

Model-Based Most Specific Concepts in Some Inexpressive Description Logics

Felix Distel*

TU Dresden, Germany

Abstract. Model-based most specific concepts are a non-standard reasoning service in Description Logics. They have turned out to be useful in knowledge base completion for ontologies that are written in certain extensions of \mathcal{EL} . There is indication that model-based most specific concepts can also be applied for knowledge base completion in other inexpressive description logics. We show that model-msc exist in the logics \mathcal{FL}_0 and \mathcal{FLE} with cyclic TBoxes and for $\mathcal{ALC}^{\cup*}$ with acyclic TBoxes. We provide constructions for model-msc in these three logics.

Many of the ontologies that are used today are large in scale and formulated in one of the less expressive Description Logics (DL). As ontologies become larger, tools that support engineers during the process of building and maintaining ontologies are gaining in importance. Often, the people who are in charge of building an ontology are experts in the domain of interest, rather than experts in Description Logics. They may sometimes be unable to comprehend the full consequences of a statement. Several reasoning services have been developed to assist them, such as Satisfiability Checking and Axiom Pinpointing [15]. Most of these services can be used to check the correctness of the knowledge that is already contained in the knowledge base. They cannot be used to detect what knowledge is *missing* in the knowledge base.

Knowledge base completion addresses this issue. In knowledge base completion it is assumed that there is an expert (or a group of experts) who has complete knowledge about the domain in the so-called “real world”. It is furthermore assumed that this “real world” can be represented as a model of the knowledge base. We sometimes call this model the *background model* of the knowledge base. Intuitively, a knowledge base is complete, if all questions about the background model can be answered by querying the knowledge base. Usually some kind of restriction with regard to the structure of these questions is made.

Baader and Sertkaya have developed one knowledge base completion formalism [6]. For their approach a knowledge base is complete if it can answer all questions of the form “Does the general concept inclusion (GCI) $C \sqsubseteq D$ hold (in the real world)?”. C and D cannot be arbitrary concept descriptions but must be conjunctions of concepts that are already present in the knowledge base. Rudolph’s approach (*Relational Exploration*) also asks for GCIs, but uses arbitrary concept descriptions from the DL \mathcal{FLE} [13, 14]. Like Baader and Sertkaya

* supported by Cusanuswerk e. V.

Rudolph uses an expert-assisted approach similar to attribute exploration from Formal Concept Analysis (FCA) [11]. Both systems repeatedly find GCIs that do not follow from the existing knowledge base and present them to the expert. If the expert accepts the GCI it is added to the knowledge base. If she rejects it, she is asked to add a counterexample.

Expert interaction is the most time-consuming part in the exploration process, and it is a major goal to reduce it. The problem with Rudolph's approach is that the number of expert interactions can become relatively large. To reduce expert interaction, it is important to find a small set of GCIs from which all GCIs that hold in a given background model follow. Unlike Baader and Sertkaya we still allow arbitrary GCIs from a given logic - not just GCIs formed using conjunctions of previously selected concepts.

We present model-based most specific concepts (model-msc) as a tool to reduce the number of GCIs that have to be considered. Model-msc are similar to most specific concepts for ABoxes but exhibit certain different characteristics. ABox-msc are the most specific concept describing an individual in an ABox [2, 7]. A model-msc is the most specific concept describing a subset of the domain of some model or primitive interpretation. If they exist they are unique up to equivalence, which justifies the notation $\text{mmsc}(A)$ for the model-msc of a set A .

The reason why model-msc can help to reduce expert interaction is the following. We say that $C \sqsubseteq D$ follows from $C \sqsubseteq E$ iff all models of $C \sqsubseteq E$ are models of $C \sqsubseteq D$. Knowledge base completion aims at adding a minimal set of GCIs \mathcal{B} to a knowledge base, such that all GCIs that hold in a given background model i follow from \mathcal{B} . Assume that model-msc exist for all models for a given logic. Let C be a concept description in this logic, and C^i its extension in the background model i . We shall show in this paper that every GCI $C \sqsubseteq D$ that holds in the background model i follows from a single GCI, namely $C \sqsubseteq \text{mmsc}(C^i)$. For a fixed description C there may be infinitely many GCIs $C \sqsubseteq D$ that hold in the background model i . Knowing a single GCI from which they all follow can greatly reduce the number of GCIs that need to be presented to the expert.

For the logic \mathcal{EL} with greatest fixpoint (gfp) semantics it can be shown that model-msc always exist. They can be used to construct a small set of GCIs that form a basis for the GCIs holding in some finite background model [4]. It is possible to transfer many ideas from FCA to the DL world. This can be exploited to develop a knowledge base completion formalism [5].

So model-msc have been shown to be useful in the description logic \mathcal{EL} with GFP-semantics. The results for \mathcal{EL} with GFP-semantics indicate that model-msc might also be useful for knowledge base completion in other inexpressive description logics. In this work we examine the logics \mathcal{FL}_0 , $\mathcal{FL}\mathcal{E}$ and \mathcal{ALC} . Unfortunately, there are examples of models for which model-msc do not exist for any of the three logics. However, we present extensions for each of the three logics in which they do exist. These extensions are GFP-semantics for \mathcal{FL}_0 and $\mathcal{FL}\mathcal{E}$, and the universal role and reflexive transitive closure of roles for \mathcal{ALC} . We also provide constructions for the model-msc in the extended logics.

