A constructive semantics for $\mathcal{ALC}$

Loris Bozzato[1], Mauro Ferrari[1], Camillo Fiorentini[2], Guido Fiorino[3]

[1] DICOM, Univ. degli Studi dell'Insubria, Via Mazzini 5, 21100, Varese, Italy
[2] DSI, Univ. degli Studi di Milano, Via Comelico, 39, 20135 Milano, Italy
[3] DIMEQUANT, Univ. degli Studi di Milano-Bicocca
P.zza dell'Ateneo Nuovo 1, 20126 Milano, Italy

Abstract. One of the main concerns of constructive semantics is to provide a computational interpretation for the proofs of a given logic. In this paper we introduce a constructive semantics for the basic description logic $\mathcal{ALC}$ in the spirit of the BHK interpretation. We prove that such a semantics provides an interpretation of $\mathcal{ALC}$ formulas consistent with the classical one and we show how, according to such a semantics, proofs of a suitable natural deduction calculus for $\mathcal{ALC}$ support a proofs-as-programs paradigm.

1 Introduction

In recent works, see e.g. [3, 5, 6], starting from different motivations, various constructive interpretations of description logics have been proposed. However, as far as we know, no computational interpretation for proofs has been given in this context. The aim of this paper is to propose a constructive semantics for $\mathcal{ALC}$ formulas, we call *information-terms semantics*, that allows us to give a computational interpretation of the proofs of a natural deduction calculus for $\mathcal{ALC}$. In particular, we will be able to read proofs of $\mathcal{ALC}$-"goals" as programs to compute goal answers.

The information-terms semantics is related to the BHK constructive explanation of logical connectives (see [7, 11] for a deeper discussion) and has already been applied in several frameworks [4, 8]. An information term is a mathematical object that explicitly explains the truth of a formula in a given classical model. For instance, if we prove that an individual c belongs to the concept $\exists R.C$, the information term provides the witness d such that $(c, d) \in R$ and $d \in C$. Differently from other approaches, such as [3, 5], information-terms semantics relies on the classical reading of logical connectives; as a consequence, we can read $\mathcal{ALC}$ formulas in the usual way.

In this paper we introduce the information-terms semantics and we compare it with the classical one. Then, we introduce a natural deduction calculus $\mathcal{ND}_\mathrm{c}$ for $\mathcal{ALC}$ and we show that it is sound with respect to information-terms semantics. As a by-product of the Soundness Theorem, we get a computational interpretation of proofs. We show, by means of an example, that this interpretation supports the proofs-as-programs paradigm.

2 $\mathcal{ALC}$ language and semantics

We begin introducing the language $\mathcal{L}$ for $\mathcal{ALC}$ [1, 10], based on the following denumerable sets: the set NR of *role names*, the set NC of *concept names*, the set NI of *individual names*. A *concept* H is a formula of the kind:

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

where $C \in$ NC and $R \in$ NR. Let Var be a denumerable set of *individual variables*, our calculus works on *formulas* K of $\mathcal{L}$ defined according to the following grammar:

$$K \quad ::= \quad \bot \mid (s,t):R \mid (s,t):\neg R \mid t:H \mid \forall H$$

where $s,t \in$ NI $\cup$ Var, $R \in$ NR and H is a concept. We remark that variables, that usually are not used in description logic formalization, are useful to put in evidence the "parameters" of natural deduction proofs. An *atomic formula* of $\mathcal{L}$ is a formula of the kind $\bot$, $(s,t):R$, $t:C$, with C a concept name; a *negated formula* is a formula of the kind $(s,t):\neg R$ or $t:\neg H$. A formula is *closed* if it does not contain variables. We write $\neg((s,t):R)$, $\neg((s,t):\neg R)$, $\neg(t:H)$ as abbreviations for $(s,t):\neg R$, $(s,t):R$, $t:\neg H$ respectively; $A \sqsubseteq B$ stands for $\forall(\neg A \sqcup B)$.

