Patent Valuation Using Difference in $\mathcal{ALEN}$

Naouel Karam and Adrian Paschke

Department of Mathematics and Computer Science, Freie Universität Berlin
Königin-Luise-Str. 24-26, 14195 Berlin, Germany
naouel.karam@fu-berlin.de,paschke@inf.fu-berlin.de

Abstract. In this paper we present an approach for patent claim comparison based on the difference operator in description logics. The claims are represented using $\mathcal{ALEN}$. An algorithm computing the difference between two $\mathcal{ALEN}$ concept descriptions is proposed and its usefulness for patent application valuation is described by means of some simple examples.

1 Introduction

More and more, patent material is made available in electronic format, and most of the time in textual form. Examples of such services are the European patent Office (EPO) [1] and the United States Patent and Trademark Office (USPTO) [3]. The need for a formal specification of the patent content arose recently to help retrieval, examination and classification of patent applications.

In this context one needs to evaluate the innovative degree of a patent application. A claim is the part of a patent application where the inventor specifies the invention attributes and its features, defining what can be protected by the patent law. The aim is to show the non-obviousness to a person with basic domain skills and the novelty of the application compared to the state of the art.

Patent claims are described in natural language. Terms used in the claims must find clear support in the preceding description part of the patent application such that the meaning of a term is clearly specified. Previous work had attempted to represent patent material using OWL ontologies [7, 15].

Statistical tools performing patent application evaluation are based on a set of parameters provided by the user. Based on those parameters, the overall score of a patent application is calculated.

In this paper we present a more formal approach for patent claim comparison based on description logics[1]. Patent application claims are formally represented as concept descriptions. When comparing two applications, the differences need to be pinpointed in order to prove the novelty. We compute this difference by using the difference operator and return the result to the user. The result must be intuitive and easy to understand in order to help the user construct his argumentation.

[1] This work has been partially supported by the Fact Screening and Transformation Project (FSTP) funded by the Teles Pri AG: www.fstp-expert-system.com

The difference operator computes the part of a concept that is not contained into another one. The difference operator has been defined in [14] as being the most general description that can be added to the second operand to obtain equivalence with the first. A second definition of the difference involving a syntactic criteria for minimality has been introduced in [4].

In our context we need to describe existential, universal and numerical restrictions. The difference operator has been investigated for description logics allowing either numerical restrictions [12] or existential restrictions [4]. We propose an algorithm for $\mathcal{ALEN}$ based on the structural characterization of subsumption for $\mathcal{ALEN}$ [10, 11].

The paper is organized as follows: in section 2, we give the formal definitions of the difference operator and discuss the motivation for our choice. Section 3 recalls useful results for $\mathcal{ALEN}$ and introduces some notations. In section 4 we provide an algorithm for computing the difference in $\mathcal{ALEN}$. Some examples of patent claims comparison are given in section 5 and Section 6 concludes the paper.

2 Operator Choice

The need for providing parts of a description that are not contained in another one has been motivated by different applications and different definitions exist with their advantages and drawbacks.

Informally speaking, the difference between two concept descriptions is the information contained in the first description and not in the second. The difference operator allows to remove from a given description all the information contained in another description. The difference operation between two concept descriptions was first introduced by Teege [14].

Definition 1. *(semantic difference) The difference between two concept descriptions C and D with $C \sqsubseteq D$ is given by*

$$C - D := max\{E \mid E \sqcap D \equiv C\}$$

where max is defined with respect to subsumption.

Later, the work in [9, 4] proposed a refinement of this definition by allowing the difference between incomparable descriptions (i.e. D is not required to subsume C) and taking the syntactic minimum (w.r.t. a subdescription ordering $\preceq_d$) instead of a semantic maximum.

Definition 2. *(syntactic difference) The difference between two incomparable concept descriptions C and D is defined as*

$$C - D := min\{E \mid E \sqcap D \equiv C \sqcap D\}$$

where *min* is defined with respect to a so-called subdescription ordering (denoted by $\preceq_d$), used to compare syntactical structures that we recall in the next definition.

