Journal of Information Hiding and Multimedia Signal Processing ©2021 ISSN 2073-4212
Ubiquitous International Volume **12**, Number **1**, March **2021**

Classification of Concepts Using Decision Trees for Inconsistent Knowledge Systems Based on Bisimulation

Nguyen Thi Hong Khanh

Faculty of Information Technology
Electricity Power University, Vietnam
khanhnth@epu.edu.vn

Received June 2020; revised October 2020

Abstract. *Knowledge inconsistencies can naturally arise in the application domains considered in Artificial Intelligence, for example as a result of data mining in distributed sources. To deal with inconsistent knowledge, several inconsistent descriptive logic have been proposed. In this paper, we face the problem of learning concepts for inconsistent basic knowledge systems on a bisectoral basis. Here, we present a learning conceptual system in inconsistent knowledge bases and discuss the preliminary experimental results obtained in the field of electronic applications.*
Keywords: Concept learning, Description logics, Bisimularity, Knowledge base.

1. Introduction.

Description logics(DLs) is a family of formal languages that is well suited for representation and reasoning in a domain of particular interest. DLs is of particular importance in providing theoretical models for semantic systems. It is the basic for building languages for modeling ontologies in which OWL is a language that is recommended by the W3C International Standard for use in Semantic Web systems [5, 9]. Description logics have usually been considered as syntactic variants of restricted versions of classical first-order logic [1]. On the other hand, in Semantic Web and multiagent applications, knowledge fusion frequently leads to inconsistencies. A way to deal with inconsistencies is to follow the area of paraconsistent reasoning [9, 12].

Concept learning in DLs is similar to binary classification in traditional machine learning. The differences are that in DLs objects are described not only by attributes but also by the relationship between the objects. As bisimulation is the notion for characterizing indiscernibility of objects in DLs. It is very useful for concept learning in this DLs. Consider a domain with individuals represented by single and binary predicates. In DL language, predicates such as concept name and role name, respectively. Domains can be described by different sources. For example, individuals may be some object in an area on earth and sources of information may be computer systems of different satellites objects described by some boolean attributes and binary relationships between them. Another example is the following: Some banks cooperate to share information about their customers to a certain extent; the bank's customers are individuals; atomic concepts can be credibility, wealth, and financial discipline; Atomic roles can be some relationship based on the transformation. Sources may provide consistent or inconsistent assertions. Based on them, the paraconsistent interpretation can be generated as an integrated information system.

Another situation of dealing with paraconsistent interpretations occurs when an inconsistent knowledge base is given. Such a knowledge base may result from combining or merging different knowledge bases. Usually, matching or merging ontologies is done semi-automatically by people using software tools. However, when the considered knowledge bases are dynamic, the automatic mode is unavoidable. In general, there are situations in which the considered knowledge base is inconsistent and one still wants to use it to derive only meaningful consequences. Having a knowledge base, sometimes we want to generate its models. For examples, for concept learning in DL from a knowledge base KB given as a training information system, generated models of KB can be used to guide the search process C concept [11]. When KB is inconsistent w.r.t the traditional semantics, it has only paraconistent models.

Now, consider the concept learning problem set as follows. We are given a training information system $\mathcal{I}$ with inconsistent data, which is a $\mathfrak{s}$- interpretation, together with three subsets E^+, E^- and E^p of Δ^I, which consist of positive examples, negative examples and inconsistent examples of a concept C. We may be told that C is in a restricted language and the learning concept is allowed to have some small error rate.

Concept learning in description logics has been studied by many researchers and is divided into three main approaches. The first approach focuses on the ability to learn in description logics and builds some simple algorithms [4, 7]. The second approach studies the concept learning in the description logic using refinement operators [6, 8, 2]. The third approach exploits bisimulation for concept learning problems in description logics. In [7] Lambrix and Larochia proposed a simple concept learning algorithm based on the concept. Lehmann and Hitzler [8], Badea and Nienhys [2], Iannone et al [6] studied concept learning in DLs by using refinement operators as in inductive logic programming. Apart from refinement operators, scoring functions, and search strategies also play important roles in algorithms proposed in those works.
All Last works handle on the consistent knowledge base. In this paper, we develop the bisimulation based method for inconsistent knowledge system, for concept learning in paraconsistent DLs. Concept learning problem C such that:

- **(1):** $KB \models C(a)$ for all $a \in E^+$ and $a \notin E^-$,
- **(2):** $KB \models C(a)$ for all $a \in E^-$ and $a \notin E^+$,
- **(3):** $KB \models C(a)$ for all $a \in E^+$ and $a \in E^-$.

