NFA reduction algorithms by means of regular inequalities

J.-M. Champarnaud and F. Coulon
Université de Rouen, LIFAR
F-76821 Mont Saint Aignan Cedex, France
{jmc,fcoulon}@dir.univ-rouen.fr

March 5, 2004

Abstract

We present different techniques for reducing the number of states and transitions in nondeterministic automata. These techniques are based on the two preorders over the set of states, related to the inclusion of left and right languages. Since their exact computation is $\mathcal{NP}$-hard, we focus on polynomial approximations which enable a reduction of the NFA all the same. Our main algorithm relies on a *first approximation*, which can be easily implemented by means of matrix products with an $\mathcal{O}(mn^3)$ time complexity, and optimized to an $\mathcal{O}(mn)$ time complexity, where m is the number of transitions and n is the number of states. This first algorithm appears to be more efficient than the known techniques based on equivalence relations as described by Lucian Ilie and Sheng Yu. Afterwards, we briefly describe some more accurate approximations and the exact (but exponential) calculation of these preorders by means of determinization.

1 Introduction

By NFA reduction algorithms, we mean algorithms which from a given NFA produce a smaller equivalent NFA w.r.t. the number of states. Actually, the main algorithms given in this paper also reduce the number of transitions, so that there is no ambiguity about which complexity measure is considered as being reduced. Among automata which recognize a given regular language $\mathcal{L}$, the problem of computing one or every minimal NFA w.r.t. the number of states has been shown to be $\mathcal{NP}$-hard in [10]. Indeed, known algorithms like in [11, 12, 4] are quite not practicable. Our present aim is to provide reduction algorithms which have a polynomial complexity w.r.t. the initial number of states in the given NFA. Such algorithms may be useful, for instance, to prevent or moderate the blow up during the determinization of the NFA, or to speed up either its straightforward simulation, or its simulation based on a partial determinization [14, 5].

Except for the method of Brzozowski [3], and the incremental algorithm of Watson and Daciuk [13], the classical reduction algorithms for DFAs, e.g. in [6], are based on an equivalence relation over the set of states Q which converges step by step toward the coarsest equivalence relation such that all states contained in one equivalence class have the same right language.

Whereas equivalence relations are fully appropriate to the DFAs, the related relations for NFAs seem to be preorders. Our new reduction methods are based on the two preorders $\overrightarrow{\subseteq}$ and $\overleftarrow{\subseteq}$ defined by $q\overrightarrow{\subseteq}p$ (resp. $q\overleftarrow{\subseteq}p$) if the right (resp. left) language of q is included in the right (resp. left) language of p.

Valentin Antimirov has given a containment calculus for regular expressions, in the form of a term-rewriting system [2]. Once transposed into our context, this system gives us an inductive calculus of $\overrightarrow{\subseteq}$ and $\overleftarrow{\subseteq}$. See the definitions of $\overrightarrow{\subseteq}_n$ and $\overleftarrow{\subseteq}_n$ in Section 6.

Unfortunately, computing the exact relations $\overrightarrow{\subseteq}$ and $\overleftarrow{\subseteq}$ is known to be $\mathcal{NP}$-hard, see e.g. [7], and indeed, the inductive construction of $\overleftarrow{\subseteq}_n$ and $\overrightarrow{\subseteq}_n$ in Section 6 has an exponential time and space complexity. So, the major part of our paper is devoted to a *first order* approximation of

$\overleftarrow{\subseteq}$ and $\overrightarrow{\subseteq}$, which can be computed in time $\mathcal{O}(mn)$ where m is the number of transitions, and n is the number of states. Indeed, this first approximation is at least as efficient as the best results obtained with equivalence relations by Lucian Ilie and Sheng Yu in [8], in the sense that every equality detected by the equivalence relation is detected by our preorders through double inclusion. We provide a simple example where our approximations are strictly more efficient. Moreover, we give a detailed algorithm based on matrix products, which can be easily parallelized in the case of the $\mathcal{O}(mn^3)$ version [9]. Our algorithm can also be easily modified in order to compute the equivalence relation given by Lucian Ilie and Sheng Yu, so that we provide an $\mathcal{O}(mn)$ time complexity implementation of their algorithm.

After having presented a general framework for the merging operation over automata in Section 2, the Section 3 briefly describes the reduction technique based on equivalence relations. Section 4 introduces the preorders related with the inclusion of left and right languages and their application to detecting states which can be merged. Section 5 dwells on the first order approximation which can be computed in time $\mathcal{O}(mn)$. Section 6 mentions higher order approximations whose complexity increases as they get more accurate. Section 7 sums up the different reduction techniques deduced from the above-mentioned approximations, and finally, Section 8 discusses the problem of computing the exact preorders.

2 Definitions and Basic Properties

Let X be a finite set, its cardinal is denoted $|X|$, and we let 2^X denote its power set.

An automaton is a quintuple $A =< Q, \Sigma, \delta, I, F >$ where Q is a finite set of *states*, Σ is the *alphabet*, $\delta : Q \times \Sigma \mapsto 2^Q$ is the *transition function*, I (resp. F) is a subset of Q whose elements are the *initial states* (resp. *final states*). The function δ is extended to $2^Q \times \Sigma^* \mapsto 2^Q$ by letting for all $E \subseteq Q$ and $a \in \Sigma$, $\delta(E, a) = \bigcup_{q \in E} \delta(q, a)$ and by the following recursive definition: we let $\delta(E, \varepsilon) = E$ and for all $w \in \Sigma^*$, $\delta(E, aw) = \delta(\delta(E, a), w)$. The language recognized by A, denoted $\mathcal{L}(A)$ is the set $\{w \in \Sigma^* \mid \delta(I, w) \cap F \neq \emptyset\}$. Two automata A and B are said to be *equivalent* if $\mathcal{L}(A) = \mathcal{L}(B)$.

