

Theoretical study and implementation of the canonical automaton

Jean-Marc Champarnaud and Fabien Coulon
Université de Rouen, LIFAR
F-76821 Mont-Saint-Aignan, France

Abstract

We can represent the canonical automaton of a language as the smallest automaton which contains any other automaton recognizing this language, providing equivalent states are merged. Indeed, the canonical automaton appears to be a good representative element in the equivalence class of non-deterministic automata recognizing a given language. Our aim is to provide a detailed description of the canonical automaton based on the notions of syntactical rectangle and characteristic event. In our approach, a state of the canonical automaton of a language \mathcal{L} is associated with a rectangle $(L, R) \subseteq \Sigma^* \times \Sigma^*$, which is maximal w.r.t. the property $L.R \subseteq \mathcal{L}$. We explicit the link with other characterizations, like considering a state as a residual intersection which was given by Arnold *et al.*, and the fundamental automaton defined by Matz and Potthoff. In particular, we pretend that the construction of the canonical automaton has the same time complexity as the construction of the fundamental automaton. Our last section briefly discusses the problem of searching minimal NFAs using the canonical automaton.

Keywords: Canonical / fundamental automaton, nondeterministic automata, NFA minimization, grid cover, characteristic event

1 Introduction

The set of automata which recognize a given language is an equivalence class for the usual equivalence relation over automata. Considering deterministic automata, each equivalence class owns a natural representative element, unique up to an isomorphism, that is, the residual automaton (see Definition 10). Moreover, if the associated language is regular, the residual automaton is the unique minimal automaton w.r.t. number of states.

Considering nondeterministic automata, the problem of associating to each class a unique representative element is less straightforward. Given a regular language \mathcal{L} , there may exist several non-isomorphic NFAs (Nondeterministic Finite Automata) which are minimal w.r.t. number of states, so that no unique representative may be found in this way. The solution given by Calude in [4] is

to cut the equivalence classes associated with languages into sub-classes. Indeed, two automata are said to be equivalent if they bisimulate each other. For each associated sub-class, there exists a minimal automaton in number of states, unique up to an isomorphism.

An other solution, is the so called *canonical automaton* first defined by Christian Carrez in [5]. Each language, regular or not, is associated with a unique canonical automaton. For several reasons, the properties of the canonical automaton of \mathcal{L} in the class of nondeterministic automata are similar to the properties of the residual automaton of \mathcal{L} in the class of deterministic automata, which is discussed in the conclusion.

A brief history of the canonical automaton

It seems that the canonical automaton first appeared in 1970 in a report of Christian Carrez[5] that deals with the minimization of non deterministic automata. The next publication is due to Arnold, Dicky and Nivat[2], who give in a few pages the definitions, the main properties of the canonical automaton, and its theoretical issue.

Let $\mathcal{L} \subseteq \Sigma^*$ be a language. Arnold *et al.* define a function $\phi_{\mathcal{L}}$: for any language $K \subseteq \Sigma^*$,

$$\phi_{\mathcal{L}}(K) = \{u \in \Sigma^* \mid uK \subseteq \mathcal{L}\}$$

The states of the canonical automaton $\mathcal{C}_{\mathcal{L}}$ are the subsets $\phi_{\mathcal{L}}(P)$ — for all $P \in \Sigma^*$ — such that neither P nor $\phi_{\mathcal{L}}(P)$ are empty. A state $\phi_{\mathcal{L}}(P)$ is initial if $\varepsilon \in \phi_{\mathcal{L}}(P)$ and it is final if $\phi_{\mathcal{L}}(P) \subseteq \mathcal{L}$. Finally, the transition $\phi_{\mathcal{L}}(P) \xrightarrow{a} \phi_{\mathcal{L}}(P')$ exists if and only if $\phi_{\mathcal{L}}(P)a \subseteq \phi_{\mathcal{L}}(P')$. This way, each state of the canonical automaton is equal to its own left language; we prove it in Theorem 4.

Afterwards, Arnold *et al.* give the main properties of the canonical automaton. Among the most noteworthy, the canonical automaton of a regular language contains every minimal NFA of its language. But the essential property is the following one: an automaton \mathcal{A} recognizes a sub-language of \mathcal{L} if and only if there exists a morphism from \mathcal{A} into $\mathcal{C}_{\mathcal{L}}$. We give the proof in Section 3.3. In addition, the canonical automaton is minimal for this property in the sense that any surjective morphism from a part of $\mathcal{C}_{\mathcal{L}}$ onto an automaton that recognizes a sub-language of \mathcal{L} is an isomorphism.

Next, the paper of Arnold, Dicky and Nivat[2] links to the one of Courcelle, Niwinski and Podelski[6]. The latter is about syntactical relations and their rectangular decompositions: two words u and v are in syntactical relation if and only if $u.v \in \mathcal{L}$. A rectangular decomposition of this relation is a cover of its graph with syntactical rectangles, that is, Cartesian products in the form of $H \times P$ where¹ H and P are subsets of Σ^* . This second paper does not directly deal with the NFA minimization neither with the canonical automata, but it describes a natural way to link automata to rectangular decompositions: each state of an automaton is bijectively associated with the syntactical rectangle $H \times P$ where H is its left language and P is its right language. Nevertheless, this link is sensitive since a rectangular decomposition can't be systematically

¹To keep Arnold et al.'s notation: H for history and P for prophecy.

associated with an automaton. Courcelle *et al.* give a criterion to determine whether this association can be done. Anyway, in our framework, we chose to use a weaker criterion, based on the property of maximality. A rectangular decomposition \mathcal{D} is maximal if for any rectangle $H \times P$ of \mathcal{D} , $H \times P$ can't be increased unless getting out of the syntactical relation.

The paper [2] characterizes the canonical automaton as associated with the rectangular decomposition that is the unique minimum for the order relation that we define over the maximal decompositions in the following way: a decomposition \mathcal{D} is less than a decomposition \mathcal{D}' if any rectangle in \mathcal{D}' is contained in a rectangle of \mathcal{D} . Indeed, this minimum coincides with the rectangular decomposition that contains every maximal rectangle, that we call the *greatest maximal decomposition*: cf. Proposition 6.

Our purpose

This study comprises two distinct parts. We first give an original theoretical description of the canonical automaton by making use of the notion of rectangular decomposition introduced by Courcelle *et al.* This idea, briefly suggested in [2], leads to an interesting characterization of the canonical automaton.