Where KB is a knowledge base in the considered DLs, and E^+, E^- are given sets of examples of C. As bisimulation is the notion for characterizing indiscernibility of objects in paraconsistent DLs, our method is natural and very promising.
Our method is completely different from the ones of [4, 6, 8, 2], as it is based on bisimulation for paraconsistent DLs, while all the later ones are learning on consistent knowledge systems.

The rest of this paper is structured as follows. In Section 2, we present notation and semantics of the paraconsistent DLs considered in this paper. In this Section 3, we recall bisimilarity for paraconsistent description logics. In Section 4, we present a learning algorithm based on bisimulation and in Section 5 we evaluate this algorithm by means of our implementation. Finally, in Section 6 we summarize our work and draw conclusions.

2. **Preliminaries.**

2.1. **Notation and semantic of Description logics.** In this work, we consider a finite set C of concept names, a countable set I of individual names a countable set R of role

names. We use letters like A, B to denote concept names, letters like r, s to denote role names, and letters like a, b to denote individual names.
A *$DL - signature$* is set $\Sigma = \Sigma_I \cup \Sigma_C \cup \Sigma_R$, where:

Σ_I is a finite set of *individual names*, Σ_C is a finite set of *concept names*, Σ_C is a finite set of *role names*.

Let Φ be a set of features among I (inverse roles), O (nominal), Q (qualified number restrictions), U (the universal role) and `Self` (local reflexivity of a role). In this section, we recall notations and semantics of the DLs ALC_Φ. A set of DL-features is a set consisting of some or zero of these names.
Given a DL-signature Σ, a set Φ of DL-features, $\mathcal{L} = ALC$, *roles* and *concepts* of the language $\mathcal{L}$ are defined in [11].

2.2. **Paraconsistent semantics for Description logics.** Following the recommendation of W3C for OWL, like [10, 3] we use the traditional syntax of DLs and only change its semantics to cover paraconsistency.

Recall that, using the traditional semantics, every query is a logical consequence of an inconsistent knowledge base. A knowledge base may be inconsistent, for example, when it contains both individual assertions $A(a)$ and $\neg A(a)$ for some $A \in \mathbf{C}$ and $a \in \mathbf{I}$. Paraconsistent reasoning is inconsistency-tolerant and aims to derive meaningful logical consequences even when the knowledge base is inconsistent.This problem just handles three-valued logic (t: true, f: false and i: inconsistent). We identify $\mathfrak{s}$ is paraconsistent semantics, $\mathfrak{s}$ with the tuple $\mathfrak{s}_{\mathtt{C}}, \mathfrak{s}_{\mathtt{R}}, \mathfrak{s}_{\forall\exists\mathtt{Q}}, \mathfrak{s}_{\mathtt{GCI}}$. The set of considered paraconsistent semantics is thus

$$\mathfrak{S} = \{2, 3\} \times \{2, 3\} \times \{+, \pm\} \times \{\mathsf{w}, \mathsf{m}, \mathsf{s}\}.$$

For $\mathfrak{s} \in \mathfrak{S}$, an $\mathfrak{s}$-*interpretation* 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*, $\cdot^\mathcal{I}$ is the *interpretation function*, which maps every individual name a to an element $a^\mathcal{I} \in \Delta^\mathcal{I}$, every concept name A to a pair $A^\mathcal{I} = \langle A_+^\mathcal{I}, A_-^\mathcal{I} \rangle$ of subsets of $\Delta^\mathcal{I}$, and every role name r to a pair $r^\mathcal{I} = \langle r_+^\mathcal{I}, r_-^\mathcal{I} \rangle$ of binary relations on $\Delta^\mathcal{I}$ such that:

- if $\mathfrak{s}_{\mathtt{C}} = 2$ then $A_+^\mathcal{I} = \Delta^\mathcal{I} \setminus A_-^\mathcal{I}$
- if $\mathfrak{s}_{\mathtt{C}} = 3$ then $A_+^\mathcal{I} \cup A_-^\mathcal{I} = \Delta^\mathcal{I}$
- if $\mathfrak{s}_{\mathtt{R}} = 2$ then $r_+^\mathcal{I} = (\Delta^\mathcal{I} \times \Delta^\mathcal{I}) \setminus r_-^\mathcal{I}$
- if $\mathfrak{s}_{\mathtt{R}} = 3$ then $r_+^\mathcal{I} \cup r_-^\mathcal{I} = \Delta^\mathcal{I} \times \Delta^\mathcal{I}$.

