Instance-based Non-standard Inferences in $\mathcal{EL}$ with Subjective Probabilities

Rafael Peñaloza and Anni-Yasmin Turhan[*]

Institute for Theoretical Computer Science, TU Dresden, Germany,
email: *last name*@tcs.inf.tu-dresden.de

Abstract. For practical ontology-based applications representing and reasoning with probabilities is an essential task. For Description Logics with subjective probabilities reasoning procedures for testing instance relations based on the completion method have been developed.
In this paper we extend this technique to devise algorithms for solving non-standard inferences for $\mathcal{EL}$ and its probabilistic extension Prob-$\mathcal{EL}_c^{01}$: computing the most specific concept of an individual and finding explanations for instance relations.

1 Introduction

The ontology language recommended for the semantic web OWL [11, 25] is based on Description Logics (DLs) [4]. Description logics are knowledge representation formalisms with formal semantics. Based on these semantics, powerful reasoning services have been defined and reasoning algorithms have been investigated. In recent years, so-called lightweight DLs have been devised; these DLs have a limited expressiveness, which allows for efficient reasoning [6]. For the lightweight DL $\mathcal{EL}$, typical DL reasoning services such as classification of TBoxes, i.e., computation of all sub- / superconcept relations of named concepts, or the realization of ABoxes, i.e., computation of the named concepts each of the ABox individuals belongs to, can be done in polynomial time. The basis for ABox realization is *instance checking*, which tests whether a given individual from the ABox belongs to a given concept. In the so-called $\mathcal{EL}$-family of DLs, which are the tractable extensions of $\mathcal{EL}$, this inference can be computed using completion algorithms, which extend the ones for concept subsumption [2, 3].

The DLs from the $\mathcal{EL}$-family are employed most prominently in the medical field, for instance in the well-known knowledge base SNOMED CT [23], as well as in context-aware applications. In both of these application areas, the need for characterizing uncertain observations, which are only known to hold with some probability, has been long recognized. While several probabilistic extensions of DLs have been proposed—see [14] for a survey—these are typically very expressive and thus no longer tractable and they cannot handle subjective probabilities. A simple probabilistic variant of $\mathcal{EL}$ that can express subjective probabilities is Prob-$\mathcal{EL}_c^{01}$, recently introduced in [15]. This logic allows only a fairly limited use of uncertainty. More precisely, it is only possible to

[*] Partially supported by the German Research Foundation (DFG) in the Collaborative Research Center 912 "Highly Adaptive Energy-Efficient Computing".

express that a concept *may* hold ($P_{>0}C$), or that it holds *almost surely* ($P_{=1}C$). Despite its limited expressivity, this logic is interesting due to its nice algorithmic properties; as shown in [15], subsumption and instance checking can also be performed in polynomial time.

In this paper we employ the above mentioned completion algorithms to compute two non-standard inferences for DLs that allow to express subjective probability: the most specific concept and explanation of instance relations in Prob-$\mathcal{EL}_c^{01}$.

Many practical applications that need to represent observed information, such as medical applications or context-aware applications, need to characterize that these observations only hold with certain probability. Furthermore, these applications face the problem that information from different sources does not coincide, e.g., different diagnoses yield differing results. These applications need to "integrate" differing observations for the same state of affairs [24]. A way to determine what information the different information sources agree upon is to represent this information in the ABox by different individuals and then to find a common generalization of these individuals. A description of such a generalization of a group of ABox individuals can be obtained by applying the so-called *bottom-up approach* for constructing knowledge bases [5]. In this approach a set of individuals is generalized into a single concept description by first generating the most specific concept (msc) of each individual and then applying the least common subsumer (lcs) to the set of obtained concept descriptions to extract their commonalities.

The second step, i.e., a computation procedure for the approximate lcs has been investigated for $\mathcal{EL}$ and Prob-$\mathcal{EL}_c^{01}$ in [21]. In this paper we present a similar procedure for the msc. For the Description Logic $\mathcal{EL}$ the msc need not exist [1], if computed with respect to general $\mathcal{EL}$-TBoxes. However, it is still possible to find a concept description that is the msc up to a fixed role-depth. This so-called k-msc is still a generalization of the input, but not necessarily the least one—in this sense, it is only an approximation of the msc. We first describe a practical approach for computing the role-depth bounded msc, based on the polynomial-time completion algorithm for $\mathcal{EL}$, and then extend it to the probabilistic variant Prob-$\mathcal{EL}_c^{01}$. Our algorithms are based upon the completion algorithms for ABox realization in $\mathcal{EL}$ and in Prob-$\mathcal{EL}_c^{01}$ and thus can be easily implemented on top of reasoners of these DLs. All the proofs can be found in [18].

The second non-standard inference that we explore in this paper is the *explanation* of a given consequence. In case a large knowledge base is edited by hand, it is not trivial for the developer to see why a particular consequence holds [10, 12]. In our case of instance checking, we want to identify those statements in the TBox and the ABox that cause an instance relationship to follow from the knowledge base. More precisely, we want to compute *minimal axiom sets* (MinAs) that entail the consequence. We compute these sets using a glass-box approach for axiom-pinpointing [22, 7]. Even for ontology-based context-aware systems, which may operate on automatically generated ABoxes, the identification of MinAs that cause an unwanted consequence is crucial, since it is the first step to edit the knowledge base such that the consequence is resolved. More than in the crisp case, finding the axioms that entail a consequence for a knowledge base written in a DL with probabilities is a difficult task to do by hand.

A method to compute MinAs for subsumptions in $\mathcal{EL}$ was devised in [9] as an extension of the completion algorithm for TBox classification. In this paper we devise a method to compute MinAs for instance relationships as an extension of the completion algorithm for ABox realization for Prob-$\mathcal{EL}_c^{01}$.

This paper extends earlier work presented in [20, 21] by algorithms for computing explanations for instance relationships in $\mathcal{EL}$ and Prob-$\mathcal{EL}_c^{01}$. To the best of our knowledge, explanation has not yet been investigated for DLs that allow to express probabilities. We start this undertaking by giving the basic notions in Section 2. In Section 3 we recall the completion algorithms for ABox realization. Section 4 discusses the computation algorithm for the role-depth bounded msc. In Section 5 we introduce the algorithm for computing explanations.

2 $\mathcal{EL}$ and Prob-$\mathcal{EL}$

In this section we introduce the DL $\mathcal{EL}$ and its probabilistic variant Prob-$\mathcal{EL}_c^{01}$. Let N_I, N_C and N_R be disjoint sets of *individual-*, *concept-* and *role names*, respectively. *Prob-$\mathcal{EL}_c^{01}$-concept descriptions* are built using the syntax rule

$$C, D ::= \top \mid A \mid C \sqcap D \mid \exists r.C \mid P_{>0}C \mid P_{=1}C,$$

where $A \in N_C$, and $r \in N_R$. *$\mathcal{EL}$-concept descriptions* are Prob-$\mathcal{EL}_c^{01}$-concept description that do not contain the constructors $P_{>0}$ or $P_{=1}$.

A *knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. An $\mathcal{EL}$- (Prob-$\mathcal{EL}_c^{01}$-) *TBox* is a finite set of *general concept inclusions* (GCIs) of the form $C \sqsubseteq D$, where C, D are $\mathcal{EL}$- (Prob-$\mathcal{EL}_c^{01}$-) concept descriptions. An *$\mathcal{EL}$-ABox* is a set of assertions of the form $C(a)$ or $r(a, b)$, where C is an $\mathcal{EL}$-concept description, $r \in N_R$, and $a, b \in N_I$. A *Prob-$\mathcal{EL}_c^{01}$-ABox* is a set of assertions of the form $C(a)$, $r(a, b)$, $P_{>0}r(a, b)$, or $P_{=1}r(a, b)$, where C is a Prob-$\mathcal{EL}_c^{01}$-concept description, $r \in N_R$, and $a, b \in N_I$.

The semantics of $\mathcal{EL}$ is defined by means of interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a non-empty *domain* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$ that assigns binary relations on $\Delta^{\mathcal{I}}$ to role names, subsets of $\Delta^{\mathcal{I}}$ to concepts and elements of $\Delta^{\mathcal{I}}$ to individual names. For a more detailed description of this semantics, see [4].

We say that the interpretation $\mathcal{I}$ *satisfies* a general concept inclusion $C \sqsubseteq D$, denoted as $\mathcal{I} \models C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; it *satisfies* an assertion $C(a)$, denoted as $\mathcal{I} \models C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and it *satisfies* an assertion $r(a, b)$, denoted as $\mathcal{I} \models r(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. It is a *model* of a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if it satisfies all GCIs in $\mathcal{T}$ and all assertions in $\mathcal{A}$.

