, the factorization resulting from this construction has height at most $3|H|-1$. □

Lemma 12. *For R an $\mathcal{R}$-class, every R-smooth sequence has a Ramsey factorization of height at most $3|R|-1$.*

Proof. Let $v = a_1, \dots, a_n$ be the R-smooth sequence. The construction is by induction on the number of $\mathcal{H}$-classes occurring in v. If this number is 0, then one uses (once more) the distorted case of an empty tree.

Otherwise, let H be the $\mathcal{H}$-class of a_n. Let $1 \leq k_1 < \cdots < k_m \leq n$ be the indexes such that $a_{k_i} \in H$. One defines as for the previous lemma $v_1,\dots,v_{m+1}$ to be such that $v = v_1, a_1, v_2, \dots, a_m, v_{m+1}$. Remark first that the $\mathcal{H}$-class H does not appear in any of $v_1, \dots, v_{m+1}$. Thus, one can apply the induction hypothesis for each of the v_i's, and get a Ramsey factorization T_i. We also know that $\pi(v_i, a_i) = \pi(v_i)a_i$ and hence $\pi(v_i, a_i) \leq_{\mathcal{L}} a_i$. It follows from (the $\mathcal{L}$-version of) Lemma 7 that $\pi(v_i, a_i) \mathrel{\mathcal{L}} a_i$, which means $\pi(v_i, a_i) \mathrel{\mathcal{L}} a_i \in H$. Hence, we can apply Lemma 10, and get a Ramsey factorization T' for $\pi(v_1, a_1), \dots, \pi(v_m, a_m)$.

We now construct the following factorization:



It is Ramsey since each part it is composed of (namely $T_1, \dots, T_{m+1}$ and T') is Ramsey. One just needs to check that the values are consistent at the glue points. However, this is by construction.

Concerning its height. Once more in the pathological case of a single $\mathcal{H}$-class, the construction gets simpler, and its height is simply the height of T', which is at most $3|H|-1$. Then at each step of the induction, the height increases of at most $3|H|-1$ where H is the $\mathcal{H}$-class treated at this step of the induction. Overall we can over approximate it by $3|R|-1$. □

In the above proof, we count the size of all the $\mathcal{H}$-classes separately. However, in some situations we would like to have more information. Such results exist:

Lemma 13 (Green's lemma). *Inside a $\mathcal{D}$-class,*

- *The $\mathcal{R}$-classes have all the same size (more precisely, if $ba \mathrel{\mathcal{D}} a$, then the application which to x associates bx is a bijection from the $\mathcal{R}$-class of a onto the $\mathcal{R}$-class of ba).*
- *The $\mathcal{L}$-classes have all the same size, (more precisely, if $ab \mathrel{\mathcal{D}} a$, then the application which to x associates xb is a bijection from the $\mathcal{L}$-class of a onto the $\mathcal{L}$-class of ab).*
- *All $\mathcal{H}$-classes have same size.*

Using the exact same proof, decomposing a $\mathcal{J}$-class into $\mathcal{R}$-classes, we obtain:

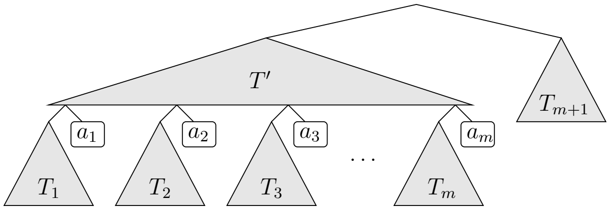
Lemma 14. *For J a $\mathcal{J}$-class, every J-smooth sequence has a Ramsey factorization of height at most $3|J|-1$.*

We are now ready to prove the factorization forest theorem, Theorem 3.

Proof. Let $v = a_1, \ldots, a_n$ be a sequence over $\mathbf{M}$. This time, the proof is by induction on the $\mathcal{J}$-class J of $\pi(v)$. Assume one knows how to construct a Ramsey factorization for each sequence w such that $\pi(w) >_{\mathcal{J}} \pi(v)$.

