Journal of Science - Hue University
ISSN 1859-1388
Vol. 96; No. 8, 2014, PP. 87-101

ENTROPY-BASED MEASURES FOR PARTITIONING THE DOMAIN OF AN INTERPRETATION IN DESCRIPTION LOGICS

Thanh-Luong Tran*, Thi-Lan-Giao Hoang

Department of Information Technology, College of Sciences, Hue University

Abstract. A basic feature of description logic-based information systems is binary relations between objects. Concept learning in these systems need to take advantages of those relationships for granulating partitions of the domain of an interpretation (i.e., a description logic-based information system). In this paper, we study a method for partitioning the domain of an interpretation in description logics using bisimulation. Apart from information gain measure, we propose a new measure, called *first depth information gain*, to choose selectors for dividing blocks in partitions. Our results show that this measure is valuable and it is an important foundation for bisimulation-based concept learning methods in description logics.

Keyworks: description logics, concept learning, bisimulation, information gain.

1 Introduction

Semantic Technologies are being applied in many areas such as Bioinformatics, Semantic Web Browser, Knowledge Managements, Software Engineering, etc. Ontologies are becoming one of the most important respects for knowledge representation and reasoning. According to the recommendation of W3C, ontologies should be modeled by using the Web Ontology Languages (OWLs) which are languages based on description logics (DLs) [1, 2]. Despite advances in researches and applications, however, we are facing with the problem lacking of available large well-known ontologies as knowledge bases. Therefore, it is natural to try an adaptation of machine learning approaches for the accomplishment of OWL ontologies. It is required to investigate and develop methods for learning in DLs, called concept learning.

Concept learning in DLs is similar to binary classification in traditional machine learning. However, the problem in the context of DLs differs from the traditional setting in that objects are described not only by attributes but also by binary relations between objects.Concept learning in DLs has been received much attention of the researchers. Apart from approaches based on "least common subsumers" proposed by Cohen and Hrish [3], and concept normal-

**Corresponding author: ttluong@hueuni.edu.vn*

Submitted: September 16, 2014; Revised: December 10, 2014; Accepted: December 15, 2014.

ization proposed by Lambrix and Larocchia [4], the approach based on inductive logic programming, namely refinement operators, was also studied by Badea and Nienhuy-Cheng [5], Iannone et al. [6], Fanizzi at al. [7], and Lehmann et al. [8]. In the recent works, Nguyen and Szałas [9], Ha et al. [10], and Tran et al. [11, 12] used bisimulation for concept learning in DLs.

There are three main settings for concept learning in DLs [10, 13]. One of them is described as follows: given a finite interpretation $\mathcal{I}$ in a DL L, learn a concept C in L such that:

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

where E^+ (resp. E^-) contains positive (resp. negative) examples of C. Note that, $\mathcal{I} \models \neg C(a)$ is the same as $\mathcal{I} \not\models C(a)$.

Bisimulation in DLs is applied to model indiscernibility of objects and blocks in partition are divided by using basic selectors [9, 11]. However, an important issue is: Which block from the current partition should be divided first? Which selector should be used to divide it? This affects both the "quality" of the final partition and complexity of the process. Nguyen and Szałas [9], Tran et al. [11] used information gain to guide in the partition process. This brings good results in favorable cases, but it is not strong enough for complex cases. To obtain a final partition, the main loop of the granulation process may need to repeat many times.

In this paper, apart from information gain, we introduce a new measure which is based on entropy, called *first depth information gain*, for partitioning the domain of an interpretation. We also give examples to illustrate the effectiveness of measures.

The rest of this paper is structured as follows. In Section 2 we recall the notation of DLs and semantics. Section 3 outlines bisimulation and indiscernibility in DLs. Section 4 presents an algorithm of concept learning. Entropy-based measures are proposed in Section 5. We conclude this work in Section 6.

2 Description Logics and Semantics

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*. 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 $A \in \Sigma_A$, $range(A)$ is a non-empty set that is countable if A is discrete, and partially ordered by $\leq$ otherwise. (For simplicity we do not subscript $\leq$ by A.) A discrete attribute is called a *Boolean attribute* if $range(A) = \{\mathsf{true}, \mathsf{false}\}$. 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 will 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 $range(A)$ or $range(\sigma)$ by letters like c and d.