A *model (interpretation)* $\mathcal{M}$ for $\mathcal{L}$ is a pair $(\mathcal{D}^{\mathcal{M}}, \cdot^{\mathcal{M}})$, where $\mathcal{D}^{\mathcal{M}}$ is a non-empty set (the *domain* of $\mathcal{M}$) and $\cdot^{\mathcal{M}}$ is a *valuation* map such that: for every $c \in$ NI, $c^{\mathcal{M}} \in \mathcal{D}^{\mathcal{M}}$; for every $C \in$ NC, $C^{\mathcal{M}} \subseteq \mathcal{D}^{\mathcal{M}}$; for every $R \in$ NR, $R^{\mathcal{M}} \subseteq \mathcal{D}^{\mathcal{M}} \times \mathcal{D}^{\mathcal{M}}$. A non atomic concept H is interpreted by a subset $H^{\mathcal{M}}$ of $\mathcal{D}^{\mathcal{M}}$:

$$(\neg A)^{\mathcal{M}} = \mathcal{D}^{\mathcal{M}} \setminus A^{\mathcal{M}} \quad (A \sqcap B)^{\mathcal{M}} = A^{\mathcal{M}} \cap B^{\mathcal{M}} \quad (A \sqcup B)^{\mathcal{M}} = A^{\mathcal{M}} \cup B^{\mathcal{M}}$$

$$(\exists R.A^{\mathcal{M}}) \;=\; \{\, d \in \mathcal{D}^{\mathcal{M}} \mid \text{there is } d' \in \mathcal{D}^{\mathcal{M}} \text{ s.t. } (d,d') \in R^{\mathcal{M}} \text{ and } d' \in A^{\mathcal{M}} \}$$

$$(\forall R.A)^{\mathcal{M}} \;=\; \{\, d \in \mathcal{D}^{\mathcal{M}} \mid \text{for all } d' \in \mathcal{D}^{\mathcal{M}}, (d,d') \in R^{\mathcal{M}} \text{ implies } d' \in A^{\mathcal{M}} \}$$

An *assignment* on a model $\mathcal{M}$ is a map $\theta : \text{Var} \to \mathcal{D}^{\mathcal{M}}$. If $t \in$ NI $\cup$ 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 $t \in$ Var; $t^{\mathcal{M},\theta} = t^{\mathcal{M}}$ if $t \in$ NI. A formula K is *valid* in $\mathcal{M}$ w.r.t. θ, and we write $\mathcal{M}, \theta \models K$, if $K \neq \bot$ and one of the following conditions holds:

$$\mathcal{M}, \theta \models (s,t):R \;\text{ iff }\; (s^{\mathcal{M},\theta}, t^{\mathcal{M},\theta}) \in R^{\mathcal{M}} \qquad \mathcal{M}, \theta \models t:H \;\text{ iff }\; t^{\mathcal{M},\theta} \in H^{\mathcal{M}}$$

$$\mathcal{M}, \theta \models (s,t):\neg R \;\text{ iff }\; (s^{\mathcal{M},\theta}, t^{\mathcal{M},\theta}) \notin R^{\mathcal{M}} \quad \mathcal{M}, \theta \models \forall H \;\text{ iff }\; H^{\mathcal{M}} = \mathcal{D}^{\mathcal{M}}$$

We write $\mathcal{M} \models K$ iff $\mathcal{M}, \theta \models K$ for every assignment θ. Note that $\mathcal{M} \models \forall H$ iff $\mathcal{M} \models x : H$, with x any variable. 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 $\mathcal{M}, \theta \models K$.

Now, we introduce *information terms*, that will be the base structure of our constructive semantics. Let $\mathcal{N}$ be a finite subset of NI. By $\mathcal{L}_{\mathcal{N}}$ we denote the set of formulas K of $\mathcal{L}$ such that all the individual names occurring in K belong to $\mathcal{N}$. Given a closed formula K of $\mathcal{L}_{\mathcal{N}}$, we define the set of information terms

$\text{IT}_{\mathcal{N}}(K)$ by induction on K as follows.

$\text{IT}_{\mathcal{N}}(K) = \{\texttt{tt}\}$, if K is an atomic or negated formula

$\text{IT}_{\mathcal{N}}(c : A \sqcap B) = \{\, (\alpha, \beta) \mid \alpha \in \text{IT}_{\mathcal{N}}(c : A) \text{ and } \beta \in \text{IT}_{\mathcal{N}}(c : B) \,\}$

$\text{IT}_{\mathcal{N}}(c : A_1 \sqcup A_2) = \{\, (k, \alpha) \mid k \in \{1, 2\} \text{ and } \alpha \in \text{IT}_{\mathcal{N}}(c : A_k) \,\}$