In a second part (Section 4), we introduce the notion of characteristic events, which is inspired by Kameda and Weiner[7]. Their definition of characteristic events is the same as our definition of characteristic *right* events, and we symmetrically define characteristic left events. The set of characteristic left (resp. right) events associated with a language is a set of elementary pairwise disjoint languages such that for any state q in any automaton which recognizes \mathcal{L} , the left (resp. right) language of q can be recovered as a union of left (resp. right) characteristic events. Moreover, if the language is regular, the set of characteristic events is finite, which gives an efficient approach to implement the canonical automaton.

Finally, the problem of extracting a minimal NFA from the canonical automaton is briefly discussed. Our present conclusion is that the canonical automaton may be a good improvement to the minimization method presented by Kameda and Weiner in [7].

2 Definitions and properties

All along this study, we consider a finite alphabet Σ . Here we provide a brief description of the structures used in the following, but the reader may find a more complete presentation of the automata theory and the regular languages for instance in [10].

Definition 1 *An automaton \mathcal{A} over an alphabet Σ is a 5-tuple $\langle Q, \Sigma, \delta, I, F \rangle$ where Q is the set of states, δ is a subset of $Q \times \Sigma \times Q$ whose elements are called transitions of the automaton, and where I and F are subsets of Q , whose elements are respectively called initial states and final states of the automaton.*

A path in \mathcal{A} is a sequence of transitions: $(q_i, a_i, q_{i+1})_{i \in \{0, \dots, n-1\}}$. A path is said to be a successful path if $q_0 \in I$ and $q_n \in F$.

The word $w = a_0.a_1 \dots a_{n-1} \in \Sigma^n$ is called the label of the path $(q_i, a_i, q_{i+1})_{i \in \{0, \dots, n-1\}}$.

Definition 2 The language recognized by \mathcal{A} is the set of words that are labels of successful paths.

The automaton \mathcal{A} is *finite* if Q is a finite set.

Definition 3 The set of regular languages is the finite language set closure by concatenation, star and union.

Theorem 1 (Kleene[8]) A language is recognized by a finite automaton if and only if it is regular.

Notation We may consider δ as a function from $Q \times \Sigma$ into $\mathcal{P}(Q)$ with $\delta(q, a) = \{r \in Q \mid (q, a, r) \in \delta\}$.

Definition 4 The left language of a state q is the set of words w such that there exists a path in \mathcal{A} whose first state is initial, whose last state is q , and whose label is w . Symmetrically, the right language of a state q is the set of words w such that there exists a path in \mathcal{A} whose first state is q , whose last state is final, and whose label is w .

Denote by $\mathcal{L}_l(q)$ the left language of q , and by $\mathcal{L}_r(q)$ its right language.

Definition 5 The reverse automaton of \mathcal{A} , denoted by $\overline{\mathcal{A}}$, is the 5-tuple $\langle Q, \Sigma, \overline{\delta}, F, I \rangle$ where $(q, a, q') \in \overline{\delta}$ is equivalent to $(q', a, q) \in \delta$.

The reverse of a word $u_0u_1 \dots u_{n-1} \in \Sigma^n$ is the word $u_{n-1}u_{n-2} \dots u_0$, and the reverse language of a given language \mathcal{L} is the set of words whose reverse is in \mathcal{L} . An immediate property of the reverse automaton is that it recognizes the reverse language of the language recognized by \mathcal{A} .

Definition 6 The automaton \mathcal{A} is said to be *deterministic* if it has a unique initial state and if for any $q \in Q$ and any $a \in \Sigma$, $\delta(q, a)$ contains at most one element.

A finite automaton is called NFA for 'nondeterministic finite automaton', while a deterministic finite automaton is also called DFA. In this study, DFAs are considered as a sub-class of NFAs. For this reason, a NFA may be deterministic: we just don't care whether it is or not.

Definition 7 Let \mathcal{A} be an automaton. It is said to be *complete* if for any $q \in Q$ and any $a \in \Sigma$, there exists at least one transition proceeding from q and labeled by a .

Definition 8 An automaton \mathcal{A} is said to be *minimized* (resp. *co-minimized*) if any two distinct states of \mathcal{A} systematically have distinct right languages (resp. left languages).

Definition 9 We denote by $w^{-1}\mathcal{L}$ the set of words $u \in \Sigma^*$ such that $wu \in \mathcal{L}$, and symmetrically, we denote by $\mathcal{L}w^{-1}$ the set of words $u \in \Sigma^*$ such that $uw \in \mathcal{L}$. We may now define the set of residual languages of \mathcal{L} , that is, $\{w^{-1}\mathcal{L} \mid w \in \Sigma^*\}$.

Definition 10 The residual automaton \mathcal{A} is defined as follows: $\mathcal{A} = \langle Q, \Sigma, \delta, I, F \rangle$ where Q is the set of residual languages of \mathcal{L} , $I = \{\mathcal{L}\}$, $F = \{R \in Q \mid \varepsilon \in R\}$, and $(q, a, q') \in \delta$ if and only if $a^{-1}q = q'$.

It can be easily proved that the residual automaton is deterministic and minimal w.r.t. number of states among the DFAs recognizing \mathcal{L} . Indeed, we have the following result:

Proposition 1 Let \mathcal{A} be a DFA that recognizes \mathcal{L} . The minimized automaton of \mathcal{A} is isomorphic to the residual automaton of \mathcal{L} .

As a consequence, the residual automaton is also called the *minimal DFA* of \mathcal{L} . We shall notice that co-minimization has no effect on deterministic automata since their left languages are pairwise disjoint and then distinct.

Definition 11 Let \mathcal{L} be a regular language. An NFA that recognizes \mathcal{L} is said to be *minimal* if it is minimal in number of states among automata which recognize \mathcal{L} .

A minimal NFA is necessarily minimized and co-minimized since minimization and co-minimization are two operations that reduce the number of states. Minimal NFAs of \mathcal{L} are smaller than its minimal DFA, and the figure 1 shows a classical example where the NFA is strictly smaller. Indeed, the minimal NFA can be exponentially smaller than the minimal DFA: for instance, the minimal NFA of the language $(a+b)^*.a.(a+b)^k$ has $k+2$ states, while its minimal DFA has 2^{k+1} states.

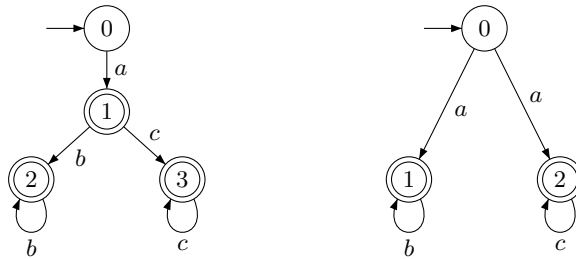


Figure 1: Minimal DFA and minimal NFA of the language $a(b^* + c^*)$.

