J Autom Reasoning (2010) 44:371–399
DOI 10.1007/s10817-009-9160-7

$\mathcal{BCDL}$: Basic Constructive Description Logic

Mauro Ferrari · Camillo Fiorentini · Guido Fiorino

Received: 27 March 2008 / Accepted: 30 November 2009 / Published online: 17 December 2009

Abstract In this paper we present $\mathcal{BCDL}$, a description logic based on information terms semantics, which allows a constructive interpretation of $\mathcal{ALC}$ formulas. In the paper we describe the information terms semantics, we define a natural deduction calculus for $\mathcal{BCDL}$ and we show it is sound and complete. As a first application of proof-theoretical properties of the calculus, we show how it fulfills the proofs-as-programs paradigm. Finally, we discuss the role of generators, the main element distinguishing our formalisation from the usual ones.

Keywords Description logics · Constructive logics · Natural deduction

1 Introduction

In Computer Science it often happens that the introduction of a classically based logical system is followed by an analysis of its constructive or intuitionistic counterparts. Indeed, if on the one hand the applicability of a logical system is often driven from

M. Ferrari
Dipartimento di Informatica e Comunicazione, Università degli Studi dell'Insubria,
Via Mazzini 5, 21100, Varese, Italy
e-mail: mauro.ferrari@uninsubria.it

C. Fiorentini (✉)
Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano,
Via Comelico 39, 20135 Milan, Italy
e-mail: fiorenti@dsi.unimi.it

G. Fiorino
Dipartimento di Metodi Quantitativi per le Scienze Economiche Aziendali,
Università degli Studi di Milano-Bicocca, P.zza dell'Ateneo Nuovo 1, 20126 Milan, Italy
e-mail: guido.fiorino@unimib.it

its classical semantics, a constructive analysis allows us to exploit the computational properties of its formulas and proofs. In line with this consideration, one of the reasons for the success of description logics as a knowledge representation formalism is surely their simple classically-based semantics and only in recent works [5, 9, 15, 16] different proposals of a constructive reinterpretation of description logics have been motivated.

In this paper we introduce $\mathcal{BCDL}$ (*Basic Constructive Description Logic*) a description logic based on *information terms semantics*. This is a constructive semantics essentially different from those considered in [5, 9, 15, 16]. Information terms semantics follows the style of BHK (*Brower-Heyting-Kolmogorov*) constructive explanation of logical connectives [22] and belongs to the family of *valuation form semantics* introduced in [13], which has already been applied in several frameworks [6, 7, 11]. In general, information term semantics is a realizability semantics according to the definition given in [10].

Informally, in our setting the truth of an $\mathcal{ALC}$ formula in a classical model must be justified by a mathematical object we call *information term*. For instance, if we prove that an individual c belongs to the concept $\exists R.C$, the information term associated with $\exists R.C$ provides the witness d such that d is an R-successor of c and recursively justifies why d belongs to C.

There are several reasons to consider information term semantics interesting. First of all, differently from other approaches, as the Kripke-style semantics proposed in [5], in our setting the reading of logical connectives is the classical one. This means that the intuition at the base of description logics applications is preserved.

The second important aspect is that our semantics admits a simple proof-theoretical characterisation. In Section 5 we present a natural deduction calculus $\mathcal{ND}_c$ which is sound and complete with respect to the logical consequence relation induced by information term semantics. As we discuss in Section 5, $\mathcal{BCDL}$ is essentially inspired by *Kuroda Logic* [8, 21], the constructive first order logic obtained by extending Intuitionistic first order logic with the axiom schema $\forall x.\neg\neg A(x) \rightarrow \neg\neg\forall x.A(x)$. The calculus $\mathcal{ND}_c$ allows us to develop a proof-theoretical investigation of the logic using the classical techniques of proof-theory. In particular we prove that $\mathcal{BCDL}$ is constructive in the sense that it satisfies an appropriate reformulation in the description logic context of *disjunction property* (DP) and *explicit definability property* (EDP)[1] (see Theorem 10). More than this, we prove that the proofs of $\mathcal{ND}_c$ support the proofs-as-programs paradigm; in particular, at the end of Section 5.1 we show how to exploit the algorithmic content of its proofs.

Finally, another interesting point is that information terms semantics supports a natural notion of *state*. Indeed, in [6] information terms are used to provide a state-semantics for a modeling language based on a first-order constructive logic. A similar characterisation can be provided for the description logic studied in this paper and can be used, e.g., to define an action language over description logics (see [2] for a preliminary work).

[1] In the first order setting a logic L satisfies the disjunction property if $A \vee B \in L$ implies $A \in L$ or $B \in L$; L meets the explicit definability property if $\exists x A(x) \in L$ implies $A(t) \in L$ for some term t.

To enter into the technical details, in this paper we study $\mathcal{BCDL}$ which is the correspondent in the information terms semantics context of the basic description logic $\mathcal{ALC}$ [1, 19]. First of all, in Section 2 we introduce the logic $\mathcal{ALCG}$ obtained by extending $\mathcal{ALC}$ with *generators*, which are concepts whose interpretation is fixed by the language and are similar to concepts defined by means of object names in $\mathcal{ALBO}$ [18]. Generators are needed to model a restricted form of subsumption in our setting; indeed, as we discuss in Section 6, it seems hard to give a constructive proof-theoretical characterisation of "pure" subsumption. In Section 3 we present a sound and complete natural deduction calculus for $\mathcal{ALCG}$. We remark that, being $\mathcal{ALCG}$ a proper extension of $\mathcal{ALC}$, disregarding the rules of the calculus treating generators, we get a sound and complete natural deduction calculus for $\mathcal{ALC}$. In Section 4 we introduce the logic $\mathcal{BCDL}$ and information term semantics. In Section 5 we present a natural deduction calculus for $\mathcal{BCDL}$ and we prove that it is sound and complete w.r.t. constructive consequence. We also show, by means of an example, how the proofs of this calculus support the proofs-as-programs paradigm.

To conclude, a remark about the treatment of negation in $\mathcal{BCDL}$ is needed. As pointed out in [9, 15], in the description logic context different kinds of negation can be given. E.g. in [15] Nelson negation and interpretations of negations in many-valued logics are discussed. In this paper, to simplify the presentation, we treat negation "classically". However a constructive negation can be introduced still preserving the soundness and completeness of the resulting calculus. We leave for future work the investigation on the various kinds of constructive negations that can be given in informationterms semantics.

2 $\mathcal{ALCG}$ Language and Semantics

The language $\mathcal{L}$ for $\mathcal{ALCG}$ is based on the following denumerable sets: the set $\mathtt{NR}$ of *role names*, the set $\mathtt{NC}$ of *concept names* and the set $\mathtt{NI}$ of *individual names*. The main differences with respect to standard presentations are the lack of subsumption and the introduction of a set $\mathtt{NG}$ of special concepts, called *generators*, where $\mathtt{NG} \cap \mathtt{NC} = \emptyset$. The lack of subsumption is due to the fact that it seems hard to give a constructive interpretation of the usual subsumption $A \sqsubseteq B$ and a calculus complete for it (we discuss this issue in Section 6). Thus, we limit subsumption to the case where A is a generator. A generator G is a concept with associated a finite set of individual names $\mathrm{dom}(G)$, we call the *domain of* G, which fixes the interpretation of G. In our language, we use bounded quantified formulas of the kind $\forall_G H$, meaning that every element of $\mathrm{dom}(G)$ belongs to the concept H.

Formally, the language for $\mathcal{L}$ is defined as follows. A *concept* H is an expression of the kind:

$$H \ ::= \ C \mid G \mid \neg H \mid H \sqcap H \mid H \sqcup H \mid \exists R.H \mid \forall R.H$$

where $C \in \mathtt{NC}$, $G \in \mathtt{NG}$ and $R \in \mathtt{NR}$. Let $\mathtt{Var}$ be a denumerable set of *individual variables*; the formulas K of $\mathcal{L}$ are defined according to the following grammar:

$$K \ ::= \ \bot \mid (s, t) : R \mid t : H \mid \forall_G H$$

where $s, t \in \mathtt{NI} \cup \mathtt{Var}$, $R \in \mathtt{NR}$, $G \in \mathtt{NG}$ and H is a concept. An *atomic formula* of $\mathcal{L}$ is a formula of the kind $\bot$, $t : H$, with $H \in \mathtt{NC} \cup \mathtt{NG}$, and $(s, t) : R$; a *negated formula* is a formula of the kind $t : \neg H$. A *simple formula* is either an atomic or a negated formula. A formula is *closed* if it does not contain variables. We use the binary relation symbol $\equiv$ to denote syntactical identity.

In the following we need to refer to languages $\mathcal{L}_{\mathcal{N}}$ generated by subsets $\mathcal{N}$ of $\texttt{NI}$; to do this we have to guarantee that generators are properly treated in $\mathcal{L}_{\mathcal{N}}$. Given $\mathcal{N} \subseteq \texttt{NI}$, let $\texttt{NG}_{\mathcal{N}}$ be the set of generators $G \in \texttt{NG}$ such that $\mathrm{dom}(G) \subseteq \mathcal{N}$; we denote by $\mathcal{L}_{\mathcal{N}}$ the language built on the set $\mathcal{N}$ of individual names, the set $\texttt{NC}$ of concept names, the set $\texttt{NR}$ of role names and the set $\texttt{NG}_{\mathcal{N}}$ of generators.

A *model (interpretation)* $\mathcal{M}$ for $\mathcal{L}_{\mathcal{N}}$ is a pair $(\mathcal{D}^{\mathcal{M}}, .^{\mathcal{M}})$, where $\mathcal{D}^{\mathcal{M}}$ is a non-empty set (the *domain* of $\mathcal{M}$) and $.^{\mathcal{M}}$ is a *valuation* map such that:

- for every $c \in \mathcal{N}$, $c^{\mathcal{M}} \in \mathcal{D}^{\mathcal{M}}$;
- for every $C \in \texttt{NC}$, $C^{\mathcal{M}} \subseteq \mathcal{D}^{\mathcal{M}}$;
- for every $R \in \texttt{NR}$, $R^{\mathcal{M}} \subseteq \mathcal{D}^{\mathcal{M}} \times \mathcal{D}^{\mathcal{M}}$;
- for every $G \in \texttt{NG}_{\mathcal{N}}$, $G^{\mathcal{M}} = \left\{c_1^{\mathcal{M}}, \ldots, c_n^{\mathcal{M}}\right\}$ where $\mathrm{dom}(G) = \{c_1, \ldots, c_n\}$.

We remark that the interpretation of a generator is fixed by the language. A non atomic concept H is interpreted by a subset $H^{\mathcal{M}}$ of $\mathcal{D}^{\mathcal{M}}$ as usual:

$$
\begin{aligned}
(\neg A)^{\mathcal{M}} &= \mathcal{D}^{\mathcal{M}} \setminus A^{\mathcal{M}} \\
(A \sqcap B)^{\mathcal{M}} &= A^{\mathcal{M}} \cap B^{\mathcal{M}} \\
(A \sqcup B)^{\mathcal{M}} &= A^{\mathcal{M}} \cup B^{\mathcal{M}} \\
(\exists R.A)^{\mathcal{M}} &= \left\{c \in \mathcal{D}^{\mathcal{M}} \mid \text{there is } d \in \mathcal{D}^{\mathcal{M}} \text{ s.t. } (c, d) \in R^{\mathcal{M}} \text{ and } d \in A^{\mathcal{M}}\right\} \\
(\forall R.A)^{\mathcal{M}} &= \left\{c \in \mathcal{D}^{\mathcal{M}} \mid \text{for all } d \in \mathcal{D}^{\mathcal{M}}, (c, d) \in R^{\mathcal{M}} \text{ implies } d \in A^{\mathcal{M}}\right\}
\end{aligned}
$$

An *assignment* on a model $\mathcal{M}$ is a map $\theta : \texttt{Var} \to \mathcal{D}^{\mathcal{M}}$. If $t \in \texttt{NI} \cup \texttt{Var}$, $t^{\mathcal{M},\theta}$ is the element of $\mathcal{D}$ denoting t in $\mathcal{M}$ w.r.t. θ, namely: $t^{\mathcal{M},\theta} = \theta(t)$ if $t \in \texttt{Var}$ and $t^{\mathcal{M},\theta} = t^{\mathcal{M}}$ if $t \in \texttt{NI}$.

A formula K is *valid* in $\mathcal{M}$ w.r.t. θ, and we write $\mathcal{M}, \theta \models K$, if $K \not\equiv \bot$ and one of the following conditions holds:

$$
\begin{aligned}
&\mathcal{M}, \theta \models (s, t) : R \ \text{ iff } \ \left(s^{\mathcal{M},\theta}, t^{\mathcal{M},\theta}\right) \in R^{\mathcal{M}} \\
&\mathcal{M}, \theta \models t : H \ \text{ iff } \ t^{\mathcal{M},\theta} \in H^{\mathcal{M}} \\
&\mathcal{M}, \theta \models \forall_G H \ \text{ iff } \ G^{\mathcal{M}} \subseteq H^{\mathcal{M}}
\end{aligned}
$$

We write $\mathcal{M} \models K$ iff $\mathcal{M}, \theta \models K$ for every assignment θ. If Γ is a set of formulas, $\mathcal{M} \models \Gamma$ means that $\mathcal{M} \models K$ for every $K \in \Gamma$. We say that K is a *logical consequence* of Γ, and we write $\Gamma \models K$ iff, for every $\mathcal{M}$ and every θ, $\mathcal{M}, \theta \models \Gamma$ implies $\mathcal{M}, \theta \models K$. We use the symbol $\not\models$ to indicate that one of the above relations does not hold.

In our approach, the domain of a generator is fixed by the language, and this simplifies the presentation of the results discussed in the paper. Alternatively, we can extend the language with singleton concepts as in $\mathcal{ALBO}$ [18]. In this case, for every $c \in \texttt{NI}$, $\{c\}$ denotes a concept which is interpreted in $\mathcal{M}$ as the set $\{c^{\mathcal{M}}\}$. A generator G such that $\mathrm{dom}(G) = \{c_1, \ldots, c_n\}$ can be defined as the concept $\{c_1\} \sqcup \cdots \sqcup \{c_n\}$. The drawback is that, in the statements of the next propositions, we have to explicitly mention the definitions of the generators at hand, whereas we assume that generator domains are implicitly defined by the language.

As usual, a theory T consists of an *ABox* and a *TBox*. An ABox is a finite set of concept assertions and role assertions, where:

- a concept assertion is a formula of the kind $c : C$, with $c \in \texttt{NI}$ and $C \in \texttt{NC}$;
- a role assertion is a formula of the kind $(c, d) : R$, with $c, d \in \texttt{NI}$ and $R \in \texttt{NR}$.

A TBox is a finite set of universally quantified formulas of the form $\forall_G H$, with $G \in \texttt{NG}$ and H a concept.

Now, let us introduce the example we use all along this paper.

Example 1 Our example is inspired by the classical one of [3]. Let $\mathcal{T}$ be the TBox consisting of the formulas:

$$(Ax_1) \;=\; \forall_{\texttt{FOOD}}\exists\texttt{goesWith}.\texttt{COLOR}$$

$$(Ax_2) \;=\; \forall_{\texttt{COLOR}}\exists\texttt{isColorOf}.\texttt{WINE}$$

where $\texttt{WINE}$ is a concept name, $\texttt{isColorOf}$ and $\texttt{goesWith}$ are role names, $\texttt{FOOD}$ and $\texttt{COLOR}$ are generators. Let us consider the set of individual names

$$\mathcal{W} \;=\; \{\texttt{barolo}, \texttt{chardonnay}, \texttt{meat}, \texttt{fish}, \texttt{red}, \texttt{white}\}$$

and let

$$\text{dom}(\texttt{FOOD}) \;=\; \{\texttt{fish}, \texttt{meat}\} \qquad \text{dom}(\texttt{COLOR}) \;=\; \{\texttt{red}, \texttt{white}\}$$

the domain of the generators. Finally, let $\mathcal{A}$ be the ABox consisting of the following role and concept assertions:

```
    barolo:WINE                    (red,barolo):isColorOf
chardonnay:WINE              (white,chardonnay):isColorOf
                                   (fish,white):goesWith
                                     (meat,red):goesWith
```

A model $\mathcal{M}$ of $\mathcal{A} \cup \mathcal{T}$ must interpret $\texttt{FOOD}$ as the set $\{\texttt{fish}^{\mathcal{M}}, \texttt{meat}^{\mathcal{M}}\}$ and $\texttt{COLOR}$ as $\{\texttt{white}^{\mathcal{M}}, \texttt{red}^{\mathcal{M}}\}$. The intuitive meaning of (Ax_1) is that every food has an appropriate color (the color of the more appropriate wine for the food) represented by the role name $\texttt{goesWith}$; (Ax_2) pairs a color to a wine. In $\mathcal{M}$, the set of wines (i.e., the set interpreting the $\texttt{WINE}$ concept) contains the elements $\texttt{barolo}^{\mathcal{M}}$ and $\texttt{chardonnay}^{\mathcal{M}}$ and possibly other elements (indeed, since $\texttt{WINE}$ is not a generator, its interpretation is not fixed by the language).

