[Array 15 (2022) 100203](https://doi.org/10.1016/j.array.2022.100203)



Contents lists available at [ScienceDirect](http://www.elsevier.com/locate/array)

Array

journal homepage: [www.elsevier.com/locate/array](http://www.elsevier.com/locate/array)

Decision trees for regular factorial languages

Mikhail Moshkov

*Computer, Electrical and Mathematical Sciences and Engineering Division and Computational Bioscience Research Center, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia*

A R T I C L E I N F O A B S T R A C T

*Keywords:*

Regular factorial language Recognition problem Membership problem Deterministic decision tree Nondeterministic decision tree

In this paper, we study arbitrary regular factorial languages over a finite alphabet *𝛴*. For the set of words

*𝐿*(*𝑛*) of the length *𝑛* belonging to a regular factorial language *𝐿*, we investigate the depth of decision trees

recognition problem, for a given word from *𝐿*(*𝑛*), we should recognize it using queries each of which, for some solving the recognition and the membership problems deterministically and nondeterministically. In the case of

*𝑖* ∈ {1*,* … *, 𝑛*}, returns the *𝑖*th letter of the word. In the case of membership problem, for a given word over the alphabet *𝛴* of the length *𝑛*, we should recognize if it belongs to the set *𝐿*(*𝑛*) using the same queries. For a given problem and type of trees, instead of the minimum depth *ℎ*(*𝑛*) of a decision tree of the considered type solving the problem for *𝐿*(*𝑛*), we study the smoothed minimum depth *𝐻* (*𝑛*) = max{*ℎ*(*𝑚*) ∶ *𝑚* ≤ *𝑛*}. With the growth of *𝑛*, the smoothed minimum depth of decision trees solving the problem of recognition deterministically is

either bounded from above by a constant, or grows as a logarithm, or linearly. For other cases (decision trees

deterministically and nondeterministically), with the growth of *𝑛*, the smoothed minimum depth of decision solving the problem of recognition nondeterministically, and decision trees solving the membership problem

trees is either bounded from above by a constant or grows linearly. As corollaries of the obtained results, we study joint behavior of smoothed minimum depths of decision trees for the considered four cases and describe five complexity classes of regular factorial languages. We also investigate the class of regular factorial languages

over the alphabet {0*,* 1} each of which is given by one forbidden word.

# Introduction

finite alphabet *𝛴*. A factorial language satisfies the following condition: In this paper, we study arbitrary regular factorial languages over a if a word *𝑤*1*𝑢𝑤*2 belongs to the language, then the word *𝑢* also belongs to it. For the set of words *𝐿*(*𝑛*) of the length *𝑛* belonging to a regular factorial language *𝐿*, we investigate the depth of decision trees solving

the recognition and the membership problems deterministically and nondeterministically. In the case of recognition problem, for a given

word from *𝐿*(*𝑛*), we should recognize it using queries each of which,

for some *𝑖* ∈ {1*,* … *, 𝑛*}, returns the *𝑖*th letter of the word. In the case of membership problem, for a given word over the alphabet *𝛴* of the length *𝑛*, we should recognize if it belongs to *𝐿*(*𝑛*) using the same

queries.

For a given problem (problem of recognition or membership prob- lem) and type of trees (solving the problem deterministically or non-

deterministically), instead of the minimum depth *ℎ*(*𝑛*) of a decision

tree of the considered type solving the problem for *𝐿*(*𝑛*), we study the smoothed minimum depth *𝐻* (*𝑛*) = max{*ℎ*(*𝑚*) ∶ *𝑚* ≤ *𝑛*}.

For an arbitrary regular factorial language, with the growth of *𝑛*,

the smoothed minimum depth of decision trees solving the problem

of recognition deterministically is either bounded from above by a

constant, or grows as a logarithm, or linearly. These results follow immediately from more general, obtained in [[1](#_bookmark23)] for arbitrary regular languages.

For other cases (decision trees solving the problem of recognition nondeterministically, and decision trees solving the membership prob- lem deterministically and nondeterministically), with the growth of

*𝑛*, the smoothed minimum depth of decision trees is either bounded

from above by a constant, or grows linearly. In the conference pa-

per [[2](#_bookmark24)], a classification of arbitrary regular languages depending on the smoothed minimum depth of decision trees solving the problem of recognition nondeterministically was announced without proofs. In the present paper, we consider simpler classification for regular factorial languages with full proof. Results related to the decision trees solving the membership problem are new.

As corollaries of the obtained results, we study joint behavior of smoothed minimum depths of decision trees for the considered four cases and describe five complexity classes of regular factorial lan- guages. We also investigate the class of regular factorial languages over

the alphabet *𝐸* = {0*,* 1} each of which is given by one forbidden word.

guage *𝐿* over a finite alphabet *𝛴* is to study its so-called combinatorial A well-known approach to evaluate complexity of an infinite lan-

*E-mail address:* [mikhail.moshkov@kaust.edu.sa](mailto:mikhail.moshkov@kaust.edu.sa).

<https://doi.org/10.1016/j.array.2022.100203>

Received 7 January 2022; Received in revised form 2 June 2022; Accepted 3 June 2022

Available online 8 June 2022

2590-0056/© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license ([http://creativecommons.org/licenses/by-](http://creativecommons.org/licenses/by-nc-nd/4.0/) [nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)).

complexity (known also as counting function) *𝑓𝐿*(*𝑛*) that is the number of words of the length *𝑛* in *𝐿* [[3](#_bookmark25),[4](#_bookmark26)]. The present paper proposes additional ways to evaluate the complexity of the language *𝐿* based on

the study how the depth of decision trees solving the recognition and the membership problems deterministically and nondeterministically depends on the length of words. This way is more complicated, but

compare languages generated by diagrams *𝐼*3 and *𝐼*4 depicted in [Figs.](#_bookmark11) [5](#_bookmark11) can give more detailed classification of languages. To show this, we

and [6](#_bookmark13). For both languages, the counting function grows linearly. For the first language, the minimum depth of decision trees solving the problem of recognition deterministically grows as a logarithm, but for the second language, the minimum depth of decision trees solving the problem of recognition deterministically grows linearly.

obtained for languages over the alphabet *𝐸* that are subword-closed: We should mention a recent paper [[5](#_bookmark27)] in which similar results were if a word *𝑤*1*𝑢*1*𝑤*2 ⋯ *𝑤𝑚𝑢𝑚𝑤𝑚*+1 belongs to the language, then the word

*𝑢*1 ⋯ *𝑢𝑚* also belongs to it.

It is clear that each subword-closed language is a factorial language.

regular language [[6](#_bookmark28)]. One can show that the language *𝐿*(00) over Moreover, each subword-closed language over a finite alphabet is a the alphabet *𝐸* given by one forbidden word 00 is a regular factorial

closed languages over the alphabet *𝐸* is a proper subclass of the class language, which is not subword-closed. Therefore the class of subword- of regular factorial languages over the alphabet *𝐸*.

The main difference between the present paper and [[5](#_bookmark27)] is that, in the latter paper, we do not assume that the subword-closed lan- guages are given by deterministic finite automata. Instead of this, we describe simple criteria (based on the presence in the language of words of special types) for the behavior of the minimum depths of decision trees solving the problem of recognition deterministically and nondeterministically. Differently formulated criteria for the behavior of the minimum depth of decision trees solving the recognition problem require very different proofs. One more difference is that in [[5](#_bookmark27)] we directly consider the minimum depth of decision trees.

The rest of the paper is organized as follows. In Section [2](#_bookmark0), we consider main notions, in Section [3](#_bookmark1) – main results, and in Section [4](#_bookmark12) – two corollaries of these results.

# Main notions

In this section, we discuss the notions related to regular facto- rial languages and decision trees solving problems of recognition and membership for these languages.

* 1. *Regular factorial languages*

Let *𝜔* = {0*,* 1*,* 2*,* …} be the set of nonnegative integers and *𝛴* be a

node are labeled with pairwise different letters, *𝑞*0 is a node of *𝐺* called starting, and *𝑄* is a nonempty set of the graph *𝐺* nodes called final.

A path of the diagram *𝐼* is an arbitrary sequence *𝜉* = *𝑣*1*, 𝑑*1*,* … *, 𝑣𝑚,*

*𝑑𝑚, 𝑣𝑚*+1 of nodes and edges of *𝐺* such that the edge *𝑑𝑖* leaves the node

*𝑣𝑖* and enters the node *𝑣𝑖*+1 for *𝑖* = 1*,* … *, 𝑚*. We now define a word *𝑤*(*𝜉*) from *𝛴*∗ in the following way: if *𝑚* = 0, then *𝑤*(*𝜉*) = *𝜆*. Let *𝑚 >* 0 and let *𝛿𝑗* be the letter attached to the edge *𝑑𝑗* , *𝑗* = 1*,* … *, 𝑚*. Then

*𝑤*(*𝜉*) = *𝛿*1 ⋯ *𝛿𝑚*. We say that the path *𝜉* generates the word *𝑤*(*𝜉*). Note

that different paths which start in the same node generate different

words.