Let $k_1 \geq 1$ be the least index such that $\pi(a_1, \ldots, a_{k_1}) \in J$. Continue by constructing $k_2 > k_1$ minimal such that $\pi(a_{k_1+1}, \ldots, a_{k_2}) \in J$, and so on, producing in the end $k_1 < \cdots < k_m$. One decomposes v as $v_1, a_{k_1}, v_2, \ldots, a_{k_m}, v_{m+1}$ as expected. By minimality property in the definition of the k_i's, $\pi(v_i) \notin J$. Since furthermore, $\pi(v_i) \geq_{\mathcal{J}} \pi(v) \in J$, we obtain $\pi(v_i) >_{\mathcal{J}} \pi(v)$. Thus we can apply the induction hypothesis, and get a Ramsey factorization T_i for v_i. Furthermore, for all $i \leq m$, $\pi(v_i, a_i) \in J$ by construction. Hence, we can apply Lemma 14 and get a Ramsey factorization T' for $\pi(v_1, a_1), \ldots, \pi(v_m, a_m)$.

We construct now the following factorization:



As in the previous lemmas, it is Ramsey simply because all its components are Ramsey.

Concerning the height. For a maximal $\mathcal{J}$-class J, the construction can be slightly simplified since all the T_i's are empty. Hence, the height is the one of T', which is at most $3|J|-1$. Then, the height increases of the height of T' plus one at each step of the induction, which is at most $3|J|$ for a $\mathcal{J}$-class J. Overall, we reach a factorization of height at most $3|M|-1$. □

An interesting point in this proof is that it is completely driven by the decomposition of the monoid according to Green's relations.

4 On ω-Semigroups and Monadic Logic

The remaining results which we consider involve the study of languages of infinite words, of length ω. We call such words *ω-words*. Regular languages of ω-words are usually defined using Büchi automata. We present in this section the corresponding algebraic notion, ω-semigroups. This is the extension of the notion of monoids (in fact semigroups) which is best suited for dealing with languages of infinite words.

***ω*-semigroups.** An *ω-semigroup* is an algebra $\mathbf{S} = (S_+, S_\omega, \pi)$ consisting of two disjoint sets S_+ and S_ω and a product π mapping finite sequences in $(S_+)^+$ to S_+, finite sequences in $(S_+)^* S_\omega$ to S_ω, and infinite sequences in $(S_+)^\omega$ to S_ω. The product π is required to be associative, i.e., for all meaningful choices of sequences u_1, u_2,...,

$$\pi(\pi(u_1), \pi(u_2), \dots) = \pi(u_1, u_2, \dots) \ .$$

As an example, the *free ω-semigroup* generated by A consists of $S_+ = A^+$, $S_\omega = A^\omega$, and the product is simply the concatenation product for sequences (possibly infinite).

An example of a finite[3] ω-semigroup consists in $S_+ = \{a, b\}$ and $S_\omega = \{0, 1\}$. For all finite sequences u over $\{a, b\}$, $\pi(u) = a$ if a occurs in u, b otherwise. For all finite sequences u, $\pi(u, 0) = 0$ and $\pi(u, 1) = 1$, and for all ω-sequences w over $\{a, b\}$, $\pi(w) = 1$ if w contains infinitely many occurrences of a, $\pi(w) = 0$ otherwise.

A *morphism of ω semigroups* from $\mathbf{S}$ to $\mathbf{S}'$ is an application α mapping S_+ to S'_+ and S_ω to S'_ω which preserves the product, i.e., such that for all meaningful (possibly infinite) sequences $a_1, a_2, \dots$ from S,

$$\alpha(\pi(a_1, a_2, \dots)) = \pi'(\alpha(a_1), \alpha(a_2), \dots) \ .$$

A language of ω-words L is *recognizable* by an ω-semigroup $\mathbf{S}$ if there exists a morphism α from the free ω-semigroup to S, such that for every ω-word w, $w \in L$ if and only if $\alpha(w) \in F$, where $F \subseteq S_\omega$. One also says that L is *recognized by* $\mathbf{S}, \alpha, F$.