3 The Natural Calculus $\mathcal{ND}$ for $\mathcal{ALCG}$

In this section we introduce a calculus $\mathcal{ND}$ for $\mathcal{ALCG}$ similar to the usual natural deduction calculus for classical logic and we prove it is sound and complete. We refer the reader to [17, 23] for a detailed presentation of natural deduction calculi and their notation. The rules of $\mathcal{ND}$ are those given in Tables 1 and 2 and the rule

$$\dfrac{\begin{array}{c}\Gamma, [t : \neg H] \\ \vdots\,\pi' \\ \bot\end{array}}{t : H}\;\neg E$$

We remark that we have *introduction* and *elimination* rules for all the logical constants; some rules (namely, $\neg I$, $\neg E$, $\sqcup E$, $\exists E$ and $\forall I$) allow us to *discharge* some of the assumptions (we put between square brackets the discharged assumptions). The rules $\exists E$ and $\forall I$ need a side condition on the rule parameter to guarantee correctness. The

Table 1 Rules of the calculi $\mathcal{ND}$ and $\mathcal{ND}_c$

$$\dfrac{\begin{array}{c}\Gamma_1 \\ \vdots\,\pi_1 \\ t : H\end{array} \quad \begin{array}{c}\Gamma_2 \\ \vdots\,\pi_2 \\ t : \neg H\end{array}}{\bot}\;\bot I \qquad \dfrac{\begin{array}{c}\Gamma \\ \vdots\,\pi' \\ \bot\end{array}}{K}\;\bot E \qquad \dfrac{\begin{array}{c}\Gamma, [t : H] \\ \vdots\,\pi' \\ \bot\end{array}}{t : \neg H}\;\neg I$$

$$\dfrac{\begin{array}{c}\Gamma_1 \\ \vdots\,\pi_1 \\ t : A_1\end{array} \quad \begin{array}{c}\Gamma_2 \\ \vdots\,\pi_2 \\ t : A_2\end{array}}{t : A_1 \sqcap A_2}\;\sqcap I \qquad \dfrac{\begin{array}{c}\Gamma \\ \vdots\,\pi' \\ t : A_1 \sqcap A_2\end{array}}{t : A_k}\;\sqcap E_k \; k \in \{1, 2\}$$

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots\,\pi' \\ t : A_k\end{array}}{t : A_1 \sqcup A_2}\;\sqcup I_k \; k \in \{1, 2\} \qquad \dfrac{\begin{array}{c}\Gamma_1 \\ \vdots\,\pi_1 \\ t : A_1 \sqcup A_2\end{array} \quad \begin{array}{c}\Gamma_2, [t : A_1] \\ \vdots\,\pi_2 \\ K\end{array} \quad \begin{array}{c}\Gamma_3, [t : A_2] \\ \vdots\,\pi_3 \\ K\end{array}}{K}\;\sqcup E$$

$$\dfrac{(t, u) : R \quad \begin{array}{c}\Gamma' \\ \vdots\,\pi' \\ u : A\end{array}}{t : \exists R.A}\;\exists I \qquad \dfrac{\begin{array}{c}\Gamma_1 \\ \vdots\,\pi_1 \\ t : \exists R.A\end{array} \quad \begin{array}{c}\Gamma_2, [(t, p) : R,\ p : A] \\ \vdots\,\pi_2 \\ K\end{array}}{K}\;\exists E$$

where $p \in \mathtt{Var}$ does not occur in $\Gamma_2 \cup \{K\}$ and $p \neq t$

$$\dfrac{\begin{array}{c}\Gamma, [(t, p) : R] \\ \vdots\,\pi' \\ p : A\end{array}}{t : \forall R.A}\;\forall I$$

where $p \in \mathtt{Var}$, p does not occur in Γ and $p \neq t$

$$\dfrac{(s, t) : R \quad \begin{array}{c}\Gamma' \\ \vdots\,\pi' \\ s : \forall R.A\end{array}}{t : A}\;\forall E$$

Table 2 Rules for generators

$$\frac{}{c : G}\ \text{AxGen} \quad \text{where } c \in \mathrm{dom}(G) \qquad\qquad \frac{\begin{array}{c}\Gamma \\ \vdots\ \pi' \\ c : G\end{array}}{\bot}\ \bot_{Dom} \quad \text{where } c \in \mathtt{NI} \text{ and } c \notin \mathrm{dom}(G)$$

$$\frac{\begin{array}{ccc}\Gamma_1 & & \Gamma_n \\ \vdots\ \pi_1 & & \vdots\ \pi_n \\ c_1 : A & \ldots & c_n : A\end{array}}{\forall_G A}\ \forall_G I \quad \begin{array}{l}\text{where} \\ \mathrm{dom}(G) = \{c_1, \ldots, c_n\}\end{array} \qquad\qquad \frac{\forall_G A \quad \begin{array}{c}\Gamma' \\ \vdots\ \pi' \\ t : G\end{array}}{t : A}\ \forall_G E$$

rules for generators given in Table 2 depend on the domain of the generators. Finally, the above rule $\neg E$ corresponds to the classical rule of *reductio ad absurdum*.

By $\pi : \Gamma \vdash K$, with Γ a set of formulas, we denote a proof of $\Gamma \vdash K$, that is a proof of the formula K with undischarged assumptions in Γ. We write $\Gamma \vdash_{\mathcal{ALCG}} K$ if there exists a proof $\pi : \Gamma \vdash K$ of $\mathcal{ND}$. Obviously, $\Gamma \nvdash_{\mathcal{ALCG}} K$ means that no proof of $\Gamma \vdash K$ exists in $\mathcal{ND}$.

It is easy to prove by induction on the depth of proofs, the soundness of $\mathcal{ND}$.

Theorem 1 (Soundness) *Let $\pi : \Gamma \vdash K$ be a proof of $\mathcal{ND}$. For every model $\mathcal{M}$ and assignment θ, $\mathcal{M}, \theta \models \Gamma$ implies $\mathcal{M}, \theta \models K$.*

The proof of completeness follows the usual lines. First of all we give an appropriate notion of saturated set and we show that any consistent set can be extended to a consistent and saturated set (Lemma 1). Then we prove that any consistent saturated set is satisfiable (Theorem 2).

Definition 1 ($\mathcal{ALCG}$-saturated set) Let $\mathcal{N} \subseteq \mathtt{NI}$ be a set of individual names and let Δ be a set of closed formulas of $\mathcal{L}_{\mathcal{N}}$. Δ is *$\mathcal{ALCG}$-saturated* in $\mathcal{N}$ iff the following conditions hold:

(1) $c : A \sqcap B \in \Delta$ implies $c : A \in \Delta$ and $c : B \in \Delta$.
(2) $c : A \sqcup B \in \Delta$ implies $c : A \in \Delta$ or $c : B \in \Delta$.
(3) $c : \exists R.A \in \Delta$ implies that there exists $d \in \mathcal{N}$ such that $(c, d) : R \in \Delta$ and $d : A \in \Delta$.
(4) $c : \forall R.A \in \Delta$ and $(c, d) : R \in \Delta$ imply $d : A \in \Delta$.
(5) $\forall_G A \in \Delta$ implies

   (i) $c : G \in \Delta$ iff $c \in \mathrm{dom}(G)$;
   (ii) for every $c \in \mathrm{dom}(G)$, $c : A \in \Delta$.

(6) $c : \neg(A \sqcap B) \in \Delta$ implies $c : \neg A \in \Delta$ or $c : \neg B \in \Delta$.
(7) $c : \neg(A \sqcup B) \in \Delta$ implies $c : \neg A \in \Delta$ and $c : \neg B \in \Delta$.
(8) $c : \neg\exists R.A \in \Delta$ and $(c, d) : R \in \Delta$ imply $d : \neg A \in \Delta$.

(9) $c : \neg\forall R.A \in \Delta$ implies that there exists $d \in \mathcal{N}$ such that $(c, d) : R \in \Delta$ and $d : \neg A \in \Delta$.
(10) $c : \neg\neg A \in \Delta$ implies $c : A \in \Delta$.

Lemma 1 *Let $\mathcal{N}$ be a finite set of individual names, let Γ be a finite set of closed formulas of $\mathcal{L}_{\mathcal{N}}$ and let K be a closed formula of $\mathcal{L}_{\mathcal{N}}$ such that $\Gamma \not\vdash_{\mathcal{ALCG}} K$. There exists a finite set of individual names $\overline{\mathcal{N}} \supseteq \mathcal{N}$ and a finite set $\overline{\Delta} \supseteq \Gamma$ of closed formulas of $\mathcal{L}_{\overline{\mathcal{N}}}$ such that:*

(1) *$\overline{\Delta}$ is $\mathcal{ALCG}$-saturated in $\overline{\mathcal{N}}$.*
(2) *$\overline{\Delta} \not\vdash_{\mathcal{ALCG}} K$.*
(3) *For every $c, d \in \mathcal{N}$ and every $R \in \mathtt{NR}$, $(c, d) : R \in \overline{\Delta}$ iff $(c, d) : R \in \Gamma$.*
(4) *For every $G \in \mathtt{NG}_{\mathcal{N}}$, $c : G \in \overline{\Delta}$ iff $c \in dom(G)$.*

Proof This is a quite standard saturation construction, but some care is needed to guarantee the finiteness of $\overline{\Delta}$. Let $\mathcal{C}$ be a numerable set of individual names such that $\mathcal{C} \cap \mathcal{N} = \emptyset$. We build a sequence $\mathcal{C}_k$ of finite sets of individual names, a sequence Δ_k of finite sets of closed formulas of $\mathcal{L}_{\mathcal{C}_k}$ such that $\Delta_k \not\vdash_{\mathcal{ALCG}} K$ and a sequence Φ_k of formulas treated up to step k. We begin with:

$$\mathcal{C}_0 = \mathcal{N} \qquad \Delta_0 = \Gamma \cup \{ c : G \mid G \in \mathtt{NG}_{\mathcal{N}} \text{ and } c \in \mathrm{dom}(G) \} \qquad \Phi_o = \emptyset\,.$$

Since $\mathcal{ND}$ contains the rule $AxGen$, $\Gamma \not\vdash_{\mathcal{ALCG}} K$ implies $\Delta_0 \not\vdash_{\mathcal{ALCG}} K$. Let $c \in \mathtt{NC}$. We say that Δ_k is saturated w.r.t. c iff the following conditions hold:

- The non-atomic formulas $c : H$ belonging to Δ_k are of the kind $c : \forall R.A$ or $c : \neg\exists R.A$.
- If $(c, c) : R \in \Delta_k$, we also require that:
  - for every $c : \forall R.A \in \Delta_k$, there exists $i < k$ such that $c : A \in \Delta_i$;
  - for every $c : \neg\exists R.B \in \Delta_k$, there exists $j < k$ such that $c : \neg B \in \Delta_j$.

Given $k > 0$, let H be a formula of Δ_{k-1} such that $H \notin \Phi_{k-1}$. We define the sets $\mathcal{C}_k$, Δ_k and Φ_k according to the form of H; by Δ^H_{k-1} we denote the set $\Delta_{k-1} \setminus \{H\}$.

(i) If $H \equiv c : C$ with $C \in \mathtt{NC} \cup \mathtt{NG}_{\mathcal{N}}$ or $H \equiv (c, d) : R$ with $R \in \mathtt{NR}$, then $\mathcal{C}_k = \mathcal{C}_{k-1}$, $\Delta_k = \Delta_{k-1}$ and $\Phi_k = \Phi_{k-1} \cup \{H\}$.
(ii) If $H \equiv c : A \sqcap B$, then $\mathcal{C}_k = \mathcal{C}_{k-1}$, $\Delta_k = \Delta^H_{k-1} \cup \{c : A, c : B\}$ and $\Phi_k = \Phi_{k-1} \cup \{H\}$.
(iii) If $H \equiv c : A \sqcup B$, then $\mathcal{C}_k = \mathcal{C}_{k-1}$ and $\Phi_k = \Phi_{k-1} \cup \{H\}$. Moreover, $\Delta_k = \Delta^H_{k-1} \cup \{c : A\}$ if $\Delta^H_{k-1} \cup \{c : A\} \not\vdash_{\mathcal{ALCG}} K$ and $\Delta_k = \Delta^H_{k-1} \cup \{c : B\}$ otherwise.
(iv) If $H \equiv c : \exists R.A$, let $d \in \mathcal{C}$ such that $d \notin \mathcal{C}_{k-1}$. Then, $\mathcal{C}_k = \mathcal{C}_{k-1} \cup \{d\}$ and $\Delta_k = \Delta^H_{k-1} \cup \{(c, d) : R, d : A\}$. Moreover, $\Phi_k = (\Phi_{k-1} \cup \{H\}) \setminus \Sigma$ where Σ is the set of the formulas of the kind $c : \forall R.B$ and $c : \neg\exists R.B$ occurring in Φ_{k-1}.
(v) If $H \equiv c : \forall R.A$, let $\{d_1, \ldots, d_n\}$ be the set of $d \in \mathcal{C}_{k-1}$ s.t. $(c, d) : R \in \Delta_{k-1}$ and let $\Theta = \{d_1 : A, \ldots, d_n : A\}$. Then, $\mathcal{C}_k = \mathcal{C}_{k-1}$ and $\Phi_k = \Phi_{k-1} \cup \{H\}$. Moreover, if Δ_{k-1} is not saturated w.r.t. c, then $\Delta_k = \Delta_{k-1} \cup \Theta$, otherwise $\Delta_k = \Delta^H_{k-1} \cup \Theta$.

(vi) If $H \equiv \forall_G A$, then $G \in \mathrm{NG}_{\mathcal{N}}$ and $\mathrm{dom}(G) = \{c_1, \ldots, c_n\} \subseteq \mathcal{N}$. Then, $\mathcal{C}_k = \mathcal{C}_{k-1}$, $\Delta_k = \Delta^H_{k-1} \cup \{c_1 : A, \ldots, c_n : A\}$ and $\Phi_k = \Phi_{k-1} \cup \{H\}$.

(vii) If $H \equiv c : \neg(A \sqcap B)$, then $\mathcal{C}_k = \mathcal{C}_{k-1}$ and $\Phi_k = \Phi_{k-1} \cup \{H\}$. Moreover, $\Delta_k = \Delta^H_{k-1} \cup \{c : \neg A\}$ if $\Delta^H_{k-1}, c : \neg A \not\vdash_{\mathcal{ALCG}} K$ and $\Delta_k = \Delta^H_{k-1} \cup \{c : \neg B\}$ otherwise.

(viii) If $H \equiv c : \neg(A \sqcup B)$, then $\mathcal{C}_k = \mathcal{C}_{k-1}$, $\Delta_k = \Delta^H_{k-1} \cup \{c : \neg A, c : \neg B\}$ and $\Phi_k = \Phi_{k-1} \cup \{H\}$.

(ix) If $H \equiv c : \neg\exists R.A$ let $\{d_1, \ldots, d_n\}$ be the set of $d \in \mathcal{C}_{k-1}$ s.t. $(c, d) : R \in \Delta_{k-1}$ and let $\Theta = \{d_1 : \neg A, \ldots, d_n : \neg A\}$. Then, $\mathcal{C}_k = \mathcal{C}_{k-1}$ and $\Phi_k = \Phi_{k-1} \cup \{H\}$. Moreover, if Δ_{k-1} is not saturated w.r.t. c, then $\Delta_k = \Delta_{k-1} \cup \Theta$, otherwise $\Delta_k = \Delta^H_{k-1} \cup \Theta$.

(x) If $H \equiv c : \neg\forall R.A$, let $d \in \mathcal{C}$ such that $d \notin \mathcal{C}_{k-1}$. Then, $\mathcal{C}_k = \mathcal{C}_{k-1} \cup \{d\}$ and $\Delta_k = \Delta^H_{k-1} \cup \{(c, d) : R, d : \neg A\}$. Moreover, $\Phi_k = (\Phi_{k-1} \cup \{H\}) \setminus \Sigma$ where Σ is the set of the formulas of the kind $c : \forall R.B$ and $c : \neg\exists R.B$ occurring in Φ_{k-1}.

(xi) If $H \equiv c : \neg\neg A$, then $\mathcal{C}_k = \mathcal{C}_{k-1}$, $\Delta_k = \Delta^H_{k-1} \cup \{c : A\}$ and $\Phi_k = \Phi_{k-1} \cup \{H\}$.

