Vietnam J Comput Sci (2018) 5:3–14
https://doi.org/10.1007/s40595-017-0094-4

REGULAR PAPER

# On the possibility of correct concept learning in description logics

**Ali Rezaei Divroodi**[1] · **Quang-Thuy Ha**[2] · **Linh Anh Nguyen**[3,4] · **Hung Son Nguyen**[4]

Received: 6 August 2016 / Accepted: 4 January 2017 / Published online: 15 February 2017
© The Author(s) 2017. This article is published with open access at Springerlink.com

**Abstract** It is well known that any Boolean function in classical propositional calculus can be learned correctly if the training information system is good enough. In this paper, we extend that result for description logics. We prove that any concept in any description logic that extends $\mathcal{ALC}$ with some features amongst $I$ (inverse roles), $Q_k$ (qualified number restrictions with numbers bounded by a constant $k$), and Self (local reflexivity of a role) can be learned correctly if the training information system (specified as a finite interpretation) is good enough. That is, there exists a learning algorithm such that, for every concept $C$ of those logics, there exists a training information system such that applying the learning algorithm to it results in a concept equivalent to $C$. For this result, we introduce universal interpretations and bounded bisimulation in description logics and develop an appropriate learning algorithm. We also generalize common types of queries for description logics, introduce interpretation queries, and present some consequences.

**Keywords** Concept learning · Learnability · Description logics · Bisimulation

✉ Linh Anh Nguyen
nguyenanhlinh@tdt.edu.vn

Ali Rezaei Divroodi
rezaeial83@yahoo.com

Quang-Thuy Ha
thuyhq@vnu.edu.vn

Hung Son Nguyen
son@mimuw.edu.pl

[1] Education Department, Enghelab 2, 46731-83354 Noshahr, Iran

[2] Faculty of Information Technology, VNU University of Engineering and Technology, 144 Xuan Thuy, Hanoi, Vietnam

[3] Division of Knowledge and System Engineering for ICT, Ton Duc Thang University, No. 19, Nguyen Huu Tho Street, Tan Phong Ward, District 7, Ho Chi Minh City, Vietnam

[4] Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland

## 1 Introduction

It is well known that any Boolean function in classical propositional calculus can be learned correctly if the training information system is good enough. In this paper, we extend that result for description logics.

Description logics (DLs) are a family of formal languages suitable for representing and reasoning about terminological knowledge. They are of particular importance in providing a logical formalism for ontologies and the Semantic Web. Binary classification in the context of DLs is called concept learning, as the function to be learned is expected to be characterizable by a concept. This differs from the traditional setting in that objects are described not only by attributes but also by relationship between the objects (i.e., by object roles). The major settings of concept learning in DLs are as follows:

- *Setting 1* Given a knowledge base $KB$ in a description logic $L$ and sets $E^+$ and $E^-$ of individuals, learn a concept $C$ in $L$ such that:
  1. $KB \models C(a)$ for all $a \in E^+$, and
  2. $KB \models \neg C(a)$ for all $a \in E^-$.

  The set $E^+$ contains positive examples of $C$, while $E^-$ contains negative ones.
- *Setting 2* This differs from the previous one only in that the condition 2 is weakened:

1. $KB \models C(a)$ for all $a \in E^+$, and
2. $KB \not\models C(a)$ for all $a \in E^-$.

- *Setting 3* Given a finite interpretation $\mathcal{I}$ and sets $E^+$ and $E^-$ of individuals, learn a concept $C$ in $L$ such that:

  1. $\mathcal{I} \models C(a)$ for all $a \in E^+$, and
  2. $\mathcal{I} \models \neg C(a)$ for all $a \in E^-$.

  Note that $\mathcal{I} \not\models C(a)$ is the same as $\mathcal{I} \models \neg C(a)$.

Settings 1 and 2 are useful for ontology engineering in suggesting concept definitions. Setting 3 is closer to binary classification in traditional machine learning, where the interpretation $\mathcal{I}$ and the sets $E^+$, $E^-$ form a training information system.

In this paper, we study the possibility of correct concept learning in DLs using Setting 3.

### 1.1 Related work on learnability in traditional machine learning

PAC learning (probably approximately correct learning) is a framework for mathematical analysis of machine learning proposed by Valiant [32]. In this framework, the learner receives samples and must select from a certain class a hypothesis that approximates the function to be learned. The goal is that, with high probability, the selected hypothesis will have low generalization error. As efficient PAC learnability, the learner must be able to learn the concept in polynomial time given any arbitrary approximation ratio, probability of success, or distribution of the samples. Valiant [32] provided some results on PAC learnability of bounded CNF expressions, DNF expressions, and $\mu$-expressions in classical propositional logic.

Angluin [2] studied exact and probably exact learnability using different types of queries, such as membership, equivalence, subset, superset, disjointness, and exhaustiveness. She provided some general lower bound techniques and compared equivalence queries with Valiant's criterion of PAC identification under random sampling.

Bshouty et al. [4] studied a model of probably exactly correct (PExact) learning that can be viewed either as the Exact model [2] relaxed so that counterexamples to equivalence queries are distributionally drawn rather than adversarially chosen, or as the PAC model [32] strengthened to require a perfect hypothesis. They also introduced a model of probably almost exactly correct learning (PAExact) that lies between the PExact and PAC models. They obtained a number of separation results between these models and some results on efficient parallel learning in the PAExact model.

There are many other works related to learnability of concepts/theories in traditional machine learning or inductive logic programming. For example, De Raedt and Dzeroski [28] showed that first-order $jk$-clausal theories are PAC learnable. A survey on this subject is beyond the scope of this paper.

### 1.2 Related work on concept learning in description logics

Regarding learnability in DLs, we are only aware of the following works:

- Cohen and Hirsh [5] proved that concepts in the C-CLASSIC description logic are PAC learnable using subsumption queries (i.e., subset queries). This logic is an early DL formalism that differs from the basic DL $\mathcal{ALC}$ in that: it allows unqualified number restrictions as well as the MIN and MAX operators, but it allows the union constructor and the existential restriction only for atomic concepts and it disallows the complement constructor. The authors also proposed an algorithm called LCSLearn for learning concepts and disjunctions of concepts in C-CLASSIC from individuals, which is based on "least common subsumers".
- Frazier and Pitt [12] proved that concepts in the CLASSIC description logic can be learned using equivalence and subsumption queries. The logic CLASSIC differs from C-CLASSIC in that it allows the concept constructor specified by equality between two chains of functional roles, but disallows the MIN and MAX operators. They also showed that learning concepts in CLASSIC from individuals is as hard as predicting arbitrary polynomial-sized circuits and that subsumption queries alone do not suffice for learning CLASSIC.
- Konev et al. [18] studied exact learnability of TBoxes in lightweight DLs using subsumption and equivalence queries. They proved that: TBoxes formulated in DL-Lite with role inclusions and composite concepts on the right-hand side of concept inclusions can be learned in polynomial time; $\mathcal{EL}$ TBoxes with only concept names on the right-hand side of concept inclusions can be learned in polynomial time. They also gave some negative results on learnability.

Regarding concept learning in DLs, in an early work [20], Lambrix and Larocchia proposed a simple algorithm based on concept normalization. The other works [3,11,16,21] study concept learning in DLs using refinement operators as in inductive logic programming. The works [3] by Badea and Nienhuys-Cheng and [16] by Iannone et al. use Setting 1, while the works [11] by Fanizzi et al. and [21] by Lehmann and Hitzler use Setting 2. Apart from refinement operators, scoring functions and search strategies also play important roles in the algorithms proposed in those works. The DL-Learner system [21] exploits genetic programming

techniques, while the DL-FOIL algorithm [11] considers also unlabeled data as in semi-supervised learning. A comparison between DL-Learner, YinYang [16], and LCSLearn can be found in [14].

Nguyen and Szałas [26] applied bisimulation to concept learning in DLs. They also studied concept approximation using bisimulation and the rough set theory of Pawlak [27]. Tran et al. [29,31] generalized and extended the concept learning method of [26] for DL-based information systems. They took attributes as the basic elements of the language. The works [26,29,31] use Setting 3.

Ha et al. [13] gave a bisimulation-based method, called BBCL, for concept learning in DLs using Setting 1. Tran et al. [30] gave a bisimulation-based method, called BBCL2, for concept learning in DLs using Setting 2.

There are also other related works on learning terminological axioms or ontologies. The works [1,6,23] study theory learning or concept inclusion axioms learning in DLs. The works [1,23] involve probabilistic DLs. Some other researchers combined concept learning in DLs with inductive logic programming (e.g., [17]) or studied concept learning in DLs via inductive logic programming (e.g., [19]). The work [22] gives a characterization of concept learning in DLs using Setting 1.

### 1.3 Our contributions

In this paper, we prove that any concept in any description logic that extends the basic DL $\mathcal{ALC}$ with some features amongst $I$ (inverse roles), $Q_k$ (qualified number restrictions with numbers bounded by a constant $k$), and Self (local reflexivity of a role) can be learned if the training information system (specified as a finite interpretation) is good enough. That is, there exists a learning algorithm such that, for every concept $C$ of those logics, there exists a training information system such that applying the learning algorithm to it results in a concept equivalent to $C$. We call this property *C-learnability* (possibility of correct learning). Our work uses Setting 3.

