Journal of Intelligent Learning Systems and Applications, 2025, 17(3), 193-210
https://www.scirp.org/journal/jilsa
ISSN Online: 2150-8410
ISSN Print: 2150-8402

Recognizing Properties of Decision Rule Systems Using Deterministic and Nondeterministic Decision Trees

Kerven Durdymyradov, Mikhail Moshkov

Computer, Electrical and Mathematical Sciences & Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia
Email: kerven.durdymyradov@kaust.edu.sa, mikhail.moshkov@kaust.edu.sa

How to cite this paper: Durdymyradov, K. and Moshkov, M. (2025) Recognizing Properties of Decision Rule Systems Using Deterministic and Nondeterministic Decision Trees. *Journal of Intelligent Learning Systems and Applications*, **17**, 193-210.
https://doi.org/10.4236/jilsa.2025.173013

Received: July 8, 2025
Accepted: August 16, 2025
Published: August 19, 2025

Abstract

We consider various tasks of recognizing properties of DRSs (Decision Rule Systems) in this paper. As solution algorithms, DDTs (Deterministic Decision Trees) and NDTs (Nondeterministic Decision Trees) are used. An NDT can be considered as a representation of a DRS that satisfies the conditions of the considered task and covers all potential inputs. It has been shown that the minimum depth of a DDT solving the task does not exceed the square of the minimum depth of an NDT. The growth of the minimum number of nodes in DDTs and NDTs can be exponential with the size of the original DRSs. Therefore, in the general case, it is better to simulate the behavior of the DT (Decision Tree) on the given tuple of feature values rather than building the entire tree. We propose a greedy algorithm for such modeling and study its efficiency for a class of tasks of recognizing properties of DRSs. The obtained results may be of interest for data analysis in which both DRSs and DTs are intensively studied. In particular, these results make one think about the possibilities of transforming DRSs into DTs.

Keywords

Deterministic Decision Tree, Nondeterministic Decision Tree, Decision Rule System

1. Introduction

DRSs (Decision Rule Systems) [1]-[5] and DTs (Decision Trees) [6]-[9] serve as common models for structuring and representing the knowledge. They act as classifiers that predict solutions for unseen instances, and operate as algorithms addressing tasks in combinatorial optimization, fault diagnosis, and related domains. Among classification and knowledge representation techniques, both DTs and DRSs stand out for their high level of interpretability [10]. Exploring the relationships between DTs and DRSs represents an important research direction in computer science.

In this paper, we study DDTs (Deterministic Decision Trees) and NDTs (Nondeterministic Decision Trees) for solving tasks related to recognizing properties of DRSs. Note that an NDT solving such a task, can be considered as a representation of a DRS that is true for this task and covers the whole set of inputs.

Compared to classical decision tree models, which rely on information-theoretic criteria (e.g., entropy or Gini index) and typically require labeled data, our approach uses a syntactic method that operates directly on a known DRS without the need for input data. Moreover, our framework preserves full transparency of decision paths, thereby maintaining a high level of interpretability.

In the present paper, we continue to develop a syntactic approach to solving the tasks under consideration, proposed in [11] [12]. It assumes that the DRS is known, but the input data is unavailable. The results of previous studies in this area are summarized in the book [13]. Compared to [13], this paper, which extends two conference papers [14] [15], considers a wider range of tasks related to recognizing the properties of DRSs.

Let S be a finite DRS represented in the form

$$\left(f_{i_1}=d_1\right)\wedge\cdots\wedge\left(f_{i_m}=d_m\right)\to d,$$

where $f_{i_1},\cdots,f_{i_m}$ are features, $d_1,\cdots,d_m$ represent feature values selected from the set $E_k=\{0,\cdots,k-1\}$, where $k\geq 2$, and d denotes a nonnegative integer serving as the decision.

Let $f_1,\cdots,f_n$ be all features from the DRs (Decision Rules) in the DRS S, and $\bar{b}=\left(b_1,\cdots,b_n\right)\in E_k^n$ be a tuple of values of the features $f_1,\cdots,f_n$. The considered DR is called applicable for the tuple $\bar{b}$ when its left part is true for this tuple, *i.e.*, when $b_{i_1}=d_1,\cdots,b_{i_m}=d_m$.

A solution map is a function φ that associates a tuple $\bar{b}$ of values of features from S with a corresponding value $\varphi\left(\bar{b}\right)$, referred to as the solution. Let us consider examples of solution maps (more examples can be found in Sect. 2.1):

- $\varphi\left(\bar{b}\right)=1$ if there exists a DR from S that is applicable for the tuple $\bar{b}$ and $\varphi\left(\bar{b}\right)=0$ otherwise.
- $\varphi\left(\bar{b}\right)$ is the set of all DRs in S which are applicable for the tuple $\bar{b}$.
- $\varphi\left(\bar{b}\right)$ is the set of decisions from the right part of DRs in S which are applicable for the tuple $\bar{b}$.

We study the task of finding $\varphi\left(\bar{b}\right)$, where $\bar{b}\in E_k^n$ is a tuple of feature values from S. We denote this task as $T\left(\varphi,S,k\right)$. It is important to note that while solving the task $T\left(\varphi,S,k\right)$, the tuple $\bar{b}$ is not directly accessible. To determine the value of a feature, we need to find it for the given input, which can be an expensive procedure. To minimize the number of queries related to feature values, we consider DDTs and NDTs solving the task $T\left(\varphi,S,k\right)$.

We pay special attention to the task of All Applicable Rules (AAR task) for which $\varphi(\bar{b})$ is the set of all DRs from S that are applicable for the tuple $\bar{b}$. For this task, we consider DRS examples where DDTs and NDTs can solve the AAR task with minimum depths much smaller than the number of distinct features in S. In such a situation, the use of DTs seems appropriate. We also consider examples of DRSs, in which the minimal number of nodes in DDTs and NDTs solving the AAR task grows exponentially with the size of the DRSs. Therefore, in general, instead of constructing the entire DT, we should model its operation on a given tuple $\bar{b}$ using a sufficiently efficient algorithm. Note that similar examples were considered in the book [13].

First, we investigate how the minimum depths of DDTs and NDTs relate when solving the task $T(\varphi,S,k)$. We use $h^d(\varphi,S,k)$ to represent the minimal depth of a DDT solving this task, and $h^a(\varphi,S,k)$ for the minimal depth of an NDT. We show the inequalities $h^a(\varphi,S,k) \le h^d(\varphi,S,k) \le h^a(\varphi,S,k)^2$. These bounds were obtained in the conference paper [15].

To obtain the upper bound for $h^d(\varphi,S,k)$, we analyze an NDT $\mathcal{G}$ that solves the task $T(\varphi,S,k)$ with depth $h^a(\varphi,S,k)$. Using this tree, we describe the operation of a DDT G, solving the same task for a given tuple $\bar{b} \in E_k^n$, has the depth at most $h^a(\varphi,S,k)^2$. It should be emphasized that this description of the DDT cannot be regarded as an efficient algorithm, since $\mathcal{G}$ may possess a substantial number of nodes. Note also that a similar upper bound was derived in [13] for functions of k-valued logic, $k \ge 2$ and in [16]-[18] for Boolean functions (see [19] for details).

We additionally introduce a greedy algorithm $\mathcal{U}$ which, for a given DRS S and a tuple $\bar{b} \in E_k^n$ of feature values, simulate the operation of a DDT solving the task $T(\varphi,S,k)$ on $\bar{b}$. For this algorithm, we need to have an uncertainty measure γ for the task $T(\varphi,S,k)$, which is defined on the set of equation systems of the form $\alpha = \{f_{i_1} = d_1, \cdots, f_{i_m} = d_m\}$, such that $f_{i_1}, \cdots, f_{i_m}$ are pairwise different features from the set $\{f_1, \cdots, f_n\}$ and $d_1, \cdots, d_m \in E_k$. The system of equations α describes information already obtained by the DT. At the beginning of the tree work, $\alpha = \varnothing$. The DT will stop when $\gamma(\alpha) = 0$. We prove that the depth of the considered DDT is at most $h^a(\varphi,S,k)\ln\gamma(\varnothing)+1$.

It should be noted that this algorithm has some similarities with the one presented in Sect. 4.2 of [3]. However, the method in [3] operates on a decision table T instead of a DRS S, and uses very different measures of uncertainty, in particular, the value $P(T)$, which counts unordered pairs of rows in T having distinct decisions.

The uncertainty measure should satisfy some additional requirements, and the search for appropriate uncertainty measures is a nontrivial task. In the conference paper [14], we found a suitable uncertainty measure for the AAR task and proved all the results related to the greedy algorithm only for the AAR task. In the current paper, we extend these results to the general case of an arbitrary task $T(\varphi,S,k)$ and an appropriate uncertainty measure γ for it.