We remark that in points (iv) and (x) we delete the formulas of the kind $c : \forall R.B$ and $c : \neg\exists R.B$ from the set of the formulas treated up to step k. This is needed to guarantee that these formulas are correctly saturated w.r.t. the new role assertion $(c, d) : R$ introduced in points (iv) and (x). Clearly, for every $k \geq 0$, $\mathcal{C}_k$ and Δ_k are finite. It is easy to check, by induction on $k \geq 0$, that $\Delta_k \not\vdash_{\mathcal{ALCG}} K$. For instance, let us suppose that Δ_k is defined as in Point (vii) when $\Delta^H_k, c : \neg A \vdash_{\mathcal{ALCG}} K$; we have to show that $\Delta^H_k, c : \neg B \not\vdash_{\mathcal{ALCG}} K$. If $\Delta^H_k, c : \neg B \vdash_{\mathcal{ALCG}} K$, by the fact that $c : \neg(A \sqcap B) \vdash_{\mathcal{ALCG}} c : \neg A \sqcup \neg B$, we get $\Delta^H_k, c : \neg(A \sqcap B) \vdash_{\mathcal{ALCG}} K$, namely $\Delta_{k-1} \vdash_{\mathcal{ALCG}} K$, in contradiction with the induction hypothesis. The other cases are similar. In particular, for negated formulas we exploit the following facts:

- $c : \neg(A \sqcup B) \vdash_{\mathcal{ALCG}} c : \neg A$ and $c : \neg(A \sqcup B) \vdash_{\mathcal{ALCG}} c : \neg B$.
- $c : \neg\exists R.A \vdash_{\mathcal{ALCG}} c : \forall R.\neg A$.
- $c : \neg\forall R.A \vdash_{\mathcal{ALCG}} c : \exists R.\neg A$.
- $c : \neg\neg A \vdash_{\mathcal{ALCG}} c : A$.

One can prove the following properties:

(a) If $c : \forall R.A \in \Delta_k$, there is $i \geq k$ s.t. $c : \forall R.A \in \Delta_i$ and Δ_i is saturated w.r.t. c.
(b) If $c : \neg\exists R.B \in \Delta_k$, there is $j \geq k$ s.t. $c : \neg\exists R.B \in \Delta_j$ and Δ_j is saturated w.r.t. c.
(c) If Δ_m is saturated w.r.t. c then, for every $l \geq m$, $(c, d) : R \in \Delta_l$ iff $(c, d) : R \in \Delta_m$.

Now, let $\overline{\mathcal{N}} = \bigcup_{i \geq 0} \mathcal{C}_i$ and $\overline{\Delta} = \bigcup_{i \geq 0} \Delta_i$. Clearly, $\overline{\mathcal{N}} \supseteq \mathcal{N}$, $\overline{\Delta} \supseteq \Gamma$ and $\overline{\Delta} \not\vdash_{\mathcal{ALCG}} K$. It is easy to prove that $\overline{\Delta}$ is $\mathcal{ALCG}$-saturated in $\overline{\mathcal{N}}$. As an example, let $c : \forall R.A \in \overline{\Delta}$ and $(c, d) : R \in \overline{\Delta}$. By Point (a), there is a Δ_m saturated w.r.t. c such that $c : \forall R.A \in \Delta_m$. By Point (b), $(c, d) : R \in \Delta_m$. Thus, for some $j > m$, $d : A \in \Delta_j$, which implies $d : A \in \overline{\Delta}$. Point (3) follows from the fact that, whenever we add $(c, d) : R$ to some Δ_k, d is a new constant. Thus, if $(c, d) : R \in \overline{\Delta}$ and both c and d belong to $\mathcal{N}$, we have $(c, d) : R \in \Gamma$. As for Point (4), if $c : G \in \overline{\Delta}$ then $c \in \mathrm{dom}(G)$, otherwise by applying rule $\bot_{Dom}$ of Table 2 we would get $\overline{\Delta} \vdash_{\mathcal{ALCG}} \bot$, in contradiction with $\overline{\Delta} \not\vdash_{\mathcal{ALCG}} K$. Finally, we outline the proof of the finiteness of $\overline{\Delta}$. Firstly, we note that, except the cases (v) and (ix), the set Δ_k is obtained from Δ_{k-1} by replacing a formula H with one or more formulas simpler than H. In cases (v) and (ix), this happens only if Δ_{k-1} is saturated w.r.t. c. By points (a) and (b), this eventually happens, so we cannot have infinitely many applications of cases (v) and (ix). $\square$

We point out that the "only if" part of Point (3) of the previous lemma (and of Point (iii) of the next theorem) is not required to prove the completeness of $\mathcal{ND}$, but it is crucial to prove the completeness of the constructive calculus of Section 5.

Theorem 2 *Let $\mathcal{N}$ be a finite set of individual names, let Γ be a finite set of closed formulas of $\mathcal{L}_\mathcal{N}$ and let K be a simple closed formula of $\mathcal{L}_\mathcal{N}$ such that $\Gamma \not\vdash_{\mathcal{ALCG}} K$. There exists a finite model $\mathcal{M}$ for $\mathcal{L}_\mathcal{N}$ such that:*

(i) $\mathcal{M} \models \Gamma$.
(ii) $\mathcal{M} \not\models K$.
(iii) *For every $c, d \in \mathcal{N}$ and every $R \in \mathtt{NR}$, $\mathcal{M} \models (c, d) : R$ iff $(c, d) : R \in \Gamma$.*
(iv) *For every $G \in \mathtt{NG}_\mathcal{N}$, $\mathcal{M} \models c : G$ iff $c \in dom(G)$.*

Proof Let us assume K to be an atomic formula. By Lemma 1, there exist a finite set of individual names $\overline{\mathcal{N}} \supseteq \mathcal{N}$ and a finite set $\overline{\Delta}$ such that $\Gamma \subseteq \overline{\Delta} \subseteq \mathcal{L}_{\overline{\mathcal{N}}}$, $\overline{\Delta}$ is $\mathcal{ALCG}$-saturated in $\overline{\mathcal{N}}$, $\overline{\Delta} \not\vdash_{\mathcal{ALCG}} K$ and $\overline{\Delta}$ satisfies Points (3)–(4) of the lemma. Let $\mathcal{M} = (\mathcal{D}^\mathcal{M}, .^\mathcal{M})$ be the model for $\mathcal{L}_{\overline{\mathcal{N}}}$ done as follows:

- $\mathcal{D}^\mathcal{M} = \overline{\mathcal{N}}$;
- For every $C \in \mathtt{NC} \cup \mathtt{NG}_\mathcal{N}$, $C^\mathcal{M}$ is the set of $c \in \mathcal{D}^\mathcal{M}$ such that $c : C \in \overline{\Delta}$;
- For every $R \in \mathtt{NR}$, $R^\mathcal{M}$ is the set of pairs $(c, d) \in \mathcal{D}^\mathcal{M} \times \mathcal{D}^\mathcal{M}$ s.t. $(c, d) : R \in \overline{\Delta}$.

By Points (3)–(4) of Lemma 1, we immediately have:

(3') For every $R \in \mathtt{NR}$, $(c, d) \in R^\mathcal{M}$ iff $(c, d) : R \in \Gamma$;
(4') For every $G \in \mathtt{NG}_\mathcal{N}$, $c \in G^\mathcal{M}$ iff $c \in \mathrm{dom}(G)$.

One can easily check that the definition of $\mathcal{M}$ is sound and that $\mathcal{M}$ is finite. Now we prove that:

(*) $D \in \overline{\Delta}$ implies $\mathcal{M} \models D$.

The proof is by induction on the structure of D. If $D \equiv c : C$, with $C \in \mathtt{NC} \cup \mathtt{NG}_\mathcal{N}$, or $D \equiv (c, d) : R$, (*) follows by the definition of $\mathcal{M}$. If $D \equiv c : \neg C$, with $C \in \mathtt{NC} \cup \mathtt{NG}_\mathcal{N}$, we cannot have $c : C \in \overline{\Delta}$, otherwise $\overline{\Delta} \vdash_{\mathcal{ALCG}} \bot$ and $\overline{\Delta} \vdash_{\mathcal{ALCG}} K$ would follow. By definition of $C^\mathcal{M}$, we have $\mathcal{M} \not\models c : C$, hence $\mathcal{M} \models c : \neg C$. The other cases easily follow by the properties of $\mathcal{ALCG}$-saturated sets.

Point (i) of the assertion immediately follows from (*). Now, we prove Point (ii). If $K \equiv \bot$, then $\mathcal{M} \not\models \bot$. If $K \equiv c : C$, with $C \in \mathtt{NC} \cup \mathtt{NG}_\mathcal{N}$ and $\mathcal{M} \models c : C$, by definition of $C^\mathcal{M}$ we should have $c : C \in \overline{\Delta}$, in contradiction with the hypothesis $\overline{\Delta} \not\vdash_{\mathcal{ALCG}} K$. The case $K \equiv (c, d) : R$ is similar. Points (iii) and (iv) follow from (3') and (4') respectively. Since $\mathcal{M}$ is a model for $\mathcal{L}_\mathcal{N}$, this concludes the proof of the assertion in the case where K is an atomic formula.

Now let us consider the case $K \equiv t : \neg H$. Since $\overline{\Gamma} \not\vdash_{\mathcal{ALCG}} t : \neg H$, we have $\overline{\Gamma}, t : H \not\vdash_{\mathcal{ALCG}} \bot$. As proved in the previous case, there exists a model $\mathcal{M}$ for $\mathcal{L}_\mathcal{N}$ such that $\mathcal{M} \models \overline{\Gamma}$, $\mathcal{M} \models t : H$, and $\mathcal{M}$ satisfies (iii) and (iv). Thus, $\mathcal{M} \not\models t : \neg H$, and the theorem holds. $\square$

The above theorem implies that, if Γ is a consistent set of closed formulas (i.e., $\Gamma \not\vdash_{\mathcal{ALCG}} \bot$), then Γ has a model. The completeness theorem follows along the usual lines:

Theorem 3 (Completeness) *For every finite* Γ, $\Gamma \models K$ *implies* $\Gamma \vdash_{\mathcal{ALCG}} K$.

To conclude this section we remark that disregarding generators and quantification over generators in $\mathcal{ALCG}$ we get the usual language of $\mathcal{ALC}$. More precisely, let $\mathcal{ND}'$ be the natural deduction calculus only consisting of the rules of Table 1 and the rule $\neg E$. From the above proofs we get that $\mathcal{ND}'$ is a sound and complete calculus for $\mathcal{ALC}$.

4 $\mathcal{BCDL}$ and Information Terms Semantics

In this section we introduce the logic $\mathcal{BCDL}$. It uses $\mathcal{ALCG}$ language, but its semantics is based on *information terms*. Intuitively, an information term α for a formula A is a possible explanation of the truth of A in the spirit of the BHK interpretation of logical connectives [22]. Formally, given a finite set of individual names $\mathcal{N} \subseteq \mathtt{NI}$ and a closed formula K of $\mathcal{L}_{\mathcal{N}}$, we define the set of information terms $\mathrm{IT}_{\mathcal{N}}(K)$ by induction on K as follows.

$$
\begin{aligned}
\mathrm{IT}_{\mathcal{N}}(K) &= \{\mathtt{tt}\}, \text{ if } K \text{ is a simple formula}\\
\mathrm{IT}_{\mathcal{N}}(c : A_1 \sqcap A_2) &= \{ (\alpha, \beta) \mid \alpha \in \mathrm{IT}_{\mathcal{N}}(c : A_1) \text{ and } \beta \in \mathrm{IT}_{\mathcal{N}}(c : A_2) \}\\
\mathrm{IT}_{\mathcal{N}}(c : A_1 \sqcup A_2) &= \{ (k, \alpha) \mid k \in \{1, 2\} \text{ and } \alpha \in \mathrm{IT}_{\mathcal{N}}(c : A_k) \}\\
\mathrm{IT}_{\mathcal{N}}(c : \exists R.A) &= \{ (d, \alpha) \mid d \in \mathcal{N} \text{ and } \alpha \in \mathrm{IT}_{\mathcal{N}}(d : A) \}\\
\mathrm{IT}_{\mathcal{N}}(c : \forall R.A) &= \{ \phi : \mathcal{N} \to \textstyle\bigcup_{d \in \mathcal{N}} \mathrm{IT}_{\mathcal{N}}(d : A) \mid \phi(d) \in \mathrm{IT}_{\mathcal{N}}(d : A) \}\\
\mathrm{IT}_{\mathcal{N}}(\forall_G A) &= \{ \phi : \mathrm{dom}(G) \to \textstyle\bigcup_{d \in \mathrm{dom}(G)} \mathrm{IT}_{\mathcal{N}}(d : A) \mid \phi(d) \in \mathrm{IT}_{\mathcal{N}}(d : A) \}
\end{aligned}
$$

We recall that in the last case $\mathrm{dom}(G) \subseteq \mathcal{N}$. Let $\mathcal{M}$ be a model for $\mathcal{L}_{\mathcal{N}}$, K a closed formula of $\mathcal{L}_{\mathcal{N}}$ and $\eta \in \mathrm{IT}_{\mathcal{N}}(K)$. We define the *realizability relation* $\mathcal{M} \rhd \langle \eta \rangle\, K$ by induction on the structure of K.

$$
\begin{aligned}
&\mathcal{M} \rhd \langle \mathtt{tt} \rangle\, K \text{ iff } \mathcal{M} \models K, \text{ where } K \text{ is a simple formula}\\
&\mathcal{M} \rhd \langle (\alpha, \beta) \rangle\, c : A_1 \sqcap A_2 \text{ iff } \mathcal{M} \rhd \langle \alpha \rangle\, c : A_1 \text{ and } \mathcal{M} \rhd \langle \beta \rangle\, c : A_2\\
&\mathcal{M} \rhd \langle (k, \alpha) \rangle\, c : A_1 \sqcup A_2 \text{ iff } \mathcal{M} \rhd \langle \alpha \rangle\, c : A_k\\
&\mathcal{M} \rhd \langle (d, \alpha) \rangle\, c : \exists R.A \text{ iff } \mathcal{M} \models (c, d) : R \text{ and } \mathcal{M} \rhd \langle \alpha \rangle\, d : A\\
&\mathcal{M} \rhd \langle \phi \rangle\, c : \forall R.A \text{ iff } \mathcal{M} \models c : \forall R.A \text{ and, for every } d \in \mathcal{N},\\
&\qquad\qquad \mathcal{M} \models (c, d) : R \text{ implies } \mathcal{M} \rhd \langle \phi(d) \rangle\, d : A\\
&\mathcal{M} \rhd \langle \phi \rangle\, \forall_G A \text{ iff, for every } d \in \mathrm{dom}(G),\ \mathcal{M} \rhd \langle \phi(d) \rangle\, d : A
\end{aligned}
$$

It is easy to prove that:

Lemma 2 *Let* $\mathcal{N}$ *be a finite subset of* $\mathtt{NI}$, K *a closed formula of* $\mathcal{L}_{\mathcal{N}}$ *and* $\eta \in \mathrm{IT}_{\mathcal{N}}(K)$. *For every model* $\mathcal{M}$, $\mathcal{M} \rhd \langle \eta \rangle\, K$ *implies* $\mathcal{M} \models K$.

This means that the constructive semantics is compatible with the classical one. If Γ is a finite set of closed formulas $\{K_1, \ldots, K_n\}$ of $\mathcal{L}_{\mathcal{N}}$, $\textsc{it}_{\mathcal{N}}(\Gamma)$ denotes the set of n-tuples $\overline{\eta} = (\eta_1, \ldots, \eta_n)$ such that, for every $1 \leq j \leq n$, $\eta_j \in \textsc{it}_{\mathcal{N}}(K_j)$; $\mathcal{M} \rhd \langle\overline{\eta}\rangle\, \Gamma$ iff, for every $1 \leq j \leq n$, $\mathcal{M} \rhd \langle\eta_j\rangle\, K_j$.

We introduce the constructive consequence relation.

Definition 2 (Constructive consequence) Let $\Gamma \cup \{K\}$ be a set of closed formulas of $\mathcal{L}$ and let $\mathcal{N}$ be a finite set of individual names such that $\Gamma \cup \{K\} \subseteq \mathcal{L}_{\mathcal{N}}$. We say that *K is a constructive consequence of* Γ, and we write $\Gamma \models_{\mathrm{c}} K$, iff, for every $\overline{\gamma} \in \textsc{it}_{\mathcal{N}}(\Gamma)$, there exists $\eta \in \textsc{it}_{\mathcal{N}}(K)$ such that, for every model $\mathcal{M}$ for $\mathcal{L}_{\mathcal{N}}$, $\mathcal{M} \rhd \langle\overline{\gamma}\rangle\, \Gamma$ implies $\mathcal{M} \rhd \langle\eta\rangle\, K$.

Thus, the relation $\Gamma \models_{\mathrm{c}} K$ implicitly defines a map $\Phi_{\mathcal{N}}$ from $\textsc{it}_{\mathcal{N}}(\Gamma)$ to $\textsc{it}_{\mathcal{N}}(K)$ such that, for every model $\mathcal{M}$, $\mathcal{M} \rhd \langle\overline{\gamma}\rangle\, \Gamma$ implies $\mathcal{M} \rhd \langle\Phi_{\mathcal{N}}(\overline{\gamma})\rangle\, K$. The key point is that $\Phi_{\mathcal{N}}$ is independent of the choice of the models.

