ELSEVIER
Theoretical Computer Science 233 (2000) 75–90
Theoretical Computer Science
www.elsevier.com/locate/tcs

Characterization of Glushkov automata

Pascal Caron *, Djelloul Ziadi

Laboratoire d'Informatique de Rouen, Université de Rouen, 76821 Mont-Saint-Aignan, Cédex, France

Received May 1996; revised October 1997
Communicated by M. Crochemore

Abstract

Glushkov's algorithm computes a nondeterministic finite automaton without ε-transitions and with $n+1$ states from a regular expression having n occurrences of letters. The aim of this paper is to give a set of necessary and sufficient conditions characterizing this automaton. Our characterization theorem is formulated in terms of directed graphs. Moreover these conditions allow us to produce an algorithm of conversion of a Glushkov automaton into a regular expression of small size. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Automata theory, Glushkov automaton, Graphs theory, regular expression

1. Introduction

During the last 40 years, the synthesis of automata[1] has stimulated a good deal of research. The taxonomy that Watson [21] has dedicated to this topic enlights two wide categories of algorithms: the first ones yield automata with ε-transitions [20, 14], the second ones, nondeterministic automata without ε-transition [17, 13, 18]. The second approach provides in a "natural" way [6] automata of "small" size [4] currently called Glushkov automata. This construction is implemented in many automata manipulation softwares such as AUTOMATE [10, 9], AMoRE [15, 16] or Automap [7, 8]. It was studied and improved by Chang and Paige [11], Brüggemann-Klein [6], and Ziadi *et al.* [22], who produced quadratic time algorithms on the size of the expression.

The best known property of Glushkov automata is certainly the fact that a given state is always reached using the same letter. However, this property is not sufficient for an automaton to be a Glushkov one. Our aim is to study structural properties of Glushkov automata and to state a characterization of these automata. Beyond its theoretical interest, this study yields an algorithm computing a regular expression from a

---

* Corresponding author.
E-mail address: caron@dir.univ-rouen.fr (P. Caron)

[1] Better known today under the name of conversion of regular expressions into automata.

PII: S0304-3975(97)00296-X

Glushkov automaton. This expression is particularly "short" (it contains $n-1$ occurrences of letters if the initial Glushkov automaton has n states). Let us notice that our characterization is entirely formulated in terms of graphs. Section 2 aims at defining the notion of Glushkov automaton of an expression. The third section gathers definitions and notations we use for the characterization in terms of graphs. Section 4 enumerates necessary conditions for an automaton to be the Glushkov automaton of an expression. The last section establishes the characterization theorem.

2. Glushkov automata

This section introduces notations used in this paper. General references concerning automata are [1, 3, 12, 14].

Kleene's theorem asserts that, given a regular expression, there exists a finite automaton recognizing the language that it defines. This automaton can be chosen without ε-transitions and nondeterministic. We are interested by a particular class of recognizers: Glushkov automata. These automata are computed by an algorithm whose best known variants are due to Glushkov [13], and McNaughton and Yamada [17]. This algorithm will be called "Glushkov's algorithm" and we refer the reader to the papers of Berry and Sethi [2], Berstel [4], Brüggeman-Klein [6], Berstel and Pin [5] for descriptions and analyses of this algorithm. Our notation follows [19].

In order to specify their position in the expression, symbols are subscripted following the order of reading. For example, starting from $E = (a + \varepsilon).b.a$, one obtains the subscripted expression $\overline{E} = (a_1 + \varepsilon)\cdot b_2 \cdot a_3$. The set of positions for an expression E is denoted by $Pos(E)$. If F is a subexpression of E, we denote by $\mathrm{Pos}_E(F)$ the subset[2] of positions of E which are symbols of F. We shall write $\mathrm{Pos}(F)$ for $\mathrm{Pos}_E(F)$ whenever it is not ambiguous. We denote by χ the application which maps each position in $Pos(E)$ to the symbol of Σ which appears at this position in E.

In order to construct a nondeterministic finite automaton recognizing $L(E)$, Glushkov defines the following sets: $First(E)$ is the set of initial positions of words of $L(E)$, $Last(E)$ is the set of final positions of words of $L(E)$, and $Follow(E,x)$ is the set of positions which immediately follow the position[3] x in E.

Formally $First(E)$ is defined by induction according to the following rules:

$$\begin{aligned}
First(\emptyset) &= First(\varepsilon) = \emptyset\\
First(x) &= \{x\}\\
First(F+G) &= First(F) \cup First(G)\\
First(F\cdot G) &= \begin{cases} First(F) & \text{if } \varepsilon \notin L(F)\\ First(F) \cup First(G) & \text{if } \varepsilon \in L(F)\end{cases}\\
First(E^*) &= First(E^+) = First(E)
\end{aligned}$$

---

[2] For $E = F + G$ and $E = F \cdot G$, $Pos(F) \cap Pos(G) = \emptyset$; for E^*, $Pos(E^*) = Pos(E)$.

[3] $x \notin Pos(E) \Rightarrow Follow(E,x) = \emptyset$.

In order to obtain rules for $Last(E)$ substitute "*Last*" for "*First*" and replace the last but one rule by

$$Last(F\cdot G) = \begin{cases} Last(G) & \text{if } \varepsilon \notin L(G) \\ Last(F) \cup Last(G) & \text{if } \varepsilon \in L(G) \end{cases}$$

The set $Follow(E,x)$ can be inductively computed as follows:

$$Follow(\varepsilon,x) = Follow(\emptyset,x) = Follow(a,x) = \emptyset$$

$$Follow(F+G,x) = \begin{cases} Follow(F,x) & \text{if } x \in Pos(F) \\ Follow(G,x) & \text{if } x \in Pos(G) \end{cases}$$

$$Follow(F\cdot G,x) = \begin{cases} Follow(F,x) & \text{if } x \in Pos(F) \setminus Last(F) \\ Follow(F,x) \cup First(G) & \text{if } x \in Last(F) \\ Follow(G,x) & \text{if } x \in Pos(G) \end{cases}$$

$$Follow(E^+,x) = \begin{cases} Follow(F,x) & \text{if } x \in Pos(F) \setminus Last(F) \\ Follow(F,x) \cup First(F) & \text{if } x \in Last(F) \end{cases}$$

$$Follow(E^*,x) = Follow(E^+,x)$$

The Glushkov automaton of E is the automaton $M_E = (Q, \Sigma, \{0\}, F, \delta)$ where the set of states is $Q = Pos(E) \cup \{0\}$ for some element 0 not in $Pos(E)$, the set of final states is $F = Last(E)$ if $\varepsilon \notin L(E)$ and $Last(E) \cup \{0\}$ otherwise, and the set of transitions is $\delta = \{\ (x, \chi(y), y) \mid x \in Pos(E) \text{ and } y \in Follow(E,x)\} \cup \{\ (0, \chi(y), y) \mid y \in First(E)\}$.