The intuition behind $A^\mathcal{I} = \langle A_+^\mathcal{I}, A_-^\mathcal{I} \rangle$ is that $A_+^\mathcal{I}$ gathers positive evidence about A, while $A_-^\mathcal{I}$ gathers negative evidence about A. Thus, $A^\mathcal{I}$ can be treated as the function from $\Delta^\mathcal{I}$ to $\{t, f, i\}$ defined below:

$$A^\mathcal{I}(x) = \begin{cases} t & \text{for } x \in A_+^\mathcal{I} \text{ and } x \notin A_-^\mathcal{I} \\ f & \text{for } x \in A_-^\mathcal{I} \text{ and } x \notin A_+^\mathcal{I} \\ i & \text{for } x \in A_+^\mathcal{I} \text{ and } x \in A_-^\mathcal{I} \end{cases} \tag{1}$$

Informally, $A^\mathcal{I}(x)$ can be thought of as the truth value of $x \in A^\mathcal{I}$. Note that $A^\mathcal{I}(x) \in \{t, f\}$ if $\mathfrak{s}_{\mathtt{C}} = 2$, and $A^\mathcal{I}(x) \in \{t, f, i\}$ if $\mathfrak{s}_{\mathtt{C}} = 3$. The intuition behind $r^\mathcal{I} = \langle r_+^\mathcal{I}, r_-^\mathcal{I} \rangle$ is similar, and under which $r^\mathcal{I}(x, y) \in \{t, f\}$ if $\mathfrak{s}_{\mathtt{R}} = 2$, and $r^\mathcal{I}(x, y) \in \{t, f, i\}$ if $\mathfrak{s}_{\mathtt{R}} = 3$.

The interpretation function $\cdot^\mathcal{I}$ maps a role R to a pair $R^\mathcal{I} = R_+^\mathcal{I}, R_-^\mathcal{I}$, defined for the case $R \notin \mathbf{R}$ as follows:

$$\begin{aligned} (r^-)^\mathcal{I} &= (r_+^\mathcal{I})^{-1}, (r_-^\mathcal{I})^{-1} \\ U^\mathcal{I} &= \Delta^\mathcal{I} \times \Delta^\mathcal{I}, \emptyset. \end{aligned}$$

Function $\cdot^{\mathcal{I}}$ maps a complex concept C to a pair $C^{\mathcal{I}} = \langle C_+^{\mathcal{I}}, C_-^{\mathcal{I}} \rangle$ of subsets of $\Delta^{\mathcal{I}}$ defined as follows:

$$
\begin{array}{rcl}
\top^{\mathcal{I}} & = & \langle \Delta^{\mathcal{I}}, \emptyset \rangle \\
\bot^{\mathcal{I}} & = & \langle \emptyset, \Delta^{\mathcal{I}} \rangle \\
(\{a\})^{\mathcal{I}} & = & \{a^{\mathcal{I}}\}, \Delta^{\mathcal{I}} \setminus \{a^{\mathcal{I}}\} \\
(\neg C)^{\mathcal{I}} & = & \langle C_-^{\mathcal{I}}, C_+^{\mathcal{I}} \rangle \\
(C \sqcap D)^{\mathcal{I}} & = & \langle C_+^{\mathcal{I}} \cap D_+^{\mathcal{I}}, C_-^{\mathcal{I}} \cup D_-^{\mathcal{I}} \rangle \\
(C \sqcup D)^{\mathcal{I}} & = & \langle C_+^{\mathcal{I}} \cup D_+^{\mathcal{I}}, C_-^{\mathcal{I}} \cap D_-^{\mathcal{I}} \rangle \\
(\exists R.\mathtt{Self})^{\mathcal{I}} & = & \big\langle \{x \in \Delta^{\mathcal{I}} \mid (x,x) \in R_+^{\mathcal{I}}\}, \\
 & & \ \{x \in \Delta^{\mathcal{I}} \mid (x,x) \in R_-^{\mathcal{I}}\} \big\rangle;
\end{array}
$$