We generalized essentially the results obtained in [14] by studying the task of recognition of applicable special subsets of the DRS S. Let W be a nonempty family of nonempty subsets of S that are called special subsets of S. We assume that, for each subset w from W, there exists a tuple $\bar{b} \in E_k^n$ for which all DRs from w are applicable. In this case, we will say that the subset w of DRs is applicable for the tuple $\bar{b}$. We consider the following task $T(\varphi_W, S, k)$: for a given tuple $\bar{b} \in E_k^n$, we should find all subsets $w \in W$ that are applicable for the tuple $\bar{b}$. For this task, we found an appropriate uncertainty measure γ_W such that $\gamma_W(\varnothing) \le |W| n$ and showed that the algorithm $\mathcal{U}$ has polynomial time complexity depending of the sizes of the DRS S and the family W. The depth of a DDT, the operation of which is modeled by the algorithm $\mathcal{U}$, does not exceed $h^a(\varphi_W, S, k)(\ln|W| + \ln n) + 1$.

The results obtained in this paper may be of interest for data analysis, in which both DRSs and DTs are intensively studied. In particular, these results make one think about the possibilities of transforming DRSs into DTs.

The structure of the paper is as follows. Section 2 provides the key definitions and notation, closely aligned with those in [13]. Section 3 explores both the possibilities and the constraints of using DTs. In Section 4, we analyze and compare the minimum depths of DDTs and NDTs. Section 5 focuses on a greedy algorithm designed to simulate the operation of a DDT. Section 6 addresses the task of identifying applicable special subsets of a DRS. Finally, Section 7 offers concluding remarks.

2. Definitions

This section introduces the fundamental definitions and notations associated with DRSs and DTs.

2.1. DRSs—Decision Rule Systems

Let $\mathbb{N}_0 = \{0, 1, 2, \cdots\}$, and define the set of *features* as $F = \{f_i : i \in \mathbb{N}_0\}$. For any $k \in \mathbb{N}_0 \setminus \{0, 1\}$, let $E_k = \{0, 1, \cdots, k-1\}$.

Definition 1 *We use* ES *to denote the set of systems of equations of the following form*:

$$\{f_{i_1} = b_1, \cdots, f_{i_m} = b_m\},$$

where $m \in \mathbb{N}_0$, $f_{i_1}, \cdots, f_{i_m} \in F$ and $b_1, \cdots, b_m \in \mathbb{N}_0$. The system will be *inconsistent* if there exist $t, p \in \{1, \cdots, m\}$ such that $t \ne p$, $i_t = i_p$, and $b_t \ne b_p$. Otherwise, the system will be called *consistent*.

Definition 2 *A DR* (*Decision Rule*) *is defined as an expression of the following form*:

$$(f_{i_1} = b_1) \wedge \cdots \wedge (f_{i_m} = b_m) \to d,$$

where $m \in \mathbb{N}_0 \setminus \{0\}$, $f_{i_1}, \cdots, f_{i_m}$ are pairwise different features from F, and $b_1, \cdots, b_m, d \in \mathbb{N}_0$.

We denote this DR by r. The value d is referred to as the *decision* of the decision rule r. The integer m is called the *length* of r and is denoted by $l(r)$. We define $F(r)=\{f_{i_1},\cdots,f_{i_m}\}$ as the set of features in r and $K(r)=\{f_{i_1}=b_1,\cdots,f_{i_m}=b_m\}$. Let $V(r)$ denote the minimum integer $k\in\mathbb{N}_0\setminus\{0,1\}$ for which $\{b_1,\cdots,b_m\}\subseteq E_k$.

Definition 3 *A Decision Rule System (DRS) S is defined as a finite, nonempty set of DRs.*

Denote $F(S)=\bigcup_{r\in S}F(r)$, $n(S)=|F(S)|$, $L(S)=\sum_{r\in S}l(r)$, and $V(S)=\max\{V(r):r\in S\}$. Let $n(S)=n$ and $F(S)=\{f_{j_1},\cdots,f_{j_n}\}$, where $j_1<\cdots<j_n$. For $\bar{b}=(b_1,\cdots,b_n)\in\mathbb{N}_0^n$, denote $K(S,\bar{b})=\{f_{j_1}=b_1,\cdots,f_{j_n}=b_n\}$.

Definition 4 *A DR r from S is called applicable for a tuple $\bar{b}\in\mathbb{N}_0^{n(S)}$ if $K(r)\subseteq K(S,\bar{b})$.*

Definition 5 *Consider a DRS S and an integer $k\geq V(S)$. A solution map for the pair (S,k) is a mapping φ that assigns to each tuple $\bar{b}\in E_k^{n(S)}$ a value $\varphi(\bar{b})$ from some set. This value is interpreted as a solution. The map φ will be called degenerate if it is constant on the set $E_k^{n(S)}$ and nondegenerate otherwise.*

Let S be a DRS, $k\geq V(S)$, and $\bar{b}\in E_k^{n(S)}$. We now consider a number of examples of solution maps φ for the pair (S,k):

- $\varphi(\bar{b})=1$ if there exists a DR from S that is applicable for the tuple $\bar{b}$ and $\varphi(\bar{b})=0$ otherwise.
- $\varphi(\bar{b})$ is the set of all DRs from S that are applicable for the tuple $\bar{b}$.
- $\varphi(\bar{b})$ is the number of DRs from S that are applicable for the tuple $\bar{b}$.
- $\varphi(\bar{b})$ is the set of decisions of DRs from S that are applicable for the tuple $\bar{b}$.
- $\varphi(\bar{b})$ is the minimum number $d\in\mathbb{N}_0$ such that the number of DRs from S that are applicable for the tuple $\bar{b}$ and have d as the decision is maximum.

Let S be a DRS with $n(S)=n$ and $k\geq V(S)$.

Definition 6 *Let φ be a solution map for the pair (S,k). Task $T(\varphi,S,k)$ is defined as follows: for a given tuple $\bar{b}\in E_k^n$, it is required to find the solution $\varphi(\bar{b})$.*

When addressing the task $T(\varphi,S,k)$, it is important to note that we do not have direct access to the tuple $\bar{b}$. To determine the value of a feature $f_i\in F(S)$, it is necessary to compute it for the given input, which can be an expensive procedure. To minimize the number of queries related to feature values, we consider DTs for solving the task $T(\varphi,S,k)$.

2.2. DTs—Decision Trees

A *finite directed tree with a root* is a finite directed tree in which exactly one node has no entering edges. This node is referred to as the *root*. Nodes that do not have any leaving edges are called *leaf* nodes, while those that are neither root nor leaf are called *internal* nodes. A *full path* in such a tree is a sequence $\mathcal{P}=v_1,d_1,\cdots,v_m,d_m,v_{m+1}$ consisting of alternating nodes and edges, where v_1 is the root, v_{m+1} is a leaf, and for each $i=1,\cdots,m$, the edge d_i leaves from node v_i and enters to node v_{i+1}.

Consider a DRS S, $n(S)=n$, $k \ge V(S)$, and φ be a solution map for the pair (S,k).

Definition 7 *We define a DT (Decision Tree) over the task* $T(\varphi,S,k)$ *as a finite marked directed tree with root* $\mathcal{G}$, *containing at least two nodes, and satisfying the conditions below:*

- The root and edges leaving the root are not marked.
- In the tree $\mathcal{G}$, every internal node is marked with a feature from $F(S)$, and each leaving edge marked with a numbers from E_k.
- Each leaf node of the tree $\mathcal{G}$ is marked with a solution from the set $\{\varphi(\bar{b}):\bar{b}\in E_k^n\}$.

Definition 8 *A DT for the task* $T(\varphi,S,k)$ *is a deterministic if its root has exactly one leaving edge, and the leaving edges of every internal node are marked with pairwise distinct values.*

Suppose $\mathcal{G}$ is a DT over the task $T(\varphi,S,k)$. We denote by $FP(\mathcal{G})$ the set of full paths in the DT $\mathcal{G}$. Consider a full path $\mathcal{P}=v_1,d_1,\cdots,v_m,d_m,v_{m+1}$ from $\mathcal{G}$. To this path, we associate an equation system $K(\mathcal{P})\in ES$. If $m=1$ and $\mathcal{P}=v_1,d_1,v_2$, then $K(\mathcal{P})=\varnothing$. Let $m\ge 2$ and, for $j=2,\cdots,m$, the node v_j be marked with the feature f_{i_j} and the corresponding edge d_j be marked with the value $b_j\in E_k$. In this case, we set $K(\mathcal{P})=\{f_{i_2}=b_2,\cdots,f_{i_m}=b_m\}$. We denote by $\tau(\mathcal{P})$ the solution associated with the leaf node v_{m+1}.

Let $\mathcal{G}$ be a DT over $T(\varphi,S,k)$, $\bar{b}\in E_k^n$, and $\mathcal{P}\in FP(\mathcal{G})$. We say that a full path $\mathcal{P}$ *accepts the tuple* $\bar{b}$ if $K(\mathcal{P})\subseteq K(S,\bar{b})$. Furthermore, the solution $\tau(\mathcal{P})$ is called *derivable from the equation system* $K(\mathcal{P})$ if $\varphi(\bar{b})=\tau(\mathcal{P})$ for any tuple $\bar{b}\in E_k^n$ such that the path $\mathcal{P}$ accepts the tuple $\bar{b}$.