Proposition 2.1. *For every regular expression* E, $L(E) = L(M_E)$. □

Notice that the Glushkov automaton of an expression E which does not contain a reference to the empty set is a trim automaton (i.e. every state is on a path from the initial state to some final state).

An automaton is *homogeneous* if for all (p,a,q), $(p',a',q') \in \delta$, $q = q' \Rightarrow a = a'$.

It is easy to see that the Glushkov automaton of an expression E is homogeneous. Consequently, this automaton can be seen as a vertex-labeled directed graph $G_E = (X, U)$ where (1) the set X of vertices is the set made up with $Pos(E)$, the state 0 called "root", and in the case where $F \neq \emptyset$, the state Φ called "antiroot" (the state Φ is introduced in order to "forget" the final states); (2) the set U of directed edges is made up with the pairs (x, y) such that $(x, \chi(y), y) \in \delta$ or $x \in F$ and $y = \Phi$; and (3) the function χ assigns the label $\chi(x) \in \Sigma \cup \{\varepsilon\}$ to each vertex x ($\chi(\Phi) = \chi(0) = \varepsilon$). This graph is called the Glushkov graph of the expression E. We can notice that the only Glushkov graph with a unique vertex is the graph $G_\emptyset$ associated to the empty set. Moreover $G_\emptyset$ has no loop.

3. Graph properties

Let $G = (X, U)$ be a graph. A *hammock* is a graph with the following properties. If G has a unique vertex, then it has no loop, otherwise G has two distinguished vertices

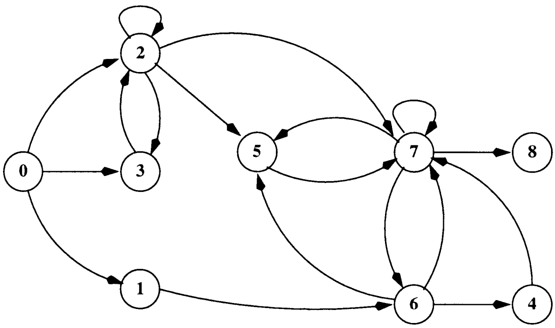


Fig. 1. The hammock T includes eight orbits.

i and t such that for any vertex x of X, (1) there exists a path from i to t going through x, (2) there is no path from t to x nor from x to i. One can notice that a hammock has a unique root (the vertex i) and a unique antiroot (the vertex t); these vertices are distinct and have no loop. A hammock is connected and is not strongly connected.

$\mathcal{O} \subseteq X$ is an *orbit* of G if and only if for all x and x' in $\mathcal{O}$ there exists a non trivial path from x to x'.

An orbit is *maximal*, if for each vertex x of $\mathcal{O}$ and for each vertex x' out of $\mathcal{O}$, there do not exist at the same time a path from x to x' and a path from x' to x. In other words, an orbit is maximal if it is not contained in an other orbit. A maximal orbit is a strongly connected component but the converse is not true, since a vertex without loop is not an orbit.

The set of direct successors (resp. direct predecessors) of $x \in X$ is denoted by $Q^+(x)$ (resp. $Q^-(x)$). For an orbit $\mathcal{O} \subset X$, $\mathcal{O}^+(x)$ denotes $Q^+(x) \cap (X \setminus \mathcal{O})$ and $\mathcal{O}^-(x)$ denotes the set $Q^-(x) \cap (X \setminus \mathcal{O})$. In other words, $\mathcal{O}^+(x)$ is the set of vertices which are directly reached from x and which are not in $\mathcal{O}$.

$In(\mathcal{O}) = \{x \in \mathcal{O} \mid \mathcal{O}^-(x) \neq \emptyset\}$ and $Out(\mathcal{O}) = \{x \in \mathcal{O} \mid \mathcal{O}^+(x) \neq \emptyset\}$ denote the *input* and the *output* of the orbit $\mathcal{O}$.

The hammock T (see Fig. 1) includes eight orbits, two of which are maximal, namely $\mathcal{O}_1 = \{2,3\}$ and $\mathcal{O}_3 = \{4,5,6,7\}$. $\mathcal{O}_2 = \{5,6,7\}$ is not a maximal orbit because it is contained in $\mathcal{O}_3$. $In(\mathcal{O}_1) = \{2,3\}$, $Out(\mathcal{O}_1) = \{2\}$.

An orbit $\mathcal{O}$ is *stable* if $Out(\mathcal{O}) \times In(\mathcal{O}) \subset U$. Remark that if an orbit is stable, every vertex which is at the same time in $Out(\mathcal{O})$ and in $In(\mathcal{O})$ is equipped with a loop. The orbit $\mathcal{O}_2 = \{5,6,7\}$ of the hammock T is not stable. $Out(\mathcal{O}_2) = \{6,7\}$, $In(\mathcal{O}_2) = \{5,6,7\}$. $Out(\mathcal{O}_2) \times In(\mathcal{O}_2) = \{(6,5),(6,6),(6,7),(7,5),(7,6),(7,7)\}$. The edge $(6,6)$ does not exist, so $\mathcal{O}_2$ is not stable.

A maximal orbit $\mathcal{O}$ is *strongly stable* if it is stable and if after deleting the edges in $Out(\mathcal{O}) \times In(\mathcal{O})$ every maximal suborbit is strongly stable.

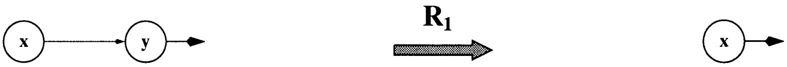
Let us consider the orbit $\mathcal{O}'_1 = \{1,2,3,4\}$ (see Fig. 2). $Out(\mathcal{O}'_1) = \{4\}$, $In(\mathcal{O}'_1) = \{1\}$ and the edge $(4,1)$ exists. After deleting this edge, we consider the unique (maximal) suborbit $\mathcal{O}'_2 = \{2,3\}$. We perform the same process on $\mathcal{O}'_2$. After deleting the edges $(2,2)$, $(2,3)$, $(3,2)$ and $(3,3)$ there is no suborbit anymore. Therefore $\mathcal{O}'_1$ is strongly stable.