Definition 12 (Automata morphisms) Let $\mathcal{A}, \mathcal{A}'$ be two automata and let h be a function from $Q_{\mathcal{A}}$ into $Q_{\mathcal{A}'}$. The function h is a morphism if for all

q in I_A (resp. F_A), $h(q) \in I_{A'}$ (resp. $F_{A'}$) and if for all transitions from q to p labeled by a letter a in \mathcal{A} , there exists a transition from $h(q)$ to $h(p)$ labeled by a in \mathcal{A}' .

The following result illustrates the point of automata morphisms in dealing with minimal NFAs.

Theorem 2 *Let \mathcal{L} be a regular language on Σ , and let \mathcal{A} be one of its minimal NFAs. Any morphism from \mathcal{A} into an automaton that recognizes \mathcal{L} is injective.*

Theorem 3 (Brzowski [3]) *Let \mathcal{L} be a regular language and let A be a DFA recognizing \mathcal{L} . Then the determinized automaton of \overline{A} is isomorphic to the minimal DFA of $\overline{\mathcal{L}}$.*

3 Automata geometrical description

The geometrical approach to automata in the space $\Sigma^* \times \Sigma^*$ that was given by Courcelle, Niwinski and Podelski[6] makes it possible to give a more concrete idea of the canonical automaton.

Therefore, the bulk of the definitions involved in this section are inspired by [6]. But we introduce the notion of maximal rectangular decomposition which enables us to throw light on the link between the canonical automaton and the rectangular decompositions that was pointed out by [2].

3.1 Syntactical relation and rectangular decomposition

In this particular representation, we shall consider a slightly different characterization of the language recognized by an automaton:

Proposition 2 *An automaton $\mathcal{A} = (Q, \Sigma, \delta, I, F)$ recognizes the language \mathcal{L} if and only if for any couple (u, v) in $\Sigma^* \times \Sigma^*$ we have*

$$u.v \in \mathcal{L} \iff (\exists q \in Q) u \in \mathcal{L}_i(q), v \in \mathcal{L}_r(q)$$

Definition 13 *Let \mathcal{L} be an arbitrary language on the alphabet Σ . We define the syntactical relation of \mathcal{L} on Σ^* as follows: two words u and v are in relation if and only if $u.v \in \mathcal{L}$.*

So, the syntactical relation of a language is a subset of the space $\Sigma^* \times \Sigma^*$. In the following, to put the emphasis on the geometrical aspect of certain properties on languages and the associated automata, we may sometimes call this relation the *syntactical graph*.

Definition 14 *A Cartesian product $A \times B$ where A and B are subsets of Σ^* is called a *syntactical rectangle*.*

Definition 15 Given a language \mathcal{L} in Σ^* , a set $\{(L_i, R_i) | i \in I\}$ is said to be a rectangular decomposition if it is formed of syntactical rectangles whose union is exactly equal to the syntactical graph of \mathcal{L} .

Regarding these last definitions, we can rewrite the Proposition 2 in order to connect automata to rectangular decompositions:

Corollary 1 Let \mathcal{L} be a language over the alphabet Σ and let $\mathcal{A} = \langle Q, \Sigma, \delta, I, F \rangle$ be an automaton. The two following properties are equivalent:

1. The automaton \mathcal{A} recognizes the language \mathcal{L} .
2. The set $\{(\mathcal{L}_l(q), \mathcal{L}_r(q))\}_{q \in Q}$ is a rectangular decomposition of \mathcal{L} .

Notation . Let's denote by $\mathcal{D}_{\mathcal{A}}$ the rectangular decomposition $\{(\mathcal{L}_l(q), \mathcal{L}_r(q))\}_{q \in Q}$.

We have just shown that a rectangular decomposition may be associated with any automaton. As we can expect, some properties on automata are associated with geometrical properties on rectangular decompositions. Let's examine the example of determinism:

Definition 16 (Courcelle et al. 1991) A rectangular decomposition $\{(L_i, R_i)\}_{i \in I}$ is said to be deterministic if $L_i \cap L_j = \emptyset$ as soon as $i \neq j$.

Proposition 3 (Courcelle et al. 1991) An automaton \mathcal{A} is deterministic if and only if its associated rectangular decomposition $\mathcal{D}_{\mathcal{A}}$ is deterministic.

Actually, an automaton is deterministic if and only if its left languages are pairwise disjoint.

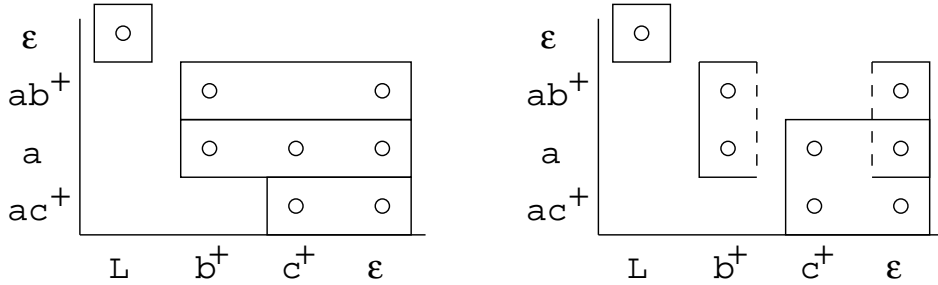


Figure 2: Two rectangular decompositions for the syntactical relation of the language $L = a(b^* + c^*)$. Left languages are on the vertical axis. The first decomposition is deterministic. They are associated with automata in Figure 1.

The Corollary 1 enables us to get a rectangular decomposition from an automaton. However, the reverse is not systematically possible: there exist decompositions that can't be obtained from an automaton in the last way. The class of decompositions for which it is possible is characterized in [6].

3.2 The maximal rectangular decompositions

Definition 17 Let (L, R) be a syntactical rectangle of the syntactical relation \mathcal{R} associated with a given language \mathcal{L} . The syntactical rectangle (L, R) is said to be maximal if for all $(L', R') \in \Sigma^* \times \Sigma^*$ we have:

$$L \times R \subseteq L' \times R' \subseteq \mathcal{R} \implies (L', R') = (L, R)$$

A rectangular decomposition \mathcal{D} is said to be maximal if it is formed of maximal rectangles.