An automaton A is *deterministic* if it has a unique initial state and for all $q \in Q$, $a \in \Sigma$, we have $|\delta(q, a)| = 1$. In the following, DFA stands for deterministic finite automaton, and NFA stands for nondeterministic finite automaton.

Let $A =< Q, \Sigma, \delta, I, F >$ be an NFA.

Definition 1 *The* left language *of a state q, denoted $\overleftarrow{\mathcal{L}}^A(q)$ is the set $\{w \in \Sigma^* \mid q \in \delta(I, w)\}$.*

Symmetrically, the right language *of a state q, denoted $\overrightarrow{\mathcal{L}}^A(q)$ is the set $\{w \in \Sigma^* \mid \delta(q, w) \cap F \neq \emptyset\}$.*

Definition 2 *The* reverse automaton *of A, denoted by $\overline{A}$, is the quintuple $< Q, \Sigma, \overline{\delta}, F, I >$ where $q' \in \overline{\delta}(q, a)$ iff $q \in \delta(q', a)$ for all $q, q' \in Q$ and $a \in \Sigma$.*

Definition 3 *Let $q \in Q$. The* fan out *of q is denoted $\overrightarrow{q}$ and defined by $\overrightarrow{q} = \{(a, q') \in \Sigma \times Q \mid q' \in \delta(q, a)\}$. Symmetrically, the* fan in *of q is denoted $\overleftarrow{q}$ and defined by $\overleftarrow{q} = \{(q', a) \in Q \times \Sigma \mid q \in \delta(q', a)\}$.*

The elements of the fan in and fan out of q are called arrows.

Reduction algorithms which are considered in the following are algorithms which proceed by *merging* some states of the original NFA. Let us introduce a general framework for this kind of manipulation.

Definition 4 *Let q and p be two states of A. We let $\mathrm{merge}(A, q, p)$ stand for an automaton obtained from A by merging q and p, that is, p is deleted and some arrows are added to $\overleftarrow{q}$ and $\overrightarrow{q}$, so that $\overleftarrow{\mathcal{L}}^{\mathrm{merge}(A,q,p)}(q) = \overleftarrow{\mathcal{L}}^A(q) \cup \overleftarrow{\mathcal{L}}^A(p)$ and $\overrightarrow{\mathcal{L}}^{\mathrm{merge}(A,q,p)}(q) = \overrightarrow{\mathcal{L}}^A(q) \cup \overrightarrow{\mathcal{L}}^A(p)$. We define a relation $\sim$ on Q by letting $q \sim p$ if and only if $\mathrm{merge}(A, q, p)$ is equivalent to A.*

Whatever we know about A, we can always build merge(A, q, p) by adding $\overleftarrow{p}$ to $\overleftarrow{q}$ and adding $\overrightarrow{p}$ to $\overrightarrow{q}$.

Proposition 1 *Let $q, p \in Q$, we have $q \sim p$ if and only if $\overleftarrow{\mathcal{L}}^A(q) \cdot \overrightarrow{\mathcal{L}}^A(p) \subseteq \mathcal{L}(A)$ and $\overleftarrow{\mathcal{L}}^A(p) \cdot \overrightarrow{\mathcal{L}}^A(q) \subseteq \mathcal{L}(A)$.*

Proof. An automaton B recognizes $\mathcal{L}(A)$ if and only if for all $u, v \in \Sigma^*$, we have

$$u \cdot v \in \mathcal{L}(A) \iff \left[\ (\ \exists q' \in Q_B\ ) \quad u \in \overleftarrow{\mathcal{L}}^B(q') \wedge v \in \overrightarrow{\mathcal{L}}^B(q')\ \right]$$

Hence the result. ■

In particular, the relation $\sim$ is well defined: it is independent of the choice of merge(A, q, p). Unfortunately, it is trivially not an equivalence relation.

3 Equalities of left and right languages

A first reduction algorithm is obtained by adapting the Moore's algorithm (which concerns the minimization of DFAs) to the case of NFAs. It has been described by Lucian Ilie and Sheng Yu in [8].

Define the equivalence relation $\equiv$ on Q by letting $q \equiv p$ if and only if $\overrightarrow{\mathcal{L}}(q) = \overrightarrow{\mathcal{L}}(p)$. Indeed, $\equiv$ is the Nerode equivalence. Let $\equiv_0$ be another equivalence relation on Q contained in $\equiv$, that is, $q \equiv_0 p$ implies $q \equiv p$. Then, for all $q, p \in Q$, we have $q \equiv_0 p \Longrightarrow q \sim p$, that is, q and p can be merged as soon as $q \equiv_0 p$.