Besides being useful in knowledge base completion, model-msc are also interesting from a theoretical perspective. By comparing them to the similar msc for ABoxes one can learn more about the implications of open-world semantics versus closed-world semantics. We shall see that model-msc are not as closely related to classical msc for ABoxes as one might think. Their computational behaviour can be different. For example model-msc need not exist in \mathcal{FL}_0 with acyclic TBoxes, but ABox-msc do exist for this logic.

1 Preliminaries

1.1 Greatest Fixpoint Semantics

The description logics that we are considering in this work are \mathcal{FL}_0 , $\mathcal{FL}\mathcal{E}$, and \mathcal{ALC} . \mathcal{FL}_0 provides for the concept constructors conjunction (\sqcap), value restrictions (\forall), and the top concept (\top). $\mathcal{FL}\mathcal{E}$ extends \mathcal{FL}_0 with existential restrictions (\exists). In addition to these constructors \mathcal{ALC} provides for disjunction (\sqcup) and negation \neg and the bottom concept (\perp).

An acyclic or unfoldable TBox is a set of equivalence statements of the form $A \equiv C$ where A is a concept name and C is a concept description in the respective logic. Every concept name may occur at most once as the right hand side of an equivalence. No concept name may be used explicitly or implicitly in its own definition. A cyclic TBox is like an acyclic TBox but without this last restriction.

An interpretation $i = (\Delta_i, \cdot^i)$ consists of a set Δ_i and a function mapping every concept name A to a subset $A^i \subseteq \Delta_i$ and every role name r to a relation $r^i \subseteq \Delta_i \times \Delta_i$. An interpretation can be extended such that it maps every concept description to a subset of Δ_i . The semantics of concept descriptions in \mathcal{FL}_0 , $\mathcal{FL}\mathcal{E}$ and \mathcal{ALC} are defined in the usual way (cf. [8] for details). An interpretation i is a model of an acyclic TBox \mathcal{T} if $A^i = C^i$ holds for all statements $A \equiv C$ in \mathcal{T} .

In the case of cyclic TBoxes several possible semantics exist [12]. The semantics that we use in this work are greatest-fixpoint semantics (or gfp-semantics for short). We denote by \mathcal{N}_p the set of primitive concept names, i.e. the set of concept names that do not occur on the left hand side of some equivalence statement. A *primitive interpretation* i is a mapping that assigns a binary relation $r^i \subseteq \Delta_i \times \Delta_i$ to every $r \in \mathcal{N}_r$ and a set $P^i \subseteq \Delta_i$ to every $P \in \mathcal{N}_p$. An interpretation j is *based on* the primitive interpretation i if it coincides with i on \mathcal{N}_r and \mathcal{N}_p and maps every defined concept name A to some $A^j \subseteq \Delta_i$. It can be shown that for the logics \mathcal{FL}_0 and $\mathcal{FL}\mathcal{E}$, for every TBox \mathcal{T} and for every primitive interpretation of \mathcal{T} there is a so-called gfp-model i among the interpretations that are based on i . We are not going to give a full definition of greatest fixpoint semantics. For more details consider for example [12].

Both for acyclic and for cyclic TBoxes a model is uniquely determined by an interpretation of the primitive concept names and role names. Therefore, we will use the terms *model* and *primitive interpretation* interchangeably and denote both by the letter i .

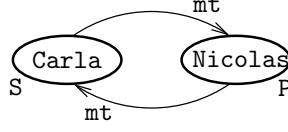


Fig. 1. A Graph Representation of a Model

Models can be represented by directed graphs with labelled vertices and edges. Given a model i the vertices of this graph are the elements of Δ_i . A vertex x is labelled with a primitive concept name P if $x \in P^i$ holds. Vertices x and y are connected by an r -labelled edge iff $(x, y) \in r^i$ holds. Let us look at an example for such a graph. As primitive concept names we use S (*Singer*) and P (*Politician*), and as role name we use mt (*isMarriedTo*). The model i_{Couple} consists of the domain $\Delta_{i_{\text{Couple}}} = \{\text{Carla}, \text{Nicolas}\}$ and a mapping $\cdot^{i_{\text{Couple}}}$ which maps S to $\{\text{Carla}\}$, P to $\{\text{Nicolas}\}$ and mt to $\{(\text{Carla}, \text{Nicolas}), (\text{Nicolas}, \text{Carla})\}$. Figure 1.1 shows a graph representation of this model.

1.2 Model-Based Most Specific Concepts

As usual we write $C \sqsubseteq_{\mathcal{T}} D$ if C and D are concept names occurring in \mathcal{T} and $C^i \subseteq D^i$ holds for all models i of \mathcal{T} . A *conservative extension* of a TBox \mathcal{T} is a TBox \mathcal{T}' such that $\mathcal{T} \subseteq \mathcal{T}'$, and if A and B are concept names used in \mathcal{T} then $A \sqsubseteq_{\mathcal{T}'} B$ iff $A \sqsubseteq_{\mathcal{T}} B$.

