Mergible States in Large NFA

Cezar Câmpeanu [a] Nicolae Sântean [b] Sheng Yu [b]

[a] *Department of Computer Science and Information Technology*
University of Prince Edward Island, Charlottetown, PEI C1A 4P3, Canada

[b] *Department of Computer Science*
University of Western Ontario, London, ON N6A 5B8, Canada

Abstract

Quite often, trivial problems stated for Deterministic Finite Automata (DFA) are surprisingly difficult for the nondeterministic case (NFA). In any non-minimal DFA for a given regular language, we can find two equivalent states wich can be "merged" without changing the accepted language. This is not the case for NFA, where we can have non-minimal automata with no "mergible" states. In this paper, we prove a very basic result for NFA, that for a given regular language, any NFA of size greater than a computable constant must contain mergible states. Even more, we parameterized this constant in order to guarantee groups of an arbitrary number of mergible states.

Key words: Nondeterministic finite automata, mergible states, number of states, equivalent states

1 Introduction

Deterministic Finite Automata are among the simplest structures in Formal Language Theory. Therefore, many interesting properties of DFA were the subject of early developments in this area. The existence of a finite number of Myhill-Nerode equivalence classes for regular languages is an example of such properties. As a consequence of Myhill-Nerode Theorem, all DFAs for a given regular language with a number of states greater than the index of the corresponding Myhill-Nerode equivalence must have equivalent states (states that can be "merged" into one state, preserving the recognized language). If we try to apply a similar idea to NFAs, we discover that merging states may be done in different ways (preserving all transitions, or just some of them) and that Myhill-Nerode equivalence is not powerful enough to detect such states, or to at least guarantee their existence. Moreover, so far there are no efficient algorithms (computational complexity wise) for reducing the number of states and transitions of NFAs.

In this paper we propose a method to detect(guarantee) mergible states in NFA solely based on their size (number of states). Our results confirm the intuition that, for a given regular language,

one cannot construct an arbitrarily large NFA with no mergible states. More precisely, we answer to the following:

Problem 1 Let L be an arbitrary regular language, and $k \geq 2$ an arbitrary integer. Does it exist (and if "yes", effectively construct it) a constant $E_{L,k}$ such that any ε-NFA of size at least $E_{L,k}$ has at least k mergible states?

In spite of its descriptive simplicity, the problem turned out to be quite difficult to solve by means of just classical tools. In order to alleviate such technical difficulties, we define for each state in an NFA two new equivalence relations on words derived from the Myhill-Nerode equivalence and syntactic congruence of the given regular language. In the first section we introduce basic notions and notations, and we prove an initial property of states in large NFA. In particular, we solve the problem for the easiest case, of finite languages. In Section 3 we solve the problem for the general case, i.e., for arbitrary regular languages.

2 Preliminaries and Initial Results

We begin this section with `Dirichlet's Box Principle` (also known as pigeonhole principle), extensively used throughout this paper:

"Given n boxes (with $n \geq 1$) containing $m > n$ objects altogether, there exist at least one box containing at least two objects."

We can generalize this principle as following: given $n \geq 1$ boxes containing $m \geq (k-1)n+1$ objects altogether, $k \geq 2$, there exist a box containing at least k objects. (For further reference consult [2, p.38])

Let n be a positive integer. By S_n^j we denote the `Stirling number of the second kind`, which gives the number of ways to partition a set of n elements into j nonempty disjoint subsets (see [6, p.65] or [3, §2.6.2]). It is given by the formula

$$S_n^j = \frac{1}{j!}\sum_{i=0}^{j-1}(-1)^i C_j^i (j-i)^n \quad . \tag{1}$$

Then, the number of all distinct partitions of the set $\{1,\ldots,n\}$ - called `Bell number`, as in [3, §2.6.3] - will be denoted by $P(n)$, given by

$$P(n) = \sum_{j=1}^{n} S_n^j \quad . \tag{2}$$

Let A, B be two arbitrary sets. The `Cartesian product` of A and B is denoted by $A \times B = \{(a,b) \mid a \in A, b \in B\}$. A `binary relation` over A and B is a subset R of $A \times B$. The `inverse`

