The Universal Automaton

Sylvain Lombardy[1]
Jacques Sakarovitch[2]

[1] Institut Gaspard Monge
Université de Marne-la-Vallée
5, boulevard Descartes
Champs-sur-Marne
77454 Marne-la-Vallée Cedex 2, France
Sylvain.Lombardy@univ-mlv.fr

[2] École Nationale Supérieure des Télécommunications
46, rue Barrault
75634 Paris Cedex 13, France
Jacques.Sakarovitch@enst.fr

Abstract

This paper is a survey on the universal automaton, which is an automaton canonically associated with every language. In the last forty years, many objects have been defined or studied, that are indeed closely related to the universal automaton.

We first show that every automaton that accepts a given language has a morphic image which is a subautomaton of the universal automaton of this language. This property justifies the name "universal" that we have coined for this automaton. The universal automaton of a regular language is finite and can be effectively computed in the syntactic monoid or, more efficiently, from the minimal automaton of the language. We describe the construction that leads to tight bounds on the size of the universal automaton. Another outcome of the effective construction of the universal automaton is the computation of a minimal NFA accepting a given language, or approximations of such a minimal NFA. From another point of view, the universal automaton of a language is based on the factorisations of this language, and is thus involved in the problems of factorisations and approximations of languages. Last, but not least, we show how the universal automaton gives an elegant solution to the star height problem for some classes of languages (pure-group or reversible languages).

With every language is canonically associated an automaton, called *the universal automaton of the language*, which is finite whenever the language is regular. It is large, it is complex, it is complicated to compute, but it contains, hopefully, many interesting informations on the language. In the last forty years, it has been described a number of times, more or less explicitly, more or less approximately, in relation with one or another property of the language. This is what we review here systematically.

1 A brief history of the universal automaton

The origin of the universal automaton is not completely clear. A well-publicized note [ADN92] credits Christian Carrez of what seems to be the first definition of the universal automaton in a report that remained unpublished [Car70]. The problem at stake was the computation of the, or of *a*, NFA with minimal number of states that recognizes a given regular language L. And Carrez's report states the existence of an automaton $\mathcal{U}_L$, very much in the way we do in Section 2, with the property that it contains a morphic image of any automaton which recognizes L, and thus a copy of any minimal NFA which recognizes L.

At about the same time, Kameda and Weiner tackled the same problem and, *without stating the existence of* $\mathcal{U}_L$, described a construction for a NFA recognizing L with minimal number of states [KW70], a construction which we recognize now as being similar to the construction of $\mathcal{U}_L$ we propose in Section s.con-uni-aut.

Soon afterwards, in another context, and with no connexion of any kind with the previous problem (*cf.* Section 6) Conway proposed the definition of what can be seen also as an automaton attached to L and which is again equal to $\mathcal{U}_L$ [Con71] (*cf.* Section 3.1).

Among other work related to $\mathcal{U}_L$, but without reference to the previous one, let us quote [CNP91] and [MP95]. Eventually, we got interested in the universal automaton as we discovered it contains other informations on the languages that those studied before (see Section 7.1) and we made the connexion between the different instances [LS03, LS02].

2 Creation of the universal automaton

No wonder, we first fix some notations. If X is a set, $\mathfrak{P}(X)$ denote the *power set* of X, *i.e.* the set of subsets of X. We denote by A^* the free monoid generated by a set A. Elements of A^* are *words*, the identity of A^* is the *empty word*, written 1_{A^*}. The product in A^* is denoted by concatenation and is extended by additivity to $\mathfrak{P}(A^*)$: $XY = \{uv \mid u \in X, v \in Y\}$.

An automaton $\mathcal{A}$ is a 5-tuple $\mathcal{A} = \langle Q, A, E, I, T\rangle$, where Q is a finite set of states, A is a finite set of letters, E, the set of transitions, is a subset of $Q \times A \times Q$, and I (*resp.* T), the set of initial (*resp.* terminal) states, is a subset of Q. Such an automaton $\mathcal{A}$ defines an *action*[1] $\triangleright$ of A^* on $\mathfrak{P}(Q)$,

[1] Normally, we would have denoted the action by a simple $\cdot$; but later, in Section 5, we shall need to consider an action on the right *and* an action on the left, hence a lateralized symbol which makes the reading easier. Moreover, when necessary, *i.e.* when several automata are considered at the same time, we shall even specify as a

by setting first for all $p \in Q$ and all $a \in A$

$$p \rhd a = \{q \in Q \mid (p, a, q) \in E\},$$

and then by additivity and the definition of an action for all $X \in \mathfrak{P}(Q)$

$$X \rhd 1_{A^*} = X\,, \quad X \rhd a = \bigcup_{p \in X} p \rhd a\,, \quad X \rhd wa = (X \rhd w) \rhd a.$$

The behaviour $|\mathcal{A}|$ (or the accepted language) of an automaton $\mathcal{A} = \langle Q, A, E, I, T\rangle$ is the set of words that label a path from an initial state to a terminal state, *i.e.*

$$|\mathcal{A}| = \{w \in A^* \mid \exists i \in I, \quad t \in T \quad i \xrightarrow[\mathcal{A}]{w} t\} = \{w \in A^* \mid I \rhd w \cap T \neq \varnothing\}.$$

A subset of A^* is called a *language* and a language is *regular* if it is the behaviour of some finite automaton.

Let $\mathcal{A} = \langle Q, A, E, I, T\rangle$ be an automaton over A^*. For each state p of $\mathcal{A}$, the *past of* p is the set of labels of computation which go from an initial state of $\mathcal{A}$ to p, and we write it $\mathsf{Past}_{\mathcal{A}}(p)$; *i.e.*

$$\mathsf{Past}_{\mathcal{A}}(p) = \{w \in A^* \mid \exists i \in I \quad i \xrightarrow[\mathcal{A}]{w} p\} = \{w \in A^* \mid p \in I \rhd w\}.$$

Dually, the *future of* p is the set of labels of computations that go from p to a final state of $\mathcal{A}$ and we write it $\mathsf{Fut}_{\mathcal{A}}(p)$, *i.e.*:

$$\mathsf{Fut}_{\mathcal{A}}(p) = \{w \in A^* \mid \exists t \in T \quad p \xrightarrow[\mathcal{A}]{w} t\} = \{w \in A^* \mid p \rhd w \cap T \neq \varnothing\}.$$

Likewise, for each pair of states (p, q) of $\mathcal{A}$, the transitional language of (p, q) is the set of labels of computations that go from p to q and we write it $\mathsf{Trans}_{\mathcal{A}}(p, q)$, *i.e.*:

$$\mathsf{Trans}_{\mathcal{A}}(p, q) = \{w \in A^* \mid p \xrightarrow[\mathcal{A}]{w} q\} = \{w \in A^* \mid q \in p \rhd w\}.$$

For each p, q in Q, we clearly have

$$[\mathsf{Past}_{\mathcal{A}}(q)]\,[\mathsf{Fut}_{\mathcal{A}}(q)] \subseteq |\mathcal{A}|. \qquad (*)$$

Thus, in every automaton, each state induces a set of ‘factorisations’ — which is the name we give to equations of the type $(*)$ — of the language it recognizes. The starting point of the construction is to prove the converse of this observation, namely that we can construct from the set of factorisations of a language L of A^* an automaton which accepts L.

subscript the automaton that defines the action in action: $p \underset{\mathcal{A}}{\rhd} a$.

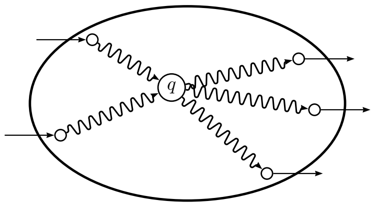


FIGURE 1. Representation of the past and future of q in $\mathcal{A}$

2.1 Factorisations of a language

In the rest of this paper, L is a language of A^*. We call *subfactorisation* of L a pair (X, Y) of languages of A^* such that $XY \subseteq L$ and *factorisation* a subfactorisation (X, Y) which is maximal for the inclusion, that is, if $X \subseteq X'$, $Y \subseteq Y'$ and $X'Y' \subseteq L$ then $X = X'$ and $Y = Y'$. We write $\mathcal{F}_L$ for the set of factorisations of L. If (X, Y) is in $\mathcal{F}_L$ then X is called a *left factor* and Y a *right factor* of L. The maximality condition on factorisations already implies that the left and right factors are in a 1-1 correspondence. The notion of quotient allows to be even more precise.

The *left quotient* (*resp.* the *right quotient*) of L by a word v is the language[2] $v^{-1}L = \{w \in A^* \mid vw \in L\}$ (*resp.* the language $Lv^{-1} = \{u \in A^* \mid uv \in L\}$).

If $WZ \subseteq L$ then Z is contained in $Y = \bigcap_{w \in W} w^{-1}L$ and thus Y is maximum such that $WY \subseteq L$. From which a series of properties are easily derived, that are worth stating for further usage, with, or without, explicit reference.

Proposition 2.1.

(i) For every (X, Y) in $\mathcal{F}_L$,

$$Y = \bigcap_{x \in X} x^{-1}L \quad \text{and} \quad X = \bigcap_{y \in Y} Ly^{-1}.$$

(ii) Conversely, any intersection of left quotients is a right factor, and any intersection of right quotients is a left factor.

(iii) If W and Z are such that $WZ \subseteq L$, then there exists (at least) one factorisation (X, Y) of L such that $W \subseteq X$ and $Z \subseteq Y$

(iv) The property '(X, Y) is a factorisation of L' induces a bijection between the left and right factors of L.

[2] Sometimes called *residual* of L.

Corollary 2.2. A language is regular if, and only if, it has a finite number of factorisations.

Remark 2.3. We write L^{t} for the *transpose* of L, that is, the set of mirror image of words in L. If (X,Y) is a factorisation of L, $(Y^{\mathrm{t}},X^{\mathrm{t}})$ is a factorisation of L^{t}. By *duality*, we unterstand the change from L to L^{t}.

2.2 Universal automaton of a language

The definition of factorisations of a language allows in turn to set up the definition we are aiming at.

Definition 2.4. The *universal automaton* $\mathcal{U}_L$ of L is defined as $\mathcal{U}_L = \langle \mathcal{F}_L, A, E^L, I^L, T^L\rangle$, where:

$$I^L = \{(X,Y)\in\mathcal{F}_L \mid 1_{A^*}\in X\}\,,\quad T^L = \{(X,Y)\in\mathcal{F}_L \mid 1_{A^*}\in Y\},$$
$$E^L = \{\big((X,Y),a,(X',Y')\big)\in\mathcal{F}_L\times A\times\mathcal{F}_L \mid XaY'\subseteq L\}.$$

From the maximality of the factorisations follows:

$$(X,Y)\in I^L \iff Y\subseteq L\,,\quad (X,Y)\in T^L \iff X\subseteq L, \tag{1.1}$$
$$\big((X,Y),a,(X',Y')\big)\in E^L \iff Xa\subseteq X' \iff aY'\subseteq Y. \tag{1.2}$$

The description of computations in the universal automaton is then a generalisation of the above equation.

Lemma 2.5. For all (X,Y) and (X',Y') in $\mathcal{F}_L$ and for every w in A^+, it holds:

$$(X,Y)\xrightarrow[\mathcal{U}_L]{w}(X',Y') \iff XwY'\subseteq L \iff Xw\subseteq X' \iff wY'\subseteq Y.$$

Proof. By induction on $|w|$. The property holds true for $|w|=1$, by definition of E^L and by (1.2).

Suppose that $(X,Y)\xrightarrow[\mathcal{U}_L]{aw}(X',Y')$; there exists then (X'',Y'') in $\mathcal{F}_L$ such that $(X,Y)\xrightarrow[\mathcal{U}_L]{a}(X'',Y'')$ and $(X'',Y'')\xrightarrow[\mathcal{U}_L]{w}(X',Y')$. We thus have $Xa\subseteq X''$ and $X''w\subseteq X'$, hence $XawY'\subseteq L$.
Conversely, $XawY' = [Xa]\,[wY']\subseteq L$ implies that there exists (X'',Y'') in $\mathcal{F}_L$ such that $Xa\subseteq X''$ and $wY'\subseteq Y''$, thus $XaY''\subseteq L$ and $X''wY'\subseteq L$, which, by induction hypothesis, gives $(X,Y)\xrightarrow[\mathcal{U}_L]{aw}(X',Y')$. Q.E.D.

A fundamental property of the universal automaton is given by the following.

Proposition 2.6. If (X,Y) is a factorisation of L, it then holds:

$$\mathsf{Past}_{\mathcal{U}_L}((X,Y)) = X \quad\text{and}\quad \mathsf{Fut}_{\mathcal{U}_L}((X,Y)) = Y.$$

Proof. The definition of T^L itself states that $\mathsf{Fut}_{\mathcal{U}_L}((X,Y))$ contains 1_{A^*} if, and only if, Y contains 1_{A^*}. Let w be a non empty word in $\mathsf{Fut}_{\mathcal{U}_L}((X,Y))$, that is, $(X,Y) \xrightarrow[\mathcal{U}_L]{w} (X',Y')$ with 1_{A^*} in Y'. By Lemma 2.5, $XwY' \subseteq L$; as 1_{A^*} is in Y', $Xw \subseteq L$ and w is in Y by maximality of Y. Therefore $\mathsf{Fut}_{\mathcal{U}_L}((X,Y)) \subseteq Y$.

Conversely, if (X,Y) is in $\mathcal{F}_L$, $XY = [XY]\,[1_{A^*}] \subseteq L$ and there exists a right factor Y', containing 1_{A^*}, such that $XYY' \subseteq L$. By Lemma 2.5 again, $Y \subseteq \mathsf{Fut}_{\mathcal{U}_L}((X,Y))$.

The other equality is obtained by duality. Q.E.D.

As $1_{A^*}L = L1_{A^*} = L$, L is both a right and a left factor, to which correspond the left factor X_s and the right factor Y_e. We call (X_s, L) the *starting factorisation*, and (L, Y_e) the *ending factorisation*.[3]

Corollary 2.7. $\mathcal{U}_L$ recognises L.

Proof. For any factorisation (X,Y) in I^L, $Y \subseteq L$ since $1_{A^*} \in X$. Then $|\mathcal{U}_L| = \bigcup_{(X,Y)\in I^L} \mathsf{Fut}_{\mathcal{U}_L}((X,Y)) = \bigcup_{(X,Y)\in I^L} Y$ is contained in L. Since $(X_s, L) \in I^L$ then $|\mathcal{U}_L| = L$. Q.E.D.

The universal automaton is canonically associated with L, like the *minimal deterministic automaton* or the *minimal co-deterministic automaton*; unlike them, it is not lateralised, that is, not oriented from left to right nor from right to left. It is a restatement of Corollary 2.2 that L is regular if, and only if, $\mathcal{U}_L$ is finite. And the universal automaton $\mathcal{U}_{L^{\mathrm{t}}}$ of L^{t} is the transpose automaton of $\mathcal{U}_L$.

Example 2.8.

(i) Let $L_1 = A^*abA^*$. The set $\mathcal{F}_{L_1} = \{u, v, w\}$ is easily computed: $u = (A^*, A^*abA^*)$, $v = (A^*aA^*, A^*bA^*)$ and $w = (A^*abA^*, A^*)$. Figure 2 shows $\mathcal{U}_{L_1}$.

(ii) Figure 2 also shows the universal automaton of $L_2 = aA^*$. This example allows us to see that a universal automaton is not necessarily trim: a factorisation $(\varnothing, A^*)$ (*resp.* $(A^*, \varnothing)$), if it exists, corresponds to a non-accessible (*resp.* a non-co-accessible) state.

2.3 Universality of the universal automaton

We begin with the definition of *morphism of automata* and some related notions that will be central to our purpose.

[3] Conway, who did not define the universal automaton as such, called them *initial* and *final* factorisation respectively, an option that is not open to us.

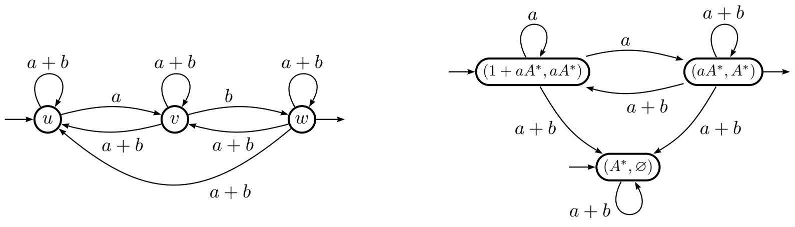


FIGURE 2. The universal automaton of L_1 (left) and of L_2 (right)

2.3.1 Morphisms, quotients and minimal automata

In the sequel, $\mathcal{A} = \langle Q, A, E, I, T\rangle$ and $\mathcal{B} = \langle R, A, F, J, U\rangle$ are two automata over A^*.

Definition 2.9. A map φ from Q into R is a *morphism of automata*, and we write $\varphi\colon \mathcal{A} \to \mathcal{B}$ if, and only if,

$$\varphi(I) \subseteq J, \quad \varphi(T) \subseteq U, \quad \text{and } \varphi(E) = \{\big(\varphi(p), a, \varphi(q)\big) \mid (p,a,q) \in E\} \subseteq F\,.$$

The morphism φ is *surjective* if $\mathcal{B} = \langle \varphi(Q), A, \varphi(E), \varphi(I), \varphi(T)\rangle$ (we also say that $\mathcal{B}$ is a *morphic image* of $\mathcal{A}$).

