A Bisimulation-based Method of Concept Learning for Knowledge Bases in Description Logics

Quang-Thuy Ha*, Thi-Lan-Giao Hoang†, Linh Anh Nguyen‡,
Hung Son Nguyen‡, Andrzej Szałas‡§ and Thanh-Luong Tran†

*Faculty of Information Technology, College of Technology, Vietnam National University
144 Xuan Thuy, Hanoi, Vietnam
Email: thuyhq@vnu.edu.vn

†Department of Information Technology, College of Sciences, Hue University
77 Nguyen Hue, Hue city, Vietnam
Email: ttluong@hueuni.edu.vn, hlgiao@hueuni.edu.vn

‡Faculty of Mathematics, Informatics and Mechanics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
Email: {nguyen,son,andsz}@mimuw.edu.pl

§Dept. of Computer and Information Science, Linköping University
SE-581 83 Linköping, Sweden

***Abstract*— We develop the first bisimulation-based method of concept learning, called BBCL, for knowledge bases in description logics (DLs). Our method is formulated for a large class of useful DLs, with well-known DLs like $\mathcal{ALC}$, $\mathcal{SHIQ}$, $\mathcal{SHOIQ}$, $\mathcal{SROIQ}$. As bisimulation is the notion for characterizing indiscernibility of objects in DLs, our method is natural and very promising.**

I. Introduction

Description logics (DLs) are formal languages suitable for representing terminological knowledge [1]. They are of particular importance in providing a logical formalism for ontologies and the Semantic Web. In DLs the domain of interest is described in terms of individuals (objects), concepts, object roles and data roles. A concept stands for a set of objects, an object role stands for a binary relation between objects, and a data role stands for a binary predicate relating objects to data values. Complex concepts are built from concept names, role names and individual names by using constructors. A knowledge base in a DL consists of role axioms, terminological axioms and assertions about individuals.

In this paper we study concept learning in 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 relationship 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^+ contains positive examples of C, while E^- contains negative ones.
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 $\mathcal{I} \models \neg C(a)$.

A. Previous Work on Concept Learning in DLs

Concept learning in DLs has been studied by a considerable number of researchers [2], [3], [4], [5], [6], [7], [8], [9] (see also [10], [11], [12], [13], [14] for works on related problems).

As an early work on concept learning in DLs, Cohen and Hirsh [2] studied PAC-learnability of the CLASSIC description logic (an early DL formalism) and its sublogic called C-CLASSIC. They proposed a concept learning algorithm called LCSLearn, which is based on “least common subsumers”. In [3] Lambrix and Larocchia proposed a simple concept learning algorithm based on concept normalization.

Badea and Nienhuys-Cheng [4], Iannone et al. [5], Fanizzi et al. [6], Lehmann and Hitzler [7] studied concept learning in DLs by using refinement operators as in inductive logic programming. The works [4], [5] use the first mentioned setting, while the works [6], [7] 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 [7] exploits genetic programming techniques, while DL-FOIL [6] considers also

SoICT 2012, August 23-24, 2012, Ha-Long, Vietnam.

unlabeled data as in semi-supervised learning. A comparison between DL-Learner [7], YinYang [5] and LCSLearn [2] can be found in Hellmann's master thesis [15].

Nguyen and Szałas [8] applied bisimulation in DLs [16] to model indiscernibility of objects. Their work is pioneering in using bisimulation for concept learning in DLs. It concerns also concept approximation by using bisimulation and Pawlak's rough set theory [17], [18]. In [9] we generalized and extended the concept learning method of [8] for DL-based information systems. We took attributes as basic elements of the language. An information system in a DL is a finite interpretation in that logic. It can be given explicitly or specified somehow, e.g., by a knowledge base in the rule language OWL 2 RL$^+$ [19] (using the standard semantics) or WORL [20] (using the well-founded semantics) or SWORL [20] (using the stratified semantics) or by an acyclic knowledge base [9] (using the closed world assumption). Thus, both the works [8], [9] use the third mentioned setting.

B. Contributions of This Paper

In this paper, we develop the first bisimulation-based method, called BBCL, for concept learning in DLs using the first mentioned setting, i.e., for learning a concept C such that:

- $KB \models C(a)$ for all $a \in E^+$, and
- $KB \models \neg C(a)$ 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.

The idea is to use models of KB and bisimulation in those models to guide the search for C. Our method is formulated for a large class of useful DLs, with well-known DLs like $\mathcal{ALC}$, $\mathcal{SHIQ}$, $\mathcal{SHOIQ}$, $\mathcal{SROIQ}$. As bisimulation is the notion for characterizing indiscernibility of objects in DLs, our method is natural and very promising.