if $\mathfrak{s}_{\forall\exists\mathtt{Q}} = +$ then

$$
\begin{array}{l}
(\exists R.C)^{\mathcal{I}} = \\
\quad \big\langle \{x \in \Delta^{\mathcal{I}} \mid \exists y((x,y) \in R_+^{\mathcal{I}} \wedge y \in C_+^{\mathcal{I}})\}, \\
\quad \ \{x \in \Delta^{\mathcal{I}} \mid \forall y((x,y) \in R_+^{\mathcal{I}} \to y \in C_-^{\mathcal{I}})\} \big\rangle \\
(\forall R.C)^{\mathcal{I}} = \\
\quad \big\langle \{x \in \Delta^{\mathcal{I}} \mid \forall y((x,y) \in R_+^{\mathcal{I}} \to y \in C_+^{\mathcal{I}})\}, \\
\quad \ \{x \in \Delta^{\mathcal{I}} \mid \exists y((x,y) \in R_+^{\mathcal{I}} \wedge y \in C_-^{\mathcal{I}})\} \big\rangle;
\end{array}
$$

$$
\begin{array}{l}
(\geq n\, R.C)^{\mathcal{I}} = \\
\big\langle \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid (x,y) \in R_+^{\mathcal{I}} \wedge y \in C_+^{\mathcal{I}}\} \geq n\}, \\
\ \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid (x,y) \in R_+^{\mathcal{I}} \wedge y \notin C_-^{\mathcal{I}}\} < n\} \big\rangle \\
(\leq n\, R.C)^{\mathcal{I}} = \\
\big\langle \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid (x,y) \in R_+^{\mathcal{I}} \wedge y \notin C_-^{\mathcal{I}}\} \leq n\}, \\
\ \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid (x,y) \in R_+^{\mathcal{I}} \wedge y \in C_+^{\mathcal{I}}\} > n\} \big\rangle;
\end{array}
$$

if $\mathfrak{s}_{\forall\exists\mathtt{Q}} = \pm$ then

$$
\begin{array}{l}
(\exists R.C)^{\mathcal{I}} = \\
\quad \big\langle \{x \in \Delta^{\mathcal{I}} \mid \exists y((x,y) \in R_+^{\mathcal{I}} \wedge y \in C_+^{\mathcal{I}})\}, \\
\quad \ \{x \in \Delta^{\mathcal{I}} \mid \forall y((x,y) \notin R_-^{\mathcal{I}} \to y \in C_-^{\mathcal{I}})\} \big\rangle \\
(\forall R.C)^{\mathcal{I}} = \\
\quad \big\langle \{x \in \Delta^{\mathcal{I}} \mid \forall y((x,y) \notin R_-^{\mathcal{I}} \to y \in C_+^{\mathcal{I}})\}, \\
\quad \ \{x \in \Delta^{\mathcal{I}} \mid \exists y((x,y) \in R_+^{\mathcal{I}} \wedge y \in C_-^{\mathcal{I}})\} \big\rangle;
\end{array}
$$

$$
\begin{array}{l}
(\geq n\, R.C)^{\mathcal{I}} = \\
\big\langle \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid (x,y) \in R_+^{\mathcal{I}} \wedge y \in C_+^{\mathcal{I}}\} \geq n\}, \\
\ \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid (x,y) \notin R_-^{\mathcal{I}} \wedge y \notin C_-^{\mathcal{I}}\} < n\} \big\rangle \\
(\leq n\, R.C)^{\mathcal{I}} = \\
\big\langle \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid (x,y) \notin R_-^{\mathcal{I}} \wedge y \notin nC_-^{\mathcal{I}}\} \leq n\}, \\
\ \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid (x,y) \in R_+^{\mathcal{I}} \wedge y \in C_+^{\mathcal{I}}\} > n\} \big\rangle.
\end{array}
$$

We denote Γ is a set of concepts, $\Gamma_+^{\mathcal{I}} = \bigcap\{C_+^{\mathcal{I}} \mid C \in \Gamma\}$, $\Gamma_-^{\mathcal{I}} = \bigcup\{C_-^{\mathcal{I}} \mid C \in \Gamma\}$ and $\Gamma^{\mathcal{I}} = \Gamma_+^{\mathcal{I}}, \Gamma_-^{\mathcal{I}}$. Observe that, if Γ is finite, then $\Gamma^{\mathcal{I}} = (\Gamma)^{\mathcal{I}}$.