Such a relation $\equiv_0$ can be computed as the coarsest equivalence relation on Q that satisfies the two following axioms:

1. For all $q \in F$ and $p \in Q \setminus F$, we have $q \not\equiv_0 p$,
2. Let $q, p \in Q$ and $a \in \Sigma$,

$$q \equiv_0 p \Longrightarrow \Big[(\forall q' \in \delta(q, a))(\exists p' \in \delta(p, a)) \quad q' \equiv_0 p'\Big]$$

Let $q, p, q', p' \in Q$, and suppose that $q \equiv_0 p$ and $q' \equiv_0 p'$. Then we still have $q' \equiv_0 p'$ in the automaton merge(A, q, p), so that the merging operations can be successively applied in any order and lead to a k-state automaton where k is the number of equivalence classes for $\equiv_0$.

A dual relation can be obtained the same way by considering left languages instead of right languages.

4 Inequalities of left and right languages

We shall see in this section that using preorders instead of equivalence relations gives better results for nondeterministic automata. Let us recall that a relation is a preorder if and only if it is both reflexive and transitive. An antisymmetric preorder is a partial order, and a symmetric preorder is an equivalence relation.

Definition 5 *Let $\overrightarrow{\subseteq}$ and $\overleftarrow{\subseteq}$ be two preorders on Q defined by letting $q \overrightarrow{\subseteq} p$ if and only if $\overrightarrow{\mathcal{L}}(q) \subseteq \overrightarrow{\mathcal{L}}(p)$, and $q \overleftarrow{\subseteq} p$ if and only if $\overleftarrow{\mathcal{L}}(q) \subseteq \overleftarrow{\mathcal{L}}(p)$.*

Definition 6 *Let R and R' be two relations on Q. The relation R is said to be* smaller *than R' if for all $q, p \in Q$, qRp implies $qR'p$.*

This partial order coincides with the inclusion if the relations are considered as subsets of $Q \times Q$.

Proposition 2 *Let q and p be two states of A. Let $\overrightarrow{\subseteq}_0$ and $\overleftarrow{\subseteq}_0$ be two relations on Q respectively smaller than $\overrightarrow{\subseteq}$ and $\overleftarrow{\subseteq}$. We have $q \sim p$ as soon as one of the following conditions holds:*

1. *$q \overrightarrow{\subseteq}_0 p$ and $p \overrightarrow{\subseteq}_0 q$,*
2. *$q \overleftarrow{\subseteq}_0 p$ and $p \overleftarrow{\subseteq}_0 q$,*
3. *$q \overrightarrow{\subseteq}_0 p$ and $q \overleftarrow{\subseteq}_0 p$,*
4. *$p \overrightarrow{\subseteq}_0 q$ and $p \overleftarrow{\subseteq}_0 q$.*

Proof. Trivial from Proposition 1. ■

Such relations $\overrightarrow{\subseteq}_0$ and $\overleftarrow{\subseteq}_0$ are said to be *approximations* of the inequality relations $\overrightarrow{\subseteq}$ and $\overleftarrow{\subseteq}$.

When merging two states q and p which satisfy one of the conditions of Proposition 2, the relations $\overrightarrow{\subseteq}_0$ and $\overleftarrow{\subseteq}_0$ should be updated in order to remain smaller than $\overrightarrow{\subseteq}$ and $\overleftarrow{\subseteq}$ in the automaton obtained by merging. Let us enumerate the conditions of Proposition 2 and give the associated reduction operations:

1. In merge(A, q, p), add $\overleftarrow{p}$ to $\overleftarrow{q}$, and let $\overleftarrow{\subseteq}_0 \leftarrow \overleftarrow{\subseteq}_0 \setminus \{(q, p') \mid p' \in Q,\ p \overleftarrow{\not\subseteq}_0 p'\}$,
2. In merge(A, q, p), add $\overrightarrow{p}$ to $\overrightarrow{q}$, and let $\overrightarrow{\subseteq}_0 \leftarrow \overrightarrow{\subseteq}_0 \setminus \{(q, p') \mid p' \in Q,\ p \overrightarrow{\not\subseteq}_0 p'\}$,
3. Compute merge(A, p, q),
4. Compute merge(A, q, p).

5 First order appoximation of inequalities

Let $\overrightarrow{\subseteq}_1$ be the greatest preorder on Q which satisfies the two following axioms:

1. For all $q \in F$ and $p \in Q \setminus F$, we have $q \overrightarrow{\not\subseteq}_1 p$,
2. Let $q, p \in Q$ and $a \in \Sigma$,

$$q \overrightarrow{\subseteq}_1 p \Longrightarrow \left[(\ \forall q' \in \delta(q, a)\ )(\ \exists p' \in \delta(p, a)\ ) \quad q' \overrightarrow{\subseteq}_1 p' \right]$$

Proposition 3 *The relation $\overrightarrow{\subseteq}_1$ is smaller than $\overrightarrow{\subseteq}$.*

Proof. We shall prove by induction on the length of the word w that for all $q, p \in Q$, $\left[q \overrightarrow{\subseteq}_1 p \wedge w \in \overrightarrow{\mathcal{L}}(q)\right] \Longrightarrow w \in \overrightarrow{\mathcal{L}}(p)$.

Let $q \overrightarrow{\subseteq}_1 p$ and $\varepsilon \in \overrightarrow{\mathcal{L}}(q)$. Since $q \in F$, the first axiom of $\overrightarrow{\subseteq}_1$ implies that $p \in F$ and $\varepsilon \in \overrightarrow{\mathcal{L}}(p)$.

