Complexity of $\overline{\sum\square\overline{A}}$ and its Connection with Logic

Manju, Rajesh Kumar

Abstract— *We investigate the connection of logic with complexity of basic operations. Upper and lower bounds for the finite-state complexity of arbitrary strings, and for strings of particular types, are given and incompressible strings are studied. We prove that the upper bounds on the state complexity of these operations, which were known to be tight for larger alphabets, are tight also for binary alphabets.*

Index Terms—*Finate Automat, Formal Languages, Logic, State Complexity.*

I. INTRODUCTION

This paper answers the following question asked by Jean-Eric Pin. Let be a finite alphabet and let A⊆ ∑* be a regular language, recognized by an NFA (non-deterministic finite automaton) or a DFA (deterministic finite automaton) with *n* states. How many states are sufficient (and necessary in the worst case) for an NFA, respectively a DFA, if it is to recognize $\overline{\sum\square\overline{A}} = \sum^* - \sum^*.\overline{A}$ (In general, ∑-A=Ā denotes the complement of a set A in ∑*, and A.B or AB denotes concatenation). The author shows an upper bound of 2^{n-1} states for a complete DFA recognizing $\overline{\sum\square\overline{A}}$, if A has an n-state DFA. It also show that this upper bound is optimal, even if NFAs are used to recognize $\overline{\sum\square\overline{A}}$. If A has an n-state NFA then $\overline{\sum\square\overline{A}}$ has an NFA with $\leq 2^{n-1}$ states, and this bound is close to optimal. n spite of its complicated appearance $\overline{\sum\square\overline{A}}$ has a rather simple description:

$\overline{\sum\square\overline{A}}$ = {w ∊ ∑*| every suffix of w belongs to A} . (Recall that the empty word and w itself are also suffixes of w.) Note that this expression implies that $\overline{\sum\square\overline{A}} = \phi$ A does not contain the empty word.

II. CONNECTION WITH LOGIC

The motivation of Pin's question comes from the word model of Propositional Temporal Logic; for terminology and further references see [5]. Here the set of all models of a formula φ (over a fixed alphabet ∑) is a formal language A(φ) ⊆ ∑* , which has the non-trivial property of being regular and aperiodic. Some of the temporal operators used in this logic are ○ ("next") and ◊ ("eventually", or "at some moment in the future"); there are also the usual boolean operations ¯,∧ ,∨. A natural dual to the "eventually" operator is the "forever" (or, "always in the future") operator □, defined to be ¯◊¯ ("not eventually not"). If only ○ , ◊ (or □) and the boolean operations are used, one obtains the Restricted Propositional Temporal Logic (RPTL). One of the main results in [5] is that a language A⊆ ∑* is the set of models of a formula in RPTL if and only if the syntactic semigroup of A is "locally $\mathscr{L}$-tivial" (see [5] for the definition). Formulas and their models are related as follows (as is easy to check):

$A(\overline{\varphi}) = \overline{A(\varphi)}$, $A(\varphi \wedge \psi) = A(\varphi) \cap A(\psi)$,
$A(\varphi \wedge \psi) = A(\varphi) \cup A(\psi)$, $A(\bigcirc \varphi) = \sum.A(\varphi)$,
$A(\Diamond\varphi) = \sum^*.A(\varphi)$,

Thus $A(\square\varphi) = A(\overline{\Diamond\overline{\varphi}}) = \overline{\sum^*\,\overline{A(\varphi)}}$. In other words, in this paper we study the state-complexity of the "forever" operator.

For more information on NFAs and complete DFAs, see [6]; a DFA is "complete" if the next state is always defined. The author will also use AFAs (alternating finite automata), because of their obvious ties to Logic (see [4,3,8,9] for the definition of AFA.

III. THEOREM 1

Upper Bounds (a) if A⊆ ∑* is recognized by an AFA (or, in particular, by an NFA or a DFA) with n states, then $\sum\square\overline{A}$ is recognized by an AFA with ≤ n+1 states, and $(\sum\square\overline{A})^{rev}$ is recognized by a DFA with $\leq 2^{n+1}$ states. Hence $\sum\square\overline{A}$ is recognized by an NFA with $\leq 2^{n+1}$ +1 states.

