Fundamenta Informaticae XX (2005) 1–11
IOS Press

Results on Transforming NFA into DFCA*

Cezar CÂMPEANU†
Department of Computer Science and Information Technology,
University of Prince Edward Island, Charlottetown, P.E.I.,
Canada, C1A 4P3; email: cezar@sun11.math.upei.ca

Lila KARI‡
Department of Computer Science,
University of Western Ontario, London, Ontario,
Canada, N6A 5B7; email: lila@csd.uwo.ca

Andrei PĂUN§
Department of Computer Science, College of Engineering and Science,
Louisiana Tech University, Ruston, Louisiana,
P.O. Box 10348, LA-71272, USA; email: apaun@latech.edu

Abstract. In this paper we consider the transformation from (minimal) Non-deterministic Finite Automata (NFAs) to Deterministic Finite Cover Automata (DFCAs). We want to compare the two equivalent accepting devices with respect to their number of states; this becomes in fact a comparison between the expression power of the nondeterministic device and the expression power of the deterministic with loops device. We prove a lower bound for the maximum state complexity of Deterministic Finite Cover Automata obtained from Non-deterministic Finite Automata of a given state complexity n, considering the case of a binary alphabet. We show, for such binary alphabets, that the difference between maximum blow-up state complexity of DFA and DFCA can be as small as $2^{\lceil \frac{n}{2} \rceil - 2}$ compared to the number of states of the minimal DFA. Moreover, we show the structure of automata for worst case exponential blow-up complexity from NFA to DFCA. We conjecture that the lower bound given in the paper is also the upper bound. Several results clarifying some of the

*A preliminary version of the paper was presented at DCGARS 2004 conference.
†Work supported by Natural Sciences and Engineering Research Council of Canada (NSERC) grant 600089
‡Work supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada Research Chair Program to L.K.
§Work supported by Louisiana Board of Regents grant 32-0967-40766 and a LATECH-CenIT grant.

structure of the automata in the worst case are given (we strongly believe they will be pivotal in the upper bound proof).

Keywords: Finite automata, deterministic automata, nondeterministic automata, cover automata, state complexity

1. Introduction

State complexity of deterministic automata is important because it gives an accurate estimate of the memory space needed to store the automaton. In case of finite languages, Deterministic Finite Cover Automata reduce this space by taking into account the length of the longest word in the language, so that in practice the amount of memory necessary to store such a structure is significantly reduced (we refer the reader to [6] for examples of languages that exhibit such high degree of reduction in the number of states when they are described with a DFCA). In [1], [2], [3] it is proved that for a given finite language the state complexity of a minimal DFCA is always less than or equal to the state complexity of a DFA recognizing the same language. Using this idea, it is interesting to know whether this improvement can always be significant or not in the number of states of the automaton, since transforming a DFA to a DFCA is also time consuming, the best known algorithm has the time complexity $O(n \log n)$ (see [3] for a detailed description of the algorithm).

The main purpose of this paper is to study the state complexity of the transformation from NFA to DFCA. We will give a lower bound in the worst case for this transformation and also give some results that we expect will be important in proving the upper bound of the transformation.

In [5] it is given an upper bound for converting NFA to minimal DFA for finite languages and non-unary alphabets, and it is proved that the upper bound is reached in case of a binary alphabet. However, in the general case there is no result about the structure of states/transitions of these automata.

We consider this question important and prove in the section 4 of the paper some properties of such high complexity automata, for an arbitrary alphabet.

The unary case is not interesting for this particular problem, since for a language containing only a word of length $n-1$ (a^{n-1} if our alphabet has only the letter a), a minimal NFA has n states. The minimal DFA in this case has $n+1$ states, and the minimal DFCA has n states. The problem is solved, since if a minimal NFA has n states and the associated DFCA has more than n states, the DFCA is not minimal.

The main results of the paper is Theorem 1 , where we prove a lower bound for state complexity of NFA to DFCA transformations for the case of a binary alphabet, and the results in section 4 dealing with arbitrary alphabets.

We prove that in the worst case the number of states of a minimal DFCA for a finite language L over a binary alphabet generated by an n-state minimal NFA can be at least as high as $2^{n-t} - 2^{t-2} + 2^t - 1$, where $t = \lceil \frac{n}{2} \rceil$. Notice that this bound is just with 2^{t-2} states lower than the bound obtained in [5] for the worst case transformation from NFA to DFA.

In the next section we give some basic notations and in Section 3 we give an example of NFA of size n, for which the corresponding DFCA has at least $2^{n-t} - 2^{t-2} + 2^t - 1$ states, proving our lower bound.