Now, let $q \overrightarrow{\subseteq}_1 p$ and $aw \in \overrightarrow{\mathcal{L}}(q)$, where $a \in \Sigma$, $w \in \Sigma^*$ and $q, p \in Q$. There exists $q' \in \delta(q, a)$ such that $w \in \overrightarrow{\mathcal{L}}(q')$. The second axiom implies that there exists $p' \in \delta(p, a)$ such that $q' \overrightarrow{\subseteq}_1 p'$. By induction, we have $w \in \overrightarrow{\mathcal{L}}(p')$, hence $aw \in \overrightarrow{\mathcal{L}}(p)$. ■

Symmetrically, we have a preorder $\overleftarrow{\subseteq}_1$, smaller than $\overleftarrow{\subseteq}$, obtained by considering the reverse automaton.

Indeed, reducing A by simply using the first order approximations $\overrightarrow{\subseteq}_1$ and $\overleftarrow{\subseteq}_1$ with the conditions 1 and 2 of Proposition 2 is at least as efficient as reducing A by using the relation $\equiv_0$ of Section 3:

Proposition 4 *Let $q, p \in Q$ such that $q \equiv_0 p$. We have $q \overrightarrow{\subseteq}_1 p$ and $p \overrightarrow{\subseteq}_1 q$.*

Proof. Actually, the constraint "R is a preorder" is weaker than "R is an equivalence relation", and the axioms of $\overrightarrow{\subseteq}_1$ are weaker than the axioms of $\equiv_0$. ■

We can see on Figure 1 an example where $\overrightarrow{\subseteq}_1$ and $\overleftarrow{\subseteq}_1$ are strictly more efficient than $\equiv_0$. We easily verify that $1 \overrightarrow{\subseteq}_1 2$ and $2 \overrightarrow{\subseteq}_1 1$, though we do not have $1 \equiv_0 2$.

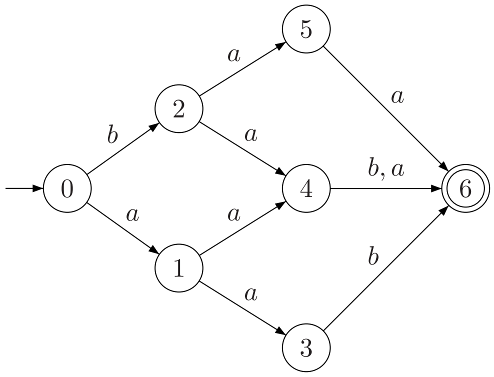

Figure 1: States 1 and 2 have the same right language.

5.1 Implementation of the first order approximation

Consider the semigroup $\mathcal{M}_2(n)$ of $n \times n$-matrices over the boolean semiring. This semigroup is partially ordered: a matrix A is smaller than a matrix B iff $A_{i,j} = 1 \Rightarrow B_{i,j} = 1$ for all i, j. Consider that the set Q is equal to $\{1, 2, \dots, n\}$, so that each function from Q to Q, as well as each binary relation R on Q, is naturally associated with a matrix X in $\mathcal{M}_2(n)$ by letting $X_{i,j} = 1 \Longleftrightarrow jRi$, which is the same as $X_{i,j} = 1 \Longleftrightarrow i \in R(j)$.

Let X_0 be the matrix associated with the relation $F \times (Q \setminus F)$. For all $a \in \Sigma$, let δ_a be the matrix associated with the application $q \mapsto \delta(q, a)$ and let ${}^t\delta_a$ be the transposition of δ_a in $\mathcal{M}_2(n)$. Indeed, ${}^t\delta_a$ is associated with $q \mapsto \overline{\delta}(q, a)$.

Let $\mathcal{M}_{\mathbb{N}}(n)$ be the semiring of $n \times n$-matrices over the semiring of natural integers. Let '.' be the matrix product in $\mathcal{M}_2(n)$ and '$\circ$' be the matrix product in $\mathcal{M}_{\mathbb{N}}(n)$. Let X be a matrix in $\mathcal{M}_{\mathbb{N}}(n)$ and V be a vector of $\mathbb{N}^n$, we denote by $\overline{X}^V$ the matrix of $\mathcal{M}_2(n)$ defined by

$$\begin{array}{rcl} \overline{X}^V_{i,j} & = & 1 \text{ if } X_{i,j} = V_j, \\ & = & 0 \text{ otherwise} \end{array}$$

For all $a \in \Sigma$, let $V(a) \in \mathbb{N}^n$ be defined by

$$V(a)_q = |\delta(q, a)| \quad (\ \forall q \in Q\ )$$

Let Δ_a, be defined for all $X \in \mathcal{M}_2(n)$ by $\Delta_a(X) = {}^t\delta_a.\overline{(X \circ \delta_a)}^{V(a)} + X$.

Proposition 5 *Let X be the transposition of the matrix associated with the relation $\overrightarrow{\not\subseteq}_1$.*
X is the smallest matrix which is a fix point for all Δ_a ($a \in \Sigma$) and is greater than ${}^tX_0$.