Our method is completely different from the ones of [4], [5], [6], [7], as it is based on bisimulation, while all the latter ones are based on refinement operators as in inductive logic programming. This work also differs essentially from the work [8] by Nguyen and Szałas and our previous work [9] because the setting is different: while in [8], [9] concept learning is done on the basis of a given interpretation (and examples of the concept to be learned), in the current work concept learning is done on the basis of a given knowledge base, which may have many models.

C. The Structure of the Rest of This Paper

In Section II, we first present notation and define semantics of DLs, and then recall bisimulation in DLs and its properties concerning indiscernibility. We present our BBCL method in Section III and illustrate it by examples in Section IV. We conclude in Section V.

II. Preliminaries

A. 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 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 called 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 (*unquantified number restriction*), Q (*quantified 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.

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 recursively 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 $\mathcal{L}_{\Sigma,\Phi}$
  - if Self $\in \Phi$ then $\exists r.\mathsf{Self}$ 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.

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

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

If $\mathbb{C} = \{C_1, \ldots, C_n\}$ is a finite set of concepts then by $\bigsqcup \mathbb{C}$ we denote $C_1 \sqcup \ldots \sqcup C_n$. Let's assume that $\bigsqcup \emptyset = \bot$.

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 $A^\mathcal{I} : \Delta^\mathcal{I} \to dom(A)$, 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 $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ 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:

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

where the operator $\circ$ stands for the composition of 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 φ is *valid* in $\mathcal{I}$ (or $\mathcal{I}$ *validates* φ) if $\mathcal{I} \models \varphi$.

An *RBox* in $\mathcal{L}_{\Sigma,\Phi}$ is a finite set of role axioms in $\mathcal{L}_{\Sigma,\Phi}$. An interpretation $\mathcal{I}$ is a *model* of an RBox $\mathcal{R}$, denoted by $\mathcal{I} \models \mathcal{R}$, if it validates all the role axioms of $\mathcal{R}$.

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 $C^\mathcal{I} \subseteq D^\mathcal{I}$.

A *TBox* in $\mathcal{L}_{\Sigma,\Phi}$ is a finite set of terminological axioms in $\mathcal{L}_{\Sigma,\Phi}$. An interpretation $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$, denoted by $\mathcal{I} \models \mathcal{T}$, if it validates all the axioms of $\mathcal{T}$.

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 $\mathcal{L}_{\Sigma,\Phi}$. Given an interpretation $\mathcal{I}$, define that:

$\mathcal{I} \models a = b$	if	$a^\mathcal{I} = b^\mathcal{I}$
$\mathcal{I} \models a \neq b$	if	$a^\mathcal{I} \neq b^\mathcal{I}$
$\mathcal{I} \models C(a)$	if	$C^\mathcal{I}(a^\mathcal{I})$ holds
$\mathcal{I} \models r(a,b)$	if	$r^\mathcal{I}(a^\mathcal{I}, b^\mathcal{I})$ holds
$\mathcal{I} \models \neg r(a,b)$	if	$r^\mathcal{I}(a^\mathcal{I}, b^\mathcal{I})$ does not hold.

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

An *ABox* in $\mathcal{L}_{\Sigma,\Phi}$ is a finite set of individual assertions in $\mathcal{L}_{\Sigma,\Phi}$. An interpretation $\mathcal{I}$ is a *model* of an ABox $\mathcal{A}$, denoted by $\mathcal{I} \models \mathcal{A}$, if it satisfies all the assertions of $\mathcal{A}$.

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 knowledge base $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ if it is a model of all $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{A}$. A knowledge base is *satisfiable* if it has a model. An individual a is said to be an

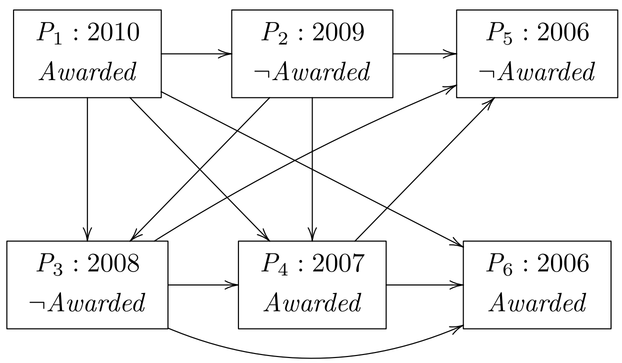

Fig. 2. An illustration for the knowledge base given in Example 1

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, $a^\mathcal{I} \in C^\mathcal{I}$.

