How to prove that a language is regular or star-free? [1]

Jean-Éric Pin[2]

December 19, 2019

Abstract

This survey article presents some standard and less standard methods used to prove that a language is regular or star-free.

Most books of automata theory [9, 23, 29, 46, 50] offer exercises on regular languages, including some difficult ones. Further examples can be found on the web sites `math.stackexchange.com` and `cs.stackexchange.com`. Another good source of tough questions is the recent book *200 Problems in Formal Languages and Automata Theory* [37]. Surprisingly, there is hardly any exercise to prove that a language is star-free. In this paper, we present various methods to prove that a language is regular or star-free.

1 Background

1.1 Regular and star-free languages

Let's start by reminding us what a regular language and a star-free language are.

Definition 1.1. The class of *regular languages* is the smallest class of languages containing the finite languages that is closed under finite union, finite product and star.

The definition of star-free languages follows the same pattern, with the difference that the star operation is replaced by the complement:

Definition 1.2. The class of *star-free languages* is the smallest class of languages containing the finite languages that is closed under finite union, finite product and complement.

---

[2] IRIF, CNRS and Université Paris-Diderot, Case 7014, 75205 Paris Cedex 13, France.

For instance, the language A^* is star-free, since $A^* = \emptyset^c$. More generally, if B is a subset of A, then B^* is star-free since

$$B^* = A^* - \sum_{a \in A-B} A^* a A^* = \left(\sum_{a \in A-B} \emptyset^c a \emptyset^c \right)^c$$

On the alphabet $\{a, b\}$, the language $(ab)^*$ is star-free since

$$(ab)^* = (b\emptyset^c + \emptyset^c a + \emptyset^c aa\emptyset^c + \emptyset^c bb\emptyset^c)^c.$$

Since regular languages are closed under complement, every star-free language is regular, but the converse is not true: one can show that the language $(aa)^*$ is not star-free.

1.2   Early results and their consequences

Kleene's theorem [26] states that regular languages are accepted by finite automata.

Theorem 1.3. *Let L be a language. The following conditions are equivalent:*
  (1) *L is regular,*
  (2) *L is accepted by a finite deterministic automaton,*
  (3) *L is accepted by a finite non-deterministic automaton.*

Given a language L and a word u, the *left [right] quotient* of L by u are defined by $u^{-1}L = \{v \mid uv \in L\}$ and $Lu^{-1} = \{v \mid vu \in L\}$, respectively. The quotients of a regular [star-free] language are also regular [star-free].

Here is another standard result, due to Nerode.

Theorem 1.4. *A language is regular if and only if it has finitely many left (respectively right) quotients.*

Example 1.1. Nerode's theorem suffices to show that if L_1 and L_2 are regular [star-free], then the language

$$L = \{uw \mid \text{there exists } v \text{ such that } uv \in L_1 \text{ and } vw \in L_2\}$$

is also regular [star-free]. Indeed $L = \bigcup_{v \in A^*}(L_1 v^{-1})(v^{-1} L_2)$ and since L_1 and L_2 are regular [star-free], this apparently infinite union can be rewritten as a finite union. Thus L is regular [star-free].

1.3   Recognition by a monoid and syntactic monoid

It is often useful to have a more algebraic definition of regular languages, based on the following result.

Proposition 1.5. *Let L be a language. The following conditions are equivalent:*

(1) *L is regular,*

(2) *L is recognised by a finite monoid,*

(3) *the syntactic monoid of L is finite.*

For readers who may have forgotten the definitions used in this proposition, here are some reminders. A language L of A^* is *recognised by a monoid M* if there is a surjective monoid morphism $f \colon A^* \to M$ and a subset P of M such that $f^{-1}(P) = L$.

The *syntactic congruence* of a language L of A^* is the equivalence relation $\sim_L$ on A^* defined as follows: $u \sim_L v$ if and only if, for every $x, y \in A^*$, xuy and xvy are either both in L or both outside of L. The *syntactic monoid* of L is the quotient monoid $A^*/\!\sim_L$.

Moreover, the syntactic monoid of a regular language is the transition monoid of its minimal automaton, which gives a convenient algorithm to compute it. It is also the minimal monoid (in size, but also for the division ordering[1]) that recognises the language.

Syntactic monoids are particularly useful to show that a language is star-free. Recall that a finite monoid M is *aperiodic* if, for every $x \in M$, there exists $n \geqslant 0$ such that $x^{n+1} = x^n$.

Theorem 1.6 (Schützenberger [47])**.** *For a language L, the following conditions are equivalent:*

(1) *L is star-free,*

(2) *L is recognised by a finite aperiodic monoid,*

(3) *the syntactic monoid of L is a finite aperiodic monoid.*

Schützenberger's theorem is considered, right after Kleene's theorem, as the most important result of the algebraic theory of automata.

Example 1.2. The languages $(ab + ba)^*$ and $(a(ab)^*b)^*$ are star-free, but the languages $(aa)^*$ and $(a + bab)^*$ are not. This is easy to prove by computing the syntactic monoid of these languages.

The following classic example is a good example of the usefulness of the monoid approach. For each language L, let $\sqrt{L} = \{u \mid uu \in L\}$.

Proposition 1.7. *If L is regular [star-free], then so is $\sqrt{L}$.*

Proof. Let $h \colon A^* \to M$ be the syntactic morphism of L, let $P = h(L)$ and let $Q = \{x \in M \mid x^2 \in P\}$. Then

$$h^{-1}(Q) = \{u \in A^* \mid h(u) \in Q\} = \{u \in A^* \mid h(u)^2 \in P\}$$
$$= \{u \in A^* \mid h(u^2) \in P\} = \{u \in A^* \mid u^2 \in L\} = \sqrt{L}.$$

---

[1]Let M and N be monoids. We say that M *divides* N if there is a submonoid R of N and a monoid morphism that maps R onto M.

Thus M recognises $\sqrt{L}$ and the result follows. □

Although the star operation is prohibited in the definition of a star-free language, some languages of the form L^* are star-free. A submonoid M of A^* is *pure* if, for all $u \in A^*$ and $n > 0$, the condition $u^n \in M$ implies $u \in M$. The following result is due to Restivo [44] for finite languages and to Straubing [53] for the general case.

Theorem 1.8. *If L is star-free and L^* is pure, then L^* is star-free.*

Here is another example, based on [52, Theorem 5]. For each language L, let

$$f_W(L) = \{u \in A^* \mid \text{there exists } x, z \in A^* \text{ and } y \in W \text{ with } u = xz \text{ and } xyz \in L\}$$

Proposition 1.9. *If L is regular [star-free], then so is $f_W(L)$.*

Proof. Let $h\colon A^* \to M$ be the syntactic morphism of L and let $P = h(L)$. Note that the conditions $x^{-1}Lz^{-1} \cap W \neq \emptyset$ and $P \cap h(x)h(W)h(z) \neq \emptyset$ are equivalent, for any $x, z \in A^*$. Setting $R = h(W)$ and $T = \{(u,v) \in M \times M \mid uRv \cap P \neq \emptyset\}$ one gets

$$f_W(L) = \bigcup_{(u,v)\in T} h^{-1}(u)h^{-1}(v).$$

and the result now follows easily. □

2 Iteration properties

The bible on this topic is the book of de Luca and Varricchio [13]. I only present here a selection of their numerous results.

2.1 Pumping

The standard pumping lemma is designed to prove that a language is *non-regular*, although some students try to use it to prove the opposite. In a commendable effort to comfort these poor students, several authors have proposed extensions of the pumping lemma that *characterise* regular languages. The first is due to Jaffe [24]:

Theorem 2.1. *A language L is regular if and only if there is an integer m such that every word x of length $\geqslant m$ can be written as $x = uvw$, with $v \neq 1$, and for all words z and for all $k \geqslant 0$, $xz \in L$ if and only if $uv^k wz \in L$.*

Stronger versions were proposed by Stanat and Weiss [51] and Ehrenfeucht, Parikh and Rozenberg [15], but the most powerful version was given by Varricchio [55].