Note that our result is completely different from the ones of [5,12,18], as we consider learning concepts from individuals, while the learnability results of those works do not (they use subsumption and equivalence queries). Furthermore, the DLs considered by us are more advanced than C-CLASSIC, CLASSIC, DL-Lite, and $\mathcal{EL}$. In addition, note that C-learnability is different from PAC learnability.

To obtain the mentioned result, our work uses bounded bisimulation in DLs and a new version of the algorithms proposed in the works [26,31] that minimizes modal depths of the resulting concepts. It shows a good property of the bisimulation-based concept learning methods.

This paper is a revised and extended version of our conference paper [8] and a part of the Ph.D. dissertation [7] of the first author. In comparison with [8], it contains full proofs of the results, illustrative examples, and a correction for a normalization rule. Furthermore, we also generalize common types of queries for DLs, introduce *interpretation queries*, and present some consequences.

Based on our work [8], Tran et al. [30] studied C-learnability in a certain class of DLs using Setting 2. That class is a bit larger than the one considered in our work, as it also allows the features $N_k$ (unqualified number restrictions with numbers bounded by a constant $k$) and $F$ (role functionality), which are special forms of $Q_k$ (qualified number restrictions with numbers bounded by a constant $k$). Their paper refers to our work [8] for some notions and proofs. The current paper differs from their paper in that it uses Setting 3, while the latter uses Setting 2. These two settings are essentially different. Furthermore, the current paper contains all details, including full proofs. In addition, note that all the other previous joint papers by any co-author of the current paper do not deal with learnability in DLs.

### 1.4 The structure of this paper

The rest of this paper is structured as follows. In Sect. 2, we introduce notation and semantics of DLs. In Sect. 3, we present concept normalization and introduce universal interpretations. In Sect. 4, we define bounded bisimulation in DLs and state its properties. In Sect. 5, we present a concept learning algorithm, which is used in Sect. 6 for analyzing C-learnability in DLs. In Sect. 7, we discuss concept learning in DLs using queries. Concluding remarks are given in Sect. 8.

## 2 Notation and semantics of description logics

A *DL-signature* is a set $\Sigma = \Sigma_I \cup \Sigma_C \cup \Sigma_R$, where $\Sigma_I$ is a finite set of individual names, $\Sigma_C$ is a finite set of *concept names*, and $\Sigma_R$ is a finite set of *role names*. Concept names stand for unary predicates, while role names stand for binary predicates. We denote concept names by capital letters, such as $A$ and $B$, role names by lower case letters, such as $r$ and $s$, and individual names by lower case letters, such as $a$ and $b$.

We will consider *DL-features* denoted by $I$ (*inverse roles*), $Q_k$ (*qualified number restrictions with numbers bounded by a constant k*), and Self (*local reflexivity of a role*). In this paper, by a *set of DL-features*, we mean an empty set or a finite set consisting of some of these names.

Let $\Sigma$ be a DL-signature and $\Phi$ be a set of DL-features. Let $\mathcal{L}$ stand for $\mathcal{ALC}$, which is the name of a basic DL (we treat $\mathcal{L}$ as a language, but not a logic). The DL language $\mathcal{L}_{\Sigma,\Phi}$ allows *roles* and *concepts* defined inductively as follows:

- If $r \in \Sigma_R$, then $r$ is a role of $\mathcal{L}_{\Sigma,\Phi}$.

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad \bot^{\mathcal{I}} = \emptyset \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$$
$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$
$$(\exists r.\mathsf{Self})^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid r^{\mathcal{I}}(x,x)\}$$
$$(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y\,[R^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y)]\}$$
$$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y\,[R^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y)]\}$$
$$(\geq h\, R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \mid R^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y)\} \geq h\}$$
$$(< h\, R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \mid R^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y)\} < h\}$$

**Fig. 1** Interpretation of complex concepts

- If $I \in \Phi$, then $r^-$ is a role of $\mathcal{L}_{\Sigma,\Phi}$.
- If $A \in \Sigma_C$, then $A$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$.
- If $C$ and $D$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$, $R$ is a role of $\mathcal{L}_{\Sigma,\Phi}$, $r \in \Sigma_R$, and $h$ and $k$ are natural numbers, then
  - $\top, \bot, \neg C, C \sqcap D, C \sqcup D, \forall R.C$ and $\exists R.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$.
  - If $Q_k \in \Phi$ and $h \leq k$, then $\geq h\, R.C$ and $< h\, R.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$ (we use $< h\, R.C$ instead of $\leq h\, R.C$, because it is more "dual" to $\geq h\, R.C$).
  - If $\mathsf{Self} \in \Phi$, then $\exists r.\mathsf{Self}$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$.

A role $r^-$ is called the *inverse* of $r$. The symbols $\top$ and $\bot$ stand for *truth* and *falsity*, respectively. The constructors $\neg$, $\sqcap$, and $\sqcup$ stand for *complement*, *intersection*, and *union*, respectively. The constructors $\forall R.C$ and $\exists R.C$ are called *universal restriction* and *existential restriction*, respectively. The constructors $\geq h\, R.C$ and $< h\, R.C$ are called *qualified number restrictions*. The constructor $\exists r.\mathsf{Self}$ stands for *local reflexivity* of $r$.

An *interpretation* over $\Sigma$ is a pair $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a non-empty set called the *domain* of $\mathcal{I}$ and $\cdot^{\mathcal{I}}$ is a mapping called the *interpretation function* of $\mathcal{I}$ that associates each individual $a \in \Sigma_I$ with an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each concept name $A \in \Sigma_C$ with a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each role name $r \in \Sigma_R$ with a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. For $r \in \Sigma_R$, define $(r^-)^{\mathcal{I}} = (r^{\mathcal{I}})^{-1}$. The interpretation function $\cdot^{\mathcal{I}}$ is extended to complex concepts, as shown in Fig. 1, where $\sharp\Gamma$ stands for the cardinality of the set $\Gamma$.

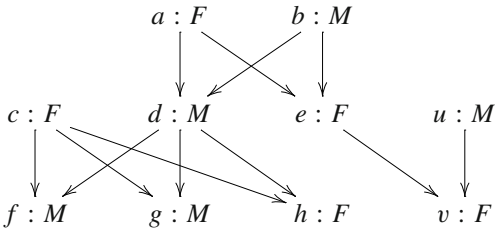
An *information system* over $\Sigma$ is defined to be a finite interpretation over $\Sigma$.

*Example 2.1* Let

- $\Sigma_I = \{Alice, Bob, Claudia, Dave, Eva, Frank, George, Helen\}$,
- $\Sigma_C = \{Male, Female, Father, Mother\}$,
- $\Sigma_R = \{hasChild, hasParent\}$.

Consider the information system $\mathcal{I}$ specified by

- $\Delta^{\mathcal{I}} = \{a, b, c, d, e, f, g, h, u, v\}$,
- $Alice^{\mathcal{I}} = a$, $Bob^{\mathcal{I}} = b$, …, $Helen^{\mathcal{I}} = h$ ($u$ and $v$ are unnamed individuals),
- $hasChild^{\mathcal{I}}$ consists of elements illustrated by edges in the following graph:




(in this graph, the letter $M$ denotes *Male*, and $F$ denotes *Female*),

- $hasParent^{\mathcal{I}} = (hasChild^{-1})^{\mathcal{I}} = (hasChild^{\mathcal{I}})^{-1}$,
- $Male^{\mathcal{I}} = \{b, d, f, g, u\}$,
- $Female^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus Male^{\mathcal{I}} = \{a, c, e, h, v\}$,
- $Father^{\mathcal{I}} = (Male \sqcap \exists hasChild.\top)^{\mathcal{I}} = \{b, d, u\}$,
- $Mother^{\mathcal{I}} = (Female \sqcap \exists hasChild.\top)^{\mathcal{I}} = \{a, c, e\}$.

As examples, we have that:

- $(\exists hasChild.\mathsf{Self})^{\mathcal{I}} = \emptyset$.
- $(\geq 3\, hasChild.\top)^{\mathcal{I}} = \{c, d\}$.
- $(\geq 2\, hasChild.Male)^{\mathcal{I}} = \{c, d\}$.
- $(Female \sqcap < 2\, hasChild.\top)^{\mathcal{I}} = \{e, h, v\}$. □

A concept $C$ of $\mathcal{L}_{\Sigma,\Phi}$ is *satisfiable* if there exists an interpretation $\mathcal{I}$ over $\Sigma$ such that $C^{\mathcal{I}} \neq \emptyset$. We say that concepts $C$ and $D$ of $\mathcal{L}_{\Sigma,\Phi}$ are *equivalent* if $C^{\mathcal{I}} = D^{\mathcal{I}}$ for every interpretation $\mathcal{I}$ over $\Sigma$.

The *modal depth* of a concept $C$, denoted by $mdepth(C)$, is defined to be:

- 0 if $C$ is of the form $\top$, $\bot$, $A$ or $\exists r.\mathsf{Self}$;
- $\mathsf{mdepth}(D)$ if $C$ is of the form $\neg D$;
- $\max(\mathsf{mdepth}(D), \mathsf{mdepth}(D'))$ if $C$ is of the form $D \sqcap D'$ or $D \sqcup D'$;
- $\mathsf{mdepth}(D)+1$ if $C$ is of the form $\forall R.D, \exists R.D, \geq h\, R.C$ or $< h\, R.C$.