A model-based most specific concept for a set $X \subseteq \Delta_i$ in some model i is a concept description whose extension contains X and is minimal with respect to subsumption among all concept descriptions whose extension contains X .

Definition 1 (model-based most specific concepts). *Let \mathcal{T} be a TBox and i a model of \mathcal{T} . Let $X \subseteq \Delta_i$ be some subset of the domain of i and E a defined concept in \mathcal{T} . Then E is called the most specific concept of X iff the following conditions hold*

- $X \subseteq E^i$
- If \mathcal{T}' is a conservative extension of \mathcal{T} that uses the same primitive concept names and role names then for every defined concept F in \mathcal{T}' with $X \subseteq F^i$, it holds that $E \sqsubseteq_{\mathcal{T}'} F$.

If model-msc exist, they are unique up to equivalence. Therefore the notation $\text{mmsc}(X)$ for the model-msc of some $X \subseteq \Delta_i$ is justified.

We say that a GCI $C \sqsubseteq D$ holds in a model i if $C^i \subseteq D^i$. A GCI $C \sqsubseteq D$ follows from a set of GCIs \mathcal{L} if $C \sqsubseteq D$ holds in all models in which all GCIs from \mathcal{L} hold. Knowledge base completion attempts to add GCIs to a knowledge base such that eventually all GCIs that hold in some background model i follow from the GCIs in the knowledge base. It is desirable to keep the number of newly added GCIs small to the purpose of minimizing expert interaction.

Lemma 1. *Assume that we are dealing with a logic such that model-msc always exist, regardless of the model that is being used. If i is a model for the GCI $\{C \sqsubseteq D\}$, then $C \sqsubseteq D$ follows from $\{C \sqsubseteq \text{mmsc}(C^i)\}$.*

Proof. Since i is a model for $\{C \sqsubseteq D\}$ it holds that $C^i \sqsubseteq D^i$. The definition of model-msc tells us that

$$\text{mmsc}(C^i) \sqsubseteq D \quad (1)$$

since $\text{mmsc}(C^i)$ is the most specific concept whose extension contains C^i .

Let j be some model of $\{C \sqsubseteq \text{mmsc}(C^i)\}$, i. e. $C^j \sqsubseteq \text{mmsc}(C^i)^j$. We have assumed that the model-msc for C^j in the model j exists. Because of $C^j \sqsubseteq \text{mmsc}(C^i)^j$ it must hold that $\text{mmsc}(C^j) \sqsubseteq \text{mmsc}(C^i)$ (By Definition 1 $\text{mmsc}(C^j)$ is subsumed by any concept description whose extension in j contains C^j). Together with (1) we obtain $\text{mmsc}(C^j) \sqsubseteq D$. But then $C^j \sqsubseteq \text{mmsc}(C^j)^j \sqsubseteq D^j$ must hold. We have shown that if j is a model of $\{C \sqsubseteq \text{mmsc}(C^i)\}$ then j is also a model of $C \sqsubseteq D$. Therefore $C \sqsubseteq D$ follows from $C \sqsubseteq \text{mmsc}(C^i)$.

This shows that in logics where model-msc exist, we do not need to add more than one GCI for a given left-hand side C . Thus the number of GCIs can be reduced. One might argue that this is true whenever the used logic allows for conjunction and whenever there is a finite number of axioms with left-hand side C . However, the axioms are what is meant to be computed, and $\text{mmsc}(C^i)$ can be computed without knowing any axioms beforehand. The following sections focus on identifying logics for which model-msc do exist. Due to the page restrictions we can only sketch most of the following proofs. Where proofs or details of proofs are omitted, they can be found in [10].

2 Model-Based Most Specific Concepts in Three Different Logics

2.1 Unfoldable TBoxes

In \mathcal{FL}_0 , $\mathcal{FL}\mathcal{E}$ and \mathcal{ALC} most specific concepts do not exist if TBoxes are acyclic. Informally, this is because the models can contain cycles, which cannot be described using an acyclic terminology. Consider the model from Figure 1.1. $E_0 \equiv \mathbf{S}$, $E_1 \equiv \forall \text{mt}.\forall \text{mt}.\mathbf{S}$, $E_2 \equiv \forall \text{mt}.\forall \text{mt}.\forall \text{mt}.\mathbf{S}$, ... is a sequence of concept descriptions of increasing role depth. Still $\text{Carla} \in E_k^i$ holds for all $k \in \mathbb{N}$. If there were a model-msc M for $\{\text{Carla}\}$ then $M \sqsubseteq E_k$ for all $k \in \mathbb{N}$ would hold. But in all three logics this would mean that M could be unfolded to a concept description whose role depth is greater than that of E_k for all $k \in \mathbb{N}$. Since the role depth of the E_k goes toward infinity this is impossible.