Proposition 4 Given a syntactical rectangle (L, R) , the following conditions are equivalent:

1. The syntactical rectangle (L, R) is maximal.
2. $R = L^{-1}\mathcal{L}$ and $L = \mathcal{L}R^{-1}$

Proposition 5 Let (L_1, R_1) and (L_2, R_2) be two maximal syntactical rectangles, we have:

$$aR_2 \subseteq R_1 \iff L_1a \subseteq L_2$$

Proof. Let us prove that $aR_2 \subseteq R_1 \implies L_1a \subseteq L_2$; the reverse is symmetrical. Assume that $aR_2 \subseteq R_1$, and let $w \in L_1$. We have $wR_1 \subseteq \mathcal{L}$, and thus, $waR_2 \subseteq \mathcal{L}$, which is the same as $wa \in L_2$. ■

The results we develop here are to define the canonical automaton (cf. Theorem 4). But before, we examine the link between the canonical automaton and maximal decompositions as it was pointed out by [2].

Denote by \preceq the relation defined on rectangular decompositions in the following way: if \mathcal{D} and \mathcal{D}' are two rectangular decompositions of one syntactical relation \mathcal{R} , then $\mathcal{D} \preceq \mathcal{D}'$ if and only if each rectangle in \mathcal{D}' is contained in a rectangle of \mathcal{D} .

According to Arnold *et al.*, the canonical automaton is associated with the decomposition that is the minimum of the relation \preceq , but this relation is not an order relation over the set of rectangular decompositions. Although, we have the following result:

Proposition 6 The relation \preceq is an order relation on the set of maximal rectangular decompositions, and for any two maximal rectangular decompositions \mathcal{D} and \mathcal{D}' , we have

$$\mathcal{D} \preceq \mathcal{D}' \iff \mathcal{D}' \subseteq \mathcal{D}$$

In particular, \preceq admits a unique minimum that is also the greatest maximal decomposition.

Proof. Let \mathcal{D} and \mathcal{D}' be two maximal rectangular decompositions such that $\mathcal{D} \preceq \mathcal{D}'$, and let (L', R') be a rectangle of \mathcal{D}' . There exists a rectangle $(L, R) \in \mathcal{D}$ containing (L', R') , but since (L', R') is maximal, we get $(L', R') = (L, R)$. Yet we have proved that $\mathcal{D}' \subseteq \mathcal{D}$ and the reciprocal implication is straightforward.

At last, the uniqueness of the greatest maximal decomposition is not a problem: it is the decomposition which contains every maximal rectangle. ■

The following result becomes obvious when rewritten by means of the relation \preceq :

Corollary 2 *Let \mathcal{L} be a language on the alphabet Σ and let \mathcal{D} be the greatest maximal rectangular decomposition of \mathcal{L} . Then any syntactical rectangle of \mathcal{L} is contained in a rectangle of \mathcal{D} .*

3.3 The canonical automaton

Definition 18 *Let \mathcal{L} be a language on the alphabet Σ and let \mathcal{D} be the greatest maximal rectangular decomposition of \mathcal{L} . We define the canonical automaton associated with \mathcal{L} by $\mathcal{C}_{\mathcal{L}} = \langle \mathcal{D}, \Sigma, \delta, I, F \rangle$ where: for two given states (L_1, R_1) , (L_2, R_2) and any letter $a \in \Sigma$, we let*

$$\begin{aligned} (L_2, R_2) \in \delta((L_1, R_1), a) &\iff a.R_2 \subseteq R_1 \\ &\iff L_1 a \subseteq L_2 \end{aligned}$$

then, a state (L, R) is initial if $\varepsilon \in L$, and it is final if $\varepsilon \in R$.

In accordance with Arnold *et al.*, we denote the canonical automaton of \mathcal{L} by $\mathcal{C}_{\mathcal{L}}$.

Theorem 4 *The canonical automaton $\mathcal{C}_{\mathcal{L}}$ recognizes \mathcal{L} and for any state $(L, R) \in \mathcal{D}$, L is its left language, and R is its right language.*

Proof. We first prove that for any state (L, R) of $\mathcal{C}_{\mathcal{L}}$, L is its left language; the proof for R is symmetrical.

Let's proceed by induction on n : we have to prove that $L \cap \Sigma^n = \mathcal{L}_l((L, R)) \cap \Sigma^n$ ($\forall (L, R) \in \mathcal{D}$) for all n .

For $n = 0$, we just have to notice

$$\varepsilon \in L \Leftrightarrow (L, R) \text{ is initial} \Leftrightarrow \varepsilon \in \mathcal{L}_l((L, R))$$

Assume that the hypothesis is true for n .

Let $(L, R) \in \mathcal{D}$ and $w \in L \cap \Sigma^{n+1}$, $w = w_1 a$ ($a \in \Sigma$).

Then there exists an element of \mathcal{D} in the form of (La^{-1}, R') . Actually, (La^{-1}, aR) satisfies $(La^{-1})(aR) \subseteq \mathcal{L}$, so, there exists a state $(L', R') \in \mathcal{D}$ where $La^{-1} \subseteq L'$ and $aR \subseteq R'$ (cf. Corollary 2). Now, let $w \in L'$, so that $La^{-1} \cup \{w\} \subseteq L'$. We have $(La^{-1} \cup \{w\})(aR) \subseteq \mathcal{L}$, so, $waR \subseteq \mathcal{L}$, then $wa \in L$ by maximality and finally $w \in La^{-1}$. This proves $L' = La^{-1}$. So we have $(La^{-1}, R') \xrightarrow{a} (L, R)$ since $(La^{-1})a \subseteq L$ and $w_1 \in La^{-1}$.

From the induction hypothesis, we get $w_1 \in \mathcal{L}_l((La^{-1}, R'))$, and thus $w = w_1a \in \mathcal{L}_l((L, R))$.

Reciprocally, let $(L, R) \in \mathcal{D}$ and $w \in \mathcal{L}_l((L, R)) \cap \Sigma^{n+1}$, $w = w_1a$ ($a \in \Sigma$). There exists $(L_1, R_1) \in \mathcal{D}$ such that $(L_1, R_1) \xrightarrow{a} (L, R)$ with $w_1 \in \mathcal{L}_l((L_1, R_1))$. So $w_1 \in L_1$ from the recursion hypothesis. As $L_1a \subseteq L$, we have $w_1a \in L$ and then $w \in L \cap \Sigma^{n+1}$.