For instance, the language of infinite words over $\{a, b, c\}$ which contains infinitely many occurrences of letter a is recognized by the above finite ω-semigroup. The morphism α maps each non-empty finite word to a if it contains an occurrence of the letter a, to b otherwise. The morphism also sends each ω-word to 1 if it contains infinitely many occurrences of the letter a, to 0 otherwise. The finite subset of S_ω is $F = \{1\}$.

Given an ω-semigroup (S_+, S_ω, π), one defines the binary product $\cdot$ over S_+ by $ab \overset{\text{def}}{=} \pi(a, b)$. One also uses it from $S_+ \times S_\omega$ to S_ω (with the same definition). The exponentiation by ω from S_+ to S_ω is defined with $a^\omega \overset{\text{def}}{=} \pi(a, a, a, \dots)$. It turns out that if S_+ and S_ω are finite, then π is entirely determined by the product $\cdot$ and the exponent by ω [22]. We will not use this direction, however, this explains why it is sufficient to know a finite amount of information for working effectively with ω-semigroups.

The relationship with the monoids we have been using so far is that $(S_+, \cdot)$ is a semigroup: a *semigroup* $\mathbf{S} = (S, \cdot)$ is a set S together with an associative operator $\cdot$. Hence, this is simply a monoid without necessary a neutral element. This difference is not essential. In our case, it simply reflects the special property of the empty word (which is the neutral element of the free monoid) in the study

[3] Finite means that both S_+ and S_ω are finite, though *a priori*, the product π requires an infinite quantity of information for being described.

of infinite words: it is the only finite word which, when iterated ω times, does not yield an infinite word. Using semigroups rather than monoids means avoiding to treat this particular case.

The structure of semigroups and monoids are highly related. Given a semigroup $\mathbf{S}$, one defines $\mathbf{S}^1$ to be $\mathbf{S}$ to which has been added a new neutral element 1, if necessary. This makes a monoid out of a semigroup. When we refer to the Green's relations of the semigroup, we refer in fact implicitly to the Green's relations of the corresponding monoid $\mathbf{S}^1$.

Monadic second-order logic. Let us recall here that *monadic (second-order) logic* is the first-order logic extended with the ability to quantify over sets. I.e., it is possible to quantify existentially or universally over elements (e.g., $\exists x, \forall y, \dots$) and sets of elements (e.g., $\exists X, \forall Y, \dots$), to test membership (e.g., $x \in Y$), to use boolean connectives (i.e., $\vee$,$\wedge$, $\neg$) and to use the predicates of the structure. In our case, we consider ω-words. In this case, the elements are the positions in the word, i.e., non-negative integers, and there are two kinds of predicates. For all letters a, the predicate $a(x)$ allows to test whether the letter at the position x is an a, and the predicate $x \leq y$ tests whether the position x occurs to the left of or at y. Given an ω-word, one says that its *monadic theory is decidable* if there is an algorithm which, given any sentence of monadic logic, decides whether it holds or not over the ω-word. See for instance [21] for more on the subject.

Those various notions are tied together by the following theorem.

Theorem 4 ([2],[12]). *A language of ω-words is regular (i.e., definable by Büchi automata) if and only if it is recognized by a finite ω-semigroup, if and only if it is definable in monadic logic. Furthermore, the translations are effective.*

For this reason, we do not use explicitly monadic logic from now on.

5 The Characterization of Decidable Theories by Semenov

The result of Semenov we are mentioning answers the following question:

> When is the monadic theory of an ω-word decidable?

This question differs, though it is related, to the original question solved by Büchi [2], which aims at deciding whether a monadic sentence has a model, i.e., decide if it is satisfied by some ω-word.

A possible answer to the above question is:

Theorem 5 (variation of Semenov[18]). *An ω-word w has a decidable monadic theory if and only if the following questions are decidable for all regular languages of finite words L:*

(A) Does there exist a finite prefix of w in L? I.e., $w \in LA^\omega$?
(B) Does there exist recurrent factors of L in w ? I.e., $w \in (A^*L)^\omega$?