Definition 9 *It is called* $\mathcal{G}$ *solves the task* $T(\varphi,S,k)$ *nondeterministically if, for every tuple* $\bar{b}\in E_k^n$, *there exists a path* $\mathcal{P}\in FP(\mathcal{G})$, *that accepts* $\bar{b}$ *and, for every path* $\mathcal{P}\in FP(\mathcal{G})$ *such that the system of equations* $K(\mathcal{P})$ *is consistent, the solution* $\tau(\mathcal{P})$ *is derivable from* $K(\mathcal{P})$. *In this case, we also refer to* $\mathcal{G}$ *as an NDT that solves the task* $T(\varphi,S,k)$.

Suppose $\mathcal{G}$ be a NDT solving the task $T(\varphi,S,k)$, $\bar{b}\in E_k^n$, and $\mathcal{P}\in FP(\mathcal{G})$. If the path $\mathcal{P}$ accepts the tuple $\bar{b}$, then $K(\mathcal{P})$ is a consistent system of equations, the solution $\tau(\mathcal{P})$ is derivable from $K(\mathcal{P})$, and $\varphi(\bar{b})=\tau(\mathcal{P})$.

Definition 10 *It is called* $\mathcal{G}$ *solves the task* $T(\varphi,S,k)$ *deterministically if* $\mathcal{G}$ *is a DDT that solves the task* $T(\varphi,S,k)$ *nondeterministically. In this case, it's called* $\mathcal{G}$ *is a DDT solving the task* $T(\varphi,S,k)$.

See Figure 1 for an example of a DDT and an NDT that solve the task described above.

Definition 11 *We define* $h(\mathcal{P})$ *as the number of internal nodes in any full path* $\mathcal{P}\in FP(\mathcal{G})$. *The value* $h(\mathcal{G})=max\{h(\mathcal{P}):\mathcal{P}\in FP(\mathcal{G})\}$ *is the depth of the DT* $\mathcal{G}$.

Suppose S is a DRS, $k\ge V(S)$ and φ is a solution map for the pair (S,k).

We denote by $h^a(\varphi, S, k)$ the minimum depth of a NDT over the task $T(\varphi, S, k)$, which solves this task. We denote by $h^d(\varphi, S, k)$ the minimum depth of a DDT over the task $T(\varphi, S, k)$, which solves this task.

For example, let us consider the problem $T(\varphi_1, S_0, 2)$ where $S_0 = \{(a_1 = 0) \to 0, (a_2 = 1) \to 1\}$, φ_1 is a solution map for the pair $(S_0, 2)$: $\varphi_1(\bar{b}) = 1$ if there exists a DR from S_0 that is applicable for the tuple $\bar{b}$ and $\varphi_1(\bar{b}) = 0$ otherwise, from [15]. Then the decision trees depicted in **Figure 1** are deterministic (left) and nondeterministic (right) decision trees that solve the considered problem.

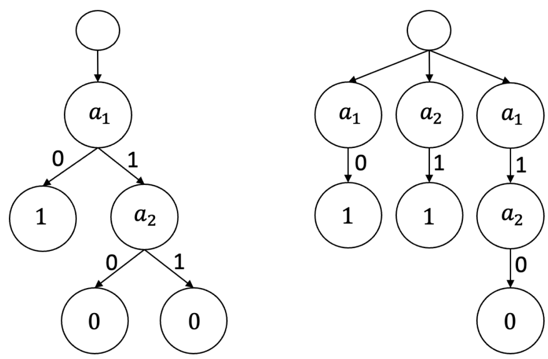

Figure 1. Deterministic and nondeterministic decision trees solving the problem $T(\varphi_1, S_0, 2)$.

3. Preliminary Discussion

Let S be a DRS, $k \geq V(S)$, and φ_{AAR} be a solution map for the pair (S, k) such that, for any $\bar{b} \in E_k^{n(S)}$, $\varphi_{AAR}(\bar{b})$ is the set of all DRs from S that are applicable for the tuple $\bar{b}$. The task $T(\varphi_{AAR}, S, k)$ is called the All Applicable Rules task. This is one of the most important tasks arising within the framework of the syntactic approach to the study of DRSs. We will study its generalizations in Sect. 6.

This section explores the possibilities and constraints of using DTs for addressing the All Applicable Rules task. Accordingly, we analyze two different sequences of DRSs. We will use the sum of lengths of DRs from S, denoted by $L(S)$, as a parameter characterizing the complexity of the DRS S.

We begin by defining a sequence of DRSs, denoted as S_m, where $m = 2, 3, \cdots$. For each S_m, the minimum depth of a DDT that solves the task $T(\varphi_{AAR}, S_m, 2)$ is significantly less than the number $n(S_m)$ of distinct features present in S_m. The number of nodes of the corresponding DT is at most $L(S_m)$ nodes. This example demonstrates the feasibility of using DTs to solve the task $T(\varphi_{AAR}, S_m, 2)$.

We then define a sequence of DRSs Q_m, for $m = 2, 3, \cdots$, such that the minimal number of nodes in both DDTs and NDTs solving the task $T(\varphi_{AAR}, Q_m, 2)$ grows exponentially with respect to the parameter $L(Q_m)$. Consequently, constructing complete DTs becomes impractical in general. As an alternative, one can simulate the operation of a DDT on a tuple of feature values with some algorithm.

Let us begin with the first sequence of DRSs. Let $m \in \mathbb{N}_0 \setminus \{0, 1\}$. A complete binary tree with depth m is a finite directed tree with root such that every non-leaf node has exactly two leaving edges, and all full paths from the root to the leaves have length m. The nodes in this tree are organized into $m+1$ layers: for each $i=0,\cdots,m$, the i th layer consists of all nodes at distance i from the root. The i th level contains exactly 2^i nodes. In total, there are 2^m-1 non-leaf nodes and 2^m leaf nodes.

Let H_m denote a marked complete binary tree of depth m, where non-leaf nodes are marked with features $f_1,\cdots,f_{2^m-1}$, and the leaves are marked with integers 1 through 2^m. Each non-leaf node has two leaving edges, marked with 0 and 1. For each $j=1,\cdots,2^m$, we define a DR r_j as follows. Consider a full path in H_m from the root to the leaf marked j, represented as $v_0,d_0,\cdots,v_{m-1},d_{m-1},v_m$, where for every $i=0,\cdots,m-1$, the node v_i marked with f_{l_i}, and the edge d_i is marked with $b_i\in E_2$. Then the DR r_j is given by:

$$\left(f_{l_0}=b_0\right)\wedge\cdots\wedge\left(f_{l_{m-1}}=b_{m-1}\right)\to j.$$

We define S_m as the set of DRs $r_1,\cdots,r_{2^m}$. It follows that $n\left(S_m\right)=2^m-1$ and $V\left(S_m\right)=2$.

We examine the task $T\left(\varphi_{AAR},S_m,2\right)$ and prove that $h^d\left(\varphi_{AAR},S_m,2\right)\le m$. To this end, we convert the tree H_m into a DDT, denoted by K_m, that corresponds to the task $T\left(\varphi_{AAR},S_m,2\right)$. This is done by introducing a new node v and an edge d directed from v to the root of H_m, leaving both v and d unmarked. Each leaf node in H_m previously marked with j (for $j=1,\cdots,2^m$) is now marked with $\left\{r_j\right\}$. The resulting DDT K_m solves the task $T\left(\varphi_{AAR},S_m,2\right)$ and has a depth of m.

Thus, for every $m\in\mathbb{N}_0\setminus\{0,1\}$, we obtained an example of a DRS S_m satisfying $n\left(S_m\right)=2^m-1$ and $L\left(S_m\right)=m2^m$. Moreover, there exists a DDT K_m that solves the task $T\left(\varphi_{AAR},S_m,2\right)$, having depth exactly m and number of nodes is 2^{m+1} nodes.

We now turn to the second sequence of DRSs. Let $m\in\mathbb{N}_0\setminus\{0,1\}$. Define Q_m as the DRS:

$$\left\{\left(f_1=0\right)\to 0,\left(f_1=1\right)\to 1,\cdots,\left(f_m=0\right)\to 0,\left(f_m=1\right)\to 1\right\}.$$

Clearly, $V\left(Q_m\right)=2$. Consider the task $T\left(\varphi_{AAR},Q_m,2\right)$. It can be shown that for any two distinct tuples $\bar{b},\bar{d}\in E_2^m$, we have $\varphi_{AAR}\left(\bar{b}\right)\ne\varphi_{AAR}\left(\bar{d}\right)$. As a result, any NDT that solves the task $T\left(\varphi_{AAR},Q_m,2\right)$ must contain at least 2^m leaf nodes.

Consequently, for every $m\in\mathbb{N}_0\setminus\{0,1\}$, we have obtained a DRS Q_m such that $n\left(Q_m\right)=m$ and $L\left(Q_m\right)=2m$. Furthermore, any decision tree, whether deterministic or nondeterministic, that solves the task $T\left(\varphi_{AAR},S_m,2\right)$ must contain at least 2^m nodes.

