Computing Probabilistic Least Common Subsumers in Description Logics

Thomas Mantay, Ralf Möller, and Alissa Kaplunova

Artificial Intelligence Lab, Department of Computer Science, University of Hamburg

Abstract. Computing least common subsumers in description logics is an important reasoning service useful for a number of applications. As shown in the literature, it can, for instance, be used for similarity-based information retrieval where information retrieval is performed on the basis of the similarities of user-specified examples. In this article, we first show that, for crisp DLs, in certain cases the set of retrieved information items can be too large. Then we propose a probabilistic least common subsumer operation based on a probabilistic extension of the description logic language $\mathcal{ALN}$. We show that by this operator the amount of retrieved data can be reduced avoiding information flood.

1 Introduction

Knowledge representation languages based on description logics (DLs) have proven to be a useful means for representing the terminological knowledge of an application domain in a structured and formally well understood way. In DLs, knowledge bases are formed out of *concepts* representing sets of individuals. Using the concept constructors provided by the DL language, complex concepts are built out of atomic concepts and atomic roles. Roles represent binary relations between individuals. For example, in the context of a TV information system, the set of all football broadcasts can be described with a concept term using the atomic concepts teamsports-broadcast and football and the atomic role has-sports-tool: teamsports-broadcast $\sqcap \forall$ has-sports-tool.football.

A central feature of knowledge representation systems based on DLs is a set of reasoning services with the possibility to deduce implicit knowledge from explicitly represented knowledge. For instance, the subsumption relation between two concepts can be determined. Intuitively speaking, a concept C *subsumes* a concept D if the set of individuals represented by C is a superset of the set of individuals represented by D, i.e., if C is more general than D. Furthermore, *retrieval* describes the problem of determining all individuals which are instances of a given concept.

As another reasoning service, the *least common subsumer* (LCS) operation, applied to concepts $C_1, \ldots, C_m$, computes the most specific concept which subsumes $C_1, \ldots, C_m$. The LCS operation is an important and non-trivial reasoning service useful for a number of applications. In [2], an LCS operator is considered for the DL $\mathcal{ALN}$ including feature chains in order to approximate a disjunction operation which is not explicitly included in $\mathcal{ALN}$. In addition, the operator is used as a subtask for the "bottom-up" construction of knowledge bases based on the DLs $\mathcal{ALN}$ with cyclic concept definitions [1]. See also [3] for a similar application concerning the constructive induction of a P-Classic KB from data.

In our applications, the LCS operation is used as a subtask for similarity-based information retrieval [5]. The goal is to provide a user of an information system with an example-based query mechanism. The data of an information system are modeled as DL individuals. For instance, a specific TV broadcast about football could be modeled as an instance of the concept for football broadcasts introduced above. The "commonalities" of the selected examples of interest to the user are formalized by a DL concept of which (i) the user-selected examples are instances and (ii) which is the most specific concept (w.r.t. subsumption) with property (i). A concept fulfilling properties (i) and (ii) will then be used as a retrieval filter. The task of similarity-based information retrieval can be split into three subtasks: First, the most specific concepts of a finite set of individuals are computed yielding a finite set of concepts. Then, the LCS of these concepts is computed. Finally, by determining its instances the LCS concept is used as a retrieval concept. For the purpose of similarity-based information retrieval, the first task is fulfilled by the well-known realization inference service. The third subtask, determining the instances of the LCS concept, is accomplished by the instance retrieval inference service of the knowledge representation system.

In certain cases, computing the LCS of concepts yields a very general concept. As a consequence, a large set of information items are retrieved resulting in an information flood if all items are displayed at once. Thus, at least a ranking is needed or we have to define a new operator for computing the commonalities between concepts. In this paper, we pursue the second approach and define an LCS operator that takes additional domain knowledge into account.

The main contribution of this paper is the proposal of a probabilistic LCS operation for a probabilistic extension of the DL $\mathcal{ALN}$ which has been introduced in [4] for the knowledge representation system P-Classic. The probabilistic LCS operator makes use of P-Classic's ability to model the degree of overlap between concepts. With the probabilistic LCS operator we investigate an example-based retrieval approach in which well known information retrieval techniques are integrated with formally well investigated inference services of DLs.

2 Preliminaries

In this section, we introduce syntax and semantics of the underlying knowledge representation language $\mathcal{ALN}$ and give formal definitions of relevant terms.

Definition 1 (Syntax). *Let $\mathcal{C}$ be a set of* atomic concepts *and $\mathcal{R}$ a set of* atomic roles *disjoint from $\mathcal{C}$. $\mathcal{ALN}$ concepts are recursively defined as follows:*

- *The symbols $\top$ and $\bot$ are $\mathcal{ALN}$ concepts (top concept, bottom concept).*
- *A and $\neg A$ are $\mathcal{ALN}$ concepts for each $A \in \mathcal{C}$ (atomic concept, negated atomic concept).*
- *Let C and D be $\mathcal{ALN}$ concepts, $R \in \mathcal{R}$ an atomic role, and $n \in I\!N \cup \{0\}$. Then $C \sqcap D$ (concept conjunction), $\forall\, R.C$ (universal role quantification), $(\geq\ n\, R)$ ($\geq$-restriction), and $(\leq\ n\, R)$ ($\leq$-restriction) are also concepts.*

We set $(=\ n\, R)$ as an abbreviation for $(\geq\ n\, R) \sqcap (\leq\ n\, R)$. The semantics of concepts is given in terms of an interpretation.