For example,

$$\mathsf{mdepth}(\exists r.(\forall s^-.(A \sqcup \exists r.\mathsf{Self}) \sqcap \exists s.(\neg A))) = 2.$$

Let $d$ denote a natural number. By $\mathcal{L}_{\Sigma,\Phi,d}$, we denote the sublanguage of $\mathcal{L}_{\Sigma,\Phi}$ that consists of concepts with modal depth not greater than $d$.

## 3 Concept normalization

There are different normal forms for formulas or concepts (see, e.g., [24]). We provide below such a form. The aim is to introduce the notion of *universal interpretation* and a lemma about its existence, which in turn is used in Sect. 6 to prove our result about C-learnability in DLs. Our normal form uses the following normalization rules:

- Replace $\forall R.C$ by $\neg\exists R.\neg C$.
  Replace $< h\ R.C$ by $\neg \geq h\ R.C$.
- Replace $\geq 0\ R.C$ by $\top$. Replace $\geq 1\ R.C$ by $\exists R.C$.
- Push $\neg$ in depth through $\top$, $\bot$, $\neg$, $\sqcap$, $\sqcup$ according to De Morgan's laws.
- Represent $C_1 \sqcap \cdots \sqcap C_n$ as an "and"-set $\sqcap\{C_1, \ldots, C_n\}$ to make the order inessential and eliminate duplicates. Use a dual rule for $\sqcup$ and "or"-sets.
- Flatten an "and"-set $\sqcap\{\sqcap\{C_1, \ldots, C_i\}, C_{i+1}, \ldots, C_n\}$ to $\sqcap\{C_1, \ldots, C_n\}$.
  Replace $\sqcap\{C\}$ by $C$.
  Replace $\sqcap\{\top, C_1, \ldots, C_n\}$ by $\sqcap\{C_1, \ldots, C_n\}$.
  Replace $\sqcap\{\bot, C_1, \ldots, C_n\}$ by $\bot$.
  Use the dual rules for "or"-sets.
- Replace $\exists R.\sqcup\{C_1, \ldots, C_n\}$ by $\sqcup\{\exists R.C_1, \ldots, \exists R.C_n\}$.
- Replace $\geq h\ R.\sqcup\{C_1, \ldots, C_n\}$ by the union (using $\sqcup$) of all concepts of the form $\sqcap\{\geq h_1 R.D_1, \ldots, \geq h_n R.D_n\}$, where $D_i = \sqcap\{C_i, \neg C_1, \ldots, \neg C_{i-1}\}$ for $1 \leq i \leq n$, and $h_1, \ldots, h_n$ are natural numbers such that $h_1 + \cdots + h_n = h$.[1]
- Distribute $\sqcap$ over $\sqcup$.

A concept is said to be in the *DEG normal form* (in short, *DEGNF*)[2] if it cannot be changed by any one of the above rules. The following two lemmas can easily be proved.

**Lemma 3.1** *Every concept can be translated to the DEG-normal form. If $C'$ is the DEG normal form of $C$ then they are equivalent. A concept in the DEG normal form may contain $\sqcup$ only at the most outer level (i.e., either it does not contain $\sqcup$ or it must be of the form $\sqcup\{C_1, \ldots, C_n\}$, where $C_1, \ldots, C_n$ do not contain $\sqcup$).*

**Lemma 3.2** *$\mathcal{L}_{\Sigma,\Phi,d}$ has only finitely many concepts in the DEG normal form. All of them can effectively be constructed.*

In the case $\Phi = \{I, Q_k, \mathsf{Self}\}$, $|\Sigma_C| = m$ and $|\Sigma_R| = n$, an upper bound $T(d)$ for the number of concepts in the DEG normal form of $\mathcal{L}_{\Sigma,\Phi,d}$ can be estimated as follows:

$$T'(0) = 2^{2m+2n+2}$$
$$T'(l+1) = 2^{4k.n.T'(l)+2m+2n+2} \quad \text{for } l \geq 0$$
$$T(d) = 2^{T'(d)},$$

where $T'(l)$ is an upper bound for the number of concepts in the DEG normal form of $\mathcal{L}_{\Sigma,\Phi,d}$ that do not use $\sqcup$ and have a modal depth not greater than $l$.

We say that an interpretation $\mathcal{I}$ over $\Sigma$ is *universal* with respect to a sublanguage of $\mathcal{L}_{\Sigma,\Phi}$ if, for every satisfiable concept $C$ of that sublanguage, $C^{\mathcal{I}} \neq \emptyset$.

**Lemma 3.3** *There exists a finite universal interpretation with respect to $\mathcal{L}_{\Sigma,\Phi,d}$, which can effectively be constructed.*

*Proof* Let $C_1, \ldots, C_n$ be all the satisfiable concepts in the DEG normal form of $\mathcal{L}_{\Sigma,\Phi,d}$. (By Lemma 3.2, the number of such concepts is finite.) For each $1 \leq i \leq n$, let $\mathcal{I}_i$ be a finite model satisfying $C_i$, which can effectively be constructed using some tableau algorithm (e.g., [15,25]).[3] Without loss of generality we assume that these interpretations have pairwise disjoint domains. Let $\mathcal{I}$ be any interpretation such that: $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}_1} \cup \cdots \cup \Delta^{\mathcal{I}_n}$; for $A \in \Sigma_C$, $A^{\mathcal{I}} = A^{\mathcal{I}_1} \cup \cdots \cup A^{\mathcal{I}_n}$; for $r \in \Sigma_R$, $r^{\mathcal{I}} = r^{\mathcal{I}_1} \cup \cdots \cup r^{\mathcal{I}_n}$. (Individual names can be interpreted in $\mathcal{I}$ arbitrarily.) It is easy to see that $\mathcal{I}$ is finite and universal with respect to $\mathcal{L}_{\Sigma,\Phi,d}$. □

## 4 Bounded bisimulation for description logics

Indiscernibility in DLs is related to bisimulation. Divroodi and Nguyen [9,10] studied bisimulations for a number of DLs. Nguyen and Szałas [26] generalized that notion to model indiscernibility of objects and study concept learning. Tran et al. [31] and Ha et al. [13] generalized that notion further for concept learning. In this section, we present bounded bisimulation for the DLs studied in the current paper. The theorems given in this section differ from the ones in the mentioned works mainly in that we are now dealing with *bounded* bisimulation (but not bisimulation) and the considered class of DLs is different. They serve as technical tools for proving our main result about C-learnability in DLs.

Let $d$ be a natural number and let

- $\Sigma$ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$,
- $\Phi$ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$,
- $\mathcal{I}$ and $\mathcal{I}'$ be interpretations over $\Sigma$.

A binary relation $Z_d \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}'}$ is called an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$ if there exists a sequence of

[1] This corrects [7,8].

[2] DEGNF stands for disjunctive-existential-greater-or-equal normal form.

[3] As RBoxes and TBoxes are not considered, $\mathcal{L}_{\Sigma,\Phi,d}$ has the finite model property.

binary relations $Z_d \subseteq \cdots \subseteq Z_0 \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}'}$ such that the following conditions hold for every $0 \leq i \leq d$, $0 \leq j < d$, $a \in \Sigma_I^\dagger$, $A \in \Sigma_C^\dagger$, $x, y \in \Delta^{\mathcal{I}}$, $x', y' \in \Delta^{\mathcal{I}'}$, and every role $R$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$:

$$Z_i(a^{\mathcal{I}}, a^{\mathcal{I}'}) \tag{1}$$

$$Z_0(x, x') \Rightarrow [A^{\mathcal{I}}(x) \Leftrightarrow A^{\mathcal{I}'}(x')] \tag{2}$$

$$[Z_{j+1}(x, x') \wedge R^{\mathcal{I}}(x, y)] \Rightarrow \exists y' \in \Delta^{\mathcal{I}'}[Z_j(y, y') \wedge R^{\mathcal{I}'}(x', y')] \tag{3}$$

$$[Z_{j+1}(x, x') \wedge R^{\mathcal{I}'}(x', y')] \Rightarrow \exists y \in \Delta^{\mathcal{I}}[Z_j(y, y') \wedge R^{\mathcal{I}}(x, y)], \tag{4}$$

if $Q_k \in \Phi^\dagger$ and $1 \leq h \leq k$ then
if $Z_{j+1}(x, x')$ holds and $y_1, \ldots, y_h$ are pairwise different elements of $\Delta^{\mathcal{I}}$ such that $R^{\mathcal{I}}(x, y_l)$ holds for every $1 \leq l \leq h$, then there exist pairwise different elements $y'_1, \ldots, y'_h$ of $\Delta^{\mathcal{I}'}$ such that $R^{\mathcal{I}'}(x', y'_l)$ and $Z_j(y_l, y'_l)$ hold for every

$$1 \leq l \leq h \tag{5}$$

if $Z_{j+1}(x, x')$ holds and $y'_1, \ldots, y'_h$ are pairwise different elements of $\Delta^{\mathcal{I}'}$ such that $R^{\mathcal{I}'}(x', y'_l)$ holds for every $1 \leq l \leq h$, then there exist pairwise different elements $y_1, \ldots, y_h$ of $\Delta^{\mathcal{I}}$ such that $R^{\mathcal{I}}(x, y_l)$ and

$$Z_j(y_l, y'_l) \text{ hold for every} 1 \leq l \leq h, \tag{6}$$

if $\mathsf{Self} \in \Phi^\dagger$, then

$$Z_0(x, x') \Rightarrow [r^{\mathcal{I}}(x, x) \Leftrightarrow r^{\mathcal{I}'}(x', x')]. \tag{7}$$

**Lemma 4.1** *Let $\mathcal{I}$, $\mathcal{I}'$, and $\mathcal{I}''$ be interpretations.*

1. *The relation $\{\langle x, x\rangle \mid x \in \Delta^{\mathcal{I}}\}$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between $\mathcal{I}$ and $\mathcal{I}$.*
2. *If $Z$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$, then $Z^{-1}$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between $\mathcal{I}'$ and $\mathcal{I}$.*
3. *If $Z_1$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$, and $Z_2$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between $\mathcal{I}'$ and $\mathcal{I}''$, then $Z_1 \circ Z_2$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between $\mathcal{I}$ and $\mathcal{I}''$.*
4. *If $\mathcal{Z}$ is a set of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulations between $\mathcal{I}$ and $\mathcal{I}'$, then $\bigcup \mathcal{Z}$ is also an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$.*

The proof of this lemma is straightforward.