One direction is straightforward. Indeed, it is easy to translate the properties "$w \in LA^\omega$" and "$w \in (A^*L)^\omega$" into equivalent monadic formulas. Hence properties (A) and (B) can be reduced to the decidability of the monadic theory of w.

The interesting implication is the opposite one. We will use as our major tool Lemma 18 below. This requires beforehand to disclose some extra facts concerning the Green's relations.

An element a in a monoid is called *regular* if there exists s such that $asa = a$.

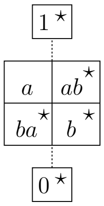
Lemma 15. *Let J be a $\mathcal{J}$-class J in a finite monoid. The following items are equivalent:*

- *J contains a regular element,*
- *J contains an idempotent,*
- *every element in J is regular,*
- *every $\mathcal{R}$-class in J contains an idempotent,*
- *every $\mathcal{L}$-class in J contains an idempotent,*
- *there exist two elements in J, the product of which belongs to J.*

Such $\mathcal{J}$-classes are naturally called *regular*.

Keeping the same example as above, one enriches the presentation by adding information concerning the idempotents. Each $\mathcal{H}$-class is now decorated by a star if it contains an idempotent.

This gives an important information:



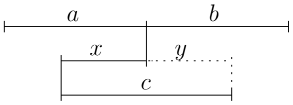
Lemma 16. *In a finite semigroup, if $a \mathcal{J} b$, then $ab \mathcal{J} a$ if and only if there exists an idempotent e such that $e \mathcal{R} b$ and $e \mathcal{L} a$, and in this case, $ab \mathcal{R} a$ and $ab \mathcal{L} b$.*

In our example, on the one hand, $abba$ stays in the same $\mathcal{J}$-class since b is an idempotent, and the result is a. On the other hand, $baab$ falls in the lower $\mathcal{J}$-class since a is not an idempotent.

The following technical lemma contains the key arguments we need:

Lemma 17. *If $a\mathcal{J}b\mathcal{J}ab\mathcal{J}c$ and $a\mathcal{L}x\mathcal{R}c$, then there exists y such that $c = xy$ and $c \mathcal{L} y \mathcal{R} b$.*

This statement can be depicted as shown. I.e., x can be completed to the left into a ($a\mathcal{L}x$), and to the right into c ($x\mathcal{R}c$). Then it is possible to complete x to the right into $c = xy$ such that y can be completed into b ($b \mathcal{R} y$).



Proof. We have $x \mathcal{L} a$, hence $ab \mathcal{L} xb$ by Lemma 3. Thus $xb \mathcal{J} ab \mathcal{J} x$. It follows $xb \mathcal{R} x \mathcal{R} c$, once more by Lemma 3. Hence $c = xbz$ for some z. Let $y = bz$. Clearly $c = xbz = xy$. Furthermore $y = bz \leq_{\mathcal{R}} b$, and $bz \geq_{\mathcal{L}} xbz = c$. Hence by (twice) Lemma 7, $c \mathcal{L} y \mathcal{R} b$. □

We then obtain by iterated applications of Lemma 17.

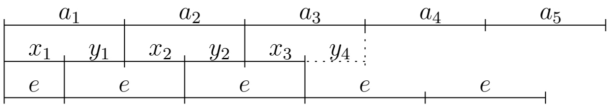
Lemma 18. *If $u = a_1, a_2, \ldots \in (S_+)^\omega$ is $\mathcal{J}$-smooth (i.e., is J-smooth for some $\mathcal{J}$-class J) then (a) there exists an idempotent e such that $e \,\mathcal{R}\, a_1$, and (b) $\pi(u) = e^\omega$ for all idempotents e such that $e \,\mathcal{R}\, a_1$.*

Proof. Let J be the $\mathcal{J}$-class of a_1. Since a_1, a_2, and $a_1 a_2$ all belong to J, it follows that J is regular by Lemma 15. Hence, still by Lemma 15, there exists an idempotent e in the $\mathcal{R}$-class of a_1 (a).