Definition 2 (Interpretation, Model, Consistency). *An interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *of an* $\mathcal{ALN}$ *concept consists of a non-empty set* $\Delta^{\mathcal{I}}$ *(the* domain *of* $\mathcal{I}$*) and an* interpretation function $\cdot^{\mathcal{I}}$*. The interpretation function maps every atomic concept* A *to a subset* $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ *and every role* R *to a subset* $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$*. The interpretation function is recursively extended to complex* $\mathcal{ALN}$ *concepts as follows. Assume that* $A^{\mathcal{I}}, C^{\mathcal{I}}, D^{\mathcal{I}}$*, and* $R^{\mathcal{I}}$ *are already given and* $n \in I\!N \cup \{0\}$*. Then*

- $\top^{\mathcal{I}} := \Delta^{\mathcal{I}}$, $\bot^{\mathcal{I}} := \emptyset$, $(\neg A)^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$, $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
- $\forall\, R.C^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} | \forall d' : (d, d') \in R^{\mathcal{I}} \Rightarrow d' \in C^{\mathcal{I}}\}$,
- $(\geq\ n\ R)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} | \sharp\{d' | (d, d') \in R^{\mathcal{I}}\} \geq n\}$, *and*
- $(\leq\ n\ R)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} | \sharp\{d' | (d, d') \in R^{\mathcal{I}}\} \leq n\}$.

An interpretation $\mathcal{I}$ *is a* model *of an* $\mathcal{ALN}$ *concept* C *iff* $C^{\mathcal{I}} \neq \emptyset$*.* C *is called* consistent *iff* C *has a model.*

Note that both constructors $\top$ and $\bot$ are expressible by $(\geq\ 0\, R)$ and $A \sqcap \neg A$, respectively.

Definition 3 (Subsumption, Equivalence, Instance). *A concept* C is subsumed by *a concept* D $(C \sqsubseteq D)$ *iff* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ *for all interpretations* $\mathcal{I}$*. Two concepts* C *and* D *are* equivalent *iff* $C^{\mathcal{I}} = D^{\mathcal{I}}$ *holds for all interpretations* $\mathcal{I}$*. The interpretation function* $\cdot^{\mathcal{I}}$ *is extended to individuals by mapping them to elements of* $\Delta^{\mathcal{I}}$ *such that* $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ *if* $a \neq b$*. An individual* $d \in \Delta^{\mathcal{I}}$ *is an* instance *of a concept* C *iff* $d^{\mathcal{I}} \in C^{\mathcal{I}}$ *holds for all interpretations* $\mathcal{I}$*.*

Definition 4 (Depth). *The* depth *of a concept is recursively defined as follows:*

- *If* $C = A$, $C = \neg A$, $C = (\geq\ n\ R)$, *or* $(\leq\ n\ R)$, *then* $depth(C) := 0$.
- *If* $C = \forall\, R.C'$, *then* $depth(C) := 1 + depth(C')$.

Note that, in contrast to usual definitions of the concept depth, we define the depth of number restrictions as 0.

Definition 5 (Canonical form). *Let* $C_1, \ldots, C_m$ *be concepts and* $\{R_1, \ldots, R_M\}$ *the set of all roles occuring in* $C_1, \ldots, C_m$*. Then* C_i *is in* canonical form *iff*

$$C_i = \alpha_{i1} \sqcap \cdots \sqcap \alpha_{in_i} \sqcap \beta_{iR_1} \sqcap \cdots \sqcap \beta_{iR_{j_i}}$$

where $j_i \in \{0, \ldots, M\}$*,* α_{ik} *is an atomic concept or negated atomic concept with no atomic concept appearing more than once and* $\beta_{iR_j} = (\geq\ l_{iR_j}\ R_j) \sqcap (\leq\ m_{iR_j}\ R_j) \sqcap \forall\, R_j.C'_{iR_j}$ *with* C'_{iR_j} *also being in canonical form.*

It is easy to see that any concept can be transformed into an equivalent concept in canonical form in linear time.

Definition 6 (LCS). *Let* $C_1, \ldots, C_m$ *be concepts. Then we define the set of* least common subsumers *(LCSs) of* $C_1, \ldots, C_m$ *as*

$$lcs(C_1, \ldots, C_m) := \{E | C_1 \sqsubseteq E \wedge \cdots \wedge C_m \sqsubseteq E \wedge \forall E' : C_1 \sqsubseteq E' \wedge \cdots \wedge C_m \sqsubseteq E' \Rightarrow E \sqsubseteq E'\}.$$

In [2], it is shown that, for the DL $\mathcal{ALN}$, all elements of $lcs(C_1, \ldots, C_m)$ are equivalent. Therefore, we will consider $lcs(C_1, \ldots, C_m)$ as a single concept instead of a set of concepts.

The following example shows that the concept computed by the LCS is sometimes too general and, thus, might not always be a useful retrieval concept. Let sports-broadcast (SB), team-sports-broadcast (TSB), individual-sports-broadcast (ISB), basketball (B), football (FB), and tennis-racket (TR) be atomic concepts, has-sports-tool an atomic role, and

$$\begin{aligned}
\text{basketball-broadcast (BB)} := {} & \text{team-sports-broadcast} \sqcap (= \ 1 \ \text{has-sports-tool}) \sqcap \\
& \forall \, \text{has-sports-tool.basketball}, \\
\text{football-broadcast (FB)} := {} & \text{team-sports-broadcast} \sqcap (= \ 1 \ \text{has-sports-tool}) \sqcap \\
& \forall \, \text{has-sports-tool.football}, \ \text{and} \\
\text{tennis-broadcast (TB)} := {} & \text{individual-sports-broadcast} \sqcap (= \ 1 \ \text{has-sports-tool}) \sqcap \\
& \forall \, \text{has-sports-tool.tennis-racket}
\end{aligned}$$