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

Definition 2.1 (The $\mathcal{L}_{\Sigma,\Phi}$ Language)**.** Let Σ be a DL-signature, Φ be a set of DL-features and $\mathcal{L}$ stands for $\mathcal{ALC}$. 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 $d \in range(A)$ then $A = d$ and $A \neq d$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$,
- if $A \in \Sigma_{nA}$ and $d \in range(A)$ then $A \leq d$, $A < d$, $A \geq d$ and $A > d$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$,
- if R is an object role of $\mathcal{L}_{\Sigma,\Phi}$, C and D are concepts 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 $\mathcal{I} \in \Phi$ then R^- is an object role of $\mathcal{L}_{\Sigma,\Phi}$,
  - if $\mathcal{O} \in \Phi$ then $\{a\}$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$,
  - if $\mathcal{F} \in \Phi$ then $\leq 1\, r$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$,
  - if $\{\mathcal{F}, \mathcal{I}\} \subseteq \Phi$ then $\leq 1\, r^-$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$,
  - if $\mathcal{N} \in \Phi$ then $\geq n r$ and $\leq n r$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$,
  - if $\{\mathcal{N}, \mathcal{I}\} \subseteq \Phi$ then $\geq n\, r^-$ and $\leq n\, r^-$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$,
  - if $\mathcal{Q} \in \Phi$ then $\geq n\, r.C$ and $\leq n\, r.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$,
  - if $\{\mathcal{Q}, \mathcal{I}\} \subseteq \Phi$ then $\geq n\, r^-.C$ and $\leq n\, r^-.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$,
  - if $\mathcal{U} \in \Phi$ then U is an object role of $\mathcal{L}_{\Sigma,\Phi}$,
  - if $\mathsf{Self} \in \Phi$ then $\exists r.\mathsf{Self}$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$. $\square$

Definition 2.2 (Semantics). 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 $A^\mathcal{I} : \Delta^\mathcal{I} \to range(A)$, 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. □

We say that $C^\mathcal{I}$ (reps. $R^\mathcal{I}$) is the *extension* of the concept C (resp. role R) in the interpretation $\mathcal{I}$. If $a^\mathcal{I} \in C^\mathcal{I}$, then we say that a is an *instance* of C in the interpretation $\mathcal{I}$. For abbreviation, we write $C^\mathcal{I}(x)$ (resp. $R^\mathcal{I}(x,y)$, $\sigma^\mathcal{I}(x,d)$) instead of $x \in C^\mathcal{I}$ (resp. $\langle x,y\rangle \in R^\mathcal{I}$, $\langle x,d\rangle \in \sigma^\mathcal{I}$).

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

Fig. 1: *Interpretation of complexity concepts and complexity object roles*

Example 1. Let $\Phi = \{\mathcal{I}\}$, $\Sigma_C = \{Human, Male, Female\}$, $\Sigma_{nA} = \{BirthYear\}$, $\Sigma_{dA} = \Sigma_C$, $\Sigma_{oR}=\{hasChild, marriedTo\}$, $\Sigma_{dR}=\emptyset$ and $\Sigma_I=\{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t\}$. Consider the interpretation $\mathcal{I}$ specified by:
$\Delta^\mathcal{I} = \{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t\}$, $a^\mathcal{I} = a, \ldots, t^\mathcal{I} = t$, $Human^\mathcal{I} = \Delta^\mathcal{I}$,
$Female = \{a,c,f,i,j,n,p,s\}$, $Male^\mathcal{I} = \Delta^\mathcal{I} \setminus Female = \{b,d,e,g,h,k,l,m,o,q,r,t\}$,
$marriedTo^\mathcal{I} = \{\langle a,b\rangle, \langle b,a\rangle, \langle b,c\rangle, \langle c,b\rangle, \langle j,k\rangle, \langle k,j\rangle\}$,
$hasChild^\mathcal{I} = \{\langle a,e\rangle, \langle a,f\rangle, \langle b,e\rangle, \langle b,f\rangle, \langle e,g\rangle, \langle e,h\rangle, \langle d,g\rangle, \langle d,h\rangle, \langle e,i\rangle, \langle f,j\rangle, \langle g,k\rangle, \langle h,l\rangle,$
$\langle i,m\rangle, \langle j,n\rangle, \langle j,o\rangle, \langle k,n\rangle, \langle k,o\rangle, \langle l,p\rangle, \langle m,q\rangle, \langle n,r\rangle, \langle o,s\rangle, \langle p,t\rangle\}$,
the function $BirthYear^\mathcal{I}$ are specified as usual.

This interpretation is illustrated in Figure 2. In this figure, each node denotes a person, the letter M stands for $Male$, the letter F stands for $Female$, the numbers (two last digits for short) stand for $BirthYear$, the dashed edges denote assertions of the role $hasChild$, and the solid edges denote assertions of the role $marriedTo$. $\square$