Example 1: This example is about publications. It is based on an example of [9]. Let

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

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 assertion $(\neg\exists cited\_by.\top)(P_1)$ states that P_1 is not cited by any publication, and the assertion $(\forall cited\_by.\{P_2, P_3, P_4\})(P_5)$ states that P_5 is cited only by P_2, P_3 and P_4. The knowledge base KB_0 is illustrated in Figure 2 (on page 4). In the 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 $cites$. ◁

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 satisfiablity 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 [21], 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 $\mathcal{SROIQ}$ [22] - the logical base of the Web Ontology Language OWL 2. This logic has the semi-finite model property.

B. Bisimulation and Indiscernibility

Indiscernibility in DLs is related to bisimulation. In [16] Divroodi and Nguyen studied bisimulations for a number of DLs. In [8] Nguyen and Szałas generalized that notion to model indiscernibility of objects and study concept learning. In [9] we generalized their notion of bisimulation further for dealing with attributes, data roles, unquantified number restrictions and role functionality. The classes of DLs studied in [16], [8], [9] allow object role constructors of $\mathcal{ALC}_{reg}$, which correspond to program constructors of PDL (propositional dynamic logic). In this paper we omit such object role constructors, and the class of DLs studied here is the subclass of the one studied in [9] obtained by adopting that restriction. The conditions for bisimulation remain the same, as the object role constructors of $\mathcal{ALC}_{reg}$ are "safe" for these conditions. We recall them below. Let:

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

A binary relation $Z \subseteq \Delta^\mathcal{I} \times \Delta^{\mathcal{I}'}$ is called an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-*bisimulation* between $\mathcal{I}$ and $\mathcal{I}'$ if the following conditions hold for every $a \in \Sigma_I^\dagger$, $A \in \Sigma_C^\dagger$, $B \in \Sigma_A^\dagger \setminus \Sigma_C^\dagger$, $r \in \Sigma_{oR}^\dagger$, $\sigma \in \Sigma_{dR}^\dagger$, $d \in range(\sigma)$, $x, y \in \Delta^\mathcal{I}$, $x', y' \in \Delta^{\mathcal{I}'}$:

$$Z(a^\mathcal{I}, a^{\mathcal{I}'}) \tag{1}$$
$$Z(x, x') \Rightarrow [A^\mathcal{I}(x) \Leftrightarrow A^{\mathcal{I}'}(x')] \tag{2}$$
$$Z(x, x') \Rightarrow [B^\mathcal{I}(x) = B^{\mathcal{I}'}(x') \text{ or both are undefined}] \tag{3}$$
$$[Z(x, x') \wedge r^\mathcal{I}(x, y)] \Rightarrow \exists y' \in \Delta^{\mathcal{I}'}[Z(y, y') \wedge r^{\mathcal{I}'}(x', y')] \tag{4}$$
$$[Z(x, x') \wedge r^{\mathcal{I}'}(x', y')] \Rightarrow \exists y \in \Delta^\mathcal{I}[Z(y, y') \wedge r^\mathcal{I}(x, y)] \tag{5}$$
$$Z(x, x') \Rightarrow [\sigma^\mathcal{I}(x, d) \Leftrightarrow \sigma^{\mathcal{I}'}(x', d)], \tag{6}$$

if $I \in \Phi^\dagger$ then

$$[Z(x, x') \wedge r^\mathcal{I}(y, x)] \Rightarrow \exists y' \in \Delta^{\mathcal{I}'}[Z(y, y') \wedge r^{\mathcal{I}'}(y', x')] \tag{7}$$
$$[Z(x, x') \wedge r^{\mathcal{I}'}(y', x')] \Rightarrow \exists y \in \Delta^\mathcal{I}[Z(y, y') \wedge r^\mathcal{I}(y, x)], \tag{8}$$

if $O \in \Phi^\dagger$ then

$$Z(x, x') \Rightarrow [x = a^\mathcal{I} \Leftrightarrow x' = a^{\mathcal{I}'}], \tag{9}$$

if $N \in \Phi^\dagger$ then

$$Z(x,x') \Rightarrow \#\{y \mid r^\mathcal{I}(x,y)\} = \#\{y' \mid r^{\mathcal{I}'}(x',y')\}, \quad (10)$$