be concepts. Subsequently, we will use the concept abbreviations given in brackets. Let us consider a user interested in TV broadcasts similar to FB and BB. Then, computing the LCS of FB and BB would result in a useful retrieval concept: $\mathsf{TSB} \sqcap (= 1 \, \text{has-sports-tool}) \sqcap \forall \, \text{has-sports-tool.ST}$. However, let us consider a user whose interests are expressed by FB and TB. The LCS computation then yields the retrieval concept $A := \mathsf{SB} \sqcap (= 1 \text{has-sports-tool}) \sqcap \forall \text{has-sports-tool.ST}$ denoting the set of *all* sports broadcasts with a sports tool. Since A is a very general concept, using A as a retrieval concept would result in a large amount of TV broadcasts, which might not be acceptable on the part of the user. A more suitable result would be to allow for $B := \mathsf{TSB} \sqcap (= 1 \text{has-sports-tool}) \sqcap \forall \text{has-sports-tool.ST}$ and $C := \mathsf{ISB} \sqcap (= 1 \, \text{has-sports-tool}) \sqcap \forall \, \text{has-sports-tool.ST}$ as alternative retrieval concepts. This is plausible because in Davis Cup matches, for instance, teams of tennis players compete against each other. Hence, in our intuition, there is a non-empty overlap between the concepts TSB and ISB which cannot be adequately quantified in $\mathcal{ALN}$. In order to model the degree of overlap between concepts by probabilities, the knowledge representation system P-Classic was introduced in [4]. The DL underlying P-Classic is a probabilistic extension of $\mathcal{ALN}$ augmented by functional roles (attributes).

One of the goals of P-Classic is to compute probabilistic subsumption relationships of the form $P(D|C)$ denoting the probability of an individual to be an instance of D given that it is an instance of C. In case $C \equiv \top$, we write $P(D)$. In order to fully describe a concept, its atomic concept components and the properties of number restrictions and universal role quantifications need to be described. Therefore, a set $\mathcal{P}$ of probabilistic classes (p-classes) is introduced describing a probability distribution over the properties of individuals conditioned on the knowledge that the individuals occur on the right-hand side of a role. Each p-class is represented by a Bayesian network and one of the p-classes $P^* \in \mathcal{P}$ is the root p-class. The root p-class describes the distribution over all individuals and all other p-classes describe the distribution over role successors assuming independence between distinct individuals. The Bayesian networks are

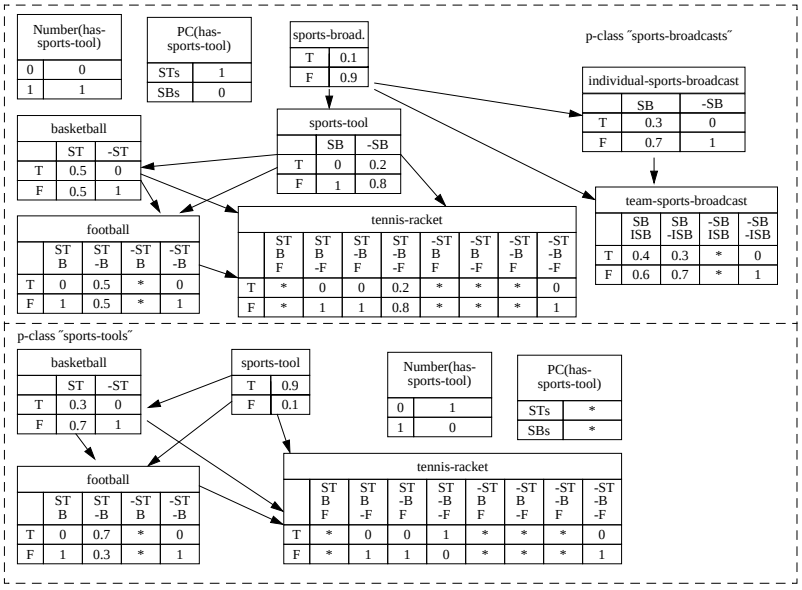

Fig. 1. P-Classic KB about sports broadcasts.

modeled as DAGs whose nodes represent atomic concepts, number restrictions $[Number(R)]$, and the p-class from which role successors are drawn $[PC(R)]$. In addition to P-Classic, we introduce extra nodes for negations of atomic concepts. Dependencies are used to model conditional probabilities and are modeled by edges in the network. For instance, for an individual, we can state the probability of this individual to be an instance of ISB under the condition that it is an instance of SB. The range of the variables of a node representing an atomic or negated atomic concept can be either *true* or *false* and for $Number(R)$ it is a subset of $\mathbb{N}$. In order to guarantee termination of the inference algorithm for computing $P(D|C)$, this subset must be finite. Thus, the number of role successors for a role is bounded. The function $bound(R)$ indicates the maximum number of role successors for R. The range of a $PC(R)$ node is the set of p-classes $\mathcal{P}$ indicating the p-classes the R-successors are drawn from. The reason for introducing special nodes for negations of atomic concepts is that this extension enables us to evaluate expressions of the form $P(A \sqcap \neg A)$ as 0 which will be a necessary property subsequently. In order to demonstrate the advantages of the probabilistic LCS operator, we will now create a P-Classic KB with overlapping concepts. Figure 1 shows a knowledge base about sports broadcasts enriched by probability information. For instance, it is stated that a broadcast is considered to be about team-sports (TSB) with probability 0.3 given that it is a broadcast about sports (SB) but no individual-sports ($\mathsf{\text{-}ISB}$). Two p-classes are represented. The concept $\mathsf{sports\text{-}broadcasts}$ is the root p-class and the role successors for the role $\mathsf{has\text{-}sports\text{-}tool}$ are drawn from the p-class $\mathsf{sports\text{-}tools}$. For each concept C, the probability $P_{P^*}(C)$ with which an individual is an instance of C can then be computed by a standard inference algorithm for Bayesian networks. For example, the probability of $P_{P*}(\mathsf{TSB} \sqcap (= 1\, \mathsf{has\text{-}sports\text{-}tool}) \sqcap \forall\, \mathsf{has\text{-}sports\text{-}tool.B})$ is computed