Let e be such an idempotent. One constructs inductively the elements $x_n, y_n \in S_+$ such that for all positive integer n:

(1) $e \,\mathcal{L}\, x_n \,\mathcal{R}\, a_n$, and if $n = 1$, $x_1 = e$, otherwise $y_{n-1} x_n = e$,
(2) $a_n \,\mathcal{L}\, y_n \,\mathcal{R}\, e$, and $x_n y_n = a_n$.

The constructions can be illustrated as follows:



For $n = 1$, one chooses $x_1 = e$, and we know by choice of e that $e \,\mathcal{L}\, x_1 \,\mathcal{R}\, a_1$. Hence (a) holds for $n = 1$. Then for all n, assuming (1) establishes (2) using simply Lemma 17. Similarly, assuming (2), one establishes (1) for $n+1$ using again Lemma 17.

It is then easy to conclude. Indeed, using the associativity of π, we have $\pi(a_1, a_2, \ldots) = \pi(x_1 y_1, x_2 y_2, \ldots) = \pi(x_1, y_1, x_2, y_2, \ldots) = \pi(x_1, y_1 x_2, y_2 x_3, \ldots) = e^\omega$. Property (b) holds. □

Lemma 18 provides us a lot of information concerning the monadic theory of an ω-word. Given a $\mathcal{J}$-class J and a word w, we say that J *occurs in w* if $w \in A^* L_J A^\omega$ where $L_J = \{u \in A^+ \ :\ \alpha(u) \in J\}$. We say that J *is recurrent in w* if $w \in (A^* L_J)^\omega$.

Lemma 19. *For all w, there is a minimum (for the $\mathcal{J}$-pre-order) $\mathcal{J}$-class recurrent in w.*

Proof. Assume that both J and J' are recurrent, we shall prove that there is a $\mathcal{J}$-class below or equal to both J and J' which is recurrent. Since both J and J' are recurrent, w can be written as $u_1 v_1 u'_1 v'_1 u_2 u'_2 \ldots$, where $v_i \in L_J$ and $v'_i \in L_{J'}$ for all i. Let J_i be the $\mathcal{J}$-class of $\alpha(v_i u'_i v'_i)$. By definition of the $\mathcal{J}$-pre-order, $J_i \leq_{\mathcal{J}} J$ and $J_i \leq_{\mathcal{J}} J'$. Furthermore, since there are only finitely many $\mathcal{J}$-class, one of the J_i's has to occur infinitely often, which means it is recurrent in w. □

Lemma 20. *If J is recurrent in w, and no $\mathcal{J}$-class $J' \not\geq_{\mathcal{J}} J$ occurs in w, then w can be decomposed into $v_1 v_2 \ldots$ such that $\alpha(v_1), \alpha(v_2), \ldots$ is J-smooth.*

Proof. Find the first non-empty prefix v_1 of w such that $\alpha(v_1) \in J$. This is possible since J is recurrent. Then proceeds with the remaining suffix, and construct v_2,... For the sake of contradiction, assume $\alpha(v_1), \alpha(v_2), \dots$ is not $\mathcal{J}$-smooth, this would mean that $\alpha(v_i \dots v_j) \notin J$ for some $i \leq j$. However, we have $\alpha(v_i \dots v_j) \leq_{\mathcal{J}} \alpha(v_i) \mathrel{\mathcal{J}} J$, thus this means that $\alpha(v_i \dots v_j) <_{\mathcal{J}} J$. A contradiction since by hypothesis no $\mathcal{J}$-class below J does occur in w. □

Corollary 3. *Assume the minimum recurrent $\mathcal{J}$-class of w is J and all $\mathcal{J}$-classes occurring in w are above or equal to J, then:*

- *w has a finite prefix u such that $\alpha(u) \in J$,*
- *for all prefix u of w such that $\alpha(u) \in J$,*

$$\alpha(w) = \alpha(u)^{\rightarrow} ,$$

where $a^{\rightarrow} \overset{def}{=} e^{\omega}$ for some/all idempotents e such that $e \mathrel{\mathcal{R}} a$ (if defined).

