1

Content-Based Information Retrieval by Computation of Least Common Subsumers in a Probabilistic Description Logic

Thomas Mantay, Ralf Möller
Artificial Intelligence Lab, Department of Computer Science, University of Hamburg
{mantay|moeller}@informatik.uni-hamburg.de

ABSTRACT Due to the constantly growing number of information sources, intelligent information retrieval becomes a more and more important task. We model information sources by description logic (DL) terminologies. The commonalities of user-specified examples can be computed by the least common subsumer (LCS) operator. However, in some cases this operator delivers too general results. In this article we solve this problem by presenting a probabilistic extension of the LCS operator for a probabilistic description logic. By computing gradual commonalities between description logic concepts, this operator serves as a crucial means for content-based information retrieval for all kinds of information sources. We also describe an extension of our operator to consider unwanted information. The probabilistic LCS can be applied for information retrieval in a scenario of multiple information sources.

1.1 Introduction

The number of structured but heterogeneous information sources that are available online is growing rapidly. In particular, many sources in the World-Wide Web offer information about all kinds of themes. Often the user must manually combine information items from multiple sources. If information is distributed in different semi-structured formats (see the XML discussion), automatic integration techniques are required to provide adequate information systems. Basically, the same situation occurs in standard database contexts, and thus, many of the well-known integration techniques can be reused in the Web context (see, e.g., [CL93]). As a remedy to the integration and combination problems, the "information agent" abstraction has been proposed (e.g., SIMS [AKS96]). Information agents are understood as systems that provide a uniform query interface to multiple information sources.

In the Web context, most users of information systems are only casual users. Hence, they are often overtaxed when asked to (formally) describe the exact kind of information they desire. In many applications they can, however, supply examples concerning the information of interest. In contrast to approaches where the user has to learn query languages (or agent communication languages), in this paper we focus on providing the theoretical background for information retrieval on the basis of *user-specified examples* which express his information demands. An information system can automatically determine a description of the user's information demands by evaluating the *commonalities* between these examples. The information source(s) can then be queried for corresponding data objects that "match" the description(s). In the approach presented in this article, we do not pursue the standard case-based information retrieval approaches where abstract statistical distance measures are computed

to associate examples and information "items" in various sources. Instead, we explicitly represent the commonalities based on formal logical models for background knowledge. Thus, commonalities can be reasoned about and, for instance, the background knowledge can be accessed to infer implicit information etc.

As an example domain we consider a scenario where a user of a so-called "TV-Assistant" can retrieve broadcasts "similar" to example broadcasts he specified beforehand. The idea is to let the user build a (structured) collection of films of interest to him. Referring to (a subset of) the collection a user can ask the TV-Assistant to retrieve similar films for Saturday evening, say. In a more general perspective, the TV-Assistant can be viewed as an instance of an application where queries can be posed which are based on previously collected related information items. In this *content-based* information retrieval approach, information sources are modeled using description logic theory. DL systems offer a number of reasoning services which are declaratively specified with reference to a formal model-theoretic semantics. For example, they determine whether a "data description" (called concept, see below) is consistent, they determine whether two concepts are disjoint, whether one concept is subsumed by another one (automatic concept classification), etc. In addition, concepts can be used to specify queries (instance classification).

The use of description logics for information access is described in various publications. In a similar way as [MMB97] we use the description logic system CLASSIC (see also [LRO96, LP97, LSW98]). For (Basic-)CLASSIC Cohen et al. [CBH92] have defined an operation for computing the *least common subsumer* (LCS) of concepts. With this operation and an operation to extract a conceptual description (i.e., a concept) from an item of a collection, the commonalities of a collection of items can be computed and, in turn, explicitly be represented as a concept. In our TV-Assistant scenario, the notion of similarity between two (or more) TV broadcasts is formalized by computing the LCS between the "parent concepts" of the corresponding individual broadcasts. The LCS concept is then used as a query. The result will be the set of broadcast individuals which are subsumed by the LCS concept. This kind of case-based information retrieval based on DLs is investigated in [MHN98] and also considered in [SV97].

Although the "LCS idea" works in principle, sometimes it is very hard to provide deep domain models such that the LCS operator returns concepts that really describe the commonalities. In other words, sometimes the commonalities will be just "thing" due to necessarily incomplete models and due to the sharp semantics behind the logical modeling techniques. For instance, broadcasts about football do not have much in common with tennis broadcasts. Hence, computing the LCS between the corresponding concepts would result in a rather general concept (e.g., the concept for sports broadcasts). Once this concept is used as a query, a probably too large set of sports broadcasts might be returned. A more suitable result would be to allow the concept representing all broadcasts where teams are involved as an LCS concept. The definition should regard that the concept team-sports-broadcast should be considered an LCS of football-broadcast and tennis-broadcast. This is a plausible model because in Davis Cup matches, for instance, teams of tennis players compete against each other. However, since there are also a lot of TV broadcasts from ATP tournaments where no teams are involved, team-sports-broadcast does not "completely" subsume both football-broadcast and tennis-broadcast but only with some probability smaller than 1. To come back to the information retrieval aspect, a user might want to be presented team sports broadcasts in this case rather than a (possibly large) set of all sports broadcasts which the crisp version of the LCS operator would suggest. In other words, we would like to be able to express that a certain concept is an *LCS only with a certain probability*, since not every tennis match

involves teams.

One of the main problems is that the *degree of overlap* between concepts cannot be adequately modeled in crisp DLs. Therefore, several probabilistic extensions of DLs have been suggested. In [Jae94] cross entropy minimization is used to combine the modeling of statements about conditional probabilities between concepts and statements expressing uncertain knowledge about specific objects. [Hei94] represents knowledge on the basis of probability intervals and thus enables the modeling of ignorance. In addition, Fuzzy approaches for modeling vague and imprecise knowledge have been developed [Str98, TM98]. The underlying crisp DL is $\mathcal{ALC}$. A probabilistic extension of CLASSIC based on Bayes nets is introduced in [KLP97] (the language is called P-CLASSIC).