$\text{IT}_{\mathcal{N}}(c : \exists R.A) = \{\, (d, \alpha) \mid d \in \mathcal{N} \text{ and } \alpha \in \text{IT}_{\mathcal{N}}(d : A) \,\}$

$\text{IT}_{\mathcal{N}}(c : \forall R.A) = \text{IT}_{\mathcal{N}}(\forall A) = \{\, \phi : \mathcal{N} \to \bigcup_{d \in \mathcal{N}} \text{IT}_{\mathcal{N}}(d : A) \mid \phi(d) \in \text{IT}_{\mathcal{N}}(d : A) \,\}$

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

$\mathcal{M} \rhd \langle \texttt{tt} \rangle K$ iff $\mathcal{M} \models K$, where K is an atomic or negated formula

$\mathcal{M} \rhd \langle (\alpha, \beta) \rangle c : A \sqcap B$ iff $\mathcal{M} \rhd \langle \alpha \rangle c : A$ and $\mathcal{M} \rhd \langle \beta \rangle c : B$

$\mathcal{M} \rhd \langle (k, \alpha) \rangle c : A_1 \sqcup A_2$ iff $\mathcal{M} \rhd \langle \alpha \rangle c : A_k$

$\mathcal{M} \rhd \langle (d, \alpha) \rangle c : \exists R.A$ iff $\mathcal{M} \models (c, d) : R$ and $\mathcal{M} \rhd \langle \alpha \rangle d : A$

$\mathcal{M} \rhd \langle \phi \rangle c : \forall R.A$ iff $\mathcal{M} \models c : \forall R.A$ and, for every $d \in \mathcal{N}$,
$\qquad\qquad \mathcal{M} \models (c, d) : R$ implies $\mathcal{M} \rhd \langle \phi(d) \rangle d : A$

$\mathcal{M} \rhd \langle \phi \rangle \forall A$ iff $\mathcal{M} \models \forall A$ and, for every $d \in \mathcal{N}, \mathcal{M} \rhd \langle \phi(d) \rangle d : A$

If Γ is a set of closed formulas $\{K_1, \ldots, K_n\}$ of $\mathcal{L}_{\mathcal{N}}$, $\text{IT}_{\mathcal{N}}(\Gamma)$ denotes the set of n-tuples $\overline{\eta} = (\eta_1, \ldots, \eta_n)$ such that, for every $1 \leq j \leq n$, $\eta_j \in \text{IT}_{\mathcal{N}}(K_j)$; $\mathcal{M} \rhd \langle \overline{\eta} \rangle \Gamma$ iff, for every $1 \leq j \leq n$, $\mathcal{M} \rhd \langle \eta_j \rangle K_j$.

We remark that $\mathcal{M} \rhd \langle \eta \rangle K$ implies $\mathcal{M} \models K$, hence the constructive semantics is compatible with the usual classical one. The converse in general does not hold and stronger conditions are required:

Proposition 1. *Let K be a closed formula of $\mathcal{L}$ and let $\mathcal{M}$ be a finite model for $\mathcal{L}$. If $\mathcal{M} \models K$, there exists a finite subset $\mathcal{N}$ of $\texttt{NI}$ and $\eta \in \text{IT}_{\mathcal{N}}(K)$ such that $\mathcal{M} \rhd \langle \eta \rangle K$.*

We point out that in our setting negation has a classical meaning, thus negated formulas are not constructively explained by an information term. However, how we will discuss in future works, information terms semantics can be extended to treat various kinds of constructive negation as those discussed in [6].

In the following example, we show how an information term provides all the information needed to "constructively" explain the meaning of a formula.

Example 1. Let us consider the knowledge base, inspired to the classical example of [2], consisting of the Tbox $\mathcal{T}$

$$(Ax_1) : \forall(\neg\texttt{FOOD} \sqcup \exists\texttt{goesWith.COLOR}) \equiv \texttt{FOOD} \sqsubseteq \exists\texttt{goesWith.COLOR}$$

$$(Ax_2) : \forall(\neg\texttt{COLOR} \sqcup \exists\texttt{isColorOf.WINE}) \equiv \texttt{COLOR} \sqsubseteq \exists\texttt{isColorOf.WINE}$$