The upper bound is not yet determined precisely as opposed to the results from [5]; the reason is that the similarity relation is more complex than equivalence relations (similarity is not transitive) making

the discussion more involved. In Section 4 we prove that if an NFA has a particular structure, the corresponding minimal DFCA cannot exceed our lower bound, thus restricting the number of cases that can produce a higher complexity.

2. Notations and Preliminary Results

The number of elements of a finite set A is $|A|$, the empty set $\emptyset$ has no elements, so $|\emptyset| = 0$. An alphabet is a finite non-empty set, usually denoted by Σ, and an element of Σ is a letter. A word is a finite sequence of letters, and the empty string, denoted by ε, is the word with no letters. The length of a string $w = w_1 \dots w_n$, $w_i \in \Sigma$, $1 \leq i \leq n$, is the number n of letters of the word and is denoted by $|w|$. The length of ε is 0. The set of all words over the alphabet Σ is denoted by Σ^* and the set of words of length k is Σ^k.

We assume the reader to be familiar with the basics in automata theory as contained in [4], [7].

A *deterministic finite automaton* (shortly, a DFA) A is a quintuple $A = (Q, \Sigma, \delta, q_0, F)$, where:

- Q is the finite set of states;
- Σ is the input alphabet;
- $\delta : Q \times \Sigma \longrightarrow Q$ is the state transition function;
- $q_0 \in Q$ is the starting state, and
- $F \subseteq Q$ is the set of final states.

A *nondeterministic finite automaton* A, (denoted in the following text as NFA), is a quintuple $A = (Q, \Sigma, \delta, q_0, F)$, where Q, Σ, q_0, and F are defined exactly the same way as for DFA, and $\delta : Q \times \Sigma \longrightarrow 2^Q$ is the transition function, where 2^Q denotes the power set of the finite set Q.

Let $A = (Q, \Sigma, 0, \delta, F)$ be a deterministic acyclic automaton. We denote the minimum and maximum level of a state q as $lev_A(q) = \min\{|w| \mid \delta(0, w) = q\}$ and respectively, by $Lev_A(q) = \max\{|w| \mid \delta(0, w) = q\}$.

The set of states of minimum and maximum level i is $lev_{A,i} = \{q \in Q \mid lev_A(q) = i\}$ and $Lev_{A,i} = \{q \in Q \mid Lev_A(q) = i\}$, respectively.

When the automaton A is understood, we can omit writing A as subscript in the previous notation.

Let $|\Sigma| = p$ be the number of letters in the alphabet Σ. Let L be a finite language over Σ with l the maximum length of the words in L. We denote by $N_L = (\Sigma, Q_N, \delta_N, 0, F_N)$ a minimal NFA with $L = L(N_L)$, and by $D_L = (\Sigma, Q_D, \delta_D, 0, F_D)$, the DFA obtained using the subset construction from the NFA N_L. Therefore, we consider without any loss of generality that since $|Q_N| = n$ is the number of states in NFA, then we can re-number the states from 0 to $n-1$: $Q_N = \{0, 1, \dots, n-1\}$.

Since N_L is minimal, then all states are useful and, also, there is a state $f \in F_N$ such that

1. for all $q \in Q_N$, there is $w \in \Sigma^*$ such that $f \in \delta_N(q, w)$, and
2. $\delta_N(f, a) = \emptyset$, for all $a \in \Sigma$.

Without any loss of generality, we may assume that in N_L the first state q_0 is 0, and the last final state is $n-1=f$.

For the following results and definitions, we refer the reader to [2].

A cover automaton for a language L with words of length less than or equal to l is a DFA accepting a cover language L', i.e., a language with the property that $L' \cap \Sigma^{\leq l} = L$. Two words x, y are L-similar if for any word z with $|z| \leq \min(l-|x|, l-|y|)$, we have that $xz \in L$ if and only if $yz \in L$, and write this $x \sim_L y$. Two words are L-dissimilar if they are not L-similar. We can/will omit the subscript when the language L is understood.

A sequence of words $x_1, x_2, \ldots, x_n$ is an L-dissimilar sequence if any two words in the sequence are L-dissimilar.

In the same way we did for words, we can define the notion of similar states with respect with the DFA D_L as follows: s is similar to q in D_L if for any word z of length less than or equal to $\min(l - lev_{D_L}(q), l - lev_{D_L}(s))$, $\delta_D(s,z) \in F_D$ if and only if $\delta_D(q,z) \in F_D$. We write this as $s \sim_{D_L} q$.