4. Comparative analysis of DDTs (Deterministic Decision Trees) and NDTs (Nondeterministic Decision Trees)

This section focuses on analyzing the minimum depths of DDTs and NDTs that solve the task $T(\varphi,S,k)$.

Theorem 1 *Suppose* S *be a DRS,* $k \geq V(S)$, *and* φ *be a solution map for the pair* (S,k). *Then*

$$h^a(\varphi,S,k) \leq h^d(\varphi,S,k) \leq h^a(\varphi,S,k)^2.$$

Proof. Since any DDT that solves the task $T(\varphi,S,k)$ is a NDT that solves the task $T(\varphi,S,k)$, so the inequality $h^a(\varphi,S,k) \leq h^d(\varphi,S,k)$ holds. We now show that $h^d(\varphi,S,k) \leq h^a(\varphi,S,k)^2$.

Let $n(S)=n$ and let, for definiteness, $F(S)=\{f_1,\cdots,f_n\}$. Let $\mathcal{G}$ be a NDT over $T(\varphi,S,k)$, that solves the task $T(\varphi,S,k)$, where $h(\mathcal{G})=h^a(\varphi,S,k)$. Let $FP^+(\mathcal{G})$ denote the set of full paths $\mathcal{P}$ from $FP(\mathcal{G})$ for which the corresponding equation system $K(\mathcal{P})$ is consistent. We assign a number from $\mathbb{N}_0$ to each path in $FP^+(\mathcal{G})$ to ensure all paths are distinctly identified. Two full paths $\mathcal{P}_1,\mathcal{P}_2 \in FP^+(\mathcal{G})$ are said to be *equivalent* if $\tau(\mathcal{P}_1)=\tau(\mathcal{P}_2)$. This equivalence relation partitions $FP^+(\mathcal{G})$ into equivalence classes $C_1,\cdots,C_t$.

We now prove that for any two full paths $\mathcal{P}_1,\mathcal{P}_2 \in FP^+(\mathcal{G})$ are not equivalent, then the system $K(\mathcal{P}_1)\cup K(\mathcal{P}_2)$ is inconsistent. Suppose, for contradiction, that $K(\mathcal{P}_1)\cup K(\mathcal{P}_2)$ is consistent. Then there exists a tuple $\bar{d} \in E_k^n$ such that $K(\mathcal{P}_1)\subseteq K(S,\bar{d})$ and $K(\mathcal{P}_2)\subseteq K(S,\bar{d})$. This implies that both paths accept $\bar{d}$, which leads to a contradiction because $\tau(\mathcal{P}_1)\neq\tau(\mathcal{P}_2)$.

Suppose $\bar{b}=(b_1,\cdots,b_n)\in E_k^n$. Let us describe the behavior of a DDT G with respect to the tuple $\bar{b}$, where G solves the task $T(\varphi,S,k)$. As a consequence, we derive the description of a full path $\rho(\bar{b})$ in the DT G that accepts the tuple $\bar{b}$. The set of full paths of the DT G coincides with the set $\{\rho(\bar{b}):\bar{b}\in E_k^n\}$.

Step 1:

Initialize $\Xi := FP^+(\mathcal{G})$. For every path $\mathcal{P}\in FP^+(\mathcal{G})$, define $Q(\mathcal{P}) := K(\mathcal{P})$. Proceed to Step 2.

Step 2:

This step is divided into three phases:

(a) If there exists a full path $\mathcal{P}\in\Xi$ such that $Q(\mathcal{P})$ is empty, then the DT G terminates and returns the solution $\tau(\rho(\bar{b}))=\tau(\mathcal{P})$, which will be attached to the leaf node of the path $\rho(\bar{b})$. Otherwise, proceed to (b).

(b) Set i_0 the minimum index i in $\{1,\cdots,t\}$ such that $C_i\cap\Xi\neq\varnothing$. Choose a full path $\mathcal{P}\in C_{i_0}\cap\Xi$ with the minimum number. If $C_{i_0}\cap\Xi=\Xi$, then the tree G terminates and returns the solution $\tau(\rho(\bar{b}))=\tau(\mathcal{P})$, which will be attached to the leaf node of the path $\rho(\bar{b})$. Otherwise, proceed to (c).

(c) Let $\{f_{i_1},\cdots,f_{i_m}\}$ be all features from the equations containing in $Q(\mathcal{P})$. The DT G computes values of these features and obtains the set of equations $K=\{f_{i_1}=b_{i_1},\cdots,f_{i_m}=b_{i_m}\}$. Then, for every full path $\mathcal{P}\in\Xi$: if $K\cup Q(\mathcal{P})$ is inconsistent, then set $\Xi := \Xi\setminus\{\mathcal{P}\}$, otherwise, set $Q(\mathcal{P}) := Q(\mathcal{P})\setminus K$. Proceed to Step 2.

We observe that for the full path $\rho(\bar{b})$, we have $K(\rho(\bar{b}))\subseteq K(S,\bar{b})$, meaning that this path accepts the tuple $\bar{b}$. We now show that the solution $\tau\left(\rho\left(\bar{b}\right)\right)$, which is attached to the leaf node of this path, is derivable from $K\left(\rho\left(\bar{b}\right)\right)$. The two variants of the finishing the work of the DT G are described in phases (a) and (b) of Step 2.

(a) There exists a full path $\mathcal{P} \in \Xi$ such that $Q(\mathcal{P})$ is empty. In this situation, the DT G terminates and returns the solution $\tau\left(\rho\left(\bar{b}\right)\right) = \tau(\mathcal{P})$, which will be attached to the leaf node of the path $\rho\left(\bar{b}\right)$. Clearly, we have $K(\mathcal{P}) \subseteq K\left(\rho\left(\bar{b}\right)\right) \subseteq K\left(S, \bar{b}\right)$. Since $\tau(\mathcal{P})$ is derivable from $K(\mathcal{P})$, it follows that $\tau\left(\rho\left(\bar{b}\right)\right) = \tau(\mathcal{P})$ is derivable from $K\left(\rho\left(\bar{b}\right)\right)$.

(b) There is no any full path $\mathcal{P} \in \Xi$ such that $Q(\mathcal{P})$ is empty and there is $i_0 \in \{1, \cdots, t\}$ for which $C_{i_0} \cap \Xi \neq \varnothing$ and $C_{i_0} \cap \Xi = \Xi$. Let us select the path $\mathcal{P} \in C_{i_0} \cap \Xi$ with the minimum number. In this case, the DT G terminates and returns the solution $\tau\left(\rho\left(\bar{b}\right)\right) = \tau(\mathcal{P})$, which will be attached to the leaf node of the path $\rho\left(\bar{b}\right)$.

Since $\mathcal{G}$ solves the task $T(\varphi, S, k)$, there exists a full path $\zeta \in FP^{+}(\mathcal{G})$ that accepts the tuple $\bar{b}$. For this path, we have $K(\zeta) \subseteq K\left(S, \bar{b}\right)$ and $\tau(\zeta) = \varphi\left(\bar{b}\right)$. Given that $K\left(\rho\left(\bar{b}\right)\right) \subseteq K\left(S, \bar{b}\right)$, the path ζ belongs to Ξ and consequently, belongs to C_{i_0}. Hence, leaf nodes of the paths from C_{i_0} are marked with the solution $\varphi\left(\bar{b}\right)$. In particular, this implies $\tau(\mathcal{P}) = \varphi\left(\bar{b}\right)$.

Let us now prove that $\tau\left(\rho\left(\bar{b}\right)\right) = \tau(\mathcal{P})$ is derivable from $K\left(\rho\left(\bar{b}\right)\right)$. Assume the contrary, that it is not derivable. Then there exists a tuple $\bar{d} \in E_k^n$, which is accepted by the path $\rho\left(\bar{b}\right)$ and for which $\varphi\left(\bar{d}\right) \neq \varphi\left(\bar{b}\right)$. Since $\mathcal{G}$ solves the task $T(\varphi, S, k)$, there is a full path $\theta \in FP^{+}(\mathcal{G})$ that accepts the tuple $\bar{d}$. For this path, $K(\theta) \subseteq K\left(S, \bar{d}\right)$ and $\tau(\theta) = \varphi\left(\bar{d}\right)$. Since $K\left(\rho\left(\bar{b}\right)\right) \subseteq K\left(S, \bar{d}\right)$, the path θ belongs to the set Ξ and therefore, to the set C_{i_0}, but this is impossible since leaf nodes of all paths from C_{i_0} are marked with the solution $\varphi\left(\bar{b}\right)$ and $\varphi\left(\bar{d}\right) \neq \varphi\left(\bar{b}\right)$.

Therefore, we obtain that the DT G solves the task $T(\varphi, S, k)$.

It is clear that in each full iteration of Step 2, including phase (c), the DT G evaluates at most $h(\mathcal{G})$ feature values. We now prove that the number of such full iterations of Step 2 is at most $h(\mathcal{G})$.