The semantics of Prob-$\mathcal{EL}_c^{01}$ is a generalization of the semantics of $\mathcal{EL}$, that considers a set of possible worlds. A *probabilistic interpretation* is of the form

$$\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu),$$

where $\Delta^{\mathcal{I}}$ is the (non-empty) *domain*, W is a (non-empty) set of *worlds*, μ is a discrete probability distribution on W, and for each world $w \in W$, $\mathcal{I}_w$ is a classical $\mathcal{EL}$ interpretation with domain $\Delta^{\mathcal{I}}$, where $a^{\mathcal{I}_w} = a^{\mathcal{I}_{w'}}$ for all $a \in N_I$, $w, w' \in W$. The probability that a given element of the domain $d \in \Delta^{\mathcal{I}}$ belongs to the interpretation of a concept name A is

$$p_d^{\mathcal{I}}(A) := \mu(\{w \in W \mid d \in A^{\mathcal{I}_w}\}).$$

The functions $\mathcal{I}_w$ and $p_d^{\mathcal{I}}$ are extended to complex concepts in the usual way for the classical $\mathcal{EL}$-constructors, where the extension to the new constructors P_* is defined as

$$(P_{>0}C)^{\mathcal{I}_w} := \{d \in \Delta^{\mathcal{I}} \mid p_d^{\mathcal{I}}(C) > 0\},$$
$$(P_{=1}C)^{\mathcal{I}_w} := \{d \in \Delta^{\mathcal{I}} \mid p_d^{\mathcal{I}}(C) = 1\}.$$

The probabilistic interpretation $\mathcal{I}$ *satisfies* a general concept inclusion $C \sqsubseteq D$, denoted as $\mathcal{I} \models C \sqsubseteq D$, if for every $w \in W$ it holds that $C^{\mathcal{I}_w} \subseteq D^{\mathcal{I}_w}$. It is a *model* of a TBox $\mathcal{T}$ if it satisfies all general concept inclusions in $\mathcal{T}$. Let C, D be two Prob-$\mathcal{EL}_c^{01}$ concepts and $\mathcal{T}$ a TBox. We say that C is *subsumed* by D w.r.t. $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$) if for every model $\mathcal{I}$ of $\mathcal{T}$ it holds that $\mathcal{I} \models C \sqsubseteq D$. The concepts c and D are equivalent, if $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$ holds. The probabilistic interpretation $\mathcal{I}$ *satisfies* the assertion $P_{>0}r(a, b)$ if $\mu(\{w \in W \mid \mathcal{I}_w \models r(a, b)\}) > 0$, and analogously for $P_{=1}r(a, b)$. $\mathcal{I}$ *satisfies* the ABox $\mathcal{A}$ if there is a $w \in W$ such that $\mathcal{I}_w \models \mathcal{A}$.

Finally, an individual $a \in N_I$ is an *instance* of a concept description C w.r.t. $\mathcal{K}$ ($\mathcal{K} \models C(a)$) if $\mathcal{I} \models C(a)$ for all models $\mathcal{I}$ of $\mathcal{K}$. The *ABox realization problem* is to compute for each individual a in $\mathcal{A}$ the set of named concepts from $\mathcal{K}$ that have a as an instance and that are least (w.r.t. $\sqsubseteq$). One of our main interests in this paper is to compute most specific concepts.

Definition 1 (most specific concept). *Let $\mathcal{L}$ be a DL, $\mathcal{K}$ = ($\mathcal{T}$, $\mathcal{A}$) be a $\mathcal{L}$-knowledge base. The* most specific concept *(msc) of an individual a from $\mathcal{A}$ is the $\mathcal{L}$-concept description C s. t.*

1. *$\mathcal{K} \models C(a)$, and*
2. *for each $\mathcal{L}$-concept description D holds: $\mathcal{K} \models D(a)$ implies $C \sqsubseteq_{\mathcal{T}} D$.*

The msc depends on the DL in use. For the DLs with conjunction as concept constructor the msc is, if it exists, unique up to equivalence. Thus it is justified to speak of *the* msc.

3 Completion Algorithms for ABox Realization

In this section we briefly sketch the completion algorithms for instance checking in the DLs $\mathcal{EL}$ [2] and Prob-$\mathcal{EL}_c^{01}$ [15].

3.1 The Completion Algorithm for $\mathcal{EL}$

Assume we want to test for an $\mathcal{EL}$-knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ whether $\mathcal{K} \models D(a)$ holds. The completion algorithm first augments the knowledge base by introducing a concept name for the complex concept description D for the instance check; that is, it redefines the knowledge base to $\mathcal{K} = (\mathcal{T} \cup \{A_q \equiv D\}, \mathcal{A})$, where A_q is a new concept name not appearing in $\mathcal{K}$. The instance checking algorithm for $\mathcal{EL}$ works on knowledge

$$
\begin{array}{lrcl}
\textbf{NF1} & C \sqcap \hat{D} \sqsubseteq E & \longrightarrow & \{\ \hat{D} \sqsubseteq A, C \sqcap A \sqsubseteq E\ \} \\
\textbf{NF2} & \exists r.\hat{C} \sqsubseteq D & \longrightarrow & \{\ \hat{C} \sqsubseteq A, \exists r.A \sqsubseteq D\ \} \\
\textbf{NF3} & \hat{C} \sqsubseteq \hat{D} & \longrightarrow & \{\ \hat{C} \sqsubseteq A, A \sqsubseteq \hat{D}\ \} \\
\textbf{NF4} & B \sqsubseteq \exists r.\hat{C} & \longrightarrow & \{\ B \sqsubseteq \exists r.A, A \sqsubseteq \hat{C}\ \} \\
\textbf{NF5} & B \sqsubseteq C \sqcap D & \longrightarrow & \{\ B \sqsubseteq C, B \sqsubseteq D\ \}
\end{array}
$$

where $\hat{C}, \hat{D} \notin \mathsf{BC}_{\mathcal{T}}$ and A is a new concept name.

Fig. 1. $\mathcal{EL}$ normalization rules (from [2])

bases containing only axioms in a structured *normal form*. Every knowledge base can be transformed into a normalized one via a two-step procedure.

First the ABox is transformed into a simple ABox. An ABox $\mathcal{A}$ is a *simple ABox*, if for every concept assertion $C(a) \in \mathcal{A}$, C is a concept name. An arbitrary $\mathcal{EL}$-ABox $\mathcal{A}$ can be transformed into a simple ABox by first replacing each complex assertion $C(A)$ in $\mathcal{A}$ by $A(a)$ where A is a fresh concept name and, second, introducing $A \equiv C$ into the TBox.

After this step, the TBox is transformed into a normal form as well. For a concept description C let $\mathsf{CN}(C)$ denote the set of all concept names and $\mathsf{RN}(C)$ denote the set of all role names that appear in C. The *signature of a concept description C* (denoted $\mathsf{sig}(C)$) is given by $\mathsf{CN}(C) \cup \mathsf{RN}(C)$. Similarly, the set of concept (respectively role) names that appear in a TBox is denoted by $\mathsf{CN}(\mathcal{T})$ (respectively $\mathsf{RN}(\mathcal{T})$). The *signature of a TBox $\mathcal{T}$* (denoted $\mathsf{sig}(\mathcal{T})$) is $\mathsf{CN}(\mathcal{T}) \cup \mathsf{RN}(\mathcal{T})$. The *signature of an ABox $\mathcal{A}$* (denoted $\mathsf{sig}(\mathcal{A})$) is the set of concept (role / individual) names $\mathsf{CN}(\mathcal{A})$ ($\mathsf{RN}(\mathcal{A})$/$\mathsf{IN}(\mathcal{A})$ resp.) that appear in $\mathcal{A}$. The signature of a knowledge base $\mathcal{K}$ = ($\mathcal{T}$, $\mathcal{A}$) (denoted $\mathsf{sig}(\mathcal{K})$) is $\mathsf{sig}(\mathcal{T}) \cup \mathsf{sig}(\mathcal{A})$.

An $\mathcal{EL}$-TBox $\mathcal{T}$ is in *normal form* if all concept axioms have one of the following forms, where $C_1, C_2 \in \mathsf{sig}(\mathcal{T})$ and $D \in \mathsf{sig}(\mathcal{T}) \cup \{\bot\}$:

$$C_1 \sqsubseteq D, \quad C_1 \sqcap C_2 \sqsubseteq D, \quad C_1 \sqsubseteq \exists r.C_2 \quad \text{or} \quad \exists r.C_1 \sqsubseteq D.$$

Any $\mathcal{EL}$-TBox can be transformed into normal form by introducing new concept names and by applying the normalization rules displayed in Figure 1 exhaustively, where $\mathsf{BC}_{\mathcal{T}}$ is the set containing all the concept names appearing in $\mathcal{T}$ and the concept $\top$. These rules replace the GCI on the left-hand side of the rules with the set of GCIs on the right-hand side. Clearly, for a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ the signature of $\mathcal{A}$ may be changed only during the first of the two normalization steps and the signature of $\mathcal{T}$ may be extended during both of them. The normalization of the knowledge base can be done in linear time.