Proof. By hypothesis and Lemma 20, w can be written as $w = v_1 v_2 \dots$ such that $\alpha(v_1), \alpha(v_2), \dots$ is $\mathcal{J}$-smooth. Thus by Lemma 18, $\alpha(w) = \alpha(v_1)^{\rightarrow}$.

Furthermore, u is either a prefix or a suffix of v_1, yielding $\alpha(u) \geq_{\mathcal{J}} \alpha(v_1)$ or $\alpha(u) \leq_{\mathcal{J}} \alpha(v_1)$ respectively. In any case, since $\alpha(u) \mathrel{\mathcal{J}} \alpha(v_1)$, we have $\alpha(u) \mathrel{\mathcal{R}} \alpha(v_1)$ by Lemma 7. Hence $e \mathrel{\mathcal{R}} \alpha(v_1)$ if and only if $e \mathrel{\mathcal{R}} \alpha(u)$. This means $\alpha(w) = \alpha(v_1)^{\rightarrow} = \alpha(u)^{\rightarrow}$. □

We are now ready to establish Theorem 5.

Proof. Assume that properties (A) and (B) hold for an ω-word w, and that one is given a monadic sentence φ. This sentence φ defines a regular language of ω-words which is recognized by an ω-semigroup $\mathbf{S}$ by Theorem 4.

Using (A), we can decide what is the minimum recurrent $\mathcal{J}$-class in w (it exists by Lemma 19). Call it J. The next step consists in finding a decomposition of w in uw' such that all $\mathcal{J}$-class occurring in w' are above or equal to J. Such a word u exists. For finding it, one just tries all possible u's, and stop when both $w \in uA^{\omega}$, and $w \notin uA^{*}L_{\not\geq_{\mathcal{J}} J}A^{\omega}$, where $L_{\not\geq_{\mathcal{J}} J}$ is the set of non-empty words which are not mapped by α to J or above. This is obviously doable using iterated applications of item (A). Then one finds v such that uv is a prefix of w, and $\alpha(v) \in J$. It is sufficient once more to test all possible such v's using (A). Then, we have $\alpha(w) = \alpha(u)\alpha(v)^{\rightarrow}$ by Corollary 3. Hence, we can decide if φ holds or not. □

6 Deterministic Automata over ω-Words: McNaughton

A *Büchi automaton* is a tuple (Q, A, I, Δ, B) where Q is a finite set of *states*, A is the alphabet, $I \subseteq Q$ is a set of *initial states*, $\Delta \subseteq Q \times A \times Q$ is the *transition relation*, and $B \subseteq \Delta$ is the set of *Büchi transitions*. An automaton is deterministic if Δ is a function from $Q \times A$ to Q. A *run of the automaton over an ω-word w* is defined as usual as an infinite sequence of transitions such that the first state is initial, the letters in the transitions yield w, and consecutive transitions agree on the common states. A run is *accepting* if it contains infinitely many Büchi transitions. The *language accepted* by an automaton $\mathcal{A}$ is the set of ω-words over which there is an accepting run of the automaton. A language is said *regular* if it is accepted by some Büchi automaton. A language is *deterministic Büchi* if it is accepted by a deterministic Büchi automaton.

It is known that not all regular languages are deterministic Büchi. However McNaughton's result still gives a strong relationship:

Theorem 6. *A language of ω-words is regular if and only if it is a Boolean combination of deterministic Büchi languages.*

Usually, this theorem is stated as the existence of deterministic automata belonging to more general classes of automata (such as parity/Rabin/Streett/Müller automata). In fact, with just a slightly more involved construction, one can immediately get a deterministic parity automaton along the lines presented here. We choose here the simplest presentation. Standard constructions are directly performed on the automaton, here we translate an ω-semigroup directly into a Boolean combination of deterministic Büchi automata. The first direct translation of ω-semigroup to deterministic automata is due to Carton [3].

