# Bisimulation-Based Concept Learning in Description Logics

Thanh-Luong Tran[1,3], Quang-Thuy Ha[2], Thi-Lan-Giao Hoang[1], Linh Anh Nguyen[3,2], and Hung Son Nguyen[3,2]

[1] Department of Information Technology,
Hue University of Sciences, 77 Nguyen Hue, Hue city, Vietnam
{ttluong,hlgiao}@hueuni.edu.vn
[2] Faculty of Information Technology,
VNU University of Engineering and Technology, 144 Xuan Thuy, Hanoi, Vietnam
thuyhq@vnu.edu.vn
[3] Faculty of Mathematics, Informatics and Mechanics,
University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
{nguyen,son}@mimuw.edu.pl

**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*.

## 1 Introduction

In this paper we continue our study [12, 15, 7, 4] on concept learning in description logics (DLs). This problem 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. The major settings of concept learning in DLs are as follows:

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 mentioned setting, while the works [6, 10] use the second mentioned 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 [12] 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 [13, 14]. In [15] Tran et al. generalized and extended the concept learning method of [12] 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 [12, 15] use the third mentioned setting. In [7] Ha et al. developed the first bisimulation-based method, called BBCL, for concept learning in DLs using the first mentioned 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 mentioned setting.

In this paper, we develop the first bisimulation-based method, called BBCL2, for concept learning in DLs using the second mentioned 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 mentioned 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 $a \in E^-$".

The rest of this paper is structured as follows. In Section 2, we recall notation and semantics of DLs. We present our BBCL2 method in Section 3 and illustrate it by examples in Section 4. We conclude 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 Notation and Semantics 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 $\Sigma_I$ is a set of *individuals*, $\Sigma_{dA}$ is a set of *discrete attributes*, $\Sigma_{nA}$ is a set of *numeric attributes*, $\Sigma_{oR}$ is a set of *object role names*, and $\Sigma_{dR}$ is a set of *data roles*.[4] All the sets $\Sigma_I$, $\Sigma_{dA}$, $\Sigma_{nA}$, $\Sigma_{oR}$, $\Sigma_{dR}$ 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.[5] (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 $\Sigma$.

An object role name stands for a binary predicate between individuals. A data role $\sigma$ 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 $\sigma$ and $\varrho$, 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*), $N$ (*unquantified number restriction*), $Q$ (*quantified number restriction*), $U$ (*universal role*), $\mathsf{Self}$ (*local reflexivity of an object role*). A *set of DL-features* is a set consisting of some or zero 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 *object roles* and *concepts* defined as follows:

- if $r \in \Sigma_{oR}$ then $r$ is an object role of $\mathcal{L}_{\Sigma,\Phi}$
- if $A \in \Sigma_C$ then $A$ is concept of $\mathcal{L}_{\Sigma,\Phi}$
- if $A \in \Sigma_A \setminus \Sigma_C$ and $d \in dom(A)$ then $A = d$ and $A \neq d$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
- 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 $\mathcal{L}_{\Sigma,\Phi}$
- 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 $\mathcal{L}_{\Sigma,\Phi}$
  - 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 $\mathcal{L}_{\Sigma,\Phi}$

[4] Object role names are atomic object roles.

[5] 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.

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

**Fig. 1.** Interpretation of complex object roles and complex concepts.

- if $\mathsf{Self} \in \Phi$ then $\exists r.\mathsf{Self}$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$.

An *interpretation* in $\mathcal{L}_{\Sigma,\Phi}$ 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 $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, each attribute $A \in \Sigma_A \setminus \Sigma_C$ with a partial function $A^\mathcal{I} : \Delta^\mathcal{I} \to \mathit{dom}(A)$, 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 $\sigma^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \mathit{range}(\sigma)$. 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 $\Gamma$.

Given an interpretation $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ 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:

- $x_k = x$ 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.

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 $U$. Given an interpretation $\mathcal{I}$, define that:

$$
\begin{array}{ll}
\mathcal{I} \models R_1 \circ \ldots \circ R_k \sqsubseteq r & \text{if } R_1^{\mathcal{I}} \circ \ldots \circ R_k^{\mathcal{I}} \subseteq r^{\mathcal{I}} \\
\mathcal{I} \models \mathtt{Ref}(r) & \text{if } r^{\mathcal{I}} \text{ is reflexive} \\
\mathcal{I} \models \mathtt{Irr}(r) & \text{if } r^{\mathcal{I}} \text{ is irreflexive} \\
\mathcal{I} \models \mathtt{Sym}(r) & \text{if } r^{\mathcal{I}} \text{ is symmetric} \\
\mathcal{I} \models \mathtt{Tra}(r) & \text{if } r^{\mathcal{I}} \text{ is transitive} \\
\mathcal{I} \models \mathtt{Dis}(R, S) & \text{if } R^{\mathcal{I}} \text{ and } S^{\mathcal{I}} \text{ are disjoint,}
\end{array}
$$