We denote by *𝛯*(*𝐼* ) the set of all paths of the diagram *𝐼* each of which starts in the node *𝑞*0 and finishes in a node from *𝑄*. Let

*𝐿𝐼* = {*𝑤*(*𝜉*) ∶ *𝜉* ∈ *𝛯*(*𝐼* )}*.*

We say that the diagram *𝐼* generates the language *𝐿𝐼* . It is well known that *𝐿𝐼* is a regular language.

The diagram *𝐼* is called complete over the alphabet *𝛴* if exactly

|*𝛴*| edges leave each node of *𝐺*. Note that these edges are labeled with pairwise different letters from *𝛴*. Such diagram corresponds to a complete DFA [[8](#_bookmark30)]. The diagram *𝐼* is called reduced if, for each node of *𝐺*, there exists a path from *𝛯*(*𝐼* ), which contains this node. Such

each regular language over the alphabet *𝛴*, there exists a complete over diagram corresponds to a reduced DFA [[8](#_bookmark30)]. It is known [[8](#_bookmark30)] that, for the alphabet *𝛴* diagram, which generates this language. Therefore, for

each nonempty regular language, there exists a reduced diagram, which generates this language.

Let *𝐿* be a regular factorial language and *𝐼* = (*𝐺, 𝑞*0*, 𝑄*) be a

reduced diagram that generates the language *𝐿*. Since the language *𝐿* is factorial, we can assume additionally that each node of the graph *𝐺* is final — it will not change the language generated by *𝐼* since with each word the language *𝐿* contains each prefix of this word. The diagram

*𝐼* will be called f-reduced if it is reduced and each node of the graph

*𝐺* is final. Further we will assume that a considered regular factorial language *𝐿* is nonempty and it is given by an f-reduced diagram, which

generates this language.

We will not consider nondeterministic finite automata (NFA) to rep- resent regular factorial languages since the study of NFA is essentially more complicated task.

* 1. *Decision trees for recognition and membership problems*

Let *𝐿* be a regular factorial language over the alphabet *𝛴*. For any natural *𝑛*, denote *𝐿*(*𝑛*) = *𝐿* ∩ *𝛴𝑛*, where *𝛴𝑛* is the set of words over the alphabet *𝛴*, which length is equal to *𝑛*. We consider two problems related to the set *𝐿*(*𝑛*). The problem of recognition: for a given word from *𝐿*(*𝑛*), we should recognize it using attributes (queries)

*𝑙𝑛,* … *, 𝑙𝑛*, where *𝑙𝑛*, *𝑖* ∈ {1*,* … *, 𝑛*}, is a function from *𝛴𝑛* to *𝛴* such that

finite alphabet with at least two letters. By *𝛴*∗, we denote the set of all

1 *𝑛 𝑖*

*𝑙𝑛*(*𝑎*1 ⋯ *𝑎𝑛*) = *𝑎𝑖* for any word *𝑎*1 ⋯ *𝑎𝑛* ∈ *𝛴𝑛*. The problem of membership:

finite words over the alphabet *𝛴*, including the empty word *𝜆*. A word *𝑖 𝑛*

*𝑤* ∈ *𝛴*∗ is called a factor of a word *𝑢* ∈ *𝛴*∗ if *𝑢* = *𝑣 𝑤𝑣*

and *𝑣 , 𝑣*

∈ *𝛴*∗.

for a given word from *𝛴* , we should recognize if this word belongs to

A language *𝐿 ⊆ 𝛴*∗

1 2 1 2

is called factorial if it contains all factors of its

the set *𝐿*(*𝑛*) using the same attributes. To solve these problems, we use

decision trees over *𝐿*(*𝑛*).

words. A word *𝑤* ∈ *𝛴*∗ is called a minimal forbidden word for *𝐿* if

*𝑤* ∉ *𝐿* and all proper factors of *𝑤* belong to *𝐿*. We denote by *𝑀𝐹* (*𝐿*) the language of minimal forbidden words for *𝐿*. It is known [[7](#_bookmark29)] that a factorial language *𝐿* is regular if and only if the language *𝑀𝐹* (*𝐿*) is regular. In particular, a factorial language *𝐿* with a finite set of minimal forbidden words *𝑀𝐹* (*𝐿*) is regular. In this paper, we study arbitrary

nonempty regular factorial languages.

It is well known that each regular language can be represented by a deterministic finite automaton (DFA) [[8](#_bookmark30)]. As in [[8](#_bookmark30)], we will consider not only complete DFA with total transition function but also partial DFA with partial transition function. Such DFA can be represented by its transition diagram (diagram for short) [[9](#_bookmark31)].

A diagram over the alphabet *𝛴* is a triple *𝐼* = (*𝐺, 𝑞*0*, 𝑄*), where *𝐺*

which each edge is labeled with a letter from *𝛴* and edges leaving each is a finite directed graph, possibly with multiple edges and loops, in

A decision tree over *𝐿*(*𝑛*) is a marked finite directed tree with root,

which has the following properties:

* The root and the edges leaving the root are not labeled.

with an attribute from the set {*𝑙𝑛,* … *, 𝑙𝑛*}. • Each node, which is not the root nor terminal node, is labeled

1 *𝑛*

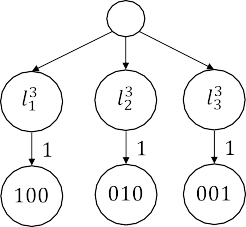
letter from the alphabet *𝛴*. • Each edge leaving a node, which is not a root, is labeled with a

A decision tree over *𝐿*(*𝑛*) is called deterministic if it satisfies the

following conditions:

* Exactly one edge leaves the root.
* For any node, which is not the root nor terminal node, the edges leaving this node are labeled with pairwise different letters.

# Bounds on decision tree depth

Let *𝐿* be a nonempty factorial regular language. In this section,

we consider the behavior of four functions *𝐻 𝑟𝑎*, *𝐻 𝑟𝑑* , *𝐻𝑚𝑎*, and *𝐻𝑚𝑑*

*𝐿 𝐿 𝐿 𝐿*

defined on the set *𝜔* ⧵ {0} and with values from *𝜔*. For any natural *𝑛*,

*𝐻 𝑟𝑎*(*𝑛*) = max{*ℎ𝑟𝑎*(*𝑚*) ∶ 1 ≤ *𝑚* ≤ *𝑛*}*,*

*𝐿 𝐿*

*𝐻 𝑟𝑑* (*𝑛*) = max{*ℎ𝑟𝑑* (*𝑚*) ∶ 1 ≤ *𝑚* ≤ *𝑛*}*,*

*𝐿 𝐿*

*𝐻𝑚𝑎*(*𝑛*) = max{*ℎ𝑚𝑎*(*𝑚*) ∶ 1 ≤ *𝑚* ≤ *𝑛*}*,*

*𝐿 𝐿*

*𝐻𝑚𝑑* (*𝑛*) = max{*ℎ𝑚𝑑* (*𝑚*) ∶ 1 ≤ *𝑚* ≤ *𝑛*}*.*

*𝐿 𝐿*

For any pair *𝑏𝑐* ∈ {*𝑟𝑎, 𝑟𝑑, 𝑚𝑎, 𝑚𝑑*}, the function *𝐻 𝑏𝑐*(*𝑛*) is a smoothed

*𝐿*

*𝐿*

**/ig. 1.** Decision trees that solve the problem of recognition for the set of words

{100*,* 010*,* 001} deterministically and nondeterministically.

Let *𝛤* be a decision tree over *𝐿*(*𝑛*). A complete path in *𝛤* is any sequence *𝜉* = *𝑣*0*, 𝑒*0*,* … *, 𝑣𝑚, 𝑒𝑚, 𝑣𝑚*+1 of nodes and edges of *𝛤* such that

*𝑣*0 is the root, *𝑣𝑚*+1 is a terminal node, and *𝑣𝑖* is the initial and *𝑣𝑖*+1 is

the terminal node of the edge *𝑒𝑖* for *𝑖* = 0*,* … *, 𝑚*. We define a subset

*𝛴*(*𝑛, 𝜉*) of the set *𝛴𝑛* in the following way: if *𝑚* = 0, then *𝛴*(*𝑛, 𝜉*) = *𝛴𝑛*.

analog of the function *ℎ𝑏𝑐*(*𝑛*).