`relation` of R is $R^{-1} = \{(b,a) \mid (a,b) \in R\}$. The `identity` of A is the relation $id_A = \{(x,x) \mid x \in A\}$. The `composition of two relations` $R_1 \subseteq A \times B$ and $R_2 \subseteq B \times C$ is the relation $R_2 \circ R_1 = \{(a,c) \mid \exists b \in B : (a,b) \in R_1 \text{ and } (b,c) \in R_2\}$. We say that a relation R_1 is `coarser` than relation R_2 if $R_2 \subseteq R_1$. $R \in A \times A$ is an `equivalence` over A if it is reflexive ($id_A \subseteq R$), symmetric ($R^{-1} = R$), and transitive ($R \circ R \subseteq R$). A `binary operation` over A is a total function $\odot : A \times A \to A$. We use the infix notation to denote binary operations: $a \odot b := \odot(a,b)$. An equivalence R over A is `right-invariant` with respect to $\odot$ if $(a,b) \in R \Rightarrow (a \odot c, b \odot c) \in R, \forall c \in A$, and is `left-invariant` if $(a,b) \in R \Rightarrow (c \odot a, c \odot b) \in R, \forall c \in A$. R is `right-invariant with respect to` $C \subseteq A$ if $(a,b) \in R \Rightarrow (a \odot c, b \odot c) \in R, \forall c \in C$. Given an equivalence R over A and an element $a \in A$, the `equivalence class` of a with respect to R is the set $[a]_R := \{b \in A \mid (a,b) \in R\}$. If a subset D is included in one class of R, than we used the notation $[D]_R$ to denote the including class. All equivalence classes of R represent a `partition` of A, i.e., they do not overlap and they cover A. The set of all classes of R is called the `quotient` of A by R, denoted by A/R. The `index` of R is the cardinal of A/R, denoted by $|A/R|$. R is a `congruence` if it is both right- and left- invariant with respect to $\odot$ (also said that R is compatible with $\odot$, i.e., that $(a,b) \in R \Rightarrow (x \odot a \odot y, x \odot b \odot y) \in R, \forall x,y \in C$). Consult [5] for more information on basic algebraic concepts.

Remark 1 Let A be an arbitrary set and R be an equivalence over A of finite index, namely n. One can observe that there exist at most $P(n)$ distinct equivalences R' over A, such that $R \subseteq R'$. Indeed, since R' is coarser than R, it follows that any equivalence class of R is included in some equivalence class of R', hence the index of R' is smaller than that of R. Furthermore, R' induces an equivalence relation over A/R, given by

$$[x]_R \sim [y]_R \quad \Leftrightarrow \quad (x,y) \in R' \ . \tag{3}$$

(It can easily verify that it is an equivalence over A/R). Since $\sim$ is an equivalence over a set with n elements, it is clear that there exist at most $P(n)$ such distinct equivalences. The mapping

$$\phi : A/R' \to (A/R)/\sim \quad , \quad \phi\left([x]_{R'}\right) := \left[[x]_R\right]_\sim \tag{4}$$

is a bijection, hence there can exist at most $P(n)$ equivalences R' as well. The relationship between various quotient sets is depicted by the commutative diagram in Fig.1, where π, π' and $\pi^\sim$ are the canonical projections of R, R' and $\sim$ (the canonical projection of an equivalence maps an element onto its corresponding equivalence class).

$$\begin{array}{ccc} A & \xrightarrow{\ \pi\ } & A/R \\ {\scriptstyle \pi'}\downarrow & & \downarrow{\scriptstyle \pi^\sim} \\ A/R' & \xrightarrow[\phi, bij.]{} & (A/R)/\sim \end{array}$$

Fig. 1. The number of distinct equivalences $R' \supseteq R$ is bound by $P(\,|A/R|\,)$.

Let Σ be an alphabet, i.e., a nonempty, finite set of symbols. By Σ^* we denote the set of all finite words (strings of symbols) over Σ and by ε we denote the empty word (a word having zero sym-

bols). The operation of concatenation (juxtaposition) of two words u and v is denoted by $u \cdot v$, or simply uv.

Definition 1 *([4]) A* `nondeterministic finite automaton` *over* Σ*, NFA for short, is a tuple* $A = (Q, \Sigma, \delta, q_0, F)$*, where*

(1) Q *is a finite set of states,*
(2) $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ *is a next-state function, and*
(3) q_0 *is an initial state and* $F \subseteq Q$ *is a set of final states.*

The next-state (or transition) function is extended to work on words as following: $q \in \delta(q, \varepsilon), \forall q \in Q$ and $\delta(q, aw) = \delta(\delta(q, a), w), \forall a \in \Sigma, w \in \Sigma^*$ and $q \in Q$. The language recognized by A is $L(A) = \{w \in \Sigma^* \mid \delta(q_0, w) \cap F \neq \emptyset\}$ (a `regular language` over Σ is any language recognized by some NFA over Σ). A state of A is `accessible` if there exists a path in the associated transition graph starting from q_0 and ending in that state. A state is `coaccessible` if there exists a path form that state to some final state. A state is `useful` if it is both accessible and coaccessible. A NFA is `trim` if it has only useful states.

