Two Types of Definite Descriptions: Theory and Implementation (Extended Abstract)

Michał Sochański[1], Przemysław Andrzej Wałęga[1,2] and Michał Zawidzki[1]

[1]University of Łódź
[2]Queen Mary University of London

Abstract
We investigate extensions of description logics with operators for definite descriptions—expressions intended to uniquely refer to elements based on their properties. Building on recent developments in both modal and description logics, we introduce two extensions of $\mathcal{ALC}$, namely $\mathcal{ALC}\iota^{\ell}$ and $\mathcal{ALC}\iota^{g}$, incorporating definite description operators from recent literature. We compare their expressive power and develop tableau-based decision procedures for both. Our ongoing work includes implementing these tableaux and conducting experimental evaluations to assess their practical viability in knowledge representation scenarios.

Keywords
Description Logics, Definite Descriptions, Tableau System

1. Introduction

In this ongoing research we study description logics with operators for definite descriptions, that is, for expressions aiming to refer to a single element by stating its unique property. As an example of such an expression, consider Bertrand Russell's famous 'the present king of France', which aims (unsuccessfully) to refer to a unique object [1]. Research on definite descriptions has a longstanding tradition [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] and, more recently, it has been conducted within the area of Knowledge Representation [14, 15, 16]. In this setting, definite descriptions are used for identifying objects in databases and for providing more informative answers to queries [17, 18, 19, 20].

In particular, definite descriptions have recently been introduced to description [21] and modal [22, 23] logics. The description logic $\mathcal{ALCO}_u^{\iota}$ [21] extends $\mathcal{ALC}$ with nominals, the universal role, and concepts for definite descriptions, which are of the form $\{\iota C\}$. The extension of $\{\iota C\}$ is a singleton that contains the unique element of the model of which C holds, or an empty set, if such an element does not exist. The modal logic $\mathcal{ML}(\mathsf{DD})$ [22, 23], introduces a different operator for definite descriptions, by allowing for operators of the form $@_{\varphi}$. A formula $@_{\varphi}\psi$, states that ψ holds in the unique modal world which satisfies φ.

Our current research aims to compare the above two types of definite descriptions in the common setting of description logics. To this end, we introduce two extensions of $\mathcal{ALC}$, namely $\mathcal{ALC}\iota^{\ell}$ and $\mathcal{ALC}\iota^{g}$ which allow for definite descriptions in the style of $\mathcal{ALCO}_u^{\iota}$ and $\mathcal{ML}(\mathsf{DD})$, respectively. We study the expressive power of these description logics and introduce tableau

DL 2025: 38th International Workshop on Description Logics, September 3–6, 2025, Opole, Poland

systems for them. Currently, we are working on an implementation of the tableaux and its experimental evaluation.

2. Description Logics with Definite Descriptions

In what follows, we will briefly introduce the syntax and semantics of $\mathcal{ALC}\iota^\ell$ and $\mathcal{ALC}\iota^g$, as well as $\mathcal{ALC}\iota$, which contains both.

As usual, 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 grammar of $\mathcal{ALC}\iota$ *concepts* C extends the grammar of $\mathcal{ALC}$ with two types of definite descriptions, namely $\{\iota C\}$ and $\iota C.D$. The first aims to represent the unique element that satisfies C, whereas the second aims to state that D holds for the unique element satisfying C. Hence, $\mathcal{ALC}\iota$ concepts are generated by the grammar

$$C \coloneqq A \mid \neg C \mid (C \sqcap C) \mid \exists r.C \mid \{\iota C\} \mid \iota C.C,$$

for $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. An $\mathcal{ALC}\iota$ *concept inclusion (CI)* is of the form $C \sqsubseteq D$, for any $\mathcal{ALC}\iota$ concepts C and D. An $\mathcal{ALC}\iota$ *term* τ is of the form a or ιC, for $a \in \mathsf{N_I}$, and C being an $\mathcal{ALC}\iota$ concept. An $\mathcal{ALC}\iota$ *assertion* is either of the form $C(\tau)$ or $r(\tau_1, \tau_2)$, for terms τ, τ_1, τ_2, concept C, and $r \in \mathsf{N_R}$. An $\mathcal{ALC}\iota$ ontology $\mathcal{O}$ is a finite set of $\mathcal{ALC}\iota$ CIs and assertions.

The description logic $\mathcal{ALC}\iota^\ell$ is the fragment of $\mathcal{ALC}\iota$ which allows for *local definite descriptions* $\{\iota C\}$, but not for *global* $\iota C.C$. Hence, the syntax is defined as for $\mathcal{ALC}\iota$, except that the grammar of $\mathcal{ALC}\iota^\ell$ concepts does not mention $\iota C.C$. As a result, $\mathcal{ALC}\iota^\ell$ corresponds to the fragment of $\mathcal{ALCO}^\iota_u$ without nominals and the universal role [18]. Description logic $\mathcal{ALC}\iota^g$, in turn, is the fragment of $\mathcal{ALC}\iota$ with only global descriptions, so it is obtained from $\mathcal{ALC}\iota$ by deleting $\{\iota C\}$ from the concepts and terms.

Semantics is obtained by extending the standard semantics of $\mathcal{ALC}$. An *interpretation* is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, of a non-empty *domain* $\Delta^\mathcal{I}$ and an *interpretation function* mapping atomic concepts $A \in \mathsf{N_C}$ to subsets of $\Delta^\mathcal{I}$, roles $r \in \mathsf{N_R}$ to subsets of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and individual names $a \in \mathsf{N_I}$ to elements in $\Delta^\mathcal{I}$. Function $\mathcal{I}$ extends to complex $\mathcal{ALC}$ concepts in a standard way, and for the concepts with definite descriptions as follows:

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

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

Satisfaction of concepts, axioms, and ontologies is defined in a standard way.

3. Expressive Power

In this section, we will compare the expressive power of $\mathcal{ALC}\iota^\ell$, $\mathcal{ALC}\iota^g$, $\mathcal{ALC}\iota$, and $\mathcal{ALCO}^\iota_u$. First we observe that $\{\iota C\}$ can be expressed as $C \sqcap \iota C.\top$. On the other hand, $\{\iota C\}$ can also be expressed as $A_C \sqcap \iota A_C.\top$, for a fresh concept name A_C, but we need to add axioms $A_C \sqsubseteq C$

and $C \sqsubseteq A_C$. The first translation preserves equivalence of concepts, but is exponential. The second is polynomial, but requires adding axioms, and does not preserve equivalence.

Proposition 1. *There is an exponential equivalence-preserving translation of $\mathcal{ALC}\iota^\ell$ to $\mathcal{ALC}\iota^g$ concepts, and a polynomial translation that maps $\mathcal{ALC}\iota^\ell$ ontologies into $\mathcal{ALC}\iota^g$ ontologies that conservatively extend the former.*

In order to study further the expressive power, we will adequately tailor the standard notion of bisimulation, so that it fits the new logics. To this end, we let *Names*$(\Delta', \mathcal{I})$, for a subset $\Delta' \subseteq \Delta^\mathcal{I}$ of the domain of $\mathcal{I}$, be the set of all $\mathcal{ALC}$ concepts C such that $C^\mathcal{I} = \{d\}$ for some $d \in \Delta'$. Now, we define the bisimulations as follows.

Definition 2. *An $\mathcal{ALC}\iota$ bisimulation between interpretations $\mathcal{I}$ and $\mathcal{J}$ is a standard $\mathcal{ALC}$ bisimulation $Z \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{J}$ (i.e., it satisfies conditions **Atom**, **Forth**, and **Back**) which additionally satisfies the new condition:*

Names *Names*$(\Delta^\mathcal{I}, \mathcal{I}) =$ *Names*$(\Delta^\mathcal{J}, \mathcal{J})$.

*An $\mathcal{ALC}\iota^\ell$ bisimulation is defined analogously with the caveat that **Names** is replaced by:*

Names$^\ell$ *Names*$(Dom(Z), \mathcal{I}) =$ *Names*$(Rng(Z), \mathcal{J})$.

For $\mathcal{L} \in \{\mathcal{ALC}\iota, \mathcal{ALC}\iota^\ell\}$ we write $\mathcal{I}, d \sim^\mathcal{L} \mathcal{J}, e$ if $(d, e) \in Z$ for some $\mathcal{L}$ bisimulation Z between $\mathcal{I}$ and $\mathcal{J}$.

Example 3. *We observe that some $\mathcal{ALC}\iota^\ell$ bisimulations are not $\mathcal{ALC}\iota$ bisimulatons. Consider $\mathcal{I}_1$ and $\mathcal{I}_2$ such that $\Delta^{\mathcal{I}_1} = (a, b)$ and $A^{\mathcal{I}_1} = \{\emptyset\}$, whereas $\Delta^{\mathcal{I}_2} = (c, d, e)$ and $A^{\mathcal{I}_2} = \{e\}$. The relation $Z = \{(a, c), (b, d))\}$ is an $\mathcal{ALC}\iota^\ell$ bisimulation, but not an $\mathcal{ALC}\iota$ bisimulation.*

It is worth observing that our bisimulations differ from those used for $\mathcal{ALCO}_u^\iota$ by Artale et al. [18], which rely on totality and "counting up to one". We can show that indeed, their conditions are too strong for $\mathcal{ALC}\iota^\ell$. The next result shows that our bisimulations behave similarly to the case of $\mathcal{ALC}$, that is, as intended. They preserve equivalence of concepts, and the opposite implication holds for ω-saturated interpretations.

Theorem 4. *For all pointed interpretations $(\mathcal{I}, d)$ and $(\mathcal{J}, e)$, and both $\mathcal{L} \in \{\mathcal{ALC}\iota, \mathcal{ALC}\iota^\ell\}$:*

1. *if $(\mathcal{I}, d) \sim^\mathcal{L} (\mathcal{J}, e)$, then $(\mathcal{I}, d) \equiv^\mathcal{L} (\mathcal{J}, e)$,*
2. *if $(\mathcal{I}, d) \equiv^\mathcal{L} (\mathcal{J}, e)$ and $\mathcal{I}, \mathcal{J}$ are ω-saturated, then $(\mathcal{I}, d) \sim^\mathcal{L} (\mathcal{J}, e)$,*

where $(\mathcal{I}, d) \equiv^\mathcal{L} (\mathcal{J}, e)$ means that both pointer interpretations satisfy the same $\mathcal{L}$-concepts.

We can use the above result to compare the expressive power of the described description logics mentioned, as stated below. In the following theorem, $A \leq B$ means that each concept of logic B can be translated into an equivalent concept of logic A. The relations $<$ and $=$ can be defined upon $\leq$ in a standard way.

Theorem 5. *The following expressive power results hold: $\mathcal{ALC}\iota^\ell < \mathcal{ALC}\iota^g = \mathcal{ALC}\iota$ and $\mathcal{ALCO}_u^\iota \not\leq \mathcal{ALC}\iota$.*

4. Tableau System

In this section we briefly introduce a tableau system for $\mathcal{ALC}\iota$, which provides us with a terminating, sound, and complete reasoning procedure. The rules of the tableau are presented in Figure 1. The main difference with respect to the standard tableau for $\mathcal{ALC}$ is that we introduce additional rules for global and local definite descriptions. For example, the rules for local descriptions $\{\iota C\}$ guarantee that if an element a satisfies $\{\iota C\}$, then it also satisfies C, and moreover, every other element a' satisfying C satisfies the same concepts as a (intuitively, a' and a need to refer to the same element). Furthermore, if a does not satisfy $\{\iota C\}$, then either it does not satisfy C, or some other element satisfies C. Finally, the cut rule allows us to determine if C or $\neg C$ holds in an arbitrary element, for any definite description $\{C\}$ occurring on the branch. We are currently working on an implementation of the tableau system in Python and its evaluation. In the basic form, the implementation will make it possible to check the satisfiability of a set of formulas, as well as of an ABox and TBox, as optional elements, and to inspect the model if it exists. We also intend to analyse the efficiency of our implementation depending on various syntactic features of input formulas, in particular those related to both kinds of definite descriptions. This will be done by generation of thousands of random formulas, checking their satisfiability using our implementation, and gathering data, such as the running times of the implementation on various inputs.

ABox rules:

$$(ABox_{|})\ \frac{a : C \in ABox}{a : C} \qquad (ABox_{|}^{\iota})\ \frac{\iota C : D \in ABox}{b_C : \{\iota C\}, b_C : D} \qquad (ABox_r)\ \frac{r(a, a') \in ABox}{r(a, a')}$$

TBox rule:

$$(TBox)\ \frac{C \sqsubseteq D \in TBox}{a : \neg(C \sqcap \neg D)}$$

$$(ABox_r^{\iota\ell})\ \frac{r(\iota C, a) \in ABox}{b_C : \{\iota C\}, r(b_C, a)} \qquad (ABox_r^{\iota r})\ \frac{r(a, \iota C) \in ABox}{b_C : \{\iota C\}, r(a, b_C)} \qquad (ABox_r^{\iota b})\ \frac{r(\iota C, \iota D) \in ABox}{b_C : \{\iota C\}, b_D : \{\iota D\}, r(b_C, b_D)}$$

Clash rule:

$$(\bot)\ \frac{a : C, a : \neg C}{\bot}$$

Propositional rules:

$$(\neg\neg)\ \frac{a : \neg\neg C}{a : C} \qquad (\sqcap)\ \frac{a : C \sqcap D}{a : C, a : D} \qquad (\neg\sqcap)\ \frac{a : \neg(C \sqcap D)}{a : \neg C \mid a : \neg D}$$

Role rules:

$$(\exists r)^1\ \frac{a : \exists r.C}{b : C, r(a, b)} \qquad (\neg\exists r)\ \frac{a : \neg\exists r.C, r(a, a')}{a' : \neg C}$$

Global definite description rules:

$$(\iota_1^g)^2\ \frac{a : \iota C.D}{b : C, b : D} \qquad (\iota_2^g)\ \frac{a : \iota C.D, a' : C, a'' : C, a' : E}{a'' : E} \qquad (\neg\iota^g)^3\ \frac{a : \neg\iota C.D}{a' : \neg C \mid a' : \neg D \mid \begin{array}{c} b : C, b : A_C^g, \\ b' : C, b : \neg A_C^g \end{array}} \qquad (cut_\iota^g)\ \frac{a : \iota C.D}{a' : C \mid a' : \neg C}$$

Local definite description rules:

$$(\iota_1^\ell)\ \frac{a : \{\iota C\}}{a : C} \qquad (\iota_2^\ell)\ \frac{a : \{\iota C\}, a' : C, a'' : C, a' : D}{a'' : D} \qquad (\neg\iota^\ell)^4\ \frac{a : \neg\{\iota C\}}{a : \neg C \mid \begin{array}{c} a : A_C, \\ b : C, b : \neg A_C \end{array}} \qquad (cut_\iota^\ell)\ \frac{a : \{\iota C\}}{a' : C \mid a' : \neg C}$$

---

[1] The rule is not applied if on the branch there is an individual a' such that for any concept $E \in \{C\} \cup \{\neg D \mid a : \neg\exists r.D \in \mathcal{B}\}, a' : E \in \mathcal{B}$.

[2] The rule is not applied if on the branch there is an individual a' such that either (1) $a' : C \in \mathcal{B}$ and $a' : D \notin \mathcal{B}$ or (2) $a' : C, a' : D \in \mathcal{B}$. In the former case $a'' : D$ is added to $\mathcal{B}$, where a'' is the first individual that occurred on $\mathcal{B}$ such that $a'' : C \in \mathcal{B}$. In the latter case no action is taken.

[3] A_C^g is an atomic concept not occurring in the input concept; if A_C^g or $\neg A_C^g$ occurs on the branch, the rule $(\neg\iota)$ cannot be applied to any premise of the form $a : \neg\iota C.E$ occurring on this branch, where E is an arbitrary concept.

[4] A_C is an atomic concept not occurring in the input concept; if there is an individual a' on the branch such that $a' : \neg A_C$ occurs on the branch, then in the left branch of the rule only the expression $a : A_C$ is introduced to the branch.

Figure 1: Rules of the tableau calculi for $\mathcal{ALC}\iota$

Acknowledgments

This research is funded by the European Union (ERC, ExtenDD, project number: 101054714). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] B. Russell, On denoting, Mind 14 (1905) 479–493.
[2] F. J. Pelletier, B. Linsky, What is Frege's theory of descriptions, in: G. Imagure, B. Linsky (Eds.), On Denoting: 1905–2005, Philosophia, München, 2005, pp. 195–250.
[3] D. Hilbert, P. Bernays, Grundlagen der Mathematik I, volume 40, 1968.
[4] J. B. Rosser, Logic for Mathematicians, 1978.
[5] K. Lambert, Free logic and definite descriptions, in: E. Morscher, A. Hieke (Eds.), New Essays in Free Logic, volume 23 of *Applied Logic Series*, Springer, Dordrecht, 2001, pp. 37–48.
[6] M. Fitting, R. L. Mendelsohn, First-order modal logic, volume 480 of *Synthese Library*, 2 ed., Springer Nature, Cham, 2023.
[7] A. Indrzejczak, M. Zawidzki, Tableaux for free logics with descriptions, in: Proc. of TABLEAUX, 2021, pp. 56–73.
[8] A. Indrzejczak, M. Zawidzki, When iota meets lambda, Synthese 201 (2023) 1–33.
[9] A. Indrzejczak, Fregean description theory in proof-theoretic setting, Logic and Logical Philosophy 28 (2019) 137–155.
[10] A. Indrzejczak, Russellian definite description theory—A proof-theoretic approach, The Review of Symbolic Logic 16 (2023) 624–649.
[11] E. Orlandelli, Labelled calculi for quantified modal logics with definite descriptions, Journal of Logic and Computation 31 (2021) 923–946.
[12] N. Kürbis, A binary quantifier for definite descriptions for cut free free logics, Studia Logica 110 (2022) 219–239.
[13] N. Kürbis, Definite descriptions in intuitionist positive free logic, Logic and Logical Philosophy 30 (2021) 327–358.
[14] A. Borgida, D. Toman, G. Weddell, On referring expressions in information systems derived from conceptual modelling, in: Proc. of ER, 2016, pp. 183–197.
[15] D. Toman, G. E. Weddell, Identity resolution in conjunctive querying over DL-based knowledge bases, in: Proc. of DL, 2018.
[16] D. Toman, G. Weddell, Identity resolution in ontology based data access to structured data sources, in: Proc. of PRICAI, 2019, pp. 473–485.
[17] A. Borgida, D. Toman, G. Weddell, On referring expressions in query answering over first order knowledge bases, in: Proc. of KR, 2016, pp. 319–328.
[18] A. Artale, A. Mazzullo, A. Ozaki, F. Wolter, On free description logics with definite descriptions, in: Proc. of KR, 2021, pp. 63–73.

[19] A. Borgida, D. Toman, G. E. Weddell, Concerning referring expressions in query answers, in: Proc. of IJCAI, 2017, pp. 4791–4795.
[20] D. Toman, G. Weddell, Finding all answers to OBDA queries using referring expressions, in: Proc. of AI, 2019, pp. 117–129.
[21] A. Mazzullo, Finite Traces and Definite Descriptions. A Knowledge Representation Journey, Ph.D. thesis, Free University of Bozen-Bolzano, 2022.
[22] P. A. Wałęga, M. Zawidzki, Hybrid modal operators for definite descriptions, in: Proc. of JELIA, 2023, pp. 712–726.
[23] P. A. Wałęga, Expressive power of definite descriptions in modal logics, in: Proc. of KR, 2024, pp. 687–696.