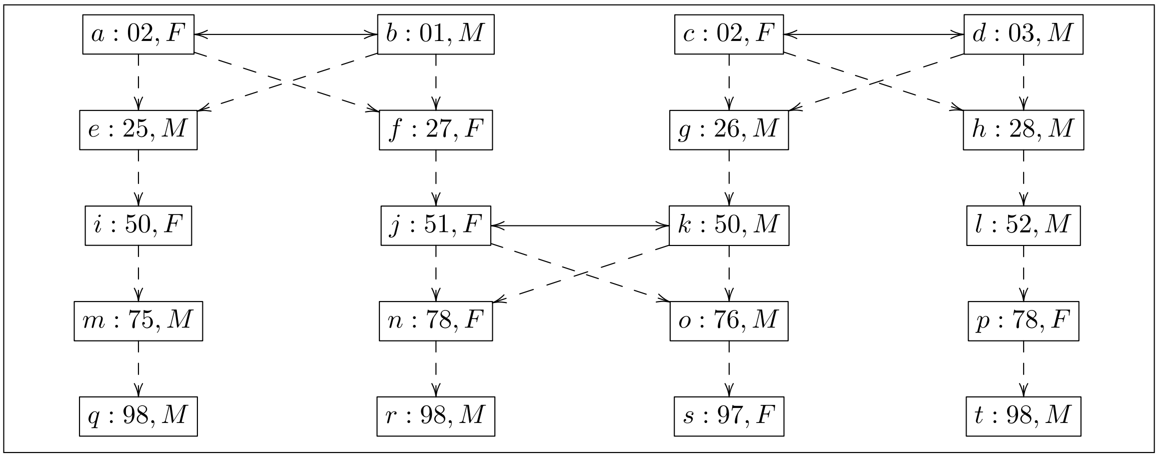

Fig. 2: *An illustration for the interpretation given in Example 1.*

3 Bisimulation and Indiscernibility

In [14] Divroodi and Nguyen studied bisimulation for a number of DLs. Nguyen and Szałas [9], Tran et al. [11] generalized that notion to model indiscernibility of objects and study concept learning. In this section, we recall the notion of bisimulation for DLs.

Definition 3.1 (Bisimulation)**.** 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 $\mathcal{I}$ 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 $a \in \Sigma_I^\dagger$, $A \in \Sigma_C^\dagger$, $B \in \Sigma_A^\dagger \setminus \Sigma_C^\dagger$, $r \in \Sigma_{oR}^\dagger$, $\sigma \in \Sigma_{dR}^\dagger$, $d \in \mathit{range}(\sigma)$, $x, y \in \Delta^\mathcal{I}$, $x', y' \in \Delta^{\mathcal{I}'}$:

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

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

$$Z(x, x') \Rightarrow [B^\mathcal{I}(x) = B^{\mathcal{I}'}(x') \text{ or both are undefined}] \tag{3}$$

$$[Z(x, x') \wedge r^\mathcal{I}(x, y)] \Rightarrow \exists y' \in \Delta^{\mathcal{I}'}[Z(y, y') \wedge r^{\mathcal{I}'}(x', y')] \tag{4}$$

$$[Z(x, x') \wedge r^{\mathcal{I}'}(x', y')] \Rightarrow \exists y \in \Delta^\mathcal{I}[Z(y, y') \wedge r^\mathcal{I}(x, y)] \tag{5}$$

$$Z(x, x') \Rightarrow [\sigma^\mathcal{I}(x, d) \Leftrightarrow \sigma^{\mathcal{I}'}(x', d)], \tag{6}$$

if $\mathcal{I} \in \Phi^\dagger$ 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 $\mathcal{O} \in \Phi^\dagger$ then

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

if $\mathcal{N} \in \Phi^\dagger$ then

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

if $\{\mathcal{N},\mathcal{I}\} \subseteq \Phi^\dagger$ then

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

if $\mathcal{F} \in \Phi^\dagger$ then

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

if $\{\mathcal{F},\mathcal{I}\} \subseteq \Phi^\dagger$ then

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

if $\mathcal{Q} \in \Phi^\dagger$ then

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

if $\{\mathcal{Q},\mathcal{I}\} \subseteq \Phi^\dagger$ then

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

if $\mathcal{U} \in \Phi^\dagger$ then

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

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

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

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

A concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ is said to be *invariant for $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation* if, for every interpretations $\mathcal{I}$ and $\mathcal{I}'$ in $\mathcal{L}_{\Sigma,\Phi}$ with $\Sigma \supseteq \Sigma^\dagger$ and $\Phi \supseteq \Phi^\dagger$, and every $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation Z between $\mathcal{I}$ and $\mathcal{I}'$, if $Z(x,x')$ holds then $x \in C^{\mathcal{I}}$ iff $x' \in C^{\mathcal{I}'}$.

Theorem 3.1. *All concepts of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ are invariant for $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation.* □

This theorem can be proved in a similar way as [14, Theorem 3.4]. By this theorem, $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimilarity formalizes indiscernibility by the sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$.

Definition 3.2. 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 $\mathcal{I} \in \Phi^\dagger$ then the set $\{y \in \Delta^{\mathcal{I}} \mid r^{\mathcal{I}}(y,x)\}$ is finite. $\square$

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

Definition 3.3 (Largest Auto-bisimulation)**.** An $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$*-auto-bisimulation of* $\mathcal{I}$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation between $\mathcal{I}$ and itself. 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 $\mathcal{I}$. $\square$

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'.

Theorem 3.2. *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:*

- *the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of $\mathcal{I}$ exists and is an equivalence relation.*
- *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).* $\square$

This theorem can be similarly proved as [14, 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 $\mathbb{Y} = \{Y_1, Y_2, \ldots, Y_n\}$ is *consistent* with a set X if, for every $1 \leq i \leq n$, Y_i is not divided by X.