Example 2 Let us consider the ABox $\mathcal{A}$ and the TBox $\mathcal{T}$ defined in Example 1. We recall that $\mathcal{W}$ is the set of all the individual names occurring in $\mathcal{A}$. An element of $\textsc{it}_{\mathcal{W}}(Ax_1)$ is a function mapping each $f \in \mathrm{dom}(\mathtt{FOOD})$ to an element $\delta \in \textsc{it}_{\mathcal{W}}(f : \exists\mathtt{goesWith}.\mathtt{COLOR})$, where $\delta = (c, \mathtt{tt})$ (intuitively, c is a wine color which goes with food f). For instance, let us consider the following $\psi_1 \in \textsc{it}_{\mathcal{W}}(Ax_1)$, where we denote with $f \mapsto \psi_1(f)$ the pairs belonging to the function ψ_1:

$$[\, \mathtt{fish} \mapsto (\mathtt{white,tt})\,,\ \mathtt{meat} \mapsto (\mathtt{red,tt})\,]$$

Let $\mathcal{M}$ be a model of $\mathcal{A}$. One can easily check that $\mathcal{M} \rhd \langle\psi_1\rangle\, Ax_1$. Similarly, if $\psi_2 \in \textsc{it}_{\mathcal{W}}(Ax_2)$ is the information term

$$[\, \mathtt{red} \mapsto (\mathtt{barolo,tt})\,,\ \mathtt{white} \mapsto (\mathtt{chardonnay,tt})\,]$$

then $\mathcal{M} \rhd \langle\psi_2\rangle\, Ax_2$ as well. We conclude $\mathcal{M} \rhd \langle(\psi_1, \psi_2)\rangle\, \mathcal{T}$. We can prove that

$$\mathcal{T} \models_{\mathrm{c}} \forall_{\mathtt{FOOD}}\exists\mathtt{goesWith}.(\mathtt{COLOR} \sqcap \exists\mathtt{isColorOf}.\mathtt{WINE}) \tag{1}$$

Indeed, let $\mathcal{N}$ be such that

$$\mathcal{T} \cup \{\, \forall_{\mathtt{FOOD}}\exists\mathtt{goesWith}.(\mathtt{COLOR} \sqcap \exists\mathtt{isColorOf}.\mathtt{WINE})\,\} \subseteq \mathcal{L}_{\mathcal{N}}$$

and let $(\phi_1, \phi_2) \in \textsc{it}_{\mathcal{N}}(\mathcal{T})$. We define

$$\psi : \mathrm{dom}(\mathtt{FOOD}) \to \bigcup_{f \in \mathrm{dom}(\mathtt{FOOD})} \textsc{it}_{\mathcal{N}}(f : \exists\mathtt{goesWith}.(\mathtt{COLOR} \sqcap \exists\mathtt{isColorOf}.\mathtt{WINE}))$$

as follows: for every $f \in \mathrm{dom}(\mathtt{FOOD})$, $\psi(f) = (c, (\mathtt{tt}, (w, \mathtt{tt})))$ where c and w satisfy $\phi_1(f) = (c, \mathtt{tt})$, $\phi_2(c) = (w, \mathtt{tt})$. One can check that, for every model $\mathcal{M}$ for $\mathcal{L}_{\mathcal{N}}$, $\mathcal{M} \rhd \langle(\phi_1, \phi_2)\rangle\, \mathcal{T}$ implies

$$\mathcal{M} \rhd \langle\psi\rangle\, \forall_{\mathtt{FOOD}}\exists\mathtt{goesWith}.(\mathtt{COLOR} \sqcap \exists\mathtt{isColorOf}.\mathtt{WINE})$$

and this proves (1).

On the other hand, $\mathcal{T} \not\models_{c} \texttt{meat} : \texttt{WINE} \sqcup \neg\texttt{WINE}$. Indeed, let us consider the pair $(\psi_1, \psi_2) \in \text{IT}_{\mathcal{W}}(\mathcal{T})$ defined above. There exist models $\mathcal{M}_1$ and $\mathcal{M}_2$ for $\mathcal{L}_{\mathcal{W}}$ such that

$$\mathcal{M}_1 \rhd \langle(\psi_1, \psi_2)\rangle\, \mathcal{T} \qquad \mathcal{M}_2 \rhd \langle(\psi_1, \psi_2)\rangle\, \mathcal{T}$$

$$\mathcal{M}_1 \models \texttt{meat} : \neg\texttt{WINE} \qquad \mathcal{M}_2 \models \texttt{meat} : \texttt{WINE}$$

Thus, there is no $\eta \in \text{IT}_{\mathcal{W}}(\texttt{meat} : \texttt{WINE} \sqcup \neg\texttt{WINE})$ such that *for every* model $\mathcal{M}$ for $\mathcal{L}_{\mathcal{W}}$, $\mathcal{M} \rhd \langle(\psi_1, \psi_2)\rangle\, \mathcal{T}$ implies $\mathcal{M} \rhd \langle\eta\rangle\, \texttt{meat} : \texttt{WINE} \sqcup \neg\texttt{WINE}$. We conclude by noticing that $\mathcal{T} \models \texttt{meat} : \texttt{WINE} \sqcup \neg\texttt{WINE}$. Hence, $\texttt{meat} : \texttt{WINE} \sqcup \neg\texttt{WINE}$ classically holds but it cannot be constructively justified in $\mathcal{T}$.

In the following section we present the calculus $\mathcal{ND}_{\text{c}}$ from which proofs one can extract the map $\Phi_{\mathcal{N}}$ involved in the definition of $\Gamma \models_{c} K$.

5 The Natural Calculus $\mathcal{ND}_{\text{c}}$ for $\mathcal{BCDL}$

In this section we introduce the calculus $\mathcal{ND}_{\text{c}}$ for the logic $\mathcal{BCDL}$ and we prove it is sound and complete w.r.t. the constructive consequence relation of Definition 2.

The calculus $\mathcal{ND}_{\text{c}}$ is obtained by adding to the rules in Tables 1 and 2 the following rules:

$$\frac{\begin{array}{c}\Gamma\\ \vdots\,\pi'\\ t : \neg\neg C\end{array}}{t : C}\,\text{At} \quad C \in \text{NC} \cup \text{NG} \qquad\qquad \frac{\begin{array}{c}\Gamma\\ \vdots\,\pi'\\ t : \forall R.\neg\neg H\end{array}}{t : \neg\neg\forall R.H}\,\text{KUR}$$

The rule At is double negation elimination for atomic concepts and generators. It corresponds to the rule of double negation elimination on atomic formulas in the first order setting. The addition of this principle to Intuitionistic first order logic **Int** gives rise to a system still satisfying the *disjunction property* (if $A \vee B$ is provable, then either A or B is provable) and the *explicit definability property* (if $\exists x A(x)$ is provable then also $A(t)$ is provable for some term t) [12, 13]. Instead, the calculus obtained by adding to $\mathcal{ND}_{\text{c}}$ the unrestricted version of At, where C is any concept, is equivalent to the calculus $\mathcal{ND}$ for $\mathcal{ALCG}$.

As for the rule KUR, it corresponds to the first order axiom schema

$$\forall x.\neg\neg A(x) \rightarrow \neg\neg\forall x.A(x)$$

which is well-known in the literature of constructive logics [8, 21]. Indeed, adding this schema to **Int**, we get a proper extension of **Int**, called *Kuroda Logic*, that satisfies the disjunction property and the explicit definability property. An important property of Kuroda Logic is that a theory T is classically consistent iff it is consistent w.r.t. Kuroda Logic. In our setting the rule KUR has a similar role, indeed it allows to prove that there is a proof $\pi : \Gamma \vdash \bot$ in $\mathcal{ND}_{\text{c}}$ iff $\Gamma \models_{\mathcal{ALCG}} \bot$ (see Corollary 1 below).

To avoid inessential technicalities and improve the readability of the paper, we introduce the following assumption on the individual names occurring in a proof of $\mathcal{ND}_{\text{c}}$. Let $\pi : \Gamma \vdash K$ be a proof of $\mathcal{ND}_{\text{c}}$ and let $\pi' : \Gamma' \vdash K'$ a subproof of π.

– We say that π' is *simple* iff K' is a simple formula.

– Let r be the last rule applied to π'. We say that r *is relevant in* π iff π' is not a subproof of a simple subproof of π.

Assumption on individual names Let $\pi : \Gamma \vdash K$ be a proof of $\mathcal{ND}_c$, let $\mathcal{N}$ be a finite set of individual names such that $\Gamma \cup \{K\} \subseteq \mathcal{L}_{\mathcal{N}}$, let π' be a subproof of π and let r be the last rule applied in π'. We assume that:

– If r is the rule $\sqcup E$ of Table 1 with major premise $t : A_1 \sqcup A_2$ and r is relevant in π, then $t \in \mathtt{Var} \cup \mathcal{N}$.
– If r is the rule $\exists E$ in Table 1 with major premise $t : \exists R.A$ and r is relevant in π, then $t \in \mathtt{Var} \cup \mathcal{N}$.

We write $\Gamma \vdash_{\overline{\mathcal{BCDL}}} K$ to mean that there exists a proof $\pi : \Gamma \vdash K$ of $\mathcal{ND}_c$ satisfying the above assumption.

First of all, we notice that the rules At and KUR are derivable in $\mathcal{ND}$, hence:

Theorem 4 $\Gamma \vdash_{\overline{\mathcal{BCDL}}} K$ *implies* $\Gamma \vdash_{\overline{\mathcal{ALCG}}} K$.

Another interesting relation between $\mathcal{ND}$ and $\mathcal{ND}_c$ is that they prove the same set of simple formulas. To prove this, let us define the following map on concepts and formulas:

– $H^{\neg\neg} = H$ if $H \in \mathtt{NC} \cup \mathtt{NG}$.
– $H^{\neg\neg} = \neg\neg H$ if H is a concept such that $H \notin \mathtt{NC} \cup \mathtt{NG}$.
– $\bot^{\neg\neg} = \bot$.
– $((s,t) : R)^{\neg\neg} = (s,t) : R$.
– $(t : H)^{\neg\neg} = t : H^{\neg\neg}$.
– $(\forall_G H)^{\neg\neg} = \forall_G H^{\neg\neg}$.

It is easy to prove by induction on the depth of $\pi : \Gamma \vdash K$ the following result:

Theorem 5 $\Gamma \vdash_{\overline{\mathcal{ALCG}}} K$ *implies* $\Gamma \vdash_{\overline{\mathcal{BCDL}}} K^{\neg\neg}$.

As a corollary we get:

Corollary 1 *For every simple formula* K *of* $\mathcal{L}$, $\Gamma \vdash_{\overline{\mathcal{BCDL}}} K$ *iff* $\Gamma \vdash_{\overline{\mathcal{ALCG}}} K$.

The following is an admissible rule of $\mathcal{ND}_c$ allowing us to prove $\forall_G A$ using one "generic proof" instead of as many proofs as the elements of dom(G).

$$\dfrac{\begin{array}{c}\Gamma, [p : G] \\ \vdots\ \pi' \\ p : A\end{array}}{\forall_G A}\ \forall_G I' \qquad \text{where } p \in \mathtt{Var} \text{ and } p \text{ does not occur in } \Gamma$$

The admissibility follows from the fact that by instantiating the parameter p of π' and by applying the rule AxGen, we get the proofs $\pi'_k : \Gamma \vdash c_k : A$, for every $c_k \in$ dom(G). By applying the rule $\forall_G I$ to these proofs, we get a proof $\pi : \Gamma \vdash \forall_G A$.

5.1 Soundness of $\mathcal{ND}_c$

We prove the soundness of $\mathcal{ND}_c$, namely $\Gamma \vdash_{\overline{\mathcal{BCDL}}} K$ implies $\Gamma \models_c K$. First of all, from Theorem 1 and the fact that the rules At and KUR are derivable in $\mathcal{ND}$ we get that $\mathcal{ND}_c$ preserves validity of formulas. More precisely:

(P1) Let $\pi : \Gamma \vdash K$ be a proof of $\mathcal{ND}_c$. For every model $\mathcal{M}$ for $\mathcal{L}_\mathcal{N}$ and every assignment θ, $\mathcal{M}, \theta \models \Gamma$ implies $\mathcal{M}, \theta \models K$.

As a consequence, $\pi : \Gamma \vdash K$ implies $\Gamma \models K$.

Let $\mathcal{N}$ be a finite subset of $\mathtt{NI}$, an $\mathcal{N}$*-substitution* σ is a map $\sigma : \mathtt{Var} \to \mathcal{N}$. We extend σ to $\mathcal{L}_\mathcal{N}$ as usual:

- If $c \in \mathcal{N}$, $\sigma c = c$;
- For a formula K of $\mathcal{L}_\mathcal{N}$, σK denotes the closed formula of $\mathcal{L}_\mathcal{N}$ obtained by replacing every variable x occurring in K with $\sigma(x)$;
- If Γ is a set of formulas, $\sigma\Gamma$ is the set of σK such that $K \in \Gamma$.

If $c \in \mathcal{N}$, $\sigma[c/p]$ is the $\mathcal{N}$-substitution σ' such that $\sigma'(p) = c$ and $\sigma'(x) = \sigma(x)$ for $x \neq p$.

Let $\pi : \Gamma \vdash K$ be a proof of $\mathcal{ND}_c$, let $\mathcal{N}$ be a finite set of individual names such that $\Gamma \cup \{K\} \subseteq \mathcal{L}_\mathcal{N}$ and let σ be a $\mathcal{N}$-substitution. We define a computable function

$$\Phi^\pi_{\sigma,\mathcal{N}} : \mathrm{IT}_\mathcal{N}(\sigma\Gamma) \to \mathrm{IT}_\mathcal{N}(\sigma K)$$

that will provide the computational interpretation of π. $\Phi^\pi_{\sigma,\mathcal{N}}$ is defined, by induction on the depth of π, in order to fulfill the following property:

(P2) For every model $\mathcal{M}$ of $\mathcal{L}_\mathcal{N}$ and for every $\overline{\gamma} \in \mathrm{IT}_\mathcal{N}(\sigma\Gamma)$, $\mathcal{M} \rhd \langle\overline{\gamma}\rangle\, \sigma\Gamma$ implies $\mathcal{M} \rhd \langle\Phi^\pi_{\sigma,\mathcal{N}}(\overline{\gamma})\rangle\, \sigma K$.

If π only consists of the introduction of an assumption K, then $\Phi^\pi_{\sigma,\mathcal{N}}$ is the identity function on $\mathrm{IT}_\mathcal{N}(\sigma K)$. If K is a simple formula, then $\Phi^\pi_{\sigma,\mathcal{N}}(\overline{\gamma}) = \mathtt{tt}$. Otherwise, let r be the last rule applied in π. By definition, r is a relevant rule in π.

(1) $r = \bot E$. Then, $\Phi^\pi_{\sigma,\mathcal{N}} : \mathrm{IT}_\mathcal{N}(\sigma\Gamma) \to \mathrm{IT}_\mathcal{N}(\sigma K)$ and $\Phi^\pi_{\sigma,\mathcal{N}}(\overline{\gamma}) = \eta^+$, where η^+ is an element of $\mathrm{IT}_\mathcal{N}(K)$.

(2) $r = \sqcap I$. Then, $\Phi^\pi_{\sigma,\mathcal{N}} : \mathrm{IT}_\mathcal{N}(\sigma\Gamma_1) \times \mathrm{IT}_\mathcal{N}(\sigma\Gamma_2) \to \mathrm{IT}_\mathcal{N}(\sigma t : A \sqcap B)$ and

$$\Phi^\pi_{\sigma,\mathcal{N}}\left(\overline{\gamma}_1, \overline{\gamma}_2\right) = \left(\Phi^{\pi_1}_{\sigma,\mathcal{N}}(\overline{\gamma}_1),\ \Phi^{\pi_2}_{\sigma,\mathcal{N}}(\overline{\gamma}_2)\right)$$

(3) $r = \sqcap E_k$ ($k \in \{1, 2\}$). Then, $\Phi^\pi_{\sigma,\mathcal{N}} : \mathrm{IT}_\mathcal{N}(\sigma\Gamma) \to \mathrm{IT}_\mathcal{N}(\sigma t : A_k)$ and

$$\Phi^\pi_{\sigma,\mathcal{N}}(\overline{\gamma}) = Pro_k\left(\Phi^{\pi'}_{\sigma,\mathcal{N}}(\overline{\gamma})\right)$$

where Pro_k is the k-projection function.