We can construct a minimal cover DFA $C_L = (\Sigma, Q_C, \delta_C, 0, F_C)$ using the DFA D_L by merging similar states. Please, note that minimal DFCA may not be unique, we may have several non isomorphic minimal DFCAs for the same language, but all these DFCAs have the same number of states.

The number of states in a minimal DFCA for a language L is equal to the length of any maximal dissimilar sequence, which is equal to the number of states in the minimal DFA minus the number of similarities on states in the minimal DFA (see [2] for the formal definitions and proofs).

Hence, the number of states of a minimal DFCA for L is less than or at most equal to the number of states in D_L (equality only when no states are similar in the DFA).

For $0 \leq i \leq l$, let us denote by $Q_{D,i} = \bigcup_{S \in lev_{D_L,i}} S$. Please note that $Q_{D,i} \subseteq Q_N$, while $lev_{D_L,i} \subseteq 2^{Q_N}$.

Using Theorem 3 given by Salomaa and Yu in [5], we conclude that

$$|Q_C| \leq |Q_D| \leq \frac{\left(2^{\lceil \frac{n \log_2 p}{1+\log_2 p} \rceil + 1} - 1\right)}{(p-1)}.$$

We investigate if this upper bound is also the lowest for the $|Q_C|$ in terms of n, and give a lower bound for the worst case.

In order to do this, we denote by $t = \min\{m \mid p^m \geq 2^{n-m}\} = \min\{m \mid m \geq \frac{n}{1+\log_2 p}\} = \lceil \frac{n}{1+\log_2 p} \rceil$. As we will see in the following, this number has a special role in separating states of N_L and D_L (we recall that by N_L we understand a minimal nondeterministic automaton for L, and by D_L the corresponding DFA obtained from N_L using the subset construction).

We set

$UB(n,p) = \frac{(p^t-1)}{(p-1)} + 2^{n-t} - 2^{n-t-2}$, $UB(n) = UB(n,2)$,

$LB(n,p) = \frac{(p^t-1)}{(p-1)} + 2^{n-t} - 2^{t-2}$, and $LB(n) = LB(n,2)$.

In the current paper we will prove that $LB(n)$ is the lower bound that can be reached.

We can see that $UB(n) = LB(n)$, if n is even. For $p=2$, the number $UB(n)$ is

$$UB(n) = 2^{n-t-1} + 2^{n-t-2} + 2^t - 1,$$

and

$$LB(n) = 2^{n-t} - 2^{t-2} + 2^t - 1 = \begin{cases} 2^{t-1} + 2^{t-2} + 2^t - 1, & \text{if } n \text{ is even} \\ 2^{t-2} + 2^t - 1, & \text{if } n \text{ is odd.} \end{cases}$$

3. The lower bound for the worst case DFCA complexity

In this section we provide examples to show that the number $LB(n)$ given in the previous section can be reached.

Theorem 3.1. For each integer $n > 1$, there exists a finite language $L \subseteq \{a, b\}$ such that L is accepted by a minimal acyclic n-state NFA, and any complete DFCA for L has at least $LB(n)$ states.

Proof:
Let $\Sigma = \{a, b\}$. We distinguish two cases: n can be either even or n is odd.

I. If n is even we consider the language $L_n = L'_n \cup L''_n$, $L'_n = \{w \mid w = w_1 b,\ |w| = t\}$, $L''_n = \{w \mid w = uava$, such that $|w| < n,$ and $|v| = \lfloor \frac{n}{2} \rfloor - 2\}$.

The language L_n is accepted by the nondeterministic automaton with n states $0,\ 1, ...,\ n-1$ with $\delta_N(i, a) = \{i + 1, t\}$, $\delta_N(i, b) = \{i + 1\}$ for all $0 \leq i \leq t - 2$, $\delta_N(i, a) = \delta_N(i, b) = \{i + 1\}$ for all $t \leq i \leq n - 3$, $\delta_N(t - 1, a) = \{t\}$, $\delta_N(t - 1, b) = \{n - 1\}$, and $\delta_N(n - 2, a) = \{n - 1\}$.

This NFA is presented in Figure 1 (please, recall that $f = n - 1$).

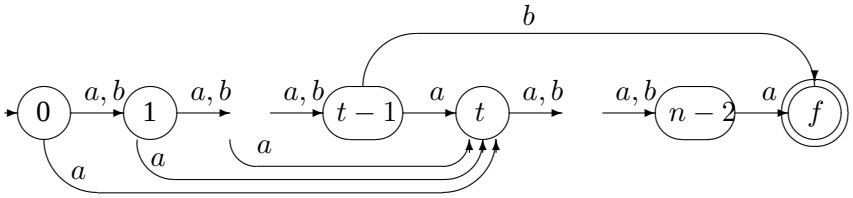

Figure 1. An example of NFA for which the DFCA reaches $LB(n)$ states.