Note 1 Throughout this paper we consider only trim NFA. Notice that by an NFA we actually understand ε*-NFA, i.e., NFA which may have* ε*-transitions.*

For background knowledge in automata theory, the reader may refer to [7,8,4,9].

Let L be a regular language and $A = (Q, \Sigma, \delta, q_0, F)$ be an NFA for L with $\mid Q \mid = n$. By the `size` of A we understand the number of its states, namely n. For some state $q \in Q$ we denote by

(1) L_q the `left language of` q, obtained by setting q to be the only final state of A, i.e., $L_q = \{w \in \Sigma^* / q \in \delta(q_0, w)\}$,
(2) R_q the `right language of` q, obtained by setting q to be the initial state of A, i.e., $R_q = \{w \in \Sigma^* / \delta(q, w) \cap F \neq \emptyset\}$,
(3) I_q the `inner language of` q, obtained by setting q to be both the initial and the only final state in A, i.e., $I_q = \{w \in \Sigma^* / q \in \delta(q, w)\}$,

as illustrated in Fig.2.

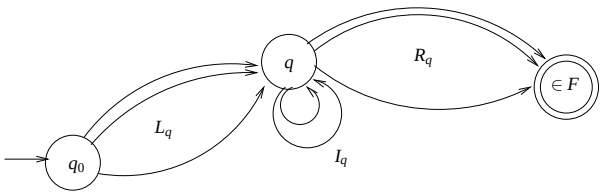

Fig. 2. The "(L)eft", "(I)nner" and "(R)ight" language of a state q.

Denote by $pref(L)$ the set of all prefixes of words in L and by $suf(L)$ the set of all suffixes of words in L. Notice that $\forall q \in Q : \varepsilon \in I_q$, $I_q^* = I_q$ and $I_q \subseteq suf(L_q) \cap pref(R_q)$. Notice also that $I_{q_0} = L_{q_0}$ and that $\forall q \in F : I_q \subseteq R_q$.

Considering these observations, one can verify that A induces a decomposition of L written as a

union of languages as following:

$$L = \bigcup_{q \in Q} L_q I_q R_q = \bigcup_{q \in Q} L_q R_q \ . \tag{5}$$

Definition 2 *Two distinct states p and q are* `mergible` *in A if and only if by adding $\varepsilon-$ transitions from one state to the other the newly obtained automaton accepts the same language L.*

More formally, let A' be the automaton obtained from A by adding $\delta(p,\varepsilon) = q$ and $\delta(q,\varepsilon) = p$ to the transition table of A. Then p and q are mergible in A if and only if $L(A) = L(A') = L$ (see Fig.3).

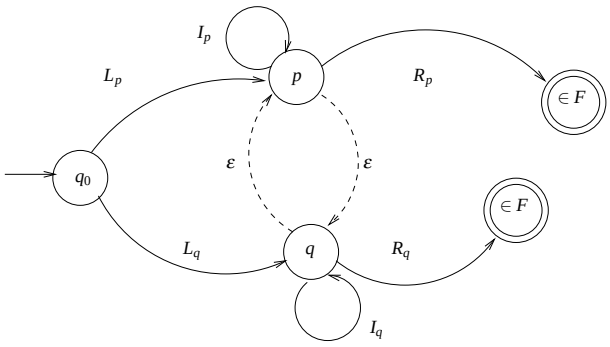

Fig. 3. p and q are mergible if L does not change when adding the dotted transitions.

Remark 2 A necessary and sufficient condition for ensuring that p and q are mergible is:

$$\begin{cases} (L_p I_q \cup L_q)(I_p I_q)^* R_p \subseteq L \\ \\ (L_q I_p \cup L_p)(I_q I_p)^* R_q \subseteq L \end{cases} \tag{6}$$

The definition of mergible states can readily be generalized to $k \geq 2$ states as following: the states $q_1, \ldots, q_k$ are mergible if by adding ε-transitions in between all states q_i with $1 \leq i \leq k$, the newly created automaton will still accept language L. The following is a useful characterization of mergible states.

Lemma 1 *(working definition) Let $p_1, \ldots, p_k$ be arbitrary states in A. Then these states are mergible if and only if*

$$\Big(\bigcup_{i=1}^{k} L_{p_i}\Big) \cdot \Big(\bigcup_{i=1}^{k} I_{p_i}\Big)^* \cdot \Big(\bigcup_{i=1}^{k} R_{p_i}\Big) \subseteq L \ . \tag{7}$$

Proof: It can be proved either directly or, for $k = 2$, by relating it to Remark 2. Remark 2 can be proved by induction. Both proofs are left to the reader. □

Remark 3 Given an NFA of size n which has a group of k mergible states, there exists an equivalent NFA of size $n-k+1$. Indeed, we can replace all k mergible states of the initial automaton with a single state which will consolidate the inward and outward transitions of all states of the group. By the definition of mergible states, we obtain an equivalent NFA.

