==
Language in India www.languageinindia.com ISSN 1930-2940 Vol. 13:5 May 2013
==

SDVFA Languages

A. Jain jainarihant@live.com and S. Sinha smitas6@yahoo.com

SDVFA Languages

A. Jain and S. Sinha

Abstract. The notion of SDVFA of order (s,t) has already been introduced by the first author in [8]. In this paper, we discuss the languages accepted and not accepted by an SDVFA of order (s,t).

AMS Subject Classification (2000): 68Q45

Keywords: Semi-deterministic virtual finite automaton(SDVFA), Language processing

1. Introduction

The notion of SDVFA of order (s,t) has already been introduced by the first author in [8]. In this paper, we study the languages accepted and not accepted by an SDVFA of order (s,t). We begin with the definition of SDVFA of order (s,t) [8].

Definition 1.1. A semi-deterministic virtual finite automaton (SDVFA) of order (s,t) is a finite automaton that can make atmost "s"$(s \geq 1)$ transitions on receiving a real input and atmost "t"$(t \geq 0)$ transitions on virtual input (or no input). (Zero transition means the automaton remains in the same state).

Remark 1.1. For an SDVFA having n states, we have the following:

(i) If $s = 1$ and $t = 0$, then an SDVFA of order $(1,0)$ is simply a DFA [1, 2, 6].

(ii) If $s = 1$ and $t = n$, then an SDVFA of order $(1,n)$ is simply a VDFA [7].

(iii) If $s = n$ and $t = 0$, then an SDVFA of order $(n,0)$ is simply an NFA [1, 2, 6].

(iv) If $s = n$ and $t = n$, then an SDVFA of order (n,n) is simply an ϵ-NFA [1, 2, 6].

We formally define a semi-deterministic virtual finite automaton (SDVFA) of order (s,t) as follows:

Definition 1.2. A semi-deterministic virtual finite automaton (SDVFA) of order (s, t) consists of

1. A finite set of states (including the dead state) often denoted by Q.
2. A finite set of input symbols including the empty string symbol ϵ. This is often denoted by $\Sigma \cup \{\epsilon\}$. Σ is called real alphabet.
3. A transition function $\delta_{(s,t)}$ that takes as arguments a state and an input symbol. On real input symbol i.e. if the symbol is a member of real alphabet Σ, $\delta_{(s,t)}$ returns a set of atmost "s" states while on virtual input ϵ, the transition function returns a set of atmost "t" states.
4. A start state S which is one of the states in Q.
5. A set of final or accepting states F. The set F is a subset of Q. Dead state is never an accepting state and it makes a transition to itself on every possible input symbol.

We can also denote an SDVFA of order (s, t) by a "five tuple" notation:

$$V = (Q, \Sigma \cup \{\epsilon\}, \delta_{(s,t)}, q_0, F)$$

where V is the name of the SDVFA, Q is the set of states, $\Sigma \cup \{\epsilon\}$ is the set of input symbols, $\delta_{(s,t)}$ is the transition function, q_0 is the start state and F is the set of accepting states.

2. SDVFA Languages

An SDVFA of order (s, t) can be used as devices to recognize (accept) sentences in a language. Let $O = \{0, 1\}$ be the output alphabet of an SDVFA of order (s, t). A state is said to be an **accepting state** if its output is 1. A state is said to **rejecting state** if its output is 0. Consequently, an input sequence is said to the be **accepted** by the SDVFA of order (s, t) if it leads the machine from the initial state to an accepting state. On the other hand, an input sequence is said to be **rejected** by the SDVFA of order (s, t) if it leads the machine from the initial state to a rejecting state.

Example 2.1. Fig. 2.1 shows an SDVFA of order $(1,0)$ that accepts all binary sequences that end with the digits 001. When an SDVFA is used as an acceptor, the states of the SDVFA are divided into only two classes, viz., accepting and rejecting states. Therefore, we introduce the slightly simpler notation of circling the names of the accepting states instead of writing down the output of each state as in Fig.2.1 where accepting state D is represented by a double circle.