4

Theorem 2.2. *A language L is regular if and only if there is an integer $m > 0$ such that, for all words x, $u_1, \ldots, u_m$ and y, there exist i, j with $1 \leqslant i < j \leqslant m$ such that for all $k > 0$,*

$$xu_1 \cdots u_{i-1}(u_i \cdots u_j)^k u_{j+1} \cdots u_m y \in L \iff xu_1 \cdots u_m y \in L$$

2.2 Periodicity and permutation

Definition 2.3. Let L be a language of A^*.

(1) L is *periodic* if, for any $u \in A^*$, there exist integers $n, k > 0$ such that, for all $x, y \in A^*$, $xu^n y \in L \iff xu^{n+k}y \in L$.

(2) L is *n-permutable* if, for any sequence $u_1, \ldots, u_n$ of n words of A^*, there exists a nontrivial permutation σ of $\{1, \ldots, n\}$ such that, for all $x, y \in A^*$, $xu_1 \cdots u_n y \in L \iff xu_{\sigma(1)} \cdots u_{\sigma(n)}y \in L$.

(3) L is *permutable* if it is permutable for some $n > 1$.

These definitions were introduced by Restivo and Reutenauer [45], who proved the following result.

Proposition 2.4. *A language is regular if and only if it is periodic and permutable.*

2.3 Iteration properties

The book of de Luca and Varricchio [13] also contains many results about iterations properties. Here is an example of this type of results.

Proposition 2.5. *A language L is regular if and only if there exist integers m and s such that for any $z_1, \ldots, z_m \in A^*$, there exist integers h, k with $1 \leqslant h \leqslant k \leqslant m$, such that for all for all $u, v \in A^*$,*

$$uz_1 \cdots z_m v \in L \iff uz_1 \cdots z_{h-1}(z_h \cdots z_k)^n z_{k+1} \cdots z_m v \in L, \qquad (2.1)$$

for all $n \geqslant s$.

3 Rewriting systems and well quasi-orders

Rewriting systems and well quasi-orders are two powerful methods to prove the regularity of a language. We follow the terminology of Otto's survey [38].

3.1 Rewriting systems

A *rewriting system* is a binary relation R on A^*. A pair (ℓ, r) from R is usually referred to as the *rewrite rule* or simply the *rule* $\ell \to r$. A rule is *special* if $r = 1$, *context-free* if $|\ell| \leqslant 1$, *inverse context-free* if $|r| \leqslant 1$, *length-reducing* if $|r| < |l|$. It is *monadic* if it is length-reducing and inverse context-free. A rewriting system is special (context-free, inverse context-free, length-reducing, monadic) if its rules have the corresponding properties.

The *reduction relation* $\xrightarrow{*}_R$ the reflexive and transitive closure of the *single-step reduction relation* $\to_R$ defined as follows: $u \to_R v$ if $u = x\ell y$ and $v = xry$ for some $(\ell, r) \in R$ and some $x, y \in A^*$. For each language L, we set

$$[L]_{\xrightarrow{*}_R} = \{v \in A^* \mid \text{ there exists } u \in L \text{ such that } u \xrightarrow{*}_R v\}$$

A rewriting system R is said to *preserve regularity* if, for each regular language L, the language $[L]_{\xrightarrow{*}_R}$ is regular. The following result is well-known.

Theorem 3.1. *Inverse context-free rewriting systems preserve regularity.*

Proof. Let R be an inverse context-free rewriting system and let L be a regular language. Starting from the minimal deterministic automaton of L, construct an automaton with the same set of states, but with 1-transitions, by iterating the following process: for each rule $u \to 1$ and for each path $p \xrightarrow{u} q$, create a new transition $p \xrightarrow{1} q$; for each rule $v \to a$ with $a \in A$ and for each path $p \xrightarrow{v} q$, create a new transition $p \xrightarrow{a} q$. The automaton obtained at the end of the iteration process will accept $[L]_{\xrightarrow{*}_R}$. $\qquad\square$

A similar technique can be used to prove the following result [39]. If K is a regular language, then the smallest language L containing K and such that $xu^+y \subseteq L \implies xy \in L$ is regular.

3.2 Suffix rewriting systems

A *suffix rewriting system* is a binary relation S on A^*. Its elements are called *suffix rules*. The *suffix-reduction* relation $\xrightarrow{*}_S$ defined by S is the reflexive transitive closure of the *single-step suffix-reduction* relation defined as follows: $u \to_S v$ if $u = x\ell$ and $v = xr$ for some $(\ell, r) \in S$ and some $x \in A^*$. *Prefix rewriting systems* are defined symmetrically. For each language L, we set

$$[L]_{\xrightarrow{*}_S} = \{v \in A^* \mid \text{ there exists } u \in L \text{ such that } u \xrightarrow{*}_S v\}$$

The following early result is due to Büchi [8].

Theorem 3.2. *Suffix (prefix) rewriting systems preserve regularity.*

3.3   Deleting rewriting systems

We follow Hofbauer and Waldmann [22] for the definition of deleting systems. If u is a word, the *content* of u is the set $c(u)$ of all letters of u occurring in u. A *precedence relation* is an irreflexive and transitive binary relation. A precedence relation $<$ on an alphabet A can be extended to a precedence relation on A^*, by setting $u < v$ if $c(u) \neq c(v)$ and, for each $a \in c(u)$, there exists $b \in c(v)$ such that $a < b$. A rewriting system R is $<$-*deleting* if for each rule $\ell \to r$ of R, $\ell < r$.

Hofbauer and Waldmann [22] proved the following result.

Theorem 3.3. *Every deleting string rewriting system preserves regularity.*

3.4   Rules of the form $u^n \to u^m$

Rules of the form $u^n \to u^m$ were studied in several papers, for instance [16, 5, 34].

The following result is due to Bovet and Varricchio [5].

Proposition 3.4. *The rewriting systems $\{(u \to uu) \mid u \in \{a,b\}^*\}$ and $\{(u \to uu) \mid u \in \{a,b\}^*, |u| \leqslant 2\}$ both preserve regularity.*

This result can be used to solve the following exercise. Let L be a language such that, for all $x, y \in A^*$, $x^{-1}Ly^{-1}$ is a semigroup. Prove that L is regular. Indeed, this condition implies that $xuy \in L$ implies $xu^2y \in L$.

Several results were obtained by Leupold [33, 34]. Let us say that a rewriting system is *k-period-expanding* [*k-period-reducing*] if its rules are of the form $u^n \to u^m$, with $n < m$ [$m < n$] and $|u| = k$. Any union of finitely many k-period-expanding and k-period reducing SRSs is called a *k-periodic rewriting system*.

Proposition 3.5 (Leupold)**.**
  (1) *Every k-periodic rewriting system preserves regularity.*
  (2) *For each $k \leqslant 3$, the rewriting system $\{(u \to uu) \mid |u| \leqslant k\}$ preserves regularity.*
  (3) *For each k and for $m \geqslant n$, the rewriting system $\{u^n \to u^m \mid |u| \leqslant k\}$ preserves regularity.*

3.5   Well quasi-orders

A *quasi-order* (or *preorder*) on A^* is a reflexive and transitive relation. A quasi-order $\leqslant$ is *stable* (or *monotone*) if, for all words u, v, x, y, the condition $u \leqslant v$ implies $xuy \leqslant xvy$. A language U is an *upper set* with respect to a quasi-order $\leqslant$ is the conditions $u \in U$ and $u \leqslant v$ imply $v \in U$. The *upper set generated by a language L* is the language $\uparrow L = \{u \in A^* \mid \text{there exists } v \in L \text{ such that } v \leqslant u\}$

A quasi-order $\leqslant$ on A^* is a *well quasi-order* (*wqo*) if every upper set is generated by some finite language. The connection with regular languages was first established in [14] (see also [13, Theorem 6.3.1, p. 203] and [12]).