Since the LCS operation is defined only for (Basic-)CLASSIC and not for $\mathcal{ALC}$ we investigated the use of P-CLASSIC from [KLP97] for modeling vague knowledge in the TV-Assistant domain. Whereas in crisp CLASSIC the determination of commonalities of concepts can be performed by computing the LCS operator, such an operation is not yet available for P-CLASSIC. We propose an approach to combine the two notions of the LCS operation and the P-CLASSIC description logic.

In Section 1.2 we give a short introduction to the underlying syntax and semantics of P-CLASSIC and sketch an application of this logic to the problems in the TV-Assistant domain. In Section 1.3 we define the probabilistic least common subsumer operator for a set of concepts and present an algorithm to compute it. In addition, measures to quantify the *suitability* or appropriateness of an LCS concept are defined. Section 1.4 shows how the definition can be extended to consider negative examples, i.e., explicit information items representing the kind of information a user is *not* interested in. We conclude with a discussion on possible optimizations of the presented algorithms.

1.2 The Underlying Description Logic

We assume three disjoint alphabets of symbols, called *atomic concepts*, *atomic roles*, and abstract *individuals*. The special concept name $\top$ is called *top*. Concepts names denote sets of domain objects. Roles denote tuples of domain objects. For a given domain object, the objects related to it by a role are referred to as its *fillers*. Roles taking only one role filler are called *attributes* (or *features*). Complex *concepts* are recursively defined as follows. Let P be an atomic concept, R a role, F an attribute, f an abstract individual, and $n \in \mathbb{N}$. If C and D are concepts already defined, then $P, \neg P, C \sqcap D, \forall R.C, (\geq\ n\ R), (\leq\ n\ R)$, and $(\mathsf{fills}\ F\ f)$ are also concepts. Furthermore, let $(=\ n\ R)$ be an abbreviation for $(\geq\ n\ R) \sqcap (\leq\ n\ R)$. The semantics for concepts is given by an *interpretation* $\mathcal{I}$ which is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a non-empty set $\Delta^{\mathcal{I}}$ (the *domain*) and an *interpretation function* $\cdot^{\mathcal{I}}$. It maps every atomic concept P to a subset $P^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, every role R to a subset $R^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and every attribute F to a partial function $F^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow \Delta^{\mathcal{I}}$. Assume that $C^{\mathcal{I}}$, $D^{\mathcal{I}}$, and $R^{\mathcal{I}}$ are already given. Then $(C \sqcap D)^{\mathcal{I}} \stackrel{\text{def}}{=} C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $(\neg P)^{\mathcal{I}} = \Delta^{\mathcal{I}} - P^{\mathcal{I}}$, $(\mathsf{fills}\ F\ f)^{\mathcal{I}} \stackrel{\text{def}}{=} \{d \in \Delta^{\mathcal{I}} | F^{\mathcal{I}}(d) = f\}, (\forall R.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{d \in \Delta^{\mathcal{I}} | \forall (d, d') \in R^{\mathcal{I}} \Rightarrow d' \in C^{\mathcal{I}}\}, (\geq\ n\ R)^{\mathcal{I}} \stackrel{\text{def}}{=} \{d \in \Delta^{\mathcal{I}} || R^{\mathcal{I}}(d)| \geq n\}$, $(\leq\ n\ R)^{\mathcal{I}} \stackrel{\text{def}}{=} \{d \in \Delta^{\mathcal{I}} || R^{\mathcal{I}}(d)| \leq n\}$. The DL also allows for introducing abbreviations for complex concepts and concept specializations. Let P be an atomic concept and C a concept. A *terminological axiom* is a statement of the form $P \doteq C$ or $P \sqsubseteq C$. A finite set of terminological axioms is called a *terminology* (or *TBox*) if no concept appears more than once on the left-hand side of a terminological axiom and if no cyclic definitions are present. An interpretation $\mathcal{I}$ *satisfies* a

$$
\begin{array}{rcl}
\text{sports-kind} & \sqsubseteq & \top \\
\text{individual-sports} & \sqsubseteq & \text{sports-kind} \\
\text{team-sports} & \sqsubseteq & \text{sports-kind} \\
\text{sports-tool} & \sqsubseteq & \top \\
\text{football} & \sqsubseteq & \text{sports-tool} \\
\text{basketball} & \sqsubseteq & \text{sports-tool} \\
\text{tennis-racket} & \sqsubseteq & \text{sports-tool} \\
\text{sports-broadcast} & \sqsubseteq & \top \\
\text{team-sports-broadcast} & \doteq & \text{sports-broadcast} \sqcap (= \ 1 \ \text{kind-of-sports}) \\
 & & \forall \text{kind-of-sports.team-sports} \\
\text{individual-sports-broadcast} & \doteq & \text{sports-broadcast} \sqcap (= \ 1 \ \text{kind-of-sports}) \\
 & & \forall \text{kind-of-sports.individual-sports} \\
\text{football-broadcast} & \doteq & \text{team-sports-broadcast} \sqcap (= \ 1 \ \text{has-sports-tool}) \\
 & & \forall \text{has-sports-tool.football} \\
\text{basketball-broadcast} & \doteq & \text{team-sports-broadcast} \sqcap (= \ 1 \ \text{has-sports-tool}) \\
 & & \forall \text{has-sports-tool.basketball} \\
\text{tennis-broadcast} & \doteq & \text{individual-sports-broadcast} \sqcap (= \ 1 \ \text{has-sports-tool}) \\
 & & \forall \text{has-sports-tool.tennis-racket} \\
\text{figure-skating-broadcast} & \sqsubseteq & \text{individual-sports-broadcast}
\end{array}
$$

FIGURE 1.1. A terminology describing knowledge about sports broadcasts.

terminological axiom $P \doteq C$ ($P \sqsubseteq C$) iff $P^{\mathcal{I}} = C^{\mathcal{I}}$ ($P^{\mathcal{I}} \subseteq C^{\mathcal{I}}$). An interpretation $\mathcal{I}$ is a *model* of a terminology iff all terminological axioms are satisfied. A concept C is *satisfiable* w.r.t. a terminology iff there exists a model such that $C^{\mathcal{I}} \neq \emptyset$. A concept C *subsumes* a concept D w.r.t. a terminology ($C \succeq D$) iff $D^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$. C and D are *equivalent* ($C \approx D$) iff $C \succeq D$ and $D \succeq C$. These notions can be extended for dealing with individuals. Due to space constraints, we omit the technical details.