by setting the nodes for TSB and B to *true*, $Number(\mathsf{has\text{-}sports\text{-}tool}) = 1$, and $PC(\mathsf{has\text{-}sports\text{-}tool}) = \mathsf{STs}$. By Bayesian network propagation we yield a value of 0.015. With the formalism for computing expressions of the form $P(D|C)$ it is possible to express the degree of overlap between C and D by a probability.

Based on the probabilistic description logic summarized in this section, it is possible to define a probabilistic LCS operator which takes into account the degree of overlap between concepts.

3 A Probabilistic Extension of the LCS Operator

Intuitively, given concepts $C_1, \ldots, C_m$, the key idea is to allow those concepts for candidates of a probabilistic least common subsumer (PLCS) of $C_1, \ldots, C_m$ which have a non-empty overlap with $C_1, \ldots, C_m$. In order to keep the set of candidates finite, we consider only concepts whose depth is not larger than $\max\{depth(C_i)|i \in \{1, \ldots, m\}\}$. From the viewpoint of information retrieval this is no severe restriction, since in practical applications deeply nested concepts usually do not have any relevant individuals as instances (e.g., the concept $\mathsf{FB} \sqcap \forall \mathsf{has\text{-}sports\text{-}tool}.\forall \mathsf{has\text{-}sports\text{-}tool}.\mathsf{F}$ in our example).

Definition 7. *Let $C_1, \ldots, C_m$ be $\mathcal{ALN}$ concepts and P^* the root p-class of a* P-Classic *KB. Then we define the set of* PLCS concept candidates *of $C_1, \ldots, C_m$ as*

$$Can(C_1, \ldots, C_m) := \{E | P_{P^*}(E \sqcap C_1) > 0 \wedge \cdots \wedge P_{P^*}(E \sqcap C_m) > 0 \wedge depth(E) \leq \max\{depth(C_i)|i = 1, \ldots, m\}\}.$$

Definition 7 induces the following observation.

Proposition 1. *Let $C_1, \ldots, C_m$ be $\mathcal{ALN}$ concepts. Then, in the worst case, the cardinality of $Can(C_1, \ldots, C_m)$ is exponential in m.*

Proof. Given a P-Classic KB in which $C_1, \ldots, C_m$ are all atomic concepts with $\forall i, j \in \{1, \ldots, m\} : P(C_i \sqcap C_j) > 0$, we can bound $\sharp Can(C_1, \ldots, C_m)$ by the exponential function 2^m. □

In the next step, we want to measure the effectiveness of using a certain PLCS candidate for retrieval. It will be helpful to be able to express the probability of an individual to be an instance of a concept disjunction. Since this language operator is not contained in $\mathcal{ALN}$, we use the following definition which is essentially taken from [6].

Definition 8. *Let $C_1, \ldots, C_m$ be $\mathcal{ALN}$ concepts. Then we define*

$$P(C_1 \sqcup \cdots \sqcup C_m) := (-1)^2 \sum_{k=1,\ldots,m} P(C_k) \; + \; (-1)^3 \sum_{k_1 < k_2} P(C_{k_1} \sqcap C_{k_2}) \; + \; (-1)^4 \sum_{k_1 < k_2 < k_3} P(C_{k_1} \sqcap C_{k_2} \sqcap C_{k_3}) + \; \ldots \; + (-1)^{m+1} P(C_1 \sqcap \cdots \sqcap C_m).$$

It should be noted that by Definition 8 we do not extend the syntax of the underlying DL.

Proposition 2. *Let $C_1, \ldots, C_m$ be concepts. Then computing $P(C_1 \sqcup \cdots \sqcup C_m)$ is exponential in m.*

The proof is obvious and is omitted here.

In many retrieval environments, it is customary to introduce two real numbers: recall and precision. Both values indicate the quality of a concept E to function as an appropriate PLCS. By these measures the qualities of potential PLCSs can be compared to one another. The comparison will be formalized by the notion of dominance between triples $(E, r_{E,C_1,\ldots,C_m}, p_{E,C_1,\ldots,C_m})$ and $(E', r_{E',C_1,\ldots,C_m}, p_{E',C_1,\ldots,C_m})$.

Definition 9 (Recall). *Let E and $C_1, \ldots, C_m$ be $\mathcal{ALN}$ concepts. Then we define E's* recall *of $C_1, \ldots, C_m$ as*

$$r_{E,C_1,\ldots,C_m} := P(C_1 \sqcup \cdots \sqcup C_m | E) = \frac{P(E \sqcap (C_1 \sqcup \cdots \sqcup C_m))}{P(E)}.$$

According to this definition, the larger the recall measure of a concept E, the more specific it is w.r.t. probabilistic subsumption of $C_1, \ldots, C_m$. For a concept E, a perfect recall is yielded iff $r_{E,C_1,\ldots,C_m} = 1$. For example, if E is a PLCS candidate and A an atomic concept such that $A \sqsubseteq E$, then $r_{E,A,\neg A} = 1$. Unlike in the definition of the (crisp) LCS, a concept expression does not necessarily need to subsume $C_1, \ldots, C_m$ (completely) in order to be a PLCS candidate. This motivates the introduction of the precision measure.