Definition 3. *(Subdescriptions) Let C be an $\mathcal{ALEN}$-concept description, $\widehat{C}$ is a subdescription of C ($\widehat{C} \preceq_d C$) iff,*

1. $\widehat{C} = \bot$*, or*
2. $\widehat{C}$ *is obtained by removing from the top-level of C: $\top$, concept names, number restrictions, value restrictions, existential restrictions. For the remaining restrictions $\forall r.E$ or $\exists r.E$, replace the concept description E with a subdescription of E.*

As an example, consider the $\mathcal{ALEN}$-concept description C:

$$\mathsf{P} \sqcap \mathsf{P} \sqcap \forall \mathsf{r}.\mathsf{P} \sqcap \exists \mathsf{r}.(\mathsf{P} \sqcap \exists \mathsf{r}.\mathsf{Q}) \sqcap \leq 3r$$

A possible subdescription:

$$\widehat{C} = \mathsf{P} \sqcap \forall \mathsf{r}.\mathsf{P} \sqcap \exists \mathsf{r}.(\exists \mathsf{r}.\mathsf{Q}) \sqcap \leq 3r$$

is obtained from C by removing P from the top-level of C and P from the subexpression $\exists \mathsf{r}.(\mathsf{P} \sqcap \exists \mathsf{r}.\mathsf{Q})$. Note that $\widehat{C}$ is equivalent to C and there exists no subdescription of $\widehat{C}$ owning this property. $\widehat{C}$ is the minimal subdescription which is equivalent to C.

In [12], the authors defined a semantic difference for $\mathcal{ALN}$. They also compare the semantic and syntactic variants and state that the two approaches differ mainly on the bottom handling. Indeed, the bottom decomposition leads to more than one result for the semantic difference. On the other hand, the syntactic difference always generates a unique result that is a subdescription of the input description and hence more intuitive and comprehensive for the user.

Another operator called “concept abduction“ has been introduced to provide an explanation for matchmaking results in an e-marketplace. Abduction was first defined for $\mathcal{ALN}$ [5] and has been recently extended to a more expressive DL $\mathcal{SH}$, using tableau calculus [13].

This operator does not require subsumption between the concept descriptions neither and thus is more general than the semantic difference. As stated in [6], the result of the difference is included in the set of solutions of a CAP.

In our context, we chose to use the syntactic difference for three reasons:

1. It does not require subsumption between the concept descriptions being compared, which is more realistic in real world applications,
2. it generates a unique result which is a syntactic variation of the input and hence more intuitive for the user,
3. when the inputs are free from insatisfiable subexpressions the result is equivalent to the semantic difference.

3 Preliminaries

In this section we recall some results for subsumption in $\mathcal{ALEN}$ and define some notations that will be useful to derive the difference algorithm.

The normal form of an $\mathcal{ALEN}$ concept description is obtained by applying the following transformation rules:

- $(\geq mr) \sqcap (\geq nr) \rightarrow (\geq nr)$ if $n \geq m$,
- $(\leq mr) \sqcap (\leq nr) \rightarrow (\leq nr)$ if $n \leq m$, and
- $\forall r.C \sqcap \forall r.D \rightarrow \forall r.(C \sqcap D)$.

To access the different component of a concept description, we will use the following notations:

- $\mathsf{prim}(C)$ denotes the set of concept names and the bottom concept occurring on the top-level of C,
- $\mathsf{inf}_r(C) = (\leq nr)$ if there exists a number restriction of the form $(\leq nr)$ on the top-level of C, $\mathsf{inf}_r(C) = \top$ otherwise,
- $\mathsf{sup}_r(C) = (\geq nr)$ if there exists a number restriction of the form $(\geq nr)$ on the top-level of C, $\mathsf{sup}_r(C) = \top$ otherwise,
- $\mathsf{min}_r(C) = max\{k \mid C \sqsubseteq (\geq kr)\}$,
- $\mathsf{max}_r(C) = min\{k \mid C \sqsubseteq (\leq kr)\}$,
- $\forall_r(C) = E$ if there exists a value restriction $\forall r.E$ on the top-level of C; $\forall_r(C) = \top$ otherwise,
- $\exists_r(C) = \{C' \mid \exists r.C' \text{occurs on the top-level of } C\}$.

For the sake of clarity, we will assume that the set N_R is reduced to the role r. The algorithm can be easily generalized to arbitrary sets of role names.

