Assertional Mining in Description Logics

Stefan Schlobach
Department of Computer Science, King's College London
Strand, London WC2R 2LS, UK
Email: schlobac@dcs.kcl.ac.uk

Abstract

This paper introduces a new framework to integrate knowledge discovery methods into knowledge representation systems based on description logic reasoning. We define necessary criteria for a concept to be accepted as learned from an A–Box in this framework and present a family of algorithms to calculate these concepts. For the simple description logic $\mathcal{AL}^{\mathbb{R},=}$ we show that we can simulate results of state-of-the-art data mining mechanisms based on these algorithms.

1 Introduction

Knowledge acquisition has always played an important role in Artificial Intelligence and the relevance of learning in description logics (DL) is widely accepted. Learning methods are the core of the data mining part of the knowledge discovery process from databases (KDD), where the databases can be potentially huge and not usually built for learning purposes.

The research presented in this paper is focused on the combination of knowledge representation using description logics and knowledge discovery with the application of rough set theory as a well established data mining method. In this paper we define the scope of learning in description logics and present algorithms to calculate concepts that are *learned* from the assertional component of a knowledge representation system.

For this purpose we introduce *generalised decision concepts* (GDCs) and present a general algorithmic framework $\mathbf{ABM}_{\mathcal{DL}}$ to calculate GDCs for a family $\mathcal{DL}$ of description logics. A GDC for a given A–Box $\mathcal{A}$ and an expert-defined decision concept D is a concept which is covered for all the examples by D and excludes all counterexamples for D in $\mathcal{A}$.

Optimality of GDCs is defined for the very simple description logic $\mathcal{AL}^{\mathbb{R},=}$ and it is shown that the algorithm $\mathbf{ABM}_{\mathcal{AL}^{\mathbb{R},=}}$ can easily be adapted to calculate $\mathcal{AL}^{\mathbb{R},=}$–optimal GDCs. Furthermore we prove that classification using optimal decision rules calculated through rough set data analysis can be simulated through instance checking for $\mathcal{AL}^{\mathbb{R},=}$–optimal GDCs.

This paper is organised in the following way. In section 2 we present the paradigm for learning in description logics. Section 3 gives a brief introduction into the method of data mining using rough set theory. These methods are closely linked to the algorithmic framework $\mathbf{ABM}_{\mathcal{DL}}$ which is presented in section 4.1. Theorem 4.1 provides soundness and completeness for this method and therefore constitutes one of the core results of this paper. Of more practical importance is a sketch of the algorithm $\mathbf{ABM}_{\mathcal{AL}^{\mathbb{R},=}}$ for the logic $\mathcal{AL}^{\mathbb{R},=}$ as presented in section 4.2 and theorem 4.3 which describes the similarity of our approach with the rough set data mining method with respect to learning of decision systems.

A more detailed description of the algorithms and the proofs of the theorems can be found in [Sch00].

2 A–Box Mining in Description Logics

Description Logics Preliminaries. For the purpose of A–Box Mining we investigate a very simple description logic $\mathcal{AL}^{\mathbb{R},=}$ which allows for conjunction ($C \sqcap D$), atomic negation ($\neg A$), $\top$, $\bot$, qualified universal ($\forall R.C$), unqualified existential quantification ($\exists R.\top$), and number-valued functions over rational numbers with equality statements $f = x$ ($x \in \mathbb{R}$).

A *T–Box* is a finite set of terminological axioms[1] $A \sqsubseteq C$ or $A \doteq C$ (where A is atomic). Similarly, an *A–Box* is a finite set of conceptual ($a : C$) and relational ($(a, b) : R$) assertional axioms. The set theoretical semantics and the reasoning mechanisms satisfiability (we abbreviate $C \neq \emptyset$), subsumption ($C \sqsubseteq D$) and instance checking ($d \in_{\mathcal{A}} C$) with respect to an A–Box $\mathcal{A}$ are defined as usual.

The general ideas of A–Box mining are independent of a particular description logic. For this reason we define the set of languages $\mathcal{DL}$, to which our approach is applicable as the set of description logics containing $\mathcal{AL}^{\mathbb{R},=}$ where subsumption, consistency and instance checking is decidable.