Theorem 3.6. *A language is regular if and only if it is an upper set with respect to some stable well quasi-order on A^*.*

It follows that if the reduction relation defined by a rewriting system is a well quasi-order, then this rewriting system preserves regularity. Actually, a stronger property holds. Following Conway [11], let us say that a rewriting system R is a *total regulator* if for *any* language L, the language $[L]_{\xrightarrow{*}_R}$ is regular.

Theorem 3.7. *Any rewriting system whose reduction relation is a well quasi-order is a total regulator.*

The most famous example is the rewriting system $\{1 \to a \mid a \in A\}$, which defines the subword ordering. A word $u = a_1 \cdots a_n$ is a *subword* of a word v if $v \in A^* a_1 A^* \cdots A^* a_n A^*$. Higman's theorem states that if A is finite, the subword relation is a well quasi-order on A^*. It follows that for any language L (regular or not), the shuffle product $L \shuffle A^*$ is regular.

The following result extends Higman's theorem on the subword order. Let us say that a set H of words of A^* is *unavoidable* if the language $A^* - A^* H A^*$ is finite.

Theorem 3.8 (Ehrenfeucht, Haussler, Rozenberg [14, Theorem 4.8]). *If H is a unavoidable finite set of words of A^*, then the reduction relation of the rewriting system $\{1 \to u \mid u \in H\}$ is a well quasi-order on A^*.*

A similar result holds for rewriting systems with rules of the form $a \to u$, where a is a letter.

Theorem 3.9 (Bucher, Ehrenfeucht and Haussler [6, Theorem 2.3]). *Let R be a finite rewriting system with rules of the form $a \to x$ with $a \in A$ and $x \in A^*$. The following conditions are equivalent:*

(1) *the relation $\xrightarrow{*}_R$ is a well quasi-order,*

(2) *The set $\{ax \mid x \in A^* \text{ and } a \xrightarrow{*}_R ax\} \cup \{xa \mid x \in A^* \text{ and } a \xrightarrow{*}_R xa\}$ is unavoidable,*

(3) *The set $\{axa \mid x \in A^* \text{ and } a \xrightarrow{*}_R axa\}$ is unavoidable.*

It follows for instance that the following rewriting systems are total regulators:

$$R_1 = \{a \to aa, a \to aba, b \to bb, b \to bab\}$$
$$R_2 = \{a \to b, b \to a, b \to bb\}$$

Bucher, Ehrenfeucht and Haussler [6] considered context-free rewriting systems related to semigroup morphims. Recall that an *ordered semigroup* is a semigroup equipped with a stable partial order. Let $(S, \leqslant)$ be a finite ordered semigroup and let $\sigma \colon A^+ \to S$ be a semigroup morphism. Consider the rewriting system

$$R_\sigma = \{a \to u \mid a \in A, \ u \in A^+ \text{ and } \sigma(a) \leqslant \sigma(u)\}.$$

8

Let $\mathcal{L}$ be a finite set of languages of A^+. Consider a (possibly infinite) system of inequations of the form

$$P_i(X_1, \ldots, X_n) \subseteq E_i(X_1, \ldots, X_n) \quad (i \in I) \tag{3.1}$$

where each $P_i(X_1, \ldots, X_n)$ is a product built from the variables $X_1, \ldots, X_n$ and arbitrary constant languages and each $E_i(X_1, \ldots, X_n)$ is an expression built from the variables $X_1, \ldots, X_n$ and constant languages belonging to the set $\mathcal{L} \cup \{1\}$, using concatenation, possibly infinite union and possibly infinite intersection. Note that the expressions E_i can also use Kleene star, since it can be rewritten as an infinite union of products.

Theorem 3.10 (Kunc [30]). *Let $\sigma \colon A^+ \to S$ be a semigroup morphism that recognises all languages in $\mathcal{L}$. If $\xrightarrow{*}_{R_\sigma}$ is a well quasi-order on A^*, then the components of every maximal solution of (3.1) is regular and they are star-free is S is aperiodic.*

Characterising the semigroup morphisms for which $\xrightarrow{*}_{R_\sigma}$ is a well quasi-order, is an open problem. However, Kunc found a complete answer for finite semigroups $(S, =)$ ordered by the equality relation.

Theorem 3.11 (Kunc [30]). *Let $(S, =)$ be a finite ordered semigroup ordered by the equality relation and let $\sigma \colon A^+ \to S$ be a surjective semigroup morphism. Then the relation $\xrightarrow{*}_{R_\sigma}$ is a well quasi-order on A^* if and only if S is a chain of simple semigroups.*

In particular any finite group is a simple semigroup. It follows that if L is a language recognised by a finite group, then, for any subset S of $\mathbb{N}$, the language $\bigcup_{n \in S} L^n$ is regular.

Example 3.1. The following example is given by Kunc [30, Example 19]. Let L be the language consisting of those words $u \in A^+$ which contain some occurrence of b and where the difference between the length of u and the number of blocks of occurrences of b in u is even. Here is the minimal automaton of this language.
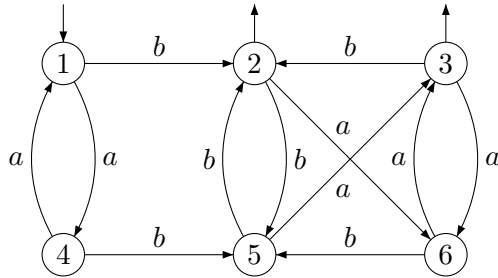
The syntactic semigroup of L is defined by the relations $a^3 = a$, $b^3 = b$, $ab^2 = a^2b$, $ba^2 = b^2a$ and $bab = b^2$. It is a chain of two simple semigroups whose elements are represented by the words a, a^2 and $b, b^2, ab, ab^2, ba, b^2a, aba, ab^2a$, respectively.

Let us consider the inequality $aXaXa \subseteq LXL$ with one variable X. It is easy to verify that this inequality has a largest solution, namely the regular language $(a^2)^*ab^2a(a^2)^* \cup A^*bA^+bA^*$.

3.6 Equations and inequalities

Inequations in languages in which the right hand side is a constant language were first considered by Conway [11], see also Bala [1]. In Chapter 21 of the forthcoming Handbook of Automata Theory, Kunc and Okhotin [32] give the following remarkable result. Consider a finite system of inequations of the form

$$\bigcup_{j \in J_i} P_{i,j}(X_1, \ldots, X_n) \subseteq C_i \quad (1 \leqslant i \leqslant k) \tag{3.2}$$

where each $P_{i,j}(X_1, \ldots, X_n)$ is a product of arbitrary constant languages and variables, each C_i is a constant regular language and each index set J_i is possibly infinite.

Theorem 3.12 (Kunc and Okhotin [32]). *Every system of the form* (3.2) *has only finitely many maximal solutions and every maximal solution has all components regular. If all C_i are star-free, then the maximal solutions are star-free. Furthermore, the result still holds if any inequalities are replaced by equations.*

Proof. Let $h\colon A^* \to M$ be the simultaneous syntactic monoid of the languages C_i. If $(L_1, \ldots, L_n)$ is a solution, then so is $(h^{-1}h(L_1), \ldots, h^{-1}h(L_n))$. It follows that every solution is contained in a solution in which all components are recognised by h and the result follows. $\square$

Inequations of the form $XK \subseteq LX$ were considered by Kunc [30].

Theorem 3.13 (Kunc [30]). *Let K be an arbitrary language and let L be a regular language. Then the greatest solution of the inequality $XK \subseteq LX$ is regular.*

The situation is totally different for equations of the type $XK = LX$. Indeed Kunc [31] has shown that there exists a finite language L such that the greatest solution of the equation $XL = LX$ is co-recursively enumerable complete.