We will show that there are at least $LB(n)$ dissimilar words with respect to L.

If two words of length less than or equal to t have different length, they are not equivalent. Indeed, let $x, y \in \Sigma^*$, $y \neq \varepsilon$, and $t \geq |x| = i > |y| = j$. Then $xb^{t-j} \notin L_n$, since its length is greater than t (so is not in L'_n) and also ends in b (thus, it is not in L''_n). But $yb^{t-j} \in L'_n$, since it has length t and the last letter is b ($j < i \leq t$ and $t - j > t - i \geq 0$), thus $x \not\sim_L y$, since $|yb^{t-j}| < |xb^{t-j}| = i + t - j \leq 2t - 1$ and $2t - 1$ is exactly the length of the longest words.

For ε and words of length t, we check similarities with words ending in b and with words ending in a. For the first case, $\varepsilon \notin L_n$, but $w = w_1 b \in L_n$, for all words with $|w_1| = t - 1$. For the second case, if $w = w_1 a$, $wa^{n-t-1} \in L$, but $a^{n-t-1} \notin L$, for all words with $|w_1| = t - 1$. Hence $\varepsilon \not\sim_L w$, for all w with $|w| = t$.

Now we consider the case when $|x| = |y| = i$ for $1 \leq i \leq t - 1$. We prove that if x is equivalent with y, they are equal. Indeed, if $x \equiv_L y$, and we append both words with another non-empty word, the

results must be both in L'_n or in L''_n, since they have the same last letter. Assuming that $x \neq y$, let k be the first position on which they differ. Without any loss of generality we can assume that on the position $k \geq 1$, x has a and y has b. We consider the word $z = a^{t-i+k-1}$, so xz has an a on position t counting from the right of the word, whereas yz has a b, so $yz \notin L''_n$. Because yz ends in a, it cannot be in L'_n either. Since, xz ends in a and $|xz| = t-i+k-1+i = t+k-1$ and $t \leq t+k-1 \leq 2t-1$, $xz \in L''_n$.

Hence, the words x and y are not equivalent; thus, all words of length at most $t-1$ are not equivalent.

We have proved that these words are also non-similar, since $|xz| = |yz| \leq 2t-1$.

Let us count the number of dissimilar words of length t. First, let us note that two words of length t are similar iff they are equivalent. It is easy to see that all the words $x = aw_2b \equiv y = bw_2b$ are equivalent, since they differ only on the first letter, they are both in the language, and for any word z of length greater than 1, $xz \in L''_n$ and $yz \in L''_n$ or $xz \notin L''_n$ and $yz \notin L''_n$ (the t-th letter from the right is the same). In what follows, we prove that all other words of length t are not equivalent. Let us consider two words x, y of length t having the same letter on the first position and having a different letter on position $k \geq 2$. We may assume that k is the greatest with this property, and on that position is the letter a in x and the letter b in y.

Let us consider the word $z = a^{k-1}$. Then $xz \in L''_n$, since ends in a ($k \geq 2$, thus $k-1 \geq 1$) and also has an a on the position t, counted from the right (the a on position k of x). But at the same time, $yz \notin L_n$. Since is longer than t, it is not in L'_n, and has a b on the position t counted from the right (the b on the position k of y), thus is not in L''_n either. So, we just proved that all the words that differ on a position greater than first letter are not equivalent, and not similar either.

Let us consider the words x and y of length t, that differ *only* on the first letter, so $x = aw_1$ and $y = bw_1$. If the last letter of w_1 is a, then x is in the language, but y is not. Therefore, all these words are L-dissimilar.

Counting the number of dissimilar words with respect to L, we get all words of length 1, 2, ..., $t-1$, and all words of length t, except 2^{t-2} of them. Therefore, our number is $1 + 2^2 + 2^3 + ... + 2^{t-1} + 2^t - 2^{t-2} = 2^{t+1} - 1 - 2^{t-2} = 2^t + 2^{n-t} - 2^{n-t-2} - 1 = 2^t + 2^{n-t-1} + 2^{n-t-2} - 1 = LB(n)$.

II. We will consider now the second case, when n is odd. We will prove that in this case we have $2^t - 1 + 2^{t-2}$ dissimilar words, meaning that we reach again $LB(n)$.