2.2 \mathcal{FL}_0 with Cyclic TBoxes and gfp-Semantics

We thus need to find a way to describe cycles in our models. One way to do this is to use gfp-semantics. \mathcal{FL}_0 with gfp-semantics has been studied by Baader et

al. [1]. The semantics proposed there make use of *semi-automata with word transitions*. A semi-automaton with word transitions is basically a finite automaton where no initial state or final states have been specified (they will be specified later on). Baader uses word transitions, so the transition function of the automaton takes words over an alphabet Σ , not symbols from Σ as input. More formally, a semi-automaton \mathcal{A} is a triple $\mathcal{A} = (\Sigma, Q, E)$ where Σ is a finite alphabet, Q is a set of states, and $E \subseteq Q \times \Sigma^* \times Q$ is a set of labelled edges. In every step the automaton can move from a state q_1 to a state q_2 on some input $w \in \Sigma^*$ if there are words $w_1, w_2 \in \Sigma^*$ such that $w = w_1 w_2$ and $(q_1, w_1, q_2) \in E$.

A finite model i can easily be translated into a semi-automaton $\mathcal{A}_i = (\Sigma_i, Q_i, E_i)$, where $\Sigma_i = \mathcal{N}_r$, $Q_i = \Delta_i$ and E_i is the set of all labelled edges (a, r, b) such that $(a, b) \in r^i$. A state x is labelled with $P \in \mathcal{N}_p$ iff $x \in P^i$ holds.

Similarly, a cyclic \mathcal{FL}_0 -TBox \mathcal{T} can be translated into a semi-automaton $\mathcal{A}_{\mathcal{T}}$ with $\Sigma_{\mathcal{T}} = \mathcal{N}_r$. We assume that \mathcal{T} is normalized in the sense that every equivalence statement in \mathcal{T} is of the form

$$D = \forall W_1.A_1 \sqcap \dots \sqcap \forall W_k.A_k. \quad (2)$$

The states of $\mathcal{A}_{\mathcal{T}}$ are the concept names occurring in \mathcal{T} . For every defined concept D there is at most one equivalence statement of the form (2) in \mathcal{T} . $E_{\mathcal{T}}$ contains an edge (D, W_k, A_k) if this statement contains a term $\forall W_k.A_k$.

The automata $\mathcal{A}_{\mathcal{T}}$ and \mathcal{A}_i give rise to the languages $L(A, P)$, $L(x, y)$ and $L(x, P)$, where A, P are concept names occurring in \mathcal{T} and $x, y \in \Delta_i$. Define

- $L(A, P)$ consists of all words $w \in \mathcal{N}_r^*$ for which $\mathcal{A}_{\mathcal{T}}$ can move from state A to state P on input w .
- $L(x, y)$ consists of all words $w \in \mathcal{N}_r^*$ for which \mathcal{A}_i can move from state x to state y on input w .
- $L(x, P) = \{W \in \mathcal{N}_r^* \mid \forall y \in \Delta_i : W \in L(x, y) \text{ implies } y \in P^i\}$.

The following characterization of the semantics of \mathcal{FL}_0 -TBoxes with gfp-semantics makes use of these previously defined languages.

Lemma 2. *Let \mathcal{T} be a TBox and let $\mathcal{A}_{\mathcal{T}}$ be the corresponding semi-automaton. Let i be a gfp-model of \mathcal{T} and let D, E be concept names occurring in \mathcal{T} . Let $x \in \Delta_i$ be some element of Δ_i . Then*

- $x \in D^i$ iff for all primitive concepts $P \in \mathcal{N}_p$ it $L(A, P) \subseteq L(x, P)$ holds, and
- $A \sqsubseteq_{\mathcal{T}} B$ iff $L(B, P) \subseteq L(A, P)$ holds for all primitive concepts P .

A regular language is a language which is accepted by a finite state automaton. The languages $L(A, P)$ and $L(x, y)$ are all regular, since they are accepted by a finite state automaton. The languages $L(x, P)$ are also regular, which follows from this lemma.

Lemma 3. *For every model i , $x \in \Delta_i$ and $P \in \mathcal{N}_p$*

$$L(x, P) = \overline{\bigcup_{y \notin P^i} L(x, y)}$$

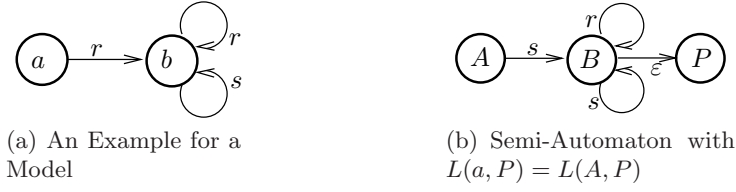


Fig. 2. A Model and Corresponding Semi-Automaton

holds. The class of regular languages is closed under complement and set union. Since the languages $L(x, c)$ are regular this implies that $L(x, P)$ is also regular.

Since $L(x, P)$ is regular for every $P \in \mathcal{N}_p$, we can find a finite state automaton which accepts $L(x, P)$. Therefore, there is a semi-automaton $\mathcal{A}_i = (\mathcal{N}_p, Q_i, E_i)$ and some $E \in Q_i$ such that $L(E, P) = L(x, P)$. By \mathcal{T}_i denote the TBox that corresponds to \mathcal{A}_i . Then Lemma 2 shows that $x \in E^i$ holds. Let \mathcal{T}'_i be conservative extension of \mathcal{T}_i and let F be a defined concept in \mathcal{T}'_i such that $x \in F^i$. Because of Lemma 2 we have $L(F, P) \subseteq L(x, P)$ for all $P \in \mathcal{N}_p$ and thus $L(F, P) \subseteq L(E, P)$. Using Lemma 2 we obtain $E \sqsubseteq_{\mathcal{T}'_i} F$. Thus E is the model-msc for x in i .