where the operator $\circ$ stands for the composition of binary relations. By a *role axiom* in $\mathcal{L}_{\Sigma,\Phi}$ we mean either a role inclusion axiom or a role assertion in $\mathcal{L}_{\Sigma,\Phi}$. We say that a role axiom $\varphi$ is *valid* in $\mathcal{I}$ (or $\mathcal{I}$ *validates* $\varphi$) if $\mathcal{I} \models \varphi$.

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 $\mathcal{L}_{\Sigma,\Phi}$. An interpretation $\mathcal{I}$ *validates* an axiom $C \sqsubseteq D$, denoted by $\mathcal{I} \models C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

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 $\mathcal{L}_{\Sigma,\Phi}$. We will write, for example, $A(a) = d$ instead $(A = d)(a)$. Given an interpretation $\mathcal{I}$, define that:

$$
\begin{array}{ll}
\mathcal{I} \models a = b & \text{if } a^{\mathcal{I}} = b^{\mathcal{I}} \\
\mathcal{I} \models a \neq b & \text{if } a^{\mathcal{I}} \neq b^{\mathcal{I}} \\
\mathcal{I} \models C(a) & \text{if } a^{\mathcal{I}} \in C^{\mathcal{I}} \\
\mathcal{I} \models r(a, b) & \text{if } \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in r^{\mathcal{I}} \\
\mathcal{I} \models \neg r(a, b) & \text{if } \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \notin r^{\mathcal{I}}.
\end{array}
$$

We say that $\mathcal{I}$ *validates* an individual assertion $\varphi$ if $\mathcal{I} \models \varphi$.

An *RBox* (resp. *TBox*, *ABox*) in $\mathcal{L}_{\Sigma,\Phi}$ is a finite set of role axioms (resp. terminological axioms, individual assertions) in $\mathcal{L}_{\Sigma,\Phi}$. 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 $\mathcal{L}_{\Sigma,\Phi}$. An interpretation $\mathcal{I}$ is a *model* of a "box" 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 $KB$, $a^{\mathcal{I}} \in C^{\mathcal{I}}$.

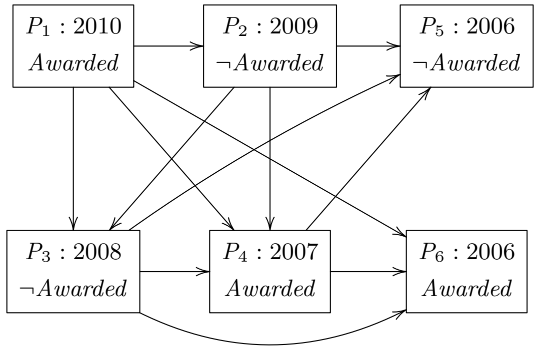


**Fig. 2.** An illustration for the knowledge base given in Example 2.1

*Example 2.1.* This example is based on an example of [15, 7]. Let

$$\begin{gathered}
\Phi = \{I, O, N, Q\}, \quad \Sigma_I = \{P_1, P_2, P_3, P_4, P_5, P_6\}, \quad \Sigma_C = \{Pub, Awarded, A_d\}, \\
\Sigma_{dA} = \Sigma_C, \quad \Sigma_{nA} = \{Year\}, \quad \Sigma_{oR} = \{cites, cited_by\}, \quad \Sigma_{dR} = \emptyset, \\
\mathcal{R} = \{cites^- \sqsubseteq cited_by, \quad cited_by^- \sqsubseteq cites\}, \quad \mathcal{T} = \{\top \sqsubseteq Pub\}, \\
\mathcal{A}_0 = \{Awarded(P_1), \neg Awarded(P_2), \neg Awarded(P_3), Awarded(P_4), \\
\neg Awarded(P_5), Awarded(P_6), Year(P_1) = 2010, Year(P_2) = 2009, \\
Year(P_3) = 2008, Year(P_4) = 2007, Year(P_5) = 2006, Year(P_6) = 2006, \\
cites(P_1, P_2), cites(P_1, P_3), cites(P_1, P_4), cites(P_1, P_6), \\
cites(P_2, P_3), cites(P_2, P_4), cites(P_2, P_5), cites(P_3, P_4), \\
cites(P_3, P_5), cites(P_3, P_6), cites(P_4, P_5), cites(P_4, P_6)\},
\end{gathered}$$