Since for any non-equivalent paths $\mathcal{P}_1, \mathcal{P}_2 \in FP^{+}(\mathcal{G})$ the equation system $K(\mathcal{P}_1) \cup K(\mathcal{P}_2)$ is inconsistent, it follows that during each full iteration of Step 2, every path $\mathcal{P}$ not in C_{i_0} (i.e., in $\Xi \setminus C_{i_0}$) will either be removed from Ξ or will have the cardinality of $Q(\mathcal{P})$ decreased by at least 1. Evidently, for each full paths $\mathcal{P} \in FP^{+}(\mathcal{G})$, the cardinality of the set $K(\mathcal{P})$ is at most $h(\mathcal{G})$. Therefore, after $h(\mathcal{G})$ complete repetitions of Step 2, we will find a full path $\mathcal{P} \in \Xi$ for which the set $Q(\mathcal{P})$ is empty or we will have $\Xi = \Xi \cap C_{i_0}$. In both cases, the DT G will finish its work.

Hence, the number of internal nodes along the path $\rho\left(\bar{b}\right)$ in G is at most $h(\mathcal{G})^2$. Taking into account that we considered an arbitrary full path in the DT G and $h(\mathcal{G}) = h^a(\varphi, S, k)$, we obtain $h(G) \leq h^a(\varphi, S, k)^2$. Since G is a DDT over $T(\varphi, S, k)$ that solves the task $T(\varphi, S, k)$, we have $h^d(\varphi, S, k) \leq h^a(\varphi, S, k)^2$. □

5. A Greedy Approach to Modeling the Operation of DDT (Deterministic Decision Tree)

Consider a DRS S with $n(S)=n$, and $k \ge V(S)$. We define $ES(S,k)$ as the set containing equation systems of the form

$$\{f_{i_1}=b_1,\cdots,f_{i_m}=b_m\},$$

where $m \in \mathbb{N}_0$, $f_{i_1},\cdots,f_{i_m} \in F(S)$, and $b_1,\cdots,b_m \in E_k$. We denote by $ES^+(S,k)$ the set of consistent equation systems from $ES(S,k)$. For $\alpha \in ES^+(S,k)$, denote $E_k^n(\alpha)=\{\bar{b} \in E_k^n : \alpha \subseteq K(S,\bar{b})\}$. This set can be interpreted as the set of solutions from E_k^n of the equation system α.

Definition 12 *Let φ be a solution map for the pair (S,k) and $\alpha \in ES^+(S,k)$. Let us define the parameter $M_\alpha(\varphi,S,k)$. For any $\bar{b} \in E_k^n$, we denote by $M_\alpha(\varphi,S,k,\bar{b})$ the minimum number $m \in \mathbb{N}_0$ such that there exists a subsystem β of the equation system $K(S,\bar{b})$, which contains m equations and for which the map φ is constant on the set $E_k^n(\alpha \cup \beta)$. Then $M_\alpha(\varphi,S,k)=max\{M_\alpha(\varphi,S,k,\bar{b}):\bar{b} \in E_k^n\}$. Note that $M_\varnothing(\varphi,S,k)=0$ if and only if φ is a degenerate map.*

A related parameter was introduced for decision tables within test theory in [20]. In the concept classes of exact learning, a similar parameter proposed in [21], where it is known as the extended teaching dimension, generalizes the teaching dimension concept from [22].

We will prove two statements related to the parameter $M_\alpha(\varphi,S,k)$. The first one has some independent interest.

Lemma 2 *Suppose S be a DRS, $k \ge V(S)$, and φ be a solution map for the pair (S,k). Then*

$$M_\varnothing(\varphi,S,k)=h^a(\varphi,S,k).$$

Proof. Let $n(S)=n$, $\mathcal{G}$ be a NDT over $T(\varphi,S,k)$, which solves the task $T(\varphi,S,k)$, where $h(\mathcal{G})=h^a(\varphi,S,k)$, and $\bar{b}$ be a tuple from E_k^n for which $M_\varnothing(\varphi,S,k)=M_\varnothing(\varphi,S,k,\bar{b})$. Since $\mathcal{G}$ solves the task $T(\varphi,S,k)$, there is a path $\mathcal{P} \in FP(\mathcal{G})$, where $K(\mathcal{P}) \subseteq K(S,\bar{b})$ and, for any tuple $\bar{d} \in E_k^n(K(\mathcal{P}))$, $\varphi(\bar{d})=\tau(\mathcal{P})$. Hence, $K(\mathcal{P})$ is a subsystem of $K(S,\bar{b})$ where the map φ is constant on the set $E_k^n(K(\mathcal{P}))$. Therefore, there are at most $h(\mathcal{G})$ equations in $K(\mathcal{P})$. As a result, $M_\varnothing(\varphi,S,k)=M_\varnothing(\varphi,S,k,\bar{b}) \le h(\mathcal{G})=h^a(\varphi,S,k)$, i.e, $M_\varnothing(\varphi,S,k) \le h^a(\varphi,S,k)$.

We now show that $h^a(\varphi,S,k) \le M_\varnothing(\varphi,S,k)$. Let $\bar{b} \in E_k^n$. From the definition of the parameter $M_\varnothing(\varphi,S,k)$, there exists a subsystem $\alpha(\bar{b})$ of $K(S,\bar{b})$, that contains at most $M_\varnothing(\varphi,S,k)$ equations and the map φ is equal to some constant $s(\bar{b})$ on the set $E_k^n(\alpha(\bar{b}))$. It is not difficult to build a full path $\mathcal{P}(\bar{b})$ having at most $M_\varnothing(\varphi,S,k)$ internal nodes, for which $K(\mathcal{P}(\bar{b}))=\alpha(\bar{b})$ and $\tau(\mathcal{P}(\bar{b}))$ equals $s(\bar{b})$. We identify roots of the full paths $\mathcal{P}(\bar{b})$, $\bar{b} \in E_k^n$, and denote the resulting DT by $\mathcal{G}$. It can be shown that $\mathcal{G}$ is a NDT over $T(\varphi,S,k)$, that solves the task $T(\varphi,S,k)$, where $h(\mathcal{G}) \le M_\varnothing(\varphi,S,k)$. Therefore

$h^a(\varphi,S,k) \le M_\varnothing(\varphi,S,k)$. Thus, $M_\varnothing(\varphi,S,k) = h^a(\varphi,S,k)$. □

Lemma 3 *Consider a DRS, $k \ge V(S)$, $\alpha \in ES^+(S,k)$, and φ be a solution map for the pair (S,k). Then*

$$M_\varnothing(\varphi,S,k) \ge M_\alpha(\varphi,S,k).$$

Proof. Let $n(S) = n$, $\bar{b} \in E_k^n$ and β be a subsystem of $K(S,\bar{b})$, where the map φ is constant on the set $E_k^n(\beta)$. Then β is a subsystem of $K(S,\bar{b})$, where the map φ is constant on the set $E_k^n(\alpha \cup \beta)$. From here it follows that $M_\varnothing(\varphi,S,k,\bar{b}) \ge M_\alpha(\varphi,S,k,\bar{b})$. Since $\bar{b}$ is an arbitrary tuple from the set E_k^n, we obtain $M_\varnothing(\varphi,S,k) \ge M_\alpha(\varphi,S,k)$. □

Definition 13 *Consider a DRS S, $k \ge V(S)$, and φ be a solution map for the pair (S,k). An uncertainty measure for the triple (φ,S,k) is a function $\gamma : ES^+(S,k) \to \mathbb{N}_0$, which satisfies the following conditions:*

- For any $\alpha \in ES^+(S,k)$, $\gamma(\alpha) = 0$ if and only if the map φ is constant on the set $E_k^n(\alpha)$.
- Let $\alpha \cup \{f_i = b\} \in ES^+(S,k)$ and the equation $f_i = b$ do not belong to the system α. Then $\gamma(\varnothing) - \gamma(\{f_i = b\}) \ge \gamma(\alpha) - \gamma(\alpha \cup \{f_i = b\})$

Note that $\gamma(\varnothing) = 0$ if and only if the map φ is degenerate. We proceed by proving an auxiliary statement.