Let us now count the number of dissimilarities with respect to L. For any two words x, y with $|x| < |y| \leq t-1$, we can choose $z = b^{t-|y|}$ and we have that $|yz| = |y|+t-|y| = t$ and $|xz| < |yz| = t$. The word yz is in L'_n, but the word $xz \notin L'_n$ (has length less than t) and also $xz \notin L''_n$, because it ends in b and does not end in a. Therefore, all these words are not similar with respect to L. Now, let us take two distinct words x and y of equal length less than $t-1$. We may assume without any loss of generality that $x = x'ax''$, $y = y'by''$, and $x'' = y''$. Take $z = a^{t-2-|x''|}$. It follows that $|xz| = |x'| + 1 + |x''| + t - 2 - |x''| = t - 1 + |x'|$, so $xz \in L''_n \subseteq L$, but $yz \notin L$, since $yz \notin L''_n$, and if $|yz| = t$, $z \neq \varepsilon$, therefore the last letter of yz is a and $yz \notin L'_n$.

For words of length equal to $t-1$ we can apply the same proof for words ending in a and we get 2^{t-2} dissimilar words; for words ending in b we get only 2^{t-3} dissimilar words. The words ending in a and those ending in b are also dissimilar one with each other so, there are at least $2^{t-2} + 2^{t-3}$ words of length $t-1$ dissimilar one with each other.

Now let us analyze words of length t. Let $x \in \Sigma^t$ and $y \in \Sigma^*$, $|y| < t$. We distinguish the following cases:

1. $x = x'ax''$ and $y = y'by''$, $x'' = y''$. We take $z = a^{t-2-|x''|}$, so $t \leq |xz| \leq 2t-2$, $|yz| \leq 2t-2$,

$xz \in L$ but $yz \notin L$, since $yz \notin L''_n$ and $|yz| = t$ implies $z \neq \varepsilon$, so $yz \notin L'_n$.

2. $x = x'bx''$ and $y = y'ay''$, $x'' = y''$. We take $z = a^{t-2-|x''|}$, so $t \leq |xz| \leq 2t-2$, $|yz| \leq 2t-2$, $yz \in L$, but $xz \notin L$, since xz ends in a $xz \notin L'_n$, and $xz \notin L''_n$.

3. y is a suffix of x. If $|y| > 1$, we take $z = b^{t-|y|} \neq \varepsilon$, so $|xz| \leq 2t-2$, $|yz| \leq 2t-2$ and $yz \in L$, but $xz \notin L$. If $|y| \leq 1$, we take $z = a^{n-1-|x|} \neq \varepsilon$, so $xz \in L$, but $yz \notin L$, since $|yz| = |y| + n - 1 - |x| \leq t-1$.

For all cases we have that $x \not\sim_L y$, since the length of the longest word in L is $n-1 = 2t-2$.

We now analyze the similarity between words of the same length t. We take the following words $uaxa$ and $uaya$, $u \in \{a,b\}$, $x, y \in \Sigma^{t-3}$. Without any loss of generality we may assume $x = x'ax''$, $y = y'by''$, and $x'' = y''$. We take $z = a^{t-2-|x''|}$, and we get that $xz \in L$, but $yz \notin L$, using the same arguments as before. The same result we get for the words of the format $ubxa$ and $ubxb$, $u \in \{a,b\}$, $x \in \Sigma^{t-3}$.

If we take $uaxa$ and $ubya$, we can see that the first one is in the language, while the second one is not. The same thing happens if we take $uaxa$ and $ubyb$. If we take $ubxa$ and $ubyb$, let $z = a^{t-2}$. We have that $ubxaz \in L$, but $ubybz \notin L$; therefore, all words in these three categories are all L-dissimilar.

Hence, the number of L-dissimilar words can now be counted in the following way:
$\sum_{i=0}^{t-2} 2^i + 2^{t-2} + 2^{t-3} + 3 \cdot 2^{t-3} = 2^{t-1} - 1 + 2^{t-2} + 4 \cdot 2^{t-3} = 2^t - 1 + 2^{t-2} = LB(n)$, which completes the proof for the case when n is odd.

Since in both cases (n even; n odd) we succeeded to prove that there are at least $LB(n)$ dissimilar words in the given language, which actually implies that there are at least $LB(n)$ dissimilar states in the corresponding DFCA we have proved the theorem. □

Remark 3.1. The language considered in the previous theorem is the NFA constructed in [5], modified as follows: we add a transition from the state $t-1$ into $n-1$ with the letter b, and, we delete the transition from $n-2$ into $n-1$ with the letter b. The modification is required since the DFA obtained by subset construction from the NFA in the paper [5] has 2^{t-1} similarities and therefore, the DFCA has a much lower state complexity.