(4) $r = \sqcup I_k$ ($k \in \{1, 2\}$). Then, $\Phi^\pi_{\sigma,\mathcal{N}} : \mathrm{IT}_\mathcal{N}(\sigma\Gamma) \to \mathrm{IT}_\mathcal{N}(\sigma t : A_1 \sqcup A_2)$ and

$$\Phi^\pi_{\sigma,\mathcal{N}}(\overline{\gamma}) = \left(k,\ \Phi^{\pi'}_{\sigma,\mathcal{N}}(\overline{\gamma})\right)$$

(5) $r = \sqcup E$. Then, $\Phi^\pi_{\sigma,\mathcal{N}} : \mathrm{IT}_\mathcal{N}(\sigma\Gamma_1) \times \mathrm{IT}_\mathcal{N}(\sigma\Gamma_2) \times \mathrm{IT}_\mathcal{N}(\sigma\Gamma_3) \to \mathrm{IT}_\mathcal{N}(\sigma K)$. Since $\sqcup E$ is relevant, by the assumptions on the individual names, $\sigma t \in \mathcal{N}$, thus $\Phi^{\pi_1}_{\sigma,\mathcal{N}}$,

$\Phi^{\pi_2}_{\sigma,\mathcal{N}}$, and $\Phi^{\pi_3}_{\sigma,\mathcal{N}}$ are defined. We set

$$\Phi^{\pi}_{\sigma,\mathcal{N}}\left(\overline{\gamma}_1,\overline{\gamma}_2,\overline{\gamma}_3\right) = \begin{cases} \Phi^{\pi_2}_{\sigma,\mathcal{N}}(\overline{\gamma}_2,\ \alpha) & \text{if } \Phi^{\pi_1}_{\sigma,\mathcal{N}}(\overline{\gamma}_1) = (1,\alpha) \\ \Phi^{\pi_3}_{\sigma,\mathcal{N}}(\overline{\gamma}_3,\ \beta) & \text{if } \Phi^{\pi_1}_{\sigma,\mathcal{N}}(\overline{\gamma}_1) = (2,\beta) \end{cases}$$

(6) $r = \exists I$. Then, $\Phi^{\pi}_{\sigma,\mathcal{N}} : \{\texttt{tt}\} \times \mathrm{IT}_{\mathcal{N}}(\sigma\Gamma') \to \mathrm{IT}_{\mathcal{N}}(\sigma t : \exists R.A)$ and

$$\Phi^{\pi}_{\sigma,\mathcal{N}}\left(\texttt{tt},\overline{\gamma}'\right) = \left(\sigma u,\ \Phi^{\pi'}_{\sigma,\mathcal{N}}(\overline{\gamma}')\right)$$

(7) $r = \exists E$. Then, $\Phi^{\pi}_{\sigma,\mathcal{N}} : \mathrm{IT}_{\mathcal{N}}(\sigma\Gamma_1) \times \mathrm{IT}_{\mathcal{N}}(\sigma\Gamma_2) \to \mathrm{IT}_{\mathcal{N}}(\sigma K)$. By the assumption on individual names, $\sigma t \in \mathcal{N}$. Let $\Phi^{\pi_1}_{\sigma,\mathcal{N}}(\overline{\gamma}_1) = (c,\alpha)$. By the side condition on p, $(\sigma[c/p])\Gamma_2 = \sigma\Gamma_2$ and $(\sigma[c/p])K = \sigma K$. We define

$$\Phi^{\pi}_{\sigma,\mathcal{N}}\left(\overline{\gamma}_1,\overline{\gamma}_2\right) = \Phi^{\pi_2}_{\sigma[c/p],\mathcal{N}}\left(\overline{\gamma}_2,\ \texttt{tt},\ \alpha\right)$$

(8) $r = \forall I$. Then, $\Phi^{\pi}_{\sigma,\mathcal{N}} : \mathrm{IT}_{\mathcal{N}}(\sigma\Gamma) \to \mathrm{IT}_{\mathcal{N}}(\sigma t : \forall R.A)$. Let $c \in \mathcal{N}$. By the side condition on p, $(\sigma[c/p])\Gamma = \sigma\Gamma$ and $(\sigma[c/p])t : \forall R.A = \sigma t : \forall R.A$. Then $\Phi^{\pi}_{\sigma,\mathcal{N}}(\overline{\gamma})$ is the function defined as follows: for every $c \in \mathcal{N}$

$$\left[\Phi^{\pi}_{\sigma,\mathcal{N}}(\overline{\gamma})\right](c) = \Phi^{\pi'}_{\sigma[c/p],\mathcal{N}}(\overline{\gamma},\ \texttt{tt})$$

(9) $r = \forall E$. Then, $\Phi^{\pi}_{\sigma,\mathcal{N}} : \{\texttt{tt}\} \times \mathrm{IT}_{\mathcal{N}}(\sigma\Gamma') \to \mathrm{IT}_{\mathcal{N}}(\sigma t : A)$ and

$$\Phi^{\pi}_{\sigma,\mathcal{N}}\left(\texttt{tt},\overline{\gamma}'\right) = \left[\Phi^{\pi'}_{\sigma,\mathcal{N}}(\overline{\gamma}')\right](\sigma t)$$

(10) $r = \forall_G I$. Let $\mathrm{dom}(G) = \{c_1,\ldots,c_n\}$. Then, $\Phi^{\pi}_{\sigma,\mathcal{N}} : \mathrm{IT}_{\mathcal{N}}(\sigma\Gamma_1) \times \cdots \times (\sigma\Gamma_n) \to \mathrm{IT}_{\mathcal{N}}(\forall_G A)$ is the function such that, for every $1 \le k \le n$,

$$\left[\Phi^{\pi}_{\sigma,\mathcal{N}}\left(\overline{\gamma}_1,\ldots,\overline{\gamma}_n\right)\right](c_k) = \Phi^{\pi_k}_{\sigma,\mathcal{N}}(\overline{\gamma}_k)$$

(11) $r = \forall_G E$. Analogous to the case $r = \forall E$.

One can easily check that $\Phi^{\pi}_{\sigma,\mathcal{N}}$ is a well-defined function and that (P2) holds.

Note that in the definition of $\Phi^{\pi}_{\sigma,\mathcal{N}}$ the relevant rules of π are used to build up "relevant information terms", namely information terms different from tt. For instance, in the case of the rule $\sqcap I$, the map takes the information terms α and β produced by the subproofs π_1 and π_2 and builds the pair (α,β). On the other hand, a simple subproof $\pi' : \Gamma' \vdash K'$ of π does not convey any relevant information apart from the fact that $\Gamma' \models K'$.

Let $\Phi^{\pi}_{\mathcal{N}} = \Phi^{\pi}_{\sigma,\mathcal{N}}$, where σ is any $\mathcal{N}$-substitution. By $(P1)$ and $(P2)$, we get:

Theorem 6 *Let $\pi : \Gamma \vdash K$ be a proof of $\mathcal{ND}_c$ such that the formulas in $\Gamma \cup \{K\}$ are closed, and let $\mathcal{N}$ be a finite set of individual names such that $\Gamma \cup \{K\} \subseteq \mathcal{L}_{\mathcal{N}}$. Then:*

(i) $\Gamma \models K$.

(ii) *For every $\overline{\gamma} \in \mathrm{IT}_{\mathcal{N}}(\Gamma)$ and for every model $\mathcal{M}$ for $\mathcal{L}_{\mathcal{N}}$, $\mathcal{M} \rhd \langle\overline{\gamma}\rangle\, \Gamma$ implies $\mathcal{M} \rhd \langle\Phi^{\pi}_{\mathcal{N}}(\overline{\gamma})\rangle\, K$.*

As a consequence, we get:

Theorem 7 (Soundness) $\Gamma \vdash_{\mathcal{BCDL}} K$ *implies* $\Gamma \models_{c} K$.

Now we give an example of the information one can extract from a proof using the map $\Phi^{\pi}_{\mathcal{N}}$.

Example 3 Let us consider the knowledge base of Example 1. We can build a proof

$$\pi : \mathcal{T} \vdash \forall_{\mathtt{FOOD}}\exists\mathtt{goesWith}.(\mathtt{COLOR} \sqcap \exists\mathtt{isColorOf}.\mathtt{WINE})$$

in $\mathcal{ND}_{\mathrm{c}}$. The proof π is

$$\dfrac{\dfrac{\dfrac{Ax_1 \quad [y:\mathtt{FOOD}]^2}{y:\exists\mathtt{goesWith}.\mathtt{COLOR}}\,\forall_{\mathtt{FOOD}}\mathrm{E} \qquad \dfrac{[(y,z):\mathtt{goesWith}]^1 \quad \dfrac{[z:\mathtt{COLOR}]^1 \quad \dfrac{Ax_2 \quad [z:\mathtt{COLOR}]^1}{z:\exists\mathtt{isColorOf}.\mathtt{Wine}}\,\forall_{\mathtt{COLOR}}\mathrm{E}}{z:\mathtt{COLOR}\sqcap\exists\mathtt{isColorOf}.\mathtt{WINE}}\,\sqcap\mathrm{I}}{y:\exists\mathtt{goesWith}.(\mathtt{COLOR}\sqcap\exists\mathtt{isColorOf}.\mathtt{WINE})}\,\exists\mathrm{I}}{y:\exists\mathtt{goesWith}.(\mathtt{COLOR}\sqcap\exists\mathtt{isColorOf}.\mathtt{WINE})}\,\exists\mathrm{E}\,[1]}{\forall_{\mathtt{FOOD}}\exists\mathtt{goesWith}.(\mathtt{COLOR}\sqcap\exists\mathtt{isColorOf}.\mathtt{WINE})}\,\forall_{\mathtt{FOOD}}\mathrm{I}'\,[2]$$

Note that the assumption on individual names is satisfied since we do not use individual names. Let us consider, for instance, the subproof

$$\pi' : \{(y,z):\mathtt{goesWith},\, z:\mathtt{COLOR},\, Ax_2\} \vdash y : \exists\mathtt{goesWith}.(\mathtt{COLOR} \sqcap \exists\mathtt{isColorOf}.\mathtt{WINE})$$

Let $\phi_2 \in \mathrm{IT}_{\mathcal{W}}(Ax_2)$ and let σ be any $\mathcal{W}$-substitution. By the above definition, we get:

$$\Phi^{\pi'}_{\sigma,\mathcal{W}}(\mathtt{tt},\mathtt{tt},\phi_2) \;=\; (\sigma z,\ (\mathtt{tt},\ (\phi_2(\sigma z),\mathtt{tt})))$$

The map $\Phi^{\pi}_{\mathcal{W}}$ is defined as in Example 2.

To conclude this section we remark that, along the lines of the previous example, Theorem 7 allows us to interpret a proof of a "goal" as a program to solve it. More in details, let us suppose to have an "open" proof

$$\begin{array}{c}\Gamma\\ \vdots\,\pi\\ p : A \sqcup \neg A\end{array} \qquad p \in \mathtt{Var}$$

in $\mathcal{ND}_{\mathrm{c}}$ and let $\mathcal{N}$ be a finite set of individual names containing all the individual names occurring in π. Then, for every $c \in \mathtt{NI}$, the map extracted from π allows us to decide the membership of c in A w.r.t. the models of Γ. Formally, let $c \in \mathtt{NI}$, let $\mathcal{N}' = \mathcal{N} \cup \{c\}$ and let σ be a $\mathcal{N}'$-substitution such that $\sigma p = c$. We can associate with every $\overline{\gamma} \in \mathrm{IT}_{\mathcal{N}'}(\sigma\Gamma)$ the class $\mathcal{I}_{\overline{\gamma}}$ of models $\mathcal{M}$ such that $\mathcal{M} \rhd \langle\overline{\gamma}\rangle\,\sigma\Gamma$. Moreover, for every $\overline{\gamma} \in \mathrm{IT}_{\mathcal{N}'}(\sigma\Gamma)$, $\Phi^{\pi}_{\sigma,\mathcal{N}'}(\overline{\gamma})$ provides an information term $(k,\eta) \in \mathrm{IT}_{\mathcal{N}'}(c : A \sqcup \neg A)$, with $k \in \{1,2\}$. By Theorem 6, for all models $\mathcal{M} \in \mathcal{I}_{\overline{\gamma}}$, $\mathcal{M} \models c : A$ if $k = 1$ and $\mathcal{M} \models c : \neg A$ if $k = 2$.

Analogously, the map extracted from a proof $\pi : \Gamma \vdash p : \exists R.A$ in $\mathcal{ND}_{\mathrm{c}}$ allows us to compute, for every $c \in \mathtt{NI}$, an R-successor d of c such that d belongs to A w.r.t. the models of Γ.

5.2 Completeness of $\mathcal{ND}_c$

The proof of completeness consists of two steps. First, we introduce a quite standard notion of saturated set and we show that every consistent set of formulas can be extended in a saturated set. Differently from standard completeness proofs, saturated sets do not contain enough information to build up a counter model based on the information term semantics. Hence we need to rely on another construction based on the notion of c-tree.

Definition 3 (Saturated set) Let $\mathcal{N}$ be a set of individual names and let Δ be a set of closed formulas of $\mathcal{L}_{\mathcal{N}}$. Δ is *saturated* in $\mathcal{N}$ iff the following conditions hold:

(1) $c : A \sqcap B \in \Delta$ implies $c : A \in \Delta$ and $c : B \in \Delta$.
(2) $c : A \sqcup B \in \Delta$ implies $c : A \in \Delta$ or $c : B \in \Delta$.
(3) $c : \exists R.A \in \Delta$ implies that there exists $d \in \mathcal{N}$ such that $(c, d) : R \in \Delta$ and $d : A \in \Delta$.
(4) $c : \forall R.A \in \Delta$ and $(c, d) : R \in \Delta$ imply $d : A \in \Delta$.
(5) $\forall_G A \in \Delta$ implies:
  (i) $c : G \in \Delta$ iff $c \in \text{dom}(G)$;
  (ii) for every $c \in \text{dom}(G)$, $c : A \in \Delta$.

Let Γ be a set of formulas of $\mathcal{L}_{\mathcal{N}}$ and let $\mathcal{N}' \subseteq \mathcal{N}$. By $\Gamma_{/\mathcal{N}'}$ we denote the *restriction* of Γ to $\mathcal{L}_{\mathcal{N}'}$, i.e., $\Gamma_{/\mathcal{N}'} = \Gamma \cap \mathcal{L}_{\mathcal{N}'}$. Following the lines of Lemma 1 one can prove:

Lemma 3 *Let $\mathcal{N}$ be a finite set of individual names, let Γ be a finite set of closed formulas of $\mathcal{L}_{\mathcal{N}}$, let K be a closed formula of $\mathcal{L}_{\mathcal{N}}$ such that $\Gamma \not\vdash_{\mathcal{BCDL}} K$. There exist a finite set of individual names $\overline{\mathcal{N}} \supseteq \mathcal{N}$ and a finite set $\overline{\Delta} \supseteq \Gamma$ of closed formulas of $\mathcal{L}_{\overline{\mathcal{N}}}$ such that:*

(1) *$\overline{\Delta}$ is saturated in $\overline{\mathcal{N}}$.*
(2) *$\overline{\Delta} \not\vdash_{\mathcal{BCDL}} K$.*
(3) *If $\Delta' \subseteq \Gamma$ is saturated in $\mathcal{N}' \subseteq \mathcal{N}$ and $\Gamma_{/\mathcal{N}'} = \Delta'$, then $\overline{\Delta}_{/\mathcal{N}'} = \Delta'$.*

Given Δ and K such that $\Delta \not\vdash_{\mathcal{BCDL}} K$, we can build a tree $\mathcal{T}$, we call a *c-tree* for (Δ, K). From $\mathcal{T}$, we can define a finite set $\mathcal{N} \subseteq \texttt{NI}$ and a "canonical" information term $\delta \in \text{IT}_{\mathcal{N}}(\Delta)$ (denoted by $\mathcal{I}_{\mathcal{T}}(\Delta)$) with the following property:

– for every $\eta \in \text{IT}_{\mathcal{N}}(K)$, there exists a model $\mathcal{M}$ of $\mathcal{L}_{\mathcal{N}}$ such that $\mathcal{M} \rhd \langle\delta\rangle\, \Delta$ and $\mathcal{M} \not\rhd \langle\eta\rangle\, K$.

This means that $\Delta \not\models_{c} K$. This is the original part of the proof, which requires techniques different from the standard ones.

We say that a triple $(\Delta, K, \mathcal{N})$ is a *c-node* iff:

(i) $\mathcal{N}$ is a finite set of individual names;
(ii) K is a closed formula of $\mathcal{L}_{\mathcal{N}}$;
(iii) Δ is a finite set of closed formulas of $\mathcal{L}_{\mathcal{N}}$ such that Δ is saturated in $\mathcal{N}$.

We introduce the notion of c-tree.

Definition 4 (c-tree) Let Γ_0 be a finite set of closed formulas and K_0 a closed formula. We say that $\mathcal{T}$ is a *c-tree* for (Γ_0, K_0) iff the following conditions hold:

(C1) The *root* of $\mathcal{T}$ is a c-node $(\Delta_0, K_0, \mathcal{N}_0)$, such that $\Gamma_0 \subseteq \Delta_0$.

(C2) Let $(\Delta, K, \mathcal{N})$ be a node of $\mathcal{T}$ and let $(\Delta_1, K_1, \mathcal{N}_1), \ldots, (\Delta_n, K_n, \mathcal{N}_n)$ be the *immediate successors* of $(\Delta, K, \mathcal{N})$ in $\mathcal{T}$. Then

$$\frac{(\Delta_1, K_1, \mathcal{N}_1) \quad \cdots \quad (\Delta_n, K_n, \mathcal{N}_n)}{(\Delta, K, \mathcal{N})}\ \mathrm{r}$$

is an instance of a rule of Table 3.

(C3) If $(\Delta, K, \mathcal{N})$ is a *leaf* of $\mathcal{T}$ (namely, $(\Delta, K, \mathcal{N})$ has no immediate successors), then $\Delta \not\vdash_{\mathcal{ALCG}} K$.

Clearly, a c-tree $\mathcal{T}$ for (Γ_0, K_0), if exists, is finite and every node $(\Delta, K, \mathcal{N})$ of $\mathcal{T}$ is a c-node.