* 1. *Decision trees solving recognition problem deterministically*

Let *𝐼* = (*𝐺, 𝑞*0*, 𝑄*) be a f-reduced diagram over the alphabet *𝛴*. A path of the diagram *𝐼* is called a cycle of the diagram *𝐼* if there is at

the last node of this path. A cycle of the diagram *𝐼* is called elementary least one edge in this path, and the first node of this path is equal to

if nodes of this cycle, with the exception of the last node, are pairwise different.

Let *𝑚 >* 0, the attribute *𝑙𝑛*

*𝑗*

*𝑖*

be attached to the node *𝑣𝑗* , and *𝑏𝑗* be the

The diagram *𝐼* is called simple if every two different elementary

letter attached to the edge *𝑒𝑗* , *𝑗* = 1*,* … *, 𝑚*. Then

*𝛴*(*𝑛, 𝜉*) = {*𝑎*1 ⋯ *𝑎𝑛* ∈ *𝛴𝑛* ∶ *𝑎𝑖* = *𝑏*1*,* … *, 𝑎𝑖* = *𝑏𝑚*}*.*

1

*𝑚*

Let *𝐿*(*𝑛*) ≠ ∅. We say that a decision tree *𝛤* over *𝐿*(*𝑛*) solves the problem of recognition for *𝐿*(*𝑛*) nondeterministically if *𝛤* satisfies the

following conditions:

* Each terminal node of *𝛤* is labeled with a word from *𝐿*(*𝑛*).
* For any word *𝑤* ∈ *𝐿*(*𝑛*), there exists a complete path *𝜉* in the tree

*𝛤* such that *𝑤* ∈ *𝛴*(*𝑛, 𝜉*).

* For any word *𝑤* ∈ *𝐿*(*𝑛*) and for any complete path *𝜉* in the tree *𝛤* such that *𝑤* ∈ *𝛴*(*𝑛, 𝜉*), the terminal node of the path *𝜉* is labeled with the word *𝑤*.

We say that a decision tree *𝛤* over *𝐿*(*𝑛*) solves the problem of recog- nition for *𝐿*(*𝑛*) deterministically if *𝛤* is a deterministic decision tree, which solves the problem of recognition for *𝐿*(*𝑛*) nondeterministically.

Examples of decision trees illustrating the considered notions are presented in [Fig.](#_bookmark2) [1](#_bookmark2).

We say that a decision tree *𝛤* over *𝐿*(*𝑛*) solves the problem of

membership for *𝐿*(*𝑛*) nondeterministically if *𝛤* satisfies the following

conditions:

* Each terminal node of *𝛤* is labeled with a number from the set

{0*,* 1}.

* For any word *𝑤* ∈ *𝛴𝑛*, there exists a complete path *𝜉* in the tree

*𝛤* such that *𝑤* ∈ *𝛴*(*𝑛, 𝜉*).

cycles of the diagram *𝐼* do not have common nodes. Let *𝐼* be a simple diagram and *𝜉* be a path of the diagram *𝐼* . The number of different elementary cycles of the diagram *𝐼* , which have common nodes with

*𝜉*, is denoted by *𝑐𝑙*(*𝜉*) and is called the cyclic length of the path *𝜉*. The

value

*𝑐𝑙*(*𝐼* ) = max{*𝑐𝑙*(*𝜉*) ∶ *𝜉* ∈ *𝛯*(*𝐼* )}

is called the cyclic length of the diagram *𝐼* .

Let *𝐼* be a simple diagram, *𝐶* be an elementary cycle of the diagram

*𝐼* , and *𝑣* be a node of the cycle *𝐶*. Beginning with the node *𝑣*, the cycle

*𝐶* generates an infinite periodic word over the alphabet *𝛴*. This word will be denoted by *𝑊* (*𝐼 , 𝐶, 𝑣*). We denote by *𝑟*(*𝐼 , 𝐶, 𝑣*) the minimum period of the word *𝑊* (*𝐼 , 𝐶, 𝑣*). The diagram *𝐼* is called dependent if there exist two different elementary cycles *𝐶*1 and *𝐶*2 of the diagram

*𝐼* , nodes *𝑣*1 and *𝑣*2 of the cycles *𝐶*1 and *𝐶*2, respectively, and a path *𝜋*

of the diagram *𝐼* from *𝑣*1 to *𝑣*2, which satisfy the following conditions:

*𝑊* (*𝐼 , 𝐶*1*, 𝑣*1) = *𝑊* (*𝐼 , 𝐶*2*, 𝑣*2) and the length of the path *𝜋* is a number di- visible by *𝑟*(*𝐼 , 𝐶*1*, 𝑣*1). If the diagram *𝐼* is not dependent, then it is called

independent. Next theorem follows immediately from Theorem 2.1 [[1](#_bookmark23)], which is a similar statement that holds for all regular languages.

**Theorem 1.** *Let 𝐿 be a nonempty regular factorial language over the alphabet 𝛴 and 𝐼 be a f-reduced diagram, which generates the language 𝐿.*

*Then the following statements hold:*

* + 1. *If 𝐼 is an independent simple diagram and 𝑐𝑙*(*𝐼* ) ≤ 1*, then 𝐻 𝑟𝑑* (*𝑛*) =

*𝐿*

* For any word *𝑤* ∈ *𝛴𝑛* and for any complete path *𝜉* in the tree *𝛤* such that *𝑤* ∈ *𝛴*(*𝑛, 𝜉*), the terminal node of the path *𝜉* is labeled with the number 1 if *𝑤* ∈ *𝐿*(*𝑛*) and with the number 0, otherwise.

We say that a decision tree *𝛤* over *𝐿*(*𝑛*) solves the problem of mem- bership for *𝐿*(*𝑛*) deterministically if *𝛤* is a deterministic decision tree which solves the problem of membership for *𝐿*(*𝑛*) nondeterministically. Let *𝛤* be a decision tree over *𝐿*(*𝑛*). We denote by *ℎ*(*𝛤* ) the maximum number of nodes in a complete path in *𝛤* that are not the root nor terminal node. The value *ℎ*(*𝛤* ) is called the depth of the decision tree

*𝛤* .

We denote by *ℎ𝑟𝑎*(*𝑛*) (*ℎ𝑟𝑑* (*𝑛*)) the minimum depth of a decision tree

*𝑂*(1)*.*

* + 1. *If 𝐼 is an independent simple diagram and 𝑐𝑙*(*𝐼* ) ≥ 2*, then 𝐻 𝑟𝑑* (*𝑛*) =

*𝛩*(log *𝑛*)*.*

*𝐿*

* + 1. *If 𝐼 is not independent simple diagram, then 𝐻 𝑟𝑑* (*𝑛*) = *𝛩*(*𝑛*)*.*

*𝐿*

* 1. *Decision trees solving recognition problem nondeterministically*

Let *𝐿* be a nonempty regular factorial language over the alphabet

*𝛴*. For any natural *𝑛*, we define a parameter *𝑇𝐿*(*𝑛*) of the language *𝐿*. If *𝐿*(*𝑛*) = ∅, then *𝑇𝐿*(*𝑛*) = 0. Let *𝐿*(*𝑛*) ≠ ∅, *𝑤* = *𝑎*1 ⋯ *𝑎𝑛* ∈ *𝐿*(*𝑛*), and

*𝐽 ⊆* {1*,* … *, 𝑛*}. Denote *𝐿*(*𝑤, 𝐽* ) = {*𝑏*1 ⋯ *𝑏𝑛* ∈ *𝐿*(*𝑛*) ∶ *𝑏𝑗* = *𝑎𝑗 , 𝑗* ∈ *𝐽* }

over *𝐿*(*𝑛*)

*𝐿 𝐿*

*𝐿*(*𝑛*) nonde-

(if *𝐽* = ∅, then *𝐿*(*𝑤, 𝐽* ) = *𝐿*(*𝑛*)) and *𝑀𝐿*(*𝑛, 𝑤*) = min{|*𝐽* | ∶ *𝐽 ⊆*

terministically (deterministically). If *𝐿*(*𝑛*) = ∅, then *ℎ𝑟𝑎*(*𝑛*) = *ℎ𝑟𝑑* (*𝑛*) = , which solves the problem of recognition for 0.

