A note on constructive semantics for description logics

Loris Bozzato, Mauro Ferrari, and Paola Villa

Dipartimento di Informatica e Comunicazione
Università degli Studi dell'Insubria
Via Mazzini 5, 21100, Varese, Italy

Abstract. Following the approaches and motivations given in recent works about constructive interpretation of description logics, we introduce the constructive description logic $\mathcal{KALC}$. This logic is based on a Kripke-style semantics inspired by the Kripke semantics for Intuitionistic first order logic. In the paper we present the main features of our semantics and we study its relations with other approaches. Moreover, we present a tableau calculus which turns out to be sound and complete for $\mathcal{KALC}$.

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 the applicability of a logical system is often driven from its classical semantics, a constructive analysis allows us to take advantage of 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 [3, 5, 6, 9, 11, 12] different proposals of a constructive reinterpretation of description logics have been motivated.

Following this line, we discuss a Kripke-style semantics for the basic description logic $\mathcal{ALC}$ [2, 13] inspired by the Kripke semantics for Intuitionistic first order logic [14]. We call $\mathcal{KALC}$ the logic corresponding to our semantics. Basically, we may think of a *Kripke model* as a set of worlds, representing *states of knowledge*, partially ordered by their information content. Our semantics differs from the similar semantics described in [5] by the fact that we impose a condition on the partial order. In particular, we require that every world is followed by a *classical world*, that is a world where concepts are interpreted according to the usual classical semantics for the description logic $\mathcal{ALC}$. As we discuss in Section 3, such a condition seems to be essential to get the finite model property, useful to define a decidable calculus for the logic. An important feature of our semantics is that such a condition can be characterized by the axiom-schema $\forall R.\neg\neg A \rightarrow \neg\neg\forall R.A$. This schema is obtained by translating in the description logics setting the *Kuroda principle* for first order logic, a principle which has been deeply studied in the literature of constructive logics [7, 14].

In the paper we also present a tableau calculus which is sound and complete with respect to our semantics. This calculus forms the base of our attempt to prove the finite model property for $\mathcal{KALC}$. Moreover, it allows us to directly compare to classical tableaux based algorithms that represent the main reasoning method of actual description logics implementations. In this paper we use the tableau calculus to prove that $\mathcal{KALC}$ meets the *disjunction property*, that is, if $A \sqcup B$ belongs to $\mathcal{KALC}$, then either A or B belongs to $\mathcal{KALC}$.

The rest of the paper is organized as follows. In Section 2, we introduce the syntax and the classical semantics for $\mathcal{ALC}$. In Section 3 we present the constructive semantics of $\mathcal{KALC}$ and we describe the relations between our semantics, the classical semantics for $\mathcal{ALC}$ and the constructive semantics described in [5, 11, 12]. In Section 4, we present sound and complete tableau calculus $\mathcal{T}$ for $\mathcal{KALC}$. Finally, conclusions are drawn in Section 5.

2 Syntax and classical semantics

We begin by introducing the language of the basic description logic $\mathcal{ALC}$ [2, 13] and its classical semantics. The language $\mathcal{L}$ of $\mathcal{ALC}$ is based on the following denumerable sets: the set $\mathtt{NR}$ of *role names*, the set $\mathtt{NC}$ of *concept names*, the set $\mathtt{NI}$ of *individual names*. A *concept* A is an expression of the kind:

$$A \ ::= \ C \mid \neg A \mid A \sqcap A \mid A \sqcup A \mid A \rightarrow A \mid \exists R.A \mid \forall R.A$$

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

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

where $s, t \in \mathtt{NI} \cup \mathtt{Var}$, $R \in \mathtt{NR}$ and A is a concept. We write $H \in \mathcal{L}$ to mean that H is a formula of $\mathcal{L}$. An *atomic formula* of $\mathcal{L}$ is a formula of the kind $\bot$, $(s,t) : R$ or $t : C$, where $R \in \mathtt{NR}$ and $C \in \mathtt{NC}$. A *negated formula* is a formula of the kind $t : \neg A$. A formula is *closed* if it does not contain variables. A *concept formula* is a formula of the kind $t : A$ or A, with A a concept. A *role formula* is a formula of the kind $(t,s) : R$ with $R \in \mathtt{NR}$.

We remark that in the classical setting $A \rightarrow B$ is equivalent to $\neg A \sqcup B$, but this equivalence does not hold in the constructive setting. We define the *concept inclusion* relation (*subsumption*) $A \sqsubseteq B$ as $A \rightarrow B$.

As usual, a *knowledge base* in $\mathcal{ALC}$ consists of a *TBox* and an *ABox*. A TBox is a finite set of terminological axioms in the form $C \sqsubseteq D$ or $C \equiv D$, where C, D are concepts and $C \equiv D$ is an abbreviation for $C \sqsubseteq D$ and $D \sqsubseteq C$. An ABox is a finite set of closed concept and role assertions, of the kind $t : C$ or $(t,s) : R$, where $R \in \mathtt{NR}$ and $t, s \in \mathtt{NI}$.