Figure 1.2 shows a knowledge base about sports broadcasts. Consider a user who likes to retrieve TV broadcasts similar to football-broadcast and basketball-broadcast. Then, computing the LCS of football-broadcast and basketball-broadcast would result in team-sports-broadcast $\sqcap$ ($=$ 1 has-sports-tool) $\sqcap$ $\forall$has-sports-tool.sports-tool and all knowledge bases would be queried for TV broadcast individuals which are subsumed by this concept. However, consider a user who collects football broadcasts and tennis broadcasts. The LCS computation then yields the concept sports-broadcast $\sqcap$ ($=$ 1 has-sports-tool) $\sqcap$ $\forall$has-sports-tool.sports-tool. Now querying the knowledge bases for individuals which are subsumed by such a general concept would result in a large amount of TV broadcasts which was not intended by the user. This unwanted behavior is due to the fact that the (crisp) LCS operator computes only those concepts which *fully* subsume football-broadcast and tennis-broadcast. The probabilistic information that tennis-broadcast with some probability is a team-sports-broadcast is not used in the computation process. In order to take such information into account, [KLP97] introduced a probabilistic extension of the DL which we will summarize briefly in the following. The DL underlying our probabilistic LCS operator was chosen such that this probabilistic extension is applicable to it.

To fully describe a concept, we need to describe the atomic concepts it is subsumed by, the properties of attribute fillers, number restrictions on roles and the properties of its role

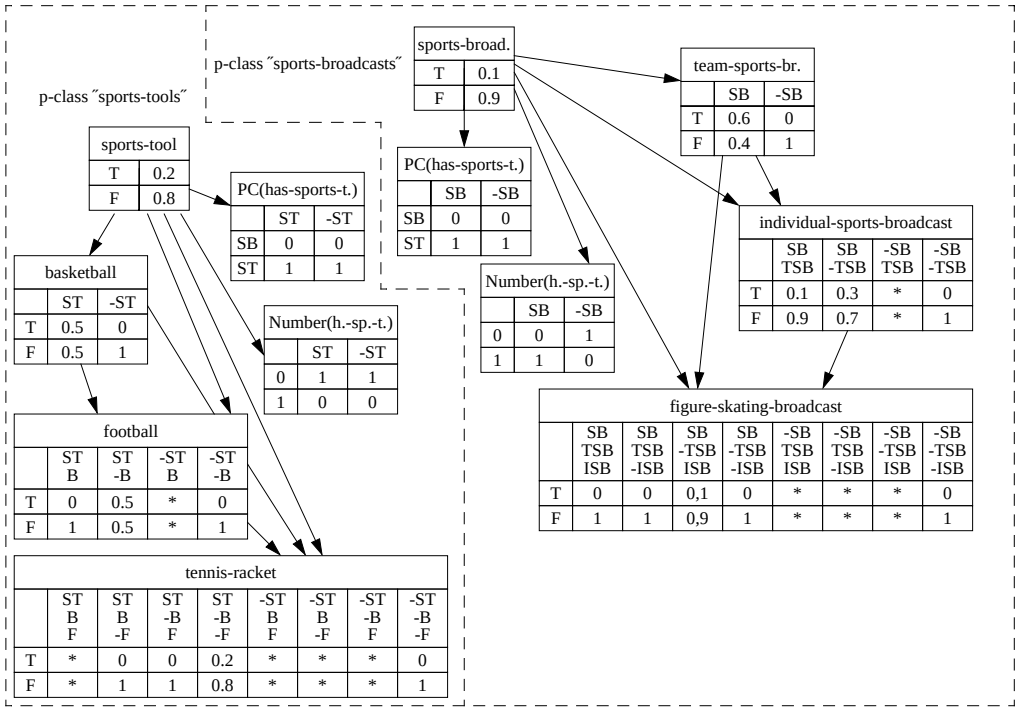

FIGURE 1.2. P-classes for the knowledge base about sports broadcasts.

fillers. Therefore, the terminology is extended by a set $\mathcal{PCL}$ of p-classes. Each p-class consists of a Bayesian network and one of the p-classes is the root p-class PCL^*. The root p-class describes the properties of concepts and all other p-classes describe the properties of role fillers. The Bayesian networks are modeled as DAGs whose nodes represent atomic concepts (P), attribute fillers ($Fills(F)$), number restrictions for roles ($Number(R)$) and properties of role fillers ($PC(R)$). Dependencies in the network are modeled by edges. For atomic concepts the range of the variables of each node can be either $true$ or $false$, for $Fills(F)$ it consists of abstract individuals, and for $Number(R)$ it is a subset of $\mathbb{N}$. In order to guarantee termination of the inference algorithm, this subset must be finite. Thus, the number of role fillers for a role is bounded. $bound(R)$ indicates the maximum number of role fillers for R. A $PC(R)$-node determines the p-class which the role fillers of a role are drawn from.

