Fundamenta Informaticae 133 (2014) 287–303
DOI 10.3233/FI-2014-1077
IOS Press

Bisimulation-Based Concept Learning in Description Logics

Thanh-Luong Tran
Faculty of Information Technology
College of Sciences, Hue University
77 Nguyen Hue, Hue city, Vietnam

Quang-Thuy Ha
Faculty of Information Technology
VNU University of Engineering and Technology
144 Xuan Thuy, Hanoi, Vietnam

Thi-Lan-Giao Hoang
Faculty of Information Technology
College of Sciences, Hue University
77 Nguyen Hue, Hue city, Vietnam

Linh Anh Nguyen
Institute of Informatics
University of Warsaw
Banacha 2, 02-097 Warsaw, Poland

Hung Son Nguyen
Institute of Mathematics
University of Warsaw
Banacha 2, 02-097 Warsaw, Poland

Abstract. Concept learning in description logics (DLs) is similar to binary classification in traditional machine learning. The difference is that in DLs objects are described not only by attributes but also by binary relationships between objects. In this paper, we develop the first bisimulation-based method of concept learning in DLs for the following setting: given a knowledge base *KB* in a DL, a set of objects standing for positive examples and a set of objects standing for negative examples, learn a concept *C* in that DL such that the positive examples are instances of *C* w.r.t. *KB*, while the negative examples are not instances of *C* w.r.t. *KB*. We also prove soundness of our method and investigate its C-learnability.

1. Introduction

Concept learning in description logics (DLs) is similar to binary classification in traditional machine learning. The difference is that in DLs objects are described not only by attributes but also by binary relationships between objects. In ontology engineering, concept learning is helpful for suggesting important concepts and their definitions. The major settings of concept learning in DLs are as follows:

Address for correspondence: Institute of Mathematics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland

1. Given a knowledge base KB in a DL L and sets E^+, E^- of individuals, learn a concept C in L such that: (a) $KB \models C(a)$ for all $a \in E^+$, and (b) $KB \models \neg C(a)$ for all $a \in E^-$. The set E^+ (resp. E^-) contains positive (resp. negative) examples of C.

2. The second setting differs from the previous one only in that the condition (b) is replaced by the weaker one: $KB \not\models C(a)$ for all $a \in E^-$.

3. Given an interpretation $\mathcal{I}$ and sets E^+, E^- of individuals, learn a concept C in L such that: (a) $\mathcal{I} \models C(a)$ for all $a \in E^+$, and (b) $\mathcal{I} \models \neg C(a)$ for all $a \in E^-$. Note that $\mathcal{I} \not\models C(a)$ is the same as $\mathcal{I} \models \neg C(a)$.

As an early work on concept learning in DLs, Cohen and Hirsh [3] studied PAC-learnability of the CLASSIC description logic (an early DL formalism) and its sublogic C-CLASSIC. They proposed a concept learning algorithm based on "least common subsumers". In [9] Lambrix and Larocchia proposed a simple concept learning algorithm based on concept normalization.

Badea and Nienhuys-Cheng [1], Iannone et al. [8], Fanizzi et al. [6], Lehmann and Hitzler [10] studied concept learning in DLs by using refinement operators as in inductive logic programming. The works [1, 8] use the first setting, while the works [6, 10] use the second setting. Apart from refinement operators, scoring functions and search strategies also play important roles in algorithms proposed in those works. The algorithm DL-Learner [10] exploits genetic programming techniques, while DL-FOIL [6] considers also unlabeled data as in semi-supervised learning.

Nguyen and Szałas [13] applied bisimulation in DLs [5] to model indiscernibility of objects. Their work is pioneering in using bisimulation for concept learning in DLs. It also concerns concept approximation by using bisimulation and Pawlak's rough set theory [14, 15]. In [17] Tran et al. generalized and extended the concept learning method of [13] for DL-based information systems. They took attributes as basic elements of the language. An information system in a DL is a finite interpretation in that logic. Thus, both the works [13, 17] use the third setting. In [7] Ha et al. developed the first bisimulation-based method, called BBCL, for concept learning in DLs using the first setting. Their method uses models of KB and bisimulation in those models to guide the search for the concept to be learned. It is formulated for a large class of useful DLs, with well-known DLs like $\mathcal{ALC}$, $\mathcal{SHIQ}$, $\mathcal{SHOIQ}$, $\mathcal{SROIQ}$. The work [7] also introduced dual-BBCL, a variant of BBCL, for concept learning in DLs using the first setting. In [4] Divroodi et al. studied C-learnability in some DLs using the third setting.

This paper improves and extends [16]. In this work, we develop the first bisimulation-based method, called BBCL2, for concept learning in DLs using the second setting, i.e., for learning a concept C such that: $KB \models C(a)$ for all $a \in E^+$, and $KB \not\models C(a)$ for all $a \in E^-$, where KB is a given knowledge base in the considered DL, and E^+, E^- are given sets of examples of C. This method is based on the dual-BBCL method (of concept learning in DLs using the first setting) from our joint work [7]. We make appropriate changes for dealing with the condition "$KB \not\models C(a)$ for all $a \in E^-$" instead of "$KB \models \neg C(a)$ for all $a \in E^-$". As an extension of [16], in the current paper we prove that the BBCL2 method is sound and a variant BBCL2-MiMoD of BBCL2 has C-learnability for a certain class of DLs.

The rest of this paper is structured as follows. We recall syntax and semantics of DLs in Section 2, present our BBCL2 method in Section 3, study C-learnability of BBCL2 in Section 4, discuss and conclude the work in Section 5. Due to the lack of space, we will not recall the notion of bisimulation in DLs [5, 7], but just mention the use of the largest auto-bisimulation relations and list the bisimulation-based selectors [7].

2. Preliminaries

2.1. Syntax of Description Logics

A *DL-signature* is a finite set $\Sigma = \Sigma_I \cup \Sigma_{dA} \cup \Sigma_{nA} \cup \Sigma_{oR} \cup \Sigma_{dR}$, where Σ_I is a set of *individuals*, Σ_{dA} is a set of *discrete attributes*, Σ_{nA} is a set of *numeric attributes*, Σ_{oR} is a set of *object role names*, and Σ_{dR} is a set of *data roles*.[1] All the sets Σ_I, Σ_{dA}, Σ_{nA}, Σ_{oR}, Σ_{dR} are pairwise disjoint.

Let $\Sigma_A = \Sigma_{dA} \cup \Sigma_{nA}$. Each attribute $A \in \Sigma_A$ has a domain $dom(A)$, which is a non-empty set that is countable if A is discrete, and partially ordered by $\leq$ otherwise.[2] (For simplicity we do not subscript $\leq$ by A.) A discrete attribute is a *Boolean attribute* if $dom(A) = \{\mathsf{true}, \mathsf{false}\}$. We refer to Boolean attributes also as *concept names*. Let $\Sigma_C \subseteq \Sigma_{dA}$ be the set of all concept names of Σ.

An object role name stands for a binary predicate between individuals. A data role σ stands for a binary predicate relating individuals to elements of a set $range(\sigma)$.

We denote individuals by letters like a and b, attributes by letters like A and B, object role names by letters like r and s, data roles by letters like σ and ϱ, and elements of sets of the form $dom(A)$ or $range(\sigma)$ by letters like c and d.

We will consider some (additional) *DL-features* denoted by I (*inverse*), O (*nominal*), F (*functionality*), N (*unqualified number restriction*), Q (*qualified number restriction*), U (*universal role*), Self (*local reflexivity of an object role*). A *set of DL-features* is a set consisting of some or zero of these names.

Definition 2.1. (The $\mathcal{L}_{\Sigma,\Phi}$ Language)
Let Σ be a DL-signature and Φ 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 *object roles* and *concepts* defined as follows:

1. if $r \in \Sigma_{oR}$ then r is an object role of $\mathcal{L}_{\Sigma,\Phi}$
2. if $A \in \Sigma_C$ then A is concept of $\mathcal{L}_{\Sigma,\Phi}$
3. if $A \in \Sigma_A \setminus \Sigma_C$ and $d \in dom(A)$ then $A = d$ and $A \neq d$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
4. if $A \in \Sigma_{nA}$ and $d \in dom(A)$ then $A \leq d$, $A < d$, $A \geq d$ and $A > d$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
5. if C and D are concepts of $\mathcal{L}_{\Sigma,\Phi}$, R is an object role of $\mathcal{L}_{\Sigma,\Phi}$, $r \in \Sigma_{oR}$, $\sigma \in \Sigma_{dR}$, $a \in \Sigma_I$, and n is a natural number 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 $d \in range(\sigma)$ then $\exists \sigma.\{d\}$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$
   - if $I \in \Phi$ then r^- is an object role of $\mathcal{L}_{\Sigma,\Phi}$
   - if $O \in \Phi$ then $\{a\}$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$
   - if $F \in \Phi$ then $\leq 1\, r$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$
   - if $\{F, I\} \subseteq \Phi$ then $\leq 1\, r^-$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$

[1] Object role names are atomic object roles.
[2] One can assume that, if A is a numeric attribute, then $dom(A)$ is the set of real numbers and $\leq$ is the usual linear order between real numbers.