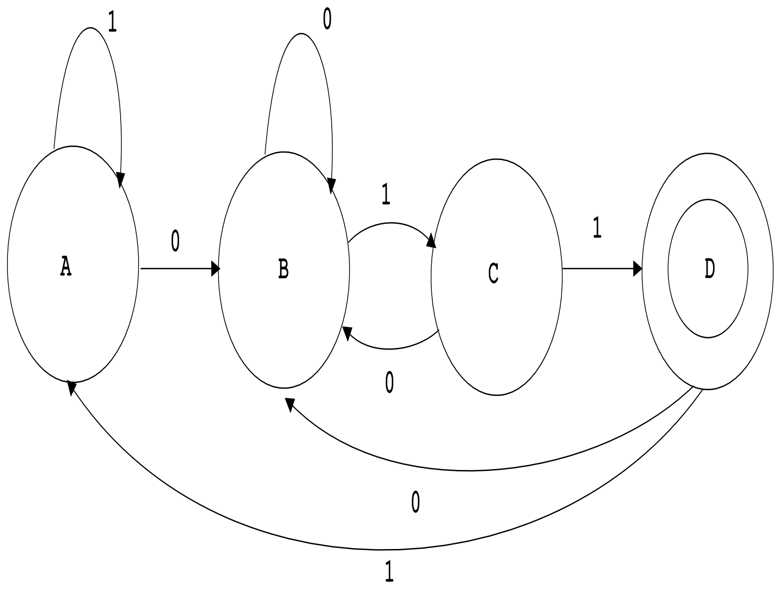

Fig. 2.1

A language is said to be an **SDVFA** if there is an SDVFA of order (s,t) that accepts exactly all sentences in the language. Thus, according to above example, the language consisting of all binary sequences that end with 011 is an SDVFA language. Clearly, any given SDVFA of order (s,t) defines an SDVFA language. On the other hand, a given language might or might not be an SDVFA. We see this fact with the help of following examples.

Example 2.2. Consider the language $L = \{a^k b^k | k \geq 1\}$. This language is not an SDVFA language.

To prove this, let us assume the contrary i.e. there exists an SDVFA of order (s,t) that accepts the sentence in L. Suppose this machine has N states. Clearly, the machines accepts the sentence $a^N b^N$. Starting from the initial state, the machine will visit N states after receiving the N $a's$ in the input sequence as shown in Figure 2.2(a), where s_{j_0} is the initial state and $s_{j_1}, s_{j_2}, \cdots, s_{j_N}$ are the states, the machine is in after receiving the sequence a^N. Also, $s_{j_{2N}}$ is the state, the machine is in after receiving the sequence $a^N b^N$. Clearly, $s_{j_{2N}}$ is an accepting state. According to the "Pigeonhole" principle, among $N+1$ states $s_{j_0}, s_{j_1}, s_{j_2}, \cdots, s_{j_N}$, there are two of them that are the same. Suppose the machine visits state s_k twice as shown in Fig 2.2(b) and there are x a's between the first and the second visit to state s_k, then the sequence $a^{N-x}b^N$ which is not a sentence in the language will also be accepted by the finite state machine. Consequently, we can conclude that the language L is not an SDVFA language.

$$\overbrace{a\ a\ a\ a\ a\ a\ a\ \cdots a\ a\ a\ a}^{N}\ \overbrace{b\ b\ b\ b\ b\ b\ b\ b\ \cdots b\ b\ b\ b}^{N}$$

$$s_{j_0}\ s_{j_1} s_{j_2} \cdots\cdots\ \cdots \qquad \cdots\ s_{j_N} \cdots\cdots\ \cdots\cdots \qquad \cdots\ s_{j_{2N}}$$

Fig. 2.2(a)

$$\overbrace{a\ a\ a\ a\ a\ \underbrace{a\ a\cdots a}_{x}\ a\ a\ a}^{N}\ \overbrace{b\ b\ b\ b\ b\ b\ b\ b\ \cdots b\ b\ b\ b}^{N}$$

$$s_{j_0}\ s_{j_1} \cdots\ \ s_k \cdots\cdots \quad s_k \cdots s_{j_N} \cdots\cdots\ \cdots \qquad \cdots\ s_{j_{2N}}$$

Fig. 2.2(b)

Example 2.3. Consider the language $L = \{a^k | k = i^2, i \geq 1\}$. This language is not an SDVFA language.

