The factorisation Forest Theorem

Thomas Colcombet[1]

[1]LIAFA
Université Paris Diderot-Paris 7 and
Case 7014
F-75205 Paris Cedex 13
email: thomas.colcombes@liafa.jussieu.fr

August 23, 2012 22 h 21

2010 Mathematics Subject Classification: 68Q45

Key words: Monoids, semigroups, Ramsey, languages.

Abstract. This chapter is devoted to the presentation of the factorisation forest theorem, a deep result due to Simon, which provides advanced Ramsey-like arguments in the context of algebra, automata, and logic. We present several proofs and several variants the result, as well as applications.

1 Introduction

In automata theory, it is a very common and elementary argument to remark that, beyond a certain size, every run of a finite state automaton contains some repetition of a state. Once this repetition is witnessed, using copy and paste of the piece of run between those two occurrences, one can produce arbitrarily long valid versions of this run. This is the content of the "pumping lemma", which is nothing but a direct consequence of the pigeonhole principle.

The argument can also be used in the reverse way; whenever a run is too long, it contains a repetition of states, and it is possible to delete the piece of run separating those two occurrences. In this case, it is used for reducing the size of the input. These arguments are used in many situations. The first one is typically used for proving the impossibility for a finite state automaton to perform some task, such as recognising a given language. The second is used for proving the existence of small witnesses to the non-emptiness of a regular language, *i.e.*, a small model property in the terminology of logic. Those arguments are among the most important and useful ones in automata theory. They illustrate in the most basic way the central importance of "finding repetitions" in this context. All the content of this chapter is about "finding repetitions" in a more advanced way.

In some situations, the above argument of “state repetition” is not sufficient, and one is interested in finding the repetition of a “behaviour” of the automaton. A behaviour here is any piece of information of bounded size associated to a word. A typical behaviour of a (non-deterministic finite state) automaton over a word u is the set of pairs (p, q) such that the automaton has a run from p to q while reading the word u. This set of pairs gathers all the relevant information concerning how the automaton can behave while reading the word, whatever is the context in which the word is plugged. Given an input word, one can associate a behaviour to each factor of the word, and the theorem of Ramsey tells us that every sufficiently big word contains long repetitions of identical behaviours: for each n, every sufficiently long word u can be decomposed into

$$v\, u_1\, u_2\, \ldots\, u_n\, w\ ,$$

in which all the words $u_i u_{i+1} \cdots u_j$ for $i \leqslant j$ exhibit the same behaviour. Let us emphasise the difference with the pumping argument given above. Indeed, a run is a labelling of the positions in a word, while behaviours label factors of the word: the number of labels in a run is linear, while the number of behaviours is quadratic. However, the theorem of Ramsey is of similar nature as the pumping lemma as it relies on a pigeonhole principle.

A famous use of this Ramsey argument in automata theory is the proof of closure under complement of Büchi automata [9]. A Büchi automaton is a non-deterministic finite state automaton running over infinite words. It accepts a word if there is a run visiting infinitely many times a certain set of states (the so-called Büchi condition). Since the input words are infinite, there are uncountably many potential runs of this automaton over each input, some of them being accepting, some other not. The problem of an automaton for the complement is to provide a proof (which takes the form of a run) that none of these many runs of the original automaton is accepting. Of course, it is not possible to keep separately track of each run of the original automaton, since there are too many of them. It is also not possible–as one would do for finite word automata–to keep track only of the reachable states at each step, as one would loose the information relevant for the Büchi condition. The key idea of Büchi is to guess a decomposition of the input word into an infinite repetition of the same behaviour. This is possible thanks to the theorem of Ramsey as we described above in the finite case. This idea granted, the construction becomes easy: the automaton guesses this repetition, checks that it is consistent with the input word (this involves easy local verifications over finite words), and since the infinitely repeated behaviour contains sufficient information for asserting whether the word is accepted or not, the automaton can deduce from it when no run of the original automaton is accepting. Here, the theorem of Ramsey is used for making explicit the regularity in the behaviours of the original automaton. The reader is welcome to proceed to Chapter **??** for a thorough presentation of this technique.

The factorisation forest theorem, which is the subject of this chapter, goes even one step further. It does not only establish the existence of the repetition of some behaviour (as the theorem of Ramsey does), but it completely factorises each word into a structure (a factorisation tree) which exhibits repetitions of behaviours everywhere.

The theorem can be understood as a nested variant of the theorem of Ramsey. Consider some input word. A single use of the theorem of Ramsey splits the word into several sub-words corresponding to the same behaviour (plus two words for the extremities). However, each of those sub-words can itself also be very long, and one could again use the theorem of Ramsey on each of them, thus providing a sub-decomposition. In fact, one would like to iterate this process until the word is entirely decomposed, in the sense that the remaining words are just isolated letters that we are not interested in factorising. The result of this process can be represented as a tree, the root of which is the initial word, which has as children the words obtained by the first application of Ramsey, etc... This tree is a *Ramsey factorisation tree*.

In general, there is not much one can say about such an iteration. For instance, there is a priori no upper bound on the number of iterations required to completely decompose the word. What Simon's factorisation forest theorem teaches us is that, under the correct assumptions, this induction need only be iterated a bounded number of times. Said differently, there is a bound such that every input word admits a Ramsey factorisation tree of height at most this bound.

The required assumption is that the behaviours are equipped with a finite semigroup structure, and that the labelling of the input word by behaviours is consistent with this structure. This means that the behaviours S are equipped with an associative product $\cdot$, such that if u has behaviour a and v has behaviour b, then uv has behaviour $a \cdot b$. Formally, it amounts to require that the mapping from words to behaviours is a morphism of semigroups. The factorisation forest theorem can then be presented as follows:

> *factorisation forest theorem (Simon [30]):* For all finite semigroups S and all morphisms α from A^+ to S, there exists a bound k such that every word in A^+ has a Ramsey factorisation tree of height at most k.

Though very close in spirit, the factorisation forest theorem and the theorem of Ramsey are incomparable. Simon's theorem is weaker since it requires an extra hypothesis, namely that the behaviours be equipped with a semigroup structure, and this is a very strong assumption. But under this assumption, the factorisation forest theorem gives a much more precise result than the theorem of Ramsey. Technically, the two results are also proved using very different arguments. The theorem of Ramsey is proved by successive extraction processes, *i.e.* an extended pigeon-hole principle, while the proof of the factorisation forest theorem is based on algebraic arguments involving the theory of semigroup ideals (the relations of Green).

To conclude, the factorisation forest theorem is to be used when arguments based on the pigeonhole principle and the theorem of Ramsey are not sufficient anymore. The price to pay is to provide a semigroup structure for describing the problem. This is often the case when problems arise from automata or logic.

The factorisation forest theorem was introduced by Simon [30] as a generalisation of the lemma of Brown [7, 8] about locally finite semigroups. Simon gave several proofs of this theorem [30, 32]. Other proofs improving on the bounds have later be proposed [12, 15, 20]. Section 3 is devoted to several presentations of the theorem (Theorems 3.1 and 3.4), to its proof, and to some optimality considerations. We also presents some extensions of the result.

Concerning applications, the factorisation forest theorem allows us to give a very simple proof of the lemma of Brown. It has also been used by Simon to prove the decidability of the finite closure property in the tropical semiring. The tropical semiring is the set of non-negative integers $\mathbb{N}$ augmented with infinity and equipped with addition as product, and minimum as sum. The finite closure problem is, given a finite set of square matrices of the same size over the tropical semiring, to determine if their closure under product is finite. Simon proved the decidability of this problem in [28]. The finite section problem is more general, and consists in determining precisely what entries in the matrices can take arbitrary high values. This problem is equivalent to an automaton related problem: the limitedness of distance automata. Distance automata are non-deterministic finite state automata with weak counting capabilities. These automata compute functions, and the problem of limitedness of distance automata consists in determining whether the function computed by a given distance automaton is bounded. Hashiguchi established the decidability of this problem in [19]. Several proofs are known of this result. Leung proposed a very natural algorithm for solving this problem, the proof of correctness of which is rather complex [22, 23]. Simon gave a simplified proof of Leung's algorithm using the forest factorisations theorem [31]. Another application of the forest factorisations theorem is in the characterisation of certain classes of languages. For instance, it has been used by Pin and Weil for giving an effective characterising of the polynomial languages [26]. Polynomials are languages describable as a finite sum of languages of the form $A_0^*a_1A_2^* \dots a_nA_n^*$ in which $a_1, \dots, a_n$ are letters, and $A_1, \dots, A_n$ are sets of letters. It is possible to characterise the syntactic ordered monoids of languages are described by polynomials. The technique has been used for an extended result in [6]. Inspired by these works, a similar technique has also been used for characterising another pseudovariety of regular languages [1]. Section 4 is devoted to presenting and proving those applications. In fact, these applications can be simplified if we do not refer to factorization trees. That is why we provide two variant presentations of the forest factorization theorem, often easier to use (Theorems 4.1 and 4.2).

The factorisation forest theorem, in its original form, can only be used for words. There exists a variant of this theorem which allows us–in some specific situations–to apply it to trees [13]. We present this result in Section 5 (Theorem 5.2).

In Section 6, we present another use of the factorisation forest theorem, as an accelerating structure. This kind of application is very naturally performed on trees, using the tree-related variant of the theorem from Section 5. The principle consists in precomputing a factorisation tree over an input, such that one is able to answer specific queries very efficiently. The first result of this form was to show that every monadic second-order formula using free first-order variables can be effectively transformed into an equivalent (equivalent on trees equipped with the ancestor relation) first-order (in fact Σ_2) formula using some extra monadic second-order definable unary predicates. This technique has also been used in database theory to give constant delay enumeration problem for trees. It consists, given a query, to pre-process the database (a tree), and then to enumerate all solutions to the query, each of them in time linear in its size (linear time in the solution if the solutions consists of sets).

2 Some definitions

2.1 Semigroups and monoids

An *alphabet* A is a finite set of *letters*. A *word* over a finite alphabet A is a sequence of letters $u = a_1 \dots a_n$. If n is null, it is called the *empty word* and is written ε. The set of words over A is A^* and the set of non-empty words is A^+.

A *semigroup* $\mathbf{S} = \langle S, \cdot \rangle$ is a set equipped S with an associative operation $\cdot$. A *monoid* $\mathbf{M}$ is a semigroup $\langle M, \cdot \rangle$ that contains a *neutral element* $1_{\mathbf{M}}$, *i.e.*, an element such that $1_{\mathbf{M}} \cdot x = x \cdot 1_{\mathbf{M}} = x$ for all $x \in M$. Note that we do not enforce a semigroup to be non-empty (while a monoid is). A *group* is a monoid such that for every element x, there exists x^{-1} such that $x \cdot x^{-1} = x^{-1} \cdot x = 1$. An *idempotent* in a semigroup is an element e such that $e^2 = e$.

A *semigroup morphism* from a semigroup $\mathbf{S} = \langle S, \cdot \rangle$ to a semigroup $\mathbf{T} = \langle T, \cdot \rangle$ is a function f from S to T sucht hat $f(x \cdot y) = f(x) \cdot f(y)$ for all x, y in S. A *monoid morphism* is a semigroup morphism from a monoid to another monoid which is further required to map the neutral element of the first monoid to the neutral element of the second monoid.

Given a semigroup $\mathbf{S} = \langle S, \cdot \rangle$, one denotes by $\pi_{\mathbf{S}}$ the unique semigroup morphism from S^+ to $\mathbf{S}$ which coincides with the identity on letters, *i.e.*, $\pi_{\mathbf{S}}(a) = a$ and $\pi_{\mathbf{S}}(ua) = \pi_{\mathbf{S}}(u) \cdot a$, where $a \in S$. For simplicity, we often omit the $\mathbf{S}$ subscript and simply write π.

A semigroup $\mathbf{T} = \langle T, \cdot' \rangle$ is a *sub-semigroup* of a semigroup $\mathbf{S} = \langle S, \cdot \rangle$ if $T \subseteq S$ and $\cdot'$ coincides with $\cdot$ on T. One usually use the same notation $\cdot$ for $\cdot'$. For monoids, one also requires that the neutral elements of the monoid and its *submonoid* coincide. Given a set $A \subseteq S$, $\langle A \rangle_{\mathbf{S}}$ is the least subsemigroup of $\mathbf{S}$ which contains A. It is equal to $\langle \pi(A^+), \cdot \rangle$. One uses the same notation for monoids.

For a thorough introduction to semigroups, we refer the reader to [21, 25].

2.2 Linear orderings and multiplicative labellings

A *linear ordering* is a set equipped with a total order. Appart from Section 3.5, we will only consider finite linear orderings. Typically, given a word $u = a_1 \dots a_n$, we consider its *domain* $dom(u) = \{1, \dots, n\}$ (we can see a word as a function from its domain to its alphabet) and its set of *cuts* $cuts(u) = \{0, \dots, n\}$. A cut is a position between letters. The cut i for $i = 1, \dots, n-1$ represents the position between letters i and $i+1$. The cut 0 represents the beginning of the word, and the cut n the end of the word. Cuts among $1, \dots, n-1$ are called *inner cuts*. The set of inner cuts is $inner\text{-}cuts(u)$. Given two cuts $i < j$, the factor between positions i and j is $u_{i,j} = a_{i+1} a_{i+2} \cdots a_j$.

Let α be a linear ordering and $\langle S, \cdot \rangle$ a semigroup. A *multiplicative labelling*[1] is a mapping σ from the set of ordered pairs $(x, y) \in \alpha^2$ such that $x < y$ to S such that:

$$\text{for all } x < y < z \text{ in } \alpha, \quad \sigma(x, y) \cdot \sigma(y, z) = \sigma(x, z)\,.$$

Given a semigroup morphism φ from A^+ to some semigroup $\langle S, \cdot \rangle$ and a word u in A^+, there is a natural way to construct a multiplicative labelling φ_u from $cuts(u)$ to $\langle S, \cdot \rangle$ as

[1] It is called an additive labelling in the context of the composition method [27].

follows. For every two cuts $x < y$ in $cuts(u)$, set

$$\varphi_u(x,y) \overset{\text{def}}{=} \varphi(u_{x,y})\,.$$

This mapping is naturally a multiplicative labelling since for all $x < y < z$ in $cuts(u)$, $\varphi_u(x,y) \cdot \varphi_u(y,z) = \varphi(u_{x,y}) \cdot \varphi(u_{y,z}) = \varphi(u_{x,y}u_{y,z}) = \varphi(u_{x,z}) = \varphi_u(x,z)$.

This view of susing linear orderings and multiplicative labellings rather than words and morphisms is non-standard. It has several advantages in the present context. A first technical advantage is that some operations are easier to describe, for instance restricting a multiplicative labelling to a sub-ordering is straightforward (this is used several times in the main proof in Section 3.2). Another advantage is that its extension to infinite linear orderings is more natural than the use of infinite words (see Section 3.5).

2.3 Standard results on finite semigroups

In this section, we recall some basic definitions and gather results concerning finite semigroups. The reader can refer to [21, 25] for more details on the subject.

Given a semigroup S, S^1 denotes the monoid S itself if S is a monoid, or the semigroup S augmented with a new neutral element 1 otherwise, thus making S a monoid.

The important notions to prove the factorisation forest theorem are Green's relations. Those relations give a comprehensive understanding of the structure of a (finite) semigroup. However, in this survey, we need Green's relations only for proving the result of forest factorisation (and its deterministic variant). Green's relations are not used in the various applications of those theorems. In fact, one way to see the result of factorisation forests is as a convenient and easy to use result which gives access to non-trivial consequences of the theory of Green's relations.

The Green's relations are defined by:

$$\begin{array}{lll@{\qquad}lll}
a \leqslant_{\mathcal{L}} b & \text{if} & a = cb \text{ for some } c \text{ in } S^1 & a\,\mathcal{L}\,b & \text{if} & a \leqslant_{\mathcal{L}} b \text{ and } b \leqslant_{\mathcal{L}} a \\
a \leqslant_{\mathcal{R}} b & \text{if} & a = bc \text{ for some } c \text{ in } S^1 & a\,\mathcal{R}\,b & \text{if} & a \leqslant_{\mathcal{R}} b \text{ and } b \leqslant_{\mathcal{R}} a \\
a \leqslant_{\mathcal{J}} b & \text{if} & a = cbc' \text{ for some } c, c' \text{ in } S^1 & a\,\mathcal{J}\,b & \text{if} & a \leqslant_{\mathcal{J}} b \text{ and } b \leqslant_{\mathcal{J}} a \\
a \leqslant_{\mathcal{H}} b & \text{if} & a \leqslant_{\mathcal{L}} b \text{ and } a \leqslant_{\mathcal{R}} b & a\,\mathcal{H}\,b & \text{if} & a\,\mathcal{L}\,b \text{ and } a\,\mathcal{R}\,b
\end{array}$$

Fact 2.1. Let a, b, c be in S. If $a\,\mathcal{L}\,b$ then $ac\,\mathcal{L}\,bc$. If $a\,\mathcal{R}\,b$ then $ca\,\mathcal{R}\,cb$. For every a, b in S, $a\,\mathcal{L}\,c\,\mathcal{R}\,b$ for some c iff $a\,\mathcal{R}\,c'\,\mathcal{L}\,b$ for some c'.