- if $N \in \Phi$ then $\geq n\, r$ and $\leq n\, r$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
- if $\{N, I\} \subseteq \Phi$ then $\geq n\, r^-$ and $\leq n\, r^-$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
- if $Q \in \Phi$ then $\geq n\, r.C$ and $\leq n\, r.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
- if $\{Q, I\} \subseteq \Phi$ then $\geq n\, r^-.C$ and $\leq n\, r^-.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
- if $U \in \Phi$ then U is an object role of $\mathcal{L}_{\Sigma,\Phi}$
- if $\mathsf{Self} \in \Phi$ then $\exists r.\mathsf{Self}$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$. □

Definition 2.2. (RBox – Box of Role Axioms)
A *role inclusion axiom* in $\mathcal{L}_{\Sigma,\Phi}$ is an expression of the form $R_1 \circ \ldots \circ R_k \sqsubseteq r$, where $k \geq 1$, $r \in \Sigma_{oR}$ and $R_1, \ldots, R_k$ are object roles of $\mathcal{L}_{\Sigma,\Phi}$ different from U. A *role assertion* in $\mathcal{L}_{\Sigma,\Phi}$ is an expression of the form $\mathtt{Ref}(r)$, $\mathtt{Irr}(r)$, $\mathtt{Sym}(r)$, $\mathtt{Tra}(r)$, or $\mathtt{Dis}(R, S)$, where $r \in \Sigma_{oR}$ and R, S are object roles of $\mathcal{L}_{\Sigma,\Phi}$ different from U. By a *role axiom* in $\mathcal{L}_{\Sigma,\Phi}$ we mean either a role inclusion axiom or a role assertion in $\mathcal{L}_{\Sigma,\Phi}$. An *RBox* in $\mathcal{L}_{\Sigma,\Phi}$ is a finite set of role axioms in $\mathcal{L}_{\Sigma,\Phi}$. □

Definition 2.3. (TBox – Box of Terminological Axioms)
A *terminological axiom* in $\mathcal{L}_{\Sigma,\Phi}$, also called a *general concept inclusion* (GCI) in $\mathcal{L}_{\Sigma,\Phi}$, is an expression of the form $C \sqsubseteq D$, where C and D are concepts in $\mathcal{L}_{\Sigma,\Phi}$. A *TBox* in $\mathcal{L}_{\Sigma,\Phi}$ is a finite set of terminological axioms in $\mathcal{L}_{\Sigma,\Phi}$. □

Definition 2.4. (ABox – Box of Individual Assertions)
An *individual assertion* in $\mathcal{L}_{\Sigma,\Phi}$ is an expression of one of the forms $C(a)$ (*concept assertion*), $r(a, b)$ (*positive role assertion*), $\neg r(a, b)$ (*negative role assertion*), $a = b$, and $a \neq b$, where $r \in \Sigma_{oR}$ and C is a concept of $\mathcal{L}_{\Sigma,\Phi}$. An *ABox* in $\mathcal{L}_{\Sigma,\Phi}$ is a finite set of individual assertions in $\mathcal{L}_{\Sigma,\Phi}$. □

We will write, for example, $A(a) = d$ instead of $(A = d)(a)$.

Definition 2.5. (Knowledge Base)
A *knowledge base* in $\mathcal{L}_{\Sigma,\Phi}$ is a triple $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{R}$ (resp. $\mathcal{T}$, $\mathcal{A}$) is an RBox (resp. a TBox, an ABox) in $\mathcal{L}_{\Sigma,\Phi}$. □

2.2. Semantics of Description Logics

As usual, the semantics of a logic is specified by interpretations and the satisfaction relation.

Definition 2.6. (Interpretation)
An *interpretation* in $\mathcal{L}_{\Sigma,\Phi}$ is a pair $\mathcal{I} = \left\langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \right\rangle$, 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 $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, each attribute $A \in \Sigma_A \setminus \Sigma_C$ with a partial function $A^\mathcal{I} : \Delta^\mathcal{I} \to dom(A)$, each object role name $r \in \Sigma_{oR}$ with a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and each data role $\sigma \in \Sigma_{dR}$ with a binary relation $\sigma^\mathcal{I} \subseteq \Delta^\mathcal{I} \times range(\sigma)$. The interpretation function $\cdot^\mathcal{I}$ is extended to complex object roles and complex concepts as shown in Figure 1, where $\#\Gamma$ stands for the cardinality of the set Γ. □

Given an interpretation $\mathcal{I} = \left\langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \right\rangle$ in $\mathcal{L}_{\Sigma,\Phi}$, we say that an object $x \in \Delta^\mathcal{I}$ has *depth* k if k is the maximal natural number such that there are pairwise different objects $x_0, \ldots, x_k$ of $\Delta^\mathcal{I}$ with the properties that:

$(r^-)^\mathcal{I} = (r^\mathcal{I})^{-1}$	$\top^\mathcal{I} = \Delta^\mathcal{I}$	$\bot^\mathcal{I} = \emptyset$
$U^\mathcal{I} = \Delta^\mathcal{I} \times \Delta^\mathcal{I}$	$(\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I}$	$\{a\}^\mathcal{I} = \{a^\mathcal{I}\}$
$(A \leq d)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid A^\mathcal{I}(x)$ is defined, $A^\mathcal{I}(x) \leq d\}$		$(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$
$(A \geq d)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid A^\mathcal{I}(x)$ is defined, $A^\mathcal{I}(x) \geq d\}$		$(C \sqcup D)^\mathcal{I} = C^\mathcal{I} \cup D^\mathcal{I}$
$(A = d)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid A^\mathcal{I}(x) = d\}$		$(A \neq d)^\mathcal{I} = (\neg(A = d))^\mathcal{I}$
$(A < d)^\mathcal{I} = ((A \leq d) \sqcap (A \neq d))^\mathcal{I}$		$(A > d)^\mathcal{I} = ((A \geq d) \sqcap (A \neq d))^\mathcal{I}$
$(\forall R.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \forall y\,[R^\mathcal{I}(x,y) \Rightarrow C^\mathcal{I}(y)]\}$		$(\exists r.\mathsf{Self})^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid r^\mathcal{I}(x,x)\}$
$(\exists R.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \exists y\,[R^\mathcal{I}(x,y) \wedge C^\mathcal{I}(y)]\}$		$(\exists \sigma.\{d\})^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \sigma^\mathcal{I}(x,d)\}$
$(\geq n\, R.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \#\{y \mid R^\mathcal{I}(x,y) \wedge C^\mathcal{I}(y)\} \geq n\}$		$(\geq n\, R)^\mathcal{I} = (\geq n\, R.\top)^\mathcal{I}$
$(\leq n\, R.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \#\{y \mid R^\mathcal{I}(x,y) \wedge C^\mathcal{I}(y)\} \leq n\}$		$(\leq n\, R)^\mathcal{I} = (\leq n\, R.\top)^\mathcal{I}$

Figure 1. Interpretation of complex object roles and complex concepts.

- $x_k = x$ and $x_0 = a^\mathcal{I}$ for some $a \in \Sigma_I$,
- $x_i \neq b^\mathcal{I}$ for all $1 \leq i \leq k$ and all $b \in \Sigma_I$,
- for each $1 \leq i \leq k$, there exists an object role R_i of $\mathcal{L}_{\Sigma,\Phi}$ such that $\langle x_{i-1}, x_i \rangle \in R_i^\mathcal{I}$.

By $\mathcal{I}_{|k}$ we denote the interpretation obtained from $\mathcal{I}$ by restricting the domain to the set of objects with depth not greater than k and restricting the interpretation function accordingly.