Fig.4(a) shows that the property of being mergible is not transitive. Also notice that any $j < k$ states of a group of $k(>2)$ mergible states are mergible; however the reciprocal does not hold - as exemplified in Fig.4(b).

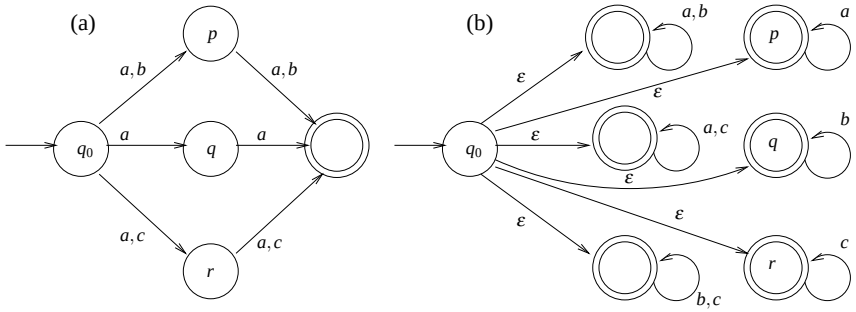

Fig. 4. (a) p and q are mergible, q and r are mergible; however p and r are not. (b) p,q,r are mergible two by two, however $\{p,q,r\}$ is not a group of mergible states.

Unlike the case of deterministic finite automata (DFA), a non-minimal (size-wise) NFA may have no mergible states. An example of such situation is given in Fig.5, which shows a non-minimal NFA (state q can readily be eliminated) with none of its states mergible. A language $L \subseteq \Sigma^*$ induces

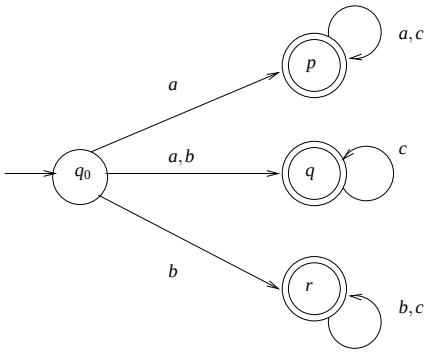

Fig. 5. State q is obsolete; however, no mergible states are present.

two important equivalence relations over Σ^*:

(1) `Myhill-Nerode Equivalence:` $u \equiv_L v \Leftrightarrow \forall z \in \Sigma^* : (uz \in L \Leftrightarrow vz \in L)$
(a right-invariant equivalence),

(2) `Syntactic Congruence:` $u \cong_L v \Leftrightarrow \forall x,y \in \Sigma^* : (xuy \in L \Leftrightarrow xvy \in L)$.

We denote by N_L the index of $\equiv_L$ and by H_L the index of $\cong_L$. It is well known that if L is regular, then both N_L and H_L are finite (consult [1, Th. 4.5]).

In the following we define the first out of two equivalence relations on words introduced in this

paper – equivalences which are central to the proof of existence of mergible states in large NFA.

Definition 3 *Let $A = (Q, \Sigma, \delta, q_0, F)$ be an NFA for a regular language L. For any state $q \in Q$ define the following relation over Σ^*:*

$$\forall u, v \in \Sigma^* : \quad u \sim_q v \Leftrightarrow \left[\forall z \in R_q : \quad (uz \in L \Leftrightarrow vz \in L)\right] \ . \tag{8}$$

Notice that this relation is derived from Myhill-Nerode equivalence by restricting the domain of the "probe" word z to R_q. Clearly $\sim_q$ is coarser than $\equiv_L$.

Lemma 2 *The relation $\sim_q$ has the following properties:*

(1) $\sim_q$ *is an equivalence (easily verifiable).*
(2) $(\equiv_L) \subseteq (\sim_q)$*; consequently,* $|\ \Sigma^*/\sim_q| \leq N_L = |\ \Sigma^*/\equiv_L|$.
(3) $(\cap_{q\in Q} \sim_q) = (\equiv_L)$.
(4) L_q *is included in one class of* $\sim_q$*, class denoted by* $[L_q]_{\sim_q}$*. In other words,*

$$q \in \delta(s_0, u) \cap \delta(s_0, v) \quad \Rightarrow \quad u \sim_q v \ . \tag{9}$$

(5) There are at most $P(N_L)$ *distinct equivalences* $\sim_q$*, i.e.,* $|\ \{\sim_q\}_{q\in Q}\ | \leq P(N_L)$.