The completion algorithm for instance checking is based on the one for classifying $\mathcal{EL}$-TBoxes introduced in [2]. The completion algorithm constructs a representation of the minimal model of $\mathcal{K}$. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a normalized $\mathcal{EL}$-knowledge base, i.e., with a simple ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ in normal form. The completion algorithm works on four kinds of *completion sets*: $S(a), S(a,r), S(C)$ and $S(C,r)$ for each $a \in \mathsf{IN}(\mathcal{A})$, $C \in \mathsf{CN}(\mathcal{K})$ and $r \in \mathsf{RN}(\mathcal{K})$. These completion sets contain concept names

CR1 If $C \in S(X)$, $C \sqsubseteq D \in \mathcal{T}$, and $D \notin S(X)$
then $S(X) := S(X) \cup \{D\}$

CR2 If $C_1, C_2 \in S(X)$, $C_1 \sqcap C_2 \sqsubseteq D \in \mathcal{T}$, and $D \notin S(X)$
then $S(X) := S(X) \cup \{D\}$

CR3 If $C \in S(X)$, $C \sqsubseteq \exists r.D \in \mathcal{T}$, and $D \notin S(X, r)$
then $S(X, r) := S(X, r) \cup \{D\}$

CR4 If $Y \in S(X, r)$, $C \in S(Y)$, $\exists r.C \sqsubseteq D \in \mathcal{T}$, and
$D \notin S(X)$ then $S(X) := S(X) \cup \{D\}$

Fig. 2. $\mathcal{EL}$ completion rules

from $\mathsf{CN}(\mathcal{K})$. Intuitively, the completion rules make implicit subsumption and instance relationships explicit in the following sense:

- $D \in S(C)$ implies that $C \sqsubseteq_{\mathcal{T}} D$,
- $D \in S(C, r)$ implies that $C \sqsubseteq_{\mathcal{T}} \exists r.D$.
- $D \in S(a)$ implies that a is an instance of D w.r.t. $\mathcal{K}$,
- $D \in S(a, r)$ implies that a is an instance of $\exists r.D$ w.r.t. $\mathcal{K}$.

$\mathsf{S}_{\mathcal{K}}$ denotes the set of all completion sets of a normalized $\mathcal{K}$. The completion sets are initialized for each $a \in \mathsf{IN}(\mathcal{A})$ and each $C \in \mathsf{CN}(\mathcal{K})$ as follows:

- $S(C) := \{C, \top\}$ for each $C \in \mathsf{CN}(\mathcal{K})$,
- $S(C, r) := \emptyset$ for each $r \in \mathsf{RN}(\mathcal{K})$,
- $S(a) := \{C \in \mathsf{CN}(\mathcal{A}) \mid C(a) \text{ appears in } \mathcal{A}\} \cup \{\top\}$, and
- $S(a, r) := \{b \in \mathsf{IN}(\mathcal{A}) \mid r(a, b) \text{ appears in } \mathcal{A}\}$ for each $r \in \mathsf{RN}(\mathcal{K})$.

Then these sets are extended by applying the completion rules shown in Figure 2 until no more rule applies. In these rules X and Y can refer to concept or individual names, while C, C_1, C_2 and D are concept names and r is a role name. After the completion has terminated, the following relations hold between an individual a, a role r and named concepts A and B:

- subsumption relation between A and B from $\mathcal{K}$ holds iff $B \in S(A)$
- instance relation between a and B from $\mathcal{K}$ holds iff $B \in S(a)$,

as shown in [2]. To decide the initial query: $\mathcal{K} \models D(a)$, one has to test whether A_q appears in $S(a)$. In fact, instance queries for all individuals and all named concepts from the knowledge base can be answered from the resulting completion sets; the completion algorithm does not only perform one instance check, but complete ABox realization. The completion algorithm runs in polynomial time in size of the knowledge base.

3.2 The Completion Algorithm for Prob-$\mathcal{EL}_c^{01}$

Before describing the completion algorithm for Prob-$\mathcal{EL}_c^{01}$, we modify the notion of basic concepts. The set $\mathsf{BC}_{\mathcal{T}}$ of Prob-$\mathcal{EL}_c^{01}$ *basic concepts* for a knowledge base $\mathcal{K}$ is the smallest set that contains

PR1	If $C' \in S_*(X, v)$ and $C' \sqsubseteq D \in \mathcal{T}$, then $S_*(X, v) := S_*(X, v) \cup \{D\}$
PR2	If $C_1, C_2 \in S_*(X, v)$ and $C_1 \sqcap C_2 \sqsubseteq D \in \mathcal{T}$, then $S_*(X, v) := S_*(X, v) \cup \{D\}$
PR3	If $C' \in S_*(X, v)$ and $C' \sqsubseteq \exists r.D \in \mathcal{T}$, then $S_*(X, r, v) := S_*(X, r, v) \cup \{D\}$
PR4	If $D \in S_*(X, r, v)$, $D' \in S_{\gamma(v)}(D, \gamma(v))$ and $\exists r.D' \sqsubseteq E \in \mathcal{T}$, then $S_*(X, v) := S_*(X, v) \cup \{E\}$
PR5	If $P_{>0}A \in S_*(X, v)$, then $S_*(X, P_{>0}A) := S_*(X, P_{>0}A) \cup \{A\}$
PR6	If $P_{=1}A \in S_*(X, v)$, $v \neq 0$, then $S_*(X, v) := S_*(X, v) \cup \{A\}$
PR7	If $A \in S_*(X, v)$ and $v \neq 0$, $P_{>0}A \in \mathcal{P}_0^{\mathcal{T}}$, then $S_*(X, v') := S_*(X, v') \cup \{P_{>0}A\}$
PR8	If $A \in S_*(X, 1)$ and $P_{=1}A \in \mathcal{P}_1^{\mathcal{T}}$, then $S_*(X, v) := S_*(X, v) \cup \{P_{=1}A\}$
PR9	If $r(a, b) \in \mathcal{A}, C \in S(b, 0), \exists r.C \sqsubseteq D \in \mathcal{T}$, then $S(a, 0) := S(a, 0) \cup \{D\}$
PR10	If $P_{>0}r(a, b) \in \mathcal{A}$, $C \in S(b, P_{>0}r(a, b))$ and $\exists r.C \sqsubseteq D \in \mathcal{T}$, then $S(a, P_{>0}r(a, b)) := S(a, P_{>0}r(a, b)) \cup \{D\}$
PR11	If $P_{=1}r(a, b) \in \mathcal{A}$, $C \in S(b, v)$ with $v \neq 0$ and $\exists r.C \sqsubseteq D \in \mathcal{T}$, then $S(a, v) := S(a, v) \cup \{D\}$

Fig. 3. Prob-$\mathcal{EL}_c^{01}$ completion rules

1. the concept $\top$,
2. all concept names used in $\mathcal{K}$, and
3. all concepts of the form $P_{>0}A$ or $P_{=1}A$,

where A is a concept name in $\mathcal{K}$. A Prob-$\mathcal{EL}_c^{01}$-TBox $\mathcal{T}$ is in *normal form* if all its axioms are of one of the following forms

$$C \sqsubseteq D, \quad C_1 \sqcap C_2 \sqsubseteq D, \quad C \sqsubseteq \exists r.A, \quad \exists r.A \sqsubseteq D,$$

where $C, C_1, C_2, D \in \mathsf{BC}_{\mathcal{T}}$ and A is a concept name. The normalization rules in Figure 1 can also be used to transform a Prob-$\mathcal{EL}_c^{01}$-TBox into this extended normal form. We still assume that the ABox $\mathcal{A}$ is a simple ABox; that is, for all assertions $C(a)$ in $\mathcal{A}$, C is a concept name. We denote as $\mathcal{P}_0^{\mathcal{T}}$ and $\mathcal{P}_1^{\mathcal{T}}$ the set of all concepts of the form $P_{>0}A$ and $P_{=1}A$ respectively, occurring in a normalized knowledge base $\mathcal{K}$. Analogously, $\mathcal{R}_0^{\mathcal{T}}$ denotes the set of all assertions of the form $P_{>0}r(a, b)$ appearing in $\mathcal{K}$.

The completion algorithm for Prob-$\mathcal{EL}_c^{01}$ follows the same idea as the algorithm for $\mathcal{EL}$, but uses several completion sets to deal with the information of what needs to be satisfied in the different worlds of a model. Intuitively, we will build a general description of all models, using the set of worlds $V := \{0, \varepsilon, 1\} \cup \mathcal{P}_0^{\mathcal{T}} \cup \mathcal{R}_0^{\mathcal{T}}$, where the probability distribution μ assigns a probability of 0 to the world 0, and the uniform probability $1/(|V|-1)$ to all other worlds. The main idea is that the world 1 will include all the entailments that hold with probability 1, and ε those that hold with probability greater than 0.

For each individual name a, concept name A, role name r and world v, we store the completion sets $S_0(A, v)$, $S_\varepsilon(A, v)$, $S_0(A, r, v)$, $S_\varepsilon(A, r, v)$, $S(a, v)$, and $S(a, r, v)$.

