Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 321 (2016) 23–39

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

On the Real-state Processing of Regular Operations and The Sakoda-Sipser Problem

J. Andres Montoya[1,2] David Casas[3]

Departamento de Matemáticas
Universidad Nacional de Colombia
Bogotá, Colombia

Abstract

In this work we study some aspects of state-complexity related to the very famous Sakoda-Sipser problem. We study the state-complexity of the regular operations, we survey the known facts and, by the way, we find some new and simpler proofs of some well known results. The analysis of the state of art allowed us to find a new and meaningful notion: Real-state processing. We investigate this notion, looking for a model of deterministic finite automata holding such an interesting property. We establish some preliminary results, which seem to indicate that there does not exists a model of deterministic finite automata having real-state processing of regular expressions, but, on the other hand, we are able of exhibiting a deterministic model of finite automata having real-state processing of star free regular expressions.

Keywords: Finite automata, State complexity, Regular expressions, Regular operation.

1 Introduction

It is known that nondeterministic finite state automata (1NFAs) are as powerful as deterministic finite state automata (1DFAs), in the sense that 1NFAs can only recognize regular languages. It is also known that 1NFAs are more powerful than 1DFAs, because 1NFAs cannot be simulated by 1DFAs with a polynomial overhead in the number of states. Sakoda and Sipser [8] asked if 1NFAs can be simulated by two-way deterministic finite state automata (2DFAs) with a polynomial overhead in the number of states. It is one of the questions included in the, so called, *Sakoda-Sipser Problem.* The Sakoda-Sipser question is a question about: how, when and to which extent can two-wayness replace nondeterminism? It would be great news if such a question would have an affirmative answer. It is the case, given

[1] The first author would like to thank Universidad Nacional de Colombia and the support provided through the R.P. Hermes 16860.

[2] Email: `jamontoyaa@unal.edu.co`

[3] Email: `dfcasast@unal.edu.co`

http://dx.doi.org/10.1016/j.entcs.2016.02.003
1571-0661/

that 1NFAs (and 2NFAs) are unreliable automata which cannot be used in practice. But, in despite of their *purely theoretical value*, 1NFAs have some remarkable features, which we would like to have in some model of reliable and implementable finite state automata. Thus, we think that the Sakoda-Sipser question is a special case of the following more general question: how, when and to which extent can deterministic finite state automata with added abilities be as powerful and efficient as their nondeterministic counterparts?

Before attacking the later question we will have to consider the following one: which are those remarkable features of nondeterministic finite automata? We won't provide an exhaustive list of remarkable characteristics, but we would like to point out, and to discuss, below, one of those features which has captured our attention.

If one is asked to prove that the set of regular languages is closed under the regular operations, it is very easy to figure out such a proof, if one is allowed to use 1NFAs. Things become harder if one is obligated to employ, in the proof, the weaker model of 1DFAs. Moreover, such an easy proof using 1NFAs yields a linear time algorithm, called Thompson's algorithm, which, on input α (where α is a regular expression), computes an $O(|\alpha|)$-state 1NFA recognizing the language $L(\alpha)$.

We say that a model of deterministic finite automata has *Thompson property*, if and only if, there exists a polynomial time algorithm, which, on input α, computes an $O(|\alpha|^c)$-state automaton within the model, and which recognizes the language $L(\alpha)$, (where c is some fixed constant).

Thus, we have that Thompson property holds for 1NFA, while it is very easy to prove it does not hold for 1DFA. We consider that Thompson property is a remarkable feature of 1NFAs, given that it allows those automata to efficiently process regular expressions. Take into account that the processing of regular expressions is (one of) the main task(s) assigned to finite automata. Unfortunately, the nondeterministic nature of 1NFAs makes them become a nonimplementable solution to the aforementioned problem. Thus, it would be great news if we could exhibit a deterministic model of automata for which Thompson property holds.

In this work we investigate the following question: how, when, and to which extent is it possible to define a model of deterministic finite automata for which Thompson property holds?

Remark 1.1 We understand the Sakoda-Sipser problem as the question: does there exist a deterministic model of finite state automata which can efficiently simulate nondeterministic automata? It is clear that a positive answer to Sakoda-Sipser implies that our problem can be positively solved. On the other hand, if we could give a positive answer to our question, we could not immediately conclude that Sakoda-Sipser also has a positive answer. It is the case because nondeterministic automata are exponentially more succinct than regular expressions [4].

Remark 1.2 We assume that the reader knows the definition of the basic models of finite state automata such as DFAs, NFAs, 2DFAs an so on. The interested reader can consult the excellent reference [9].

Organization of the work and contributions. This work is organized into