Figure 1.2 shows our knowledge base about sports broadcasts enriched by probability information. For instance, it is stated that a broadcast is considered to be about an individual sports (ISB) with probability 0.7 given that it is a broadcast about sports (SB) where no teams are involved ($\neg$TSB). Two p-classes are represented. sports-broadcasts is the root p-class and the role fillers for has-sports-tool are chosen from the p-class sports-tools. For each concept C the probability $P_{PCL^*}(C)$ with which an individual is subsumed by C can then be computed by a standard inference algorithm for Bayesian networks. In our example the probability of P_{PCL*}(team-sports-broadcast $\sqcap$ (= 1 has-sports-tool) $\sqcap$ $\forall$has-sports-tool.basketball) is computed by setting the nodes for team-sports-broadcast and basketball to *true*, $number$(has-sports-tool) = 1, and PC(has-sports-tool) = sports-tools. By Bayes net propagation we yield a value of 0.006. With this formalism it is possible to express the degree of overlap between concepts by a probability.

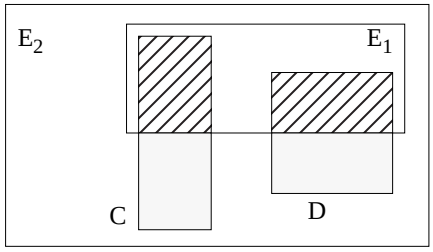

FIGURE 1.3. Scenario of four concepts illustrating the meaning of the degree of LCS subsumption and the LCS subsumption probability.

1.3 A Probabilistic Extension of the LCS Operator

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

Intuitively, given two concepts C and D, the key idea is to allow those concepts as candidates for a *probabilistic least common subsumer* of C and D which have a non-empty overlap with C and D. In order to keep the set of these concepts finite, we consider only concepts whose "depth" is not larger than the maximum depth of C and D. From the viewpoint of information retrieval this is no severe restriction, since in practical applications deeply nested concepts usually do not subsume any relevant individuals (e.g., $\mathsf{FB} \sqcap \forall \mathsf{has\text{-}sports\text{-}tool}.\forall \mathsf{has\text{-}sports\text{-}tool}.\mathsf{F}$ in our example). We introduce the *degree of LCS subsumption* as a quality measure for such a concept to be a probabilistic LCS of C and D. By this extension the qualities of potential least common subsumers can be compared to one another and the ones with "bad" quality will be eliminated from the set of possible probabilistic LCSs of C and D.

Furthermore, unlike in the definition of the crisp LCS, a concept expression does not necessarily need to subsume C and D *completely* in order to be a probabilistic LCS. In addition to the degree of LCS subsumption, we introduce a second quality measure which indicates the *LCS subsumption probability*, i.e., the probability with which a randomly chosen individual, which is subsumed by this probabilistic LCS, subsumes both C and D.

Figure 1.3 illustrates the meaning of both measures given four concepts represented as areas in the 2D space. The larger the hatched area compared to E_1, the higher the value for the degree of LCS subsumption of E_1. Conversely, the larger the area of $C \cup D$ that is not contained in E_1 (the gray area), the lower the LCS subsumption probability for E_1. For example, the concept E_2 has a better LCS subsumption probability than E_1, because the areas for C and D are both completely contained in E_2. On the other hand, the degree of LCS subsumption of E_2 is lower than the one of E_1, since the area $E_2 \setminus (C \cup D)$ is larger than $E_1 \setminus (C \cup D)$.

With the above considerations we will define the set of probabilistic LCSs of two concepts C and D as a set of triples where the first component is a concept and the other components are probabilities indicating both the degree of LCS subsumption and the LCS subsumption probability, respectively. In a concrete application a user should be able to specify minimal values for both measures that he is willing to accept.

The probabilistic LCS operator is applied to the parent concepts of individuals representing examples of a user's information demands. In our analysis, we use the canonical form of a concept. The canonical form of a concept C_i is given by

$$C_i \quad = \quad \alpha_i \sqcap \beta_{i1} \sqcap \ldots \sqcap \beta_{in_i} \qquad \text{with} \tag{1.1}$$

$$\begin{aligned}\alpha_i &= P_{i1} \sqcap \ldots \sqcap P_{ik_i} \sqcap (\mathsf{fills}\ F_{i1}\ f_{i1}) \sqcap \ldots \sqcap (\mathsf{fills}\ F_{il_i}\ f_{il_i}) \\ \beta_{ij} &= (\geq\ l_{ij}\ R_{ij}) \sqcap (\leq\ m_{ij}\ R_{ij}) \sqcap (\forall R_{ij}.C'_{ij})\end{aligned}$$

where C'_{ij} is also in canonical form, P_{ik} are atomic concepts or negations of atomic concepts. No P_{ik} or attribute F_{il} appears more than once in α_i. The canonical form of a concept is unique and every concept can be transformed into canonical form in linear time. The *depth* of a concept C in canonical form is defined as follows: $depth(\alpha) = 0, depth((\geq\ l\ R) \sqcap (\leq\ m\ R)) = 0, depth(\forall R.C') = 1 + depth(C')$, and $depth(C \sqcap D) = \max\{depth(C), depth(D)\}$.

In the following let $C = \{C_1, \ldots, C_m\}$ be a set of m concepts in canonical form. When defining the set of probabilistic least common subsumers $p\text{-}lcs(C)$ we consider only concepts with a non-empty overlap with each of the $C_1, \ldots, C_m$. The maximum depth of the concepts in $p\text{-}lcs(C)$ is limited by the maximum depth of the $C_1, \ldots, C_m$ in order to guarantee $p\text{-}lcs(C)$ to be finite. In order to formalize the idea that only the triples with "best" degree of LCS subsumption and LCS subsumption probability should be considered, we say that a tuple (p, q) *dominates* (p', q'), iff $p < p' \Rightarrow q > q' \wedge q < q' \Rightarrow p > p'$. We can now define the set of probabilistic least common subsumers of $C_1, \ldots, C_m$ as follows:

Definition 1 *Let C be a set of concepts in canonical form and PCL^* the root p-class of the* P-Classic *KB. Then we define the set of* probabilistic least common subsumers *of C as*

$$\begin{aligned}p\text{-}lcs(C) \stackrel{\text{def}}{=}\ & \{(E, p, q) \in \mathcal{C} \times I\!R \times I\!R \mid \\ & \forall i \in \{1, \ldots, m\} : P_{PCL^*}(E \sqcap C_i) > 0 \wedge depth(E) \leq \max_{i=1,\ldots,m} \{depth(C_i)\} \wedge \\ & p = \frac{P'(E, C)}{P_{PCL^*}(E)} \wedge q = \frac{P_{PCL^*}(E)}{P_{PCL^*}(E) + P''(E, C)} \wedge \\ & \forall (E', p', q')\ with\ P_{PCL^*}(E' \sqcap C_1) > 0, \ldots, P_{PCL^*}(E' \sqcap C_m) > 0, \\ & depth(E') \leq \max_{i=1,\ldots,m} \{depth(C_i)\} :\ (p, q)\ dominates\ (p', q')\} \quad where\end{aligned}$$

$$\begin{aligned}P'(E, C) \stackrel{\text{def}}{=}\ & P_{PCL^*}(E \sqcap \neg C_1 \sqcap \ldots \sqcap \neg C_{m-1} \sqcap C_m) + \\ & P_{PCL^*}(E \sqcap \neg C_1 \sqcap \ldots \sqcap \neg C_{m-2} \sqcap C_{m-1} \sqcap \neg C_m) + \ldots + \\ & P_{PCL^*}(E \sqcap C_1 \sqcap \ldots \sqcap C_{m-1} \sqcap \neg C_m) + \\ & P_{PCL^*}(E \sqcap C_1 \sqcap \ldots \sqcap C_m), \\ P''(\neg E, C) \stackrel{\text{def}}{=}\ & P_{PCL^*}(\neg E \sqcap \neg C_1 \sqcap \ldots \sqcap \neg C_{m-1} \sqcap C_m) + \\ & P_{PCL^*}(\neg E \sqcap \neg C_1 \sqcap \ldots \sqcap \neg C_{m-2} \sqcap C_{m-1} \sqcap \neg C_m) + \ldots + \\ & P_{PCL^*}(\neg E \sqcap C_1 \sqcap \ldots \sqcap C_{m-1} \sqcap \neg C_m) + \\ & P_{PCL^*}(\neg E \sqcap C_1 \sqcap \ldots \sqcap C_m).\end{aligned}$$

$p\text{-}lcs(C)$ is called minimal *iff* $\forall (E, p, q) \in p\text{-}lcs(C) : \forall (E', p', q') \in p\text{-}lcs(C) : E' \neq E \Rightarrow E' \not\approx E$.

In Definition 1 we formalize the ideas of Figure 1.3 conditioned on the general case of m concepts. The degree of LCS subsumption and the LCS subsumption probability are expressed by p and q, respectively.

Proposition 1 *The set $p\text{-}lcs(C)$ is well-defined in the following sense:*

(1) p and q are probabilities, i.e., $p, q \in [0,1]$.

(2) $\forall(E,p,q) \in p\text{-}lcs(C): \neg\exists(E',p',q'): P_{PCL^*}(E' \sqcap C_1) > 0 \wedge \ldots \wedge P_{PCL^*}(E' \sqcap C_m) > 0 \wedge depth(E') \leq \max_{i=1,\ldots,m}\{depth(C_i)\} \wedge p' > p \wedge q' > q.$

Proof. Let C be a set of concepts and $(E,p,q) \in p\text{-}lcs(C)$. Then $P_{PCL^*}(E) = P_{PCL^*}(E \sqcap \neg C_1 \sqcap \ldots \sqcap \neg C_{m-1} \sqcap C_m) + \ldots + P_{PCL^*}(E \sqcap C_1 \sqcap \ldots \sqcap C_{m-1} \sqcap \neg C_m) + P_{PCL^*}(E \sqcap C_1 \sqcap \ldots \sqcap C_m) + P_{PCL^*}(E \sqcap \neg C_1 \sqcap \ldots \sqcap \neg C_m) = P'(E,C) + P_{PCL^*}(E \sqcap \neg C_1 \sqcap \ldots \sqcap \neg C_m)$. Since all addends of $P'(E,C)$ and $P_{PCL^*}(E \sqcap \neg C_1 \sqcap \ldots \sqcap \neg C_m)$ are greater than or equal to 0, we have

$$0 \leq p = \frac{P'(E,C)}{P'(E,C) + P(E \sqcap \neg C_1 \sqcap \ldots \sqcap \neg C_m)} \leq 1$$

In order to show $q \in [0,1]$, it is sufficient to prove $P_{PCL^*}(\neg E \sqcap \neg C_1 \sqcap \ldots \sqcap \neg C_{m-1} \sqcap C_m) + P_{PCL^*}(\neg E \sqcap C_1 \sqcap \ldots \sqcap C_{m-1} \sqcap \neg C_m) + \ldots + P_{PCL^*}(\neg E \sqcap C_1 \sqcap \ldots \sqcap C_m) = P''(E,C) > 0$. However, this is obvious, since for all addends the Bayes net generates positive numbers or zero which proves (1). (2) is an immediate consequence of Definition 1. □

In order to compute $p\text{-}lcs(C)$ we must first find the set of concepts which have a non-empty overlap with each of the $C_1, \ldots, C_m$.