As a consequence of the last equivalence, one defines the last of Green's relations:

$$\begin{array}{lll}
a\,\mathcal{D}\,b & \text{iff} & a\,\mathcal{L}\,c\,\mathcal{R}\,b \text{ for some } c \text{ in } S\,, \\
 & \text{iff} & a\,\mathcal{R}\,c'\,\mathcal{L}\,b \text{ for some } c' \text{ in } S\,.
\end{array}$$

The key result being (here the hypothesis of finiteness of S is mandatory):

Fact 2.2. $\mathcal{D} = \mathcal{J}$.

For this reason, we refer from now on only to $\mathcal{D}$ and not $\mathcal{J}$. However, we will use the preorder $\leqslant_{\mathcal{J}}$ (which is an order of the $\mathcal{D}$-classes).

An element a in S is called *regular* if $asa = a$ for some s in S. A $\mathcal{D}$-class is *regular* if all its elements are regular.

Fact 2.3. A $\mathcal{D}$-class D is regular, iff it contains an idempotent, iff every $\mathcal{L}$-class in D contains an idempotent, iff every $\mathcal{R}$-class in D contains an idempotent, iff there exists a, b in D such that $ab \in D$.

Fact 2.4. For every a, b in D such that $ab \in D$, $a \mathrel{\mathcal{R}} ab$ and $b \mathrel{\mathcal{L}} ab$. Furthermore, there is an idempotent e in D such that $a \mathrel{\mathcal{L}} e$ and $b \mathrel{\mathcal{R}} e$.

Fact 2.5. All $\mathcal{H}$-classes in a $\mathcal{D}$-class have the same cardinality.

Fact 2.6. Let H be an $\mathcal{H}$-class in S. Either for all a, b in H, $ab \notin H$; or for all a, b in H, $ab \in H$, and furthermore $(H, .)$ is a group.

3 The factorisation forest theorem

In this section, we give various statements for the factorisation forest theorem. We start with a formulation via splits. We then give a presentation in terms of Ramsey trees, the original one of Simon. A last presentation, more algebraic, is also given in Section 4.

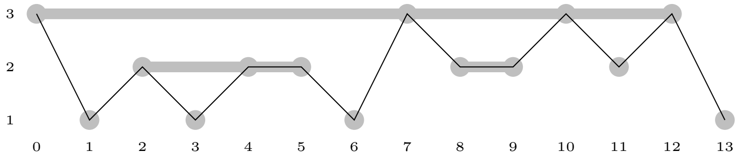
3.1 A statement via splits

A *split of height* h, h being a non-negative integer, over a linear order α is a mapping from the positions of α to $\{1, \ldots, h\}$. A split s induces an equivalence relation $\sim_s$ over α defined by:

$$x \sim_s y \qquad \text{if } s(x) = s(y) \text{ and } s(x) \geqslant s(z) \text{ for all } x \leqslant z \leqslant y\ .$$

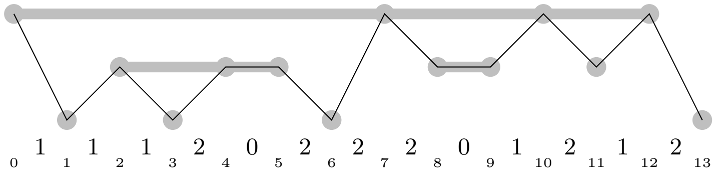
A split s of height h is called *normalised* if $s(\min \alpha) = h$.

In the following drawing, the points of the linear order $\langle\{0, \ldots, 13\}, <\rangle$ are drawn at different height according to the value of the split, yielding some sort of a landscape of mountains. The equivalence classes between points are depicted using gray lines. Graphically, two points are equivalent for the split if it is possible to go from one to the other by flying horizontally without crashing into a mountain.



A split over α is *Ramsey for a multiplicative labelling* σ over α if, for every equivalence classe C of $\sim$, there exists an idempotent e such that $\sigma(x, y) = e$ for all $x < y$ in C.

Consider for instance the (semi)group $\mathbb{Z}/3\mathbb{Z}$ and the alphabet $\{0, 1, 2\}$, with the obvious morphism. Consider now the word $u = 1112022201212$ over this alphabet which induces a multiplicative labelling σ_u. The split of the previous example happens to be Ramsey for this labelling:



One can verify that $\sigma_u(0,7) = \sigma_u(7,10) = \sigma_u(10,12) = 0$ (top most equivalence class), $\sigma_u(2,3) = \sigma_u(4,5) = 0$, and $\sigma_u(8,9) = 0$. All the other classes are singletons. Overall, the split is Ramsey for σ_u.

The factorisation forest theorem states that there is a bound such that every multiplicative labelling over a finite linear order admits a Ramsey split of height at most this bound. The statement below provides the bound $N(\mathbf{S})$ (defininition below). In practice, one often over-approximates this value simply by $|S|$, though this approximation may be far from optimal in some cases.

Theorem 3.1. *For every multiplicative labelling σ of a finite linear ordering by a finite semigroup $\mathbf{S}$, there exists a normalised split s which is Ramsey for σ, of height at most $N(\mathbf{S})$. In the above statement, the value $N(\mathbf{S})$ is the maximum over all chains of $\mathcal{D}$-classes*

$$D_1 <_{\mathcal{J}} \cdots <_{\mathcal{J}} D_k$$

of the sum $\sum_{i=1}^{k} N(D_i)$, where $N(S) = 1$ if D is irregular, and $N(D)$ is the number of elements in D which are $\mathcal{H}$-equivalent to an idempotent, otherwise.

The proof of this theorem is the subject of the next section.

3.2 Proof of the factorisation forest theorem

This proof follows the scheme used in [14, 15]. Other modern proofs of the factorisation forest theorem such as in [20], [3] or [16] do not use the split formalism, but involve essentially the same arguments.

Along this proof, σ denotes a multiplicative labelling from a finite non-empty linear ordering α to the finite semigroup $\langle S, \cdot\rangle$ of Theorem 3.1. We denote by $\sigma(\alpha)$ the set of elements of the semigroup occurring in σ, *i.e.*, $\{\sigma(x,y) \ : \ x, y \in \alpha, \ x < y\}$.

The proof consists of a case distinction according to Green's relations. In each case, a different argument is used for constructing the split. The first one is the case of a single $\mathcal{H}$-class which contains an idempotent, *i.e.*, the case of sub-groups.

Lemma 3.2. *Let $H \subseteq S$ be an $\mathcal{H}$-class such that $\langle H, \cdot\rangle$ is a group and let σ be a multiplicative labelling such that $\sigma(\alpha) \subseteq H$. Then there exists a normalised Ramsey split s of height at most $|H|$ of α, σ.*

Proof. Let $a_1, \ldots, a_{|H|}$ be an enumeration of the elements in H such that $a_{|H|} = 1_H$ (1_H denotes the neutral element of $\langle H, \cdot\rangle$). Let x_0 be $\min \alpha$. We define s by $s(x_0) = |H|$

and, for all $y \in \alpha$ with $y > x_0$, $s(y) = k$ where k is the unique number such that $a_k = \sigma(x_0, y)$.

$$s(y) = \begin{cases} |H| & \text{if } y = x_0 \\ k & \text{if } y > x_0 \text{ and } a_k = \sigma(x_0, y). \end{cases}$$

We prove that s is Ramsey for α, σ by showing that $\sigma(x, y) = 1_H$ for all $x < y$ such that $s(x) = s(y)$. We distinguish two cases. If $x = x_0$, then $s(y) = |H|$. This means by construction of s that $\sigma(x_0, y) = a_{|H|} = 1_H$. Otherwise, one knows by construction that $\sigma(x_0, x) = \sigma(x_0, y)$. Call this value a. We have

$$a = \sigma(x_0, y) = \sigma(x_0, x) \cdot \sigma(x, y) = a \cdot \sigma(x, y)\ .$$

Since $\langle H, \cdot\rangle$ is a group, dividing by a, one gets to $\sigma(x, y) = 1_H$. Consequently s is Ramsey. It is also clear that s is normalised by construction. □

The second case corresponds to a single regular $\mathcal{D}$-class.

Lemma 3.3. *Let D be a regular $\mathcal{D}$-class in S and σ be a multiplicative labeling over a linear ordering α such that $\sigma(\alpha) \subseteq D$. Then there exists a normalised Ramsey split of height at most $N(D)$ for α, σ.*

Proof. We first associate to each element $x \in \alpha$ an $\mathcal{L}$-class $L(x)$ and an $\mathcal{R}$-class $R(x)$ as follows. For all non-maximal elements $x \in \alpha$, we fix some $y > x$, and set $R(x)$ to be the $\mathcal{R}$-class of $\sigma(x, y)$. According to Fact 2.4 this definition does not depend on the choice of y. Similarly, for all non-minimal $x \in L$, we set $y < x$ and set $L(x)$ to be the $\mathcal{L}$-class of $\sigma(y, x)$. For x maximal, choose $R(x)$ to be any $\mathcal{R}$-class included in D such that $\langle R(x) \cap L(x), \cdot\rangle$ is a group: this is possible according to Fact 2.3. We similarly choose $L(x)$ for x minimal such that $\langle R(x) \cap L(x), \cdot\rangle$ is a group.

We claim that $\langle L(x) \cap R(x), \cdot\rangle$ is a group for all $x \in \alpha$, . This holds by construction when x is minimal or maximal. Consider now some non-minimal, non-maximal element $x \in \alpha$ and some $y < x$ and $z > x$. By construction, $\sigma(y, x) \in L(x)$, and $\sigma(x, z) \in R(x)$. Since furthermore $\sigma(y, x) \cdot \sigma(x, z) = \sigma(y, z) \in D$, using Fact 2.4, there exists an idempotent $e \in L(x) \cap R(x)$. This means by Fact 2.6 that $\langle L(x) \cap R(x), \cdot\rangle$ is a group. The claim holds.

Let now $H_1, \dots, H_k$ be an enumeration of $\mathcal{H}$-classes included in D that induce groups. Without loss of generality, we choose $H_k = L(\min \alpha) \cap R(\min \alpha)$. Let n be the size of $H_1 s$ (recall that all H-classes inside a D-class have same size according to Fact 2.5, and hence n is also the size of $H_2, \dots, H_k$). Note finally that $N(D) = kn$.

Set X_i to $\{x\ :\ L(x) \cap R(x) = H_i\}$ for all $i = 1 \dots k$. The X_i's are disjoint and, according to the above claim, their union equals α. For each $i = 1, \dots, k$ for which $X_i \neq \emptyset$, Lemma 3.2 provides a normalised split s_i over X_i of height n which is Ramsey for X_i, σ. Define now the split s for all $x \in \alpha$ by:

$$s(x) = s_i(x) + (i - 1)n\ , \qquad \text{in which } i \text{ is such that } x \in X_i\ .$$

Let us prove that s is Ramsey. Consider an equivalence class C for $\sim_s$. By construction of s, there is some i such that $C \subseteq X_i$. Hence, C is also an equivalence class for $\sim_{s_i}$

in X_i. Since s_i is Ramsey for X_i, σ by construction, this means that there exists an idempotent e such that $\sigma(x,y) = e$ for all $x < y$ in C. Hence s is Ramsey.

Furthermore, the height of s is $nk = N(D)$, and since $\min \alpha \in X_k$ by choice of H_k, and σ_k is normalised, we get $s(\min \alpha) = nk = N(D)$, i.e., s is normalised. □

We can now for complete the proof of Theorem 3.1.

Proof. For every $a \in S$, let $a{\uparrow}_{\mathcal{J}}$ be $\{b \ : \ a \leqslant_{\mathcal{J}} b\}$, and let $M(a)$ be the sum of $N(D)$ for D ranging over the $\mathcal{D}$-classes included in $a{\uparrow}_{\mathcal{J}}$.

The proof is by induction on the size of $a{\uparrow}_{\mathcal{J}}$. The induction hypothesis is that for every multiplicative labelling σ of a finite linear order α such that $\sigma(\alpha) \subseteq a{\uparrow}_{\mathcal{J}}$ we have:

- if a is regular, there exists a normalised Ramsey split of height at most $M(a)$ for α, σ,
- otherwise, there is a Ramsey split of height at most $M(a)$ for $(\alpha \setminus \{\min \alpha\}), \sigma$.

Let $a \in S$ and α be an order such that $\sigma(\alpha) \subseteq a{\uparrow}_{\mathcal{J}}$. We define x_i by induction on $i = 0 \dots$ by:

$$x_0 = \min \alpha, \qquad \text{and for all } i \geqslant 1, \quad x_i = \min\{x > x_{i-1} \ : \ \sigma(x_{i-1}, x) \; \mathcal{D} \; a\} \ .$$

(if there is no such element x, the constructions stops). In the end, a sequence $x_0 < x_1 < \cdots < x_m$ of elements in α is produced. Let $X = \{x_0, \dots, x_m\}$. One also defines $Y_1, \dots, Y_m$ to be the intervals of positions occurring between the x_i's: formally, $Y_0, \dots, Y_m$ are defined such that the union of $X, Y_0, \dots, Y_m$ is α, and $x_0 < Y_0 < x_1 < Y_1 < x_2 < \cdots < x_m < Y_m$ (note that some of the sets $Y_1, \dots, Y_m$ may be empty).
Remark a: For all i, j such that $0 \leqslant i < j \leqslant m$, one has $\sigma(x_i, x_j) \; \mathcal{D} \; a$ by construction. Said differently, $\sigma(X) \subseteq \mathcal{D}(a)$.
Remark b: For all $i = 0 \dots m$, $\sigma(\{x_i\} \cup Y_i) \cap \mathcal{D}(a) = \emptyset$. This comes from the minimality argument in the choice of each x_i.
Case 1: Let us assume first that a is regular. The principle of the construction is to use Lemma 3.3 over X, and the induction hypothesis over each of the Y_i's, and combine those splits. Set N to $N(\mathcal{D}(a))$, and M to $M(a)$.

By Remark a, $\sigma(X) \subseteq \mathcal{D}(a)$. Thus one can apply Lemma 3.3, and get a normalised Ramsey split s' of height N for X, σ. Thanks to Remark b, one can use the induction hypothesis and get for all $i = 0 \dots m$ a Ramsey split s_i for Y_i, σ of height at most $M(a) - N$. We combine the splits s',s_1,… into a split s by:

$$s(x) = \begin{cases} s'(x) + M - N & \text{if } x' \in X \ , \\ s_i(x) & \text{for } x \in Y_i \text{ otherwise.} \end{cases}$$

It is clear that, since s' is normalised, the same holds for s. Let us show that this split is Ramsey. Consider an equivalence class C for $\sim_s$. We distinguish two cases.

If $s(x) > M - N$ for some $x \in C$, this means that the first case in the definition of $s(x)$ is used for all elements $x \in C$. Hence, $C \subseteq X$, and C is an equivalence class for s'. Since s' is Ramsey, there exists an idempotent e such that $\sigma(x,y) = e$ for all $x < y$ in C.

Otherwise, $s(x) \leqslant M - N$ for one/all $x \in C$. Since $s(x) > M - N$ for all $x \in X$, it is not possible that C contains two elements which are separated by an element from X. We deduce that $C \subseteq Y_i$ for some i. Furthermore, since s and s' coincide over Y_i, C is also an equivalence class of $\sim_{s'}$. As s_i is Ramsey, this means that there exists an idempotent e such that $\sigma(x, y) = e$ for all $x < y$ in C.

Overall, s' is Ramsey for σ, α. This completes the proof of the first case of the induction hypothesis.

Case 2: It remains the case where a is irregular. We claim first that $|X| \leqslant 2$. Indeed assume that there exists $x < y < z$ in X. Then we have $\sigma(x, y)$, $\sigma(y, z)$ and $\sigma(x, z) = \sigma(x, y) \cdot \sigma(y, z)$ all belong to $\mathcal{D}(a)$. By Fact 2.3, this means that $\mathcal{D}(a)$ is a regular $\mathcal{D}$-class, contradicting the irregularity of a. This establishes the claim.