Definition 2.7. (The Satisfaction Relation)
Given an interpretation $\mathcal{I}$, define that:

$$
\begin{array}{lll}
\mathcal{I} \models C \sqsubseteq D & \text{if} & C^\mathcal{I} \subseteq D^\mathcal{I} \\
\mathcal{I} \models R_1 \circ \ldots \circ R_k \sqsubseteq r & \text{if} & R_1^\mathcal{I} \circ \ldots \circ R_k^\mathcal{I} \subseteq r^\mathcal{I} \\
\mathcal{I} \models \mathtt{Ref}(r) & \text{if} & r^\mathcal{I} \text{ is reflexive} \\
\mathcal{I} \models \mathtt{Irr}(r) & \text{if} & r^\mathcal{I} \text{ is irreflexive} \\
\mathcal{I} \models \mathtt{Sym}(r) & \text{if} & r^\mathcal{I} \text{ is symmetric} \\
\mathcal{I} \models \mathtt{Tra}(r) & \text{if} & r^\mathcal{I} \text{ is transitive} \\
\mathcal{I} \models \mathtt{Dis}(R, S) & \text{if} & R^\mathcal{I} \text{ and } S^\mathcal{I} \text{ are disjoint} \\
\mathcal{I} \models a = b & \text{if} & a^\mathcal{I} = b^\mathcal{I} \\
\mathcal{I} \models a \neq b & \text{if} & a^\mathcal{I} \neq b^\mathcal{I} \\
\mathcal{I} \models C(a) & \text{if} & a^\mathcal{I} \in C^\mathcal{I} \\
\mathcal{I} \models r(a, b) & \text{if} & \langle a^\mathcal{I}, b^\mathcal{I} \rangle \in r^\mathcal{I} \\
\mathcal{I} \models \neg r(a, b) & \text{if} & \langle a^\mathcal{I}, b^\mathcal{I} \rangle \notin r^\mathcal{I},
\end{array}
$$

where the operator $\circ$ stands for the composition of binary relations. We say that $\mathcal{I}$ *validates* an axiom or assertion φ if $\mathcal{I} \models \varphi$. □

Definition 2.8. (Semantics)
An interpretation $\mathcal{I}$ is a *model* of a "box" (RBox, TBox or ABox) if it validates all the axioms/assertions of that "box". It is a *model* of a knowledge base $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ if it is a model of $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{A}$. A knowledge base is *satisfiable* if it has a model. An individual a is said to be an *instance* of a concept C w.r.t. a knowledge base KB, denoted by $KB \models C(a)$, if, for every model $\mathcal{I}$ of KB, $a^{\mathcal{I}} \in C^{\mathcal{I}}$. □

An $\mathcal{L}_{\Sigma,\Phi}$ *logic* is specified by a number of restrictions adopted for the language $\mathcal{L}_{\Sigma,\Phi}$. We say that a logic L is *decidable* if the problem of checking satisfiability of a given knowledge base in L is decidable. A logic L has the *finite model property* if every satisfiable knowledge base in L has a finite model. We say that a logic L has the *semi-finite model property* if every satisfiable knowledge base in L has a model $\mathcal{I}$ such that, for any natural number k, $\mathcal{I}_{|k}$ is finite and constructible.

As the general satisfiability problem of context-free grammar logics is undecidable [2], the most general $\mathcal{L}_{\Sigma,\Phi}$ logics (without restrictions) are also undecidable. The considered class of DLs contains, however, many decidable and useful logics. One of them is $\mathcal{SROIQ}$ - the logical base of the Web Ontology Language OWL 2. This logic has the semi-finite model property.

Example 2.1. This example concerns people at a university. Let

$$\begin{aligned}
&\Phi = \{I\}, \quad \Sigma_I = \{a, b, c, d, e, f, g, h, i\}, \quad \Sigma_C = \{\mathit{Human}, \mathit{Prof}, \mathit{PhD}, \mathit{Student}, A_d\},\\
&\Sigma_{dA} = \Sigma_C, \quad \Sigma_{nA} = \{\mathit{Age}\}, \quad \Sigma_{oR} = \{\mathit{supervised}, \mathit{hasFriend}\}, \quad \Sigma_{dR} = \emptyset,\\
&\mathcal{R} = \{\mathtt{Sym}(\mathit{hasFriend})\},\\
&\mathcal{T} = \{\top \sqsubseteq \mathit{Human},\ \mathit{Prof} \sqsubseteq \mathit{PhD} \sqcap \exists \mathit{supervised}.\top\},\\
&\mathcal{A}_0 = \{\mathit{Prof}(a), \mathit{Prof}(b), \mathit{PhD}(c), \mathit{PhD}(d), \mathit{PhD}(e), \mathit{Student}(h), \mathit{Student}(i),\\
&\qquad \mathit{Age}(a) = 55, \mathit{Age}(b) = 40, \mathit{Age}(c) = 30, \mathit{Age}(d) = 45, \mathit{Age}(e) = 44, \mathit{Age}(f) = 41,\\
&\qquad \mathit{Age}(g) = 32, \mathit{Age}(h) = 24, \mathit{Age}(i) = 24, \mathit{supervised}(a, d), \mathit{supervised}(a, f),\\
&\qquad \mathit{supervised}(b, c), \mathit{supervised}(b, g), \mathit{supervised}(b, h), \mathit{supervised}(c, h), \mathit{supervised}(d, i),\\
&\qquad \mathit{hasFriend}(a, b), \mathit{hasFriend}(d, e), \mathit{hasFriend}(f, g), \mathit{hasFriend}(h, i)\}.
\end{aligned}$$

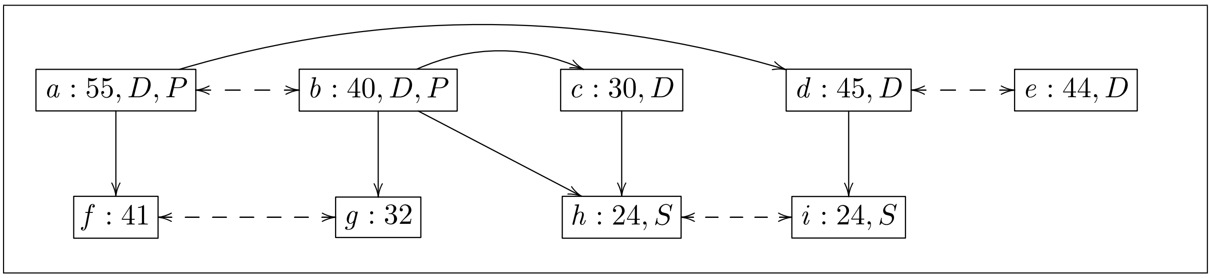

Figure 2. An illustration for the knowledge base given in Example 2.1.

Then $KB_0 = \langle \mathcal{R}, \mathcal{T}, \mathcal{A}_0 \rangle$ is a knowledge base in $\mathcal{L}_{\Sigma,\Phi}$. The axiom $\top \sqsubseteq \mathit{Human}$ states that the domain of each model of KB_0 consists of only people. The knowledge base KB_0 is illustrated in Figure 2. In this figure, each node denotes a person, the number denotes his/her age, and the letters D, P, S stand for PhD, Prof, $\mathit{Student}$, respectively. The solid edges denote assertions of the role $\mathit{supervised}$ and the dashed edges denote assertions of the role $\mathit{hasFriend}$. □

3. Concept Learning for Knowledge Bases in Description Logics

Let L be a decidable $\mathcal{L}_{\Sigma,\Phi}$ logic with the semi-finite model property, $A_d \in \Sigma_C$ be a special concept name standing for the "decision attribute", and $KB_0 = \langle \mathcal{R}, \mathcal{T}, \mathcal{A}_0 \rangle$ be a knowledge base in L without using A_d. Let E^+ and E^- be disjoint subsets of Σ_I such that the knowledge base $KB = \langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ with $\mathcal{A} = \mathcal{A}_0 \cup \{A_d(a) \mid a \in E^+\} \cup \{\neg A_d(a) \mid a \in E^-\}$ is satisfiable. The set E^+ (resp. E^-) is called the set of *positive* (resp. *negative*) *examples* of A_d. Let $E = \langle E^+, E^- \rangle$. The problem is to learn a concept C as a definition of A_d in the logic L restricted to a given sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ with $\Sigma^\dagger \subseteq \Sigma \setminus \{A_d\}$ and $\Phi^\dagger \subseteq \Phi$ such that: $KB \models C(a)$ for all $a \in E^+$, and $KB \not\models C(a)$ for all $a \in E^-$.