An orbit $\mathcal{O}$ is *transverse* if for all $x, y \in Out(\mathcal{O})$, $\mathcal{O}^+(x) = \mathcal{O}^+(y)$ and for all $x, y \in In(\mathcal{O})$, $\mathcal{O}^-(x) = \mathcal{O}^-(y)$. The orbit $\mathcal{O}_3 = \{4,5,6,7\}$ of the hammock T is not transverse because $In(\mathcal{O}_3) = \{5,6,7\}$, $1 \notin \mathcal{O}_3$, the edge $(1,6)$ exists but the edge $(1,5)$ does not. A maximal orbit $\mathcal{O}$ is *strongly transverse* if it is transverse and if every maximal suborbit obtained by deleting the edges in $Out(\mathcal{O}) \times In(\mathcal{O})$ is strongly transverse.

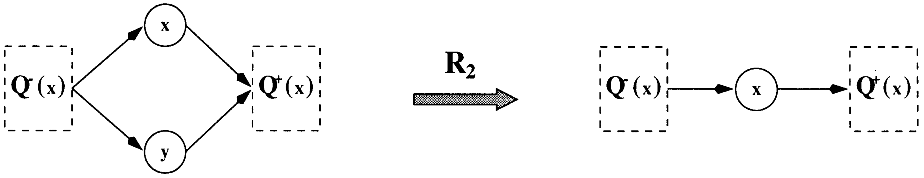
Let G be a graph in which all the orbits are strongly stable. We call graph without orbit of G and denote by $SO(G)$ the acyclic graph obtained by recursively deleting, for every maximal orbit $\mathcal{O}$ of G, the edges in $Out(\mathcal{O}) \times In(\mathcal{O})$.

A graph G is *reducible* if it has no orbit and if it can be reduced to one vertex by iterated applications of any of the three rules R_1, R_2, R_3 described below.

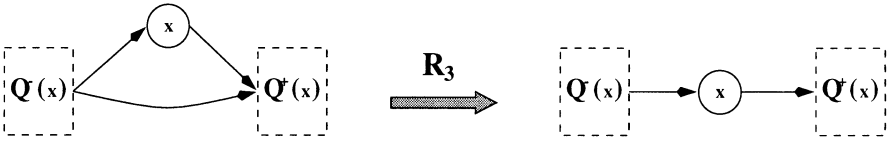
Rule **R₁**: If x and y are vertices such that $Q^-(y) = \{x\}$ and $Q^+(x) = \{y\}$, then delete y and define $Q^+(x) := Q^+(y)$.



Rule **R₂**: If x and y are vertices such that $Q^-(x) = Q^-(y)$ and $Q^+(x) = Q^+(y)$, then delete y and any edge connected to y.



Rule **R₃**: If x is a vertex such that for all $y \in Q^-(x)$, $Q^+(x) \subset Q^+(y)$, then delete edges in $Q^-(x) \times Q^+(x)$.



Example 3.1. We show using Figs. 3–10 an example of reduction of a Glushkov graph.

4. Properties of Glushkov graphs

We now study some conditions which are necessary for a graph to be a Glushkov graph.

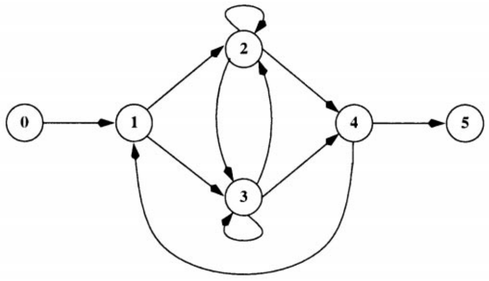


Fig. 2. The orbit $\mathcal{O}'_1 = \{1,2,3,4\}$ is strongly stable.

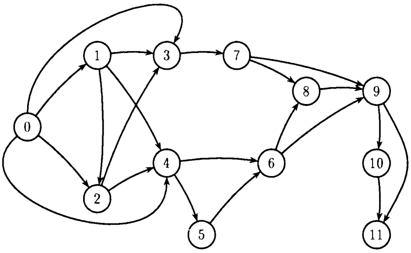


Fig. 3. Example of reduction of an automaton without orbit.

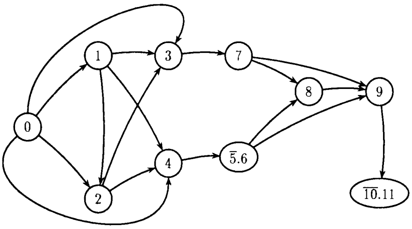


Fig. 4. Reduction by the rule R_3 then R_1.

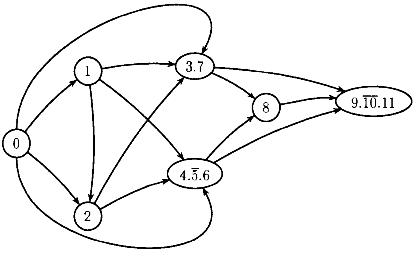


Fig. 5. Reduction by the rule R_1.

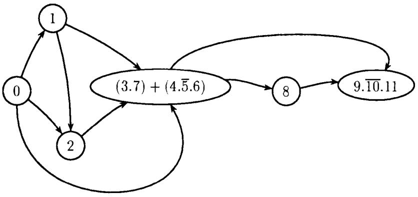


Fig. 6. Reduction by the rule R_2.

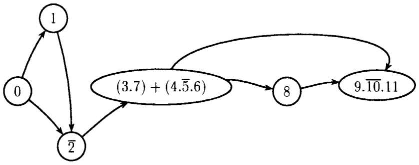


Fig. 7. Reduction by the rule R_3.

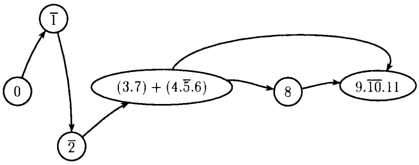


Fig. 8. Reduction by the rule R_3.

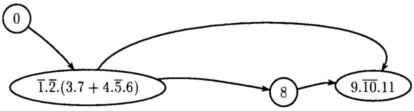


Fig. 9. Sequence of reductions by the rule R_1.

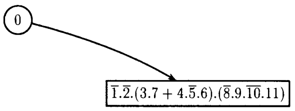

0
$\bar{1}.\bar{2}.(3.7 + 4.\bar{5}.6).(\bar{8}.9.\overline{10}.11)$

Fig. 10. Reduction by the rules R_1 and R_3.