We can now define the split s for all $x \in \alpha \setminus \{\min \alpha\}$ of height M by:

$$s(x) = \begin{cases} M & \text{if } x' \in X \setminus \{\min \alpha\}\,, \\ s_i(x) & \text{for } x \in Y_i \text{ otherwise.} \end{cases}$$

Let us show that this split is Ramsey. Consider an equivalence class C for $\sim_s$. Again we distinguish two cases. If $s(x) \geqslant M$ for some $x \in C$, this means that $C \subseteq X \setminus \{\min \alpha\}$. Since $\min \alpha \in X$ and $|X| \leqslant 2$, $|C| = 1$. Hence this class is homogeneous. Otherwise $s(x) \leqslant M - 1$ for some $x \in C$. The same argument as in the first case of the induction hypothesis can be used. Overall, s is Ramsey for $\sigma, \alpha \setminus \{\min \alpha\}$.

Thus the induction hypothesis holds for all elements a. We can use it to establish Theorem 3.1. Let a be some element in the minimal $\mathcal{J}$-class of $\mathbf{S}$. This means $S = a{\uparrow}_{\mathcal{J}}$. Let α be a finite linear ordering, and σ a multiplicative labelling of α by S. Since $S = a{\uparrow}_{\mathcal{J}}$, and a is regular, one can apply the first case of the induction hypothesis on α, σ: there exists a normalised Ramsey split for α, σ. □

Let us now turn to the original statement as proposed by Simon.

3.3 The original statement using factorisation trees

Theorem 3.1 is stated in terms of splits as in [15]. The original statement of Simon [30] uses a different presentation that we describe in this section.

Fix an alphabet A and a semigroup morphism φ from A^+ to a finite semigroup $\langle S, \cdot \rangle$. A *factorisation tree* is an unranked ordered tree in which each node is either a leaf labeled by a letter, or an internal node. The *value* of a node is the word obtained by reading the leaves below from left to right. A *factorisation tree* of a word $u \in A^+$ is a factorisation tree with value u. The *height* of the tree is defined as usual, with the convention that the height of a single leaf is 0. A factorisation tree is *Ramsey* (for φ) if every node either

(1) is a leaf, or
(2) has two children, or
(3) the values of its children are all mapped by φ to the same idempotent of S.

Figure 1 presents a Ramsey factorisation tree for the word 1112022201212 over the alphabet $\{0, 1, 2\}$, with respect to the natural morphism to $\mathbb{Z}/3\mathbb{Z}$. Each non-leaf node of the tree is depicted as an horizontal line. The only node which satisfies property 3 is highlighted in a grey surrounding. One can check that indeed, the image by the morphism of the value of each child of this node is 0.

The factorisation forest theorem reads as follows, in which $N(\mathbf{S})$ is the value introduced in Theorem 3.1:

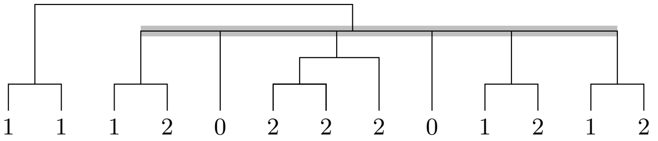


Figure 1. A Ramsey factorisation tree in $\mathbb{Z}/3\mathbb{Z}$

Theorem 3.4 (Factorisation Forest [30]). *For all alphabets A, all semigroup morphisms φ from A^+ to a finite semigroup $\mathbf{S}$, and all words $u \in A^+$, there exists a Ramsey factorisation tree for u, φ of height at most $k = 3N(\mathbf{S}) - 1$.*

The various references given for this result differ in the value of the bound k. In the original proof of Simon [30], the bound is $k = 9|S|$. Simon gave then a simplified proof [32] yielding a worse bound of $2^{|S|+1} - 2$ (this proof relies on the deep Krohn-Rhodes decomposition theorem). A bound of $k = 7|S|$ is achieved by Chalopin and Leung [12]. A bound of $3|S|$ is given in [14, 15]. The optimal bound is $3|S| - 1$ [20], see also [16][2]. Since $N(\mathbf{S}) \leqslant |S|$ the present result improves on the bound of $3|S| - 1$ to $3N(\mathbf{S}) - 1$. This better bound is essentially obtained by a more careful analysis of the construction.

Lemma 3.5 describes the relationship between Ramsey splits and Ramsey factorisations. Using it, Theorem 3.4 immediately follows from Theorem 3.1 (recall the definitions from Section 2.2).

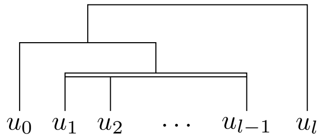
Lemma 3.5. *Let A be an alphabet, a morphism φ from A^+ to a finite semigroup $\mathbf{S}$ and a word $u \in A^+$.*

(a) Every Ramsey factorisation tree of height k of u induces a Ramsey split of height at most k for $\mathit{inner\text{-}cuts}(u), \varphi_u$.

(b) Every Ramsey split of height k for $\mathit{cuts}(u), \varphi_u$ indeuces a factorisation tree of height at most $3k$ of u; of height $3k - 1$ if the split is furthermore normalised.

Proof. For (a), we set the value of the split for $x \in \mathit{inner\text{-}cuts}(u)$, say for x the cut between letter i and letter $i + 1$ in u, to be the maximal depth of a node that has the ith and the $(i + 1)th$ letter below it. It is not difficult to see that this defines a split of height at most k, and that it is Ramseyan for $\mathit{inner\text{-}cuts}(u), \varphi_u$.

For (b), note that the only class of value 1 according to the split (we assume that there is one) factorises the word u into $u = u_0 u_1 \dots u_l$ in such a way that $\phi(u_1) = \dots = \phi(u_{l-1})$ is an idempotent. Hence we construct the prefix of a tree as:



and then proceed inductively with the subwords $u_0, \dots, u_l$. We get at the end a Ramseyan factorisation tree, and its height is at most $3k$. Furthermore note that, if the split

[2] Unlike stated in [16], the bound is due to Kufleitner.

is normalised, there is no need to use the root node of the above gadget for the highest $\sim$-class. □

3.4 Optimality of the bound

We have seen a bound of $N(\mathbf{S})$ in Theorem 3.1, and a bound of $3N(\mathbf{S})-1$ for Theorem 3.4. The question we are addressing in this section is whether this bound is optimal. This question has been the source of some investigations [12, 20]. Indeed, in some applications, this parameter may have a significant complexity impact (see the applications in Sections 4 and 6). It is also natural that a better understanding of this parameter requires a better understanding of the structure of semigroups. This remark itself justifies the interest in this question.

Chalopin and Leung [12] and Kufleitner [20] derived lower bounds. The following result of Kufleitner shows that the bound of $3|S|-1$ of Theorem 3.4 is optimal for groups (in the case of groups, $N(S)=|S|$).

Theorem 3.6 ([20]). *For all non-trivial finite groups $\mathbf{G}$ there exists a word $w\in G^+$ such that every factorisation tree of w,φ has height at least $3|G|-1$, where $\varphi : G^+\to G$ is the evaluation morphism.*

One can also deduce from it the optimality of Theorem 3.1.

Corollary 3.7. *For all non-trivial finite groups $\mathbf{G}$ there exists a multiplicative labelling σ from a finite linear ordering to $\mathbf{G}$ such that every Ramsey split of σ has height at least $|G|$.*

Proof. Consider the word w from Theorem 3.6 and the corresponding multiplicative labelling $\sigma=\varphi_w$. For the sake of a contradiction, assume that there is a Ramsey split of height $|S|-1$ for σ. By Lemma 3.5 this means that there exist a Ramsey factorisation tree of height at most $3(|S|-1)$ for w,φ, contradicting Theorem 3.6. □

In this chapter, we have given an optimised result, with a bound of $N(\mathbf{S})$ in terms of splits (Theorem 3.1), and $3N(\mathbf{S})-1$ in terms of factorisation forest (Theorem 3.4). In some cases $N(\mathbf{S})=|S|$, this is the case for instance when $\mathbf{S}$ is a group, but not only (consider for instance the semigroup $\langle\{1,\dots,n\},\max\rangle$). However, it can also happen that the gap between $N(\mathbf{S})$ and $|S|$ can be arbitrarily high: consider for instance for each positive integer n the semigroup $\mathbf{S}_n$ with elements $\{a_1,\dots,a_n,0\}$ for which the product is defined by $0\cdot x=x\cdot 0$ for all x, and $a_i\cdot a_j$ is a_i if $j=i$, and 0 otherwise. This semigroup has size $n+1$, but $N(\mathbf{S}_n)=2$ for all n. This shows that a careful analysis can drastically improve on the original upper bound of $|S|$.

However, one can still wonder whether the bound $N(\mathbf{S})$ is optimal. More precisely, given a semigroup $\mathbf{S}$, does there exist always a multiplicative labelling such that no split Ramsey for it has height less than $N(\mathbf{S})$?

The answer to this question is negative. Consider for instance the semigroup $\mathbf{S}_n=\langle\{1,2,\dots,n-1,\infty\},+\rangle$ (in which the sum is defined in the natural way). Then $N(\mathbf{S})=|\mathbf{S}|=n$. However, for every multiplicative labelling from a (finite) linear ordering to $\mathbf{S}_n$

there exists a Ramsey split of height at most $\lceil \log_2 n\rceil + 2$. We give a proof using factorisation trees (this extend to splits using Lemma 3.5). Note that (a) that in this semigroup, every word of length greater than n has value ∞. Note that (b) that in any semigroup, every word of size at most k admits a factorisation tree of height at most $\lceil \log_2 k\rceil$ (using a balanced binary factorisation tree of logarithmic height). Combining these remarks, we can construct a factorisation for every word u as follows. One factorises u into $u_1 \dots u_l v$ in which $|u_1| = \dots = |u_l| = n$, and $|v| < n$. By remark (b), all words $u_1, \dots, u_l, v$ admit Ramsey factorisation trees $t_1, \dots, t_l, t'$ of height at most $\lceil \log_2 n\rceil$. Futhermore, by Remark (b) all words $u_1, \dots, u_l$ have same value ∞ (which is an idempotent). This means that one can construct a Ramsey factorisation tree of height $\lceil \log_2 n\rceil + 2$ for it: the root is binary, the right child being the root of t', and the left child being an idempotent node with n chidren, which are the roots of respectively $t_1, \dots, t_l$. It is clear that this tree is a Ramsey factorisation for u, and also that it has height at most $\lceil \log_2 n\rceil + 2$.

Thus, the question of characterising the optimal bound for the factorisation forest theorem is still open.

Kufleitner gives a finer analysis of the bound for aperiodic semigroups using factorisation trees. Indeed, the result is optimal for groups. What about group-trivial semigroups? The answer is that it is possible to obtain a better upper bound in this case:

Theorem 3.8 ([18],[20]). *For every aperiodic (i.e., group-trivial) semigroup $\mathbf{S}$, and every morphism from A^+ to $\mathbf{S}$, every word $u \in A^+$ admits a Ramsey factorisation tree of height at most $2|S|$. Furthermore, for each n, there exists an aperiodic semigroup of size n such that this bound is optimal.*

3.5 Infinitary variants

So far, we have seen the factorisation forest theorem for finite linear orderings/finite words. In fact, the finiteness assumption is not so relevant for the result. For the presentation of presenting of infinitary variants, the machinery of splits is easier to use than factorisation trees. We only consider splits in this section.

From what we have seen so far, we can already deduce a first infinitary variant of the result. Consider the linear ordering $\langle \mathbb{N}, <\rangle$, and a multiplicative labelling σ from it to some finite semigroup $\mathbf{S}$. By Theorem 3.1, for every n, there exists a Ramsey split s_n of $\sigma, \{0, \dots, n\}$ of height $N(\mathbf{S})$. By compactness (of the Cantor space, see Chapter **??**) there exists a split of $\langle \mathbb{N}, <\rangle$ of height at most $N(\mathbf{S})$ such that for every i, s coincides with some s_n over $\{0, \dots, i\}$. It is not difficult to see that, since all the s_n splits are Ramsey, the same holds for s.

In fact, the result goes beyond $\langle \mathbb{N}, <\rangle$, but for that one needs a new proof.

Theorem 3.9 ([15]). *For all finite semigroups $\mathbf{S}$ and all multiplicative labelings σ of a (possibly infinite) linear ordering α to S, there exists a split of α that is Ramsey for σ, α, and has height at most $2|S|$. The split has height at most $|S|$ for ordinals.*

Here, we just state the bound in terms of $|S|$, though it is likely that in the case of ordinals, the bound of $N(\mathbf{S})$ still holds, and that a similar improved bound can be given in the general case. However, this would require to reprove the results of [15], and this goes beyond the subject of this survey.

But, what is the interest of having an infinitary variant of the factorisation forest theorem? An application is given in [15], namely for the complementation of automata over scattered countable linear orderings (a linear ordering is scattered if it does not contain a dense linear subordering). This result is known from [11]. Theorem 3.9 allows us to give a much simpler proof of this result. Although outside the scope of this survey, it is still possible to explain why the factorisation forest theorem (in its infinitary variant) helps.

Recall from the introduction and Chapter **??**, that a very classical use of the theorem of Ramsey is to prove the complementation of Büchi automata over words indexed by ω. The idea is to construct an automaton which guesses a good Ramsey decomposition of the word. This decomposition splits the word into finite sub-words over which one can use standard finite word automata. In the case of an infinite linear ordering, a use of the theorem of Ramsey can decompose the word into infinitely many words, which themselves are infinite. The next step would be to sub-factorise those subwords, etc... But there is no reason that these nested factorisations terminates. The factorisation forest theorem is perfectly suited for this kind of applications. It provides a bound of $2|S|$ such that this induction is guaranteed to terminate within this bound.

This technique has been pushed even further in [10] for proving that a language of words of countable length is recognised (using a suitable form of algebra) if and only if it is definable in monadic second-order logic.

In the next section, we will see several other applications of the factorisation forest theorem over finite words/finite linear orderings. The extension of some of these applications (e.g., the limitedness of distance automata) to the infinite context is possible. Theorem 3.9 is a good starting point if one is interested in pushing further in this direction.

4 Algebraic applications

The purpose of this section is to give algebraic consequences to the factorisation forest theorem. In those applications, we deliberately chose to use another presentation of the result, which is at the same time weaker (we lose the information concerning the bound), but much more easy to apply (no more trees).

4.1 An algebraic presentation

In this section, we give two other equivalent presentations of the factorization forest theorem. Depending on the context, the various presentation may prove easier to use. In particular, the two presentations avoid use of trees or splits.

The first presentation below is particularly interesting wen one is interested in effectively computing a presentation for the semigroup generated by a given subset of a monoid. We do not present in this survey any examples of this kind of applications.

Theorem 4.1. *Let* $\mathbf{S}$ *be a semigroup,* φ *be a semigroup morphism from* $\mathbf{S}$ *to a finite* *semigroup $\mathbf{T}$, and $X \subseteq \mathbf{S}$, then $\langle X\rangle = X_{3N(\mathbf{T})-1}$ where X_n is defined by*

$$X_0 = X \quad \textit{and} \quad X_{n+1} = X_n \cup X_n \cdot X_n \cup \bigcup_{e\cdot e=e} \langle X_n \cap \varphi^{-1}(e)\rangle_{\mathbf{S}} \quad \textit{for all } n \geqslant 0.$$

Proof. It is clear, by induction on n, that $X_n \subseteq \langle X\rangle$. Quite naturally, the proof of the converse inclusion is by induction on the height of factorization trees. For all $n \geqslant 0$, set

$$Y_n = \{\pi(u) \ : \ u \in X^+,\ u \text{ has a Ramsey factorization tree of height at most } n\}\ ,$$

where the Ramsey factorization is with respect to the morphism $\varphi \circ \pi$. Let us show by induction on n that $Y_n \subseteq X_n$. Assuming this, Theorem 3.4, implies that $\langle X\rangle \subseteq X_{3N(\mathbf{T})-1}$, and the results follows. The induction remains to be established. Clearly, for $n = 0$, $X_0 = X = Y_0$.