3.1. Partitioning the Domain of an Interpretation

Given an interpretation $\mathcal{I}$ in $\mathcal{L}_{\Sigma,\Phi}$, by $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$ we denote the equivalence relation on $\Delta^\mathcal{I}$ with the property that $x \equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}} x'$ iff x is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-equivalent to x' (i.e., for every concept D of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, $x \in D^\mathcal{I}$ iff $x' \in D^\mathcal{I}$). By [7, Theorem 3], this equivalence relation coincides with the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation $\sim_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$ of $\mathcal{I}$ (see [7] for the definition of this notion).

Let $\mathcal{I}$ be an interpretation. We say that a set $Y \subseteq \Delta^\mathcal{I}$ is *divided* by E if there exist $a \in E^+$ and $b \in E^-$ such that $\{a^\mathcal{I}, b^\mathcal{I}\} \subseteq Y$. A partition $P = \{Y_1, \ldots, Y_k\}$ of $\Delta^\mathcal{I}$ is said to be *consistent* with E if, for every $1 \leq i \leq n$, Y_i is not divided by E. Observe that if $\mathcal{I}$ is a model of KB then:

- since C is a concept of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, by [7, Theorems 2 and 3], $C^\mathcal{I}$ should be the union of a number of equivalence classes of $\Delta^\mathcal{I}$ w.r.t. $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$
- we should have that $a^\mathcal{I} \in C^\mathcal{I}$ for all $a \in E^+$, and $a^\mathcal{I} \notin C^\mathcal{I}$ for all $a \in E^-$.

Having a model $\mathcal{I}$ of KB, the partition $\{Y_{i_1}, \ldots, Y_{i_k}\}$ of $\Delta^\mathcal{I}$ that corresponds to $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$ can be constructed by granulating the initial partition $\{\Delta^\mathcal{I}\}$ as follows:

- In the granulation process, we denote the blocks created so far in all steps by $Y_1, \ldots, Y_n$, where the current partition $\{Y_{i_1}, \ldots, Y_{i_k}\}$ consists 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 $1 \leq i \leq n$, Y_i is characterized by an appropriate concept C_i (such that $Y_i = C_i^\mathcal{I}$).
- Following [13, 17] we use the concepts listed in Figure 3 (on page 294) as *selectors* for the granulation process. If a block Y_{i_j} ($1 \leq j \leq k$) is divided by $D^\mathcal{I}$, where D is a selector, then dividing Y_{i_j} by D is done as follows:
  - $s := n + 1,\ t := n + 2,\ n := n + 2,$
  - $Y_s := Y_{i_j} \cap D^\mathcal{I},\ C_s := C_{i_j} \sqcap D,$
  - $Y_t := Y_{i_j} \cap (\neg D)^\mathcal{I},\ C_t := C_{i_j} \sqcap \neg D,$
  - the new partition of $\Delta^\mathcal{I}$ becomes $\{Y_{i_1}, \ldots, Y_{i_k}\} \setminus \{Y_{i_j}\} \cup \{Y_s, Y_t\}$.
- Which block from the current partition should be divided first and which selector should be used to divide it are left open for heuristics. For example, one can apply some gain function like the

- A, where $A \in \Sigma_C^\dagger$
- $A = d$, where $A \in \Sigma_A^\dagger \setminus \Sigma_C^\dagger$ and $d \in dom(A)$
- $A \leq d$ and $A < d$, where $A \in \Sigma_{nA}^\dagger$, $d \in dom(A)$ and d is not a minimal element of $dom(A)$
- $A \geq d$ and $A > d$, where $A \in \Sigma_{nA}^\dagger$, $d \in dom(A)$ and d is not a maximal element of $dom(A)$
- $\exists\sigma.\{d\}$, where $\sigma \in \Sigma_{dR}^\dagger$ and $d \in range(\sigma)$
- $\exists r.C_i$, $\exists r.\top$ and $\forall r.C_i$, where $r \in \Sigma_{oR}^\dagger$ and $1 \leq i \leq n$
- $\exists r^-.C_i$, $\exists r^-.\top$ and $\forall r^-.C_i$, if $I \in \Phi^\dagger$, $r \in \Sigma_{oR}^\dagger$ and $1 \leq i \leq n$
- $\{a\}$, if $O \in \Phi^\dagger$ and $a \in \Sigma_I^\dagger$
- $\leq 1\, r$, if $F \in \Phi^\dagger$ and $r \in \Sigma_{oR}^\dagger$
- $\leq 1\, r^-$, if $\{F, I\} \subseteq \Phi^\dagger$ and $r \in \Sigma_{oR}^\dagger$
- $\geq l\, r$ and $\leq m\, r$, if $N \in \Phi^\dagger$, $r \in \Sigma_{oR}^\dagger$, $0 < l \leq \#\Delta^\mathcal{I}$ and $0 \leq m < \#\Delta^\mathcal{I}$
- $\geq l\, r^-$ and $\leq m\, r^-$, if $\{N, I\} \subseteq \Phi^\dagger$, $r \in \Sigma_{oR}^\dagger$, $0 < l \leq \#\Delta^\mathcal{I}$ and $0 \leq m < \#\Delta^\mathcal{I}$
- $\geq l\, r.C_i$ and $\leq m\, r.C_i$, if $Q \in \Phi^\dagger$, $r \in \Sigma_{oR}^\dagger$, $1 \leq i \leq n$, $0 < l \leq \#C_i$ and $0 \leq m < \#C_i$
- $\geq l\, r^-.C_i$ and $\leq m\, r^-.C_i$, if $\{Q, I\} \subseteq \Phi^\dagger$, $r \in \Sigma_{oR}^\dagger$, $1 \leq i \leq n$, $0 < l \leq \#C_i$ and $0 \leq m < \#C_i$
- $\exists r.\mathsf{Self}$, if $\mathsf{Self} \in \Phi^\dagger$ and $r \in \Sigma_{oR}^\dagger$

Figure 3. Selectors. Here, n is the number of blocks created so far when granulating $\Delta^\mathcal{I}$, and C_i is the concept characterizing the block Y_i. It was proved in [17] that using these selectors is sufficient to granulate $\Delta^\mathcal{I}$ to obtain the partition corresponding to $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$.

information gain measure, while taking into account also simplicity of selectors and the concepts characterizing the blocks. Randomization is used to a certain extent. For example, if some selectors give the same gain and are the best then randomly choose any one of them.

The granulation process can be terminated as soon as the current partition is consistent with E (or when some criteria are met). However, as will be seen later, if it is hard to learn the concept C, then this loosening should be discarded.

As the result of the granulation process, apart from the partition $\{Y_{i_1}, \ldots, Y_{i_k}\}$ we also have concepts C_{i_j} ($1 \leq j \leq k$) that characterize the blocks Y_{i_j} (i.e., $Y_{i_j} = C_{i_j}^\mathcal{I}$).

3.2. The BBCL2 Concept Learning Method

We now describe our method *BBCL2* (*Bisimulation-Based Concept Learning* for knowledge bases in DLs using the *second* setting). Recall that the problem is to learn a concept C in the restricted sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ such that: $KB \models C(a)$ for all $a \in E^+$, and $KB \not\models C(a)$ for all $a \in E^-$. The idea is to use models of KB and bisimulation in those models to guide the search for C. Our method constructs a set of E_0^- of individuals and sets of concepts $\mathbb{C}$, $\mathbb{C}_0$. E_0^- will cover more and more individuals from E^-. The meaning of $\mathbb{C}$ is to collect concepts D such that $KB \models D(a)$ for all $a \in E^+$. The set $\mathbb{C}_0$ is auxiliary for the construction of $\mathbb{C}$. When a concept D does not satisfy the mentioned condition but is a "good" candidate for that, we put it into $\mathbb{C}_0$. Later, when necessary, we take disjunctions of some concepts from $\mathbb{C}_0$ and check whether they are good for adding to $\mathbb{C}$. During the learning process, we will always have:

- $KB \models (\bigsqcap \mathbb{C})(a)$ for all $a \in E^+$,
- $KB \not\models (\bigsqcap \mathbb{C})(a)$ for all $a \in E_0^-$,

where $\bigsqcap\{D_1, \ldots, D_n\} = D_1 \sqcap \ldots \sqcap D_n$ and $\bigsqcap \emptyset = \top$. We try to extend $\mathbb{C}$ to satisfy $KB \not\models (\bigsqcap \mathbb{C})(a)$ for more and more $a \in E^-$. Extending $\mathbb{C}$ enables extension of E_0^-. When E_0^- reaches E^-, we return the concept $\bigsqcap \mathbb{C}$ after normalization and simplification. Our method is not a detailed algorithm, as we leave some steps at an abstract level, open to implementation heuristics. In particular, we assume that it is known whether L has the finite model property, how to construct models of KB, and how to do instance checking $KB \models D(a)$ for arbitrary D and a. The steps of our method are as follows.