Let $\mathcal{T}$ be a T–Box and $\mathcal{A}$ be an A–Box over $\mathcal{DL}$. A *D–Box* $\mathcal{D} = \{D_1, \ldots, D_n\}$ (for *decision*) is a finite set of $\mathcal{DL}$ concepts covering the A–Box universe (i.e., for every a occurring in $\mathcal{A}$ there must be a $D_i \in \mathcal{D}$ such that $a \in_{\mathcal{A}} D_i$) which will be called decision concepts. Instance checking for an object a and a decision D_i will be called *deciding object a*, the result of deciding all objects in the A–Box for all decisions is a *classification* of the A–Box. The triple $(\mathcal{T}, \mathcal{A}, \mathcal{D})$ will be called the *hybrid system over $\mathcal{DL}$*.

[1]For complexity reasons we restrict our T–Boxes to be unfoldable.

Learning and Mining in Description Logics. The main objective of A–Box mining is to generate concepts from a given A–Box $\mathcal{A}$ that can replace the decisions in the D–Box for classification. We will introduce and define two necessary criteria (called covering and exclusiveness) which define our notion of concepts *learned* from $\mathcal{A}$ and $\mathcal{D}$ (and possibly $\mathcal{T}$). These concepts will be called *generalised decision concepts* (GDCs).

The first condition states that all objects in the A–Box that are instances of a GDC learned from a decision D must be instances of D, i.e., all A–Box instances of a GDC must be covered by the corresponding decision. This condition ensures correctness of the classification through the GDC with respect to the original decision. The second condition denotes the fact that any A–Box object, which is an instance of a GDC, must by unambiguously decidable with the corresponding original decision only. Let us define these notions formally:

Definition 2.1 *A concept* $\underline{\mathcal{A}}D_i$ *is called* $\mathcal{A}$-covered *by a decision* D_i *if for all A–Box objects* a *in* $\mathcal{A}$*:*

$$a \in_{\mathcal{A}} \underline{\mathcal{A}}D_i \Rightarrow a \in_{\mathcal{A}} D_i,$$

and if there is an A–Box object o *such that* $o \in_{\mathcal{A}} \underline{\mathcal{A}}D_i$*. A concept which is* $\mathcal{A}$*-covered by a decision* D_i *is called* exclusive *if for all A–Box objects* a *in* $\mathcal{A}$ *and for all* $j \neq i$*:*

$$a \in_{\mathcal{A}} \underline{\mathcal{A}}D_i \Rightarrow a \notin_{\mathcal{A}} D_j.$$

This gives rise to the definition of the class of learned concepts that we are interested in; the generalised decision concepts (GDCs).

Definition 2.2 *A* generalised decision concept (GDC) *for a decision* D *is a concept, which is exclusive and* $\mathcal{A}$*-covered by* D*.*

Remark 2.3 *The underlying paradigm for learning with Least Common Subsumers could be formalised using a dual notion, called* $\mathcal{A}$*-cover. Let* $a \in^{classify} D_i$ *denote that an object is classified into a decision* D_i*, independent of the classification method. A concept* $LCS(D_i)$ *is an* $\mathcal{A}$-cover *for a decision* D_i *if for all A–Box elements* a*:* $a \in^{classify} D_i \Rightarrow a \in_{\mathcal{A}} LCS(D_i)$*. In the presence of rough data, i.e., if the decisions are not disjoint with respect to* $\mathcal{A}$*, this condition contradicts exclusiveness and is therefore not considered in the rest of this paper.*

Simplicity and Optimality. Obviously the definition for GDCs gives necessary but not sufficient criteria to justify the quality of learned concepts. Depending on the chosen language, there might be infinitely many GDCs for a decision. There are two further criteria accepted in the learning community which we introduce informally, namely that GDCs should be "as general as possible" (semantic optimality) and "as simple as possible" (syntactic simplicity). For some languages (e.g. for $\mathcal{AL}^{\mathbb{R},=}$ as we will claim in section 4.2) the formal notions of optimality and simplicity coincide, and sound and complete mechanisms to calculate optimal GDCs can be defined. For other languages and learning methods, the quality of new notions for "most general and simple concepts" will be more difficult to assess and both empirically and theoretically developped heuristics have to be incorporated into the learning process.

3 Data Mining with Rough Sets

Rough set theory was created by Pawlak in the early eighties [Paw82] and has since then become very popular. One of its applications is data mining. We will present the basic ideas in a very compact way following Bazan [Baz98].