If $\varphi\colon \mathcal{A} \to \mathcal{B}$ is a morphism, the image of a computation in $\mathcal{A}$ is a computation in $\mathcal{B}$, with the same label, which directly implies the following.

Proposition 2.10. Let φ be a morphism from $\mathcal{A}$ into $\mathcal{B}$. Then, for every state p of $\mathcal{A}$,

$$\mathsf{Past}_{\mathcal{A}}(p) \subseteq \mathsf{Past}_{\mathcal{B}}(\varphi(p))\,, \quad \mathsf{Fut}_{\mathcal{A}}(p) \subseteq \mathsf{Fut}_{\mathcal{B}}(\varphi(p)), \tag{1.3}$$

and then

$$|\mathcal{A}| \subseteq |\mathcal{B}|. \tag{1.4}$$

The notion of morphism is not lateralised and if $\varphi\colon \mathcal{A} \to \mathcal{B}$ is a morphism then so is $\varphi\colon \mathcal{A}^{\mathrm{t}} \to \mathcal{B}^{\mathrm{t}}$. If $\varphi\colon \mathcal{A} \to \mathcal{B}$ is a surjective morphism and if moreover $|\mathcal{A}| = |\mathcal{B}|$, then any two states p and q of $\mathcal{A}$ such that $\varphi(p) = \varphi(q)$ are said to be *mergible* (in $\mathcal{A}$).

Proposition 2.11. The universal automaton $\mathcal{U}_L$ has no mergible states.

Proof. Suppose, by way of contradiction, that $\varphi\colon \mathcal{U}_L \to \mathcal{C}$ is a surjective morphism and that $|\mathcal{C}| = L$.

If $\varphi((X,Y)) = \varphi((X',Y')) = s$ the combination of Proposition 2.6 and Proposition 2.10 yields $X \cup X' \subseteq \mathsf{Past}_{\mathcal{C}}(s)$ and $Y \cup Y' \subseteq \mathsf{Fut}_{\mathcal{C}}(s)$ from which follows $(X \cup X')(Y \cup Y') \subseteq \mathsf{Past}_{\mathcal{C}}(s)\mathsf{Fut}_{\mathcal{C}}(s) \subseteq L$, impossible by the maximality of factorisations. Q.E.D.

Definition 2.12. A morphism $\varphi\colon \mathcal{A} \to \mathcal{B}$ is *Out-surjective* if

(i) for every (r, a, s) in F and every p such that $\varphi(p) = r$ there exists q such that $\varphi(q) = s$ and (p, a, q) in E;

(ii) for every p in Q, if $\varphi(p)$ is in U then p is in T.

The notion of Out-surjectivity is lateralised and $\varphi\colon \mathcal{A} \to \mathcal{B}$ is said to be *In-surjective* if $\varphi\colon \mathcal{A}^{\mathrm{t}} \to \mathcal{B}^{\mathrm{t}}$ is Out-surjective. If $\varphi\colon \mathcal{A} \to \mathcal{B}$ is both surjective and Out-surjective (*resp.* and In-surjective) $\mathcal{B}$ — and φ — is called a *quotient* (*resp.* a *co-quotient*) of $\mathcal{A}$. An easy proof by induction on the length of the computations establishes the following.

Proposition 2.13. If the automaton $\mathcal{B}$ is a quotient (*resp.* a co-quotient) of the automaton $\mathcal{A}$ then $|\mathcal{A}| = |\mathcal{B}|$.

We thus have three distinct notions of maps for automata: morphism, quotient, and co-quotient, that lead to three distinct notions of *minimality.* The minimal quotient of a (non deterministic) automaton $\mathcal{A}$ exists and is unique, canonically associated with $\mathcal{A}$ — not with $|\mathcal{A}|$ unless $\mathcal{A}$ is deterministic —, defined by a generalisation of the so-called Nerode equivalence, and computed, if necessary, by a kind of Moore algorithm. The same is true of co-quotient, up to a transposition. The notion of minimality with respect to morphism is slightly more tricky and unicity is lost.

Definition 2.14. Let $\mathcal{A}$ be an automaton over A^* that accepts a language L. We say that $\mathcal{A}$ is *m-minimal* if the following two properties hold:

(i) every proper subautomaton of the *trim* part of $\mathcal{A}$ accepts a language that is strictly contained in L;

(ii) every proper morphic image of $\mathcal{A}$ accepts a language that contains strictly L.

In other words, an automaton is m-minimal if every state is necessary –unless it is a sink or a co-sink– and no two states are mergible.

We have decided to coin that new term 'm-minimal' for there are too many 'minimal' around. A minimal quotient is not necessarily m-minimal and the sentence '*A minimal quotient is not necessarily minimal*' sounds definitively too awkward. Of course, neither a minimal quotient, nor a m-minimal automaton have a minimal number of states for accepting the same language. Some consistency is given by the following.

Proposition 2.15. The *minimal automaton* of a language L (which is the minimal quotient of any *deterministic* automaton that recognises L) is m-minimal.

Proof. Every state p of the minimal automaton of L is characterised by its future which is equal to $u^{-1}L$, for any u in its past. If p and q are two distinct states there is one, say p, whose future contains a word w which is not in the future of q. For any v in the past of q, w does not belong to $v^{-1}L$, that is, vw does not belong to L and still would be accepted in any morphic image where p and q were merged. Q.E.D.

2.3.2 Morphisms into the universal automaton

The following property of the universal automaton is the one that has been appealing to most people. We call it 'universality property' and the universal automaton gets its name from it.

Theorem 2.16. If $\mathcal{A}$ is an automaton that recognises any subset K of L, then there exists a morphism from $\mathcal{A}$ into $\mathcal{U}_L$.

This result is established via the definition of a map from $\mathcal{A}$ into $\mathcal{U}_L$, canonically associated with $\mathcal{A}$, and which is then shown to be a morphism.

Definition 2.17. Let $\mathcal{A} = \langle Q, A, E, I, T \rangle$ be an automaton that recognises a subset K of L. The *(left) canonical map* $\varphi\colon Q \to \mathcal{F}_L$ is defined by $\varphi(p) = (X_p, Y_p)$ with

$$Y_p = \{v \in A^* \mid \mathsf{Past}_{\mathcal{A}}(p)v \subseteq L\} = \bigcap_{u \in \mathsf{Past}_{\mathcal{A}}(p)} u^{-1}L. \qquad (1.5)$$

In other words, φ is defined by associating with every state p of $\mathcal{A}$ the factorisation of L with the largest possible right factor that is compatible with the *past* of p in $\mathcal{A}$.

Proof of Theorem 2.16. Let p in Q and $\varphi(p) = (X_p, Y_p)$. It follows directly from the definition that $\mathsf{Past}_{\mathcal{A}}(p) \subseteq X_p$ and $\mathsf{Fut}_{\mathcal{A}}(p) \subseteq Y_p$ from which we deduce that $\varphi(I) \subseteq I^L$ and $\varphi(T) \subseteq T^L$.

Moreover, (p, a, q) in E implies $\mathsf{Past}_{\mathcal{A}}(p)a \subseteq \mathsf{Past}_{\mathcal{A}}(q)$ from which one deduces $\mathsf{Past}_{\mathcal{A}}(p)aY_q \subseteq \mathsf{Past}_{\mathcal{A}}(q)Y_q \subseteq L$ hence $aY_q \subseteq Y_p$ and by (1.2), φ is a morphism, that will be called *(left) canonical morphism* (from $\mathcal{A}$ to $\mathcal{U}_L$). Q.E.D.

If we apply Theorem 2.16 to a m-minimal automaton $\mathcal{A}$ accepting L we get a morphism from $\mathcal{A}$ into $\mathcal{U}_L$ that has to be injective since $\mathcal{A}$ is m-minimal. We have thus proved (see an example at Figure 3):

Corollary 2.18. Every m-minimal automaton accepting L is a subautomaton of $\mathcal{U}_L$.

On the other hand, an automaton $\mathcal{A}$ accepting L and that has stricly more states than $\mathcal{U}_L$ is sent into $\mathcal{U}_L$ by a morphism which is necessarily non injective. We have thus proved:

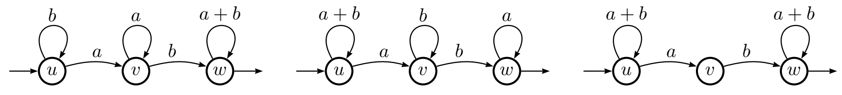


FIGURE 3. Three m-minimal subautomata of $\mathcal{U}_{L_1}$

Corollary 2.19. The universal automaton $\mathcal{U}_L$ is the largest automaton recognizing L without merging states.

Proposition 2.20. The universal automaton $\mathcal{U}_L$ is minimal for the universality property.

Proof. Suppose $\mathcal{C}$ has the universality property (with respect to L). As $\mathcal{U}_L$ accepts L, there should be a morphism from $\mathcal{U}_L$ into $\mathcal{C}$; as $\mathcal{U}_L$ has no merging states, this morphism should be injective: $\mathcal{C}$ has at least as many states as $\mathcal{U}_L$. Q.E.D.

3 Exploration of the universal automaton

The universal automaton we have just defined may be seen in different ways, from different perspective, bringing to light other characteristics and properties of this unique and canonical object. We consider here three of them. The first one is Conway's method, that yields the 'fatest' version of the universal automaton. The second one follows a universal algebra track that eventually makes easy and natural a geometric description of factorisations that was presented by Courcelle, Niwinski and Podelski ([CNP91]). The third one, due to Lombardy [Lom01] produces the most 'emaciated' version, an automaton where only the minimal information is kept and where an interesting and hidden structure is thus discovered, especially in the case of pure group languages.

3.1 The factor matrix

We keep the previous notation: $\mathcal{U}_L = \langle \mathcal{F}_L, A, E^L, I^L, T^L \rangle$ is the universal automaton of the language L of A^*. Automata are matrices (and vectors); this is the way we look at them in this section. As we are interested in matrices (and vectors) of dimension $\mathcal{F}_L$, we use throughout the section the following notation: if M is a square matrix of dimension $\mathcal{F}_L$ and for brevity, we write $M_{X,Y'}$ instead of $M_{(X,Y),(X',Y')}$ for the entry at row (X, Y) and column (X', Y'), for all (X, Y) and (X', Y') in $\mathcal{F}_L$. For a row-vector (*resp.* a column-vector) V we write V_Y (*resp.* V_X) instead of $V_{(X,Y)}$. Proposition 2.1 (iv) legitimates this shorthand.

A first example is E^L itself, viewed as a matrix with entries in $\mathfrak{P}(A^*)$ (indeed in $\mathfrak{P}(A)$):

$$E^L_{X,Y'} = \{a \in A \mid XaY' \subseteq L\}$$

for all factorisations (X,Y) and (X',Y') in $\mathcal{F}_L$. On the other hand, the left factors are naturally ordered by inclusion, an order that carries over on $\mathcal{F}_L$:

$$(X,Y) \leqslant (X',Y') \iff X \subseteq X' \iff Y' \subseteq Y.$$

As any relation on $\mathcal{F}_L$, this order is described by a Boolean matrix C^L:

$$\forall (X,Y),(X',Y') \in \mathcal{F}_L \qquad C^L_{X,Y'} = 1 \iff X \subseteq X' \iff XY' \subseteq L;$$

and since C^L is the matrix of a reflexive and transitive relation, it holds:

$$(C^L)^* = C^L. \tag{1.6}$$

The characterisation of E^L by Equation (1.2) yields that $C^L_{X,Y'} = 1$ implies, for all (X'',Y'') in $\mathcal{F}_L$, $E^L_{X',Y''} \subseteq E^L_{X,Y''}$ and $E^L_{X'',Y} \subseteq E^L_{X'',Y'}$, which means:

$$C^L \cdot E^L = E^L \cdot C^L = E^L. \tag{1.7}$$

Definition 3.1. The *factor matrix* of a language L is the matrix F^L of dimension $\mathcal{F}_L$ with entries in $\mathfrak{P}(A^*)$ defined by:

$$F^L_{X,Y'} = \{w \in A^* \mid XwY' \subseteq L\}$$

for all factorisations (X,Y) and (X',Y') in $\mathcal{F}_L$. Every entry of F^L is called *a factor of* L.

By definition, $F^L_{X,Y'}$ is the maximal Z such that $XZY' \subseteq L$. By definition[4] also, $F^L \cap \{1_{A^*}\} = C^L$ and $F^L \cap \{A\} = E^L$. Lemma 2.5 states exactly that

$$F^L \cap \{A^+\} = (E^L)^+ \quad \text{and thus} \quad F^L = C^L + (E^L)^+ = C^L + (E^L)^*.$$

Classical formulas for the star of a sum, together with (1.6) and (1.7) yields:

Proposition 3.2. $F^L = (C^L + E^L)^*$.

From which one deduces:

Corollary 3.3. $F^L = (F^L)^*$.

A direct consequence of which is:

$$\forall (X,Y),(X',Y'),(X'',Y'') \in \mathcal{F}_L \qquad F^L_{X,Y'} F^L_{X',Y''} \subseteq F^L_{X,Y''}. \tag{1.8}$$

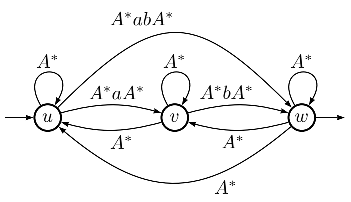
Conversely, we have:

[4] It should be obvious that $F^L \cap K$ is the matrix of dimension $\mathcal{F}_L$ obtained by taking the intersection of every entry of F^L with K.

Lemma 3.4. If $W, Z \subseteq A^*$ and (X,Y), (X',Y') in $\mathcal{F}_L$ are such that $WZ \subseteq F^L_{X,Y'}$ then there exists (X'',Y'') in $\mathcal{F}_L$ such that $W \subseteq F^L_{X,Y''}$ and $Z \subseteq F^L_{X'',Y'}$.

Proof. If $WZ \subseteq F^L_{X,Y'}$ then $XWZY' \subseteq L$ and there exists a factorisation (X'',Y'') that dominates the subfactorisation (XW, ZY') of L. The inclusions $XW \subseteq X''$ and $ZY' \subseteq Y''$ yield the conclusion. Q.E.D.

A matrix, together with initial and final vectors, is an automaton and one can see $\langle \mathcal{F}_L, A, F^L, I^L, T^L \rangle$ as a generalised automaton where the transitions are labelled by the factors of L, instead of by letters. Figure 4 shows the factor matrix of the languages L_1 and L_2 of Example 2.8 in this way.



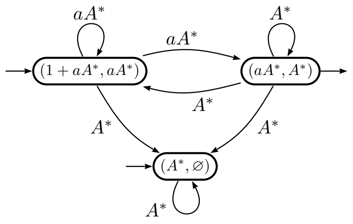


FIGURE 4. The factor matrix of L_1 (left) and of L_2 (right)

The starting and ending factorisations play a special role in the factor matrix. Since $X_sL = LY_e = L$, we have $X_sLY_e = L$ where L is obviously maximal: $F^L_{X_s,Y_e} = L$.

For every (X,Y) in $\mathcal{F}_L$, $F^L_{X_s,Y}$ is maximal in $X_sF^L_{X_s,Y}Y \subseteq L$ thus in the factorisation $(F^L_{X_s,Y}, Y)$ of $Y_s = L$, hence $F^L_{X_s,Y} = X$ and dually $F^L_{X,Y_e} = Y$.

3.2 The syntactic nature of the universal automaton

All that has been done so far for *languages*, that is, subsets of a free monoid, could have easily been done as well for subsets in any monoid: the freeness of the base monoid A^* was not involved, at the very most the generators of A^* were considered but this also could have been bypassed, especially with the help of the factor matrix. If M is a monoid and K a subset of M, a *subfactorisation* of K is a pair (X,Y) of subsets of M such that $XY \subseteq K$ and a *factorisation* is a subfactorisation (X,Y) that is maximal for the inclusion, that is, if $X \subseteq X'$, $Y \subseteq Y'$ and $X'Y' \subseteq K$ then $X = X'$ and $Y = Y'$. We write $\mathcal{F}_K$ for the set of factorisations of K. If (X,Y) is in $\mathcal{F}_K$ then X is called a *left factor* and Y a *right factor* of K. And so on.

On the other hand, the study of regular languages relies heavily on the notion of morphisms (of monoids) and that of syntactic monoid (of a language). A language L of A^* is said to be *recognised by a morphism* α,

$\alpha\colon A^* \to N$, if $\alpha^{-1}(\alpha(L)) = L$ or, which is the same, if L is a union of classes for the map equivalence of α, that is a congruence of A^*. The same could be said of a subset K, replacing L, of a monoid M, replacing A^*. The quotient of A^* by the coarsest congruence that saturates L is the *syntactic monoid* of L, denoted $\mathrm{Synt}(L)$. A language of A^*, a subset of a monoid M, is said to be *recognisable* if it is recognised by a morphism into a *finite* monoid, or, which is the same, if *its syntactic monoid is finite*.