where the concept *Pub* stands for "publication". Then $KB_0 = \langle \mathcal{R}, \mathcal{T}, \mathcal{A}_0 \rangle$ is a knowledge base in $\mathcal{L}_{\Sigma,\Phi}$. The axiom $\top \sqsubseteq Pub$ states that the domain of any model of $KB_0$ consists of only publications. The knowledge base $KB_0$ is illustrated in Figure 2 (on page 426). In this figure, nodes denote publications and edges denote citations (i.e., assertions of the role *cites*), and we display only information concerning assertions about *Year*, *Awarded* and *cites*. $\lhd$

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 constructable.

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.

## 3 Concept Learning for Knowledge Bases in DLs

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 $\Sigma_I$ 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 $a \in E^-$.

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 $a \in E^-$.

The idea is to use models of $KB$ and bisimulation in those models to guide the search for $C$. We now describe our method *BBCL2* (*Bisimulation-Based Concept Learning* for knowledge bases in DLs using the *second* setting). It 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 that:

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

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. Starting from the partition $\{\Delta^{\mathcal{I}}\}$, make subsequent granulations to reach the partition $\{Y_{i_1}, \ldots, Y_{i_k}\}$ corresponding to $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$, where each $Y_{i_j}$ is characterized by an appropriate concept $C_{i_j}$ (such that $Y_{i_j} = C_{i_j}^{\mathcal{I}}$).
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 $KB \not\models (\bigsqcap \mathbb{C})(a)$, 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 stop 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 $C$ be a normalized form of $\bigsqcap \mathbb{C}$.[6] Observe that $KB \models C(a)$ for all $a \in E^+$, and $KB \not\models C(a)$ for all $a \in E^-$. Try to simplify $C$ 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.

We describe Step 3 in more details:

- 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

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

- $A$, 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 $r \in \Sigma_{oR}^\dagger$

**Fig. 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 [15] that using these selectors is sufficient to granulate $\Delta^\mathcal{I}$ to obtain the partition corresponding to $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$.

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 [12, 15] we use the concepts listed in Figure 3 (on page 429) 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 partitioning $Y_{i_j}$ 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 partitioned first and which selector should be used to partition it are left open for heuristics. For example, one can apply some gain function like the entropy gain measure, while taking into account also simplicity of selectors and the concepts characterizing the blocks. Once again, 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.

As a modification for Step 3, the granulation process can be stopped 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 [12], 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 $\mathbb{C}$.

## 4 Illustrative Examples

*Example 4.1.* Let $KB_0 = \langle \mathcal{R}, \mathcal{T}, \mathcal{A}_0 \rangle$ be the knowledge base given in Example 2.1. Let $E^+ = \{P_4, P_6\}$, $E^- = \{P_1, P_2, P_3, P_5\}$, $\Sigma^\dagger = \{Awarded, cited\_by\}$ 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 our 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 will be used first:

$$\Delta^{\mathcal{I}} = \{P_1, P_2, P_3, P_4, P_5, P_6\}, \;\; x^{\mathcal{I}} = x \text{ for } x \in \{P_1, P_2, P_3, P_4, P_5, P_6\},$$
$$Pub^{\mathcal{I}} = \Delta^{\mathcal{I}}, \;\; Awarded^{\mathcal{I}} = \{P_1, P_4, P_6\},$$
$$cites^{\mathcal{I}} = \{\langle P_1, P_2\rangle, \langle P_1, P_3\rangle, \langle P_1, P_4\rangle, \langle P_1, P_6\rangle, \langle P_2, P_3\rangle, \langle P_2, P_4\rangle,$$
$$\langle P_2, P_5\rangle, \langle P_3, P_4\rangle, \langle P_3, P_5\rangle, \langle P_3, P_6\rangle, \langle P_4, P_5\rangle, \langle P_4, P_6\rangle\},$$
$$cited\_by^{\mathcal{I}} = (cites^{\mathcal{I}})^{-1}, \;\; \text{the function } Year^{\mathcal{I}} \text{ is specified as usual.}$$

3. $Y_1 := \Delta^{\mathcal{I}}$, $partition := \{Y_1\}$
4. Partitioning $Y_1$ by *Awarded*:
   - $Y_2 := \{P_1, P_4, P_6\}$, $C_2 := Awarded$,
   - $Y_3 := \{P_2, P_3, P_5\}$, $C_3 := \neg Awarded$,
   - $partition := \{Y_2, Y_3\}$.
5. Partitioning $Y_2$:
   - All the selectors $\exists cited\_by.\top$, $\exists cited\_by.C_2$ and $\exists cited\_by.C_3$ partition $Y_2$ in the same way. We choose $\exists cited\_by.\top$, as it is the simplest one.
   - $Y_4 := \{P_4, P_6\}$, $C_4 := C_2 \sqcap \exists cited\_by.\top$,

– $Y_5 := \{P_1\}$, $C_5 := C_2 \sqcap \neg\exists cited\_by.\top$,
– $partition := \{Y_3, Y_4, Y_5\}$.