Algorithm 1 compute-concept-candidates$(C_1, \ldots, C_m, PCL_1, \ldots, PCL_m, depth)$

$B_1 := \emptyset, B_2 := \emptyset, B_3 := \emptyset$
for all atomic concepts and concepts of the form (fills F f) C in $C_1, \ldots, C_m$ **do**
  add C to B_1 if $P_{PCL_1}(C \sqcap C_1) > 0 \wedge \ldots \wedge P_{PCL_m}(C \sqcap C_m) > 0$
end for
repeat
  $B_1' := B_1$
  recursively add all concepts $E = C \sqcap D$ to B_1 where C and D are either atomic or negated atomic concepts or of the form (fills F f) in B_1 if $P_{PCL_1}(E \sqcap C_1) > 0 \wedge \ldots \wedge P_{PCL_m}(E \sqcap C_m) > 0 \wedge \forall C' \in B_1 : E \not\approx C'$
until $B_1' \neq B_1$
for all $(\geq\ i\ R) \sqcap (\leq\ j\ R)$ in $C_1, \ldots, C_m$ **do**
  add $(\geq\ k\ R) \sqcap (\leq\ l\ R)$ to B_2 if $0 \leq k \leq j \wedge bound(R) \geq l \geq i \wedge k \leq l$
end for
if $depth > 0$ **then**
  for all $\forall R.C_1', \ldots, \forall R.C_m'$ in $C_1, \ldots, C_m$ **do**
    $B_3 :=$ *compute-concept-candidates*$(C_1', \ldots, C_m', PC(\forall R.C_1'), \ldots, PC(\forall R.C_m'), depth-1)$
  end for.
end if
return $\{X \sqcap Y \sqcap Z \mid X \in B_1 \cup \{\top\} \wedge Y \in B_2 \cup \{\top\} \wedge Z \in B_3 \cup \{\top\} \wedge \forall i \in \{1, \ldots, m\} : P(X \sqcap Y \sqcap Z \sqcap C_i) > 0\}$

Algorithm 1 computes this set given concepts $C_1, \ldots, C_m$, p-classes $PCL_1, \ldots, PCL_m$ which are initialized with the root p-class at first call and the maximum depth over $C_1, \ldots, C_m$. In the first step all relevant atomic and fills-concepts are collected in the set B_1. Let us compute the set $p\text{-}lcs(\{\text{football-transmission,tennis-transmission}\})$. For $p\text{-}lcs(\{\mathsf{FB},\mathsf{TB}\})$ we have $B_1 = \{\mathsf{SB},\mathsf{TSB},\mathsf{ISB}\}$. In addition, their conjunctions and negations (in the case of atomic concepts)

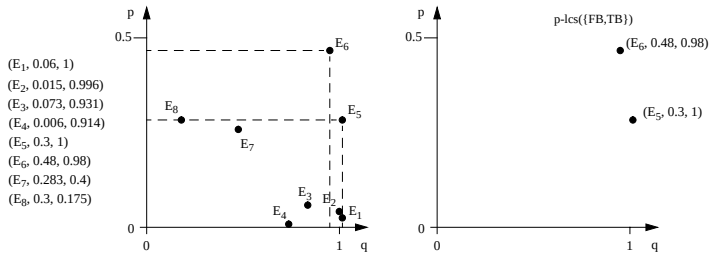

FIGURE 1.4. Example demonstrating how $p\text{-}lcs(C)$ is computed.

are added as candidates for probabilistic LCS concepts to B_1. Therefore, in the example we have: $B_1 := B_1 \cup \{\mathsf{TSB} \sqcap \mathsf{ISB}\}$. B_2 contains all number restrictions which overlap with $C_1, \ldots, C_m$. In our case we compute $B_2 = \{(= \ 1 \ \mathsf{has\text{-}sports\text{-}tool})\}$. Universal quantifications are handled recursively in the last part of the algorithm. Thereby, the maximum number of recursions is limited to *depth* which guarantees termination of the algorithm. In our example we get $B_3 = \{\forall \mathsf{has\text{-}sports\text{-}tool.ST}\}$. Let A be an abbreviation for $(= \ 1 \ \mathsf{has\text{-}sports\text{-}tool}) \sqcap \forall \mathsf{has\text{-}sports\text{-}tool.ST}$. Then the algorithm returns the set $\{E_1, \ldots, E_8\} = \{\mathsf{SB}, \mathsf{TSB}, \mathsf{ISB}, \mathsf{TSB} \sqcap \mathsf{ISB}, \mathsf{SB} \sqcap A, \mathsf{TSB} \sqcap A, \mathsf{ISB} \sqcap A, \mathsf{TSB} \sqcap \mathsf{ISB} \sqcap A\}$.

Theorem 1 *Algorithm* compute-concept-candidates *is sound and complete. In other words, for concepts $C_1, \ldots, C_m$, p-classes $PCL_1, \ldots, PCL_m$ initialized with the root p-class PCL^*, and* depth $\in \mathbb{N}$, *depth* $= \max_{i=1,\ldots,m}\{depth(C_i)\}$ *it computes the minimal set of concepts* $\{E \mid P_{PCL^*}(E \sqcap C_1) > 0, \ldots, P_{PCL^*}(E \sqcap C_m) > 0 \wedge depth(E) \leq \max_{i=1,\ldots,m}\{depth(C_i)\}\}$. □

Once the set of possible candidates for probabilistic LCS concepts is computed, the parameters p and q must be determined for each candidate by means of the formulas given in Definition 1. The set $p\text{-}lcs(C)$ contains only those triples whose quality measures dominate those of other triples. In the left part of Figure 1.4 both value pairs for each of the candidates are shown.

If only two candidates are present, $p\text{-}lcs(C)$ can be determined by a simple *if-then-else*-test. In the general case this set can be effectively computed. First, the triples $(E_1, p_1, q_1), \ldots, (E_n, p_n, q_n)$ are arranged as points in a diagram where each point is defined by its p- and q-parameter as shown in Figure 1.4. Then $p\text{-}lcs(C)$ can be computed by the following procedure:

$p\text{-}lcs(C) := \text{sort}(((E_1, p_1, q_1), \ldots, (E_n, p_n, q_n)), q_i)$
for $i = 1$ to n **do**
  eliminate all (E', p', q') from $p\text{-}lcs(C)$ with $p' < p$ and $q' < q$
end for.