We like to say that a property is '*syntactic*' if true for a language L, or a subset K, recognised by a *surjective* morphism α, it is true for $\alpha(L)$ or $\alpha(K)$. The factorisations, the universal automaton, are 'syntactic objects', as shown by the following.

Proposition 3.5. *Let L be a language of A^*, recognised by a surjective morphism α.*

(i) *Any factor of L is recognised by α.*

(ii) *If (X, Y) is a factorisation of L, $(\alpha(X), \alpha(Y))$ is a factorisation of $\alpha(L)$.*

(iii) *α establishes a bijection between the factorisations of L and those of $\alpha(L)$.*

Proof. Let (X, Y) be a factorisation of L: $XY \subseteq L$. Then $\alpha(X)\alpha(Y) \subseteq \alpha(L)$ and $(\alpha(X), \alpha(Y))$ is a *sub*factorisation of $\alpha(L)$ which we suppose dominated by a factorisation (U, V). From $\alpha(X) \subseteq U$ and $\alpha(Y) \subseteq V$ we deduce $X \subseteq \alpha^{-1}(\alpha(X)) \subseteq \alpha^{-1}(U)$ and $Y \subseteq \alpha^{-1}(\alpha(Y)) \subseteq \alpha^{-1}(V)$ and $\alpha^{-1}(U)\alpha^{-1}(V) \subseteq \alpha^{-1}(\alpha(L)) = L$. Since (X, Y) is a factorisation, $X = \alpha^{-1}(U)$ and $Y = \alpha^{-1}(V)$.

This shows at the same time that, (i) $X = \alpha^{-1}(\alpha(X))$ and $Y = \alpha^{-1}(\alpha(Y))$, and (ii) $\alpha(X) = U$ and $\alpha(Y) = V$: $(\alpha(X), \alpha(Y))$ is a factorisation of $\alpha(L)$.

For the same reason, $\alpha^{-1}(\alpha(F^L_{X,Y})) = F^L_{X,Y}$ for all factorisations (X, Y) and (X', Y') of L.

Conversely, let (U, V) be a factorisation of $\alpha(L)$; then $(\alpha^{-1}(U), \alpha^{-1}(V))$ is a *sub*factorisation of $\alpha^{-1}(\alpha(L)) = L$ which we suppose dominated by a factorisation (X, Y). Since (U, V) is a factorisation, neither $U \subseteq \alpha(X)$ or $V \subseteq \alpha(Y)$ may be strict inclusion and $(\alpha^{-1}(U), \alpha^{-1}(V))$ is a factorisation. Q.E.D.

Example 3.6. The syntactic monoid of L_1 is $M_1 = \{1_{M_1}, x, y, t, z\}$ defined by the relations $xx = x$, $yy = y$, $yx = t$, and $xy = xt = ty = z$. The syntactic morphism $\alpha\colon A^* \to M_1$ sends a onto x and b onto y. Then $\alpha(L_1) = z$ and the factorisations of z in M_1 are $(\{z\}, M_1)$, $(M_1, \{z\})$, and $(\{x, t, z\}, \{y, t, z\})$.

Let M be any monoid and let $\psi_M \colon M \times M \to M$ be the map defined by $\psi_M((u,v)) = uv$. (This map is not a morphism unless M is commutative.) It can be seen as the *multiplication table* of M: in a matrix T of size $M \times M$, each element m appears as the entry (u,v) of T, for all (u,v) in $\psi_M^{-1}(m)$.

A factorisation of a subset K of M appears as a maximal rectangle in the subset $\psi_M^{-1}(K)$ and this point of view, possible in the general case, is without doubt the simplest *when M is finite*.

Example 3.7. The table of the monoid M_1, cleverly laid out (we have inverted the order of the elements x and y by row and column) is shown in Figure 5. The factorisations of the subset $\{z\}$ are made clearly visible with rectangles. The figure also show *the factor matrix* of $\{z\}$, under the form of an automaton labelled with subsets.

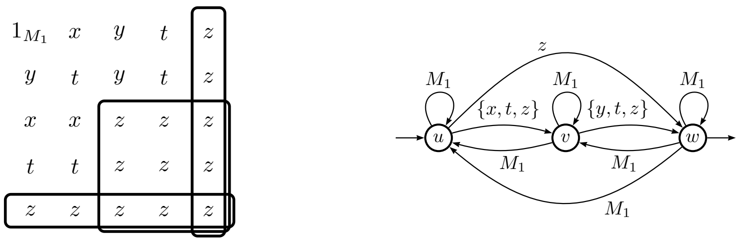


FIGURE 5. Factorisations and factor matrix of $\{z\}$ in M_1

Example 3.8. We consider the (additive) monoid $\mathbb{Z}/3\mathbb{Z}$. Figure 6 shows the factorisations of the subset $\{1,2\}$ and its universal automaton. The states are labelled by the left factor of the corresponding factorisation, and the label of the transitions is always the generator 1. This automaton is thus (up the addition of the transitions having the label b) the universal automaton of the language $L_3 = \{w \in \{a,b\}^* \mid |w|_a \not\equiv |w|_b \mod 3\}$ since $\mathrm{Synt}(L_3) = \mathbb{Z}/3\mathbb{Z}$ and the image of L_3 there is $\{1,2\}$.

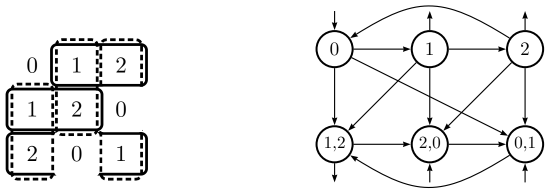


FIGURE 6. Factorisations and universal automaton of $\{1,2\}$ in $\mathbb{Z}/3\mathbb{Z}$

Proposition 3.5 holds for a recognisable subset K of any monoid M. This implies that such a subset is accepted by an automaton with a finite number of states, whose transition matrix is the factor matrix F^L. To make this automaton really finite, the monoid is required to be generated by a finite set G, and the transitions of the universal automaton of K are then given by $F^L \cap G$. This automaton accepts K, and more precisely, for every element x of K, for every factorisation $x = x_1 \ldots x_n$ of x over G, the sequence $(x_1, \ldots, x_n)$ is the label of (at least) one computation of the universal automaton. This is the reason why the universal automaton relates to recognisable subsets and not to rational subsets.

Actually, a subset K of a monoid is rational if and only if there exists a finite automaton such that for every element of K, there is at least a factorisation of this element that labels a computation, whereas a subset K of a finitely generated monoid is recognisable if and only if there exists a finite automaton such that for every element of K, *every* factorisation of this element (*w.r.t.* the generators) is accepted. [5]

3.3 The écorché of the universal automaton

The order on the factorisations of L considered above (and induced by the inclusion order on the left factors) can be used to give a simplified description of $\mathcal{U}_L$. Indeed, if $(X,Y) \xrightarrow[\mathcal{U}_L]{a} (X',Y')$ then,

$$\begin{aligned} &\forall (X_1,Y_1) \in \mathcal{F}_L \qquad (X_1,Y_1) \leqslant (X,Y) \Longrightarrow (X_1,Y_1) \xrightarrow[\mathcal{U}_L]{a} (X',Y'), \\ &\forall (X_2,Y_2) \in \mathcal{F}_L \qquad (X',Y') \leqslant (X_2,Y_2) \Longrightarrow (X,Y) \xrightarrow[\mathcal{U}_L]{a} (X_2,Y_2). \end{aligned}$$

Moreover, if (X,Y) is initial, any larger factorisation is initial and, dually, if it is final, any smaller factorisation is final. The order on factorisations is described by the matrix C^L and what we have just observed is a rewording of (1.7): $C^L \cdot E^L = E^L \cdot C^L = E^L$ and of $I^L = C^L \cdot I^L$ and $T^L = T^L \cdot C^L$.

A solution of $XaY' \subseteq L$ is *maximal* if

$$X_1 a Y_2 \subseteq L \quad \text{and} \quad X \subseteq X_1\,, \quad Y' \subseteq Y_2 \quad \Longrightarrow \quad X = X_1 \quad \text{and} \quad Y' = Y_2$$

for all factorisations (X_1,Y_1) and (X_2,Y_2) in $\mathcal{F}_L$. That is, (X,Y) is as large as possible, and (X',Y') as small as possible such that $(X,Y) \xrightarrow[\mathcal{U}_L]{a} (X',Y')$. We then define the matrix H^L of dimension $\mathcal{F}_L$ and with entries in $\mathfrak{P}(A)$ by

$$a \in H^L_{X,Y'} \quad \Longrightarrow \quad XaY' \subseteq L \quad \text{is maximal.}$$

[5] By virtue of Kleene Theorem, in the free monoid, recognisable subsets and rational subsets are the same: they are regular languages.

On the other hand, we note that the *starting factorisation* (X_s, L) is the smallest factorisation that is initial in $\mathcal{U}_L$ and, dually, the *ending factorisation* (L, Y_e) is the largest factorisation that is final. All these observation amount to the following.

Proposition 3.9.

(i) H^L is the minimal matrix such that $E^L = C^L \cdot H^L \cdot C^L$;

(ii) I^L is the X_s-th row of C^L;

(iii) T^L is the Y_e-th column of C^L.

Further economy in the description consists in considering the "maximal" solutions of $XY' \subseteq L$ that are not in $\mathcal{F}_L$ and in defining the Boolean matrix D^L by:

$$D^L_{X,Y'} = 1 \quad \Longleftrightarrow \quad (X,Y) \in \max\{(X'',Y'') \in \mathcal{F}_L \mid X \subset X'\}.$$

That is, $D^L_{X,Y'}$ is the matrix of the Hasse diagram of the order on factorisations.

This definition directly yields

Proposition 3.10. D^L is the minimal matrix such that $C^L = (D^L)^*$.

Definition 3.11. We call *écorché* of $\mathcal{U}_L$ the automaton:
$\mathcal{E}_L = \langle \mathcal{F}_L, A, D^L \cup H^L, \{(X_s, L)\}, \{(L, Y_e)\} \rangle$.

The automaton $\mathcal{U}_L$ is then obtained from $\mathcal{E}_L$ by backward *and* forward closure of the spontaneous transitions. In the sequel, we rather draw écorchés instead of universal automata, because they have less transitions and it is often easier to understand the structure of the universal automaton on the écorché.

Example 3.12. The factorisations of $L_1 = A^*abA^*$ are totally ordered

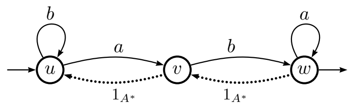
$$u = (A^*, L_1) \geqslant v = (A^*aA^*, A^*bA^*) \geqslant w = (L_1, A^*),$$

and so are the factorisations of $L_2 = aA^*$:

$$(aA^*, A^*) \leqslant (1 + aA^*, aA^*) \leqslant (A^*, \varnothing).$$

Figure 7 shows the écorché of the universal automata of these two languages.

In the case of pure-group languages, that is, languages whose syntactic monoid is a group, the écorché of the universal automaton has a very special form. The states of the strongly connected components are the pairwise uncomparable factorisations. The non spontaneous transitions, that is,



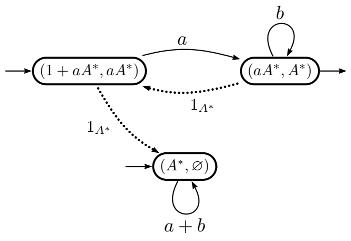


FIGURE 7. The écorché of $\mathcal{U}_{L_1}$ (left) and $\mathcal{U}_{L_2}$ (right)

the transitions described by the matrix H^L, are all the transitions in these strongly connected components whereas the spontaneous transitions put an order on the strongly connected components and the écorché is thus decomposed into levels. Figure 8 shows the écorché of $\mathcal{U}_{L_3}$. A more complicated écorché for a pure group language is shown at Figure 15, where the levels appear even more clearly. We shall characterise them at Subsection 7.2.

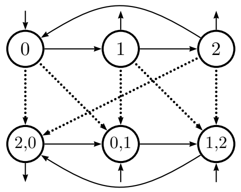


FIGURE 8. Ecorché of the universal automaton of $\{1, 2\}$ in $\mathbb{Z}/3\mathbb{Z}$

4 Construction of the universal automaton

The universal automaton has been defined, and then described. From what we have already seen, it follows immediately that the universal automaton of a regular language is effectively computable. We present now an algorithm [Lom02], somehow optimal, which performs the task. From this construction of $\mathcal{U}_L$ we then derive an effective description of the *(left) canonical morphism* from any automaton $\mathcal{B}$ which accepts L into $\mathcal{U}_L$. An example of a method for finding a small NFA accepting a given language is described in the last subsection.

4.1 Computation of the factorisations

As above, let L be a language of A^*. The key for the construction of $\mathcal{U}_L$ is the computation of the factorisations of L. From Proposition 2.1, every right factor of a language is an intersection of left quotients of this language. As the quotients of the languages are the futures of the states of any deterministic automaton $\mathcal{A}$ that accepts the language, for every factorization (X_i, Y_i) of the language L, there exists a subset P of states of $\mathcal{A}$ such that $Y_i = \bigcap_{p \in P} \mathsf{Fut}_{\mathcal{A}}(p)$. But this subset may be not unique. The set of subsets P such that the intersection of the futures of states in P is equal to Y_i is closed under union, thus there exists a unique maximal P_i such that $Y_i = \bigcap_{p \in P_i} \mathsf{Fut}_{\mathcal{A}}(p)$. To get an efficient representation of factorisations, we have to compute these maximal subsets corresponding to factorisations.

Let $\mathcal{A} = \langle Q, A, \delta, i, T \rangle$ be a *complete accessible deterministic* automaton that accepts L. Let $\mathcal{Q}_{\mathcal{A}}$ be the set of states of $\mathcal{A}_{\mathsf{cod}}$, the co-determinisation of $\mathcal{A}$; $\mathcal{Q}_{\mathcal{A}}$ is a subset of $\mathfrak{P}(Q)$, *i.e.* an element of $\mathfrak{P}(\mathfrak{P}(Q))$. We denote by $\mathcal{I}_{\mathcal{A}}$ the closure under intersection of $\mathcal{Q}_{\mathcal{A}}$. Notice that $\mathcal{I}_{\mathcal{A}}$ always contains Q itself (as the intersection of an *empty* set of elements of $\mathcal{Q}_{\mathcal{A}}$).

Theorem 4.1. The mapping $\psi_{\mathcal{A}}$ from $\mathcal{I}_{\mathcal{A}}$ into $\mathcal{F}_L$ defined by:

$$\begin{aligned} \psi_{\mathcal{A}} \colon \mathcal{I}_{\mathcal{A}} &\longrightarrow \mathcal{F}_L \\ P &\longmapsto (X, Y), \text{ with } Y = \bigcap_{p \in P} \mathsf{Fut}_{\mathcal{A}}(p) \end{aligned}$$

is a bijection.

Proof. As every intersection of left quotient is a right factor of the language, this mapping is well defined. In order to prove that this is a bijection, we prove that

$$\begin{aligned} \mathcal{F}_L &\longrightarrow \mathfrak{P}(Q) \\ (X, Y) &\longmapsto \{p \mid Y \subseteq \mathsf{Fut}_{\mathcal{A}}(p)\} \end{aligned}$$

is a mapping from $\mathcal{F}_L$ onto $\mathcal{I}_{\mathcal{A}}$. Let (X, Y) be a factorisation and $P = \{p \mid Y \subseteq \mathsf{Fut}_{\mathcal{A}}(p)\}$. Let $\mathcal{A}_{\mathsf{cod}} = \langle \mathcal{Q}_{\mathcal{A}}, A, H, J, t \rangle$ be the co-determinisation of $\mathcal{A}$ and let R be the set of states of $\mathcal{A}_{\mathsf{cod}}$ that contain P.

By construction of the co-determinisation, for every state s in $\mathcal{A}_{\mathsf{cod}}$ and for every word u in $\mathsf{Fut}_{\mathcal{A}_{\mathsf{cod}}}(s)$, it holds: $s = \{p \mid u \in \mathsf{Fut}_{\mathcal{A}}(p)\}$. Hence R is the set of states of $\mathcal{A}_{\mathsf{cod}}$ whose future has a non empty intersection with Y. Moreover, $Y = \bigcup_{s \in R} \mathsf{Fut}_{\mathcal{A}_{\mathsf{cod}}}(s)$. Hence, a state p of $\mathcal{A}$ belongs to every state of R if and only if its future contains Y. Thus, $P = \bigcap_{s \in R} s \in \mathcal{I}_{\mathcal{A}}$. Q.E.D.

Remark 4.2. This construction can take any deterministic automaton as input and give the same result. Indeed, when a deterministic automaton is co-determinised, states that are Nerode-equivalent (*i.e.* that would be merged by a minimisation algorithm) appear exactly in the same states of the co-determinisation. They become indissociable and the set $\mathcal{I}_{\mathcal{A}}$ actually does not depend on the input, but only on the language L.

Remark 4.3. The order on factorisations is realised on $\mathcal{I}_{\mathcal{A}}$ by the inclusion order.

