iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** IX **Month of publication:** September 2014

DOI:

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Non Determinism of Finite Automata

Akshit Chauhan[1], Deepak Kumar[2], Deepak Chandel[3],

[1,2,3]CSE Department
DCE Gurgaon

Abstract: The basic finite automata model has been extended over the years with different acceptancemodes (nondeterminism, alternation), new or improved devices (two-way heads, Pebbles, nested pebbles) and with cooperation. None of these additions permits recognitionof non-regular languages. The purpose of this work is to investigate a new kindof automata which is inspired by an extension of 2DPDAs. Mogensen enhanced thesewith what he called a WORM (write once, read many) track and showed that Cook's Linear-time simulation result still holds. Here we trade the pushdown store for nondeterminismor a pebble and show that the languages of these new types of finite automataare still regular. The conjunction of alternation or of nondeterminism and a pebblepermits the recognition of non-regular languages. We have given examples of languages thatare easy to recognize and of operations that are easy to perform using these WORM tracks under nondeterminism. While somewhat similar to Henie machines, our modelsdo not require an explicit time bound on their computations.

Keywords: finite automata, deterministic automata, non-deterministic automata.

I. INTRODUCTION

Two way deterministic automata(2DPDAs) have played an important role in development of formal language theory. it is awell-known fact that the class of language recognizible by multihued(or single-head with polynomailpadding) 2DPDA is strongly equal to pin in the sense thta the polynimial exponent is closely related to the number of heads. By cooks results, a k-head 2DPDA can be simulated on a random-access machine with unit cost in time)(mk) where m is the lenght of the unit. this has inspired some interesting algorithms such as the Knuth-Morrsi-Pratt[12] algorithm or a liner time algorithm for recognizing " PALSTAR". The question of an adequate notion of recognizability of graph properties has recently attracted much attention, and many competing approaches have been developed.

The starting point in this research is the notion of (nondeterministic or deterministic) _niTEautomaton over words. In a _rst step towards more general inputs than words,_nite tree automata were introduced by Doner and Thatcher and Wright.It was shown that many characterizations of recognizable word languages, namelyin terms of regular expressions, recognizability in _nite algebras, and de_nability inMonadic second-order logic, are all naturally preserved when passing from words to trees. WORM tracks thus serve as an auxiliary storage device permitting, for instance, lexical tokenization in linear time [17], as opposed to the quadratic worst-case time ofcurrent lexical scanners. WORM tracks are also useful for recognizing some languagesmore easily, such as {uuRvvR | u, v 2 {a, b}_} (PALSQUARE). However it is still anopen question whether or not WORM-2DPDAs recognize more languages than 2DPDAs.It seems natural to investigate the simpler case of a standard finite automatonprovided with a WORM track. This gives a two-way deterministic automaton witha WORM track (a WORM-2DFA) which is easily shown to accept regular languagesonly. But if one introduces nondeterminism or a pebble (thus obtaining what we callWORM-2NFAs and P-WORM-2DFAs), the regularity of the recognized languages isno longer trivial, and is the main result of this article.

A DFA represents a finite state machine that recognizes a RE. For example, the following DFA:

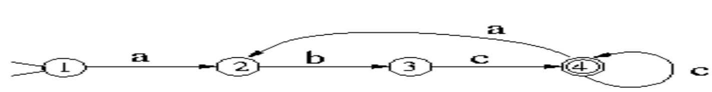


Fig .1 Graph of deterministic finite automata

Vol. 2 IssueIX, September 2014

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Recognizes $(abc^{+})^{+}$. A finite automaton consists of a finite set of states, a set of transitions (moves), one start state, and a set of final states (accepting states). In addition, a DFA has a unique transition for every state-character combination. For example, the previous figure has 4 states, state 1 is the start state, and state 4 is the only final state.

A DFA accepts a string if starting from the start state and moving from state to state, each time following the arrow that corresponds the current input character, it reaches a final state when the entire input string is consumed. Otherwise, it rejects thestring.