An interpretation $\mathcal{I}$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-*bisimilar* to $\mathcal{I}'$ if there exists an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between them. By Lemma 4.1, this $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimilarity relation is an equivalence relation between interpretations. We say that $x \in \Delta^{\mathcal{I}}$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-*bisimilar* to $x' \in \Delta^{\mathcal{I}'}$ if there exists an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation $Z_d$ between $\mathcal{I}$ and $\mathcal{I}'$ such that $Z_d(x, x')$ holds. This latter $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimilarity relation is also an equivalence relation (between elements of interpretations' domains).

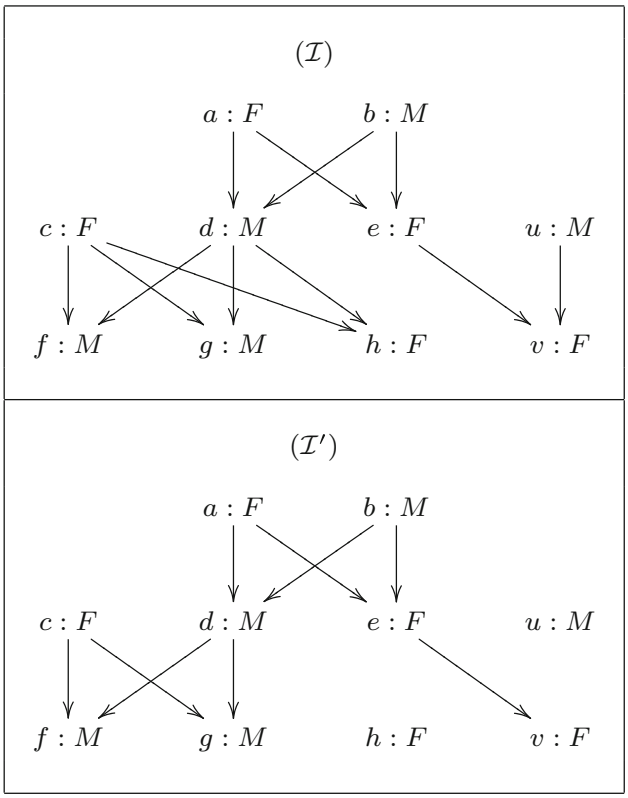



**Fig. 2** Illustration for Example 4.2

*Example 4.2* Let $\Sigma$ be the signature and $\mathcal{I}$ be the interpretation specified in Example 2.1. This interpretation is illustrated in Fig. 2 together with its modification $\mathcal{I}'$, which differs from $\mathcal{I}$ in that: $hasChild^{\mathcal{I}'}$ consists of only the elements illustrated by the edges shown at the lower part of Fig. 2 and $hasParent^{\mathcal{I}'}$, $Mother^{\mathcal{I}'}$, and $Father^{\mathcal{I}'}$ are defined accordingly.

Let $\Phi = \{I, Q_2, Q_3, \mathsf{Self}\}$ (we add $Q_2$ to $\Phi$ just for convenience), $\Sigma_I^\dagger = \Sigma_I$, $\Sigma_C^\dagger = \{Male\}$, and $\Sigma_R^\dagger = \{hasChild\}$. Consider the following cases.

- The interpretations $\mathcal{I}$ and $\mathcal{I}'$ are $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,0}$-bisimilar (with respect to any $\Phi^\dagger \subseteq \Phi$).
- Case $\Phi^\dagger \subseteq \{Q_2, \mathsf{Self}\}$: $\mathcal{I}$ and $\mathcal{I}'$ are $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimilar (with respect to any $d$).
- Case $I \in \Phi^\dagger$ and $d \geq 1$: $\mathcal{I}$ and $\mathcal{I}'$ are not $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimilar, because $Helen^{\mathcal{I}}$ ($h$ in $\mathcal{I}$) is not $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimilar to $Helen^{\mathcal{I}'}$ ($h$ in $\mathcal{I}'$).
- Case $Q_3 \in \Phi^\dagger$ and $d \geq 1$: $\mathcal{I}$ and $\mathcal{I}'$ are not $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimilar, because $Claudia^{\mathcal{I}}$ ($c$ in $\mathcal{I}$) is not $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimilar to $Claudia^{\mathcal{I}'}$ ($c$ in $\mathcal{I}'$).

If $\Sigma_I^\dagger = \{Alice, Bob\}$ and $\Phi^\dagger \subseteq \Phi$, then $\mathcal{I}$ and $\mathcal{I}'$ are $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,1}$-bisimilar. □

**Lemma 4.3** *Let $Z_d$ be an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between interpretations $\mathcal{I}$ and $\mathcal{I}'$. Then, for every $x \in \Delta^{\mathcal{I}}$, every $x' \in \Delta^{\mathcal{I}'}$, and every concept $C$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$, it holds that*

$$Z_d(x, x') \Rightarrow [C^{\mathcal{I}}(x) \Leftrightarrow C^{\mathcal{I}'}(x')].$$

*Proof* We prove this lemma by induction on the structure of $C$. Let $x \in \Delta^{\mathcal{I}}$, $x' \in \Delta^{\mathcal{I}'}$, $C$ be a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ and suppose $Z_d(x, x')$ and $C^{\mathcal{I}}(x)$ hold. We show that $C^{\mathcal{I}'}(x')$ also holds.

- The cases when $C$ is of the form $\top$, $\bot$, $A$, $\neg D$, $D \sqcup D'$ or $D \sqcap D'$ are trivial.
- Case $C = \exists R.D$, where $R$ is a role of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ and $D$ is a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d-1}$: Since $C^{\mathcal{I}}(x)$ holds, there exists $y \in \Delta^{\mathcal{I}}$ such that $R^{\mathcal{I}}(x, y)$ and $D^{\mathcal{I}}(y)$ hold. By the assertion (3), there exists $y' \in \Delta^{\mathcal{I}'}$ such that $Z_{d-1}(y, y')$ and $R^{\mathcal{I}'}(x', y')$ hold. By the induction assumption, it follows that $D^{\mathcal{I}'}(y')$ holds. Since $R^{\mathcal{I}'}(x', y')$ and $D^{\mathcal{I}'}(y')$ hold, it follows that $C^{\mathcal{I}'}(x')$ holds.
- Case $C = \forall R.D$, where $R$ is a role of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ and $D$ is a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d-1}$, is reduced to the above case, treating $\forall R.D$ as $\neg\exists R.\neg D$.
- Case $Q_k \in \Phi^\dagger$ and $C = (\geq h\, R.D)$, where $0 \leq h \leq k$, $R$ is a role of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, and $D$ is a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d-1}$: Since $C^{\mathcal{I}}(x)$ holds, there exist pairwise different $y_1, \ldots, y_h \in \Delta^{\mathcal{I}}$ such that $R^{\mathcal{I}}(x, y_i)$ and $D^{\mathcal{I}}(y_i)$ hold for all $1 \leq i \leq h$. Since $Z_d(x, x')$ holds, by the assertion (5), there exist pairwise different $y'_1, \ldots, y'_h \in \Delta^{\mathcal{I}'}$ such that $R^{\mathcal{I}'}(x', y'_i)$ and $Z_{d-1}(y_i, y'_i)$ hold for all $1 \leq i \leq h$. Since $Z_{d-1}(y_i, y'_i)$ and $D^{\mathcal{I}}(y_i)$ hold for every $1 \leq i \leq h$, by the induction assumption, it follows that $D^{\mathcal{I}'}(y'_i)$ holds for every $1 \leq i \leq h$. Therefore, $C^{\mathcal{I}'}(x')$ holds.
- Case $Q_k \in \Phi$ and $C = (< h\, R.D)$, where $0 \leq h \leq k$, $R$ is a role of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, and $D$ is a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d-1}$: this case is reduced to the above case, treating $< h\, R.D$ as $\neg(\geq h\, R.D)$.
- Case $\mathsf{Self} \in \Phi^\dagger$ and $C = \exists r.\mathsf{Self}$: since $C^{\mathcal{I}}(x)$ holds, we have that $r^{\mathcal{I}}(x, x)$ holds. Since $Z_d(x, x')$ holds and $Z_d \subseteq Z_0$, it follows that $Z_0(x, x')$ holds. By the assertion (7), we have that $r^{\mathcal{I}'}(x', x')$ holds. Hence, $C^{\mathcal{I}'}(x')$ holds.

