Theoretical Computer Science 536 (2014) 1–20

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A description based on languages of the final non-deterministic automaton

A. Ballester-Bolinches [a], E. Cosme-Llópez [a], R. Esteban-Romero [a,b,*]

[a] Departament d'Àlgebra, Universitat de València, Dr. Moliner, 50, E-46100 Burjassot (València), Spain
[b] Institut Universitari de Matemàtica Pura i Aplicada, Universitat Politècnica de València, Camí de Vera, s/n, E-46022 València, Spain

ARTICLE INFO

Article history:
Received 20 December 2012
Received in revised form 12 December 2013
Accepted 14 January 2014
Communicated by R. van Glabbeek

Keywords:
Non-deterministic automaton
Formal language
Coalgebra
Bisimilarity
Final automaton

ABSTRACT

The study of the behaviour of non-deterministic automata has traditionally focused on the languages which can be associated to the different states. Under this interpretation, the different branches that can be taken at every step are ignored. However, we can also take into account the different decisions which can be made at every state, that is, the branches that can be taken, and these decisions might change the possible future behaviour. In this case, the behaviour of the automata can be described with the help of the concept of bisimilarity. This is the kind of description that is usually obtained when the automata are regarded as labelled transition systems or coalgebras.

Contrarily to what happens with deterministic automata, it is not possible to describe the behaviour up to bisimilarity of states of a non-deterministic automaton by considering just the languages associated to them. In this paper we present a description of a final object for the category of non-deterministic automata, regarded as labelled transition systems, with the help of some structures defined in terms of languages. As a consequence, we obtain a characterisation of bisimilarity of states of automata in terms of languages and a method to minimise non-deterministic automata with respect to bisimilarity of states. This confirms that languages can be considered as the natural objects to describe the behaviour of automata.

1. Introduction

The aim of this paper is to present a description of the final object of the category of non-deterministic automata, regarded as labelled transition systems, by means of languages. Our description emphasises the role of languages as natural objects to describe the behaviour of automata.

In this paper we will use the terminology of category theory. We will assume the reader to be familiar with the basic concepts of category theory, as categories, functors, and final or terminal objects. The reader is referred to [20] for more information about category theory.

We can assign to every state of an automaton an associated language, consisting of all words which send this state to a final or terminal state. Traditionally, many authors have considered as the behaviour of a state of an automaton simply its associated language. Under this point of view, the different decisions that may be taken from each state are ignored. However, we can take into account the different branches or decisions that may be taken at every state. They might change the future behaviour of the automaton. From this point of view, automata are regarded as labelled transition systems or

* Corresponding author.
E-mail addresses: Adolfo.Ballester@uv.es (A. Ballester-Bolinches), Enric.Cosme@uv.es (E. Cosme-Llópez), Ramon.Esteban@uv.es, resteban@mat.upv.es (R. Esteban-Romero).

0304-3975/$ – see front matter
http://dx.doi.org/10.1016/j.tcs.2014.01.018

coalgebras for suitable endofunctors on the category **Set**. In this scope, the idea of the behaviour of the states of the coalgebra is related to the notion of bisimilarity, a concept originated in the field of concurrency (its precise definition will be given in Section 2, see Definition 2.13). We can say that two states have the same behaviour when they are bisimilar. Under very general hypotheses, which hold for automata, when a category of coalgebras possesses a final object, two states are bisimilar if and only if both states have the same image by the unique homomorphism into the final object. This motivates the interest in studying the final objects in some categories of coalgebras, like automata.

Up to now, most known descriptions of final coalgebras are of a very general theoretical nature or are given as a quotient of a coalgebra by the bisimilarity relation. We will present some of them in Section 3. When they are applied to the functor $\mathcal{N} = 2 \times \mathcal{P}_\omega(\mathrm{Id})^A$ associated to non-deterministic automata, it seems that they do not give a clear idea of the role of languages, which are incontestably a central notion in this theory, in the final automaton. Hence the question of whether languages can be used to describe the behaviour of non-deterministic automata as labelled transition systems remains open. The aim of this paper is to give a positive answer to this question. This also allows us to characterise bisimilarity of states of automata in terms of languages, which has been a long-standing unsolved problem in this theory.

We have done our best to keep our paper self-contained. Accordingly, Section 2 covers several topics of formal languages, automata, and coalgebras. Our main result is presented in Section 3. We conclude the paper by justifying why our description is the most natural one and by establishing some questions for future research.

2. Automata and formal languages

An introduction to the classical theory of finite automata can be found in [15]. Since our treatment of automata differs from the usual with respect to the initial state, we have preferred to recall first some basic concepts:

Definition 2.1. An *alphabet* is a finite non-empty set, whose elements are called *letters*.

Definition 2.2. A *finite word* over an alphabet A is either the empty word ϵ or a sequence $a_1a_2\dots a_r$ of letters of A. The set of all finite words over A is denoted by A^*.

Note that A^* can be regarded as the free monoid on the set A, where the multiplication in A^* is defined as the juxtaposition of words. In the rest of the paper, we will only consider finite words.

Definition 2.3. A *language* (or *formal language*) over an alphabet A is a subset of A^*, that is, a set of words over A.

Definition 2.4 *(Operations with languages).* If L, L_1, and L_2 are languages, we define:

1. the *sum* $L_1 + L_2 = L_1 \cup L_2$ of L_1 and L_2, which coincides with the set-theoretical union of L_1 and L_2,
2. the *product* $L_1L_2 = \{w_1w_2 \mid w_1 \in L_1,\ w_2 \in L_2\}$ of L_1 and L_2, composed by the words which are the result of concatenating one word of L_1 and one word of L_2, and
3. the *Kleene star* $L^* = \bigcup_{n\geqslant 0} L^n$ of L, where $L^0 = \{\epsilon\}$, $L^1 = L$ and $L^{n+1} = L^nL$ for $n \in \mathbb{N}$.

Definition 2.5. The set of all regular languages $\mathcal{R}$ is the smallest set of languages containing all finite languages and which is closed under taking sums, products, and Kleene stars.

It is usual to identify a letter a with the language $\{a\}$. With this criterion, we can identify the regular languages with the so-called *regular expressions*.

Regular languages are closely connected with finite automata. In this paper we will deal with the next generalisations of the notion of finite automata, in which infinite sets of states are allowed.

Definition 2.6. A *non-deterministic automaton* (respectively, a *deterministic automaton*, a *partial deterministic automaton*) is a quadruple $\mathcal{A} = (S, A, S_f, \delta)$ in which S is a set (not necessarily finite) whose elements are called *states*, A is an alphabet, S_f is a subset of S whose members will be called *final states* or *accepting states*, and the function $\delta : S \times A \longrightarrow \mathcal{P}_\omega(S)$ (respectively, the function $\delta : S \times A \longrightarrow S$ or the partial function $\delta : S \times A \rightharpoonup S$), called the *transition function*, assigns to each letter and to each state a *finite* set of states (respectively, a state, at most one state). When the set of states is finite we say that the corresponding automaton is *finite*.

Here $\mathcal{P}_\omega(S)$ denotes the set of all finite subsets of the set S. The finiteness restriction on the set of possible transitions from a given state is imposed here to ensure the existence of a final automaton.

It is also common to consider an initial state or a set of initial states in the study of finite automata, but we will not need it in our development, because eventually all states might play the role of the initial state. A deterministic automaton can be considered as a non-deterministic automaton by identifying an image s' of a state under the transition function with