A **deterministic finite automaton** M is a 5-tuple, (Q, Σ, δ, q_0, F), consisting of

- a finite set of states (Q)
- a finite set of input symbols called the alphabet (Σ)
- a transition function ($\delta : Q \times \Sigma \rightarrow Q$)
- a startstate ($q_0 \in Q$)
- a set of acceptstates ($F \subseteq Q$)

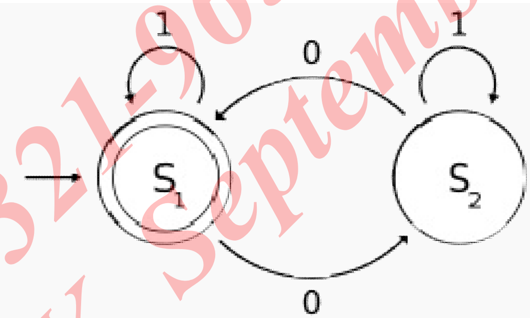
Let $w = a_1a_2 \ldots a_n$ be a string over the alphabet Σ. The automaton M accepts the string w if a sequence of states, $r_0, r_1, \ldots, r_n$, exists in Q with the following conditions:

1. $r_0 = q_0$
2. $r_{i+1} = \delta(r_i, a_{i+1})$, for $i = 0, \ldots, n-1$
3. $r_n \in F$.

In words, the first condition says that the machine starts in the start state q_0. The second condition says that given each character of string w, the machine will transition from state to state according to the transition function δ. The last condition says that the machine accepts w if the last input of w causes the machine to halt in one of the accepting states. Otherwise, it is said that the automaton *rejects* the string. The set of strings M accepts is the language *recognized* by M and this language is denoted by $L(M)$.A deterministic finite automaton without accept states and without a starting state is known as a transition system or semi automaton

EXAMPLE:

The following example is of a DFA M, with a binary alphabet, which requires that the input contains an even number of 0s.



The statediagram for M

M = (Q, Σ, δ, q_0, F) where

- $Q = \{S_1, S_2\}$,
- $\Sigma = \{0, 1\}$,
- $q_0 = S_1$,
- $F = \{S_1\}$, and
- δ is defined by the following statetransitiontable:

	0	**1**
S_1	S_2	S_1
S_2	S_1	S_2

The state S_1 represents that there has been an even number of 0s in the input so far, while S_2 signifies an odd number.

Vol. 2 IssueIX, September 2014

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

A 1 in the input does not change the state of the automaton. When the input ends, the state will show whether the input contained an even number of 0s or not. If the input did contain an even number of 0s, *M* will finish in stateS_1, an accepting state, so the input string will be accepted.

The language recognized by *M* is the regularlanguage given by the regularexpression 1*(0 (1*) 0 (1*))*, where "*" is the Kleenestar, e.g., 1* denotes any non-negative number (possibly zero) of symbols "1".

II. NON DETERMINISTIC FINITE AUTOMATA

A **nondeterministic finite automaton** (NFA), or nondeterministic finite state machine, is a finite state machine that ***(1)*** does not require input symbols for state transitions and is capable of transitioning to zero or two or more states for a given start state and input symbol. This distinguishes it from a deterministic finite automaton (DFA), in which all transitions are uniquely determined and in which an input symbol is required for all state transitions. Although NFA and DFA have distinct definitions, all NFAs can be translated to equivalent DFAs using the subset construction algorithm, i.e., constructed DFAs and their corresponding NFAs recognize the same formal language. Like DFAs, NFAs only recognize regular languages. NFAs were introduced in 1959 by Michael O. Rabin and Dana Scott, who also showed their equivalence to DFAs.

III. WORM-2NFAs

WORM-2NFAs are somewhat similar to nondeterministic Hennie machines. These aresingle-head Turing machines whose heads do not leave the input portion of their tape, and which have the bounded visit property, that is, there is a constant c such thatthe machine never visits any given position more than c times [2]. These machinesrecognize regular languages only. In Hennie's original paper [11] it was shown thatdeterministic linear-time Turing machines have the bounded visit property. It shouldbe noted that there exists linear-time nondeterministic Turing machines recognizing on-regular, NP-complete languages2 [13]. Furthermore, the linearity of running timeis a non-trivial and thus UNdecidable property of Turing machines, making the class

An NFA is represented formally by a 5-tuple, (Q, Σ, Δ, q_0, F), consisting of

- a finite set of states Q
- a finite set of inputsymbols Σ
- a transition function $Q \times \Sigma \rightarrow P(Q)$.
- an initial (or *start*) state $q_0 \in Q$
- a set of states F distinguished as accepting (or *final*) *states* $F \subseteq Q$.