We still have to prove that $\mathcal{C}_{\mathcal{L}}$ recognizes \mathcal{L} . Indeed, $(\varepsilon, \mathcal{L})$ belongs to the syntactical relation of \mathcal{L} and thence, is included in a rectangle $(L, R) \in \mathcal{D}$ that is initial and whose right language contains \mathcal{L} . Reciprocally, for all initial state (L, R) , L contains ε and then $R \subseteq \mathcal{L}$ so that the words recognized by $\mathcal{C}_{\mathcal{L}}$ are in \mathcal{L} . ■

Lemma 1 *Let (L_0, R_0) be a syntactical rectangle and let D be the set of maximal rectangles containing (L_0, R_0) . The element $(L, R) \in D$ for which L is minimal is unique and is defined by the relations $R = \{w \in \Sigma^* \mid L_0.w \subseteq \mathcal{L}\}$ and $L = \{w \in \Sigma^* \mid w.R \subseteq \mathcal{L}\}$.*

Proof. Let's prove that this (L, R) is maximal. Let (L', R') be a syntactical rectangle such as $L \times R \subseteq L' \times R' \in D$. Since $L_0 \subseteq L'$, we have $L_0.R' \subseteq \mathcal{L}$ and thence $R' \subseteq R$, so $R' = R$. As a consequence, $L'.R \subseteq \mathcal{L}$, hence $L' \subseteq L$, and finally $L' = L$.

Then we prove that L is minimal. Let $(L', R') \in D$, and show that $L \subseteq L'$. Since $L_0 \subseteq L'$, we necessarily have $L_0.R' \subseteq \mathcal{L}$ and so $R' \subseteq R$. Thence for any word $w \in L$, we have $w.R \subseteq \mathcal{L}$ and then $w.R' \subseteq \mathcal{L}$, i.e. $w \in L'$.

The uniqueness of the couple (L, R) is a consequence of the fact that L is minimal and that the rectangles of D are maximal: there can't exist two maximal rectangles (L_1, R_1) et (L_2, R_2) such that $L_1 = L_2$. ■

Theorem 5 *Let \mathcal{L} be a regular language, and let \mathcal{A} be an automaton that recognizes a subset of \mathcal{L} . There exists a morphism which maps \mathcal{A} into $\mathcal{C}_{\mathcal{L}}$.*

Proof. Denote by \mathcal{D} the greatest maximal rectangular decomposition of \mathcal{L} . For each state q in \mathcal{A} , denote by (L_q, R_q) the element of the following set for which L is minimal:

$$\{(L, R) \in \mathcal{D} \mid \mathcal{L}_l(q) \subseteq L \wedge \mathcal{L}_r(q) \subseteq R\}$$

According to Lemma 1, we know that $R_q = \{w \in \Sigma^* \mid \mathcal{L}_l(q).w \subseteq \mathcal{L}\}$ and $L_q = \{w \in \Sigma^* \mid w.R_q \subseteq \mathcal{L}\}$.

Now define the morphism h from \mathcal{A} into $\mathcal{C}_{\mathcal{L}}$ — remember that \mathcal{D} is the set of states of $\mathcal{C}_{\mathcal{L}}$. For all states q in \mathcal{A} , we let $h(q) = (L_q, R_q)$. We have to prove this is a morphism. If q is an initial state of \mathcal{A} , then $\varepsilon \in \mathcal{L}_l(q)$ so $\varepsilon \in L_q$, and hence $h(q)$ is initial in $\mathcal{C}_{\mathcal{L}}$. In the same way, if q is a final state of \mathcal{A} then $\varepsilon \in \mathcal{L}_r(q)$ so $\varepsilon \in R_q$ and $h(q)$ is final in $\mathcal{C}_{\mathcal{L}}$.

Trickier, let q and q' be two states of \mathcal{A} and let a be a letter such that the transition $q \xrightarrow{a} q'$ exists. We now have to prove that the transition $h(q) \xrightarrow{a} h(q')$ exists in $\mathcal{C}_{\mathcal{L}}$. We have $\mathcal{L}_l(q).a \subseteq \mathcal{L}_l(q')$ by hypothesis. First notice that we have $L_q.a \subseteq L_{q'}$ if and only if for all $w \in \Sigma^*$ the following implication is true:

$$w.R_q \subseteq \mathcal{L} \implies wa.R_{q'} \subseteq \mathcal{L}$$

Indeed, it's enough to verify the inclusion $L_q.a \subseteq L_{q'} \cap (\Sigma^*.a)$ and let's detail the content of these sets: $L_q.a = \{wa \mid w.R_q \subseteq \mathcal{L}\}$ and $L_{q'} \cap (\Sigma^*.a) = \{wa \mid wa.R_{q'} \subseteq \mathcal{L}\}$.

Let $w \in \Sigma^*$ such as $w.R_q \subseteq \mathcal{L}$, i.e. such that

$$(\forall u \mid \mathcal{L}_l(q)u \subseteq \mathcal{L}) \quad wu \in \mathcal{L}. \quad (1)$$

Now let $v \in R_{q'}$ so that $\mathcal{L}_l(q').v \subseteq \mathcal{L}$, then $\mathcal{L}_l(q).av \subseteq \mathcal{L}$ from the first hypothesis. We can then substitute av to u in (1) and we get $wav \in \mathcal{L}$. This ends the proof of $wa.R_{q'} \subseteq \mathcal{L}$. In conclusion, we have $L_q.a \subseteq L_{q'}$ and the searched transition exists. ■

Corollary 3 *Let \mathcal{L} be a regular language. Any minimal NFA recognizing \mathcal{L} is isomorphic to a sub-automaton of $\mathcal{C}_{\mathcal{L}}$.*

Proof. Let \mathcal{A} be a minimal NFA, and let h be a morphism from \mathcal{A} into $\mathcal{C}_{\mathcal{L}}$. The morphism h is injective according to the Theorem 2. ■

4 Implementation of $\mathcal{C}_{\mathcal{L}}$

This section is about implementing the canonical automaton of a regular language. The notions of Reduced Automaton Matrix and Characteristic Event are due in particular to Kameda and Weiner [7]. Our objective is to give an implementable characterization of the canonical automaton based on these objects. Then, the *fundamental automaton* as defined by Matz and Potthoff in [9] appears to be a particular case of our characterization.

4.1 Characteristic events

Let $\mathcal{A} = \langle Q, \Sigma, \delta, 0, F \rangle$ be the minimal DFA of the language \mathcal{L} , where $Q = \{0, \dots, n-1\}$ is the set of states; the initial state is 0. Let $\mathcal{B} = \text{Det}(\overline{\mathcal{A}})$ obtained from the reverse of \mathcal{A} by subset determinization. By Theorem 3, \mathcal{B} is also a minimal DFA. States of \mathcal{B} are arbitrarily ordered and named q_i ($0 \leq i < n_b$). Each q_i is a subset of Q , and q_0 is initial.