Consider now some $n \geqslant 0$, and let $a \in Y_{n+1}$. One aims at $a \in X_{n+1}$. By definition, there exists a Ramsey factorization T of height at most $n+1$ for some $u \in X^+$ with $\pi(u) = a$. There are three cases. If T has height at most n, then u is also a witness that $a \in Y_n \subseteq X_n$, and $X_n \subseteq X_{n+1}$ by definition of X_{n+1}. Thus $a \in X_{n+1}$. Otherwise, assume the root of T is a binary node. Then u can be decomposed as vw, such that $\pi(v) \in X_n$ and $\pi(w) \in X_n$. It follows, by induction hypothesis and definition of X_{n+1} that $a = \pi(u) = \pi(v)\cdot\pi(w) \in T_n \cdot T_n \subseteq X_n \cdot X_n \subseteq X_{n+1}$. Finally, assume the root of T is an idempotent node. This means that u can be decomposed as $v_1 \dots v_k$ such that there exists an idempotent e with $\varphi(\pi(v_i)) = e$ for all i, and $\pi(v_i) \in T_n$ for all i. Hence, by induction hypothesis, $\pi(v_i) \in X_n$ for all i and thus $\pi(v_i) \in X_n \cap \varphi^{-1}(e)$. Thus $a = \pi(u) = \pi(v_1)\cdots\pi(v_k) \in \langle X_n \cap \varphi^{-1}(e)\rangle \subseteq X_{n+1}$ by definition of X_{n+1}. □

In fact, a closer inspection reveals that the above theorem is equivalent to the forest factorisation theorem. Indeed, a similar inductive proof estabblishes that $T_n = X_n$ for all n (where T_n is as in the above proof). Thus, if one applies Theorem 4.1 to $\mathbf{S} = A^*$, one directly deduces Theorem 3.4.

Our second variant can be understood as follows. Theorem 4.1 can be seen as an iteration reaching a least fix-point. Theorem 4.2 formalizes differently this view of the result.

Theorem 4.2. *Let $\mathbf{S}$ be a semigroup, φ be a semigroup morphism from $\mathbf{S}$ to a finite semigroup $\mathbf{T}$, and $X \subseteq \mathbf{S}$. Then every family $P \subseteq \mathcal{P}(\mathbf{S})$ such that*

(1) for all $a \in \mathbf{T}$, $\{x \in X \ : \ \varphi(x) = a\} \in P$;
(2) for all $A, B \in P$, $A \cup B \in P$;
(3) for all $A, B \in P$, $A \cdot B \in P$; and,
(4) for all $A \in P$ with $f(A) = \{e\}$ for some idempotent $e \in \mathbf{T}$, $\langle A\rangle_{\mathbf{S}} \in P$,

satisfies $\langle X\rangle \in P$.

Remark 4.3. In practice, instead of (1), we will frequently use the following slightly stronger conditions:

(1') for all $A \subseteq B \in P$, $A \in P$;
(1") $X \in P$.

It is clear that (1') and (1") together imply (1).

Proof. Let us assume first that P is minimal such that it satisfies conditions (1), (2), (3) and (4).

One claims first that every $A \in P$ has the *restriction property*, *i.e.*, for all $c \in \mathbf{T}$, then $A \cap \varphi^{-1}(c) \in P \cup \{\emptyset\}$. It is sufficient forus to prove that having the restriction property is preserved under the rules (1) to (4). Indeed, if $A = \{x \in X \ : \ \varphi(x) = a\} \in P$, then clearly, $A \cap \varphi^{-1}(c)$ equals A if $c = a$, or $\emptyset$. This settles the case (1). Consider now the case $A \cup B$. It is clear that if both A and B have the restriction property, then $A \cup B$ also has the property since $(A \cup B) \cap \varphi^{-1}(c) = (A \cap \varphi^{-1}(c)) \cap (B \cap \varphi^{-1}(c)) \in P$ (by (2)). This proves the case (2). Consider now the case $A \cdot B$. We have

$$(A \cdot B) \cap \varphi^{-1}(c) = \bigcup_{a \cdot b = c} (A \cap \varphi^{-1}(a)) \cdot (B \cap \varphi^{-1}(b)) \ .$$

Thus, assuming that A, B have the restriction property, using (3), $A \cdot B$ also has the restriction property. This establishes the case of (3). Finally, assume $A \in P$ and $\varphi(A) = \{e\}$ for some idempotent e. Then clearly, if $c \neq e$, then $\langle A \rangle \cap \varphi^{-1}(c) = \emptyset$. Otherwise when $c = e$, this implies $\langle A \rangle \cap \varphi^{-1}(c) = \langle A \rangle \in P$. Hence $\langle A \rangle$ has the restriction property, which is the case (4). It follows that every $A \in P$ (using the minimality assumption) has the restriction property. The claim is established.

Let now the X_n's be as in Theorem 4.1 In this case, let us prove by induction on n that $X_n \in P$. For $n = 0$, from (1) and (2), $X_0 = X \in P$. Otherwise, assume $X_n \in P$, then clearly, using the properties (1) to (4) and the above claim, $X_{n+1} \in P$ (the claim is mandatory for proving that if $X_n \in P$, then $X_n \cap \varphi^{-1}(e) \in P$). It follows, using Theorem 4.1, that $\langle X \rangle = X_{3N(\mathbf{T})-1} \in P$.

Consider now some P' that satisfies conditions (1) to (4) (without any minimality assumption). This means $P \subseteq P'$ (where P' is minimal). One has $\langle X \rangle \in P \subseteq P'$. This establishes the general case. □

Once more, it is easy to show that this result is equivalent to the forest factorization theorem, as far as the precise bound of $3N(\mathbf{S}) - 1$ is not concerned.

4.2 Brown's lemma

In this section we show how to derive Brown's lemma from the above result. Extending Brown's lemma was one of the motivations of Simon when introducing the factorisation forest theorem.

A semigroup $\mathbf{S}$ is *locally finite* if every finite subset $X \subseteq \mathbf{S}$ generates a finite subsemigroup $\langle X \rangle_{\mathbf{S}}$. Brown's theorem is stated as follows:

Lemma 4.4 ([8]). *Let $f : \mathbf{S} \to \mathbf{T}$ be a semigroup morphism. If $\mathbf{T}$ is locally finite and for every idempotent $e \in \mathbf{T}$, $f^{-1}(e)$ is locally finite, then $\mathbf{S}$ is locally finite.*

Proof. Let $f, \mathbf{S}$ and $\mathbf{T}$ be as in the statement of the theorem. Let $X \subseteq S$ be finite. We want to show that $S' = \langle X \rangle_{\mathbf{S}}$ is finite. Let $T' = f(S')$. Since $f(X)$ is finite and $\mathbf{T}$ is locally finite, we get that $T' = f(S') = f(\langle X \rangle_{\mathbf{S}}) = \langle f(X) \rangle_{\mathbf{T}}$ is finite. Let P be the set of finite subsets of S'. Clearly, P satisfies conditions (1'), (1"), (2) and (3) of Theorem 4.2 and Remark 4.3. Let us establish the missing (4). Consider $A \in P$ such that $f(A) = \{e\}$.

This means $A \subseteq f^{-1}(e)$. Since by hypothesis $f^{-1}(e)$ is locally finite and A is finite, it follows that $\langle A\rangle_{\mathbf{S}}$ is finite, i.e., $\langle A\rangle_{\mathbf{S}} \in P$. Using Theorem 4.2 we obtain $S' \in P$, i.e., S' is finite. Since this holds for all X, $\mathbf{S}$ is locally finite. □

4.3 The finite closure property in the tropical semiring

In this section, we show how to use Brown's lemma for deciding the finite closure problem in the tropical semiring. In the next section, we will extend those techniques to solve a more general result, this time using the factorisation forest theorem. This theory is nicely surveyed in [29].

We consider here the *tropical semiring* $\mathbb{T} = (\mathbb{N} \cup \{\infty\}, \min, +)$, (also called the *Min-Plus-semiring*). We use standard notation for matrices over this semiring. Matrices over a semiring form themselves a semiring when equipped with the usual multiplication and sum. In this section, we consider the multiplicative group of this matrix semiring.

The *finite closure problem* is the following:

Input: A positive integer n and matrices $A_1, \dots, A_k \in \mathbb{T}^{n\times n}$.
Output: "Yes", if the set $\langle A_1, \dots, A_k\rangle_{\mathbb{T}^{n\times n}}$ is finite; "no" otherwise.

We prove below that this problem is decidable. On the way we show that the corresponding Burnside problem admits a positive answer. More precisely, one says that a semigroup $\mathbf{S}$ is *torsion* if for every element $x \in \mathbf{S}$, $\langle x\rangle_{\mathbf{S}}$ is finite. It is clear that every finite semigroup is both finitely generated and torsion. The *Burnside problem* consists in determining for which semigroups the converse holds. The proof of Simon shows that the Burnside problem admits a positive answer for semigroups of matrices over the tropical semiring, i.e., a subsemigroup of $\mathbb{T}^{n\times n}$ is finite iff it is both finitely generated and torsion. Phrased differently:

Theorem 4.5 ([28]). *Every torsion subsemigroup of $\mathbb{T}^{n\times n}$ is locally finite.*

The corresponding decidability result is established at the same time:

Theorem 4.6 ([28]). *The finite closure property is decidable inside $\mathbb{T}^{n\times n}$.*

The problem for the decidability proof is that the tropical semiring is infinite, which prevents exploring entirely. For this reason, the essential argument in the proof consists in translating the question to a question concerning a finite algebraic object. Formally, one constructs a morphism from the tropical semiring to a finite semiring which forgets the exact values of the matrix entries.

Let us consider the *reduced semiring* $\mathbb{T}_1 = (\{0, 1, \infty\}, \min, +)$ (in which all operations are natural, and $1 + 1$ equals 1). Given an element $a \in \mathbb{T}$, denote by $\overline{a}$ its reduced version defined by $\overline{0} = 0$, $\overline{\infty} = \infty$ and $\overline{a} = 1$ in all other cases. I.e., one approximates every positive integer by 1. The function $\overline{}$ is a morphism of semirings. This is the reason it extends in the usual way to matrices, yielding once more a morphism of semirings: the morphism which replaces every positive integer entry in a matrix by 1.

Call a matrix A in $\mathbb{T}^{n\times n}$ *idempotent* if its image under $\overline{}$ is an idempotent (of $\mathbb{T}_1^{n\times n}$). Conversely, given an element $a \in \{0, 1, \infty\}$, and a positive integer k, we denote by $k \times a$

the element a if $a \in \{0, \infty\}$, and k otherwise. We also extend this operation to matrices. Given a matrix A, denote by $||A||$ the maximal positive integer entry it contains (or 1 if there is no such entry).

An idempotent matrix A over $\mathbb{T}$ is called *stable* if the set $\langle A\rangle_{\mathbb{T}^{n\times n}}$ is finite.

Lemma 4.7. *For every idempotent matrix $A \in \mathbb{T}^{n\times n}$, the following statements are equivalent:*

(1) A is stable,
(2) for all i, j such that $A_{i,j} \neq \infty$, there exists k such that $A_{i,k} \neq \infty$, $A_{k,k} = 0$ and $A_{k,j} \neq \infty$,
(3) $||A^p|| \leqslant 2||A||$ for all $p \geqslant 1$.

Proof. (1) ⇒ (2) Assume that A is stable, and consider i, j such that $A_{i,j} \neq \infty$. Since A is idempotent, $A^p_{i,j} \neq \infty$ for all $p \geqslant 1$. Since furthermore A is stable, $A^p_{i,j}$ can take only finitely many values when p ranges. Let m be the highest such value, i.e., $A^p_{i,j} \leqslant m$ for all $p \geqslant 1$. In particular, for $p = (m + 1)|Q| + 2$, this is witnessed by the existence of $i_0, i_1, \ldots, i_p$ such that $i_0 = i$, $i_p = j$, and $A_{i_0,i_1} + A_{i_1,i_2} + \cdots + A_{i_{p-1},i_p} \leqslant m$. Since $p = (m+1)|Q|+2$, there exist $1 \leqslant l < s < p$ such that $A_{i_l,i_{l+1}} +\cdots+ A_{i_{s-1},i_s} = 0$, and $i_l = i_s$. Using the idempotency of A, we get for $k = i_l$ that $A_{i,k} \neq \infty$, $A_{k,k} = 0$, and $A_{i,k} \neq \infty$.

(2) ⇒ (3) Assume (2) holds. For $p = 1$, (3) is obvious. Consider some $p \geqslant 2$. Let $1 \leqslant i, j \leqslant n$. If $A_{i,j} = 0$, then by idempotency of A, $A^p_{i,j} = 0$. The same holds for $A_{i,j} = \infty$. Now if $A_{i,j} \in \mathbb{N}^+$, then, by hypothesis, there exists k such that $A_{i,k} \neq \infty$, $A_{k,k} = 0$ and $A_{k,j} \neq \infty$. This means that the term $A_{i,k} + A_{k,k} + \cdots + A_{k,k} + A_{k,j}$ is involved in the minimum defining the value of $A^p_{i,j}$. It follows that $A^p_{i,j} \leqslant 2||A||$. Overall, we obtain that $A^p_{i,j} \leqslant 2||A||$. Since this holds for all i, j, $A^p \leqslant (2||A||) \times \overline{A}$.

(3) ⇒ (1) Assume (3) holds. Each matrix $B \in \langle A\rangle$ is such that both $\overline{B} = \overline{A}$ (by idempotency) and $||B|| \leqslant 2||A||$ (by Item 3). There are only finitely many such matrices satisfying these properties. Hence $\langle A\rangle$ is finite, which means that A is stable. □

Corollary 4.8. *Let A, B in $\mathbb{T}^{n\times n}$ with $\overline{A} = \overline{B}$, A is stable iff B is stable. Furthermore, the stability of a matrix is decidable.*

Thanks to the above corollary, it is meaningful to say that a matrix A over $\mathbb{T}_1$ is *stable* if there exist one matrix $B \in \mathbb{T}^{n\times n}$ such that $\overline{B} = A$ is stable, or equivalently if this holds for every matric B such that $\overline{B} = A$.

The core of the proof is embedded in the following lemma.

Lemma 4.9. *Given matrices $A_1, \ldots, A_k \in \mathbb{T}^{n\times n}$, $\langle A_1, \ldots, A_k\rangle_{\mathbb{T}^{n\times n}}$ is finite iff every idempotent matrix in $\langle \overline{A_1}, \ldots, \overline{A_k}\rangle_{\mathbb{T}_1^{n\times n}}$ is stable.*

Proof. Set $C = \langle A_1, \ldots, A_k\rangle_{\mathbb{T}^{n\times n}}$. Then $\overline{C} = \langle \overline{A_1}, \ldots, \overline{A_k}\rangle_{\mathbb{T}_1^{n\times n}}$.

If there is an unstable matrix in $\overline{C}$, this means that there exists an unstable matrix in $A \in C$. By definition of stability, this means that $\langle A\rangle_{\mathbb{T}^{n\times n}}$ is infinite, and hence C is infinite.

Conversely, assume that every idempotent matrix in $\overline{C}$ is stable. We apply Brown's lemma to the morphism $\overline{}$ which sends C to $\overline{C}$. Since $\overline{C} \subseteq \mathbb{T}_1^{n\times n}$, it is finite, and as a consequence also locally finite. Consider now some idempotent matrix $A \in \mathbb{T}_1^{n\times n}$, let us show that $\{B \ : \ \overline{B} = A\}$ is locally finite. For this, consider some finite set $X \subseteq \mathbb{T}^{n\times n}$ such that $\overline{X} = \{A\}$. Let $m = \max_{B\in X}(||B||)$, and consider some $B_1, \dots, B_n \in X$, we have:

$$||B_1 \cdots B_n|| \leqslant ||(m \times A) \cdots (m \times A)|| \leqslant 2m\ .$$

(the last inequality is from (3) of Lemma 4.7). Since there are only finitely many B's such that $\overline{B} = A$ and $||B|| \leqslant 2m$, it follows that $\langle X\rangle_{\mathbb{T}^{n\times n}}$ is finite, and hence $\{B \ : \ \overline{B} = A\}$ is locally finite.

Hence by Brown's lemma, we directly get that C is locally finite. Since C is generated by finitely many matrices (namely $A_1, \dots A_k$), this means that C is finite. □

From the above lemma, one immediately obtains Theorem 4.5: consider a torsion subsemigroup S of $\mathbb{T}^{n\times n}$. Then every idempotent matrix in S is stable. Hence if X is a finite subset of S, then $\langle\overline{X}\rangle$ does only contain stable idempotents. By Lemma 4.9, this means that $\langle X\rangle$ is finite. We conclude that S is locally finite.