Theorem 1. *Let i be a model and $x \in \Delta_i$. Let \mathcal{A}_i be a semi-automaton with a state E for which $L(E, P) = L(x, P)$ for all $P \in \mathcal{N}_p$. Let \mathcal{T}_i be the TBox that corresponds to \mathcal{A}_i . Then E in \mathcal{T}_i is the model-msc for x in i .*

Since it is known that least common subsumers exists in \mathcal{FL}_0 with greatest fixpoint semantics, model-msc must also exist for non-singleton sets $\{x_1, \dots, x_n\}$. More precisely, we obtain $\text{mmsc}(\{x_1, \dots, x_n\}) = \text{lcs}(\text{mmsc}(x_1), \dots, \text{mmsc}(x_n))$. Note that when we construct a finite state automaton that accepts the complement to a language we implicitly use the subset construction to make a non-deterministic finite state automaton deterministic. This construction involves an exponential blowup in size which cannot be avoided. Thus also the model-msc may become exponentially large in the size of the model i .

An Example We give a short example of a model i_0 and show how to construct a model-based most specific concept for $a \in \Delta_{i_0}$. Let the model i_0 be defined by $\Delta_{i_0} = \{a, b\}$, $r^{i_0} = \{(a, b), (b, b)\}$, $s^{i_0} = \{(b, b)\}$ and $P^{i_0} = \emptyset$, where $\mathcal{N}_r = \{r, s\}$ and $\mathcal{N}_p = \{P\}$. The semi-automaton \mathcal{A}_{i_0} is depicted in Figure 2(a). To construct a model-based most specific concept we need to compute $L(a, P)$ first. From Lemma 3) we obtain $L(a, P) = \overline{L(a, a)} \cup L(a, b)$. The languages $L(a, a)$, and $L(a, b)$ formulated as regular expressions are

$$L(a, a) = \{\epsilon\}, \quad L(a, b) = L(r\{r, s\}^*)$$

So we obtain $L(a, P) = \overline{\{\epsilon\} \cup L(r\{r, s\}^*)} = L(s\{r, s\}^*)$. A semi-automaton with $L(a, P) = L(A, P)$ is depicted in Figure 2(b). This automaton gives rise to the TBox $\mathcal{T}_i = \{A \equiv \forall s.B, B \equiv P \sqcap \forall r.B \sqcap \forall s.B\}$. A is the model-msc for a in i_0 .

Informally, the model-msc describes individuals for which all s -successors and all successors of s -successors are in the extension of P . This is true for a in i since we know that a does not have any s -successors.

To illustrate the differences between open-world and closed-world semantics consider an ABox \mathcal{A} containing the assertions $r(a, b)$, $s(b, b)$ and $r(b, b)$. A graphical representation of this ABox would look exactly like Figure 2(a). Still the msc for a in \mathcal{A} would be \top and not A , since because of the open-world semantics of \mathcal{A} we do not know whether A has s -successors or not.

2.3 $\mathcal{FL}\mathcal{E}$ with Cyclic TBoxes and gfp-Semantics

We have seen in Section 2.1 that model-msc need not exist if we only allow for unfoldable TBoxes. For the case of \mathcal{FL}_0 we have successfully shown that this problem can be overcome by allowing for cyclic TBoxes with greatest fixpoint semantics. The same thing can be done for $\mathcal{FL}\mathcal{E}$.

$\mathcal{FL}\mathcal{E}$ -TBoxes can be visualized using so-called $\mathcal{FL}\mathcal{E}$ -description graphs that have been introduced by Baader et al. [8, 9]. Let \mathcal{T} be a normalized $\mathcal{FL}\mathcal{E}$ -TBox. In this context we call \mathcal{T} normalized if every concept definition in \mathcal{T} is of the form

$$A \equiv P_1 \sqcap \dots \sqcap P_k \sqcap \forall r_1. B_1 \dots \sqcap \forall r_l. B_l \sqcap \exists s_1. C_1 \dots \sqcap \exists s_m. C_m, \quad (3)$$

where $A, B_1, \dots, B_l, C_1, \dots, C_m$ are the names of defined concepts, P_1, \dots, P_k are primitive concept names and $r_1, \dots, r_l, s_1, \dots, s_m$ are defined concept names.