Proposition 4.1. *A Glushkov graph is a hammock.*

Proposition 4.2. *Every maximal orbit of a Glushkov graph is strongly stable and strongly transverse.*

In order to prove this proposition, we first study the link between the subexpressions of E and the orbits of the graph G_E, which leads us to state several lemmas. We shall use the expression "closure operation" both for Kleene closure and for positive closure. The expression E^+ and E^* will be called closure expressions. Let G_E be the Glushkov graph of the expression E. An edge (i,j) of G_E is a *forward edge* if $i < j$, otherwise it is a *back edge*. Let us recall that the set of vertices of G_E is the set of positions of E augmented by a position 0 and a position Φ (observe that edges (i,Φ) are forward edges for each i, $i \neq \Phi$).

The following lemma explicits the origin of the forward edges and of the back edges in a Glushkov graph.

Lemma 4.1. *Consider the Glushkov graph associated with an expression E.*

(a) *A union expression does not create any edge.*

(b) *A concatenation expression only creates forward edges* (*and at least one edge if the second operand is not ε nor $\emptyset$*).

(c) *A closure expression can create both forward edges and back edges, all the back edges going from the Last set to the First set of the expression.*

(d) *A closure expression creates at least one back edge.*

(e) *For a given subexpression F of E, every position x in $Last(F)$ has the same set $Follow(E,x)$.*

Proof. Properties (a)–(c) and (e) immediately infer from computation of the set $Follow(E,x)$. Property (d) comes from the fact that every expression E verifies: $min(First(E)) \leqslant min(Last(E))$. Let us remark that the relation $max(First(E)) \leqslant max(Last(E))$ holds too. □

The link between back edges and closure subexpressions is specified in the following lemma which is a corollary of Lemma 4.1.

Lemma 4.2. *Each back edge (j,i) of G_E is produced by a closure operation acting on a subexpression F of E such that* (1) $[i,j] \subseteq Pos(F)$, (2) $j \in Last(F)$ *and* (3) $i \in First(F)$.

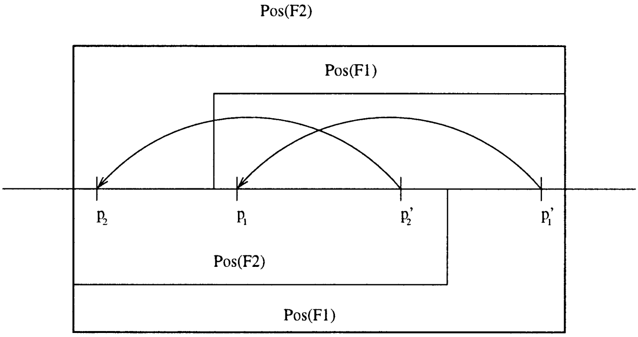
Conversely every closure subexpression F of E induces at least one back edge (j,i) in G_E satisfying the properties (1)–(3).

Two edges (p_1', p_1) and (p_2', p_2) overlap if $p_2 < p_1 < p_2' < p_1'$ or $p_1 < p_2 < p_1' < p_2'$. Lemma 4.3 studies the case of overlapping back edges.

Lemma 4.3. *Let G_E be the Glushkov graph of the expression E and let (p'_1, p_1) and (p'_2, p_2) be two overlapping back edges of G_E. Then there exist two closure subexpressions F_1 and F_2 such that*

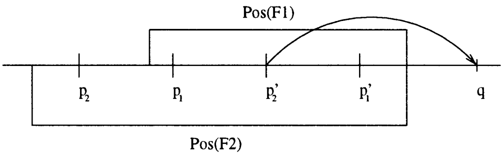
1. $Pos(F_1) \subseteq Pos(F_2)$ *or* $Pos(F_2) \subseteq Pos(F_1)$,
2. $First(F_2) \subseteq First(F_1)$ *or* $Last(F_1) \subseteq Last(F_2)$.

Proof.



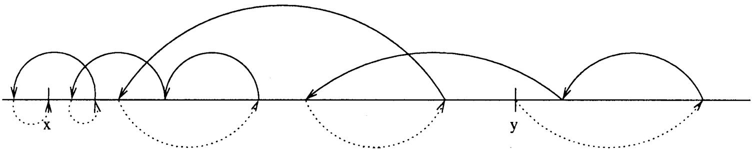
According to Lemma 4.2, there exist two closure expressions F_1 and F_2 such that $[p_1, p'_1] \subseteq Pos(F_1) \subseteq Pos(E)$, $p'_1 \in Last(F_1)$, $p_1 \in First(F_1)$ and $[p_2, p'_2] \subseteq Pos(F_2) \subseteq Pos(E)$, $p'_2 \in Last(F_2)$, $p_2 \in First(F_2)$.

Notice that F_1 and F_2 are either disjoined ($Pos(F_1) \cap Pos(F_2) = \emptyset$), or one is a subexpression of the other. As $[p_1, p'_1] \cap [p_2, p'_2] \neq \emptyset$ we are obviously in the second case. This proves part (1). Consider the situation where $[p_1, p'_1]$ and $[p_2, p'_2]$ overlap. Suppose that F_1 is a subexpression of F_2; we must prove that $Last(F_1) \subseteq Last(F_2)$.

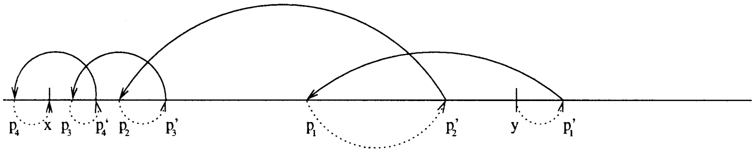


As $p'_2 \in Last(F_2)$, there exists $q \in Pos(E) \setminus Pos(F_2)$, $q > p'_2$, such that $q \in Follow(E, p'_2)$. We also have $p'_2 \in Pos(F_1)$ and $q \notin Pos(F_1)$. Consequently the existence of the edge (p'_2, q) implies $p'_2 \in Last(F_1)$. From Lemma 4.1 (e) it comes that for p in $Last(F_1)$ we have $q \in Follow(E, p)$. Therefore $p \in Pos(F_1)$ implies $p \in Pos(F_2)$. So $p \in Last(F_2)$. A similar proof gives $First(F_2) \subseteq First(F_1)$ in the case where F_2 is a subexpression of F_1. □

The following lemma enables us to study the paths from y to x with $x < y$. Every path from y to x with $x < y$ necessarily includes one or more back edges; the head of a back edge is linked to the tail of the following edge by a path of forward edges. Due to the possible back edges overlapping, one can obtain a path from y to x such as in the following figure:



where dotted lines denote paths. This path can be simplified

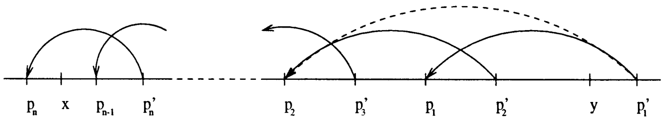


by absorbing some of the back edges in fore-paths. Notice that p_i and p'_{i+1} can be the same vertex.