if $\{N, I\} \subseteq \Phi^\dagger$ then (additionally)

$$Z(x,x') \Rightarrow \#\{y \mid r^\mathcal{I}(y,x)\} = \#\{y' \mid r^{\mathcal{I}'}(y',x')\}, \quad (11)$$

if $F \in \Phi^\dagger$ then

$$Z(x,x') \Rightarrow [\#\{y \mid r^\mathcal{I}(x,y)\} \leq 1 \Leftrightarrow \#\{y' \mid r^{\mathcal{I}'}(x',y')\} \leq 1], \quad (12)$$

if $\{F, I\} \subseteq \Phi^\dagger$ then (additionally)

$$Z(x,x') \Rightarrow [\#\{y \mid r^\mathcal{I}(y,x)\} \leq 1 \Leftrightarrow \#\{y' \mid r^{\mathcal{I}'}(y',x')\} \leq 1], \quad (13)$$

if $Q \in \Phi^\dagger$ then

if $Z(x,x')$ holds then, for every $r \in \Sigma^\dagger_{oR}$, there exists a bijection $h : \{y \mid r^\mathcal{I}(x,y)\} \to \{y' \mid r^{\mathcal{I}'}(x',y')\}$ such that $h \subseteq Z$, (14)

if $\{Q, I\} \subseteq \Phi^\dagger$ then (additionally)

if $Z(x,x')$ holds then, for every $r \in \Sigma^\dagger_{oR}$, there exists a bijection $h : \{y \mid r^\mathcal{I}(y,x)\} \to \{y' \mid r^{\mathcal{I}'}(y',x')\}$ such that $h \subseteq Z$, (15)

if $U \in \Phi^\dagger$ then

$$\forall x \in \Delta^\mathcal{I} \, \exists x' \in \Delta^{\mathcal{I}'} \, Z(x,x') \quad (16)$$

$$\forall x' \in \Delta^{\mathcal{I}'} \, \exists x \in \Delta^\mathcal{I} \, Z(x,x'), \quad (17)$$

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

$$Z(x,x') \Rightarrow [r^\mathcal{I}(x,x) \Leftrightarrow r^{\mathcal{I}'}(x',x')]. \quad (18)$$

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

Given an interpretation $\mathcal{I}$ in $\mathcal{L}_{\Sigma,\Phi}$, by $\sim_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$ we denote the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of $\mathcal{I}$, and by $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$ we denote the binary 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 C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, $x \in C^\mathcal{I}$ iff $x' \in C^\mathcal{I}$) .

An interpretation $\mathcal{I}$ is *finitely branching* (or *image-finite*) w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ if, for every $x \in \Delta^\mathcal{I}$ and every $r \in \Sigma^\dagger_{oR}$:

- the set $\{y \in \Delta^\mathcal{I} \mid r^\mathcal{I}(x,y)\}$ is finite
- if $I \in \Phi^\dagger$ then the set $\{y \in \Delta^\mathcal{I} \mid r^\mathcal{I}(y,x)\}$ is finite.

Theorem 2: Let Σ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$, Φ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$, and $\mathcal{I}$ be an interpretation in $\mathcal{L}_{\Sigma,\Phi}$. Then:

1) *the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of $\mathcal{I}$ exists and is an equivalence relation*
2) *if $\mathcal{I}$ is finitely branching w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ then the relation $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$ is the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of $\mathcal{I}$ (i.e. the relations $\equiv_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$ and $\sim_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$ coincide).* $\triangleleft$

This theorem differs from the one of [8], [9] only in the studied class of DLs. It can be proved analogously to [16, Proposition 5.1 and Theorem 5.2].

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

Theorem 3: Let $\mathcal{I}$ be an interpretation in $\mathcal{L}_{\Sigma,\Phi}$, and let $X \subseteq \Delta^\mathcal{I}$, $\Sigma^\dagger \subseteq \Sigma$ and $\Phi^\dagger \subseteq \Phi$. Then:

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

This theorem differs from the one of [8], [9] only in the studied class of DLs. It can be proved analogously to [8, Theorem 4].

III. 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 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 $\Phi^\dagger \subseteq \Phi$. The concept C should satisfy the following conditions:

- $KB \models C(a)$ for all $a \in E^+$
- $KB \models \neg C(a)$ for all $a \in E^-$.

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 the first assertion of Theorem 3, $C^\mathcal{I}$ should be the union of a number of equivalence classes of $\Delta^\mathcal{I}$ w.r.t. $\sim_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$
- 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 $a \in E^-$.