(b) If L is recognized by a DFA (complete or not) with n states, then $\overline{\sum\square\overline{A}}$ is recognized by a complete DFA with ≤ 2^{n-1} states.

IV. THEOREM 2

Lower bounds (a) For every n ≥ 2 there exists a 3-letter alphabet ∑ and a language A(⊆ ∑*) which is recognized by a complete DFA with n states, but such that every NFA (hence every DFA) recognizing $\overline{\sum\square\overline{A}}$ has at least 2^{n-1} states.

Revised Version Manuscript Received on June 20, 2017.
Ms. Manju, Assistant Professor, Department of Computer Science & Applications, CRM Jat College, Hisar (Haryana), India, E-mail: duham.manju@gmail.com
Mr. Rajesh Kumar, Assistant Professor, Department of Computer Science & Applications, CRM Jat College, Hisar (Haryana), India, E-mail: rajtaya@kuk.ac.in

(b) For every $n \geq 2$ there exists a 2-letter alphabet $\sum$ and a language A($\subseteq \sum^*$) which is recognized by a complete DFA with n states, and which is expressible in RPTL (in fact, A is the complement of a finite language, so it can be expressed in RPTL without using ◊); however, every complete DFA recognizing $\overline{\sum\square A}$ (or $\sum^*.\bar{A}$) has at least 2^{n-1} states.

Theorem 2 implies that for complete DFAs the upper bound 2^{n-1} of Theorem I(b) is optimal; for NFAs, the upper bound in Theorem l(a) is almost optimal.

V.PROOF OF THEOREM 1

A. *Upper Bound 1{a]*

Suppose $A \subseteq \sum^*$ is recognized by an AFA A_1, with n states, and with initial boolean function f_1. Then $\bar{A}$ is also recognized by an AFA A_2, with n states and with initial boolean function f_2 (one only has to negate the initial boolean function: $f_2 = \dot{f_1}$). From this one obtains an AFA A_3,with n + 1 states, recognizing $A \subseteq \sum^*$ (one adds a new start state s and introduces the transitions s.a = {s} ∪ {start states of A_2}, for each $a \in \sum$; the new initial boolean function is $f_3 = s \vee f_2$). Finally, we obtain an AFA A_4 recognizing $\overline{\sum\square A}$ by negating the initial boolean function of A_3: $f_4 = \overline{s \vee f2}$; the number of states of A $_4$ is n + 1. Author obtain an NFA with $2^{n+1}+1$ states for $\overline{\sum\square A}$ by applying the following theorem of Kozen (see [7,4]) to the AFA A_4: If a language R is recognized by an AFA with m states, then R^{rev} (the reverse of R) is recognized by a complete DFA with 2^m states.

Thus $\overline{\sum\square A}^{rev}$ has a complete DFA with 2^{n+1} states. By reversing this DFA (i.e., reversing the direction of every arrow, and exchanging accept and start states) an NFA with 2^{n+1} +1 states is obtained which recognized the $\overline{\sum\square A}$. (An additional state had to be added to the NFA since the DFA could have had many accept states, which would yield an NFA with many start states; but we want an NFA to have only one start state; this is a classical construction.)

B. *Lover Bound 1(b)*

Let □ = (Q,$\sum$, . , q_0, F) be a DFA recognizing A with |Q| = n. Recall that $\overline{\sum\square A}$ = {w $\in \sum^*$| every suffix of w belongs to A}. Since $\overline{\sum\square A}=\varnothing$ if A does not contain the empty word, the claimed upper bound certainly holds in this case. Let us henceforth assume that $q_0 \in$ F. The following complete DFA, inspired from the subset construction (see [6]), recognizes $\overline{\sum\square A}$:

B=({P ∈ ℘ (Q)| $q_0 \in$ P}, $\sum$, ∘, {q_0},

{P ∈ ℘ (Q)| $q_0 \in$ P and $P \subseteq F$};

here ℘ (Q) denotes the power set of Q. The next-state function ∘ is defined as follows for $a \in \sum$:

P ∘ a={q_0} ∪ P . a ={q_0 } ∪ { p . a | p ∈ P }.