Lemma 4.4. *Let G_E be the Glushkov graph of the expression E and x and y be two vertices such that $x < y$. If there exists a path from y to x, then there exists a closure subexpression F such that*

1. $[x, y] \subseteq Pos(F)$,
2. *For each u, v in $Pos(F)$, there exist a path from u to v.*

Proof. Consider the path going from y to x and including back edges $(p'_1, p_1), \ldots, (p'_n, p_n)$ and let us show that there exists a back edge (p'_1, p_n).



The two back edges (p'_2, p_2) and (p'_1, p_1) are such that $[p_2, p'_2] \cap [p_1, p'_1] \neq \emptyset$. According to Lemma 4.3 there exists a closure subexpression F such that $[p_2, p'_2] \cup [p_1, p'_1] \subseteq Pos(F)$, $p'_1 \in Last(F)$ and $p_2 \in First(F)$. As F is a closure expression, the back edge (p'_1, p_2) exists. By iterating this process on back edges, we can show that there exists a back edge (p'_1, p_n). Therefore there exists a subexpression G such that $[x, y] \subset [p_n, p'_1] \subseteq Pos(G)$. This proves part (1).

For each u, v in $Pos(F)$, there exists two vertices $t \in Last(F)$ and $i \in First(F)$ such that there is a path from i to v and a path from u to t. Since H is a closure expression, the edge (t, i) exists. Therefore there exists a path from u to v. □

The following lemma relates maximal orbits of G_E to closure subexpressions of E. A closure subexpression F is *maximal* if there does not exist a closure expression F' such that $Pos(F) \subset Pos(F')$.

Lemma 4.5. *Let G_E be the Glushkov graph of the expression E and let $\mathcal{O}$ be a maximal orbit. Then E contains a maximal closure subexpression F such that $\mathcal{O} = Pos(F)$, $In(\mathcal{O}) = First(F)$ and $Out(\mathcal{O}) = Last(F)$.*

Proof. Let $\mathcal{O}$ be a maximal orbit of G_E. Let $x_1 = \min\{x \mid x \in \mathcal{O}\}$ and $x_2 = \max\{x \mid x \in \mathcal{O}\}$. There is a path from x_1 to x_2 and a path from x_2 to x_1. From Lemma 4.4 there exists a closure subexpression F of E such that $[x_1, x_2] \subseteq Pos(F)$; therefore $\mathcal{O} \subseteq Pos(F)$. In the other hand, as F is a closure expression, and $x_1 \in Pos(F)$, we have: for all x in $Pos(F)$, there is a path from x to x_1 and a path from x_1 to x. As $\mathcal{O}$ is a maximal orbit, we deduce that $x \in \mathcal{O}$. Therefore $Pos(F) \subseteq \mathcal{O}$. Finally $Pos(F) = \mathcal{O}$ and as $\mathcal{O}$ is maximal, so is F. If $x \in In(\mathcal{O})$ then there exists a vertex r in $X \setminus \mathcal{O}$ such that the edge (r,x) exists. Since $\mathcal{O}$ is maximal, there is no edge from x to r. So we have $x \in First(F)$. Conversely if $x \in First(F)$, there exists $r \in Pos(E) \setminus Pos(F)$ such that the edge (r,x) exists. As F is maximal there is no path from x to r. It implies that $x \in In(\mathcal{O})$. $Out(\mathcal{O}) = Last(F)$ can be proved in a similar way. $\square$

Lemma 4.6. *Every maximal orbit of a Glushkov graph is stable.*

Proof. Let $G_E = (X, U)$ be a Glushkov graph and let $\mathcal{O}$ be a maximal orbit of G_E. Consider the subexpression F of E such that $Pos(F) = \mathcal{O}$. As F is a closure expression, we have: $x \in Last(F)$ implies $First(F) \subseteq Follow(E,x)$. Consequently $Out(\mathcal{O}) \times In(\mathcal{O}) \subset U$. $\square$

Lemma 4.7. *Every maximal orbit of a Glushkov graph is transverse.*

Proof. Let $\mathcal{O}$ be a maximal orbit of a Glushkov graph G_E. Assume that $\mathcal{O}$ is not transverse. Then (a) there exist $x, y \in Out(\mathcal{O})$ such that $\mathcal{O}^+(x) \neq \mathcal{O}^+(y)$ or (b) there exist $x, y \in In(\mathcal{O})$ such that $\mathcal{O}^-(x) \neq \mathcal{O}^-(y)$. Suppose that (a) holds. Then we have $Follow(E,x) \neq Follow(E,y)$ or $x \in Last(E)$ and $y \notin Last(E)$. Contradiction with Lemmas 4.1 and 4.2. The second case (b) leads us to a similar contradiction. $\square$

Let us recall now the notion of star normal form introduced in [6]. An expression E is in *star normal form* (SNF) if it verifies the following condition, for each H such that H^* is a subexpression of E:

$$\forall x \in Last(H), Follow(H,x) \cap First(H) = \emptyset$$

Brüggemann-Klein showed that to each expression E can be associated an expression $E^\bullet$, such that $E^\bullet$ is in star normal form and $M_{E^\bullet} = M_E$.

Lemma 4.8. *Let $G_E = (X, U)$ be the Glushkov graph of the expression E. Let $\mathcal{O}$ be a maximal orbit of G_E corresponding to the closure subexpression F of E. Let G' be the graph obtained by removing the edges in $Out(\mathcal{O}) \times In(\mathcal{O})$. Then G' is the Glushkov graph of the expression E' deduced from E by substituting $F^\bullet$ to F^+ or $(F^\bullet + \varepsilon)$ to F^*, where $F^\bullet$ is the star normal form of F.*

Proof. It is easy to verify that: $First(E') = First(E)$, $Last(E') = Last(E)$ and that

$$Follow(E',x) = \begin{cases} Follow(E,x) \setminus First(F) & \text{if } x \in Last(F), \\ Follow(E,x) & \text{otherwise.} \end{cases}$$

It follows that $G_{E'} = (X, U \setminus Last(F) \times First(F)) = (X, U \setminus Out(\mathcal{O}) \times In(\mathcal{O})) = G'$. □

We can now give a proof of Proposition 4.2 which asserts that every maximal orbit of a Glushkov graph is strongly stable and strongly transverse.