and the Abox $\mathcal{A}$

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

Let WNI be the set of individual names occurring in $\mathcal{A}$. An element of $\text{IT}_{\text{WNI}}(Ax_1)$ is a function ϕ mapping each $c \in \text{WNI}$ to an element $\delta \in \text{IT}_{\text{WNI}}(c : \neg\text{FOOD}\sqcup\exists\text{goesWith}.\text{COLOR})$, where either $\delta = (1, \texttt{tt})$ (intuitively, c is not a food) or $\delta = (2, (d, \texttt{tt}))$ (intuitively, d is a wine color which goes with food c). For instance, let us consider the following $\gamma_1 \in \text{IT}_{\text{WNI}}(Ax_1)$, where we enclose between square brackets the pairs $(c, \phi(c))$:

```
[ (barolo,(1,tt)), (chardonnay,(1,tt)), (red,(1,tt)), (white,(1,tt))
  (fish,(2,(white,tt))), (meat,(2,(red,tt))) ]
```

Let $\mathcal{M}_{\text{W}}$ be a model of $\mathcal{A} \cup \mathcal{T}$. One can easily check that $\mathcal{M}_{\text{W}} \rhd \langle \gamma_1 \rangle Ax_1$. Similarly, if $\gamma_2 \in \text{IT}_{\text{WNI}}(Ax_2)$ is the information term

```
[ (barolo,(1,tt)), (chardonnay,(1,tt)),(red,(2,(barolo,tt))),
  (white,(2,(chardonnay,tt))), (fish,(1,tt)), (meat,(1,tt)) ]
```

then $\mathcal{M}_{\text{W}} \rhd \langle \gamma_2 \rangle Ax_2$ as well. We conclude $\mathcal{M}_{\text{W}} \rhd \langle (\gamma_1, \gamma_2) \rangle \mathcal{T}$.

3 The natural calculus $\mathcal{ND}_{\text{c}}$

In this section we introduce a calculus $\mathcal{ND}_{\text{c}}$ for $\mathcal{ALC}$ similar to the usual natural deduction calculi for classical and intuitionistic logic (see, e.g., [9]). The rules of $\mathcal{ND}_{\text{c}}$ are given in Figure 1. We remark that we have *introduction* and *elimination* rules for all the logical constants; some rules (namely, $\sqcup E$, $\exists E$ and $\forall I$) allow to discharge some of the assumptions (we put them between square brackets). The rules $\exists E$, $\forall I$ and $\forall_U I$ need a side condition on the rule parameter to guarantee correctness. We notice that the rule $\perp E$ is intuitionistic, we will briefly discuss in the conclusions the relation with the calculus using the classical rule of *reductio ad absurdum*.

By $\pi : \Gamma \vdash K$, with Γ a set of formulas, we denote a proof of K with undischarged formulas Γ. We say that $\pi : \Gamma \vdash K$ is over $\mathcal{L}_{\mathcal{N}}$ if all the formulas occurring in the proof belong to $\mathcal{L}_{\mathcal{N}}$.

First of all, one can easily check that $\mathcal{ND}_{\text{c}}$ preserves the validity of formulas. Indeed, let $\pi : \Gamma \vdash K$ be a proof of $\mathcal{ND}_{\text{c}}$; then:

(P1). For every model $\mathcal{M}$ and 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 NI. An $\mathcal{N}$-*substitution* σ is a map $\sigma : \text{Var} \to \mathcal{N}$. We extend σ to $\mathcal{L}$ as usual: if $c \in \text{NI}$, $\sigma c = c$; for a formula K, 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 $c \in \text{NI}$, $\sigma[c/p]$ is the $\mathcal{N}$-substitution σ' such that $\sigma'(p) = c$ and $\sigma'(x) = \sigma(x)$ for $x \neq p$.

We associate with every proof $\pi : \Gamma \vdash K$ of $\mathcal{ND}_{\text{c}}$ over $\mathcal{L}_{\mathcal{N}}$ and every $\mathcal{N}$-substitution σ a function

$$\Phi_{\sigma, \mathcal{N}}^{\pi} : \text{IT}_{\mathcal{N}}(\sigma\Gamma) \to \text{IT}_{\mathcal{N}}(\sigma K)$$