An $\mathcal{FL}\mathcal{E}$ -description graph $\mathcal{G}_{\mathcal{T}}$ is a directed graph with labelled nodes and edges. The nodes of $\mathcal{G}_{\mathcal{T}}$ are the names of defined concepts in \mathcal{T} . For every node A there is exactly one statement of the form (3) in \mathcal{T} . A node A is labelled with the primitive concept names P_1, \dots, P_k that occur in its definition. For every term $\forall r_n. B_n$ in (3) there is an edge from A to B_n labelled $\forall r_n$, and for every term $\exists s_n. C_n$ there is an edge from A to C_n labelled $\exists s_n$. The description graph of an unfoldable TBox will be a tree while the description graph of a cyclic TBox will naturally be a cyclic graph. Apart from being a handy visualization tool description graphs are also used to characterize the semantics of $\mathcal{FL}\mathcal{E}$ -TBoxes, for the cyclic case [10]. For an example of an $\mathcal{FL}\mathcal{E}$ -description graph see the example at the end of this section.

The construction for model-based most specific concepts in $\mathcal{FL}\mathcal{E}$ with cyclic TBoxes is a hybrid between the constructions for model-msc in \mathcal{EL} and \mathcal{FL}_0 . It is similar to the subset construction that is used to turn a nondeterministic finite state automaton into a deterministic finite state automaton. This has been used implicitly in Section 2.2. On the other hand the proofs and the semantical characterizations that are involved make heavy use of graph simulations which are also used in the case of \mathcal{EL} [3].

For a given model i we define the *canonical terminology of i* as follows. Let $\mathcal{G}_{\mathcal{T}_i}$ be a description graph whose nodes are all subsets $U \subseteq \Delta_i$ where

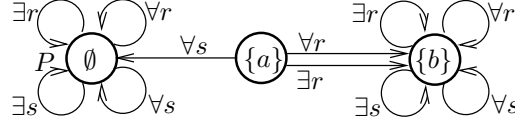


Fig. 3. The Canonical Terminology for the Model from Figure 2(a)

- a node U is labelled with $P \in \mathcal{N}_p$ iff all elements of U are in the extension of P , i. e. $x \in P^i$ for all $x \in U$, and
- there is an edge labelled $\exists r$ from U to $V \subseteq \Delta_i$ iff every $x \in U$ has an r -successor y in V , and
- there is an edge labelled $\forall r$ from U to $S_{r,U}$ where $S_{r,U} = \{x \in \Delta_i \mid \exists y \in U : (x, y) \in r^i\}$ is the set of all r -successors of some element of U .

We call the TBox \mathcal{T}_i that has $\mathcal{G}_{\mathcal{T}_i}$ as its description graph the *canonical terminology of i* . For every $U \subseteq \Delta_i$ we denote the corresponding concept name by A_U , to avoid confusion.

Why does this help us with the construction of model-msc? The underlying idea of the canonical terminology is that every defined concept A_U in \mathcal{T} should be the model-msc for U in i . Now, let us look at the description A_U in detail. What is the most specific description of the form $\forall r.B$ that can subsume A_U ? Under the assumption that $A_{S_{r,U}}$ is a most specific concept for $S_{r,U}$ the answer is $\forall r.A_{S_{r,U}}$. Since $S_{r,U}$ is the set of all r -successors of some element of U , $U \subseteq (\forall r.A_{S_{r,U}})^i$ must hold. On the other hand, $A_{S_{r,U}} \sqsubseteq B$ holds for every B with $S_{r,U} \subseteq B^i$ (and thus for every B with $U \subseteq (\forall r.B)^i$). Therefore, $\forall r.A_{S_{r,U}} \sqsubseteq \forall r.B$ holds for all such B . It therefore makes perfect sense to add to $\mathcal{G}_{\mathcal{T}_i}$ an edge labelled $\forall r$ pointing from U to $S_{r,U}$. The intuition for the existential restrictions is similar.

The complete proof of the following theorem uses a variant of the Knaster-Tarski-fixpoint theorem and a characterization of gfp-semantics that is based on graph-simulations between $\mathcal{FL}\mathcal{E}$ -description graphs [10].

Theorem 2. *Let i be a model for a given set of primitive concept names \mathcal{N}_p and a set of role names \mathcal{N}_r . Let $U \subseteq \Delta_i$ be a subset of its domain and let \mathcal{T}_i be the canonical terminology for i . Then A_U in \mathcal{T}_i is the model-msc for i .*

An Example Let us look at the example from Section 2.2 again. The model i_0 has the canonical terminology whose description graph is depicted in Figure 2.3. The vertex $\{a, b\}$ is redundant and has been omitted to keep the diagram somewhat legible. From the description graph we obtain the terminology $\mathcal{T}_i = \{A_a \equiv \forall r.A_b \sqcap \forall s.A_\emptyset \sqcap \exists r.A_b, A_b \equiv \forall r.A_b \sqcap \forall s.A_b \sqcap \exists r.A_b \sqcap \exists s.A_b, A_\emptyset \equiv P \sqcap \forall r.A_\emptyset \sqcap \forall s.A_\emptyset \sqcap \exists r.A_\emptyset \sqcap \exists s.A_\emptyset\}$.

2.4 $\mathcal{ALC}^{\cup*}$

In the case of \mathcal{ALC} it is not necessary to use the full expressive power of greatest-fixpoint semantics in order to make sure that most specific concepts always exist.

It suffices to add union of roles (\cup) and reflexive-transitive closure of roles (r^*) to guarantee existence of model-msc. The semantics of \cup and \cdot^* are intuitively defined as $(r_1 \cup r_2)^i = r_1^i \cup r_2^i$ and $(r^*)^i = (r^i)^*$ for all models i , where $(r^i)^*$ denotes the reflexive transitive closure of the binary relation r^i .