*𝐿 𝐿*

{1*,* … *, 𝑛*}*,* |*𝐿*(*𝑤, 𝐽* )| = 1}. Then

*𝑇𝐿*(*𝑛*) = max{*𝑀𝐿*(*𝑛, 𝑤*) ∶ *𝑤* ∈ *𝐿*(*𝑛*)}*.*

We denote by *ℎ𝑚𝑎*(*𝑛*) (*ℎ𝑚𝑑* (*𝑛*)) the minimum depth of a decision

*𝐿 𝐿*

tree over *𝐿*(*𝑛*), which solves the problem of membership for *𝐿*(*𝑛*)

nondeterministically (deterministically). If *𝐿*(*𝑛*) = ∅, then *ℎ𝑚𝑎*(*𝑛*) =

Note that, for any word *𝑤* ∈ *𝐿*(*𝑛*), *𝑀𝐿*(*𝑛, 𝑤*) is the minimum number of

letters of the word *𝑤*, which allow us to distinguish it from all other

*ℎ𝑚𝑑* (*𝑛*) = 0.

*𝐿*

*𝐿*

words belonging to *𝐿*(*𝑛*).

**Lemma 2.** *Let 𝐿 be a nonempty regular factorial language over the alphabet 𝛴. Then ℎ𝑟𝑎*(*𝑛*) = *𝑇𝐿*(*𝑛*) *for any natural 𝑛.*

*𝐿*

**Proof.** First, we prove that *ℎ𝑟𝑎*(*𝑛*) ≥ *𝑇𝐿*(*𝑛*). Let *𝛤* be a decision tree over

*𝐿*(*𝑛*), which solves the problem of recognition for *𝐿*(*𝑛*) nondeterministi-

*𝐿*

cally and for which *ℎ*(*𝛤* ) = *ℎ𝑟𝑎*(*𝑛*). Let *𝑤* be a word from *𝐿*(*𝑛*) for which

*𝑇* (*𝑛*) = *𝑀* (*𝑛, 𝑤*)

*𝐿*

*𝛤* contains a complete path *𝜉*

**/ig. 2.** Diagram *𝐼*0.

*𝐿 𝐿*

. Then the decision tree

such that *𝑤* ∈ *𝛴*(*𝑛, 𝜉*) and the terminal node of the path *𝜉* is labeled with

the word *𝑤*. It is clear that *𝛴*(*𝑛, 𝜉*) ∩ *𝐿*(*𝑛*) = {*𝑤*}. Let *𝜉* contain *𝑚* nodes

that are not the root nor terminal node and *𝑙𝑛 ,* … *, 𝑙𝑛* be attributes

*𝐼* is a f-reduced diagram, it contains a path *𝜉* from the node *𝑞*0 to the

attached to these nodes. Denote

*𝑖*1

*𝑖𝑚*

node *𝑣*1, and all nodes of the graph *𝐺* are final. Let the path *𝜉* generate

*𝐽* = {*𝑖*1*,* … *, 𝑖𝑚*}. Then *𝐿*(*𝑤, 𝐽* ) = {*𝑤*}.

Therefore *𝑚* ≥ *𝑀𝐿*(*𝑛, 𝑤*) = *𝑇𝐿*(*𝑛*). It is clear that *ℎ*(*𝛤* ) ≥ *𝑚*. Thus,

*ℎ𝑟𝑎*(*𝑛*) = *ℎ*(*𝛤* ) ≥ *𝑚* ≥ *𝑀* (*𝑛, 𝑤*) = *𝑇* (*𝑛*).

the word *𝛼* of the length *𝑎*. Denote *𝑟* = *𝑟*(*𝐼 , 𝐶*1*, 𝑣*1). Let the length of the cycle *𝐶*1 be equal to *𝑏𝑟*, the length of the path *𝜋* be equal to *𝑐𝑟*, and

*𝐿 𝐿*

*𝐿*

*𝑟𝑎*

the path *𝜋* generate the word *𝛽*. Denote by *𝛾* the prefix of the length *𝑟*

We now prove that *ℎ𝐿* (*𝑛*) ≤ *𝑇𝐿*(*𝑛*). One can show that, for each

*𝑤* ∈ *𝐿*(*𝑛*), we can construct a complete path *𝜉𝑤*, which satisfies the following conditions: the number of nodes in *𝜉𝑤* that are not the root

of the word *𝑊* (*𝐼 , 𝐶*1*, 𝑣*1). We now define two words of the length *𝑟𝑏𝑐*:

*𝑢* = *𝛾𝑏𝑐* and *𝑤* = *𝛽𝛾𝑐*(*𝑏*−1). It is clear that *𝑢* ≠ *𝑤*.

Consider the sequence of numbers *𝑛* = *𝑎* + *𝑖𝑟𝑏𝑐*, *𝑖* = 1*,* 2*,* …. Let

nor terminal node is equal to *𝑀𝐿*(*𝑛, 𝑤*), *𝛴*(*𝑛, 𝜉𝑤*) ∩ *𝐿*(*𝑛*) = {*𝑤*}, and the

*𝑖*

*𝑖 𝑗*

*𝑖*−*𝑗*−1

terminal node of *𝜉𝑤* is labeled with the word *𝑤*. If we merge roots of all

*𝑖* ∈ *𝜔*⧵{0}. The set *𝐿*(*𝑛𝑖*) contains the word *𝛼𝑢* and the words *𝛼𝑢 𝑤𝑢*

for *𝑗* = 0*,* … *, 𝑖* − 1. It is easy to show that *𝑀𝐿*(*𝑛, 𝛼𝑢𝑖*) ≥ *𝑖*: to distinguish

paths *𝜉𝑤*, *𝑤* ∈ *𝐿*(*𝑛*), we obtain a decision tree, which solves the problem

of recognition for *𝐿*(*𝑛*) nondeterministically and which depth is equal

to *𝑇𝐿*(*𝑛*). Thus, *ℎ𝑟𝑎*(*𝑛*) ≤ *𝑇𝐿*(*𝑛*) and *ℎ𝑟𝑎*(*𝑛*) = *𝑇𝐿*(*𝑛*). □

the word *𝛼𝑢𝑖* from the words *𝛼𝑢𝑗 𝑤𝑢𝑖*−*𝑗*−1, *𝑗* = 0*,* … *, 𝑖* − 1, we need to use at least one letter from each of *𝑖* words *𝑢* appearing in *𝛼𝑢𝑖*. Therefore

*𝑟𝑎*

*𝐿 𝐿*

*𝑇𝐿*(*𝑛𝑖*) ≥ *𝑖* and, by [Lemma](#_bookmark4) [2](#_bookmark4), *ℎ𝐿* (*𝑛𝑖*) ≥ *𝑖* = (*𝑛𝑖* − *𝑎*)∕(*𝑟𝑏𝑐*). Let *𝑛* ≥ *𝑛*1

and let *𝑖* be the maximum natural number such that *𝑛* ≥ *𝑛𝑖*. Evidently,

**Theorem 3.** *Let 𝐿 be a nonempty regular factorial language over the*

*𝑛* − *𝑛𝑖* ≤ *𝑟𝑏𝑐*. Hence *𝐻 𝑟𝑎*(*𝑛*) ≥ *ℎ𝑟𝑎*(*𝑛𝑖*) ≥ (*𝑛* − *𝑟𝑏𝑐* − *𝑎*)∕(*𝑟𝑏𝑐*). Therefore

*𝐿 𝐿*

*alphabet 𝛴 and 𝐼* = (*𝐺, 𝑞*0*, 𝑄*) *be a f-reduced diagram, which generates the*

*language 𝐿. Then the following statements hold:*

* + 1. *If 𝐼 is an independent simple diagram, then 𝐻 𝑟𝑎*(*𝑛*) = *𝑂*(1)*.*
    2. *If 𝐼 is not independent simple diagram, then 𝐻 𝑟𝑎*(*𝑛*) = *𝛩*(*𝑛*)*.*

*𝐿*

*𝐿*

**Proof.** (a) Let *𝐼* be an independent simple diagram and *𝑐𝑙*(*𝐼* ) ≤ 1. By

[Theorem](#_bookmark3) [1](#_bookmark3), *𝐻 𝑟𝑑* (*𝑛*) = *𝑂*(1). It is clear that *𝐻 𝑟𝑎*(*𝑛*) ≤ *𝐻 𝑟𝑑* (*𝑛*). Therefore

*𝐻 𝑟𝑎*(*𝑛*) ≥ *𝑛*∕(2*𝑟𝑏𝑐*) for large enough *𝑛*. The inequality *𝐻 𝑟𝑎*(*𝑛*) ≤ *𝑛* is obvious. Thus, *𝐻 𝑟𝑎*(*𝑛*) = *𝛩*(*𝑛*). □

Note that in general case (when we consider not only factorial languages) the classification of reduced diagrams depending on the minimum depth of decision trees solving the problem of recognition nondeterministically is more complicated [[2](#_bookmark24)]. In particular, there exists

*𝐿*

*𝐿 𝐿*

*𝐿*

*𝐻 𝑟𝑎*(*𝑛*) = *𝑂*(1).