We break down the description of rough set data mining techniques into two parts, the description of rough set approximation and of rough analysis. Rough analysis is the classification of objects into equivalence classes, calculation of reducts (or dependencies and keys) and the deduction of decision rules.

Rough Set Theory. Let R be an equivalence relation on a universe U. The $R-$lower and $R-$upper approximation of a subset $X \subseteq U$ of U are:

$$\underline{R}X = \{x \in U \mid [x]_R \subseteq X\} \text{ and } \overline{R}X = \{x \in U \mid [x]_R \cap X \neq \emptyset\}$$

An approximation for X through R is the pair $app_R(X) = (\underline{R}X, \overline{R}X)$ with the support degree $spt_R(X) = (\frac{|\underline{R}X|}{|U|})$ for X and the accuracy $acc_R(X) = (\frac{|\underline{R}X|}{|\overline{R}X|})$. The R-Boundary (R-Outside) region of X is defined as $bound_R(X) = \overline{R}X - \underline{R}X$ ($out_R(X) = U - \overline{R}X$). A set is said to be *rough* (resp. *crisp*) if the boundary region is non-empty (resp. empty).

Rough Analysis. A decision system $\mathbf{D}$ is a tuple $(U, C \cup \{d\})$, where U is a finite (non-empty) set of objects, called the universe, C a set of condition attributes and $d \notin C$ an attribute called the decision attribute.

With an attribute a we associate the equivalence relation R_a (indiscernability) on U so that uR_av if and only if $a(u) = a(v)$. The collection $U/R_a = \{[u]_{R_a} \mid u \in U\}$ is called a classification. With a set of attributes A we associate an equivalence relation R_A on U so that uR_Av if and only if for all $a \in A$, $a(u) = a(v)$. Again, the collection $U/R_C = \{[u]_{R_A} \mid u \in U\}$ is a classification. The approximations $app_{R_C}(D) = (\underline{R_C}D, \overline{R_C}D)$ for a given decision system $\mathbf{D}$ and for each set $D \in U/R_d$ can be calculated in polynomial time.

These approximations are the starting point for the calculation of generalisations of the decisions. We will use reducts to compute optimal basic decision rules [Sko93]. Let $\mathbf{D} = (U, C \cup \{d\})$ be a decision system. A *reduct* of $\mathbf{D}$ for an equivalence class $E' \in U/R_C$ with an element e' is the smallest set of attributes $C' \subseteq C$ such that

$$\{v \mid \exists e\ (eR_C e'\ \&\ d(e) = v)\} = \{v \mid \exists e\ (eR_{C'} e'\ \&\ d(e) = v)\}.$$

If $Red = \{a_1, \ldots, a_k\}$ is a reduct, d a decision attribute and v_i a value for attribute a_i (and v_d for d), *optimal decision rules* are of the form

$$(a_1 = v_1) \wedge (a_2 = v_2) \wedge \ldots \wedge (a_k = v_k) \Rightarrow (d = v_d).$$

An object o is *matched* by a decision rule $\phi \Rightarrow (d = v_d)$ iff $o \in (\phi)^{\mathcal{I}}$ where $\mathcal{I}$ is the standard Boolean set-theoretic interpretation over U. Optimal decision rules are usually created for all equivalence classes $E \subseteq \underline{R_C}D$ for $D \in U/R_d$.

The *discernibility matrix modulo* d of $\mathbf{D}$ is $M(\mathbf{D}) = (c_{ij})$ assuming $c_{ij} = \{a \in C \mid a(e_i) \neq a(e_j)\}$ if $d(e_i) \neq d(e_j)$ for $e_i \in E_i$ and $e_j \in E_j$ and $c_{ij} = \emptyset$ otherwise. The *k-relative discernibility function* for an equivalence class $E_k \in U/R_C$ is defined as

$$f(\mathbf{D}, k) = \bigwedge_{i \neq k, \overline{c_{ik}} \neq \emptyset} \left(\bigvee_{\overline{a} \in \overline{c_{ik}}} \overline{a} \right)$$

where $\overline{c_{ik}} = \{\overline{a} \mid a \in c_{ik}\}$ and $\overline{a}$ is a Boolean attribute corresponding to a. It can be shown that the set of all *prime implicants* of $f(\mathbf{D}, k)$ determines the set of all k-relative reducts [Sko93] of $\mathbf{D}$.