Table 3 Rules to build a c-tree

$$\frac{(\Delta,\ c : A,\ \mathcal{N})}{(\Delta,\ c : A \sqcap B,\ \mathcal{N})}\ r_{\sqcap_1} \qquad \frac{(\Delta,\ c : B,\ \mathcal{N})}{(\Delta,\ c : A \sqcap B,\ \mathcal{N})}\ r_{\sqcap_2}$$

$$\frac{(\Delta,\ c : A,\ \mathcal{N}) \qquad (\Delta,\ c : B,\ \mathcal{N})}{(\Delta,\ c : A \sqcup B,\ \mathcal{N})}\ r_{\sqcup}$$

$$\frac{(\Delta,\ d_1 : A,\ \mathcal{N}) \quad \cdots \quad (\Delta,\ d_n : A,\ \mathcal{N})}{(\Delta,\ c : \exists R.A,\ \mathcal{N})}\ r_{\exists R_1} \quad \begin{array}{l}\{d_1, \ldots d_n\} = \\ \{d \mid (c, d) : R \in \Delta\}\ (n \geq 1)\end{array}$$

$$\frac{(\Delta,\ \bot,\ \mathcal{N})}{(\Delta,\ c : \exists R.A,\ \mathcal{N})}\ r_{\exists R_2} \quad \text{if, for all } d \in \mathcal{N}, (c, d) : R \notin \Delta$$

$$\frac{(\Delta',\ d : A,\ \mathcal{N}')}{(\Delta,\ c : \forall R.A,\ \mathcal{N})}\ r_{\forall_R}$$ where $(\Delta', d : A, \mathcal{N}')$ is a c-node such that

1. $\mathcal{N} \subseteq \mathcal{N}'$ and $d \in \mathcal{N}' \setminus \mathcal{N}$;
2. $\Delta \cup \{(c, d) : R\} \subseteq \Delta'$;
3. $\Delta'_{/\mathcal{N}} = \Delta$.

$$\frac{(\Delta,\ d : A,\ \mathcal{N})}{(\Delta,\ \forall_G A,\ \mathcal{N})}\ r_{\forall_G} \quad \text{where } d \in \mathrm{dom}(G)$$

Let $\mathcal{T}$ be a c-tree and let $\preceq$ denote the reflexive and transitive closure of the immediate successor relation between the nodes of $\mathcal{T}$. Given two nodes $(\Delta_1, K_1, \mathcal{N}_1)$ and $(\Delta_2, K_2, \mathcal{N}_2)$ of $\mathcal{T}$, the *greatest lower bound (g.l.b.)* of $(\Delta_1, K_1, \mathcal{N}_1)$ and $(\Delta_2, K_2, \mathcal{N}_2)$ is the node $(\Delta, K, \mathcal{N})$ of maximal depth among the nodes $(\Delta, K, \mathcal{N})$ satisfying both the conditions $(\Delta, K, \mathcal{N}) \preceq (\Delta_1, K_1, \mathcal{N}_1)$ and $(\Delta, K, \mathcal{N}) \preceq (\Delta_2, K_2, \mathcal{N}_2)$. We note that, if we rename the individual names $c \notin \mathcal{N}_0$ occurring in $\mathcal{T}$, we again obtain a c-tree for (Γ_0, K_0). Thus, without loss of generality, we assume that the following property holds:

(C4) Let $(\Delta_1, K_1, \mathcal{N}_1)$ and $(\Delta_2, K_2, \mathcal{N}_2)$ be two nodes of a c-tree and let $(\Delta, K, \mathcal{N})$ the g.l.b. of $(\Delta_1, K_1, \mathcal{N}_1)$ and $(\Delta_2, K_2, \mathcal{N}_2)$. Then, $\mathcal{N} = \mathcal{N}_1 \cap \mathcal{N}_2$.

Inspecting the rules in Table 3, using Property (C4), one can easily prove:

Lemma 4 *Let $(\Delta_1, K_1, \mathcal{N}_1)$ and $(\Delta_2, K_2, \mathcal{N}_2)$ be two nodes of a c-tree.*

(i) $(\Delta_1, K_1, \mathcal{N}_1) \preceq (\Delta_2, K_2, \mathcal{N}_2)$ *implies* $(\Delta_2)_{/\mathcal{N}_1} = \Delta_1$.
(ii) *Let $(\Delta, K, \mathcal{N})$ be the g.l.b. of $(\Delta_1, K_1, \mathcal{N}_1)$ and $(\Delta_2, K_2, \mathcal{N}_2)$. Then, $\Delta = \Delta_1 \cap \Delta_2$.*

We give an example of c-tree.

Example 4 Let

$$H = (\forall R.\exists S.A) \sqcup (\forall R.\exists S.B) \qquad \Delta_0 = \{\, c : \forall R.\neg\neg\exists S.A,\ c : \forall R.\exists S.(A \sqcup B)\,\}$$

The set Δ_0 is saturated in $\{c\}$. We can build a c-tree $\mathcal{T}$ for $(\Delta_0,\ c : H)$ as follows:

$$\dfrac{\dfrac{\dfrac{(\Delta_1,\ d_2 : A,\ \{c, d_1, d_2\})}{(\Delta_1,\ d_1 : \exists S.A,\ \{c, d_1, d_2\})}\,r_{\exists S_1}}{(\Delta_0,\ c : \forall R.\exists S.A,\ \{c\})}\,r_{\forall R} \qquad \dfrac{\dfrac{(\Delta_2,\ d_4 : B,\ \{c, d_3, d_4\})}{(\Delta_2,\ d_3 : \exists S.B,\ \{c, d_3, d_4\})}\,r_{\exists S_1}}{(\Delta_0,\ c : \forall R.\exists S.B,\ \{c\})}\,r_{\forall R}}{(\Delta_0,\ c : (\forall R.\exists S.A) \sqcup (\forall R.\exists S.B),\ \{c\})}\,r\sqcup$$

where

$$\Delta_1 = \Delta_0 \cup \{\, (c, d_1) : R,\ d_1 : \neg\neg\exists S.A,\ d_1 : \exists S.(A \sqcup B),\ (d_1, d_2) : S,\ d_2 : A \sqcup B,\ d_2 : B \,\}$$
$$\Delta_2 = \Delta_0 \cup \{\, (c, d_3) : R,\ d_3 : \neg\neg\exists S.A,\ d_3 : \exists S.(A \sqcup B),\ (d_3, d_4) : S,\ d_4 : A \sqcup B,\ d_4 : A \,\}$$

On the other hand, we are not able to build a c-tree for $(\Delta', c : H)$ where $\Delta' = \Delta_0 \cup \{c : \forall R.\exists S.A\}$. Indeed, such a c-tree should contain a leaf $(\Delta_l, d : A, \mathcal{N}_l)$ such that, for some $d' \in \mathcal{N}_l$, the formulas $(c, d') : R$, $(d', d) : S$ and $d : A$ belong to Δ_l. This implies $\Delta_l \vdash_{\mathcal{ALCG}} d : A$, in contradiction with the definition of c-tree. Note that $\Delta' \vdash_{\mathcal{BCDL}} c : H$.

We state a sufficient condition for the existence of a c-tree.

Lemma 5 *Let $(\Delta, K, \mathcal{N})$ be a c-node such that $\Delta \nvdash_{\mathcal{BCDL}} K$. Then, there exists a c-tree having root $(\Delta, K, \mathcal{N})$.*

Proof We prove the lemma by induction on K. If K is a simple formula, by Corollary 1 we have $\Delta \not\vdash_{\mathcal{ALCG}} K$, hence $(\Delta, K, \mathcal{N})$ is a c-tree. Let $K \equiv c : A \sqcap B$ and suppose $\Delta \not\vdash_{\mathcal{BCDL}} c : A$ (the case $\Delta \not\vdash_{\mathcal{BCDL}} c : B$ is similar). By induction hypothesis, there exists a c-tree $\mathcal{T}_A$ with root the c-node $(\Delta, c : A, \mathcal{N})$. It follows that

$$\dfrac{\begin{array}{c}\vdots\, \mathcal{T}_A \\ (\Delta,\ c : A,\ \mathcal{N})\end{array}}{(\Delta,\ c : A \sqcap B,\ \mathcal{N})}\ r_{\sqcap_1}$$

is a c-tree with root $(\Delta, c : A \sqcap B, \mathcal{N})$. If $K \equiv c : A \sqcup B$, then $\Delta \not\vdash_{\mathcal{BCDL}} c : A$ and $\Delta \not\vdash_{\mathcal{BCDL}} c : B$. By induction hypothesis, there are two c-trees $\mathcal{T}_A$ and $\mathcal{T}_B$ having roots $(\Delta, c : A, \mathcal{N})$ and $(\Delta, c : B, \mathcal{N})$ respectively. Then (possibly renaming individual names not belonging to $\mathcal{N}$), we can build the c-tree

$$\dfrac{\begin{array}{c}\vdots\, \mathcal{T}_A \\ (\Delta,\ c : A,\ \mathcal{N})\end{array} \qquad \begin{array}{c}\vdots\, \mathcal{T}_B \\ (\Delta,\ c : B,\ \mathcal{N})\end{array}}{(\Delta,\ c : A \sqcup B,\ \mathcal{N})}\ r_{\sqcup}$$

The other cases are similar. For $K \equiv c : \forall R.A$, the existence of Δ' and $\mathcal{N}'$ follows by the fact that $\Delta, (c, d) : R \not\vdash_{\mathcal{BCDL}} d : A$ (where $d \notin \mathcal{N}$) and by Lemma 3. □

As a consequence, we get:

Theorem 8 *If* $\Gamma \not\vdash_{\mathcal{BCDL}} K$*, then there exists a c-tree for* (Γ, K).

Proof By Lemma 3, there exists a c-node $(\Delta, K, \mathcal{N})$ such that $\Delta \not\vdash_{\mathcal{BCDL}} K$. By Lemma 5, there exists a c-tree with root $(\Delta, K, \mathcal{N})$, which is a c-tree for (Γ, K). □

A c-tree $\mathcal{T}$ for (Γ, K) describes the countermodels needed to show that $\Gamma \not\models_{c} K$. Firstly, we define a canonical way to associate an information term with every formula occurring in a set Δ of a node of $\mathcal{T}$. Let

$$\begin{aligned} \mathcal{D}_{\mathcal{T}} &= \bigcup \{\, \Delta \mid (\Delta, K, \mathcal{N}) \text{ is a node of } \mathcal{T} \,\} \\ \mathcal{N}_{\mathcal{T}} &= \bigcup \{\, \mathcal{N} \mid (\Delta, K, \mathcal{N}) \text{ is a node of } \mathcal{T} \,\} \end{aligned}$$

By Lemma 4, for every $D \in \mathcal{D}_{\mathcal{T}}$ the set of nodes $(\Delta, K, \mathcal{N})$ such that $D \in \Delta$ has a minimum element $(\Delta_M, K_M, \mathcal{N}_M)$ w.r.t. $\preceq$. We call $(\Delta_M, K_M, \mathcal{N}_M)$ the *minimum node associated with* D.

Now, for every $D \in \mathcal{D}_{\mathcal{T}}$ we define the information term $\mathcal{I}_{\mathcal{T}}(D) \in \mathrm{IT}_{\mathcal{N}_{\mathcal{T}}}(D)$ by induction on the structure of D.

- $\mathcal{I}_{\mathcal{T}}(D) = \mathtt{tt}$ if D is a simple formula.
- $\mathcal{I}_{\mathcal{T}}(c : A \sqcap B) = (\mathcal{I}_{\mathcal{T}}(c : A), \mathcal{I}_{\mathcal{T}}(c : B))$.
- If $D \equiv c : A \sqcup B$, let $(\Delta_M, K_M, \mathcal{N}_M)$ be the minimum node associated with D. Then:

$$\mathcal{I}_{\mathcal{T}}(c : A \sqcup B) = \begin{cases} (1, \mathcal{I}_{\mathcal{T}}(c : A)) & \text{if } c : A \in \Delta_M \\ (2, \mathcal{I}_{\mathcal{T}}(c : B)) & \text{otherwise} \end{cases}$$

We remark that if $c : A \notin \Delta_M$ then $c : B \in \Delta_M$ must hold.

– If $D \equiv c : \exists R.A$, let $(\Delta_M, K_M, \mathcal{N}_M)$ be the minimum node associated with D. Then:

$$\mathcal{I}_{\mathcal{T}}(c : \exists R.A) \;=\; (d,\, \mathcal{I}_{\mathcal{T}}(d : A))$$

where d is any element in $\mathcal{N}_M$ such that $(c, d) : R \in \Delta_M$ and $d : A \in \Delta_M$ (being Δ_M saturated in $\mathcal{N}_M$, there exists at least one).

– $\mathcal{I}_{\mathcal{T}}(c : \forall R.A)$ is the map ϕ such that, for every $d \in \mathcal{N}_{\mathcal{T}}$:

$$\phi(d) \;=\; \begin{cases} \mathcal{I}_{\mathcal{T}}(d : A) & \text{if } (c, d) : R \in \mathcal{D}_{\mathcal{T}} \\ \alpha^+ \in \textsc{it}_{\mathcal{N}_{\mathcal{T}}}(d : A) & \text{otherwise} \end{cases}$$

where α^+ is an element of $\textsc{it}_{\mathcal{N}_{\mathcal{T}}}(d : A)$.

– $\mathcal{I}_{\mathcal{T}}(\forall_G A)$ is the map ϕ such that, for every $d \in \mathrm{dom}(G)$:

$$\phi(d) \;=\; \mathcal{I}_{\mathcal{T}}(d : A)$$

One can easily check that the above definition is sound. For instance, let $c : \forall R.A \in \mathcal{D}_{\mathcal{T}}$ and $(c, d) : R \in \mathcal{D}_{\mathcal{T}}$. There are two nodes $(\Delta_1, K_1, \mathcal{N}_1)$ and $(\Delta_2, K_2, \mathcal{N}_2)$ such that $c : \forall R.A \in \Delta_1$ and $(c, d) : R \in \Delta_2$. By Lemma 4 we get $c : \forall R.A \in \Delta_2$, hence $d : A \in \Delta_2$, which implies $d : A \in \mathcal{D}_{\mathcal{T}}$. Thus, $\mathcal{I}_{\mathcal{T}}(d : A)$ is inductively defined. As usual, if $\Delta = \{D_1, \ldots, D_n\}$, $\mathcal{I}_{\mathcal{T}}(\Delta)$ denotes the tuple $(\mathcal{I}_{\mathcal{T}}(D_1), \ldots, \mathcal{I}_{\mathcal{T}}(D_n))$. Thus, $\mathcal{M} \rhd \langle \mathcal{I}_{\mathcal{T}}(\Delta) \rangle\, \Delta$ iff, for every $D \in \Delta$, $\mathcal{M} \rhd \langle \mathcal{I}_{\mathcal{T}}(D) \rangle\, D$.

Example 5 Let us consider the c-tree in Example 4. We have:

$$\mathcal{D}_{\mathcal{T}} \;=\; \Delta_1 \cup \Delta_2 \qquad \mathcal{N}_{\mathcal{T}} \;=\; \{c,\, d_1,\, d_2,\, d_3,\, d_4\}$$

The definition of $\mathcal{I}_{\mathcal{T}}(D)$ for non-simple formulas $D \in \mathcal{D}_{\mathcal{T}}$ is:

$$\mathcal{I}_{\mathcal{T}}(d_2 : A \sqcup B) \;=\; (2, \mathtt{tt}) \qquad \mathcal{I}_{\mathcal{T}}(d_1 : \exists S.(A \sqcup B)) \;=\; (d_2, (2, \mathtt{tt}))$$

$$\mathcal{I}_{\mathcal{T}}(d_4 : A \sqcup B) \;=\; (1, \mathtt{tt}) \qquad \mathcal{I}_{\mathcal{T}}(d_3 : \exists S.(A \sqcup B)) \;=\; (d_4, (1, \mathtt{tt}))$$

$$\mathcal{I}_{\mathcal{T}}(c : \forall R.\neg\neg\exists S.A) \;=\; \psi \;\text{ such that, for every } d \in \mathcal{N}_{\mathcal{T}},\, \psi(d) = \mathtt{tt}$$

$$\mathcal{I}_{\mathcal{T}}(c : \forall R.\exists S.(A \sqcup B)) \;=\; \phi \;\text{ such that, for every } d \in \mathcal{N}_{\mathcal{T}}:$$

$$\phi(d) \;=\; \begin{cases} (d_2, (2, \mathtt{tt})) & \text{if } d \equiv d_1 \\ (d_4, (1, \mathtt{tt})) & \text{if } d \equiv d_3 \\ \alpha^+ & \text{otherwise} \end{cases}$$

where we can take, for instance, $\alpha^+ = (c, (1, \mathtt{tt}))$.