Theorem 3.3. *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:*

- *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.*
- *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 $C^{\mathcal{I}} = X$.* $\square$

This theorem can be similarly proved as [9, Theorem 4].

4 Partitioning the Domain of an Interpretation

Let $\mathcal{I}$ be a finite interpretation in $\mathcal{L}_{\Sigma,\Phi}$ given as a training information system and $A_d \in \Sigma_C$ be a concept name standing for the "decision attribute". Let $E = \langle E^+, E^- \rangle$, where $E^+ = \{a \mid a^\mathcal{I} \in A_d^\mathcal{I}\}$ and $E^- = \{a \mid a^\mathcal{I} \in (\neg A_d)^\mathcal{I}\}$ be sets of *positive examples* and *negative examples* of A_d in $\mathcal{I}$, respectively. Suppose that A_d can be expressed by a concept C in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, where $\Sigma^\dagger \subseteq \Sigma \setminus \{A_d\}$ and $\Phi^\dagger \subseteq \Phi$. How can we learn that concept C on the basis of $\mathcal{I}$, E^+ and E^- such that:

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

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 $\mathbb{Y} = \{Y_1, Y_2, \ldots, Y_n\}$ 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 A_d is definable in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ by a concept C then:

- by the first assertion of Theorem 3.3, $C^\mathcal{I}$ 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 $a \in E^-$.

We now describe a bisimulation-based concept learning method for information systems in DLs via Algorithm 1. It is based on the general method of [9]. In this algorithm, the partition process of the domain of an interpretation uses basic selectors. A *basic selector* D_u in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ is a concept of the form given in Figure 3. In the work [11], together with our colleagues, we formulated and proved an important theorem on basic selectors, which states that: to reach the partition corresponding to the equivalence relation characterizing indiscernibility of objects w.r.t. to a given sublanguage of the considered logic it suffices to start from the full block consisting of all objects and repeatedly granulate it by using the basic selectors of the sublanguage. The correctness of Algorithm 1 can be proved by using this theorem.

Note that, in the partition process, we denote the blocks created so far in all steps by $Y_1, Y_2, \ldots, Y_n$, where the current partition $\mathbb{Y} = \{Y_{i_1}, Y_{i_2}, \ldots, Y_{i_k}\}$ consists of only some of them. We always use a new subscript for each newly created block by increasing n. We also take care that, for each $1 \leq i \leq n$:

- Y_i is characterized by a concept C_i such that $C_i^\mathcal{I} = Y_i$,
- we keep information whether Y_i is divided by E,

Algorithm 1: *Concept learning for description logic-based information systems*

Input: $\mathcal{I}$, $\Sigma^\dagger$, $\Phi^\dagger$, $E = \langle E^-, E^+ \rangle$
Output: A concept C such that:

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

$n := 1$; $Y_1 := \Delta^{\mathcal{I}}$; $\mathbb{Y} := \{Y_1\}$; $\mathbb{D} = \emptyset$; $\mathit{LargestContainer}[1] := 1$;
create and add basic selectors into $\mathbb{D}$; `/* using the rules in Figure 3 */`
while ($\mathbb{Y}$ *is not consistent with* E) and ($\mathbb{Y}$ *can partition*) **do**
    choose a selector $D_u \in \mathbb{D}$ and a block $Y_{i_j} \in \mathbb{Y}$ such that D_u divide Y_{i_j} into two non-empty parts;
    $s := n + 1$; $t := n + 2$; $n := n + 2$;
    $Y_s = Y_{i_j} \cap D_u^{\mathcal{I}}$; $C_s := C_{i_j} \sqcap D_u$;
    $Y_t = Y_{i_j} \cap (\neg D_u)^{\mathcal{I}}$; $C_t := C_{i_j} \sqcap \neg D_u$;
    if (Y_{i_j} *is not divided by* E) **then**
        $\mathit{LargestContainer}[s] := \mathit{LargestContainer}[i_j]$;
        $\mathit{LargestContainer}[t] := \mathit{LargestContainer}[i_j]$;
    else
        if (Y_s *is not divided by* E) **then**
            $\mathit{LargestContainer}[s] := s$;
        if (Y_t *is not divided by* E) **then**
            $\mathit{LargestContainer}[t] := t$;
    $\mathbb{Y} := \mathbb{Y} \cup \{Y_s, Y_t\} \setminus \{Y_{i_j}\}$;
    create and add new basic selectors into $\mathbb{D}$; `/* using the rules in Figure 3 */`
$\mathbb{J} = \emptyset$; $\mathbb{C} = \emptyset$;
if ($\mathbb{Y}$ *is consistent with* E) **then**
    foreach $Y_{i_j} \in \mathbb{Y}$ **do**
        if (Y_{i_j} *contains some* $a^{\mathcal{I}}$ *with* $a \in E^+$) **then**
            $\mathbb{J} = \mathbb{J} \cup \{\mathit{LargestContainer}[i_j]\}$;
    foreach $l \in \mathbb{J}$ **do**
        $\mathbb{C} = \mathbb{C} \cup \{C_l\}$;
    return $C := \bigsqcup \mathbb{C}$;
else
    return failure;

- $\mathit{LargestContainer}[i] := j$ is the subscript of the largest block Y_j such that $Y_i \subseteq Y_j$ and Y_j is not divided by E,
- the current set of selectors is denoted by $\mathbb{D} = \{D_1, D_2, \ldots, D_m\}$.