The lemma also yields a decision procedure: compute the closure of $\{\overline{A_1}, \dots, \overline{A_k}\}$, and check whether there is an unstable matrix in this set. We obtain Theorem 4.6.

Technically, in this application, we did not use directly the factorisation forest theorem, but rather Brown's lemma which is one of its consequences. In the next section, we study a generalisation of the above problem, and this time, Brown's lemma is not sufficient anymore.

4.4 The bounded section in the tropical semiring

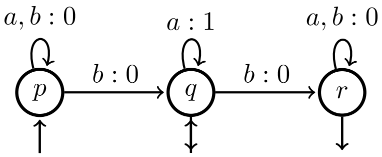
We have seen in the above section how to decide whether the closure under product of a set of matrices over the tropical semiring is finite. The bounded section problem is a generalisation of this problem, which requires a more subtle analysis. The problem now is not to check whether infinitely many matrices are generated, but more precisely to determine what are the entries in the matrices which can get unbounded.

Formally, the *bounded section problem* is the following:

Input: A positive integer n, a finite set of matrices $X \subseteq \mathbb{T}^{n\times n}$, and two n-tuples $I, F \in \{0, \infty\}^n$.

Output: "Yes", if there is m such that for all $A \in \langle X\rangle_{\mathbb{T}^{n\times n}}$, $I^tAF \leqslant m$. "No" otherwise.

Before presenting a decision procedure for this problem (Theorem 4.10 below), we introduce a related problem, the limitedness problem for distance automata. Distance automata are non-deterministic finite automata in which each transition is labelled by a *cost* among $0, 1$. The cost of a run of such an automaton is the sum of the costs of its transitions. The cost of a word is the minimum cost over all possible runs of the automaton over this input. This value can be ∞ if there are no such runs, otherwise it is a non-negative integer. For instance, the following automaton computes the minimal size of a maximal segment of consecutive a's (*i.e.*, maps $a^{n_1}ba^{n_2}\dots ba^{n_k}$ to $\max(n_1, \dots n_k)$):



It does this by guessing the position of the maximal segment of consecutive a's of shortest size and using state q along this segment, state p before, and the state r after. The corresponding run computes the length of this segment by using cost 1 for each a-transition between q-states.

The *boundedness problem* is the following:

Input: A distance automaton $\mathcal{A}$.
Output: "Yes", if the function it computes is bounded. "No", otherwise.

This problem is very close to the original *limitedness problem* studied by Hashiguchi which asks whether the function is bounded over its domain, *i.e.*, over the words that are mapped to an integer by the automaton (a closer inspection shows simple mutual reductions between the two problems; we do not develop it here).

The bounded section problem and the boundedness problem are in fact the same problem. The proof of this equivalence uses the classical argument that weighted automata can be represented by matrices (see Chapter **??**). Indeed, given a distance automaton, it is possible to associate with each letter a a transition matrix A over $\{0, 1, \infty\}$ whose rows and columns are indexed by Q in the following way. The entry with index p, q of the matrix is 0 if in the automaton there is a transition from p to q reading letter a with cost 0 in the automaton, it is 1 a transition of cost 1 (but none of cost 0), and finally it is ∞ if there are no transitions of the automaton at all from p to q while reading letter a. Using this translation, each finite word over $a_1 \dots a_l$ can be transformed into a sequence of matrices $A_1, \dots, A_l \in \mathbb{T}^{n \times n}$. One can prove (by induction) that the entry p, q in the product matrix $A_1 \cdots A_l$ has value m if and only if there is a run of the automaton over $a_1 \dots a_l$ starting in state p, ending in state q, and m is the least cost among all such runs. The entry is ∞ if there is no such run. The sets of initial states and final states can be translated into vectors I and F over $\{0, \infty\}$ by $I(p) = 0$ if p is initial, ∞ otherwise, and $F(p) = 0$ if p is final, ∞ otherwise. It is then easy to see that $I^t A_1 \cdots A_l F$ is exactly the value computed by the automaton while reading the word $a_1 \dots a_l$. Hence the existence of a bound on the function computed by the automaton has been reduced to the bounded section problem. The converse reduction is similar: there is a straightforward translation from a set of matrices $A_1, \dots, A_k$ to a distance automaton over an alphabet of size k (note here that a distance automaton does only use the costs 0 and 1. Therefore the reduction replaces every positive integer by 1. This approximation is valid since we are only interested in boundedness questions.)

Theorem 4.10 ([19]). *The bounded section problem is decidable.*

We present a proof based on the algorithm of Leung [23] and the theorem of factorisation forest for establishing its correctness. Simon later gave another proof for Theorem 4.10 using the factorisation forest theorem [31], but the complexity is not as good as the one obtained by Leung (which is optimal).

The principal idea of this algorithm is similar to the one for the finite closure problem: one "approximates" the matrices in $\mathbb{T}^{n\times n}$ by matrices in $\mathbb{T}_1^{n\times n}$. However, checking whether there exist unstable matrices in $\langle X\rangle_{\mathbb{T}^{n\times n}}$ is now no longer sufficient: indeed, the stability property ensures that no entry in the matrix gets unbounded. Here, we are interested in determining which entries get unbounded. For this, we use the *stabilisation operation* $\sharp$ introduced by Leung. Given any idempotent matrix M in $\mathbb{T}_1^{n\times n}$, it transforms it into a stable matrix $M^\sharp$.

Indeed, when an idempotent matrix A in $\mathbb{T}^{n\times n}$ is not stable, iterating it yields an infinite set of matrices. Thus, some of its entries get to have arbitrary high values when the matrix is iterated. We define $\overline{A}^\sharp$, the *stabilisation* of the matrix $\overline{A}$, to be obtained from the matrix $\overline{A}$ by setting those entries to ∞ whose values are unbounded when the matrix is iterated. This matrix happens to be stable, and it essentially represents the result of iterating the matrix A "many times".

For instance, consider the following idempotent matrix A and its iterations:

$$A = \begin{pmatrix} 0 & 1 \\ \infty & 1 \end{pmatrix}, \qquad A^2 = \begin{pmatrix} 0 & 1 \\ \infty & 2 \end{pmatrix}, \qquad \dots \qquad A^n = \begin{pmatrix} 0 & 1 \\ \infty & n \end{pmatrix}, \qquad \dots$$

The right bottom entry is the only non-infinity one which tends toward infinity in this sequence. The stabilisation reflects this fact in that the corresponding entry is set to infinity:

$$\overline{A} = \begin{pmatrix} 0 & 1 \\ \infty & 1 \end{pmatrix} \quad \text{is stabilised into} \quad \overline{A}^\sharp = \begin{pmatrix} 0 & 1 \\ \infty & \infty \end{pmatrix}\ .$$

Formally, given an idempotent $M \in \mathbb{T}_1^{n\times n}$, the matrix $M^\sharp \in \mathbb{T}_1^{n\times n}$ is defined by:

$$M^\sharp_{i,j} = \begin{cases} 0 & \text{if } M_{i,j} = 0 \\ 1 & \text{if } M_{i,j} = 1 \text{ and for some } k,\ M_{i,k} \neq \infty, M_{k,k} = 0 \text{ and } M_{k,j} \neq \infty \\ \infty & \text{otherwise.} \end{cases}$$

Keeping Lemma 4.7 in mind, one clearly sees then $M^\sharp = M$ iff M is stable. It is also easy to verify that $M^\sharp$ is always idempotent and stable.

Given $Z \subseteq \mathbb{T}_1^{n\times n}$, define $\langle Z\rangle^\sharp \subseteq \mathbb{T}_1^{n\times n}$ to be the closure of Z under product and stabilisation of idempotents. In the remainder of the section we shall prove that:

Lemma 4.11. *For every finite set $X \subseteq \mathbb{T}^{n\times n}$ and all $I, F \in \mathbb{T}^n$, the following statements are equivalent:*

(1) there exists $M \in \langle \overline{X}\rangle^\sharp$ such that $\overline{I}^t M \overline{F} = \infty$,
(2) for all k, there exists some $A \in \langle X\rangle$ such that $I^t A F > k$.

Since the second statement exactly corresponds to the case of a negative answer to the bounded section problem, we obtain a decision procedure for the boundedness problem by taking the set of input matrices X, closing it under product and stabilisation of idempotents, and verifying that $\overline{I}^t B \overline{F} \neq \infty$ for all the resulting matrices. This completes the proof of Theorem 4.10. This procedure is exponential, but a closer inspection of the structure of $\langle X\rangle^\sharp$ reveals in fact that the algorithm can be performed in PSPACE [23]. This also matches the known lower bound from [24].

The remainder of this section is devoted to the proof of Lemma 4.11. This requires to introduce some notations. Given $a \in \mathbb{T}$ and some $k \geqslant 1$ let $\overline{a}^k \in \mathbb{T}_1$ be 0 if $a = 0$, 1 if $1 \leqslant a \leqslant k$, and ∞ otherwise. The intention is that $1 \in \mathbb{T}_1$ represents a “small” value, while $\infty \in \mathbb{T}_1$ represents a “big” value (not necessarily infinity). Seen like this, the mapping which to a associates $\overline{a}^k$ tells us whether the value a should be considered as small or big, where k denotes the threshold between “small” and “big”. One easily checks that $\overline{ab}^{2k} \leqslant \overline{a}^k\overline{b}^k \leqslant \overline{ab}^k$. Since this operation is non-decreasing, this inequality extends to matrices in a natural way: if A, B are matrices over the tropical semiring, then $\overline{AB}^{2k} \leqslant \overline{A}^k\overline{B}^k \leqslant \overline{AB}^k$ where $\overline{\quad}^k$ is extended to matrices componentwise. More generally, $\overline{A_1 \cdots A_m}^{mk} \leqslant \overline{A_1}^k \cdots \overline{A_m}^k \leqslant \overline{A_1 \cdots A_m}^k$.

Given matrices $A_1, \ldots, A_m \in \mathbb{T}^{n\times n}$ (we also use the same definition for matrices in $\mathbb{T}_1^{n\times n}$), a *path from i_0 to i_l in $A_1 \ldots A_m$* is a sequence $p = i_0, \ldots, i_m$ of elements among $1 \ldots n$ such that $i = i_0$ and $i_m = j$. Its *value* $v(p)$ is the sum $(A_1)_{i_0,i_1} + \cdots + (A_m)_{i_{m-1},i_m}$. This definition is related to the product of matrices in the following way: $(A_1 \cdots A_m)_{i,j}$ is the minimum value over all paths from i to j in $A_1, \ldots, A_m$.

Lemma 4.12. *For all $M \in \langle\overline{X}\rangle^\sharp$ and all $k \geqslant 1$, there exists $A \in \langle X\rangle$ such that $M \leqslant \overline{A}^k$.*

Proof. The proof is by induction on the number of multiplications needed to produce the matrix M from matrices in X. Fix k. If $M \in \overline{X}$, then $M = \overline{A}$ for some $A \in X$. Hence $M = \overline{A} \leqslant \overline{A}^k$ (whatever k is). If the induction hypothesis holds for M, N, *i.e.*, there are $A, B \in \langle X\rangle$ such that $M \leqslant \overline{A}^k$ and $N \leqslant \overline{B}^k$, then it holds for MN since $AB \in \langle X\rangle$ and $MN \leqslant \overline{A}^k\overline{B}^k \leqslant \overline{AB}^k$.

Finally, the interesting case is when the induction hypothesis holds for an idempotent matrix E. Assume there exists $B \in \langle X\rangle$ such that $E \leqslant \overline{B}^k$, and consider K sufficiently big (for instance $K = kn+3$). We claim that $E^\sharp \leqslant \overline{A}^k$ where $A = B^K$ (which belongs to $\langle X\rangle$). Consider $i, j = 1 \ldots n$, and a path $p = i_0, \ldots, i_K$ from i to j in B^K with value v. We have to prove that $E^\sharp_{i,j} \leqslant \overline{v}^k$. Since we already know that $E = E^K \leqslant \overline{A}^k \cdots \overline{A}^k \leqslant \overline{A^K}^k$, the only interesting case is for entries for which E and $E^\sharp$ differ, *i.e.*, when $E_{i,j} = 1$ and $E^\sharp_{i,j} = \infty$. By definition of stabilisation, this implies that for all $l = 1 \ldots n$,

$$\text{either} \quad E_{i,l} = \infty, \qquad \text{or} \quad E_{l,j} = \infty, \qquad \text{or} \quad E_{l,l} \geqslant 1\,. \qquad (\star)$$

Since we have chosen K sufficiently large, there is some state l which appears at least $k+1$ times among $i_1, \ldots, i_{K-1}$. This induces a decomposition of p into $p_0, \ldots, p_{k+1}$, in which each p_m is a path in some B^{K_m}. The path p_0 is from i to l, p_1,...,p_k are from l to l, and p_{k+1} is from l to j. We distinguish three cases depending on $\star$. If $E_{i,l} = \infty$, then:

$$\infty = E_{i,l} = (E^{K_0})_{i,l} \leqslant \left(\overline{A}^k\right)^{K_0}_{i,l} \leqslant \overline{A^{K_0}}^k_{i,l} \leqslant \overline{v(p_0)}^k \leqslant \overline{v(p)}^k\,,$$

from which we deduce that $v(p) > k$. The same holds if $E_{j,l} = \infty$. The third case is when $E_{l,l} = 1$. Then, the same chain of inequalities yields $\overline{v(p_m)}^k \geqslant 1$ for all $m = 1 \ldots k$. Hence $v(p_m) \geqslant 1$. As a consequence, we have once more $v(p) > k$. □

Corollary 4.13. *Statement (1) of Lemma 4.11 implies Statement (2).*

Proof. Assume that $\overline{I}^t M\overline{F} = \infty$, and fix k. By Lemma 4.12, there exists a matrix $A \in \langle X\rangle$ such that $M \leqslant \overline{A}^k$. Hence $\infty = \overline{I}^t M\overline{F} \leqslant \overline{I}^t\overline{A}^k\overline{F} \leqslant \overline{I^tAF}^k$, *i.e.*, $I^tAF > k$. □

The second implication is the more involved one. It amounts to proving the following lemma.

Lemma 4.14. *There exists k such that for all $A \in \langle X\rangle$, $\overline{A}^k \leqslant M$ for some $M \in \langle\overline{X}\rangle^\sharp$.*

Corollary 4.15. *Statement (2) of Lemma 4.11 implies Statement (1).*

Proof. Assume (2), and let k be the positive integer obtained from Lemma 4.14. Then there is some $A \in \langle X\rangle$ such that $I^tAF > k$, *i.e.*, $\overline{I^tAF}^k = \infty$. Furthermore, by Lemma 4.11, $\overline{A}^k \leqslant M$ for some $M \in \langle\overline{X}\rangle^\sharp$. We obtain $\infty = \overline{I^tAF}^k = \overline{I}^t\overline{A}^k\overline{F} \leqslant \overline{I}^t M\overline{F}$. This establishes (1). □

It remains to prove Lemma 4.14. For the rest of this section, let us say that a set $Y \subseteq \mathbb{T}_1^{n\times n}$ *covers* a set $X \in \mathbb{T}^{n\times n}$ if there exists $k \geqslant 1$ such that, for all $A \in X$, there exists $M \in Y$ such that $\overline{A}^k \leqslant M$. In this case, k is called the *witness*. Using this terminology, Lemma 4.14 can be rephrased simply as '$\langle\overline{X}\rangle^\sharp$ covers $\langle X\rangle$'. Call two matrices over $\mathbb{T}^{n\times n}$ 0-*equivalent* if they coincide on their 0 entries. Call a matrix 0-*idempotent* if it is 0-equivalent to its square.

Lemma 4.16. *If Y covers X, and all matrices in X are 0-equivalent and all are 0-idempotents, then $\langle Y\rangle^\sharp$ covers $\langle X\rangle$.*

Proof. Let $A_1,\ldots,A_n \in X$, and set $A = A_1\cdots A_n$. We have to prove that there is some $M \in \langle Y\rangle^\sharp$ such that $\overline{A}^k \leqslant M$, in which k must be constructed independently from $A_1,\ldots,A_n$ (and in particular independently from n).