*𝐿*

*𝐿 𝐿*

a dependent simple reduced diagram *𝐼*0 (see [Fig.](#_bookmark5) [2](#_bookmark5)) with the starting

node labeled with the symbol + and the unique final node labeled with

Let *𝐼* be an independent simple diagram and *𝑐𝑙*(*𝐼* ) ≥ 2. Let *𝑛* be a

natural number. If *𝐿*(*𝑛*) = ∅, then *𝑇𝐿*(*𝑛*) = 0. Let *𝐿*(*𝑛*) ≠ ∅. Denote by *𝑑*

the symbol ∗ that generates the regular language *𝐿*0 = {0*𝑖*10*𝑗* ∶ *𝑖, 𝑗* ∈ *𝜔*}

over the alphabet {0*,* 1}, which is not factorial and for which *𝐻 𝑟𝑎* (*𝑛*) =

the number of nodes in the graph *𝐺*. In the proof of Lemma 4.5 [[1](#_bookmark23)], it

was proved that *𝑀𝐿*(*𝑛, 𝑤*) ≤ *𝑑*(4*𝑑* + 1) for any word *𝑤* ∈ *𝐿*(*𝑛*). Therefore

*𝑇𝐿*(*𝑛*) ≤ *𝑑*(4*𝑑* + 1). Thus, by [Lemma](#_bookmark4) [2](#_bookmark4), *ℎ𝑟𝑎*(*𝑛*) ≤ *𝑑*(4*𝑑* + 1) for any natural

*𝐿*

*𝑂*(1).

* 1. *Decision trees solving membership problem*

*𝐿*0

*𝑛* and *𝐻 𝑟𝑎*(*𝑛*) = *𝑂*(1).

*𝐿*

(b) Let *𝐼* be not simple diagram and *𝐶*1*, 𝐶*2 be different elementary cycles of the diagram *𝐼* , which have a common node *𝑣*. Since *𝐼* is a f-reduced diagram, it contains a path *𝜉* from the node *𝑞*0 to the node

*𝑣*, and *𝑣* is a final node. Let the length of the path *𝜉* be equal to *𝑎*, the length of the cycle *𝐶*1 be equal to *𝑏*, and the length of the cycle

*𝐶*2 be equal to *𝑐*. Let *𝛼* be the word generated by the path *𝜉*, *𝛽* be the

For a regular factorial language *𝐿*, the notation |*𝐿*| = ∞ means that

*𝐿* is an infinite language, and the notation |*𝐿*| *<* ∞ means that *𝐿* is a

finite language.

**Theorem 4.** *Let 𝐿 be a regular factorial language over the alphabet 𝛴.*

1. *If* |*𝐿*| = ∞ *and 𝐿* ≠ *𝛴*∗*, then 𝐻𝑚𝑑* (*𝑛*) = *𝛩*(*𝑛*) *and 𝐻𝑚𝑎*(*𝑛*) = *𝛩*(*𝑛*)*.*

*𝐿 𝐿*

word generated by a path from *𝑣* to *𝑣* obtained by the passage *𝑐* times

along the cycle *𝐶*1, and *𝛾* be the word generated by a path from *𝑣* to *𝑣*

obtained by the passage *𝑏* times along the cycle *𝐶*2. The words *𝛽* and *𝛾*

are different and they have the same length *𝑏𝑐*.

1. *If* |*𝐿*| *<* ∞ *or 𝐿* = *𝛴*∗*, then 𝐻𝑚𝑑* (*𝑛*) = *𝑂*(1) *and 𝐻𝑚𝑎*(*𝑛*) = *𝑂*(1)*.*

**Proof.** It is clear that *ℎ𝑚𝑎*(*𝑛*) ≤ *ℎ𝑚𝑑* (*𝑛*) for any natural *𝑛*.

*𝐿 𝐿*

1. Let |*𝐿*| = ∞, ∗, and be a word with the minimum length

*𝐿 𝐿*

Consider the sequence of numbers *𝑛*

= *𝑎* + *𝑖𝑏𝑐*, *𝑖* = 1*,* 2*,* …. Let

*𝐿* ≠ *𝛴*

∗

*𝑤*0

*𝑖*

*𝑖 𝑗*

*𝑖*−*𝑗*−1

from *𝛴* ⧵ *𝐿*. Denote by *𝑡* the length of *𝑤*0. Since |*𝐿*| = ∞, *𝐿*(*𝑛*) ≠ ∅ for

*𝑖* ∈ *𝜔*⧵{0}. The set *𝐿*(*𝑛𝑖*) contains the word *𝛼𝛾* and the words *𝛼𝛾 𝛽𝛾*

for *𝑗* = 0*,* … *, 𝑖* − 1. It is easy to show that *𝑀𝐿*(*𝑛𝑖, 𝛼𝛾𝑖*) ≥ *𝑖*: to distinguish the word *𝛼𝛾𝑖* from the words *𝛼𝛾𝑗 𝛽𝛾𝑖*−*𝑗*−1, *𝑗* = 0*,* … *, 𝑖* − 1, we need to use at least one letter from each of *𝑖* words *𝛾* appearing in *𝛼𝛾𝑖*. Therefore

*𝑇𝐿*(*𝑛𝑖*) ≥ *𝑖* and, by [Lemma](#_bookmark4) [2](#_bookmark4), *ℎ𝑟𝑎*(*𝑛𝑖*) ≥ *𝑖* = (*𝑛𝑖* − *𝑎*)∕(*𝑏𝑐*). Let *𝑛* ≥ *𝑛*1

and let *𝑖* be the maximum natural number such that *𝑛* ≥ *𝑛𝑖*. Evidently,

*𝐿*

*𝑛* − *𝑛𝑖* ≤ *𝑏𝑐*. Hence *𝐻 𝑟𝑎*(*𝑛*) ≥ *ℎ𝑟𝑎*(*𝑛𝑖*) ≥ (*𝑛* − *𝑏𝑐* − *𝑎*)∕(*𝑏𝑐*). Therefore

*𝐻 𝑟𝑎*(*𝑛*) ≥ *𝑛*∕(2*𝑏𝑐*) for large enough *𝑛*. The inequality *𝐻 𝑟𝑎*(*𝑛*) ≤ *𝑛* is

*𝐿*

*𝐿*

any natural *𝑛*. Let *𝑛* be a natural number such that *𝑛 > 𝑡* and *𝛤* be a decision tree over *𝐿*(*𝑛*) that solves the problem of membership for *𝐿*(*𝑛*) nondeterministically and has the minimum depth. Let *𝑤* ∈ *𝐿*(*𝑛*) and *𝜉* be a complete path in *𝛤* such that *𝑤* ∈ *𝛴*(*𝑛, 𝜉*). Then the terminal node of *𝜉* is labeled with the number 1. Beginning with the first letter, we divide the word *𝑤* into ⌊*𝑛*∕*𝑡*⌋ blocks with *𝑡* letters in each and the suffix of the length *𝑛* − *𝑡* ⌊*𝑛*∕*𝑡*⌋. Let us assume that the number of nodes labeled

with attributes in *𝜉* is less than ⌊*𝑛*∕*𝑡*⌋. Then there is a block such that

*𝐿*

obvious. Thus, *𝐻 𝑟𝑎*(*𝑛*) = *𝛩*(*𝑛*).

*𝐿*

*𝐿*

queries (attributes) attached to nodes of *𝜉* does not ask about letters

Let *𝐼* be a dependent simple diagram. Then there exist two different

elementary cycles *𝐶*1 and *𝐶*2 of the diagram *𝐼* , nodes *𝑣*1 and *𝑣*2 of the cycles *𝐶*1 and *𝐶*2, respectively, and a path *𝜋* of the diagram *𝐼* from *𝑣*1 to

*𝑣*2, which satisfy the following conditions: *𝑊* (*𝐼 , 𝐶*1*, 𝑣*1) = *𝑊* (*𝐼 , 𝐶*2*, 𝑣*2)

and the length of the path *𝜋* is a number divisible by *𝑟*(*𝐼 , 𝐶*1*, 𝑣*1). Let us remind that, for *𝑖* = 1*,* 2, *𝑊* (*𝐼 , 𝐶𝑖, 𝑣𝑖*) is the infinite periodic word over the alphabet *𝛴* generated by the cycle *𝐶𝑖* beginning with the node *𝑣𝑖*,

from the block. We replace this block in the word *𝑤* with the word

*𝑤*0 and denote by *𝑤*′ the obtained word. It is clear that *𝑤*′ ∉ *𝐿* and

*𝑤*′ ∈ *𝛴*(*𝑛, 𝜉*), but this is impossible since the terminal node of the path

*𝜉* is labeled with the number 1. Therefore the depth of *𝛤* is greater than or equal to ⌊*𝑛*∕*𝑡*⌋. Thus, *ℎ𝑚𝑎*(*𝑛*) ≥ ⌊*𝑛*∕*𝑡*⌋. It is easy to construct a decision tree over *𝐿*(*𝑛*) that solves the problem of membership for *𝐿*(*𝑛*) deterministically and has the depth equals to *𝑛*. Therefore *ℎ𝑚𝑑* (*𝑛*) ≤ *𝑛*.

