Journal of Automata, Languages and Combinatorics **9** (2004) 1, 121–146

BIMACHINES AND STRUCTURALLY-REVERSED AUTOMATA

NICOLAE SANTEAN
Department of Computer Science, The University of Western Ontario
1151 Richmond Street, Suite 2, London, Ontario N6A 5B8, Canada
e-mail: nic@csd.uwo.ca

ABSTRACT

Although bimachines are not widely used in practice, they represent a central concept in the study of rational functions. Indeed, they are finite state machines specifically designed to implement rational word functions. Their modelling power is equal to that of single-valued finite transducers. From the theoretical point of view, bimachines reflect the decomposition of a rational function into a left and a right sequential function. In this paper we define three new types of bimachines, classified according to the scanning direction of their reading heads. Then we prove that these types of bimachines are equivalent to the classical one and for doing so, we define and use a new concept, of structurally-reversed automaton. Consequently, we prove that the scanning directions of bimachines are irrelevant from the point of view of their modelling power. This leads to a method of simulating a bimachine by a left sequential transducer (or generalized sequential machines – *GSM* for short). Indeed, a preprocessing of the input word allows sequential transducers to realize the full range of rational functions. Remarkably enough, we basically show that the so versatile functional transducers – nondeterministic and with λ-input transitions – can successfully be replaced by a simple deterministic setup: a "trimmer" coupled with a *GSM*. Intuitively, this fact proves that sequential functions are not much weaker than rational functions.

Keywords: Structurally-reversed automata, rational functions, bimachines, *GSM*

1. Introduction

The interest in providing finite and effective descriptions of sets in certain algebraic structures dates as early as 40's. Formal machines, initially designed as nets of formalized neurons (McCulloch-Pitts nets, comprising of synchronized elements, each capable of some boolean function) were introduced by McCulloch and Pitts in 1943 [14] in order to carry out the control operations of a Turing Machine [25]. The idea was further refined by Kleene in 1956 [11], who interrelated regular sets (or regular events in nerve nets), regular expressions and finite automata.

In parallel, a special interest in formal and natural language processing was developing. Indeed, in addition to classification ([3, 4] – Chomsky, 1956–59), recognition and generation of languages, a growing interest for the study of devices with output emerged. In [16] (Mealy, 1955), [18] (Moore, 1956) and [22] (Raney, 1958) we find