The difference algorithm is based on the structural characterization of subsumption in $\mathcal{ALEN}$ given in [10]. In the sequel we present the intuitions used to compute induced concept descriptions and recall the structural subsumption characterization. We also introduce some notations that we will use in the difference algorithm.

The main issue when dealing with $\mathcal{ALEN}$ is to compute the "non-trivial" concept descriptions subsuming a concept description C; such concepts do not appear in the top-level of C and are called "induced". Let us recall this process by means of examples.

Number restrictions:

They can be induced if two existential restrictions involve disjoint concepts. For example, if $C := \exists r.P \sqcap \exists r.\neg P$ then $\geq 2r$ is induced by C.

Let us note the induced number restrictions as follows: $\mathsf{min}^*_r(C)$. Note that if $\mathsf{min}^*_r(C) = k$, we have $C \sqsubseteq \geq kr$ which means that $\mathsf{min}^*_r(C)$ is taken into account in $\mathsf{min}_r(C)$ defined above.

Existential restrictions:

Due to $\leq$-number restrictions, if the number of existential restrictions in the top-level of a concept C is greater than the $\leq$-number restriction, those existential restrictions must be merged.

For example, if we have:

$$C := \exists r.(P \sqcap A) \sqcap \exists r.(P \sqcap B) \sqcap \exists r.(\neg P \sqcap A) \sqcap \exists r.Q \sqcap \exists r.\neg Q \sqcap \leq 2r,$$

the merging process results in new concepts descriptions where the only consistent ones are:

$$\exists r.(P \sqcap Q \sqcap A \sqcap B) \sqcap \exists r.(\neg P \sqcap \neg Q \sqcap A) \leq 2r, \text{and}$$
$$\exists r.(P \sqcap \neg Q \sqcap A \sqcap B) \sqcap \exists r.(\neg P \sqcap Q \sqcap A) \leq 2r.$$

In [10], each merging is represented by a mapping α leading to a concept C^{α} and the set of mappings is denoted by $\Gamma_r(C)$. The mappings in $\Gamma_r(C)$ are required to obey the following conditions:

1. $\alpha(i) \neq \emptyset$ for all $1 \leq i \leq n$;
2. $\bigcup_{1 \leq i \leq n} \alpha(i) = \{1, ..., m\}$ and $\alpha(i) \cap \alpha(j) = \emptyset$ for all $1 \leq i < j \leq n$;
3. $\bigsqcap_{j \in \alpha(i)} C_j \sqcap \forall_r(C) \not\equiv \bot$ for all $1 \leq i \leq n$,

where $n := min\{\mathsf{max}_r(C), \mid \exists_r(C) \mid\}$ and $m :=\mid \exists_r(C) \mid$.

In our example, this set consists of two mappings α_1 and α_2 leading to the two concepts C^{α_1} and C^{α_2} above with $C \equiv C^{\alpha_1} \sqcup C^{\alpha_2}$.
The set of C' such that $\exists r.C'$ in C^{α} for all mappings is noted as follows:

$$\exists_r^*(C) := \bigcup_{\alpha \in \Gamma_r(C)} \exists_r(C^{\alpha})$$

For our example we obtain:

$$\exists_r^*(C) := \{P \sqcap Q \sqcap A \sqcap B, \neg P \sqcap \neg Q \sqcap A, P \sqcap \neg Q \sqcap A \sqcap B, \neg P \sqcap Q \sqcap A\}$$

In case $\exists_r(C) = \emptyset$ (there is no existential restriction on the top level of C) one needs to take into account the $\geq$-restrictions together with value restrictions to deduce non-trivial existential restrictions. Let us illustrate the case on the following concept:

$$D := (\geq 2r) \sqcap \forall r.(A \sqcap B).$$

The instances of D have at least two r-successors and due to the value restriction we can deduce that $D \sqsubseteq \exists r.(A \sqcap B)$.

This concludes the existential restrictions case.

Value restrictions:
It is clear from the former example that every instance of C has exactly two r-successors, in that case one can deduce from the existential restrictions in C^{α_1} and C^{α_2} that all r-successors belong to A and thus that the value restriction $\forall r.A$ is induced by C.