The algorithm initializes the sets as follows for every $A \in \mathsf{BC}_{\mathcal{T}}, r \in \mathsf{RN}(\mathcal{K})$, and $a \in \mathsf{IN}(\mathcal{A})$:

- $S_0(A, 0) = \{\top, A\}$ and $S_0(A, v) = \{\top\}$ for all $v \in V \setminus \{0\}$,
- $S_\varepsilon(A, \varepsilon) = \{\top, A\}$ and $S_\varepsilon(A, v) = \{\top\}$ for all $v \in V \setminus \{\varepsilon\}$,
- $S(a, 0) = \{\top\} \cup \{A \mid A(a) \in \mathcal{A}\}$, $S(a, v) = \{\top\}$ for all $v \neq 0$,
- $S_0(A, r, v) = S_\varepsilon(A, r, v) = \emptyset$ for all $v \in V$, $S(a, r, v) = \emptyset$ for $v \neq 0$,
- $S(a, r, 0) = \{b \in \mathsf{IN}(\mathcal{A}) \mid r(a, b) \in \mathcal{A}\}$.

These sets are then extended by exhaustively applying the rules shown in Figure 3, where X ranges over $\mathsf{BC}_\mathcal{T} \cup \mathsf{IN}(\mathcal{A})$, $S_*(X, v)$ stands for $S(X, v)$ if X is an individual and for $S_0(X, v), S_\varepsilon(X, v)$ if $X \in \mathsf{BC}_\mathcal{T}$, and $\gamma : V \to \{0, \varepsilon\}$ is defined by $\gamma(0) = 0$, and $\gamma(v) = \varepsilon$ for all $v \in V \setminus \{0\}$.

This algorithm terminates in polynomial time. After termination, the completion sets store all the information necessary to decide subsumption of concept names, as well as checking whether an individual is an instance of a given concept name [15]. For the former decision, it holds that for every pair A, B of concept names: $B \in S_0(A, 0)$ iff $A \sqsubseteq_\mathcal{K} B$. In the case of instance checking, we have that $\mathcal{K} \models A(a)$ iff $A \in S(a, 0)$.

4 Computing the k-MSC using Completion

The msc was first investigated for $\mathcal{EL}$-concept descriptions and w.r.t. unfoldable TBoxes and possibly cyclic ABoxes in [13]. It was shown that the msc does not need to exists for cyclic ABoxes. Consider the ABox $\mathcal{A} = \{r(a, a), C(a)\}$. The msc of a is then

$$C \sqcap \exists r.(C \sqcap \exists r.(C \sqcap \exists r.(C \sqcap \cdots$$

and cannot be expressed by a finite concept description. For cyclic TBoxes it has been shown in [1] that the msc does not need to exists even if the ABox is acyclic.

To avoid infinite nestings in presence of cyclic ABoxes it was proposed in [13] to limit the role-depth of the concept description to be computed. This limitation yields an approximation of the msc, which is still a concept description with the input individual as an instance, but it does not need to be the least one (w.r.t. $\sqsubseteq$) with this property. We follow this idea to compute approximations of the msc also in presence of general TBoxes.

The *role-depth* of a concept description C (denoted $rd(C)$) is the maximal number of nested quantifiers of C. This allows us to define the msc with limited role-depth for $\mathcal{EL}$.

Definition 2 (role-depth bounded $\mathcal{EL}$-msc). *Let $\mathcal{K}$ =($\mathcal{T}$, $\mathcal{A}$) be an $\mathcal{EL}$-knowledge base and a an individual in $\mathcal{A}$ and $k \in \mathbb{N}$. Then the $\mathcal{EL}$-concept description C is the* role-depth bounded $\mathcal{EL}$-most specific concept *of a w.r.t. $\mathcal{K}$ and role-depth k (written k-$msc_\mathcal{K}(a)$) iff*

1. *$rd(C) \leq k$,*
2. *$\mathcal{K} \models C(a)$, and*
3. *for all $\mathcal{EL}$-concept descriptions E with $rd(E) \leq k$ holds: $\mathcal{K} \models E(a)$ implies $C \sqsubseteq_\mathcal{T} E$.*

Notice that in case the exact msc has a role-depth less or equal to k the role-depth bounded msc is the exact msc.

Example 3. As an example we consider the labeled knowledge base $\mathcal{K}_{ex} = (\mathcal{T}_{ex}, \mathcal{A}_{ex})$. In this labeled knowledge base each axiom and assertion is associated with a label (printed in the same line), which will be used later.

$$\begin{array}{rlll} \mathcal{T}_{ex} = \{ & \exists r.\top \sqsubseteq A, & ax1 \\ & B \sqsubseteq \exists r.C, & ax2 \\ & D \sqsubseteq E\} & ax3 \end{array} \quad \text{and} \quad \begin{array}{rll} \mathcal{A}_{ex} = \{ & B(a), & as1 \\ & D(b), & as2 \\ & r(a,b), & as3 \\ & s(a,c), & as4 \\ & r(c,a)\ \} & as5 \end{array}$$

Obviously the ABox $\mathcal{A}_{ex}$ is cyclic due to the last two assertions. Note, that c is an instance of A due to $as5$ and $ax1$. Now, for $k = 3$ we obtain the following role-depth bounded msc for a:

$$\begin{aligned} 3\text{-}msc_{\mathcal{K}_{ex}}(a) = \ & B \sqcap \\ & \exists r.D \sqcap \\ & \exists s.(A \sqcap \exists r.(B \sqcap \exists r.D \sqcap \exists s.A))). \end{aligned}$$

Next we describe how to obtain the k-msc in general.

4.1 Computing the k-msc in $\mathcal{EL}$ by Completion

The computation of the msc relies on a characterization of the instance relation. While in earlier works this was given by homomorphisms [13] or simulations [1] between graph representations of the knowledge base and the concept in question, we use the completion algorithm as such a characterization. Moreover, we construct the msc by traversing the completion sets to "collect" the msc. More precisely, the set of completion sets encodes a graph structure, where the sets $S(X)$ are the nodes and the sets $S(X, r)$ encode the edges. Traversing this graph structure, one can construct an $\mathcal{EL}$-concept. To obtain a finite concept in the presence of cyclic ABoxes or TBoxes one can limit the number of edges than can be traversed during this construction.

Definition 4 (traversal concept). *Let $\mathcal{K}$ be an $\mathcal{EL}$-knowledge base, $\mathcal{K}'$ be its normalized form,* $\mathsf{S}_\mathcal{K}$ *the completion set obtained from $\mathcal{K}$ and $k \in \mathbb{N}$. Then the* traversal concept of a named concept A *(denoted k-$\mathsf{C}_{\mathsf{S}_\mathcal{K}}(A)$) with* $\mathsf{sig}(A) \subseteq \mathsf{sig}(\mathcal{K}')$ *is the concept obtained from executing the procedure call* traversal-concept-c*(A, $\mathsf{S}_\mathcal{K}$, k) shown in Algorithm 1.*

The traversal concept of an individual a *(denoted k-$\mathsf{C}_{\mathsf{S}_\mathcal{K}}(a)$) with $a \in$* $\mathsf{sig}(\mathcal{K})$ *is the concept description obtained from executing the procedure call* traversal-concept-i*(a,* $\mathsf{S}_\mathcal{K}$*, k) shown in Algorithm 1.*

The idea is that the traversal concept of an individual yields its msc. However, the traversal concept contains names from $\mathsf{sig}(\mathcal{K}') \setminus \mathsf{sig}(\mathcal{K})$, i.e., concept names that were introduced during normalization—we call this kind of concept names *normalization names* in the following. The returned msc should be formulated w.r.t. the signature of the original knowledge base, thus the normalization names need to be removed or replaced.

Algorithm 1 Computation of a role-depth bounded $\mathcal{EL}$-msc.

```
Procedure k-msc $(a, \mathcal{K}, k)$
Input: $a$: individual from $\mathcal{K}$; $\mathcal{K}$ =($\mathcal{T}$, $\mathcal{A}$) an $\mathcal{EL}$-knowledge base; $k \in \mathbb{N}$
Output: role-depth bounded $\mathcal{EL}$-msc of $a$ w.r.t. $\mathcal{K}$ and $k$.
1: $(\mathcal{T}', \mathcal{A}')$ := simplify-ABox($\mathcal{T}$, $\mathcal{A}$)
2: $\mathcal{K}'$ := (normalize($\mathcal{T}'$), $\mathcal{A}'$)
3: $\mathsf{S}_{\mathcal{K}}$ := apply-completion-rules($\mathcal{K}$)
4: return Remove-normalization-names ( traversal-concept-i$(a, \mathsf{S}_{\mathcal{K}}, k)$)

Procedure traversal-concept-i ($a$, S, $k$)
Input: $a$: individual name from $\mathcal{K}$; S: set of completion sets; $k \in \mathbb{N}$
Output: role-depth traversal concept (w.r.t. $\mathcal{K}$) and $k$.
1: if $k = 0$ then return $\bigsqcap_{A \,\in\, S(a)} A$
2: else return $\bigsqcap_{A \,\in\, \mathsf{S}(a)} A \sqcap$
       $\bigsqcap_{r \in \mathsf{RN}(\mathcal{K}')} \bigsqcap_{A \,\in\, \mathsf{CN}(\mathcal{K}') \cap S(a,r)} \exists r.$ traversal-concept-c $(A, \mathsf{S}, k-1) \sqcap$
       $\bigsqcap_{r \in \mathsf{RN}(\mathcal{K}')} \bigsqcap_{b \,\in\, \mathsf{IN}(\mathcal{K}') \cap S(a,r)} \exists r.$ traversal-concept-i $(b, \mathsf{S}, k-1)$
3: end if

Procedure traversal-concept-c ($A$, S, $k$)
Input: $A$: concept name from $\mathcal{K}'$; S: set of completion sets; $k \in \mathbb{N}$
Output: role-depth bounded traversal concept.
1: if $k = 0$ then return $\bigsqcap_{B \in S(A)} B$
2: else return $\bigsqcap_{B \in S(A)} B \sqcap \bigsqcap_{r \in \mathsf{RN}(\mathcal{K}')} \bigsqcap_{B \in S(A,r)} \exists r.$traversal-concept-c $(B, \mathsf{S}, k-1)$
3: end if
```