Once more, we start from a regular language L which is presented by an ω-semigroup (S_+, S_ω, π), a morphism α, and some subset $F \subseteq S_\omega$.

Lemma 21. *Given a $\mathcal{J}$-class J, the language of words such that the minimum recurrent $\mathcal{J}$-class J' is such that $J' \not>_{\mathcal{J}} J$ is deterministic Büchi.*

Proof. Without loss of generality, one assumes $1 \notin J$ (otherwise, this is the language A^ω). Let K be the set of elements $\{a \in S^1_+ \ : \ a >_{\mathcal{J}} J\}$. By the assumption $1 \notin J$, we have $1 \in K$. One constructs the following deterministic Büchi automaton:

- The set of states is K.
- The initial state is 1.
- The transition from state a, reading letter x, ends in state

$$\Delta(a,x) = \begin{cases} a\alpha(x) \text{ if } a\alpha(x) \in K, & (a) \\ 1 \qquad\ \ \text{otherwise.} & (b) \end{cases}$$

- The automaton accepts if some transition of kind (b) is seen infinitely often.

This automaton decomposes an input ω-word into $w = u_1 u_2 \ldots$ in such a way that each u_i is minimal such that $\alpha(u_i) \not>_{\mathcal{J}} J$. It accepts if and only if this decomposition is infinite.

Assume the automaton accepts an ω-word w. Then there is a $\mathcal{J}$-class visited by infinitely many u_i's, which is a witness that the minimum recurrent $\mathcal{J}$-class is not above J. Conversely, assume the automaton does not accept an ω-word w. This means that after some time no more transitions of kind (b) are visited anymore. Thus all $\mathcal{J}$-classes appearing after this moment are above J. □

Lemma 22. *Given a $\mathcal{J}$-class J, the language of ω-words:*

$$L \cap \{w \in A^\omega \ : \ J \textit{ is the minimum recurrent } \mathcal{J}\textit{-class in } w\}$$

is the difference of two deterministic Büchi languages.

Proof. We construct a deterministic Büchi automaton such that:

A. It accepts all ω-words such that the minimum recurrent $\mathcal{J}$ class is $\not\geq_{\mathcal{J}} J$.
B. An ω-word such that the minimum recurrent $\mathcal{J}$-class is J is accepted if and only if it does not belong to L.

Then, it is easy to see that if we subtract this language to the one of Lemma 21, we obtain the expected language.

Let K be the set of elements $\{a \in S^1_+ \ : \ a \geq_{\mathcal{J}} J\}$. One constructs the following deterministic Büchi automaton:

- The set of states is $S^1_+ \times K$.
- The initial state is $(1, 1)$.
- The transition from state (a, b) while reading letter x goes to state:

$$\Delta((a,b),x) = \begin{cases} (a, b\alpha(x)) & \text{if } b\alpha(x) \in K \quad (a) \\ (ab\alpha(x), 1) & \text{otherwise.} \qquad (b) \end{cases}$$

  Transition (a) is called $(a1)$ if $a(b\alpha(x))^{\rightarrow} \in F$, otherwise, it is called $(a2)$.
- The automaton accepts if a transition of the kind (b) or $(a2)$ is visited infinitely often.

First of all, remark that, for the same reasons as in the proof of Lemma 21, the transitions of kind (b) are visited infinitely often if and only if the minimum recurrent $\mathcal{J}$-class is not above or equal to $\mathcal{J}$. This settles item A.

Consider now some ω-word such that the minimum recurrent $\mathcal{J}$-class is J. This means that after some steps, no more (b)-transitions will be encountered. This uniquely decomposes the ω-word into $w = uv$ in which u is the prefix of w which terminates when the last (b)-transition is visited (possibly $u = \varepsilon$ as a pathological case if no (b)-transition is ever encountered).