Proposition 4.4. Let $\mathcal{A} = \langle Q, A, \delta, i, T\rangle$ be a complete deterministic automaton. Let P in $\mathcal{I}_{\mathcal{A}}$ and $(X, Y) = \psi_{\mathcal{A}}(P)$. Then

$$X = \bigcup_{p \in P} \mathsf{Past}_{\mathcal{A}}(p) \qquad \text{and} \qquad P = i \triangleright X.$$

Proof. Let (X, Y) be a factorisation; $X = \{u \mid uY \subseteq L\} = \{u \mid Y \subseteq u^{-1}L\}$. For every word u in X, let $p = i \triangleright u$; as $\mathcal{A}$ is deterministic, $u^{-1}L = \mathsf{Fut}_{\mathcal{A}}(p)$. Hence, p is in P; therefore, $X \subseteq \bigcup_{p\in P} \mathsf{Past}_{\mathcal{A}}(p)$. Conversely, let v be a word in the past of some state p in P. It holds $v\mathsf{Fut}_{\mathcal{A}}(p) \subseteq L$ and $Y \subseteq \mathsf{Fut}_{\mathcal{A}}(p)$, hence v is in X. Q.E.D.

We have thus characterized the factorisations of the language, that is the states of the universal automaton. We can now give a construction for the universal automaton.

Proposition 4.5. Let $\mathcal{A} = \langle Q, A, \delta, i, T\rangle$ be a complete deterministic automaton that accepts L. The automaton $\langle \mathcal{I}_{\mathcal{A}}, A, D, J, U\rangle$ defined by:

$$D = \{(P, a, S) \in \mathcal{I}_{\mathcal{A}} \times A \times \mathcal{I}_{\mathcal{A}} \mid P \triangleright a \subseteq S\}\,, \tag{1.9}$$

$$J = \{P \in \mathcal{I}_{\mathcal{A}} \mid i \in P\}\,, \quad U = \{P \in \mathcal{I}_{\mathcal{A}} \mid P \subseteq T\}, \tag{1.10}$$

is isomorphic to the universal automaton of L: $\mathcal{U}_L = \langle \mathcal{F}_L, A, E^L, I^L, T^L\rangle$.

Proof. Theorem 4.1 defines a bijection from $\mathcal{I}_{\mathcal{A}}$ onto $\mathcal{F}_L$. We have to check that the definitions of D, J and U correspond to E^L, I^L and T^L of Definition 2.4. Let (X_P, Y_P) and (X_S, Y_S) the factorisations corresponding to P and S:

$$Y_P = \bigcap_{p\in P} \mathsf{Fut}_{\mathcal{A}}(p), \qquad Y_S = \bigcap_{p \in S} \mathsf{Fut}_{\mathcal{A}}(p).$$

We have

$$\begin{aligned} P \triangleright a \subseteq S \quad &\Longleftrightarrow \quad Y_S \subseteq \bigcap_{p \in P\triangleright a} \mathsf{Fut}_{\mathcal{A}}(p) \\ &\Longleftrightarrow \quad aY_S \subseteq \bigcap_{p\in P} \mathsf{Fut}_{\mathcal{A}}(p) = Y_P \quad \Longleftrightarrow \quad X_P a Y_S \subseteq L. \end{aligned}$$

$$\begin{aligned} \psi_{\mathcal{A}}(J) =& \{(X, Y) \in \mathcal{F}_L \mid Y \subseteq L\} \\ =& \{(X, Y) \in \mathcal{F}_L \mid 1_{A^*} \in X\}. \end{aligned}$$

$$\psi_{\mathcal{A}}(U) = \{(X,Y) \in \mathcal{F}_L \mid 1_{A^*} \in Y\}.$$

Q.E.D.

Remark 4.6. Once $\mathcal{I}_{\mathcal{A}}$ is computed, the construction of $\mathcal{U}_L$ goes as follow: $\mathcal{I}_{\mathcal{A}}$ is the set of states; for every P in $\mathcal{I}_{\mathcal{A}}$, if i is in P, make P initial, if P is a subset of T, make P final; for every letter a, compute $P \triangleright a$, and for every R in $\mathcal{I}_{\mathcal{A}}$ that contains $P \triangleright a$, add a transition (P, a, R).

4.2 Computation of the canonical morphism

If the universal automaton is computed from a complete deterministic accessible automaton $\mathcal{A}$, the left canonical morphism from any equivalent automaton $\mathcal{B}$ into the universal automaton can be computed in polynomial time.

Let $\mathcal{P}$ be the accessible part of the product of $\mathcal{A}$ by $\mathcal{B}$. Every state of $\mathcal{P}$ is a pair (p,q) of a state of $\mathcal{A}$ and a state of $\mathcal{B}$. Let R_q be the set of states p of $\mathcal{A}$ such that (p,q) is a state of $\mathcal{P}$. We define an application $\varphi_{\mathcal{B}}$ from the states of $\mathcal{B}$ into $\mathcal{I}_{\mathcal{A}}$: $\varphi_{\mathcal{B}}(q)$ is the smallest element of $\mathcal{I}_{\mathcal{A}}$ which contains R_q.

Proposition 4.7. The morphism from $\mathcal{B}$ into $\mathcal{U}_L$ induced by $\varphi_{\mathcal{B}}$ is the left canonical morphism.

Proof. Let r be a state of $\mathcal{B}$. It holds $R_q = i \underset{\mathcal{A}}{\triangleright} \mathsf{Past}_{\mathcal{B}}(q)$. Let $Y = \bigcap_{p \in R_q} \mathsf{Fut}_{\mathcal{A}}(p)$. As $\mathcal{A}$ is deterministic, the futures of its states are quotients and thus Y is a right factor. We show that this is the largest right factor such that $[\mathsf{Past}_{\mathcal{B}}(q)]\,[Y] \subseteq L$.

$$\mathsf{Past}_{\mathcal{B}}(q) = \bigcup_{p \in R_q} \mathsf{Past}_{\mathcal{P}}((p,q)) \subseteq \bigcup_{p \in R_q} \mathsf{Past}_{\mathcal{A}}(p)$$

As $\left[\bigcup_{p \in R_q} \mathsf{Past}_{\mathcal{A}}(p)\right] \left[\bigcap_{p \in R_q} \mathsf{Fut}_{\mathcal{A}}(p)\right] \subseteq L$, $[\mathsf{Past}_{\mathcal{B}}(q)]\,[Y] \subseteq L$. Let v be a word which is not in Y. There exists a state p in R_q such that v is not in $\mathsf{Fut}_{\mathcal{A}}(p)$ and there exists a word u in $\mathsf{Past}_{\mathcal{B}}(q)$ such that $p = i \underset{\mathcal{A}}{\triangleright} u$. We have $v \notin u^{-1}L$ and $uv \notin L$. This proves that Y is maximal.

We show now that Y is the right factor of $\psi_{\mathcal{A}}(\varphi_{\mathcal{B}}(q))$. As $\varphi_{\mathcal{B}}(q)$ is the smallest element of $\mathcal{I}_{\mathcal{A}}$ which contains R_q, they correspond to the same right factor, *i.e.* $Y = \bigcap_{p \in \varphi_{\mathcal{B}}(q)} \mathsf{Fut}_{\mathcal{A}}(p)$. Q.E.D.

4.3 Searching for NFA of minimal size

It is known that the computation of a NFA with minimal size from the minimal automaton of a language is a PSPACE-complete problem [JR93]. However, the universal automaton is a good framework to explain exact algorithms or to describe heuristics that give approximate solutions.

First, the universal automaton of a language contains any equivalent NFA with minimal size, since the canonical morphism from this NFA into the universal automaton is injective.

Then an exact algorithm would consist in enumerating all subautomata of the universal automaton (starting with the smallest) and testing if they accept every word of the language.

This fact is the base of many heuristics. There exist several conditions on subautomata of the universal automaton built as in Proposition 4.5. Each of these conditions is either necessary or sufficient for the subautomaton accepts the language. In [Pol05], Polák has made a comparison between a large set of these conditions. They all give tractable algorithms that compute NFA accepting the language, hopefully small, but not necessarily of minimal size.

The first authors that give such a condition are Kameda and Weiner in [KW70]. They build a table, whose rows are indexed by the states of the minimal automaton and the columns by the states of its co-determinisation, and read factorisations in this table. They define a property of *cover*, that guarantees that a set of factorisations corresponds to an automaton (actually a subautomaton of the universal automaton, even if they do not define it), that accepts the language.

Along the same line of work, Matz and Potthoff [MP95] have defined another automaton, which they call *fundamental automaton* and which contains, strictly in some cases, the universal automaton. They then give a condition that guarantees that a subautomaton of the fundamental automaton accepts the language. We present here a condition that is inspired by this one and which is an example of a heuristic that search for small NFA.

Proposition 4.8. Let $\mathcal{A} = \langle Q, A, \delta, i, T \rangle$ be a deterministic complete automaton and let $\mathcal{U}_L$ be the universal automaton built from $\mathcal{A}$. Let R be a subset of $\mathcal{I}_{\mathcal{A}}$ such that:

(i) $\bigcup_{P \in R, P \subseteq T} P = T$;

(ii) for every P in R, for every letter a, for every q in Q such that $q \triangleright a = p$, there exists S in R, such that q is in S and $S \triangleright a \subseteq P$.

Then the subautomaton of $\mathcal{U}_L$ with set of states R accepts the language.

Proof. Let u be a word of L. Let $p_0 = i, p_1, \ldots, p_k$ the states of the computation in $\mathcal{A}$ labeled by u. The state p_k is final, hence there exists P_k final in R that contains p_k. If there exists P_i that contains p_i, as $p_i = p_{i-1} \triangleright u_i$, there exists P_{i-1} in R that contains p_{i-1} such that there is a transition from P_{i-1} to P_i labeled by a. Hence, by induction, the word u is accepted by the subautomaton. Q.E.D.

5 Size of the universal automaton

It follows from the construction of the universal automaton that $\mathcal{U}_L$ has at most 2^n states if the size of the minimal deterministic automaton is n. The computation for the language $\{w \mid |w| \neq 0 \mod n\}$ shows that this bound is tight (*cf.* Section 7.2 below and also [GKP06]).

As the determinisation of a non deterministic n-state automaton may give at most a 2^n-state automaton, we immediately get a 2^{2^n} upper bound for the size of the universal automaton with respect to the minimal non deterministic automaton that accepts the language.

This bound is not tight, for the worst cases in determinisation and in the construction of the universal automaton cannot occur in a row. We give here the proof that the tight bound is given by the Dedekind numbers and also that, in the case of a unary alphabet, the conjunction of worst cases may occur, but the determinisation does not yield a 2^n blow-up then.

5.1 Bounds for the universal automaton

The n-th Dedekind number $D(n)$ is defined as the number of monotonous Boolean functions with n variables. Since such a function is characterised by a Boolean expression in disjunctive normal form whose clauses are pairwise uncomparable, it is also the number of antichains of $\mathfrak{P}([1;n])$ (ordered by inclusion).

Theorem 5.1 (Lombardy, [Lom07])**.** Let $\mathcal{A}$ be an NFA with n states that accepts a language L. Then:

(i) $\mathcal{U}_L$ has at most $D(n)$ states;

(ii) the trim part of $\mathcal{U}_L$ has at most $D(n) - 2$ states.

For every integer n, there exist automata with n states for which these bounds are reached.

Remark 5.2. There is no closed form expression for $D(n)$, and its exact value is only known for n smaller than 9 (*cf.* [Wie91]). However, Korshunov [Kor81] has given an approximate expression of $D(n)$. For instance, if n is even,

$$D(n) \sim 2^{\binom{n}{n/2}} \exp\left(\binom{n}{n/2-1}(2^{-n/2} + n^2 2^{-n-5} - n 2^{-n-4})\right).$$

Figure 9 gives a visual comparison between $D(n)$ and the double exponential function $n \mapsto 2^{2^n}$.

Definition 5.3. Let O be an ordered set. An *upset* $\mathcal{V}$ of O is an upperly closed subset of O:

$$\forall x \in \mathcal{V},\ \forall y \in O,\ x \leqslant y \Longrightarrow y \in \mathcal{V}. \tag{1.11}$$

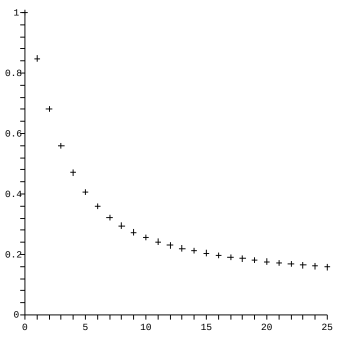


FIGURE 9. The graph of $\frac{\log_2 D(n)}{2^n}$

Notice that an upset may be empty and may also be equal to O itself. If Q is a set, $\mathfrak{P}(Q)$ or every subset of $\mathfrak{P}(Q)$ is naturally ordered by inclusion. Upsets of $\mathfrak{P}(Q)$ are naturally in bijection with *antichains* by taking their *minimal* elements.

We now use the construction given in the previous section. As we start with a non deterministic automaton, we first determinize it to obtain an automaton $\mathcal{D}$ that is used to build the universal automaton.

Proposition 5.4. Let $\mathcal{A} = \langle Q, A, E, I, T\rangle$ be a non deterministic automaton. Let $\mathcal{D} = \langle R, A, F, \{I\}, U\rangle$ be the determinisation of $\mathcal{A}$ and $\mathcal{C} = \langle S, A, G, K, \{U\}\rangle$ the co-determinisation of $\mathcal{D}$. Every element of S is an upset of R.

Proof. Let X and Y be two states of $\mathcal{D}$ such that $X \subseteq Y$. It holds $\mathsf{Fut}_{\mathcal{D}}(X) = \bigcup_{p\in X} \mathsf{Fut}_{\mathcal{A}}(p) \subseteq \bigcup_{p\in Y} \mathsf{Fut}_{\mathcal{A}}(p) = \mathsf{Fut}_{\mathcal{D}}(Y)$. Let P be a state of $\mathcal{C}$ which contains X. For every v in $\mathsf{Fut}_{\mathcal{C}}(P)$, $P = v \underset{\mathcal{D}}{\lhd} U$. As X is in P, v is in $\mathsf{Fut}_{\mathcal{D}}(X)$, thus in $\mathsf{Fut}_{\mathcal{D}}(Y)$. Hence, Y is in P. Q.E.D.

Proposition 5.5. Let $\mathcal{A} = \langle Q, A, E, I, T\rangle$ be a non deterministic automaton that recognizes a language L. The universal automaton of L has at most $D(\mathsf{card}(Q))$ states, where $D(n)$ it the n-th Dedekind number.

Proof. Let $n = \mathsf{card}(Q)$ and let $\mathcal{D}$ be the determinisation of $\mathcal{A}$. As the intersection of two upsets is an upset, the elements of $\mathcal{I}_{\mathcal{D}}$ are upsets of $\mathfrak{P}(Q)$, and $D(n)$ is equal to the number of upsets of $\mathfrak{P}(Q)$. Q.E.D.

Proposition 5.6. Let $\mathcal{A} = \langle Q, A, E, I, T\rangle$ be an NFA that recognizes a language L. The number of states of the trim universal automaton of L is bounded by $D(\mathsf{card}(Q)) - 2$.

Proof. Actually, if a state corresponds to the empty upset, it has an empty past and it is therefore not accessible. Likewise, if a state corresponds to the upset $\{\varnothing\}$, it has an empty future and it is therefore not co-accessible. Q.E.D.

The first part of Theorem 5.1 is thus established.

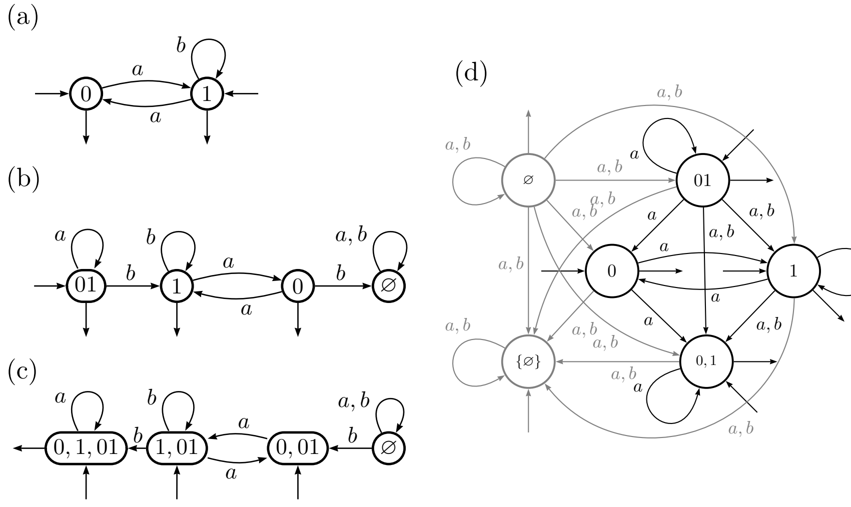


FIGURE 10. The construction of the universal automaton from $\mathcal{Z}_2$

Example 5.7. We give here an example for the construction of the universal automaton. Let $\mathcal{Z}_2$ be the automaton of Figure 10(a). Let $\mathcal{D}_2$ be the determinized automaton of $\mathcal{Z}_2$, drawn on Figure 10(b). Each of its states is a subset of the set of states of $\mathcal{Z}_2$. We denote this set by a word whose letters are the elements of the state: the word 01 stands for the set $\{0, 1\}$. The states of the universal automaton (Figure 10(c)) are upsets of the power set of states of $\mathcal{Z}_2$. We denote an upset by the setr of its minimal elements. For instance 0, 1 means $\{\{1\}, \{2\}, \{0, 1\}\}$. Notice that $\varnothing$ is the empty upset, whereas $\{\varnothing\}$ is the upset with $\varnothing$ as minimal element, *i.e.* the power set itself. The non accessible part of the universal automaton is drawn in gray. The automaton $\mathcal{Z}_2$ is an example of the worst case in the construction of the universal automaton. Indeed, $D(2) = 6$.