Lemma 4 *Consider a DRS S, $k \ge V(S)$, φ be a nondegenerate solution map for the pair (S,k), and γ be an uncertainty measure for the triple (φ,S,k). Let $n(S) = n$, $F(S) = \{f_1,\cdots,f_n\}$, $\alpha \in ES^+(S,k)$ and $\gamma(\alpha) > 0$. Let, for $i = 1,\cdots,n$, d_i be a number from E_k such that $\gamma(\alpha \cup \{f_i = d_i\}) = max\{\gamma(\alpha \cup \{f_i = d\}) : d \in E_k\}$, and t be the minimum $i \in \{1,\cdots,n\}$ for which $\gamma(\alpha \cup \{f_i = d_i\})$ has the minimum value. Then*

$$\gamma(\alpha \cup \{f_t = d_t\}) \le \gamma(\alpha)(1 - 1/M_\alpha(\varphi,S,k)).$$

Proof. By the definition of the parameter $M_\alpha(\varphi,S,k)$ and from $\gamma(\alpha) > 0$, we conclude that there exist pairwise distinct features $f_{i_1},\cdots,f_{i_m} \in \{f_1,\cdots,f_n\}$ such that $\gamma(\alpha \cup \{f_{i_1} = d_{i_1},\cdots,f_{i_m} = d_{i_m}\}) = 0$ and $1 \le m \le M_\alpha(\varphi,S,k)$. Then

$$\begin{aligned}
&\gamma(\alpha) - \left(\gamma(\alpha) - \gamma(\alpha \cup \{f_{i_1} = d_{i_1}\})\right) \\
&- \left(\gamma(\alpha \cup \{f_{i_1} = d_{i_1}\}) - \gamma(\alpha \cup \{f_{i_1} = d_{i_1}, f_{i_2} = d_{i_2}\})\right) - \cdots \\
&- \left(\gamma(\alpha \cup \{f_{i_1} = d_{i_1},\cdots,f_{i_{m-1}} = d_{i_{m-1}}\}) - \gamma(\alpha \cup \{f_{i_1} = d_{i_1},\cdots,f_{i_m} = d_{i_m}\})\right) \\
&= \gamma(\alpha \cup \{f_{i_1} = d_{i_1},\cdots,f_{i_m} = d_{i_m}\}) = 0.
\end{aligned}$$

Since the features $f_{i_1},\cdots,f_{i_m}$ are pairwise different, for $j = 1,\cdots,m-1$, the equation system $\{f_{i_1} = d_{i_1},\cdots,f_{i_{j+1}} = d_{i_{j+1}}\}$ is consistent and the equation $f_{i_{j+1}} = d_{i_{j+1}}$ does not belong to the equation system $\{f_{i_1} = d_{i_1},\cdots,f_{i_j} = d_{i_j}\}$. Using the definition of uncertainty measure γ, we obtain that, for $j = 1,\cdots,m-1$,

$$\begin{aligned}
&\gamma(\alpha \cup \{f_{i_1} = d_{i_1},\cdots,f_{i_j} = d_{i_j}\}) - \gamma(\alpha \cup \{f_{i_1} = d_{i_1},\cdots,f_{i_j} = d_{i_j}, f_{i_{j+1}} = d_{i_{j+1}}\}) \\
&\le \gamma(\alpha) - \gamma(\alpha \cup \{f_{i_{j+1}} = d_{i_{j+1}}\}).
\end{aligned}$$

Therefore $\gamma(\alpha)-\sum_{j=1}^{m}\left(\gamma(\alpha)-\gamma\left(\alpha\cup\left\{f_{i_j}=d_{i_j}\right\}\right)\right)\le 0$. Since $\gamma(\alpha\cup\{f_t=d_t\})\le\gamma\left(\alpha\cup\left\{f_{i_j}=d_{i_j}\right\}\right)$ for $j=1,\cdots,m$, we have $\gamma(\alpha)-m(\gamma(\alpha)-\gamma(\alpha\cup\{f_t=d_t\}))\le 0$ and $\gamma(\alpha\cup\{f_t=d_t\})\le\gamma(\alpha)(1-1/m)$. Given that $m\le M_\alpha(\varphi,S,k)$, it follows that $\gamma(\alpha\cup\{f_t=d_t\})\le\gamma(\alpha)(1-1/M_\alpha(\varphi,S,k))$. □

Let S be a DRS with $n(S)=n$, $k\ge V(S)$, φ be a nondegenerate solution map for the pair (S,k), and γ be an uncertainty measure for the triple (φ,S,k). We now consider an algorithm $\mathcal{U}$ that, for the DRS S, the number k, the solution map φ, the uncertainty measure γ, and a tuple $\bar{b}=(b_1,\cdots,b_n)\in E_k^n$, simulates the work of a DDT $G_{\mathcal{U}}(\varphi,S,k,\gamma)$ over $T(\varphi,S,k)$, that solves the task $T(\varphi,S,k)$. Therefore, we get the description of a full path $\rho(\bar{b})$ from the DT $G_{\mathcal{U}}(\varphi,S,k,\gamma)$ which accepts the tuple $\bar{b}$. The set of full paths of the DT $G_{\mathcal{U}}(\varphi,S,k,\gamma)$ coincides with the set $\left\{\rho(\bar{b}):\bar{b}\in E_k^n\right\}$. Let, for definiteness, $F(S)=\{f_1,\cdots,f_n\}$.

Algorithm $\mathcal{U}$

Step 1:

Set $j:=1$ and $\alpha:=\varnothing$. Proceed to Step 2.

Step 2:

If $\gamma(\alpha)=0$, then the DT $G_{\mathcal{U}}(\varphi,S,k,\gamma)$ terminates and returns the solution $\tau(\rho(\bar{b}))=\varphi(\bar{d})$, where $\bar{d}$ is an arbitrary tuple from the set $E_k^n(\alpha)$. This solution is attached to the leaf node of the full path $\rho(\bar{b})$. Otherwise, proceed to Step 3.

Step 3:

Set t_j the minimum $i\in\{1,\cdots,n\}$ for which $\max\left\{\gamma(\alpha\cup\{f_i=d\}):d\in E_k\right\}$ has the minimum value. Compute the value of the feature f_{t_j}. Set $\alpha:=\alpha\cup\left\{f_{t_j}=b_{t_j}\right\}$ and $j:=j+1$. Proceed to Step 2.

Theorem 5 *Let* S *be a DRS with* $n(S)=n$, $k\ge V(S)$, φ *be a nondegenerate solution map for the pair* (S,k), *and* γ *be an uncertainty measure for the triple* (φ,S,k). *Then* $G_{\mathcal{U}}(\varphi,S,k,\gamma)$ *is a DDT over* $T(\varphi,S,k)$, *that solves the task* $T(\varphi,S,k)$ *such that*

$$h(G_{\mathcal{U}}(\varphi,S,k,\gamma))\le M_\varnothing(\varphi,S,k)\ln\gamma(\varnothing)+1.$$

Proof. Let, for definiteness, $F(S)=\{f_1,\cdots,f_n\}$. Consider an arbitrary tuple $\bar{b}=(b_1,\cdots,b_n)\in E_k^n$. The algorithm $\mathcal{U}$ constructs the full path $\rho(\bar{b})$ within the DT $G_{\mathcal{U}}(\varphi,S,k,\gamma)$, that accepts this tuple. It is not difficult to show that the solution $\tau(\rho(\bar{b}))$ is derivable from $K(\rho(\bar{b}))$. Since $\rho(\bar{b})$ is an arbitrary full path in the DT $G_{\mathcal{U}}(\varphi,S,k,\gamma)$, it follows that this tree is a DDT over $T(\varphi,S,k)$, that solves the task $T(\varphi,S,k)$.

We now analyze the number of internal nodes along the path $\rho(\bar{b})$. Suppose Step 3 of algorithm $\mathcal{U}$ is repeated p times during its work. Then $\rho(\bar{b})=v_0,d_0,v_1,d_1,\cdots,v_p,d_p,v_{p+1}$, where v_0 is the root of DT $G_{\mathcal{U}}(\varphi,S,k,\gamma)$, v_{p+1} is a leaf node and, for $j=0,\cdots,p$, the edge d_j leaves the node v_j and enters the node v_{j+1}. The node v_0 and the edge d_0 are not marked. For

$j=1,\cdots,p$, the node v_j is marked with the feature f_{t_j} and the edge d_j is marked with b_{t_j}. The leaf node v_{p+1} is marked with $\tau\left(\rho\left(\bar{b}\right)\right)$. Since the map φ is nondegenerate, $\gamma(\varnothing)\geq 1$ and $p\geq 1$. We now show that $p\leq M_{\varnothing}(\varphi,S,k)\ln\gamma(\varnothing)+1$.

For $j=1,\cdots,p$, let $\alpha_j=\left\{f_{t_1}=b_{t_1},\cdots,f_{t_j}=b_{t_j}\right\}$. Denote $\alpha_0=\varnothing$. It is clear that, for $j=0,\cdots,p$, the equation system α_j is consistent. From the description of the algorithm $\mathcal{U}$ it follows that, for $j=0,\cdots,p-1$, $\gamma\left(\alpha_j\right)>0$. By Lemma 4, for $j=0,\cdots,p-1$, $\gamma\left(\alpha_{j+1}\right)\leq\gamma\left(\alpha_j\right)\left(1-1/M_{\alpha_j}(\varphi,S,k)\right)$. From Lemma 3, we get that, for $j=0,\cdots,p-1$,

$$M_{\alpha_j}(\varphi,S,k)\leq M_{\varnothing}(\varphi,S,k).$$

Therefore, for $j=0,\cdots,p-1$, $\gamma\left(\alpha_{j+1}\right)\leq\ \gamma\left(\alpha_j\right)\left(1-1/M_{\varnothing}(\varphi,S,k)\right)$.

Since the map φ is nondegenerate, $M_{\varnothing}(\varphi,S,k)\geq 1$. Let $M_{\varnothing}(\varphi,S,k)=1$. Then $\gamma\left(\alpha_1\right)\leq 0$, i.e., $p=1$. In this case, $p\leq M_{\varnothing}(\varphi,S,k)\ln\gamma(\varnothing)+1$.