Definition 10 (Precision). *Let E and $C_1, \ldots, C_m$ be $\mathcal{ALN}$ concepts. Then we define E's* precision *of $C_1, \ldots, C_m$ as*

$$p_{E,C_1,\ldots,C_m} := P(E | C_1 \sqcup \cdots \sqcup C_m) = \frac{P(E \sqcap (C_1 \sqcup \cdots \sqcup C_m))}{P(C_1 \sqcup \cdots \sqcup C_m)}.$$

The precision measures the probability with which a randomly chosen individual, which is an instance of any of the C_i, $i \in \{1, \ldots, m\}$, is also an instance of the PLCS candidate E. As a consequence of Definition 10, if $E = lcs(C_1, \ldots, C_m)$, we have $q_{E,C_1,\ldots,C_m} = 1$.

Figure 2 illustrates the meaning of both measures given four concepts represented as areas in the 2D space. The recall of E_1, $r_{E_1,C,D}$, corresponds to the ratio of the size of the hatched area and the size of E_1. E_1's precision, $p_{E_1,C,D}$, is the ratio of the size of E_1 and the size of the union of E_1, C, and D. Given the appropriate values for E_2 we see that $r_{E_2,C,D}$ is smaller than $r_{E_1,C,D}$ but $p_{E_2,C,D}$ is larger than $p_{E_1,C,D}$.

Proposition 3. *Let E and $C_1, \ldots, C_m$ be $\mathcal{ALN}$ concepts. Then, computing $r_{E,C_1,\ldots,C_m}$ and $p_{E,C_1,\ldots,C_m}$ takes time exponential in the length of $E, C_1, \ldots, C_m$.*

Proof. Since $P(E \sqcap (C_1 \sqcup \cdots \sqcup C_m)) = P(E) - (P(E \sqcup C_1 \sqcup \cdots \sqcup C_m) - P(C_1 \sqcup \cdots \sqcup C_m))$, the claim follows from Proposition 2. □

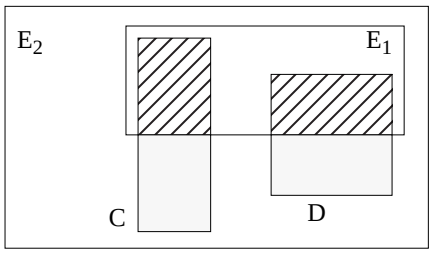

Fig. 2. Scenario of four concepts illustrating the meaning of "recall" and "precision".

With the above considerations, we will define the set of PLCSs of concepts $C_1, \ldots, C_m$ as a set of triples where the first component is a concept $E \in Can(C_1, \ldots, C_m)$ and the other components are E's recall and precision. In a concrete application, a user should be able to specify minimum values for at least one of the measures that he is willing to accept. For example, he could specify a recall of 0.8 preventing him from obtaining too general PLCS concepts and, thus, restricting the amount of retrieved data.

With the notion of dominance between candidates we can define the set of probabilistic least common subsumers.

Definition 11 (Dominance). *Let E, E' and $C_1, \ldots, C_m$ be $\mathcal{ALN}$ concepts. Then $(E, r_{E,C_1,\ldots,C_m}, p_{E,C_1,\ldots,C_m})$* dominates *$(E', r_{E',C_1,\ldots,C_m}, p_{E',C_1,\ldots,C_m})$ iff $r_{E,C_1,\ldots,C_m} > r_{E',C_1,\ldots,C_m} \wedge p_{E,C_1,\ldots,C_m} > p_{E',C_1,\ldots,C_m}$.*

Definition 12 (Set of Probabilistic Least Common Subsumers). *Let $C_1, \ldots, C_m$ be $\mathcal{ALN}$ concepts. Then we define the set of* probabilistic least common subsumers *of $C_1, \ldots, C_m$ as*

$$\begin{aligned} \textit{p-lcs}(C_1, \ldots, C_m) := \{&(E, r_{E,C_1,\ldots,C_m}, p_{E,C_1,\ldots,C_m}) \in Can(C_1, \ldots, C_m) \times I\!R \times I\!R \mid \\ &\neg\exists (E', r_{E',C_1,\ldots,C_m}, p_{E',C_1,\ldots,C_m}) : \\ &(E', r_{E',C_1,\ldots,C_m}, p_{E',C_1,\ldots,C_m}) \textit{ dominates} \\ &(E, r_{E,C_1,\ldots,C_m}, p_{E,C_1,\ldots,C_m})\}. \end{aligned}$$

p-lcs$(C_1, \ldots, C_m)$ is called minimal *iff $\forall (E, r_{E,C_1,\ldots,C_m}, p_{E,C_1,\ldots,C_m}), (E', r_{E',C_1,\ldots,C_m}, p_{E',C_1,\ldots,C_m}) \in$ p-lcs$(C_1, \ldots, C_m) : E \not\equiv E'$.*

In Definition 12 we formalize the ideas of Fig. 2 conditioned on the general case of m concepts. When defining *p-lcs*$(C_1, \ldots, C_m)$ we consider only concepts with a non-empty overlap with each of the $C_1, \ldots, C_m$. We only accept triples with the best quality measures and, therefore, accept only dominating triples in *p-lcs*$(C_1, \ldots, C_m)$. From this definition we can derive the following statement.