4 GDC–Algorithms for $\mathcal{DL}$ Languages

4.1 The general framework

In section 2 a definition has been given for the class of *interesting* concepts *learned* from an A–Box: the generalised decision concepts. We will now present an algorithmic framework $\mathbf{ABM}_{\mathcal{DL}}$ to calculate GDCs.

Input:	An A–Box $\mathcal{A}$ and a D–Box $\mathcal{D} = \{D_1, \ldots, D_n\}$.
Output:	The set $\underline{G}(D_i)$ of all GDCs for each decision D_i.

1. Algorithm $\mathbf{AClass}_{\mathcal{DL}}$: *A–Box classification* is the collection of A–Box objects into equivalence classes. A–Box classification of $\mathcal{A}$ using an equivalence relation $\equiv$ will produce a set $E = \{E_1, \ldots, E_m\}$ of temporary concept axioms where each of the E_i corresponds to the logical description of an equivalence class $[u]^{E_i}_{\equiv}$. The only requirement for the equivalence relation $\equiv$ and the logical description E_i is that for all equivalence classes $[u]^{E_i}_{\equiv}$ and for all objects v in the A–Box $\mathcal{A}$, $v \in [u]^{E_i}_{\equiv} \Rightarrow v \in_{\mathcal{A}} E_i$.[2]

[2]This condition is sufficient because GDCs are defined related to a decision whereas the

2. Algorithm $\mathbf{TClass}_{\mathcal{DL}}$**:** If we consider $Temp = E \cup \mathcal{D}$ as a temporary T–Box we can calculate the corresponding subsumption hierarchy. The set S of the direct subconcepts $S^{D_i} = \{S_1^{D_i}, \ldots, S_{m_i}^{D_i}\}$ of a decision concept D_i will be the starting point for the calculation of the approximations.

3. Algorithm $\mathbf{Sep}_{\mathcal{DL}}$**:** In this step we analyse the sets S^{D_i} by separating the concepts into two classes, $\underline{S^{D_i}}$, which are the subconcepts of D_i that are inconsistent with any other subconcept of the other decisions and the boundary cases $bound(D_i)$ which are subconcepts that might be consistent with subconcepts of other decisions.

$$\underline{S^{D_i}} = \{S_j^{D_i} \mid S_j^{D_i} \sqcap S_l^{D_k} = \emptyset, \text{ for all } k \neq i, 1 \leq l \leq m_k, 1 \leq k \leq m\}.$$

Since $S^{D_i} = \{S_1^{D_i}, \ldots, S_{m_i}^{D_i}\} = \overline{S^{D_i}}$ can be seen as an upper approximation for D_i the boundary region for D_i in $\mathcal{A}$ will be defined as: $bound(D_i) = S^{D_i} - \underline{S^{D_i}}$.

4. Algorithm $\mathbf{Gen}_{\mathcal{DL}}$**:** The next step is to generalise the concepts in $\underline{S^{D_i}}$. A *lower generalisation* $\underline{gen}(D_i, S_j^{D_i})$ for a subconcept $S_j^{D_i} \in \underline{S^{D_i}}$ is the set of superconcepts that are inconsistent with any of the subconcepts of all the other decision concepts.

$$\begin{aligned}\underline{gen}(D_i, S_j^{D_i}) = \quad \{X \mid \; & S_j^{D_i} \sqsubseteq X \text{ and } S_l^{D_k} \sqcap X = \emptyset \\ & \text{for all } D_k \in \mathcal{D}, 1 \leq k \leq m, 1 \leq l \leq m_k, l \neq i\}\end{aligned}$$

The set of *generalised decisions* $\underline{G}(D_i)$ for a decision concept D_i is defined as the set of all those generalisations for $S_j^{D_i} \in \underline{S^{D_i}}$.

The following theorem is a soundness and completeness result for our learning framework in the sense that all the concepts calculated by the method are in fact GDCs and that $\underline{G}D_i$ is indeed the set of all GDCs.

Theorem 4.1 *$\underline{G}(D_i)$ is the set of all generalised decision concepts* (GDCs) *for a decision D_i as defined in definition 2.2.*

The completeness part of this result has to be interpreted carefully. Since there might be infinitely many GDCs the definition of lower generalisations cannot be constructive in some cases. For others, as shown in section 4.2 for the logic $\mathcal{AL}^{\mathbb{R},=}$, the set of superconcepts is finite and therefore $\underline{G}(D_i)$ can be constructed.