Example 2.1. *An example of inconsistent knowlege base in $\mathcal{L}$ refers to electronic devices:*

Let Σ_I={*Cellphone, Bluetooth, Laptop, Memory, Size, Weight*}
Σ_C={*Device, General*}
Σ_R={*hasGeneral*}

$\mathcal{A}$={*General(Memory), General(Size), General(Weight), General(Bluetooth), Device(Cellphone), Device(Laptop), Device(Memory), hasGeneral(Cellphone,Size), hasGeneral(Cellphone,Memory), hasGeneral(Cellphone,Bluetooth), hasGeneral(Laptop,Size), hasGeneral(Cellphone,Weight), hasGeneral(Cellphone,Memory)* }

$\mathcal{T}=\{Device = \exists hasGeneral.\top\}$

This knowledge base is inconsistent because both concepts Device and General contain the object Memory.

An interpretation of the inconsistent knowledge base following:

$\Delta^{\mathcal{I}} = \{a, b, c, d, e, f\}$, $Cellphone^{\mathcal{I}} = a$, $Bluetooth^{\mathcal{I}} = b$, $Laptop^{\mathcal{I}} = c$, $Memory^{\mathcal{I}} = d$, $Size^{\mathcal{I}} = e$, $Weight^{\mathcal{I}} = f$.

$Device^{\mathcal{I}} = \{a, c, d\}$, $General^{\mathcal{I}} = \{b, d, e, f\}$.

3. **Bisimulation for Concept Learning in Paraconsistent Description Logics.**
Bisimulation is as a binary relation between nodes of a labeled in a graph. We will demonstrate how to modify and extend bisimulation to deal with richer logic languages. The approach is as follows: Fix a logic language, for example, a description logic, and define bisimulation so the Hennessy-Milner property still holds.

Bisimulation is of interest to researchers and it is applied in practice in which three main applications are mentioned: (i) Separating the expressive powers of logic languages; (ii) Minimizing interpretations and labeled state transition systems; (iii) Concept learning in description logics.

In this section, we consider an implementation of bisimilarity for concept learning in description logics when inconsistencies occur. Bisimilation applied for Description Logics in concept learning problems with consistent knowledge [11]. We repeat the definition from the idea is to use models of KB and bisimularity in this model to guide the search for C concept.

Let $\Phi \subseteq \{I, O, Q, U, \mathtt{Self}\}$ be a set of features, $\mathfrak{s} \in \mathfrak{S}$ a paraconsistent semantics, and $\mathcal{I}$, $\mathcal{I}'$ $\mathfrak{s}$-interpretations. A non-empty binary relation $Z \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}'}$ is called a $(\Phi, \mathfrak{s})$-*bisimulation* between $\mathcal{I}$ and $\mathcal{I}'$ if the following conditions hold for every $a \in \mathbf{I}$, $x, y \in \Delta^{\mathcal{I}}$, $x', y' \in \Delta^{\mathcal{I}'}$, $A \in \mathbf{C}$, $r \in \mathbf{R}$ and every role R of ALC_Φ different from U:

- **(1):** $Z(a^{\mathcal{I}}, a^{\mathcal{I}'})$
- **(2):** $Z(x, x') \Rightarrow [A_+^{\mathcal{I}}(x) \Rightarrow A_+^{\mathcal{I}'}(x')]$
- **(3):** $Z(x, x') \Rightarrow [A_-^{\mathcal{I}}(x) \Rightarrow A_-^{\mathcal{I}'}(x')]$
- **(4):** $[Z(x, x') \wedge R_+^{\mathcal{I}}(x, y)] \Rightarrow \exists y' \in \Delta^{\mathcal{I}'}[Z(y, y') \wedge R_+^{\mathcal{I}'}(x', y')]$,
- **(5):** if $\mathfrak{s}_{\forall\exists\mathtt{Q}} = +$ then
  $[Z(x, x') \wedge R_+^{\mathcal{I}'}(x', y')] \Rightarrow \exists y \in \Delta^{\mathcal{I}}[Z(y, y') \wedge R_+^{\mathcal{I}}(x, y)]$
- **(6):** if $\mathfrak{s}_{\forall\exists\mathtt{Q}} = \pm$ then
  $[Z(x, x') \wedge \neg R_-^{\mathcal{I}'}(x', y')] \Rightarrow \exists y \in \Delta^{\mathcal{I}}[Z(y, y') \wedge \neg R_-^{\mathcal{I}}(x, y)]$,
- **(7):** if $O \in \Phi$ then
  $Z(x, x') \Rightarrow (x = a^{\mathcal{I}} \Leftrightarrow x' = a^{\mathcal{I}'})$,
- **(8):** if $Q \in \Phi$ then
  if $Z(x, x')$ holds and $y_1, \ldots, y_n$ $(n \geq 1)$ are pairwise different elements of $\Delta^{\mathcal{I}}$ such that $R_+^{\mathcal{I}}(x, y_i)$ holds for every $1 \leq i \leq n$, then there exist pairwise different elements $y'_1, \ldots, y'_n$ of $\Delta^{\mathcal{I}'}$ such that $R_+^{\mathcal{I}'}(x', y'_i)$ and $Z(y_i, y'_i)$ hold for every $1 \leq i \leq n$,
- **(9):** if $Q \in \Phi$ and $\mathfrak{s}_{\forall\exists\mathtt{Q}} = +$ then
  if $Z(x, x')$ holds and $y'_1, \ldots, y'_n$ $(n \geq 1)$ are pairwise different elements of $\Delta^{\mathcal{I}'}$ such that $R_+^{\mathcal{I}'}(x', y'_i)$ holds for every $1 \leq i \leq n$, then there exist pairwise different elements $y_1, \ldots, y_n$ of $\Delta^{\mathcal{I}}$ such that $R_+^{\mathcal{I}}(x, y_i)$ and $Z(y_i, y'_i)$ hold for every $1 \leq i \leq n$,

(10): if $Q \in \Phi$ and $\mathfrak{s}_{\forall\exists Q} = \pm$ then if $Z(x, x')$ holds and $y'_1, \ldots, y'_n$ $(n \geq 1)$ are pairwise different elements of $\Delta^{\mathcal{I}'}$ such that $\neg R_-^{\mathcal{I}'}(x', y'_i)$ holds for every $1 \leq i \leq n$, then there exist pairwise different elements $y_1, \ldots, y_n$ of $\Delta^{\mathcal{I}}$ such that $\neg R_-^{\mathcal{I}}(x, y_i)$ and $Z(y_i, y'_i)$ hold for every $1 \leq i \leq n$,

if $U \in \Phi$ then

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

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

if $\texttt{Self} \in \Phi$ then

(13): $Z(x, x') \Rightarrow [r_+^{\mathcal{I}}(x, x) \Rightarrow r_+^{\mathcal{I}'}(x', x')]$

(14): $Z(x, x') \Rightarrow [r_-^{\mathcal{I}}(x, x) \Rightarrow r_-^{\mathcal{I}'}(x', x')]$

As a consequence, if one of the above conditions holds and $\mathcal{I}$, $\mathcal{I}'$ are $\mathfrak{s}$-interpretations $(\Phi, \mathfrak{s})$-bisimilar to each other, then $\mathcal{I} \models_{\mathfrak{s}} \mathcal{A}$ iff $\mathcal{I}' \models_{\mathfrak{s}} \mathcal{A}$.

4. **Concept Learning for Paraconsistent Description Logics.** Concept learning problem is similar to binary classification in traditional machine learning. The difference is that in paraconsistent description logics objects are described not only by attributes but also by binary relationships between objects. As bisimulation is the notion for characterizing indiscernibility of objects in paraconsistent description logics. It is very useful for concept learning in inconsistent knowledge base systems.

Definition 4.1. *(Learning problem in paraconsistent description logics). Let $\mathcal{I}$ be a finite interpretation (given as a training information system), a knowledge base KB in a DL L and sets E^+, E^- of individuals, learn a concept C in L such that:*

(1): $KB \models C(a)$ for all $a \in E^+$ and $a \notin E^-$,
(2): $KB \models C(a)$ for all $a \in E^-$ and $a \notin E^+$,
(3): $KB \models C(a)$ for all $a \in E^+$ and $a \in E^-$.

The goal of learning is to find a correct concept with respect to the examples. This can be seen as a search process in the space of concepts. A natural idea is imposing an ordering on this search space and use models of KB and bisimulation in those models to guide the search for C.