Lemma 5. *Let $\mathcal{K}$ be an $\mathcal{EL}$-knowledge base, $\mathcal{K}'$ its normalized version, $\mathsf{S}_{\mathcal{K}}$ be the set of completion sets obtained for $\mathcal{K}$, $k \in \mathbb{N}$ a natural number and $a \in \mathsf{IN}(\mathcal{K})$. If $C = k$-$\mathsf{C}_{\mathsf{S}_{\mathcal{K}}}(a)$ and $\widehat{C}$ is obtained from C by removing the normalization names, then*

$$\mathcal{K}' \models C(a) \textit{ iff } \mathcal{K} \models \widehat{C}(a).$$

This lemma guarantees that removing the normalization names from the traversal concept preserves the instance relationships. Intuitively, this lemma holds since the construction of the traversal concept conjoins exhaustively all named subsumers and all subsuming existential restrictions to a normalization name up to the role-depth bound. Thus removing the normalization name does not change the extension of the conjunction. The proof can be found in [18].

The procedure k-msc uses an individual a from a knowledge base $\mathcal{K}$, the knowledge base $\mathcal{K}$ itself and a number k for the role depth-bound as parameters. It first performs the two normalization steps on $\mathcal{K}$, then applies the completion rules from Figure 2 to the normalized knowledge base $\mathcal{K}'$, and then stores the set of completion sets in $\mathsf{S}_{\mathcal{K}}$. Afterwards it computes the traversal-concept of a from $\mathsf{S}_{\mathcal{K}}$ w.r.t. role-depth bound k. In a post-processing step it applies Remove-normalization-names to the traversal concept obtained in the previous step.

Example 6. We use the knowledge base from Example 3, to apply the algorithm k-msc to the individual a from $\mathcal{A}_{ex}$ again for $k = 3$. Since the TBox $\mathcal{T}_{ex}$ is in normal form and the ABox $\mathcal{A}_{ex}$ is simple, completion can be applied directly. After completion we have the following elements in the completion sets:

$$\begin{array}{lll} S(A) = \{\top, A\} & & S(B,r) = \{\top, C\} \\ S(B) = \{\top, A, B\} & S(a) = \{\top, A, B\} & S(a,r) = \{\top, D, E\} \\ S(C) = \{\top, C\} & S(b) = \{\top, D, E\} & S(a,s) = \{\top, A\} \\ S(D) = \{\top, D, E\} & S(c) = \{\top, A\} & S(c,r) = \{\top, A\} \end{array}$$

The here omitted completion sets do not change after initialization and are empty. We obtain:

$$\begin{aligned} \mathsf{k\text{-}msc}(a, \mathcal{K}_{ex}, 3) = \; & \top \sqcap A \sqcap B \sqcap \\ & \exists r.(\top \sqcap D \sqcap E) \sqcap \\ & \exists s.(\top \sqcap A \sqcap \exists r.(\top \sqcap A \sqcap B \sqcap \exists r.(\top \sqcap D \sqcap E) \sqcap \exists s.(\top \sqcap A))). \end{aligned}$$

The resulting concept description is larger than the $k\text{-}msc$ derived in Example 3, since all the elements from the completion set are conjoined to the result concept description in traversal-concept-i and traversal-concept-c. However, it is easy to see that the result is a concept description equivalent to the $k\text{-}msc$ w.r.t. $\mathcal{K}_{ex}$.

Obviously, the concept description returned from the procedure k-msc has a role-depth less or equal to k. Thus the first condition of Definition 2 is fulfilled. As we prove next, the concept description obtained from the procedure k-msc fulfills also the second condition from Definition 2.

Lemma 7. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{EL}$-knowledge base and a an individual in $\mathcal{A}$ and $k \in \mathbb{N}$. If $C =$ k-msc$(a, \mathcal{K}, k)$, then $\mathcal{K} \models C(a)$.*

The claim can be shown by induction on k. Each name in C is from a completion set of (1) an individual or (2) a concept, which is connected via existential restrictions to an individual. The full proof can be found in [18].

Lemma 8. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{EL}$-knowledge base, a an individual appearing in $\mathcal{A}$, and $k \in \mathbb{N}$. If $C =$ k-msc$(a, \mathcal{K}, k)$, then for every $\mathcal{EL}$-concept description E with $rd(E) \leq k$ the following holds: $\mathcal{K} \models E(a)$ implies $C \sqsubseteq_{\mathcal{T}} E$.*

Again, the full proof can be found in [18]. Together, these two Lemmas yield the correctness of the overall procedure.

Theorem 9. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{EL}$-knowledge base and a an individual in $\mathcal{A}$ and $k \in \mathbb{N}$.*
Then k-msc$(a, \mathcal{K}, k) \equiv k\text{-}msc_{\mathcal{K}}(a)$.

It is important to notice that, while the completion sets can be computed in polynomial time, the $k\text{-}msc$ can grow exponential in the size of the knowledge base. In addition to that and as the example already indicated, the concept description obtained from k-msc contains a lot of redundant information and thus is quite larger. However for practical usability it is necessary to rewrite the concept to an equivalent, but smaller one. A heuristic for this has been proposed in [16]. The algorithm and the rewriting heuristic are implemented in the GEL system[1].

[1] See http://gen-el.sourceforge.net/

4.2 Computing the k-msc in Prob-$\mathcal{EL}_c^{01}$ by Completion

The role-bounded msc for a Prob-$\mathcal{EL}_c^{01}$-knowledge base can be computed in a similar fashion to the one described before for $\mathcal{EL}$. The knowledge base is first normalized and the completion procedure is executed to obtain all the completion sets.

In order to compute the msc, we simply accumulate all concepts to which the individual a belongs, given the information stored in the completion sets. This process needs to be done recursively in order to account for both, the successors of a explicitly encoded in the ABox, and the nesting of existential restrictions masked by normalization names. In the following we use the abbreviation $S^{>0}(a,r) := \bigcup_{v \in V \setminus \{0\}} S(a,r,v)$. We then define $\mathsf{traversal\text{-}concept\text{-}i}(a, \mathsf{S}, k)$ as

$$\begin{aligned}\bigsqcap_{B \in S(a,0)} B \sqcap \bigsqcap_{r \in \mathsf{RN}(\mathcal{K}'')} \Big(\bigsqcap_{r(a,b) \in \mathcal{K}''} & \exists r.\mathsf{traversal\text{-}concept\text{-}i}(b, \mathsf{S}, k-1) \sqcap \\ \bigsqcap_{B \in \mathsf{CN}(\mathcal{K}'') \cap S(a,r,0)} & \exists r.\mathsf{traversal\text{-}concept\text{-}c}(B, \mathsf{S}, k-1) \sqcap \\ \bigsqcap_{B \in \mathsf{CN}(\mathcal{K}'') \cap S(a,r,1)} & P_{=1}(\exists r.\mathsf{traversal\text{-}concept\text{-}c}(B, \mathsf{S}, k-1)) \sqcap \\ \bigsqcap_{B \in \mathsf{CN}(\mathcal{K}'') \cap S^{>0}(a,r)} & P_{>0}(\exists r.\mathsf{traversal\text{-}concept\text{-}c}(B, \mathsf{S}, k-1))\Big),\end{aligned}$$

where $\mathsf{traversal\text{-}concept\text{-}c}(B, \mathsf{S}, k+1)$ is

$$\begin{aligned}\bigsqcap_{C \in S_0(B,0)} B \sqcap \bigsqcap_{r \in \mathsf{RN}} \Big(\bigsqcap_{C \in S_0(B,r,0)} & \exists r.\mathsf{traversal\text{-}concept\text{-}c}(C, \mathsf{S}, k) \sqcap \\ \bigsqcap_{C \in S_0(B,r,1)} & P_{=1}(\exists r.\mathsf{traversal\text{-}concept\text{-}c}(C, \mathsf{S}, k)) \sqcap \\ \bigsqcap_{C \in S_0^{>0}(B,r)} & P_{>0}(\exists r.\mathsf{traversal\text{-}concept\text{-}c}(C, \mathsf{S}, k))\Big)\end{aligned}$$

and $\mathsf{traversal\text{-}concept\text{-}c}(B, \mathsf{S}, 0) = \bigsqcap_{C \in S_0(B,0)} B$. Once the traversal concept has been computed, it is possible to remove all normalization names preserving the instance relation, which gives us the msc in the original signature of $\mathcal{K}$. As in the case for $\mathcal{EL}$, the proof of correctness of this method can be found in [18].