4 Logic

Logic can be used in various ways to characterise regular languages. We consider successively logic on words, linear temporal logic and logic on trees.

4.1 Logic on words

Let $u = a_1 \ldots a_n$ be a nonempty word on the alphabet A. The *domain* of u, denoted by $\mathrm{Dom}(u)$, is the set $\mathrm{Dom}(u) = \{1, \ldots, n\}$. For each letter $a \in A$, let $\mathbf{a}$ be a unary predicate symbol, where $\mathbf{a}x$ is interpreted as "the letter in position x is an a". We also use the binary predicate symbols $<$ and S, interpreted as the usual order relation and the successor relation on $\mathrm{Dom}(u)$, respectively. The language defined by a sentence φ is the set

$$L(\varphi) = \{u \in A^+ \mid u \text{ satisfies } \varphi\}.$$

We let $\mathbf{FO}[<]$ and $\mathbf{MSO}[<]$ denote the set of *first-order* and *monadic second-order* formulas of signature $\{<, (\mathbf{a})_{a \in A}\}$, respectively. Similarly, we let $\mathbf{FO}[S]$ and $\mathbf{MSO}[S]$ denote the same sets of formulas of signature $\{S, (\mathbf{a})_{a \in A}\}$.

Let us say that a syntactic fragment of logic F *captures* a class of languages $\mathcal{C}$ if every sentence of the fragment F defines a language of $\mathcal{C}$ and every language of $\mathcal{C}$ can be defined by a sentence of F.

Two famous results are a natural ingredient of this survey. The first one is due to Buchi [7] and was independently obtained by Elgot [20] and Trakhtenbrot [54].

Theorem 4.1 (Buchi [7]). $\mathbf{MSO}[S]$ *captures the class of regular languages.*

The second one relates first order logic and star-free languages.

Theorem 4.2 (McNaughton [36]). $\mathbf{FO}[<]$ *captures the class of star-free languages.*

Second order logic $\mathbf{SO}$ is much more expressive than monadic second order, but two successive results led to a complete characterisation of the syntactic fragments of $\mathbf{SO}$ — in the signature $\{S, (\mathbf{a})_{a \in A}\}$ — that capture the regular languages.

A *quantifier prefix* is any word on the alphabet $\{\exists, \forall\}$. A *quantifier prefix class* is any set of quantifier prefixes. For any quantifier prefix Q, let $\Sigma_0^1(Q)$ (resp. $\Pi_0^1(Q)$) be the set of all formulas of the shape $\exists \mathbf{R}\, Q\varphi$ (resp. $\forall \mathbf{R}\, Q\varphi$) where $\mathbf{R}$ is a list of relations and φ is quantifier free. For every $k \geqslant 0$, let $\Sigma_{k+1}^1(Q)$ (resp., $\Pi_{k+1}^1(Q)$) be the set of all formulas of the form $\exists \mathbf{R}\, \Phi$ (resp. $\forall \mathbf{R}\, \Phi$) where Φ is a $\Pi_k^1(Q)$ (resp. $\Sigma_k^1(Q)$) formula. Finally, for every quantifier prefix class $\mathcal{Q}$, let $\Sigma_k^1(\mathcal{Q}) = \bigcup_{Q \in \mathcal{Q}} \Sigma_k^1(Q)$.

The fragment Σ_1^1, also known as *existential second order* and frequently denoted by $\mathbf{ESO}$, was first explored by Eiter, Gottlob and Gurevich [17].

Theorem 4.3 (Eiter, Gottlob and Gurevich [17]). *A syntactic fragment $\mathbf{ESO}(\mathcal{Q})$ captures the regular languages if and only if $\mathcal{Q}$ is a quantifier prefix class contained in $\exists^*\forall(\forall \cup \exists^*)$ whose intersection with $\exists^*\forall\{\exists, \forall\}^+$ is nonempty.*

The proof of this result is very difficult. It relies on combinatorial methods related to hypergraph transversals for the fragment $\exists^*\forall\exists^*$ and on more logical techniques for the fragment $\exists^*\forall\forall$. Eiter, Gottlob and Gurevich further proved the following dichotomy theorem: a class **ESO**$[\mathcal{Q}]$ either expresses only regular languages or it expresses some NP-complete languages.

The fragments $\Sigma_k^1(\mathcal{Q})$, with $k \geqslant 2$, were explored by Eiter, Gottlob and Schwentick [18].

Theorem 4.4 (Eiter, Gottlob and Schwentick [18]). *The fragments $\Sigma_2^1(\forall\forall)$ and $\Sigma_2^1(\forall\exists)$ capture the class of regular languages. Furthermore, for each $k \geqslant 0$, the fragments $\Sigma_k^1(\forall)$ and $\Sigma_k^1(\exists)$ only define regular languages.*

For more information on this topic, the reader is invited to read the beautiful survey of Eiter, Gottlob and Schwentick [19].

4.2 Linear temporal logic

Linear temporal logic (LTL for short) on an alphabet A is defined as follows. The vocabulary consists of an atomic proposition p_a (for each letter $a \in A$), the usual connectives $\vee$, $\wedge$ and $\neg$ and the temporal operators $\mathbf{X}$ (*next*), $\mathbf{F}$ (*eventually*) and $\mathbf{U}$ (*until*). The formulas are constructed according to the following rules:

(1) for every $a \in A$, p_a is a formula,

(2) if φ and ψ are formulas, so are $\varphi \vee \psi$, $\varphi \wedge \psi$, $\neg\varphi$, $\mathbf{X}\varphi$, $\mathbf{F}\varphi$ and $\varphi \, \mathbf{U} \, \psi$.

Semantics are defined by induction on the formation rules. Given a word $w \in A^+$, and $n \in \{1, 2, ..., |w|\}$, we define the expression "w satisfies φ at the instant n" (denoted $(w, n) \models \varphi$) as follows:

(1) $(w, n) \models p_a$ if the n-th letter of w is an a.

(2) $(w, n) \models \varphi \vee \psi$ (resp. $\varphi \wedge \psi$, $\neg\varphi$) if $(w, n) \models \varphi$ or $(w, n) \models \psi$ (resp. if $(w, n) \models \varphi$ and $(w, n) \models \psi$, if (w, n) does not satisfy φ).

(3) $(w, n) \models \mathbf{X}\varphi$ if $(w, n + 1)$ satisfies φ.

(4) $(w, n) \models \mathbf{F}\varphi$ if there exists m such that $n \leqslant m \leqslant |w|$ and $(w, m) \models \varphi$.

(5) $(w, n) \models \varphi \, \mathbf{U} \, \psi$ if there exists m such that $n \leqslant m \leqslant |w|$, $(w, m) \models \psi$ and, for every k such that $n \leqslant k < m$, $(w, k) \models \varphi$.

Note that, if $w = w_1 w_2 \cdots w_{|w|}$, $(w, n) \models \varphi$ only depends on the word $w = w_n w_{n+1} \cdots w_{|w|}$.

Example 4.1. Let $w = abbababcba$. Then $(w, 4) \models p_a$ since the fourth letter of w is an a, $(w, 4) \models \mathbf{X}p_b$ since the fifth letter of w is a b and $(w, 4) \models \mathbf{F}(p_c \wedge \mathbf{X}p_b)$ since cb is a factor of $babcba$.

If φ is a temporal formula, we say that w satisfies φ if $(w, 1) \models \varphi$. The language defined by a LTL formula φ is the set $L(\varphi)$ of all words of A^+ that satisfy φ.

A famous result of Kamp [25] states that LTL is equivalent to the first-order logic of order. As a consequence, one gets the following result.