This deduction can be performed only when all r-successors are known and it is represented by the condition:

$$\mathsf{max}_r(C) = \mathsf{min}_r^*(C).$$

The second case where a value restriction can be induced is when $\mathsf{max}_r(C) = 0$ then $C \sqsubseteq \forall r.\bot$.

4 Syntactic Difference between $\mathcal{ALEN}$-Concept Descriptions

The algorithm computing the difference between two $\mathcal{ALEN}$-concept descriptions C and D is depicted in figure 1. We extended the algorithm proposed in [9] to compute the difference between two $\mathcal{ALE}$-concept descriptions.

Before going into detail about the algorithm, and as it is based on the structural characterization of the subsumption [10], let us recall the conditions for each kind of conjunct occurring on the top level of the concept descriptions.

Theorem 1. *Let C, D be two $\mathcal{ALEN}$-concept descriptions with $\exists_r(C) = \{C_1, ..., C_n\}$. Then $C \sqsubseteq D$ iff $C \equiv \bot$, or $D \equiv \top$, or the following conditions hold:*

1. *$prim(C) \subseteq prim(D)$,*
2. *$max_r(C) \leq max_r(D)$,*
3. *$min_r(C) \geq min_r(D)$,*
4. *for all $D' \in \exists_r(D)$ it holds that*
   (a) *$\exists_r(C) = \emptyset$, $min_r(C) \geq 1$, and $\forall_r(C) \sqsubseteq D'$; or*
   (b) *$\exists_r(C) \neq \emptyset$, and for each $\alpha \in \Gamma_r(C)$, there exists $C' \in \exists_r(C^\alpha)$ such that $C' \sqcap \forall_r(C) \sqsubseteq D'$.*
5. *if $\forall_r(C) \not\equiv \top$, then*
   (a) *$max_r(C) = 0$; or*
   (b) *$min_r(C) < max_r(C)$ and $\forall_r(C) \sqsubseteq \forall_r(D)$; or*
   (c) *$0 < min_r(C) = max_r(C)$ and $\forall_r(C) \sqcap C' \sqsubseteq \forall_r(D)$ for all $C' \in \exists^*_r(C)$.*

According to the definition of the difference (c.f Definition 2), the output of the algorithm must verify:

$$\mathsf{ComputeDiff}(C, D) \sqcap D \equiv C \sqcap D$$

Let start with concept names and numerical restrictions. One can easily verify conditions 1.−3. of Theorem 1:

- The set of primitive concepts $\mathsf{prim}(\mathsf{ComputeDiff}(C, D) \sqcap D)$ is equal to $\mathsf{prim}(\mathsf{ComputeDiff}(C, D)) \cup \mathsf{prim}(D)$, which by definition is equal to $(\mathsf{prim}(C) \backslash \mathsf{prim}(D)) \cup \mathsf{prim}(D)$. This is again equal to $\mathsf{prim}(C) \cup \mathsf{prim}(D)$, the set of primitive concepts of $(C \sqcap D)$.

- For $\mathsf{max}_r(\mathsf{ComputeDiff}(C, D) \sqcap D)$, we distinguish two cases: it is equal to $\mathsf{max}_r(D)$, when $\mathsf{max}_r(C) \geq \mathsf{max}_r(D)$ and to $\mathsf{max}_r(\mathsf{inf}_r(C) \sqcap D)$, when $\mathsf{max}_r(C) < \mathsf{max}_r(D)$. Both cases are equal to $\mathsf{max}_r(C \sqcap D)$.