We first claim that for all k and all idempotent $E \in \langle Y\rangle^\sharp$, if $\overline{A_1}^k \leqslant E$ and $\overline{A_n}^k \leqslant E$ then $\overline{A}^{2k} \leqslant E^\sharp$ (note that we do not make here any assumptions on $A_2,\ldots,A_{n-1}$). Indeed, consider $i,j = 1\ldots n$. If $E^\sharp_{i,j} = \infty$, we of course have $\overline{A}^{2k} \leqslant \infty = E^\sharp_{i,j}$. If $E^\sharp_{i,j} = 0$, this means that $E_{i,j} = 0$ and, as a consequence, there is a path from i to j in E^n with value 0. Since all the 0-entries in E are also 0-entries in each A_m, the same path can be used in $A_1,\ldots,A_n$. The last case is $E^\sharp_{i,j} = 1$. By definition of stabilisation, this implies that there is some l such that $E_{i,l} \leqslant 1$, $E_{l,l} = 0$ and $E_{l,j} \leqslant 1$. Consider the path $p = i,l,\ldots,l,j$ in $A_1,\ldots,A_n$. Since $\overline{A_1}^k \leqslant E$ and $E_{i,l} \leqslant 1$, we have $(A_1)_{i,l} \leqslant k$. In the same way $(A_n)_{l,j} \leqslant k$. Furthermore, since $E_{l,l} = 0$ and using the 0-equivalence assumption, we obtain $(A_m)_{l,l} = 0$ for all m. Hence the value of p is at most $2k$. This concludes the claim.

Consider now the general case. Let k be the witness that Y covers X anf fix $M_1,\ldots,M_n \in Y$ such that $\overline{A_i}^k \leqslant M_i$ for each i.

We choose a sufficiently large K. Given an element $N \in \langle\overline{Y}\rangle^\sharp$, we say that N *appears* in $M_1,\ldots,M_n$ between positions m,m' if $N = M_m\cdots M_{m'}$ and $m'-m < K$.

The proof is by induction on the number of idempotents appearing in $M_1, \dots, M_n$. More precisely, we prove that for each i there exist a constant k_i such that, if at most i distinct idempotents appear in $M_1, \dots, M_n$, then $\overline{A_1 \dots A_n}^{k_i} \leqslant M$ for some $M \in \langle X \rangle^\sharp$.

For $i = 0$, no idempotents appear in $M_1, \dots, M_n$. This means that n is small, *i.e.*, $n < K$ (indeed, by the theorem of Ramsey or the factorisation forest theorem, every sufficiently long product has to contain an idempotent). We have:

$$\overline{A_1 \cdots A_n}^{Kk} \leqslant \overline{A_1 \cdots A_n}^{nk} \leqslant \overline{A_1}^{k} \cdots \overline{A_n}^{k} \leqslant M_1 \cdots M_n \in \langle X \rangle^\sharp \,.$$

Suppose now that $i \geqslant 1$ idempotents appear in $M_1, \dots, M_n$. Let E be one of them. We first treat the case where E appears both at the beginning and the end of $M_1, \dots, M_n$, *i.e.*, both between positions $1, m$, and between position m', n. There are two cases. If $m + 1 \geqslant m'$, the two appearances of E overlap or are contiguous. In this case, by definition of appearance, $n \leqslant 2K$ and, as in the case $i = 0$, we obtain that $\overline{A_1 \cdots A_n}^{2Kk} \leqslant N$ for some $N \in \langle X \rangle^\sharp$. Otherwise, we know that $\overline{A_1 \cdots A_m}^{Kk} \leqslant E$, and $\overline{A_{m'} \cdots A_n}^{Kk} \leqslant E$. Hence we can use our first claim on the following sequence of matrices:

$$(A_1 \cdots A_m),\, A_{m+1},\, \dots,\, A_{m'-1},\, (A_{m'} \cdots A_n) \,,$$

and we obtain $\overline{A_1 \cdots A_n}^{2Kk} \leqslant E^\sharp$.

The general case is now easy. Consider a sequence $A_1, \dots, A_n$. It can be decomposed into three sequences

$$U = (A_1, \dots, A_{m-1}),\ V = (A_m, \dots, A_{m'-1}),\ W = (A_{m'}, \dots, A_n) \,,$$

such that E does not appear in U nor W, but both at the beginning and the end of V. According to the induction hypothesis on U and W, there exists $M, M' \in \langle X \rangle^\sharp$ such that $\overline{A_1 \cdots A_{m-1}}^{k_{i-1}} \leqslant M$ and $\overline{A_{m'} \cdots A_n}^{k_{i-1}} \leqslant M'$. Using the previous case with E appearing at the beginning and the end of V, we also have $\overline{A_m \cdots A_{m'-1}}^{2Kk} \leqslant N$ for some $N \in \langle X \rangle^\sharp$. Overall,

$$\begin{aligned} \overline{A_1 \cdots A_m}^{2Kk + 2k_{i-1}} &\leqslant \overline{A_1 \cdots A_{m-1}}^{k_{i-1}} \overline{A_m \cdots A_{m'-1}}^{2Kk} \overline{A_{m'} \cdots A_n)}^{k_{i-1}} \\ &\leqslant MNM' \in \langle X \rangle^\sharp \,, \end{aligned}$$

This establishes the induction hypothesis with $k_i = 2Kk + 2k_{i-1}$. □

We can now conclude the proof of Lemma 4.11 using the factorisation forest theorem.

Proof of Lemma 4.11. Let P be the set of all subsets $Y \subseteq \langle X \rangle$ that are covered by $\langle \overline{X} \rangle^\sharp$. We also say that a set covered by $\langle X \rangle$ *has porperty* P. Consider the morphism f mapping each element of $\langle X \rangle$ to its 0-equivalence class.

Let us show that one can apply Theorem 4.2 to $\langle X \rangle$, which is generated by X, the morphism being f and the family P:

(1') If Y is covered by $\langle \overline{X} \rangle^\sharp$, it is clear that the same holds for every subset of Y.

(1") Let k be the maximum over $||A||$ for all $A \in X$. Then, we have $\overline{A}^k \leqslant \overline{A} \in \overline{X}$ for all $A \in X$. Hence X is covered by $\overline{X}$.

(2) If Y, Z are covered by $\langle \overline{X} \rangle^\sharp$ with respective witnesses k_Y and k_Z, then $Y \cup Z$ is covered by $\langle \overline{X} \rangle^\sharp$, taking as witness $\max(k_Y, k_Z)$.

(3) If Y, Z are covered by $\langle\overline{X}\rangle^\sharp$, witnessed by k_Y and k_Z, then $\overline{AB}^{k_A+k_B} \leqslant \overline{A}^{k_A}\overline{B}^{k_B}$. Hence $k_A + k_B$ is a witness that $(Y \cdot Z)$ is covered by $\langle\overline{X}\rangle^\sharp$.

(4) Finally, suppose that Y is covered by $\langle\overline{X}\rangle^\sharp$ and that $f(Y) = \{E\}$ an idempotent E. Since Y is covered by $\langle\overline{X}\rangle^\sharp$, Lemma 4.14 implies that $\langle Y\rangle$ is covered by $\langle\langle\overline{X}\rangle^\sharp\rangle^\sharp$, *i.e.*, by $\langle\overline{X}\rangle^\sharp$.

Overall, by Theorem 4.2, we conclude that $\langle\overline{X}\rangle^\sharp$ covers $\langle X\rangle$. This concludes the proof of Lemma 4.14, and hence of Theorem 4.10. □

4.5 Polynomial closure

Our last algebraic application of the factorisation forest theorem concerns the problem of finding characterisations of families of regular languages. In Chapters **??** and **??** of this handbook, this topic is treated much more deeply.

The factorisation forest theorem is used in this context to obtain characterisations (possibly non-effective) of the polynomial closure of a class of languages. Given a class of languages $\mathcal{L}$, a language K belongs to its polynomial closure $\mathrm{Pol}(\mathcal{L})$ if it is a finite union of languages of the form $L_0a1L_1\ldots a_nL_n$, where each L_i belongs to $\mathcal{L}$ and the a_i's are letters. In general, the idea is to transform a characterisation of $\mathcal{L}$ (by profinite equations, identities, $\ldots$) into another one for $\mathrm{Pol}(\mathcal{L})$. The first use of this technique appear in Pin and Weil [26] for positive varieties, and the most general and recent such result treats the case of the polynomial closure of any lattice of regular languages [6]. A similar technique is used for characterising another pseudovariety of regular languages. We present here the simplest among the results of this kind: the characterisation of polynomial languages. This corresponds to the case when $\mathcal{L}$ contains the languages of the form B^* where B is any set of letters. The interest of this particular case is that the family of languages obtained in this way coincide with the ones definable in Σ_2, *i.e.*, the fragment of first-order logic consisting of formulas which take the form of a block of existential quantifiers, followed by a block of universal quantifiers, followed by a quantifier-free formula.

A *monomial language* is a language of the form $A_0^*a_1A_1^*\ldots a_nA_n^*$ in which $a_1,\ldots,a_n$ are letters, and $A_0,\ldots,A_n$ are sets of letters. For instance $\{\varepsilon\}$ is the monomial language defined by $\emptyset^*$, and $\{a\}$ is defined as $\emptyset^*a\emptyset^*$. A *polynomial language* is a finite union of monomial languages.

Theorem 4.17 ([26]). *A language is a polynomial langauge if and only if its syntactic ordered monoid satisfies* $e \geqslant e\langle\{s \,:\, e \leqslant_{\mathcal{J}} s\}\rangle e$ *for all idempotent* e.

The exact content of the inequality $e \geqslant e\langle\{s \,:\, e \leqslant_{\mathcal{J}} s\}\rangle e$ may be at first sight uneasy to grasp. To give some more intuition, let us make the following remark before entering the proof.

Remark 4.18. At first glance the constraint $e \geqslant e\langle\{s \,:\, e \leqslant_{\mathcal{J}} s\}\rangle e$ is quite unintuitive. Let us denote by $alph(u)$ the set of letters occurring in a word $u \in A^*$. Then

$$\langle\{s \,:\, a \leqslant_{\mathcal{J}} s\}\rangle = \{f(u) \,:\, alph(u) \subseteq alph(f^{-1}(a))\}$$

for all $a \in M$, *i.e.*, this set represents the possible values of all words that consists of letters that could appear in a word evaluating to a. Hence, the condition $e \geqslant e\langle\{s \ : \ e \leqslant_{\mathcal{J}} s\}\rangle e$ tests the occurring letters in an idempotent context.

Consider now the particularly simple monomial langauge B^* for some $B \subseteq A$. Then clearly, the membership to this language has only to do with the occurring letters; more precisely, if $alph(u) \subseteq alph(v)$ and $v \in B^*$, then $u \in B^*$. The above property $e \geqslant e\langle\{s \ : \ e \leqslant_{\mathcal{J}} s\}\rangle e$ is a form of generalisation of this remark to polynomial languages: in particular it implies that whenever u is an idempotent word, if xuy is in the language, $xuvuy$ is also in the language for all v such hat $alph(v) \subseteq alph(u)$.

Proof. From left to right. Assume that L is a polynomial language and let k be the maximal degree of a monomial it contains. Consider now an idempotent word u and assume $xuy \in L$ for some words x, y, then $xu^{k+1}y \in L$ (by idempotency of u). This word belongs to one of the monomials of L, say $K = A_0^*a_1 \dots a_l A_l^*$, with $l \leqslant k$. Since there are $k+1$ occurrences of the word u in v, at least one is contained in one of the A_i^*. This means that $xu^s \in A_0^*a_1 \dots a_i A_i^*$, $u \in A_i^*$, and $u^{k-s} \in A_i^* a_{i+1} \dots a_l A_l^*$. Let now w be any word such that $alph(w) \subseteq alph(u)$. From the above decomposition, we have that $xu^s uwuu^{k-s}y$ also belong to K and hence to L. Using the idempotency of u, this is also the case for $xuwuy$.

From right to left. This direction uses the factorisation forest theorem.

We denote by $P \subseteq \mathcal{P}(A^*)$ the set

$$\{X \subseteq A^* \ : \ \text{for every } a \in M, \text{ there exists a polynomial language } K_a \\ \text{such that } X \cap f^{-1}(a{\downarrow}) \subseteq K_a \subseteq f^{-1}(a{\downarrow})\}$$

We will apply Theorem 4.2 to the family P to show that $A^* \in P$. By definitions of P, this means that, for every $a \in M$ there exists a polynomial K_a such that $K_a = f^{-1}(a{\downarrow})$. Since polynomial languages are closed under finite union of polynomials, it follows that, for every ideal $I \subseteq M$, $f^{-1}(I)$ is a polynomial language. This concludes this direction of the proof.

What needs to be proved concerning ordered monoids?

It remains to show that Theorem 4.2 can indeed be applied to P. It is clear that $A \in P$ since every finite language is a polynomial language. It is also clear from the definition that P is closed under taking subsets, and under unions (since polynomial languages are closed under unions). Consider now A, B in P. Let us show that $A \cdot B \in P$. Let $(K_x)_{x \in M}$ (resp. $(K'_x)_{x \in M}$) be the polynomial languages witnessing the fact that $A \in P$ (resp. $B \in P$). Consider now the polynomial language:

$$K = \sum_{x \cdot y \leqslant a} K_x K'_y$$

By construction, $f(u) \leqslant a$ for every word $u \in K$. Hence $K \subseteq f^{-1}(a{\downarrow})$. Consider now $u \in (A \cdot B) \cap f^{-1}(a{\downarrow})$. Since $u \in A \cdot B$, u can be decomposed as $u = vw$ with $v \in A$, $w \in B$. By hypothesis, $v \in A$, and hence $v \in K_{f(v)}$. Similarly $w \in K'_{f(w)}$. We get $u = vw \in K_{f(v)}K'_{f(w)}$. Since furthermore $u \in f^{-1}(a{\downarrow})$, we have $f(v) \cdot f(w) = f(u) \leqslant a$ and, as a consequence, $K_{f(v)}K'_{f(w)} \subseteq K$. Overall $u \in K$.

It remains to check the last condition. Assume $A \in P$ and $f(A) = \{e\}$ for some idempotent e. Let K_x for $x \in M$ be the polynomial languages witnessing that $A \in P$.

Consider the following polynomial:

$$K' = K_e + K_e\,alph(f^{-1}(e))^* K_e\ .$$

By the above remark, we know that $f(K') \subseteq f^{-1}(e{\downarrow})$. Conversely, assume that $u \in A^+$. Then u can be written as $u_1 \dots u_n$ with $u_i \in A$ for all i. Clearly, if $n = 1$, then $u \in K_e \subseteq K'$. Otherwise, since $u_2, \dots, u_{n-1}$ belong to $alph(f^{-1}(e))$, $u \in K'$. Hence

$$A^+ \subseteq K' \subseteq f^{-1}(e{\downarrow})\ .$$

Let us prove that $A^+ \in P$ using the above inequalities. Consider some $a \in M$. If $e \notin f^{-1}(a{\downarrow})$, set $K_a = \emptyset$ since $f(A^+) = \{e\}$ we have $A^+ \cap f^{-1}(a{\downarrow}) = \emptyset \subseteq K_a \subseteq f^{-1}(a{\downarrow})$. Otherwise, set $K_a = K'$, and we have by the above inqualities

$$A^+ \cap f^{-1}(a{\downarrow}) \subseteq K' = K_a \subseteq f^{-1}(e{\downarrow}) \subseteq f^{-1}(a{\downarrow})\ .$$

Hence $A^+ \in A$, and Theorem 4.2 can indeed be applied to P. □

5 A deterministic variant of the factorisation forest theorem

The factorisation forest theorem states the existence of a factorisation (of bounded depth) for each word. A natural question to ask is whether a similar result holds for trees. We do not fully resolve this question, the exact formal statement of which is even unclear. Nevertheless, we present a variant of the factorisation forest theorem which has several interesting consequences over trees.

Given a tree, each of its branches can be seen as a word, on which one can apply the factorisation forest theorem. For each branch, this provides a Ramsey split (or factorisation). However, there is no reason, a priori, that two branches sharing a common prefix have a common split on this prefix. This property is desirable in several applications. In particular, it implies that a single split over the tree induces a Ramsey split over each branch. One can already have some feeling about the difference between the word approach and the tree approach by looking at the memory needed for storing the information. A single split for the tree is an object of size linear in the size of the tree, since it amounts to providing some finite quantity of information on every node. However, storing a different split for each branch requires a quadratic memory, since there can be linearly many branches of linear length (an extremal case consists in a string shaped tree of height $n-1$, the deepest node of which has n children which are leaves: this results in n branches of height n, *i.e.*, memory of size n^2 for a tree of size $2n$). Hence, constructing a single split for the tree means to have a significantly more compact representation.

The theorem described in this section provides a result which, given a tree, provides a single split (of bounded depth) such that on every branch it behaves (almost) like a Ramsey split. The proof is obtained by describing a (finite state) transducer, which reads the input and deterministically outputs the split. We will see below some applications of this result.