4. Upper bounds

In this section we give some necessary conditions for a NFA to obtain less than $UB(n,p)$ states when transformed to a DFCA. Therefore, if one "needs" an NFA which when transformed to a DFCA has to get higher state complexity than $UB(n,p)$, then the NFA must not satisfy any of the conditions mentioned in this section.

Since we are interested in finding an upper bound, once we establish that automata having a certain property will not reach $UB(n,p)$ states when transformed to a minimal DFCA, we assume that all subsequent automata are not satisfying that particular property since our quest is to settle the discussion about transformations from NFA to DFCA (i.e., *What is the highest possible number of states in the DFCA when starting with an n states NFA?*).

Let $m = \max\{i \mid lev_{D_L,i} \neq \emptyset\}$. Most of the results given in the Lemma 4.1 can be found in Salomaa and Yu [5] in Lemma 1, Lemma 2, and Corollary 1, using a slightly different notation. We view these

properties as an important staring point of the discussion; even though the results are given for the DFA, the results apply also to the DFCA:

Lemma 4.1. We have the following:

1. $|lev_{D_L,i}| \leq p^i$, 2. $Lev_{N_L,i} \neq \emptyset$, for all $1 \leq i \leq l$, 3. if $Lev_{N_L}(q) = i$, $q \notin \bigcup_{j>i} Q_{D,j}$,
4. $|\bigcup_{j\geq i} Q_{D,j}| \leq n-i$, 5. $|Q_{D,i}| \leq n-i$, 6. $|lev_{D_L,i}| \leq \min(p^i, 2^{n-i})$.

Proof:

1. We have that: $lev_{D_L,0} = \{0\}$ and $|lev_{D_L,i+1}| \leq p \cdot |lev_{D_L,i}|$.
2. Assume that there is j, $j \leq l$ such that $Lev_{N_L,j} = \emptyset$. Then $l < j$, contradiction.
3. If $q \in S$, $S \in lev_{D_L,j}$ and $j > i$, if follows that there is $w \in \Sigma^*$, such that $q \in \delta_N(0, w)$, i.e., $Lev_{N_L}(q) \geq j > i$, contradiction.
4. This follows from the fact that for each i, $1 \leq i \leq m$, there is at least one state $q \in Q_{D,i}$ for which $q \notin \bigcup_{j>i} Q_{D,j}$.
5. Follows from 4.
6. Follows from 5 and 1.

□

Lemma 4.2. If the maximum level in the DFA is less than t, (i.e., $m < t$,) then we have $|Q_D| < UB(n,p)$.

Proof:
If $m < t$ (m the maximum level in DFA), then we have that $m \leq t-1$, and using the Lemma 4.1 we obtain: $|Q_D| \leq \sum_{i=0}^{m} |lev_{D_L,i}| \leq \sum_{i=0}^{m} p^i \leq \sum_{i=0}^{t-1} p^i < UB(n,p)$, which proves the statement. □

Since for $m < t$ the number of states in Q_D is less than $UB(n,p)$, in what follows we consider only the case $t \leq m$.

For the states of level less than t we analyze what happens if two of them have the same maximum level in the NFA.

Lemma 4.3. Assume there is $1 \leq i \leq t-1$ and there are two states $s, q \in \bigcup_{j\geq i} Q_{D,j}$ such that $s, q \notin \bigcup_{j>i} Q_{D,j}$. In these conditions we have that $|Q_D| < UB(n,p)$.

Proof:
Let $s,\ q$ satisfying the properties mentioned in the lemma. Then for all $j > i \mid \bigcup_{k \geq j} Q_{D,k} \mid \leq 2^{n-(j+1)}$ using the same reasoning as in Lemma 4.1 property 4, but now we have two states that appear up to level i. Using this and the fact that the number of subsets of a set with $n-(j-1)-2$ elements is at most $2^{n-(j+1)}$ we get the inequality.

The next step is to approximate the number of states in Q_D by considering the maximal possible number of states on level 0, on level 1, on level 2, and so on up to level j and, then the rest of states that can be found on a level greater than j using the previous inequality. It is easy to notice that this particular value is maximal when $j = t$.