Theorem 4.5. *A language of A^+ is star-free if and only if it is LTL-definable.*

We just defined *future* temporal formulas but one can define in the same way *past* temporal formulas by reversing time: it suffices to replace *next* by *previous*, *eventually* by *sometimes* and *until* by *since*. The expressive power of this extended temporal logic remains the same: it still captures the class of star-free languages.

4.3 Rabin's tree theorem

We now consider the structure $(A^*, (S_a)_{a \in A})$, where each S_a is a binary relation symbol, interpreted on A^* as follows: $S_a(u, v)$ if and only if $v = ua$. Let $\varphi(X)$ be a monadic second order formula with a free set-variable X. We write $\exists! X \, \varphi(X)$ as a short hand for the formula $\exists X \Big(\varphi(X) \wedge \big(\forall Y \, [\varphi(Y) \to (Y = X)] \big) \Big)$. A language L is said to be definable in $\mathbf{MSO}[(S_a)_{a \in A}]$ if there exists a monadic second order formula $\varphi(X)$ such that L satisfies $\exists! X \, \varphi(X)$.

The following result is a consequence of Rabin's tree theorem [43].

Theorem 4.6. *A language of A^* is regular if and only if it is definable in* $\mathbf{MSO}[(S_a)_{a \in A}]$.

5 Transductions

Transductions proved to be a powerful tool to study regular languages. Let us first recall some useful facts about rational and recognisable sets.

5.1 Rational and recognisable sets

Let M be a monoid. A subset P of M is *recognisable* if there exists a finite monoid F, and a monoid morphism $\varphi \colon M \to F$ such that $P = \varphi^{-1}(\varphi(P))$. It is well known that the class $\mathrm{Rec}(M)$ of recognisable subsets of M is closed under Boolean operations, left and right quotients and under inverses of monoid morphisms. The recognisable subsets of a product of monoids were described by Mezei (unpublished).

Theorem 5.1. *Let $M_1, \ldots, M_n$ be monoids. A subset of $M_1 \times \cdots \times M_n$ is recognisable if and only if it is a finite union of subsets of the form $R_1 \times \cdots \times R_n$, where $R_i \in \mathrm{Rec}(M_i)$.*

Furthermore, the following property holds:

Proposition 5.2. *Let $A_1, \ldots, A_n$ be finite alphabets. Then $\mathrm{Rec}(A_1^* \times \cdots \times A_n^*)$ is closed under product.*

The class $\mathrm{Rat}(M)$ of *rational* subsets of M is the smallest set $\mathcal{R}$ of subsets of M containing the finite subsets and closed under finite union, product and star (where X^* is the submonoid of M generated by X). Rational sets are closed under monoid morphisms.

Let A be a finite alphabet. Then, by Kleene's theorem, $\mathrm{Rec}(A^*) = \mathrm{Rat}(A^*)$. Although Kleene's theorem does not extend to arbitrary monoids, a weaker property holds for finitely generated monoids.

Theorem 5.3 (McKnight [35]). *Let M be a monoid. The following conditions are equivalent:*

(1) M *is finitely generated,*

(2) *every recognisable subset of M is rational,*

(3) *the set M is a rational subset of M.*

Furthermore one has:

Theorem 5.4 (McKnight [35]). *The intersection of a rational set and of a recognisable set is rational.*

5.2 Matrix representations of transductions

Let M be a monoid. We denote by $\mathcal{P}(M)$ the semiring of subsets of M with union as addition and the usual product of subsets as multiplication. Note that both $\mathrm{Rat}(M)$ and $\mathrm{Rec}(M)$ are subsemirings of $\mathcal{P}(M)$. Let also $\mathcal{P}(M)^{n \times n}$ denote the semiring of $n \times n$-matrices with entries in $\mathcal{P}(M)$.

Let M and N be two monoids. A *transduction* $\tau \colon M \to N$ is a relation on M and N, viewed as a function from M to $\mathcal{P}(N)$. One extends τ to a function $\mathcal{P}(M) \to \mathcal{P}(N)$ by setting $\tau(P) = \bigcup_{m \in P} \tau(m)$. The inverse transduction $\tau^{-1} \colon N \to M$ is defined by $\tau^{-1}(Q) = \{m \in M \mid \tau(m) \cap Q \neq \emptyset\}$. The transduction is *rational* if the set $\{(m, n) \in M \times N \mid n \in \tau(m)\}$ is a rational subset of $M \times N$.

A transduction $\tau \colon A^* \to M$ admits a *linear matrix representation* (λ, μ, ν) *of degree n* if there exist $n > 0$, a monoid morphism $\mu \colon A^* \to \mathcal{P}(M)^{n \times n}$, a row vector $\lambda \in \mathcal{P}(M)^{1 \times n}$ and a column vector $\nu \in \mathcal{P}(M)^{n \times 1}$ such that, for all $u \in A^*$, $\tau(u) = \lambda \mu(u) \nu$.

A *substitution* from A^* to a monoid M is a monoid morphism from A^* to $\mathcal{P}(M)$. Thus a substitution has linear matrix representation of degree 1.

Kleene-Schützenberger's theorem (see [2]) states that a transduction $\tau \colon A^* \to M$ is rational if and only if it admits a linear matrix representation with entries in $\mathrm{Rat}(M)$.

The following result already suffices for most of the applications we have in mind. It relies on the fact that every monoid morphism $M \to N$ can be extended to a semiring morphism $\mathcal{P}(M) \to \mathcal{P}(N)$ and, for each $n > 0$, to a semiring morphism $\mathcal{P}(M)^{n \times n} \to \mathcal{P}(N)^{n \times n}$.

Theorem 5.5. *Let $\tau\colon A^* \to M$ be a transduction that admits a linear matrix representation (λ, μ, ν) of degree n and let P be a subset of M recognised by a morphism $\eta\colon M \to N$. Then the language $\tau^{-1}(P)$ is recognised by the submonoid $\eta\mu(A^*)$ of the monoid of matrices $\mathcal{P}(N)^{n\times n}$.*

This result was generalised in [40, 41]. Let us say that a transduction $\tau\colon A^* \to M$ admits a *matrix representation (S, μ) of degree n* if there exist a morphism $\mu\colon A^* \to \mathcal{P}(M)^{n\times n}$ and an expression $S(X_{1,1}, \ldots, X_{n,n})$ where S is a possibly infinite union of products involving arbitrary languages and the variables $(X_{i,j})_{1\leqslant i,j\leqslant n}$, such that, for all $u \in A^*$, $\tau(u) = S[\mu_{1,1}(u), \ldots, \mu_{n,n}(u)]$. Theorem 5.5 can now be generalized as follows.

Theorem 5.6. *Let $\tau\colon A^* \to M$ be a transduction that admits a matrix representation (S, μ) of degree n and let P be a subset of M recognised by a morphism $\eta\colon M \to N$. Then the language $\tau^{-1}(P)$ is recognised by the submonoid $\eta\mu(A^*)$ of the monoid of matrices $\mathcal{P}(N)^{n\times n}$.*

Example 5.1. Let us come back to the example $\sqrt{L} = \{u \in A^* \mid u^2 \in L\}$. Observe that $\sqrt{L} = \tau^{-1}(L)$ where $\tau(u) = u^2$. Clearly τ admits the matrix representation (S, μ) where $\mu(u) = u$ and $S = X^2$.