Proof of Proposition 4.2. The graph obtained by removing edges belonging to $Out(\mathcal{O}) \times In(\mathcal{O})$ from a maximal orbit of a Glushkov graph is a Glushkov graph (Lemma 4.8). In this graph, orbits are stable (Lemma 4.6) and transverse (Lemma 4.7). Consequently, the recursive process of edges removal deduced from the definition of strong stability produces only maximal orbits which are stable and transverse. The orbit $\mathcal{O}$ is therefore strongly stable and strongly transverse. □

The following lemma is used in the proof of Proposition 4.3 which deals with the reducibility of a graph w.r.t. the rules R_1, R_2, R_3.

Lemma 4.9. *Let $G = (X, U)$ and $G' = (X', U')$ be the Glushkov graphs of the expression E and E'. Let $G'' = (X'', U'')$ the graph obtained by replacing in G the vertex α ($0 < \alpha < \Phi$ and $(\alpha, \alpha) \notin U$) by G', in the following way*:

$$\begin{aligned} X'' &= (X \setminus \{\alpha\}) \cup (X' \setminus \{0_{G'}, \Phi_{G'}\}), \\ U'' &= [U \setminus (X \times \{\alpha\} \cup \{\alpha\} \times X)] \\ &\quad \cup Q_G^-(\alpha) \times Q_{G'}^+(0_G) \cup \; Q_{G'}^-(\Phi_{G'}) \times Q_G^+(\alpha) \\ &\quad \cup [U' \setminus (X' \times \{0_{G'}, \Phi_{G'}\} \cup \{0_{G'}, \Phi_{G'}\} \times X')]. \end{aligned}$$

The graph G'' is the Glushkov graph of the expression E'' deduced from E by replacing α by the expression E'. □

Proposition 4.3. *Let G be a graph without orbit. G is a Glushkov graph if and only if one of the following propositions holds*:
- *G has only one state.*
- *There exists a rule R among R_1, R_2, R_3 such that the resulting graph G/R is a Glushkov graph.*

Proof. The first three cases we shall examine correspond to the atomic expressions: $\emptyset$, ε, a, where a is a letter.

(a) The only Glushkov graph with one vertex is the Glushkov graph of the empty set.

(b) The only Glushkov graph with two vertices is the Glushkov graph of the empty word. The only applicable rule is R_1 and G/R_1 is the Glushkov graph of the empty set.

(c) If G has three vertices, and if G is the Glushkov graph of an atomic expression with only one letter, then the only applicable rule is R_1 and G/R_1 is the Glushkov graph of the empty word. Conversely if G/R is the Glushkov graph of the empty word, R is necessarily the rule R_1 and G is the Glushkov graph of an expression which has only one letter.

(d) We now suppose that G is the Glushkov graph of a non atomic expression E, and we show that there exists a rule applicable to G. As E is not atomic, there exists a subexpression F of E such that $F = a_i \cdot a_{i+1}$ or $F = a_i + a_{i+1}$ or $F = a_i + \varepsilon$. Let us examine each of these cases:

Case 1: $F = a_i \cdot a_{i+1}$. $\mathit{Follow}(E,i) = \{i+1\}$ and for all j in $\mathit{Pos}(E)$, $j \neq i$, we have $i+1 \notin \mathit{Follow}(E,j)$. These two conditions are equivalent to the following propositions on the graph G_E: $Q^+(i) = \{i+1\}$ and $Q^-(i+1) = \{i\}$. The rule R_1 can therefore be applied.

Case 2: $F = a_i + a_{i+1}$. $\mathit{Follow}(E,i) = \mathit{Follow}(E,i+1)$ and $i \in \mathit{First}(E) \Leftrightarrow i+1 \in \mathit{First}(E)$. For all j in $\mathit{Pos}(E)$ we have: $i \in \mathit{Follow}(E,j) \Leftrightarrow i+1 \in \mathit{Follow}(E,j)$. In terms of graphs, this means that $Q^+(i) = Q^+(i+1)$ and $Q^-(i) = Q^-(i+1)$. The rule R_2 can therefore be applied.

Case 3: $F = a_i + \varepsilon$. For all j in $\mathit{Pos}(E)$ such that $i \in \mathit{Follow}(E,j)$ we have $\mathit{Follow}(E,i) \subset \mathit{Follow}(E,j)$. In terms of graphs, this means that, for all x, $x \in Q^-(y)$, we have $Q^+(y) \subset Q^+(x)$. The rule R_3 can therefore be applied.

Let us show that after applying one of the three rules, the graph is still a Glushkov graph. For this purpose, one verifies that there exists, for each rule R, an expression E' deduced from E such that the Glushkov graph of E' is equal to G_E/R. This part is a direct application of Lemma 4.9. E' is the expression deduced from E by substituting $a_i \cdot a_{i+1}$ (resp. $a_i + a_{i+1}$, $a_i + \varepsilon$) to a_i in the case of the rule R_1 (resp. R_2, R_3). □

5. Characterization theorem

Theorem 5.1. *$G = (X,U)$ is a Glushkov graph if and only if the three following conditions are satisfied*:

- *G is a hammock.*
- *Each maximal orbit in G is strongly stable and it strongly transverse.*
- *The graph without orbit of G is reducible.*

Proof. If G is a Glushkov graph the following properties hold:

(i) G is a hammock (Proposition 4.1).
(ii) Each orbit in G is strongly stable and strongly transverse (Proposition 4.2).
(iii) The graph without orbit of G is reducible (Proposition 4.3).

In order to prove the converse part of this theorem, we need the following lemma:

Lemma 5.1. *Let $G = (X,U)$ be a graph that satisfies the conditions* (ii) *and* (iii). *Let $\mathcal{O}$ be a maximal orbit in G. By iteration of the rules R_1, R_2 and R_3 in $SO(G), \mathcal{O}$*

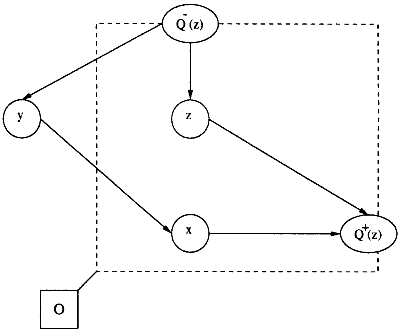
will be reduced to a unique vertex, under the assumption that rules R_1 *and* R_2 *are only applied to pairs* $(x, y) \in \mathcal{O}^2$ *or* $(x, y) \in (X \setminus \mathcal{O})^2$.