Our idea is to use models of KB and bisimulation in those models to guide the search for C. Here is our method, named *BBCL* (Bisimulation-Based Concept Learning for knowledge bases in DLs):

1) Initialize $\mathbb{C} := \emptyset$ and $\mathbb{C}_0 := \emptyset$. (The meaning of $\mathbb{C}$ is to collect concepts D such that $KB \models \neg D(a)$ for all $a \in E^-$. The set $\mathbb{C}_0$ is auxiliary for constructing $\mathbb{C}$. In the case when a concept D does not satisfy the mentioned condition but is a "good" candidate for that, we put it

- 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. In [9] we proved that it suffices to use these selectors for granulating $\Delta^\mathcal{I}$ in order to reach the partition corresponding to $\sim_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$.

into $\mathbb{C}_0$. Later, when necessary, we take conjunctions of some concepts from $\mathbb{C}_0$ and check whether they are good for adding into $\mathbb{C}$.)

2) (This is the beginning of a loop controlled by "go to".) 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). If L is one of the well known DLs, then $\mathcal{I}$ can be constructed by using tableau algorithms, e.g., [23] (for $\mathcal{ALC}$), [24] (for $\mathcal{ALCI}$), [25] (for $\mathcal{SH}$), [26], [27] (for $\mathcal{SHI}$), [28] (for $\mathcal{SHIQ}$), [29] (for $\mathcal{SHOIQ}$) and [22] (for $\mathcal{SROIQ}$). 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.

3) Starting from the partition $\{\Delta^\mathcal{I}\}$, make subsequent granulations to reach the partition corresponding to $\sim_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$.

   - The granulation process can be stopped as soon as the current partition is consistent with E (or when some criteria are met).

   - 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 [8], [9] we use the concepts listed in Figure 3 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.

4) Let $\{Y_{i_1}, \ldots, Y_{i_k}\}$ be the resulting partition of the above step. 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 C_{i_j} is not subsumed by $\bigsqcup\mathbb{C}$ w.r.t. KB (i.e. $KB \not\models (C_{i_j} \sqsubseteq \bigsqcup\mathbb{C})$) then add C_{i_j} into $\mathbb{C}$
   - else add C_{i_j} into $\mathbb{C}_0$.

5) If $KB \models (\bigsqcup\mathbb{C})(a)$ for all $a \in E^+$ 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 $\mathcal{I}$) even after tightening the strategy for Step 3 by requiring reaching the partition corresponding to $\sim_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$ before stopping the granulation process, then go to Step 7, else go to Step 2 to repeat the loop.

7) Repeat the following:
   - Randomly select some concepts $D_1, \ldots, D_l$ from $\mathbb{C}_0$ and let $D = (D_1 \sqcap \ldots \sqcap D_l)$.
   - If $KB \models \neg D(a)$ for all $a \in E^-$ and D is not subsumed by $\bigsqcup\mathbb{C}$ w.r.t. KB (i.e., $KB \not\models (D \sqsubseteq \bigsqcup\mathbb{C})$) then:
     - add D into $\mathbb{C}$
     - if $KB \models (\bigsqcup\mathbb{C})(a)$ for all $a \in E^+$ then go to Step 8.
   - 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 every $D \in \mathbb{C}$, if $KB \models \bigsqcup(\mathbb{C} \setminus \{D\})(a)$ for all $a \in E^+$ then delete D from $\mathbb{C}$.

9) Let C be a normalized form of $\bigsqcup\mathbb{C}$. (Normalizing concepts can be done as in [30].) Observe that $KB \models C(a)$

for all $a \in E^+$, and $KB \models \neg C(a)$ for all $a \in E^-$. Try to simplify C while preserving this property, and then return it.

Observe that, when C_{i_j} is added into $\mathbb{C}$, we have that $a^{\mathcal{I}} \notin C_{i_j}^{\mathcal{I}}$ 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[3], and if it holds then we add C_{i_j} into the set $\mathbb{C}$. Otherwise, we add C_{i_j} into $\mathbb{C}_0$. To increase the chance to have C_{i_j} satisfying the mentioned condition and being added into $\mathbb{C}$, we tend to make C_{i_j} strong enough. For this reason, we do not use the technique with *LargestContainer* introduced in [8], and when necessary, we tighten the strategy for Step 3 by requiring reaching the partition corresponding to $\sim_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$ before stopping the granulation process.

