A Decidable Constructive Description Logic

Loris Bozzato[1], Mauro Ferrari[1], Camillo Fiorentini[2], and 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. Recently, there has been a growing interest in constructive reinterpretations of description logics. This has been motivated by the need to model in the DLs setting problems that have a consolidate tradition in constructive logics. In this paper we introduce a constructive description logic for the language of $\mathcal{ALC}$ based on the Kripke semantics for Intuitionistic Logic. Moreover we give a tableau calculus and we show that it is sound, complete and terminating.

1 Introduction

Nowadays Description Logics (DLs) are the most prominent formalism for Knowledge Representation. Their success depends on the one side on their "natural" classical semantics and their expressivity and on the other side on the decidability and efficiency of the reasoning problems. However, in recent years there has been a growing interest in different interpretations of DLs allowing one to model knowledge domains and problems that can hardly be treated in the context of a classical semantics. Among these, we recall some proposals towards a constructive approach to DLs: paraconsistent versions of DLs [14], different interpretations of negation [12], the introduction of semantics supporting a computational reading of proofs [3,4,8] and the characterization of incomplete information and typing systems for data streams [13].

As pointed out in [13], in general a constructive reading of DLs is useful in domains with possibly dynamic and incomplete knowledge. This view is supported by the model theoretical features of constructive semantics which allows the representation of stages of information and truth evidence. Following this line, in this paper we introduce the constructive description logic $\mathcal{KALC}$. This logic is based on the same language of $\mathcal{ALC}$ and relies on a Kripke-style semantics which consists in a reformulation in the DLs setting of the Kripke semantics for first-order Intuitionistic Logic. A Kripke model can be considered as a set of worlds, representing states of knowledge, partially ordered by their information content. This permits to express partial and incomplete states of knowledge which can increase in time in the context of the Open World Assumption. We exemplify this in Section 2 by means of an example inspired by [13]. The main difference between our semantics and the one of [13] is that our refinement relation concerns a whole state of knowledge and not a single individual. This provides a

T. Janhunen and I. Niemelä (Eds.): JELIA 2010, LNAI 6341, pp. 51–63, 2010.

semantics which seems a "natural" generalization of the classical semantics for $\mathcal{ALC}$. Reasoning problems for DLs in our setting are formulated as in $\mathcal{ALC}$ but have a constructive meaning. Indeed, as we prove in Section 2, $\mathcal{KALC}$ meets the *disjunction property*: if the assertion $c : C \sqcup D$ (c belongs to concept $C \sqcup D$) holds in $\mathcal{KALC}$, then either $c : C$ or $c : D$ is true in $\mathcal{KALC}$. In particular, the classically valid assertion $c : C \sqcup \neg C$ is not true in $\mathcal{KALC}$. This is considered an essential feature of every constructive system. We show that reasoning problems for $\mathcal{KALC}$ are decidable by introducing a tableau calculus $\mathcal{T}_{\mathcal{K}}$ for $\mathcal{KALC}$. To conclude, we notice that our semantics can be viewed as a refinement of $i\mathcal{ALC}$ [6], which corresponds to a direct translation of semantics of Intuitionistic first order logics in the language of $\mathcal{ALC}$.

In the following section we first introduce the syntax and semantics of $\mathcal{KALC}$, discussing its relations with classical semantics and the disjunction property. In Section 3 we introduce the calculus $\mathcal{T}_{\mathcal{K}}$ and we prove its soundness with respect to $\mathcal{KALC}$ semantics: completeness and termination are presented in Section 4. We conclude in Section 5 by considering possible directions for future work.

2 Syntax and Semantics

We begin by introducing the language $\mathcal{L}$ of $\mathcal{KALC}$ which essentially coincides with the one of $\mathcal{ALC}$ [7]. It is based on the following denumerable sets: the set $\mathtt{NI}$ of *individual names*, the set $\mathtt{NC}$ of *atomic concept names*, the set $\mathtt{NR}$ of *role names*. *Concepts* C,D and *formulas* H are defined as follows:

$$
\begin{aligned}
C, D &::= \bot \mid A \mid C \sqcap D \mid C \sqcup D \mid C \to D \mid \exists R.C \mid \forall R.C \\
H &::= (c, d) : R \mid c : C \mid C \sqsubseteq D
\end{aligned}
$$

where $c, d \in \mathtt{NI}$, $A \in \mathtt{NC}$ and $R \in \mathtt{NR}$. As usual in constructive logics, we write $\neg C$ as an abbreviation for $C \to \bot$. Given $\mathcal{N} \subseteq \mathtt{NI}$, we denote with $\mathcal{L}_{\mathcal{N}}$ the language only containing individual names from $\mathcal{N}$. A *(classical) model* $\mathcal{M}$ for $\mathcal{L}_{\mathcal{N}}$ is a pair $(\mathcal{D}^{\mathcal{M}}, \cdot^{\mathcal{M}})$, where $\mathcal{D}^{\mathcal{M}}$ is a finite non-empty set (the *domain* of $\mathcal{M}$) and $\cdot^{\mathcal{M}}$ is a *valuation* map such that: for every $c \in \mathcal{N}$, $c^{\mathcal{M}} \in \mathcal{D}^{\mathcal{M}}$; for every $A \in \mathtt{NC}$, $A^{\mathcal{M}} \subseteq \mathcal{D}^{\mathcal{M}}$; $\bot^{\mathcal{M}}$ is the empty set; for every $R \in \mathtt{NR}$, $R^{\mathcal{M}} \subseteq \mathcal{D}^{\mathcal{M}} \times \mathcal{D}^{\mathcal{M}}$. A *Kripke model* for $\mathcal{L}_{\mathcal{N}}$ is a quadruple $\underline{K} = \langle P, \leq, \rho, \iota \rangle$, where:

- $(P, \leq)$ is a finite poset with minimum element ρ; P is the set of *worlds* of $\underline{K}$, ρ the *root* of $\underline{K}$.
- ι is a function associating with every world α a model $(\mathcal{D}^{\alpha}, \cdot^{\alpha})$ for $\mathcal{L}_{\mathcal{N}}$ such that, for every $\alpha \leq \beta$, the following holds:

  (K1) $\mathcal{D}^{\alpha} \subseteq \mathcal{D}^{\beta}$;
  (K2) for every $c \in \mathcal{N}$, $c^{\beta} = c^{\alpha}$;
  (K3) for every $A \in \mathtt{NC}$, $A^{\alpha} \subseteq A^{\beta}$;
  (K4) for every $R \in \mathtt{NR}$, $R^{\alpha} \subseteq R^{\beta}$.