Proof: Property (5) is a consequence of (2) and Remark 1; the rest of the proof is left to the reader. □

Anticipating the use of property (5) of Lemma 2, we observe that if our NFA has more than $P(N_L)$ states, then there will certainly exist at least two distinct states p and q in Q such that $\sim_p = \sim_q$ (by Dirichlet's box principle). Moreover, given a regular language and a parameter k, all large enough NFA for the language must have at least k states $q_1, \ldots, q_k$ verifying $\sim_{q_1} = \ldots = \sim_{q_k}$.

Lemma 3 *Let L be a regular language and $k \geq 2$ an arbitrary integer. Any NFA of size at least $M_{L,k}$, where*

$$M_{L,k} = (k-1) \cdot N_L \cdot P(N_L) + 1 \ , \tag{10}$$

has at least k states $\{q_1, \ldots, q_k\}$, such that

(1) $\sim_{q_1} = \ldots = \sim_{q_k}$ $(:= \sim)$ *, and*
(2) $[L_{q_1}]_\sim = \ldots = [L_{q_k}]_\sim$.

Proof: Let $A = (Q, \Sigma, \delta, q_0, F)$ be an NFA for L with $|\ Q\ | \geq M_{L,k}$. Since $|\ Q\ | \geq (k-1) \cdot N_L \cdot P(N_L) + 1$ we infer that there exist at least $n = (k-1) \cdot N_L + 1$ states $p_1, \ldots, p_n$ such that $\sim_{p_1} = \ldots = \sim_{p_n}$ (we generically denote this equivalence as $\sim$). But then, among all these states, there exist at least k states $q_1, \ldots, q_k$ with their left languages belonging to a same equivalence class of $\sim$. This is true since the index of $\sim$ is at most N_L and each of the $(k-1)N_L + 1$ left languages is included in a class of $\sim$. Then $q_1, \ldots, q_k$ is a group of states verifying the requirements of our theorem. Here we used twice Dirichlet's box principle. □

Lemma 4 *Let L be a regular language and A a corresponding NFA. If there exist $k(\geq 2)$ states $q_1, \ldots, q_k$ in A such that*

(1) $\sim_{q_1} = \ldots = \sim_{q_k} (:= \sim)$ *, and*
(2) $[L_{q_1}]_\sim = \ldots = [L_{q_k}]_\sim$ *,*

then

$$(\bigcup_{i=1}^{k} L_{q_i}) \cdot (\bigcup_{j=1}^{k} R_{q_j}) \subseteq L \ . \tag{11}$$

Proof: Take $u \in L_{q_i}$ and $z \in R_{q_j}$ with $i, j \in \{1, \ldots, k\}$ arbitrarily chosen. Since all states are useful, there exists a word $v \in L_{q_j}$, hence $vz \in L$. But since $[L_{q_i}]_\sim = [L_{q_j}]_\sim$, it follows that $u \sim v$. Then $u \sim_{q_j} v$ and since $z \in R_{q_j}$ and $vz \in L$, it follows that $uz \in L$. Therefore, $L_{q_i} \cdot R_{q_j} \subseteq L$ for arbitrary $i, j \in \{1, \ldots, k\}$. □

An application of the previous two lemmas is the solution to Problem 1 for finite languages, as captured in the following result.

Corollary 1 *Any NFA for a finite language L, of size at least $M_{L,k}$, $k \geq 2$, has at least k mergible states.*

Proof: Let A be a NFA for L of size at least $M_{L,k}$. Consequence of Lemma 3 and 4, there exist k states $q_1, \ldots, q_k$ in A such that

$$(\bigcup_{i=1}^{k} L_{q_i}) \cdot (\bigcup_{j=1}^{k} R_{q_j}) \subseteq L \ . \tag{12}$$

It now suffices to observe that any state q in a trim NFA for a finite language has $I_q = \{\varepsilon\}$. Then

$$(\bigcup_{i=1}^{k} L_{q_i}) \cdot (\bigcup_{i=1}^{k} I_{q_i})^* \cdot (\bigcup_{i=1}^{k} R_{q_i}) = (\bigcup_{i=1}^{k} L_{q_i}) \cdot (\bigcup_{j=1}^{k} R_{q_j}) \subseteq L \ , \tag{13}$$

hence $q_1, \ldots, q_k$ are mergible. □

We essentially proved that a large enough NFA for a finite language must satisfy the hypothesis of Lemma 4. Notice that satisfying only condition (1.) of Lemma 4 does not suffice. Indeed, consider the example shown in Fig.6. The states p and q satisfy the condition $\sim_p = \sim_q$, since $\Sigma^*/_{\sim_p} = \Sigma^*/_{\sim_q} = \Big\{\{a\}, \{b\}, \Sigma^* \backslash \{a, b\}\Big\}$. However, p and q are not mergible.