Let us consider the leaf $(\Delta_1, d_2 : A, \{c, d_1, d_2\})$ of $\mathcal{T}$. Since $\Delta_1 \not\vdash_{\mathcal{ALCG}} d_2 : A$, by Theorem 2 there exists a model $\mathcal{M}_1$ of Δ_1 such that $\mathcal{M}_1 \not\models d_2 : A$ and, for every $d, d' \in \mathcal{N}_{\mathcal{T}}$, the following holds:

$$\begin{array}{lll} \mathcal{M}_1 \models (d, d') : R & \text{iff} & d \equiv c \text{ and } d' \equiv d_1 \\ \mathcal{M}_1 \models (d, d') : S & \text{iff} & d \equiv d_1 \text{ and } d' \equiv d_2 \end{array}$$

It follows that $\mathcal{M}_1 \rhd \langle \phi \rangle\, c : \forall R.\exists S.(A \sqcup B)$, hence:

(i) $\mathcal{M}_1 \rhd \langle \mathcal{I}_{\mathcal{T}}(\Delta_0) \rangle\, \Delta_0$

Similarly, since $(\Delta_2, d_4 : B, \{c, d_3, d_4\})$ is a leaf of $\mathcal{T}$, there exists $\mathcal{M}_2$ defined according to Theorem 2 such that:

(ii) $\mathcal{M}_2 \rhd \langle \mathcal{I}_\mathcal{T}(\Delta_0) \rangle\, \Delta_0$

We show that:

(iii) For every $\eta \in \mathrm{IT}_{\mathcal{N}_\mathcal{T}}(c : H)$, either $\mathcal{M}_1 \not\rhd \langle \eta \rangle\, c : H$ or $\mathcal{M}_2 \not\rhd \langle \eta \rangle\, c : H$.

If $\eta \in \mathrm{IT}_{\mathcal{N}_\mathcal{T}}(c : H)$, either $\eta \equiv (1, \phi_1)$ or $\eta \equiv (2, \phi_2)$. Let us assume that $\eta = (1, \phi_1)$, with $\phi_1 \in \mathrm{IT}_{\mathcal{N}_\mathcal{T}}(c : \forall R.\exists S.A)$, and let $\phi_1(d_1) = (d, \mathtt{tt})$, with $d \in \mathcal{N}_\mathcal{T}$. If $d \equiv d_2$, by the fact that $\mathcal{M}_1 \models (c, d_1) : R$, $\mathcal{M}_1 \models (d_1, d_2) : S$ and $\mathcal{M}_1 \not\models d_2 : A$, we get $\mathcal{M}_1 \not\rhd \langle \phi_1 \rangle\, c : \forall R.\exists S.A$. If $d \not\equiv d_1$, since $\mathcal{M}_1 \models (c, d_1) : R$ and $\mathcal{M}_1 \not\models (d_1, d) : S$, we get $\mathcal{M}_1 \not\rhd \langle \phi_1 \rangle\, c : \forall R.\exists S.A$. Thus, $\eta \equiv (1, \phi_1)$ implies $\mathcal{M}_1 \not\rhd \langle \eta \rangle\, c : H$. In a similar way one can prove that $\eta \equiv (2, \phi_2)$ implies $\mathcal{M}_2 \not\rhd \langle \eta \rangle\, c : H$, hence (iii) is proved.
By (i), (ii) and (iii) we conclude $\Delta_0 \not\models_c c : H$.

The following lemmas generalise the reasoning of the previous example to any c-tree.

Lemma 6 *Let $(\Delta, K, \mathcal{N})$ be a leaf of a c-tree $\mathcal{T}$. Then, there exists a finite model $\mathcal{M}$ for $\mathcal{L}_{\mathcal{N}_\mathcal{T}}$ such that:*

(i) $\mathcal{M} \rhd \langle \mathcal{I}_\mathcal{T}(\Delta) \rangle\, \Delta$.
(ii) $\mathcal{M} \not\models K$.
(iii) *For every $c, d \in \mathcal{N}_\mathcal{T}$ and $R \in \mathrm{NR}$, $\mathcal{M} \models (c, d) : R$ iff $(c, d) : R \in \Delta$.*
(iv) *For every $G \in \mathrm{NG}_{\mathcal{N}_\mathcal{T}}$, $\mathcal{M} \models c : G$ iff $c \in dom(G)$.*

Proof By definition of $\mathcal{T}$, $\Delta \cup \{K\} \subseteq \mathcal{L}_{\mathcal{N}_\mathcal{T}}$ and $\Delta \not\vdash_{\overline{\mathcal{ALCG}}} K$. By Theorem 2 there exists a model finite $\mathcal{M}$ for $\mathcal{L}_{\mathcal{N}_\mathcal{T}}$ such that

(1) $\mathcal{M} \models \Delta$.
(2) $\mathcal{M} \not\models K$.
(3) For every $c, d \in \mathcal{N}_\mathcal{T}$ and $R \in \mathrm{NR}$, $\mathcal{M} \models (c, d) : R$ iff $(c, d) : R \in \Delta$.
(4) For every $G \in \mathrm{NG}_{\mathcal{N}_\mathcal{T}}$, $\mathcal{M} \models c : G$ iff $c \in \mathrm{dom}(G)$.

To complete the proof, it only remains to show that:

(*) $D \in \Delta$ implies $\mathcal{M} \rhd \langle \mathcal{I}_\mathcal{T}(D) \rangle\, D$.

If D is a simple formula, (*) immediately follows by (1).

Let $D \equiv c : A \sqcap B$. Then, $\mathcal{I}_\mathcal{T}(D) = (\mathcal{I}_\mathcal{T}(c : A), \mathcal{I}_\mathcal{T}(c : B))$. Since Δ is saturated in $\mathcal{N}$, we have $c : A \in \Delta$ and $c : B \in \Delta$. By induction hypothesis, $\mathcal{M} \rhd \langle \mathcal{I}_\mathcal{T}(c : A) \rangle\, c : A$ and $\mathcal{M} \rhd \langle \mathcal{I}_\mathcal{T}(c : B) \rangle\, c : B$, and (*) holds.

Let $D \equiv c : A \sqcup B$ and let $(\Delta_M, K_M, \mathcal{N}_M)$ be the minimum node associated with D. Assume that $c : A \in \Delta_M$ (the case $c : B \in \Delta_M$ is similar). Then, $\mathcal{I}_\mathcal{T}(D) = (1, \mathcal{I}_\mathcal{T}(c : A))$. By definition of minimum node, $(\Delta_M, K_M, \mathcal{N}_M) \preceq (\Delta, K, \mathcal{N})$. It follows that $c : A \in \Delta$, hence, by induction hypothesis, $\mathcal{M} \rhd \langle \mathcal{I}_\mathcal{T}(c : A) \rangle\, c : A$, which implies $\mathcal{M} \rhd \langle (1, \mathcal{I}_\mathcal{T}(c : A)) \rangle\, D$.

Let $D \equiv c : \exists R.A$ and let $(\Delta_M, K_M, \mathcal{N}_M)$ be the minimum node associated with D. We have $\mathcal{I}_\mathcal{T}(D) = (d, \mathcal{I}_\mathcal{T}(d : A))$, where $(c, d) : R \in \Delta_M$ and $d : A \in \Delta_M$. Since $(\Delta_M, K_M, \mathcal{N}_M) \preceq (\Delta, K, \mathcal{N})$, it follows that $(c, d) : R \in \Delta$ and $d : A \in \Delta$. Thus, $\mathcal{M} \models$

$(c, d) : R$ and, by the induction hypothesis, $\mathcal{M} \rhd \langle \mathcal{I}_{\mathcal{T}}(d : A)\rangle\, d : A$. We conclude $\mathcal{M} \rhd \langle (d, \mathcal{I}_{\mathcal{T}}(c : A))\rangle\, D$.

Let $D \equiv c : \forall R.A$. By (1), we have $\mathcal{M} \models c : \forall R.A$. Let $\phi = \mathcal{I}_{\mathcal{T}}(D)$ and let $d \in \mathcal{N}_{\mathcal{T}}$. If $\mathcal{M} \models (c, d) : R$, by (3) we have $(c, d) : R \in \Delta$. It follows that $d : A \in \Delta$ and $(c, d) : R \in \mathcal{D}_{\mathcal{T}}$. Thus, $\phi(d) = \mathcal{I}_{\mathcal{T}}(d : A)$ and, by induction hypothesis, $\mathcal{M} \rhd \langle \phi(d)\rangle\, d : A$. We conclude $\mathcal{M} \rhd \langle \phi\rangle\, D$.

The case $D \equiv \forall_G A$ is similar and requires Point (4). □

Lemma 7 *Let* $(\Delta, K, \mathcal{N})$ *be a node of a c-tree* $\mathcal{T}$. *Then, for every* $\eta \in \mathrm{IT}_{\mathcal{N}_{\mathcal{T}}}(K)$, *there exists a finite model* $\mathcal{M}$ *for* $\mathcal{L}_{\mathcal{N}_{\mathcal{T}}}$ *such that:*

(i) $\mathcal{M} \rhd \langle \mathcal{I}_{\mathcal{T}}(\Delta)\rangle\, \Delta$.
(ii) $\mathcal{M} \not\rhd \langle \eta\rangle\, K$.
(iii) *For every* $c, d \in \mathcal{N}$ *and* $R \in \mathrm{NR}$, $\mathcal{M} \models (c, d) : R$ *iff* $(c, d) : R \in \Delta$.
(iv) *For every* $G \in \mathrm{NG}_{\mathcal{N}_{\mathcal{T}}}$, $\mathcal{M} \models c : G$ *iff* $c \in dom(G)$.

Proof The proof is by induction on the height of the node $(\Delta, K, \mathcal{N})$ (i.e., the maximum length of a path between $(\Delta, K, \mathcal{N})$ and a leaf of $\mathcal{T}$). If $(\Delta, K, \mathcal{N})$ is a leaf, then K is a simple formula, hence $\mathrm{IT}_{\mathcal{T}}(K) = \mathtt{tt}$ and the assertion immediately follows by Lemma 6. Otherwise, $(\Delta, K, \mathcal{N})$ has one or more immediate successors according to the form of K.

Let $K \equiv c : A \sqcap B$ and let $(\Delta, c : A, \mathcal{N})$ be the only immediate successor of $(\Delta, K, \mathcal{N})$ (the other case is similar). Let $(\alpha, \beta) \in \mathrm{IT}_{\mathcal{N}_{\mathcal{T}}}(K)$. Since $\alpha \in \mathrm{IT}_{\mathcal{N}_{\mathcal{T}}}(c : A)$, by induction hypothesis there exists $\mathcal{M}$ such that $\mathcal{M} \rhd \langle \mathcal{I}_{\mathcal{T}}(\Delta)\rangle\, \Delta$ and $\mathcal{M} \not\rhd \langle \alpha\rangle\, c : A$, hence $\mathcal{M} \not\rhd \langle (\alpha, \beta)\rangle\, K$.

Let $K \equiv c : A \sqcup B$ and let, for instance, $\eta \equiv (1, \alpha)$, with $\alpha \in \mathrm{IT}_{\mathcal{N}_{\mathcal{T}}}(c : A)$ (the other case is similar). Since $(\Delta, c : A, \mathcal{N})$ is an immediate successor of $(\Delta, K, \mathcal{N})$, by induction hypothesis there is $\mathcal{M}$ such that $\mathcal{M} \rhd \langle \mathcal{I}_{\mathcal{T}}(\Delta)\rangle\, \Delta$ and $\mathcal{M} \not\rhd \langle \alpha\rangle\, c : A$, which implies $\mathcal{M} \not\rhd \langle (1, \alpha)\rangle\, K$.

Let $K \equiv c : \exists R.A$ and let $(d, \alpha) \in \mathrm{IT}_{\mathcal{N}_{\mathcal{T}}}(K)$, with $d \in \mathcal{N}_{\mathcal{T}}$ and $\alpha \in \mathrm{IT}_{\mathcal{N}_{\mathcal{T}}}(d : A)$. If $(c, d) : R \in \Delta$, then $(\Delta, d : A, \mathcal{N})$ is an immediate successor of $(\Delta, K, \mathcal{N})$. By induction hypothesis, there is $\mathcal{M}$ such that $\mathcal{M} \rhd \langle \mathcal{I}_{\mathcal{T}}(\Delta)\rangle\, \Delta$ (hence, $\mathcal{M} \models (c, d) : R$) and $\mathcal{M} \not\rhd \langle \alpha\rangle\, d : A$. It follows that $\mathcal{M} \not\rhd \langle (d, \alpha)\rangle\, K$. Let $(c, d) : R \notin \Delta$ and let $(\Delta, Z, \mathcal{N})$ be any immediate successor of $(\Delta, K, \mathcal{N})$ (there exists at least one). By induction hypothesis, there exists a model $\mathcal{M}$ such that $\mathcal{M} \rhd \langle \mathcal{I}_{\mathcal{T}}(\Delta)\rangle\, \Delta$ and $\mathcal{M} \not\models (c, d) : R$. It follows that $\mathcal{M} \not\rhd \langle (d, \alpha)\rangle\, K$.

Let $K \equiv c : \forall R.A$ and let $\phi \in \mathrm{IT}_{\mathcal{N}_{\mathcal{T}}}(K)$. Then, for some $d \in \mathcal{N}_{\mathcal{T}}$, there exists an immediate successor $(\Delta', d : A, \mathcal{N}')$ of $(\Delta, K, \mathcal{N})$ such that $\Delta \cup \{(c, d) : R\} \subseteq \Delta'$. Since $\phi(d) \in \mathrm{IT}_{\mathcal{N}_{\mathcal{T}}}(d : A)$, by the induction hypothesis there exists a model $\mathcal{M}$ such that $\mathcal{M} \rhd \langle \mathcal{I}_{\mathcal{T}}(\Delta')\rangle\, \Delta'$ and $\mathcal{M} \not\rhd \langle \phi(d)\rangle\, d : A$. It follows that $\mathcal{M} \rhd \langle \mathcal{I}_{\mathcal{T}}(\Delta)\rangle\, \Delta$ and, by the fact that $\mathcal{M} \models (c, d) : R$, we get $\mathcal{M} \not\rhd \langle \phi\rangle\, K$. Point (iii) follows by the fact that $\Delta'_{/\mathcal{N}} = \Delta$ (see the definition of the rule $r_{\forall_G}$ in Table 3); Point (iv) by the fact that $c : G \in \Delta$ iff $c : G \in \Delta'$.

The case $K \equiv \forall_G A$ is similar. □

As a consequence, we get:

Theorem 9 *If there is a c-tree for* (Γ, K), *then* $\Gamma \not\models_{\mathrm{c}} K$.

Proof Let $\mathcal{T}$ be a c-tree for (Γ, K) having root $(\Delta, K, \mathcal{N})$, where $\Gamma \subseteq \Delta$, and let $\overline{\gamma} = \mathcal{I}_{\mathcal{T}}(\Gamma)$. Let us assume $\Gamma \models_{\mathrm{c}} K$. Since $\Gamma \cup \{K\} \subseteq \mathcal{L}_{\mathcal{N}_{\mathcal{T}}}$ and $\overline{\gamma} \in \mathrm{IT}_{\mathcal{N}_{\mathcal{T}}}(\Gamma)$, by Definition 2 there is $\eta' \in \mathrm{IT}_{\mathcal{N}_{\mathcal{T}}}(K)$ such that, for every model $\mathcal{M}$ for $\mathcal{L}_{\mathcal{N}_{\mathcal{T}}}$, $\mathcal{M} \rhd \langle\overline{\gamma}\rangle\, \Gamma$ implies $\mathcal{M} \rhd \langle\eta'\rangle\, K$. On the other hand, by Lemma 7 there must be a model $\mathcal{M}'$ for $\mathcal{L}_{\mathcal{N}_{\mathcal{T}}}$ such that $\mathcal{M}' \rhd \langle\mathcal{I}_{\mathcal{T}}(\Delta)\rangle\, \Delta$ and $\mathcal{M}' \not\rhd \langle\eta'\rangle\, K$. In particular, $\mathcal{M}' \rhd \langle\overline{\gamma}\rangle\, \Gamma$, and we get a contradiction. We conclude $\Gamma \not\models_{\mathrm{c}} K$. □

We summarise the main results of this section:

Corollary 2 *The following statements are equivalent:*

(i) $\Gamma \vdash_{\mathcal{BCDL}} K$.
(ii) $\Gamma \models_{\mathrm{c}} K$.
(iii) *There is no c-tree for* (Γ, K).

Proof (i) implies (ii) by the Soundness Theorem for $\mathcal{ND}_{\mathrm{c}}$ (Theorem 7). (ii) implies (iii) by Theorem 9. (iii) implies (i) by Theorem 8. □

Example 6 Let Δ_0 and H be defined as in Example 4. Since there exists a c-tree for $(\Delta_0, c : H)$, we have $\Delta_0 \nvdash_{\mathcal{BCDL}} c : H$. Note that $\Delta_0 \vdash_{\mathcal{ALCG}} c : H$.

We conclude the section by discussing some constructive properties of $\mathcal{ND}_{\mathrm{c}}$. In particular, we prove that, under suitable conditions on the assumptions Γ of a proof, the *disjunction property* (DP) and the *explicit definability property* (EDP) hold. Namely, let Γ be a set of closed formula and let $c \in \mathtt{NI}$, the properties DP and EDP are formulated as follows:

(DP) If $\Gamma \vdash_{\mathcal{BCDL}} c : A \sqcup B$, then $\Gamma \vdash_{\mathcal{BCDL}} c : A$ or $\Gamma \vdash_{\mathcal{BCDL}} c : B$.
(EDP) If $\Gamma \vdash_{\mathcal{BCDL}} c : \exists R.A$, then there exists $d \in \mathtt{NI}$ such that $\Gamma \vdash_{\mathcal{BCDL}} (c, d) : R$ and $\Gamma \vdash_{\mathcal{BCDL}} d : A$.