Proposition 4. *The set p-lcs$(C_1, \ldots, C_m)$ has the following properties:*

(i) p-lcs$(C_1, \ldots, C_m)$ is finite.
(ii) Minimality: $(E, r_{E,C_1,\ldots,C_m}, p_{E,C_1,\ldots,C_m}) \in$ p-lcs$(C_1, \ldots, C_m) \Longrightarrow$
$\forall i \in \{1, \ldots, m\} : P(E \sqcap C_i) > 0 \wedge \neg\exists (E', r_{E',C_1,\ldots,C_m}, p_{E',C_1,\ldots,C_m}) :$
$r_{E',C_1,\ldots,C_m} > r_{E,C_1,\ldots,C_m} \wedge p_{E',C_1,\ldots,C_m} > p_{E',C_1,\ldots,C_m} \wedge depth(E') \leq depth(E)$.

Proof. (i) is obvious since the maximum depth of the concepts in $p\text{-}lcs(C_1, \ldots, C_m)$ is limited by the maximum depth of the $C_1, \ldots, C_m$ and the number of concept components of $C_1, \ldots, C_m$ is finite ensuring the number of PLCS candidates to be finite. Hence, $p\text{-}lcs(C_1, \ldots, C_m)$ is finite as well. For $(E, r_{E,C_1,\ldots,C_m}, p_{E,C_1,\ldots,C_m}) \in p\text{-}lcs(C_1, \ldots, C_m)$, the fact that $P(E \sqcap C_i) > 0$, for all $i \in \{1, \ldots, m\}$, follows immediately by the definition of $Can(C_1, \ldots, C_m)$. $\neg\exists(E', r_{E',C_1,\ldots,C_m}, p_{E',C_1,\ldots,C_m}) : r_{E',C_1,\ldots,C_m} > r_{E,C_1,\ldots,C_m} \wedge p_{E',C_1,\ldots,C_m} > p_{E,C_1,\ldots,C_m} \wedge depth(E') \leq depth(E)$ also follows since $p\text{-}lcs(C_1, \ldots, C_m)$ contains only dominating triples $(E, r_{E,C_1,\ldots,C_m}, p_{E,C_1,\ldots,C_m})$ with $depth(E) \leq \max\{depth(C_i) | i \in \{1, \ldots, m\}\}$. □

The minimal set $p\text{-}lcs(C_1, \ldots, C_m)$ can be computed in three steps: First, we must find the set of concepts which have a non-empty overlap with each of the $C_1, \ldots, C_m$. Proposition 4 (i) states a necessary criterion for a corresponding algorithm to terminate since the set of concepts E which have a non-empty overlap with each of the C_i is finite. Then, for each concept E in this set, we have to compute the parameters $r_{E,C_1,\ldots,C_m}$ and $p_{E,C_1,\ldots,C_m}$ and then build the set of dominant triples $p\text{-}lcs(C_1, \ldots, C_m)$. Proposition 4 (ii) guarantees that there is no relevant retrieval concept with better recall *and* precision than the corresponding measures of the triples in $p\text{-}lcs(C_1, \ldots, C_m)$. Finally, we must determine the minimal set $p\text{-}lcs(C_1, \ldots, C_m)$. This can be done by successively eliminating a triple $(E, r_{E,C_1,\ldots,C_m}, p_{E,C_1,\ldots,C_m})$ from $p\text{-}lcs(C_1, \ldots, C_m)$ as long as the following condition holds:

$$\forall(E, r_{E,C_1,\ldots,C_m}, p_{E,C_1,\ldots,C_m}) \in p\text{-}lcs(C_1, \ldots, C_m) :$$
$$\neg\exists(E', r_{E',C_1,\ldots,C_m}, p_{E',C_1,\ldots,C_m}) \in p\text{-}lcs(C_1, \ldots, C_m) \text{ with } E \equiv E'.$$

The necessary equivalence test can be performed by structural comparisons since the involved concepts are in normal form. In general, a minimal $p\text{-}lcs(C_1, \ldots, C_m)$ is not unique since there is no rule stating which triple to eliminate in case two triples with equivalent concepts are present. However, in our similarity-based information retrieval application this is no problem because the sets of instances of equivalent concepts are equal.

Algorithm 1 computes the set of PLCS candidates given concepts $C_1, \ldots, C_m$ and the KB as a Bayesian network BN. In the first step, all atomic concepts and negated atomic concepts in the Bayesian network are collected in the set X_1 if there is a non-empty overlap with each of the $C_1, \ldots, C_m$. Computing the concept candidates for our example, *compute-concept-candidate*(FB, TB), we get $X_1 = \{\mathsf{SB}, \mathsf{TSB}, \mathsf{ISB}\}$. Secondly, we build the set of all conjunctions of concepts of X_1 which have a non-empty overlapping with each of the $C_1, \ldots, C_m$ including the ones consisting of only one conjunct. In our case, we yield $X_2 = \{\mathsf{SB}, \mathsf{TSB}, \mathsf{ISB}, \mathsf{SB} \sqcap \mathsf{TSB}, \mathsf{SB} \sqcap \mathsf{ISB}, \mathsf{ISB} \sqcap \mathsf{TSB}, \mathsf{SB} \sqcap \mathsf{TSB} \sqcap \mathsf{ISB}\}$. In the next part of the algorithm, we collect all number restrictions having a non-empty overlap with each of the $C_1, \ldots, C_m$ in the set X_3. Since the maximum number of role successors is bounded, we can guarantee finiteness of X_3. Let X be an abbreviation for $(= 1\,\mathsf{has\text{-}sports\text{-}tool})$ and Y an abbreviation for $(\geq 0\,\mathsf{has\text{-}sports\text{-}tool}) \sqcap (\leq 1\,\mathsf{has\text{-}sports\text{-}tool})$. Then, in our example, we have $X_3 = \{X, Y\}$. Subsequently,