Likewise, $D(3) = 20$ and we give a three-state automaton which recognizes a language whose universal automaton has twenty states: the automaton $\mathcal{Z}_3$ shown on Figure 11(a). As the number of transitions of the universal automaton is to high to allow to draw them all, the more compact representation given by the écorché is drawn on Figure 11(c).

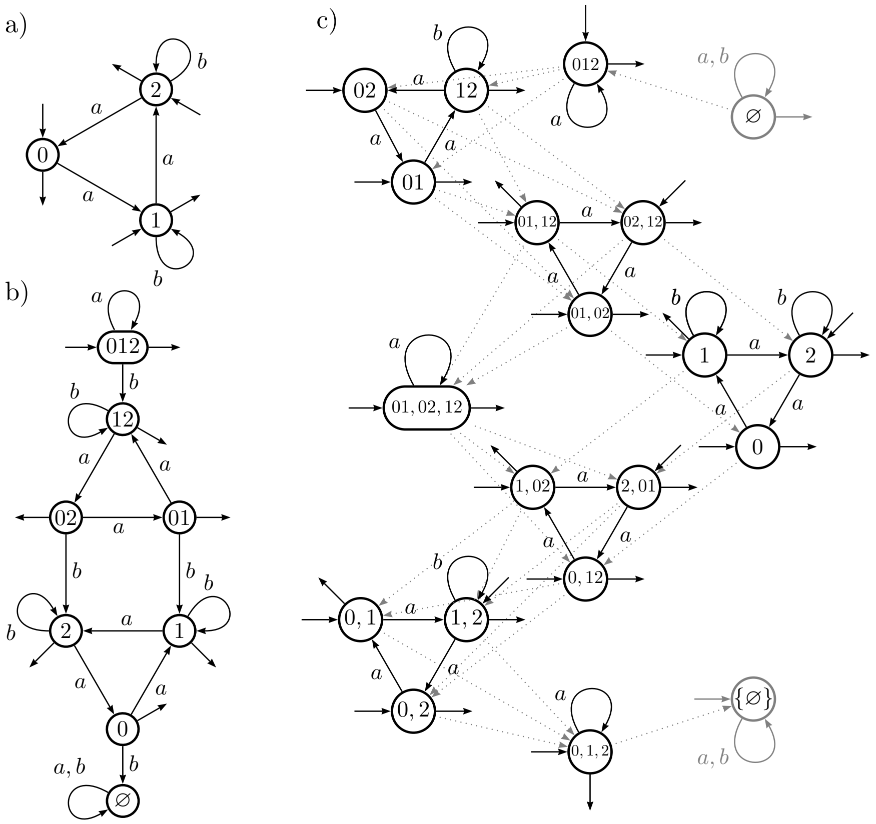


FIGURE 11. The construction of the universal automaton from $\mathscr{Z}_3$

In the following section, we generalise this example to show that, for every n, there exists a n-state NFA that accepts a language whose universal automaton has $D(n)$ states.

5.2 Reaching the bounds of the universal automata

As announced, we introduce first a notation for the dual action induced by an automaton.

Definition 5.8. Let $\mathcal{A} = \langle Q, A, E, I, T\rangle$ be an automaton. The *set of predecessors* of a state p of $\mathcal{A}$ by a letter a is $a \underset{\mathcal{A}}{\lhd} p = \{q \in Q \mid (q, a, p) \in E\}$, denoted $a \lhd p$ if there is no ambiguity. This defines a *left action* of A^* on $\mathfrak{P}(Q)$: for every letter a, every word w, and every subset X of Q, we have:

$$a \underset{\mathcal{A}}{\lhd} X = \bigcup_{p\in X} a \underset{\mathcal{A}}{\lhd} p\,, \quad 1_{A^*} \underset{\mathcal{A}}{\lhd} X = X\,, \quad aw \underset{\mathcal{A}}{\lhd} X = a \underset{\mathcal{A}}{\lhd} (w \underset{\mathcal{A}}{\lhd} X).$$

Obviously, q is in $p \underset{\mathcal{A}}{\rhd} w$ if and only if p is in $w \underset{\mathcal{A}}{\lhd} q$. In the sequel, for every positive integer n, $\mathcal{Z}_n = \langle Q, A, E, I, T\rangle$ is the automaton defined by:

$$\begin{gathered} Q = \mathbb{Z}/n\mathbb{Z}; \qquad A = \{a,b\}; \qquad I = T = Q; \\ E = \{(p,a,p+1) \mid p \in Q\} \cup \{(p,b,p) \mid p \in Q \smallsetminus \{0\}\}. \end{gathered} \tag{1.12}$$

In the sequel, if X is a subset of Q, *i.e.* a subset of $\mathbb{Z}/n\mathbb{Z}$, for every integer k, we denote $X + k = \{x + k \mid x \in X\}$.

Lemma 5.9. Let n be a positive integer. The determinisation of $\mathcal{Z}_n$ is $\mathcal{D}_n = \langle \mathfrak{P}(Q), A, F, \{Q\}, \mathfrak{P}(Q)\smallsetminus\{\varnothing\}\rangle$, with:

$$F = \{(X,a,X+1),(X,b,X\smallsetminus\{0\}) \mid X \subseteq Q\}. \tag{1.13}$$

Proof. As every state of $\mathcal{A}$ is initial, the initial state of $\mathcal{D}$ is Q. As every state of $\mathcal{A}$ is terminal, every state of $\mathcal{D}$ different from $\varnothing$ is terminal.
If X is a subset of Q, $X \underset{\mathcal{A}}{\rhd} a = \bigcup_{p\in X} p \underset{\mathcal{A}}{\rhd} a = \bigcup_{p\in X} p+1 = X+1$; likewise, $X \underset{\mathcal{A}}{\rhd} b = \bigcup_{p\in X, p\neq 0} p = X\smallsetminus\{0\}$. This gives the set of transitions F of $\mathcal{D}$.
We show that every element of $\mathfrak{P}(Q)$ is an accessible state by induction on the number of elements. The set Q itself is the initial state of $\mathcal{D}$. Let assume that X is an accessible state. Let x be an element of X, we show that $X\smallsetminus\{x\}$ is accessible. Actually, $X \rhd a^{n-x}ba^x = (X-x) \rhd ba^x = ((X-x)\smallsetminus\{0\}) \rhd a^x = ((X-x)\smallsetminus\{0\}) + x = X\smallsetminus\{x\}$. Therefore, every element of $\mathfrak{P}(Q)$ is accessible. Q.E.D.

For every subset X of Q, we denote $\underline{X} = \{Y \mid Y \subseteq X\}$, and $(\underline{X})^c = \mathfrak{P}(Q)\smallsetminus\underline{X}$; we can notice that $(\underline{X})^c$ is an upset of $\mathfrak{P}(Q)$.

Lemma 5.10. Let n be a positive integer. The co-determinisation of $\mathcal{D}_n$ is $\mathcal{C}_n = \langle S, A, G, K, V\rangle$, with:

$$\begin{gathered} S = \{(\underline{X})^c \mid X \in \mathfrak{P}(Q)\}; \qquad K = S\smallsetminus\{\varnothing\}; \qquad V = \{(\underline{\varnothing})^c\} \\ G = \{(\underline{X})^c, a, (\underline{Y})^c \mid (X,a,Y) \in F\} \cup \{\left(\underline{X \cup \{0\}}\right)^c, b, (\underline{X})^c \mid X \subseteq Q\}. \end{gathered} \tag{1.14}$$

Proof. As any state X of $\mathcal{D}_n$ different from $\varnothing$ is final, the state $t = (\underline{\varnothing})^c$ is the final state of $\mathcal{D}_n$. First, we show by induction on the word w that any state $P = w \triangleleft t$ is in S. This is obviously true if w is the empty word: $P = t$. If $P = (\underline{X})^c$ is in S, so its predecessors are:

$$\begin{aligned} a \underset{\mathcal{C}}{\triangleleft} (\underline{X})^c =& \{a \underset{\mathcal{D}}{\triangleleft} Y \mid Y \not\subseteq X\} = \{Y \mid Y \not\subseteq a \underset{\mathcal{D}}{\triangleleft} X\} = \left(\underline{a \underset{\mathcal{D}}{\triangleleft} X}\right)^c \\ =& (\underline{X-1})^c ; \end{aligned} \tag{1.15}$$

$$\begin{aligned} b \underset{\mathcal{C}}{\triangleleft} (\underline{X})^c =& \{b \underset{\mathcal{D}}{\triangleleft} Y \mid Y \not\subseteq X\} \\ =& \{Y \mid Y \not\subseteq X, 0 \notin Y\} \cup \{Y \cup \{0\} \mid Y \not\subseteq X, 0 \notin Y\} \\ =& \{Y \mid Y \not\subseteq X \cup \{0\}, 0 \notin Y\} \cup \{Y' \mid Y' \not\subseteq X \cup \{0\}, 0 \in Y'\} \\ =& \{Y \mid Y \not\subseteq X \cup \{0\}\} = \left(\underline{X \cup \{0\}}\right)^c . \end{aligned} \tag{1.16}$$

We show that every element $P = (\underline{X})^c$ of S is co-accessible from t.If $X = \varnothing$, then $P = t$. If $P = (\underline{X})^c$ is co-accessible, for any x in Q, $P' = \left(\underline{X \cup \{x\}}\right)^c$ is co-accessible too:

$$a^{n-x} b a^x \underset{\mathcal{C}}{\triangleleft} P = a^{n-x} b \underset{\mathcal{C}}{\triangleleft} (\underline{X-x})^c = a^{n-x} \underset{\mathcal{C}}{\triangleleft} \left(\underline{(X-x) \cup \{0\}}\right)^c = \left(\underline{X \cup \{x\}}\right)^c . \tag{1.17}$$

Therefore the set of states of $\mathcal{C}_n$ is exactly S. Q.E.D.

Lemma 5.11. Let Q be a finite set. The intersection closure of $\{(\underline{X})^c \mid X \in \mathfrak{P}(Q)\}$ is exactly the set of upsets of $\mathfrak{P}(Q)$.

Proof. Let $\mathcal{U}$ be an upset of $\mathfrak{P}(Q)$. For every Y in $\mathcal{U}$, for every X not in $\mathcal{U}$, $Y \not\subseteq X$. Hence, Y is in $(\underline{X})^c$ and $\mathcal{U}$ is a subset of $(\underline{X})^c$. Thus, as X is not in $(\underline{X})^c$, it comes $\mathcal{U} = \bigcap_{X \notin \mathcal{U}} (\underline{X})^c$. Q.E.D.

With this last lemma, the proof of Theorem 5.1 is now complete.

In the case of a one-letter alphabet, the determinisation algorithm is known not to be exponential. Indeed, if $\mathcal{A}$ is a one-letter NFA with n states, the determinisation of $\mathcal{A}$ (and the minimal automaton of the accepted language) has at most $G(n)$ states (*cf.* [Chr86]), where $G(n)$ is the Landau function of n, that is, the maximal least common multiple of a set of integers with sum equal to n. We show that in this case, the obvious upper bound $2^{G(n)}$ for the size of the universal automaton is tight.

Proposition 5.12. For every integer n, there exist automata with n states over a one-letter alphabet such that, if L is the accepted language:

(i) $\mathcal{U}_L$ has $2^{G(n)}$ states;

(ii) the trim part of $\mathcal{U}_L$ has $2^{G(n)} - 2$ states.

Proof. There exist an integer r and r numbers $k_1, .., k_r$ such that $k_1 + .. + k_r = n$ and $\mathsf{lcm}(k_1, k_2, ..., k_r) = G(n)$. Let Q be the disjoint union of all the $(Q_i = \mathbb{Z}/k_i\mathbb{Z})$ for i in $[1; r]$ and let $\mathcal{Y}_n = \langle Q, \{a\}, E, I, T\rangle$ be the automaton defined by:

$$I = \{0 \in Q_i \mid i \in [1; r]\}, \quad T = Q \setminus I, \quad E = \{(p, a, p+1) \mid \exists i, p \in Q_i\}.$$

The determinisation of $\mathcal{Y}_n$ is the automaton $\mathcal{D}_n = \langle R, \{a\}, F, J, U\rangle$, with $R = \mathbb{Z}/G(n)\mathbb{Z}$, $J = \{0\}$, $U = R \setminus J$ and $F = \{(p, a, p+1) \mid p \in R\}$.

The states of the co-determinisation of $\mathcal{D}_n$ are all the subset of R with $\mathsf{card}(R) - 1$ elements. The intersection closure of this set of states is equal to $\mathfrak{P}(R)$. Hence, the universal automaton of the language recognized by $\mathcal{Y}_n$ has $2^{G(n)}$ states and the trim universal automaton has $2^{G(n)} - 2$ states. Q.E.D.

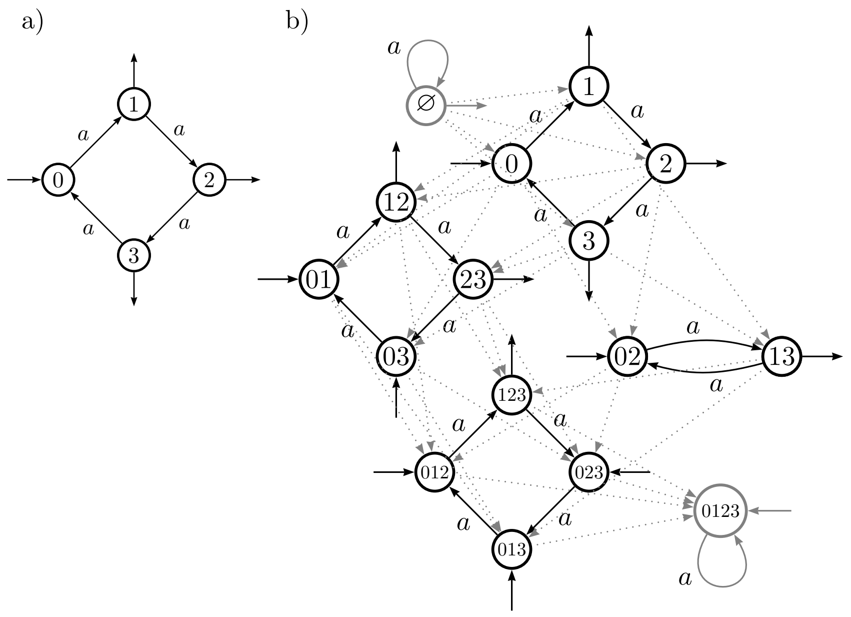


FIGURE 12. The automaton $\mathcal{Y}_4$ and its universal automaton

Remark 5.13. Starting from a one-letter DFA with n states ($n > 1$), it is not possible to obtain a trim universal automaton with $2^n - 1$ states. The state corresponding to the empty set in the construction of Theorem 4.1 cannot be accessible. If the full set corresponds to a co-accessible state, it means that every state of the DFA is final, thus every word is accepted and the universal automaton has one state, or, if the DFA is not complete, the language is a finite prefix language and the universal automaton has n states. Therefore, the trim universal automaton has at most $2^n - 2$ states.

Example 5.14. Let $\mathcal{Y}_4$ be the automaton of Figure 12 a). It is equal to $\mathcal{D}_4$ (actually, $G(4) = 4$). The universal automaton, drawn on Figure 12 b), has $2^4 = 16$ states, including a non accessible state and a non co-accessible state.

6 Equations in the universal automaton

John H. Conway who gave, in his own language and terminology [Con71], another definition of the universal automaton, was not at all interested in the computation of small NFA's for a regular language. He used the *factor matrix* of a language to solve two dual classes of problems. First, in the *approximation problem*, are given on one hand a language L in A^* and on the other hand a family $\mathcal{K} = \{K_1, \ldots, K_n\}$, all in A^*. The latter determines a *substitution* σ from X^* into A^*, with $X = \{x_1, \ldots, x_n\}$ and $\sigma(x_i) = K_i$. The construction of the universal automaton of L allows us to show that the set W of words w in X^* such that $\sigma(w)$ is contained in L is regular when L is regular (and without any hypohesis on the K_i's).

The dual problem is the one of (in)equations. The regular languages L in A^* and K in X^* being given, the universal automaton of L allows the effective computation of all maximal n-tuples of languages $\{H_1, \ldots, H_n\}$ such that $\sigma(K)$ is contained in L.

6.1 The approximation problem

The construction of the automaton $\mathcal{U}_L$ can be seen as a special case of an approximation problem: the reasoning that proves that $\mathcal{U}_L$ accepts L can be generalised to other families of subsets than the generating set of A^*, with remarkable results.