1. Initialize $E_0^- := \emptyset$, $\mathbb{C} := \emptyset$, $\mathbb{C}_0 := \emptyset$.

2. (This is the beginning of a loop controlled by "go to" at Step 6.) If L has the finite model property then construct a (next) finite model $\mathcal{I}$ of KB. Otherwise, construct a (next) interpretation $\mathcal{I}$ such that either $\mathcal{I}$ is a finite model of KB or $\mathcal{I} = \mathcal{I}'_{|K}$, where $\mathcal{I}'$ is an infinite model of KB and K is a parameter of the learning method (e.g., with value 5).

3. Construct the partition $\{Y_{i_1}, \ldots, Y_{i_k}\}$ of $\Delta^\mathcal{I}$ that corresponds to $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$ together with concepts C_{i_j} such that $Y_{i_j} = C_{i_j}^\mathcal{I}$ (for $1 \leq j \leq k$).

4. For each $1 \leq j \leq k$, if Y_{i_j} contains some $a^\mathcal{I}$ with $a \in E^-$ and no $a^\mathcal{I}$ with $a \in E^+$ then:
   - if $KB \models \neg C_{i_j}(a)$ for all $a \in E^+$ then

     if $\bigsqcap \mathbb{C}$ is not subsumed by $\neg C_{i_j}$ w.r.t. KB (i.e. $KB \not\models (\bigsqcap \mathbb{C} \sqsubseteq \neg C_{i_j})$) then add $\neg C_{i_j}$ to $\mathbb{C}$ and add to E_0^- all $a \in E^-$ such that $a^\mathcal{I} \in Y_{i_j}$
   - else add $\neg C_{i_j}$ to $\mathbb{C}_0$.

5. If $E_0^- = E^-$ then go to Step 8.

6. If it was hard to extend $\mathbb{C}$ during a considerable number of iterations of the loop (with different interpretations $\mathcal{I}$) then go to Step 7, else go to Step 2.

7. Repeat the following:

   (a) Randomly select some concepts $D_1, \ldots, D_l$ from $\mathbb{C}_0$ and let $D = (D_1 \sqcup \ldots \sqcup D_l)$.

   (b) If $KB \models D(a)$ for all $a \in E^+$, $\bigsqcap \mathbb{C}$ is not subsumed by D w.r.t. KB (i.e., $KB \not\models (\bigsqcap \mathbb{C}) \sqsubseteq D$), and $E^- \setminus E_0^-$ contains some a such that $KB \not\models (\bigsqcap \mathbb{C} \sqcap D)(a)$, then:

      i. add D to $\mathbb{C}$,

      ii. add to E_0^- all $a \in E^- \setminus E_0^-$ such that $KB \not\models (\bigsqcap \mathbb{C})(a)$,

      iii. if $E_0^- = E^-$ then go to Step 8.

(c) If it was still too hard to extend $\mathbb{C}$ during a considerable number of iterations of the current loop, or $\mathbb{C}$ is already too big, then terminate the process with failure.

8. For each $D \in \mathbb{C}$, if $KB \not\models \bigsqcap(\mathbb{C} \setminus \{D\})(a)$ for all $a \in E^-$ then delete D from $\mathbb{C}$.

9. Let C_{rs} be a normalized form of $\bigsqcap \mathbb{C}$.[3] Observe that $KB \models C_{rs}(a)$ for all $a \in E^+$, and $KB \not\models C_{rs}(a)$ for all $a \in E^-$. Try to simplify C_{rs} while preserving this property, and then return it.

For Step 2, if L is one of the well known DLs, then $\mathcal{I}$ can be constructed by using a tableau algorithm (see [7] for references). During the construction, randomization is used to a certain extent to make $\mathcal{I}$ different from the interpretations generated in previous iterations of the loop.

As mentioned before, the granulation process at Step 3 can be terminated as soon as the current partition is consistent with E (or when some criteria are met). But, if it is hard to extend $\mathbb{C}$ during a considerable number of iterations of the loop (with different interpretations $\mathcal{I}$), then this loosening should be discarded.

Observe that, when $\neg C_{i_j}$ is added to $\mathbb{C}$, we have that $a^{\mathcal{I}} \in (\neg C_{i_j})^{\mathcal{I}}$ for all $a \in E^+$. This is a good point for hoping that $KB \models \neg C_{i_j}(a)$ for all $a \in E^+$. We check it, for example, by using some appropriate tableau decision procedure, and if it holds then we add $\neg C_{i_j}$ to the set $\mathbb{C}$. Otherwise, we add $\neg C_{i_j}$ to $\mathbb{C}_0$. To increase the chance to have C_{i_j} satisfying the mentioned condition and being added to $\mathbb{C}$, we tend to make C_{i_j} strong enough. For this reason, we do not use the technique with $LargestContainer$ introduced in [13], and when necessary, we do not apply the above mentioned loosening for Step 3.

Note that any single concept D from $\mathbb{C}_0$ does not satisfy the condition $KB \models D(a)$ for all $a \in E^+$, but when we take a number of concepts $D_1, \ldots, D_l$ from $\mathbb{C}_0$ we may have that $KB \models (D_1 \sqcup \ldots \sqcup D_l)(a)$ for all $a \in E^+$. So, when it is really hard to extend $\mathbb{C}$ by directly using concepts $\neg C_{i_j}$ (where C_{i_j} are the concepts used for characterizing blocks of partitions of the domains of models of KB), we change to using disjunctions $D_1 \sqcup \ldots \sqcup D_l$ of concepts from $\mathbb{C}_0$ as candidates for adding to $\mathbb{C}$.

Proposition 3.1. The BBCL2 method is sound. That is, if the method returns a concept C_{rs} then C_{rs} is a solution of the considered problem.

Proof:
Observe that before the execution terminates or reaches Step 8 we always have that:

- $KB \models (\bigsqcap \mathbb{C})(a)$ for all $a \in E^+$,
- $KB \not\models (\bigsqcap \mathbb{C})(a)$ for all $a \in E_0^-$.

When the execution reaches Step 8, we have $E_0^- = E^-$. Hence, at the moments before and after executing Step 8, we have that:

- $KB \models (\bigsqcap \mathbb{C})(a)$ for all $a \in E^+$,
- $KB \not\models (\bigsqcap \mathbb{C})(a)$ for all $a \in E^-$.

[3] Normalizing concepts can be done, e.g., as in [11, 18].

At Step 9, normalization and simplification is done for $C_{rs} = \bigsqcap \mathbb{C}$. As normalization preserves equivalence and simplification is done only when the above two properties with $\bigsqcap \mathbb{C}$ replaced by C_{rs} are preserved, we conclude that if the method does not terminate with failure then the returned concept C_{rs} is a solution of the considered problem. □

Remark 3.1. (Complexity)
Concept learning using the second setting involves with automated reasoning in the considered DL. As the latter problem is EXPTIME-hard even for the basic DL $\mathcal{ALC}$, one cannot hope for a lower complexity. In general, if the satisfiability problem in the considered DL is EXPTIME-complete and we use a polynomial bound for the cardinality of the collection $\mathbb{C}_0$, then our method results in an algorithm that runs in exponential time (in the sizes of KB, E^+ and E^-, while assuming that $\Sigma^\dagger$ is fixed). □

3.3. Illustrative Examples

In [16] we gave two examples of application of the BBCL2 method. Here, we provide two more examples, which use different heuristics for choosing selectors. For more details on such heuristics, we refer the reader to [13, 18].