Theorem 10. *Let $\mathcal{K}$ a Prob-$\mathcal{EL}_c^{01}$-knowledge base, $a \in \mathsf{IN}(\mathcal{A})$, and $k \in \mathbb{N}$; then* $\mathsf{Remove\text{-}normalization\text{-}names}(\mathsf{traversal\text{-}concept\text{-}i}(a, \mathsf{S}, k)) \equiv k\text{-}msc_{\mathcal{K}}(a)$.

5 Computing Explanations for Instance Relations in Prob-$\mathcal{EL}_c^{01}$

By definition, an individual a is always an instance of its (role-depth bounded) msc. However, it is not always obvious why this is the case. We thus provide a method for describing the axiomatic causes for a to be an instance of a concept name A.

Definition 11 (MinA). *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an Prob-$\mathcal{EL}_c^{01}$-knowledge base, a an individual in $\mathcal{A}$ and A a concept name such that $\mathcal{K} \models A(a)$. A* minimal axiom set (MinA) *for $\mathcal{K}$ w.r.t. $A(a)$ is a sub-knowledge base $\mathcal{K}' = (\mathcal{S}, \mathcal{B})$, with $\mathcal{S} \subseteq \mathcal{T}, \mathcal{B} \subseteq \mathcal{A}$ such that*

– *$\mathcal{K}' \models A(a)$ and*
– *for all strict subsets $\mathcal{S}' \subset \mathcal{S}, \mathcal{B}' \subset \mathcal{B}$, it holds that (i) $(\mathcal{S}', \mathcal{B}) \not\models A(a)$ and (ii) $(\mathcal{S}, \mathcal{B}') \not\models A(a)$.*

Intuitively, a MinA is a sub-ontology that still entails the instance relationship between a and A, and that is minimal (w.r.t. set inclusion) with this property. As the following example illustrates there may be several MinAs for one consequence.

Example 12. Continuing with our running example, we have that $\mathcal{K}_{ex} \models A(a)$, and there are two MinAs for $\mathcal{K}_{ex}$ w.r.t. this instance relationship, namely

$$\mathcal{K}_1 = (\{\exists r.\top \sqsubseteq A, B \sqsubseteq \exists r.C\}, \{B(a)\}), \text{ and}$$
$$\mathcal{K}_2 = (\{\exists r.\top \sqsubseteq A\}, \{r(a,b)\}).$$

It is a simple task to verify that indeed these two knowledge bases entail $A(a)$, and that they satisfy the minimality requirement w.r.t. set inclusion.

The process of computing MinAs is called *pinpointing*. As it has been done before for other kinds of reasoning problems, we show that the completion algorithm for Prob-$\mathcal{EL}_c^{01}$ can be modified into a *pinpointing algorithm*. Rather than directly computing the MinAs, we will construct a monotone Boolean formula—called the *pinpointing formula*—that encodes all these MinAs. To define this formula, we first assume that every axiom and every assertion α in $\mathcal{K}$ is labeled with a *unique* propositional variable $\mathsf{lab}(\alpha)$ and denote as $\mathsf{lab}(\mathcal{K})$ the set of all propositional variables labeling axioms and assertions in $\mathcal{K}$. A *monotone Boolean formula* over $\mathsf{lab}(\mathcal{K})$ is a Boolean formula that uses only variables from $\mathsf{lab}(\mathcal{K})$, the binary connectives conjunction ($\wedge$) and disjunction ($\vee$), and the constant t (for "truth"). As customary in propositional logic, we identify a *valuation* with the set of propositional variables that it makes true. Finally, given a valuation $\mathcal{V} \subseteq \mathsf{lab}(\mathcal{K})$, we define

$$\mathcal{K}_\mathcal{V} := (\{\alpha \in \mathcal{T} \mid \mathsf{lab}(\alpha) \in \mathcal{V}\}, \{\alpha \in \mathcal{A} \mid \mathsf{lab}(\alpha) \in \mathcal{V}\}).$$

Definition 13 (pinpointing formula). *Given a Prob-$\mathcal{EL}_c^{01}$-knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, an individual name a occurring in $\mathcal{A}$ and a concept name A, the monotone Boolean formula ϕ over $\mathsf{lab}(\mathcal{K})$ is a* pinpointing formula *for $\mathcal{K}$ w.r.t. $A(a)$ if for every valuation $\mathcal{V} \subseteq \mathsf{lab}(\mathcal{K})$ it holds that*

$$\mathcal{K}_\mathcal{V} \models A(a) \textit{ iff } \mathcal{V} \textit{ satisfies } \phi.$$

Example 14. Recall that we have given every axiom and assertion of $\mathcal{K}_{ex}$ a unique label, depicted in Example 3. Hence, for instance $\mathsf{lab}(\exists r.\top \sqsubseteq A) = ax1$. The following is a pinpointing formula for $\mathcal{K}_{ex}$ w.r.t. $A(a)$:

$$ax1 \wedge (as3 \vee (ax2 \wedge as1)).$$

The MinAs for an instance relation can be obtained from the pinpointing formula ϕ by computing the minimal valuations that satisfy ϕ.

Proposition 15. *If ϕ is a pinpointing formula for $\mathcal{K}$ w.r.t. $A(a)$, then the set*

$$\{\mathcal{K}_{\mathcal{V}} \mid \mathcal{V} \textit{ is a minimal valuation satisfying } \phi\}$$

is the set of all MinAs for $\mathcal{K}$ w.r.t. $A(a)$.

We take advantage of this proposition and describe an algorithm that computes a pinpointing formula for a given instance relationship.[2] If one is interested in the specific MinAs, it is only necessary to find the minimal valuations that satisfy this formula. This can be done by e.g. bringing the pinpointing formula to disjunctive normal form first and then removing all the non-minimal disjuncts. In general, a pinpointing formula may yield a more compact representation of the set of all MinAs, and hence be of more practical use.

We will use a so-called glass-box approach for computing pinpointing formulas for all the instance relationships that follow from a knowledge base $\mathcal{K}$. The idea is to extend the completion algorithm for deciding instances in Prob-$\mathcal{EL}_c^{01}$ with a tracing mechanism that encodes all the axiomatic causes for a consequence—in this case, either a subsumption or an instance relation—to follow. Since $\mathcal{EL}$ is a sub-logic of Prob-$\mathcal{EL}_c^{01}$ and classification can be reduced to instance checking,[3] our approach can also find the pinpointing formulas for the different subsumption relations that follow from the knowledge base. Thus, we generalize previous results on axiom-pinpointing in $\mathcal{EL}$ [9] in two ways by developing explanations also for the entailed instance relationships and include the probabilistic concept constructors from Prob-$\mathcal{EL}_c^{01}$.

In order to describe the pinpointing algorithm, we assume first that the knowledge base $\mathcal{K}$ is already in normal form; recall that our example knowledge base $\mathcal{K}_{ex}$ is in normal form. The *pinpointing extension* of the completion algorithm for Prob-$\mathcal{EL}_c^{01}$ also stores completion sets $S(a,v), S(a,r,v), S_0(C,v), S_0(A,r,v), S_\varepsilon(A,v)$, and, $S_\varepsilon(A,r,v)$ for the different individual-, and role names a, r, respectively, and basic concept A appearing in the knowledge base. However, the elements of these sets are not only concept names from $\mathsf{CN}(\mathcal{K})$ as in Section 3, but rather pairs of the form (D,φ), where $D \in \mathsf{CN}(\mathcal{K})$ and φ is a monotone Boolean formula. Intuitively, $(D,\varphi) \in S(C)$ means that D is a subsumer of C w.r.t. $\mathcal{K}$, and φ stores information of the axioms responsible for this fact. For the other three kinds of completion sets the idea is analogous.

The pinpointing algorithm initializes these completion sets as follows: for every $A \in \mathsf{BC}_{\mathcal{T}}, r \in \mathsf{RN}(\mathcal{K})$, and $a \in \mathsf{IN}(\mathcal{A})$

- $S_0(A,0) = \{(\top,\mathsf{t}),(A,\mathsf{t})\}$ and $S_0(A,v) = \{(\top,\mathsf{t})\}$ for all $v \in V \setminus \{0\}$,
- $S_\varepsilon(A,\varepsilon) = \{(\top,\mathsf{t}),(A,\mathsf{t})\}$ and $S_\varepsilon(A,v) = \{(\top,\mathsf{t})\}$ for all $v \in V \setminus \{\varepsilon\}$,
- $S(a,0) = \{(\top,\mathsf{t})\} \cup \{(A,p) \mid A(a) \in \mathcal{A}, p = \mathsf{lab}(A(a))\}$,
- $S(a,v) = \{(\top,\mathsf{t})\}$ for all $v \neq 0$,
- $S_0(A,r,v) = S_\varepsilon(A,r,v) = \emptyset$ for all $v \in V$, $S(a,r,v) = \emptyset$ for $v \neq 0$,
- $S(a,r,0) = \{(b,p) \in \mathsf{IN}(\mathcal{A}) \mid r(a,b) \in \mathcal{A}, p = \mathsf{lab}(A(a))\}$.