- A, where $A \in \Sigma_C^\dagger$,
- $A = d$, where $A \in \Sigma_A^\dagger \setminus \Sigma_C^\dagger$ and $d \in range(A)$,
- $\exists\sigma.\{d\}$, where $\sigma \in \Sigma_{dR}^\dagger$ and $d \in range(\sigma)$,
- $\exists r.C_{i_t}$, where $r \in \Sigma_{oR}^\dagger$ and $1 \leq t \leq k$,
- $\exists r^-.C_{i_t}$, if $\mathcal{I} \in \Phi^\dagger$, $r \in \Sigma_{oR}^\dagger$ and $1 \leq t \leq k$,
- $\{a\}$, if $\mathcal{O} \in \Phi^\dagger$ and $a \in \Sigma_I^\dagger$,
- $\leq 1\, r$, if $\mathcal{F} \in \Phi^\dagger$ and $r \in \Sigma_{oR}^\dagger$,
- $\leq 1\, r^-$, if $\{\mathcal{F}, \mathcal{I}\} \subseteq \Phi^\dagger$ and $r \in \Sigma_{oR}^\dagger$,
- $\geq l\, r$ and $\leq m\, r$, if $\mathcal{N} \in \Phi^\dagger$, $r \in \Sigma_{oR}^\dagger$, $0 < l \leq \#\Delta^\mathcal{I}$ and $0 \leq m < \#\Delta^\mathcal{I}$,
- $\geq l\, r^-$ and $\leq m\, r^-$, if $\{\mathcal{N}, \mathcal{I}\} \subseteq \Phi^\dagger$, $r \in \Sigma_{oR}^\dagger$, $0 < l \leq \#\Delta^\mathcal{I}$ and $0 \leq m < \#\Delta^\mathcal{I}$,
- $\geq l\, r.C_{i_t}$ and $\leq m\, r.C_{i_t}$, if $\mathcal{Q} \in \Phi^\dagger$, $r \in \Sigma_{oR}^\dagger$, $1 \leq t \leq k$, $0 < l \leq \#C_{i_t}^\mathcal{I}$ and $0 \leq m < \#C_{i_t}^\mathcal{I}$,
- $\geq l\, r^-.C_{i_t}$ and $\leq m\, r^-.C_{i_t}$, if $\{\mathcal{Q}, \mathcal{I}\} \subseteq \Phi^\dagger$, $r \in \Sigma_{oR}^\dagger$, $1 \leq t \leq k$, $0 < l \leq \#C_{i_t}^\mathcal{I}$ and $0 \leq m < \#C_{i_t}^\mathcal{I}$,
- $\exists r.\mathsf{Self}$, if $\mathsf{Self} \in \Phi^\dagger$ and $r \in \Sigma_{oR}^\dagger$.

Fig. 3: *Basic selectors*

The efficiency of Algorithm 1 depends on the granulation process of partitions of the domain of an interpretation. For Steps 2 and 17, the set $\mathbb{D} = \{D_1, D_2, \ldots, D_m\}$ is built by using the rules in Figure 3. For Step 4, which block from the current partition should be divided first and which selector should be used to divide it are left open for heuristics. For example, one can apply some gain function like the information gain measure, while taking into account also simplicity of selectors and the concepts characterizing the blocks. In the next section, we introduce some entropy-based measures for choosing blocks and selectors in the granulation process of partitions.

5 Entropy-Based Measures

For constructing entropy-based measures, we recall the entropy notion and appropriate it in the context of DLs. Let $\Delta^\mathcal{I}$ be a domain of the interpretation $\mathcal{I}$, X and Y be subsets of $\Delta^\mathcal{I}$, where X plays the role of a set of positive examples for the concept to be learnt.

Definition 5.1. The *entropy* of Y w.r.t. X in $\Delta^\mathcal{I}$, denoted by $E_{\Delta^\mathcal{I}}(Y/X)$, is defined as follows:

$$E_{\Delta^\mathcal{I}}(Y/X) = \begin{cases} 0, \text{ if } Y \cap X = \emptyset \text{ or } Y \subseteq X \\ -\dfrac{\#XY}{\#Y}\log_2\dfrac{\#XY}{\#Y} - \dfrac{\#\overline{X}Y}{\#Y}\log_2\dfrac{\#\overline{X}Y}{\#Y}, \text{otherwise} \end{cases} \tag{19}$$

where XY stands for the set $X \cap Y$ and $\overline{X}Y$ stands for the set $\overline{X} \cap Y$. □

Entropy is an information-theoretic measure of the uncertainty contained in an information system treated as a training set, due to the presence of more than one possible classification. It takes its minimum value (zero) if and only if all the examples have the same classification, in which case there is only one non-empty class, for which the probability is 1. In other words, there exits only one set of examples which is not divided by the set of positive ones. Entropy takes its maximum value when the examples are equally distributed amongst the two possible sets.