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

Observe that we always have $KB \models \neg(\bigsqcup \mathbb{C})(a)$ for all $a \in E^-$. So, intending to return $\bigsqcup \mathbb{C}$ as the result, we try to extend $\mathbb{C}$ to satisfy $KB \models (\bigsqcup \mathbb{C})(a)$ for more and more $a \in E^+$. This is the skeleton of our method.

As a slight variant, one can exchange E^+ and E^-, apply the BBCL method to get a concept C', and then return $\neg C'$. We call this method *dual-BBCL*. Its search strategy is dual to the one of BBCL. One method may succeed when the other fails.

IV. Illustrative Examples

Example 4: Let $KB_0 = \langle \mathcal{R}, \mathcal{T}, \mathcal{A}_0 \rangle$ be the knowledge base given in Example 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 BBCL method on this example is as follows.

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

$$\begin{array}{rcl}
\Delta^{\mathcal{I}} & = & \{P_1, P_2, P_3, P_4, P_5, P_6\} \\
x^{\mathcal{I}} & = & x, \text{ for } x \in \{P_1, P_2, P_3, P_4, P_5, P_6\} \\
Pub^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
Awarded^{\mathcal{I}} & = & \{P_1, P_4, P_6\} \\
cites^{\mathcal{I}} & = & \{\langle P_1, P_2\rangle, \langle P_1, P_3\rangle, \langle P_1, P_4\rangle, \\
 & & \langle P_1, P_6\rangle, \langle P_2, P_3\rangle, \langle P_2, P_4\rangle, \\
 & & \langle P_2, P_5\rangle, \langle P_3, P_4\rangle, \langle P_3, P_5\rangle, \\
 & & \langle P_3, P_6\rangle, \langle P_4, P_5\rangle, \langle P_4, P_6\rangle\} \\
cited_by^{\mathcal{I}} & = & (cites^{\mathcal{I}})^{-1}
\end{array}$$

The function $Year^{\mathcal{I}}$ 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 $Y_4 = E^+$, $Y_3 \subset E^-$ and $Y_5 \subset E^-$. (It is not yet the partition corresponding to $\sim_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$.)
7) We have $C_4 = Awarded \sqcap \exists cited\_by.\top$. Since $KB \models \neg C_4(a)$ for all $a \in E^-$, we add C_4 to $\mathbb{C}$ and obtain $\mathbb{C} = \{C_4\}$ and $\bigsqcup \mathbb{C} = C_4$.
8) Since $KB \models (\bigsqcup \mathbb{C})(a)$ for all $a \in E^+$, and $\bigsqcup \mathbb{C} = C_4 = Awarded \sqcap \exists cited\_by.\top$ is already in the normal form and cannot be simplified, we return $Awarded \sqcap \exists cited\_by.\top$ as the result. ◁

Example 5: We now consider the dual-BBCL method. For that we take the same example as in Example 4 but exchange E^+ and E^-. Thus, we now have $E^+ = \{P_1,\ P_2,\ P_3,\ P_5\}$ and $E^- = \{P_4,\ P_6\}$. Execution of the BBCL method on this new example has the same first five steps as in Example 4, and then continues as follows.

1) The obtained partition $\{Y_3, Y_4, Y_5\}$ 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 $\sim_{\Sigma^\dagger,\Phi^\dagger,\mathcal{I}}$.)
2) We have $C_3 = \neg Awarded$. Since $KB \models \neg C_3(a)$ for all $a \in E^-$, we add C_3 to $\mathbb{C}$ and obtain $\mathbb{C} = \{C_3\}$.
3) We have $C_5 := Awarded \sqcap \neg\exists cited\_by.\top$. Since $KB \models \neg C_5(a)$ for all $a \in E^-$ and C_5 is not subsumed by $\bigsqcup \mathbb{C}$ w.r.t. KB, we add C_5 to $\mathbb{C}$ and obtain $\mathbb{C} = \{C_3, C_5\}$ and $\bigsqcup \mathbb{C} = \neg Awarded \sqcup (Awarded \sqcap \neg\exists cited\_by.\top)$.
4) Since $KB \models (\bigsqcup \mathbb{C})(a)$ for all $a \in E^+$, we normalize $\bigsqcup \mathbb{C}$ to $\neg Awarded \sqcup \neg\exists cited\_by.\top$ and return it as the result. If one wants to have a result for the dual learning problem as stated in Example 4, that concept should be negated to $Awarded \sqcap \exists cited\_by.\top$. ◁

Example 6: Let KB_0, E^+, E^-, KB and $\Phi^\dagger$ be as in Example 4, but let $\Sigma^\dagger = \{cited\_by,\ Year\}$. Execution of the BBCL method on this new example has the same first two steps as in Example 4, and then continues as follows.