[2] In fact, our method produces pinpointing formulas for *all* instance relationships that follow from the knowledge base at once.

[3] Indeed, $A \sqsubseteq_{\mathcal{K}} B$ iff $\mathcal{K} \cup \{A(a)\} \models B(a)$, where a is an individual name not appearing in $\mathcal{K}$.

PpR1 If $(C', \varphi) \in S_*(X, v)$, $\alpha = C' \sqsubseteq D \in \mathcal{T}$, and $\mathsf{lab}(\alpha) = p$ then $S_*(X, v) := S_*(X, v) \uplus (D, \varphi \wedge p)$

PpR2 If $(C_1, \varphi_1), (C_2, \varphi_2) \in S_*(X, v)$, $\alpha = C_1 \sqcap C_2 \sqsubseteq D \in \mathcal{T}$, and $\mathsf{lab}(\alpha) = p$ then $S_*(X, v) := S_*(X, v) \uplus (D, \varphi_1 \wedge \varphi_2 \wedge p)$

PpR3 If $(C', \varphi) \in S_*(X, v)$, $\alpha = C' \sqsubseteq \exists r.D \in \mathcal{T}$, and $\mathsf{lab}(\alpha) = p$ then $S_*(X, r, v) := S_*(X, r, v) \uplus (D, \varphi \wedge p)$

PpR4 If $(D, \varphi) \in S_*(X, r, v)$, $(D', \varphi') \in S_{\gamma(v)}(D, \gamma(v))$, $\alpha = \exists r.D' \sqsubseteq E \in \mathcal{T}$, and $\mathsf{lab}(\alpha) = p$ then $S_*(X, v) := S_*(X, v) \uplus (E, \varphi \wedge \varphi' \wedge p)$

PpR5 If $(P_{>0}A, \varphi) \in S_*(X, v)$, then $S_*(X, P_{>0}A) := S_*(X, P_{>0}A) \uplus (A, \varphi)$

PpR6 If $(P_{=1}A, \varphi) \in S_*(X, v)$, $v \neq 0$, then $S_*(X, v) := S_*(X, v) \uplus (A, \varphi)$

PpR7 If $(A, \varphi) \in S_*(X, v)$ and $v \neq 0$, $P_{>0}A \in \mathcal{P}_0^{\mathcal{T}}$ then $S_*(X, v') := S_*(X, v') \uplus (P_{>0}A, \varphi)$

PpR8 If $(A, \varphi) \in S_*(X, 1)$ and $P_{=1}A \in \mathcal{P}_1^{\mathcal{T}}$, then $S_*(X, v) := S_*(X, v) \uplus (P_{=1}A, \varphi)$

PpR9 If $\alpha_1 = r(a, b) \in \mathcal{A}$, $(C, \varphi) \in S(b, 0)$, $\alpha_2 = \exists r.C \sqsubseteq D \in \mathcal{T}$, $\mathsf{lab}(\alpha_1) = p_1$, and $\mathsf{lab}(\alpha_2) = p_2$ then $S(a, 0) := S(a, 0) \uplus (D, \varphi \wedge p_1 \wedge p_2)$

PpR10 If $\alpha_1 = P_{>0}r(a, b) \in \mathcal{A}$, $(C, \varphi) \in S(b, P_{>0}r(a, b))$, $\alpha_2 = \exists r.C \sqsubseteq D \in \mathcal{T}$, $\mathsf{lab}(\alpha_1) = p_1$, and $\mathsf{lab}(\alpha_2) = p_2$ then $S(a, P_{>0}r(a, b)) := S(a, P_{>0}r(a, b)) \uplus (D, \varphi \wedge p_1 \wedge p_2)$

PpR11 If $\alpha_1 = P_{=1}r(a, b) \in \mathcal{A}$, $(C, \varphi) \in S(b, v)$ with $v \neq 0$, $\alpha_2 = \exists r.C \sqsubseteq D \in \mathcal{T}$, $\mathsf{lab}(\alpha_1) = p_1$, and $\mathsf{lab}(\alpha_2) = p_2$ then $S(a, v) := S(a, v) \uplus (D, \varphi \wedge p_1 \wedge p_2)$

Fig. 4. Prob-$\mathcal{EL}_c^{01}$ completion rules for axiom-pinpointing

For describing the extended completion rules, we need some more notation. For a set S and a pair (D, φ), the operation $S \uplus (D, \varphi)$ is defined as follows: if there exists a ψ such that $(D, \psi) \in S$, then $S \uplus (D, \varphi) := S \setminus \{(D, \psi)\} \cup \{(D, \psi \vee \varphi)\}$; otherwise, $S \uplus (D, \varphi) := S \cup \{(D, \varphi)\}$. In other words, if the concept name D already belongs to S with some associated formula ψ, we modify the formula by adding φ to it as a disjunct; otherwise, we simply add the pair (D, φ) to S.

The completion sets are then extended by exhaustively applying the rules shown in Figure 4, where X ranges over $\mathsf{BC}_{\mathcal{T}} \cup \mathsf{IN}(\mathcal{A})$, $S_*(X, v)$ stands for $S(X, v)$ if X is an individual and for $S_0(X, v), S_\varepsilon(X, v)$ if $X \in \mathsf{BC}_{\mathcal{T}}$, and $\gamma : V \to \{0, \varepsilon\}$ is defined by $\gamma(0) = 0$, and $\gamma(v) = \varepsilon$ for all $v \in V \setminus \{0\}$.

To ensure termination of this algorithm, the completion can only be applied if their application modifies at least one of the completion sets; that is, if either a new pair is added, or the second element of an existing pair is modified to a (strictly) more general Boolean formula. Under this applicability condition, this modified algorithm always terminates, although not necessarily in polynomial time. In fact, every completion set can contain at most as many pairs as there are concept names in $\mathcal{K}$, and hence polynomially many. Whenever the formula of a pair is changed, it is done so by generalizing it in the sense that it has more models than the previous one. As there are exponentially many models, such changes can only be done an exponential number of times. Thus, in total we can have at most exponentially many rule applications, which take each at most exponential time; that is, the pinpointing algorithm runs in exponential time in the size of $\mathcal{K}$.

As stated before, these completion sets make the subsumption and instance relationships explicit, together with a formula that describe which axioms are responsible for each of these relationships. It is easy to see that the concepts appearing in the completion sets are exactly the same that will be obtained by applying the *standard* completion rules from Section 3. We thus know that $A \sqsubseteq_{\mathcal{K}} B$ iff there is some ψ with $(B, \psi) \in S_0(A, 0)$ and $\mathcal{K} \models A(a)$ iff $(A, \psi) \in S(a, 0)$ for some monotone Boolean formula ψ. Moreover, the pinpointing algorithm maintains the following invariants:

– if $(B, \psi) \in S_0(A, 0)$, then for every valuation $\mathcal{V}$ satisfying ψ, $A \sqsubseteq_{\mathcal{K}_\mathcal{V}} B$,
– if $(A, \psi) \in S(a, 0)$, then for every valuation $\mathcal{V}$ satisfying ψ, $\mathcal{K}_\mathcal{V} \models A(a)$.

It can also be shown that when the algorithm has terminated, the converse implications also hold; this is a consequence of the results from [8].

Theorem 16. *Given a Prob-$\mathcal{EL}_c^{01}$-knowledge base in normal form, the pinpointing algorithm terminates in exponential time. After termination, the following holds for every concept name A and individual name a appearing in $\mathcal{K}$:*

if $(A, \psi) \in S(a, 0)$, then ψ is a pinpointing formula for $\mathcal{K}$ w.r.t. $A(a)$.

We have so far described how to find the MinAs of a normalized knowledge base w.r.t. instance and subsumption relations. We now show how to extend this method to deal also with non-normalized knowledge bases; that is, to obtain the MinAs referring to the original axioms of the knowledge base and not to their normalized versions. Before going into the details, it is worth noticing that the relationship between original axioms and normalized axioms is many-to-many: one axiom in the original knowledge base may produce several axioms in the normalized one, while one axiom in the normalized knowledge base can be due to the presence of several axioms from the original one. An example of the latter can be given by the two axioms $A \sqsubseteq B$, $A \sqsubseteq B \sqcap C$. The normalization rules change these axioms into $A \sqsubseteq B, A \sqsubseteq C$, but the first axiom has two sources; that is, it will appear in the normalized knowledge base whenever *any* of the two original axioms is present.