that will provide the computational interpretation of π. To this aim $\Phi_{\sigma, \mathcal{N}}^{\pi}$ will be defined, by induction on the depth of π, in order to fulfill the following property:
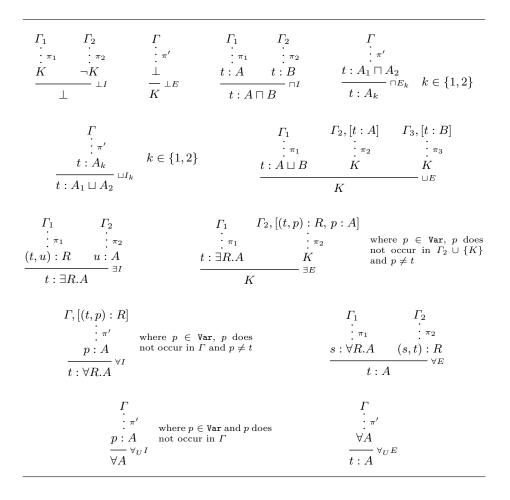
$$\dfrac{\begin{array}{cc}\begin{array}{c}\Gamma_1\\[-2pt]\vdots\,\pi_1\\[-2pt]K\end{array} & \begin{array}{c}\Gamma_2\\[-2pt]\vdots\,\pi_2\\[-2pt]\neg K\end{array}\end{array}}{\bot}\,{\bot}I \qquad \dfrac{\begin{array}{c}\Gamma\\[-2pt]\vdots\,\pi'\\[-2pt]\bot\end{array}}{K}\,{\bot}E \qquad \dfrac{\begin{array}{cc}\begin{array}{c}\Gamma_1\\[-2pt]\vdots\,\pi_1\\[-2pt]t:A\end{array} & \begin{array}{c}\Gamma_2\\[-2pt]\vdots\,\pi_2\\[-2pt]t:B\end{array}\end{array}}{t:A\sqcap B}\,{\sqcap}I \qquad \dfrac{\begin{array}{c}\Gamma\\[-2pt]\vdots\,\pi'\\[-2pt]t:A_1\sqcap A_2\end{array}}{t:A_k}\,{\sqcap}E_k \quad k\in\{1,2\}$$

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

$$\dfrac{\begin{array}{cc}\begin{array}{c}\Gamma_1\\[-2pt]\vdots\,\pi_1\\[-2pt](t,u):R\end{array} & \begin{array}{c}\Gamma_2\\[-2pt]\vdots\,\pi_2\\[-2pt]u:A\end{array}\end{array}}{t:\exists R.A}\,{\exists}I \qquad\qquad \dfrac{\begin{array}{cc}\begin{array}{c}\Gamma_1\\[-2pt]\vdots\,\pi_1\\[-2pt]t:\exists R.A\end{array} & \begin{array}{c}\Gamma_2,[(t,p):R,\,p:A]\\[-2pt]\vdots\,\pi_2\\[-2pt]K\end{array}\end{array}}{K}\,{\exists}E$$
where $p\in\mathrm{Var}$, p does not occur in $\Gamma_2\cup\{K\}$ and $p\neq t$

$$\dfrac{\begin{array}{c}\Gamma,[(t,p):R]\\[-2pt]\vdots\,\pi'\\[-2pt]p:A\end{array}}{t:\forall R.A}\,{\forall}I \qquad\qquad \dfrac{\begin{array}{cc}\begin{array}{c}\Gamma_1\\[-2pt]\vdots\,\pi_1\\[-2pt]s:\forall R.A\end{array} & \begin{array}{c}\Gamma_2\\[-2pt]\vdots\,\pi_2\\[-2pt](s,t):R\end{array}\end{array}}{t:A}\,{\forall}E$$
where $p\in\mathrm{Var}$, p does not occur in Γ and $p\neq t$

$$\dfrac{\begin{array}{c}\Gamma\\[-2pt]\vdots\,\pi'\\[-2pt]p:A\end{array}}{\forall A}\,{\forall}_U I \qquad\qquad \dfrac{\begin{array}{c}\Gamma\\[-2pt]\vdots\,\pi'\\[-2pt]\forall A\end{array}}{t:A}\,{\forall}_U E$$
where $p\in\mathrm{Var}$ and p does not occur in Γ

Fig. 1. The rules of the calculus $\mathcal{ND}_c$

(P2). For every model $\mathcal{M}$ and $\overline{\gamma}\in\mathrm{IT}_\mathcal{N}(\sigma\Gamma)$, $\mathcal{M}\rhd\langle\overline{\gamma}\rangle\,\sigma\Gamma$ 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 $\mathrm{IT}_\mathcal{N}(\sigma K)$. Otherwise, π is obtained by applying a rule r of Figure 1 to some subproofs:

(1) $r={\bot}I$. Then, $\Phi^\pi_{\sigma,\mathcal{N}}(\overline{\gamma}_1,\overline{\gamma}_2)=\mathtt{tt}$.
(2) $r={\bot}E$. 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 any element of $\mathrm{IT}_\mathcal{N}(K)$ (for the definiteness of $\Phi^\pi_{\sigma,\mathcal{N}}$, one has to assume that, for every $K\in\mathcal{L}_\mathcal{N}$, an element $\eta^+\in\mathrm{IT}_\mathcal{N}(K)$ is defined).
(3) $r={\sqcap}I$. 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

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

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

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

where Pro_k is the k-projection function.

(5) $r = \sqcup I_k$ $(k \in \{1,2\})$. 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

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

(6) $r = \sqcup E$. Then, $\Phi^\pi_{\sigma,\mathcal{N}} : \mathrm{IT}_\mathcal{N}(\sigma\Gamma_1) \times \mathrm{IT}_\mathcal{N}(\sigma\Gamma_2) \times \mathrm{IT}_\mathcal{N}(\sigma\Gamma_3) \to \mathrm{IT}_\mathcal{N}(\sigma K)$ and

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

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

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

(8) $r = \exists E$. Then, $\Phi^\pi_{\sigma,\mathcal{N}} : \mathrm{IT}_\mathcal{N}(\sigma\Gamma_1) \times \mathrm{IT}_\mathcal{N}(\sigma\Gamma_2) \to \mathrm{IT}_\mathcal{N}(\sigma K)$ and

$$\Phi^\pi_{\sigma,\mathcal{N}}(\overline{\gamma}_1,\overline{\gamma}_2) \;=\; \Phi^{\pi_2}_{\sigma[c/p],\mathcal{N}}(\overline{\gamma}_2,\,\mathtt{tt},\,\alpha)$$

where $(c,\alpha) \;=\; \Phi^{\pi_1}_{\sigma,\mathcal{N}}(\overline{\gamma}_1)^1$.

(9) $r = \forall I$. Then, $\Phi^\pi_{\sigma,\mathcal{N}} : \mathrm{IT}_\mathcal{N}(\sigma\Gamma) \to \mathrm{IT}_\mathcal{N}(\sigma t : \forall R.A)$ and[2]

$$\left[\Phi^\pi_{\sigma,\mathcal{N}}(\overline{\gamma})\right](c) \;=\; \Phi^{\pi'}_{\sigma[c/p],\mathcal{N}}(\overline{\gamma},\,\mathtt{tt}) \quad \text{for every } c \in \mathcal{N}$$

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

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

(11) $r = \forall_U I$. Analogous to the case $r = \forall I$.

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

Theorem 1 (Soundness). *Let $\mathcal{N}$ be a finite subset of $\mathtt{NI}$ and let $\pi : \Gamma \vdash K$ be a proof of $\mathcal{ND}_\mathrm{c}$ over $\mathcal{L}_\mathcal{N}$ such that the formulas in $\Gamma \cup \{K\}$ are closed. Then:*

(i) $\Gamma \models K$.

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

To conclude this section we give an example of the information one can extract from a proof using Theorem 1.

---

[1] We remark that, by the side condition on p, $(\sigma[c/p])\Gamma_2 = \sigma\Gamma_2$ and $(\sigma[c/p])K = \sigma K$.

[2] By the side condition on p, $(\sigma[c/p])\Gamma = \sigma\Gamma$ and $(\sigma[c/p])t : \forall R.A = \sigma t : \forall R.A$.

Example 2. Let us consider the knowledge base of Example 1. We can build a proof

$$\pi : \mathcal{T} \ \vdash \ \forall(\neg\texttt{FOOD} \sqcup \exists\texttt{goesWith}.(\texttt{COLOR} \sqcap \exists\texttt{isColorOf}.\texttt{WINE}))$$

in $\mathcal{ND}_c$, namely a proof of $\texttt{FOOD} \sqsubseteq \exists\texttt{goesWith}.(\texttt{COLOR} \sqcap \exists\texttt{isColorOf}.\texttt{WINE})$ from $\mathcal{T}$. The proof π is