An interpretation $\mathcal{I}$ over $\Sigma$ is *finitely branching* (or *image-finite*) with respect to $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ and $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ if, for every $x \in \Delta^{\mathcal{I}}$ and every role $R$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, the set $\{y \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(x, y)\}$ is finite.

Let $x \in \Delta^{\mathcal{I}}$ and $x' \in \Delta^{\mathcal{I}'}$. We say that $x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-*equivalent* to $x'$ if, for every concept $C$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$, $x \in C^{\mathcal{I}}$ iff $x' \in C^{\mathcal{I}'}$.

**Theorem 4.4** (The Hennessy–Milner Property) *Let $d$ be a natural number, $\Sigma$ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$, and $\Phi$ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$. Let $\mathcal{I}$ and $\mathcal{I}'$ be interpretations in $\mathcal{L}_{\Sigma,\Phi}$, finitely branching with respect to $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ such that, for every $a \in \Sigma_I^\dagger$, $a^{\mathcal{I}}$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-equivalent to $a^{\mathcal{I}'}$. Then, $x \in \Delta^{\mathcal{I}}$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-equivalent to $x' \in \Delta^{\mathcal{I}'}$ iff there exists an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation $Z_d$ between $\mathcal{I}$ and $\mathcal{I}'$ such that $Z_d(x, x')$ holds. In particular, the relation $\{\langle x, x'\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}'} \mid x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-equivalent to $x'\}$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$.*

*Proof* Consider the "⇐" direction. Suppose $Z_d$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$ such that $Z_d(x, x')$ holds. By Lemma 4.3, for every concept $C$ in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$, $C^{\mathcal{I}}(x)$ holds iff $C^{\mathcal{I}'}(x')$ holds. Therefore, $x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-equivalent to $x'$.

Now, consider the "⇒" direction. Define $Z_j = \{\langle x, x'\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}'} \mid x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j}$-equivalent to $x'\}$ for every $1 \leq j \leq d$. We show that $Z_d$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between $\mathcal{I}$ and $\mathcal{I}'$.

- The condition (1) follows from the assumption of the theorem.
- Consider the condition (2) and suppose $Z_0(x, x')$ holds. By the definition of $Z_0$, it follows that $x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,0}$-equivalent to $x'$. Therefore, for every concept name $A$, $A^{\mathcal{I}}(x)$ holds iff $A^{\mathcal{I}'}(x')$ holds.
- Consider the condition (3) and suppose $Z_{j+1}(x, x')$ and $R^{\mathcal{I}}(x, y)$ hold. Thus, $x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j+1}$-equivalent to $x'$. Let $S = \{y' \in \Delta^{\mathcal{I}'} \mid R^{\mathcal{I}'}(x', y')\}$. We show that there exists $y' \in S$ such that $Z_j(y, y')$ holds. Since $x \in (\exists R.\top)^{\mathcal{I}}$ and $x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j+1}$-equivalent to $x'$, we have that $x' \in (\exists R.\top)^{\mathcal{I}'}$. Hence, $S \neq \emptyset$. Since $\mathcal{I}'$ is finitely branching, $S$ must be finite. Let the elements of $S$ be $y'_1, \ldots, y'_n$. For the sake of contradiction, suppose that, for every $1 \leq i \leq n$, $Z_j(y, y'_i)$ does not hold, which means that $y$ is not $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j}$-equivalent to $y'_i$. Thus, for every $1 \leq i \leq n$, there exists a concept $C_i$ in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j}$ such that $C_i^{\mathcal{I}}(y)$ holds, but $C_i^{\mathcal{I}'}(y'_i)$ does not. Let $C = \exists R.(C_1 \sqcap \cdots \sqcap C_n)$. Thus, $C$ is a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j+1}$ and $C^{\mathcal{I}}(x)$ holds, but $C^{\mathcal{I}'}(x')$ does not, which contradicts the fact that $x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j+1}$-equivalent to $x'$. Therefore, there exists $y'_i \in S$ such that $Z_j(y, y'_i)$ holds.
- The condition (4) can be proved analogously as for the condition (3).
- Consider the case $Q_k \in \Phi^\dagger$ and the conditions (5) and (6). Suppose $Z_{j+1}(x, x')$ holds. Thus, $x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j+1}$-equivalent to $x'$. Let $S = \{y \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(x, y)\}$ and $S' = \{y' \in \Delta^{\mathcal{I}'} \mid R^{\mathcal{I}'}(x', y')\}$. Since $\mathcal{I}$ and $\mathcal{I}'$ are finitely branching, both $S$ and $S'$ are finite. Consider an arbitrary

$y'' \in S \cup S'$ and let $y_1, \ldots, y_n \in S$ and $y'_1, \ldots, y'_{n'} \in S'$ be all the pairwise different elements that are $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j}$-equivalent to $y''$. To prove (5) and (6), it suffices to show that either $n = n'$ or ($n \geq k$ and $n' \geq k$). For the sake of contrary, assume that $n \neq n'$ and ($n < k$ or $n' < k$). Without loss of generality, also assume that $n < n'$. Thus, $n < k$ and $n + 1 \leq k$. Let $\{t_1, \ldots, t_m\} = S \backslash \{y_1, \ldots, y_n\}$ and $\{t'_1, \ldots, t'_{m'}\} = S' \backslash \{y'_1, \ldots, y'_{n'}\}$. Let $\mathcal{I}'' = \mathcal{I}$ if $y'' \in S$, and let $\mathcal{I}'' = \mathcal{I}'$ otherwise. For each $1 \leq i \leq m$, there exists a concept $D_i$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j}$ such that $y'' \in D_i^{\mathcal{I}''}$ but $t_i \notin D_i^{\mathcal{I}}$. Similarly, for each $1 \leq i \leq m'$, there exists a concept $D'_i$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j}$ such that $y'' \in (D'_i)^{\mathcal{I}''}$ but $t'_i \notin (D'_i)^{\mathcal{I}'}$. Let $D = (D_1 \sqcap \cdots \sqcap D_m \sqcap D'_1 \sqcap \cdots \sqcap D'_{m'})$. We have that $\{y_1, \ldots, y_n\} \subseteq D^{\mathcal{I}}$ (since $y'' \in D^{\mathcal{I}''}$) and $\{t_1, \ldots, t_m\} \cap D^{\mathcal{I}} = \emptyset$. Similarly, $\{y'_1, \ldots, y'_{n'}\} \subseteq D^{\mathcal{I}'}$ and $\{t'_1, \ldots, t'_{m'}\} \cap D^{\mathcal{I}'} = \emptyset$. Observe that $C = (\geq (n+1)\, R.D)$ is a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j+1}$, $C^{\mathcal{I}'}(x')$ holds, but $C^{\mathcal{I}}(x)$ does not. This contradicts the fact that $x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,j+1}$-equivalent to $x'$.

- Consider the case $\mathsf{Self} \in \Phi^\dagger$ and the assertion (7). Suppose $Z_0(x, x')$ holds. Thus, $x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,0}$-equivalent to $x'$. Let $C = \exists r.\mathsf{Self}$. Since $\mathsf{mdepth}(C) = 0$, it follows that $C^{\mathcal{I}}(x)$ holds iff $C^{\mathcal{I}'}(x')$ holds, which means $x \in (\exists r.\mathsf{Self})^{\mathcal{I}}$ iff $x' \in (\exists r.\mathsf{Self})^{\mathcal{I}'}$. Therefore, $r^{\mathcal{I}}(x, x)$ holds iff $r^{\mathcal{I}'}(x', x')$ holds. □

An $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-bisimulation between $\mathcal{I}$ and itself is called an *$\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-auto-bisimulation of $\mathcal{I}$*. An $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-auto-bisimulation of $\mathcal{I}$ is said to be the *largest* if it is larger than or equal to ($\supseteq$) any other $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-auto-bisimulation of $\mathcal{I}$.

Given an interpretation $\mathcal{I}$ over $\Sigma$, by $\sim_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ we denote the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-auto-bisimulation of $\mathcal{I}$, and by $\equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ we denote the binary relation on $\Delta^{\mathcal{I}}$ with the property that $x \equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}} x'$ iff $x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-equivalent to $x'$.

**Theorem 4.5** *Let $d$ be a natural number, $\Sigma$ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$, $\Phi$ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$, and $\mathcal{I}$ be an interpretation over $\Sigma$. Then, the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-auto-bisimulation of $\mathcal{I}$ exists and is an equivalence relation. Furthermore, if $\mathcal{I}$ is finitely branching with respect to $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, then the relation $\equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ is the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-auto-bisimulation of $\mathcal{I}$ (i.e., the relations $\equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ and $\sim_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ coincide).*

*Proof* It follows from Lemma 4.1 that the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-auto-bisimulation of $\mathcal{I}$ exists and is an equivalence relation. Assume that $\mathcal{I}$ is finitely branching with respect to $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$. By Theorem 4.4, the relation $\equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-auto-bisimulation of $\mathcal{I}$. It remains to show that this $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-auto-bisimulation is the largest one. Suppose $Z_d$ is another $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-auto-bisimulation of $\mathcal{I}$. If $Z_d(x, x')$ holds then, by Lemma 4.3, for every concept $C$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$, $C^{\mathcal{I}}(x)$ holds iff $C^{\mathcal{I}'}(x')$ holds, and hence $x \equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}} x'$. Therefore, $Z_d \subseteq \equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$. □

We say that a set $Y$ is *split* by a set $X$ if $Y \backslash X \neq \emptyset$ and $Y \cap X \neq \emptyset$. Thus, $Y$ is not split by $X$ if either $Y \subseteq X$ or $Y \cap X = \emptyset$. A partition $P = \{Y_1, \ldots, Y_n\}$ is *consistent* with a set $X$ if, for every $1 \leq i \leq n$, $Y_i$ is not split by $X$.