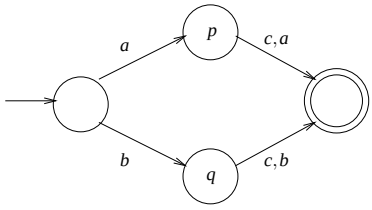

Fig. 6. The states p and q are not mergible despite the fact that $\sim_p = \sim_q$.

3 Large NFA - the General Case

In Section 2 we have defined a useful equivalence relation on words, derived from the Myhill-Nerode equivalence. We have used this new equivalence and its properties to solve Problem 1 for finite languages. For the general case, this equivalence does not suffice. Therefore, let us first define a second equivalence, this time derived from the syntactic congruence ($\cong_L$).

Definition 4 *Let L be a regular language and $A = (Q.\Sigma, \delta, q_0, F)$ a corresponding NFA. For any state $q \in Q$ we associate the following relation on words:*

$$\forall u, v \in \Sigma^* : \quad u \approx_q v \Leftrightarrow \left[\forall (x,y) \in L_q \times R_q : \quad (xuy \in L \Leftrightarrow xvy \in L)\right] \ . \tag{14}$$

Notice that this relation is derived from the syntactic congruence of L by restricting the domain of the "probe" pair (x,y) to $L_q \times R_q$. Clearly $\approx_q$ is coarser than $\cong_L$.

Lemma 5 *The relation $\approx_q$ has the following properties:*

(1) $\approx_q$ *is an equivalence (can easily be verified).*
(2) $(\cong_L) \subseteq (\approx_q)$*; consequently,* $|\ \Sigma^* / \approx_q| \leq H_L = |\ \Sigma^* / \cong_L|$.
(3) $(\cap_{q \in Q} \approx_q) = (\cong_L)$.
(4) I_q *is included in one class of* $\approx_q$*. In other words,*

$$q \in \delta(q,u) \cap \delta(q,v) \quad \Rightarrow \quad u \approx_q v \ . \tag{15}$$

Consequently, $I_q \subseteq [\varepsilon]_{\approx_q}$*, since* $\varepsilon \in I_q$.
(5) There are at most $P(H_L)$ *equivalences* $\approx_q$*, i.e.,* $|\ \{\approx_q\}_{q \in Q}\ | \leq P(H_L)$.
(6) $\approx_q$ *is right-invariant with respect to* I_q *(if* $u \approx_q v$ *and* $z \in I_q$ *then* $uz \approx_q vz$*).*

Proof: Property (5) follows from property (2) and Remark 1. For property (6), consider $u \approx_q v$, and choose an arbitrary $z \in I_q$. We must prove that $uz \approx_q vz$. Let $(x,y) \in L_q \times R_q$. We prove that $xuzy \in L \Leftrightarrow xvzy \in L$ (we prove only one implication, the relation being symmetric).

For the implication to the right, suppose that $xuzy \in L$. We have $z \in I_q$ and $y \in R_q$, therefore, we deduce that $zy \in R_q$. Since $(x, zy) \in L_q \times R_q$, $u \approx_q v$ and $xuzy \in L$, it follows that $xvzy \in L$.

The rest of the proof is left to the reader. □

Note 2 Notice that $\approx_q$ is not necessarily a right-invariant equivalence. It is just right-invariant with respect to I_q.

In order to be able to use both relations $\approx_q$ and $\sim_q$ simultaneously, we require a mean to couple them via their equivalence classes. The following corollary provides a solution.

Lemma 6 *If A is an NFA for a regular language L and q is an arbitrary state in A, then*

$$L_q[\varepsilon]_{\approx_q} \subseteq [L_q]_{\sim_q} \ . \tag{16}$$

Proof: We first observe that $L_q[\varepsilon]_{\approx_q} R_q \subseteq L$. Indeed, let $u \in L_q$, $v \in [\varepsilon]_{\approx_q}$ and $w \in R_q$. Since $\varepsilon \approx_q v$, $(u,w) \in L_q \times R_q$ and $uw \in L$, we obtain that $uvw \in L$.