Remark 1. *According to Equation* (19)*, we see that* $E_{\Delta^{\mathcal{I}}}(Y/X) = 0$ *iff* Y *is not divided by* X*.*

5.1 Information Gain Measure

We want to determine which attribute in a given set of training features is most useful for discriminating between the classes to be learned. In [15], Quinlan suggested information gain which is used for deciding the ordering of attributes in the nodes of a decision tree. In the context of DLs, we formulate information gain for a selector to divide a block in a partition.

Definition 5.2. *Information gain* (IG) for a selector D to divide Y w.r.t. X in $\Delta^{\mathcal{I}}$, denoted by $IG_{\Delta^{\mathcal{I}}}(Y/X, D)$, is defined as follows:

$$IG_{\Delta^{\mathcal{I}}}(Y/X, D) = E_{\Delta^{\mathcal{I}}}(Y/X) - \left(\frac{\#D^{\mathcal{I}}Y}{\#Y} E_{\Delta^{\mathcal{I}}}(D^{\mathcal{I}}Y/X) + \frac{\#\overline{D^{\mathcal{I}}}Y}{\#Y} E_{\Delta^{\mathcal{I}}}(\overline{D^{\mathcal{I}}}Y/X) \right), \quad (20)$$

where $D^{\mathcal{I}}Y$ stands for the set $D^{\mathcal{I}} \cap Y$ and $\overline{D^{\mathcal{I}}}Y$ stands for the set $\overline{D^{\mathcal{I}}} \cap Y$. □

IG measure is based on the decrease in entropy after an information system is split on a block by a selector. Constructing a decision tree is all about finding a block as well as a selector that returns the highest IG.

In the case $\Delta^{\mathcal{I}}$ and X are clear from the context, we write $E(Y)$ instead of $E_{\Delta^{\mathcal{I}}}(Y/X)$ and $IG(Y, D)$ instead of $IG_{\Delta^{\mathcal{I}}}(Y/X, D)$.

For each block $Y_{i_j} \in \mathbb{Y}$ $(1 \le j \le k)$, we want to choose the best selector for dividing Y_{i_j}. Let S_{i_j} be the simplest selector from the set $\arg\max_{D \in \mathbb{D}} \{IG(Y_{i_j}, D)\}$.[1] By IG measure, S_{i_j} is one of the best selectors which should be used to divide Y_{i_j}.

After choosing the selectors for blocks, we have to decide which block should be divided first. We will choose a block Y_{i_j} such that applying the selector S_{i_j} to divide Y_{i_j} maximizes IG. That is, we divide a block $Y_{i_j} \in \arg\max_{Y_{i_j} \in \mathbb{Y}} \{IG(Y_{i_j}, S_{i_j})\}$ first and have a new partition. Then we also add new selectors which are created by using the rules in Figures 3 to the current set of selectors (number of new selectors depends on DLs, roles, concepts, ... for training). This set is used to continue granulating the new partition.

Example 2. We will consider the interpretation $\mathcal{I}$ given in Example 1. Assume that we want to learn a definition of $X = \{k, l, m, o, q, r, t\}$ (i.e., $E^+ = \{k, l, m, o, q, r, t\}$ and $E^- =$

[1] The simplest selector can be determined as in [12]

$\{a, b, c, d, e, f, g, h, i, j, n, p, s\}$) in the sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, where $\Sigma^\dagger = \{hasChild, Male\}$ and $\Phi^\dagger = \{\mathcal{I}\}$. One can think of X as the set of instances of the concept $GrandSon = Male \sqcap \exists hasChild^-.(\exists hasChild^-.\top)$ in $\mathcal{I}$.

Using IG measure to choose blocks in the granulation process of partitions. The steps are illustrated by the decision tree in Figure 4.