Algorithm 1 compute-concept-candidates$(C_1, \ldots, C_m, BN)$

$X_1 := \emptyset, X_2 := \emptyset, X_3 := \emptyset, X_4 := \emptyset, X_5 := \emptyset, X_6 := \emptyset, X_7 := \emptyset$
for all nodes in BN representing an atomic or negated atomic concept A **do**
  add A to X_1 if $\forall i \in \{1, \ldots, m\} : P_{PCL_i}(A \sqcap C_i) > 0$
end for
for all $K \in 2^{X_1}$ **do**
  add $E := \sqcap_{D \in K} D$ to X_2 if $\forall i \in \{1, \ldots, m\} : P_{PCL_i}(E \sqcap C_i) > 0$
end for
for $i \in \{1, \ldots, M\}$ and subexpressions of the form $(\geq\ i\, R) \sqcap (\leq\ j\, R)$ in $C_1, \ldots, C_m$ **do**
  add $(\geq\ k\, R) \sqcap (\leq\ l\, R)$ to X_3 if $0 \leq k \leq j \wedge \mathit{bound}(R) \geq l \geq i \wedge k \leq l$
end for
for $i \in \{1, \ldots, M\}$ and subexpressions of the form $\forall R_i.C_1^{'}, \ldots, \forall R_i.C_m^{'}$ in $C_1, \ldots, C_m$ **do**
  add the concepts resulting from the invocation
  compute-concept-candidates$(C_1^{'}, \ldots, C_m^{'}, BN)$ to X_4
end for
$X_5 := \{C \sqcap \forall R_i.D | i \in \{1, \ldots, M\}$ and $C \in X_3$ and $D \in X_4$ and C refers to role $R_i\}$
for all $K \in 2^{X_5}$ **do**
  add $E := \sqcap_{D \in K} D$ to X_6 if $\forall i \in \{1, \ldots, m\} : P_{PCL_i}(E \sqcap C_i) > 0$
end for
$X_7 := X_2 \cup X_6 \cup \{C \sqcap D | C \in X_2 \wedge D \in X_6 \wedge \forall i \in \{1, \ldots, m\} : P_{PCL_i}(C \sqcap D \sqcap C_i) > 0\}$
return X_7

for all roles R_i and all $\forall$-quantifications occurring in $C_1, \ldots, C_m$ and involving R_i, we add those concepts to X_4 which have a non-empty overlap with each of the R_i quantifiers ($C_1^{'}, \ldots, C_m^{'}$ in the algorithm). In our example, we compute $X_4 := \{\mathsf{ST}\}$. Now, in X_5 we collect all conjunctions of number restrictions from X_3 involving role R and $\forall\, R.D$ where D is a concept overlapping with R's quantifiers $C_1^{'}, \ldots, C_m^{'}$. Let X' be an abbreviation for $X \sqcap \forall\, \mathsf{has\text{-}sports\text{-}tool.ST}$ and Y' an abbreviation for $Y \sqcap \forall\, \mathsf{has\text{-}sports\text{-}tool.ST}$. Then, in our example, we have $X_5 = \{X', Y'\}$. In X_6, we collect the conjunctions of elements of X_5 over all occurring roles if a conjunction has a non-empty overlap with each of the $C_1, \ldots, C_m$. Since, in our example, we have only one role, we get $X_6 = X_5$. Finally, we combine the results in X_2 (conjunctions of atomic and negated atomic concepts) and the ones in X_6 (conjunctions of number restrictions and $\forall$-quantifications) into X_7 which is returned by the algorithm. In our example, X_7 consists of 21 concepts from which we will only list the ones which are unique w.r.t. to equivalence: $\{E_1, \ldots, E_{12}\} = \{\mathsf{SB}, \mathsf{TSB}, \mathsf{ISB}, \mathsf{TSB} \sqcap \mathsf{ISB}, \mathsf{SB} \sqcap X', \mathsf{TSB} \sqcap X', \mathsf{ISB} \sqcap X', \mathsf{TSB} \sqcap \mathsf{ISB} \sqcap X', \mathsf{SB} \sqcap Y', \mathsf{TSB} \sqcap Y', \mathsf{ISB} \sqcap Y', \mathsf{TSB} \sqcap \mathsf{ISB} \sqcap Y'\}$ as desired.

Theorem 1. *For concepts $C_1, \ldots, C_m$ and a Bayesian network BN representing a* P-Classic *KB, algorithm* compute-concept-candidates *returns the set $Can(C_1, \ldots, C_m)$.*

Proof. We give only a sketch of the proof. Algorithm *compute-concept-candidates* terminates because the maximum number of iterations is bounded by the maximum depth of $C_1, \ldots, C_m$. It is sound since every output concept has a non-empty overlap with $C_1, \ldots, C_m$. It is also complete because the algorithm recursively checks all possible concepts resulting from the concept-forming operators of Definition 1 for a non-empty overlap with $C_1, \ldots, C_m$. □

The set of concept candidates computed by Algorithm 1 can easily be transformed into a set in which all pairs of concepts are not equivalent. Therefore, later no additional algorithm for transforming $p\text{-}lcs(C_1, \ldots, C_m)$ into a minimal $p\text{-}lcs(C_1, \ldots, C_m)$ will be necessary. Now recall and precision must be determined for each candidate by means of the formulae given in Definitions 9 and 10. This can be done straightforwardly by algorithms taking concepts E and $C_1, \ldots, C_m$ as input parameters and returning $r_{E,C_1,\ldots,C_m}$ and $p_{E,C_1,\ldots,C_m}$, respectively. The set $p\text{-}lcs(C_1, \ldots, C_m)$ contains only those triples whose quality measures dominate those of other triples.