6. The obtained partition is consistent with $E$, having $Y_3 = \{P_2, P_3, P_5\} \subset E^-$, $Y_4 = \{P_4, P_6\} = E^+$ and $Y_5 = \{P_1\} \subset E^-$. (It is not yet the partition corresponding to $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$.)
7. Since $Y_3 \subset E^-$ and $KB \models \neg C_3(a)$ for all $a \in E^+$, we add $\neg C_3$ to $\mathbb{C}$ and add the elements of $Y_3$ to $E_0^-$. Thus, $\mathbb{C} = \{\neg C_3\}$ and $E_0^- = \{P_2, P_3, P_5\}$.
8. Since $Y_5 \subset E^-$ and $KB \models \neg C_5(a)$ for all $a \in E^+$ 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, $\mathbb{C} = \{\neg C_3, \neg C_5\}$, $\bigsqcap \mathbb{C} = \neg\neg Awarded \sqcap \neg(Awarded \sqcap \neg\exists cited\_by.\top)$ and $E_0^- = \{P_1, P_2, P_3, P_5\}$.
9. Since $E_0^- = E^-$, we normalize $\bigsqcap \mathbb{C}$ to $Awarded \sqcap \exists cited\_by.\top$ and return it as the result. (This concept denotes the set of publications which were awarded and cited.) ◁

*Example 4.2.* Let $KB_0$, $E^+$, $E^-$, $KB$ and $\Phi^\dagger$ be as in Example 4.1, but let $\Sigma^\dagger = \{cited\_by, Year\}$. Execution of the BBCL2 method on this new example has the same first two steps as in Example 4.1, and then continues as follows.

1. Granulating $\{\Delta^\mathcal{I}\}$ as in [15, Example 11] we reach the partition $\{Y_4, Y_6, Y_7, Y_8, Y_9\}$ consistent with $E$ and have that:
   – $Y_4 = \{P_4\}$, $Y_6 = \{P_1\}$, $Y_7 = \{P_2, P_3\}$, $Y_8 = \{P_6\}$, $Y_9 = \{P_5\}$,
   – $C_2 = (Year \geq 2008)$, $C_3 = (Year < 2008)$,
   $C_5 = C_3 \sqcap (Year < 2007)$, $C_6 = C_2 \sqcap (Year \geq 2010)$,
   $C_7 = C_2 \sqcap (Year < 2010)$, $C_9 = C_5 \sqcap \neg\exists cited\_by.C_6$.
2. We have $C_6 = (Year \geq 2008) \sqcap (Year \geq 2010)$. Since $Y_6 \subset E^-$ and $KB \models \neg C_6(a)$ for all $a \in E^+$, we add $\neg C_6$ to $\mathbb{C}$ and add the elements of $Y_6$ to $E_0^-$. Thus, $\mathbb{C} = \{\neg C_6\}$ and $E_0^- = \{P_1\}$.
3. We have $C_7 := (Year \geq 2008) \sqcap (Year < 2010)$. Since $Y_7 \subset E^-$ and $KB \models \neg C_7(a)$ for all $a \in E^+$ and $\bigsqcap \mathbb{C}$ is not subsumed by $\neg C_7$ w.r.t. $KB$, we add $\neg C_7$ to $\mathbb{C}$ and add the elements of $Y_7$ to $E_0^-$. Thus, $\mathbb{C} = \{\neg C_6, \neg C_7\}$ and $E_0^- = \{P_1, P_2, P_3\}$.
4. We have $C_9 := (Year < 2008) \sqcap (Year < 2007) \sqcap \neg\exists cited\_by.((Year \geq 2008) \sqcap (Year \geq 2010))$. Since $Y_9 \subset E^-$ and $KB \models \neg C_9(a)$ for all $a \in E^+$ and $\bigsqcap \mathbb{C}$ is not subsumed by $\neg C_9$ w.r.t. $KB$, we add $\neg C_9$ to $\mathbb{C}$ and add the elements of $Y_9$ to $E_0^-$. Thus, $\mathbb{C} = \{\neg C_6, \neg C_7, \neg C_9\}$ and $E_0^- = \{P_1, P_2, P_3, P_5\}$.
5. Since $E_0^- = E^-$, we normalize and simplify $\bigsqcap \mathbb{C}$ before returning it as the result. Without exploiting the fact that publication years are integers, $\bigsqcap \mathbb{C}$ can be normalized to

$$(Year < 2008) \sqcap [(Year \geq 2007) \sqcup \exists cited\_by.(Year \geq 2010)].$$