Worlds of $\underline{K}$ represent states of knowledge that can be updated or refined by the relation $\leq$; conditions (K1)–(K4) settle that the knowledge is monotonic. Given

$\underline{K} = \langle P, \leq, \rho, \iota \rangle$ and $\alpha \in P$, we denote with $\mathcal{L}_\alpha$ the language obtained by adding to the individual names of $\mathcal{L}_\mathcal{N}$ every element $d \in \mathcal{D}^\alpha$ and setting $d^\alpha = d$. Note that, by Condition (K1), $\alpha \leq \beta$ implies $\mathcal{L}_\alpha \subseteq \mathcal{L}_\beta$. Let α be a world of $\underline{K}$ and H a formula of $\mathcal{L}_\alpha$; we inductively define the *forcing relation* $\alpha \Vdash H$ as follows:

- $\alpha \Vdash c : A$, where $A \in \mathtt{NC}$ or $A = \bot$, iff $c^\alpha \in A^\alpha$;
- $\alpha \Vdash (c, d) : R$ where $R \in \mathtt{NR}$, iff $(c^\alpha, d^\alpha) \in R^\alpha$;
- $\alpha \Vdash c : C \sqcap D$ iff $\alpha \Vdash c : C$ and $\alpha \Vdash c : D$;
- $\alpha \Vdash c : C \sqcup D$ iff $\alpha \Vdash c : C$ or $\alpha \Vdash c : D$;
- $\alpha \Vdash c : C \to D$ iff, for every $\beta \geq \alpha$, $\beta \Vdash c : C$ implies $\beta \Vdash c : D$;
- $\alpha \Vdash c : \exists R.C$ iff there is $d \in \mathcal{D}^\alpha$ such that $\alpha \Vdash (c, d) : R$ and $\alpha \Vdash d : C$;
- $\alpha \Vdash c : \forall R.C$ iff, for every $\beta \geq \alpha$ and $d \in \mathcal{D}^\beta$, $\beta \Vdash (c, d) : R$ implies $\beta \Vdash d : C$;
- $\alpha \Vdash C \sqsubseteq D$ iff, for every $\beta \geq \alpha$ and $c \in \mathcal{D}^\beta$, $\beta \Vdash c : C$ implies $\beta \Vdash c : D$.

We remark that, differently from $\mathcal{ALC}$, the logical connectives are not interdefinable; e.g., $C \to D$ is not equivalent to $\neg C \sqcup D$. As for negation, being $\neg C$ an abbreviation for $C \to \bot$, we get $\alpha \Vdash c : \neg C$ iff for every $\beta \geq \alpha$, $\beta \nVdash c : C$.

By conditions (K1)–(K4) the forcing relation satisfies the *monotonicity property*: if $\alpha \Vdash H$ then $\beta \Vdash H$ for every $\beta \geq \alpha$. A *final* world ϕ of $\underline{K}$ is a maximal element of $(P, \leq)$. Note that $\phi \nVdash H$ implies $\phi \Vdash \neg H$. As a consequence, in ϕ any formula $c : C \sqcup \neg C$ is valid, as in classical models for $\mathcal{ALC}$, hence a final world represents a state of complete knowledge.

We now introduce the notion of *$\mathcal{KALC}$-logical consequence*, denoted by $\models$, which allows us to represent in $\mathcal{KALC}$ the usual inference problems for DLs. Let $\mathcal{F}$ be a set of formulas; by $\alpha \Vdash \mathcal{F}$ we mean that $\alpha \Vdash H$ for every $H \in \mathcal{F}$. Given a formula H, the relation $\mathcal{F} \models H$ holds iff:

- for every $\underline{K} = \langle P, \leq, \rho, \iota \rangle$ and $\alpha \in P$, if $\alpha \Vdash \mathcal{F}$ then $\alpha \Vdash H$.

By the above discussion, it follows that if $\mathcal{F} \models H$ then H is a logical consequence of $\mathcal{F}$ as understood in $\mathcal{ALC}$. Thus, $\mathcal{KALC}$-logical consequence refines the corresponding notion for $\mathcal{ALC}$.

In our setting, an ABox $\mathcal{A}$ is a set of assertions of the kind $(c, d) : R$ and $c : C$, a TBox $\mathcal{T}$ is a set of inclusions of the form $C \sqsubseteq D$. The inference problems for $\mathcal{KALC}$ are formulated as in $\mathcal{ALC}$, using the $\models$ relation. To exemplify:

- *Concept satisfiability.* C is satisfiable w.r.t. $(\mathcal{A}, \mathcal{T})$ iff $\mathcal{A} \cup \mathcal{T} \cup \{q : C\} \not\models c : \bot$, with q not occurring in $\mathcal{A}$.
- *Instance checking.* $c : C$ is entailed by $(\mathcal{A}, \mathcal{T})$ iff $\mathcal{A} \cup \mathcal{T} \models c : C$.
- *Subsumption.* D subsumes C w.r.t. $(\mathcal{A}, \mathcal{T})$ iff $\mathcal{A} \cup \mathcal{T} \models C \sqsubseteq D$.

In the next example we show how our semantics allows us to represent partial and incomplete information and supports constructive reasoning.

Example 1 (Auditing). We reconsider the auditing example of [13]. Let us suppose to have a knowledge base defined by the following ABox $\mathcal{A}$ (`CW` stands for "Credit Worthy"):

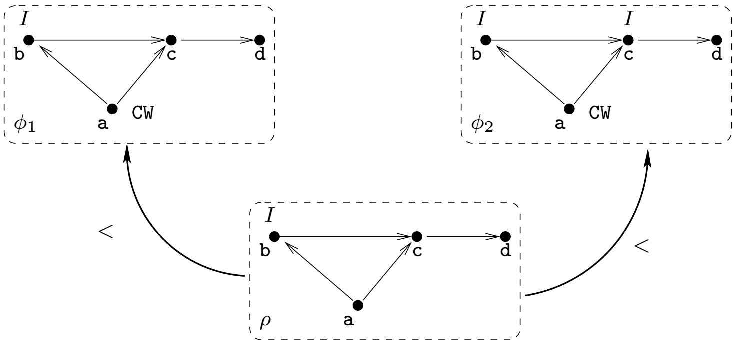

Fig. 1. The model $\underline{K}$

a:Company	d:Company	(a,b):hasCustomer	(c,d):hasCustomer
b:Company	b:Insolvent	(a,c):hasCustomer	a:$D \rightarrow$ CW
c:Company	d:¬Insolvent	(b,c):hasCustomer	

where D is the concept ∃hasCustomer.(Insolvent ⊓ ∃hasCustomer.¬Insolvent)

The formula a : $D \rightarrow$ CW states that if the company a has an insolvent customer *solv* which in turn can rely on at least one non-insolvent customer *unsolv*, then a can be trusted as CW (here and in Fig. 1 I abbreviates Insolvent). In $\mathcal{ALC}$ the assertion a:CW is entailed by $\mathcal{A}$. To prove this, let $\mathcal{M}$ be any model of $\mathcal{A}$. We first show that a : D holds in $\mathcal{M}$. Since *tertium non datur* classically holds, in $\mathcal{M}$ the customer c is either insolvent or not insolvent. Let us consider the worlds ϕ_2 and ϕ_1 in Fig. 1 representing the two possibilities; in both cases, we can find out the clients *solv* and *unsolv* required by D: *solv* = c and *unsolv* = d in the former case, *solv* = b and *unsolv* = c in the latter. Since a : $D \rightarrow$ CW holds in $\mathcal{M}$, it follows that a : CW holds in $\mathcal{M}$. Thus, in $\mathcal{ALC}$ the company a is trusted as CW though we have not any knowledge about the identity of the customers *solv* and *unsolv*.