To prove this, let us assume that there is an SDVFA of order (s,t) that accepts language L. Let N denote the number of states in the SDVFA. Let i be an integer that is sufficiently large such that

$$(i+1)^2 - i^2 > N.$$

Consider the situation depicted in Figure 2.3. Since between the $i^2 th$ a and the $(i+1)^2 th$ a , the SDVFA will visit a certain state s_k more than once, removal of the a's between these two visits will yield a sequence that will also be accepted by the SDVFA. However, this sequence is not a sentence in the language because it contains more than i^2 but less than $(i+1)^2$ a's. Thus, we conclude that L is not an SDVFA language.

$$\overbrace{\overbrace{a\;a\;a\;a\;\cdots\;a}^{i^2}\;a\;a\;a\;a\;a\;\cdots\;a\;a\;a\cdots\;a\;a}^{(i+1)^2}$$
$$s_0\;\cdots\qquad\qquad\cdots\qquad\qquad s_k\cdots\quad s_k\quad\cdots\;s_f$$

Fig. 2.3

Now, we present a theorem which gives the criteria for testing the SDVFA languages.

Theorem 2.1. Let L be an SDVFA language accepted by an SDVFA of order (s,t) with N states. For any sequence α whose length is N or larger in the language, α can be written as uvw such that v is nonempty and uv^iw is also in the language for $i \geq 0$, where v^i denotes the concatenation of i copies of the sequence v. (In other words, $uw, uvw, uvvw, uvvvw, \cdots$ are all in the language).

Proof. Without any loss of generality, let the length of α be N. Let $\alpha = a_1a_2a_3\cdots a_N$. Let $s_{j_0}, s_{j_1}, s_{j_2}, \cdots, s_{j_N}$ denote the states of the SDVFA where s_{j_0} is the initial state and s_{j_N} is an accepting state. Again, among the $N+1$ states $s_{j_0}, s_{j_1}, s_{j_2}, \cdots, s_{j_N}$ there are two of them that are the same. Suppose that is state s_k, as shown in Fig. 2.4. If we divide α into three segments as shown in Fig. 2.4, we realize that the sequences $uw, uvw, uvvw, uvvvw, \cdots, uv^iw, \cdots$ will all lead the SDVFA from the initial state s_{j_0} to the accepting state s_{j_N}.

$$\overbrace{a_1\;a_2\;a_3\;\cdots\;a_p}^{u}\;\overbrace{\cdots\;a_q}^{v}\overbrace{\cdots\;a_N}^{w}$$
$$s_{j_0}\;s_{j_1}\;s_{j_2}\;s_{j_3}\cdots\;s_k\cdots\qquad s_k\quad\cdots s_{j_N}$$

Fig. 2.4

4. Conclusion

In this paper, we have discussed the languages accepted and not accepted by an SDVFa of order (s, t)

References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[2] A.V. Aho and J.D. Ullman, Foundations of Computer Science, Computer Science Press, New York, 1994.

[3] J. Anderson, Automata Theory with Modern Applications, Cambridge University Press, 2006.

[4] N. Chomsky, in review of Belevitch, V., *Language des machines et langage humain*, Language, 34 (1958), p.100.

[5] N. Chomsky, *Three models for the description of languages*, IRE Trans. on Information Theory, 2:3 (1956), pp.113-124.

[6] S. Eilenberg, *Automata, Languages and Machines*, Vol. A-B, Academic Press, New York, 1974.

[7] M. Ganesan, P. Bhattacharya and A. Jain, *A Notion of Virtual Deterministic Finite Automaton (VDFA) in Language Processing*, J. Comput. Math. Optim., 2 (2006), 181-206.

[8] A. Jain, *Semi-deterministic Virtual Finite Automaton(SDVFA) of order* (s, t), J. Compt. Math. Optim., 5(2009), 1-22.

[9] A. Jain, *Equivalence of Semi-deterministic Virtual Finite Automaton(SDVFA) with DFA, VDFA, NFA and* ϵ*-NFA*, to appear in Ars Combinatoria.

[10] A. Jain, *Semi-deterministic Virtual Finite Automaton(SDVFA) of order* (s, t) *and Regular Grammar*, to appear in Ars combinatoria.

[11] M.O. Rabin and D. Scott, *Finite automata and their decision problems*, IBM J. Research and Development 3:2 (1959), pp.115-125.

[12] M.P. Schutzenberger, *On the definition of a family of automata*, Information and Control, 4 (1961).

[13] C.E. Shannon and J. McCarthy, Automata Studies, Princeton Univ. Press, 1956.

Department of Linguistics
Berhampur University
Odisha 760007
India
E-mail: jainarihant@live.com

Department of Linguistics
Berhampur University
Odisha 760007
India
E-mail: smitas6@yahoo.com