Let us assume now that $M_{\varnothing}(\varphi,S,k)\geq 2$. In this case, $\gamma\left(\alpha_{p-1}\right)\leq\gamma(\varnothing)\left(1-1/M_{\varnothing}(\varphi,S,k)\right)^{p-1}$. From the description of algorithm $\mathcal{U}$, it follows that $\gamma\left(\alpha_{p-1}\right)\geq 1$. Thus, $1\leq\gamma(\varnothing)\left(1-1/M_{\varnothing}(\varphi,S,k)\right)^{p-1}$, $\left(M_{\varnothing}(\varphi,S,k)/\left(M_{\varnothing}(\varphi,S,k)-1\right)\right)^{p-1}\leq\gamma(\varnothing)$ and $\left(1+1/\left(M_{\varnothing}(\varphi,S,k)-1\right)\right)^{p-1}\leq\gamma(\varnothing)$. By taking the natural logarithm on both sides of the inequality, we get $(p-1)\ln\left(1+1/\left(M_{\varnothing}(\varphi,S,k)-1\right)\right)\leq\ln L(S)$. It is well-known fact that, the inequality $\ln(1+1/q)>1/(q+1)$ holds, for any natural q. Given that $M_{\varnothing}(\varphi,S,k)\geq 2$, it follows that $(p-1)/M_{\varnothing}(\varphi,S,k)<\ln\gamma(\varnothing)$ and $p<M_{\varnothing}(\varphi,S,k)\ln\gamma(\varnothing)+1$. Since $\rho\left(\bar{b}\right)$ is an arbitrary full path in the DT $G_{\mathcal{U}}(\varphi,S,k,\gamma)$, we conclude that $h\left(G_{\mathcal{U}}(\varphi,S,k,\gamma)\right)\leq M_{\varnothing}(\varphi,S,k)\ln\gamma(\varnothing)+1$. □

From Lemma 2 and Theorem 1, we can conclude the statement below.

Corollary 1 *Let* S *be a DRS with* $n(S)=n$, $k\geq V(S)$, φ *be a nondegenerate solution map for the pair* (S,k), *and* γ *be an uncertainty measure for the triple* (φ,S,k). *Then* $G_{\mathcal{U}}(\varphi,S,k,\gamma)$ *is a DDT over* $T(\varphi,S,k)$, *that solves the task* $T(\varphi,S,k)$ *such that*

$$h\left(G_{\mathcal{U}}(\varphi,S,k,\gamma)\right)\leq h^a(\varphi,S,k)\ln\gamma(\varnothing)+1\leq h^d(\varphi,S,k)\ln\gamma(\varnothing)+1.$$

In order to consider the work of algorithm $\mathcal{U}$ efficient, we need to have

- An efficient algorithm that finds the value $\varphi\left(\bar{d}\right)$ for a given tuple $\bar{d}\in E_k^n$. In this case, we have direct access to the tuple $\bar{d}$ and we do not need to compute values of the features from the set $F(S)$.
- An efficient algorithm that finds the value $\gamma(\alpha)$ for a given equation system $\alpha\in ES^+(S,k)$.

We find it difficult to study the complexity of algorithms for computing individual functions φ and γ with finite domains. We defer a detailed discussion of this issue to the next section, in which we study infinite parametric families of functions φ and γ.

6. Recognition of Applicable Special Sets of Rules

Let S be a DRS with $n(S)=n$, $k\geq V(S)$, and W be a nonempty set of nonempty subsets of S such that, for any $w \in W$, the set of equations $K(w) = \bigcup_{r \in w} K(r)$ is consistent. We will call subsets w from W *special subsets of* S and will interpret them as subsets of S of special interest. We will call the set W a *family of special subsets for the DRS* S.

Let $\bar{b} \in E_k^n$. We will say that a special subset w is *applicable for the tuple* $\bar{b}$ if each DR $r \in w$ is applicable for this tuple, i.e., if $K(w) \subseteq K(S, \bar{b})$. Let us define a solution map φ_W for the pair (S, k) as follows: for $\bar{b} \in E_k^n$, $\varphi_W(\bar{b})$ is the set of all special subsets from W that are applicable for the tuple $\bar{b}$. It is clear that the solution map φ_W is nondegenerate. In this section, we will study the task $T(\varphi_W, S, k)$. Let us consider two examples:

- $W_1 = \{\{r\} : r \in S\}$. The task $T(\varphi_{W_1}, S, k)$ coincides with the task All Applicable Rules.
- W_2 is the set of all subsets of S of the kind $\{r_1, r_2\}$, where r_1 and r_2 have different decisions and $K(r_1) \cup K(r_2)$ is consistent. If the set W_2 is nonempty, then it is a family of special subsets for the DRS S.

Let us define an uncertainty measure γ_W for the triple (φ_W, S, k). Let $\alpha \in ES^+(S, k)$. For each subset $w \in W$, we consider two parameters: $p_\alpha(w) = |K(w) \setminus \alpha|$ and $q_\alpha(w)$ that is equal to 1 if the equation system $K(w) \cup \alpha$ is consistent and to 0 otherwise. Denote $\pi_\alpha(w) = p_\alpha(w) q_\alpha(w)$ and $\gamma_W(\alpha) = \sum_{w \in W} \pi_\alpha(w)$.

Proposition 6 *Let* S *be a DRS with* $n(S) = n$, $k \geq V(S)$, *and* W *be a family of special subsets for the DRS* S. *Then* γ_W *is an uncertainty measure for the triple* (φ_W, S, k).

Proof. (a) First, we show that, for any $\alpha \in ES^+(S, k)$, $\gamma_W(\alpha) = 0$ if and only if the map φ_W is constant on the set $E_k^n(\alpha)$. Let $\gamma_W(\alpha) = 0$. Then, for any $w \in W$, $\pi_\alpha(w) = 0$. Let us consider an arbitrary $w \in W$. Since $\pi_\alpha(w) = 0$, either $p_\alpha(w) = 0$ or $q_\alpha(w) = 0$ (obviously, the equalities $p_\alpha(w) = 0$ and $q_\alpha(w) = 0$ cannot be fulfilled simultaneously). If $p_\alpha(w) = 0$, then, for any $\bar{b} \in E_k^n(\alpha)$, the subset of DRs w is applicable for $\bar{b}$. If $q_\alpha(w) = 0$, then, for any $\bar{b} \in E_k^n(\alpha)$, the subset of DRs w is not applicable for $\bar{b}$. From here it follows that the map φ_W is constant on the set $E_k^n(\alpha)$.

Let the map φ_W be constant on the set $E_k^n(\alpha)$. Let us consider an arbitrary subset $w \in W$. Then either, for any $\bar{b} \in E_k^n(\alpha)$, the subset of DRs w is applicable for $\bar{b}$ or, for any $\bar{b} \in E_k^n(\alpha)$, the subset of DRs w is not applicable for $\bar{b}$. Let, for any $\bar{b} \in E_k^n(\alpha)$, the subset of DRs w be applicable for $\bar{b}$. Then $K(w) \subseteq \alpha$ and $p_\alpha(w) = 0$. Let, for any $\bar{b} \in E_k^n(\alpha)$, the subset of DRs w be not applicable for $\bar{b}$. Then the equation system $K(w) \cup \alpha$ is inconsistent and $q_\alpha(w) = 0$. Thus, $\pi_\alpha(w) = 0$. Since w is an arbitrary subset of the set W, we obtain $\gamma_W(\alpha) = 0$.

(b) We now show that, for any $\alpha \cup \{f_i = b\} \in ES^+(S, k)$, if $\{f_i = b\}$ is not a subset of α, then

$$\gamma_W(\varnothing) - \gamma_W(\{f_i = b\}) \geq \gamma_W(\alpha) - \gamma_W(\alpha \cup \{f_i = b\}). \tag{1}$$

We denote by β the equation system $\{f_i = b\}$. To prove (1), it is enough to show that, for any $w \in W$, the following inequality holds:

$$\pi_{\varnothing}(w) - \pi_{\beta}(w) \ge \pi_{\alpha}(w) - \pi_{\alpha \cup \beta}(w). \quad (2)$$

It is easy to see that $p_{\varnothing}(w) \ge p_{\alpha}(w)$ and $q_{\varnothing}(w) \ge q_{\alpha}(w)$. Therefore

$$\pi_{\varnothing}(w) \ge \pi_{\alpha}(w). \quad (3)$$

We now consider a number of cases.

(i) Let the equation system $K(w) \cup \beta$ be inconsistent. In this case, $\pi_{\beta}(w) = \pi_{\alpha \cup \beta}(w) = 0$. Using (3), we obtain (2).

(ii) Let the equation system $K(w) \cup \beta$ be consistent and the equation system $K(w) \cup \alpha$ be inconsistent. Then $\pi_{\alpha}(w) - \pi_{\alpha \cup \beta}(w) = 0$. It is clear that $\pi_{\varnothing}(w) - \pi_{\beta}(w) \ge 0$. Therefore (2) holds.