On the contrary, in $\mathcal{KALC}$ the information in $\mathcal{A}$ does not enable to assert that a is CW. The point is that we have not enough knowledge on c, thus neither "c is insolvent" nor "c is not insolvent" can be asserted (indeed, in a future world c might become insolvent). This can be formalized in $\mathcal{KALC}$ semantics as follows. Let $\mathcal{N} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ and let us consider the Kripke model $\underline{K} = \langle P, \leq, \rho, \iota \rangle$ for $\mathcal{L}_{\mathcal{N}}$ in Fig. 1, consisting of the root ρ, two final worlds ϕ_1 and ϕ_2 such that:

- for every $\alpha \in P$, $\mathcal{D}^\alpha = \mathcal{N}$ and, for every $z \in \mathcal{N}$, $z^\alpha = z$;
- atomic concepts and roles are interpreted as follows:

World	Company	Insolvent	CW	hasCustomer
ρ	$\mathcal{N}$	{b}	$\emptyset$	{ (a, b), (a, c), (b, c), (c, d) }
ϕ_1	$\mathcal{N}$	{b}	{a}	{ (a, b), (a, c), (b, c), (c, d) }
ϕ_2	$\mathcal{N}$	{ b, c }	{a}	{ (a, b), (a, c), (b, c), (c, d) }

Since $\phi_1 \not\Vdash$ c : Insolvent (actually, $\phi_1 \Vdash$ c : ¬Insolvent) and $\phi_2 \Vdash$ c : Insolvent, we have that $\rho \not\Vdash$ c : Insolvent and $\rho \not\Vdash$ c : ¬Insolvent, hence

$\rho \not\Vdash$ $\mathtt{c : Insolvent} \sqcup \neg\mathtt{Insolvent}$. Note that, $\mathtt{b}$ and $\mathtt{c}$ are the only individuals such that $(\mathtt{a}, \mathtt{b}) \in \mathtt{hasCustomer}^\rho$, $\mathtt{b} \in \mathtt{Insolvent}^\rho$ and $(\mathtt{b}, \mathtt{c}) \in \mathtt{hasCustomer}^\rho$, but $\rho \not\Vdash \mathtt{c} : \neg\mathtt{Insolvent}$. Thus $\rho \not\Vdash \mathtt{a} : D$. Since $\phi_1 \Vdash \mathtt{a} : \mathtt{CW}$ and $\phi_2 \Vdash \mathtt{a} : \mathtt{CW}$, we have $\rho \Vdash \mathtt{a} : D \to \mathtt{CW}$. To sum up, $\rho \Vdash \mathcal{A}$ and $\rho \not\Vdash \mathtt{a} : \mathtt{CW}$; we conclude that $\mathtt{a} : \mathtt{CW}$ is not a $\mathcal{KALC}$-logical consequence of $\mathcal{A}$. Observe that the final worlds correspond to the two possible ways of acquiring a complete knowledge about the insolvency of $\mathtt{c}$: clearly, in the final worlds $\mathtt{a}$ must be $\mathtt{CW}$. $\diamond$

We conclude the discussion on Kripke semantics by remarking that $\mathcal{KALC}$ satisfies the *Disjunction Property (DP)*:

- $\models c : C_1 \sqcup C_2$ implies $\models c : C_1$ or $\models c : C_2$.

As an immediate consequence, the classically valid assertion $c : C \sqcup \neg C$ is not valid in $\mathcal{KALC}$. The proof of (DP) exploits the standard technique of gluing models: given two Kripke models $\langle P_j, \leq_j, \rho_j, \iota_j \rangle$ $(j = 1, 2)$ such that $\rho_j \not\Vdash_j c : C_j$, one can build a model $\langle P, \leq, \rho, \iota \rangle$, with $\rho \notin P_1 \cup P_2$, such that the immediate successors of ρ are ρ_1 and ρ_2. It follows that $\rho \not\Vdash c : C_1 \sqcup C_2$. Handling with care the same technique, we can prove (DP) in a more general form:

- $\mathcal{F}_H \models c : C_1 \sqcup C_2$ implies $\mathcal{F}_H \models c : C_1$ or $\mathcal{F}_H \models c : C_2$

with $\mathcal{F}_H$ a set of *Harrop Formulas* (occurrences of $\sqcup$ and $\exists$ are only allowed in the left-hand scope of $\to$ or $\sqsubseteq$).

In this paper we only consider the reasoning problems over *acyclic* TBoxes $\mathcal{T}$ according to the standard definition:

1. $\mathcal{T}$ only contains inclusions $A \sqsubseteq C$, with A an atomic concept.
2. Let us say that the atomic concept A *directly uses* A' in $\mathcal{T}$ iff, for some $A \sqsubseteq C \in \mathcal{T}$, A' is a subformula of C and let *uses* be the transitive closure of the "directly uses" relation. Then, no concept occurring in $\mathcal{T}$ uses itself.

3 The Tableau Calculus $\mathcal{T}_\mathcal{K}$

The tableau calculus $\mathcal{T}_\mathcal{K}$ works on *signed formulas* $W = \mathcal{S}(H)$, with H a formula and $\mathcal{S}$ a sign in $\{\mathbf{T}, \mathbf{F}, \mathbf{T_s}\}$. Formally:

$$W \quad ::= \quad \mathbf{T}((c,d) : R) \mid \mathbf{F}(c : C) \mid \mathbf{T}(c : C) \mid \mathbf{T_s}(c : C) \mid \mathbf{T}(C \sqsubseteq D)$$

Given a Kripke model $\underline{K} = \langle P, \leq, \rho, \iota \rangle$, a world $\alpha \in P$ and a signed formula W, α *realizes* W *in* $\underline{K}$, and we write $\underline{K}, \alpha \rhd W$, iff:

- $W = \mathbf{T}(H)$ and $\alpha \Vdash H$.
- $W = \mathbf{F}(H)$ and $\alpha \not\Vdash H$.
- $W = \mathbf{T_s}(H)$ and, for every $\beta \in P$ such that $\alpha < \beta$, $\beta \Vdash H$.

The signs $\mathbf{T}$ and $\mathbf{F}$ have the usual meaning [9], whereas $\mathbf{T}_s$ refers to the successors of a world. Let Δ be a set of signed formulas and let

$$\Delta_s \;=\; \{\, \mathbf{T}(H) \mid \mathbf{T}(H) \in \Delta \,\} \;\cup\; \{\, \mathbf{T}(H) \mid \mathbf{T_s}(H) \in \Delta \,\}$$

$$\frac{\Delta, \mathbf{T}(c : C \sqcap D)}{\Delta, \mathbf{T}(c : C), \mathbf{T}(c : D)}\,\mathbf{T}\sqcap \qquad \frac{\Delta, \mathbf{F}(c : C \sqcap D)}{\Delta, \mathbf{F}(c : C) \mid \Delta, \mathbf{F}(c : D)}\,\mathbf{F}\sqcap$$

$$\frac{\Delta, \mathbf{T}(c : C \sqcup D)}{\Delta, \mathbf{T}(c : C) \mid \Delta, \mathbf{T}(c : D)}\,\mathbf{T}\sqcup \qquad \frac{\Delta, \mathbf{F}(c : C \sqcup D)}{\Delta, \mathbf{F}(c : C), \mathbf{F}(c : D)}\,\mathbf{F}\sqcup$$

$$\frac{\Delta, \mathbf{F}(c : C \to D)}{\Delta, \mathbf{T}(c : C), \mathbf{F}(c : D) \mid \Delta_s, \mathbf{T}(c : C), \mathbf{F}(c : D)}\,\mathbf{F}\!\to$$

$$\frac{\Delta, \mathbf{T}(c : C \to D)}{\Delta, \mathbf{T}(c : D) \mid \Delta, \mathbf{F}(c : C), \mathbf{T_s}(c : D) \mid \Delta_s, \mathbf{F}(c : C), \mathbf{T_s}(c : D)}\,\mathbf{T}\!\to$$

$$\frac{\Delta, \mathbf{T}(c : A), \mathbf{T}(A \sqsubseteq C)}{\Delta, \mathbf{T}(c : A), \mathbf{T}(A \sqsubseteq C), \mathbf{T}(c : C)}\,\mathbf{T}\sqsubseteq$$