In general DP and EDP do not hold; for instance, $c : A \sqcup B \vdash_{\mathcal{BCDL}} c : A \sqcup B$ but neither $c : A \sqcup B \vdash_{\mathcal{BCDL}} c : A$ nor $c : A \sqcup B \vdash_{\mathcal{BCDL}} c : B$ (the same happens for (EDP), taking $\Gamma = \{c : \exists R.A\}$). We have to restrict the set Γ to the well-known class of *Harrop formulas*, namely the formulas not containing the connectives $\sqcup$ and $\exists$. Let $\mathcal{N}$ be a finite set of individual names and Γ a finite set of closed Harrop formulas of $\mathcal{L}_{\mathcal{N}}$. One can easily check that, for every $H_f \in \Gamma$, $\mathrm{IT}_{\mathcal{N}}(H_f)$ contains exactly one information term. This implies that, for every $c \in \mathcal{N}$, the following properties hold:

(1) If $\Gamma \models_{\mathrm{c}} c : A \sqcup B$, then $\Gamma \models_{\mathrm{c}} c : A$ or $\Gamma \models_{\mathrm{c}} c : B$.
(2) If $\Gamma \models_{\mathrm{c}} c : \exists R.A$, then there exists $d \in \mathcal{N}$ such that $\Gamma \models_{\mathrm{c}} (c, d) : R$ and $\Gamma \models_{\mathrm{c}} d : A$.

Thus, by the completeness theorem, the constructivity of $\mathcal{ND}_{\mathrm{c}}$ can be stated as follows:

Theorem 10 *Let $\mathcal{N}$ be a finite set of individual names, $\mathcal{A} \subseteq \mathcal{L}_{\mathcal{N}}$ an ABox, $\mathcal{T} \subseteq \mathcal{L}_{\mathcal{N}}$ a TBox such that the formulas in $\mathcal{T}$ are closed Harrop formulas, and let $c \in \mathcal{N}$.*

(1) *If $\mathcal{A} \cup \mathcal{T} \vdash_{\mathcal{BCDL}} c : A \sqcup B$, then $\mathcal{A} \cup \mathcal{T} \vdash_{\mathcal{BCDL}} c : A$ or $\mathcal{A} \cup \mathcal{T} \vdash_{\mathcal{BCDL}} c : B$.*
(2) *If $\mathcal{A} \cup \mathcal{T} \vdash_{\mathcal{BCDL}} c : \exists R.A$, then there exists $d \in \mathcal{N}$ such that $\mathcal{A} \cup \mathcal{T} \vdash_{\mathcal{BCDL}} (c, d) : R$ and $\mathcal{A} \cup \mathcal{T} \vdash_{\mathcal{BCDL}} d : A$*

6 On Unbounded Quantification

In this paper we have considered bounded quantified formulas of the kind $\forall_G A$, where $\mathrm{dom}(G)$ is a finite domain *fixed* by the language. What happens if we admit unbounded quantified formulas of the form $\forall_C A$, where C is *any* concept name?

Let $\mathcal{L}_u$ be the language obtained by extending $\mathcal{L}$ with the unbounded quantified formulas. The validity of an unbounded quantified formulas can be defined as:

$$\mathcal{M} \models \forall_C H \;\text{ iff }\; C^{\mathcal{M}} \subseteq H^{\mathcal{M}}$$

We remark that $\forall_C H$ models in our setting the usual subsumption relation $C \sqsubseteq H$. For a finite $\mathcal{N} \subseteq \mathtt{NI}$, an information term for $\forall_C A$ can be defined as

$$\mathrm{IT}_{\mathcal{N}}(\forall_C A) \;=\; \left\{ \phi : \mathcal{N} \to \bigcup_{c\in\mathcal{N}} \mathrm{IT}_{\mathcal{N}}(c : A) \mid \phi(c) \in \mathrm{IT}_{\mathcal{N}}(c : A) \right\}$$

and $\mathcal{M} \rhd \langle\phi\rangle\, \forall_C A$ iff:

$$\mathcal{M} \models \forall_C A \text{ and, for every } c \in \mathcal{N}, \mathcal{M} \models c : C \text{ implies } \mathcal{M} \rhd \langle\phi(c)\rangle\, c : A$$

To properly treat $\forall_C$ we need the following rules

$$\dfrac{\begin{array}{c}\Gamma, [p : C] \\ \vdots\;\pi' \\ p : A\end{array}}{\forall_C A}\,\forall_C I \quad \text{where } p \in \mathtt{Var} \text{ and } p \text{ does not occur in } \Gamma \qquad\qquad \dfrac{\begin{array}{cc}\Gamma_1 & \Gamma_2 \\ \vdots\;\pi_1 & \vdots\;\pi_2 \\ \forall_C A & t : C\end{array}}{t : A}\,\forall_C E$$

Let $\mathcal{ND}^u_{\mathrm{c}}$ be the calculus obtained by adding to $\mathcal{ND}_{\mathrm{c}}$ the rules $\forall_C I$ and $\forall_C E$. We say that $\Gamma \vdash_{\overline{\mathrm{Cu}}} K$ iff there exists a proof $\pi : \Gamma \vdash K$ in $\mathcal{ND}^u_{\mathrm{c}}$. It is easy to extend the map $\Phi^{\pi}_{\sigma,\mathcal{N}}$ of Section 4 to the new rules so that Property (P2) holds, thus:

Theorem 11 (Soundness of $\mathcal{ND}^u_{\mathrm{c}}$) $\Gamma \vdash_{\overline{\mathrm{Cu}}} K$ *implies* $\Gamma \models_{\overline{\mathrm{c}}} K$.

We show that the converse of Theorem 11 (namely, the Completeness of $\mathcal{ND}^u_{\mathrm{c}}$ with respect to $\models_{\overline{\mathrm{c}}}$) does not hold. To this aim let us consider the following translation Tr from the formulas of $\mathcal{L}_u$ into the formulas of predicate first order logic:

- $\mathrm{Tr}(\bot) \;=\; \bot$
- $\mathrm{Tr}((s,t) : R) \;=\; R(s,t)$
- $\mathrm{Tr}(t : C) \;=\; C(t)$, if $C \in \mathtt{NC} \cup \mathtt{NG}$
- $\mathrm{Tr}(t : \neg A) \;=\; \neg\mathrm{Tr}(t : A)$
- $\mathrm{Tr}(t : A \sqcap B) \;=\; \mathrm{Tr}(t : A) \wedge \mathrm{Tr}(t : B)$
- $\mathrm{Tr}(t : A \sqcup B) \;=\; \mathrm{Tr}(t : A) \vee \mathrm{Tr}(t : B)$
- $\mathrm{Tr}(t : \exists R.A) \;=\; \exists x.(R(t,x) \wedge \mathrm{Tr}(x : A))$
- $\mathrm{Tr}(t : \forall R.A) \;=\; \forall x.(R(t,x) \to \mathrm{Tr}(x : A))$
- $\mathrm{Tr}(\forall_G A) \;=\; \forall x.((G(x) \leftrightarrow x = c_1 \vee \cdots \vee x = c_n) \wedge (G(x) \to \mathrm{Tr}(x : A)))$ where $\mathrm{dom}(G) = \{c_1, \ldots, c_n\}$
- $\mathrm{Tr}(\forall_C A) \;=\; \forall x.(C(x) \to \mathrm{Tr}(x : A))$

For a set of formulas Γ, $\mathrm{Tr}(\Gamma)$ denotes the set of formulas $\mathrm{Tr}(K)$ such that $K \in \Gamma$.

We write $\Gamma \vdash_{\mathbf{Int}^+} K$ to mean that there exists a proof $\pi : \Gamma \vdash K$ in a calculus for the logic obtained by adding to Intuitionistic Logic **Int** the equality theory and the following axiom schema

$$\text{(KUR)} \equiv \forall x.\neg\neg A(x) \to \neg\neg\forall x.A(x)$$

$$\text{(NegAt)} \equiv \neg\neg a \to a \quad \text{where } a \text{ is an atomic formula}$$

It is easy to prove:

Lemma 8 *$\Gamma \vdash_{\mathrm{Cu}} K$ implies $Tr(\Gamma) \vdash_{\mathbf{Int}^+} Tr(K)$.*

Now, let us consider the set $\overline{\Gamma}$ of formulas of $\mathcal{L}_u$ defined as follows:

$$\overline{\Gamma} \;=\; \{\, \forall_C(A \sqcup B),\; d : \neg\neg(A \sqcup C),\; d : \neg\neg(B \sqcup C) \,\} \quad A, B, C \in \mathrm{NC}$$

Lemma 9 $\overline{\Gamma} \nvdash_{\mathrm{Cu}} d : A \sqcup B$

Proof Suppose that $\overline{\Gamma} \vdash_{\mathrm{Cu}} d : A \sqcup B$. Let

$$K \;\equiv\; \forall x(\, C(x) \to A(x) \vee B(x)\,) \;\wedge\; \neg\neg(\, A(d) \vee C(d)\,) \;\wedge\; \neg\neg(\, B(d) \vee C(d)\,)$$

Then, by Lemma 8 we should have

$$K \vdash_{\mathbf{Int}^+} A(d) \vee B(d) \tag{2}$$

Using standard techniques based on Kripke semantics for intermediate logics (see e.g. [20]), one can prove that Fact (2) cannot hold.[2] □

Lemma 10 $\overline{\Gamma} \models_{c} d : A \sqcup B$

Proof Let $\mathcal{N}$ be any finite set of individual names containing d. We have to prove that, for every $\phi \in \mathrm{IT}_{\mathcal{N}}(\forall_C(A \sqcup B))$, there exists $\eta \in \mathrm{IT}_{\mathcal{N}}(d : A \sqcup B)$ such that, for every model $\mathcal{M}$ for $\mathcal{L}_{\mathcal{N}}$, the following holds:

(a) If $\mathcal{M} \rhd \langle\phi\rangle\, \forall_C(A \sqcup B)$ and $\mathcal{M} \models d : (A \sqcup C)$ and $\mathcal{M} \models d : (B \sqcup C)$, then $\mathcal{M} \rhd \langle\eta\rangle\, d : A \sqcup B$.

Let $\phi \in \mathrm{IT}_{\mathcal{N}}(\forall_C(A \sqcup B))$. We define:

$$\eta = \begin{cases} (1, \mathtt{tt}) \text{ if } \phi(d) = (1, \mathtt{tt}) \\ (2, \mathtt{tt}) \text{ if } \phi(d) = (2, \mathtt{tt}) \end{cases}$$

Let us assume $\phi(d) = (1, \mathtt{tt})$. If the premise of (a) holds, then $\mathcal{M} \models d : A$. Indeed, since $\mathcal{M} \models d : (A \sqcup C)$, either $\mathcal{M} \models d : A$ or $\mathcal{M} \models d : C$. In the latter case, since $\mathcal{M} \rhd \langle\phi\rangle\, \forall_C(A \sqcup B)$, we get $\mathcal{M} \rhd \langle\phi(d)\rangle\, d : A \sqcup B$, namely $\mathcal{M} \rhd \langle(1, \mathtt{tt})\rangle\, d : A \sqcup B$, which implies $\mathcal{M} \models d : A$. Since $\eta \equiv (1, \mathtt{tt})$, we conclude $\mathcal{M} \rhd \langle\eta\rangle\, d : A \sqcup B$, and (a) is proved. The case $\phi(d) = (2, \mathtt{tt})$ is similar. □

[2] Actually, to prove Fact (2) one needs to consider an extension of $\mathbf{Int}^+$ also including the *Kreisel and Putnam* axiom schema $(\neg A \to B \vee C) \to (\neg A \to B) \vee (\neg A \to C)$ [4].

By Lemmas 9 and 10, $\mathcal{ND}_c^u$ is not complete with respect to constructive consequence.

We think that we cannot apply the tools described in this paper to prove a completeness theorem. The major drawback is that we cannot extend Theorem 2 to $\mathcal{L}_u$ so that the following property holds:

(v) For every $c \in \mathtt{NI}$ and every $C \in \mathtt{NC}$, $\mathcal{M} \models c : C$ iff $c : C \in \Gamma$.

For instance, every model $\mathcal{M}$ of $\overline{\Gamma}$ satisfies $\mathcal{M} \models d : A$ or $\mathcal{M} \models d : B$, but neither $d : A$ nor $d : B$ belongs to $\overline{\Gamma}$. Thus, different techniques have to be studied.

As for the future works, we think that a few questions deserve to be investigated. First of all, as we remarked in the introduction, in our context negation is treated classically. However we can extend $\mathcal{BCDL}$ with a further operator modeling Nelson constructive negation [13, 14] still obtaining a sound and complete natural deduction characterisation. Another significant point is to extend information terms semantics to treat the usual quantifiers defined in the description logic context. Finally, an interesting issue is the development of a Kripke-style semantics for $\mathcal{BCDL}$. Indeed, according to the authors experience, such a semantics is an important guideline in the development of "efficient" decision procedures for a logic. This is also related to the study of complexity issues of the usual decision problems considered in description logics.

References

1. 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, Cambridge (2003)
2. Bozzato, L., Ferrari, M., Villa, P.: Actions over a constructive semantics for description logics. Fundam. Inform. **96**, 1–17 (2009)
3. Brachman, R.J., Mcguinness, D.L., Patel-Schneider, P.F., Resnick, L.A., Borgida, A.: Living with CLASSIC: When and How to Use a KL-ONE-like Language (1991)
4. Chagrov, A., Zakharyaschev, M.: Modal Logic. Oxford University Press, Oxford (1997)
5. de Paiva, V.: Constructive description logics: what, why and how. Technical report, Xerox Parc (2003)
6. Ferrari, M., Fiorentini, C., Momigliano, A., Ornaghi, M.: Snapshot generation in a constructive object-oriented modeling language. In: King, A. (ed.) Logic Based Program Synthesis and Transformation, LOPSTR 2007, Selected Papers. Lecture Notes in Computer Science, vol. 4915, pp. 169–184. Springer, New York (2008)
7. Ferrari, M., Fiorentini, C., Ornaghi, M.: Extracting exact time bounds from logical proofs. In: Petterossi, A. (ed.) Logic Based Program Synthesis and Transformation, 11th International Workshop, LOPSTR 2001, Selected Papers. Lecture Notes in Computer Science, vol. 2372, pp. 245–265. Springer, New York (2002)
8. Gabbay, D.M.: Semantical Investigations in Heyting's Intuitionistic Logic. Reidel, Dordrecht (1981)
9. Kaneiwa, K.: Negations in description logic - contraries, contradictories, and subcontraries. In: Proceedings of the 13th International Conference on Conceptual Structures (ICCS '05), pp. 66–79. Kassel University Press, Kassel (2005)
10. Lipton, J., M.J. O'Donnel: Some intuitions behind realizability semantics for constructive logic: Tableaux and Läunchli countermodels. Ann. Pure Appl. Logic **81**, 187–239 (1996)
11. Miglioli, P., Moscato, U., Ornaghi, M.: Pap: a logic programming system based on a constructive logic. In: Foundations of Logic and Functional Programming. Lecture Notes in Computer Science, vol. 306, pp. 143–156. Springer, New York (1986)
12. Miglioli, P., Moscato, U., Ornaghi, M.: Abstract parametric classes and abstract data types defined by classical and constructive logical methods. J. Symb. Comput. **18**, 41–81 (1994)

13. Miglioli, P., Moscato, U., Ornaghi, M., Usberti, G.: A constructivism based on classical truth. Notre Dame J. Form. Log. **30**(1), 67–90 (1989)
14. Nelson, D.: Constructible falsity. J. Symb. Log. **14**, 16–26 (1949)
15. Odintsov, S.P., Wansing, H.: Inconsistency-tolerant description logic. Motivation and basic systems. In: Hendricks, V., Malinowski, J. (eds.) Trends in Logic. 50 Years of Studia Logica, pp. 301–335. Kluwer Academic, Dordrecht (2003)
16. Odintsov, S.P., Wansing, H.: Inconsistency-tolerant description logic. Part II: a tableau algorithm for $\mathcal{CACL}^{c}$. Journal of Applied Logic **6**(3), 343–360 (2008)
17. Prawitz, D.: Natural Deduction. Almquist and Wiksell, Stockholm (1965)
18. Schmidt, R.A.,. Tishkovsky, D.: Using tableau to decide expressive description logics with role negation. In: ISWC/ASWC 2007. Lecture Notes in Computer Science, vol. 4825, pp. 438–451 (2007)
19. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. Artif. Intell. **48**(1), 1–26 (1991)
20. Smorynski, C.A.: Applications of Kripke semantics. In: Troelstra, A.S. (ed.) Metamathematical Investigation of Intuitionistic Arithmetic and Analysis. Lecture Notes in Mathematics, vol. 344, pp. 324–391. Springer, New York (1973)
21. Troelstra, A.S. (ed.): Metamathematical Investigation of intuitionistic arithmetic and analysis. In: Lecture Notes in Mathematics, vol. 344. Springer, New York (1973)
22. Troelstra, A.S.: From constructivism to computer science. Theor. Comp. Sci. **211**(1–2), 233–252 (1999)
23. Troelstra, A.S., Schwichtenberg, H.: Basic proof theory. In: Cambridge Tracts in Theoretical Computer Science, vol. 43. Cambridge University Press, Cambridge (1996)