As in [7], we define the RAM of the language \mathcal{L} , that is unique up to a permutation of rows and columns:

Definition 19 The reduced automaton map (RAM) associated with \mathcal{L} , denoted M , contains n rows and n_b columns. The element at the intersection of the i^{th} row and the j^{th} column is defined by:

$$M_{i,j} = 1 \text{ if } i \in q_j \\ = 0 \text{ otherwise}$$

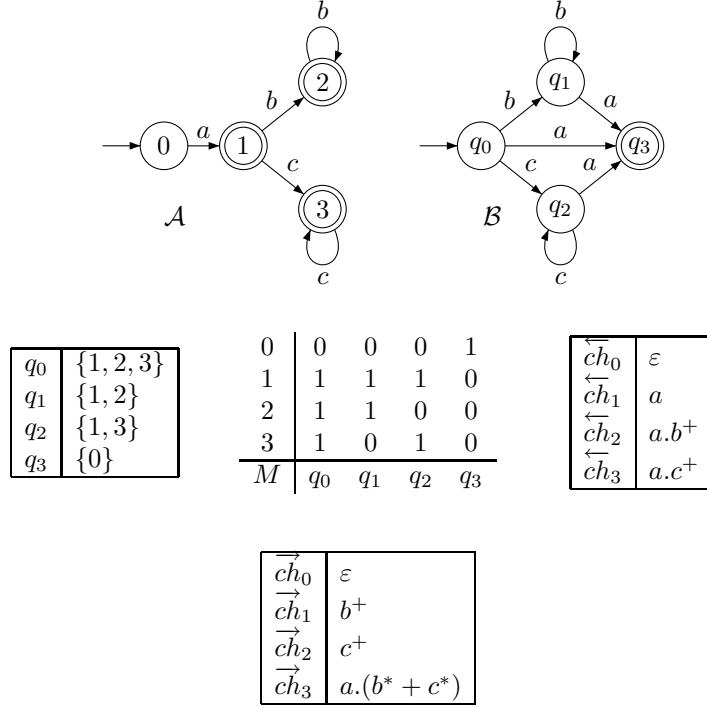


Figure 3: Automata \mathcal{A} and \mathcal{B} for the language $a(b^* + c^*)$, the associated RAM M , and the associated characteristic events.

Definition 20 Characteristic left events of the language \mathcal{L} are the languages $\overleftarrow{ch}_i = \underline{\mathcal{L}}_i^{\mathcal{A}}(i)$ ($0 \leq i < n$), while characteristic right events of \mathcal{L} are the languages $\overrightarrow{ch}_j = \underline{\mathcal{L}}_i^{\mathcal{B}}(q_j)$ ($0 \leq j < n_b$).

The following Proposition sums up some results that can be found in [7]. Since the context is rather different, they are proved once more:

Proposition 7 We have the following results about characteristic events:

1. Left (resp. right) characteristic events of \mathcal{L} are pairwise disjoint.
2. Let A be an automaton that recognizes \mathcal{L} , and let q be a state of A . The left (resp. right) language of q in A is a union of left (resp. right) characteristic events of \mathcal{L} .

3. Let $i \in \{0, \dots, n-1\}$ and $j \in \{0, \dots, n_b-1\}$, we have

$$\overleftarrow{ch}_i \cdot \overrightarrow{ch}_j \subseteq \mathcal{L} \iff M_{i,j} = 1$$

Proof. (1) is trivial since characteristic events are left languages of deterministic automata.

(2): Let A be an automaton that recognizes \mathcal{L} . Consider that the states of \mathcal{A} and \mathcal{B} are obtained from A by subset determinization, so that their states are sets of states of A . Let q be a state of A , the left (resp. right) language of q is the union of characteristic left (resp. right) events associated with the states of \mathcal{A} (resp. \mathcal{B}) which contain q .

(3): Let $i \in \llbracket 0, n-1 \rrbracket$ and $j \in \llbracket 0, n_b-1 \rrbracket$.

Suppose $M_{i,j} = 1$, that is, $i \in q_j$. Since \mathcal{B} is obtained from $\overline{\mathcal{A}}$ by subset determinization, we have $\mathcal{L}_i^{\mathcal{B}}(q_j) \subseteq \mathcal{L}_i^{\overline{\mathcal{A}}}(i)$, that is, $\overrightarrow{ch}_j \subseteq \mathcal{L}_r^{\mathcal{A}}(i)$, hence $\overleftarrow{ch}_i \cdot \overrightarrow{ch}_j \subseteq \mathcal{L}$.

Now suppose that $\overleftarrow{ch}_i \cdot \overrightarrow{ch}_j \subseteq \mathcal{L}$. Since characteristic events are pairwise disjoint, this implies $\overrightarrow{ch}_j \subseteq \mathcal{L}_r^{\mathcal{A}}(i)$, that is, $\mathcal{L}_i^{\mathcal{B}}(q_j) \subseteq \mathcal{L}_i^{\overline{\mathcal{A}}}(i)$. Because of the subset determinization, this implies $i \in q_j$, that is, $M_{i,j} = 1$. ■

Definition 21 A grid in the RAM M is a pair $[l, r]$ where l is a subset of $\{0, \dots, n-1\}$, r is a subset of $\{0, \dots, n_b-1\}$, and for all $i \in l$ and $j \in r$, we have $M_{i,j} = 1$.

A grid $[l, r]$ is said to be a prime grid if for all other grid $[l', r']$ we have

$$l \subseteq l' \wedge r \subseteq r' \implies [l, r] = [l', r']$$

Naturally, each state of the canonical automaton, that is, each maximal syntactical rectangle (L, R) is associated with a prime grid $[l, r]$ such that $\bigcup_{i \in l} \overleftarrow{ch}_i = L$ and $\bigcup_{j \in r} \overrightarrow{ch}_j = R$.

4.2 Computing states of the canonical automaton

Let \mathcal{D} be the set of maximal syntactical rectangles of \mathcal{L} .