Proof. Let R be a relation on Q. For all $a \in \Sigma$, we let

$$A_a(R) \equiv \Big[(\ \forall q, p \in Q\ )\Big[(\ \exists q' \in \delta(q, a)\ )(\ \forall p' \in \delta(p, a)\ )\ \ q'Rp'\Big] \Longrightarrow qRp\Big]$$

Let X be the transposition of the matrix associated with R. For all $q, p \in Q$, $(X \circ \delta_a)_{q',p}$ is the number of states p' such that $p' \in \delta(p, a)$ and $q'Rp'$. If this number is equal to $|\delta(p, a)|$, that is, if $\left(\overline{X \circ \delta_a}^{V(a)}\right)_{q',p}$ is equal to 1, then we have $q'Rp'$ for all $p' \in \delta(p, a)$.

Hence, for all $q, p \in Q$, $\left({}^t\delta_a.\overline{X \circ \delta_a}^{V(a)}\right)_{q,p} = 1$ is equivalent to $(\ \exists q' \in \delta(q, a)\ )(\ \forall p' \in \delta(p, a)\ )\ \ q'Rp'$. So, X is a fix point for Δ_a if and only if $A_a(R)$.

Now, let X be the transposition of the matrix associated with $\overrightarrow{\not\subseteq}_1$. We know from the axioms of $\overrightarrow{\subseteq}_1$ that $\overrightarrow{\not\subseteq}_1$ is the smallest relation on Q which contains $F \times (Q \setminus F)$ and such that $A_a(\overrightarrow{\not\subseteq}_1)$ for all $a \in \Sigma$. Hence, X is the smallest matrix which is a fix point for all Δ_a and contains ${}^tX_0$. ■

Figure 2 illustrates the Proof. Let $q'Rp'$ for all states p' in the figure. Then (q, p) is added to R because $(\exists q' \in \delta(q, a))(\forall p' \in \delta(p, a)) \quad q'Rp'$.

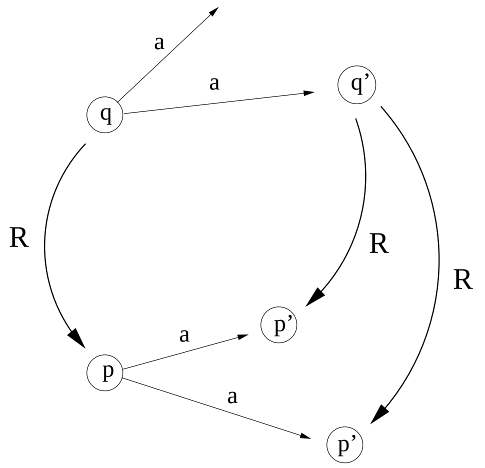

Figure 2: Principle of the recursive construction of $\not\subseteq_1$.

The following algorithm is then straightforward:

FIRST ORDER(A)

1 $X \leftarrow {}^tX_0$
2 **repeat**
3 | **for each** $a \in \Sigma$ **do**
4 | | $X \leftarrow \Delta_a(X)$
5 **until** X is a fix point for all Δ_a $(a \in \Sigma)$
▷ ${}^tX$ is now associated with $\overrightarrow{\not\subseteq}_1$

Let us digress and get back on the first relation $\equiv_0$. Indeed, $\equiv_0$ can be computed by a slightly different algorithm:

- Line 1, replace $X \leftarrow {}^tX_0$ by $X \leftarrow Sym({}^tX_0)$ where Sym stands for the symmetric closure,
- Line 4, replace $X \leftarrow \Delta_a(X)$ by $X \leftarrow Sym(\Delta_a(X))$,
- the algorithm ends with X associated with $\not\equiv_0$.

5.2 Complexity

Let m be the number of transitions in A. Hence, the average number of non-zero coefficients in the matrices δ_a and ${}^t\delta_a$ is $\frac{m}{|\Sigma|}$.

The number of iterations is bound by n^2. On the other hand, Line 4 consists in two sparse matrix products, so that its complexity is bound by $\mathcal{O}(\frac{m}{|\Sigma|}n)$, and we get an $\mathcal{O}(mn^3)$ overall complexity. The implemention of Line 4 using matrix products may be interesting for parallelization, using standard parallelization techniques for matrix product [9].

But indeed, the matrix X is increasing, so that during the whole algorithm, there are less than n^2 non-zero coefficients added to X. We can verify that Δ_a is linear for all a, in the sense that for any two matrices X_1 and X_2, we have $\Delta_a(X_1 + X_2) = \Delta_a(X_1) + \Delta_a(X_2)$. Here is the algorithm that details Line 4 of Algorithm FIRST ORDER. For all $a \in \Sigma$ we maintain two boolean $(n \times n)$-matrices M_a and N_a which are initially set to zero.

$[X \leftarrow \Delta_a(X)](a, M_a, N_a, X)$
▷ **Auxiliary:** M'_a : $n \times n$ matrix
1 **Begin**
2 $\quad M'_a \leftarrow M_a$
3 $\quad M_a \leftarrow \overline{(X - N_a) \circ \delta_a}^{V(a)} + M_a$
4 $\quad N_a \leftarrow X$
5 $\quad X \leftarrow {}^t\delta_a.(M_a - M'_a) + X$
6 **End**

For each $a \in \Sigma$, the two following assertions hold during the whole algorithm First Order:

$$I_a \equiv \left[M_a = \overline{N_a \circ \delta_a}^{V(a)}\right] \quad \text{and} \quad J_a \equiv \left[X \geq {}^t\delta_a.M_a\right]$$

Assertion J_a is conserved from one iteration step to another because X is increasing. The conservation of I_a through Lines 3 and 4 is due to the linearity of matrix product. At Line 5, J_a gives $X \geq {}^t\delta_a M'_a$, so that by linearity, Line 5 is equivalent to $X \leftarrow {}^t\delta_a.M_a + X$, that is, $X \leftarrow \Delta_a(X)$. Hence the correctness of the algorithm.