   $C = (Year < 2008) \sqcap \exists cited\_by.(Year \geq 2010)$ is a simplified form of the above concept, which still satisfies that $KB \models C(a)$ for all $a \in E^+$ and $KB \not\models C(a)$ for all $a \in E^-$. Thus, we return it as the result. (The returned concept denotes the set of publications released before 2008 that are cited by some publications released since 2010.) ◁

## 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 this paper and in [7], apart from detailed technical differences in concept learning, it can be seen that the first setting requires more knowledge[7] 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 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 [15, 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. We intend to implement BBCL2 in the near future.

## Acknowledgments

This paper was written during the first author's visit at Warsaw Center of Mathematics and Computer Science (WCMCS). He would like to thank WCMCS for the support. This work was also supported by Polish National Science Centre (NCN) under Grant No. 2011/01/B/ST6/02759 as well as by Polish National Center 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".

## References

1. L. Badea and S.-H. Nienhuys-Cheng. A refinement operator for description logics. In *Proceedings of ILP'2000*, volume 1866 of *LNCS*, pages 40–59. Springer, 2000.

[7] 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 $P_4$

2. M. Baldoni, L. Giordano, and A. Martelli. A tableau for multimodal logics and some (un)decidability results. In *Proceedings of TABLEAUX'1998*, volume 1397 of *LNCS*, pages 44–59. Springer, 1998.
3. W.W. Cohen and H. Hirsh. Learning the Classic description logic: Theoretical and experimental results. In *Proceedings of KR'1994*, pages 121–133.
4. A.R. Divroodi, Q.-T. Ha, L.A. Nguyen, and H.S. Nguyen. On C-learnability in description logics. In *Proceedings of ICCCI'2012 (1)*, volume 7653 of *LNCS*, pages 230–238. Springer, 2012.
5. A.R. Divroodi and L.A. Nguyen. On bisimulations for description logics. In *Proceedings of CS&P'2011*, pages 99–110 (see also arXiv:1104.1964).
6. N. Fanizzi, C. d'Amato, and F. Esposito. DL-FOIL concept learning in description logics. In *Proc. of ILP'2008*, volume 5194 of *LNCS*, pages 107–121. Springer, 2008.
7. Q.-T. Ha, T.-L.-G. Hoang, L.A. Nguyen, H.S. Nguyen, A. Szałas, and T.-L. Tran. A bisimulation-based method of concept learning for knowledge bases in description logics. In *Proceedings of SoICT'2012*, pages 241–249. ACM, 2012.
8. L. Iannone, I. Palmisano, and N. Fanizzi. An algorithm based on counterfactuals for concept learning in the Semantic Web. *Appl. Intell.*, 26(2):139–159, 2007.
9. P. Lambrix and P. Larocchia. Learning composite concepts. In *Proc. of DL'1998*.
10. J. Lehmann and P. Hitzler. Concept learning in description logics using refinement operators. *Machine Learning*, 78(1-2):203–250, 2010.
11. L.A. Nguyen. An efficient tableau prover using global caching for the description logic $\mathcal{ALC}$. *Fundamenta Informaticae*, 93(1-3):273–288, 2009.
12. L.A. Nguyen and A. Szałas. Logic-based roughification. In A. Skowron and Z. Suraj, editors, *Rough Sets and Intelligent Systems (To the Memory of Professor Zdzisław Pawlak), Vol. 1*, pages 529–556. Springer, 2012.
13. Z. Pawlak. *Rough Sets. Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, Dordrecht, 1991.
14. Z. Pawlak and A. Skowron. Rudiments of rough sets. *Inf. Sci.*, 177(1):3–27, 2007.
15. T.-L. Tran, Q.-T. Ha, T.-L.-G. Hoang, L.A. Nguyen, H.S. Nguyen, and A. Szałas. Concept learning for description logic-based information systems. In *Proceedings of KSE'2012*, pages 65–73. IEEE Computer Society, 2012.