Let's go back to the example from Section 1.1. First of all, notice that there is a concept description (e. g. S) that has only **Carla** in its extension. Likewise, P has only **Nicolas** in its extension. If such concept descriptions exist, we say that **Carla** and **Nicolas** are distinguishable.

We try to find a model-msc for **Carla**. For a start consider the concept descriptions $D_{\text{Carla}} \equiv S \sqcap \neg P \sqcap \exists \text{mt}. P \sqcap \forall \text{mt}. P$ and $D_{\text{Nicolas}} \equiv P \sqcap \neg S \sqcap \exists \text{mt}. S \sqcap \forall \text{mt}. S$. D_{Carla} and D_{Nicolas} describe the elements **Carla** and **Nicolas** in the model i , but they are not specific enough to be model-msc. For example $\forall \text{mt}. \exists \text{mt}. S$ also describes **Carla**, but does not subsume D_{Carla} . But the concept description $\forall \text{mt}. P$ does. We know that **Nicolas** is the only Politician in the model i . Therefore, it would be helpful to require something like "Every politician is an instance of D_{Nicolas} ". With the help of the transitive closure of roles we can do something very similar, by defining

$$E_{\text{Carla}} \equiv D_{\text{Carla}} \sqcap \forall \text{mt}^*. (\neg P \sqcup D_{\text{Nicolas}}) \sqcap \forall \text{mt}^*. (\neg S \sqcup D_{\text{Carla}}).$$

Let j be some model with the same signature as i and $x \in E_{\text{Carla}}^j$. The term $\forall \text{mt}^*. (\neg P \sqcup D_{\text{Nicolas}})$ makes sure that every mt^* -successor of x is in D_{Nicolas}^j if it is in P^j (and it is in D_{Carla}^j if it is in S^j). Because x is an instance of D_{Carla} we can be sure that every mt -successor of x is a Politician. This means that this successor has to be an instance of D_{Nicolas} , which in turn requires that all its successors are Singers, and so on. This way, we have implicitly added a flavour of cycles to E_{Carla} without using cycles explicitly.

This gives us an idea how to formally construct model-msc for a given model i . We first assume that all elements of Δ_i are distinguishable, i. e. that for every $x \in \Delta_i$ there is some concept description C_x such that $\{x\} = C_x^i$. Then define

$$D_x \equiv \prod_{\substack{P \in \mathcal{N}_p \\ x \in P^i}} P \sqcap \prod_{\substack{P \in \mathcal{N}_p \\ x \notin P^i}} \neg P \sqcap \prod_{r \in \mathcal{N}_r} \left(\left(\prod_{\substack{y \in \Delta_i \\ (x,y) \in r^i}} \exists r. C_y \right) \sqcap \left(\forall r. \bigsqcup_{\substack{y \in \Delta_i \\ (x,y) \in r^i}} C_y \right) \right),$$

where we define the empty disjunction to be \perp and the empty conjunction to be \top . Furthermore define

$$M_x \equiv C_x \sqcap \prod_{y \in \Delta_i} \forall \left(\bigcup_{r \in \mathcal{N}_r} r \right)^*. (\neg C_y \sqcup D_y). \quad (4)$$

A bisimulation is a relation $\zeta \subseteq \Delta_i \times \Delta_j$ with the following three properties. Let $(y, y') \in \zeta$.

- For every $P \in \mathcal{N}_p$ it holds that $y \in P^i$ iff $y' \in P^j$.
- *Forth condition.* $z \in \Delta_i, (y, z) \in r^i \Rightarrow (\exists z' \in \Delta_j : (y', z') \in r^j \text{ and } (z, z') \in \zeta)$
- *Back condition.* $z' \in \Delta_j, (y', z') \in r^j \Rightarrow (\exists z \in \Delta_i : (y, z) \in r^i \text{ and } (z, z') \in \zeta)$

Let $j = (\Delta_j, \cdot^j)$ be a second model over the same signature and let $x' \in M_x^j$. \mathcal{ALC} and $\mathcal{ALC}^{\cup*}$ can be viewed as modal logics. It is a well-known fact from modal logics that if there is a bisimulation from x to x' then x and x' are indistinguishable, i. e. there is no $\mathcal{ALC}^{\cup*}$ -concept description C with $x \in C^i$ but $x' \notin C^j$. It is easy but tedious to show that

$$Z = \{(y, y') \mid y \in \Delta_i, y' \in C_y^j, (x', y') \in \left(\left(\bigcup_{r \in \mathcal{N}_r} r \right)^* \right)^j\}$$

is a bisimulation. This proves that x and x' are indistinguishable. Therefore for every $\mathcal{ALC}^{\cup*}$ concept description D with $x \in D^i$ it holds that $x' \in D^j$ for all $x' \in M_x^j$. Thus $M_x^j \subseteq D^j$. Since j was an arbitrary model this implies that $M_x \sqsubseteq D$. Therefore, M_x is a most specific concept for $\{x\}$. Most specific concepts for non-singleton sets $\{x_1, \dots, x_n\}$ can be obtained as the disjunction of M_{x_1}, \dots, M_{x_n} .