Let $a \in \Sigma$ and let k_i (resp. l_i) be the number of non-zero coefficients in matrix $(X - N_a)$ at Line 3 (resp. matrix $(M_a - M'_a)$ at Line 5) of the i^{th} iteration. The matrices X and M_a are both increasing, so that $\sum_i k_i$ and $\sum_i l_i$ are both lower than n^2. Hence the total complexity of matrix products is lower than mn. The calculation of $X - N_a$ and $M_a - M'_a$ can be performed by maintaining the list of added non-zero coefficients. On the other hand, the total number of iterations is bound by n^2.

Hence, the first order approximation preorders can be computed with a space complexity of $\mathcal{O}(n^2)$ and a time complexity of $\mathcal{O}(mn)$.

6 Higher order approximations

For all finite sets X and all $k \in \mathbb{N}$, we let $\mathcal{P}_k(X)$ stand for $\{P \subseteq X \mid |P| \leq k\}$.

For all $k \geq 1$, let $\overrightarrow{\subseteq}_k$ be the relation between Q and $\mathcal{P}_k(Q)$ that is the greatest relation which satisfies the two following axioms:

1. $q \overrightarrow{\not\subseteq}_k P$ for all $q \in F$ and $P \in \mathcal{P}_k(Q \setminus F)$,
2. Let $q \in Q$, $P \in \mathcal{P}_k(Q)$ and $a \in \Sigma$,

$$q \overrightarrow{\subseteq}_k P \Longrightarrow \left[(\ \forall q' \in \delta(q,a)\ )(\ \exists P' \in \mathcal{P}_k(\delta(P,a))\ )\quad q' \overrightarrow{\subseteq}_k P'\right]$$

Let $q, p \in Q$, we note $q \overrightarrow{\subseteq}_k p$ instead of $q \overrightarrow{\subseteq}_k \{p\}$, and we shall consider the trace of $\overrightarrow{\subseteq}_k$ on Q^2 as the relation $\{(q,p) \in Q \mid q \overrightarrow{\subseteq}_k p\}$. For the case $k = 1$, we recover our first relation $\overrightarrow{\subseteq}_1$.

Proposition 6 *The trace of $\overrightarrow{\subseteq}_k$ on Q^2 is smaller than or equal to $\overrightarrow{\subseteq}$ for all $k \geq 1$.*

Proof. We shall prove by induction on the length of the word w that for all $q \in Q$ and $P \in \mathcal{P}_k(Q)$, having $q \overrightarrow{\subseteq}_k P$ and $w \in \overrightarrow{\mathcal{L}}(q)$ implies $w \in \overrightarrow{\mathcal{L}}(p)$ for some $p \in P$.

Let $\varepsilon \in \overrightarrow{\mathcal{L}}(q)$. Since $q \in F$, the first axiom of $\overrightarrow{\subseteq}_k$ implies that there exists $p \in P \cap F$, hence $\varepsilon \in \overrightarrow{\mathcal{L}}(p)$.

Now, let $aw \in \overrightarrow{\mathcal{L}}(q)$. There exists $q' \in \delta(q,a)$ such that $w \in \overrightarrow{\mathcal{L}}(q')$. The second axiom implies that there exists $P' \subseteq \delta(P,a)$, with $|P'| \leq k$, such that $q' \overrightarrow{\subseteq}_k P'$. By induction, we have $w \in \overrightarrow{\mathcal{L}}(p')$ for some $p' \in P'$, hence, let p in P such that $p' \in \delta(p,a)$, we have $aw \in \overrightarrow{\mathcal{L}}(p)$. ■

The trace of $\overrightarrow{\subseteq}_k$ on Q^2 is increasing w.r.t. k and reaches $\overrightarrow{\subseteq}$. Indeed, the trace of $\overrightarrow{\subseteq}_n$ is equal to $\overrightarrow{\subseteq}$.

7 Reduction methods

Let $\overleftarrow{\subseteq}_0$ and $\overrightarrow{\subseteq}_0$ be respectively two approximations of $\overleftarrow{\subseteq}$ and $\overrightarrow{\subseteq}$. One can simply apply Proposition 2. This method reduces the number of states as well as the number of transitions.

If one aims to reduce the number of states, without considering the number of transitions, the relations $\overleftarrow{\subseteq}_0$ and $\overrightarrow{\subseteq}_0$ can be grown first in the following way. Let q and p such that $q\overrightarrow{\subseteq}_0 p$, the fan in $\overleftarrow{p}$ can be added to the fan in $\overleftarrow{q}$. Then (p,q) can be added to $\overleftarrow{\subseteq}_0$. Symmetrically, if $q\overleftarrow{\subseteq}p$, then $\overrightarrow{p}$ can be added to $\overrightarrow{q}$, and then (p,q) can be added to $\overrightarrow{\subseteq}_0$. Hence we have the following algorithm:

Reduction(A)