In Figure 1.4 the area of points which would be eliminated in two steps of the loop is indicated by dotted lines. For our example we get $p\text{-}lcs(\{\mathsf{FB}, \mathsf{TB}\}) = \{(\mathsf{SB} \sqcap (= \ 1 \ \mathsf{has\text{-}sports\text{-}tool}) \sqcap \forall \mathsf{has\text{-}sports\text{-}tool.ST}, 0.3, 1), (\mathsf{TSB} \sqcap (= \ 1 \ \mathsf{has\text{-}sports\text{-}tool}) \sqcap \forall \mathsf{has\text{-}sports\text{-}tool.ST}, 0.48, 0.98)\}$. As a result we get two possible concepts: the one the crisp LCS would have computed and an additional one with a better representation of the commonalities of FB and TB at the expense of an imperfect LCS subsumption probability. The second concept does not subsume any figure skating broadcasts, since such broadcasts involve no team sports (unlike FB and TB) according to our KB.

1.4 Negative Examples

Often it is desirable for a user to not only characterize information he likes to be supplied with, but also information he is *not* interested in. In a concrete application such as the TV-Assistant a user should be able to specify *negative examples* — TV broadcasts whose presentation should be avoided when retrieving TV program information if possible. Let us assume a user likes to retrieve sports broadcasts similar to football broadcasts and tennis broadcasts which do not have the properties of basketball broadcasts if possible.

In terms of the probabilistic LCS operator we are looking for the set of probabilistic LCS concepts (representing the characteristics of the positive examples) which share as few as possible commonalities with the parent concepts of the negative examples. Therefore, we add a third parameter to every triple in $p\text{-}lcs(C)$. This degree of *LCS subsumption of a negative example* indicates the probability of occurrence of an individual which is subsumed by one of the parent concepts of the negative examples provided that it is subsumed by the probabilistic LCS concept. A potential probabilistic LCS candidate is considered the *better*, the smaller this probability is. From our example KB it follows that TSB overlaps with BB more than SB. Thus, its degree of LCS subsumption of a negative example will be higher. We say that a triple (p, q, r) *dominates* (p', q', r'), iff $(p < p' \wedge q < q' \Rightarrow r < r') \wedge (p < p' \wedge r > r' \Rightarrow q > q') \wedge (q < q' \wedge r > r' \Rightarrow p > p')$. In the following let $N = \{N_1, \ldots, N_n\}$ denote the set of parent concepts of the negative examples given as KB individuals.

Definition 2 *Let C, N be sets of concepts in canonical form representing the parent concepts of m positive and n negative examples of information items, respectively. Then we define the set of* probabilistic least common subsumers *of C* w.r.t. *N as*

$$
\begin{aligned}
p\text{-}lcs^*(C,N) \stackrel{\text{def}}{=} \; & \{(E,p,q,r) \in \mathcal{C} \times I\!R \times I\!R \times I\!R \mid \\
& \forall i \in \{1,\ldots,m\} : P_{PCL^*}(E \sqcap C_i) > 0 \wedge depth(E) \leq \max_{i=1,\ldots,m}\{depth(C_i)\} \wedge \\
& p = \frac{P'(E,C)}{P_{PCL^*}(E)} \wedge q = \frac{P_{PCL^*}(E)}{P_{PCL^*}(E) + P''(E,C)} \wedge r = \frac{P'(E,N)}{P_{PCL^*}(E)} \\
& \forall (E',p',q',r') \text{ with } P_{PCL^*}(E' \sqcap C_1) > 0, \ldots, P_{PCL^*}(E' \sqcap C_m) > 0, \\
& depth(E') \leq \max_{i=1,\ldots,m}\{depth(C_i)\} : \; (p,q,r) \text{ dominates } (p',q',r')\}
\end{aligned}
$$

$p\text{-}^*lcs(C,N)$ *is called* minimal *iff* $\forall (E,p,q,r) \in p\text{-}lcs^*(C,N) : \forall (E',p',q',r') \in p\text{-}lcs^*(C,N) : E \neq E' \Rightarrow E \not\approx E'$.

Let $C = \{\mathsf{FB}, \mathsf{TB}\}$ and $N = \{\mathsf{BB}\}$, i.e., a user likes to retrieve TV broadcasts similar to FB and TB which should not have the properties of BB if possible. Then for $\{E_1, \ldots, E_8\}$, p and q are computed as in Definition 1. Additionally, we have to compute r for each of the elements in $p\text{-}lcs(C)$. We get $\{r_{E_1}, \ldots, r_{E_8}\} = \{0.06, 0.1, 0.03, 0.1, 0.3, 0.5, 0.17, 0.5\}$. The rest of the computation of $p\text{-}lcs^*(C,N)$ can be performed the same way as $p\text{-}lcs(C)$. However, due to the third parameter r in Definition 2, the algorithm for searching for the probabilistic LCS quadruples must be executed in the 3D space. Hence we have $p\text{-}lcs^*(C,N) = \{(\mathsf{SB} \sqcap (=\ 1\ \mathsf{has\text{-}sports\text{-}tool}) \sqcap \forall \mathsf{has\text{-}sports\text{-}tool.ST}, 0.3, 1, 0.3), (\mathsf{TSB} \sqcap (=\ 1\ \mathsf{has\text{-}sports\text{-}tool}) \sqcap \forall \mathsf{has\text{-}sports\text{-}tool.ST}, 0.48, 0.98, 0.5), (\mathsf{ISB}, 0.07, 0.93, 0.03)\}$. Compared to the result of the last section we additionally get the concept ISB which has the best possible value for the degree of LCS subsumption of a negative example.

Proposition 2 *The set p-lcs*(C, N) is well-defined in the following sense:*

(1) p, q *and* r *are probabilities, i.e.,* $p, q, r \in [0, 1]$.

(2) $\forall(E, p, q, r) \in p\text{-}lcs^*(C, N) : \neg\exists(E', p', q', r') : P_{PCL^*}(E' \sqcap C_1) > 0 \wedge \ldots \wedge P_{PCL^*}(E' \sqcap C_m) > 0 \wedge ((p' > p \wedge q' > q \wedge r' < r) \Rightarrow E' = E)$.