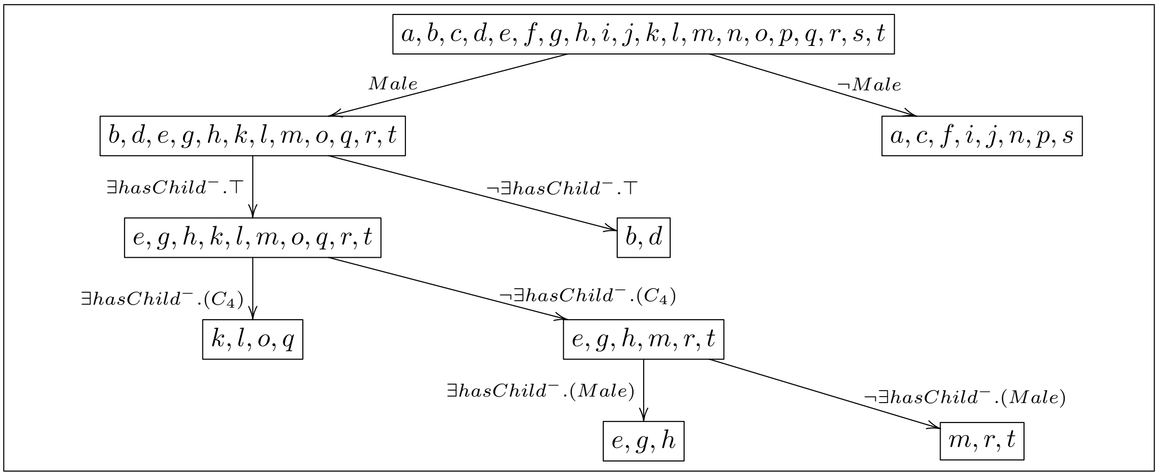

Fig. 4: *A decision tree illustrating the granulation process using IG measure.*

In Figure 4, the concept C_4 is $Male \sqcap \exists hasChild^-.\top$. It characterizes the set of instances $\{e, g, h, k, l, m, o, q, r, t\}$ in the interpretation $\mathcal{I}$.

The resulting concept is $Male \sqcap \exists hasChild^-.\top \sqcap (\exists hasChild^-.(Male \sqcap \exists hasChild^-.\top) \sqcup (\neg\exists hasChild^-.(Male \sqcap \exists hasChild^-.\top) \sqcap \neg\exists hasChild^-.Male))$.

This concept can be simplified to $Male \sqcap \exists hasChild^-.\top \sqcap (\exists hasChild^-.(Male \sqcap \exists hasChild^-.\top) \sqcup \neg\exists hasChild^-.Male)$. □

The granulation process of partitions in the context of DLs differs from the traditional decision tree in that blocks are divided not only by attributes (understood as concepts) but also by concepts which are constructed from roles and other ones. The list of concepts will be changed and depends on choosing blocks and selectors to be divided in the granulation process as in Section 4. Therefore, the traditional IG measure is not good enough for choosing blocks and selectors in the granulation process. In the next subsection, we propose another entropy-based measure for this problem.

5.2 First Depth Information Gain Measure

Different from IG, it is worth to look one or a few steps deeper to see which block should be divided first and which selector should be used to divide it. This process is like finding a move in a game with a given depth. A selector is recognized as the best at the current step can be not good for the next steps and otherwise.

For each block $Y_{i_j} \in \mathbb{Y}$ $(1 \leq j \leq k)$ we will consider all selectors which can be used to divide Y_{i_j}. Let $D_u \in \mathbb{D}$ be a selector for dividing Y_{i_j}. We obtain a new partition $\mathbb{Y} = \mathbb{Y} \cup \{Y_s, Y_t\} \setminus \{Y_{i_j}\}$ and a new set of selectors $\mathbb{D}^u_{i_j} = \mathbb{D} \cup \{D^u_{i_j,1}, D^u_{i_j,2}, \ldots, D^u_{i_j,m^u_{i_j}}\}$, where $m^u_{i_j}$ is the number of new selectors and $D^u_{i_j,l}$ $(1 \leq l \leq m^u_{i_j})$ is a new selector which is created by using the rules in Figure 3.

Definition 5.3. *First depth information gain* (fdIG) for a selector D_u to divide a block Y_{i_j}, denoted by $fdIG(Y_{i_j}, D_u)$, is defined as follows:

$$fdIG(Y_{i_j}, D_u) = IG(Y_{i_j}, D_u) + \sum_{\substack{v=1 \\ v \neq j}}^{k} \max_{D \in \mathbb{D}^u_{i_j}} \{IG(Y_{i_v}, D)\} + \max_{D \in \mathbb{D}^u_{i_j}} \{IG(Y_s, D)\} + \max_{D \in \mathbb{D}^u_{i_j}} \{IG(Y_t, D)\}. \square \quad (21)$$

Compared to IG, fdIG is a better measure simply because fdIG allows us to look forward the benefits from new selectors when a block is divided. By fdIG, we first choose $Y_{i_j} \in \mathbb{Y}$ as well as $D_u \in \mathbb{D}$ for dividing such that $fdIG(Y_{i_j}, D_u)$ is maximum.

Example 3. Consider Example 2 but using fdIG measure to choose blocks in the granulation process of partitions. The steps are illustrated by the decision tree in Figure 5.