Proof. Let $G = (X, U)$ be a graph and $\mathcal{O}$ be a maximal orbit in G. Let us suppose that after the iteration of the rules R_1, R_2 and R_3 in $\mathcal{O}$ and out of $\mathcal{O}$, $\mathcal{O}$ cannot be reduced to one vertex. As $SO(G)$ is reducible, there exists at least one rule R_1 or R_2 which is applicable on a pair (x, y) with $x \in \mathcal{O}$ and $y \notin \mathcal{O}$. Let us examine these two cases.

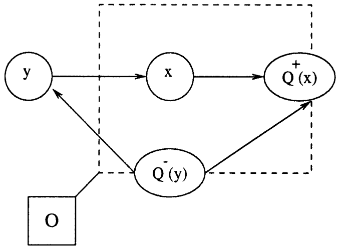
Rule $\mathbf{R_1}$: $Q^+(y) = \{x\}$, $Q^-(x) = \{y\}$.

Let us consider $x \in \mathcal{O}$ and $y \notin \mathcal{O}$ such that the conditions of reduction by R_1 are verified. The cases $(y,x) \in U$ and $(x, y) \in U$ are symmetric. We examine the case $(y,x) \in U$. We have $Q^+(y) = \{x\}$, thus x is the unique element of $In(\mathcal{O})$. We show that the application of the rule R_1 on (y,x) does not lead to any new applicable rule on the pair $(x,z) \in \mathcal{O}^2$.

- If there exists a new applicable rule R_1 on (x,z), with $z \in \mathcal{O}$, then necessarily $Q^+(x) = \{z\}$ and $Q^-(z) = \{x\}$. Consequently R_1 would have been applicable on (x,z) in $\mathcal{O}$, before applying R_1 on (x, y). By assumption, it is not the case.
- If there exists a new applicable rule R_2 on (y,z), with $z \in \mathcal{O}$, one has the following diagram:



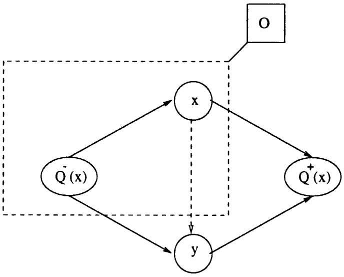
  The vertex x is the unique element of $In(\mathcal{O})$. Thus $Q^-(z)$ is included in $\mathcal{O}$. We have $y \notin \mathcal{O}$, thus $Q^-(z) \subset Out(\mathcal{O})$. As $x \in In(\mathcal{O})$, there exists an element s of $Out(\mathcal{O})$ such that there exists a path from x to s. As there is an edge from every vertex of $Q^-(z)$ to y, as $Q^-(z) \subset Out(\mathcal{O})$, and as $\mathcal{O}$ is transverse, there exists an edge from s to y. Therefore there exists a cycle in $SO(G)$. Contradiction.
- If there exists a new applicable rule R_3 on a vertex x of $\mathcal{O}$, one has the following diagram:



If there exists z, $z \in \mathcal{O} \cap Q^-(y)$ then we prove, in a similar way, the existence of a cycle. It implies $Q^-(y) \cap \mathcal{O} = \emptyset$. If $Q^+(y) \cap \mathcal{O} = \emptyset$ then x is the unique element of $\mathcal{O}$, but $\mathcal{O}$ has at least two vertices. If there exist z, $z \in Q^+(y) \cap \mathcal{O}$, then $z \in In(\mathcal{O})$ but x is the unique element of $In(\mathcal{O})$. Thus there does not exist a new applicable rule R_3.
Rule $\mathbf{R_2}$: $Q^+(y) = Q^+(x)$, $Q^-(y) = Q^-(x)$.

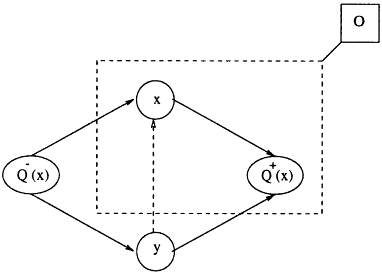
Notice that if z and t are two vertices of $Q^-(x)$, we have: $z \in \mathcal{O}$ implies $t \in \mathcal{O}$. Indeed if $z \in \mathcal{O}$ and $t \notin \mathcal{O}$, we have $z \in Q^-(y)$, hence $z \in Out(\mathcal{O})$ and $t \in Q^-(x)$ and then $x \in In(\mathcal{O})$. The edge (z,x) does not exist in $SO(G)$ because it belongs to $Out(\mathcal{O}) \times In(\mathcal{O})$. Similarily if z and t are two vertices in $Q^+(x)$, we have: $z \in \mathcal{O}$ implies $t \in \mathcal{O}$. Hence there are four cases to consider:

- $Q^-(x) \cap \mathcal{O} = \emptyset$ and $Q^+(x) \subset \mathcal{O}$.



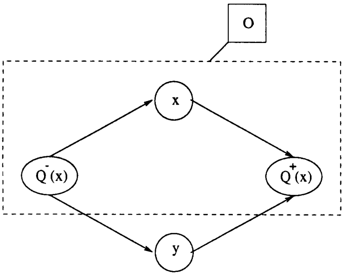
As $Q^-(x) \cap \mathcal{O} = \emptyset$, $x \in In(\mathcal{O})$. By assumption $Q^+(x) \subset \mathcal{O}$ and $y \notin \mathcal{O}$. If $Q^+(x) = Q^+(y)$, then, for all z in $Q^+(x)$, we have $z \in In(\mathcal{O})$. By transversality of $\mathcal{O}$, we get $[(y,z) \in U \Rightarrow (y,x) \in U]$. Contradiction.

- $Q^+(x) \cap \mathcal{O} = \emptyset$ and $Q^-(x) \subset \mathcal{O}$.



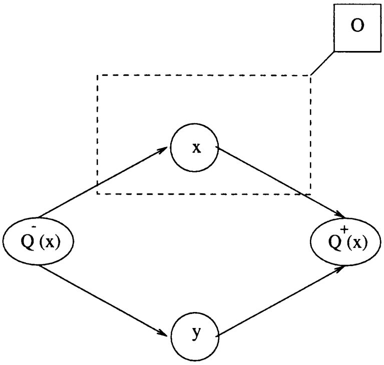
By a similar proof, we show the existence of the edge (x,y) which leads to a contradiction.

- $Q^+(x) \subset \mathcal{O}$ and $Q^-(x) \subset \mathcal{O}$.



Let $z \in Q^-(x)$ and $z' \in Q^+(x)$. By assumption, there is a path from z to z' containing y. As vertices z and z' are in the orbit, there exists a path from z' to z in G; this implies $y \in \mathcal{O}$. Contradiction.

- $Q^-(x) \cap \mathcal{O} = \emptyset$ and $Q^+(x) \cap \mathcal{O} = \emptyset$.