Example 5.2. Let us show that if L is a regular language and S is a subset of $\mathbb{N}^2$ then the language

$$L_S = \{u \in A^* \mid \text{there exist } (x, y) \in A^* \text{ and } (p, q) \in S$$
$$\text{such that } |x| = p|u|, \ |y| = q|u| \text{ and } xuy \in L\}$$

is also regular. It suffices to observe that $L_S = \tau^{-1}(L)$ where the transduction $\tau(u) = \bigcup_{(p,q)\in S} A^{p|u|} u A^{q|u|}$ admits the matrix representation (S, μ), where

$$\mu(u) = \begin{pmatrix} A^{|u|} & \emptyset & \emptyset \\ \emptyset & u & \emptyset \\ \emptyset & \emptyset & A^{|u|} \end{pmatrix} \text{ and } S(X_{1,1}, \ldots, X_{3,3}) = \bigcup_{(p,q)\in S} X_{1,1}^p X_{2,2} X_{3,3}^q$$

Example 5.3. Finally the reader who likes more complicated examples may prove by the same method that if $L \subseteq \{a, b\}^*$ is regular, then the following language is also regular (D^* is the Dyck language):

$$L' = \bigcup_{n \text{ square-free}} \{u \in A^* \mid D^* u^{\lfloor\sqrt{n}\rfloor} a^n u^{n!} b \cap L \neq \emptyset\}$$

Many more examples can be found in [40, 41].

5.3 Decompositions of languages

For each $n > 0$, consider the transduction $\tau_n \colon A^* \to (A^*)^n$ defined by

$$\tau_n(u) = \{(u_1, \ldots, u_n) \mid u_1 \cdots u_n = u\}$$

Theorem 5.7. *Let L be a language of A^*. The following conditions are equivalent:*

(1) *L is rational,*

(2) *for some $n > 0$, $\tau_n(L)$ is a recognisable subset of $(A^*)^n$,*

(3) *for all $n > 0$, $\tau_n(L)$ is a recognisable subset of $(A^*)^n$.*

Proof. (1) implies (3). Let $\mathcal{A} = (Q, A, \cdot, i, F)$ be the minimal automaton of L. For each state p, q of $\mathcal{A}$, let $L_{p,q}$ be the language accepted by $\mathcal{A}$ with p as initial state and q as unique final state. Let $S = \{i\} \times Q^{n-2} \times F$. We claim that

$$\tau_n(L) = \bigcup_{(q_0, \ldots, q_n) \in S} L_{q_0, q_1} \times L_{q_1, q_2} \times \cdots \times L_{q_{n-1}, q_n} \tag{5.1}$$

Let R be the right hand side of (5.1). Let $(u_1, \ldots, u_n) \in \tau_n(L)$. Let $q_0 = i$, $q_1 = q_0 \cdot u_1$, ..., $q_n = q_{n-1} \cdot u_n$. Since $u_1 \cdots u_n \in L$, one has $q_n \in F$ and hence $(q_0, \ldots, q_n) \in S$. Moreover, by construction, $u_1 \in L_{q_0, q_1}, \ldots, u_n \in L_{q_{n-1}, q_n}$ and hence $(u_1, \ldots, u_n) \in R$.

Let now $(u_1, \ldots, u_n) \in R$. Then, for some $(q_0, \ldots, q_n) \in S$, one has $u_1 \in L_{q_0, q_1}$, ..., $u_n \in L_{q_{n-1}, q_n}$. It follows that $q_1 = q_0 \cdot u_1$, ..., $q_n = q_{n-1} \cdot u_n$ and thus $i \cdot u_1 \cdots u_n = q_n \in F$ and hence $u \in L$. $\qquad\square$

6 Profinite topology

Let M be a monoid. A monoid morphism $\varphi \colon M \to N$ *separates* two elements u and v of M if $\varphi(u) \neq \varphi(v)$. By extension, we say that a monoid N *separates* two elements of M if there exists a morphism $\varphi \colon M \to N$ which separates them. A monoid is *residually finite* if any pair of distinct elements of M can be separated by a finite monoid.

Let us consider the class $\mathcal{M}$ of monoids that are finitely generated and residually finite. This class include finite monoids, free monoids, free groups, free commutative monoids and many others. It is closed under direct products and thus monoids of the form $A_1^* \times A_2^* \times \cdots \times A_n^*$ are also in $\mathcal{M}$.

Each monoid M of $\mathcal{M}$ can be equipped with the *profinite metric*, defined as follows. Let, for each $(u, v) \in M^2$,

$$r(u, v) = \min\{\operatorname{Card}(N) \mid N \text{ separates } u \text{ and } v \}$$

Then we set $d(u, v) = 2^{-r(u,v)}$, with the usual conventions $\min \emptyset = +\infty$ and $2^{-\infty} = 0$. One can show that d is an ultrametric and that the product on M is uniformly continuous for this metric.

6.1 Uniformly continuous functions and recognisable sets

The connection with recognisable sets is given by the following result:

Proposition 6.1. *Let $M, N \in \mathcal{M}$ and let $f \colon M \to N$ be a function. Then the following conditions are equivalent:*

  (1) *for every $L \in \mathrm{Rec}(N)$, one has $f^{-1}(L) \in \mathrm{Rec}(M)$,*

  (2) *the function f is uniformly continuous for the profinite metric.*

Here is an interesting example [42].

Proposition 6.2. *The function $g \colon A^* \times \mathbb{N} \to A^*$ defined by $g(x, n) = x^n$ is uniformly continuous.*

Example 6.1. As an application, let us show that if L is a regular language of A^*, then the language

$$K = \{u \in A^* \mid u^{|u|} \in L\}$$

is also regular. Indeed, $K = h^{-1}(L)$, where h is the function defined by $h(u) = u^{|u|}$. Observe that $h = g \circ f$, where $f : A^* \to A^* \times \mathbb{N}$ is the monoid morphism defined by $f(u) = (u, |u|)$ and g is the function defined in Proposition 6.2. Now since $L \in \mathrm{Rec}(A^*)$, one gets $g^{-1}(L) \in \mathrm{Rec}(A^* \times \mathbb{N})$ by Proposition 6.2 and $f^{-1}(g^{-1}(L)) \in \mathrm{Rec}(A^*)$ since f is a monoid morphism. Thus K is regular.

Uniformly continuous functions from $\mathbb{N}$ to $\mathbb{N}$ are of special interest. A function $f \colon \mathbb{N} \to \mathbb{N}$ is *residually ultimately periodic* (rup) if, for each monoid morphism h from $\mathbb{N}$ to a finite monoid F, the sequence $h(f(n))$ is ultimately periodic. It is *cyclically ultimately periodic* if, for every $p > 0$, there exist two integers $m \geqslant 0$ and $r > 0$ such that, for each $n \geqslant m$, $f(n) \equiv f(n + r) \pmod{p}$. It is *ultimately periodic threshold t* if the function $\min(f(n), t)$ is ultimately periodic.

For instance, the functions n^2 and $n!$ are residually ultimately periodic. The function $\binom{2n}{n}$ is not cyclically ultimately periodic. Indeed, it is known that $\binom{2n}{n} \equiv 2 \bmod 4$ if and only if n is a power of 2. It is shown in [49] that the sequence $\lfloor \sqrt{n} \rfloor$ is not cyclically ultimately periodic.

Let us mention a last example, first given in [10]. Let b_n be a non-ultimately periodic sequence of 0 and 1. The function $f(n) = (\sum_{0 \leqslant i \leqslant n} b_i)!$ is residually ultimately periodic. It follows that the function $\Delta f(n) = f(n + 1) - f(n)$ is not residually ultimately periodic since $\min(\Delta f(n), 1) = b_n$.

The following result was proved in [3].

Proposition 6.3. *For a function $f \colon \mathbb{N} \to \mathbb{N}$, the following conditions are equivalent:*

  (1) *f is uniformly continuous,*

  (2) *f is residually ultimately periodic,*

  (3) *f is cyclically ultimately periodic and ultimately periodic threshold t for all $t \geqslant 0$.*

The class of cyclically ultimately periodic functions has been studied by Siefkes [49], who gave in particular a recursion scheme for producing such functions. The class of residually ultimately periodic sequences was also thoroughly studied in [10, 56] (see also [27, 28, 48]). Their properties are summarized in the next proposition.