Proof that B recognizes $\overline{\sum\square A}$:

B accepts w = a_1, a_2 . . . . .a_m, if and only if{q_0} ∘ a_1a_2 . . . .a_m={q_0} ∪ {q_0 . a_k . . . .a_{m-1} a_m | k=1, ... m} $\subseteq F$ this holds if and only if for all k ∈ {1, . m}: q_0 . a_k . .a_{m-1} a_m ∈ F (already assumed q_0 ∈ F) ; this hold if and only if every suffix a_k . . a_{m-1} a_m of w (and the empty suffix as well, by assumption) belongs to A; this holds if and only if w ∈ $\overline{\sum\square A}$.

VI. PROOF OF THEOREM 2

A. *Upper Bound 2(a)*

For every $n \geq 1$, let n = (l, . . . , n), and let F_n, be the set of all total functions from **n** to **n**. For x ∈ n and f ∈F_n , we denote the image of x under f by (x)f; in this notation, functions compose from left to right, e.g., (x)(f_1 f_2 f_3= ((($x)f_1)f_2)f_3$.

we will pick F_n as our alphabet, and for $n \geq 2$ we consider the following language:

A_n= {w ∈ $(F_n)^*$ | (l) f $_1$ . . . $f_k \neq 2$,

where w=(f_1 , . . . f_k), $k \geq 0$}.(The empty word is also in A_n, when k = 0 in the above definition.)

Then A is recognized by the complete DFA □ = (**n,** F_n, . , 1, **n** - {2}}, where the next-state function " ." is defined by i .f= (i)f, for i ∈ **n** and f ∈ F_n,. So A_n has an n-state complete DFA.

The alphabet F_n has size n^n but we shall see later how one can modify the above example (without changing the main properties of the languages) so that the alphabet has size 3.

Fact 1. The minimum complete DFA **B** of $\overline{\sum\square A}$ has 2^{n-1} states.

Proof. We consider the complete DFA B that was constructed in the proof of Theorem l(b), and we show that B is minimum for this example. Thus the minimum complete DFA for $\overline{\sum\square A}$ has 2^{n-1} states. Here B=({P $\subseteq$ ***n*** | 1 ∈ P} , F_n, ∘ , {l}, {P $\subseteq$ ***n***| **1**∈ P and 2 ∉ P}), where the next-state function ∘ is given by P ∘ a=(l) ∪ {(i)a| i ∈ P) when a ∈ F_n, and P $\subseteq$ ***n*** . Let us prove minimality of **B**.

Claim 1 (Reachability from the start state (1)). For every P $\subseteq$ ***n*** with 1 ∈ P there exists $u_p \in (F_n)^*$ such that {1} ∘ u_p = P

B. *Proff of Claim 1*

Let P = {1, p_1, . . ., p_k) $\subseteq$ ***n*** with l <p_1< . . . <p_k. We let u_p= f_1f_2 . . .f_k∈ $(F_n)^*$ where f_i (for $1 \leq i \leq k$) is defined by:

(1)f_i = p_i, and (x) f_i = x for x ≠ 1. It is straightforward to check that {l}∘ f_1 = {1, p_1}, {1, p_1}∘ f_2 = { 1, p_2, p_1 }, { 1, p_2, p_1 }∘ f_3 = { 1, p_3, p_2, p_1 }, etc.,

and {1} ∘ u_p = P.

Fact 2. *Every NFA recognizing* $\overline{\sum\square An}$ *has* $\geq 2^{n-1}$ *states.*
The following lemma from [1,2] is a convenient tool for proving lower bounds on the number of states of NFAs.
(See [l] for a proof.)