Hence, we have p^k possible states on each of the levels $0 \leq k \leq t-1$ plus the number of states that are of level t of higher using the result obtained above.

$$\begin{aligned}
|Q_D| &\leq 1 + p + \ldots p^{i-1} + p^i + \ldots + p^{t-1} + 2^{n-t-1} \\
&= 1 + p + \ldots p^{i-1} + p^i + \ldots + p^{t-1} + 2^{n-t} - 2^{n-t-1} \\
&= 1 + p + \ldots p^{i-1} + p^i + \ldots + p^{t-1} + 2^{n-t} - 2^{n-t-1} + 2^{n-t-2} - 2^{n-t-2} \\
&= 1 + p + \ldots p^{i-1} + p^i + \ldots + p^{t-1} + 2^{n-t} - 2^{n-t-2} + 2^{n-t-2} - 2^{n-t-1} \\
&= UB(n,p) + 2^{n-t-2} - 2^{n-t-1} < UB(n,p).
\end{aligned}$$

□

Therefore, at each level i, $0 \leq i \leq t-1$, in DFA there is one state and only one i present in all states from that level $S \in lev_{D_L,i}$, and is only in these states. Without any loss of generality, we may assume that the name of the state on level i is exactly i, i.e., $i+1 \in \delta_N(i,a)$, for all $a \in \Sigma$, when $0 \leq i \leq t-1$. Also, for states greater than t we may assume that they are topologically ordered, i.e., $\delta(i,a) = j$ implies $i < j$, for all $t \leq i, j \leq n-1$.

As a consequence, for any state $S \in Q_D$, $S \subseteq \{t, t+1, \ldots, f\}$, $S \neq \delta_D(R,a)$, for any $R \in Q_D$ and $R \cap \{0, \ldots, t-2\} \neq \emptyset$.

Lemma 4.4. If there exists i, $0 \leq i \leq t-2$, $\delta_N(i,a) = \delta_N(i,b)$, $|Q_C| \leq UB(n,p)$.

Proof:
Since $t = \lceil \frac{n}{1+\log_2 p} \rceil$, $n \leq 2t$, so $n-t-2 \leq t-2$. Now, assume there is i, $0 \leq i \leq t-2$, such that $\delta_N(i,a) = \delta_N(i,b)$. Then

$$\begin{aligned}
|Q_D| &\leq 1 + p + p^2 + \ldots + p^i + p^i + p^{i+1} + \ldots + p^{t-2} + 2^{n-t} \\
&= 1 + p + p^2 + \ldots + p^i + p^i + p^{i+1} + \ldots + p^{t-2} + p^{t-1} + 2^{n-t} - 2^{n-t-2} + 2^{n-t-2} - p^{t-1} \\
&= UB(n,p) + p^i + 2^{n-t-2} - p^{t-1} \ \leq \ UB(n,p) + p^{t-2} - p^{t-1} + 2^{n-t-2} \\
&= UB(n,p) - p^{t-2} + 2^{n-t-2} \ \leq \ UB(n,p) + 2^{n-t-2} - 2^{t-2} \\
&\leq UB(n,p)
\end{aligned}$$

□

One can easily observe from the proof of the previous lemma, that for $p > 2$, or $p = 2$ and $i < t-2$, or $p = 2$, $i = t-2$, and n odd, the inequality is strict.

The next lemma proves that if we have a final state $s \in Q_N$, $t \leq s < f$, we cannot reach the upper bound for Q_D (therefore, neither for Q_C).

Lemma 4.5. If $q \in F_N$, $q \neq f$, $q \geq t$, for any $S \subseteq \{t, t+1, \ldots, n-2\}$, with $(S \cup \{f, q\}), (S \cup \{q\}) \in Q_D$, then we have $(S \cup \{f, q\}) \equiv (S \cup \{q\})$.

Proof:
Recall that $f = n-1$. We have that $f \in \delta_D(S \cup \{f, q\}, \varepsilon) \in F_D$, $q \in \delta_D(S \cup \{q\}, \varepsilon) \in F_D$. So we cannot distinguish with ε.

If $w \in \Sigma^+$, $\delta_D(S \cup \{f, q\}, w) = \delta_D(S \cup \{q\}, w) \cup \delta_D(\{f\}, w) = \delta_D(S \cup \{q\}, w) \cup \emptyset = \delta_D(S \cup \{q\}, w)$. □

Corollary 4.1. If $q \in F_N$, $q \neq f$, $q \geq t$, $|Q_C| < UB(n, p)$.

Proof:
We have that the number of states in a cover automaton:

$$\begin{aligned} |Q_C| &\leq 1 + p + \ldots + p^{t-1} + \frac{2^{n-t}}{2} \\ &\leq 1 + p + \ldots + p^{t-1} + 2^{n-t} - 2^{n-t-1} \\ &< 1 + p + \ldots + p^{t-1} + 2^{n-t} - 2^{n-t-2} \\ &= UB(n, p). \end{aligned}$$

This happens because we lost all the equivalent sets of states from the level greater than or equal to t that contained q. □

The following lemma continues the discussion for the states that appear on a level greater than t.