(iii) Let both equation systems $K(w) \cup \beta$ and $K(w) \cup \alpha$ be consistent. Since the equation system $\alpha \cup \beta$ is consistent, we obtain that $K(w) \cup \alpha \cup \beta$ is consistent. In this case, $\pi_{\varnothing}(w) - \pi_{\beta}(w) = p_{\varnothing}(w) - p_{\beta}(w)$ and $\pi_{\alpha}(w) - \pi_{\alpha \cup \beta}(w) = p_{\alpha}(w) - p_{\alpha \cup \beta}(w)$. We know that β is not a subset of α. Using this fact, it is easy to show that $p_{\varnothing}(w) - p_{\beta}(w) = p_{\alpha}(w) - p_{\alpha \cup \beta}(w)$. Therefore (2) holds. □

Suppose S be a DRS, $k \ge V(S)$, and W be a family of special subsets for the DRS S. First, we evaluate the depth of the DT $G_{\mathcal{U}}(\varphi_W, S, k, \gamma_W)$. According to Corollary 1, it is at most $h^a(\varphi_W, S, k) \ln \gamma_W(\varnothing) + 1$. It is clear that $\gamma_W(\varnothing) \le |W| n(S)$. Therefore

$$h(G_{\mathcal{U}}(\varphi_W, S, k, \gamma_W)) \le h^a(\varphi_W, S, k)(\ln|W| + \ln n(S)) + 1.$$

We now discuss the complexity of the algorithm $\mathcal{U}$ when it generates a full path in the DT $G_{\mathcal{U}}(\varphi_W, S, k, \gamma_W)$.

The DRS S can be represented by a word over the alphabet $\{(,), f, =, \wedge, \to, 0, 1, ;\}$ in which numbers from $\mathbb{N}_0$ (feature indices, feature values, and decisions of the DRs) are in binary representation (are represented by words over the alphabet $\{0,1\}$) and the symbol ";" is used to separate two DRs. The length of this word will be called the *size of the DRS* S and will be denoted $size(S)$. This word defines an order on the set S, that is, all DRs from S are numbered with indices $1, \cdots, |S|$. Each subset $w \in W$ is represented by the set of indices of DRs belonging to it. The set W can be represented by a word over the alphabet $\{\{,\}, 0, 1, ;\}$ in which numbers from $\mathbb{N}_0$ (indices of the DRs) are in binary representation and the symbol ";" is used to separate two indices. The length of this word will be called the *size of the family* W and will be denoted $size(W)$.

One can show that there is an algorithm, which finds the value $\varphi_W(\bar{d})$ for a given tuple $\bar{d} \in E_k^{n(S)}$ and which time complexity is bounded from above by a polynomial depending on $size(S)$ and $size(W)$. One can show also that there is an algorithm, which finds the value $\gamma_W(\alpha)$ for a given equation system $\alpha \in ES^+(S,k)$ and which time complexity is bounded from above by a polynomial depending on $size(S)$ and $size(W)$. It is easy to show that $h^a(\varphi_W, S, k) \le n(S)$. Therefore $h(G_{\mathcal{U}}(\varphi_W, S, k, \gamma_W)) \le n(S)(\ln|W| + \ln n(S)) + 1$.

Using these facts, one can prove that the algorithm $\mathcal{U}$ can be designed to have polynomial time complexity depending on $size(S)$ and $size(W)$.

7. Conclusion

In this paper, various tasks of recognizing the properties of DRSs are investigated. It is proved that the minimum depth of a DDT solving the task is bounded from above by the square of the minimum depth of an NDT. A greedy algorithm for modeling the operation of a DDT on a given tuple of feature values is proposed and investigated. Its complexity is studied for a class of tasks of recognizing the properties of DRSs. A more in-depth theoretical and experimental study of this and similar algorithms is planned in the future.

Acknowledgements

Research reported in this publication was supported by King Abdullah University of Science and Technology (KAUST).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Boros, E., Hammer, P.L., Ibaraki, T. and Kogan, A. (1997) Logical Analysis of Numerical Data. *Mathematical Programming*, **79**, 163-190. https://doi.org/10.1007/bf02614316

[2] Fürnkranz, J., Gamberger, D. and Lavrac, N. (2012) Foundations of Rule Learning. Cognitive Technologies. https://doi.org/10.1007/978-3-540-75197-7

[3] Moshkov, M. and Zielosko, B. (2011) Combinatorial Machine Learning—A Rough Set Approach. Studies in Computational Intelligence. https://doi.org/10.1007/978-3-642-20995-6

[4] Pawlak, Z. (1991) Rough Sets—Theoretical Aspects of Reasoning about Data. Theory and Decision Library: Series D.

[5] Skowron, A. and Rauszer, C. (1992) The Discernibility Matrices and Functions in Information Systems. In: Słowiński, R., Ed., *Intelligent Decision Support—Handbook of Applications and Advances of the Rough Sets Theory*, Springer, 331-362. https://doi.org/10.1007/978-94-015-7975-9_21

[6] Breiman, L., Friedman, J.H. and Olshen, R.A. (1984) Classification and Regression Trees. Wadsworth and Brooks.

[7] Moshkov, M.J. (2005) Time Complexity of Decision Trees. In: Peters, J.F. and Skowron, A., Eds., *Transactions on Rough Sets III*, Springer, 244-459. https://doi.org/10.1007/11427834_12

[8] Quinlan, J.R. (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann.

[9] Rokach, L. and Maimon, O. (2007) Data Mining with Decision Trees—Theory and Applications. World Scientific Publishing Co. Pte. Ltd. https://doi.org/10.1142/9789812771728

[10] Molnar, C. (2022) Interpretable Machine Learning. A Guide for Making Black Box Models Explainable, Self-Published.

https://christophm.github.io/interpretable-ml-book/

[11] Moshkov, M. (1998) Some Relationships between Decision Trees and Decision Rule Systems. In: Polkowski, L. and Skowron, A., Eds., *Rough Sets and Current Trends in Computing*, Springer, 499-505. https://doi.org/10.1007/3-540-69115-4_68

[12] Moshkov, M. (2001) On transformation of Decision Rule Systems into Decision Trees. *Proceedings of the Seventh International Workshop Discrete Mathematics and Its Applications*, Moscow, 29 January-2 February 2001, 21-26.

[13] Durdymyradov, K., Moshkov, M. and Ostonov, A. (2025) Decision Trees Versus Systems of Decision Rules: A Rough Set Approach. Springer. https://doi.org/10.1007/978-3-031-71586-0

[14] Durdymyradov, K. and Moshkov, M. (2025) Deterministic and Nondeterministic Decision Trees for Recognition of All Realizable Decision Rules. In: Nguyen, N.T., *et al.*, Eds., *Lecture Notes in Computer Science*, Springer Nature Singapore, 3-17. https://doi.org/10.1007/978-981-96-6005-6_1

[15] Durdymyradov, K. and Moshkov, M. (2025) Deterministic and Nondeterministic Decision Trees for Recognition of Properties of Decision Rule Systems. In: Zhang, Q., *et al.*, Eds., *Lecture Notes in Computer Science*, Springer Nature Switzerland, 426-435. https://doi.org/10.1007/978-3-031-92744-7_29

[16] Blum, M. and Impagliazzo, R. (1987) Generic Oracles and Oracle Classes. 28*th Annual Symposium on Foundations of Computer Science* (*SFCS* 1987), Los Angeles, 12-14 October 1987, 118-126. https://doi.org/10.1109/sfcs.1987.30

[17] Hartmanis, J. and Hemachandra, L.A. (1987) One-Way Functions, Robustness, and the Non-Isomorphism of NP-Complete Sets. *Proceedings of the Second Annual Conference on Structure in Complexity Theory*, Ithaca, 16-19 June 1987, 160-174. https://doi.org/10.1109/PSCT.1987.10319267

[18] Tardos, G. (1989) Query Complexity, or Why Is It Difficult to Separate $NP^A \cap coNP^A$ From P^A by Random Oracles A? *Combinatorica*, **9**, 385-392. https://doi.org/10.1007/bf02125350

[19] Buhrman, H. and de Wolf, R. (2002) Complexity Measures and Decision Tree Complexity: A Survey. *Theoretical Computer Science*, **288**, 21-43. https://doi.org/10.1016/s0304-3975(01)00144-x

[20] Moshkov, M. (1982) On Conditional Tests. *Doklady Physics*, **27**, 528-530. http://mi.mathnet.ru/dan45454

[21] Hegedűs, T. (1995) Generalized Teaching Dimensions and the Query Complexity of Learning. *Proceedings of the Eighth Annual Conference on Computational Learning theory*, Santa Cruz, 5-8 July 1995, 108-117. https://doi.org/10.1145/225298.225311

[22] Goldman, S.A. and Kearns, M.J. (1995) On the Complexity of Teaching. *Journal of Computer and System Sciences*, **50**, 20-31. https://doi.org/10.1006/jcss.1995.1003