Algorithm 2 compute-minimal-plcs$((E_1, r_1, p_1), \ldots, (E_n, r_n, p_n))$

$p\text{-}lcs(C_1, \ldots, C_m) := sort(((E_1, r_1, p_1), \ldots, (E_n, r_n, p_n)), p_i)$
for $i = 1$ to n **do**
  eliminate all (E', r', p') from $p\text{-}lcs(C_1, \ldots, C_m)$ with $r' < r_i$ and $p' < p_i$
end for.

Algorithm 2 computes the largest subset of dominant triples of $\{(E_1, r_{E_1,C_1,\ldots,C_m}, p_{E_1,C_1,\ldots,C_m}), \ldots, (E_n, r_{E_n,C_1,\ldots,C_m}, p_{E_n,C_1,\ldots,C_m})\}$. In the example, we get $p\text{-}lcs(\mathsf{FB}, \mathsf{TB}) = \{(\mathsf{SB} \sqcap X', 0.22, 1), (\mathsf{SB} \sqcap Y', 0.22, 1), (\mathsf{TSB} \sqcap X', 0.24, 0.354), (\mathsf{TSB} \sqcap Y', 0.24, 0.354), (\mathsf{ISB} \sqcap X', 0.26, 0.345), (\mathsf{ISB} \sqcap Y', 0.26, 0.345)\}$. As a result we get six possible retrieval concepts. $\mathsf{SB} \sqcap X'$ is the (crisp) LCS of FB and TB. Naturally, this concept has a precision of 1.0 since, according to Definition 6, $lcs(\mathsf{FB}, \mathsf{TB})$ is a concept which (completely) subsumes FB and TB. Alternatively, the result suggests the use of $\mathsf{TSB} \sqcap X'$ or $\mathsf{ISB} \sqcap X'$ as retrieval concepts. Both concepts have a better recall measure, and using them for retrieval results in a smaller set of information items. On the other hand, $\mathsf{TSB} \sqcap X'$ and $\mathsf{ISB} \sqcap X'$ have a worse precision measure than $\mathsf{SB} \sqcap X'$. Hence, the probability of meeting an individual which does not incorporate the commonalities represented by the concepts FB and TB is higher. The three concepts involving Y' have the same quality measures than the ones involving X'. The reason is that from our P-Classic KB it follows that $P(Number(\mathsf{has\text{-}sports\text{-}tool}) = 0) = 1.0$, i.e, we do not need to consider them.

Theorem 2. *Let $C_1, \ldots, C_m$ be $\mathcal{ALN}$ concepts. Then, in the worst case, computing $p\text{-}lcs(C_1, \ldots, C_m)$ takes time exponential in m.*

Proof. This result follows from Proposition 1 since computing the set of PLCS candidates of $C_1, \ldots, C_m$ is a subtask of computing $p\text{-}lcs(C_1, \ldots, C_m)$. □

Propositions 2, 1, and 3 show the sources of complexity for the presented inference task. Due to the subterms $P(C_1 \sqcup \cdots \sqcup C_m)$ and $P(E \sqcap (C_1 \sqcup \cdots \sqcup C_m))$

occuring in Definitions 9 and 10, the computation of the precision and the recall measure take time exponential in the number of m. Also the computation of the set of PLCS candidates takes time exponential in the number of concepts. In practice, however, the exponential behavior of the computation comes into effect only for knowledge bases with many overlapping concepts. Thus, when building a KB, the number of concept overlaps should be kept small.

4 Conclusion

In this article, we contributed to the problem of similarity-based information retrieval on the basis of the DL $\mathcal{ALN}$. It is shown that in certain cases the computation of commonalities with the (crisp) LCS operation yields too general retrieval concepts which can result in an information flood in a retrieval context. In order to circumvent this problem, we introduced a probabilistic LCS for a probabilistic extension of the DL $\mathcal{ALN}$. It is proved that the retrieval concepts provided by this operation are in some sense optimal and can be used as an alternative to retrieval concepts computed by a crisp LCS operation. By demonstrating the performances of the PLCS operator with an example we showed that meaningful retrieval results can be achieved with this operator. In the retrieval approach we integrated known information retrieval techniques with formally investigated inference services of DLs. Further research can be done on extending the expressivity of the underlying DL—especially integrating a disjunction operator. It is not clear if the disjunction $C \sqcup D$ of two concepts C and D should also belong to the set of PLCS candidates since it is questionable if it sufficiently represents the commonalities of C and D. Another problem is that the number of PLCS candidates will dramatically increase in the presence of an or-operator.

References

1. F. Baader and R. Küsters. Computing the Least Common Subsumer and the Most Specific Concept in the Presence of Cyclic $\mathcal{ALN}$-concept Descriptions. In O. Herzog and A. Günter, editors, *Proc. of the 22nd KI-98*, volume 1504, pages 129–140, 1998.
2. W. W. Cohen, A. Borgida, and H. Hirsh. Computing Least Common Subsumers in Description Logics. In *Proceedings of the National Conference on Artificial Intelligence AAAI'92*, pages 754–760. AAAI Press/The MIT Press, 1992.
3. J.U. Kietz and K. Morik. A polynomial approach to the constructive induction of structural knowledge. *Machine Learning*, 14:193–217, 1994.
4. D. Koller, A. Levy, and A. Pfeffer. P-Classic: A tractable probabilistic description logic. In *Proc. of AAAI 97*, pages 390–397, Providence, Rhode Island, 1997.
5. R. Möller, V. Haarslev, and B. Neumann. Semantics-based Information Retrieval. In *Int. Conf. on Inf. Techn. and Knowl. Systems*, Vienna, Budapest, 1998.
6. V. K. Rohatgi. *An Introduction to Probability Theory and Mathematical Statistics*. Wiley Series in Probability and Mathematical Statistics, 1976.