Let L be a language of A^* and $\mathcal{K} = \{K_1, \ldots, K_n\}$ a family of n languages of A^*. We set $X = \{x_1, x_2, \ldots, x_n\}$ an n-letter alphabet and $\sigma\colon B^* \to A^*$ the *substitution* defined by

$$\forall i \in [1;n] \qquad \sigma(x_i) = K_i .$$

The sole consideration of the syntactic morphism allows us to show that the language W over B^*,

$$W = \{f \in B^* \mid \sigma(f) \subseteq L\} ,$$

is recognisable if L is recognisable and without any assumption on the K_i's — a corollary of a result in [Reu79], see [Sak03]. But here we prove the result and give it a more precise interpretation using the universal automaton.

To simplify the statements, with K_i and hence σ being fixed, we write $\breve{\sigma}$ for the map from $\mathfrak{P}(A^*)$ to $\mathfrak{P}(B^*)$ defined by

$$\forall L \in \mathfrak{P}(A^*) \qquad \breve{\sigma}(L) = \{f \in B^* \mid \sigma(f) \subseteq L\}\,, \tag{1.18}$$

that is, $\breve{\sigma}(L)$ is equal to the language W defined above. The map $\breve{\sigma}$ acts as the *inverse* of the substitution σ but retains only those words whose image under σ is *contained* in L. In other words, $\sigma(\breve{\sigma}(L))$ is the *best possible approximation* (by default) to L as a *sum of products* of languages K_i and $\breve{\sigma}(L)$ describes how this approximation is constructed.

Let L be a language of A^*, $\mathcal{U}_L$ its universal automaton and F^L its factor matrix. We write $\mathcal{S}_L^{\mathcal{K}}$ for the automaton over X^* obtained from $\mathcal{U}_L$ by replacing each label $E^L_{X,Y'}$ by the set of letters in X whose image under σ is contained in $F^L_{X,Y'}$:

$$\forall (X,Y),(X',Y') \in \mathcal{F}_L \qquad (X,Y) \xrightarrow[\mathcal{S}_L^{\mathcal{K}}]{x} (X',Y') \Longleftrightarrow \sigma(x) \subseteq F^L_{X,Y'}.$$

Theorem 6.1 (Conway [Con71]). $\breve{\sigma}(L) = |\mathcal{S}_L^{\mathcal{K}}|$.

Proof. The proof goes by induction on the length of f and amounts to establish that, for all (X,Y), (X',Y') in $\mathcal{F}_L$, and all f in B^*, it holds:

$$(X,Y) \xrightarrow[\mathcal{S}_L^{\mathcal{K}}]{f} (X',Y') \quad \Longleftrightarrow \quad \sigma(f) \subseteq F^L_{X,Y'}.$$

For $|f| = 1$, this is exactly the definition of $\mathcal{S}_L^{\mathcal{K}}$.

Suppose then that we have $(X,Y) \xrightarrow[\mathcal{S}_L^{\mathcal{K}}]{xf} (X',Y')$; then there exists (X'',Y'') in $\mathcal{F}_L$ such that $(X,Y) \xrightarrow[\mathcal{S}_L^{\mathcal{K}}]{x} (X'',Y'')$ and $(X'',Y'') \xrightarrow[\mathcal{S}_L^{\mathcal{K}}]{f} (X',Y')$. We thus have $\sigma(x) \subseteq F^L_{X,Y''}$ by definition of $\mathcal{S}_L^{\mathcal{K}}$ and $\sigma(f) \subseteq F^L_{X'',Y'}$ by induction hypothesis. Then, by Equation (1.8),

$$\sigma(xf) \subseteq F^L_{X,Y''} F^L_{X'',Y'} \subseteq F^L_{X,Y'}.$$

Conversely, suppose that $\sigma(xf) = \sigma(x)\sigma(f) \subseteq F^L_{X,Y'}$. By Lemma 3.4, there exists (X'',Y'') in $\mathcal{F}_L$ such that $\sigma(x) \subseteq F^L_{X,Y''}$ and $\sigma(f) \subseteq F^L_{X'',Y'}$. This, in turn, by definition of $\mathcal{S}_L^{\mathcal{K}}$ and by induction hypothesis, implies $(X,Y) \xrightarrow[\mathcal{S}_L^{\mathcal{K}}]{xf} (X',Y')$. Q.E.D.

As announced, a mere consequence of Theorem 6.1 is that if L is regular, $\mathcal{U}_L$ has a finite number of states and $\breve{\sigma}(L)$ is regular. The definition of $\mathcal{S}_L^{\mathcal{K}}$ is itself a procedure for computing the 'best approximation' to L, on condition that we know how to compute effectively the factors of L and the inclusion of K_i in these factors. These conditions are fulfilled in particular when considering the rational sets of a free monoid. We then deduce:

Corollary 6.2. Given a regular language L and a *finite* family $\mathcal{K}$ of regular languages over A^*, we can decide whether L belongs to $\mathsf{Rat}\mathcal{K}$, the rational closure of $\mathcal{K}$.

Proof. We compute the best approximation to L by the n languages of the family $\mathcal{K}$ and then decide whether this approximation is equal to L. Q.E.D.

The elegance of this proof, and the efficiency of the computations it entails is to be compared with those of the proofs given subsequently for the same result (*e.g.* [Has83]).

6.2 Solutions of pure language equations

The problem of approximation is susceptible to a 'dual' approach.[6] The (recognisable) subset L of A^* having been fixed, instead of choosing the subsets K_i, that is the substitution $\sigma\colon B^* \to A^*$, and trying to compute the language $\breve{\sigma}(L)$ over B^*, we can choose a language W (not even necessarily regular) over a free monoid B^* and seek a substitution $\sigma\colon B^* \to A^*$ such that $\sigma(W) \subseteq L$, which will be called a *sub-solution* of the *problem* (L, W). The sub-solutions are naturally (and partially) ordered by inclusion of the images of the letters of B, and the interesting sub-solutions are the maximal ones.

Theorem 6.3 (Conway, [Con71])**.** Let L be a subset of A^*, W a language of B^* and $\sigma\colon B^* \to A^*$ a maximal sub-solution of the problem (L, W). Then for each x in B, $\sigma(x)$ is an intersection of factors of L.

Proof. Let $f = x_1 x_2 \ldots x_n$ be a word of W. If σ is a solution of (L, W), $\sigma(x_1)\sigma(x_2)\ldots\sigma(x_n) \subseteq L$. By Lemma 3.4, and an induction argument, there exist $(X_0, Y_0), (X_1, Y_1), \ldots, (X_n, Y_n)$ in $\mathcal{F}_L$ such that

$$\sigma(x_i) \subseteq F^L_{X_{i-1},Y_i}$$

for each i in $[1; n]$. As these inclusions are verified for each f in W, each $\sigma(x_i)$ is contained in an intersection of factors and such an intersection is a maximal component in a sub-solution of the problem. Q.E.D.

Corollary 6.4 (Conway, [Con71])**.** If L is regular, the maximal sub-solutions of the problem (L, W) are k-tuples ($k = \mathrm{Card}(B)$) of regular subsets of A^*. If in addition W is regular, we can effectively compute all the maximal sub-solutions of the problem (L, W).

Proof. If L is regular, there is only a finite number of factors that are all regular and their intersections are finite in number and regular. There is only a finite number of k-tuples of intersections among which all the maximal

[6] This is not the left-right duality of automata, but rather a vector–linear form duality.

sub-solutions are found. If W is regular we can effectively find all the k-tuples which are sub-solutions and keep only the maximal ones. Q.E.D.

Example 6.5. A regular language L of A^* being chosen, let us find all the subsets U such that $U^2 \subseteq L$ and U is maximal for this property (*i.e.* find the maximal sub-solutions of the problem (L, x^2)). If $U^2 \subseteq L$, (U, U) is a subfactorisation of L, it is dominated by (at least) one factorisation (X, Y), and $U \subseteq X \cap Y$. The maximal sub-solutions are thus among the $X \cap Y$ when (X, Y) varies over $\mathcal{F}_L$.

7 Stars in the universal automaton

Last but not least, the universal automaton contains informations on the *star height* of the language if it is a regular one, may be not always but certainly for some subfamilies of regular languages — and this was what motivated first the interest of the authors in this construction.

The computation of the star height problem is a hard question that was stated by Eggan [Egg63] in 1963. It was positively solved in 1988 by Hashiguchi [Has88] and Kirsten gave more recently a particulary elegant proof for its decidability [Kir05]. The results we present here do not give the solution of the star height problem for any regular language, but in the cases where they can be applied, they give more precise informations on the form of the result than the other works.

7.1 Star height and loop complexity

The *star height* of a regular expression E, denoted by $\mathsf{h}(\mathsf{E})$, is defined recursively by:

$$\begin{aligned} &\text{if } \mathsf{E} = 0,\ \mathsf{E} = 1 \text{ or } \mathsf{E} = a \in A, && \mathsf{h}(\mathsf{E}) = 0\,, \\ &\text{if } \mathsf{E} = \mathsf{E}' + \mathsf{E}'' \text{ or } \mathsf{E} = \mathsf{E}' \cdot \mathsf{E}'', && \mathsf{h}(\mathsf{E}) = \max(\mathsf{h}(\mathsf{E}'), \mathsf{h}(\mathsf{E}''))\,, \\ &\text{if } \mathsf{E} = \mathsf{F}^*, && \mathsf{h}(\mathsf{E}) = 1 + \mathsf{h}(\mathsf{F})\,. \end{aligned}$$

Example 7.1. The expressions $(a+1)(a^2+b)^*a+1$ and $(b^*a+1)(ab^*a)^*$ have star height 1 and 2 respectively. As they both denote the same language accepted by the automaton $\mathcal{A}_2$ shown at Figure 13, two equivalent expressions may have different star heights.

Definition 7.2. The star height of a regular language L of A^*, which we note as $\mathsf{h}(L)$, is the *minimum* of the star height of the expressions that denote the language L:

$$\mathsf{h}(L) = \min\{\mathsf{h}(\mathsf{E}) \mid \mathsf{E} \in \mathsf{RatE}A^* \quad |\mathsf{E}| = L\}\,.$$

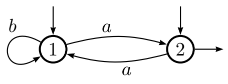


FIGURE 13. The automaton $\mathcal{A}_2$

The star height induces a hierarchy on regular languages. We shall give examples for the fact (see Corollary 7.12):

Fact 7.3. There exist regular languages of arbitrary large star height.

The star height of an expression reflects also a structural property of an automaton which corresponds to that expression (more precisely, of the underlying graph of an automaton). In order to state it, we first define the notion of *a ball* of a graph: a ball in a graph is a strongly connected component that contains at least one arc.

Definition 7.4. The *loop complexity*[7] of a graph $\mathcal{G}$ is the integer $\mathsf{lc}(\mathcal{G})$ recursively defined by:

$$\mathsf{lc}(\mathcal{G}) = 0 \qquad \text{if } \mathcal{G} \text{ contains no ball (in particular, if } \mathcal{G} \text{ is empty);}$$
$$\mathsf{lc}(\mathcal{G}) = \max\{\mathsf{lc}(\mathcal{P}) \mid \mathcal{P} \text{ ball of } \mathcal{G}\} \qquad \text{if } \mathcal{G} \text{ is not a ball itself;}$$
$$\mathsf{lc}(\mathcal{G}) = 1 + \min\{\mathsf{lc}(\mathcal{G} \setminus \{s\}) \mid s \text{ vertex of } \mathcal{G}\} \qquad \text{if } \mathcal{G} \text{ is a ball.}$$

As Eggan showed, star height and loop complexity are the two faces of the same notion:

Theorem 7.5 (Eggan [Egg63])**.** The star height of a language L is equal to the minimal loop complexity of an automaton that accepts L.

More precisely, from every automaton with loop complexity n, an expression with star height n can be computed, and *vice-versa.* Theorem 7.5 allows to deal with automata instead of expressions, and to look for automata of minimal loop complexity instead of expressions of minimal star height. A reason why star height, or loop complexity is not an easy parameter to compute is given by the following fact, for which we give an example below (see Example 7.13).

Fact 7.6. The minimal automaton is not always of minimal loop complexity (for the language it recognises).

[7] Eggan [Egg63] as well as Cohen [Coh70] and Hashiguchi [HH79] call it 'cycle rank', Büchi calls it 'feedback complexity'. McNaughton [McN67] calls loop complexity of a language the minimum cycle rank of an automaton that accepts the language. We have taken this terminology and made it parallel to star height, for 'rank' is a word of already many different meanings.

The following structural result gives a criterium to bound the loop complexity of an automaton.

Definition 7.7. Let $\mathcal{A}$ and $\mathcal{B}$ be two automata and let φ be a surjective morphism from $\mathcal{A}$ onto $\mathcal{B}$. The morphism φ is *conformal* if every path in $\mathcal{B}$ is the image of a path in $\mathcal{A}$.

Theorem 7.8 (McNaughton, [McN67])**.** If $\varphi\colon \mathcal{B} \to \mathcal{A}$ is a conformal morphism, the loop complexity of $\mathcal{B}$ is larger than or equal to that of $\mathcal{A}$: that is, $\mathsf{lc}(\mathcal{B}) \geqslant \mathsf{lc}(\mathcal{A})$.

We first show a lemma:

Lemma 7.9. Let $\varphi\colon \mathcal{B} \to \mathcal{A}$ be a conformal morphism. For every ball $\mathcal{P}$ in $\mathcal{A}$, there exists a ball $\mathcal{Q}$ in $\mathcal{B}$ such that the restriction of φ to $\mathcal{Q}$ is a conformal morphism from $\mathcal{Q}$ to $\mathcal{P}$.

Proof. This lemma (like the theorem) is in fact a proposition about graphs, but we shall use automata-theoretic notions to simplify the proof. We assume, possibly by changing them all, that each transition of $\mathcal{A}$ bears a distinct label, and that each state of $\mathcal{A}$ is both initial and final; this may change the language accepted by $\mathcal{A}$ but has no effect on its loop complexity. The words of the language recognised by $\mathcal{A}$ (*resp.* by a subautomaton $\mathcal{P}$ of $\mathcal{A}$) describe paths in the *graph* $\mathcal{A}$ (*resp.* in the sub-graph $\mathcal{P}$). The transitions of $\mathcal{B}$ are labeled in such a way that φ is an automata morphism and each state of $\mathcal{B}$ is both initial and final.

Let $\mathcal{P}$ be a ball in $\mathcal{A}$ and $\mathcal{R} = \mathcal{P}\varphi^{-1}$. Set $n = \|\mathcal{R}\|$ and $m = \|\mathcal{P}\|$ to be the number of states of $\mathcal{R}$ and $\mathcal{P}$ respectively and consider a circuit (hence a word) w which visits all the paths in $\mathcal{P}$ of length less than 2^{n+m}. The circuit w^n is a path in $\mathcal{P}$ which can be lifted to a path in $\mathcal{R}$ (since φ is conformal). By the proof of the block star lemma, a factor w^k is the label of a *circuit* in $\mathcal{R}$; let $\mathcal{Q}$ be the ball in $\mathcal{R}$, and hence in $\mathcal{B}$, that contains this circuit. By construction, $\mathcal{Q}$ recognises all words of length less than 2^{n+m} of the language recognised by $\mathcal{P}$, hence $\mathcal{Q}$ is equivalent to $\mathcal{P}$, hence all the paths in $\mathcal{P}$ become paths in $\mathcal{Q}$: thus, φ is conformal from $\mathcal{Q}$ to $\mathcal{P}$. Q.E.D.

Proof of Theorem 7.8. Suppose that the property is false, and proceed by *reductio ad absurdum.* Among the automata which are sent by a conformal morphism to an automaton of strictly greater complexity, let $\mathcal{B}$ be an automaton of minimal loop complexity d, and let $\mathcal{A}$, of complexity c, be the image of $\mathcal{B}$ under a conformal morphism: thus, $c > d$.

If $d = 0$, the length of the paths in $\mathcal{B}$ is bounded and it is impossible for φ to be conformal, hence $d > 0$.

By definition, there is a ball $\mathcal{P}$ in $\mathcal{A}$ of complexity c and, by Lemma 7.9, a ball $\mathcal{Q}$ in $\mathcal{B}$ whose image under φ is $\mathcal{P}$. This ball is of complexity at most d

but also, by the minimality of d, at least d. There exists a state q in $\mathcal{Q}$ such that

$$\mathsf{lc}(\mathcal{Q}\smallsetminus\{q\}) = d - 1. \tag{1.19}$$

Let $p = q\varphi$, $\mathcal{P}' = \mathcal{P}\smallsetminus\{p\}$ and $\mathcal{Q}' = \mathcal{Q}\smallsetminus\{p\varphi^{-1}\}$; we have $\mathsf{lc}(\mathcal{Q}') \leqslant \mathsf{lc}(\mathcal{Q}\smallsetminus\{q\}) = d - 1$ and $\mathsf{lc}(\mathcal{P}') \geqslant c - 1 > d - 1$.