Theorem 3. *If i is a model over a signature $(\mathcal{N}_p, \mathcal{N}_r)$ then M_x as defined in (4) is a model-msc for $\{x\}$. Since least common subsumers exist in $\mathcal{ALC}^{\cup*}$ ($\mathcal{ALC}^{\cup*}$ provides for disjunction) this shows that model-msc exist for all subsets of Δ_i .*

Theorem 3 can be generalized to the case of models that do contain indistinguishable states [10].

3 Conclusion and Future Work

We have seen that in many standard logics model-msc need not exist. In the present work we have presented constructions for model-mscs in \mathcal{ALC} with union of roles and reflexive-transitive closure of roles and in \mathcal{FL}_0 and \mathcal{FLE} with cyclic TBoxes and gfp-semantics. There is hope that the model-msc can be used for knowledge base completion. The existence of model-msc ensures that a basis for the set of GCIs that hold in a given background model need not contain more than one GCI with a given left-hand side and that the right-hand-side of this GCI can be computed without knowing a finite set of axioms beforehand. However, the number of left-hand sides may still be large. For \mathcal{EL} one can use ideas from Formal Concept Analysis to further reduce the number of GCIs and to obtain a knowledge base completion formalism [5]. It remains to be examined whether this can also be done for the three logics presented here.

The three logics used here are admittedly not very common. This is firstly because of the gfp-semantics involved. For the case of \mathcal{EL} it has been shown that one can get rid of cyclic concept descriptions (and gfp-semantics) easily after a complete set of GCIs has been found. A similar result for the logics presented here is needed to make the results useful in practice.

Secondly, one might argue that the logics \mathcal{FLE} , and \mathcal{FL}_0 are still not commonly used, even without gfp-semantics. However, they are fragments of commonly used logics. Completeness with respect to GCIs formulated using one of these fragments is still much stronger than completeness with respect to GCIs that are only allowed to use conjunction.

References

1. Franz Baader. Using automata theory for characterizing the semantics of terminological cycles. *Ann. of Mathematics and Artificial Intelligence*, 18:175–219, 1996.
2. Franz Baader. Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In Georg Gottlob and Toby Walsh, editors, *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence*, pages 319–324. Morgan Kaufmann, 2003.
3. Franz Baader. Terminological cycles in a description logic with existential restrictions. In Georg Gottlob and Toby Walsh, editors, *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 325–330, Acapulco, Mexico, 2003. Morgan Kaufmann, Los Altos.
4. Franz Baader and Felix Distel. A finite basis for the set of \mathcal{EL} -implications holding in a finite model. In Raoul Medina and Sergei Obiedkov, editors, *Proc. of the 6th Int. Conf. on Formal Concept Analysis (ICFCA '08)*, volume 4933 of *Lecture Notes in Artificial Intelligence*, pages 46–61. Springer, 2008.
5. Franz Baader and Felix Distel. Exploring finite models in the description logic $\mathcal{EL}_{\text{gfp}}$. In *Proc. of ICFCA 2009*. Springer, 2009.
6. Franz Baader, Bernhard Ganter, Ulrike Sattler, and Barış Sertkaya. Completing description logic knowledge bases using formal concept analysis. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI 2007)*. AAAI Press/The MIT Press, 2007.
7. Franz Baader and Ralf Küsters. Computing the least common subsumer and the most specific concept in the presence of cyclic \mathcal{ALN} -concept descriptions. In *Proc. of the 22nd German Annual Conf. on Artificial Intelligence (KI'98)*, volume 1504 of *Lecture Notes in Computer Science*, pages 129–140. Springer-Verlag, 1998.
8. Franz Baader, Ralf Küsters, and Ralf Molitor. Computing least common subsumers in description logics with existential restrictions. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI'99)*, pages 96–101, 1999.
9. Sebastian Brandt and Anni-Yasmin Turhan. Computing least common subsumers for $\mathcal{FL}_{\mathcal{E}}^+$. In *Proc. of DL 2003*, CEUR-WS, 2003.
10. Felix Distel. Model-based most specific concepts in description logics with value restrictions. LTCS-Report 08-04, Institute for theoretical computer science, TU Dresden, Dresden, Germany, 2008. <http://lat.inf.tu-dresden.de/research/reports.html>.
11. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, New York, 1997.
12. Bernhard Nebel. Terminological cycles: Semantics and computational properties. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 331–361. Morgan Kaufmann, Los Altos, 1991.
13. Sebastian Rudolph. Exploring relational structures via FLE. In K. E. Wolff, H. D. Pfeiffer, and H. S. Delugach, editors, *ICCS*, volume 3127 of *LNCS*, pages 196–212, 2004.
14. Sebastian Rudolph. *Relational Exploration – Combining Description Logics and Formal Concept Analysis for Knowledge Specification*. PhD thesis, Technische Universität Dresden, 2006.
15. Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In Georg Gottlob and Toby Walsh, editors, *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 355–362, Acapulco, Mexico, 2003. Morgan Kaufmann, Los Altos.