$$\frac{\Delta, \mathbf{T}(c : \exists R.C)}{\Delta, \mathbf{T}((c, q) : R), \mathbf{T}(q : C)}\,\mathbf{T}\exists^{*} \qquad \frac{\Delta, \mathbf{T}((c, d) : R), \mathbf{F}(c : \exists R.C)}{\Delta, \mathbf{T}((c, d) : R), \mathbf{F}(c : \exists R.C), \mathbf{F}(d : C)}\,\mathbf{F}\exists$$

$$\frac{\Delta, \mathbf{T}((c, d) : R), \mathbf{T}(c : \forall R.C)}{\Delta, \mathbf{T}((c, d) : R), \mathbf{T}(c : \forall R.C), \mathbf{T}(d : C)}\,\mathbf{T}\forall$$

$$\frac{\Delta, \mathbf{F}(c : \forall R.C)}{\Delta, \mathbf{T}((c, q) : R), \mathbf{F}(q : C) \mid \Delta_s, \mathbf{T}((c, q) : R), \mathbf{F}(q : C)}\,\mathbf{F}\forall^{*}$$

$^{*}q$ does not occur in the premise

Fig. 2. Rules of $\mathcal{T}_{\mathcal{K}}$

Then, $\underline{K}, \alpha \rhd \Delta$ implies $\underline{K}, \beta \rhd \Delta_s$ for every $\beta > \alpha$ ($\underline{K}, \alpha \rhd \Delta$ means $\underline{K}, \alpha \rhd W$ for every $W \in \Delta$). We also note that $\underline{K}, \alpha \rhd \mathbf{T_s}(c : \bot)$ iff α is final. We say that Δ *is realizable* if $\underline{K}, \alpha \rhd \Delta$ for some $\underline{K}$ and α. Given a set of formulas $\mathcal{F}$ and a sign $\mathcal{S}$, $\mathcal{S}(\mathcal{F})$ denotes the set of signed formulas $\mathcal{S}(H)$ such that $H \in \mathcal{F}$. The relations among realizability, $\mathcal{KALC}$-logical consequence and $\mathcal{ALC}$-logical consequence are stated by the following theorem, which can be easily proved:

Theorem 1. *Let $\mathcal{F}$ be a set of formulas and q an individual name not in $\mathcal{F}$.*

(i) *$\mathcal{F} \models c : C$ iff the set $\mathbf{T}(\mathcal{F}) \cup \{\mathbf{F}(c : C)\}$ is not realizable.*
(ii) *$\mathcal{F} \models C \sqsubseteq D$ iff the set $\mathbf{T}(\mathcal{F}) \cup \{\mathbf{F}(q : C \to D)\}$ is not realizable.*
(iii) *H is an $\mathcal{ALC}$-logical consequence of $\mathcal{F}$ iff $\mathbf{T}(\mathcal{F}) \cup \{\mathbf{T_s}(c : \bot), \mathbf{F}(H)\}$ is not realizable.*
(iv) *$c : C$ is an $\mathcal{ALC}$-logical consequence of $\mathcal{F}$ iff $\mathcal{F} \models c : \neg\neg C$.* □

The rules of the tableau calculus $\mathcal{T}_{\mathcal{K}}$ are shown in Fig. 2. In the rules we write Δ, W as a shorthand for $\Delta \cup \{W\}$; moreover, if Δ, W is the premise of a rule, we assume $W \notin \Delta$. Every rule applies to a set of signed formulas, but only acts on the signed formula W explicitly indicated in the premise. The consequence of a rule consists of one or more sets of signed formulas separated by the symbol '|'.

In the rules $\mathbf{T}\exists$ and $\mathbf{F}\forall$, q is a fresh individual name. Formulas of the kind $\mathbf{F}(c : \exists R.C)$, $\mathbf{T}(c : \forall R.C)$ and $\mathbf{T}(A \sqsubseteq C)$ must be duplicated in rule application to guarantee the completeness; we call them *dup-formulas*. Note that in the intuitionistic case the treatment of $\mathbf{T} \to$-rule is problematic and requires duplications [1]; in $\mathcal{T}_{\mathcal{K}}$ duplications are avoided by the introduction of the sign $\mathbf{T_s}$. A set Δ *clashes* iff $\{\mathbf{F}(c : C), \mathbf{T}(c : C)\} \subseteq \Delta$ or $\mathbf{T}(c : \bot) \in \Delta$. Clearly, a clashing set is not realizable. A *proof table* for Δ is a finite tree τ with Δ as root and such that all the children of a node Δ' of τ are the sets in the consequence of a rule applied to Δ'. If all the leaves of τ clash, τ is a *closed proof table* for Δ and we say that Δ is *provable (in $\mathcal{T}_{\mathcal{K}}$)*; Δ is *consistent* iff Δ is not provable.

Before proving soundness and completeness we give an example of a proof.

Example 2. Let $H = C \sqcup \neg C$. Since $c : H$ is valid in $\mathcal{ALC}$, by Theorem 1 $c : \neg\neg H$ is valid in $\mathcal{KALC}$. We show a proof of $c : \neg\neg H$ (recall that $\neg D = D \to \bot$). The proof is displayed according to the standard notation [9]. In the proof we underline the clashing formulas, we denote with X a clashing set and we label with an integer the formulas treated by the rules when needed.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathbf{F}(c:\neg\neg H)}{\mathbf{T}(c:\neg H),\mathbf{F}(c:\bot)}\;\mathbf{F}{\to}}{\underline{\mathbf{T}(c:\bot)},\mathbf{F}(c:\bot) \mid \mathbf{F}(c:H)^1,\mathbf{T_s}(c:\bot),\mathbf{F}(c:\bot) \mid \mathbf{F}(c:H)^2,\mathbf{T_s}(c:\bot)}\;\mathbf{T}{\to}}{\mathrm{X} \mid \mathbf{F}(c:C),\mathbf{F}(c:\neg C)^3,\mathbf{T_s}(c:\bot),\mathbf{F}(c:\bot) \mid \Delta=\mathbf{F}(c:C),\mathbf{F}(c:\neg C)^4,\mathbf{T_s}(c:\bot)}\;\mathbf{F}{\sqcup}^1,\mathbf{F}{\sqcup}^2}{\mathrm{X} \mid \underline{\mathbf{F}(c:C)},\underline{\mathbf{T}(c:C)},\mathbf{T_s}(c:\bot),\mathbf{F}(c:\bot) \mid \mathbf{T}(c:C),\mathbf{F}(c:\bot),\underline{\mathbf{T}(c:\bot)} \mid \Delta}\;\mathbf{F}{\to}^3}{\mathrm{X} \mid \mathrm{X} \mid \mathrm{X} \mid \underline{\mathbf{F}(c:C)},\underline{\mathbf{T}(c:C)},\mathbf{F}(c:\bot),\mathbf{T_s}(c:\bot) \mid \mathbf{T}(c:C),\mathbf{F}(c:\bot),\underline{\mathbf{T}(c:\bot)}}\;\mathbf{F}{\to}^4$$

Note that, if $\mathbf{T_s}(c : \bot) \in \Delta$, then Δ_s clashes. Thus, in applying one of the rules $\mathbf{F} \to$, $\mathbf{T} \to$ and $\mathbf{F}\forall$ to Δ, we can drop out the rightmost set in the conclusion and a proof table for $\Delta, \mathbf{T_s}(c : \bot)$ resembles an $\mathcal{ALC}$ proof table. ◇

Soundness. The following is the main lemma to prove the soundness of $\mathcal{T}_{\mathcal{K}}$.