Calculate the preorders $\overleftarrow{\subseteq}_0$ and $\overrightarrow{\subseteq}_0$ associated with A
for each $(q,p) \in \overleftarrow{\subseteq}_0$ **do**
  | Add $\overrightarrow{p}$ to $\overrightarrow{q}$
  | Add (p,q) to $\overrightarrow{\subseteq}_0$
for each $(q,p) \in \overrightarrow{\subseteq}_0$ **do**
  | Add $\overleftarrow{p}$ to $\overleftarrow{q}$
  | Add (p,q) to $\overleftarrow{\subseteq}_0$
for each (q,p) such that $q\overrightarrow{\subseteq}p$ and $p\overrightarrow{\subseteq}q$ **do**
  | Add $\overleftarrow{p}$ to $\overleftarrow{q}$
  | Delete p
for each (q,p) such that $q\overrightarrow{\subseteq}p$ and $q\overleftarrow{\subseteq}p$ **do**
  | Delete q

If we have computed the approximations $\overleftarrow{\subseteq}_k$ and $\overrightarrow{\subseteq}_k$ for some $k \geq 2$, then we can consider merging one state q with a set of states P:

Definition 7 *Let $q \in Q$ and $P \subseteq Q$, we define the automaton* merge(A,q,P) *obtained from A by deleting q and adding $\overrightarrow{q}$ to each $\overrightarrow{p}$ $(p \in Q)$, and adding $\overleftarrow{q}$ to each $\overleftarrow{p}$ $(p \in P)$.*

Proposition 7 *Let $q \in Q$ and $P \subseteq Q$. The automaton* merge(A,q,P) *is equivalent to A as soon as one of the following conditions is verified:*

1. *$q\overrightarrow{\subseteq}_k P$ and for all $p \in P$, $p\overrightarrow{\subseteq}_k q$,*
2. *$q\overleftarrow{\subseteq}_k P$ and for all $p \in P$, $p\overleftarrow{\subseteq}_k q$.*

This kind of reduction was pointed out by Amilhastre, Janssen and Vilarem in [1] for the case of homogeneous languages, that is, languages which are contained in Σ^h for some $h \in \mathbb{N}$. Let $k = n$ in Proposition 7 and consider that $\mathcal{L}(A)$ is an homogeneous language, a pair (q,P) which satisfies condition 1 is called a *union reduction* by Amilhastre *et al.*

8 Computing the exact inequality relations

Let $k \leq n$, computing the relations $\overrightarrow{\subseteq}_k$ and $\overleftarrow{\subseteq}_k$ has a polynomial complexity w.r.t. n, but the complexity grows exponentially w.r.t. k, which is unavoidable since computing $\overrightarrow{\subseteq}$ and $\overleftarrow{\subseteq}$ is known to be $\mathcal{NP}$-hard [7]. Nevertheless, $\overrightarrow{\subseteq}$ and $\overleftarrow{\subseteq}$ may be computed if we are able to determinize A and $\overline{A}$ with a reasonable complexity. Of course, proceeding this way does not make sense if our aim is to prevent a combinatorial blow up during the determinization by first reducing A.

This method is a straightforward application of the notion of characteristic events, which can be found in [11] and is reformulated in a way closer to our context in [4].

Let $\mathcal{A} =< Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}}, I_{\mathcal{A}}, F_{\mathcal{A}} >$ be the minimal DFA of $\mathcal{L}$ obtained by subset determinization from A, so that states of $\mathcal{A}$ are subsets of Q. Let $\mathcal{B} =< Q_{\mathcal{B}}, \Sigma, \delta_{\mathcal{B}}, I_{\mathcal{B}}, F_{\mathcal{B}} >$ be the minimal DFA of $\overline{\mathcal{L}}$ obtained by subset determinization from $\overline{A}$, so that states of $\mathcal{B}$ are also subsets of Q.

For all $q \in Q$, let $\lambda(q) = \{P \in Q_{\mathcal{A}} \mid q \in P\}$ and $\rho(q) = \{P \in Q_{\mathcal{B}} \mid q \in P\}$.

Proposition 8 *Let $q, p \in Q$, we have $q \overrightarrow{\subseteq} p$ if and only if $\rho(q) \subseteq \rho(p)$ and $q \overleftarrow{\subseteq} p$ if and only if $\lambda(q) \subseteq \lambda(p)$.*

This property leads to an $\mathcal{O}(n^2 N)$-time complexity algorithm for computing $\overleftarrow{\subseteq}$ and $\overrightarrow{\subseteq}$, where $N = \max(|Q_{\mathcal{A}}|, |Q_{\mathcal{B}}|)$.

9 Practical tests

We have carried out some tests for the first approximation, using randomly generated automata. For random automata, the average reduction ratio is correlated with the probability that two given right or left languages are equal. This probability decreases exponentially while the size of the alphabet or the size of the automaton is increasing, which appears on the tests. Reduction of NFAs, like reduction of DFAs, makes sense only if one is manipulating automata with a high level of redundancy, which is the case in most of applications. Tests with random automata just make the comparison possible between reduction by equivalence relations and reduction with first order preorders in the most general case. Preorders appear to be at least twice more efficient. Figures 3 and 4 give respectively results with a two and a three letter alphabet. The reduction algorithm that is considered is the algorithm REDUCTION deprived of Lines 2 – 7.