1) Granulating $\{\Delta^{\mathcal{I}}\}$ as in [9, Example 11] we reach the following partition, which is consistent with E.
   - $partition = \{Y_4, Y_6, Y_7, Y_8, Y_9\}$
   - $Y_4 = \{P_4\}$, $Y_6 = \{P_1\}$, $Y_7 = \{P_2, P_3\}$, $Y_8 = \{P_6\}$, $Y_9 = \{P_5\}$

[3]e.g., [23], [24], [25], [26], [27], [28], [29], [22]

- $C_2 = (\mathit{Year} \geq 2008)$, $C_3 = (\mathit{Year} < 2008)$,
  $C_4 = C_3 \sqcap (\mathit{Year} \geq 2007)$,
  $C_5 = C_3 \sqcap (\mathit{Year} < 2007)$,
  $C_6 = C_2 \sqcap (\mathit{Year} \geq 2010)$,
  $C_8 = C_5 \sqcap \exists cited\_by.C_6$.

2) We have $Y_4 \subset E^+$. Since $KB \models \neg C_4(a)$ for all $a \in E^-$, we add C_4 to $\mathbb{C}$ and obtain $\mathbb{C} = \{C_4\}$.
3) We have $Y_8 \subset E^+$. Since $KB \models \neg C_8(a)$ for all $a \in E^-$ and C_8 is not subsumed by $\bigsqcup \mathbb{C}$ w.r.t. KB, we add C_8 to $\mathbb{C}$ and obtain $\mathbb{C} = \{C_4, C_8\}$ with $\bigsqcup \mathbb{C}$ equal to

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

4) Since $KB \models (\bigsqcup \mathbb{C})(a)$ for all $a \in E^+$, we normalize and simplify $\bigsqcup \mathbb{C}$ before returning it as the result. Without exploiting the fact that publication years are integers, $\bigsqcup \mathbb{C}$ can be normalized to

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

$C = (\mathit{Year} < 2008) \sqcap \exists cited\_by.(\mathit{Year} \geq 2010)$ is a simplified form of the above concept, which still satisfies that $KB \models C(a)$ for all $a \in E^+$ and $KB \models \neg C(a)$ for all $a \in E^-$. Thus, we return it as the result. $\lhd$

V. Conclusions

We have developed the *first bisimulation-based* method, called BBCL, for concept learning in DLs. 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 [9, Theorem 10]).

The idea of our method is to use models of the considered knowledge base and bisimulation in those models to guide the search for the concept. The skeleton of our search strategy has also a special design. It allows dual search (dual-BBCL). Our method is thus completely different from methods of the previous works [4], [5], [6], [7] with similar learning settings. As bisimulation is the notion for characterizing indiscernibility of objects in DLs, our method is natural and very promising.

As future work, we intend to implement our learning method. We will use efficient tableau decision procedures with global caching like [30], [23], [24], [25], [27], [31] for the task. Global caching is important because during the learning process many queries will be processed for the same knowledge base.

Acknowledgments

This work was partially supported by Polish National Science Centre (NCN) under Grants No. 2011/01/B/ST6/02759 and 2011/01/B/ST6/027569 as well as by 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".

References

[1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds., *Description Logic Handbook*. Cambridge University Press, 2002.

[2] W. Cohen and H. Hirsh, "Learning the Classic description logic: Theoretical and experimental results," in *Proceedings of KR'1994*, pp. 121–133.

[3] P. Lambrix and P. Larocchia, "Learning composite concepts," in *Proceedings of DL'1998*.

[4] L. Badea and S.-H. Nienhuys-Cheng, "A refinement operator for description logics," in *Proceedings of ILP'2000*, ser. LNCS, vol. 1866. Springer, 2000, pp. 40–59.

[5] L. Iannone, I. Palmisano, and N. Fanizzi, "An algorithm based on counterfactuals for concept learning in the Semantic Web," *Appl. Intell.*, vol. 26, no. 2, pp. 139–159, 2007.

[6] N. Fanizzi, C. d'Amato, and F. Esposito, "DL-FOIL concept learning in description logics," in *Proceedings of ILP'2008*, ser. LNCS, vol. 5194. Springer, 2008, pp. 107–121.