Define the functions λ and ρ such that for all states of the canonical automaton $(L, R) \in \mathcal{D}$, $\lambda(L)$ is the subset of $\{0, \dots, n-1\}$ which verifies

$$L = \bigcup_{i \in \lambda(L)} \overleftarrow{ch}_i \tag{2}$$

and $\rho(R)$ is the subset of $\{0, \dots, n_b-1\}$ which verifies

$$R = \bigcup_{j \in \rho(R)} \overrightarrow{ch}_{q_j}$$

An implementation of the canonical automaton may store a state (L, R) either as $\lambda(L)$ or $\rho(R)$, or even as the pair $(\lambda(L), \rho(R))$. We choose the first

solution by letting $\mathcal{Q} = \{\lambda(L) \mid (L, R) \in \mathcal{R}\}$. The elements of \mathcal{Q} are representatives of the states of the canonical automaton. Indeed, a maximal syntactical rectangle (L, R) is fully characterized by $\lambda(L)$ since $\rho(R)$ can be recovered from $\lambda(L)$ as the maximal subset of $\{0, \dots, n_b - 1\}$ such that $[\lambda(L), \rho(R)]$ is a grid.

Notice that R can also be directly recovered from $\lambda(L)$:

Proposition 8 *Let (L, R) be a state of the canonical automaton, we have:*

$$R = \bigcap \mathcal{L}_r^A(i)$$

Proof. Let $w \in \Sigma^*$,

$$w \in R \iff \bigcup_{i \in \lambda(L)} \overleftarrow{ch}_i.w \subseteq \mathcal{L}$$

$$\iff \overleftarrow{ch}_i.w \subseteq \mathcal{L} \quad (\forall i \in \lambda(L))$$

$$\iff w \in \mathcal{L}_r^A(i) \quad (\forall i \in \lambda(L))$$

The sets \overleftarrow{ch}_i are pairwise disjoint, which justify the last equivalence. ■

Since the right languages of \mathcal{A} are the residuals of \mathcal{L} , this Proposition implies that the states of the canonical automaton can also be characterized as residual intersections, which was the original definition given by Arnold *et al.* in [2].

Computing states of $\mathcal{C}(\mathcal{L})$ consists in enumerating prime grids of the RAM M . For this purpose, we propose the following recursive algorithm `GetPrimeGrid`. Let M_j stands for the j^{th} column of M :

```

GetPrimeGrid (  $I$ : subset of  $\{0, \dots, n - 1\}$ ,  $J$ : subset of  $\{0, \dots, n_b - 1\}$ ,
column: integer )
  If  $I = \emptyset$  Then return;
  If  $column = n_b$  Then add the Grid  $[I, J]$ 
  Else
    GetPrimeGrid (  $I \cap M_{column}$ ,  $J \cup \{column\}$ ,  $column + 1$  );
    GetPrimeGrid (  $I, J, column + 1$  );
  End
End
GetPrimeGrid( $\{0, \dots, n - 1\}, \emptyset, 0$  );

```

This algorithm has an $\mathcal{O}(n2^{n_b})$ complexity. Notice that if $n_b > n$, we may transpose the matrix, so that we get an $\mathcal{O}(\max(n, n_b)2^{\min(n, n_b)})$ complexity.

4.3 Computing transitions, initial and final states

Let \mathcal{A}° be the completed automaton of \mathcal{A} to which a sink state denoted by -1 may have been added. We extend characteristic left events by letting $\overleftarrow{ch}_{-1} = \mathcal{L}_i^{\mathcal{A}^\circ}(-1)$.

We have seen that given two states of the canonical automaton (L, R) , (L_1, R_1) and given a letter a , the transition $(L, R) \xrightarrow{a} (L_1, R_1)$ exists if and only if $L.a \subseteq L_1$. We are going to show that this property is easily characterized using representatives of the states in \mathcal{Q} .

Proposition 9 Let (L, R) and (L_1, R_1) be two states of the canonical automaton and $a \in \Sigma$. The transition $(L, R) \xrightarrow{a} (L_1, R_1)$ exists if and only if $\delta_{\mathcal{A}^\circ}(\lambda(L)) \subseteq \lambda(L_1)$.

Proof. In the one hand we have:

$$L.a = \left(\bigcup_{l \in \lambda(L)} \overleftarrow{ch}_l \right).a = \bigcup_{l \in \lambda(L)} (\overleftarrow{ch}_l.a)$$

In the other hand,

$$L_1 = \bigcup_{l \in \lambda(L_1)} \overleftarrow{ch}_l$$

Further, according to the completeness of \mathcal{A}° , we have

$$\overleftarrow{ch}_l.a \subseteq \overleftarrow{ch}_{\delta_{\mathcal{A}^\circ}(l,a)} \quad (\forall l \in \lambda(L))$$

Yet, the family $(\overleftarrow{ch}_i)_{i \in \{-1, 0, \dots, n\}}$ is a partitioning of Σ^* since \mathcal{A}° is deterministic and complete. It follows that $L.a \subseteq L_1$ if and only if

$$\delta_{\mathcal{A}^\circ}(\lambda(L), a) \subseteq \lambda(L_1)$$

■

Let (L, R) be a state of the canonical automaton. Knowing $\lambda(L)$, we shall determine whether it is initial and whether it is final.

Proposition 10 The state (L, R) is initial if and only if $0 \in \lambda(L)$.
It is final if and only if for all $i \in \lambda(L)$, i is a final state of \mathcal{A} .

Proof. Trivial from Equation 2 and Proposition 8.

■

4.4 The fundamental automaton

In this section, we take up the fundamental automaton construction presented by Matz and Potthoff in [9] in order to compare it with the canonical automaton.

Definition 22 The fundamental automaton $\mathcal{F} = \langle Q_{\mathcal{F}}, \Sigma, \delta_{\mathcal{F}}, I_{\mathcal{F}}, F_{\mathcal{F}} \rangle$ can be defined as follows:

1. $Q_{\mathcal{F}} = \{P \subseteq \{0, \dots, n_b - 1\} \mid \bigcap_{j \in P} q_j \neq \emptyset\}$
2. $I_{\mathcal{F}} = \{P \in Q_{\mathcal{F}} \mid (\forall j \in P) q_j \in F_{\mathcal{B}}\}$
3. $F_{\mathcal{F}} = \{P \in Q_{\mathcal{F}} \mid 0 \in P\}$
4. $\delta_{\mathcal{F}}(P, a) = \{P' \in Q_{\mathcal{F}} \mid P' \subseteq \delta_{\overline{\mathcal{B}}}(P, a)\}$ for all P in $Q_{\mathcal{F}}$ and a in Σ .

By replacing the definition of $Q_{\mathcal{F}}$ by $Q_{\mathcal{F}} = \{\rho(R) \mid (L, R) \in \mathcal{D}\}$, one recovers the canonical automaton, where the states (L, R) are represented by $\rho(R)$.

