Information Processing Letters 116 (2016) 356–360

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Complexity of atoms, combinatorially

Szabolcs Iván [1]

University of Szeged, Hungary

ARTICLE INFO

Article history:
Received 26 April 2014
Received in revised form 6 January 2016
Accepted 7 January 2016
Available online 12 January 2016
Communicated by A. Muscholl

Keywords:
Regular language
Atoms of a language
Syntactic left-congruence classes
Automata theory
Formal languages

ABSTRACT

Atoms of a (regular) language L were introduced by Brzozowski and Tamm in 2011 as intersections of complemented and uncomplemented quotients of L. They derived tight upper bounds on the complexity of atoms in 2012. In 2014, Brzozowski and Davies characterized the regular languages meeting these bounds. To achieve these results, they used the so-called "átomaton" of a language, introduced by Brzozowski and Tamm in 2011. In this note we give an alternative proof of their characterization, via a purely combinatorial approach.

1. Introduction

The state complexity of a regular language L is the number of states of its minimal automaton. An atom of a language is a non-empty intersection of its quotients, some of which may be complemented. Brzozowski and Tamm introduced atoms in [6] and found tight upper bounds for their state complexity in [7], carefully analyzing a particular nondeterministic finite automaton, the so-called "átomaton" of a regular language also introduced in [6].

A language is defined to be *maximally atomic* in [5] if it has the maximal number of atoms possible and each of the individual atoms has the maximal possible state complexity. In [4], Brzozowski and Davies showed that maximal syntactic complexity implies maximal atomicity and in [5] they gave necessary and sufficient conditions for a language to be maximally atomic.

In this paper we introduce another tool which we call the "disjoint power square automaton" of a regular language, and give a self-contained, purely combinatorial and automata-theoretic proof of their characterization.

2. Notation

A *semigroup* $(S,\cdot)$ is a set S equipped with a binary associative operation $\cdot$. We usually omit the sign $\cdot$ and write st for $s\cdot t$. A *monoid* is a semigroup $(S,\cdot)$ having a neutral element 1 satisfying $s1=1s=s$ for each $s\in S$. Given a finite nonempty set Q, two particular semigroups are $\mathcal{T}_Q$ consisting of all the transformations of Q (i.e. functions $Q\to Q$ with function composition as product) and its subsemigroup $\mathcal{P}_Q$ consisting of the *permutations* of Q. In order to ease notation in the automata theoretic part, we write function application in diagrammatic order, i.e. if $p\in Q$ and $f\in\mathcal{T}_Q$, then pf stands for the value to which f maps p, and for $f,g\in\mathcal{T}_Q$ their product is fg defined as $p(fg)=(pf)g$ for each p. Also, when $f\in\mathcal{T}_Q$ and $S\subseteq Q$, then Sf stands for the set $\{sf : s\in S\}$. The *rank* of a transformation $f\in\mathcal{T}_Q$ is the cardinality of its image Qf; transformations of rank n are called *permutations*, while all other transformations are called *singular* transformations.

E-mail address: szabivan@inf.u-szeged.hu.

[1] This research was supported by the European Union and the State of Hungary, co-financed by the European Social Fund in the framework of TÁMOP 4.2.4.A/2-11-1-2012-0001 'National Excellence Program' and by the NKFI grant no. K 108448.

http://dx.doi.org/10.1016/j.ipl.2016.01.003
0020-0190/

When $n \geq 1$ is an integer, then $\mathcal{T}_n$ stands for the transformation semigroup $\mathcal{T}_{\{1,\ldots,n\}}$.

An *alphabet* is a finite nonempty set Σ of symbols. A Σ-*word* is a finite sequence $w = a_1 a_2 \ldots a_n$ with each a_i being in Σ. For $n = 0$ we get the *empty word*, denoted ε. The set Σ^* of all words forms a monoid with the operation being (con)catenation, or simply product of words given by $a_1 \ldots a_n \cdot b_1 \ldots b_k = a_1 \ldots a_n b_1 \ldots b_k$. In this monoid, ε is the neutral element. The semigroup $\Sigma^+ = \Sigma^* - \{\varepsilon\}$ is the semigroup of nonempty words.

A *language* (over Σ) is an arbitrary subset of Σ^*. A *finite automaton* is a system $M = (Q, \Sigma, \delta, q_0, F)$ with Q being the finite nonempty set of states, Σ being the input alphabet, $\delta : Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the start state and $F \subseteq Q$ is the set of final states. Given M, the monoid Σ^* acts on Q from the right as $q \cdot_M \varepsilon = q$ and $q \cdot_M ua = \delta(q \cdot_M u, a)$ for each $q \in Q$, $u \in \Sigma^*$ and $a \in \Sigma$. When M is clear from the context, we omit the subscript and, in most cases, also the period and write only qw for $q \cdot_M w$. Then, each word w *induces* a function $Q \to Q$, denoted by w_M, defined as $q \mapsto qw$. The *transformation semigroup* of M is $\mathcal{T}(M) = \{w_M : w \in \Sigma^+\}$ – it is clear that $u_M v_M = (uv)_M$ so $\mathcal{T}(M)$ is indeed a semigroup. Most of the time, when M is clear, we omit the subscript also here and identify w with w_M. Another semigroup associated to M is that of its permutation group $\mathcal{P}(M) = \{w_M : w \in \Sigma^*, Q\,w = Q\}$. The language recognized by M is the language $L(M) = \{w : q_0 w \in F\}$. A language is called *regular* if it can be recognized by a finite automaton. It is well-known that for each regular language there exists a *minimal* automaton, unique up to isomorphism having the least number of states among all the automata recognizing L.