The main idea of this method is to smooth the Δ domain of the information system $\mathcal{I}$ using the selectors. Based on that idea, the concept learning approach is broadly described as follows:

Let $A_p \in \Sigma_C$ be a concept name standing for the *decisionattribute* and suppose that A_p can be expressed by a concept C in $\mathcal{L}_{\Sigma^+,\Phi^+}$, where $\Sigma^+ \subseteq \Sigma \backslash A_p$ and $\Phi^+ \subseteq \Phi$. Let $\mathcal{I}$ be a training information system over Σ. How can we learn that concept C on the basis of I.

The main idea of this method is to smooth the Δ domain of the information system $\mathcal{I}$ using the selectors. Based on that idea, the concept learning approach is broadly described as follows:

1. Starting from $\Delta^{\mathcal{I}}$ partition, we smooth this partition sequentially until we reach the partition corresponding to A_p. This smoothing process can be stopped sooner when the current partition is consistent with E or satisfies certain conditions.

2. In the process of smoothing $\Delta^{\mathcal{I}}$ partition, the blocks created at all steps are $Y_1, Y_2, ..., Y_n$. Each generated block is denoted by a new index by increasing the value of n. For each $1 \leq i \leq n$, we set the following information:
   - Y_i is characterized by a concept C_i such that $C_i^{\mathcal{I}} = Y_i$,
   - Record information about Y_i is split by E,
   - Saving the index of the largest block Y_j such that $Y_i \subseteq Y_j$ and Y_j is not split by E.
3. The current partition is denoted $Y = \{Y_{i_1}, Y_{i_2}, ..., Y_{i_k}\} \subseteq \{Y_1, Y_2, ..., Y_n\}$
4. When the current partition becomes consistent with A_p, return $C_{i_1} \sqcup ... \sqcup C_{i_1}$, where $i_1...i_j$ are indices such that $Y_{i_1}...Y_{i_j}$ are all the blocks of the current partition that are subsets of A_p.

Example 4.1. *Consider knowledge base has been given in Example 1. Suppose we want to learn a concept C such that $C^{\mathcal{I}} = \{a, c\}$*

1. $Y_1 := \Delta^{\mathcal{I}}$, $C_1 = \top$, *partition* $:= \{Y_1\}$,
2. *splitting Y_1 by Device*
   - $Y_2 := \{a, c, d\}, C_2 := Device$,
   - $Y_3 := \{b, d, e, f\}, C_3 := \neg Device$,
   - *partition* $:= \{Y_2, Y_3\}$,
3. *splitting Y_2 by $\exists hasGeneral.\top$:*
   - $Y_4 := \{a, c\}, C_4 := C_2 \sqcap \exists hasGeneral.\top$,
   - $Y_5 := \{d\}, C_5 := C_2 \sqcap \neg\exists hasGeneral.\top$,
   - *partition* $:= \{Y_3, Y_4, Y_5\}$,
4. *The partition is consistent with $\{a, c\}$*
   The returned concept is $C = C_4 = Device \sqcap \exists hasGeneral.\top$.

5. Preliminary Evaluation.

5.1. **The datasets.** We applied the proposed model on the set of electric devices. We buid concepts data, roles data and individuals data. We labelled and testing datasets with different numbers of inconsistent concepts. The device data set contains electronic device information and attributes including information on 941 types of configurations (concepts), 32 links between objects (roles), and 521 objects (individual). Each object in the data set is expressed by the concept. We use data from 7 out of 627 subjects for training and validate. Data of the other types of equipment is used for test.

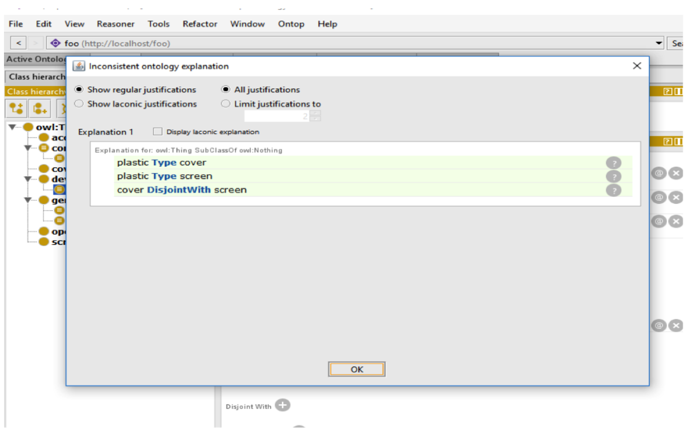

FIGURE 1. Test the inconsistent data set on Protege

After some preprocessing steps on these datasets, i.e. tested the inconsistent data set on Protege. Protege error when testing Reasoner with HermiT.