Every path in $\mathcal{P}'$ is a path in $\mathcal{P}$ which does not visit p, hence the image of a path in $\mathcal{Q}$ which does not go through any of the vertices of $p\varphi^{-1}$; that is, the image of a path in $\mathcal{Q}'$: thus, φ is a conformal morphism from $\mathcal{Q}'$ to $\mathcal{P}'$, which contradicts the assumed minimality of d. Q.E.D.

7.2 Star height of group languages

The star height of a group language can be computed within the universal automaton. The simplest instance of this fact is the following statement which provides a new, easier, and clearer presentation of McNaughton's proof of computability of the star height of pure group languages.

Theorem 7.10 (Lombardy-Sakarovitch, [LS03])**.** The universal automaton of a regular group language L contains a subautomaton of minimal loop complexity that recognises L.

Since the universal automaton of a regular language is finite, we can enumerate its subautomata, keeping those that recognise the language, and from among them find those of minimal loop complexity. We therefore have:

Corollary 7.11 (McNaughton, [McN67])**.** The star height of a regular group language is computable.

Furthermore, the same theorem allows us to establish directly, a result whose original proof relied on a highly subtle combinatorial method. Let W_q be the language defined by:

$$W_q = \{w \in \{a,b\}^* \mid |w|_a \equiv |w|_b \mod 2^q\}.$$

Corollary 7.12 (Dejean-Schützenberger, [DS66])**.** $\mathsf{lc}(W_q) = q$.

In this case indeed the universal automaton is isomorphic to the minimal automaton, which has thus the minimal loop complexity (see below).

Example 7.13. Let H_2 and H_3 be the languages over $A^* = \{a,b\}^*$ consisting of words whose number of a's is *congruent* to the number of b's *plus* 1 modulo 2 and 3 respectively and H_6 their union:

$$H_2 = \{f \mid |f|_a - |f|_b \equiv 1 \mod 2\}\,, \quad H_3 = \{f \mid |f|_a - |f|_b \equiv 1 \mod 3\}$$
$$\text{and} \qquad H_6 = \{f \mid |f|_a - |f|_b \equiv 1, 3, 4 \text{ or } 5 \mod 6\}.$$

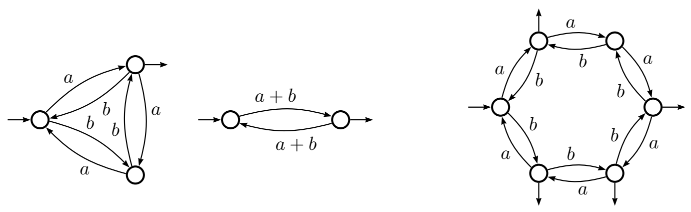


FIGURE 14. An automaton of minimal loop complexity (left) which is not the minimal automaton (right) for H_6

The minimal automaton of H_6 is the 'double ring' of length 6 whose loop complexity is 3. The minimal automata of H_2 and H_3 have complexity 1 and 2, hence the star height of H_6 is at most 2 (*cf.* Figure 14).

Figure 15 shows the écorché of the universal automaton of H_6. We see, all the better for its grey background, a subautomaton of this universal automaton which recognises H_6, with a minimal complexity. This subautomaton is equal to the union of the minimal automata of H_2 and H_3 seen above, and this is not a coincidence.

Let $\mathcal{B}$ be an automaton of minimal loop complexity which recognises L and $\varphi\colon \mathcal{B} \to \mathcal{U}_L$ a morphism from $\mathcal{B}$ to the universal automaton of L. If φ is a *conformal* morphism from $\mathcal{B}$ to its image $\varphi(\mathcal{B})$ in $\mathcal{U}_L$, this subautomaton of $\mathcal{U}_L$ is of lesser or equal complexity to that of $\mathcal{B}$ by Theorem 7.8 and the property is proved. However, in the general case φ is not conformal. The proof comes down to showing that nonetheless φ is conformal on some subautomata of $\mathcal{B}$ (on some balls) which are crucial for the complexity. We start by proving some properties of the structure of the universal automaton of a group language.

7.2.1 The universal automaton of a group language

In what follows, $L \subseteq A^*$ is a group language, $\alpha\colon A^* \to G$ is the syntactic morphism, $P = \alpha(L)$ and $\mathcal{A}_L = \langle G, A, \delta, 1_G, P\rangle$ is a complete accessible deterministic automaton that recognises L. For w in A^* and g in G we therefore write $g \triangleright w$ for $g\alpha(w)$, multiplication in G.

As we have seen (Subsection 3.2), the universal automaton $\mathcal{U}_L$ of L, is obtained by considering the factorisations (X, Y) of P in G and that if

$$(X_1, Y_1) \xrightarrow[\mathcal{U}_L]{a} (X_2, Y_2)$$

is a transition of $\mathcal{U}_L$, then $X_1(a\alpha)Y_2 \subseteq P$ and hence

$$X_1 \triangleright a \subseteq X_2 \quad \text{and} \quad \alpha(a)^{-1} Y_2 \subseteq Y_1.$$

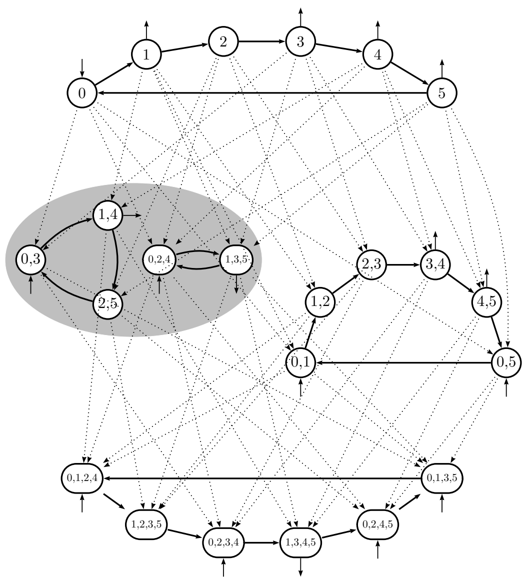


FIGURE 15. The écorché of the universal automaton of H_6 (without the sink and co-sink states). The bold arrows represent a double transition, one labeled a in the direction of the arrow and one labeled b in the opposite direction; the dotted arrows represent the spontaneous transitions.

Lemma 7.14. The balls of $\mathcal{U}_L$ are deterministic and complete.

Proof. Let (X_1, Y_1) and (X_2, Y_2) be two states of $\mathcal{U}_L$ belonging to the same ball. There exists u and v in A^* such that $X_1 \triangleright u \subseteq X_2$ and $X_2 \triangleright v \subseteq X_1$. As G is a group, the action of every element is injective and $\|X_1\| \leqslant \|X_2\| \leqslant \|X_1\|$ hence $\|X_1\| = \|X_2\|$ and $X_1 \triangleright u = X_2$. That is, X_2 is *uniquely determined* by X_1 and u: the ball is deterministic.

Furthermore, if (X, Y) is a factorisation of P, then $(X(u\alpha), (u\alpha)^{-1}Y)$ is also a factorisation of P, for all u in A^*, and there exists a transition labeled u from the first to the second. For all u, there exists v such that $(uv)\alpha = 1_G$, and hence a transition labeled v from $(X(u\alpha), (u\alpha)^{-1}Y)$ to (X, Y). Thus, $(X(u\alpha), (u\alpha)^{-1}Y)$ belongs to the same ball as (X, Y) and the ball is complete. Q.E.D.

A direct consequence of Lemma 7.14 is the following.

Corollary 7.15. Let L be a group language whose image in its syntactic monoid is reduced to one element. Then $\mathcal{U}_L$ is isomorphic with the minimal automaton of L whose loop complexity is thus minimal.

7.2.2 Proof of Theorem 7.10

Lemma 7.16. For every integer k, there exists a word w_k in A^* whose image in G is 1_G and such that every computation of length k of every ball C in $\mathcal{U}_L$ is contained in every computation of C labeled w_k.

Proof. Every word whose image in G is 1_G labels a circuit in every ball of $\mathcal{U}_L$ and for every source vertex. For each ball, and each vertex of this ball, we construct a circuit which visits every computation of length k of this ball. The product of the labels of all these circuits is a word w_k that answers the question. Q.E.D.

We now turn to the proof of the theorem itself.

Proof of Theorem 7.10. The automaton $\mathcal{B}$, an automaton of minimal loop complexity which recognises L, has n states. Let g be in P, a final state of $\mathcal{A}_L$, and u_g be a word in A^* that is sent to g by α. For every integer k, the word $(w_k)^n u_g$ is in L and is hence accepted by $\mathcal{B}$. The Block Star Lemma, applied to the factors w_k, ensures that there exists a state p_k of $\mathcal{B}$ such that there exists a circuit with source p_k labeled by a certain power (w_k^l). Let $\mathcal{D}_k$ be the ball in $\mathcal{B}$ which contains p_k, and hence this circuit. We thus obtain an infinite sequence of balls $\mathcal{D}_k$ in which at least one ball $\mathcal{D}$ in $\mathcal{B}$ appears infinitely often.

Let $\mathcal{C}$ be the ball in $\mathcal{U}_L$ which contains the image of $\mathcal{D}$ under the morphism $\varphi\colon \mathcal{B} \to \mathcal{U}_L$. For every path c in $\mathcal{C}$, there exists a k greater than the length of c, an integer l and a state p of $\mathcal{D}$ such that there exists a loop in $\mathcal{D}$ with source p labeled $(w_k)^l$. This same word $(w_k)^l$ labels a loop in $\mathcal{C}$ which contains all the computations of length less than or equal to k; it thus contains c in particular. That is, c is the image of a computation of $\mathcal{D}$, hence on one hand, $\mathcal{C}$ is the image of $\mathcal{D}$ under φ and on the other, the restriction of φ to $\mathcal{D}$ is *conformal.* By Theorem 7.8, $\mathsf{lc}(\mathcal{D}) \geqslant \mathsf{lc}(\mathcal{C})$.

Let (X, Y) be the factorisation, which is the image of p under φ (the state p that was defined just above). Since $(w_k)^{l'}$ is in $\mathsf{Past}_{\mathcal{B}}(p)$, 1_G is in $\mathsf{Past}_{\mathcal{U}_L}((X, Y))$ and hence 1_G is in X; that is, (X, Y) is an initial state of $\mathcal{U}_L$. Likewise, $(w_k)^{l''} u_g$ is in $\mathsf{Fut}_{\mathcal{B}}(p)$ and g is in Y. Every word u of A^* such that $u\alpha = g$ labels a computation of $\mathcal{C}$ with source (X, Y) and destination $(Xg, g^{-1}Y)$, a final state of $\mathcal{U}_L$, since $1_G \in g^{-1}Y$. Hence u is accepted by $\mathcal{C}$.

We can repeat this construction for each g in P and finally obtain a set of balls of $\mathcal{U}_L$ that recognise all of L and each of which has complexity less than or equal to at least one ball in $\mathcal{B}$. The complexity of the set is at most equal to that of $\mathcal{B}$, which was assumed to be minimal. Q.E.D.

7.3 Star height of reversible languages

The method of the proof of Theorem 7.10 can be both deepened and generalised in order to settle the question of star height for a larger class of languages.

Definition 7.17. An automaton $\mathcal{A}$ is *reversible* if the letters induce partial bijections on the set of states, that is, if for every state p and every letter a, $\mathsf{card}(p \triangleright a) \leqslant 1$ and $\mathsf{card}(a \triangleleft p) \leqslant 1$.

A language is *reversible* if it is recognised by a reversible automaton.

Remark 7.18. A reversible automaton may be not deterministic, nor co-deterministic, for the definition puts no restriction on the number of initial or final sates.

The minimal automaton of a reversible language may be not reversible.

Nevertheless, given an automaton, it can be decided (in polynomial time) whether the language it accepts is reversible or not (see [Pin92]). It is to be stressed that this decision procedure *does not* yield a reversible automaton for a regular language that is determined to be reversible but only the information that such a reversible automaton exists.

Theorem 7.19 (Lombardy-Sakarovitch, [LS02])**.** The universal automaton of a reversible language contains an equivalent subautomaton of minimal loop complexity.

The subautomaton quoted in this result is not necessarily reversible, but it is 'not far' of being so. We then introduce a weaker notion for automata, that will not change the class of accepted languages and that will be useful for both the statement and the proof of the result.

Definition 7.20. An automaton $\mathcal{A}$ is *quasi-reversible* if for every state p and every letter a the following holds:

(i) if $\mathsf{card}(p \triangleright a) > 1$, none of the states in $p \triangleright a$ is in the same ball as p;

(ii) if $\mathsf{card}(a \triangleleft p) > 1$, none of the states is $a \triangleleft p$ is in the same ball as p.

Quasi-reversible automata will be analysed by means of the following decomposition.

Definition 7.21. Let $\mathcal{A}$ be an automaton. A subautomaton $\mathcal{B}$ of $\mathcal{A}$ is *a $\mathcal{A}$-constituent* if the following holds:

(i) any ball of $\mathcal{A}$ is either contained in, or disjoint from, $\mathcal{B}$;

(ii) there is at most one incoming transition to, and one outgoing transition from, every ball of $\mathcal{B}$;

(iii) $\mathcal{B}$ has one initial state and one final state.

It follows from the definition that every finite automaton $\mathcal{A}$ has a finite (but exponential) number of $\mathcal{A}$-constituents and that any $\mathcal{A}$-constituent of a quasi-reversible automaton $\mathcal{A}$ is a reversible automaton. It then holds:

Proposition 7.22. The language accepted by a quasi-reversible automaton is reversible.

We can now give the main result of this section its true form.

Theorem 7.23 (Lombardy, [Lom01])**.** The universal automaton of a reversible language contains an equivalent quasi-reversible subautomaton of minimal loop complexity.

The overall scheme of the proof is illustrated by the figure below.

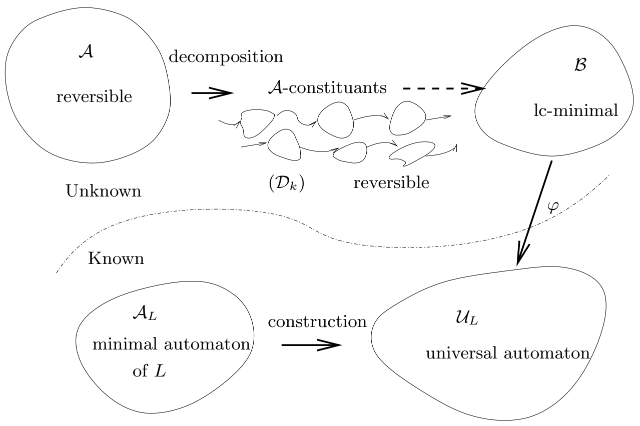


FIGURE 16. The construction underlying the proof of Theorem 7.23

Let L be a reversible language. We know that there exists an unknown automaton $\mathcal{A}$ that recognizes this language and there exists an unknown automaton $\mathcal{B}$ that recognizes this language with a minimal loop complexity. On the other side, we can build the minimal automaton $\mathcal{A}_L$ of the language and the universal automaton $\mathcal{U}_L$. We know that there exists a morphism φ from $\mathcal{B}$ into $\mathcal{U}_L$. Notice that the image of $\mathcal{B}$ by φ may have a loop complexity greater than the loop complexity of $\mathcal{B}$.

Thanks to the reversible automaton $\mathcal{A}$, we decompose L into a union of sub-languages, and we prove that the images of the computations in $\mathcal{B}$ labeled by these sub-languages give a subautomaton of $\mathcal{U}_L$ which is both quasi-reversible and with minimal loop complexity.

To prove the theorem, we must give first a more precise description of the structure of the universal automaton.

7.3.1 The universal automaton of a reversible language

To handle the particular structure of the universal automaton of a reversible language, we consider the construction of the universal automaton from a reversible automaton $\mathcal{A}$ with set of states Q. From Proposition 5.4, every state of the universal automaton is an upset of $\mathfrak{P}(Q)$.

Every upset is characterized by the anti-chain of its minimal elements. The *shape* of an upset R of $\mathfrak{P}(Q)$ is a $|Q|+1$-uplet $s(R)$ of integers such that, for every $k \in [0;|Q|]$, $s(R)_k$ is the number of subsets of Q with cardinal k among minimal elements of R. We define a lexicographic order on shapes:

$$\begin{aligned}&s(R) < s(R') \quad \Longleftrightarrow \\ &\exists k \in [0;|Q|]\,,\ \forall l \in [0;k-1] \qquad s(R)_l = s(R')_l \quad \text{and} \quad s(R)_k < s(R')_k.\end{aligned}$$

Proposition 7.24. If there is a path in the universal automaton from a state with index R to a state with index R', then $s(R) \leqslant s(R')$

Proof. Let w be the label of the path. The state R' contains $\{X \triangleright w \mid X \in R\}$. For every minimal element X in R, either $X \triangleright w$ has the same cardinal as X, or it has a smaller cardinal.

If there is some X such that $|X \triangleright w| < |X|$, thanks to the reversibility of $\mathcal{A}$ there is no X' such that $X' \triangleright w = X \triangleright w$ and $|X'| = |X' \triangleright w|$, hence, $s(R) \leqslant s(R')$. Otherwise, let M be the set of minimal elements of R; the set $\{X \triangleright w \mid X \in M\}$ is a subset of the set of minimal elements of R' and $s(R) \leqslant s(R')$. Q.E.D.