Lemma 4.6. If for all $w \in \Sigma^*$ with the property that $\delta(t, w) = f$ we have that $|w| = n-1-t$, then $|Q_C| < UB(n, p)$.

Proof:
We prove the states $\{t\} \cup S$ and S are similar for every $S \subseteq \{t+1, \ldots, f\}$.

Indeed, if a state $S \subseteq \{t+1, \ldots, f\}$ is reachable in D_L its level is at least $t+1$, therefore we need to check the states $\{t\} \cup S$ and S with all words of length at most $n-1-(t+1) = n-t-2$. Since for such words w, $\delta_N(t, w) \cap F_N = \emptyset$, it follows that $\delta_D((\{t\} \cup S), w) \in F_D$ iff $\delta_D(S, w) \in F_D$, for all $w \in \Sigma^{\leq n-t-2}$, i.e., $(\{t\} \cup S) \sim_{D_L} S$. The number of such pairs of similar states is $2^{n-1-(t+1)+1} = 2^{n-t-1}$. Since, all reachable similar states in D_L are merged into one in the minimal DFCA C_L, the number of states in C_L is at most $\frac{p^t-1}{p-1} + 2^{n-t} - 2^{n-t-1} < UB(n, p)$ □

If the condition on the above lemma is not satisfied, there is a word $w \in \Sigma^*$ with $\delta(t, w) = f$ and $|w| < n-1-t$. Let s be the first state greater than $t-1$ for which $(s+1), q \in \delta(s, a),\ a \in \Sigma$ or $\{(s+1)\} = \delta(s, a)$, and there is another letter $b \in \Sigma$ such that $q \in \delta(s, b)$, and $f \in \delta(q, u)$ for some $|u| < n-t-2$. We can continue the discussion by considering these cases:

1. $(s+1), q \in \delta(s,a)$ and
2. $(s+1) \in \delta(s,a)$, $q \in \delta(s,b)$, and $f \in \delta(q,u)$ for some $a,b \in \Sigma$ and $|u| < n-t-2$.

In the first case $s+1$ cannot occur in any state $S \in Q_D$, $S \subseteq \{t,\ldots,n-1\}$ without q. Therefore, in this case $|Q_D| \leq \frac{p^t-1}{p-1} + 2^{n-t} - 2^{n-t-1+1-1} = \frac{p^t-1}{p-1} + 2^{n-t} - 2^{n-t-1} < UB(n,p)$.

In the second case, the problem is still open.

5. Conclusion

We have proved that for an NFA with n states accepting a finite language over a binary alphabet the equivalent minimal DFCA has at least $2^{\lceil \frac{n}{2} \rceil - 2}$ less states than the number of states of the minimal DFA.

Moreover, the number of languages for which this (associated DFCA) complexity is high, could be viewed as low when it is compared with the total number of NFAs of size n. This could prove very useful if one needs to make memory estimations according to the structure of an NFA given as input. In the section 4 we have given several results that provide more understanding of the structure of automatons that will yield the worst number of states when they are transformed into a DFCA. The discussion was given for a general alphabet of size p, one could consider the restriction to binary alphabets to obtain a better understanding of the structure of the NFA in that case. Of course, the discussion becomes more involved if one considers arbitrary alphabets.

References

[1] Cezar Câmpeanu and Andrei Păun, Counting The Number of Minimal DFCA Obtained by Merging States, *International Journal of Foundations of Computer Science*, Vol. **14**, No 6, December (2003), 995 – 1006.

[2] Cezar Câmpeanu, Nicolae Sântean, and Sheng Yu, Finite Languages and Cover Automata, *Theoretical Computer Science*, 267, 1-2 (2001), 3 – 16.

[3] Heiko Göeman, On Minimizing Cover Automata in $O(n \log n)$ Time, *Proc. of Seventh International Conference on Implementation and Application of Automata*, J.M. Champarnaud and D. Maurel eds., University of Tours, July 2002.

[4] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, Reading Mass., 1979.

[5] Kai Salomaa, Sheng Yu, NFA to DFA transformations for finite languages over arbitrary alphabets, *Journal of Automata, Languages and Combinatorics*, Vol. **2** (1997), 177 – 186.

[6] Nicolae Sântean, Towards a Minimal Representation for Finite Languages: Theory and Practice, Masters Thesis, The University of Western Ontario, January 2000.

[7] Sheng Yu, Regular Languages, in: A. SALOMAA AND G. ROZENBERG (eds.), *Handbook of Formal Languages*. Springer Verlag, Berlin, 1997, 41 – 110.