Lemma 1. *Let Δ be a set of signed formulas, $\underline{K} = \langle P, \leq, \rho, \iota\rangle$ a Kripke model such that $\underline{K}, \alpha \rhd \Delta$, with $\alpha \in P$, and r a rule of $\mathcal{T}_{\mathcal{K}}$ applicable to Δ. Then, there is a set Δ' in the consequence of r and $\beta \in P$ such that $\underline{K}, \beta \rhd \Delta'$.*

Proof. We only discuss the case of rule $\mathbf{T} \to$. Let $W = \mathbf{T}(c : C \to D)$ and let us assume $\underline{K}, \alpha \rhd \Delta, W$. If $\underline{K}, \alpha \rhd \mathbf{T}(c : D)$, the assertion holds. Otherwise, $\underline{K}, \alpha \rhd \mathbf{F}(c : C)$; being $\underline{K}$ finite, there exists $\beta \geq \alpha$ such that $\underline{K}, \beta \rhd \mathbf{F}(c : C)$ and $\underline{K}, \beta \rhd \mathbf{T_s}(c : C)$, which implies $\underline{K}, \beta \rhd \mathbf{T_s}(c : D)$. If $\beta = \alpha$ then $\underline{K}, \beta \rhd \Delta$, otherwise $\underline{K}, \beta \rhd \Delta_s$, and the assertion is proved. □

By the previous lemma we get:

Theorem 2 (Soundness). *Let Δ be a set of signed formulas. If Δ is realizable, then Δ is consistent.* □

4 Completeness and Termination

In this section we prove the completeness of $\mathcal{T}_{\mathcal{K}}$ and we provide a decision procedure for $\mathcal{KALC}$ based on $\mathcal{T}_{\mathcal{K}}$. Let Δ be a set of signed formulas; we say that Δ

is *acyclic* iff the set of $A \sqsubseteq C$ such that $\mathbf{T}(A \sqsubseteq C) \in \Delta$ is an acyclic TBox. Note that, according to Theorem 1, to solve the inference problems w.r.t. an acyclic TBox is equivalent to decide the realizability of an acyclic set. We show that, given a finite acyclic consistent set Δ, we can build in finite time a countermodel for Δ, i.e. a Kripke model $\underline{K} = \langle P, \leq, \rho, \iota \rangle$ such that $\underline{K}, \rho \rhd \Delta$. Our construction is inspired to the standard technique used for $\mathcal{ALC}$ [2] based on graph expansion. A labelled graph $\mathcal{G}$ refers to a world α of the countermodel $\underline{K}$ under construction: the nodes of $\mathcal{G}$ form the domain $\mathcal{D}^\alpha$ of α, while the labelled arcs (c, d, R) of $\mathcal{G}$ define the interpretation of R in α. Each node c is associated with a finite set of signed formulas $\mathcal{S}(c : C)$, representing the formulas that must be realized in α. To get this, we repeatedly apply the following transformation rules on $\mathcal{G}$:

1. Firstly, we apply to $\mathcal{G}$ *expansion rules* as in the standard construction of a downward saturated set. We call *expanded graph* the graph $\mathrm{Exp}(\mathcal{G})$ obtained at the end of this step; $\mathrm{Exp}(\mathcal{G})$ completely describes a world α of $\underline{K}$.
2. Let $\mathcal{G}_e$ be an expanded graph describing a world α. We give rules to compute the *successor graphs* $\mathcal{G}'$ of $\mathcal{G}_e$ so that the graphs $\mathrm{Exp}(\mathcal{G}')$ will be all the immediate successors of α in $\underline{K}$.

We need some care to guarantee the termination. We partition the formulas associated with a node in *primary* and *secondary formulas*. Roughly speaking, primary formulas drive the graph construction. At every step a primary formula or a TBox axiom is selected and the graph is expanded according to the chosen formula. The formulas already considered are collected in the set of secondary formulas. Dup-formulas require an ad-hoc treatment to avoid infinite loops: for every dup-formula we store the individual names already considered in the expansion procedure. The TBox formulas can be seen as "global constraints" on $\mathcal{G}$ and are not affected by the transformation rules, thus we take them apart.

Formally, we consider *labelled graphs* $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}, \textsc{Pf}, \textsc{Sf}, \textsc{Tb}, \textsc{Df} \rangle$ where:

– $\mathcal{N}$ is the set of *nodes*, with $\mathcal{N}$ a finite subset of $\mathtt{NI}$.
– $\mathcal{E}$ is the set of *labelled edges* (c, d, R), with $c, d \in \mathcal{N}$ and $R \in \mathtt{NR}$.
– Pf and Sf are functions associating with every node c a finite set of signed formulas $\mathcal{S}(c : C)$, called the *primary* and *secondary* formulas of c respectively.
– Tb has the form $\mathbf{T}(\mathcal{T})$, with $\mathcal{T}$ a finite acyclic TBox.
– Df is a function mapping a dup-formula to a finite set of nodes.

The sets $\textsc{Form}(\mathcal{G})$ and $\textsc{Form}^*(\mathcal{G})$ are defined as:

$$\begin{aligned} \textsc{Form}(\mathcal{G}) &= \textstyle\bigcup_{c \in \mathcal{N}} \textsc{Pf}(c) \cup \{\, \mathbf{T}((c,d) : R) \mid (c,d,R) \in \mathcal{E} \,\} \cup \textsc{Tb} \\ \textsc{Form}^*(\mathcal{G}) &= \textsc{Form}(\mathcal{G}) \cup \textstyle\bigcup_{c \in \mathcal{N}} \textsc{Sf}(c) \end{aligned}$$

Assumptions on $\mathcal{G}$*.* In the following we assume that at any step of the countermodel construction a graph $\mathcal{G}$ satisfies the following properties (G1) and (G2):

(G1) $\textsc{Form}(\mathcal{G})$ is consistent.
(G2) The following closure properties hold:

- If $\mathbf{T}(c : C \sqcap D) \in \text{SF}(c)$, then $\{\mathbf{T}(c : C), \mathbf{T}(c : D)\} \subseteq \text{FORM}^*(\mathcal{G})$.
- If $\mathbf{F}(c : C \sqcap D) \in \text{SF}(c)$, then $\mathbf{F}(c : C) \in \text{FORM}^*(\mathcal{G})$ or $\mathbf{F}(c : D) \in \text{FORM}^*(\mathcal{G})$.
- If $\mathbf{T}(c : C \sqcup D) \in \text{SF}(c)$, then $\mathbf{T}(c : C) \in \text{FORM}^*(\mathcal{G})$ or $\mathbf{T}(c : D) \in \text{FORM}^*(\mathcal{G})$.
- If $\mathbf{F}(c : C \sqcup D) \in \text{SF}(c)$, then $\{\mathbf{F}(c : C), \mathbf{F}(c : D)\} \subseteq \text{FORM}^*(\mathcal{G})$.
- If $\mathbf{T}(c : C \to D) \in \text{SF}(c)$, then $\mathbf{T}(c : D) \in \text{FORM}^*(\mathcal{G})$ or $\{\mathbf{F}(c : C), \mathbf{T_s}(c : D)\} \subseteq \text{FORM}^*(\mathcal{G})$.
- If $\mathbf{F}(c : C \to D) \in \text{SF}(c)$, then $\{\mathbf{T}(c : C), \mathbf{F}(c : D)\} \in \text{FORM}^*(\mathcal{G})$.
- If $\mathbf{T}(c : \exists R.C) \in \text{SF}(c)$, then there is $(c, q, R) \in \mathcal{E}$ s.t. $\mathbf{T}(q : C) \in \text{FORM}^*(\mathcal{G})$.
- If $W = \mathbf{F}(c : \exists R.C) \in \text{PF}(c)$ and $d \in \text{DF}(W)$, then $\mathbf{F}(d : C) \in \text{FORM}^*(\mathcal{G})$.
- If $W = \mathbf{T}(c : \forall R.C) \in \text{PF}(c)$ and $d \in \text{DF}(W)$, then $\mathbf{T}(d : C) \in \text{FORM}^*(\mathcal{G})$.
- If $\mathbf{F}(c : \forall R.C) \in \text{SF}(c)$, then there is $(c, q, R) \in \mathcal{E}$ s.t. $\mathbf{F}(q : C) \in \text{FORM}^*(\mathcal{G})$.
- If $W = \mathbf{T}(A \sqsubseteq C) \in \text{TB}$ and $c \in \text{DF}(W)$ and $\mathbf{T}(c : A) \in \text{PF}(c)$, then $\mathbf{T}(c : C) \in \text{FORM}^*(\mathcal{G})$.

The starting graph $\mathcal{G}_\Delta$. The countermodel construction for Δ starts with the graph $\mathcal{G}_\Delta = \langle \mathcal{N}_\Delta, \mathcal{E}_\Delta, \text{PF}_\Delta, \text{SF}_\Delta, \text{TB}, \text{DF}_\Delta \rangle$, where $\mathcal{N}_\Delta$ is the set of individual names occurring in Δ, $\mathcal{E}_\Delta$ is the set of (c, d, R) such that $\mathbf{T}((c, d) : R) \in \Delta$, $\text{PF}_\Delta(c)$ is the set of $\mathcal{S}(c : A) \in \Delta$, SF_Δ and DF_Δ maps any element to the empty set, TB is the set of $\mathbf{T}(A \sqsubseteq C) \in \Delta$. One can easily check that $\mathcal{G}_\Delta$ satisfies (G1) and (G2).

Expansion of a graph $\mathcal{G}$. Let $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}, \text{PF}, \text{SF}, \text{TB}, \text{DF} \rangle$ be a finite graph and $W \in \text{FORM}(\mathcal{G})$. *Expansion rules* are defined in Fig. 3. Given W, the corresponding expansion rule transforms $\mathcal{G}$ in a new graph $\mathcal{G}' = \langle \mathcal{N}', \mathcal{E}', \text{PF}', \text{SF}', \text{TB}, \text{DF}' \rangle$. In the rules, $\mathcal{S}_W$ denotes the sign of W. We only indicate the components of the graph that are actually modified; if an element E of $\mathcal{G}$ is not mentioned, it is understood that the corresponding element E' of $\mathcal{G}'$ coincides with E. In some cases rules have no effect (for instance, in the case $W = \mathbf{F}(c : C \to D)$ when the if condition does not hold).

We repeatedly apply expansion rules to $\mathcal{G}$ until no rule is applicable. Let $\text{Exp}(\mathcal{G})$ denote the *expanded graph* $\mathcal{G}_e = \langle \mathcal{N}_e, \mathcal{E}_e, \text{PF}_e, \text{SF}_e, \text{TB}, \text{DF}_e \rangle$ obtained at the end of the expansion step; $\text{Mod}(\mathcal{G}_e)$ is the model $(\mathcal{D}^\alpha, \cdot^\alpha)$ for $\mathcal{L}_{\mathcal{N}_e}$ representing the world α such that:

- $\mathcal{D}^\alpha = \mathcal{N}_e$ and, for every $c \in \mathcal{N}_e$, $c^\alpha = c$;
- for every $A \in \texttt{NC}$, A^α is the set of c such that $\mathbf{T}(c : A) \in \text{PF}_e(c)$;
- for every $R \in \texttt{NR}$, R^α is the set of pairs (c, d) such that $(c, d, R) \in \mathcal{E}_e$.

The following properties are crucial to prove the finiteness of $\text{Exp}(\mathcal{G})$.

(P1) For every $c \in \mathcal{N}_e$, the set of R-successors of c in $\mathcal{G}_e$ is finite.

(P2) Let $c_0 \in \mathcal{N}_e \setminus \mathcal{N}$, let $\sigma = c_0, c_1, \ldots$ be an R-chain of nodes of $\mathcal{G}_e$, namely: $(c_k, c_{k+1}, R) \in \mathcal{E}_e$ for every $k \geq 0$. Let A be a concept name and $\mathcal{B}(\sigma, A)$ the set of c in σ such that $\mathbf{T}(c : A) \in \text{PF}_e(c)$. Then, $\mathcal{B}(\sigma, A)$ is finite.

We only give a sketch of the proof. As for (P1), d is an R-successor of c in $\mathcal{G}_e$ iff $(c, d, R) \in \mathcal{E}$ or d has been generated by a formula $W = \mathbf{T}(c : \exists R.C)$ or $W = \mathbf{F}(c : \forall R.C)$. In the former case the assertion follows by finiteness of $\mathcal{G}$. In