- Analogously to the precedent case, for $\mathsf{min}_r(\mathsf{ComputeDiff}(C, D) \sqcap D)$ we have two cases: equal to $\mathsf{min}_r(D)$ when $\mathsf{min}_r(C) \leq \mathsf{min}_r(D)$ and to $\mathsf{min}_r(\mathsf{sup}_r(C) \sqcap D)$ when $\mathsf{min}_r(C) > \mathsf{min}_r(D)$. Which again are equal to $\mathsf{min}_r(C \sqcap D)$.

```
Require: Two $\mathcal{ALEN}$-concept descriptions $C$ and $D$ in $\mathcal{ALEN}$-normal form
Ensure: ComputeDiff$(C, D)$
if $C \sqcap D \equiv \bot$ then
    ComputeDiff$(C, D) := \bot$
else
    ComputeDiff$(C, D) := \sqcap_{A \in \mathsf{prim}(C-D)} A \sqcap \mathsf{inf}_r(C-D) \sqcap \mathsf{sup}_r(C-D) \sqcap \forall r.\mathsf{ComputeDiff}(\forall_r(C), \forall_r(D) \sqcap \forall^*_r(C \sqcap D)) \sqcap \bigsqcap_{E \in \mathcal{E}'_r} \exists r.E$,
    where the value restriction is omitted in case $\mathsf{ComputeDiff}(\forall_r(C), \forall_r(D) \sqcap \forall^*_r(C \sqcap D)) \equiv \top$ and:

    • $\mathsf{prim}(C\text{-}D) := \mathsf{prim}(C) \backslash \mathsf{prim}(D)$;
    • $\mathsf{inf}_r(C\text{-}D) = \begin{cases} \top, & \mathsf{max}_r(C) \geq \mathsf{max}_r(D), \\ \mathsf{inf}_r(C), & \mathsf{max}_r(C) < \mathsf{max}_r(D); \end{cases}$
    • $\mathsf{sup}_r(C\text{-}D) = \begin{cases} \top, & \mathsf{min}_r(C) \leq \mathsf{min}_r(D), \\ \mathsf{sup}_r(C), & \mathsf{min}_r(C) > \mathsf{min}_r(D); \end{cases}$
    • $\forall^*_r(E) = \begin{cases} \top, & \mathsf{min}^*_r(E) < \mathsf{max}_r(E), \\ \bot, & \mathsf{max}_r(E) = 0, \\ lcs(\{\forall_r(E) \sqcap E' \mid E' \in \exists^*_r(E)\}), & 0 < \mathsf{min}^*_r(E) = \mathsf{max}_r(E); \end{cases}$
    • $\mathcal{E}'_r$ is computed as follows:

    Let $\mathcal{E}_r := \exists_r(C) = \{C_1, ..., C_n\}$ and we define $C^{\backslash E}$ as being $C$ without the conjunct $\exists r.E$.
    for $i = 1$ to $n$ do
        if (i) $\Gamma_r(C^{\backslash C_i} \sqcap D) \neq \emptyset$ and there exists $D' \in \exists_r((C^{\backslash C_i} \sqcap D)^\alpha)$ forall $\alpha \in \Gamma_r(C^{\backslash C_i} \sqcap D)$ with $\forall_r(C) \sqcap \forall_r(D) \sqcap D' \sqsubseteq C_i$, or
        (ii) $\Gamma_r(C^{\backslash C_i} \sqcap D) = \emptyset$ and $\mathsf{min}_r(C \sqcap D) \geq 1$ and $\forall_r(C) \sqcap \forall_r(D) \sqsubseteq C_i$,
        then
            $\mathcal{E}_r := \mathcal{E}_r \backslash \{C_i\}$;
        end if
    end for
    $\mathcal{E}'_r = \{E^* \mid E \in \mathcal{E}_r\}$ where $E^* := \mathsf{ComputeDiff}(E, \forall_r(C) \sqcap \forall_r(D))$.
end if
```

Fig. 1. The algorithm ComputeDiff

Let us now consider the value restrictions. In case $0 < \mathsf{min}^*_r(C \sqcap D) = \mathsf{max}_r(C \sqcap D)$, a value restriction $\forall_r(C \sqcap D)$ can be deduced from $C \sqcap D$, which is the least common subsumer of all the existential restrictions appearing in $\exists^*_r(C \sqcap D)$.