There are two elements of this framework which depend on the choice of a particular language in $\mathcal{DL}$ and on the different new learning method, namely the definition of the equivalence relation on the objects of the A–Box and the calculation of the superconcepts of a concept. It will be these two notions that have to be investigated for each particular case.

---

quality of the equivalence relation already affects the classification of A–Box objects into decisions. The definition of the equivalence relation is of great importance for the learning process, even though not directly for the construction of GDCs. For any particular language in $\mathcal{DL}$ and for each different learning method, the equivalence relation has to be defined separately, for example using logical equivalence of most specific concepts for each A–Box object or explicite equivalence definitions using special (i.e. similarity or equivalence) roles.

4.2 Optimal GDCs for $\mathcal{AL}^{\mathbb{R},=}$

The structure of the language $\mathcal{AL}^{\mathbb{R},=}$ allows us to define simplicity and optimality in a very simple way using concept subsumption. This is due to the fact that because there is no disjunction, more general concepts are usually shorter.

Definition 4.2 *An $\mathcal{AL}^{\mathbb{R},=}$ concept is an $\mathcal{AL}^{\mathbb{R},=}$–*optimal *GDC if it is a GDC which is not subsumed by any other GDC.*

We show that we can use our framework to calculate optimal GDCs for $\mathcal{AL}^{\mathbb{R},=}$ by constructing an equivalence relation on A–Box elements and adapting the calculation of the lower generalisations to $\mathcal{AL}^{\mathbb{R},=}$. Let $\mathbf{ABM}_{\mathcal{AL}^{\mathbb{R},=}}$ denote the instance of the framework $\mathbf{ABM}_{\mathcal{DL}}$ for the language $\mathcal{AL}^{\mathbb{R},=}$.

The best way to define an equivalence relation for $\mathcal{AL}^{\mathbb{R},=}$ A–Boxes is to use logical equivalence on the most specific concept (msc) for each object of the A–Box. In $\mathcal{AL}^{\mathbb{R},=}$ this is possible, because the calculation of mscs in $\mathcal{AL}^{\mathbb{R},=}$ is the straightforward collection of conceptual assertions and implicit knowledge, where only assertional combinations such as $(a, b) : r$ and $a : \forall rC$ (implying $b \in_{\mathcal{A}} C$ so that $b : C$ should be added for A–Box classification) need to be saturated.

Since the set of superconcepts for any $\mathcal{AL}^{\mathbb{R},=}$ concept is finite, the generalisation process (step 4 of $\mathbf{ABM}_{\mathcal{DL}}$) is straightforward as well. Naive methods could involve the creation of all the superconcepts and consistency checks with all subconcepts of the other decisions. More elaborated versions could use Boolean methods (for example prime implicants) similar to those proposed in data mining with rough sets [Sko93]. Both methods describes sound and complete procedures to calculate optimal GDCs for an $\mathcal{AL}^{\mathbb{R},=}$ A–Box.

Generalised Decision Concepts and Decision Systems. We decided to present an algorithm to calculate GDCs for $\mathcal{AL}^{\mathbb{R},=}$ A–Boxes because of the close connection between $\mathcal{AL}^{\mathbb{R},=}$ and decision systems and the existence of efficient and reliable algorithms for data mining of decision systems. Decision systems as defined in section 3 can be translated into $\mathcal{AL}^{\mathbb{R},=}$ A–Boxes. Every continuous attribute is translated into a number valued function, every Boolean attribute into atomic concepts and every nominal attribute into a set of atomic concepts plus inconsistency axioms. Furthermore, the syntactical notion of a reduct coincides with the semantical notion of maximality in the subsumption hierarchy of lower generalisations. These observations and the construction of $\mathbf{ABM}_{\mathcal{DL}}$ imply the following theorem:

Theorem 4.3 *For each decision system $\mathbf{D} = (U, A \cup \{d\})$ there is a hybrid system $(\emptyset, \mathcal{A}_{\mathcal{AL}^{\mathbb{R},=}}, \mathcal{D}_{\mathcal{AL}^{\mathbb{R},=}})$ such that instance checking for $\mathcal{AL}^{\mathbb{R},=}$–optimal* GDCs *for decisions $(d = v_i)$ is equivalent to the matching of optimal decision rules as defined in section 3.*