□

The proof of Proposition 1 can be easily extended to proof Proposition 2.

As Proposition 2 shows, our definition of the probabilistic least common subsumer considering negative examples does indeed what we expect it to do. It can be assured that the concepts as first parameter of the quadruples in $p\text{-}lcs^*(C, N)$ either have the best possible degree of LCS subsumption, LCS subsumption probability, or degree of LCS subsumption of a negative example.

1.5 Discussion and Conclusion

Unlike the crisp LCS algorithm presented in [CBH92], our algorithm is no longer polynomial in the number of the concepts in the worst case. Principally this is due to the factor computation of the parameters p, q, and r which in the worst case demands exponentially many Bayes net propagations in the number of concepts. In practice, a lot of the involved probability multiplications will evaluate to zero and therefore need not be considered in the further computation process. The number of parameter sets that have to be computed depends on the set of possible $p\text{-}lcs$-candidates. In general, only for knowledge bases with many overlapping concepts this number is high. Usually it can be computed by only a few iterations as in our example of sports broadcasts. Sometimes parameters do no even have to be computed at all. Let $(E_1, p_1, 1, r_1), E_2, p_2$ and r_2 already be computed, $p_1 \geq p_2$ and $r_1 \leq r_2$. Then computing q_2 can be omitted, since it is clear that E_2 is no $p\text{-}lcs$-candidate. Furthermore, the q-parameter should always be determined last (if necessary) due to the higher computational costs for computing $P''(E, C)$ compared to $P'(E, C)$.

Considering the growing number of information sources especially on the WWW, intelligent information retrieval becomes an important task. In this paper we have made a new approach to tackle this problem. Information retrieval is modeled by knowledge base queries based on the common properties of example individuals of a description logic. We introduced a least common subsumer operator in order to compute commonalities between concepts in a probabilistic description logic. Furthermore we have presented an extension to the probabilistic LCS operator to also regard negative examples in information queries, i.e., examples of unwanted information. It can be proved that our algorithm finds the set of probabilistic least common subsumers with optimal parameters for degree of LCS subsumption, LCS subsumption probability, and degree of LCS subsumption of a negative example. The probabilistic LCS operator serves as a key mechanism for content-based information retrieval. Whereas in this paper we focused on the TV-Assistant example application, our operator is applicable for content-based information retrieval in all kinds of heterogeneous information sources. The idea is to compute the probabilistic LCS to the schema of each of the component information sources. Thus, the dominating LCS results (see Figure 1.4) might refer to different information sources. However, details of this approach still have to be developed.

1.6 References

[AKS96] Y. Arens, C. Knoblock, and W. Shen. Query Reformulation for Dynamic Information Integration. *Journal of Intelligent Information Systems*, 1996.

[CBH92] W. W. Cohen, A. Borgida, and H. Hirsh. Computing Least Common Subsumers in Description Logics. In *Proc. of the Int. Conf. on Fifth Generation Computer Systems*, pages 1036–1043, ICOT, Japan, 1992. Ass. for Computing Machinery.

[CL93] T. Catarci and M. Lenzerini. Interschema Knowledge in Cooperative Information Systems. In G. Schlageter M. Huhns, M.P. Papazoglou, editor, *Proc. of the Int. Conf. on Int. and Cooper. Inf. Sys.*, pages 55–63. IEEE Comp. Society Press, 1993.

[Hei94] J. Heinsohn. Probabilistic Description Logics. In Ramon Lopez de Mantaras and David Poole, editors, *Proc. of the 10th Conf. on Uncertainty in AI*, pages 311–318, San Francisco, CA, USA, July 1994. Morgan Kaufmann Publishers.

[Jae94] M. Jaeger. Probabilistic Reasoning in Terminological Logics. In Pietro Torasso Jon Doyle, Erik Sandewall, editor, *Proc. of the 4th Int. Conf. on Principles of Knowledge Representation and Reasoning*, pages 305–316, Bonn, FRG, May 1994. Morgan Kaufmann.

[KLP97] 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.

[LP97] P. Lambrix and L. Padham. A description logic model for querying knowledge bases for structured documents. In *Proc. of the Tenth Int. Symposium on Methodologies for Intelligent Systems, LNAI 1325*, pages 72–83. Springer-Verlag, 1997.

[LRO96] A. Y. Levy, A. Rajaraman, and J. J. Ordille. Query-answering Algorithms for Information Agents. In *Proc. of the Thirteenth National Conf. on AI and the Eighth Innovative Applications of AI Conf.*, pages 40–47, Menlo Park, August4–8 1996. AAAI Press / MIT Press.

[LSW98] P. Lambrix, N. Shahmehri, and N. Wahlöf. A Default Extension to Description Logics forUse in an Intelligent Search Engine. In *Proc. of the 1st Hawaiian Int. Conf. on System Science*, pages 28–35, 1998.

[MHN98] R. Möller, V. Haarslev, and B. Neumann. Semantics-based Information Retrieval. In *Int. Conf. on Information Technology and Knowledge Systems*, Vienna, Budapest, August–September, 1998.

[MMB97] D. L. McGuinness, H. Manning, and T. Beattie. Knowledge Assisted Search. In *Proc. of the Int. Joint Conf. on AI Workshop on The Future of AI and the Internet*, 1997.

[Str98] U. Straccia. A Fuzzy Description Logic. In *Proc. of the AAAI*, Wisconsin, 1998.

[SV97] S. Salotti and V. Ventos. Study and Formalization of a Case-Based Reasoning System with a Description Logic. In *11th Int. Workshop on Description Logics DL'97, Rousset et al. (editors), Gif sur Yvette, Universite Paris Sud, Laboratoire de Recherche en Informatique (LRI), CNRS, 1997*, pages 114–118, 1997.

[TM98] C. B. Tresp and R. Molitor. A Description Logic for Vague Knowledge. Technical Report LTCS-Report 98-01, LuFg Theoretische Informatik, RWTH Aachen, 1998.