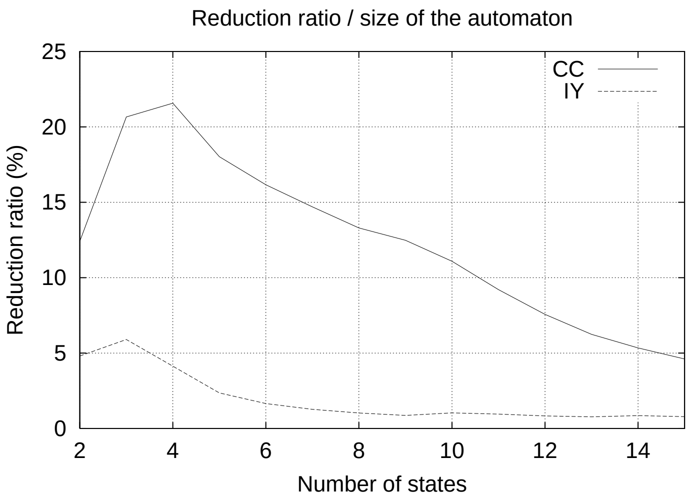

Figure 3: Tests on randomly generated automata over a 2-letter alphabet.

10 Conclusion

We would like to mention first that these reduction techniques work well with some slightly different definitions of automata, like labeled transition systems with finite recognition, which are automata without final states. Recognized words are words that do not cause a deadlock.

Our opinion is that the most interesting algorithm is the one based on the first order approximation, which has a reasonable worst case complexity, and should prove to be very fast in practice. To our knowledge, this algorithm is the most efficient one in this level of performance. Higher order approximations may be useful if one really cares about the succinctness of the NFA.

We tried to be the most exhaustive as we could in the domain of reduction by means of the preorders related to regular inequalities. But this does not exhaust the realm of reduction heuristics. In particular, we restrained us to algorithms which only proceed by merging states. Studies about the full minimization of NFAs [11, 12, 4] revealed that to reduce an NFA, we also

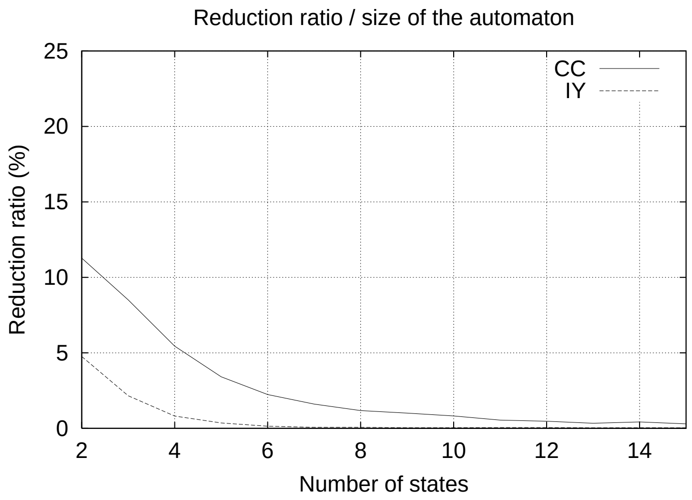

Figure 4: Tests on randomly generated automata over a 3-letter alphabet.

have to split some states, in order to merge them differently. Heuristics in this domain may overflow the imagination, but we can hardly expect them to produce a fast algorithm.

Acknowledgments

We wish to thank Dimitri Guillot for its implementation of the first order approximation.

References

[1] J. Amilhastre, P. Janssen, and M.C. Vilarem. FA minimization heuristics for a class of finite languages. In O. Boldt and H. Jürgensen, editors, *Automata Implementation, WIA'99, Lecture Notes in Computer Science*, volume 2214, pages 1–12. Springer, 2001.

[2] V. Antimirov. Rewriting regular inequalities. *Lecture notes in computer science*, 965:116–125, 1995.

[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] J.-M. Champarnaud and F. Coulon. Theoretical study and implementation of the canonical automaton. *Fundamenta Informaticae*, 55:23–38, April 2003.

[5] J.-M. Champarnaud, F. Coulon, and T. Paranthoen. Compact and fast algorithms for regular expression search. To appear in IJCM.

[6] John E. Hopcroft. An $n \log n$ algorithm for minimizing the states in a finite automaton. In Z. Kohavi, editor, *The Theory of Machines and Computations*, pages 189–196. Academic Press, 1971.

[7] H. Hunt, D. Rosenkrantz, and T. Szymanski. On the equivalence, containment and covering problems for the regular and context-free languages. *J. Comput. System Sci.*, 12:222–268, 1976.

[8] L. Ilie and S. Yu. Algorithms for computing small NFAs. In K. Diks and W. Rytter, editors, *Lecture Notes in Computer Science*, volume 2420, pages 328–340. Springer, 2002.

[9] J. Jaja. *An introduction to parallel algorithms.* Addison-Wesley, 1992.

[10] T. Jiang and B. Ravikumar. Minimal NFA problems are hard. *SIAM J. Comput. Vol 22, No 6*, pages 1117–1141, 1993.

[11] T. Kameda and P. Weiner. On the state minimization of nondeterministic finite automata. *IEEE Trans. Comp.*, C(19):617–627, 1970.

[12] H. Sengoku. Minimization of nondeterministic finite automata. Master's thesis, Kyoto University, 1992.

[13] B.-W. Watson and J. Daciuk. An efficient incremental DFA minimization algorithm. *Natural Language Engineering*, 9(1):49–64, march 2003.

[14] S. Wu and U. Manber. Fast text searching algorithm allowing errors. In *Communication of the ACM*, 31, pages 83–91, October 1992.