A state q of an automaton is *reachable* from a state p if $pw = q$ for some word w. States that are reachable from q_0 are simply called *reachable* states. A *sink* is a non-final state $p \notin F$ such that $pa = p$ for each $a \in \Sigma$ (thus, $pw = p$ for each word w as well). Two states p, q are called *distinguishable* if there exists a word w such that exactly one of the states pw and qw belongs to F. It is known that M is minimal iff each pair $p \neq q$ of its states is distinguishable and all its states are reachable. When $M = (Q, \Sigma, \delta, q_0, F)$ is an automaton and $q \in Q$, then M_q stands for the automaton $(Q, \Sigma, \delta, q, F)$ and L_q for the language recognized by M_q. A state q is *empty* if so is L_q. In a minimal automaton, there is at most one empty state which is then a sink. For a subset $S \subseteq Q$ of states, let L_S stand for $\cup_{q \in S} L_q$.

Given a (regular) language $L \subseteq \Sigma^*$, a well-known associated congruence on words is its *syntactic right congruence* $\sim_L$ defined as

$$x \sim_L y \;\Leftrightarrow\; (\forall z : \; xz \in L \Leftrightarrow yz \in L).$$

It is known that the minimal automaton of a regular language L is isomorphic to $(\Sigma^*/\sim_L, \Sigma, \delta_L, \varepsilon/\sim_L, L/\sim_L)$ where $\delta_L(x/\sim_L, a) = xa/\sim_L$.

Similarly,[2] one can define the syntactic *left* congruence ${}_L\!\sim$ of a language defined dually as

$$x \;{}_L\!\sim\; y \;\Leftrightarrow\; (\forall z : \; zx \in L \Leftrightarrow zy \in L).$$

The *reversal* of a word $w = a_1 \ldots a_n$ is the word $w^R = a_n \ldots a_1$, and the reversal of the language L is $L^R = \{w^R : w \in L\}$. Then obviously, $x \;{}_L\!\sim\; y$ if and only if $x^R \sim_{L^R} y^R$, since $zx \in L$ holds iff $x^R z^R \in L^R$. Hence, classes of the syntactic left congruence are precisely the reversals of the syntactic right congruence classes of L^R.

3. Atoms of a regular language

Let $L \subseteq \Sigma^*$ be a regular language and $M = (Q, \Sigma, \delta, q_0, F)$ be its minimal automaton. Let n stand for $|Q|$, the *state complexity* of L. An *atom* of L, as defined in [7], is a nonempty language of the form

$$A_S \;=\; \bigcap_{q \in S} L_q \cap \bigcap_{q \notin S} \overline{L_q},$$

for some $S \subseteq Q$. Here $\overline{X}$ stands for complementation with respect to Σ^*, i.e. $\Sigma^* - X$. It is clear that L has at most 2^n atoms.

That is, a word w is in A_S if $qw \in F$ iff $q \in S$, or equivalently, if $Sw \subseteq F$ and $\overline{S}w \subseteq \overline{F}$. (For $X \subseteq Q$, $\overline{X}$ denotes $Q - X$.) An immediate consequence of this characterization is that the atoms of a language are precisely the classes of its syntactic left congruence. Indeed, first observe that each word u belongs to precisely one atom A_S – to which $S = \{q : qu \in F\} = \{q_0 w : wu \in L\}$. Hence, u and v belong to the same atom A_S iff $S = \{q_0 w : wu \in L\} = \{q_0 w : wv \in L\}$ iff $u \;{}_L\!\sim\; v$. Thus, atoms are in a one-to-one correspondence with the states of the minimal automaton of L^R, in particular the number of atoms of L coincides with the state complexity of L^R.

In [5,7], the authors achieved results on properties of atoms such as the number and state complexity of individual atoms, via studying the "átomaton" of L, which is a nondeterministic automaton, actually being isomorphic to the reversal of the determinized reversal of M. In this paper we suggest another way to study atoms and reprove the characterization of the so-called maximally atomic languages.

To achieve this, we define a modified power set automaton, the *disjoint power square (DPS) automaton* $\mathrm{DPS}(M) = (Q', \Sigma, \Delta, p, F')$ of M as follows:

- $Q' \subseteq \big(P(Q) \times P(Q)\big) \cup \{\bot\}$ consists of the state pairs (S, T) for $S, T \subseteq Q$ with $S \cap T = \emptyset$, and a sink state $\bot$.
- $\Delta(\bot, a) = \bot$ and

$$\Delta((S,T), a) = \begin{cases} (Sa, Ta) & \text{if } Sa \cap Ta = \emptyset \\ \bot & \text{otherwise} \end{cases}.$$

- p is an arbitrary state.
- $F' = P(F) \times P(\overline{F})$, that is, $\{(S,T) : S \subseteq F, T \subseteq \overline{F}\}$.

[2] Though the notion of syntactic left congruence seems to be natural, we have not found any of its appearance in the literature in this form. However, as it turns out, atoms are precisely the classes of this equivalence relation.