Formula	Expansion rule
$W = \mathbf{T}(c : C \sqcap D)$ $W = \mathbf{F}(c : C \sqcup D)$	$\textsc{Pf}'(c) = (\textsc{Pf}(c) \setminus \{W\}) \cup \{\mathcal{S}_W(c : C), \mathcal{S}_W(c : D)\}$ $\textsc{Sf}'(c) = \textsc{Sf}(c) \cup \{W\}$
$W = \mathbf{F}(c : C \sqcap D)$ $W = \mathbf{T}(c : C \sqcup D)$	If $(\Delta \setminus \{W\}) \cup \{\mathcal{S}_W(c : C)\}$ is consistent, then $\textsc{Pf}'(c) = (\textsc{Pf}(c) \setminus \{W\}) \cup \{\mathcal{S}_W(c : C)\}$ else $\textsc{Pf}'(c) = (\textsc{Pf}(c) \setminus \{W\}) \cup \{\mathcal{S}_W(c : D)\}$ $\textsc{Sf}'(c) = \textsc{Sf}(c) \cup \{W\}$
$W = \mathbf{F}(c : C \to D)$	If $(\Delta \setminus \{W\}) \cup \{\mathbf{T}(c : C), \mathbf{F}(c : D)\}$ is consistent then $\textsc{Pf}'(c) = (\textsc{Pf}(c) \setminus \{W\}) \cup \{\mathbf{T}(c : C), \mathbf{F}(c : D)\}$ $\textsc{Sf}'(c) = \textsc{Sf}(c) \cup \{W\}$
$W = \mathbf{T}(c : C \to D)$	If $(\Delta \setminus \{W\}) \cup \{\mathbf{T}(c : D)\}$ is consistent then $\textsc{Pf}'(c) = (\textsc{Pf}(c) \setminus \{W\}) \cup \{\mathbf{T}(c : D)\}$ $\textsc{Sf}'(c) = \textsc{Sf}(c) \cup \{W\}$ else if $(\Delta \setminus \{W\}) \cup \{\mathbf{F}(c : C), \mathbf{T_s}(c : D)\}$ is consistent then $\textsc{Pf}'(c) = (\textsc{Pf}(c) \setminus \{W\}) \cup \{\mathbf{F}(c : C), \mathbf{T_s}(c : D)\}$ $\textsc{Sf}'(c) = \textsc{Sf}(c) \cup \{W\}$
$W = \mathbf{T}(A \sqsubseteq C)$	Let $c \in \mathcal{N} \setminus \textsc{Df}(W)$ If $\mathbf{T}(c : A) \in \textsc{Pf}(c)$ then $\textsc{Pf}'(c) = \textsc{Pf}(c) \cup \{\mathbf{T}(c : C)\}$ $\textsc{Df}'(W) = \textsc{Df}(W) \cup \{c\}$
$W = \mathbf{T}(c : \exists R.C)$	Let $q \notin \mathcal{N}$. $\mathcal{N}' = \mathcal{N} \cup \{q\}$ $\mathcal{E}' = \mathcal{E} \cup \{(c, q, R)\}$ $\textsc{Pf}'(c) = \textsc{Pf}(c) \setminus \{W\}$ $\textsc{Pf}'(q) = \{\mathbf{T}(q : C)\}$ $\textsc{Sf}'(c) = \textsc{Sf}(c) \cup \{W\}$ $\textsc{Sf}'(q) = \emptyset$
$W = \mathbf{F}(c : \exists R.C)$ $W = \mathbf{T}(c : \forall R.C)$	Let $d \in \mathcal{N}$ such that $(c, d, R) \in \mathcal{E}$ and $d \notin \textsc{Df}(W)$ $\textsc{Pf}'(d) = \textsc{Pf}(d) \cup \{\mathcal{S}_W(d : C)\}$ $\textsc{Df}'(W) = \textsc{Df}(W) \cup \{d\}$
$W = \mathbf{F}(c : \forall R.C)$	Let $q \notin \mathcal{N}$ If $(\Delta \setminus \{W\}) \cup \{\mathbf{T}((c, q) : R), \mathbf{F}(q : C)\}$ is consistent $\mathcal{N}' = \mathcal{N} \cup \{q\}$ $\mathcal{E}' = \mathcal{E} \cup \{(c, q, R)\}$ $\textsc{Pf}'(c) = \textsc{Pf}(c) \setminus \{W\}$ $\textsc{Pf}'(q) = \{\mathbf{F}(q : C)\}$ $\textsc{Sf}'(c) = \textsc{Sf}(c) \cup \{W\}$ $\textsc{Sf}'(q) = \emptyset$

$\mathcal{S}_W$ denotes the sign of W

Fig. 3. Expansion rules

the latter two cases, W must be a subformula of a formula in $\textsc{Form}^*(\mathcal{G})$, and only finitely many such W exist.

Let $\textsc{Tb} = \mathbf{T}(\mathcal{T})$ and let $\prec$ be the "uses" relation induced by the TBox $\mathcal{T}$. We prove (P2) by induction on $\prec$ (recall that $\mathcal{T}$ is finite and acyclic, hence $\prec$ is well-founded). If A is minimal w.r.t. $\prec$ then, for every $\mathbf{T}(A' \sqsubseteq C) \in \textsc{Tb}$, A is not a subformula of C. Thus, $c \in \mathcal{B}(\sigma, A)$ iff $\mathbf{T}(c : A)$ has been generated by some formula in $\textsc{Pf}_e(c_0)$ or $\textsc{Sf}_e(c_0)$, and this implies that $\mathcal{B}(\sigma, A)$ is finite. Suppose that A is not minimal. If $\mathcal{B}(\sigma, A)$ is infinite, there must exist a formula $W = \mathbf{T}(A' \sqsubseteq C) \in \textsc{Tb}$ such that A is a subformula of C and the rule $\mathbf{T} \sqsubseteq$ has been applied infinitely many times on W. It follows that $\mathcal{B}(\sigma, A')$ is infinite. Since $A' \prec A$, this contradicts the induction hypothesis.

Formula	Successor graph
$W = \mathbf{F}(c : C \to D)$ $W = \mathbf{T}(c : C \to D)$	$\mathcal{N}' = \mathcal{N} \quad \mathcal{E}' = \mathcal{E} \quad \mathrm{DF}' = \mathrm{DF}_s$ $\mathrm{PF}'(c) = (\mathrm{PF}(c))_s \cup \mathcal{R}_W \quad \mathrm{PF}'(d) = (\mathrm{PF}(d))_s$ for every $d \neq c$ $\mathrm{SF}'(c) = (\mathrm{SF}(c))_s \cup \{W\} \quad \mathrm{SF}'(d) = (\mathrm{SF}(d))_s$ for every $d \neq c$ where $\mathcal{R}_{\mathbf{F}(c:C \to D)} = \{\mathbf{T}(c : C), \mathbf{F}(c : D)\}$ and $\mathcal{R}_{\mathbf{T}(c:C \to D)} = \{\mathbf{F}(c : C), \mathbf{T_s}(c : D)\}$
$W = \mathbf{F}(c : \forall R.C)$	Let $q \notin \mathcal{N}$. $\mathcal{N}' = \mathcal{N} \cup \{q\} \quad \mathcal{E}' = \mathcal{E} \cup \{ (c, q, R) \} \quad \mathrm{DF}' = \mathrm{DF}_s$ $\mathrm{PF}'(q) = \{ \mathbf{F}(q : C) \} \quad \mathrm{PF}'(d) = (\mathrm{PF}(d))_s$ for every $d \in \mathcal{N}$ $\mathrm{SF}'(c) = (\mathrm{SF}(c))_s \cup \{W\} \quad \mathrm{SF}'(q) = \emptyset$ $\mathrm{SF}'(e) = (\mathrm{SF}(e))_s$ for every $e \in \mathcal{N} \setminus \{c\}$

$\mathrm{DF}_s(Z) = \mathrm{DF}(Z)$ if $Z = \mathbf{T}(H)$, otherwise $\mathrm{DF}_s(Z) = \emptyset$

Fig. 4. Successor graphs

By (P2) it follows that $\mathcal{G}_e$ does not contain infinite R-chains starting from a node $c_0 \in \mathcal{N}_e \setminus \mathcal{N}$. We conclude:

Lemma 2. Exp($\mathcal{G}$) *is finite.* □

Successor of an expanded graph $\mathcal{G}$. Let W be a formula of $\mathrm{FORM}(\mathcal{G})$. The *successor graph* of $\mathcal{G}$ generated by W is the graph $\mathcal{G}' = \langle \mathcal{N}', \mathcal{E}', \mathrm{PF}', \mathrm{SF}', \mathrm{TB}, \mathrm{DF}' \rangle$ defined according to the form of W as specified in Fig. 4.

Countermodel construction. Let Δ be a finite acyclic consistent set of signed formulas. The countermodel $\underline{K}(\Delta) = \langle P, \leq, \rho, \iota \rangle$ for Δ is built as follows.

- The root ρ coincides with Mod(Exp($\mathcal{G}_\Delta$)), where $\mathcal{G}_\Delta$ is the starting graph.
- Let α = Mod($\mathcal{G}_\alpha$) be a world of $\underline{K}(\Delta)$ and let $\mathcal{G}_1, \ldots, \mathcal{G}_m$ be all the successors of $\mathcal{G}_\alpha$. Then, the immediate successors of α in $\underline{K}(\Delta)$ are the models Mod(Exp($\mathcal{G}_1$)), ..., Mod(Exp($\mathcal{G}_m$)).
- $\leq$ is the reflexive and transitive closure of the immediate successor relation.