$$
\cfrac{
\cfrac{
\cfrac{\quad Ax_1 \quad}{y : \neg\texttt{FOOD} \sqcup \exists\texttt{goesWith}.\texttt{COLOR}}\ {\scriptstyle\forall_U E}
\qquad
\cfrac{[y : \neg\texttt{FOOD}]}{K}\ {\scriptstyle\sqcup I}
\qquad
\cfrac{Ax_2 \ [y : \exists\texttt{goesWith}.\texttt{COLOR}] \atop \vdots\ \pi_1}{K}
}{K \equiv y : \neg\texttt{FOOD} \sqcup \exists\texttt{goesWith}.(\texttt{COLOR} \sqcap \exists\texttt{isColorOf}.\texttt{WINE})}\ {\scriptstyle\sqcup E}
}{\forall(\neg\texttt{FOOD} \sqcup \exists\texttt{goesWith}.(\texttt{COLOR} \sqcap \exists\texttt{isColorOf}.\texttt{WINE}))}\ {\scriptstyle\forall_U I}
$$

where π_1 is the proof

$$
\cfrac{
\cfrac{
[(y,z) : \texttt{goesWith}]
\qquad
\cfrac{z : \texttt{COLOR} \sqcap \exists\texttt{isColorOf}.\texttt{WINE} \qquad {Ax_2 \ [z : \texttt{COLOR}] \atop \vdots\ \pi_2}}{y : \exists\texttt{goesWith}.(\texttt{COLOR} \sqcap \exists\texttt{isColorOf}.\texttt{WINE})}\ {\scriptstyle\exists I}
}{
\cfrac{y : \exists\texttt{goesWith}.\texttt{COLOR} \qquad\qquad y : \exists\texttt{goesWith}.(\texttt{COLOR} \sqcap \exists\texttt{isColorOf}.\texttt{WINE})}{y : \exists\texttt{goesWith}.(\texttt{COLOR} \sqcap \exists\texttt{isColorOf}.\texttt{WINE})}\ {\scriptstyle\exists E}
}
}{y : \neg\texttt{FOOD} \sqcup \exists\texttt{goesWith}.(\texttt{COLOR} \sqcap \exists\texttt{isColorOf}.\texttt{WINE})}\ {\scriptstyle\sqcup I}
$$

and π_2 is the proof

$$
\cfrac{
\cfrac{Ax_2}{z : \neg\texttt{COLOR} \sqcup \exists\texttt{isColorOf}.\texttt{Wine}}\ {\scriptstyle\forall_U E}
\quad
\cfrac{\cfrac{z : \texttt{COLOR} \ [z : \neg\texttt{COLOR}]}{\bot}\ {\scriptstyle\bot I}}{H}\ {\scriptstyle\bot E}
\quad
\cfrac{z : \texttt{COLOR} \ [z : \exists\texttt{isColorOf}.\texttt{WINE}]}{H}\ {\scriptstyle\sqcap I}
}{H \equiv z : \texttt{COLOR} \sqcap \exists\texttt{isColorOf}.\texttt{WINE}}\ {\scriptstyle\sqcup E}
$$

Note that individual names do not occur in π. Let $\mathcal{M}_\texttt{W}$, γ_1 and γ_2 be defined as in Example 1. Since $\mathcal{M}_\texttt{W} \triangleright \langle(\gamma_1, \gamma_2)\rangle \mathcal{T}$, by Theorem 1 we get that $\Phi^\pi_\texttt{WNI}(\gamma_1, \gamma_2)$ is a function ψ such that, for every $c \in \texttt{WNI}$:

$$\mathcal{M}_\texttt{W} \triangleright \langle\psi(c)\rangle c : \neg\texttt{FOOD} \sqcup \exists\texttt{goesWith}.(\texttt{COLOR} \sqcap \exists\texttt{isColorOf}.\texttt{WINE})$$

If $\psi(c) = (1, \texttt{tt})$, then $c^{\mathcal{M}_\texttt{W}} \notin \texttt{FOOD}^{\mathcal{M}_\texttt{W}}$ (c is not a food). Otherwise, $\psi(c)$ has the form $(2, (d, (\texttt{tt}, (e, \texttt{tt}))))$, meaning that $(c^{\mathcal{M}_\texttt{W}}, d^{\mathcal{M}_\texttt{W}}) \in \texttt{goesWith}^{\mathcal{M}_\texttt{W}}$ (food c goes with color d) and $(d^{\mathcal{M}_\texttt{W}}, e^{\mathcal{M}_\texttt{W}}) \in \texttt{isColorOf}^{\mathcal{M}_\texttt{W}}$ (wine e has color d), hence we have found a wine e to pair with c. In our example we get

$$
\begin{aligned}
\psi(\texttt{meat}) &= (2, (\texttt{red}, (\texttt{tt}, (\texttt{barolo}, \texttt{tt})))) \\
\psi(\texttt{fish}) &= (2, (\texttt{white}, (\texttt{tt}, (\texttt{chardonnay}, \texttt{tt}))))
\end{aligned}
$$

and $\psi(c) = (1, \texttt{tt})$ for all the other $c \in \texttt{WNI}$.