The fact that $L_q[\varepsilon]_{\approx_q} R_q \subseteq L$ implies that $L_q[\varepsilon]_{\approx_q}$ is included in one equivalence class of $\sim_q$ (by the definition of $\sim_q$). But since $L_q \subseteq L_q[\varepsilon]_{\approx_q} \cap [L_q]_{\sim_q}$, this class can only be $[L_q]_{\sim_q}$. Concluding, $L_q[\varepsilon]_{\approx_q} \subseteq [L_q]_{\sim_q}$. $\square$

This property allows us to prove a result similar to Lemma 4, with the improvement of introducing $[\varepsilon]_\approx$ in between the two unions of left and right languages. $[\varepsilon]_\approx$ will later be used as a mean to accommodate the inner languages.

Corollary 2 *Let A be an NFA for a regular language L and $q_1, \ldots, q_k$ states in A such that*

(1) $\approx_{q_1} = \approx_{q_2} = \ldots = \approx_{q_k} (:= \approx)$,
(2) $\sim_{q_1} = \sim_{q_2} = \ldots = \sim_{q_k} (:= \sim)$, *and*
(3) $[L_{q_1}]_\sim = \ldots = [L_{q_k}]_\sim$ *(i.e $\bigcup_{i=1}^k L_{q_i}$ is included in one class of $\sim$) .*

Then the following relation holds:

$$\Big(\bigcup_{i=1}^{k} L_{q_i}\Big)[\varepsilon]_\approx \Big(\bigcup_{j=1}^{k} R_{q_j}\Big) \subseteq L \ . \tag{17}$$

Proof: We prove that $L_{q_i}[\varepsilon]_\approx R_{q_j} \subseteq L$, $\forall i, j \in \{1, \ldots, k\}$. Arbitrarily choose $i, j \in \{1, \ldots, k\}$. We have the following relations:

$$L_{q_i}[\varepsilon]_\approx R_{q_j} = L_{q_i}[\varepsilon]_{\approx_{q_i}} R_{q_j} \subseteq [L_{q_i}]_{\sim_{q_i}} R_{q_j} = [L_{q_i}]_\sim R_{q_j} = [L_{q_j}]_\sim R_{q_j} \subseteq L \ . \tag{18}$$

We have used the fact that $L_{q_i}[\varepsilon]_\approx \subseteq [L_{q_i}]_{\sim_{q_i}}$ (by Lemma 6) and that $[L_{q_i}]_\sim = [L_{q_j}]_\sim$ by hypothesis. $\square$

In order to take into consideration the inner languages as well, it now suffices to relate them to $[\varepsilon]_\approx$ – as stated in the context of Corollary 2. The result follows.

Lemma 7 *Let A be an NFA and $q_1, \ldots, q_k$ be arbitrary states in A. If $\approx_{q_1} = \ldots = \approx_{q_k} (:= \approx)$ then*

$$\Big(\bigcup_{i=1}^{k} I_{q_i}\Big)^* \subseteq [\varepsilon]_{\approx} \ . \tag{19}$$

Proof: Let $z \in (\bigcup_{i=1}^{k} I_{q_i})^*$ and consider a factorization $z = z_1 \ldots z_n$ with $z_i \in (\bigcup_{j=1}^{k} I_{q_j})$, $\forall 1 \leq i \leq n$. We prove that $z \in [\varepsilon]_{\approx}$ by induction on n. The property is true for $n = 1$ since it is easy to notice that $\bigcup_{j=1}^{k} I_{q_j} \subseteq [\varepsilon]_{\approx}$ (from property (4) of Lemma 5). Assume that the property holds for an arbitrary n and choose $z_{n+1} \in \bigcup_{j=1}^{k} I_{q_j}$. It remains to prove that $z_1 \ldots z_n z_{n+1} \in [\varepsilon]_{\approx}$. Consider $z_{n+1} \in I_{q_t}$ for an arbitrary $t \in \{1, \ldots, k\}$. By induction hypothesis we have that $z_1 \ldots z_n \in [\varepsilon]_{\approx} = [\varepsilon]_{\approx_{q_t}}$. Since $z_{n+1} \in I_{q_t}$ and since $z_1 \ldots z_k \approx_{q_t} \varepsilon$, it follows that $z_1 \ldots z_n z_{n+1} \approx_{q_t} z_{n+1}$ (using property (6) of Lemma 5). But $z_{n+1} \approx_{q_t} \varepsilon$ since $\varepsilon \in I_{q_t}$, hence $z_1 \ldots z_n z_{n+1} \approx_{q_t} \varepsilon$. It follows that $z_1 \ldots z_n z_{n+1} \in [\varepsilon]_{\approx}$. $\square$

We now have sufficient ingredients for solving Problem 1 for the general case.