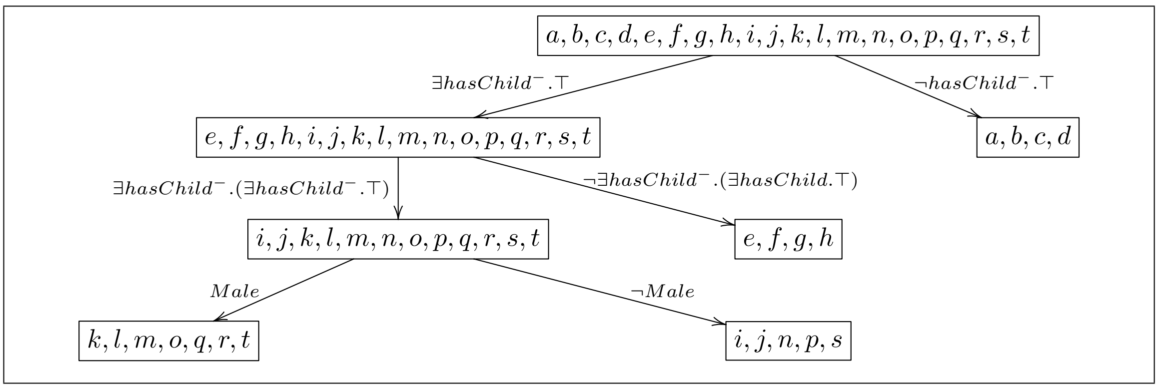

Fig. 5: *A decision tree illustrating the granulation process using fdIG measure.*

The resulting concept is $\exists hasChild^{-}.\top \sqcap \exists hasChild^{-}.(\exists hasChild^{-}.\top) \sqcap Male$.

This concept can be simplified to $Male \sqcap \exists hasChild^{-}.(\exists hasChild^{-}.\top)$. $\square$

Compare to Example 2, the selector $\exists hasChild^{-}.\top$ is chosen to divide the block $Y_1 = \{a, b, c, d, e, f, g, h, i, j, k, l, m\}$ instead of the selector $Male$ because using the concept $\exists hasChild^{-}.\top$ allows us to create new selectors ($\exists hasChild^{-}.(\exists hasChild^{-}.\top)$) which are better than using the concept $Male$ in the next steps. This would also suitable for other blocks of partitions which are created in next steps.

We see that the decision tree in Example 3 is simpler than the one in Example 2. Therefore, the resulting concept in Example 2 is more complex than the one in Example 3. These examples show that using fdIG measure is better than using IG one. The former reduces the number of iterations of the main loop and the complexity of resulting concept.

6 Conclusion

We describe a large of class of DLs and semantics. We outline bisimulation-based concept learning method for partitioning the domain of an interpretation in DLs. Apart from information gain, we introduce a new entropy-based measure, called first depth information gain, for choosing blocks as well as selectors in the granulation process of partitions. The illustrative examples point out the efficiency of the proposed measure. As future work, we intend to implement our learning method using mentioned measures and evaluate its performance.

Acknowledgment

This work was supported by Hue University under Grant No. DHH2013-01-41.

References

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

[2] I. Horrocks, P. F. Patel-Schneider, D. L. Mcguinness, and C. A. Welty, "OWL: a description logic based ontology language for the semantic web," in *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2007, pp. 458–486.

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

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

[5] L. Badea and S.-H. Nienhuys-Cheng, "A refinement operator for description logics," in *Proceedings of the 10th International Conference on Inductive Logic Programming*, ser. ILP'2000. Springer-Verlag, 2000, pp. 40–59.

[6] L. Iannone, I. Palmisano, and N. Fanizzi, "An algorithm based on counterfactuals for concept learning in the semantic web," *Applied Intelligence*, vol. 26, no. 2, pp. 139–159, 2007.

[7] N. Fanizzi, S. Ferilli, L. Iannone, I. Palmisano, and G. Semeraro, "Downward refinement in the $\mathcal{ALN}$ description logic," in *Proceedings of HIS'2004*. IEEE Computer Society, 2004, pp. 68–73.

[8] J. Lehmann and P. Hitzler, "A refinement operator based learning algorithm for the $\mathcal{ALC}$ description logic," in *Proceedings of ILP'2007*. Springer-Verlag, 2008, pp. 147–160.

[9] L. A. Nguyen and A. Szałas, "Logic-based roughification," in *Rough Sets and Intelligent Systems (1)*. Springer, 2013, pp. 517–543.

[10] 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*. ACM, 2012, pp. 241–249.

[11] 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*. IEEE Computer Society, 2012, pp. 65–73.

[12] T.-L. Tran, L. A. Nguyen, and T.-L.-G. Hoang, "A domain partitioning method for bisimulation-based concept learning in description logics," in *Proceedings of ICC-SAMA'2014*, vol. 282. Springer International Publishing, 2014, pp. 297–312.

[13] T.-L. Tran, Q.-T. Ha, T.-L.-G. Hoang, L. A. Nguyen, and H. S. Nguyen, "Bisimulation-based concept learning in description logics." in *Proceedings of CS&P'2013*. CEUR-WS.org, 2013, pp. 421–433.

[14] A. R. Divroodi and L. A. Nguyen, "On bisimulations for description logics," in *Proceedings of CS&P'2011*, 2011, pp. 99–110.

[15] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.