The following theorem is similar to the ones given in [13, 26, 31]. The difference is that it deals with a different class of languages, which contain concepts with bounded modal depth.

**Theorem 4.6** *Let $d$ be a natural number, $\Sigma$ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$, $\Phi$ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$, $\mathcal{I}$ be a finitely branching interpretation with respect to $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, and let $X \subseteq \Delta^{\mathcal{I}}$. Then:*

1. *If there exists a concept $C$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ such that $X = C^{\mathcal{I}}$, then the partition of $\Delta^{\mathcal{I}}$ by $\sim_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ is consistent with $X$.*
2. *If the partition of $\Delta^{\mathcal{I}}$ by $\sim_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ is consistent with $X$, then there exists a concept $C$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ such that $C^{\mathcal{I}} = X$.*

*Proof* By Theorem 4.5, $\sim_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ coincides with $\equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$.

Consider the first assertion and assume that $X = C^{\mathcal{I}}$ for some concept $C$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$. Let $Y$ be any element of the partition of $\Delta^{\mathcal{I}}$ by $\sim_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ such that $X \cap Y \neq \emptyset$. It suffices to show that $Y \subseteq X$. Let $x$ be an arbitrary element of $Y$. Since $X \cap Y \neq \emptyset$, there exists $x' \in X \cap Y$. Since both $x$ and $x'$ belong to $Y$, $x' \sim_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}} x$. Since $\sim_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ coincides with $\equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$, we also have that $x' \equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}} x$. Since $x' \in X$ and $X = C^{\mathcal{I}}$, $C^{\mathcal{I}}(x')$ holds, which together with $x' \equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}} x$ implies that $C^{\mathcal{I}}(x)$ holds. Thus, $x \in X$ and we can conclude that $Y \subseteq X$.

Consider the second assertion and assume that the partition of $\Delta^{\mathcal{I}}$ by $\sim_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ is consistent with $X$. Let that partition be $\{Y_1, \ldots, Y_m, Y'_1, \ldots, Y'_n\}$, where $Y_i \subseteq X$ for all $1 \leq i \leq m$ and $Y'_j \cap X = \emptyset$ for all $1 \leq j \leq n$. We have that $X = Y_1 \cup \cdots \cup Y_m$. For each $1 \leq i \leq m$ and $1 \leq j \leq n$, since $Y_i$ and $Y'_j$ are different equivalence classes of $\equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ (the same as $\sim_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$), there exists a concept $C_{i,j}$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ such that $Y_i \subseteq C_{i,j}^{\mathcal{I}}$ and $Y'_j \cap C_{i,j}^{\mathcal{I}} = \emptyset$. For each $1 \leq i \leq m$, let $C_i = C_{i,1} \sqcap \cdots \sqcap C_{i,n}$. Thus, $Y_i \subseteq C_i^{\mathcal{I}}$ and $Y'_j \cap C_i^{\mathcal{I}} = \emptyset$ for all $1 \leq j \leq n$. Let $C = C_1 \sqcup \cdots \sqcup C_m$. Thus, $Y_i \subseteq C^{\mathcal{I}}$ for all $1 \leq i \leq m$ and $Y'_j \cap C^{\mathcal{I}} = \emptyset$ for all $1 \leq j \leq n$. Therefore, $C^{\mathcal{I}} = X$. □

## 5 A concept learning algorithm

Let $A_0 \in \Sigma_C$ be a concept name standing for the "decision attribute" and suppose that $A_0$ can be expressed by a concept $C$ in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, where $\Sigma^\dagger \subseteq \Sigma\backslash\{A_0\}$ and $\Phi^\dagger \subseteq \Phi$. Let $\mathcal{I}$ be a training information system over $\Sigma$. How can we learn that concept $C$ on the basis of $\mathcal{I}$? Nguyen and Szałas [26] gave a bisimulation-based method for this learning problem. In this section, by adopting a specific strategy, we present a modified version of that method, called the MIMOD (minimizing modal depth) concept learning algorithm. This algorithm is used for analyzing C-learnability in the next section.

Our MiMoD algorithm is as follows:

1. Starting from the partition $\{\Delta^{\mathcal{I}}\}$, make subsequent granulations to reach a partition consistent with $A_0^{\mathcal{I}}$. In the granulation process, we denote the blocks created so far in all steps by $Y_1, \ldots, Y_n$, where the current partition may consist of only some of them. We do not use the same subscript to denote blocks of different contents (i.e., we always use new subscripts obtained by increasing $n$ for new blocks). We take care that, for each $1 \le i \le n$, $Y_i$ is characterized by a concept $C_i$ such that $Y_i = C_i^{\mathcal{I}}$.
2. We use the following concepts as *selectors* for the granulation process, where $1 \le i \le n$:
   (a) $A$, where $A \in \Sigma_C^\dagger$,
   (b) $\exists r.\mathsf{Self}$, if $\mathsf{Self} \in \Phi^\dagger$ and $r \in \Sigma_R^\dagger$,
   (c) $\exists r.C_i$, where $r \in \Sigma_R^\dagger$,
   (d) $\exists r^-.C_i$, if $I \in \Phi^\dagger$ and $r \in \Sigma_R^\dagger$,
   (e) $\geq h\, r.C_i$, if $Q_k \in \Phi^\dagger$, $r \in \Sigma_R^\dagger$, and $1 \le h \le k$,
   (f) $\geq h\, r^-.C_i$, if $\{Q_k, I\} \subseteq \Phi^\dagger$, $r \in \Sigma_R^\dagger$ and $1 \le h \le k$.

   A selector $D$ has a *higher priority* than $D'$ if $\mathsf{mdepth}(D) < \mathsf{mdepth}(D')$.
3. During the granulation process if
   - a block $Y_i$ of the current partition is split by $D^{\mathcal{I}}$, where $D$ is a selector, and
   - there do not exist a block $Y_j$ of the current partition and a selector $D'$ with a higher priority than $D$ such that $Y_j$ is split by $D'$,

   then split $Y_i$ by $D$ as follows:
   - $s := n+1,\ t := n+2,\ n := n+2$,
   - $Y_s := Y_i \cap D^{\mathcal{I}},\ C_s := C_i \sqcap D$,
   - $Y_t := Y_i \cap (\neg D)^{\mathcal{I}},\ C_t := C_i \sqcap \neg D$,
   - replace $Y_i$ in the current partition by $Y_s$ and $Y_t$.
4. When the current partition becomes consistent with $A_0^{\mathcal{I}}$, return $C_{i_1} \sqcup \cdots \sqcup C_{i_j}$, where $i_1, \ldots, i_j$ are indices such that $Y_{i_1}, \ldots, Y_{i_j}$ are all the blocks of the current partition that are subsets of $A_0^{\mathcal{I}}$.

Observe that the above algorithm always terminates.

*Example 5.1* Consider the information system $\mathcal{I}$ given in Example 2.1. Let $\Sigma^\dagger = \{Male, hasChild\}$ and $\Phi^\dagger = \emptyset$. We want to apply the MiMoD algorithm to learn a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ that describes the concept *Father*. Recall that $Father^{\mathcal{I}} = \{b, d, u\}$. One of possible runs of the algorithm is as follows:

1. $Y_1 := \Delta^{\mathcal{I}}$, $C_1 = \top$, $partition := \{Y_1\}$.
2. Splitting $Y_1$ by *Male*:
   - $Y_2 := \{b, d, f, g, u\}$, $C_2 := Male$.
   - $Y_3 := \{a, c, e, h, v\}$, $C_3 := \neg Male$.
   - $partition := \{Y_2, Y_3\}$.
3. Splitting $Y_2$ by $\exists hasChild.\top$:
   - $Y_4 := \{b, d, u\}$, $C_4 := C_2 \sqcap \exists hasChild.\top$.
   - $Y_5 := \{f, g\}$, $C_5 := C_2 \sqcap \neg\exists hasChild.\top$.
   - $partition := \{Y_3, Y_4, Y_5\}$.

The obtained partition is consistent with $Father^{\mathcal{I}}$, having $Y_4 = Father^{\mathcal{I}}$ and $Y_3$, $Y_5$ disjoint with $Father^{\mathcal{I}}$. The returned concept is $C_4 = Male \sqcap \exists hasChild.\top$. □

*Example 5.2* Consider once again the information system $\mathcal{I}$ given in Example 2.1. Now, let $\Sigma^\dagger = \{Male, hasChild\}$, $\Phi^\dagger = \{Q_3\}$ and let $A_0$ be a new concept name interpreted in $\mathcal{I}$ as $A_0^{\mathcal{I}} = \{c, d\}$. We want to apply the MiMoD algorithm to learn a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ that describes $A_0$. One of possible runs of the algorithm has first two steps as in Example 5.1 and then continues as follows:

3. Splitting $Y_2$ by $\geq 3\, hasChild.\top$:
   - $Y_4 := \{d\}$, $C_4 := C_2 \sqcap (\geq 3\, hasChild.\top)$.
   - $Y_5 := \{b, f, g, u\}$. $C_5 := C_2 \sqcap \neg(\geq 3\, hasChild.\top)$,
   - $partition := \{Y_3, Y_4, Y_5\}$.
4. Splitting $Y_3$ by $\geq 3\, hasChild.\top$:
   - $Y_6 := \{c\}$, $C_6 := C_3 \sqcap (\geq 3\, hasChild.\top)$.
   - $Y_7 := \{a, e, h, v\}$, $C_7 := C_3 \sqcap \neg(\geq 3\, hasChild.\top)$.
   - $partition := \{Y_4, Y_5, Y_6, Y_7\}$.