Theorem 1 *Let L be an arbitrary regular language and k a positive integer. There exists a constant $E_{L,k}$ (effectively constructed) such that any NFA for L of size at least $E_{L,k}$ has at least k mergible states.*

Proof: Let us define $E_{L,k}$ to be

$$E_{L,k} := M_{L,[(k-1)\cdot P(H_L)+1]} \ , \tag{20}$$

and prove that indeed it satisfies theorem's requirements. Let A be an NFA for L of size at least $E_{L,k}$. Applying Lemma 3, we infer that A has at least $n := (k-1)\cdot P(H_L) + 1$ states $p_1, \ldots, p_n$ such that

(1) $\sim_{p_1} = \ldots = \sim_{p_n} (:= \sim)$, and
(2) $[L_{p_1}]_{\sim} = \ldots = [L_{p_n}]_{\sim}$.

But among these states there are at least k states $q_1, \ldots, q_k$ such that $\approx_{q_1} = \ldots = \approx_{q_k} (:= \approx)$. This follows from the fact that there exist at most $P(H_L)$ distinct equivalences $\approx$ (we applied yet again Dirichlet's box principle). Summing up what we found so far, we proved that the NFA A has at least k states $q_1, \ldots, q_k$ which verify the following properties:

(1) $\approx_{q_1} = \ldots = \approx_{q_k} (:= \approx)$,
(2) $\sim_{q_1} = \ldots = \sim q_k (:= \sim)$, and
(3) $[L_{q_1}]_{\sim} = \ldots = [L_{q_k}] \sim$.

These relations allow us to apply Corollary 2, from which we infer that

$$\Big(\bigcup_{i=1}^{k} L_{q_i}\Big)[\varepsilon]_{\approx}\Big(\bigcup_{j=1}^{k} R_{q_j}\Big) \subseteq L \ . \tag{21}$$

But since $\approx_{q_1} = \ldots = \approx_{q_k} = \approx$, we can also apply Lemma 7, and establish that

$$\Big(\bigcup_{i=1}^{k} I_{q_i}\Big)^* \subseteq [\varepsilon]_{\approx} \ . \tag{22}$$

Then, by the relations (21) and (22), the following relations hold:

$$\Big(\bigcup_{i=1}^{k} L_{q_i}\Big)\Big(\bigcup_{i=1}^{k} I_{q_i}\Big)^*\Big(\bigcup_{i=1}^{k} R_{q_i}\Big) \subseteq \Big(\bigcup_{i=1}^{k} L_{q_i}\Big)[\varepsilon]_{\approx}\Big(\bigcup_{j=1}^{k} R_{q_j}\Big) \subseteq L \ , \tag{23}$$

hence $q_1, \ldots, q_k$ are mergible by Lemma 1. □

This result completes the solution to Problem 1.

4 Conclusions and Further Work

In this paper we studied the existence of mergible states in large NFA. We have proven that given a regular language, there is a certain size beyond which any corresponding NFA has mergible states. Moreover, we effectively determined a parameterized constant for this size, which guarantees arbitrarily many (given by the parameter) mergible states. During our work we mainly focussed on proving the existence of such constants and on effectively computing them. The constants we provided are very large, some involving imbricated Stirling numbers. Left for immediate future work is to find smaller such constants, preferably sharp lower bounds. Last, but not the least, it remains to apply our results in, for example, NFA minimization algorithms or in decidability problems for NFA involving "brute-force" techniques.

References

[1] J. Berstel and D. Perrin, *Theory of codes*, Academic Press, New York, London, 1985.

[2] G. Chartrand, *Introductory Graph Theory*, Dover, New York, 1985.

[3] J. M. Harris, J. L. Hirst, M. J. Mossinghoff, *Combinatorics and graph theory*, Springer-Verlag, New York, Berlin, Heidelberg, 2000.

[4] J. E. Hopcroft, J. D. Ullman, *Introduction to automata theory, languages, and computation*, 1st edition, Addison-Wesley, Reading, Massachusetts, 1979.

[5] J. M. Howie, *An introduction to semigroup theory*, Academic Press, New York, 1976.

[6] D. E. Knuth, *The art of computer programming*, vol. 1: Fundamental Algorithms, 3rd edition, Addison-Wesley, Massachusetts, 1997.

[7] A. Salomaa, *Theory of Automata.* Pergamon Press, Oxford, 1969.

[8] A. Salomaa, *Formal languages*, Academic Press, New York, 1973.

[9] S. Yu, Regular languages. In: *Handbook of Formal Languages, Vol. I.* (G. Rozenberg, A. Salomaa, eds.) pp. 41–110, Springer-Verlag, Berlin, 1997.