*𝐿*

*𝐿*

and *𝑟*(*𝐼 , 𝐶*1*, 𝑣*1) is the minimum period of the word *𝑊* (*𝐼 , 𝐶*1*, 𝑣*1). Since

Thus, *𝐻𝑚𝑑* (*𝑛*) = *𝛩*(*𝑛*) and *𝐻𝑚𝑎*(*𝑛*) = *𝛩*(*𝑛*).

*𝐿 𝐿*

**Table 1**

Complexity classes T1 *,* … *,* T5 .

*𝐿𝐼*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *𝐼* is independent  simple diagram | *𝑐𝑙*(*𝐼* ) | *𝐿𝐼* | *𝐻 𝑟𝑑* | *𝐻 𝑟𝑎* | *𝐻𝑚𝑑* | *𝐻𝑚𝑎*  *𝐿𝐼* |
| T1 | Yes | = 0 |  | *𝑂*(1) | *𝑂*(1) | *𝑂*(1) | *𝑂*(1) |
| T2 | Yes | = 1 |  | *𝑂*(1) | *𝑂*(1) | *𝛩*(*𝑛*) | *𝛩*(*𝑛*) |
| T3 | Yes | ≥ 2 |  | *𝛩*(log *𝑛*) | *𝑂*(1) | *𝛩*(*𝑛*) | *𝛩*(*𝑛*) |
| T4 | No | ≠ *𝛴*∗ | | *𝛩*(*𝑛*) | *𝛩*(*𝑛*) | *𝛩*(*𝑛*) | *𝛩*(*𝑛*) |
| T5 | No | = *𝛴*∗ | | *𝛩*(*𝑛*) | *𝛩*(*𝑛*) | *𝑂*(1) | *𝑂*(1) |

*𝐿𝐼*

*𝐿𝐼*

**/ig. 3.** Diagram *𝐼*1.

1. Let |*𝐿*| *<* ∞. Then there exists natural *𝑚* such that *𝐿*(*𝑛*) = ∅ for any natural *𝑛* ≥ *𝑚*. Therefore, for each natural *𝑛* ≥ *𝑚*, *ℎ𝑚𝑑* (*𝑛*) = 0 and

*𝐿*

**/ig. 4.** Diagram *𝐼*2.

*ℎ𝑚𝑎*(*𝑛*) = 0. Thus, *𝐻𝑚𝑑* (*𝑛*) = *𝑂*(1) and *𝐻𝑚𝑎*(*𝑛*) = *𝑂*(1).

*𝐿* Let *𝐿* = *𝛴*∗, *𝑛 𝐿*

*𝐿*

*𝛤* be a decision tree over

be a natural number, and

*𝐿*(*𝑛*), which consists of the root, a terminal node labeled with 1, and an

that *𝛤* solves the problem of membership for *𝐿*(*𝑛*) deterministically and edge that leaves the root and enters the terminal node. One can show has the depth equals to 0. Therefore *ℎ𝑚𝑑* (*𝑛*) = 0 and *ℎ𝑚𝑎*(*𝑛*) = 0. Thus,

*𝐿 𝐿*

*𝐻𝑚𝑑* (*𝑛*) = *𝑂*(1) and *𝐻𝑚𝑎*(*𝑛*) = *𝑂*(1). □

**/ig. 5.** Diagram *𝐼*3.

*𝐿 𝐿*

# Corollaries

In this section, we consider two corollaries of [Theorems](#_bookmark3) [1](#_bookmark3), [3](#_bookmark6), and [4](#_bookmark7).

* 1. *Joint behavior of functions 𝐻 𝑟𝑎, 𝐻 𝑟𝑑 , 𝐻𝑚𝑎, and 𝐻𝑚𝑑*

*𝐿 𝐿 𝐿 𝐿*

**/ig. 6.** Diagram *𝐼*4.

the alphabet *𝛴* is given by a f-reduced diagram *𝐼* , which generates the In this section, we assume that each regular factorial language over considered language denoted by *𝐿𝐼* . To study all possible types of joint

behavior of functions *𝐻 𝑟𝑑* , *𝐻 𝑟𝑎* , *𝐻𝑚𝑑* , and *𝐻𝑚𝑎*, we consider five classes

*𝐿𝐼*

*𝐿𝐼*

*𝐿𝐼*

*𝐿𝐼*

of regular factorial languages T1*,* … *,* T5 described in the columns 2–4 of [Table](#_bookmark8) [1](#_bookmark8). In particular, T1 consists of all regular factorial languages

*𝐿𝐼* for which the diagram *𝐼* is an independent simple diagram and

*𝑐𝑙*(*𝐼* ) = 0. It is easy to show that the complexity classes T1*,* … *,* T5 are pairwise disjoint, and each regular factorial language *𝐿𝐼* belongs to one

of these classes. The behavior of functions *𝐻 𝑟𝑑* , *𝐻 𝑟𝑎* , *𝐻𝑚𝑑* , and *𝐻𝑚𝑎*

*𝐿𝐼*

*𝐿𝐼*

*𝐿𝐼*

*𝐿𝐼*

**/ig. 7.** Diagram *𝐼*5.

for languages from these classes is described in the last four columns of [Table](#_bookmark8) [1](#_bookmark8). For each class, the results considered in [Table](#_bookmark8) [1](#_bookmark8) for the

functions *𝐻 𝑟𝑑* and *𝐻 𝑟𝑎* follow directly from [Theorems](#_bookmark3) [1](#_bookmark3) and [3](#_bookmark6).

*𝐿𝐼 𝐿𝐼*

We now consider the behavior of the functions *𝐻𝑚𝑑* and *𝐻𝑚𝑎* for

Denote by *𝐼*2 the diagram over the alphabet *𝐸* depicted in [Fig.](#_bookmark10) [4](#_bookmark10).

each of the classes T *,* … *,* T . Let *𝐼* = (*𝐺, 𝑞 , 𝑄*)

*𝐿𝐼*

*𝐿𝐼*

One can show that *𝐼*2 is an independent simple f-reduced diagram and

1 5 0

be a f-reduced diagram

over the alphabet *𝛴*, which generates a regular factorial language.

Let *𝐿𝐼* ∈ T1. Since *𝑐𝑙*(*𝐼* ) = 0, *𝐺* is a directed acyclic graph, and the

language *𝐿𝐼* is finite. Using [Theorem](#_bookmark7) [4](#_bookmark7) we obtain *𝐻𝑚𝑑* (*𝑛*) = *𝑂*(1) and

*𝑐𝑙*(*𝐼*2) = 1. This diagram generates the language *𝐿𝐼*2 = {0*𝑖* ∶ *𝑖* ∈ *𝜔*},

which is factorial. Therefore *𝐿𝐼*2 ∈ T2.

Denote by *𝐼*3 the diagram over the alphabet *𝐸* depicted in [Fig.](#_bookmark11) [5](#_bookmark11).

*𝐻𝑚𝑎*(*𝑛*) = *𝑂*(1).

*𝐼*

*𝐿*

*𝐿𝐼*

One can show that *𝐼*3

is an independent simple f-reduced diagram and

Let *𝐿𝐼* ∈ T2. Since *𝑐𝑙*(*𝐼* ) = 1, *𝐺* is a graph containing a cycle, and

the language *𝐿𝐼* is infinite. By Lemma 4.2 [[1](#_bookmark23)], |*𝐿𝐼* (*𝑛*)| = *𝑂*(1). Therefore

*𝐿𝐼* ≠ *𝛴*∗. Using [Theorem](#_bookmark7) [4](#_bookmark7) we obtain *𝐻𝑚𝑑* (*𝑛*) = *𝛩*(*𝑛*) and *𝐻𝑚𝑎*(*𝑛*) = *𝛩*(*𝑛*).

*𝑐𝑙*(*𝐼*1) = 2. This diagram generates the language *𝐿𝐼*3 = {0*𝑖*1*𝑗* ∶ *𝑖, 𝑗* ∈ *𝜔*},

which is factorial. Therefore *𝐿𝐼*3 ∈ T3.

Denote by *𝐼* the diagram over the alphabet *𝐸* depicted in [Fig.](#_bookmark13) [6](#_bookmark13).