Each edge having x for tail (resp. head) has its head (resp. tail) out of the orbit $\mathcal{O}$. Therefore there does not exist a rule R_1 on (y,z) with $z \in \mathcal{O}$. If there existed a new rule R_2 on (y,z) with $z \in \mathcal{O}$, then this rule would have been applicable on (z,x) in $\mathcal{O}$. If the rule R_3 is applicable on y, it was already applicable before using R_2. □

Proof. (*Converse of Theorem* 5.1). Let $G = (X, U)$ be a graph satisfying conditions (i) to (iii). Let $SO(G) = (X', U')$ be the graph without orbit of G. Let us show that G is a Glushkov graph. Let $\mathcal{O} \subset X$ be a maximal orbit of G. The set $\mathcal{O}$ is reduced to a unique vertex after iteration of the rules R_1, R_2, R_3 in $\mathcal{O}$ or out of $\mathcal{O}$ (Lemma 5.1). Consider the graph $G_{\mathcal{O}} = (X_{\mathcal{O}}, U_{\mathcal{O}})$, where $X_{\mathcal{O}} = \mathcal{O} \cup \{0', \Phi'\}$ and $U_{\mathcal{O}} = (U' \cap \mathcal{O}^2) \cup \{0'\} \times In(\mathcal{O}) \cup Out(\mathcal{O}) \times \{\Phi'\}$. Thus defined, $G_{\mathcal{O}}$ is reducible. Consequently $G_{\mathcal{O}}$ is the Glushkov graph of a regular expression F (Proposition 4.3). We construct the Glushkov graph of F^+ by adding the edges of the set $Last(F) \times First(F)$. This set is exactly the set $Out(\mathcal{O}) \times In(\mathcal{O})$. Let o be the vertex to which $\mathcal{O}$ reduces. We can substitute in $SO(G)$ the graph $G_{\mathcal{O}}$ to the vertex o under the conditions of Lemma 4.9 (o is neither 0 nor Φ, as G is a hammock). According to this lemma, the graph we obtain is a Glushkov one. Consequently, the graph G is a Glushkov graph. □

6. Conclusion

Our study of structural properties of Glushkov automata leads to a theorem of characterization for these automata. We intend to use this characterization in order to simplify the expression computed by classical methods of conversion of an automaton into an expression. The Maple package “Automap” developed by Caron [7, 8] proves to be an efficient tool to undertake this type of survey.

Acknowledgements

We thank the anonymous referees whose valuable suggestions improved the presentation of this paper. We would like to thank J.-M. Champarnaud for his assistance and encouragement.

References

[1] A.V. Aho, J.E. Hopcroft, J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[2] G. Berry, R. Sethi, From regular expression to deterministic automata, Theoret. Comput. Sci. 48 (1) (1986) 117–126.

[3] J. Berstel, Transductions and Context-free Languages, Teubner, Stuttgart, 1979.

[4] J. Berstel, Finite automata and rational languages, an introduction, in: J.-E. Pin (Ed.), Formal Properties of Finite Automata and Applications, Lecture Notes in Computer Science, vol. 386, Springer-Verlag, Berlin, 1989, pp. 2–14.

[5] J. Berstel, J.-E. Pin, Local languages and the Berry–Sethi algorithm, Theoret. Comput. Sci. 155 (1996) 439–446.

[6] A. Brüggemann-Klein, Regular expressions into finite automata, Theoret. Comput. Sci. 120 (1) (1993) 197–213.

[7] P. Caron, AG: Manuel de l'utilisateur, Rapport LIR 96.09, Université de Rouen, France, 1996.

[8] P. Caron, AG: A set of Maple packages for manipulating automata and finite semigroups, Software–Practice & Experience 27 (8) (1997) 863–884.

[9] J.-M. Champarnaud, AUT: un langage pour la manipulation des automates et des semigroupes finis, in: D. Krob (Ed.), Actes des 2^e Journées franco-belges: Théorie des Automates et Applications, Publications de l'Université de Rouen, vol. 176, 1991, pp. 29–43.

[10] J.-M. Champarnaud, G. Hansel, Automate, a computing package for automata and finite semigroups, J. Symbol, Comput. 12 (1991) 197–220.

[11] C.-H. Chang, R. Paige, From regular expression to DFA's using NFA's, in: A. Apostolico, M. Crochemore, Z. Galil, U. Manber (Eds.), Proc. 3rd Annual Symp. on Combinatorial Pattern Matching, Tucson, AZ, Lecture Notes in Computer Science, vol. 664, Springer, Berlin, 1992, pp. 90–110.

[12] S. Eilenberg, Automata, Languages and Machines, vol. B, Academic Press, New York, 1976.

[13] V.M. Glushkov, The abstract theory of automata, Russian Math. Surveys 16 (1961) 1–53.

[14] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, Reading, MA, 1979.

[15] V. Jansen, A. Potthoff, W. Thomas, U. Wermuth, A short guide to the AMoRE system (computing Automata, MOnoids and Regular Expressions), Technical Report 90.2, Aachener Informatik-Berichte, Ahornstr 55, D-5100 Aachen, 1990.

[16] O. Matz, A. Miller, A. Potthoff, W. Thomas, E. Valkena, Report on the program AMoRE, Report, Institut für informatik und praktische mathematik, Christian-Albrechts Universität, Kiel, 1995.

[17] R. McNaughton, H. Yamada, Regular expressions and state graphs for automata, IRA Trans. Electron. Comput. EC-9 (1) (1960) 39–47.

[18] B.G. Mirkin, An algorithm for constructing a base in a language of regular expressions, Eng. Cybernet. 5 (1966) 110–116.

[19] J.-L. Ponty, D. Ziadi, J.-M. Champarnaud, A new quadratic algorithm to convert a regular expression into an automaton, in: D. Raymond, D. Wood, S. Yu (Eds.), Automata Implementation: 1st Internat. Workshop on Implementing Automata, WIA'96, London, Ontario, Lecture Notes in Computer Science, vol. 1260, Springer, Berlin, 1997, pp. 109–119.

[20] K. Thompson, Regular expression search algorithm, Comm. ACM 11 (6) (1968) 419–422.

[21] B.W. Watson, A taxonomy of finite automata construction algorithms, Report, Department of Mathematics and Computing Science, Eindhoven University of Technology, The Netherlands, 1994.

[22] D. Ziadi, J.-L. Ponty, J.-M. Champarnaud, Passage d'une expression rationnelle à un automate fini non-déterministe, Bull. Belg. Math. Soc. 4 (1997) 177–203.