Theorem 6.4. *Let g and g be rup functions. Then the following functions are also rup: $f \circ g$, $f + g$, fg, f^g, $\sum_{0 \leqslant i \leqslant g(n)} f(i)$, $\prod_{0 \leqslant i \leqslant g(n)} f(i)$. Furthermore, if $f(n) \geqslant g(n)$ for all n and $\lim_{n \to \infty} (f - g)(n) = +\infty$, then $f - g$ is also rup.*

In particular, the functions f^k and $n \to k^{f(n)}$ (for a fixed k), are rup. The *tetration function* $^n 2$ (exponential stack of 2's of height n), considered in [48], is also rup, according to the following result: if k is a positive integer, then the function $f(n)$ defined by $f(0) = 1$ and $f(n + 1) = k^{f(n)}$ is rup.

The existence of non-recursive rup functions was established in [48]: if f is a strictly increasing, non-recursive function, then the function $g(n) = n!f(n)$ is non-recursive but is rup.

Coming back to regular languages, Seiferas and McNaughton [48] proved the following result.

Theorem 6.5. *Let $f : \mathbb{N} \to \mathbb{N}$ be a rup function. If L is regular, then so is the language*

$$P(f, L) = \{x \in A^* \mid \text{there exists } y \in A^* \text{ such that } |y| = f(|x|) \text{ and } xy \in L\}.$$

Here is another application of rup functions. A *filter* is a strictly increasing function $f : \mathbb{N} \to \mathbb{N}$. *Filtering a word* $a_0 a_1 \cdots a_n$ by f consists in deleting the letters a_i such that i is not in the range of f. For each language L, let $L[f]$ denote the set of all words of L filtered by f. A filter is said to *preserve regular languages* if, for every regular language L, the language $L[f]$ is also regular. The following result was proved in [3].

Theorem 6.6. *A filter f preserves regular languages if and only if the function Δf defined by $\Delta f(n) = f(n + 1) - f(n)$ is rup.*

6.2   Transductions and recognisable sets

Some further topological results are required to extend Proposition 6.1 to transductions.

The completion of the metric space (M, d), denoted by $(\widehat{M}, d)$, is called the *profinite completion* of M. Since multiplication on M is uniformly continuous, it extends, in a unique way, to a multiplication on $\widehat{M}$, which is again uniformly continuous. One can show that $\widehat{M}$ is a metric compact monoid.

18

Let $\mathcal{K}(\widehat{M})$ be the monoid of compact subsets of $\widehat{M}$. The *Hausdorff metric* on $\mathcal{K}(\widehat{M})$ is defined as follows. For $K, K' \in \mathcal{K}(\widehat{M})$, let

$$\delta(K, K') = \sup_{x \in K} \inf_{x' \in K'} d(x, x')$$

$$h(K, K') = \begin{cases} \max(\delta(K, K'), \delta(K', K)) & \text{if } K \text{ and } K' \text{ are nonempty,} \\ 0 & \text{if } K \text{ and } K' \text{ are empty,} \\ 1 & \text{otherwise.} \end{cases}$$

By a standard result of topology, $\mathcal{K}(\widehat{M})$, equipped with this metric, is compact.

Let now $\tau \colon M \to N$ be a transduction. Define a map $\widehat{\tau} : M \to \mathcal{K}(\widehat{N})$ by setting, for each $x \in M$, $\widehat{\tau}(x) = \overline{\tau(x)}$, the topological closure of $\tau(x)$. The following extension of Proposition 6.1 was proved in [42].

Theorem 6.7. *Let $M, N \in \mathcal{M}$ and let $\tau \colon M \to N$ be a transduction. Then the following conditions are equivalent:*

(1) *for every $L \in \operatorname{Rec}(N)$, one has $\tau^{-1}(L) \in \operatorname{Rec}(M)$,*

(2) *the function $\widehat{\tau} \colon M \to \mathcal{K}(\widehat{N})$ is uniformly continuous.*

Let us say that a transduction τ is uniformly continuous, if $\widehat{\tau}$ is uniformly continuous. Uniformly continuous transductions are closed under composition and they are also closed under product:

Proposition 6.8. *Let $\tau_1 : M \to N_1$ and $\tau_2 : M \to N_2$ be uniformly continuous transductions. Then the transduction $\tau : M \to N_1 \times N_2$ defined by $\tau(x) = \tau_1(x) \times \tau_2(x)$ is uniformly continuous.*

Proposition 6.9. *For every $M \in \mathcal{M}$, the transduction $\sigma : M \to M$ defined by $\sigma(x) = x^*$ is uniformly continuous.*

7 Further examples

8 Further examples and conclusion

Here are a few results relating regular languages and Turing machines.

Theorem 8.1 ([9, Theorem 3.84, p. 185])**.** *The language accepted by a one-tape Turing machine that never writes on its input is regular.*

Theorem 8.2 (Hartmanis [21])**.** *The language accepted by a one-tape Turing machine that works in time $o(n \log n)$ is regular.*

The following result is proposed as an exercise in [9, Exercise 4.16, p. 243].

Theorem 8.3. *The language accepted by a Turing machine that works in space* $o(\log \log n)$ *is regular.*

Theorem 8.4 ([45]). *If a language and its complement are support of a rational series, then it is a regular language.*

Many other examples that could not be included in this survey, notably the work of Bertoni, Mereghetti and Palano [4, Theorem 3, p. 8] on 1-way quantum automata and the large literature on splicing systems.

I would be very grateful to any reader providing me new interesting examples to enrich this survey.

Acknowledgements

References

[1] S. BALA, Complexity of regular language matching and other decidable cases of the satisfiability problem for constraints between regular open terms, *Theory Comput. Syst.* **39**,1 (2006), 137–163.

[2] J. BERSTEL, *Transductions and context-free languages*, Teubner, 1979.

[3] J. BERSTEL, L. BOASSON, O. CARTON, B. PETAZZONI AND J.-É. PIN, Operations preserving recognizable languages, *Theor. Comput. Sci.* **354** (2006), 405–420.

[4] A. BERTONI, C. MEREGHETTI AND B. PALANO, Quantum computing: 1-way quantum automata, in *Developments in Language Theory*, pp. 1–20, *Lecture Notes in Comput. Sci.* vol. 2710, Springer, Berlin, 2003.

[5] D. P. BOVET AND S. VARRICCHIO, On the regularity of languages on a binary alphabet generated by copying systems, *Inform. Process. Lett.* **44**,3 (1992), 119–123.

[6] W. BUCHER, A. EHRENFEUCHT AND D. HAUSSLER, On total regulators generated by derivation relations, *Theor. Comput. Sci.* **40**,2-3 (1985), 131–148.

[7] J. R. BÜCHI, Weak second-order arithmetic and finite automata, *Z. Math. Logik und Grundl. Math.* **6** (1960), 66–92.

[8] J. R. BÜCHI, Regular canonical systems, *Arch. Math. Logik Grundlagenforsch.* **6** (1964), 91–111 (1964).

[9] O. CARTON, *Langages formels, calculabilité et complexité*, Vuibert, 2008.

[10] O. CARTON AND W. THOMAS, The Monadic Theory of Morphic Infinite Words and Generalizations, *Inform. Comput.* **176** (2002), 51–76.

[11] J. H. CONWAY, *Regular Algebra and Finite Machines*, Chapman and Hall, London, 1971.

[12] F. D'ALESSANDRO AND S. VARRICCHIO, Well quasi-orders in formal language theory, in *Developments in language theory*, pp. 84–95, *Lecture Notes in Comput. Sci.* vol. 5257, Springer, Berlin, 2008.

[13] A. DE LUCA AND S. VARRICCHIO, *Finiteness and regularity in semigroups and formal languages*, *Monographs in Theoretical Computer Science. An EATCS Series*, Springer-Verlag, Berlin, 1999.

[14] A. EHRENFEUCHT, D. HAUSSLER AND G. ROZENBERG, On regularity of context-free languages, *Theoret. Comput. Sci.* **27**,3 (1983), 311–332.