*𝐿𝐼*

*𝐿𝐼* 4

Let *𝐿𝐼* ∈ T3. Since *𝑐𝑙*(*𝐼* ) ≥ 2, *𝐺* is a graph containing a cycle, and the

language *𝐿𝐼* is infinite. By Lemma 4.2 [[1](#_bookmark23)], |*𝐿𝐼* (*𝑛*)| = *𝑂*(*𝑛𝑐𝑙*(*𝐼*)). Therefore

*𝐿* ≠ *𝛴*∗. Using [Theorem](#_bookmark7) [4](#_bookmark7) we obtain *𝐻𝑚𝑑* (*𝑛*) = *𝛩*(*𝑛*) and *𝐻𝑚𝑎*(*𝑛*) = *𝛩*(*𝑛*).

One can show that *𝐼*4 is a dependent simple f-reduced diagram generat- ing the language *𝐿𝐼*4 = {0*𝑖*1*𝑗* 0*𝑘* ∶ *𝑖, 𝑘* ∈ *𝜔, 𝑗* ∈ {0*,* 1}}, which is factorial.

∗

*𝐼 𝐿𝐼*

*𝐿𝐼*

It is clear that *𝐿𝐼*4 ≠ *𝐸* . Therefore *𝐿𝐼*4 ∈ T4.

Let *𝐿𝐼* ∈ T4. Since *𝐼* is not an independent simple diagram, *𝐺* is a

graph containing a cycle, and the language *𝐿𝐼* is infinite. We know that

*𝐿𝐼* ≠ *𝛴*∗. Using [Theorem](#_bookmark7) [4](#_bookmark7) we obtain *𝐻𝑚𝑑* (*𝑛*) = *𝛩*(*𝑛*) and *𝐻𝑚𝑎*(*𝑛*) = *𝛩*(*𝑛*).

Denote by *𝐼*5 the diagram over the alphabet *𝐸* depicted in [Fig.](#_bookmark14) [7](#_bookmark14). One can show that *𝐼*5 is a f-reduced diagram that is not simple. This

∗

∗ *𝐿𝐼*

*𝐿𝐼*

*𝑚𝑑*

diagram generates the language *𝐿𝐼*5 = *𝐸* , which is factorial. It is clear

Let *𝐿𝐼* ∈ T5. Then *𝐿𝐼* = *𝛴* . Using [Theorem](#_bookmark7) [4](#_bookmark7) we obtain *𝐻𝐿* (*𝑛*) =

*𝑂*(1) and *𝐻𝑚𝑎*(*𝑛*) = *𝑂*(1). *𝐼*

*𝐼*

*𝐿*

We now show that the classes T1*, ,* T5 are nonempty. For simplic- ity, we assume that *𝛴* = *𝐸*, where *𝐸* = {0*,* 1}. It is easy to generalize the considered examples to the case of an arbitrary finite alphabet *𝛴*

…

is labeled with the symbol +, and all nodes are final. with at least two letters. In the examples of diagrams, the starting node

Denote by *𝐼*1 the diagram over the alphabet *𝐸* depicted in [Fig.](#_bookmark9) [3](#_bookmark9).

that *𝐿𝐼*5 = *𝐸*∗. Therefore *𝐿𝐼*5 ∈ T5.

A regular factorial language *𝐿* can have different f-reduced dia- grams, which generate it. However, for each of such diagrams *𝐼* , the language *𝐿𝐼* = *𝐿* will belong to the same complexity class. Let us assume the contrary: there exist a regular factorial language *𝐿* and two f-reduced diagrams *𝐼*1 and *𝐼*2, which generate it and for which languages *𝐿𝐼*1 and *𝐿𝐼*2 belong to different complexity classes. Then,

One can show that *𝐼*1 is an independent simple f-reduced diagram and

for some pair *𝑏𝑐* ∈ {*𝑟𝑑, 𝑟𝑎, 𝑚𝑑, 𝑚𝑎*}, the functions *𝐻 𝑏𝑐*

*𝐿𝐼*1

and *𝐻 𝑏𝑐*

*𝐿𝐼*2

have

*𝑐𝑙*(*𝐼* ) = 0. This diagram generates the language *𝐿*

= {*𝜆,* 0}, which is

different behavior, but this is impossible since *𝐻 𝑏𝑐* (*𝑛*) = *𝐻 𝑏𝑐* (*𝑛*) for

1

factorial. Therefore *𝐿𝐼*1 ∈ T1.

*𝐼*1

any natural *𝑛*.

*𝐿𝐼*1

*𝐿𝐼*2



**/ig. 8.** Diagram *𝐼* (0).

* 1. *Languages over alphabet* {0*,* 1} *given by one forbidden word*

Let *𝐸* = {0*,* 1}, *𝛼* ∈ *𝐸*∗, and *𝛼* ≠ *𝜆*. We denote by *𝐿*(*𝛼*) the language over the alphabet *𝐸*, which consists of all words from *𝐸*∗ that do not contain *𝛼* as a factor. This is a regular factorial language with

*𝑀𝐹* (*𝐿*(*𝛼*)) = {*𝛼*}. The following theorem indicates for each nonempty word *𝛼* ∈ *𝐸*∗ the complexity class T*𝑖* to which the language *𝐿*(*𝛼*)

belongs.

**Theorem 5.** *Let 𝛼* ∈ *𝐸*∗ *and 𝛼* ≠ *𝜆.*

* + 1. *If 𝛼* ∈ {0*,* 1}*, then 𝐿*(*𝛼*) ∈ T2*.*
    2. *If 𝛼* ∈ {01*,* 10}*, then 𝐿*(*𝛼*) ∈ T3*.*
    3. *If 𝛼* ∉ {0*,* 1*,* 01*,* 10}*, then 𝐿*(*𝛼*) ∈ T4*.*

We now describe a f-reduced diagram *𝐼* (*𝛼*) that generates the lan- guage *𝐿*(*𝛼*) for a nonempty word *𝛼* ∈ *𝐸*∗. Let *𝛼* = *𝑎*1 ⋯ *𝑎𝑛*, *𝛼*0 = *𝜆*, and

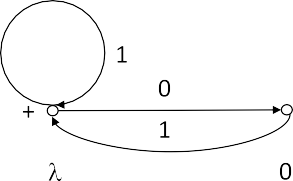
*𝛼𝑖* = *𝑎*1 ⋯ *𝑎𝑖* for *𝑖* = 1*,* … *, 𝑛* − 1. The set *𝑃* (*𝛼*) = {*𝛼*0*, 𝛼*1*,* … *, 𝛼𝑛*−1} is the

set of all proper prefixes of the word *𝛼*. Then *𝐼* (*𝛼*) = (*𝐺, 𝑞*0*, 𝑄*), where the set of nodes of the graph *𝐺* is equal to *𝑃* (*𝛼*), *𝑞*0 = *𝛼*0, and *𝑄* = *𝑃* (*𝛼*). For *𝑖* = 0*,* … *, 𝑛* − 2, an edge leaves the node *𝛼𝑖* and enters the node *𝛼𝑖*+1. This edge is labeled with the letter *𝑎𝑖*+1. For *𝑖* = 0*,* … *, 𝑛* − 1, an edge leaves the node *𝛼𝑖* and enters the node *𝛼𝑗* ∈ *𝑃* (*𝛼*) such that *𝛼𝑗* is the longest suffix of the word *𝛼𝑖𝑎̄𝑖*+1, where *𝑎̄𝑖*+1 = 0 if *𝑎𝑖*+1 = 1 and *𝑎̄𝑖*+1 = 1 if *𝑎𝑖*+1 = 0. This edge is labeled with the letter *𝑎̄𝑖*+1. It is easy to show that *𝐼* (*𝛼*) is a f-reduced diagram over the alphabet *𝐸*. From Theorem 10 [[7](#_bookmark29)] it follows that the diagram *𝐼* (*𝛼*) generates the language *𝐿*(*𝛼*).

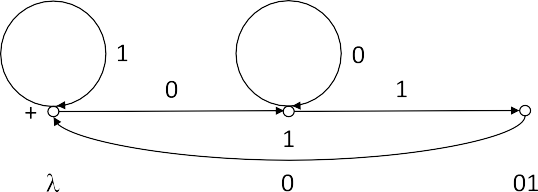
Let *𝛼* ∈ *𝐸*∗ ⧵ {*𝜆*} and *𝛼* = *𝑎*1 ⋯ *𝑎𝑛*. We denote by *𝛼̄* the word *𝑎̄*1 ⋯ *𝑎̄𝑛*.

It is easy to prove the following statement.