The termination of the countermodel construction procedure is guaranteed by the following property.

(T) Let $\mathcal{G}'$ be obtained by applying to $\mathcal{G}$ one of the rules of Fig. 3 and 4 defined by W. Then, one of the following facts holds ($|W|$ denotes the size of W):
  (1) $\mathrm{FORM}(\mathcal{G}')$ is obtained by replacing W with one or more formulas W' such that $|W'| < |W|$, possibly substituting $\mathbf{T_s}$ with $\mathbf{T}$ and discharging the **F**-formulas.
  (2) If W is a dup-formula, $\mathrm{FORM}(\mathcal{G}') = \mathrm{FORM}(\mathcal{G}) \cup \{W'\}$, with $|W'| < |W|$, and $\mathrm{DF}(W) \subset \mathrm{DF}'(W)$.

By Lemma 2 the sets $\mathrm{DF}(W)$ can not increase indefinitely, thus we cannot apply the transformation rules infinitely many times.

We now state the main results of this section.

Lemma 3. *Let Δ be a finite acyclic consistent set of signed formulas.*

(i) The model $\underline{K}(\Delta)$ is finite.
(ii) Let $\alpha = \mathrm{Mod}(\mathcal{G}_\alpha)$ be a world of $\underline{K}(\Delta)$. Then, $\underline{K}(\Delta), \alpha \rhd \textsc{Form}^*(\mathcal{G}_\alpha)$.
(iii) $\underline{K}(\Delta), \rho \rhd \Delta$.

Proof. Point (i) follows by Property (T). To prove (ii), one has to show that $W \in \textsc{Form}^*(\mathcal{G}_\alpha)$ implies $\underline{K}(\Delta), \alpha \rhd W$; the proof is by induction on W, using (G1) and (G2). Point (iii) follows by (ii), being $\Delta \subseteq \textsc{Form}(\mathcal{G}_\Delta) \subseteq \textsc{Form}^*(\mathcal{G}_\rho)$. □

By the previous lemma and by the Soundness Theorem we conclude:

Theorem 3 (Completeness). *Let Δ be a finite acyclic set of signed formulas. Then, Δ is realizable iff Δ is consistent.* □

The countermodel construction procedure can be used to decide the realizability of an acyclic Δ. Indeed, one tries to build $\underline{K}(\Delta)$ by applying the transformation rules in all possible ways; by Property (T), the search space is finite. If all the attempts fail, yielding a clashing set $\textsc{Pf}(c)$, Δ is not consistent (Lemma 3), hence it is not realizable (Theorem 3). In this case, the failed branches correspond to the branches of a closed proof table for Δ.

5 Related Works and Conclusions

The logic $\mathcal{KALC}$ we have introduced is strongly connected with the constructive DL presented in [5], let us call $\mathcal{KALC}'$. Indeed, a Kripke model $\underline{K} = \langle P \leq, \rho, \iota \rangle$ for $\mathcal{KALC}'$ is a $\mathcal{KALC}$ model where P can be infinite and, for every $\alpha \in P$, there is a final element $\phi \in P$ such that $\alpha \leq \phi$. The restriction to finite models is crucial to prove the decidability of $\mathcal{KALC}$ (whereas $\mathcal{KALC}'$ is semidecidable). Clearly, $\mathcal{KALC}' \subseteq \mathcal{KALC}$. If, as we conjecture, $\mathcal{KALC} \subseteq \mathcal{KALC}'$, we can conclude that $\mathcal{KALC} = \mathcal{KALC}'$ has the finite model property.

It is well-known that DLs have multi-modal logic counterparts [2]; likewise, intuitionistic DLs are related to intuitionistic multi-modal logics [10,15], via the standard translation between the involved languages. It is easy to prove that the multi-modal version of Fischer-Servi logic **FS** [10] is contained in $\mathcal{KALC}$. On the other hand **FS** $\neq \mathcal{KALC}$, since the formula $H = c : \forall R.\neg\neg A \to \neg\neg\forall R.A$ belongs to $\mathcal{KALC}$, while the corresponding formula $\Box\neg\neg A \to \neg\neg\Box A$ does not belong to **FS** (see the countermodel in [15]). We remark that H belongs to $\mathcal{KALC}'$ as well, due to the fact that $\mathcal{KALC}'$ models have final elements.

As for the comparison with other approaches, we notice that our notion of refinement (induced by the partial order relation of Kripke models) concerns the whole state of knowledge. So it is closer to the usual Kripke interpretation than those given in [13,14], which concern single individuals. Note that in [13] the knowledge about roles is not monotonic. We plan to investigate the relation between $\mathcal{KALC}$ and the constructive description logic $\mathcal{BCDL}$ [8], which exploits a different semantics. Finally, we aim to extend the decision procedure to treat general TBoxes and transitive and inverse role relations by introducing loop-checking mechanisms, such as *blocking* and its variants [11].

References

1. Avellone, A., Ferrari, M., Miglioli, P.: Duplication-free tableau calculi and related cut-free sequent calculi for the interpolable propositional intermediate logics. Logic J. of the IGPL 7(4), 447–480 (1999)
2. Baader, F., Nutt, W.: Basic description logics. In: [7], pp. 43–95
3. Bozzato, L., Ferrari, M., Fiorentini, C., Fiorino, G.: A constructive semantics for $\mathcal{ALC}$. In: Calvanese, D., et al. (eds.) 2007 International Workshop on Description Logics. CEUR Proceedings, vol. 250, pp. 219–226 (2007)
4. Bozzato, L., Ferrari, M., Villa, P.: Actions over a constructive semantics for $\mathcal{ALC}$. In: Baader, F., et al. (eds.) 2008 International Workshop on Description Logics. CEUR Proceedings, vol. 353 (2008)
5. Bozzato, L., Ferrari, M., Villa, P.: A note on constructive semantics for description logics. In: 24-esimo Convegno Italiano di Logica Computazionale (2009), `http://www.ing.unife.it/eventi/cilc09/accepted.shtml`
6. de Paiva, V.: Constructive description logics: what, why and how. Technical report, Xerox Parc (2003)
7. Baader, F., et al. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
8. Ferrari, M., Fiorentini, C., Fiorino, G.: BCDL: Basic Constructive Description Logic. J. of Automated Reasoning 44(4), 371–399 (2010)
9. Fitting, M.C.: Proof Methods for Modal and Intuitionistic Logics. Reidel, Dordrechtz (1983)
10. Gabbay, D.M., Kurucz, A., Wolter, F., Zakharyaschev, M.: Many-dimensional modal logics: theory and applications. Studies in logic and the foundations of mathematics. North-Holland, Amsterdam (2003)
11. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for very expressive description logics. Logic J. of the IGPL 8(3), 239–264 (2000)
12. Kaneiwa, K.: Negations in description logic - contraries, contradictories, and subcontraries. In: Proc. of the 13th International Conference on Conceptual Structures (ICCS 2005), pp. 66–79. Kassel University Press (2005)
13. Mendler, M., Scheele, S.: Towards Constructive DL for Abstraction and Refinement. J. of Automated Reasoning 44(3), 207–243 (2010)
14. Odintsov, S.P., Wansing, H.: Inconsistency-tolerant description logic. Part II: A tableau algorithm for $CALC^C$. J. of Applied Logic 6(3), 343–360 (2008)
15. Simpson, A.K.: The Proof Theory and Semantics of Intuitionistic Modal Logic. PhD thesis, University of Edinburgh (1994)