[7] J. Lehmann and P. Hitzler, "Concept learning in description logics using refinement operators," *Machine Learning*, vol. 78, no. 1-2, pp. 203–250, 2010.

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

[9] 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," Accepted for KSE'2012.

[10] J. Alvarez, "A formal framework for theory learning using description logics," in *ILP'2000 Work-in-progress reports*, 2000.

[11] J. Kietz, "Learnability of description logic programs," in *Proceedings of ILP'2002*, ser. LNCS, vol. 2583. Springer, 2002, pp. 117–132.

[12] S. Konstantopoulos and A. Charalambidis, "Formulating description logic learning as an inductive logic programming task," in *Proceedings of FUZZ-IEEE'2010*, pp. 1–7.

[13] F. Distel, "Learning description logic knowledge bases from data using methods from formal concept analysis," Ph.D. dissertation, Dresden University of Technology, 2011.

[14] J. Luna, K. Revoredo, and F. Cozman, "Learning probabilistic description logics: A framework and algorithms," in *Proceedings of MICAI'2011*, ser. LNCS, vol. 7094. Springer, 2011, pp. 28–39.

[15] S. Hellmann, "Comparison of concept learning algorithms (with emphasis on ontology engineering for the Semantic Web)," Master's thesis, Leipzig University, 2008.

[16] A.R. Divroodi and L.A. Nguyen, "On bisimulations for description logics," in *Proceedings of CS&P'2011*, pp. 99–110 (see also arXiv:1104.1964).

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

[18] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Inf. Sci.*, vol. 177, no. 1, pp. 3–27, 2007.

[19] S.T. Cao, L.A. Nguyen, and A. Szałas, "On the Web ontology rule language OWL 2 RL," in *Proceedings of ICCCI'2011*, ser. LNCS, vol. 6922. Springer, 2011, pp. 254–264.

[20] S.T. Cao, L.A. Nguyen, and A. Szalas, "WORL: a Web ontology rule language," in *Proceedings of KSE'2011*. IEEE Computer Society, 2011, pp. 32–39.

[21] M. Baldoni, L. Giordano, and A. Martelli, "A tableau for multimodal logics and some (un)decidability results," in *Proceedings of TABLEAUX'1998*, ser. LNCS, vol. 1397. Springer, 1998, pp. 44–59.

[22] I. Horrocks, O. Kutz, and U. Sattler, "The even more irresistible $\mathcal{SROIQ}$," in *Proc. of KR'2006*. AAAI Press, 2006, pp. 57–67.

[23] L.A. Nguyen and A. Szałas, "ExpTime tableaux for checking satisfiability of a knowledge base in the description logic ALC," in *Proceedings of ICCCI'2009*, ser. LNAI, vol. 5796. Springer, 2009, pp. 437–448.

[24] L.A. Nguyen, "Cut-free ExpTime tableaux for checking satisfiability of a knowledge base in the description logic ALCI," in *Proceedings of ISMIS'2011*, ser. LNCS, vol. 6804. Springer, 2011, pp. 465–475.

[25] L.A. Nguyen and A. Szałas, "Tableaux with global caching for checking satisfiability of a knowledge base in the description logic $\mathcal{SH}$," *T. Computational Collective Intelligence*, vol. 1, pp. 21–38, 2010.

[26] I. Horrocks and U. Sattler, "A description logic with transitive and inverse roles and role hierarchies," *J. Log. Comput.*, vol. 9, no. 3, pp. 385–410, 1999.

[27] L.A. Nguyen, "A cut-free ExpTime tableau decision procedure for the description logic SHI," in *Proceedings of ICCCI'2011 (1)*, ser. LNCS, vol. 6922. Springer, 2011, pp. 572–581.

[28] I. Horrocks, U. Sattler, and S. Tobies, "Reasoning with individuals for the description logic SHIQ," in *Proceedings of CADE-17*, ser. LNCS, vol. 1831. Springer, 2000, pp. 482–496.

[29] I. Horrocks and U. Sattler, "A tableau decision procedure for $\mathcal{SHOIQ}$," *J. Autom. Reasoning*, vol. 39, no. 3, pp. 249–276, 2007.

[30] L.A. Nguyen, "An efficient tableau prover using global caching for the description logic $\mathcal{ALC}$," *Fundamenta Informaticae*, vol. 93, no. 1-3, pp. 273–288, 2009.

[31] L.A. Nguyen, "A cut-free ExpTime tableau decision procedure for the logic extending converse-PDL with regular inclusion axioms," arXiv:1104.0405v1, 2011.