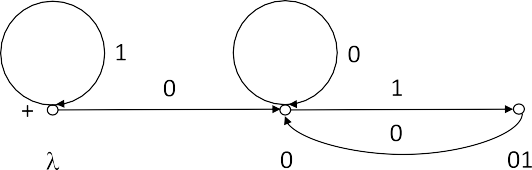
**/ig. 9.** Diagram *𝐼* (01).



**/ig. 10.** Diagram *𝐼* (00).



**/ig. 11.** Diagram *𝐼* (010).



**Lemma 6.** *Let 𝛼* ∈ *𝐸*∗ *and 𝛼* ≠ *𝜆. Then 𝐻 𝑏𝑐*

*𝐿*(*𝛼̄*)

*𝐿*(*𝛼*)

(*𝑛*) = *𝐻 𝑏𝑐*

(*𝑛*) *for any pair*

**/ig. 12.** Diagram *𝐼* (011).

*𝑏𝑐* ∈ {*𝑟𝑑, 𝑟𝑎, 𝑚𝑑, 𝑚𝑎*} *and any natural 𝑛.*

**Lemma 7.** *Let 𝛼* ∈ *𝐸*∗ ⧵ {*𝜆*}*, 𝛽* ∈ *𝐸*∗*, and 𝐿*(*𝛼*) ∈ T4*. Then 𝐿*(*𝛼𝛽*) ∈ T4*.*

**Proof.** Since *𝐿*(*𝛼*) ∈ T4, *𝐻 𝑟𝑑* (*𝑛*) = *𝛩*(*𝑛*) and *𝐻 𝑟𝑎* (*𝑛*) = *𝛩*(*𝑛*). One can

1. The diagram *𝐼* (00) is depicted in [Fig.](#_bookmark19) [10](#_bookmark19). This is not a simple diagram. It is clear that *𝐿*(00) ≠ *𝐸*∗. Therefore *𝐿*(00) ∈ T4. By [Lemma](#_bookmark17) [6](#_bookmark17),

*𝐿*(11) ∈ T4. Using [Lemma](#_bookmark22) [7](#_bookmark22) we obtain *𝐿*(000)*, 𝐿*(001)*, 𝐿*(110)*, 𝐿*(111) ∈

show that *𝐿*(*𝛼*) *⊆ 𝐿*(*𝛼𝛽*)

*𝐿*(*𝛼*)

*𝐿*(*𝛼*)

T4.

. Using this fact it is not difficult to prove that

The diagram *𝐼* (010) is depicted in [Fig.](#_bookmark20) [11](#_bookmark20). This is not a simple

*𝑟𝑑*

*𝐻*

(*𝑛*) ≤ *𝐻*

*𝐿*(*𝛼*)

*𝑟𝑑*

*𝐿*(*𝛼𝛽*)

(*𝑛*) and *𝐻 𝑟𝑎*

*𝑟𝑎*

*𝐿*(*𝛼𝛽*)

(*𝑛*) ≤ *𝐻*

(*𝑛*) for any natural *𝑛*. From

*𝑟𝑑*

diagram. It is clear that *𝐿*(010) ≠ *𝐸*∗. Therefore *𝐿*(010) ∈ T4. By

here and from [Theorems](#_bookmark3) [1](#_bookmark3) and [3](#_bookmark6) it follows that *𝐻𝐿*(*𝛼𝛽*)(*𝑛*) = *𝛩*(*𝑛*) and

*𝐿*(*𝛼*)

[Lemma](#_bookmark17) [6](#_bookmark17), *𝐿*(101) ∈ T4.

*𝑟𝑎*

*𝐻*

*𝐿*(*𝛼𝛽*)

(*𝑛*) = *𝛩*(*𝑛*).

∗

The diagram *𝐼* (011) is depicted in [Fig.](#_bookmark21) [12](#_bookmark21). This is not a simple

Since *𝛼𝛽* ∉ *𝐿*(*𝛼𝛽*), *𝐿*(*𝛼𝛽*) ≠ *𝐸* . The diagram *𝐼* (*𝛼𝛽*) contains at least

one circle formed by the edge that leaves and enters the node *𝜆* and is labeled with the letter *𝑎̄*1, where *𝑎*1 is the first letter of the word

diagram. It is clear that *𝐿*(011) ≠ *𝐸*∗. Therefore *𝐿*(011) ∈ T4. By [Lemma](#_bookmark17) [6](#_bookmark17), *𝐿*(100) ∈ T4.

We proved that, for any word *𝛼* ∈ *𝐸*∗ of the length three, *𝐿*(*𝛼*) ∈ T4.

*𝛼*. Therefore the language *𝐿*(*𝛼𝛽*) is infinite. By [Theorem](#_bookmark7) [4](#_bookmark7), *𝐻𝑚𝑑* (*𝑛*) =

*𝛩*(*𝑛*) and *𝐻𝑚𝑎* (*𝑛*) = *𝛩*(*𝑛*). Thus, *𝐿*(*𝛼𝛽*) ∈ T4. □

*𝐿*(*𝛼𝛽*)

*𝐿*(*𝛼𝛽*)

**Proof of** [**Theorem**](#_bookmark16)[**5**](#_bookmark16)**.** In each figure depicting a diagram *𝐼* (*𝛼*), *𝛼* ∈

*𝐸*∗ ⧵ {*𝜆*}, we label each node with a corresponding prefix of the word

*𝛼*.

1. The diagram *𝐼* (0) is depicted in [Fig.](#_bookmark15) [8](#_bookmark15). This is an independent simple f-reduced diagram with *𝑐𝑙*(*𝐼* (0)) = 1. Therefore *𝐿*(0) ∈ T2. By [Lemma](#_bookmark17) [6](#_bookmark17), *𝐿*(1) ∈ T2.
2. The diagram *𝐼* (01) is depicted in [Fig.](#_bookmark18) [9](#_bookmark18). This is an independent simple f-reduced diagram with *𝑐𝑙*(*𝐼* (01)) = 2. Therefore *𝐿*(01) ∈ T3. By [Lemma](#_bookmark17) [6](#_bookmark17), *𝐿*(10) ∈ T3.

Using [Lemma](#_bookmark22) [7](#_bookmark22) we obtain that, for any word *𝛼* ∈ *𝐸*∗ of the length greater than or equal to four, *𝐿*(*𝛼*) ∈ T4. □

# Declaration of competing interest

The authors declare that they have no known competing finan- cial interests or personal relationships that could have appeared to influence the work reported in this paper.

# Acknowledgments

Research reported in this publication was supported by King Abdul- lah University of Science and Technology (KAUST), Saudi Arabia. The

author is grateful to the anonymous reviewers for useful remarks and suggestions.

# References

1. [Moshkov M. Decision trees for regular language word recognition. Fund Inform](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb1) [2000;41(4):449–61.](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb1)
2. [Moshkov M. Complexity of deterministic and nondeterministic decision trees](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb2) [for regular language word recognition. In: Bozapalidis S, editor. Proceedings of](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb2) [the 3rd international conference developments in language theory, DLT 1997,](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb2) [Thessaloniki, Greece, July 20–23, 1997. Aristotle University of Thessaloniki; 1997,](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb2)

[p. 343–9.](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb2)

1. [D’Alessandro F, Intrigila B, Varricchio S. On the structure of the counting function](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb3) [of sparse context-free languages. Theoret Comput Sci 2006;356(1–2):104–17.](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb3)
2. [Shur AM. Combinatorial complexity of regular languages. In: Hirsch EA,](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb4) [Razborov AA, Semenov AL, Slissenko A, editors. Computer science - theory and](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb4) [applications, third international computer science symposium in Russia, CSR 2008,](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb4) [Moscow, Russia, June 7-12, 2008, proceedings. Lecture notes in computer science,](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb4) [vol. 5010, Springer; 2008, p. 289–301.](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb4)
3. Moshkov M. Decision trees for binary subword-closed languages. 2022, CoRR, [arXiv:2201.01493](http://arxiv.org/abs/2201.01493).
4. [Haines LH. On free monoids partially ordered by embedding. J Combin Theory](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb6) [1969;6:94–8.](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb6)
5. [Crochemore M, Mignosi F, Restivo A. Automata and forbidden words. Inf Process](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb7) [Lett 1998;67(3):111–7.](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb7)
6. [Yu S. Regular languages. In: Rozenberg G, Salomaa A, editors. Handbook of formal](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb8) [languages, Volume 1: word, language, grammar. Springer; 1997, p. 41–110.](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb8)
7. [Hopcroft JE, Motwani R, Ullman JD. Introduction to automata theory, languages,](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb9) [and computation. Pearson international edition, third ed.. Addison-Wesley; 2007.](http://refhub.elsevier.com/S2590-0056(22)00051-0/sb9)