One easily sees that for all finite prefix of w of the form uv', the automaton reaches the state $(\alpha(u), \alpha(v'))$ after reading it. Since the minimum recurrent $\mathcal{J}$-class is J, $\alpha(v')$ will eventually enter $\mathcal{J}$. By Corollary 3, if $w \in L$, then all transitions from this moment are of kind (a1) and the word is rejected, while if $w \notin L$, all transitions from this moment are of kind (a2), and the word is accepted. This settles item B. □

Of course, Theorem 6 follows directly since L is the union of the languages of Lemma 22 for J ranging over the possible $\mathcal{J}$-classes.

Acknowledgments

I am grateful to Olivier Carton, Denis Kuperberg and Jean-Éric Pin for helping me in the preparation of this document.

References

1. Bojańczyk, M.: Factorization forests. In: Developments in Language Theory. Volume 5583 (2009) 1–17
2. Büchi, J.R.: On a decision method in restricted second order arithmetic. In: Proceedings of the International Congress on Logic, Methodology and Philosophy of Science, Stanford Univ. Press (1962) 1–11
3. Carton, O.: Mots infinis, ω-semigroupes et topologie. PhD thesis, University Paris VII (1993)
4. Colcombet, T.: Factorisation forests for infinite words and applications to countable scattered linear orderings. Theoretical Computer Science **411** (2010) 751–764
5. Grillet, P.A.: Semigroups. An introduction to the structure theory. Pure and Applied Mathematics, Marcel Dekker. 193. New York, NY: Marcel Dekker, Inc. ix, 398 p. (1995)
6. Hashiguchi, K.: Limitedness theorem on finite automata with distance functions. J. Comput. Syst. Sci. **24**(2) (1982) 233–244
7. Kufleitner, M.: The height of factorization forests. In: MFCS. Volume 5162 (2008) 443–454
8. Leung, H.: Limitedness theorem on finite automata with distance functions: An algebraic proof. Theoretical Computer Science **81**(1) (1991) 137–145
9. McNaughton, R.: Testing and generating infinite sequences by a finite automaton. Information and Control **9** (1966) 521–530
10. McNaughton, R., Papert, S.: Counter-free Automata. MIT Press (1971)
11. Perrin, D.: Finite automata. In: Handbook of theoretical computer science, Vol. B. Elsevier, Amsterdam (1990) 1–57
12. Perrin, D., Pin, J.E.: Semigroups and automata on infinite words. In Fountain, J., ed.: NATO Advanced Study Institute Semigroups, Formal Languages and Groups. Kluwer academic publishers (1995) 49–72
13. Pin, J.E.: Varieties of Formal Languages. North Oxford Academic, London and Plenum, New York (1986)
14. Pin, J.E.: Syntactic semigroups. In Rozenberg, G., Salomaa, A., eds.: Handbook of formal languages. Volume 1. Springer Verlag (1997) 679–746
15. Pin, J.E., Weil, P.: Polynomial closure and unambiguous product. Theory Comput. Syst. **30**(4) (1997) 383–422
16. Rabin, M.O., Scott, D.: Finite automata and their decision problems. IBM J. Res. and Develop. **3** (April 1959) 114–125
17. Schützenberger, M.P.: On finite monoids having only trivial subgroups. Information and Control **8** (1965) 190–194
18. Semenov, A.L.: Decidability of monadic theories. In: MFCS. Lecture Notes in Computer Science, Springer (1984) 162–175
19. Simon, I.: Factorization forests of finite height. Theoretical Computer Science **72** (1990) 65–94
20. Simon, I.: On semigroups of matrices over the tropical semiring. RAIRO ITA **28**(3-4) (1994) 277–294
21. Thomas, W.: Languages, automata and logic. In Rozenberg, G., Salomaa, A., eds.: Handbook of language theory. Volume 3. Springer Verlag (1997) 389–455
22. Wilke, T.: An Eilenberg theorem for ∞-languages. In: Automata, Languages and Programming. Volume 510 of Lecture Notes in Computer Science, Springer (1991) 588–599