Example 3.1. Let $KB_0 = \langle \mathcal{R}, \mathcal{T}, \mathcal{A}_0 \rangle$ be the knowledge base given in Example 2.1. Let $E^+ = \{a, b\}$, $E^- = \{c, d, e, f, g, h, i\}$, $\Sigma^\dagger = \{PhD, supervised\}$ and $\Phi^\dagger = \emptyset$. As usual, let $KB = \langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{A} = \mathcal{A}_0 \cup \{A_d(a) \mid a \in E^+\} \cup \{\neg A_d(a) \mid a \in E^-\}$. Execution of the BBCL2 method on this example is as follows.

1. $E_0^- := \emptyset$, $\mathbb{C} := \emptyset$, $\mathbb{C}_0 := \emptyset$.
2. KB has infinitely many models, but the most natural one is $\mathcal{I}$ specified below, which is used first:

   $\Delta^\mathcal{I} = \{a, b, c, d, e, f, g, h, i\},\ x^\mathcal{I} = x$ for $x \in \{a, b, c, d, e, f, g, h, i\}$,
   $Human^\mathcal{I} = \Delta^\mathcal{I},\ Prof^\mathcal{I} = \{a, b\},\ PhD^\mathcal{I} = \{a, b, c, d, e\},\ Student^\mathcal{I} = \{h, i\}$,
   $supervised^\mathcal{I} = \{\langle a, d\rangle, \langle a, f\rangle, \langle b, c\rangle, \langle b, g\rangle, \langle b, h\rangle, \langle c, h\rangle, \langle d, i\rangle\}$,
   $hasFriend^\mathcal{I} = \{\langle a, b\rangle, \langle b, a\rangle, \langle d, e\rangle, \langle e, d\rangle, \langle f, g\rangle, \langle g, f\rangle, \langle h, i\rangle, \langle i, h\rangle\}$,
   the function $Age^\mathcal{I}$ is specified according to $\mathcal{A}_0$.

3. $Y_1 := \Delta^\mathcal{I},\ C_1 := \top,\ partition := \{Y_1\}$.
4. Dividing Y_1: The possible selectors are PhD and $\exists supervised.\top$. The selector $\exists supervised.\top$ gives a higher information gain, but the selector PhD is simpler. Assume that the latter is preferred.
   - $Y_2 := \{a, b, c, d, e\},\ C_2 := C_1 \sqcap PhD$,
   - $Y_3 := \{f, g, h, i\},\ C_3 := C_1 \sqcap \neg PhD$,
   - $partition := \{Y_2, Y_3\}$.
5. Dividing Y_2: The selector $\exists supervised.C_2$ gives the highest information gain and is used.
   - $Y_4 := \{a, b\},\ C_4 := C_2 \sqcap \exists supervised.C_2$,

- $Y_5 := \{c, d, e\},\ C_5 := C_2 \sqcap \neg\exists supervised.C_2$,
- $partition := \{Y_3, Y_4, Y_5\}$.

6. The obtained partition is consistent with E, having $Y_3 = \{f, g, h, i\} \subset E^-$, $Y_4 = \{a, b\} = E^+$ and $Y_5 = \{c, d, e\} \subset E^-$. (It is not yet the partition corresponding to $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$.)

7. Since $Y_3 \subset E^-$ and $KB \models \neg C_3(x)$ for all $x \in E^+$, we add $\neg C_3$ to $\mathbb{C}$ and add the elements of Y_3 to E_0^-. Thus, $\mathbb{C} = \{\neg C_3\}$, $\bigsqcap \mathbb{C} = \neg(\top \sqcap \neg PhD)$ and $E_0^- = \{f, g, h, i\}$.

8. Since $Y_5 \subset E^-$ and $KB \models \neg C_5(x)$ for all $x \in E^+$ and $\bigsqcap \mathbb{C}$ is not subsumed by $\neg C_5$ w.r.t. KB, we add $\neg C_5$ to $\mathbb{C}$ and add the elements of Y_5 to E_0^-. Thus, $\mathbb{C} = \{\neg C_3, \neg C_5\}$, $\bigsqcap \mathbb{C} = \neg(\top \sqcap \neg PhD) \sqcap \neg((\top \sqcap PhD) \sqcap \neg\exists supervised.(\top \sqcap PhD))$ and $E_0^- = \{c, d, e, f, g, h, i\}$.

9. Since $E_0^- = E^-$, we normalize $\bigsqcap \mathbb{C}$ to $PhD \sqcap \exists supervised.PhD$ and return it as the result. This concept denotes the set of PhDs who supervised some PhDs. □

Example 3.2. Let $KB_0 = \langle \mathcal{R}, \mathcal{T}, \mathcal{A}_0 \rangle$ be as in Examples 2.1 and 3.1. Let $E^+ = \{a, b, d\}$, $E^- = \{c, e, f, g, h, i\}$, $\Sigma^\dagger = \{Age, supervised\}$ and $\Phi^\dagger = \emptyset$. As usual, let $KB = \langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{A} = \mathcal{A}_0 \cup \{A_d(a) \mid a \in E^+\} \cup \{\neg A_d(a) \mid a \in E^-\}$. Execution of the BBCL2 method on this new example has the same first three steps as in Example 3.1, and then continues as follows.

4. Dividing Y_1: The selector $\exists supervised.\top$ gives the highest information gain and is used.
   - $Y_2 := \{a, b, c, d\},\ C_2 := C_1 \sqcap \exists supervised.\top$,
   - $Y_3 := \{e, f, g, h, i\},\ C_3 := C_1 \sqcap \neg\exists supervised.\top$,
   - $partition := \{Y_2, Y_3\}$.

5. Dividing Y_2: The selector $Age \geq 40$ gives the highest information gain and is used.
   - $Y_4 := \{a, b, d\},\ C_4 := C_2 \sqcap (Age \geq 40)$,
   - $Y_5 := \{c\},\ C_5 := C_2 \sqcap (Age < 40)$,
   - $partition := \{Y_3, Y_4, Y_5\}$.

6. The obtained partition is consistent with E, having $Y_3 = \{e, f, g, h, i\} \subset E^-$, $Y_4 = \{a, b, d\} = E^+$ and $Y_5 = \{c\} \subset E^-$. (It is not yet the partition corresponding to $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$.)

7. Since $Y_3 \subset E^-$ and $KB \models \neg C_3(x)$ for all $x \in E^+$, we add $\neg C_3$ to $\mathbb{C}$ and add the elements of Y_3 to E_0^-. Thus, $\mathbb{C} = \{\neg C_3\}$, $\bigsqcap \mathbb{C} = \neg(\top \sqcap \neg\exists supervised.\top)$ and $E_0^- = \{e, f, g, h, i\}$.

8. Since $Y_5 \subset E^-$ and $KB \models \neg C_5(x)$ for all $x \in E^+$ and $\bigsqcap \mathbb{C}$ is not subsumed by $\neg C_5$ w.r.t. KB, we add $\neg C_5$ to $\mathbb{C}$ and add the elements of Y_5 to E_0^-. Thus, $\mathbb{C} = \{\neg C_3, \neg C_5\}$, $\bigsqcap \mathbb{C} = \neg(\top \sqcap \neg\exists supervised.\top) \sqcap \neg((\top \sqcap \exists supervised.\top) \sqcap (Age < 40))$ and $E_0^- = \{c, e, f, g, h, i\}$.

9. Since $E_0^- = E^-$, we normalize $\bigsqcap \mathbb{C}$ to $\exists supervised.\top \sqcap (Age \geq 40)$ and return it as the result. □

4. On C-Learnability of BBCL2

In [4] Divroodi et al. proved that any concept in any description logic that extends the basic DL $\mathcal{ALC}$ with some features amongst I, Self, Q_k (qualified number restrictions with numbers bounded by a constant k) can be learned if the training information system (specified as an 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 consistent with C such that applying the learning algorithm to the system results in a concept equivalent to C. They called this property *C-learnability* (possibility of correct learning). Their work uses the third setting. In this section, we provide a similar result for the second setting.

From now on, we assume that $\Sigma_{nA} = \Sigma_{dR} = \emptyset$ (i.e., there are no numeric attributes nor data roles) and $\Phi \subseteq \{I, F, N_k, Q_k, \mathsf{Self}\}$, where N_k and Q_k are restricted versions of the features N and Q with numbers bounded by a constant k. Furthermore, we assume that Σ is specified also by a subset Σ_{soR} of Σ_{oR} that consists of so called *simple object roles*. The language $\mathcal{L}_{\Sigma,\Phi}$ is redefined by adding the following subitems to the item 5 of Definition 2.1 (and deleting the redundant items):

- if $n \leq k$ and R is of the form r or r^- with $r \in \Sigma_{soR}$ then
  - if $N_k \in \Phi$ then $\geq n\, R$ and $\leq n\, R$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
  - if $Q_k \in \Phi$ then $\geq n\, R.C$ and $\leq n\, R.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$.

