On Free Description Logics with Definite Descriptions

Alessandro Artale[1], Andrea Mazzullo[1], Ana Ozaki[2], and Frank Wolter[3]

[1] Free University of Bozen-Bolzano
`{artale,mazzullo}@inf.unibz.it`
[2] University of Bergen
`ana.ozaki@uib.no`
[3] University of Liverpool
`wolter@liverpool.ac.uk`

Abstract. Definite descriptions are phrases of the form 'the x such that φ'. Together with individual names, they are used to refer to single entities in a context. In some cases, however, names and descriptions may fail to denote any object at all, as witnessed by the name 'DL 1993', for a workshop that never took place, or the description 'the Special Session of DL 2020', for a non-occurring event. In this work, we introduce and investigate DL languages with individual names and definite descriptions which may both fail to denote. We focus on $\mathcal{ALCO}$-based and $\mathcal{ELO}$-based languages. A generic polynomial time reduction of the resulting expressive *free DLs with definite descriptions* into classical DLs is provided, and we show that free $\mathcal{ELO}$ with definite descriptions is still in PTime. Moreover, we characterise the expressive power of concepts relative to first-order formulas interpreted on partial interpretations using a suitable notion of bisimulation.

1 Introduction

A noun phrase that can be used to refer to a single object in a context is known in linguistics as a *referring expression*. These include both *individual names*, such as 'DL 2020', and *definite descriptions*, such as 'the General Chair of DL 2020' [33, 16]. Another feature of individual names and definite descriptions in natural language is that they might also *fail* to denote any object at all. For instance, 'DL 1993' is a non-denoting individual name, since no DL workshop took place in 1993, while, 'the Program Chair of DL 2020' and 'the Special Session of DL 2020' are non-denoting definite descriptions, since this workshop in 2020 has, respectively, two Program Chairs and no Special Session at all.

When it comes to formalisation, however, this behaviour is not easily captured in frameworks based on classical first-order logic, where an individual name is always assigned to an element of the domain by the interpretation function, and definite descriptions are not included among the terms of the language. An approach dating back to Russell [39] paraphrases sentences with definite descriptions in classical first-order logic by making their existence and uniqueness conditions explicit. Logics that allow instead for possibly non-denoting terms, either definite descriptions or individual names, are known as *free logics* in the literature, since their terms may lack existential import. Several syntactical and semantical options for free logics with definite descriptions have been proposed [11, 32].

In this work, we introduce and study a family of DL languages with both individual names and definite descriptions, that we call *free DLs with definite descriptions*, or *free DLs*, for short. Syntactically, these languages extend the classical ones with nominals of the form $\{\iota C\}$, where ιC is a term standing for the definite description 'the object that is C' and C is a concept. We denote the resulting DLs with an upperscript ι, focussing in particular on $\mathcal{ALCO}^\iota$ and $\mathcal{ELO}^\iota$. Their semantics is based on *partial interpretations*, that generalise the classical ones by letting the interpretation function to be *partial* on individual names, meaning that only a *subset* of all the individual names has its elements assigned to objects of the domain. Moreover, the extension of $\{\iota C\}$ in a partial interpretation coincides with that of the concept C, if C is interpreted as a singleton, and it is empty otherwise. This semantic choice respects the main tenets of Russell's paraphrase, while also preserving definite descriptions as terms.

One motivation behind the introduction of this family of languages is to add, at the modelling level, the flexibility of empty-valued individual names, as well as the possibility to single out elements of a domain via definite descriptions. These additional features can also be used in the context of query answering over DL knowledge bases. For example, the Boolean instance queries

$$\top(\mathsf{dl93}), \qquad \exists \mathsf{isProgramChairOf}.\{\mathsf{dl09}\}(\iota \exists \mathsf{isGeneralChairOf}.\{\mathsf{dl20}\}),$$

ask whether 'DL 1993' names anything at all, and whether the General Chair of DL 2020 was also a Program Chair of DL 2009, respectively. One can now retrieve not only individual names, but also definite descriptions as answers to queries. Moreover, nominals involving definite descriptions can be used to form concept inclusions with different satisfaction conditions. Consider for instance

$$\exists \mathsf{organises}.\{\mathsf{dl20}\} \sqsubseteq \exists \mathsf{reportsTo}.\{\iota \exists \mathsf{isGeneralChairOf}.\{\mathsf{dl20}\}\},$$
$$\{\iota \exists \mathsf{isProgramChairOf}.\{\mathsf{dl20}\}\} \sqsubseteq \exists \mathsf{selects}.\mathsf{Reviewer}.$$

The former, stating that every organiser of DL 2020 reports to the General Chair of DL 2020, forces $\exists \mathsf{isGeneralChairOf}.\{\mathsf{dl20}\}$ to have exactly one element in all its models satisfying $\exists \mathsf{organises}.\{\mathsf{dl20}\}$. The latter holds if, whenever there is exactly one Program Chair of DL 2020, that individual $\mathsf{selects}$ some $\mathsf{Reviewer}$, but also in interpretations without, or with more than one, Program Chair of DL 2020.

On the technical side, we show that reasoning in free DLs with definite descriptions can be performed at no additional costs. For (extensions of) $\mathcal{ALCO}^\iota$, we employ a reduction to languages covered by the OWL 2 standard, so that efficient off-the-shelf reasoners can be used. In particular, we prove that satisfiability in $\mathcal{ALCO}^{\iota}$ can be polynomially reduced (via a translation that can be applied to other constructors as well) to $\mathcal{ALCO}_u$, i.e., $\mathcal{ALCO}$ extended with the universal role. Moreover, we show that entailment in $\mathcal{ELO}^{\iota}$ knowledge bases remains tractable, using a modified version of the algorithm for classical $\mathcal{ELO}$. Finally, we focus on $\mathcal{ALCO}^{\iota}$ expressive power, showing that its concepts can be characterised in terms of first-order formulas on partial interpretations that are invariant under a suitable notion of bisimulation. We conclude the paper with a discussion of related work and future directions.

2 Free Description Logics

We introduce basic notions for free DLs (with definite descriptions) by presenting the syntax and semantics of $\mathcal{ALCO}^{\iota}$, which we define as a free DL based on the classical DL $\mathcal{ALCO}$ [5], and other related languages.

2.1 Syntax

Let $\mathsf{N_C}$, $\mathsf{N_R}$ and $\mathsf{N_I}$ be countably infinite and pairwise disjoint sets of *concept names*, *role names*, and *individual names*, respectively. The $\mathcal{ALCO}^{\iota}$ *terms* and *concepts* are constructed by mutual induction as follows:

$$\tau ::= a \mid \iota C, \qquad\qquad C ::= A \mid \neg C \mid (C \sqcap C) \mid \exists r.C \mid \{\tau\},$$

with $a \in \mathsf{N_I}$, $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. A term of the form ιC is called a *definite description*, and a concept $\{\tau\}$ is called a *(term) nominal*. An $\mathcal{ALCO}^{\iota}$ *atom* is either an $\mathcal{ALCO}^{\iota}$ *concept inclusion* (*CI*) of the form $C \sqsubseteq D$ or an $\mathcal{ALCO}^{\iota}$ *assertion* of the form $A(\tau)$ or $r(\tau_1, \tau_2)$, where C, D are $\mathcal{ALCO}^{\iota}$ concepts, $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and τ, τ_1, τ_2 are $\mathcal{ALCO}^{\iota}$ terms. An *instance query* is either an assertion or an expression of the form $C(\tau)$, where C is an $\mathcal{ALCO}^{\iota}$ concept and τ is a term. We may omit '$\mathcal{ALCO}^{\iota}$' if this is clear from the context. Thus, a *TBox* $\mathcal{T}$ is a finite set of CIs, an *ABox* $\mathcal{A}$ is a finite set of assertions, and a *knowledge base* (KB) $\mathcal{K}$ is a pair $(\mathcal{T}, \mathcal{A})$. Although we are particularly interested in working with KBs, for the presentation of the results it is convenient to combine CIs and assertions into $\mathcal{ALCO}^{\iota}$ *formulas*, defined as expressions of the form

$$\varphi ::= \alpha \mid \neg\varphi \mid (\varphi \land \varphi),$$

where α is an atom. All the usual syntactic abbreviations and conventions are assumed. In particular, for concepts, we set $\bot = A \sqcap \neg A$, $\top = \neg\bot$, $C \sqcup D = \neg(\neg C \sqcap \neg D)$, $C \Rightarrow D = \neg C \sqcup D$, and $\forall r.C = \neg\exists r.\neg C$, while a *concept equivalence* (*CE*) $C \equiv D$ abbreviates $C \sqsubseteq D, D \sqsubseteq C$ (as a formula, it stands for the conjunction of the two). The *signature* of φ, Σ_φ, is the set of all concept, role and individual names occurring in φ, while $\mathsf{con}(\varphi)$ is the set of all concepts occurring in φ (and similarly for concepts, TBoxes, ABoxes and KBs).

In the rest of this paper, we will consider other DL languages with nominals, that we introduce briefly here. Since $\mathcal{ALCO}^\iota$ nominals are constructed using arbitrary terms, the classical $\mathcal{ALCO}$ is the sublanguage of $\mathcal{ALCO}^\iota$ where terms can only be in $\mathsf{N_I}$. The language $\mathcal{ALCO}_u$ extends $\mathcal{ALCO}$ with the *universal role* u, allowing for concepts of the form $\exists u.C$ [5]. Moreover, $\mathcal{ELO}^\iota$ is the language obtained from $\mathcal{ALCO}^\iota$ by allowing only for $\bot$, $\top$ (considered now as primitive logical symbols), concept names, term nominals, conjunctions (both on concepts and formulas) and existential restrictions, while negations and disjunctions can be applied to formulas only. Finally, $\mathcal{ELO}$ is the sublanguage of $\mathcal{ELO}^\iota$ with only individual names as terms.

2.2 Semantics

For the DL languages with nominals considered in this work, we introduce semantics that generalise the classical ones through the notion of *partial* interpretation. A *partial interpretation* is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set, called the *domain* of $\mathcal{I}$, and $\cdot^\mathcal{I}$ is a function that maps every $A \in \mathsf{N_C}$ to a subset of $\Delta^\mathcal{I}$, every $r \in \mathsf{N_R}$ to a subset of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$, the universal role u to the set $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$ itself, and every a in a *subset* of $\mathsf{N_I}$ to an element in $\Delta^\mathcal{I}$. In other words, $\cdot^\mathcal{I}$ is a total function on $\mathsf{N_C} \cup \mathsf{N_R}$ and a partial function on $\mathsf{N_I}$. A *total interpretation* is a partial interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ in which $\cdot^\mathcal{I}$ is also total on $\mathsf{N_I}$. The *value* $\tau^\mathcal{I}$ of a term τ in $\mathcal{I}$ and the *extension* $C^\mathcal{I}$ of a concept C in $\mathcal{I}$ are defined by mutual induction:

$$(\iota C)^\mathcal{I} = \begin{cases} d, & \text{if } C^\mathcal{I} = \{d\}, \text{ for some } d \in \Delta^\mathcal{I}; \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

We say that τ *denotes* in $\mathcal{I}$ iff $\tau^\mathcal{I} = d$, for a $d \in \Delta^\mathcal{I}$. Thus, in particular, an individual name a denotes in $\mathcal{I}$ iff $a^\mathcal{I}$ is defined. In addition:

$$(\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I}, \qquad (C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I},$$
$$(\exists r.C)^\mathcal{I} = \{d \in \Delta^\mathcal{I} \mid \exists e \in C^\mathcal{I} : (d, e) \in r^\mathcal{I}\}.$$

Moreover, we set $\{\tau\}^\mathcal{I} = \{\tau^\mathcal{I}\}$, if τ denotes in $\mathcal{I}$, and $\{\tau\}^\mathcal{I} = \emptyset$, otherwise.

A concept C is *satisfied in* $\mathcal{I}$ iff $C^\mathcal{I} \neq \emptyset$, and it is *satisfiable* iff there is a partial interpretation in which it is satisfied. The *satisfaction of an* $\mathcal{ALCO}^\iota$ *formula* φ *in* $\mathcal{I}$, written $\mathcal{I} \models \varphi$, is defined as follows. For CIs:

$$\mathcal{I} \models C \sqsubseteq D \quad \text{iff} \quad C^\mathcal{I} \subseteq D^\mathcal{I}.$$

For instance queries (generalizing assertions):

$$\mathcal{I} \models C(\tau) \quad \text{iff} \quad \tau \text{ denotes in } \mathcal{I} \text{ and } \tau^\mathcal{I} \in C^\mathcal{I},$$
$$\mathcal{I} \models r(\tau_1, \tau_2) \quad \text{iff} \quad \tau_1, \tau_2 \text{ denote in } \mathcal{I} \text{ and } (\tau_1^\mathcal{I}, \tau_2^\mathcal{I}) \in r^\mathcal{I}.$$

Finally, for the remaining formulas:

$$\mathcal{I} \models \neg\psi \quad \text{iff} \quad \mathcal{I} \not\models \psi, \qquad \mathcal{I} \models \psi \wedge \chi \quad \text{iff} \quad \mathcal{I} \models \psi \text{ and } \mathcal{I} \models \chi.$$

We say that φ is *satisfied* in a partial interpretation $\mathcal{I}$ (or that $\mathcal{I}$ *satisfies*, or is a *model* of, φ) iff $\mathcal{I} \models \varphi$, and that φ is *satisfiable* iff it is satisfied in some $\mathcal{I}$. Moreover, φ *entails* ψ, written $\varphi \models \psi$, if every interpretation that satisfies φ satisfies also ψ. Finally, φ and ψ are *equivalent* iff they entail each other. As usual, the *formula satisfiability problem* is the problem of deciding whether a given formula is satisfied in some (partial) interpretation. The *entailment problem* is the problem of deciding whether a given formula entails another formula (or, in particular, an instance query). We may also consider these problems restricted to total interpretations and write 'on total interpretations' explicitly whenever this is the case. These notions extend naturally to TBoxes, ABoxes and KBs, as well as to other free DLs presented in subsequent sections.

Example 1. In the context of DL workshops, $\mathsf{dl93}$ and $\mathsf{dl20}$ are examples of, respectively, a non-denoting and a denoting individual name, while the definite description $\iota\exists\mathsf{isGeneralChairOf}.\{\mathsf{dl20}\}$ denotes the General Chair of DL 2020. The following concept, applying to Program Chairs of DL 2020 that are not the only ones, shows an interaction between a nominal constructed from a definite description and the concept occurring inside it:

$$\exists\mathsf{isProgramChairOf}.\{\mathsf{dl20}\} \sqcap \neg\{\iota\exists\mathsf{isProgramChairOf}.\{\mathsf{dl20}\}\}.$$

Let $\mathsf{OneOfDL20ProgramChairs}$ abbreviate the concept above. The CI

$$\mathsf{OneOfDL20ProgramChairs} \sqsubseteq \exists\mathsf{collaboratesWith}.\exists\mathsf{isProgramChairOf}.\{\mathsf{dl20}\}$$

states that every DL 2020 Program Chair who is not the only one collaborates with someone who is a DL 2020 Program Chair.

2.3 First Observations

We discuss some properties of free DLs. Our first observation is that the $\mathcal{ALCO}^{\iota}$ formula satisfiability problem on total interpretations can be reduced to the $\mathcal{ALCO}^{\iota}$ formula satisfiability problem on partial interpretations. This is because an $\mathcal{ALCO}^{\iota}$ term τ denotes in a partial interpretation $\mathcal{I}$ iff $\mathcal{I} \models \neg(\{\tau\} \sqsubseteq \bot)$. An $\mathcal{ALCO}^{\iota}$ formula φ entails $\top(\tau)$ iff $\varphi \models \neg(\{\tau\} \sqsubseteq \bot)$ and this happens iff τ denotes in all the $\mathcal{ALCO}^{\iota}$ partial interpretations that are models of φ. Then, to solve satisfiability on total interpretations, one can simply add conjuncts of the form $\neg(\{\tau\} \sqsubseteq \bot)$ for each individual name $\tau = a$ occurring in the formula.

Our second observation is that, due to partial interpretations, an instance query $C(\tau)$ is *not* equivalent to $\{\tau\} \sqsubseteq C$. Indeed, while terms always denote in the models of the instance queries (and assertions) in which they occur, the CI $\{\tau\} \sqsubseteq C$ is satisfied in any partial interpretation where τ is not denoting. Nevertheless, instance queries are just syntactic sugar: one can replace $C(\tau)$ by $\{\tau\} \sqsubseteq C \land \neg(\{\tau\} \sqsubseteq \bot)$; and $r(\tau_1, \tau_2)$ by $\{\tau_1\} \sqsubseteq \exists r.\{\tau_2\} \land \neg(\{\tau_1\} \sqsubseteq \bot)$. Thus, from now on, we may assume without loss of generality that $\mathcal{ALCO}^{\iota}$ formulas do not contain assertions. Also, deciding whether instance queries are entailed can be reduced to formula satisfiability. Observe that this encoding yields an equivalent formula. It is possible to obtain an equisatisfiable translation that is also expressible within the $\mathcal{ELO}^\iota$ fragment by replacing $C(\tau)$ with CIs of the form $\top \sqsubseteq \exists s.\{\tau\}$, $\{\tau\} \sqsubseteq C$, and $r(\tau_1, \tau_2)$ with $\top \sqsubseteq \exists s.\{\tau_1\}$, $\{\tau_1\} \sqsubseteq \exists r.\{\tau_2\}$, where s is a fresh role name.

Our third observation is regarding Boolean operators. CIs of the form

$$\{\iota(C \sqcup D)\} \sqsubseteq \{\iota C\} \sqcup \{\iota D\}, \qquad \{\iota C\} \sqcap \{\iota D\} \sqsubseteq \{\iota(C \sqcap D)\}$$

are satisfied in every partial interpretation (but the direction $\sqsupseteq$ may not hold).

Finally, we point out that, for satisfiability checking, it suffices to consider $\mathcal{ALCO}^\iota$ formulas where all occurrences of definite descriptions are of the form ιB, where B is a concept name. Indeed, let $\iota C_1, \ldots, \iota C_n$ be all the definite descriptions in φ that do not occur in the body of another definite description $\iota C'$, where the *body* of a definite description ιC is just C. We define the $\mathcal{ALCO}^\iota$ formula φ' as the formula obtained by substituting the bodies $C_1, \ldots, C_n$ of $\iota C_1, \ldots, \iota C_n$ with fresh concept names $B_{C_1}, \ldots, B_{C_n}$, respectively. Then, φ'' is defined as the $\mathcal{ALCO}^\iota$ conjunction:

$$\varphi' \wedge \bigwedge_{1 \leq i \leq n} (B_{C_i} \equiv C_i)'',$$

where $(B_C \equiv C)''$ is the formula obtained by recursively applying the procedure just described to the CE $B_C \equiv C$. It can be checked that φ and φ'' are equisatisfiable $\mathcal{ALCO}^\iota$ formulas. Moreover, given φ'', we assume without loss of generality that φ'' does not contain any assertion (cf. second observation above). Finally, all the CIs occurring in φ'' will be assumed without loss of generality to be either of the form $E \sqsubseteq F$, where E, F are $\mathcal{ALC}$ concepts, or $\{\tau\} \sqsubseteq A$, or $A \sqsubseteq \{\tau\}$, with $A \in \mathsf{N_C}$ and τ either an individual name or of the form ιB, where $B \in \mathsf{N_C}$. Indeed, given an $\mathcal{ALCO}^\iota$ CI $C \sqsubseteq D$ occurring in φ'', we can obtain an equisatisfiable $\mathcal{ALCO}^\iota$ formula by substituting all nominals $\{\tau\}$ occurring in $C \sqsubseteq D$ with concept names A_τ, and taking the conjunction of the resulting $\mathcal{ALC}$ CI with the CEs $A_\tau \equiv \{\tau\}$. A formula in this format is said to be in *normal form*.

3 Reasoning in Free Description Logics

We analyse the complexity of reasoning in $\mathcal{ALCO}^\iota$ and in $\mathcal{ELO}^\iota$.

3.1 Satisfiability in $\mathcal{ALCO}^\iota$

We prove that satisfiability in $\mathcal{ALCO}^\iota$ is ExpTime-complete. To show this result, we provide a polynomial size equisatisfiable translation into $\mathcal{ALCO}_u$. Given an $\mathcal{ALCO}^\iota$ formula φ in normal form, we define a translation of φ into an $\mathcal{ALCO}_u$ formula φ^*. While the translation preserves concept and role names in $\mathsf{N_C} \cup \mathsf{N_R}$, nominals $\{\tau\}$ are translated in the following way:

$$\{\tau\}^* = \{\tau\}^+ \sqcap C_\tau^{\leq 1},$$

where

$$\{\tau\}^+ = \begin{cases} A_b, & \text{if } \tau \text{ is of the form } b \in \mathsf{N_I}, \\ B, & \text{if } \tau \text{ is of the form } \iota B, \end{cases}$$

with A_b fresh concept name, and $C_\tau^{\leq 1}$ stands for the concept

$$\forall u.(\{\tau\}^+ \Rightarrow \{a_\tau\}),$$

with a_τ fresh individual name. We now define φ^* inductively as follows:

$$(E \sqsubseteq F)^* = E \sqsubseteq F, \quad (\{\tau\} \sqsubseteq A)^* = \{\tau\}^* \sqsubseteq A, \quad (A \sqsubseteq \{\tau\})^* = A \sqsubseteq \{\tau\}^*,$$

$$(\neg\psi)^* = \neg\psi^*, \qquad (\psi \wedge \chi)^* = \psi^* \wedge \chi^*,$$

where E, F are $\mathcal{ALC}$ concepts, and A is a concept name. Finally, we define the formula translation as

$$\varphi^\dagger = \varphi^* \wedge \bigwedge_{\{\tau\} \in \mathsf{con}(\varphi)} \{\tau\}^+ \sqsubseteq \forall u.(\{a_\tau\} \Rightarrow \{\tau\}^+).$$

Lemma 1. *An $\mathcal{ALCO}^\iota$ formula φ is satisfiable iff the $\mathcal{ALCO}_u$ formula $\varphi^\dagger$ is satisfiable on total interpretations.*

It follows from a known result in Propositional Dynamic Logic extended with nominals and the universal modality [35, Corollary 7.7] that the $\mathcal{ALCO}_u$ formula satisfiability problem is in ExpTime. The matching lower bound comes from the $\mathcal{ALC}$ formula satisfiability problem [23]. Since the $\mathcal{ALCO}^\iota$ formula satisfiability and entailment problems on total interpretations are reducible to their counterparts on partial interpretations (Subsection 2.3), the following holds.

Theorem 1. *The $\mathcal{ALCO}^\iota$ formula satisfiability and the entailment problems on partial and total interpretations are* ExpTime*-complete.*

The reduction we presented can be easily adapted to deal with more expressive DLs, e.g., extensions of $\mathcal{ALCO}$ with inverse roles and number restrictions.

3.2 Reasoning in $\mathcal{ELO}^\iota$

We prove that satisfiability of $\mathcal{ELO}^\iota$ formulas is NP-complete and entailment in $\mathcal{ELO}^\iota$ KBs is PTime-complete. To show these results, we assume without loss of generality that the ABox and instance queries can be encoded within the TBox (cf. Subsection 2.3) and adapt the completion algorithm for $\mathcal{ELO}$ TBoxes [7]. The main idea is to add a copy of each concept name in a TBox and remove it only if the extension of it is exactly one in any model. Even though $\mathcal{ELO}^\iota$ admits a mild form of disjunction ($\{\iota A\} \sqsubseteq B$ states that the extension of A contains at least two elements *or* that $A \sqsubseteq B$), the logic remains 'Horn' in the sense that minimal models exist.

Let $\mathcal{T}$ be an $\mathcal{ELO}^\iota$ TBox. We denote by $\mathcal{BC}_\mathcal{T}$ the union of $\{\top\}$, the set of all concept names occurring in $\mathcal{T}$, and the set of all concepts $\{a\}$ with a an individual name occurring in $\mathcal{T}$. Also, we denote by $\mathcal{BC}_{\mathcal{T}}^{+}$ the union of $\mathcal{BC}_{\mathcal{T}}$ with $\{\bot\} \cup \{\{\iota A\} \mid \{\iota A\}$ occurs in $\mathcal{T}\}$ and by $\mathcal{R}_{\mathcal{T}}$ the set of role names occurring in $\mathcal{T}$. We assume without loss of generality that any $\mathcal{ELO}^{\iota}$ TBox $\mathcal{T}$ is *normalized* and all CIs in it have one of the following forms:

$$C_1 \sqcap C_2 \sqsubseteq D, \quad \exists r.C \sqsubseteq D, \quad C \sqsubseteq \exists r.D, \quad \{\tau\} \sqsubseteq D, \quad D \sqsubseteq \{\tau\},$$

where $C_{(i)} \in (\mathcal{BC}_{\mathcal{T}} \cap \mathsf{N_C}) \cup \{\top\}$, $D \in (\mathcal{BC}_{\mathcal{T}} \cap \mathsf{N_C}) \cup \{\top, \bot\}$ and all terms τ in $\mathcal{T}$ are either of the form $\{a\}$, with $a \in \mathsf{N_I}$, or of the form $\{\iota A\}$, with $A \in \mathsf{N_C}$. Given $A, B \in \mathsf{N_C}$, we may write $A \sqsubseteq B$ instead of $A \sqcap A \sqsubseteq B$. Moreover, if $\{\iota A\}$ occurs in $\mathcal{T}$ then we assume without loss of generality that $\{\iota A\} \sqsubseteq A \in \mathcal{T}$.

The *classification graph* for $\mathcal{T}$ is a tuple $(V, V \times V, S, R)$ where

- $V = \mathcal{BC}_{\mathcal{T}} \cup \{A^c \mid A \in (\mathcal{BC}_{\mathcal{T}} \cap \mathsf{N_C})\}$, with each $A^c \in \mathsf{N_C}$ fresh;
- S is a function mapping nodes in V to subsets of $\mathcal{BC}_{\mathcal{T}}^{+}$;
- R is a function mapping edges in $V \times V$ to (possibly empty) subsets of $\mathcal{R}_{\mathcal{T}}$.

Intuitively, a concept name of the form A^c represents a second element in the extension of A, and it is removed from the classification graph if A has at most one object in its extension. We write $C \rightsquigarrow_R D$ iff there are $C_1, \ldots, C_k \in \mathcal{BC}_{\mathcal{T}}$ such that $C_1 = C$; $R(C_j, C_{j+1}) \neq \emptyset$, for all $1 \leq j < k$; $C_k = D$. One can show that the label sets satisfy the following invariants:

- $D \in S(C)$ implies $\mathcal{T} \models C \sqsubseteq D$; and
- $r \in R(C, D)$ implies $\mathcal{T} \models C \sqsubseteq \exists r.D$.

Initially, we set $S(C) := \{C, \top\}$ for all nodes $C \in V$, and $R(C, D) := \emptyset$ for all edges $(C, D) \in V \times V$. If $C \in V \setminus \mathcal{BC}_{\mathcal{T}}$ is of the form A^c, with $A \in \mathsf{N_C}$, then we add A to $S(A^c)$. The above invariants are satisfied by these initial label sets. The completion rules are given in Table 1. Assume that rules are only applied if S or R or V change after the rule application. This bounds the number of rule applications to a polynomial in the number of concept and role names in $\mathcal{T}$. We assume that A is a special concept name we want to check for satisfiability (it appears in Rule R_{10} of Table 1). To show that subsumption in $\mathcal{ELO}^{\iota}$ can be decided in polynomial time one needs to show that if no more rules are applicable, then $\mathcal{T} \models A \sqsubseteq B$ iff $B \in S(A)$. We formalise this with Lemma 2.

Lemma 2. *Given a TBox $\mathcal{T}$, let S be the node function of a complete classification graph for $\mathcal{T}$ (cf. rules in Table 1). Then, $\mathcal{T} \models \mathsf{A} \sqsubseteq B$ iff $S(\mathsf{A}) \cap \{B, \bot\} \neq \emptyset$.*

Any $\mathcal{ELO}^{\iota}$ TBox $\mathcal{T}$ can be normalized (preserving satisfiability) in polynomial time and the classification graph for an $\mathcal{ELO}^{\iota}$ TBox $\mathcal{T}$ can be constructed in polynomial time with respect to the size of $\mathcal{T}$. Then, given arbitrary $\mathcal{ELO}^{\iota}$ concepts C, D and an $\mathcal{ELO}^{\iota}$ TBox $\mathcal{T}$, one can decide whether $C \sqsubseteq D$ is entailed by $\mathcal{T}$ by adding $\mathsf{A} \equiv C$ and $B \equiv D$ to $\mathcal{T}$, normalizing it, and then checking whether $S(\mathsf{A}) \cap \{B, \bot\} \neq \emptyset$, where A, B are fresh concept names (Lemma 2). As already mentioned, ABoxes can be encoded into the TBox and instance checking can be reduced to the entailment of CIs. Formula satisfiability can be divided

Table 1. The completion rules for subsumption in $\mathcal{ELO}^\iota$ with respect to TBoxes.

R_1:	if $C \sqcap D \sqsubseteq B \in \mathcal{T}$, $C, D \in S(E)$	then add B to $S(E)$
R_2:	if $C \sqsubseteq \exists r.D \in \mathcal{T}$, $C \in S(E)$	then add r to $R(E, D)$
R_3:	if $\exists r.C \sqsubseteq D \in \mathcal{T}$, $C \in S(B)$, $r \in R(E, B)$	then add D to $S(E)$
R_4:	if $\{\tau\} \in S(E) \cap S(D)$, $\mathsf{A} \leadsto_R D$;	then $S(E) := S(E) \cup S(D)$
R_5:	if $r \in R(E, D)$, $\bot \in S(D)$	then add $\bot$ to $S(E)$
R_6:	if $\{\tau\} \sqsubseteq D \in \mathcal{T}$, $\{\tau\} \in S(E)$	then add D to $S(E)$
R_7:	if $C \sqsubseteq \{\tau\} \in \mathcal{T}$, $C \in S(E)$	then add $\{\tau\}$ to $S(E)$
R_8:	if $\{\tau\} \in S(B)$, $B \in \mathsf{N_C}$	then $V := V \setminus \{B^c\}$
R_9:	if $B \in S(E)$, $B^c \notin V$	then add $\{\iota B\}$ to $S(E)$
R_{10}:	if $\mathsf{A} \leadsto_R C'$, $C \in S(C')$, $\{\iota B\} \in S(C)$	then $V := V \setminus \{B^c\}$

into two problems: one is the satisfiability of a propositional formula obtained by replacing each CI with a fresh propositional symbol, and the other is entailment of CIs in the $\mathcal{ELO}^\iota$ dimension (to ensure that the DL and the propositional parts of the formula are satisfiable together). The next theorem formalises these results.

Theorem 2. *The $\mathcal{ELO}^\iota$ formula satisfiability problem on partial interpretations is* NP*-complete and the entailment problem is* PTime*-complete.*

4 Bisimulations and Expressive Power

We discuss the expressive power of free DLs. In particular, we define a notion of bisimulation for $\mathcal{ALCO}^\iota$ that we use to characterise the expressive power of concepts relative to first-order formulas interpreted on partial interpretations. First, we present the definitions of bisimulation for $\mathcal{ALCO}$ and $\mathcal{ALCO}_u$, which are standard in the literature [1, 17], adapted to the case of partial interpretations.

Let $\mathcal{I}$ and $\mathcal{J}$ be partial interpretations, and let $\Sigma \subseteq \mathsf{N_C} \cup \mathsf{N_R} \cup \mathsf{N_I}$ be a *signature*. An $\mathcal{ALCO}$ *Σ-bisimulation* between $\mathcal{I}$ and $\mathcal{J}$ is a non-empty relation $Z \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{J}$ such that, for every $d \in \Delta^\mathcal{I}$ and $e \in \Delta^\mathcal{J}$ with $(d, e) \in Z$, every concept name or nominal X formulated within Σ, and every role name r in Σ: (*atom*) $d \in X^\mathcal{I}$ iff $e \in X^\mathcal{J}$; (*forth*) if $(d, d') \in r^\mathcal{I}$ then there is $e' \in \Delta^\mathcal{J}$ such that $(e, e') \in r^\mathcal{J}$ and $(d', e') \in Z$; and (*back*) if $(e, e') \in r^\mathcal{J}$ then there is $d' \in \Delta^\mathcal{I}$ such that $(d, d') \in r^\mathcal{I}$ and $(d', e') \in Z$. An $\mathcal{ALCO}_u$ *Σ-bisimulation* (between $\mathcal{I}$ and $\mathcal{J}$) is an $\mathcal{ALCO}$ Σ-bisimulation that is *total*, meaning that $\Delta^\mathcal{I}$ and $\Delta^\mathcal{J}$ are the domain and range of the relation.

Given a DL language $\mathcal{L}$, a signature Σ, and *pointed interpretations* $(\mathcal{I}, d)$ and $(\mathcal{J}, e)$, we write $(\mathcal{I}, d) \sim^\mathcal{L}_\Sigma (\mathcal{J}, e)$ if there is an $\mathcal{L}$ Σ-bisimulation Z between $\mathcal{I}$ and $\mathcal{J}$ such that $(d, e) \in Z$, and we say that $(\mathcal{I}, d)$ is $\mathcal{L}$ *Σ-bisimilar* to $(\mathcal{J}, e)$, or that Z is an $\mathcal{L}$ Σ-bisimulation between $(\mathcal{I}, d)$ and $(\mathcal{J}, e)$. We now introduce a suitable notion of bisimulation for $\mathcal{ALCO}^\iota$.

Definition 1 ($\mathcal{ALCO}^\iota$ Bisimulation). A relation $Z \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{J}$ is an $\mathcal{ALCO}^\iota$ *Σ-bisimulation* between $\mathcal{I}$ and $\mathcal{J}$ iff it is an $\mathcal{ALCO}$ Σ-bisimulation between $\mathcal{I}$ and $\mathcal{J}$ that satisfies, in addition, the following conditions, for every $(d, e) \in Z$:

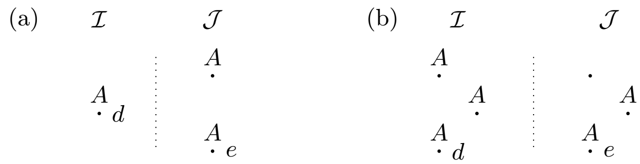

Fig. 1. (a) $\mathcal{ALCO}_u$ Σ-bisimilar but not $\mathcal{ALCO}^\iota$ Σ-bisimilar. (b) $\mathcal{ALCO}^\iota$ Σ-bisimilar but not $\mathcal{ALCO}_u$ Σ-bisimilar.

(ι.1) there exists $d' \in \Delta^{\mathcal{I}}$ such that $d \neq d'$ and $(\mathcal{I}, d) \sim_{\Sigma}^{\mathcal{ALCO}} (\mathcal{I}, d')$ iff there exists $e' \in \Delta^{\mathcal{J}}$ such that $e' \neq e$ and $(\mathcal{J}, e) \sim_{\Sigma}^{\mathcal{ALCO}} (\mathcal{J}, e')$;
(ι.2) if there is no $d' \in \Delta^{\mathcal{I}}$ such that $d \neq d'$ and $(\mathcal{I}, d) \sim_{\Sigma}^{\mathcal{ALCO}} (\mathcal{I}, d')$, then for every $u \in \Delta^{\mathcal{I}}$ there exists $v \in \Delta^{\mathcal{J}}$ such that $(u, v) \in Z$;
(ι.3) if there is no $e' \in \Delta^{\mathcal{J}}$ such that $e \neq e'$ and $(\mathcal{J}, e) \sim_{\Sigma}^{\mathcal{ALCO}} (\mathcal{J}, e')$, then for every $v \in \Delta^{\mathcal{J}}$ there exists $u \in \Delta^{\mathcal{I}}$ such that $(u, v) \in Z$.

Intuitively, Condition (ι.1) says that, for two $\mathcal{ALCO}^\iota$ Σ-bisimilar elements d and e, d has a distinct $\mathcal{ALCO}$ Σ-bisimilar object (representing an $\mathcal{ALCO}$ Σ-indistinguishable element in the same interpretation) iff e has one as well. Moreover, Condition (ι.2) (respectively, (ι.3)) states that if there is no distinct $\mathcal{ALCO}$ Σ-bisimilar object to $d \in \Delta^{\mathcal{I}}$ (respectively, $e \in \Delta^{\mathcal{J}}$), then the $\mathcal{ALCO}^\iota$ Σ-bisimulation relation is *left-total* (respectively, *right-total*), that is, $\Delta^{\mathcal{I}}$ (respectively, $\Delta^{\mathcal{J}}$) is the domain (respectively, range) of the relation.

The next theorem states that $\mathcal{ALCO}^\iota$ Σ-bisimilar elements satisfy the same $\mathcal{ALCO}^\iota$ concepts C formulated within Σ on partial interpretations (given a DL language $\mathcal{L}$, a signature Σ, and pointed interpretations $(\mathcal{I}, d)$ and $(\mathcal{J}, e)$, we write $(\mathcal{I}, d) \equiv_{\Sigma}^{\mathcal{L}} (\mathcal{J}, e)$ iff it holds that $d \in C^{\mathcal{I}}$ iff $e \in C^{\mathcal{J}}$, for every $\mathcal{L}$ concept C with $\Sigma_C \subseteq \Sigma$). Moreover, under the assumption that the partial interpretations satisfy the conditions of the class of *ω-saturated* interpretations [21] from model theory, the converse direction holds as well. These results are known for $\mathcal{ALCO}$, $\mathcal{ALCO}_u$ on total interpretations [17, Theorem 4.1.2] and can be adapted to the case with partial interpretations.

Theorem 3. *For all signatures Σ and all partial interpretations $\mathcal{I}$ and $\mathcal{J}$,*

(i) if $(\mathcal{I}, d) \sim_{\Sigma}^{\mathcal{ALCO}^\iota} (\mathcal{J}, e)$, then $(\mathcal{I}, d) \equiv_{\Sigma}^{\mathcal{ALCO}^\iota} (\mathcal{J}, e)$;
(ii) if $(\mathcal{I}, d) \equiv_{\Sigma}^{\mathcal{ALCO}^\iota} (\mathcal{J}, e)$ and $\mathcal{I}, \mathcal{J}$ are ω-saturated, then $(\mathcal{I}, d) \sim_{\Sigma}^{\mathcal{ALCO}^\iota} (\mathcal{J}, e)$.

Clearly, $\mathcal{ALCO}_u$ and $\mathcal{ALCO}^\iota$ are both more expressive than $\mathcal{ALCO}$. We now comment on the expressivity of $\mathcal{ALCO}_u$ and $\mathcal{ALCO}^\iota$. As illustrated in Figure 1, there are pointed interpretations $(\mathcal{I}, d)$ and $(\mathcal{J}, e)$ that are $\mathcal{ALCO}_u$ Σ-bisimilar but not $\mathcal{ALCO}^\iota$ Σ-bisimilar, and, $\mathcal{ALCO}^\iota$ Σ-bisimilar but not $\mathcal{ALCO}_u$ Σ-bisimilar, where $A \in \Sigma$. Since $\mathcal{ALCO}_u$ and $\mathcal{ALCO}^\iota$ are invariant under their respective notions of bisimulation, it follows that the expressivity of concept expressions in these languages is not comparable. There is no $\mathcal{ALCO}_u$ concept D equivalent to $\{\iota A\}$ and no $\mathcal{ALCO}^\iota$ concept D equivalent to $\forall u.A$.

Proposition 1. *The expressive power of $\mathcal{ALCO}_u$ and $\mathcal{ALCO}^\iota$ concepts is not comparable on partial (and on total) interpretations.*

We now characterise $\mathcal{ALCO}^\iota$ as the fragment of first-order logic on partial interpretations that is invariant under $\mathcal{ALCO}^\iota$ bisimulations. The *standard translation* of an $\mathcal{ALCO}^\iota$ concept C into a first-order formula $\pi_x(C)$ (with at most one free variable x) is defined as usual for concepts built using $\mathcal{ALCO}$ constructors. For nominals of the form $\{\iota C\}$ we have:

$$\pi_x(\{\iota C\}) = \exists x \pi_x(C) \land \forall x \forall y(\pi_x(C) \land \pi_y(C) \to x = y) \land \forall y(\pi_y(C) \to x = y).$$

We say that a first-order formula $\varphi(x)$ with free variable x and such that $\Sigma_\varphi \subseteq \Sigma$ is *invariant under* $\sim_\Sigma^{\mathcal{ALCO}^\iota}$ iff, for every $(\mathcal{I}, d)$ and $(\mathcal{J}, e)$ such that $(\mathcal{I}, d) \sim_\Sigma^{\mathcal{ALCO}^\iota} (\mathcal{J}, e)$, we have $\mathcal{I}, [x \mapsto d] \models \varphi(x)$ iff $\mathcal{J}, [x \mapsto e] \models \varphi(x)$ (where $[x \mapsto d]$ stands for any variable assignment that maps x to d).

Theorem 4. *Let $\varphi(x)$ be a first-order formula with one free variable x and such that $\Sigma_{\varphi(x)} \subseteq \Sigma$. The following conditions are equivalent:*

(i) there is an $\mathcal{ALCO}^\iota$ concept C such that $\pi_x(C)$ is equivalent to $\varphi(x)$;
(ii) $\varphi(x)$ is invariant under $\sim_\Sigma^{\mathcal{ALCO}^\iota}$.

We leave open the question of which notion can precisely capture $\mathcal{ELO}^\iota$. This notion should be less strict than $\mathcal{ALCO}^\iota$ bisimulations. One of the difficulties in finding it is that with $\{\iota C\}$ one can express a limited form of disjunction.

5 Related and Future Work

The DLs proposed in this article introduce a mild form of *cardinality constraints*, a set of constructors that has a long tradition in DL research [8, 40, 9, 6]. We refer the reader to [10] for a review of the state of the art. Using these formalisms, it is possible to constrain the number of elements in the extension of a concept. Thus, the $\mathcal{ALCO}^\iota$ instance query $\exists \mathsf{selects}.\mathsf{Reviewer}(\iota \exists \mathsf{isProgramChairOf}.\{\mathsf{dl20}\})$ can be captured by the $\mathcal{ALCO}$ CI $\exists \mathsf{isProgramChairOf}.\{\mathsf{dl20}\} \sqsubseteq \exists \mathsf{selects}.\mathsf{Reviewer}$, together with the requirement that $\exists \mathsf{isProgramChairOf}.\{\mathsf{dl20}\}$ has cardinality one. The expressivity of many of these logics goes far beyond the DLs proposed here, and novel reasoning tools are required. In contrast, we have shown that reasoning in the DLs considered here can be reduced to reasoning in standard DLs ($\mathcal{ALCO}^\iota$) or mild extensions ($\mathcal{ELO}^\iota$).

In computational linguistics, the *referring expression generation* (*REG*) problem is concerned with the (automatic) production of such noun phrases, so that they can be used to describe an entity in a given domain [36, 31, 30]. REG has been addressed in a DL setting as well, where the problem of finding a concept to describe an element is formulated with respect to a single interpretation, given as input [3, 2]. Other expressive DLs, as well as a relaxed version of the closed-world assumption, are considered in [38, 37]. Further research in the REG direction might involve adaptations of the algorithms proposed in these approaches to the case of partial interpretations and free DLs considered in our work.

Referring expressions are also relevant to other knowledge representation tasks, as in the case of *identity resolution* problems [12, 41, 42], or in *query-answering* over first-order and DL knowledge bases, where an approach allowing for referring expressions as answers to queries (in place of individual names only) has been recently proposed [13, 14]. The DLs considered in these papers are tractable languages tailored to efficient query answering in presence of functionality and path-based identification constraints. In [42], in particular, a knowledge base $\mathcal{K}$ consists of a TBox $\mathcal{T}$ and a finite set of concepts $\mathcal{C}$, called a *CBox*, introduced to replace the standard notion of an ABox. An interpretation $\mathcal{I}$ is a model of $\mathcal{K}$ iff $\mathcal{I}$ is a model of $\mathcal{T}$ and the extension in $\mathcal{I}$ of each concept in $\mathcal{C}$ has cardinality one. Moreover, given a conjunctive query φ with free variables $x_1, \ldots, x_n$, a finite list $(C_1, \ldots, C_n)$ of concepts in $\mathcal{C}$ is a *certain answer* to φ in $\mathcal{K}$ iff $\mathcal{K} \models \exists x_1 \ldots \exists x_n(\varphi \wedge C_1(x_1) \wedge \ldots \wedge C_n(x_n))$. Thus, in order to serve as referring expressions under a given knowledge base, these concepts have to satisfy an existence, a uniqueness, and a correctness (with respect to a query) condition. They are not, however, directly treated as possibly non-denoting terms of the language. We plan to explore further the connections with this approach.

Closely related are also the computation of *explicit definitions* of concepts and the *Beth definability property* (*BDP*) in DLs [17–19]. Unfortunately, it is known that the BDP fails for $\mathcal{ALCO}$, while it is regained if the use of individual names in definitions is not restricted and the language is extended with the @ operator from hybrid logic [20, 18]. Using recent results from [4], we plan to study how the BDP behaves in case of $\mathcal{ALCO}^{\iota}$ on partial interpretations, and to apply new techniques to find explicit definitions of concept names and nominals.

Finally, to the best of our knowledge, hybrid logics with non-denoting nominals have not received much attention in the literature, with the exception of [27] in the context of public announcement logics. On the other hand, formalisms involving definite descriptions, variously inspired by free logics in their accounts for non-denoting terms, have been extensively investigated in *first-order modal logic* [28, 24, 15, 26, 22, 25, 29, 34]. Here, the possible lack of referents for names and descriptions is usually paired with another feature, that of *non-rigid* denotation, i.e., the ability to refer to different objects at different states (time instants, epistemic alternatives, etc.). We intend to apply our framework for free DLs with definite descriptions to *modal* and *temporal extensions* as well, particularly in the context of query answering over temporal DL knowledge bases, where the interaction between denotation failure and non-rigidity can be at stake.

References

1. Areces, C., Blackburn, P., Marx, M.: Hybrid logics: Characterization, interpolation and complexity. J. Symb. Log. **66**(3), 977–1010 (2001)
2. Areces, C., Figueira, S., Gorín, D.: Using logic in the generation of referring expressions. In: LACL. pp. 17–32 (2011)
3. Areces, C., Koller, A., Striegnitz, K.: Referring expressions as formulas of description logic. In: INLG (2008)

4. Artale, A., Jung, J.C., Mazzullo, A., Ozaki, A., Wolter, F.: Living Without Beth and Craig: Explicit Definitions and Interpolants in Description Logics with Nominals. CoRR **abs/2007.02736** (2020), https://arxiv.org/abs/2007.02736
5. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
6. Baader, F., Bortoli, F.D.: On the expressive power of description logics with cardinality constraints on finite and infinite sets. In: FroCoS. pp. 203–219 (2019)
7. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: IJCAI. pp. 364–369 (2005)
8. Baader, F., Buchheit, M., Hollunder, B.: Cardinality restrictions on concepts. Artif. Intell. **88**(1-2), 195–213 (1996)
9. Baader, F., Ecke, A.: Extending the description logic $\mathcal{ALC}$ with more expressive cardinality constraints on concepts. In: GCAI. pp. 6–19 (2017)
10. Bednarczyk, B., Baader, F., Rudolph, S.: Satisfiability and query answering in description logics with global and local cardinality constraints. In: ECAI (2020), to appear
11. Bencivenga, E.: Free logics. In: Handbook of Philosophical Logic, pp. 147–196. Springer (2002)
12. Borgida, A., Toman, D., Weddell, G.E.: On referring expressions in information systems derived from conceptual modelling. In: ER. pp. 183–197 (2016)
13. Borgida, A., Toman, D., Weddell, G.E.: On referring expressions in query answering over first order knowledge bases. In: KR. pp. 319–328 (2016)
14. Borgida, A., Toman, D., Weddell, G.E.: Concerning referring expressions in query answers. In: IJCAI. pp. 4791–4795 (2017)
15. Braüner, T., Ghilardi, S.: First-order Modal Logic. In: Handbook of Modal Logic, pp. 549–620. Elsevier (2007)
16. Cann, R.: Formal Semantics: an Introduction. Cambridge University Press (1993)
17. ten Cate, B.: Model theory for Extended Modal Languages. Ph.D. thesis, University of Amsterdam (2005), ILLC Dissertation Series DS-2005-01
18. ten Cate, B., Conradie, W., Marx, M., Venema, Y.: Definitorially complete description logics. In: KR. pp. 79–89 (2006)
19. ten Cate, B., Franconi, E., Seylan, I.: Beth definability in expressive description logics. J. Artif. Intell. Res. **48**, 347–414 (2013)
20. ten Cate, B., Marx, M., Viana, J.P.: Hybrid logics with Sahlqvist axioms. Log. J. IGPL **13**(3), 293–300 (2005)
21. Chang, C.C., Keisler, H.J.: Model Theory. Elsevier (1990)
22. Fitting, M., Mendelsohn, R.L.: First-order Modal Logic. Springer Science & Business Media (2012)
23. Gabbay, D.M., Kurucz, A., Wolter, F., Zakharyaschev, M.: Many-dimensional Modal Logics: Theory and Applications. North Holland Publishing Company (2003)
24. Garson, J.W.: Quantification in modal logic. In: Handbook of philosophical logic, pp. 267–323. Springer (2001)
25. Garson, J.W.: Modal logic for Philosophers. Cambridge University Press (2013)
26. Goldblatt, R.: Quantifiers, propositions and identity: admissible semantics for quantified modal and substructural logics. No. 38, Cambridge University Press (2011)
27. Hansen, J.U.: A hybrid public announcement logic with distributed knowledge. Electr. Notes Theor. Comput. Sci. **273**, 33–50 (2011)

28. Hughes, G.E., Cresswell, M.J.: A New Introduction to Modal Logic. Routledge (1996)
29. Indrzejczak, A.: Cut-free modal theory of definite descriptions. In: AiML. pp. 387–406 (2018)
30. Krahmer, E., van Deemter, K.: Computational generation of referring expressions: A survey. Computational Linguistics **38**(1), 173–218 (2012)
31. Krahmer, E., van Erk, S., Verleg, A.: Graph-based generation of referring expressions. Computational Linguistics **29**(1), 53–72 (2003)
32. Lehmann, S.: More free logic. In: Handbook of Philosophical Logic, pp. 197–259. Springer (2002)
33. Neale, S.: Descriptions. MIT Press (1990)
34. Orlandelli, E., Corsi, G.: Labelled Calculi for Quantified Modal Logics with Non-rigid and Non-denoting Terms. In: ARQNL@IJCAR. pp. 64–78 (2018)
35. Passy, S., Tinchev, T.: An essay in combinatory dynamic logic. Inf. Comput. **93**(2), 263–332 (1991)
36. Reiter, E., Dale, R. (eds.): Building Natural Language Generation Systems. Cambridge University Press (2000)
37. Ren, Y., van Deemter, K., Pan, J.Z.: Charting the potential of description logic for the generation of referring expressions. In: INLG (2010)
38. Ren, Y., van Deemter, K., Pan, J.Z.: Generating referring expressions with OWL2. In: DL (2010)
39. Russell, B.: On Denoting. Mind **14**(56), 479–493 (1905)
40. Tobies, S.: The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. J. Artif. Intell. Res. **12**, 199–217 (2000)
41. Toman, D., Weddell, G.E.: Identity resolution in conjunctive querying over DL-based knowledge bases. In: DL (2018)
42. Toman, D., Weddell, G.E.: Identity resolution in ontology based data access to structured data sources. In: PRICAI Part I. pp. 473–485 (2019)