4.3 Quality Assessment

Accuracy and support degree play an important role in the data mining process. Fortunately it is easy to define and calculate the quality of a generalised concept in terms of the number of A–Box elements supporting it. Given the cardinality of a concept as the sum of the number of elements of its subconcepts related to the A–Box, the degree of support for a lower generalisation $\underline{gen}(D_i, S_j^{D_i}))$ is $spt_{\mathcal{A}}(\underline{gen}(D_i, S_j^{D_i})) = (\frac{|S_j^{D_i}|}{|U|})$. The accuracy $acc_{\mathcal{A}}(\underline{G}D_i) = (\frac{|\underline{G}D_i|}{|\overline{G}D_i|})$ also depends on the number of A–Box instances of concepts $\overline{G}D_i$, which is the upper approximation or the set of concepts that are non-exclusively covered by D_i.

4.4 Example

We formalise an example about an academic who decides the admission procedure for a postgraduate course. We assume that every student in the past achieved success (concept `S`) or failure (`F`), but we do not exclude that a student could have achieved success in some courses and failed others.

We will use concept names `K`, `E` and `R` for "Graduated at King's", at "Euphoria" and at "Rummidge", `C` for "Students with children" and `Pt` (`Ft`) for "part-time student" (resp. "full-time"), furthermore `GoodMarks` and `BadMarks`. We have a number-valued function `age` and a role "having-a-course" `hc`. The T–Box contains the following axioms: $\mathtt{K} \sqsubseteq \neg\mathtt{E} \sqcap \neg\mathtt{R}$ and $\mathtt{E} \sqsubseteq \neg\mathtt{R}$ and $\mathtt{Ft} \sqsubseteq \neg\mathtt{Pt}$, as well as $\mathtt{GoodMarks} \sqsubseteq \neg\ \mathtt{BadMarks}$.

Assume that A–Box classification of an A–Box with facts about past students gives the following temporary concepts as subconcepts for `S` and `F`.

$$
\begin{array}{lcl}
S_1^{\mathtt{S}} & \doteq & \mathtt{S} \sqcap \mathtt{K} \sqcap \mathtt{age=22} \sqcap \forall\ \mathtt{hc.GoodMarks} \\
S_2^{\mathtt{S}} & \doteq & \mathtt{S} \sqcap \mathtt{F} \sqcap \mathtt{E} \sqcap \mathtt{C} \sqcap \forall\ \mathtt{hc.BadMarks} \sqcap \exists\ \mathtt{hc}.\top \sqcap \mathtt{Pt} \\
S_1^{\mathtt{F}} & \doteq & \mathtt{S} \sqcap \mathtt{F} \sqcap \mathtt{E} \sqcap \mathtt{C} \sqcap \forall\ \mathtt{hc.BadMarks} \sqcap \exists\ \mathtt{hc}.\top \sqcap \mathtt{Pt} \\
S_2^{\mathtt{F}} & \doteq & \mathtt{F} \sqcap \mathtt{E} \sqcap \forall \mathtt{hc.BadMarks} \sqcap \exists\ \mathtt{hc}.\top \\
S_3^{\mathtt{F}} & \doteq & \mathtt{F} \sqcap \mathtt{R} \sqcap \mathtt{age=23} \sqcap \forall\ \mathtt{hc.GoodMarks} \sqcap \mathtt{Ft}
\end{array}
$$

To generalise `S` we calculate the direct subconcepts: $\underline{S^{\mathtt{S}}} = \{S_1^{\mathtt{S}}\}, \overline{S^{\mathtt{S}}} = \{S_1^{\mathtt{S}}, S_2^{\mathtt{S}}\}$. There are two optimal generalised decision concept:

$$\underline{G}(\mathtt{S}) = \{\ \mathtt{K},\ (\forall\mathtt{hc.GoodMarks} \sqcap \mathtt{age=22})\ \}$$

The academic should admit all students from King's and those students who are young and have good marks, since every student in the past was successful if he was an instance of $\underline{G}(\mathtt{S})$. Both GDCs should be tested by the academic for quality and usefullness, possibly adding new axioms $\mathtt{S} \sqsubseteq \mathtt{K}$ or $\mathtt{S} \sqsubseteq\ \forall\mathtt{hc.GoodMarks} \sqcap \mathtt{age=22}$ to the T–Box.