Let $\hat{\mathcal{K}}$ be an arbitrary Prob-$\mathcal{EL}_c^{01}$-knowledge base and $\mathcal{K}$ its normalized version. If ϕ is a pinpointing formula for $\mathcal{K}$ w.r.t. an instance or subsumption relation, that uses only basic concepts appearing in $\hat{\mathcal{K}}$, then we can modify ϕ into a pinpointing formula *for the original knowledge base* $\hat{\mathcal{K}}$ as follows. As in the case of normalized knowledge bases, each axiom in $\hat{\mathcal{K}}$ is associated with a unique propositional variable. Each normalized axiom in $\mathcal{K}$ has a finite number of original axioms that created it—at most as many as there were in the original knowledge base. We modify the pinpointing formula ϕ by replacing the propositional formula associated to each normalized axiom by the disjunction of the labels of all its sources. We thus obtain a new pinpointing formula that speaks of the original ontology $\hat{\mathcal{K}}$. In the above example, let $\mathsf{lab}(A \sqsubseteq B) = p_1$ and $\mathsf{lab}(A \sqsubseteq B \sqcap C) = p_2$, and suppose that the labels of the normalized ontology are $\mathsf{lab}(A \sqsubseteq B) = q_1, \mathsf{lab}(A \sqsubseteq C) = q_2$, and that the knowledge base also contains an assertion $A(a)$ with label q_3. The pinpointing formula for the normalized ontology w.r.t. $B(a)$ is $q_1 \wedge q_3$. For the original ontology, this formula is changed to $(p_1 \vee p_2) \wedge q_3$.

It is worth commenting on the execution time of the pinpointing algorithm and the complexity of finding *all* MinAs. Recall that computing all MinAs is crucial when resolving an unwanted consequence of a knowledge base. As described before, the algorithm takes exponential time to compute all instance and subsumption relations between concept names and individual names, with their respective pinpointing formulas. These formulas may be exponential in the size of the knowledge base $\mathcal{K}$, however finding one or all the minimal valuations satisfying a formula is only exponential on the number of propositional variables appearing in that formula, hence, we can compute one or all MinAs from each of these pinpointing formulas in exponential time in the size of $\mathcal{K}$. Since classification of an $\mathcal{EL}$ TBox is a special case of our setting—where the ABox $\mathcal{A}$ is empty and no probabilistic concepts are used—our algorithm yields an optimal upper bound on the complexity of pinpointing for Prob-$\mathcal{EL}_c^{01}$. Indeed, it has been shown that finding all MinAs for one subsumption relation in $\mathcal{EL}$ requires already exponential time [17]. Additionally, other kinds of tasks like finding a MinA of least cardinality or the first MinA w.r.t. some underlying ordering, can be also solved by computing the related valuations over the pinpointing formula; this is in particular beneficial, as the various optimizations developed in the SAT community, and in particular the very efficient modern SAT/SMT-solvers, can be exploited.

6 Conclusions

In this paper we have presented a practical method for computing the role-depth bounded msc in $\mathcal{EL}$- and in Prob-$\mathcal{EL}_c^{01}$- w.r.t. a general TBox or cyclic ABoxes. Our approach is based on the completion sets that are computed during realization of a knowledge base. Thus, any of the available implementations of the $\mathcal{EL}$ completion algorithm, as for instance JCEL[4] [16] can be easily extended to an implementation of the (approximative) msc computation algorithm – as it is provided in the GEL system[5]. We also showed that the same idea can be adapted for the computation of the msc in the probabilistic DL Prob-$\mathcal{EL}_c^{01}$.

Together with the completion-based computation of role-depth bounded (least) common subsumers given in [19] these results complete the bottom-up approach for general $\mathcal{EL}$- and Prob-$\mathcal{EL}_c^{01}$-knowledge bases. This approach yields a practical method to compute commonalities for differing observations regarding individuals. To the best of our knowledge this has not been investigated for DLs that can express uncertainty.

We have also applied the ideas of axiom-pinpointing to compute explanations for instance relationships that follow from a Prob-$\mathcal{EL}_c^{01}$-knowledge base. To the best of our knowledge this is also the first time that axiom-pinpointing has been applied to instance relationships, even for crisp DLs. The glass-box approach proposed modifies the computation of the completion sets to include an encoding of the axiomatic causes for a concept to be added to each set. Understanding the causes for some unexpected instance relationships is an important first step towards correcting a knowledge base,

[4] `http://jcel.sourceforge.net/`

[5] `http://gen-el.sourceforge.net/`

specially in the case of automatically generated ones, as done through the bottom-up approach described before. In general, finding out the precise axioms responsible for an unwanted consequence is a very hard task, even for experts, due to the large number of axioms available. When dealing with uncertainty, the difficulty grows, as the probabilities may interact in unexpected ways. Thus, being able to explain the consequences of a Prob-$\mathcal{EL}_c^{01}$ ontology automatically is of special importance.

References

1. F. Baader. Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In G. Gottlob and T. Walsh, editors, *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-03)*, pages 325–330. Morgan Kaufmann, 2003.
2. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*, Edinburgh, UK, 2005. Morgan-Kaufmann Publishers.
3. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope further. In K. Clark and P. F. Patel-Schneider, editors, *In Proc. of the OWLED Workshop*, 2008.
4. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
5. F. Baader, R. Küsters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. In T. Dean, editor, *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99)*, pages 96–101, Stockholm, Sweden, 1999. Morgan Kaufmann, Los Altos.
6. F. Baader, C. Lutz, and A.-Y. Turhan. Small is again Beautiful in Description Logics. *KI – Künstliche Intelligenz*, 24(1):25–33, April 2010.
7. F. Baader and R. Peñaloza. Axiom pinpointing in general tableaux. *Journal of Logic and Computation*, 20(1):5–34, 2010. Special Issue: Tableaux and Analytic Proof Methods.
8. F. Baader and R. Peñaloza. Axiom pinpointing in general tableaux. *Journal of Logic and Computation*, 20(1):5–34, 2010. Special Issue: Tableaux and Analytic Proof Methods.
9. F. Baader, R. Peñaloza, and B. Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In *Proc. of the 30th German Annual Conf. on Artificial Intelligence (KI'07)*, volume 4667 of *Lecture Notes In Artificial Intelligence*, pages 52–67, Osnabrück, Germany, 2007. Springer.
10. F. Baader and B. Suntisrivaraporn. Debugging SNOMED CT using axiom pinpointing in the description logic $\mathcal{EL}^+$. In *Proceedings of the International Conference on Representing and Sharing Knowledge Using SNOMED (KR-MED'08)*, Phoenix, Arizona, 2008.
11. S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL web ontology language reference. W3C Recommendation, February 2004. `http://www.w3.org/TR/owl-ref/`.
12. A. Kalyanpur, B. Parsia, M. Horridge, and E. Sirin. Finding all justifications of OWL DL entailments. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 267–280, 2007.
13. R. Küsters and R. Molitor. Approximating most specific concepts in description logics with existential restrictions. *AI Communications*, 15(1):47–59, 2002.
14. T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *J. Web Sem.*, 6(4):291–308, 2008.

15. C. Lutz and L. Schröder. Probabilistic description logics for subjective probabilities. In F. Lin and U. Sattler, editors, *Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-10)*, 2010.
16. J. Mendez, A. Ecke, and A.-Y. Turhan. Implementing completion-based inferences for the $\mathcal{EL}$-family. In R. Rosati, S. Rudolph, and M. Zakharyaschev, editors, *Proc. of the 2011 Description Logic Workshop (DL 2011)*, volume 745. CEUR, 2011.
17. R. Peñaloza and B. Sertkaya. On the complexity of axiom pinpointing in the el family of description logics. In F. Lin, U. Sattler, and M. Truszczynski, editors, *Proceedings of the Twelfth International Conference on Principles of Knowledge Representation and Reasoning (KR 2010)*. AAAI Press, 2010.
18. R. Peñaloza and A.-Y. Turhan. Completion-based computation of most specific concepts with limited role-depth for $\mathcal{EL}$ and prob-$\mathcal{EL}^{01}$. LTCS-Report LTCS-10-03, Chair f. Automata Theory, Inst. for Theoretical Computer Science, TU Dresden, Germany, 2010.
19. R. Peñaloza and A.-Y. Turhan. Role-depth bounded least common subsumers by completion for $\mathcal{EL}$- and Prob-$\mathcal{EL}$-TBoxes. In V. Haarslev, D. Toman, and G. Weddell, editors, *Proc. of the 2010 Description Logic Workshop (DL'10)*, 2010.
20. R. Peñaloza and A.-Y. Turhan. Towards approximative most specific concepts by completion for $\mathcal{EL}^{01}$ with subjective probabilities. In T. Lukasiewicz, R. Peñaloza, and A.-Y. Turhan, editors, *Proceedings of the First International Workshop on Uncertainty in Description Logics (UniDL'10)*, 2010.
21. R. Peñaloza and A.-Y. Turhan. A practical approach for computing generalization inferences in $\mathcal{EL}$. In M. Grobelnik and E. Simperl, editors, *Proc. of the 8th European Semantic Web Conf. (ESWC'11)*, Lecture Notes in Computer Science. Springer, 2011.
22. S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In G. Gottlob and T. Walsh, editors, *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-03)*, pages 355–362, Acapulco, Mexico, 2003. Morgan Kaufmann, Los Altos.
23. K. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with snomed-rt. *Journal of the American Medical Informatics Assoc.*, 2000. Fall Symposium Special Issue.
24. T. Springer and A.-Y. Turhan. Employing description logics in ambient intelligence for modeling and reasoning about complex situations. *Journal of Ambient Intelligence and Smart Environments*, 1(3):235–259, 2009.
25. W3C OWL Working Group. OWL 2 web ontology language document overview. W3C Recommendation, 27th October 2009. `http://www.w3.org/TR/2009/REC-owl2-overview-20091027/`.