To obtain the result we have to slightly weaken the conclusion of the theorem. Instead of constructing Ramsey splits, we construct the weaker form of “forward Ramsey” splits.

A split s over a labelling σ is *forward Ramsey* if, for all x, y, x', y' equivalent for $\sim_s$ and such that $x < y$ and $x' < y'$, we have $\sigma(x, y) \cdot \sigma(x', y') = \sigma(x, y)$. This is a weakening of the notion of Ramsey split, as described by the following remark.

Remark 5.1. Every Ramsey split is forward Ramsey. Indeed, for every $\sim_s$-equivalence class C of a Ramsey split, there is an idempotent e such that $\sigma(x, y) = e$ for all $x < y$ in C. In particular, if $x < y$ and $x' < y'$ belong to C, then $\sigma(x, y) \cdot \sigma(x', y') = e \cdot e = e = \sigma(x, y)$. Hence, the split is forward Ramsey.

However, in general, not every forward Ramsey split is Ramsey. Consider for instance the two element semigroup over $\{a, b\}$ defined by $a \cdot a = a \cdot b = a$ and $b \cdot a = b \cdot b = b$. Then for every two elements x, y, $x \cdot y = x$. This implies that every split is forward Ramsey for this semigroup, but not every split is Ramsey.

In some situations, the two notions coincide. For instance, in the case of groups, being Ramsey and being forward Ramsey are equivalent notions. More generally, the notions coincide if and only if $\mathcal{R} = \mathcal{D}$.

The way the notion of a forward Ramsey split is often used is by saying that for all $x < y < z$ with $x \sim_s y \sim_s z$, we have $\sigma(x, y) = \sigma(x, z)$. Indeed, we have $\sigma(x, z) = \sigma(x, y) \cdot \sigma(y, z) = \sigma(x, y)$.

Theorem 5.2 (improvement of [13]). *For all finite semigroups $\mathbf{S} = \langle S, \cdot \rangle$, alphabets A and morphism φ from A^+ to $\mathbf{S}$, there exists a deterministic and complete automaton $\mathcal{A}$ with at most $|S|^{|S|}$ states and a numbering of its states by $\{1, \ldots, |S|\}$ such that, on every input word $u \in A^*$, the numbering of the states along the unique run of the automaton over u defines a forward Ramsey split for φ_u.*

Without loss of generality, we assume in the sequel that $A = S$ and φ is simply the evaluation morphism.

Our proof consists in giving a direct construction of the automaton equipped with a numbering of its states. This automaton has the property that, when reading a word, the sequence of numbers assumed by the states form a forward Ramsey split for the word. In fact, the arguments involved in the proof are very close to the proof given above for the standard factorisation forest theorem, however, this direct construction makes it different, and likely easier to implement.

A *configuration* is a non-empty sequence $\langle a_1, \ldots, a_n \rangle$ of elements of $\mathbf{S}$. A configuration is called *valid* if furthermore:

(1) $a_i \cdot a_{i+1} \cdots a_j \mathrel{\mathcal{J}} a_i$ for all $i \leqslant j$ in $1 \ldots n$,
(2) $a_i <_{\mathcal{J}} a_j$ for all $i < j$ in $1 \ldots n$.

We construct the deterministic automaton $\mathcal{A}$ as follows:

- the states are the valid configurations,
- the initial state can be chosen arbitrarily,
- the transition function δ is defined for all configuration $\langle a_1, \ldots, a_n \rangle$ and all b by:

$$\delta(\langle a_1, \ldots, a_n \rangle, b) = \langle a_1, \ldots, a_k, (a_{k+1} \cdots a_n \cdot \varphi(b)) \rangle$$

where k is maximal such that $\langle a_1, \ldots, a_k, (a_{k+1} \cdots a_n \cdot b) \rangle$ is a valid configuration,

- each valid configuration is numbered by $h(a_1 \dots a_n) = h_0(a_n)$ where h_0 is any injective mapping from **S** to $\{1, \dots, |\mathbf{S}|\}$ such that $a <_{\mathcal{J}} b$ implies $h_0(a) > h_0(b)$.

Given two states p, q and a word u, let us write $p \overset{u:k}{\to} q$ to denote the fact that the automaton $\mathcal{A}$ can go from state p to state q, and k is the maximal value it outputs on the way. Formally, if $\delta(p, a) = q$ then $p \overset{a:h(q)}{\to} q$, and if $p \overset{u:k}{\to} q$ and $q \overset{u:l}{\to} r$ then $p \overset{uv:\max(k,l)}{\to} r$.

Let us give an example of this construction on a particularly simple example: the syntactic semigroup which recognises over the alphabet $A = \{a, b\}$ the language of words which contains a repetition of a letter, *i.e.*, $L = A^*aaA^* + A^*bbA^*$. This semigroup contains five elements: the element 0 represents the words which contain a repetition of a letter, and the four classes of respectively the words a, b, ab, and ba. The multiplication table is defined by the equations $aa = bb = 0 = 0a = a0 = b0 = 0b = 0$, $aba = a$ and $bab = b$. In terms of $\mathcal{J}$-classes, $0 <_{\mathcal{J}} a \mathrel{\mathcal{J}} b \mathrel{\mathcal{J}} ab \mathrel{\mathcal{J}} ba$. To entirely define the automaton of the construction, we also need to define the injection h. We set $h_0(a) = 1$, $h_0(b) = 2$, $h_0(ab) = 3$, $h_0(ba) = 4$ and $h_0(0) = 5$. This results in the following set of configurations–written in vertical boxes and indexed with the corresponding value of h–:

$$\left\{ \boxed{0}_5,\ \boxed{a}_1,\ \boxed{b}_2,\ \boxed{ab}_3,\ \boxed{ba}_4,\ \boxed{\begin{array}{c} a \\ 0 \end{array}}_1,\ \boxed{\begin{array}{c} b \\ 0 \end{array}}_2,\ \boxed{\begin{array}{c} ab \\ 0 \end{array}}_3,\ \boxed{\begin{array}{c} ba \\ 0 \end{array}}_4 \right\}.$$

Let us see how the resulting automaton would process the word *babbbabaa* starting from configuration $\langle a \rangle$:

$$\boxed{a}_1 \overset{b}{\to} \boxed{ab}_3 \overset{a}{\to} \boxed{a}_1 \overset{b}{\to} \boxed{ab}_3 \overset{b}{\to} \boxed{0}_5 \overset{b}{\to} \boxed{\begin{array}{c} b \\ 0 \end{array}}_2 \overset{a}{\to} \boxed{\begin{array}{c} ba \\ 0 \end{array}}_4 \overset{b}{\to} \boxed{\begin{array}{c} b \\ 0 \end{array}}_2 \overset{a}{\to} \boxed{\begin{array}{c} ba \\ 0 \end{array}}_4 \overset{a}{\to} \boxed{0}_5$$

The objective of the construction is to produce a forward Ramsey split, *i.e.*, for all elements $x < y$ and $x' < y'$ of the same class, $\sigma(x, y) \cdot \sigma(x', y') = \sigma(x, y)$. For $b = \sigma(x, y)$ and $c = \sigma(x', y')$, this means $b \cdot c = b$. The following lemma contains the argument we use to obtain such an equality from the construction:

Lemma 5.3. *Let $a, b, c \in S$ be such that $a \cdot b = a \cdot c = a$ and $a \mathrel{\mathcal{J}} b$, then $b \cdot c = b$.*

Proof. By Fact 2.4, $b \mathrel{\mathcal{J}} a \cdot b$ implies $b \mathrel{\mathcal{L}} a \cdot b = a$. Hence $b = d \cdot a$ for some d. It follows that $b \cdot c = (d \cdot a) \cdot c = d \cdot (a \cdot c) = d \cdot a = b$. □

The next lemma is an analysis of what happens when two positions are in the same class (one can recognise some of the premises of Lemma 5.3 for $b = \varphi(u)$).

Lemma 5.4. *If $\langle v, a \rangle \overset{u:h(a)}{\to} \langle w, a \rangle$ for some non-empty word u then $a \cdot \varphi(u) = a$ and $a \mathrel{\mathcal{J}} \varphi(u)$.*

Proof. We prove by induction on the length of u (possibly empty) the stronger property that whenever $\langle v, a \rangle \overset{u:k}{\to} \langle w \rangle$ with $k \leqslant h(a)$, then:

(1) $\langle w \rangle = \langle v, a_1, \dots, a_m \rangle$ for some $a_1, \dots, a_n \in S$, with

(2) $a_1 \mathrel{\mathcal{J}} a$,

(3) $a_1 \cdots a_m = a \cdot \varphi(u)$,

(4) if $m > 1$, there is a suffix u' of u such that $\varphi(u') = a_2 \cdots a_m$, and,
(5) if $u \neq \varepsilon$, and $m = 1$, then $a \mathrel{\mathcal{J}} \varphi(u)$.

This clearly yields the lemma, by (3) and (5).

One easily checks that the statement holds for $u = \varepsilon$. Assume now that (1),...,(5) holds for some u such that $\langle v, a\rangle \overset{u:k}{\to} \langle v, a_1, \ldots, a_m\rangle$ with $k \leqslant h(a)$. Let c be a letter such that $\langle v, a_1, \ldots, a_m\rangle \overset{c:k'}{\to} \langle w\rangle$ for some $k' \leqslant h(a)$. We aim at establishing the claim for the word uc.

Assume that $v = d_1, \ldots, d_s$ is not a prefix of w. Then by definition of the transition function there is some $l \leqslant s$ such that $w = d_1, \ldots, d_{l-1}, d$ with $d = d_l \cdots d_s \cdot a_1 \cdots a_m \cdot \varphi(c)$. In particular, this means that $d \leqslant_{\mathcal{J}} d_l$. Furthermore $d_l <_{\mathcal{J}} a$ since $\langle d_1, \ldots, d_s, a_1, \ldots a_m\rangle$ is a valid configuration in which $a_1 \mathrel{\mathcal{J}} a$ by (2) of the induction hypothesis. It follows that $k' = h(d) > h(a)$, by choice of h. This contradicts $k' \geqslant h(a)$. Hence (1) holds.

At this point, we know that for some $b_1, \ldots, b_n$,

$$\langle v, a_1, \ldots, a_m\rangle \overset{c:k''}{\to} \langle w\rangle = \langle v, b_1, \ldots, b_n\rangle\ .$$

According to the definition of the transition function δ, two cases can happen. Either $n > 1$ and by definition of the transition function $b_1 = a_1$. Since $a_1 \mathrel{\mathcal{J}} a$ by (2) of the induction hypothesis, we have $b_1 \mathrel{\mathcal{J}} a$. Otherwise if $n = 1$, we have $b_1 = a_1 \cdots a_m \cdot \varphi(c)$. This implies $b_1 \leqslant_{\mathcal{J}} a_1 \mathrel{\mathcal{J}} a$. Conversely, assume that $b_1 <_{\mathcal{J}} a$. This would imply $k' = h(\langle w\rangle) = h(b_1) > h(a)$. This contradicts the assumption that $k' \geqslant h(a)$. Overall, (2) holds.

By (3) of the induction hypothesis, $\varphi(u) = a_1 \cdots a_m$. By definition of δ we obtain $b_1 \cdots b_n = a_1 \cdots a_m \cdot \varphi(c)$ and $b_1 \cdots b_n = \varphi(u) \cdot \varphi(c) = \varphi(uc)$. Hence (3) holds.

Assume $n > 1$. Two cases can happen. If $m = 1$, this means that $n = 2$, $b_1 = a_1$, and $b_2 = \varphi(c)$. It follows that c is a suffix of uc, *i.e.*, a witness for (4). Otherwise $m > 1$. Let u' be the witness of (4) obtained by induction hypothesis for u. Using the definition of the transition function we obtain $\varphi(u'c) = \varphi(u') \cdot \varphi(c) = a_2 \cdots a_m \cdot \varphi(c) = b_2 \cdots b_n$. Therefore, $u'c$ is a witness for (4).

Finally, assume $n = 1$. Then $b_1 = a_1 \cdot d$ with $d = a_2 \cdots a_m \cdot \varphi(c)$. By definition of the transition function this means that $\langle v, a_1, d\rangle$ is not a valid configuration while $\langle v, a_1 \cdot d\rangle$ is valid. One knows further from (2) that $a_1 \mathcal{J} a \mathcal{J} a_1 \cdot d$. This last point implies $d \geqslant_{\mathcal{J}} a \mathcal{J} a_1$. The only possible reason for $\langle v, a_1, d\rangle$ not to be valid is that $d \not>_{\mathcal{J}} a_1$. Hence $d \mathrel{\mathcal{J}} a$. Set u' to ε if $m = 1$, otherwise u' is the suffix of u obtained by (4) of the induction hypothesis. In both case, we have $\varphi(u'c) = d \mathrel{\mathcal{J}} a$. This means that $\varphi(uc) \leqslant_{\mathcal{J}} a$. Since furthermore $\varphi(uc) \geqslant_{\mathcal{J}} a$ by (3), we have $\varphi(uc) \mathrel{\mathcal{J}} a$, *i.e.*, (5). □

Overall, we obtain the following statement:

Lemma 5.5. *Let $q_0, \ldots, q_n$ be the states successively assumed by $\mathcal{A}$ while reading a word u, then $s(0) = h(q_0), \ldots, s(n) = h(q_n)$ is a forward Ramsey split for φ_u.*

Proof. Let s be the split defined by $s(i) = h(q_i)$ for $i = 0 \ldots n$. Let $x < y$, and $x' < y'$ be $\sim_s$-equivalent elements among $1, \ldots, m$.

Since x, y, x', and y' are $\sim_s$-equivalent, $h(q_x) = h(q_y) = h(q_{x'}) = h(q_{y'})$. Let k be this value. Since h is injective on S, there exists a single a such that $h(a) = k$. By definition of h on configurations, $q_x = \langle v, a\rangle$, $q_y = \langle w, a\rangle$, $q_{x'} = \langle v', a\rangle$ and $q_{y'} = \langle w', a\rangle$ for some v, w, v', w'.

Since $x \sim_s y$, by Lemma 5.4, $a \cdot \varphi(u_{x,y}) = a$ and $\varphi(u_{x,y}) \mathrel{\mathcal{J}} a$. The same holds for x', y'. In particular, $a \cdot \varphi(u_{x',y'}) = a$. Applying Lemma 5.3 to $b = \varphi(u_{x,y})$ and $c = \varphi(u_{x',y'})$, we obtain $\varphi(u_{x,y}) \cdot \varphi(u_{x',y'}) = b \cdot c = b = \varphi(u_{x,y})$. Hence, s is forward-Ramsey. □

6 Applications as an accelerating structure

In this last section, we provide applications of the factorisation forest theorem of a different kind. The principle of these applications is that, once computed, a (forward) Ramsey split (or a factorisation tree) can be used as a data structure which allows us to perform some computations in an efficient way. We refer to this use as factorisations as an *acceleration structure*.

6.1 Fast infix evaluation

The canonical example of such an application is a solution to the following question:

> Given a regular language L and a word u, is it possible to efficitently compute a data structure (meaning here in time linear in u, possibly more complex in the presentation of L) such that every request of the form $u_{i,j} \in L$ can be answered very efficiently (meaning here in time independent from u, i, j, *i.e.*, dependent only on the presentation of L)?

Since we are not interested in the exact complexity in terms of L (which depends on the way the language is described, and would require a long and careful analysis), we consider L to be fixed. This means that every parameter depending only on L is considered as a constant. With this convention, the statement boils down to computing a data structure in time linear in u, and answering every request in constant time. In what follows, φ is assumed to be some morphism which recognises L, and $\mathbf{S} = \langle S, \cdot\rangle$ is the corresponding semigroup. The goal is to efficiently compute $\varphi(u_{i,j})$.

There are two straightforward approaches to this problem. The first consists in not performing any pre-computation. In this case, the pre-processing time in constant, however, answering to each request of the form $u_{i,j} \in L$ requires a time in $O(j - i)$, which in the worth case is $O(|u|)$. Another solution would be to pre-compute the answer to every request, and store it in a table. This requires quadratic time (and space), but reaches the constant time objective for query evaluation. These two solutions have the advantage of making weak assumptions on L, namely that it is computable in linear time, for ensuring the bounds given above. None of those solutions does provide a solution to our problem.