Proposition 11 *Let (L, R) be a state of $\mathcal{C}_{\mathcal{L}}$, we have $\bigcap_{j \in \rho(R)} q_j \neq \emptyset$.*

Proof. Since $[\lambda(L), \rho(R)]$ is a non empty grid, there exists a row i such that $M_{i,j} = 1$ for all $j \in \rho(R)$, which means $i \in q_j$ for all $j \in \rho(R)$. Hence, the intersection is non empty. ■

This proves that the canonical automaton is a sub-automaton of the fundamental automaton. Indeed, \mathcal{F} is bigger than $\mathcal{C}(\mathcal{L})$, though the construction of $\mathcal{C}(\mathcal{L})$ is rather as fast as the construction of \mathcal{F} , since computing $Q_{\mathcal{F}}$ has also an $\mathcal{O}(n2^{n_b})$ time complexity. Let us mention that there exist two fundamental automata, $\mathcal{F}(\mathcal{L})$ and $\mathcal{F}(\overline{\mathcal{L}})$, so that the complexity can also be reduced down to $\mathcal{O}(\max(n, n_b)2^{\min(n, n_b)})$.

5 Searching the minimal NFAs

The canonical automaton is tied with the NFA minimization, since we know from Corollary 3 that every minimal NFA of the language \mathcal{L} is contained in its canonical automaton as a sub-automaton. Hence, searching one or every minimal NFA of \mathcal{L} may consist in parsing sub-automata of $\mathcal{C}_{\mathcal{L}}$ by increasing number of states until we find an automaton which recognizes \mathcal{L} .

One interesting improvement consists in using the canonical automaton in order to speed up the method given by Kameda and Weiner in [7].

Definition 23 *A prime grid cover of the RAM M is a set \mathcal{P} of prime grids which has the following property. For each pair (i, j) such that $M_{i,j} = 1$, there is a prime grid $[l, r] \in \mathcal{P}$ such that $i \in l$ and $j \in r$.*

We know that each state of the canonical automaton (L, R) is associated with a prime grid $[\lambda(L), \rho(R)]$ and conversely.

Proposition 12 *Let $(L_i, R_i)_{0 \leq i < k}$ be a set of states of $\mathcal{C}_{\mathcal{L}}$. If the sub-automaton of $\mathcal{C}_{\mathcal{L}}$ whose states are $\{(L_i, R_i)\}_{0 \leq i < k}$ recognizes \mathcal{L} , then $\{[\lambda(L_i), \rho(R_i)] \mid 0 \leq i < k\}$ is a prime grid cover of M .*

Proof. Suppose $\{[\lambda(L_i), \rho(R_i)] \mid 0 \leq i < k\}$ is not a prime grid cover. There exist two words u and v with $uv \in \mathcal{L}$ such that (u, v) does not belong to any $L_i \times R_i$ ($0 \leq i < k$). Let C be the sub-automaton of $\mathcal{C}(\mathcal{L})$ whose states are $\{(L_i, R_i)\}_{0 \leq i < k}$. It is clear that $L_i^C((L_i, R_i)) \subseteq L_i$ and $L_r^C((L_i, R_i)) \subseteq R_i$ for all i . Hence, there does not exist any state of C such as u is in its left language and v is in its right language, hence C does not recognize the word uv . ■

As a consequence, the parse of sub-automata can be restricted to the sub-automata associated with prime grid covers.

This method is indeed the method of Kameda and Weiner improved in the sense that for every prime grid cover, the original method has to build an equivalent automaton and test whether the resulting automaton recognizes \mathcal{L} . By using the canonical automaton, the automata associated with prime grid covers are given as sub-automata of $\mathcal{C}(\mathcal{L})$ and the test can be carried out directly.

Conclusion

The similarity between the canonical automaton and the residual automaton of a language \mathcal{L} appears naturally in the definitions we have given. The canonical automaton is the automaton associated with the greatest maximal rectangular decomposition of the syntactical relation of \mathcal{L} , while the residual automaton is associated with the greatest maximal *deterministic* rectangular decomposition.

We tried to present a detailed approach of the canonical automaton, wide enough to give an intuitive and coherent idea of what it is. Section 3 derives from the definitions given by Arnold *et al.* in [2]. They have also been used by Amilhastre [1], who details the notion of pairs (L, R) , even though the canonical automaton is not his main purpose, and his study essentially concerns the case of homogeneous languages. In Section 4, we introduce a part of the work of Kameda and Weiner[7], which is indeed close to the notion of canonical automaton, even though their approach is in a way symmetrical: from a given prime grid cover of a Reduced Automaton Map, they compute an associated automaton through an inverse operation of the subset construction, namely, the *intersection rule* method.

Acknowledgement

We would like to thank Benoist Gaston, for some interesting previous discussions on the subject. Thanks are also due to the referees, who read carefully our first version and gave us insightful comments.

References

- [1] J. Amilhastre. *Représentation par automate d'ensemble de solutions de problèmes de satisfaction de contraintes*. PhD thesis, Université Montpellier II, 1999.
- [2] A. Arnold, A. Dicky, and M. Nivat. A note about minimal non-deterministic automata. *Bulletin of the EATCS*, number 47, pages 166–169, June 1992.
- [3] J. A. Brzozowski. Canonical regular expressions and minimal state graphs for definite events. *Mathematical Theory of Automata, MRI Symposia Series*, 12:529–561, 1962.

- [4] Cristian S. Calude, Elena Calude, and Bakhadyr Khossainov. Finite non-deterministic automata: Simulation and minimality. *Theoretical Computer Science*, 242(1–2):219–235, 2000.
- [5] C. Carrez. On the minimalization of non-deterministic automaton. Technical report, Laboratoire de Calcul de la Faculté des Sciences de l’Université de Lille, 1970.
- [6] B. Courcelle, D. Niwinski, and A. Podelski. A geometrical view of the determinization and minimization of finite-state automata. *Mathematical Systems Theory* 24, 1991.
- [7] T. Kameda and P. Weiner. On the state minimization of nondeterministic finite automata. *IEEE Trans. Comp.*, C(19):617–627, 1970.
- [8] S. C. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, pages 2–42, 1996.
- [9] O. Matz and A. Potthoff. Computing small nondeterministic finite automata. *Tools and Algorithms for the Construction and Analysis of Systems – TACAS 95, volume NS-95-2*, pages 74–88, 1995.
- [10] S. Yu. Regular languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, Volume I, Word, Language, Grammar*, pages 41–110. Springer, Berlin, 1997.