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

- 0 if C is of the form $\top$, $\bot$, A or $\exists r.\mathsf{Self}$,
- 1 if C is of the form $\geq n\, R$ or $\leq n\, R$,
- $\mathsf{mdepth}(D)$ 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 $\forall R.D$, $\exists R.D$, $\geq n\, R.D$ or $\leq n\, R.D$.

The *skeleton interpretation of an ABox* $\mathcal{A}$ is the interpretation $\mathcal{I}$ specified by:

- $\Delta^\mathcal{I}$ is the set of all individuals occurring in $\mathcal{A}$,
- if $a \in \Delta^\mathcal{I}$ then $a^\mathcal{I} = a$, else if $a \in \Sigma_I \setminus \Delta^\mathcal{I}$ then $a^\mathcal{I}$ is some fixed element of $\Delta^\mathcal{I}$,
- $A^\mathcal{I} = \{a \mid A(a) \in \mathcal{A}\}$ for $A \in \Sigma_C$, and $r^\mathcal{I} = \{\langle a, b\rangle \mid r(a, b) \in \mathcal{A}\}$ for $r \in \Sigma_{oR}$.

By BBCL2-MiMoD we denote the variant of BBCL2 with the following differences:

- At Step 2, if the skeleton interpretation of $\mathcal{A}$ is a model of KB then use it for $\mathcal{I}$ at the first iteration of the loop. (This is a very natural try.)
- At Step 3, during the granulation process, if
  - a block Y_i of the current partition is divided 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' such that Y_j is divided by D' and $\mathsf{mdepth}(D') < \mathsf{mdepth}(D)$

then divide Y_i by D. (MiMoD stands for "minimizing modal depth".)

- The granulation process at Step 3 is terminated as soon as the current partition is consistent with E.
- Step 9 is simplified to "Return the normalized form of $\bigsqcap \mathbb{C}$.".

Example 4.1. Reconsider Example 3.1 and assume that the numeric attribute Age is discarded from its text and the signature (as it is inessential for Example 3.1). The execution of BBCL2 in Example 3.1 minimizes the modal depth of the resulting concept (the choice of the selector PhD at the step 4 is essential). Hence, it is an example of application of the BBCL2-MiMoD method. $\square$

Let the notion of *regular RBoxes* be defined as in [12], which requires the introduction of simple object roles for the case $\{N_k, Q_k\} \cap \Phi \neq \emptyset$ and has the property that if $\mathcal{R}$ is a regular RBox then:

- checking satisfiability of any knowledge base $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ in $\mathcal{L}_{\Sigma,\Phi}$ is decidable,
- if $\Phi \subseteq \{I, \mathsf{Self}\}$ or $\Phi \subseteq \{F, N_k, Q_k, \mathsf{Self}\}$ then any satisfiable knowledge base $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ in $\mathcal{L}_{\Sigma,\Phi}$ has a finite model, which can be effectively constructed.

A concept C is *satisfiable* w.r.t. an RBox $\mathcal{R}$ and a TBox $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{R}$ and $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$. It is *equivalent* to D w.r.t. $\mathcal{R}$ and $\mathcal{T}$ if $C^{\mathcal{I}} = D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{R}$ and $\mathcal{T}$.

Here is the main result of this section:

Theorem 4.1. Suppose $\Phi \subseteq \{I, \mathsf{Self}\}$ or $\Phi \subseteq \{F, N_k, Q_k, \mathsf{Self}\}$, $\Phi^\dagger \subseteq \Phi$ and $\Sigma^\dagger \subseteq \Sigma \setminus \{A_d\}$. For any regular RBox $\mathcal{R}$ and any TBox $\mathcal{T}$ in $\mathcal{L}_{\Sigma,\Phi}$ and for any concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, if $\langle \mathcal{R}, \mathcal{T}, \emptyset \rangle$ is satisfiable and Σ_I is big enough[4] then there exist an ABox $\mathcal{A}$ in $\mathcal{L}_{\Sigma,\Phi}$ and subsets E^+ and E^- of Σ_I such that applying the BBCL2-MiMoD method to $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ and $E = \langle E^+, E^- \rangle$ results in a concept equivalent to C w.r.t. $\mathcal{R}$ and $\mathcal{T}$.

To prove this theorem we need some auxiliary definitions and a lemma.

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.

An interpretation $\mathcal{I}$ in $\mathcal{L}_{\Sigma,\Phi}$ is called a *universal* model of an RBox $\mathcal{R}$ and a TBox $\mathcal{T}$ w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ if, for every concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ that is satisfiable w.r.t. $\mathcal{R}$ and $\mathcal{T}$, we have that $C^{\mathcal{I}} \neq \emptyset$.

Lemma 4.1. Suppose $\Phi \subseteq \{I, \mathsf{Self}\}$ or $\Phi \subseteq \{F, N_k, Q_k, \mathsf{Self}\}$, $\Phi^\dagger \subseteq \Phi$ and $\Sigma^\dagger \subseteq \Sigma$. Let $\mathcal{R}$ be a regular RBox and $\mathcal{T}$ a TBox in $\mathcal{L}_{\Sigma,\Phi}$ such that $\langle \mathcal{R}, \mathcal{T}, \emptyset \rangle$ is satisfiable. Then there exists a finite universal model of $\mathcal{R}$ and $\mathcal{T}$ w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$, which can be effectively constructed.

This lemma can be proved analogously as done for [4, Lemma 3.3].

Proof of Theorem 4.1. Let $d = \mathsf{mdepth}(C)$ and let $\mathcal{I}_0$ be a finite universal model of $\mathcal{R}$ and $\mathcal{T}$ w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$ (by Lemma 4.1, such a model exists and can be effectively constructed). Name the elements of $\Delta^{\mathcal{I}_0}$ by individuals $a_1, \ldots, a_m$ from Σ_I (recall that Σ_I is assumed to be big enough), then choose $E^+ = C^{\mathcal{I}_0}$ and $E^- = \Delta^{\mathcal{I}_0} \setminus E^+$ (which are subsets of Σ_I). Suppose that executing Step 3 of

[4]That is, the cardinality of Σ_I is greater than the value of a certain fixed function of the sizes of $\mathcal{R}$ and $\mathcal{T}$ and the length of C.

BBCL2-MiMoD for $\mathcal{I}_0$ (in the place of $\mathcal{I}$) would result in a partition $\{Y_{i_1}, \ldots, Y_{i_k}\}$ and concepts C_{i_j} that characterize the blocks Y_{i_j}. Choose $\mathcal{A} = \mathcal{A}_0 \cup \{A_d(a) \mid a \in E^+\} \cup \{\neg A_d(a) \mid a \in E^-\}$, where

$$\begin{array}{rcl}\mathcal{A}_0 & = & \{A(a) \mid A \in \Sigma_C \text{ and } a \in A^{\mathcal{I}_0}\} \cup \{r(a,b) \mid r \in \Sigma_{oR} \text{ and } \langle a,b\rangle \in r^{\mathcal{I}_0}\} \cup \\ & & \{\neg C_{i_j}(a) \mid 1 \leq j \leq k \text{ and } a \in \Delta^{\mathcal{I}_0} \setminus Y_{i_j}\}.\end{array}$$

Let $\mathcal{I}$ be the skeleton interpretation of $\mathcal{A}$. It has the same domain as $\mathcal{I}_0$ and interprets concept names and object role names in the same way as $\mathcal{I}_0$. Hence, like $\mathcal{I}_0$, it is a finite universal model of $\mathcal{R}$ and $\mathcal{T}$ w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$. Furthermore, $\mathcal{I}$ is a model of $\mathcal{A}$, because $C_{i_j}^{\mathcal{I}_0} = Y_{i_j}$ for all $1 \leq j \leq k$. Thus, $\mathcal{I}$ is a model of KB and is used at Step 2 of BBCL2-MiMoD at the first iteration of the loop.