[15] A. EHRENFEUCHT, R. PARIKH AND G. ROZENBERG, Pumping lemmas for regular sets, *SIAM J. Comput.* **10**,3 (1981), 536–541.

[16] A. EHRENFEUCHT AND G. ROZENBERG, On regularity of languages generated by copying systems, *Discrete Appl. Math.* **8**,3 (1984), 313–317.

[17] T. EITER, G. GOTTLOB AND Y. GUREVICH, Existential second-order logic over strings, *J. Acm* **47**,1 (2000), 77–131.

[18] T. EITER, G. GOTTLOB AND T. SCHWENTICK, Second-order logic over strings: regular and non-regular fragments, in *Developments in language theory (Vienna, 2001)*, pp. 37–56, *Lecture Notes in Comput. Sci.* vol. 2295, Springer, Berlin, 2002.

[19] T. EITER, G. GOTTLOB AND T. SCHWENTICK, The model checking problem for prefix classes of second-order logic: a survey, in *Fields of logic and computation*, pp. 227–250, *Lecture Notes in Comput. Sci.* vol. 6300, Springer, Berlin, 2010.

[20] C. C. ELGOT, Decision problems of finite automata design and related arithmetics, *Trans. Amer. Math. Soc.* **98** (1961), 21–51.

[21] J. HARTMANIS, Computational complexity of one-tape Turing machine computations, *J. Assoc. Comput. Mach.* **15** (1968), 325–339.

[22] D. HOFBAUER AND J. WALDMANN, Deleting string rewriting systems preserve regularity, *Theoret. Comput. Sci.* **327**,3 (2004), 301–317.

[23] J. E. HOPCROFT AND J. D. ULLMAN, *Introduction to automata theory, languages, and computation*, Addison-Wesley Publishing Co., Reading, Mass., 1979. Addison-Wesley Series in Computer Science.

[24] J. JAFFE, A necessary and sufficient pumping lemma for regular languages, *SIGACT News* **10**,No. 2 (1978), 48–49.

[25] J. KAMP, *Tense Logic and the Theory of Linear Order*, PhD thesis, Univ. of California, Los Angeles, 1968.

[26] S. C. KLEENE, Representation of events in nerve nets and finite automata, in *Automata studies*, Princeton, N. J., 1956, pp. 3–41, Princeton University Press. Annals of mathematics studies, no. 34.

[27] S. R. KOSARAJU, Regularity preserving functions, *SIGACT News* **6 (2)** (1974), 16–17. Correction to "Regularity preserving functions", SIGACT News 6 (3), (1974), p. 22.

[28] D. KOZEN, On regularity-preserving functions, *Bull. Europ. Assoc. Theor. Comput. Sc.* **58** (1996), 131–138. Erratum: On regularity-preserving functions, Bull. Europ. Assoc. Theor. Comput. Sc. **59** (1996), 455.

[29] D. C. KOZEN, *Automata and computability*, *Undergraduate Texts in Computer Science*, Springer-Verlag, New York, 1997.

[30] M. KUNC, Regular solutions of language inequalities and well quasi-orders, *Theoret. Comput. Sci.* **348**,2-3 (2005), 277–293.

[31] M. KUNC, The power of commuting with finite sets of words, *Theory Comput. Syst.* **40**,4 (2007), 521–551.

[32] M. KUNC AND A. OKHOTIN, Language equations, in *Handbook of Automata Theory*, J.-E. Pin (ed.), vol. II, ch. 21, Eur. Math. Soc., Zürich, 2020. To appear.

[33] P. LEUPOLD, Languages generated by iterated idempotency, *Theoret. Comput. Sci.* **370**,1-3 (2007), 170–185.

[34] P. LEUPOLD, On regularity-preservation by string-rewriting systems, in *Language and automata theory and applications*, pp. 345–356, *Lecture Notes in Comput. Sci.* vol. 5196, Springer, Berlin, 2008.

[35] J. D. MCKNIGHT, JR., Kleene quotient theorems, *Pacific J. Math.* **14** (1964), 1343–1352.

[36] R. MCNAUGHTON AND S. PAPERT, *Counter-free automata*, The M.I.T. Press, Cambridge, Mass.-London, 1971. With an appendix by William Henneman, M.I.T. Research Monograph, No. 65.

[37] D. NIWIŃSKI AND W. RYTTER, *200 Problems in Formal Languages and Automata Theory*, University of Warsaw, 2017.

[38] F. OTTO, On the connections between rewriting and formal language theory, in *Rewriting techniques and applications (Trento, 1999)*, pp. 332–355, *Lecture Notes in Comput. Sci.* vol. 1631, Springer, Berlin, 1999.

[39] J.-É. PIN, Topologies for the free monoid, *J. of Algebra* **137** (1991), 297–337.

[40] J.-É. PIN AND J. SAKAROVITCH, Some operations and transductions that preserve rationality, in *6th GI Conference*, Berlin, 1983, pp. 277–288, *Lecture Notes in Comput. Sci.* n145, Springer.

[41] J.-É. Pin and J. Sakarovitch, Une application de la représentation matricielle des transductions, *Theoret. Comput. Sci.* **35** (1985), 271–293.

[42] J.-É. Pin and P. V. Silva, A topological approach to transductions, *Theoret. Comput. Sci.* **340** (2005), 443–456.

[43] M. O. Rabin, Decidability of second-order theories and automata on infinite trees., *Trans. Amer. Math. Soc.* **141** (1969), 1–35.

[44] A. Restivo, Codes and aperiodic languages, in *Erste Fachtagung der Gesellschaft für Informatik über Automatentheorie und Formale Sprachen (Bonn, 1973)*, pp. 175–181, *Lecture Notes in Comput. Sci.* vol. 2, Springer, Berlin, 1973.

[45] A. Restivo and C. Reutenauer, On cancellation properties of languages which are supports of rational power series, *J. Comput. System Sci.* **29**,2 (1984), 153–159.

[46] J. Sakarovitch, *Elements of automata theory*, Cambridge University Press, Cambridge, 2009. Translated from the 2003 French original by Reuben Thomas.

[47] M.-P. Schützenberger, On finite monoids having only trivial subgroups, *Information and Control* **8** (1965), 190–194.

[48] J. I. Seiferas and R. McNaughton, Regularity-preserving relations, *Theoret. Comput. Sci.* **2**,2 (1976), 147–154.

[49] D. Siefkes, Decidable extensions of monadic second order successor arithmetic, in *Automatentheorie und formale Sprachen (Tagung, Math. Forschungsinst., Oberwolfach, 1969)*, pp. 441–472, Bibliographisches Inst., Mannheim, 1970.

[50] M. Sipser, *Introduction to the theory of computation (third edition)*, Cengage Learning, 2012.

[51] D. F. Stanat and S. F. Weiss, A pumping theorem for regular languages, *SIGACT News* **14 (1)** (1982), 36–37.

[52] R. E. Stearns and J. Hartmanis, Regularity preserving modifications of regular expressions, *Information and Control* **6** (1963), 55–69.

[53] H. Straubing, Relational morphisms and operations on recognizable sets, *RAIRO Inf. Theor.* **15** (1981), 149–159.

[54] B. A. Trakhtenbrot and Y. M. Barzdin′, *Finite automata*, North-Holland Publishing Co., Amsterdam, 1973. Behavior and synthesis, Translated from the Russian by D. Louvish, English translation edited by E. Shamir and L. H. Landweber, Fundamental Studies in Computer Science, Vol. 1.

[55] S. Varricchio, A pumping condition for regular sets, *SIAM J. Comput.* **26**,3 (1997), 764–771.

[56] G.-Q. Zhang, Automata, Boolean matrices, and ultimate periodicity, *Inform. and Comput.* **152**,1 (1999), 138–154.

Contents