Let us illustrate this case by an example.

$$C_{ex1} := \exists r.(P \sqcap A) \sqcap \exists r.(P \sqcap B) \sqcap \exists r.\neg Q \sqcap \forall r.(A \sqcap B),$$
$$D_{ex1} := \exists r.(\neg P \sqcap A) \sqcap \exists r.Q \sqcap (\leq 2r).$$

As demonstrated in section 3, the merging of existential restrictions in $C_{ex1} \sqcap D_{ex1}$ induces the value restriction $\forall r.A$. The value restriction of the difference $C-D$ is then

$$\mathsf{ComputeDiff}(\forall_r(C), \forall_r(D) \sqcap \forall_r^*(C \sqcap D)) = \mathsf{ComputeDiff}(A \sqcap B, A) = B$$

One can verify condition 5(c) of Theorem 1, namely that $B \sqcap C' \sqsubseteq A \sqcap B$ for all C' in the set of merged existential restrictions $\exists_r^*(C_{ex1} \sqcap D_{ex1})$ (c.f. the example in Section 3).

Let us now illustrate the two cases for existential restrictions by means of examples.

- To illustrate case (i) let us take the following descriptions:

  $$\begin{aligned}&C_{ex2} := \exists r.(P \sqcap A \sqcap B) \sqcap \leq 2r,\\ &D_{ex2} := \exists r.(P \sqcap A) \sqcap \exists r.(P \sqcap B) \sqcap \exists r.(\neg P \sqcap A) \sqcap \exists r.Q \sqcap \exists r.\neg Q.\end{aligned}$$

  $C_1 = P \sqcap A \sqcap B$ subsumes $P \sqcap Q \sqcap A \sqcap B$ from $(C_{ex2}^{\setminus C_1} \sqcap D_{ex2})^{\alpha 1}$ and $P \sqcap \neg Q \sqcap A \sqcap B$ from $(C_{ex2}^{\setminus C_1} \sqcap D_{ex2})^{\alpha 2}$. We then have

  $$\mathsf{ComputeDiff}(C_{ex2}, D_{ex2}) = \leq 2r.$$

- Finally we illustrate case (ii) with the following example

  $$\begin{aligned}&C_{ex3} := P \sqcap \exists r.(A_1 \sqcap A_2) \sqcap (\geq 3r),\\ &D_{ex3} := \forall r.(A_1 \sqcap A_2 \sqcap A_3).\end{aligned}$$

  $$\mathsf{ComputeDiff}(C_{ex3}, D_{ex3}) = P \sqcap (\geq 3r)$$

  While $\exists r.(A_1 \sqcap A_2 \sqcap A_3)$ can be induced from $\forall r.(A_1 \sqcap A_2 \sqcap A_3) \sqcap (\geq 3r)$, one can verify that

  $$\begin{aligned}&\mathsf{ComputeDiff}(C_{ex3}, D_{ex3}) \sqcap D_{ex3} \equiv P \sqcap (\geq 3r) \sqcap \forall r.(A_1 \sqcap A_2 \sqcap A_3)\\ &\sqsubseteq P \sqcap (\geq 3r) \sqcap \exists r.(A_1 \sqcap A_2 \sqcap A_3) \sqcap \forall r.(A_1 \sqcap A_2 \sqcap A_3)\\ &\sqsubseteq P \sqcap (\geq 3r) \sqcap \exists r.(A_1 \sqcap A_2) \sqcap \forall r.(A_1 \sqcap A_2 \sqcap A_3)\\ &\equiv C_{ex3} \sqcap D_{ex3}\end{aligned}$$

  The case $\sqsupseteq$ is obvious.

The next lemma proves that the algorithm returns a difference that respects the condition of the difference operator (Section 2). The proof can be found in our technical report [8].

Lemma 1. *Let C, D be two $\mathcal{ALEN}$-concept descriptions in $\mathcal{ALEN}$ normal form. Then, ComputeDiff$(C, D) \sqcap D \equiv C \sqcap D$.*

5 Application to Patent Claims Comparison

In this section we are going to illustrate the difference on some simple patent examples.

Example 1. Suppose we have an application for a chair with only one leg having a seat made only from a light material. The corresponding concept description is the following:

$$= 1 hasLeg \sqcap \exists hasSeat.(\forall hasMaterial.Light),$$

that we need to compare to a previous patent describing a chair with three legs and having one seat made of light wood, given by the description

$$= 3 hasLeg \sqcap \exists hasSeat.(\forall hasMaterial.(Wood \sqcap Light)).$$

The algorithm returns $\leq 1 hasLeg$ which corresponds to the specific part of the application.