Lemma. Let R $\subseteq \sum^*$ be a regular language, and let X be a finite set. Assume that with every x $\in$ X one can associate words u_x and $v_x \in \sum^*$ such that

(1) x$\in$ X) $u_x v_x \in$ R,

(2) x,y $\in$ X with $x \neq y$) $u_x v_y \in$ R or $u_y v_x \notin$ R

Then every NFA recognizing R has $\geq | X |$ *states.*

C. *Proof of Fact 2*

We apply the lemma. For X we take the set X={P $\subseteq$ ***n***| ***1*** $\in$ P}.Then $|X|=2^{n-1}$. *With* every P $\in$ X we associate two words u_p, $v_p \in (F_n)^*$ as follows: u_p is the word defined in the proof of Fact 1, Claim 1 (Reach ability from (1)); and v_p is the function in F_n defined as follows (for any q): $(q)u_p = 1$ if q $\in$ P, and $(q)v_p = 2$ if q $\notin$ P (so v_p is just a one-letter word.)

Then we have:

(1) $u_p v_p \in \overline{\sum\square An}$ Indeed, $\{l\}\circ u_p v_p = P\circ v_p$, by the proof of Claim 1. Moreover, $P \circ v_p = \{l\}$, so $u_p v_p$ is accepted by the DFA B of $\overline{\sum\square An}$

2) $u_p\ v_s$ or $u_s\ v_p \notin \overline{\sum\square An}$ if $P \neq S$: Indeed, if $P\text{-}S \neq \Phi$ then $\{1\} \circ u_p v_p = P\circ\ v_s = \{1,2\}$ (which is a non-accept state of B, as it contains 2), so $u_p v_s \in \overline{\sum\square An}$ .This proves fact

REFERENCES

1. J.C. Birget, Intersection and union of regular languages, and state-complexity, *Inform. Process. Lett.* ***43*** (1992) 185- 190.
2. J.C. Birget, Partial orders on words, minimal elements of regular languages, and state-complexity, *Theoret. Comput. Sci.* ***119*** (1993) 267-291.
3. J. Btzozowski and E. Leiss. On equations for regular languages, finite automata, and sequential networks, *Theoret. Comput. Sci.* ***10*** (1980) 19-35.
4. A. Chandra, D. Kozen and L. Stockmeyer, Alternation, J. ACM 28 (1981) 114-133.
5. J. Cohen, D. Penin and J.-E. Pin, On the expressive power of temporal logic, *J. Comput. System Sci.* ***46*** (1993) 271-294.
6. J. Hopcroft and J. Ullman, *Introduction to Automata, Languages and Computation* (Addison-Wesley, Reading, MA, 19791.
7. D. Kozen, On parallelism in Turing machines, in: *Proc. Ann. Symp. on Founaiuions of Computer Science* (1976) 89-97.
8. E. Leiss, Succinct representation of regular languages by boolean automata, *Theoret. Comput. Sci.* ***13*** (198 1) 323-330.
9. E. Leiss, Succinct representation of regular languages by boolean automata, Part II, *Theoret. Comput. Sci.* **38** (1985) 133-136.
10. A.R. Meyer and M.J. Fischer, Economy of description by automata, grammars, and formal systems, in: Proc. 12Th IEEE Ann. Symp. on Switching atul Automata Theory (1971) 188-191.

AUTHORS PROFILE

Miss Manju, obtained her B.Sc. degree(Computer Science) from Govt. college, Jind and Master's Degree (Master of Computer Applications) from Banasthali University, Rajasthan. She is UGC NET qualified. Currently, She is Research Scholar in Guru Jambheshwar University, Hisar, Haryana,India. Her research interest in Software Quality Metrics in Object Oriented environment.

Rajesh Kumar. obtained his B.Sc.Degree, Master's degree – (Master of Computer Applications from Kurukshetra University, Kurukshetra. He had qualified UGC-NET Exam two times. Currently, He isan Assistant Professor in the Department of Computer Science and Applications, C R M Jat College, Hisar, Haryana, India. His research interests are in Genetic Algorithm, Software Testing and Design of Algorithms.