Note that, since in our setting negation has not a constructive meaning, the choice of axioms is crucial to extract information. As an example, if we replace Ax_1 with the classically equivalent formula $\forall(\neg(\texttt{FOOD} \sqcap \neg\exists\texttt{goesWith}.\texttt{COLOR}))$, we cannot build a proof of the formula $\forall(\neg\texttt{FOOD} \sqcup \exists\texttt{goesWith}.(\texttt{COLOR} \sqcap \exists\texttt{isColorOf}.\texttt{WINE}))$.

To conclude this section we remark that, along the lines of the previous example, Theorem 1 allows us to interpret a proof of a "goal" as a program to solve it. We defer to a future work a deeper discussion on the notion of "solvable goal".

4 Conclusions

First of all, we compare information-terms semantics with the classical one. Let $\overline{\mathcal{ALC}}$ denote the set of formulas K such that $\mathcal{M} \models K$, and let $\overline{\mathcal{ALC}_c}$ be the set of formulas K such that there exists a proof $\pi : \vdash K$ in $\mathcal{ND}_c$. By Theorem 1, $\overline{\mathcal{ALC}_c} \subseteq \overline{\mathcal{ALC}}$. However, one can easily prove that the classically valid formula $x : D \sqcup \neg D$ is not provable in $\mathcal{ND}_c$; hence, $\overline{\mathcal{ALC}_c} \neq \overline{\mathcal{ALC}}$. We remark that in general a constructive explanation of $x : D \sqcup \neg D$ cannot be given. If we replace the rule $\bot E$ of $\mathcal{ND}_c$ with the classical rule of *reductio ad absurdum*, the set of provable formulas of the resulting calculus coincides with $\overline{\mathcal{ALC}}$; obviously, the computational interpretation of proofs provided by Theorem 1 cannot be extended to such a rule. Finally, we remark that our constructive semantics and $\mathcal{ND}_c$ can be exploited to handle intuitionistic implication and stronger negation (as discussed in [6]). As for future works, we are developing an extension of $\mathcal{ND}_c$ sound and complete with respect to the information-terms semantics for $\mathcal{ALC}$. Moreover, we plan to extend our framework to treat other description logics.

References

1. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.
2. R. J. Brachman, D. L. Mcguinness, P. F. Patel-Schneider, L. A. Resnick, and A. Borgida. Living with CLASSIC: When and how to use a KL-ONE-like language, 1991.
3. V. de Paiva. Constructive description logics: what, why and how. Technical report, Xerox Parc, 2005.
4. M. Ferrari, C. Fiorentini, and M. Ornaghi. Extracting exact time bounds from logical proofs. In A. Pettorossi, editor, *LOPSTR 2001*, volume 2372 of *LNCS*, pages 245–265. Springer-Verlag, 2002.
5. M. Hofmann. Proof-theoretic approach to description-logic. In *LICS*, pages 229–237. IEEE Computer Society, 2005.
6. K. Kaneiwa. Negations in description logic - contraries, contradictories, and subcontraries. In *ICCS'05*, pages 66–79. Kassel University Press, 2005.
7. P. Miglioli, U. Moscato, M. Ornaghi, and G. Usberti. A constructivism based on classical truth. *Notre Dame Journal of Formal Logic*, 30(1):67–90, 1989.
8. M. Ornaghi, M.Benini, M. Ferrari, C. Fiorentini, and A.Momigliano. A Constructive Modeling Language for Object Oriented Information Systems. In *CLASE'05*, volume 153 of *ENTCS*, pages 55–75, 2006.
9. D. Prawitz. *Natural Deduction*. Almquist and Winksell, 1965.
10. M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.
11. A. S. Troelstra. From constructivism to computer science. *TCS*, 211(1-2):233–252, 1999.