The reason it's inconsistent data here is that of the plastic cover that the plastic is also on the screen. Meanwhile, the cover and the screen are disjointed (the two layers are totally unrelated, unrelated) to each other.

5.2. **Experimental results.** We took several experiments with different of rate of inconsistent data to evaluate the effect of the proposed algorithm. In order to analyse the contribution of the labelled datasets, we also generated some subsets with the size of 25, 30, 35, 40, 45, 50, 55, 60 inconsistent data rates.
To illustrate the influence of the inconsistent parameter, we additionally measured tenfold cross-validation accuracy, Recall, Precision, and F1-Measure. The results are shown in Table. Since the inconsistent parameter acts as a termination criterion, we observe, as expected that lower inconsistent values lead to significant increases in accuracy.

Table 1. The influence of the inconsistent parameters in knowledge system

Inconsistent(%)	Accuracy(%)	Precision(%)	Recall(%)	F1-Measure(%)
25	80.00	66.67	100	80.00
30	78.43	63.54	100	75.00
35	75.62	60.00	100	70.00
40	72.14	56.00	100	66.00
45	71.00	52.48	100	63.67
50	70.48	48.23	100	62.33
55	70.32	44.00	100	59.00
60	70.00	40.00	100	57.14

Overall, the presented approach is able to learn accurate and inconsistent concepts with a reasonably low number of expensive reasoner requests. Note that all the approaches are able to learn in a very expressive language with arbitrarily nested structures, as can be seen in the concept above. Learning many levels of the structure has recently been identified as a key issue for structured Machine Learning and our work provides a clear advance on this front.

The evaluations show that our approach is competitive with state-of-the-art Web Semantic systems when the approximate reasoning technique is used.

6. **Conclusion.** In this paper, a concept learning model for paraconsistent knowledge base system is introduced discussed. The key idea in this work is to use models of KB and bisimulation in those models to guide the search for C. This mathematical technique, along with the partitioning strategies used, has been tested on two theoretical and experimental aspects.

REFERENCES

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

[2] L. Badea and S. Nienhuys-Cheng. Refining concepts in description logics, *In Proceedings of the 2000 International Workshop on Description Logics (DL2000)*,Aachen, Germany, August 17-19, 2000, pages 31-44, 2000.

[3] N. Belnap. A useful four-valued logic. In G. Eptein and J. Dunn, editors, *Modern Uses of Many Valued Logic.*, pages 8-37. Reidel, 1977.

[4] N. Fanizzi, C. d'Amato, and F. Esposito. DL-FOIL concept learning in description logics, *Inductive Logic Programming, 18th International Conference, ILP 2008*,Prague, Czech Republic, September 10-12, 2008, Proceedings, pages 107-121, 2008.

[5] I. Horrocks, P. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language, *Journal of Web Semantics*, 1(1):7-26, 2003.

[6] 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

[7] J. Wang, C. Jiang, W. Hu, M. Gao, Modified design of photonic crystal fibers with flattened dispersion, *Foundations of Intelligent Systems, 9th International Symposium, ISMIS '96, Zakopane, Poland.*, June 9-13, 1996, Proceedings, pages 68-77, 1996.

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

[9] Y. Ma and P. Hitzler. Paraconsistent reasoning for OWL 2. In A. Polleres and T. Swift, editors, *Proc. of Web Reasoning and Rule Systems.*,volume 5837 of LNCS, pages 197-211.Springer, 2009.

[10] J. Maluszy´nski, A. Sza las, and A. Vit´oria. Paraconsistent logic programs with fourvalued rough sets. In C.-C. Chan, J. Grzymala-Busse, and W. Ziarko, editors, *Proc. of RSCTC'2008.*,volume 5306 of LNAI, pages 41-51, 2008.

[11] T. H. K. Nguyen, Q. Ha, and T. H. Tran. A learning method based on bisimulation in inconsistent knowledge systems, *15th International Conference on Control, Automation, Robotics and Vision, ICARCV 2018, Singapore, November 18-21, 2018.*,pages 430-435. IEEE, 2018.

[12] M. K. Sarker and P. Hitzler. Efficient concept induction for description logics, *AAAI Press*, pages 3036-3043, 2019.