5 Conclusion

In the recent past, learning and knowledge acquisition in description logics have become very popular issues. There have been different approaches like *learning as search* over the space of possible concepts for completion of T–Boxes [Alv98], *learning through observation* [LL98] and *least common subsumer* (lcs) (for example [CH94, BK98]). Least common subsumers are the most common approach for learning in DL. lcs are appropriate for precise examples and data constructed for bottom-up construction of terminologies, because common properties of examples are deductively collected, but is unsuitable for rough and noisy data.

The approach closest to the one presented in this paper is the *constructive induction of structural knowledge* of Kietz and Morik [KM94] who present a system KLUSTER which calculates most general discriminations (MGD). MGDs are generalised decision concepts which are optimal for the particular description logic presented in this paper and which is more expressive than our logic $\mathcal{AL}^{\mathbb{R},=}$.

The novelty of our method is that we present a more general framework for the calculation of GDCs, which allows learning and mining of more expressive description logics. More investigations are needed to establish whether Kietz and Morik's method can be applied in our framework and how our approach can be generalised to build optimal GDCs for the description logic used by KLUSTER.

As far as $\mathcal{AL}^{\mathbb{R},=}$ is concerned there is a simple translation of the algorithms used for data mining with rough set theory. In the case of more expressive A–Box languages, these translations do not exist any more, and complexity becomes an issue. The calculation of all optimal GDCs is an NP-hard problem even for $\mathcal{AL}^{\mathbb{R},=}$. But this result corresponds to the complexity of data mining using rough sets and can be overcome using non-optimal generalisations, for example by stopping the algorithm when a *good* GDC is found. To solve these problems becomes crucial for more expressive description logics and involve both data mining (e.g. generic algorithms) as well as description logic specific techniques.

The Group of Logic and Computation at King's College London is currently implementing a hybrid system[3] which enables reasoning with an expressive T–Box and a database–like assertional component. The integration of a knowledge acquisition facility will be an integral part of the system and the quality of the methods presented here is to be assessed empirically. The tests will have two major objectives, first to establish the quality of the newly learned concepts and secondly to study the computational feasibility of possible extensions.

We have offered a method for combining knowledge representation and acquisition in a single description logic system. The main contribution is the definition of generalised decision concepts as a basis for the A–Box mining process and algorithms to calculate these GDCs. We claim that our methods will allow

[3]More information about the `Wellington` knowledge representation and reasoning system is available on: `http://www.dcs.kcl.ac.uk/research/groups/logic/wellington`.

the mining of more complex example data using the expressivity of description logic reasoning and will be playing an important role in the development of new mining techniques. From the description logic perspective we are convinced that incorporating data mining techniques will be a crucial addition to the standard DL machinery for the bottom-up development of useful real-life knowledge bases.

Acknowledgements. I would like to thank the anonymous referees for the very helpful comments. This research has been funded by the EPSRC.

References

[Alv98] J. Alvarez. Description logic system for learning in complex domains. In DL'98 [DL'98], pages 137–141.

[Baz98] J. Bazan. A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables. In Polkowski and Skowron [PS98], pages 322–365.

[BK98] F. Baader and R. Küsters. Least common subsumer computation w.r.t. cyclic $\mathcal{ALN}$-terminologies. In DL'98 [DL'98].

[CH94] W. Cohen and H. Hirsh. Learning the classic description logic: Theoretical and experimental results. In *KR-94*, pages 121–133, Bonn, Germany, 1994.

[DL'98] *1998 International Workshop on Description Logics (DL'98)*, 1998.

[KM94] J.-U. Kietz and K. Morik. A polynomial approach to the constructive induction of structural knowledge. *Machine Learning*, 14(2):193–218, 1994.

[LL98] P. Lambrix and P. Larocchia. Learning composite concepts. In DL'98 [DL'98], pages 147–152.

[Paw82] Z. Pawlak. Rough sets. *International Journal of Computer and Information Sciences*, 11(5):341–356, 1982.

[PS98] L. Polkowski and A. Skowron, editors. *Rough Sets in Knowledge Discovery*, volume 1. Physica-Verlag, 1998.

[Sch00] S. Schlobach. A–Box mining. Technical report, King's College London, 2000.

[Sko93] A. Skowron. Extracting laws from decision tables. In W. Ziarko, editor, *Proceedings of the Second International Workshop on Rough Sets and Knowledge Discovery (RSKD'93)*, pages 101–104, 1993.