Example 2. Let us consider a patent application for a watch with two types of displays or more that are all bright. The corresponding description is:

$$>= 2 hasDisplay \sqcap \forall hasDisplay.Bright,$$

to compare to a watch with an analogue and a non analogue display described as:

$$\exists hasDisplay.Analogical \sqcap \exists hasDisplay.\neg Analogical.$$

The difference is $\forall hasDisplay.Bright$. Indeed, we can deduce the numerical restriction $>= 2 hasDisplay$ from the second concept description while the existential restrictions involve disjoint concepts.

6 Conclusion

In this paper, we have investigated the problem of computing the syntactic difference in $\mathcal{ALEN}$. This work was motivated by an application in the context of patent applications valuation. In this context one needs to compare the claims of the patent with previous patents solving a similar problem. The textual descriptions of the claims can be described in a formal way using $\mathcal{ALEN}$-concept descriptions. The difference operator provides the user with the parts that are contained in his application and not in the state of the art. Our aim is not to provide a decision-making tool regarding the novelty of a patent application, but rather a tool that can help in pinpointing the differences and assist the user in constructing his argumentation.

We are implementing a prototype of the difference algorithm based on the DL reasoner HermiT [2] for the subsumption test. A direction for Future work would be to investigate the decision making process based on the results returned by the difference and empirical rules derived from experts decisions.

References

1. The European Patent Office website: http://www.epo.org
2. The Hermit OWL Reasoner website: http://hermit-reasoner.com
3. The United States Patent and Trademark Office website: http://www.uspto.gov/
4. Brandt, S., Kusters, R., Turhan, A.Y.: Approximation and difference in description logics. In: T.Dean (ed.) Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR2002). pp. 203–214. Morgan Kaufmann, San Francisco, CA (2002)
5. Colucci, S., Noia, T.D., Sciascio, E.D., Donini, F.M., Mongiello, M.: Concept abduction and contraction in description logics. In: Description Logics (2003)
6. Di Noia, T., Di Sciascio, E., Donini, F.M.: Semantic matchmaking as non-monotonic reasoning: a description logic approach. J. Artif. Int. Res. 29(1), 269–307 (2007), http://dl.acm.org/citation.cfm?id=1622606.1622615
7. Giereth, M., Koch, S., Kompatsiaris, Y., Papadopoulos, S., Pianta, E., Serafini, L., Wanner, L.: A modular framework for ontology-based representation of patent information. In: JURIX (2007)
8. Karam, N., Paschke, A.: Patent valuation using difference in ALEN. Tech. rep., Freie Universitat Berlin, Germany (2012), http://www.csw.inf.fu-berlin.de/publications/TR-DiffAlen.pdf
9. Kusters, R.: Non-standard inferences in description logics, vol. 2100 of Lecture Notes in Artificial Intelligence. Springer-Verlag (2001)
10. Kusters, R., Molitor, R.: Computing least common subsumers in ALEN. In: Proceedings of the 17th international joint conference on Artificial intelligence - Volume 1. pp. 219–224. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001), http://portal.acm.org/citation.cfm?id=1642090.1642120
11. Kusters, R., Molitor, R.: Structural subsumption and least common subsumers in a description logic with existential and number restrictions. Studia Logica 81(2), 227–259 (2005)
12. Leger, A., Rey, C., Toumani, F.: Semantic difference in ALN. In: Description Logics (2007)
13. Noia, T.D., Sciascio, E.D., Donini, F.M.: A tableaux-based calculus for abduction in expressive description logics: preliminary results. In: Description Logics (2009)
14. Teege, G.: Making the difference: a subtraction operation for description logics. In: Doyle, J., Sandewall, E., Torasso, P. (eds.) KR'94: Principles of Knowledge Representation and Reasoning, pp. 540–550. Morgan Kaufmann, San Francisco, California (1994)
15. Wanner, L., Baeza-Yates, R., Brügmann, S., Codina, J., Diallo, B., Escorsa, E., Giereth, M., Kompatsiaris, Y., Papadopoulos, S., Pianta, E., Piella, G., Puhlmann, I., Rao, G., Rotard, M., Schoester, P., Serafini, L., Zervaki, V.: Towards content-oriented patent document processing. World Patent Information 30(1), 21–33 (2008)