The obtained partition is consistent with $A_0^{\mathcal{I}}$, having $Y_4 \subset A_0^{\mathcal{I}}$, $Y_6 \subset A_0^{\mathcal{I}}$, and $Y_5$, $Y_7$ disjoint with $A_0^{\mathcal{I}}$. The returned concept is

$$C_4 \sqcup C_6 = [Male \sqcap (\geq 3\, hasChild.\top)] \sqcup [\neg Male \sqcap (\geq 3\, hasChild.\top)]$$

which is equivalent to $\geq 3\, hasChild.\top$. □

For experimental results on the usefulness of the bisimulation-based concept learning method (not necessarily

MiMoD), we refer the reader to [29]. Here, we are interested in the following lemma.

**Lemma 5.3** *Let $\Sigma$ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$, $\Phi$ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$, and $\mathcal{I}$ be an interpretation over $\Sigma$. Suppose $A_0 \in \Sigma_C \backslash \Sigma_C^\dagger$ and $C$ is a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ such that $A_0^\mathcal{I} = C^\mathcal{I}$. Let $C'$ be a concept returned by the MiMoD algorithm for $\mathcal{I}$. Then, $C'$ is a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ such that $C'^\mathcal{I} = C^\mathcal{I}$ and $\mathsf{mdepth}(C') \leq \mathsf{mdepth}(C)$.*

*Proof* Clearly, $C'^\mathcal{I} = A_0^\mathcal{I} = C^\mathcal{I}$. Consider the execution of the MiMoD algorithm on $\mathcal{I}$ that results in $C'$. By $\mathcal{P}_d$ we denote the partition of $\Delta^\mathcal{I}$ at the moment in that execution when $\max\{\mathsf{mdepth}(C_i) \mid Y_i \in \mathcal{P}_d\} = d$ and $\mathcal{P}_d$ cannot be granulated any more without using some selector with modal depth $d+1$. Let $d_{\max}$ be the maximal value of such an index $d$ (of some $\mathcal{P}_d$). Let $Z_d$ be the equivalence relation corresponding to the partition $\mathcal{P}_d$, i.e., $Z_d = \{\langle x, x'\rangle \mid x, x' \in Y_i$ for some $Y_i \in \mathcal{P}_d\}$. It is straightforward to prove by induction on $d$ that $Z_d$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$-auto-bisimulation of $\mathcal{I}$. Hence, $Z_d \subseteq \sim_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$. Since each block of $\mathcal{P}_d$ is characterized by a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$, $Z_d$ is a superset of $\equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$. Since $\equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ and $\sim_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$ coincide (Theorem 4.5), we have that $Z_d = \equiv_{\Sigma^\dagger,\Phi^\dagger,d,\mathcal{I}}$.

Since the algorithm terminates as soon as the current partition is consistent with $C^\mathcal{I}$, it follows that $d_{\max} \leq \mathsf{mdepth}(C)$. Furthermore, if $d_{\max} < \mathsf{mdepth}(C')$, then we also have $d_{\max} < \mathsf{mdepth}(C)$. Since $\mathsf{mdepth}(C') \leq d_{\max} + 1$, we conclude that $\mathsf{mdepth}(C') \leq \mathsf{mdepth}(C)$. □

## 6 C-learnability in description logics

**Theorem 6.1** *Let $d$ be a natural number, $\Sigma$ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$, $\Phi$ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$, and $\mathcal{I}$ be a finite universal interpretation with respect to $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$. Suppose $A_0 \in \Sigma_C \backslash \Sigma_C^\dagger$ and $C$ is a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ such that $A_0^\mathcal{I} = C^\mathcal{I}$. Then, any concept returned by the MiMoD algorithm for $\mathcal{I}$ is equivalent to $C$.*

*Proof* Let $C'$ be a concept returned by the MiMoD algorithm for $\mathcal{I}$. By Lemma 5.3, $C'^\mathcal{I} = C^\mathcal{I}$ and $\mathsf{mdepth}(C') \leq \mathsf{mdepth}(C)$. For the sake of contradiction, suppose $C'$ is not equivalent to $C$. Thus, either $C \sqcap \neg C'$ or $C' \sqcap \neg C$ is satisfiable. Both of them belong to $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$. Since $\mathcal{I}$ is universal with respect to $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$, it follows that either $(C \sqcap \neg C')^\mathcal{I}$ or $(C' \sqcap \neg C)^\mathcal{I}$ is not empty, which contradicts the fact that $C'^\mathcal{I} = C^\mathcal{I}$. □

**Theorem 6.2** *For every concept $C$ in any description logic that extends $\mathcal{ALC}$ with some features amongst $I$, $Q_k$, $\mathsf{Self}$, there exists a training information system such that applying the MiMoD algorithm to it results in a concept equivalent to $C$.*

*Proof* Let the considered logic be $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ and let $d = \mathsf{mdepth}(C)$, $\Phi = \Phi^\dagger$ and $\Sigma = \Sigma^\dagger \cup \{A_0\}$, where $A_0 \notin \Sigma_C^\dagger$. By Lemma 3.3, there exists a finite universal interpretation $\mathcal{I}'$ with respect to $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$. Let $\mathcal{I}$ be the interpretation over $\Sigma$ different from $\mathcal{I}'$ only in that $A_0^\mathcal{I}$ is defined to be $C^{\mathcal{I}'}$. Clearly, $\mathcal{I}$ is universal with respect to $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ and $A_0^\mathcal{I} = C^\mathcal{I}$. By Theorem 6.1, any concept returned by the MiMoD algorithm for $\mathcal{I}$ is equivalent to $C$. □

Assuming that the language $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ is fixed, the MiMoD algorithm in the above two theorems for learning a concept $C$ does not depend on $C$ (nor the modal depth of $C$). Furthermore, the training information system $\mathcal{I}$ used for learning $C$ depends on $C$ only via its modal depth.

## 7 On concept learning using queries

Angluin [2] assumed that the learner has access to a fixed set of oracles that will answer specific kinds of queries about the concept to be learned. As mentioned earlier, she studied exact and probably exact learnability using different types of queries, such as membership, equivalence, subset, superset, disjointness, and exhaustiveness. In this section, we generalize these types of queries for DLs, introduce interpretation queries, and present some consequences. This mainly serves as a starting point for future work.

A type of queries is specified by a form of inputs and outputs for oracles. Let $C$ denote the concept to be learned, which belongs to a language $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$. It is known to the oracles, but unknown to the learner. We assume that the learner knows $\Sigma^\dagger$ and whether $\Phi^\dagger$ contains $I$ or $\mathsf{Self}$, but it may not know $d$ nor the (maximal) number $k$ with $Q_k \in \Phi^\dagger$. Generalization of the types of queries studied by Angluin [2] is as follows.

- *Membership* The input is a pair of an interpretation $\mathcal{I}$ and an element $x \in \Delta^\mathcal{I}$, and the output is *yes* if $x \in C^\mathcal{I}$ and *no* otherwise.
- *Equivalence* The input is a concept $D$ and the output is *yes* if $D \equiv C$ and *no* otherwise. If the answer is *no*, the oracle returns an interpretation $\mathcal{I}$ and an element $x \in D^\mathcal{I} \ominus C^\mathcal{I}$, where $\ominus$ denotes "symmetric difference".
- *Subset* The input is a concept $D$ and the output is *yes* if $D \sqsubseteq C$ (i.e., $D^\mathcal{I} \subseteq C^\mathcal{I}$ for every interpretation $\mathcal{I}$) and *no* otherwise. If the answer is *no*, the oracle returns an interpretation $\mathcal{I}$ and an element $x \in D^\mathcal{I} - C^\mathcal{I}$.
- *Superset* The input is a concept $D$ and the output is *yes* if $C \sqsubseteq D$ and *no* otherwise. If the answer is *no*, the oracle returns an interpretation $\mathcal{I}$ and an element $x \in C^\mathcal{I} - D^\mathcal{I}$.

- *Disjointness* The input is a concept $D$ and the output is *yes* if $D \sqcap C$ is unsatisfiable and *no* otherwise. If the answer is *no*, the oracle returns an interpretation $\mathcal{I}$ and an element $x \in D^{\mathcal{I}} \cap C^{\mathcal{I}}$.
- *Exhaustiveness* The input is a concept $D$ and the output is *yes* if $D \sqcup C \equiv \top$ (i.e., $D^{\mathcal{I}} \cup C^{\mathcal{I}} = \Delta^{\mathcal{I}}$ for every interpretation $\mathcal{I}$) and *no* otherwise. If the answer is *no*, the oracle returns an interpretation $\mathcal{I}$ and an element $x \notin D^{\mathcal{I}} \cup C^{\mathcal{I}}$.