Here, $P(Q)$ denotes the powerset of Q. Let $w = a_1a_2 \ldots a_n$ be a word over the alphabet Σ. The automaton *M* accepts the word *w* if a sequence of states, $r_0, r_1, \ldots, r_n$, exists in Q with the following conditions:

1. $r_0 = q_0$
2. $r_{i+1} \in \Delta(r_i, a_{i+1})$, for $i = 0, \ldots, n-1$
3. $r_n \in F$.

In words, the first condition says that the machine starts in the start state q_0. The second condition says that given each character of string *w*, the machine will transition from state to state according to the transition function Δ. The last condition says that the machine accepts *w* if the last input of *w* causes the machine to halt in one of the accepting states. Otherwise, it is said that the automaton rejects the string. The set of

Vol. 2 IssueIX, September 2014

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

strings *M* accepts is the language recognized by *M* and this language is denoted by *L (M).*

We can also define*L(M)* in terms of Δ^*: $Q \times \Sigma^* \rightarrow P(Q)$ such that:

1. $\Delta^*(r, \varepsilon) = \{r\}$ where ε is the empty string, and
2. If $x \in \Sigma^*$, $a \in \Sigma$, and $\Delta^*(r, x) = \{r_1, r_2, \ldots, r_k\}$ then $\Delta^*(r, xa) = \Delta(r_1, a) \cup \ldots \cup \Delta(r_k, a)$.

Now $L(M) = \{w \mid \Delta^*(q_0, w) \cap F \neq \emptyset\}$.

Note that there is a single initial *state*, which is not necessary. Sometimes, NFAs are defined with a set of initial states. There is an easy construction that translates a NFA with multiple initial states to a NFA with single initial state, which provides a convenient notation.

EXAMPLE:

Let *M* be a NFA, with a binary alphabet, that determines if the input ends with a 1.

In formal notation, let $M = (\{p, q\}, \{0, 1\}, \Delta, p, \{q\})$ where the transition function Δ can be defined by this state transition table.

	0	**1**
p	{*p*}	{*p*,*q*}
q	∅	∅

Note that $\Delta(p,1)$ has more than one state therefore *M* is nondeterministic. The language of *M* can be described by the regular language given by the regular expression (0|1)*1.