Proposition 7.25. The balls of the universal automaton of a reversible language are reversible.

Proof. Let R and R' be two such states. Let u be a word that labels a path from R to R' and let v be a word that labels a path from R' to R. By Proposition 7.24, two states that belong to the same ball have the same shape. In this case, if M the set of minimal elements of R, for every X in M, $Y = X \triangleright u$ is a minimal element of R' and $|Y| = |X|$. Thanks to the reversibility, the mapping from M into the minimal elements M' of R' is injective. Likewise, there is an injective mapping from M' into M. Therefore, the word u induces a bijection between M and M'; as these minimal elements characterize states, the balls are reversible. Q.E.D.

Corollary 7.26 (Cohen, [Coh70]). If L is a reversible language recognised by a reversible minimal automaton with only one final state, then the minimal automaton has a minimal loop complexity.

Actually, in this case, the universal automaton is the minimal automaton itself and the only subautomaton that accepts the langugae is the complete universal automaton.

7.3.2 Proof of Theorem 7.23

We begin with a series of definitions and notation that allow us to describe a decomposition of a language according to an automaton that accepts it and to state a property of the constituents of that decomposition. This is indeed an adaptation of a method devised by Hashiguchi in [HH79].

Definition 7.27. Let $\mathcal{A}$ be a reversible automaton that accepts L.

(i) We say that a word w is *an idempotent for* $\mathcal{A}$ if, for every state p, $p \triangleright w = p$ or $p \triangleright w = \varnothing$.

(ii) Let $\mathcal{C}$ be a trim $\mathcal{A}$-constituent with m balls. The *marker sequence* of $\mathcal{C}$ is the $2m$-uple $(p_1, q_1, \ldots, p_m, q_m)$ such that p_i (*resp.* q_i) is the first (*resp.* last) state of the i-th ball crossed by any computation.

(iii) A $\mathcal{A}$-constituent with marker sequence $(p_1, q_1, \ldots, p_m, q_m)$ accepts the language $v_0 H_1 v_1 H_2 \ldots v_{m-1} H_m v_m$, where H_i is the language of labels of paths from p_i to q_i.

(iv) We denote by W_i the set of idempotents for $\mathcal{A}$ that label a circuit around p_i.

$$\begin{tikzcd} i \arrow[r, "v_0"] & p_1 \arrow[r, "u_1"] & q_1 \arrow[r, "v_1"] & p_2 \arrow[r, "u_2"] & q_2 \arrow[r] & q_{m-1} \arrow[r, "v_{m-1}"] & p_m \arrow[r, "u_m"] & q_m \arrow[r, "v_m"] & t \end{tikzcd}$$

FIGURE 17. A marker sequence

Lemma 7.28. Let $\mathcal{A}$ be a reversible automaton and $\mathcal{C}$ a trim $\mathcal{A}$-constituent with m balls. Let $\mathcal{B}$ be any automaton equivalent to $\mathcal{A}$.

Then, there exist m states r_1, r_2,…,r_m in $\mathcal{B}$ such that, with the above notation, the following holds:

$$v_0\, W_{p_1} \cap \mathsf{Past}_{\mathcal{B}}(r_1) \neq \varnothing, \qquad H_m\, v_m \cap \mathsf{Fut}_{\mathcal{B}}(r_m) \neq \varnothing,$$
$$\text{and,} \quad \forall i \in [1; m-1] \qquad (H_i\, v_i\, W_{p_i}) \cap \mathsf{Trans}_{\mathcal{B}}(r_i, r_i + 1) \neq \varnothing.$$

For every i in $[1;m]$, for every circuit around p_i labeled by a word v, there exists a circuit around r_i, labeled by a word $u\,v\,w$, where u is in W_{p_i} and w in A^*.

Proof. There exists an integer k such that, for every word $v \in A^*$, the image of v^k is an idempotent for $\mathcal{A}$. Let n be the number of states of $\mathcal{B}$. Let l be an integer that will silently index the sets we define now. For every $i \in [1;m]$, let C_i be the set of words of length smaller than l that label a circuit around p_i in $\mathcal{A}$. Let w_i be the product of all k-th power of words in C_i:

$$w_i = \prod_{v \in C_i} v^k \,.$$

For every $v_0 u_1 v_1 \cdots v_m$ in the $\mathcal{A}$-constituent,

$$w = v_0 (w_1)^n u_1 v_1 (w_2)^n u_2 ... (w_m)^n u_m v_m$$

is in the $\mathcal{A}$-constituent as well. Hence, there is a successful computation labeled by w in $\mathcal{B}$. As $\mathcal{B}$ has only n states, this path contains, for every i, a loop labeled by a power of w_i around a state r_i of $\mathcal{B}$. The m-tuple $r^{(l)} = (r_1, r_2, \ldots, r_m)$ verifies i) and ii) for y shorter than l. If we consider the infinite sequence $r^{(1)}, r^{(2)}, \ldots$, we can find an m-tuple that occurs infinitly often and that verifies the lemma. Q.E.D.

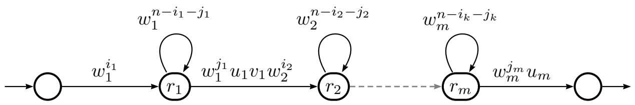


FIGURE 18. A witness word for a $\mathcal{A}$-constituent.

We can now proceed to the proof of Theorem 7.23. We consider a set C of $\mathcal{A}$-constituents such that every element of C accepts at least one word that is not accepted by the other elements of C and such that the union of elements of C is equivalent to $\mathcal{A}$.

Let $\mathcal{D}$ be an element of C and let $p_1, q_1, p_2, \ldots, q_m$ be the marker sequence of $\mathcal{D}$ and let $u = v_0 u_1 v_1 \ldots v_k$ be a word accepted only by $\mathcal{D}$ in C, with v_i labelling a path from q_{i-1} to p_i and u_i a path from p_i to q_i. Let $r_1, r_2, \ldots, r_m$ be the states of $\mathcal{B}$ defined in Lemma 7.28 w.r.t the $\mathcal{A}$-constituent $\mathcal{D}$ and $w_1, \ldots, w_m$ be the idempotents defined in the proof of the lemma. Let φ be a morphism from $\mathcal{B}$ into the universal automaton.

We deal with the strongly connected component of r_i, for $i \in [1;m]$. Let $s_i = \varphi(r_i)$ and $\mathcal{P}_i$ be the ball of $\mathcal{U}_L$ containing s_i. There exist integers $h_1, \ldots, h_m$ (*resp.* $l_1, \ldots, l_m$) such that the word $x = v_0 w_1^{h_1+l_1} u_1 v_1 \ldots w_i^{h_i}$ (*resp.* $y = w_i^{l_i} u_i v_i \ldots w_k^{h_k+l_k} v_k u_k$) is in the past (*resp.* the future) of r_i and thus of s_i.

(i) The morphism φ is conformal on $\mathcal{P}_i$. Let $\mathcal{C}$ be a path of $\mathcal{P}_i$. We can assume, up to make it longer, that this is a circuit around s_i and, up to take it several times, that it is labeled by an idempotent for $\mathcal{A} : z$. The word xzy is in L; every $\mathcal{A}$-constituent that accepts this word accepts also xy, therefore xzy is accepted only by $\mathcal{D}$ in C. As $\mathcal{D}$ is reversible, x labels a path from the initial state to p_i, y a path from p_i to the final state and z a circuit around p_i. Therefore, from Lemma 7.28, there exist a word w idempotent for $\mathcal{A}$ and a word v such that wzv labels a circuit around r_i. The image of this circuit is a circuit around s_i. As w is an idempotent for $\mathcal{A}$, it is an idempotent in the syntactic monoid; hence for every k, $w^k zv$ labels a circuit, if k is large enough, this circuit contains a sub-circuit labeled by a power of w, as the ball is reversible, this power labels a circuit around s_i, and as w is an idempotent, it labels itself a circuit around s_i. As balls are deterministic, the circuit $\mathcal{C}$ around s_i is the image of the part of the circuit around r_i labeled by z. Thus the morphism φ is conformal onto $\mathcal{P}_i$ which have a loop complexity not greater than the loop complexity of $\mathcal{B}$.

(ii) The images of the words linking balls in $\mathcal{B}$ contain no circuit. The word $w_i^{l_i} u_i v_i w_{i+1}^{h_{i+1}}$ is in $\mathsf{Trans}_{\mathcal{B}}(r_i, r_{i+1})$ thus in $\mathsf{Trans}_{\mathcal{U}_L}(s_i, s_{i+1})$. Let $s_i = (X_i, Y_i)$ and let $t_i = (X'_i, Y'_i)$ be the state in $\mathcal{P}_i$ such that u_i is in $\mathsf{Trans}_{\mathcal{U}_L}(s_i, t_i)$. By definition of the universal automaton:

$$X_i w_i^{l_i} u_i v_i w_{i+1}^{h_{i+1}} Y_{i+1} \subseteq L.$$

The words w_i and w_{i+1} are idempotents and L is reversible, therefore it holds $X_i u_i v_i Y_{i+1} \subseteq L$, and X'_i is the smallest left factor that contains $X_i u_i$, hence $X'_i v_i Y_{i+1} \subseteq L$ Thus there exists a path labeld by v_i from t_i to s_{i+1}. This holds for every i in $[1; k-1]$. We prove the same way, that there exists a path from an initial state to s_1 labeled by v_0 and a path from s_m to a final state labeled by v_m.

If one of the intern states of one of these paths labeled by v_i belongs to a ball, the word v_i can be factorised into $x_i y_i$ and there exists an idempotent w for $\mathcal{A}$ such that

$$v_0 u_1 v_1 \ldots u_i x_i w y_i u_{i+1} v_{i+1} \ldots u_k v_k \in L.$$

It can only be accepted by $\mathcal{D}$, which would imply the existence of a circuit between q_i and p_{i+1}.

(iii) The subautomaton obtained in $\mathcal{U}_L$ accepts every word accepted by $\mathcal{D}$. Such a word can be factorised into $v_0u'_1v_1su'_kv_k$, with $p_i \triangleright u'_i = q_i$. There exists a word w_i such that u'_iw_i is an idempotent and both u_iw_i and u'_iw_i label circuits around p_i. As above, these words label circuits around s_i and, as the ball is co-deterministic, the path from s_i labeled by u'_i ends in the same state as the one labeled by u_i, *i.e.* t_i.

(iv) This subautomaton is reversible. The balls of the universal automaton are reversible. Between every ball, there is only one path in the automaton, by construction. If there exists a letter a that labels two incoming transitions of s_i, this letter is the last one in v_i and there exists a circuit around p_i with a as last letter, which is a constradiction with the reversibility of $\mathcal{D}$. Hence, this subautomaton is co-deterministic; likewise, it is deterministic.

(v) Conclusion of the proof. For every constituent of $\mathcal{A}$, we prove that there exists a subautomaton of the universal automaton, with a loop complexity not greater than the star height of the language, and that accepts every word accepted by the constituent. The superposition of all these subautomata of the universal automaton gives a subautomaton of the universal automaton that recoginzes the language. More, every ball intersected by one of these subautomata is entirely included in the subautomaton, hence, the loop complexity of the superposition is not greater than the maximal loop complexity of the superposed automata. Therefore the superposition is a subautomaton of the universal automaton that have a minimal loop complexity for the language.

Moreover, as every superposed automaton is reversible, the superposition is a quasi-reversible automaton. That proves that, for every reversible language, there exists a quasi-reversible automaton, with minimal loop complexity, and that is a subautomaton of the universal automaton.

8 Conclusion

The aim of this paper is to show the soundness of the notion of universal automaton and its various applications. Its large size leads to algorithms with poor complexity, but it is a good theoretical framework to state different kinds of problems on regular languages.

We end this survey with an open question about star height. We have said that, roughly speaking, the universal automaton of a language contains every automaton that accepts this language. This is true up to morphic image, but morphisms do not preserve loop complexity. This is the reason why in the general case, we do not know how to prove the following extension of Theorems 7.10 and 7.19. *The universal automaton of a regular language contains a subautomaton with a minimal loop complexity for this language.*

The universal automaton has not revealed all its secrets.

References

[ADN92] André Arnold, Anne Dicky, and Maurice Nivat. A note about minimal non-deterministic automata. *Bulletin of the EATCS*, 47:166–169, 1992.

[Car70] Christian Carrez. On the minimalization of non-deterministic automaton. Technical report, Computing Laboratory of the Science Faculty of Lille University, 1970.

[Chr86] Marek Chrobak. Finite automata and unary languages. *Theor. Comput. Sci.*, 47(3):149–158, 1986.

[CNP91] Bruno Courcelle, Damian Niwiński, and Andreas Podelski. A geometrical view of the determinization and minimization of finite-state automata. *Mathematical Systems Theory*, 24(2):117–146, 1991.

[Coh70] Rina S. Cohen. Star height of certain families of regular events. *J. Comput. Syst. Sci.*, 4(3):281–297, 1970.

[Con71] John H. Conway. *Regular Algebra and Finite Machines.* Chapman and Hall, London, 1971.

[DS66] Françoise Dejean and Marcel Paul Schützenberger. On a question of eggan. *Information and Control*, 9(1):23–25, 1966.

[Egg63] L. C. Eggan. Transition graphs and the star-height of regular events. *Michigan Math. J.*, 10:385–397, 1963.

[GKP06] Igor Grunsky, Oleksiy Kurganskyy, and Igor Potapov. On a maximal nfa without mergible states. In Dima Grigoriev, John Harrison, and Edward A. Hirsch, editors, *CSR*, volume 3967 of *Lecture Notes in Computer Science*, pages 202–210. Springer, 2006.

[Has83] Kosaburo Hashiguchi. Representation theorems on regular languages. *J. Comput. Syst. Sci.*, 27(1):101–115, 1983.

[Has88] Kosaburo Hashiguchi. Algorithms for determining relative star height and star height. *Inf. Comput.*, 78(2):124–169, 1988.

[HH79] Kosaburo Hashiguchi and Namio Honda. The star height of reset-free events and strictly locally testable events. *Information and Control*, 40(3):267–284, 1979.

[JR93] Tao Jiang and Bala Ravikumar. Minimal nfa problems are hard. *SIAM J. Comput.*, 22(6):1117–1141, 1993.

[Kir05] Daniel Kirsten. Distance desert automata and the star height problem. *Theor. Inform. Appl.*, 39(3):455–509, 2005.

[Kor81] A. D. Korshunov. The number of monotone Boolean functions. *Problemy Kibernet.*, 38:5–108, 272, 1981.

[KW70] Tsunehiko Kameda and Peter Weiner. On the state minimization of nondeterministic finite automata. *IEEE Trans. Computers*, C-19(7):617–627, 1970.

[Lom01] Sylvain Lombardy. *Approche structurelle de quelques problèmes de la théorie des automates.* PhD thesis, ENST, Paris, 2001.

[Lom02] Sylvain Lombardy. On the construction of reversible automata for reversible languages. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo, editors, *ICALP*, volume 2380 of *Lecture Notes in Computer Science*, pages 170–182. Springer, 2002.

[Lom07] Sylvain Lombardy. On the size of the universal automaton of a regular language. In Wolfgang Thomas and Pascal Weil, editors, *STACS*, volume 4393 of *Lecture Notes in Computer Science*, pages 85–96. Springer, 2007.

[LS02] Sylvain Lombardy and Jacques Sakarovitch. Star height of reversible languages and universal automata. In *LATIN 2002: Theoretical informatics (Cancun)*, volume 2286 of *Lecture Notes in Comput. Sci.*, pages 76–90, Berlin, 2002. Springer.

[LS03] Sylvain Lombardy and Jacques Sakarovitch. On the star height of rational languages: a new presentation for two old results. In *Words, languages & combinatorics, III (Kyoto, 2000)*, pages 266–285. World Sci. Publ., River Edge, NJ, 2003.

[McN67] Robert McNaughton. The loop complexity of pure-group events. *Information and Control*, 11:167–176, 1967.

[MP95] O. Matz and A. Potthoff. Computing small finite nondeterministic automata. In *Proc. of the Workshop on Tools and Algorithms for Construction and Analysis of Systems*, BRICS Note Series, pages 74–88, Aarhus, 1995. BRICS.

[Pin92] Jean-Eric Pin. On reversible automata. In Imre Simon, editor, *LATIN*, volume 583 of *Lecture Notes in Computer Science*, pages 401–416. Springer, 1992.

[Pol05] Libor Polák. Minimalizations of nfa using the universal automaton. *Int. J. Found. Comput. Sci.*, 16(5):999–1010, 2005.

[Reu79] Christophe Reutenauer. Sur les variétés de langages et de monoídes. In Klaus Weihrauch, editor, *Theoretical Computer Science*, volume 67 of *Lecture Notes in Computer Science*, pages 260–265. Springer, 1979.

[Sak03] Jacques Sakarovitch. *Eléments de théorie des automates.* Vuibert, Paris, 2003. In French, English translation: *Elements of Automata Theory*, Cambridge University Press, to appear.

[Wie91] Doug Wiedemann. A computation of the eighth Dedekind number. *Order*, 8(1):5–6, 1991.