The input concept $D$ is usually assumed to belong to the same language as $C$. In the *restricted* version, the above oracles return only *yes* or *no* without providing a counter example $x$.

Valiant [32] studied concept learnability using membership queries and oracles that generate positive examples. One can also consider oracles that generate negative examples. These oracles do not receive inputs, but only return examples. They are generalized for DLs as follows:

- *Positive example* The output is a pair of an interpretation $\mathcal{I}$ and an element $x \in C^{\mathcal{I}}$.
- *Negative example* The output is a pair of an interpretation $\mathcal{I}$ and an element $x \in \Delta^{\mathcal{I}} - C^{\mathcal{I}}$.

Our new type of queries is as follows, which generalizes membership queries.

- *Interpretation* The input is an interpretation $\mathcal{I}$ and the output is the set $C^{\mathcal{I}}$.

As a consequence of Theorem 6.1, we have the following corollary:

**Corollary 7.1** *If $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ is known, then each of its concepts can be learned using one interpretation query.*

We say that a concept $C$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ is in the $h$-DEG normal form (in short, $h$-DEGNF) if it is in the DEG normal form and

- every conjunction occurring in $C$ has no more than $h$ conjuncts,
- if $C$ is a disjunction, then it has no more than $h$ disjuncts.

In the case $\Phi^\dagger = \{I, Q_k, \mathsf{Self}\}$, $|\Sigma_C^\dagger| = m$ and $|\Sigma_R^\dagger| = n$, an upper bound $S(d)$ for the number of concepts in the $h$-DEG normal form of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ can be estimated as follows:

$$\begin{aligned} S'(0) &= (2m + 2n + 2)^h \\ S'(l+1) &= (4k.n.S'(l) + 2m + 2n + 2)^h \quad \text{for } l \geq 0 \\ S(d) &= (S'(d))^h, \end{aligned}$$

where $S'(l)$ is an upper bound for the number of concepts in the $h$-DEG normal form of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ that do not use $\sqcup$ and have a modal depth not greater than $l$.

Thus, $S(d) = (O(k.n.(m+n)^h))^{h^{d+1}}$. In the case when $h$ and $d$ are constants, $S(d)$ is a polynomial (in $k, m$, and $n$). We arrive at the following consequence, which is related to the learnability of bounded CNF boolean formulas in classical propositional calculus [2,32].

**Proposition 7.2** *When $h$ and $d$ are fixed natural numbers, every concept $C$ in the $h$-DEG normal form of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ can be learned using a polynomial number of equivalence queries.*

## 8 Concluding remarks

We have proved that any concept in any description logic that extends $\mathcal{ALC}$ with some features amongst $I$ (inverse roles), $Q_k$ (qualified number restrictions with numbers bounded by a constant $k$), and Self (local reflexivity of a role) can be learned if the training information system (specified as a finite interpretation) is good enough. This is an interesting theoretical result, which requires advanced techniques. In particular, for this theorem, we have introduced universal interpretations and bounded bisimulation in DLs and developed the MiMoD algorithm.

The considered DLs are quite expressive. At least, they are more expressive than C-CLASSIC, DL-Lite, and $\mathcal{EL}$. Our result on C-learnability is not extensible in a straightforward way for DLs with nominals or the universal role (the proof of Lemma 3.3 does not work for such an extension). It can be extended to deal also with the features $N_k$ (unqualified number restrictions with numbers bounded by a constant $k$), $F$ (role functionality), and reformulated for Setting 2 as done in [30]. A reformulation for Setting 1 can be done analogously.

As future work, we intend to study PAC and PExact concept learnability in DLs using queries. The formulations given in Sect. 7 are a starting point.

**Acknowledgements** This work was partially supported by VNU Grant QG-15-22.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

1. Alvarez, J.: A formal framework for theory learning using description logics. In: ILP'2000 Work-in-progress reports (2000)

2. Angluin, D.: Queries and concept learning. Mach. Learn. **2**(4), 319–342 (1987)
3. Badea, L., Nienhuys-Cheng, S.H.: A refinement operator for description logics. In: Proceedings of ILP'2000, vol. 1866 of LNCS, pp. 40–59. Springer, New York (2000)
4. Bshouty, N., Jackson, J., Tamon, C.: Exploring learnability between exact and PAC. J. Comput. Syst. Sci. **70**(4), 471–484 (2005)
5. Cohen, W., Hirsh, H.: Learning the Classic description logic: Theoretical and experimental results. In: Proceedings of KR'1994, pp. 121–133 (1994)
6. Distel, F.: Learning Description Logic Knowledge Bases from Data Using Methods from Formal Concept Analysis. Ph.D. thesis, Dresden University of Technology (2011)
7. Divroodi, A.: Bisimulation Equivalence in Description Logics and Its Applications. Ph.D. thesis, University of Warsaw (2015). http://www.mimuw.edu.pl/wiadomosci/aktualnosci/doktoraty/pliki/ali_rezaei_divroodi/ad-dok.pdf
8. Divroodi, A., Ha, Q.-T., Nguyen, L., Nguyen, H.: On C-learnability in description logics. In: Proceedings of ICCCI'2012 (1), vol. 7653 of LNCS, pp. 230–238. Springer, New York (2012)
9. Divroodi, A., Nguyen, L.: On bisimulations for description logics (2011). arXiv:1104.1964 [abs]
10. Divroodi, A., Nguyen, L.: On bisimulations for description logics. Inf. Sci. **295**, 465–493 (2015)
11. Fanizzi, N., d'Amato, C., Esposito, F.: DL-FOIL concept learning in description logics. In: Proceedings of ILP'2008, vol. 5194 of LNCS, pp. 107–121. Springer, New York (2008)
12. Frazier, M., Pitt, L.: Classic learning. Mach. Learn. **25**(2–3), 151–193 (1996)
13. Ha, Q.-T., Hoang, T.-L.-G., Nguyen, L., Nguyen, H., Szałas, A., Tran, T.-L.: A bisimulation-based method of concept learning for knowledge bases in description logics. In: Proceedings of SoICT'2012, pp. 241–249. ACM, New York (2012)
14. Hellmann, S.: Comparison of concept learning algorithms (with emphasis on ontology engineering for the Semantic Web). Master's thesis, Leipzig University (2008)
15. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Proc. of KR'2006, pp. 57–67. AAAI Press, Menlo Park (2006)
16. Iannone, L., Palmisano, I., Fanizzi, N.: An algorithm based on counterfactuals for concept learning in the Semantic Web. Appl. Intell. **26**(2), 139–159 (2007)
17. Kietz, J.: Learnability of description logic programs. In: Proceedings of ILP'2002, vol. 2583 of LNCS, pp. 117–132. Springer, New York (2002)
18. Konev, B., Lutz, C., Ozaki, A., Wolter, F.: Exact learning of lightweight description logic ontologies. In: Proceedings of KR'2014. AAAI Press, Menlo Park (2014)
19. Konstantopoulos, S., Charalambidis, A.: Formulating description logic learning as an inductive logic programming task. In: Proceedings of FUZZ-IEEE, pp. 1–7 (2010)
20. Lambrix, P., Larocchia, P.: Learning composite concepts. In: Proceedings of DL (1998)
21. Lehmann, J., Hitzler, P.: Concept learning in description logics using refinement operators. Mach. Learn. **78**(1–2), 203–250 (2010)
22. Lisi, F.: A formal characterization of concept learning in description logics. In: Proceedings of DL'2012, vol. 846 of CEUR Workshop Proceedings. CEUR-WS.org (2012)
23. Luna, J., Revoredo, K., Cozman, F.: Learning probabilistic description logics: A framework and algorithms. In: Proceedings of MICAI'2011, vol. 7094 of LNCS, pp. 28–39. Springer, New York (2011)
24. Nguyen, L.: An efficient tableau prover using global caching for the description logic $\mathcal{ALC}$. Fundam. Inf. **93**(1–3), 273–288 (2009)
25. Nguyen, L.: A tableau method with optimal complexity for deciding the description logic SHIQ. In: Advanced Computational Methods for Knowledge Engineering, vol. 479 of Studies in Computational Intelligence, pp. 331–342. Springer, New York (2013)
26. Nguyen, L., Szałas, A.: Logic-based roughification. In: Skowron, A., Suraj, Z. (eds) Rough Sets and Intelligent Systems (to the Memory of Professor Zdzisław Pawlak), vol. 1, pp. 529–556. Springer, New York (2012)
27. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
28. Raedt, L.D., Dzeroski, S.: First-order jk-clausal theories are PAC-learnable. Artif. Intell. **70**(1–2), 375–392 (1994)
29. Tran, T., Nguyen, L., Hoang, T.: Bisimulation-based concept learning for information systems in description logics. Vietnam J. Comput. Sci. **2**(3), 149–167 (2015)
30. Tran, T.-L., Ha, Q.-T., Hoang, T.-L.-G., Nguyen, L., Nguyen, H.: Bisimulation-based concept learning in description logics. Fundam. Inform. **133**(2–3), 287–303 (2014)
31. Tran, T.-L., Ha, Q.-T., Hoang, T.-L.-G., Nguyen, L., Nguyen, H., Szałas, A.: Concept learning for description logic-based information systems. In: Proceedings of KSE'2012, pp. 65–73. IEEE Computer Society, Washington (2012)
32. Valiant, L.G.: A theory of the learnable. Commun. ACM **27**(11), 1134–1142 (1984)