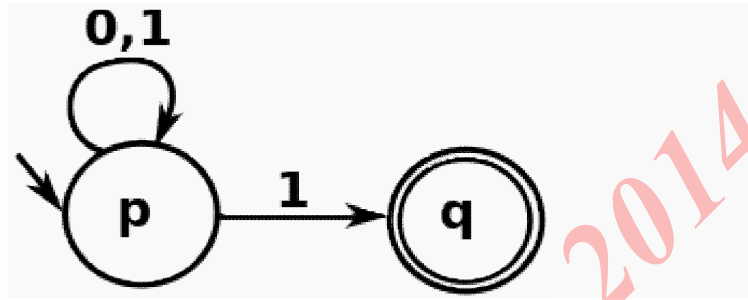


Fig. 3 the state diagram of M

IV. CONCLUSION

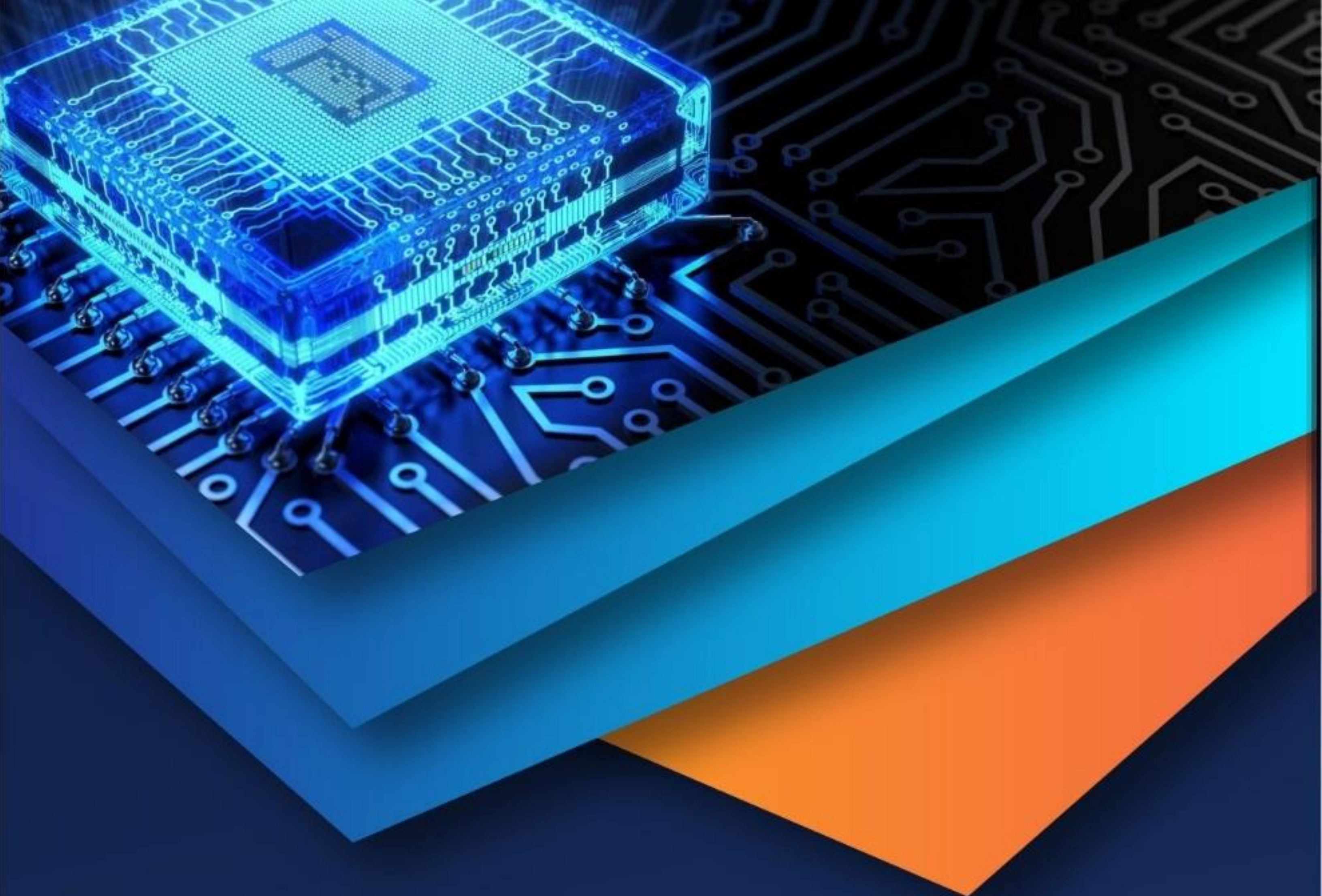
The ability of WORM-2NFAs to hold a number of guessed values linear in the size ofthe input throughout the computation appears as ability out of the reach of 2AFA,pebble-2AFA and other similar models. Also, compared to nondeterministic Henniemachines, WORM-2NFAs do not have a finite bound on the number of times they canvisit a given square. We therefore conjecture that 2AFAs as well as nondeterministicHennie machines cannot solve SAT with a polynomial number of states. We hopethat these results will shed some light on the relative power of WORM-2DPDAs vS2DPDAs.The effect of WORM cells on the languages of other kinds of finite computingDevices, such as 2DFAs with nested pebbles, 2DFAs or 2NFAs with monotonic outputtapes and tree-walking automata remains to Be Explored.

REFERENCES

- M. O. Rabin and D. Scott, "Finite Automata and their Decision Problems", *IBM Journal of Research and Development*, **3**:2 (1959) pp. 115–125.
- Michael Sipser, *Introduction to the Theory of Computation*. PWS, Boston. 1997. ISBN 0-534-94728-X. *(see section 1.2: Nondeterminism, pp.47–63.)*
- John E. Hopcroft and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-

Wesley Publishing, Reading Massachusetts, 1979. ISBN 0-201-02988-X.

- C. Choffrut, B. Durak, Collage of two-dimensional words. Theoret. Comp.Sci 340 (2005) 1, 364–380.
- J.-C. Birget, Two-way automata and length-preserving homomorphisms. Mathematical Systems Theory 29 (1996) 3, 191–226.



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY