A connection method for a defeasible extension of $\mathcal{ALCH}$

Renan Fernandes[1[0000−0001−9553−5515]], Fred Freitas[1[0000−0003−0425−6786]], and Ivan Varzinczak[2,3,4[0000−0002−0025−9632]]

[1] Centro de Informática, Universidade Federal de Pernambuco, Brazil
`{rlf5,fred}@cin.ufpe.br`
[2] LIASD, Université Paris 8, France
[3] CAIR, University of Cape Town, South Africa
[4] ISTI–CNR, Italy
`ivan.varzinczak@univ-paris8.fr`

Abstract. In this paper, we propose a connection method à la Bibel for an exception-tolerant family of description logics (DLs). As for the language, we assume the DL $\mathcal{ALCH}$ extended with two kinds of typicality operators, namely one on (complex) concepts and one on role names. The language is a variant of defeasible DLs as broadly studied in the literature over the past decade and in which most of these can be embedded. We revisit the definition of a knowledge base matrix representation and establish the conditions for a given axiom to be provable from it. In particular, we show how term substitution is dealt with and we define a suitable condition of blocking in the presence of typicality operators. We show that the calculus terminates and that it is sound and complete w.r.t. a DL version of the preferential semantics widely adopted in non-monotonic reasoning.

1 Introduction

The problem of modelling exceptions in ontologies and reasoning meaningfully in their presence has received a great deal of attention over the past decade. Among the emblematic approaches put forward in the literature feature Giordano et al.'s description logics of typicality [22,23,26], Britz et al.'s defeasible subsumption relations [10,9], Bonatti et al.'s light-weight DLs of normality [5,4,6], Bozzato et al.'s reduction to Datalog [8], besides Casini and Straccia's seminal work on the computational counterpart of non-monotonic entailment in DLs [16,17] along with its implementation [15]. These investigations have given rise to a whole family of defeasible description logics of varying expressive power and with the ability to handle exceptions at both the modelling and the reasoning levels in a number of ways [3,11,12,14,18,30].

One of the interesting characteristics of some of the aforementioned approaches is the fact that depending on the underlying DL that is assumed and given certain conditions on how exceptionality (or typicality) is expressed, the kind of non-monotonic reasoning that is performed can be reduced to (a polynomial number of calls to) classical entailment check. Therefore, the study of the automated deduction for the various flavours of defeasible DLs and its potential reduction to classical reasoning remains a relevant and active research topic in logic-based artificial intelligence.

The development of proof methods for defeasible description logics has, in a sense, followed those for classical DLs. As a result, the overwhelming majority of existing decision procedures for reasoning with defeasible ontologies are based on semantic tableaux [11,13,23,25,31]. Notwithstanding the commonly extolled virtues of tableau systems, there are equally viable alternatives in the literature on automated theorem proving (ATP). One prominent example is the connection method (CM) [2], initially defined by W. Bibel in the late '70s, which earned a good reputation in the field of ATP in the '80s and '90s. In particular, the connection method has recently been revived in the context of (classical) modal and description logics [28,29,19,20].

The connection method consists of a direct proof procedure of which the main internal structure is a matrix representation of the knowledge base and associated query. It was designed to incorporate a more parsimonious usage of memory during proof searches. Indeed, contrary to tableaux and resolution, the connection method does not create intermediate clauses or sentences along the way, keeping its search space confined to the boundaries of the matrix it started off with. The first connection calculus for (classical) description logics, $\mathcal{ALC} - \theta\ CM$ [19], incorporates several features of most DL proof systems such as blocking, absence of variables, unification and Skolem functions. Moreover, a C++ implementation of the calculus, Raccoon [27], has been developed.[5] Worthy of mention is the fact that, despite incorporating none of the optimisations commonly done for DL tableaux systems, Raccoon performed competitively in reasoning over $\mathcal{ALC}$ ontologies when compared to cutting-edge highly-optimised tableau-based reasoners which had ranked high in past competitions of DL reasoners.[6]

In this paper, we provide a concrete calculus for a defeasible DL encompassing those frequently considered in the literature and that can be used to endow Raccoon with non-monotonic reasoning features. We hope our constructions will serve as a springboard for developing the connection method in defeasible DLs of varying expressive power and further extensions of Raccoon.

Hereafter, we assume the reader's familiarity with the well-established families of description logics. The remaining text is structured as follows: we start by presenting the defeasible extension of the DL $\mathcal{ALCH}$ we build on in this work; the following section is the heart of the paper and introduces our connection method for reasoning with defeasible ontologies, of which the inner workings are illustrated with a worked-out example in the next section; we then conclude the paper with a discussion and possible directions for further investigation.

[5] `https://github.com/dmfilho/raccoon`

[6] See `https://goo.gl/V9Ewkv` for details on the comparison.

2 The defeasible DL $\mathcal{ALCH}^\bullet$

The defeasible DL $\mathcal{ALCH}^\bullet$[31] is an extension of $\mathcal{ALCH}$[1] with typicality operators on both complex concepts and role names. Intuitively, a concept expression of the form $\bullet C$ denotes the most typical (alias normal) objects in the class C, whereas a role expression of the form $\bullet r$, with r an atomic role denotes the most typical instances of the relationship represented by r. To give a glimpse of $\mathcal{ALCH}^\bullet$'s expressive power, the axioms

$$\begin{gathered}\bullet\mathsf{Muggle} \sqsubseteq \neg\mathsf{Wizard},\\ \mathsf{HalfBloodWizard} \sqsubseteq \exists\mathsf{casts.Spell} \sqcap \neg\bullet\mathsf{Wizard}, \text{ and}\\ \bullet\mathsf{Wizard} \sqsubseteq \exists\bullet\mathsf{attachedWith.Wand}\end{gathered}$$

specify, respectively, that "typical muggles are not wizards", "half-blood wizards cast spells but are not typical wizards", and "typical wizards have a typical attachment with a wand". The RBox axiom $\mathsf{masterOf} \sqsubseteq \bullet\mathsf{attachedWith}$ states that "to be a master of (a wand) means to be typically attached with (that wand)". Furthermore, the ABox assertion $\neg\bullet\mathsf{attachedWith}(\mathsf{lordVoldemort}, \mathsf{elderWand})$ formalises the intuition that $\mathsf{lordVoldemort}$ is not typically attached with $\mathsf{elderWand}$.

We assume finite sets of concepts, roles, and individual names, denoted resp., with C, R, and I. We denote atomic concepts with $A, B, \ldots$, with $r, s, \ldots$ role names, and $a, b, \ldots$ individual names. Complex roles of $\mathcal{ALCH}^\bullet$ are denoted with $R, S, \ldots$ and defined by the rule:

$$R ::= \mathsf{R} \mid \bullet R$$

Complex concepts of $\mathcal{ALCH}^\bullet$ are denoted with $C, D, \ldots$ and are built according to the following grammar:

$$C ::= \mathsf{C} \mid \top \mid \bot \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall R.C \mid \exists R.C \mid \bullet C$$

The definitions of axiom, GCI, assertion, TBox, RBox (allowing for role subsumption axioms of the form $R \sqsubseteq S$) and ABox are as in the classical $\mathcal{ALCH}$ case. If $\mathcal{T}$, $\mathcal{R}$ and $\mathcal{A}$ are, respectively, a TBox, an RBox and an ABox, with $\mathcal{K} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$, we denote henceforth a <u>knowledge base</u> (alias ontology), frequently abbreviated as KB.

Example 1 (Wizarding-World Scenario). Assume we are interested in modelling facts about the wizarding-world and its wonderful features. We have the atomic concepts $\mathsf{C} = \{\mathsf{Muggle}, \mathsf{Wizard}, \mathsf{PureBloodWizard}\}$ representing, respectively, the class of muggles, wizards, and pure-blood wizards. As for the set of atomic roles, we have $\mathsf{R} = \{\mathsf{marriedTo}, \mathsf{hasPartner}\}$, representing a marriage and a partnership between two people (wizards or muggles). The set of individuals I is $\{\mathsf{hermione}, \mathsf{ronWeasley}\}$. Below is an example of an $\mathcal{ALCH}^\bullet$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ for the wizarding-world scenario.

$$\mathcal{T} = \left\{\begin{gathered}\bullet\mathsf{Muggle} \sqsubseteq \neg\mathsf{Wizard},\\ \mathsf{PureBloodWizard} \sqsubseteq \forall\mathsf{hasPartner.Wizard}\end{gathered}\right\}$$

$$\mathcal{R} = \big\{\mathsf{marriedTo} \sqsubseteq \bullet\mathsf{hasPartner}\big\}$$

$$\mathcal{A} = \left\{\begin{array}{c}\mathsf{Muggle(hermione)},\\ \mathsf{PureBloodWizard(ronWeasley)},\\ \mathsf{marriedTo(ronWeasley, hermione)}\end{array}\right\}$$

The semantics of $\mathcal{ALCH}^\bullet$ extends that of classical $\mathcal{ALCH}$ and is in terms of partially-ordered structures called bi-ordered interpretations. Before introducing these, we recall a few notions.

A binary relation is a strict partial order if it is irreflexive and transitive. If $<$ is a strict partial order on a given set X, with $\min_< X \stackrel{\text{def}}{=} \{x \in X \mid \text{there is no } y \in X \text{ s.t. } y < x\}$ we denote the minimal elements of X w.r.t. $<$. A strict partial order on a set X is well-founded if, for every $\emptyset \neq X' \subseteq X$, we have $\min_< X' \neq \emptyset$.

Definition 1 (Bi-ordered interpretation). *An $\mathcal{ALCH}^\bullet$ bi-ordered interpretation is a tuple $\mathcal{O} \stackrel{\text{def}}{=} \langle \Delta^\mathcal{O}, \cdot^\mathcal{O}, <^\mathcal{O}, \ll^\mathcal{O}\rangle$ s.t. $\langle \Delta^\mathcal{O}, \cdot^\mathcal{O}\rangle$ is a classical $\mathcal{ALCH}$ interpretation, and both $<^\mathcal{O} \subseteq \Delta^\mathcal{O} \times \Delta^\mathcal{O}$ and $\ll^\mathcal{O} \subseteq (\Delta^\mathcal{O} \times \Delta^\mathcal{O}) \times (\Delta^\mathcal{O} \times \Delta^\mathcal{O})$ are well-founded strict partial orders.*

Given $\mathcal{O} = \langle \Delta^\mathcal{O}, \cdot^\mathcal{O}, <^\mathcal{O}, \ll^\mathcal{O}\rangle$, the intuition of $\Delta^\mathcal{O}$ and $\cdot^\mathcal{O}$ is the same as in a standard $\mathcal{ALCH}$ interpretation. The intuition underlying the orderings $<^\mathcal{O}$ and $\ll^\mathcal{O}$ is that they play the role of preference relations (or normality orderings): the objects (resp. pairs) that are lower down in the ordering $<^\mathcal{O}$ (resp. $\ll^\mathcal{O}$) are deemed more normal (or typical) than those higher up in $<^\mathcal{O}$ (resp. $\ll^\mathcal{O}$). Within the context of (the interpretation of) a concept C (resp. role R), $<^\mathcal{O}$ (resp. $\ll^\mathcal{O}$) therefore allows us to single out the most normal representatives falling under C (resp. R), which is the intuition of the semantics of concepts (resp. roles) of the form $\bullet C$ (resp. $\bullet R$):

Definition 2 (Semantics). *A bi-ordered interpretation $\mathcal{O} = \langle \Delta^\mathcal{O}, \cdot^\mathcal{O}, <^\mathcal{O}, \ll^\mathcal{O}\rangle$ interprets the classical constructors in the usual way. The typicality-based concepts are interpreted as $(\bullet C)^\mathcal{O} \stackrel{\text{def}}{=} \min_{<^\mathcal{O}} C^\mathcal{O}$, and roles as $(\bullet R)^\mathcal{O} \stackrel{\text{def}}{=} \min_{\ll^\mathcal{O}} R^\mathcal{O}$.*

Hence, to be a typical element of a concept (resp. role) amounts to being one of the most preferred elements in the interpretation of that concept (resp. role). It is easy to see that the typicality operators are both idempotent.

The definition of satisfaction of a statement α by a bi-ordered interpretation $\mathcal{O}$, denoted as $\mathcal{O} \Vdash \alpha$, carries over from the classical case. If X is a set of statements, with $\mathcal{O} \Vdash X$, we denote the fact $\mathcal{O}$ satisfies each statement in X, in which case we say $\mathcal{O}$ is a model of X. We say $\mathcal{O}$ is a model of a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$, denoted $\mathcal{O} \Vdash \mathcal{K}$, if $\mathcal{O}$ is a model of $\mathcal{T}$, $\mathcal{R}$, and $\mathcal{A}$.

Given a knowledge base $\mathcal{K}$ and a statement α, we say $\mathcal{K}$ preferentially entails α, denoted $\mathcal{K} \models \alpha$, if, for every bi-ordered interpretation $\mathcal{O}$ such that $\mathcal{O} \Vdash \mathcal{K}$, we have $\mathcal{O} \Vdash \alpha$.

3 A formula representation

The connection method represents the facts as matrices. In its classical approach to first-order logic, those facts are formulae, and we will see formulae in detail in this section.

3.1 Vocabulary

The vocabulary to build formulae contains a set V for variables; for all integer $n \geq 1$, a set of n-ary Skolem functions F_n; and the previously defined sets C, R, and I.

Definition 3 (Term). *A <u>term</u> is recursively defined as follows:*

1. *A variable is a term;*
2. *An individual name is a term;*
3. *Given a n-ary function f, and terms $t_1, \cdots, t_n$, $f(t_1, \cdots, t_n)$ is a term. With T, we represent the set of terms of a language.*

Definition 4 (Literal). *A <u>literal</u> is defined as follows:*

$$L ::= \mathsf{C}(\mathsf{T}) \mid \mathsf{R}(\mathsf{T}, \mathsf{T}) \mid \mathsf{T} < \mathsf{T} \mid (\mathsf{T}, \mathsf{T}) \ll (\mathsf{T}, \mathsf{T}) \mid \neg L$$

Definition 5 (Formula). *A <u>formula</u> is built upon the constructors $\wedge$, $\vee$, $\neg$, $\exists$, and $\forall$ as follows:*

$$F ::= L \mid F \wedge F \mid F \vee F \mid \neg F \mid \exists \mathsf{V} F \mid \forall \mathsf{V} F$$

Definition 6 (Semantics). *An <u>interpretation</u> $\mathcal{I}$ is a pair $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is the domain of interpretation, and $\cdot^{\mathcal{I}}$ is the interpretation function that interprets matrix elements as:*

- *$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, for each individual name $a \in \mathsf{I}$;*
- *$A^{\mathcal{I}} : \Delta^{\mathcal{I}} \longrightarrow \{\mathsf{true}, \mathsf{false}\}$, for each concept name $A \in \mathsf{C}$;*
- *$r^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \longrightarrow \{\mathsf{true}, \mathsf{false}\}$, for each role name $r \in \mathsf{R}$;*
- *$<^{\mathcal{I}}: \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \longrightarrow \{\mathsf{true}, \mathsf{false}\}$;*
- *$\ll^{\mathcal{I}}: (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \times (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \longrightarrow \{\mathsf{true}, \mathsf{false}\}$;*
- *$f^{\mathcal{I}} : (\Delta^{\mathcal{I}})^n \longrightarrow \Delta^{\mathcal{I}}$, for each $n \geq 1$ and Skolem function $f \in \mathsf{F}_n$.*

Definition 7 (Satisfaction). *Given a formula F and an interpretation $\mathcal{I}$, we say that $\mathcal{I}$ <u>satisfies</u> F, denoted $\mathcal{I} \Vdash F$, if:*

- *if $F = \top$, then $\top^{\mathcal{I}} = \mathsf{true}$;*
- *if $F = \bot$, then $\bot^{\mathcal{I}} = \mathsf{false}$;*
- *if $F = A(t)$, then $\mathcal{I} \Vdash F$ if $A^{\mathcal{I}}(t^{\mathcal{I}}) = \mathsf{true}$;*
- *if $F = r(t, u)$, then $\mathcal{I} \Vdash F$ if $r^{\mathcal{I}}(t^{\mathcal{I}}, u^{\mathcal{I}}) = \mathsf{true}$;*
- *if $F = t < u$, then $\mathcal{I} \Vdash F$ if $<^{\mathcal{I}} (t^{\mathcal{I}}, u^{\mathcal{I}}) = \mathsf{true}$;*
- *if $F = (t, u) \ll (v, k)$, then $\mathcal{I} \Vdash F$ if $\ll^{\mathcal{I}} (t^{\mathcal{I}}, u^{\mathcal{I}}, v^{\mathcal{I}}, k^{\mathcal{I}}) = \mathsf{true}$;*
- *if $F = \neg F_1$, then $\mathcal{I} \Vdash F$ if $F_1^{\mathcal{I}} = \mathsf{false}$;*

- *if* $F = F_1 \vee F_2$, *then* $\mathcal{I} \Vdash F$ *if* $\mathcal{I} \Vdash F_1$, *or* $\mathcal{I} \Vdash F_2$, *or both;*
- *if* $F = F_1 \wedge F_2$, *then* $\mathcal{I} \Vdash F$ *if* $\mathcal{I} \Vdash F_1$ *and* $\mathcal{I} \Vdash F_2$;
- *if* $F = \forall x F_1$, *then* $\mathcal{I} \Vdash F$ *if* $\mathcal{I}' \Vdash F_1$, *for every extension* $\mathcal{I}'$ *that coincides with* $\mathcal{I}$ *but maps* x *to different objects* $x^{\mathcal{I}}$ *of* $\Delta^{\mathcal{I}}$;
- *if* $F = \exists x F_1$, *then* $\mathcal{I} \Vdash F$ *if* $\mathcal{I}' \Vdash F_1$, *for some extension* $\mathcal{I}'$ *that coincides with* $\mathcal{I}$ *but maps* x *to different objects* $x^{\mathcal{I}}$ *of* $\Delta^{\mathcal{I}}$.

The definitions above recall the well-known classical first-order logic. In our case, the formulae are simpler than the full first-order once there is no functions (besides Skolem functions) or predicates with more than two terms.

4 Mapping from $\mathcal{ALCH}^{\bullet}$ to matrices

Definition 8 (Translation between roles and formulae). *Given the terms* t, u, *the* *translation function* $\pi_{t,u}$ *maps* $\mathcal{ALCH}^{\bullet}$*-roles to formulae with variables* x *and* y *as follows:*

$$\begin{aligned}
\pi_{t,u}(r) &\stackrel{\text{def}}{=} r(t,u),\\
\pi_{t,u}(\neg r) &\stackrel{\text{def}}{=} \neg r(t,u),\\
\pi_{t,u}(\bullet r) &\stackrel{\text{def}}{=} r(t,u) \wedge \forall x \forall y \left(\neg((x,y) \ll (t,u)) \vee \neg r(x,y)\right),\\
\pi_{t,u}(\neg\bullet r) &\stackrel{\text{def}}{=} \neg r(t,u) \vee \exists x \exists y \left((x,y) \ll (t,u) \wedge \pi_{x,y}(\bullet r)\right),
\end{aligned}$$

Definition 9 (Translation between concepts and formulae). *Given a term* t, *the* *translation function* π_t *maps* $\mathcal{ALCH}^{\bullet}$*-concepts to formulae with a variable* y *as follows:*

$$\begin{aligned}
\pi_t(A) &\stackrel{\text{def}}{=} A(t)\\
\pi_t(D \sqcap E) &\stackrel{\text{def}}{=} \pi_t(D) \wedge \pi_t(E)\\
\pi_t(D \sqcup E) &\stackrel{\text{def}}{=} \pi_t(D) \vee \pi_t(E)\\
\pi_t(\exists R.D) &\stackrel{\text{def}}{=} \exists y \left[\pi_{t,y}(R) \wedge \pi_y(D)\right]\\
\pi_t(\forall R.D) &\stackrel{\text{def}}{=} \forall y \left[\pi_{t,y}(\neg R) \vee \pi_y(D)\right]\\
\pi_t(\bullet D) &\stackrel{\text{def}}{=} \pi_t(D) \wedge \forall y \left[\neg(y < t) \vee \pi_y(\neg D)\right]\\
\pi_t(\neg\bullet D) &\stackrel{\text{def}}{=} \pi_t(\neg D) \vee \exists y \left[(y < t) \wedge \pi_y(\bullet D)\right]
\end{aligned}$$

Definition 10 (Translation between knowledge bases and formulae). *The translation function π maps a knowledge base $\mathcal{K}$ to formulae as follows:*

$$\pi(\mathcal{K}) \stackrel{\text{def}}{=} \bigwedge_{a:D\in\mathcal{K}} \pi_a(D) \wedge \bigwedge_{(a,b):R\in\mathcal{K}} \pi_{a,b}(R) \bigwedge_{D\sqsubseteq E\in\mathcal{K}} \forall x\,[\pi_x(\neg D \sqcup E)] \tag{1}$$

$$\bigwedge_{R\sqsubseteq S\in\mathcal{K}} \forall x \forall y\,[\pi_{x,y}(\neg R) \vee \pi_{x,y}(E))] \tag{2}$$

$$\wedge\, \forall x \forall y \forall z\,[\neg(x < y \wedge y < z) \vee x < z] \tag{3}$$

$$\wedge\, \forall x\,[\neg(x < x)] \tag{4}$$

$$\wedge\, \forall x \forall y\,[\neg(x < y) \vee \neg(y < x)] \tag{5}$$

$$\wedge\, \forall x \forall y \forall z \forall k \forall m \forall n\,[\neg((x,y) \ll (z,k) \wedge (z,k) \ll (m,n)) \vee (x,y) \ll (m,n)] \tag{6}$$

$$\wedge\, \forall x \forall y\,[\neg((x,y) \ll (x,y))] \tag{7}$$

$$\wedge\, \forall x \forall y \forall z \forall k\,[\neg((x,y) \ll (z,k)) \vee \neg((z,k) \ll (x,y))] \tag{8}$$

The previous definition maps each axiom to an atomic subformula (Eq. 1 and Eq. 2) from the conjunction defined above. The other conjunction's subformulae are not related to the knowledge base; they consist of transitivity (Eq. 3 and Eq. 6), irreflexivity (Eq. 4 and Eq. 7) and asymmetry (Eq. 5 and Eq. 8) axioms for $<$ and $\ll$, to ensure that such relations are strict partial orders. Hereafter, we omit those axioms unless we use them in proof.

Corollary 1. *Given a knowledge base $\mathcal{K}$ and a bi-ordered interpretation $\mathcal{O}$, if $\mathcal{O} \Vdash \mathcal{K}$, then there exists an interpretation $\mathcal{I}$ such that $\mathcal{I} \Vdash \pi(\mathcal{K})$*

Proof. This proof is a long one and, thus, is split into three parts. In the first part, we build an interpretation based on the bi-ordered interpretation $\mathcal{O}$. The second part contains the proofs for concept and role assertions. The last one is related to TBox axioms as such concept and role subsumption.

Definition of an interpretation To show this corollary, we start by building an interpretation $\mathcal{I}$ such that:

1. $\Delta^{\mathcal{I}}$ is the domain $\Delta^{\mathcal{O}}$;
2. $a^{\mathcal{I}} = a^{\mathcal{O}}$, for all individual names a;
3. for each concept name $A \in \mathsf{C}$ and each object $o \in \Delta^{\mathcal{O}}$,

$$A^{\mathcal{I}}(o) = \begin{cases} \mathsf{true}, \text{ if } o \in A^{\mathcal{O}} \\ \mathsf{false}, \text{ otherwise} \end{cases} \quad ;$$

4. for each role name $r \in \mathsf{R}$ and each pair of objects $(o_1, o_2) \in \Delta^{\mathcal{O}} \times \Delta^{\mathcal{O}}$,

$$r^{\mathcal{I}}(o_1, o_2) = \begin{cases} \mathsf{true}, \text{ if } (o_1, o_2) \in r^{\mathcal{O}} \\ \mathsf{false}, \text{ otherwise} \end{cases} \quad ;$$

5. for each pair $(o_1, o_2) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

$$<^{\mathcal{I}} (o_1, o_2) = \begin{cases} \mathsf{true}, \text{ if } (o_1, o_2) \in (<^{\mathcal{O}})^+ \\ \mathsf{false}, \text{ otherwise} \end{cases} \quad ; \text{ and}$$

6. for each pair of pairs of objects $((o_1, o_2), (o_3, o_4)) \in (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \times (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}})$

$$\ll^{\mathcal{I}} ((o_1, o_2), (o_3, o_4)) = \begin{cases} \mathsf{true}, \text{ if } ((o_1, o_2), (o_3, o_4)) \in (\ll^{\mathcal{O}})^+ \\ \mathsf{false}, \text{ otherwise} \end{cases}$$

This interpretation is suitable once there is no $A^{\mathcal{I}}(o)$ both true and false, otherwise $\mathcal{O}$ could not satisfy $\mathcal{K}$. Furthermore, $\mathcal{I}$ satisfies the transitivity, irreflexivity, and asymmetry axioms for both $<^{\mathcal{I}}$ and $\ll^{\mathcal{I}}$ since $<^{\mathcal{O}}$ and $\ll^{\mathcal{O}}$ are strict partial orders, by definition. To illustrate those properties, assume that $\mathcal{I}$ does not satisfy the irreflexivity axiom for $<$. Then, $\mathcal{I} \not\Vdash \forall x.\neg(x < x)$, which means there exists an object $o \in \Delta^{\mathcal{I}}$ such that $<^{\mathcal{I}} (o, o) = \mathsf{true}$. However, by definition of $\mathcal{I}$, $<^{\mathcal{I}} (o, o)$ must be false, as there is no $(o, o) \in <^{\mathcal{O}}$ because $<^{\mathcal{O}}$ is a strict partial order. The other properties are obtained in a similar manner and are left to the reader.

Proof for role assertions We here prove satisfaction for role assertions by induction on the structure of formulae, such that

$$[\pi_{a,b}(R)]^{\mathcal{I}} = \begin{cases} \mathsf{true}, \text{ if } (a^{\mathcal{O}}, b^{\mathcal{O}}) \in R^{\mathcal{O}} \\ \mathsf{false}, \text{ otherwise.} \end{cases} \quad .$$

The assertions which are related to role names or their negation constitute the induction base cases as follows:

- $R = r$. Then, $(a^{\mathcal{O}}, b^{\mathcal{O}}) \in r^{\mathcal{O}}$ and, by definition of $\mathcal{I}$ above, $r^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) = \mathsf{true}$; and
- $R = \neg r$. Then, $(a^{\mathcal{O}}, b^{\mathcal{O}}) \notin r^{\mathcal{O}}$ and, by definition of $\mathcal{I}$ above, $r^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) = \mathsf{false}$.

Induction steps:

- $R = \neg S$. Then, $(a^{\mathcal{O}}, b^{\mathcal{O}}) \notin S^{\mathcal{O}}$. By the IH, $[\pi_{a,b}(S)]^{\mathcal{I}} = \mathsf{false}$. Therefore, $[\pi_a(\neg S)]^{\mathcal{I}} = \mathsf{true}$;
- $R = \bullet S$. Then, $(a^{\mathcal{O}}, b^{\mathcal{O}}) \in S^{\mathcal{O}}$ and, for every pair $o_1, o_2 \in \Delta^{\mathcal{O}} \times \Delta^{\mathcal{O}}$, either $((o_1, o_2), (a^{\mathcal{O}}, b^{\mathcal{O}})) \notin (\ll^{\mathcal{O}})^+$, or $(o_1, o_2) \in (\neg S)^{\mathcal{O}}$. By the IH, the definition of $\mathcal{I}$, and the semantics of formula, we have that $[\pi_{a,b}(S)]^{\mathcal{I}} = \mathsf{true}$, and $[\forall x \forall y.\neg((x, y) \ll (a, b)) \vee \pi_{x,y}(\neg S)]^{\mathcal{I}} = \mathsf{true}$ as well. Therefore, $[\pi_{a,b}(S) \wedge \forall x \forall y.\neg((x, y) \ll (a, b)) \vee \pi_{x,y}(\neg S)]^{\mathcal{I}} = \mathsf{true}$ even as $[\pi_{a,b}(\bullet S)]^{\mathcal{I}} = \mathsf{true}$;
- $R = \neg\bullet S$. Then, $(a^{\mathcal{O}}, b^{\mathcal{O}}) \notin S^{\mathcal{O}}$ or, for some pair $o_1, o_2 \in \Delta^{\mathcal{O}} \times \Delta^{\mathcal{O}}$, $((o_1, o_2), (a^{\mathcal{O}}, b^{\mathcal{O}})) \in (\ll^{\mathcal{O}})^+$ and $(o_1, o_2) \in S^{\mathcal{O}}$. By the IH, the definition of $\mathcal{I}$, and the semantics of formula, we have that $[\pi_{a,b}(\neg S)]^{\mathcal{I}} = \mathsf{true}$, or $[\exists x \exists y.((x, y) \ll (a, b)) \wedge \pi_{x,y}(S)]^{\mathcal{I}} = \mathsf{true}$, or both. Therefore, $[\pi_{a,b}(\neg S) \vee \exists x \exists y.((x, y) \ll (a, b)) \wedge \pi_{x,y}(S)]^{\mathcal{I}} = \mathsf{true}$, meaning that $[\pi_{a,b}(\neg\bullet S)]^{\mathcal{I}} = \mathsf{true}$, too.

The proof for concept assertions We here prove the satisfaction for concept assertions, such that

$$[\pi_a(D)]^{\mathcal{I}} = \begin{cases} \textsf{true}, \text{ if } a^{\mathcal{O}} \in D^{\mathcal{O}} \\ \textsf{false}, \text{ otherwise} \end{cases}$$

The assertion related to concept names is the induction base cases. Then, $a^{\mathcal{O}} \in A^{\mathcal{O}}$ and, by definition of $\mathcal{I}$ above, $A^{\mathcal{I}}(a^{\mathcal{I}}) = \textsf{true}$.

Induction steps:

- $D = \neg E$. Then, $a^{\mathcal{O}} \in \Delta^{\mathcal{O}} \setminus E^{\mathcal{O}}$ which means that $a^{\mathcal{O}} \notin E^{\mathcal{O}}$. By the IH, $[\pi_a(E)]^{\mathcal{I}} = \textsf{false}$. Therefore, $[\pi_a(\neg E)]^{\mathcal{I}} = \textsf{true}$;
- $D = E_1 \sqcup E_2$. Then, $a^{\mathcal{O}} \in E_1^{\mathcal{O}} \cup E_2^{\mathcal{O}}$ which means that either $a^{\mathcal{O}} \in E_1^{\mathcal{O}}$, $a^{\mathcal{O}} \in E_2^{\mathcal{O}}$, or both. By the IH, either $[\pi_a(E_1)]^{\mathcal{I}} = \textsf{true}$, $[\pi_a(E_2)]^{\mathcal{I}} = \textsf{true}$, or both. Therefore, $[\pi_a(E_1 \sqcup E_2)]^{\mathcal{I}} = \textsf{true}$;
- $D = E_1 \sqcap E_2$. Then, $a^{\mathcal{O}} \in E_1^{\mathcal{O}} \cap E_2^{\mathcal{O}}$ which means that $a^{\mathcal{O}} \in E_1^{\mathcal{O}}$ and $a^{\mathcal{O}} \in E_2^{\mathcal{O}}$. By the IH, $[\pi_a(E_1)]^{\mathcal{I}} = \textsf{true}$ and $[\pi_a(E_2)]^{\mathcal{I}} = \textsf{true}$. Therefore, $[\pi_a(E_1 \sqcap E_2)]^{\mathcal{I}} = \textsf{true}$;
- $D = \exists R.E$. Then, there exists an object x such that $(a^{\mathcal{O}}, x) \in R^{\mathcal{O}}$ and $x \in E^{\mathcal{O}}$. By the IH, $[\exists y.\pi_{a,y}(R) \wedge \pi_y(E)]^{\mathcal{I}} = \textsf{true}$. Therefore, $[\pi_a(\exists R.E)]^{\mathcal{I}} = \textsf{true}$;
- $D = \forall R.E$. Then, for all objects x, $(a^{\mathcal{O}}, x) \notin R^{\mathcal{O}}$ or $x \in E^{\mathcal{O}}$. By the IH, $[\forall y.\pi_{a,y}(\neg R) \vee \pi_y(E)]^{\mathcal{I}} = \textsf{true}$. Therefore, $[\pi_a(\forall R.E)]^{\mathcal{I}} = \textsf{true}$;
- $D = \bullet E$. Then, $a^{\mathcal{O}} \in E^{\mathcal{O}}$ and, for all object $o \in \Delta^{\mathcal{O}}$, either $(o, a^{\mathcal{O}}) \notin (<^{\mathcal{O}})^+$, or $o \in (\neg E)^{\mathcal{O}}$. By the IH, the definition of $\mathcal{I}$, and the semantics of formula, we have that $[\pi_a(E)]^{\mathcal{I}} = \textsf{true}$, and $[\forall x.\neg(x < a) \vee \pi_x(\neg E)]^{\mathcal{I}} = \textsf{true}$ as well. Therefore, $[\pi_a(E) \wedge \forall x.\neg(x < a) \vee \pi_x(\neg E)]^{\mathcal{I}} = \textsf{true}$, also $[\pi_a(\bullet E)]^{\mathcal{I}} = \textsf{true}$ too;
- $D = \neg\bullet E$. Then, $a^{\mathcal{O}} \notin E^{\mathcal{O}}$ or, for some object $o \in \Delta^{\mathcal{O}}$, $(o, a^{\mathcal{O}}) \in (<^{\mathcal{O}})^+$ and $o \in E^{\mathcal{O}}$. By the IH, the definition of $\mathcal{I}$, and the semantics of formula, we have that $[\pi_a(\neg E)]^{\mathcal{I}} = \textsf{true}$, $[\exists x.x < a \wedge \pi_x(E)]^{\mathcal{I}} = \textsf{true}$, or both. Therefore, $[\pi_a(\neg E) \vee \exists x.x < a \wedge \pi_x(E)]^{\mathcal{I}} = \textsf{true}$, meaning that $[\pi_a(\neg\bullet E)]^{\mathcal{I}} = \textsf{true}$ too.

The proof for role subsumption We prove that, given an axiom $R \sqsubseteq S \in \mathcal{K}$,

$$[\forall x \forall y[\pi_{x,y}(\neg R) \vee \pi_{x,y}(S)]]^{\mathcal{I}} = \begin{cases} \textsf{true}, \text{ if } R^{\mathcal{O}} \subseteq S^{\mathcal{O}} \\ \textsf{false}, \text{ otherwise} \end{cases}$$

by contrapositive. Assume that there exists a pair of objects $(o_1, o_2) \in \Delta^{\mathcal{O}} \times \Delta^{\mathcal{O}}$, such that $(o_1, o_2) \in R^{\mathcal{O}}$, but $(o_1, o_2) \notin S^{\mathcal{O}}$. Let them be two new individual names i_1 and i_2. We show above that the translation function holds for individual names. Then, $[\pi_{i_1,i_2}(\neg S) \vee \pi_{i_1,i_2}(S)]^{\mathcal{I}} = \textsf{false}$. By the definition of the satisfaction of $\forall$, a formula is interpreted as $\textsf{true}$ only if every object in the domain is also interpreted as $\textsf{true}$. Therefore, $[\forall x \forall y.\pi_{x,y}(\neg R) \vee \pi_{x,y}(S)]^{\mathcal{I}} = \textsf{false}$.

The proof for concept subsumption Here, we prove that, given an axiom $D \sqsubseteq E \in \mathcal{K}$,

$$[\forall x.\pi_x(\neg D \sqcup E)]^{\mathcal{I}} = \begin{cases} \mathsf{true}, & \text{if } D^{\mathcal{O}} \subseteq E^{\mathcal{O}} \\ \mathsf{false}, & \text{otherwise} \end{cases}$$

by contrapositive. Assume that there exists an object o, such that $o \in D^{\mathcal{O}}$, but $o \notin E^{\mathcal{O}}$. Let it be a new individual name i. We prove above that the translation function for concept expression holds for individual names. Then, $[\pi_i(\neg D \sqcup E)]^{\mathcal{I}} = \mathsf{false}$. By the definition of the satisfaction of $\forall$, the formula is interpreted as true only if, for every object in the domain, it is also interpreted as true. Therefore, $[\forall x.\pi_x(\neg D \sqcup E)]^{\mathcal{I}} = \mathsf{false}$.

□

Corollary 2. *Given a knowledge base $\mathcal{K}$ and an interpretation $\mathcal{I}$, if $\mathcal{I} \Vdash \pi(\mathcal{K})$, then there exists a bi-ordered interpretation $\mathcal{O}$ such that $\mathcal{O} \Vdash \mathcal{K}$.*

Proof. As the previous corollary, this proof is quite long and contains three main parts. In the first part we build a bi-ordered interpretation based on the interpretation $\mathcal{I}$. The second part contains the proofs for concept and role assertions. Lastly, the proof is related to TBox axioms as such concept and role subsumption.

We prove this corollary by creating a bi-ordered interpretation such that it satisfies the knowledge base $\mathcal{K}$. So, given the interpretation $\mathcal{I}$, we build a bi-ordered interpretation, where:

- $\Delta^{\mathcal{O}} = \Delta^{\mathcal{I}}$;
- for each individual name a, $a^{\mathcal{O}} = a^{\mathcal{I}}$;
- for each concept name A and object $o \in \Delta^{\mathcal{I}}$, $o \in A^{\mathcal{O}}$ if $A^{\mathcal{I}}(o) = \mathsf{true}$;
- for each role name r and objecs $o_1, o_2 \in \Delta^{\mathcal{I}}$, $(o_1, o_2) \in r^{\mathcal{O}}$ if $r^{\mathcal{I}}(o_1, o_2) = \mathsf{true}$;
- for each pair of objects $o_1, o_2 \in \Delta^{\mathcal{I}}$, $(o_1, o_2) \in <^{\mathcal{O}}$ if $<^{\mathcal{I}} (o_1, o_2) = \mathsf{true}$; and
- for each pair of pairs of objects $o_1, o_2, o_3, o_4 \in \Delta^{\mathcal{I}}$, $((o_1, o_2), (o_3, o_4)) \in \ll^{\mathcal{O}}$ if $\ll^{\mathcal{I}} (o_1, o_2, o_3, o_4) = \mathsf{true}$.

Proof for preferential relations First, $<^{\mathcal{O}}$ and $\ll^{\mathcal{O}}$ must be strict partial orders, e.g., transitive, irreflexive and asymmetric. We prove for $<^{\mathcal{O}}$ by contrapositive, but the proof for $\ll^{\mathcal{O}}$ is analogous and left to the reader. We assume that $<^{\mathcal{O}}$ is not a strict partial order. Then, the following cases may happen:

1. there exists a pair $(o, o) \in <^{\mathcal{O}}$;
2. there exists $(o_1, o_2) \in <^{\mathcal{O}}$ and $(o_2, o_3) \in <^{\mathcal{O}}$, but $(o_1, o_3) \notin <^{\mathcal{O}}$; or
3. there exists $(o_1, o_2) \in <^{\mathcal{O}}$ and $(o_2, o_1) \in <^{\mathcal{O}}$.

If case 1 is true, then $\mathcal{I} \Vdash \exists x(x < x)$, by the Definition of $\mathcal{O}$. Hence, $\mathcal{I} \not\Vdash \forall x \neg(x < x)$. If case 2 is true, then $\mathcal{I} \Vdash \exists x \exists y \exists z\,[x < y \wedge y < z \wedge \neg(x < z)]$, by the definition of $\mathcal{O}$. Hence, $\mathcal{I} \not\Vdash \forall x \forall y \forall z\,[\neg(x < y \wedge y < z) \vee x < z]$. If case 3 is true, then $\mathcal{I} \Vdash \exists x \exists y\,[(x < y) \wedge (y < x)]$, by the definition of $\mathcal{O}$. Hence, $\mathcal{I} \not\Vdash \forall x \forall y\,[\neg(x < y) \vee \neg(y < x)]$. Therefore, $<^{\mathcal{O}}$ is a strict partial order.

Proof for role assertions We prove that $\mathcal{O}$ satisfies the role assertions in $\mathcal{K}$ by showing that if $[\pi_{a,b}(R)]^{\mathcal{I}} = \mathsf{true}$, then $(a^{\mathcal{O}}, b^{\mathcal{O}}) \in R^{\mathcal{O}}$ by induction over role expressions. There are three basis cases:

- Case R is a role name r or its negation. Then, by definition of $\mathcal{I}$, if $[r(a,b)]^{\mathcal{I}} = \mathsf{true}$, then $(a^{\mathcal{O}}, b^{\mathcal{O}}) \in r^{\mathcal{O}}$, and if $[\neg r(a,b)]^{\mathcal{I}} = \mathsf{true}$, then $[r(a,b)]^{\mathcal{I}} = \mathsf{false}$, therefore, $(a^{\mathcal{O}}, b^{\mathcal{O}}) \notin r^{\mathcal{O}}$;
- Case $R = \bullet r$. Then, $\pi_{a,b}(\bullet r)$ is $[r(a,b) \wedge \forall x \forall y\, [\neg((x,y) \ll (a,b)) \vee \neg r(x,y)]]^{\mathcal{I}} = \mathsf{true}$. Hence, $[r(a,b)]^{\mathcal{I}} = \mathsf{true}$, and for all pairs of objects in $\Delta^{\mathcal{I}}$, either $[(x,y) \ll (a,b)]^{\mathcal{I}} = \mathsf{true}$ or $[\neg r(x,y)]^{\mathcal{I}} = \mathsf{true}$. So, by the definition of $\mathcal{O}$, $(a^{\mathcal{O}}, b^{\mathcal{O}}) \in r^{\mathcal{O}}$ and, either $((o_1,o_2),(a^{\mathcal{O}}, b^{\mathcal{O}})) \notin (\ll^{\mathcal{O}})^+$ or $(o_1,o_2) \notin r^{\mathcal{O}}$, for all $o_1, o_2 \in \Delta^{\mathcal{O}}$. Therefore, $(a^{\mathcal{O}}, b^{\mathcal{O}}) \in (\bullet r)^{\mathcal{O}}$.

The induction step is for $R = \neg\bullet r$. Then, $\pi_{a,b}(\neg\bullet r)$ is $[\neg r(a,b) \vee \exists x \exists y\, [(x,y) \ll (a,b) \wedge \pi_{x,y}(\bullet r)]]^{\mathcal{I}} = \mathsf{true}$. Hence, by the semantics of intepretation, either $[\neg r(a,b)]^{\mathcal{I}} = \mathsf{true}$ or, for some pair of objects in $\Delta^{\mathcal{I}}$, $[((x,y) \ll (a,b))]^{\mathcal{I}} = \mathsf{true}$ and $[\pi_{x,y}(\bullet r)]^{\mathcal{I}} = \mathsf{true}$. So, by the definition of $\mathcal{O}$ and the IH, either $(a^{\mathcal{O}}, b^{\mathcal{O}}) \notin r^{\mathcal{O}}$, or $((o_1,o_2),(a^{\mathcal{O}}, b^{\mathcal{O}})) \in (\ll^{\mathcal{O}})^+$ and $(o_1,o_2) \in (\bullet r)^{\mathcal{O}}$, for some $o_1, o_2 \in \Delta^{\mathcal{O}}$. Therefore, $(a^{\mathcal{O}}, b^{\mathcal{O}}) \in (\neg\bullet r)^{\mathcal{O}}$.

Proof for concept assertions Here, we prove that if $(\pi_a D)^{\mathcal{I}} = \mathsf{true}$, then $a^{\mathcal{O}} \in D^{\mathcal{O}}$ by induction over concept expressions in $\mathcal{ALCH}^{\bullet}$.

The induction basis case is the one where D is a concept name. In this case, $\pi_a(A) = A(a)$, meaning that $A^{\mathcal{I}}(a^{\mathcal{I}}) = \mathsf{true}$. By the definition of $\mathcal{O}$, $a^{\mathcal{O}} \in A^{\mathcal{O}}$.

Induction steps:

- Case $D = \neg E$. Then, $(\pi_a(\neg E))^{\mathcal{I}} = \mathsf{true}$. So, $[\pi_a(E)]^{\mathcal{I}} = \mathsf{false}$, and by the IH, $a^{\mathcal{O}} \notin E^{\mathcal{O}}$. Therefore, $a^{\mathcal{O}} \in (\neg E)^{\mathcal{O}}$;
- Case $D = E_1 \sqcup E_2$. Then, either $[\pi_a(E_1)]^{\mathcal{I}} = \mathsf{true}$, or $[\pi_a(E_2)]^{\mathcal{I}} = \mathsf{true}$, or both. By the IH, $a^{\mathcal{O}} \in E_1^{\mathcal{O}} \cup E_2^{\mathcal{O}}$. Therefore, $a^{\mathcal{O}} \in (E_1 \sqcup E_2)^{\mathcal{O}}$;
- Case $D = E_1 \sqcap E_2$. Then, $[\pi_a(E_1)]^{\mathcal{I}} = \mathsf{true}$ and $[\pi_a(E_2)]^{\mathcal{I}} = \mathsf{true}$. By the IH, $a^{\mathcal{O}} \in E_1^{\mathcal{O}} \cap E_2^{\mathcal{O}}$. Therefore, $a^{\mathcal{O}} \in (E_1 \sqcap E_2)^{\mathcal{O}}$;
- Case $D = \exists R.E$. Then, $[\exists x\, [\pi_{a,x}(R) \wedge \pi_a(E)]]^{\mathcal{I}} = \mathsf{true}$. By the IH and the semantics, there exists an object $o \in \Delta^{\mathcal{O}}$ such that $(a^{\mathcal{O}}, o) \in R^{\mathcal{O}}$ and $o \in E^{\mathcal{O}}$. Therefore, $a^{\mathcal{O}} \in (\exists R.E)^{\mathcal{O}}$;
- Case $D = \forall R.E$. Then, $[\forall x\, [\pi_{a,x}(\neg R) \vee \pi_a(E)]]^{\mathcal{I}} = \mathsf{true}$. By the IH and the semantics, for all objects $o \in \Delta^{\mathcal{O}}$, either $(a^{\mathcal{O}}, o) \notin R^{\mathcal{O}}$, or $o \in E^{\mathcal{O}}$. Therefore, $a^{\mathcal{O}} \in (\forall R.E)^{\mathcal{O}}$;
- Case $D = \bullet E$. Then $[\pi_a(E) \wedge \forall x\, [\neg(x < a) \vee \pi_a(\neg E)]]^{\mathcal{I}} = \mathsf{true}$. By the IH and the semantics, for all objects $o \in \Delta^{\mathcal{O}}$, either $(o, a^{\mathcal{O}}) \notin (<)^{\mathcal{O}}$, or $o \in (\neg E)^{\mathcal{O}}$. Therefore, $a^{\mathcal{O}} \in (\bullet E)^{\mathcal{O}}$; and
- Case $D = \neg\bullet E$. Then $[\pi_a(\neg E) \vee \forall x\, [(x < a) \wedge \pi_a(\bullet E)]]^{\mathcal{I}} = \mathsf{true}$. By the IH and the semantics, for some object $o \in \Delta^{\mathcal{O}}$, $(o, a^{\mathcal{O}}) \in (<)^{\mathcal{O}}$, and $o \in (\bullet E)^{\mathcal{O}}$. Therefore, $a^{\mathcal{O}} \in (\neg\bullet E)^{\mathcal{O}}$.

Proof for role subsumptions We prove $\mathcal{O}$ satisfies role subsumptions by contrapositive. Assume that $[\forall x \forall y.\pi_{x,y}(\neg R) \vee \pi_{x,y}(S)]^{\mathcal{I}} = \mathsf{false}$. Then, there exists a pair of objects o_1, o_2 such that the formula is interpreted as false. Let them be two new individual names i_1, and i_2 such that $i_1^{\mathcal{I}} = o_1$ and $i_2^{\mathcal{I}} = o_2$. Hence, $[\pi_{i_1,i_2}(\neg R) \vee \pi_{i_1,i_2}(S)]^{\mathcal{I}} = \mathsf{false}$, and, by the proof for role assertions and satisfaction of $\neg$, $(o_1, o_2) \in R^{\mathcal{O}}$, but $(o_1, o_2) \notin S^{\mathcal{O}}$. Therefore, $R^{\mathcal{O}} \not\subseteq S^{\mathcal{O}}$.

Proof for concept subsumptions We prove $\mathcal{O}$ satisfies concept subsumptions by contrapositive. Assume that $[\forall x.\pi_x(\neg D \sqcup E)]^{\mathcal{I}} = \mathsf{false}$. Then, there exists an object o such that the formula is interpreted as false. Let it be a new individual name i, such that $i^{\mathcal{I}} = o$. Hence, $[\pi_i(\neg D \sqcup E)]^{\mathcal{I}} = \mathsf{false}$, and, by the proof for concept assertions and satisfaction of $\neg$, $o \in D^{\mathcal{O}}$, but $o \notin E^{\mathcal{O}}$. Therefore, $D^{\mathcal{O}} \not\subseteq E^{\mathcal{O}}$.

□

5 From formulae to matrices

In the previous sections, we show the translation from any knowledge base to formulae. Those formulae are fragments of the so-called first-order logic (FOL) used in various applications and automated theorem-proving algorithms. The connection method is one prominent example of such algorithms. The intuition for the connection method is its representation of formulae as matrices, extending the proof search connecting clauses of the matrix in order to check whether some formula is valid.

Given two formulae F and α, we say that F entails α, or $F \models \alpha$ if every model of F satisfies α. Furthermore, $F \models \alpha$ iff $\models \neg F \vee \alpha$, by the Deduction theorem. Thank that, when proving a knowledge base $\mathcal{K}$ entails an axiom, the entire knowledge base is negated, and the axiom remains positive.

Lemma 1. *Given a knowledge base $\mathcal{K}$ and an axiom α, $\mathcal{K} \models \alpha$ iff $\models \neg\pi(\mathcal{K}) \vee \pi(\alpha)$.*

Proof. We prove this lemma by contrapositive on both sides.

Only-if part: Assume that $\not\models \neg\pi(\mathcal{K}) \vee \pi(\alpha)$. Then, there exists an interpretation $\mathcal{I}$ such that $[\neg\pi(\mathcal{K}) \vee \pi(\alpha)]^{\mathcal{I}} = \mathsf{false}$, meaning that $[\pi(\mathcal{K}) \wedge \neg\pi(\alpha)]^{\mathcal{I}} = \mathsf{true}$. It is already proven, by the Corollary 2, that we can build a bi-ordered interpretation $\mathcal{O}$ such that $\mathcal{O} \Vdash \mathcal{K}$ but $\mathcal{O} \nVdash \alpha$. Therefore, $\mathcal{K} \not\models \alpha$.

If part: Assume that there exists a bi-ordered interpretation $\mathcal{O}$ such that $\mathcal{O} \Vdash \mathcal{K}$ but $\mathcal{O} \nVdash \alpha$. It is already proven, by the Corollary 1, that we can build an interpretation $\mathcal{I}$, such that $[\neg\pi(\mathcal{K}) \vee \pi(\alpha)]^{\mathcal{I}} = \mathsf{false}$. Therefore, $\not\models \neg\pi(\mathcal{K}) \vee \pi(\alpha)$

□

As a fragment of FOL, some useful results such as the so-called negation normal form (NNF), Prenex normal form (PNF), Skolem normal form (SNF) and their equivalence hold here as well. However, the Skolemization preserves the consistency of formulae but not the validity of them. Hence, as the connection method is a validity procedure, we use the dual version of Skolemization, the Herbrandization. In this process, the formulae contain only $\exists$, replacing the $\forall$ for functions.

Definition 11 (Prenex Normal Form (PNF)). *A formula F is in Prenex normal form if $F = Q_1x_1Q_2x_2 \cdots Q_nx_n\ \varphi$ where, for all $1 \leq i \leq n$, $Q_i \in \{\forall, \exists\}$ and φ is a quantifier-free formula.*

The FOL fragment we use here contains variables only guarded by quantifiers, e.g., no quantifier-free variables exist. As such, dealing with free variables is not the case, and the translation from any formula to its PNF is straightforward.

Definition 12 (Herbrand Normal Form (HNF)). *A formula is in Herbrand Normal Form*[7] *if it is in PNF, and for all $1 \leq i \leq n$, $Q_i = \exists$.*

In other words, there are no universal quantifiers on HNF. It is well-known that every formula in FOL can be converted to a formula in HNF due to a process called Herbrandization. It ensures that both formulae are equivalent: one is valid iff the other is valid as well. Hereafter we shall assume that formulae are quantifier-free, where the remaining variables are existentially quantified, and Skolem functions represent previous universally quantified variables.

Definition 13 (Disjunctive Normal Form (DNF)). *A formula is in Disjunctive normal form if it is a disjunction of conjunctions of literals.*

Corollary 3. *Given a formula F and an interpretation $\mathcal{I}$, there exists a formula F' in DNF, such that $\mathcal{I} \Vdash F$ iff $\mathcal{I} \Vdash F'$.*

The proof of this corollary comes from the facts that first-order operators are distributive.

Definition 14 (Validity). *Given a formula F, we say it is valid, denoted $\models F$, if $\mathcal{I} \Vdash F$, for every interpretation $\mathcal{I}$.*

Definition 15 (Matrix). *Given a formula $F = \exists x_1 \cdots \exists x_o D_1 \vee \cdots \vee D_n$, where each $D_i = L_{i,1} \wedge \cdots \wedge L_{i,m}$ and $L_{i,j}$ is a literal, a matrix is the set of clauses $\{C_1, \cdots, C_n\}$ where each clause $C_i = \{L_{i,1}, \cdots, L_{i,m}\}$ is the set of literals of each D_i.*

In other words, a matrix is the clausal form of a DNF formula without its quantifiers. So, given an interpretation $\mathcal{I}$ and a matrix $M = \{C_1, \cdots, C_n\}$, $\mathcal{I} \Vdash M$ iff $\mathcal{I} \Vdash F$, where $F = \exists x_1 \cdots \exists x_o D_1 \vee \cdots \vee D_n$, and each $D_i = L_{i,1} \wedge \cdots \wedge L_{i,m}$.

[7] The HNF is a dual normal form of Skolem Normal Form, as Herbrandization is the dual process to remove quantifiers as Skolemization process as well.

5.1 A shortcut from $\mathcal{ALCH}^{\bullet}$ knowledge bases to matrices

In the previous sections, we define and prove that a mapping function exists from $\mathcal{ALCH}^{\bullet}$ to a fragment of first-order logic. Moreover, from the formulae in first-order logic, we show a matrix which is the clausal form of the formula in DNF (and after Herbrandization) in this section. In detail, to map a knowledge base to a matrix, the following steps must occur:

1. to map the knowledge base to a formula in first-order logic;
2. to apply the negation of the formula in order to prove validity;
3. to transform the formula into Prenex normal form;
4. to apply the Herbrandization;
5. to transform the formula into DNF; and
6. to transform the conjunctions of the formula in DNF to its clausal form (the matrix).

In this subsection, we provide a direct mapping from $\mathcal{ALCH}^{\bullet}$ to matrices to obtain a more straightforward translation for knowledge bases, applying those six steps above in one translation.

At this point, we have a translation from $\mathcal{ALCH}^{\bullet}$ to a formula (step 1). Now, we define a useful translation from formulae to matrices, encompassing steps 3, 4, 5, and 6. This translation is based on the matrix representation for formulae presented in Otten, 2012. [28]

Definition 16 (Clausal union). *Let M_1 and M_2 be matrices. We define the* <u>*clausal union*</u> *of M_1 and M_2 as $M_1 \uplus M_2 \overset{\text{def}}{=} \{C_1 \cup C_2 \mid (C_1, C_2) \in M_1 \times M_2\}$.*

Definition 17 (Application of a function symbol). *Given a function f, an individual name c, and a set of variables $S = \{x_1, \cdots, x_n\}$, we define the* <u>*application of*</u> *the function f with arguments S as:*

$$\mathsf{apply}(f, S) = \begin{cases} c, \textit{if } S = \emptyset \\ f(x_1, \cdots, x_n), \textit{otherwise} \end{cases}$$

Definition 18 (Matrix of a formula). *Given a formula F, a set of variables S, a new variable x', and a new function f', the* <u>*matrix*</u> *of the formula F, denoted $M(F)$ is defined as follows:*

- $M(L, S) \overset{\text{def}}{=} \{\{L\}\}$;
- $M(F_1 \vee F_2, S) \overset{\text{def}}{=} M(F_1, S) \cup M(F_2, S)$;
- $M(F_1 \wedge F_2, S) \overset{\text{def}}{=} M(F_1, S) \uplus M(F_2, S)$;
- $M(\exists x F_1, S) \overset{\text{def}}{=} M(F_1, S \cup \{x\})[x \setminus x']$;
- $M(\forall x F_1, S) \overset{\text{def}}{=} M(F_1, S)[x \setminus \mathsf{apply}(f', S)]$.

This translation from formulae to matrices already converts the whole formula to a formula in DNF and HNF equivalent one.

Corollary 4. *Given a formula F, $M(F, \emptyset)$ is the clausal form of F', such that F' is the formula F in DNF and HNF.*

Proof. We prove this corollary by induction over the structure of the formula F. The induction basis case is when the formula is a literal. Therefore, $M(F)$ is in DNF, HNF, and is equivalent to F.

Induction steps:

- When $F = F_1 \vee F_2$. Then, by the IH, $M(F_1, S_1)$ is the formula F_1 in DNF, HNF, and equivalent to F_1, and $M(F_2, S_2)$ the same for F_2. Therefore, the union of two DNF clausal form formulae is already a DNF clausal form for $F_1 \vee F_2$;
- When $F = F_1 \wedge F_2$. Then, by the IH, $M(F_1, S)$ is the formula F_1 in DNF, HNF, and equivalent to F_1, and $M(F_2, S)$ the same for F_2. Let $F_1' = D_1 \vee \cdots \vee D_n$ and $F_2' = E_1 \vee \cdots \vee E_m$ be the formulae of $M(F_1, S_1)$ and $M(F_2, S_2)$, respectively. To convert the conjunction of two formulae in DNF we apply the distributivity property for $\vee$ and $\wedge$ as follows:

$$\begin{aligned} F_1' \wedge F_2' = &(D_1 \wedge E_1) \vee \cdots \vee (D_1 \wedge E_m) \\ &\vee (D_2 \wedge E_1) \vee \cdots \vee (D_2 \wedge E_m) \\ &\vdots \\ &\vee (D_n \wedge E_1) \vee \cdots \vee (D_n \wedge E_m) \end{aligned}$$

  which coincides with the cartesian product of their clauses. Therefore, $M(F_1, S) \uplus M(F_2, S)$ is in DNF, HNF and is equivalent to $F_1 \wedge F_2$;
- When $F = \exists x F_1$. Then, by the IH, $M(F_1, S)$ is the clausal form of the formula F_1. As the matrix is the clausal form of a formula in HNF, it is easy to see that $\exists x F'(x)$ is equivalent to $F'(x)$, as the variables are guarded by existential quantifiers, by the definition of HNF;
- When $F = \forall x F_1$. Then, by the IH, $M(F_1, S)$ is the clausal form of the formula F_1. As the matrix is the clausal form of a formula in HNF, $\forall x F'(x)$ is equivalent to $F'(\mathsf{apply}(\mathsf{f}, \mathsf{S}))$, where S is the set of variables before $\forall$, as the Herbrandization process transform universally-restricted variables to functions such that their validity hold.

Definition 19 (Translation function for role expressions). *Given a role R, two terms t, u, two variables x, y, and two functions f_i, g_i, the translation function $\delta(R, t, u)$ maps role expressions to matrices as follows:*

- $\delta(r, t, u) \stackrel{\text{def}}{=} \{\{r(t, u)\}\}$;
- $\delta(\neg r, t, u) \stackrel{\text{def}}{=} \{\{\neg r(t, u)\}\}$;
- $\delta(\neg \bullet r, t, u) \stackrel{\text{def}}{=} \{\{\neg r(t, u)\}\} \cup [\{\{(x, y) \ll (t, u)\}\} \uplus \{\{r(x, y)\}\}]$;
- $\delta(\bullet_i r, t, u) \stackrel{\text{def}}{=} \{\{r(t, u)\}\} \uplus [\{\{\neg((f_i(t), g_i(u)) \ll (t, u))\}\} \cup \delta(\neg \bullet r, f_i(t), g_i(u))]$;
- $\delta(\neg\neg R, t, u) \stackrel{\text{def}}{=} \delta(R, t, u)$.

Definition 20 (Translation function for concept expressions). *Given a concept D, a term t, and a ordered set of terms S, the translation function $\delta(D, t, S)$ maps concept expressions to matrices as follows:*

- $\delta(A, t, S) \stackrel{\text{def}}{=} \{\{A(t)\}\}$;
- $\delta(\neg A, t, S) \stackrel{\text{def}}{=} \{\{\neg A(t)\}\}$;
- $\delta(D \sqcap E, t, S) \stackrel{\text{def}}{=} \delta(D, t, S) \uplus \delta(E)$;
- $\delta(\neg(D \sqcap E), t, S) \stackrel{\text{def}}{=} \delta(\neg D, t, S) \cup \delta(\neg E)$;

- $\delta(D \sqcup E, t, S) \stackrel{\text{def}}{=} \delta(D, t, S) \cup \delta(E, t, S)$;
- $\delta(\neg(D \sqcup E), t, S) \stackrel{\text{def}}{=} \delta(\neg D, t, S) \uplus \delta(\neg E, t, S)$;
- $\delta(\exists_i R.D, t, S) \stackrel{\text{def}}{=} \delta(R, t, x, S \cup \{x\}) \uplus \delta(E, x, S \cup \{x\})$;
- $\delta(\neg(\exists_i R.D), t, S) \stackrel{\text{def}}{=} \delta(\neg R, f_i(\mathsf{var}(S)), S) \cup \delta(\neg E, f_i(\mathsf{var}(S)), S)$;
- $\delta(\forall_i R.D, t, S) \stackrel{\text{def}}{=} \delta(\neg R, f_i(\mathsf{var}(S)), S) \cup \delta(E, f_i(\mathsf{var}(S)), S)$;
- $\delta(\neg(\forall_i R.D, t, S)) \stackrel{\text{def}}{=} \delta(R, t, x, S \cup \{x\}) \uplus \delta(\neg D, x, S \cup \{x\})$;
- $\delta(\bullet_i D, t, S) \stackrel{\text{def}}{=} \delta(D, t, S) \uplus [\{\{\neg(f_i(\mathsf{var}(S)) < t)\}\} \cup \delta(\neg \bullet D, f_i(\mathsf{var}(S)), S)]$;
- $\delta(\neg \bullet D, t, S) \stackrel{\text{def}}{=} \delta(\neg D, t, S) \cup [\{\{x < t\}\} \uplus \delta(D, x, S \cup \{x\})]$;
- $\delta(\neg\neg D) \stackrel{\text{def}}{=} \delta(D)$.

Definition 21 (Translation function for knowledge bases). *Given a knowledge base $\mathcal{K}$ we define the translation function δ from $\mathcal{K}$ to a matrix as*

$$\delta(\mathcal{K}) \stackrel{\text{def}}{=} \bigcup_{D \sqsubseteq E \in \mathcal{K}} \delta(D \sqcap \neg E, x, \{x\}) \cup \bigcup_{a:D \in \mathcal{K}} \delta(\neg D, a, \{a\}) \tag{9}$$

$$\cup \bigcup_{R \sqsubseteq S \in \mathcal{K}} [\delta(R, x, y) \uplus \delta(\neg S, x, y)] \cup \bigcup_{(a,b):R} \delta(\neg R, a, b) \tag{10}$$

$$\cup \{\{x < y, y < z, \neg(x < z)\}\} \tag{11}$$

$$\cup \{\{x < x\}\} \tag{12}$$

$$\cup \{\{x < y, y < x\}\} \tag{13}$$

$$\cup \{\{(x, y) \ll (z, k), (z, k) \ll (m, n), \neg((x, y) \ll (m, n))\}\} \tag{14}$$

$$\cup \{\{(x, y) \ll (x, y)\}\} \tag{15}$$

$$\cup \{\{(x, y) \ll (z, k), (z, k) \ll (x, y)\}\} \tag{16}$$

Definition 22 (Matrices equivalence). *Given two formulas F_1 and F_2, and their matrices M_1 and M_2, respectively. We say M_1 and M_2 are equivalent if, for every interpretation $\mathcal{I}$, $\mathcal{I} \Vdash F_1$ iff $\mathcal{I} \Vdash F_2$.*

Corollary 5. *Given a knowledge base $\mathcal{K}$, $M(\neg\pi(\mathcal{K}))$ and $\delta(\mathcal{K})$ are equivalent.*

Proof. The proof of this corollary comes from the Definitions 10, 18, 21, and the semantics of FOL.

6 Connection method

When looking at the matrix, if we change our perspective, we can see the paths of the matrix.

Definition 23 (Path). *Given a matrix M, a path is a set containing exactly one literal from each clause of M.*

If the matrix is the clausal form for a DNF formula, then its paths compose the clausal form for the same formula in CNF. Therefore, an interpretation satisfies a path iff the interpretation satisfies every literal on it. Furthermore, an interpretation satisfies a set of paths iff the interpretation satisfies at least one path.

Lemma 2. *Given a matrix M and an interpretation $\mathcal{I}$, $\mathcal{I}$ satisfies M iff $\mathcal{I}$ satisfies every path of it.*

Proof. We prove this lemma by contrapositive on both sides.

Only-if part: Assume that $\mathcal{I} \nVdash p$, for some path p of M. So, $\mathcal{I} \nVdash L$, for every $L \in p$. By the definition of a path, the path must contain a literal from each clause in the matrix, meaning that $\mathcal{I} \nVdash C$, for every clause $C \in M$. Therefore, $\mathcal{I} \nVdash M$.

If-part: . Assume that $\mathcal{I} \nVdash M$. Then, at least one literal from each clause is not satisfied by $\mathcal{I}$. By the semantics and the definition of a path, there exists a path containing each literal not satisfied by $\mathcal{I}$ and, therefore, there exists a path not satisfied by $\mathcal{I}$. □

Definition 24 (Multiplicity). *The multiplicity is a function μ that maps for each clause of a matrix a natural number denoting the number of copies of such clause.*

We represent a matrix M and its copies as M^{μ}.

Corollary 6. *Given a matrix M, a multiplicity μ, and an interpretation $\mathcal{I}$, $\mathcal{I} \Vdash M$ iff $\mathcal{I} \Vdash M^{\mu}$.*

The proof idea is to show that copies preserve the semantics of a matrix. Due to the Skolemization process, there are only existentially quantified variables. Since a matrix is a formula in DNF, given an interpretation $\mathcal{I}$:

$$\mathcal{I} \Vdash \exists x_1, \cdots, x_n \varphi \text{ iff } \exists x_1, \cdots, x_n\ \varphi \vee \exists y_1, \cdots, y_n\ \varphi[x_1 \mapsto y_1, \cdots, x_n \mapsto y_n]$$

Definition 25 (Term substitution). *A term substitution is a function $\sigma : \mathsf{V} \longrightarrow \mathsf{T}$ that maps variables to terms of a matrix. We define that $\sigma(L)$ is the literal L, but its variables x are replaced by $\sigma(x)$, i.e., $\sigma(A(x)) = A(\sigma(x))$.*

Besides this definition, we shall ensure no substitution between a variable and a term that mentions it. A substitution with this behaviour is called idempotent. However, we will omit this term hereafter to a better understanding.

Definition 26 (σ-complementary literals). *Given a literal L, we say that $\overline{L}$ is the σ-complement of L if $\sigma(\overline{L}) = \sigma(\neg P)$, when $L = P$ or $\sigma(\overline{L}) = \sigma(P)$ when $L = \neg P$.*

Hereafter, in order to decrease the complexity of definitions, we shall omit σ before complementary literals or sets.

Definition 27 (Complementary set). *A complementary set is a pair $\{L_1, L_2\}$, where $\sigma(L_1) = \sigma(\overline{L_2})$.*

Definition 28 (Complementary matrix). *A matrix is called complementary if there exists a term substitution σ such that it is a complementary set for each path on it.*

Theorem 1 (Matrix characterisation). *Given a matrix M, there exists a multiplicity μ and a term substitution σ such that $\sigma(M^\mu)$ is complementary iff $\models M$.*

Proof. We prove this theorem by contrapositive.

$\Rightarrow$. Assume that $\not\models M$. Then, there exists an interpretation $\mathcal{I}$ such that $\mathcal{I} \not\Vdash M$. Hence, there exists a path in M that is not satisfied by $\mathcal{I}$ since the path corresponds to a disjunction in CNF. That path cannot contain a complementary set on it, no matter which substitution is applied to it. Otherwise, it is a validity, and $\mathcal{I}$ must satisfy it. Therefore, M is not complementary under any term substitution.

$\Leftarrow$. Assume that $\sigma(M^\mu)$ is not complementary, for all multiplicity μ and for all term substitution σ. Then, there exists a path which does not contain complementary sets on it. Hence, we can build an interpretation such that it does not satisfy every literal of the path. Let the path be p, we build an interpretation $\mathcal{I}$ such that:

- $\mathcal{I} \Vdash \neg A(t)$, for all unary predicate A and term t s.t. $A(t) \in p$;
- $\mathcal{I} \Vdash A(t)$, for all unary predicate A and term t s.t. $\neg A(t) \in p$;
- $\mathcal{I} \Vdash \neg R(t,u)$, for all binary predicate R and terms t,u s.t. $R(t,u) \in p$;
- $\mathcal{I} \Vdash R(t,u)$, for all binary predicate R and terms t,u s.t. $\neg R(t,u) \in p$;
- $\mathcal{I} \Vdash \neg(t < u)$, for all terms t,u s.t. $(t < u) \in p$;
- $\mathcal{I} \Vdash (t < u)$, for all terms t,u s.t. $\neg(t < u) \in p$;
- $\mathcal{I} \Vdash \neg((t,u) \ll (v,k))$, for all terms t,u,v,k s.t. $((t,u) \ll (v,k)) \in p$; and
- $\mathcal{I} \Vdash ((t,u) \ll (v,k))$, for all terms t,u,v,k s.t. $\neg((t,u) \ll (v,k)) \in p$.

That interpretation does not satisfy the formula in CNF. Otherwise, it should satisfy the path. Therefore, $\not\models M$. □

7 Connection calculus

We define the connection method with a formal calculus. The main structure of the calculus is a triple $\langle C, M, \mathit{Path} \rangle$, where C is the goal, M is the matrix, and *Path* is the active path of the proof. The proof starts with no goal and path (represented by ε), applying the Start rule for some clause in M. If the proof finds an empty set as the goal, then that branch is closed with the Axiom.

As $\mathcal{ALCH}^\bullet$ is a decidable-fragment of preferential DLs, a notion of blocking is needed before presenting the calculus.

Definition 29 (Set of concepts). *Given a term t, a path p, and a term substitution σ, the set of concepts of t w.r.t. the path p, denoted by $\tau_p^\sigma(t)$ is*

$$\tau_p^\sigma(t) = \{D \mid \textit{for all } D(\sigma(t)) \in \sigma(p)\}$$

Definition 30 (Copy). *Given two clauses C_1 and C_2, we say that C_2 is a copy of C_1 if they have the same literals but different variables.*

Definition 31 (i-th copy). *Given a matrix M, a clause $C \in M$, and a multiplicity μ, with C^i we denote the i-th copy of $C \in M^\mu$, where $1 \leq i \leq \mu(C)$.*

Definition 32 (Blocking). *Given a literal L and a path p, we say that $p \cup \{L\}$ is blocked w.r.t. a term substitution σ if:*

1. *$\sigma(L) \in \sigma(p)$; or*
2. *if $L \in C^n$, for some clause $C \in M^\mu$ and some $2 \leq n \leq \mu(C)$, there exists a previous copy term t_{n-1} such that either t_{n-1} is blocked or $\tau^\sigma_{p\cup\{L\}}(t_n) \subseteq \tau^\sigma_{p\cup\{L\}}(t_{n-1})$.*

Definition 33 (Blocked cause). *We say that a clause is blocked w.r.t. a path p and a term substitution σ if some literal $L \in C$ is blocked w.r.t. p.*

During the proof, the calculus connects literals on the goal in two different ways. The first one, Reduction rule, occurs when a complementary literal is already on the active path, meaning that path contains a complementary set. The second one, Extension rule, occurs when it finds another clause with a complementary literal for the goal. In this case, the proof is branched into two parts in order to check the remaining goal (without that literal complemented) and the new goal (the clause found). Every application of a rule in calculus must ensure the goal is not blocked w.r.t. to its path. Otherwise, no rule application must occur.

$$\textbf{\textit{Axiom (Ax)}} \ \frac{}{\{\}, M, \mathit{Path}}$$

$$\textbf{\textit{Start rule (St)}} \ \frac{C_1, M, \{\}}{\varepsilon, M, \varepsilon}, \text{ with } C_1 \text{ being a blocking-free copy of some } C \in M$$

$$\textbf{\textit{Reduction rule (Red)}} \ \frac{C, M, \mathit{Path} \cup \{L_2\}}{C \cup \{L_1\}, M, \mathit{Path} \cup \{L_2\}}, \\ \text{with } \sigma(L_1) = \sigma(\overline{L_2})$$

$$\textbf{\textit{Extension rule (Ext)}} \ \frac{C_1 \setminus \{L_2\}, M, \mathit{Path} \cup \{L_1\} \qquad C, M, \mathit{Path}}{C \cup \{L_1\}, M, \mathit{Path}}, \\ \text{with } L_2 \in C_1, C_1 \text{ being a blocking-free copy of some } C_2 \in M, \text{ and } \sigma(L_1) = \sigma(\overline{L_2})$$

Fig. 1. The calculus.

Definition 34 (Connection proof). *Given a triple $\langle C, M, \mathit{Path}\rangle$, we say that it is a connection proof if, applying the rules of the calculus for $\langle C, M, \mathit{Path}\rangle$,* *there exists a multiplicity μ, a substitution σ, and a proof tree such that every leaf ends with an* Axiom.

Definition 35 (Relative paths). *Given two sets of literals C and S, and a matrix M, we define the* relative paths *of C, S and M as:*

$$\begin{aligned}
\varphi(M) &\stackrel{\text{def}}{=} \{p \mid p \textit{ is a path of } M\} \\
\varphi(M, S) &\stackrel{\text{def}}{=} \{p \in \varphi(M) \mid S \subseteq p\} \\
\varphi(M, S, C) &\stackrel{\text{def}}{=} \{p \in \varphi(M, S) \mid L \in p, \textit{for some } L \in C\}
\end{aligned}$$

Corollary 7.

$$\varphi(M, S \cup \{L\}) = \varphi(M, S, \{L\})$$

Proposition 1. *Given a triple $\langle C, M, \mathit{Path}\rangle$, if it is a connection proof, for some variable substitution σ, then there exists a multiplicity μ for all path $p \in \varphi(M, \mathit{Path}, C)$, s.t. p is σ-complementary.*

Proof. We prove the lemma above by structural induction over the connection proofs. Since a connection proof can be a subtree of another connection proof, we can assume, as the Induction Hypothesis (IH), that if there exists a connection proof of a subtree for some σ, then there exists a multiplicity μ such that every path in it is σ-complementary. We demonstrate that the lemma holds for Axiom and the rules Reduction and Extension in the calculus as follows:

- **Axiom (Ax) :** Assume that $\dfrac{}{\{\}, M, \mathit{Path}}$ is a connection proof for $\langle\{\}, M, \mathit{Path}\rangle$. Therefore, $\varphi(M, \mathit{Path}, \emptyset) = \emptyset$, meaning that IH holds since there is no complementary path in the empty set;
- **Reduction (Red) :** Assume that $\dfrac{\mathit{Proof}}{C, M, \mathit{Path} \cup \{L_2\}}$ is a connection proof for $\langle C, M, \mathit{Path} \cup \{L_2\}\rangle$, for some variable substitution σ. Then, the derivation $\dfrac{\dfrac{\mathit{Proof}}{C, M, \mathit{Path} \cup \{L_2\}}}{C \cup \{L_1\}, M, \mathit{Path} \cup \{L_2\}}$ is a connection proof, where $\sigma'(\sigma(L_1)) = \sigma'(\sigma(\overline{L_2}))$. By the IH, for some multiplicity μ, every path $p' \in \varphi(M^{\mu}, \mathit{Path} \cup \{L_2\}, C)$ is σ-complementary. Furthermore, every path $p'' \in \varphi(M^{\mu}, \mathit{Path} \cup \{L_2\}, \{L_1\})$ is σ'-complementary, since $\sigma'(\sigma(L_1)) = \sigma'(\sigma(\overline{L_2}))$. Let μ' be μ and σ'' be the composition of σ and σ'. As $\varphi(M^{\mu'}, \mathit{Path} \cup \{L_2\}, C \cup \{L_1\}) = \varphi(M^{\mu'}, \mathit{Path} \cup \{L_2\}, C) \cup \varphi(M^{\mu'}, \mathit{Path} \cup \{L_2\}, \{L_1\})$, we conclude that every path in $\varphi(M^{\mu'}, \mathit{Path} \cup \{L_2\}, C \cup \{L_1\})$ is σ''complementary;
- **Extension (Ex) :** Assume that $\dfrac{\mathit{Proof1}}{C_2 \setminus \{L_2\}, M, \mathit{Path} \cup \{L_1\}}$ is a connection proof for $\langle C_2 \setminus \{L_2\}, M, \mathit{Path} \cup \{L_1\}\rangle$, and $\dfrac{\mathit{Proof2}}{C, M, \mathit{Path}}$ is a connection proof for $\langle C, M, \mathit{Path}\rangle$, for some substitution σ. By the IH, there exists a multiplicity μ_1 such that every path in $\varphi(M^{\mu_1}, \mathit{Path} \cup \{L_1\}, C_2 \setminus \{L_2\})$ is σ-complementary, and there exists a multiplicity μ_2 such that every path in $\varphi(M^{\mu_2}, \mathit{Path}, C)$ is σ-complementary. Then, the derivation

$$\cfrac{\cfrac{\text{Proof1}}{C_2 \setminus \{L_2\}, M, Path \cup \{L_1\}} \qquad \cfrac{\text{Proof2}}{C, M, Path}}{C \cup \{L_1\}, M, Path}$$

with $\sigma'(\sigma(L_1)) = \sigma'(\sigma(\overline{L_2}))$ and C_2 as a copy of some clause $C_1 \in M$, is a connection proof for $\langle C \cup \{L_1\}, M, Path\rangle$. Let μ' be the combination of μ_1 and μ_2, and σ'' be the composition of σ and σ'. Hence, every path $p' \in \varphi(M^{\mu'}, Path, C)$ is σ''-complementary and every path $p'' \in \varphi(M^{\mu'}, Path, C_2 \setminus \{L_2\})$ is also σ''complementary. Moreover, once $\sigma''(L_1) = \sigma''(\overline{L_2})$, every path in $\varphi(M^{\mu'}, Path \cup \{L_1\}, \{L_2\})$ is complementary. As $C_2 \in M^{\mu'}$, we also have that $\varphi(M^{\mu'}, Path \cup \{L_1\}, C_2) = \varphi(M, Path \cup \{L_1\})$ and every path on it is complementary. Therefore, every path in $\varphi(M^{\mu'}, Path, C \cup \{L_1\})$ is σ''-complementary since $\varphi(M^{\mu'}, Path, C \cup \{L_1\}) = \varphi(M^{\mu'}, Path \cup \{L_1\}) \cup \varphi(M, Path, C)$.

□

Lemma 3 (Soundness of the calculus). *Given a matrix M, if $\langle \varepsilon, M, \varepsilon\rangle$ is a connection proof with σ, then there exists a multiplicity μ such that every path in $\varphi(M^\mu)$ is σ-complementary.*

Proof. We prove this theorem by contrapositive. If we assume there is no multiplicity, then no copies are able to occur, and the triple is not a connection proof. Now, we assume there exists a multiplicity μ and a path in $\varphi(M^\mu)$ that is not σ-complementary, for all σ. Let $\cfrac{\cfrac{\cdots}{\langle C_2, M, \{\}\rangle}}{\langle \varepsilon, M, \varepsilon\rangle}\,St$ be the derivation for M, with C_2 being a copy of some clause $C_1 \in M$. Then, $\varphi(M^\mu, \{\}, C_2)$, is not σ-complementary as well. By Proposition 1's contrapositive, there is no connection proof for $\langle C_2, M, \{\}\rangle$, therefore, there is no connection proof for $\langle \varepsilon, M, \varepsilon$. □

Lemma 4 (Completeness of the calculus). *Given a matrix M, if there exists a multiplicity μ and variable substitution σ such that every path in $\varphi(M^\mu)$ is σ-complementary, then $\langle \varepsilon, M, \varepsilon\rangle$ is a connection proof.*

Proof. So, we prove this theorem by contrapositive. Then, if there is no connection proof for $\langle \epsilon, M, \epsilon\rangle$, then there exists no multiplicity μ and variable substitution σ such that every path in $\varphi(M^\mu)$ is σ-complementary. Thus, w.l.o.g. there exists a satured derivation branch of $\langle \epsilon, M, \epsilon, \rangle$ containing a leaf $\cfrac{\langle C, M, Path\rangle}{\cdots}$ such that there is no rule of the calculus to be applied. The clause C cannot be empty. Otherwise, the Axiom would be applied. Also, the clause C does not contain a literal L such that its complement $\overline{L} \in Path$ with σ. Otherwise, the Reduction rule would be applied. Finally, there is no blocking-free copy clause C_2 of $C_1 \in M^\mu$ such that $\overline{L} \in C_2$, for some literal $L \in C$, otherwise the extension rule would be applied. To enforce this possibility, assume there exists a clause blocked by *Path*, and it contains a complementary literal. Then, once it is blocked, it means that: (i) it exists in the *Path*, so the path would remain the same, or (ii) there exists a previous copy term such that its set of concepts is the same or greater than the set of concepts of the target term in the new clause. Once this derivation is a saturated proof, this clause was used before in the path, and it remains open. It means there exists an infinite loop for this proof, and as such, it is blocked, remaining open to this derivation. Therefore, there exists a path $p \in \varphi(M^\mu, Path)$ over the matrix M such that p is not complementary for any multiplicity μ. ☐

Algorithm 1: Inconsistent

```
Inconsistent (𝒦)
    Input  : An 𝒜ℒ𝒞ℋ• knowledge base 𝒦
    Output: True if 𝒦 ⊨ ⊥ or False otherwise
    F ← δ(𝒦);
    M stores the DNF clausal form of formula F after Herbrandization ;
    for each clause C ∈ M do
        if Proof(C, M, {}) is True then
            return True;
        end
    end
    return False;
end
```

Algorithm 2: Proof

```
Proof (C, M, Path)
    Input  : A (sub-)clause C, a matrix M and a (sub-)path Path
    Output: True if there exists a connection proof for ⟨C, M, Path⟩ or
            False otherwise
    /* for Axiom                                                    */
    if C = ∅ then
        return True;
    end
    for each rule R that is applicable to ⟨C, M, Path⟩ do
        for each triple ⟨C', M', Path'⟩ derived from applying R do
            if Proof(C', M', Path') is False then
                skip rule R;
            end
        end
        return True;
    end
    return False;
end
```

Theorem 2 (Termination). *Given any $\mathcal{ALCH}^\bullet$ knowledge base $\mathcal{K}$, **Inconsistent**($\mathcal{K}$) terminates.*

Proof. The key component that affects the termination of the algorithm is when Extension rules are applied. The reduction rule reduces the number of literals to be proven, and Axiom closes the branch with an empty goal. Hence, they lead to a finite number of applications. The Extension rule reduces the number of literals to be proven on one side of its derivation but adds a new clause to be proven on its left side. So, to ensure the application of Extension rules stops, we need to prove that at some point, every clause with complementary literals of a goal is blocked.

Let m be the number of concept names occurring in $\mathcal{K}$. During an Ext application, there exists at least a term t in the new clause that is unified with a term u in the path. Once there exists a finite number of concept names, the term t can appear at most m times in the path before being blocked (Case 1 of Blocking definition). The same idea can be used for pair of terms since there exists at most n role names. If the Ext application adds new terms to the path, it must add different concept names to the path to prevent being blocked by its previous copy term. Again, there exists a finite number of concept names. Therefore, the blocking will occur at some point.

□

Theorem 3 (Soundness). *Given a knowledge base $\mathcal{K}$, if **Inconsistent**($\mathcal{K}$) returns True, then $\mathcal{K}$ is inconsistent.*

Proof. The proof is a direct consequence of Theorem 1 and Lemma 3. □

Theorem 4 (Completeness). *Given a knowledge base $\mathcal{K}$, if $\mathcal{K}$ is inconsistent, then **Inconsistent**($\mathcal{K}$) returns True*

Proof. The proof is a consequence of Theorem 1 and Lemma 4. □

8 Example of proof

We present in this section a toy example to illustrate how the proposal checks whether a knowledge base entails an assertion. For that, assume $\mathsf{C} = \{A, B\}$, $\mathsf{R} = \{r, s\}$, $\mathsf{I} = \{a, b\}$ as the set of concept names, set of role names, and set of individual names, respectively. Let $\mathcal{K} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ be a knowledge base, where $\mathcal{T} = \{\exists s.B \sqsubseteq \bullet A\}$, $\mathcal{R} = \{\bullet r \sqsubseteq s\}$, and $\mathcal{A} = \{\bullet r(a, b), B(b)\}$. We want to check whether $\mathcal{K} \models A(a)$, i.e., that $A(a)$ is entailed by $\mathcal{K}$. The first step is to translate $\mathcal{K}$ to the matrix $M = \delta(\mathcal{K} \cup \{\neg A(a))$ as

$$\delta(\exists s.B \sqcap \neg\bullet A(x_1, \{x_1\})) \cup \delta(\bullet r, x_2, y_1) \uplus \delta(\neg s, x_2, y_1)$$
$$\cup\, \delta(\neg\bullet r(a, \emptyset)) \cup \delta(\neg B(b, \emptyset)) \cup \delta(A, a, \emptyset) \cup \dagger$$

and its graphical representation is shown in Figure 2.

It may not be necessary to use all the clauses of the matrix besides the connection method making connections through all the paths. In our example, the second clause and the transitivity, irreflexivity, and asymmetry representation will not be used,[8] and we shall omit them hereafter.

[8] The axioms are denoted with the † in the matrix as we mention in Section **??**.

$$\begin{bmatrix} s(x,y) & s(x,y) & r(x,y) & r(x,y) & r(x,y) & \neg r(a,b) & (x',y') \ll (a,b) & \neg B(b) & A(a) & \dagger \\ B(y) & B(y) & \neg((f(x),g(y)) \ll (x,y)) & \neg r(f(x),g(y)) & (x',y') \ll (f(x),g(y)) & & r(x',y') & & & \\ \neg A(x) & x' < x & \neg s(x,y) & \neg s(x,y) & r(x',y') & & & & & \\ & A(x') & & & \neg s(x,y) & & & & & \end{bmatrix}$$

Fig. 2. Matrix representation of $\delta(\mathcal{K} \cup \{\neg A(a)\})$. The matrix displays clauses as columns in the same order as shown in $\mathcal{K}$. Clauses may have different lengths, thus causing gaps in the matrix.

$$\begin{bmatrix} A(a) & s(x_1,y_1) & \neg B(b) & \neg s(x_2,y_2) & \neg r(a,b) & (x_3,y_3) \ll (a,b) & \neg s(x_4,y_4) \\ & B(y_1) & & r(x_2,y_2) & & r(x_3,y_3) & r(x_4,y_4) \\ & \neg A(x_1) & & \neg r(f(x_2),g(y_2)) & & & \neg((f(x_4),g(y_4)) \ll (x_4,y_4)) \end{bmatrix}$$

$$\sigma = \{x_1/a, y_1/b, x_2/a, y_2/b, x_3/f(x_2), y_3/g(y_2), x_4/a, y_4/b\}$$

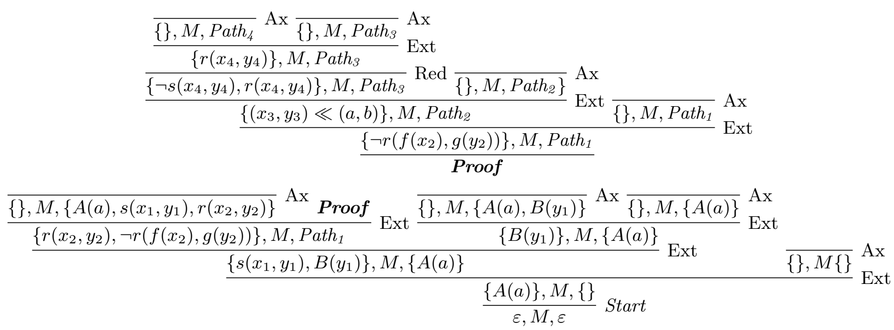
Fig. 3. Connection method for M. The clauses were rearranged to a simpler representation by omitting not used clauses as well.

In Figure 3, we perform the connection method in a graphical way. The red curves between literals of the matrix represent the application of the Start, the Extension, or the Reduction rules. The goal is to check if every clause associate is fully connected. Otherwise, the derivation is not a connection proof. It starts with $A(a)$, the possible entailed assertion, connecting it to a complementary literal $\neg A(a)$ in the second clause. As there is no $\neg A(a)$ in the matrix, we perform a substitution to x_1 to make σ-substitution and unify the literals. Therefore, $\sigma = \{x_1/a\}$ at this point. The other connections follow the same approach, and as every clause is fully connected, we prove that $\mathcal{K} \models A(a)$.

Another useful representation of the connection method is the sequent-style derivation tree. It shows the triples during the search proof and applies the rules until no rule can be applied. If every leaf is an Axiom, i.e., the subgoal is fully proven, then it is a connection proof of M. Figure 8 shows this representation.

9 Concluding remarks

In this work, we have defined a connection method for $\mathcal{ALCH}^{\bullet}$, a defeasible description logic encompassing most defeasible DLs considered in the literature. The calculus extends $\mathcal{ALC} - \theta\ CM$, in a number of aspects: (i) it complies with the preferential-DL semantics developed by Britz et al. and by Giordano et al. and which is widely assumed in the literature on reasoning with defeasible ontologies; (ii) it relies on a tailor-made matrix translation we introduced to cater for typicality in concepts and in roles, and (iii) it includes new submatrices, allowing for elegant handling of connections involving typicality.



$$
\begin{aligned}
Path_1 &= \{A(a), s(x_1, y_1)\}\\
Path_2 &= \{A(a), s(x_1, y_1), \neg r(f(x_2), g(y_2))\}\\
Path_3 &= \{A(a), s(x_1, y_1), \neg r(f(x_2), g(y_2)), (x_3, y_3) \ll (a, b)\}\\
Path_4 &= \{A(a), s(x_1, y_1), \neg r(f(x_2), g(y_2)), (x_3, y_3) \ll (a, b), r(x_4, y_4)\}
\end{aligned}
$$

Fig. 4. The connection method in sequent-style for M. It starts from the bottom to the top. As every leaf is an Axiom, we call this proof derivation a connection proof.

As already alluded to in the introduction, the work reported here is one of a pioneer study of connection methods as viable alternatives for reasoning with defeasible ontologies. It is, therefore, part of a broader long-term agenda, of which an immediate next step is endowing RACCOON with the ability to reason over defeasible extensions of $\mathcal{ALCH}$.

The reader conversant with preferential reasoning would have noticed that in this work, we assume preferential entailment, which is a Tarskian notion of consequence and, therefore, monotonic. As pointed out in the literature on non-monotonic reasoning, preferential entailment is not always enough for reasoning defeasibly with exceptions. Stronger, more venturous forms of entailment are often called for. One particular definition thereof, namely the rational closure of a defeasible ontology, has been thoroughly investigated in the context of defeasible $\mathcal{ALC}$ [9,16,26]. Nevertheless, a case has been made for sticking to preferential reasoning in some contexts [24] or for investigating weaker forms of rationality [7,21]. This suggests the debate around rational closure being the baseline for defeasible reasoning remains, in a sense, open.

In any case, the aforementioned approaches to the computation of the rational closure of a defeasible knowledge base rely on a number of calls to a classical (monotonic) reasoner, and therefore the availability of a method for preferential reasoning in $\mathcal{ALCH}^\bullet$ as the one we propose here is an important step in the investigation of stronger forms of entailment in the presence of typicality operators.

The limitations of adopting a single preference ordering in modelling object typicality have already been pointed out [12]. They introduce a notion of context in multi-preference semantics, making it possible for some objects to be more typical than others w.r.t. a context but less typical w.r.t. a different one. Part of our research agenda is, therefore, to extend the method proposed here to deal with multiple preference relations (on both sets of objects and pairs of objects). The tableau system of Britz and Varzinczak can serve as a springboard with which to investigate such an extension to contextual defeasible reasoning.

Acknowledgments

This study has been financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. It also has been supported in part by the project Reconciling Description Logics and Non-Monotonic Reasoning in the Legal Domain (PRC CNRS–FACEPE, France–Brazil). Fred Freitas is partially supported by a Research Productivity grant provided by the Brazilian funding agency National Council for Scientific and Technological Development – CNPq.

This work was partially supported by the ANR Chaire IA BE4musIA: BElief change FOR better MUlti-Source Information Analysis.

References

1. Baader, F., Nutt, W.: Basic Description Logics. In: 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 (2007)
2. Bibel, W.: Automated Theorem Proving. Vieweg Verlag, Wiesbaden (1987)
3. Bonatti, P.: Rational closure for all description logics. Artificial Intelligence **274**, 197–223 (2019)
4. Bonatti, P., Faella, M., Petrova, I., Sauro, L.: A new semantics for overriding in description logics. Artificial Intelligence **222**, 1–48 (2015)
5. Bonatti, P., Faella, M., Sauro, L.: Defeasible inclusions in low-complexity DLs. Journal of Artificial Intelligence Research **42**, 719–764 (2011)
6. Bonatti, P., Sauro, L.: On the logical properties of the nonmonotonic description logic DL^N. Artificial Intelligence **248**, 85–111 (2017)
7. Booth, R., Varzinczak, I.: Conditional inference under disjunctive rationality. In: Leyton-Brown, K., Mausam (eds.) Proceedings of the 35th AAAI Conference on Artificial Intelligence. pp. 6227–6234. AAAI (2021)
8. Bozzato, L., Eiter, T., Serafini, L.: Enhancing context knowledge repositories with justifiable exceptions. Artificial Intelligence **257**, 72–126 (2018)
9. Britz, K., Casini, G., Meyer, T., Moodley, K., Sattler, U., Varzinczak, I.: Principles of KLM-style defeasible description logics. ACM Transactions on Computational Logic **22**(1) (2021)
10. Britz, K., Heidema, J., Meyer, T.: Semantic preferential subsumption. In: Lang, J., Brewka, G. (eds.) Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR). pp. 476–484. AAAI Press/MIT Press (2008)

11. Britz, K., Varzinczak, I.: Toward defeasible $\mathcal{SROIQ}$. In: Proceedings of the 30th International Workshop on Description Logics (2017)
12. Britz, K., Varzinczak, I.: Contextual rational closure for defeasible $\mathcal{ALC}$. Annals of Mathematics and Artificial Intelligence **87**(1-2), 83–108 (2019)
13. Britz, K., Varzinczak, I.: Preferential tableaux for contextual defeasible $\mathcal{ALC}$. In: Cerrito, S., Popescu, A. (eds.) Proceedings of the 28th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX). pp. 39–57. No. 11714 in LNCS, Springer (2019)
14. Casini, G., Meyer, T., Moodley, K., Nortjé, R.: Relevant closure: A new form of defeasible reasoning for description logics. In: Fermé, E., Leite, J. (eds.) Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA). pp. 92–106. No. 8761 in LNCS, Springer (2014)
15. Casini, G., Meyer, T., Moodley, K., Sattler, U., Varzinczak, I.: Introducing defeasibility into OWL ontologies. In: Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K., Staab, S. (eds.) Proceedings of the 14th International Semantic Web Conference (ISWC). pp. 409–426. No. 9367 in LNCS, Springer (2015)
16. Casini, G., Straccia, U.: Rational closure for defeasible description logics. In: Janhunen, T., Niemelä, I. (eds.) Proceedings of the 12th European Conference on Logics in Artificial Intelligence (JELIA). pp. 77–90. No. 6341 in LNCS, Springer-Verlag (2010)
17. Casini, G., Straccia, U.: Defeasible inheritance-based description logics. Journal of Artificial Intelligence Research (JAIR) **48**, 415–473 (2013)
18. Casini, G., Straccia, U., Meyer, T.: A polynomial time subsumption algorithm for nominal safe $\mathcal{ELO}_\perp$ under rational closure. Information Sciences **501**, 588–620 (2019)
19. Freitas, F., Otten, J.: A Connection Calculus for the Description Logic $\mathcal{ALC}$. In: 29th Canadian Conference on Artificial Intelligence. pp. 243–256. Springer, Victoria, BC, Canadá (2016)
20. Freitas, F., Varzinczak, I.: Cardinality restrictions within description logic connection calculi. In: Benzmüller, C., Ricca, F., Parent, X., Roman, D. (eds.) Rules and Reasoning - Second International Joint Conference, RuleML+RR 2018, Luxembourg, September 18-21, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11092, pp. 65–80. Springer (2018). `https://doi.org/10.1007/978-3-319-99906-7\_5`, `https://doi.org/10.1007/978-3-319-99906-7_5`
21. Freund, M.: Injective models and disjunctive relations. Journal of Logic and Computation **3**(3), 231–247 (1993)
22. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.: Preferential description logics. In: Dershowitz, N., Voronkov, A. (eds.) Logic for Programming, Artificial Intelligence, and Reasoning (LPAR). pp. 257–272. No. 4790 in LNAI, Springer (2007)
23. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.: $\mathcal{ALC}+T$: a preferential extension of description logics. Fundamenta Informaticae **96**(3), 341–372 (2009)
24. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.: Preferential vs rational description logics: which one for reasoning about typicality? In: Proceedings of the European Conference on Artificial Intelligence (ECAI). pp. 1069–1070 (2010)
25. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.: A non-monotonic description logic for reasoning about typicality. Artificial Intelligence **195**, 165–202 (2013)
26. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.: Semantic characterization of rational closure: From propositional logic to description logics. Artificial Intelligence **226**, 1–33 (2015)

27. Melo Filho, D., Freitas, F., Otten, J.: RACCOON: A Connection Reasoner for the Description Logic $\mathcal{ALC}$. In: 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning. vol. 46, pp. 200–211. EasyChair, Maun, Botswana (2017)
28. Otten, J.: Implementing connection calculi for first-order modal logics. In: Korovin, K., Schulz, S., Ternovska, E. (eds.) IWIL 2012: The 9th International Workshop on the Implementation of Logics, Merida, Venezuela, March 10, 2012. EPiC Series in Computing, vol. 22, pp. 18–32. EasyChair (2012). https://doi.org/10.29007/82m9, https://doi.org/10.29007/82m9
29. Otten, J.: Advancing automated theorem proving for the modal logics D and S5. In: Benzmüller, C., Otten, J. (eds.) Proceedings of the 4th International Workshop on Automated Reasoning in Quantified Non-Classical Logics (ARQNL 2022) affiliated with the 11th International Joint Conference on Automated Reasoning (IJCAR 2022), Haifa, Israel, August 11, 2022. CEUR Workshop Proceedings, vol. 3326, pp. 81–91. CEUR-WS.org (2022), https://ceur-ws.org/Vol-3326/ARQNL2022_paper5.pdf
30. Pensel, M., Turhan, A.Y.: Reasoning in the defeasible description logic $\mathcal{EL}_\perp$ – computing standard inferences under rational and relevant semantics. International Journal of Approximate Reasoning **112**, 28–70 (2018)
31. Varzinczak, I.: A note on a description logic of concept and role typicality for defeasible reasoning over ontologies. Logica Universalis **12**(3-4), 297–325 (2018)