A third attempt would be a simple divide and conquer algorithm. The data structure consists of a binary tree, the leaves of which yield the input word when read from left to right. Furthermore, one enriches each node of this tree with the value $\varphi(v)$ in which v is the word obtained by reading the leaves below the node from left to right. For i a leaf below a node n, we define $\mathrm{suff}(i,n)$ to be $\varphi(v)$ where v is the node obtained by reading the leaves below n from left to right, starting from leaf i. This function can be computed using the following formula:

$$\mathrm{suff}(i,n) = \begin{cases} \varphi(i) & \text{if } n = i \\ \mathrm{suff}(i,m) & \text{if } i \text{ appears below the right child } m \text{ of } n, \\ \mathrm{suff}(i,m)\cdot\varphi(m') & \text{if } i \text{ appears below the left child } m \text{ of } n, \\ & \qquad \text{and } m' \text{ is the right child of } n. \end{cases}$$

The correctness of this equation follows from the definition. Using this equation we can compute $\mathrm{suff}(i,n)$ in at most h recursion steps, where h is the height of the tree. The same argument gives an algorithm for computing $\mathrm{pref}(j,m)$ which computes the image under φ of the word below m up to letter j. Given $i<j$, one computes the value $\varphi(u_{i,j})$ as $\mathrm{suff}(i,m)\cdot\mathrm{pref}(j,n)$ where m,n are two sibling nodes such that i appears below m and j below n. Overall, we obtain an algorithm which is linear in h. Since it is clear that we can use an almost balanced tree as data structure, this means that computing $u_{i,j}$ is logarithmic in the length of the word. It is also easy to verify that the tree can be computed in linear time.

If one uses a Ramsey factorisation tree as data structure instead of a binary tree, then one obtains a similar result, but this time the height of the tree being bounded by $3|S|-1$, answering a single query becomes constant time. Note that, for this to work, we have to be able to compute a factorisation tree in linear time, and this is possible. We then reach the following theorem:

Theorem 6.1. *There exists an algorithm which, given a language of finite words L, and a word u,*

- *pre-processes u in time linear in $|u|$, and then;*
- *is able to answer each query of the form "does the factor of u between position i and position j belong to L?" in constant time (constant wrt. $|u|$, but not L).*

When replacing the binary tree of the above algorithm by a Ramsey factorisation tree we have to explain how to compute $\mathrm{suff}(i,n)$ when n is a node of arity at least 3, *i.e*, a node such that $\varphi(n) = e$ is an idempotent, and $\varphi(n_1) = \cdots = \varphi(n_k) = e$ where $n_1,\dots,n_k$ are the children of n read from left to right. In this case, we evaluate $\mathrm{suff}(i,n)$ using:

$$\mathrm{suff}(i,n) = \begin{cases} \mathrm{suff}(i,n_k) & \text{if } i \text{ is below } n_k\ , \\ \mathrm{suff}(i,n_l)\cdot e & \text{if } i \text{ is below } n_l \text{ for some } l<k. \end{cases}$$

Indeed, if i appears below n_l for some $l<k$, then

$$\mathrm{suff}(i,n) = \mathrm{suff}(i,n_l)\cdot\varphi(n_{l+1})\cdots\varphi(n_k) = \mathrm{suff}(i,n_l)\cdot e\cdots e = \mathrm{suff}(i,n)\cdot e\ .$$

In fact, forward Ramsey splits are as good as Ramsey splits for this kind of applications, and, in this case, it is obvious how to construct the structure in linear time: simply by evaluating the deterministic transducer of Theorem 5.2. This is clearly linear in the length of the word.

This use of Ramsey factorisations has been employed in several papers of Bojanczyk and Parys [3, 5, 4], improving on the complexity in terms of the input language, and using the technique for solving XPath queries.

6.2 Acceleration in monadic second-order logic

TODO: Is there a reference to MSO in another chapter?

Let us recall here that Π_1 is the fragment of first-order logic consisting of formulas which, do only contain universal quantifiers (no existential ones) when transformed into prenex normal form. The fragment Σ_2 consists of formulas for which no existential quantifier is in the scope of a universal one. In general, and even on words, Σ_2 is strictly less expressive than full first-order logic, which is itself strictly less expressive than MSO.

The following result was the first use of Ramsey factorisations as an acceleration structure:

Theorem 6.2 (Theorem 2 in [13][3]). *Given an MSO formula $\Psi(x_1, \dots, x_n)$ with free first-order variables, there effectively exists a Σ_2-formula $\Psi^*(x_1, \dots, x_n)$ which uses MSO-definable unary predicates, such that Ψ and Ψ^* are equivalent over trees equipped with the ancestor relation $\sqsubseteq$ and unary predicates.*

This result shows that, in some sense, the gab in expressive power between MSO and Σ_2, which is a weak fragment of first-order logic, can be collapsed by simply adding some extra local (*i.e.*, unary) information.

Below, we give a high level presentation of this proof. It is decomposed into two steps. For simplicity, we assume a fixed tree for the explanations, though of course, the construction is uniform, *i.e.*, the same construction works for all trees.

During the first step, we establishe the result for a binary formula $\Psi(x, y)$ such that $\Psi(x, y)$ implies $x \sqsubseteq y$. In this case, using the standard relationship between MSO and recognizable languages, we can construct a semigroup $\mathbf{M} = \langle M, \cdot \rangle$ (this semigroup depends on the formula, but not on the tree), and an additive labelling σ over the tree, such that it is sufficient to know $\sigma(x, y)$ to determine whether $\Psi(x, y)$ holds. Furthermore, this additive labelling is MSO-definable in the sense that for each $a \in \mathbf{M}$, there is an MSO-formula $\Phi_a(x, y)$ which holds if and only if $x \sqsubseteq y$ and $\sigma(x, y) = a$. This can be established either using tree automata, or directly using the composition method ([27]). Both approaches would go beyond the scope of this chapter.

Our goal is to enhance the tree with suitable MSO-definable unary predicates such that $\sigma(x, y)$ can be reconstructed based on those extra predicates. This would clearly complete the first step. The main extra information we use is a forward Ramsey split s for σ obtained from Theorem 5.2. According to Theorem 5.2, this forward Ramsey split can be computed by a transducer. This means in the present case that the unary predicates "$s(x) = k$" for each fixed $k = 1, \dots, |M|$ are definable in MSO, using this time the standard translation from automata to logic. Let us define for each $k = 1 \dots |M|$ and each node y, $\text{parent}_k(y)$ to be, if it exists, the (unique) ancestor x of y such that $s(x) = k$, and $s(z) < k$ for all z such that $x < z < y$. We label for each $k = 1 \dots |M|$ and each $a \in$

[3] The statement in [13] does only mention first-order rather than Σ_2. The proof is the same.

M, the tree with the MSO-definable unary predicate "$\sigma_k(x) = a$" which expresses that $parent_k(x)$ exists and $\sigma(\text{parent}_k(x), x) = a$.

What remains to be done is to prove that, using only the new predicates "$s(x) = k$" and "$\sigma_k(z) = a$", as well as the ancestor relation, it is possible to reconstruct the value of $\sigma(x, y)$ for any $x \sqsubseteq y$. More formally, we need to provide for each $a \in M$ a Σ_2-formula $\phi_a(x, y)$ which can use the above predicates, and such that $\phi_a(x, y)$ holds if and only if $\sigma(x, y) = a$.

Note at this point that given x and y, we can express in Π_1 whether $x \sim_s y$, simply by implementing the definition of $\sim_s$ ($\star$). In the same way, we can express in Π_1 whether $x = \text{parent}_k(y)$ ($\star\star$). It is also very easy, given some $x \sqsubseteq y$ such that $x \sim_s y$, to check using a Σ_2-formula the value of $\sigma(x, y)$ ($\star\star\star$). Indeed, either $x = y$, and the result is 1 (the unit of the monoid $\mathbf{M}$). Or there exist k and z such that $s(z) = k$ and $parent_k(z) = x$, and in this case the value is simply $\sigma_k(z)$, and we have introduced the suitable unary predicate for testing this.

To conclude the argument, it is sufficient to note that given any two nodes $x \sqsubseteq y$, there exists a sequence $x = z_1 \sqsubseteq \cdots \sqsubseteq z_n = y$ of length at most $4|\mathbf{M}|$ such that for all $i = 1 \ldots n-1$, either $z_i = parent_k(z_{i+1})$ for some k, or $z_i \sim_s z_{i+1}$ ($\natural$).

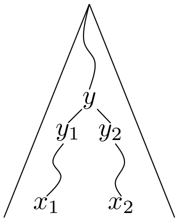
Consider now the following Σ_2-formula:

$$\bigvee_{a_1 \cdot a_2 \cdots a_{n-1} = a,\ n \leqslant 4|M|} \exists z_1 \ldots z_n . \bigwedge_{i=1 \ldots n-1}$$

$$\left(\underbrace{z_i \sim_s z_{i+1}}_{\text{in } \Pi_1 \text{ by } (\star)} \wedge \underbrace{\sigma(z_i, z_{i+1}) = a_i}_{\text{in } \Sigma_2 \text{ by } (\star\star\star) \text{ since } z_i \sim_s z_{i+1}} \right) \vee \left(\bigvee_k \underbrace{\text{parent}_k(z_i, z_{i+1})}_{\text{in } \Pi_1 \text{ by } (\star\star)} \wedge \underbrace{\sigma_k(z_{i+1}) = a}_{\text{new unary predicate}} \right)$$

This formulas holds if and only if $\sigma(x, y) = a$. Indeed, by construction, whenever this formula holds, $\sigma(x, y) = a$. For the converse, assume $\sigma(x, y) = a$. Then the formula holds for the choice of $z_1, \ldots, z_n$ obtained from remark ($\natural$).

The goal of the second step is to generalise the first step to any formula $\Psi(x_1, \ldots, x_n)$. We do not develop this part further. It consists in separating the cases depending on the relationship between x_i's with respect to $\sqsubseteq$. For instance, if $n = 2$, and x_1 and x_2 are incomparable, one chooses y to be the least common ancestor of x_1 and x_2, and y_1 (resp y_2) to be the child of y ancestor of x_1 (resp. x_2). This yields the following picture.



Using syntactic transformations of the formula, e.g., using once more the composition method, it follows that whether $\Psi(x_1, x_2)$ holds depends solely on some local information concerning y, y_1 and y_2, and some formulas involving either y_1, x_1 or y_2, x_2. Those two last cases can be treated using the first step since $y_1 \sqsubseteq x_1$, and $y_2 \sqsubseteq x_2$. Since furthermore x, y_1, y_2 are definable in Σ_2 from x_1 and y_2, one can construct a Σ_2-formula equivalent to $\Psi(x_1, x_2)$.

A consequence in graph theory. The analysis of the structure of graphs is also related to definability questions. There exists a parameter for graphs (and more generally structures) called the clique-width [17] – which we do not develop here – such that a lot of graph problems admit solutions with better complexities when this parameter is bounded (more precisely when the decomposition witnessing the bounded clique-width is known). In this paper, the following fact concerning clique-width is sufficient:

Fact 6.3. A family of graphs has bounded clique-width if and only if it is MSO-definable in a family of trees.

In other words, graphs of bounded-clique width consist in some sense of a tree skeleton on which the graph can be reconstructed, using solely MSO-definable relations. Since our result applies to trees we directly obtain the corollary:

Corollary 6.4. *A family of graphs/structure is of bounded clique width if and only if it is* Σ_2*-definable in a family of trees.*

We redirect the reader to [2] for more pointers in this direction.

Acknowledgments

I am very grateful to Achim Blumensath and Laure Daviaud for their help in the preparation of this document.

References

[1] J. Almeida and O. Klíma. New decidable upper bound of the second level in the straubing-thérien concatenation hierarchy of star-free languages. *Discrete Mathematics & Theoretical Computer Science*, 12(4):41–58, 2010.

[2] A. Blumensath, T. Colcombet, and C. Löding. Logical theories and compatible operations. In *Logic and automata*, volume 2 of *Texts Log. Games*, pages 73–106. Amsterdam Univ. Press, Amsterdam, 2008.

[3] M. Bojanczyk. Factorization forests. In *Developments in Language Theory*, volume 5583, pages 1–17, 2009.

[4] M. Bojanczyk and P. Parys. Efficient evaluation of nondeterministic automata using factorization forests. In *ICALP (1)*, volume 6198 of *Lecture Notes in Computer Science*, pages 515–526. Springer, 2010.

[5] M. Bojanczyk and P. Parys. Xpath evaluation in linear time. *J. ACM*, 58(4):17, 2011.

[6] M. J. J. Branco and J.-E. Pin. Equations defining the polynomial closure of a lattice of regular languages. In *ICALP (2)*, pages 115–126, 2009.

[7] T. C. Brown. On van der waerden's theorem on arithmetic progressions. *Notices Amer. Math. Soc.*, 16:245, 1969.

[8] T. C. Brown. An interesting combinatorial method in the theory of locally finite semigroups. *Pacific J. Math.*, 36:285–289, 1971.

[9] J. R. Büchi. On a decision method in restricted second order arithmetic. In *Proceedings of the International Congress on Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford Univ. Press, 1962.

[10] O. Carton, T. Colcombet, and G. Puppis. Regular languages of words over countable linear orderings. In L. Aceto, M. Henzinger, and J. Sgall, editors, *ICALP (2)*, volume 6756 of *Lecture Notes in Computer Science*, pages 125–136. Springer, 2011.

[11] O. Carton and C. Rispal. Complementation of rational sets on scattered linear orderings of finite rank. *Theoretical Computer Science*, 382(2):109–119, 2007.

[12] J. Chalopin and H. Leung. On factorization forests of finite height. *Theoretical Computer Science*, 310(1–3):489–499, jan 2004.

[13] T. Colcombet. A combinatorial theorem for trees: applications to monadic logic and infinite structures. In *Automata, languages and programming*, volume 4596 of *Lecture Notes in Comput. Sci.*, pages 901–912. Springer, Berlin, 2007.

[14] T. Colcombet. Factorisation forests for infinite words. In *FCT 07*, number 4639 in Lecture Notes in Computer Science, pages 226–237. Springer, 2007.

[15] T. Colcombet. Factorization forests for infinite words and applications to countable scattered linear orderings. *Theoret. Comput. Sci.*, 411(4-5):751–764, 2010.

[16] T. Colcombet. Green's relations and their use in automata theory. In A. H. Dediu, S. Inenaga, and C. Martín-Vide, editors, *LATA*, volume 6638 of *Lecture Notes in Computer Science*, pages 1–21. Springer, 2011. Invited lecture.

[17] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

[18] V. Diekert, P. Gastin, and M. Kufleitner. A survey on small fragments of first-order logic over finite words. *Int. J. Found. Comput. Sci.*, 19(3):513–548, 2008.

[19] K. Hashiguchi. Limitedness theorem on finite automata with distance functions. *J. Comput. Syst. Sci.*, 24(2):233–244, 1982.

[20] M. Kufleitner. The height of factorization forests. In *MFCS*, volume 5162, pages 443–454, 2008.

[21] G. Lallement. *Semigroups and Combinatorial Applications*. Wiley, 1979.

[22] H. Leung. *An Algebraic Method for Solving Decision Problems in Finite Automata Theory*. PhD thesis, Pennsylvania State University, Department of Computer Science, 1987.

[23] H. Leung. Limitedness theorem on finite automata with distance functions: An algebraic proof. *Theoretical Computer Science*, 81(1):137–145, 1991.

[24] H. Leung and V. Podolskiy. The limitedness problem on distance automata: Hashiguchi's method revisited. *Theoretical Computer Science*, 310(1-3):147–158, 2004.

[25] J.-E. Pin. *Varieties of Formal Languages*. North Oxford Academic, London and Plenum, New York, 1986.

[26] J.-E. Pin and P. Weil. Polynomial closure and unambiguous product. *Theory Comput. Syst.*, 30(4):383–422, 1997.

[27] S. Shelah. The monadic theory of order. *Annals of Math.*, 102:379–419, 1975.

[28] I. Simon. Limited subsets of a free monoid. In *FOCS*, pages 143–150. IEEE, 1978.

[29] I. Simon. Recognizable sets with multiplicities in the tropical semiring. In *MFCS*, volume 324 of *Lecture Notes in Computer Science*, pages 107–120. Springer, 1988.

[30] I. Simon. Factorization forests of finite height. *Theoretical Computer Science*, 72:65–94, 1990.

[31] I. Simon. On semigroups of matrices over the tropical semiring. *RAIRO ITA*, 28(3-4):277–294, 1994.

[32] I. Simon. A short proof of the factorization forest theorem. *Tree Automata and Languages*, pages 433–438, 92.