The execution of Step 3 of BBCL2-MiMoD using that model $\mathcal{I}$ is the same as the one for $\mathcal{I}_0$ and results in the same partition $\{Y_{i_1}, \ldots, Y_{i_k}\}$ and the same concepts C_{i_j}. It is exactly the execution of the MiMoD algorithm given in [4] for learning a definition of the concept C when given $\mathcal{I}$ and the set $A_0^{\mathcal{I}} = C^{\mathcal{I}} = C^{\mathcal{I}_0} = E^+$ (i.e., using the third setting). That algorithm returns $C' = \bigsqcup\{C_{i_j} \mid 1 \leq j \leq k,\ Y_{i_j} \subseteq E^+\}$, where $\bigsqcup\{D_1, \ldots, D_n\} = D_1 \sqcup \ldots \sqcup D_n$ and $\bigsqcup\emptyset = \bot$. By [4, Lemma 5.1], we have that $C'^{\mathcal{I}} = C^{\mathcal{I}}$ and $\mathsf{mdepth}(C') \leq \mathsf{mdepth}(C)$. The proof of [4, Lemma 5.1] also shows that $\mathsf{mdepth}(C_{i_j}) \leq \mathsf{mdepth}(C)$ for all $1 \leq j \leq k$.

Consider the moment after executing Step 4 of BBCL2-MiMoD (for $\mathcal{I}$). Observe that $\mathbb{C} = \{\neg C_{i_j} \mid Y_{i_j} \subseteq E^-\}$ and $E_0^- = E^-$. It follows that $\mathsf{mdepth}(\bigsqcap\mathbb{C}) \leq \mathsf{mdepth}(C)$ and $(\bigsqcap\mathbb{C})^{\mathcal{I}} = C'^{\mathcal{I}}$. Since $E_0^- = E^-$, the execution of BBCL2-MiMoD continues at Step 8, which does not modify $\mathbb{C}$, and then returns $\bigsqcap\mathbb{C}$ at Step 9.

For the sake of contradiction, suppose $\bigsqcap\mathbb{C}$ is not equivalent to C w.r.t. $\mathcal{R}$ and $\mathcal{T}$. Thus, either $\bigsqcap\mathbb{C} \sqcap \neg C$ or $C \sqcap \neg\bigsqcap\mathbb{C}$ is satisfiable w.r.t. $\mathcal{R}$ and $\mathcal{T}$. Both of them belong to $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$. Since $\mathcal{I}$ is a universal model of $\mathcal{R}$ and $\mathcal{T}$ w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger,d}$, it follows that either $(\bigsqcap\mathbb{C} \sqcap \neg C)^{\mathcal{I}}$ or $(C \sqcap \neg\bigsqcap\mathbb{C})^{\mathcal{I}}$ is not empty, which contradicts the fact $(\bigsqcap\mathbb{C})^{\mathcal{I}} = C'^{\mathcal{I}} = C^{\mathcal{I}}$. $\square$

5. Discussion and Conclusion

We first compare the BBCL2 method with the BBCL and dual-BBCL methods from our joint work [7]. First of all, BBCL2 is used for the second setting of concept learning in DLs, while BBCL and dual-BBCL are used for the first setting. BBCL2 is derived from dual-BBCL, but it contains substantial modifications needed for the change of setting. BBCL2 differs from BBCL at Steps 1, 4, 5, 7, 8, 9, and differs from dual-BBCL by the use of E_0^- at Steps 1, 4, 5 and 7. Comparing the examples given in [16] and [7], apart from detailed technical differences in concept learning, it can be seen that the first setting requires more knowledge[5] in order to obtain similar effects as the second setting. In other words, the second setting has effects of a kind of closed world assumption, while the first setting does not. The overall impression is that the second setting is more convenient than the first one.

Recall that our BBCL2 method is the *first bisimulation-based* method for concept learning in DLs using the second setting. As for the case of BBCL and dual-BBCL, it is formulated for the class of decidable $\mathcal{ALC}_{\Sigma,\Phi}$ DLs that have the finite or semi-finite model property, where $\Phi \subseteq \{I, O, F, N, Q, U, \mathsf{Self}\}$. This class contains many useful DLs. For example, $\mathcal{SROIQ}$ (the logical base of OWL 2) belongs to

[5] like the assertions $(\neg\exists cited\_by.\top)(P_1)$ and $(\forall cited\_by.\{P_2, P_3, P_4\})(P_5)$, which state that P_1 is not cited by any publication and P_5 is only cited by P_2, P_3 and P_4

this class. Our method is applicable also to other decidable DLs with the finite or semi-finite model property. The only additional requirement is that those DLs have a good set of selectors (in the sense of [17, Theorem 10]).

Like BBCL and dual-BBCL, the idea of BBCL2 is to use models of the considered knowledge base and bisimulation in those models to guide the search for the concept. Thus, it is completely different from the previous works [6, 10] on concept learning in DLs using the second setting. As bisimulation is the notion for characterizing indiscernibility of objects in DLs, our BBCL2 method is natural and very promising. Theorem 4.1 on C-learnability shows a good property of BBCL2. We intend to implement BBCL2 in the near future.

Acknowledgments

This work was supported by:

- Polish National Science Centre (NCN) under Grant No. 2011/01/B/ST6/02759,
- Polish National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information",
- Hue University under Grant No. DHH2013-01-41.

We would also like to thank the anonymous reviewers for helpful suggestions.

References

[1] Badea, L., Nienhuys-Cheng, S.-H.: A Refinement Operator for Description Logics, *Proceedings of ILP'2000*, 1866, Springer, 2000.

[2] Baldoni, M., Giordano, L., Martelli, A.: A Tableau for Multimodal Logics and Some (Un)Decidability Results, *Proceedings of TABLEAUX'1998*, 1397, Springer, 1998.

[3] Cohen, W., Hirsh, H.: Learning the Classic Description Logic: Theoretical and Experimental Results, *Proceedings of KR'1994*.

[4] Divroodi, A., Ha, Q.-T., Nguyen, L., Nguyen, H.: On C-Learnability in Description Logics, *Proceedings of ICCCI'2012 (1)*, 7653, Springer, 2012.

[5] Divroodi, A., Nguyen, L.: On Bisimulations for Description Logics, *Proceedings of CS&P'2011*.

[6] Fanizzi, N., d'Amato, C., Esposito, F.: DL-FOIL Concept Learning in Description Logics, *Proc. of ILP'2008*, 5194, Springer, 2008.

[7] 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, *Proceedings of SoICT'2012*, ACM, 2012.

[8] Iannone, L., Palmisano, I., Fanizzi, N.: An algorithm based on counterfactuals for concept learning in the Semantic Web, *Appl. Intell.*, **26**(2), 2007, 139–159.

[9] Lambrix, P., Larocchia, P.: Learning Composite Concepts, *Proc. of DL'1998*.

[10] Lehmann, J., Hitzler, P.: Concept learning in description logics using refinement operators, *Machine Learning*, **78**(1-2), 2010, 203–250.

[11] Nguyen, L.: An Efficient Tableau Prover using Global Caching for the Description Logic $\mathcal{ALC}$, *Fundamenta Informaticae*, **93**(1-3), 2009, 273–288.

[12] Nguyen, L., Nguyen, T.-B.-L., Szałas, A.: Horn-DL: An Expressive Horn Description Logic with PTime Data Complexity, *Proceedings of RR'2013*, 7994, Springer, 2013.

[13] Nguyen, L., Szałas, A.: Logic-based Roughification, in: *Rough Sets and Intelligent Systems (To the Memory of Professor Zdzisław Pawlak), Vol. 1* (A. Skowron, Z. Suraj, Eds.), Springer, 2012, 529–556.

[14] Pawlak, Z.: *Rough Sets. Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1991.

[15] Pawlak, Z., Skowron, A.: Rudiments of rough sets, *Inf. Sci.*, **177**(1), 2007, 3–27.

[16] Tran, T.-L., Ha, Q.-T., Hoang, T.-L.-G., Nguyen, L., Nguyen, H.: Bisimulation-Based Concept Learning in Description Logics, *Proceedings of CS&P'2013*, CEUR-WS.org, 2013.

[17] Tran, T.-L., Ha, Q.-T., Hoang, T.-L.-G., Nguyen, L., Nguyen, H., Szałas, A.: Concept Learning for Description Logic-Based Information Systems, *Proceedings of KSE'2012*, IEEE Computer Society, 2012.

[18] Tran, T.-L., Nguyen, L., Hoang, T.-L.-G.: A Domain Partitioning Method for Bisimulation-Based Concept Learning in Description Logics, *Proceedings of ICCSAMA'2014*, 282, Springer, 2014.