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 $c \in \mathtt{NI}$, $c^{\mathcal{M}} \in \mathcal{D}^{\mathcal{M}}$; for every $C \in \mathtt{NC}$, $C^{\mathcal{M}} \subseteq \mathcal{D}^{\mathcal{M}}$; for every $R \in \mathtt{NR}$, $R^{\mathcal{M}} \subseteq \mathcal{D}^{\mathcal{M}} \times \mathcal{D}^{\mathcal{M}}$. A non atomic concept A is interpreted by a subset $A^{\mathcal{M}}$ of $\mathcal{D}^{\mathcal{M}}$ as follows:

- $(\neg A)^{\mathcal{M}} = \mathcal{D}^{\mathcal{M}} \setminus A^{\mathcal{M}}$
- $(A \sqcap B)^{\mathcal{M}} = A^{\mathcal{M}} \cap B^{\mathcal{M}}$
- $(A \sqcup B)^{\mathcal{M}} = A^{\mathcal{M}} \cup B^{\mathcal{M}}$
- $(A \rightarrow B)^{\mathcal{M}} = \{\, d \in \mathcal{D}^{\mathcal{M}} \mid d \notin A^{\mathcal{M}} \text{ or } d \in B^{\mathcal{M}} \,\}$
- $(\exists R.A)^{\mathcal{M}} = \{\, d \in \mathcal{D}^{\mathcal{M}} \mid \exists d' \in \mathcal{D}^{\mathcal{M}} \text{ s.t. } (d, d') \in R^{\mathcal{M}} \text{ and } d' \in A^{\mathcal{M}} \,\}$
- $(\forall R.A)^{\mathcal{M}} = \{\, d \in \mathcal{D}^{\mathcal{M}} \mid \forall d' \in \mathcal{D}^{\mathcal{M}}, (d, d') \in R^{\mathcal{M}} \text{ implies } d' \in A^{\mathcal{M}} \,\}$

An *assignment* on a model $\mathcal{M}$ is a map $\theta : \mathtt{Var} \rightarrow \mathcal{D}^{\mathcal{M}}$. If $t \in \mathtt{NI} \cup \mathtt{Var}$, $t^{\mathcal{M},\theta}$ is the element of $\mathcal{D}^{\mathcal{M}}$ denoting t in $\mathcal{M}$ w.r.t. θ, namely: $t^{\mathcal{M},\theta} = \theta(t)$ if $t \in \mathtt{Var}$, $t^{\mathcal{M},\theta} = t^{\mathcal{M}}$ if $t \in \mathtt{NI}$. A formula H is *valid* in $\mathcal{M}$ w.r.t. θ, and we write $\mathcal{M}, \theta \models H$, if $H \neq \bot$ and one of the following conditions holds:

- $\mathcal{M}, \theta \models t : A$ iff $t^{\mathcal{M},\theta} \in A^{\mathcal{M}}$;
- $\mathcal{M}, \theta \models (s, t) : R$ iff $(s^{\mathcal{M},\theta}, t^{\mathcal{M},\theta}) \in R^{\mathcal{M}}$;
- $\mathcal{M}, \theta \models A$ iff $A^{\mathcal{M}} = \mathcal{D}^{\mathcal{M}}$.

We write $\mathcal{M} \models H$ iff $\mathcal{M}, \theta \models H$ for every assignment θ. Note that, given a concept A of $\mathcal{L}$, $\mathcal{M} \models A$ iff $\mathcal{M} \models x : A$, with x any variable. If Γ is a set of formulas, $\mathcal{M} \models \Gamma$ means that $\mathcal{M} \models H$ for every $H \in \Gamma$. We say that H is a *logical consequence* of Γ, and we write $\Gamma \models H$, iff, for every $\mathcal{M}$ and every θ, $\mathcal{M}, \theta \models \Gamma$ implies $\mathcal{M}, \theta \models H$.

3 Kripke semantics

In this section we introduce a Kripke-style semantics for $\mathcal{ALC}$ inspired by the *Kripke semantics* for Intuitionistic first order logic. A similar semantics for description logics has already been proposed in [5]: however, our semantics differs from this one by the fact that we additionally impose a condition on the partial order. As we discuss at the end of this section, such a condition seems to be essential to get the finite model property.

Given a partially ordered set (poset) $(P, \leq)$, we call *final* an element $\phi \in P$ such that, for every $\alpha \in P$, $\phi \leq \alpha$ implies $\phi = \alpha$. Given $\alpha \in P$, we denote with $\mathrm{Fin}(\alpha)$ the set of final elements $\phi \in P$ such that $\alpha \leq \phi$. We call *K-poset* every poset $(P, \leq)$ such that, for every $\alpha \in P$, $\mathrm{Fin}(\alpha) \neq \emptyset$. We remark that every finite poset is a K-poset. A *$\mathcal{KALC}$-model* is a quadruple $\underline{K} = \langle P, \leq, \rho, \iota \rangle$, where:

- $(P, \leq)$ is a K-poset with root ρ;
- ι is a function associating with every $\alpha \in P$ a model $\iota(\alpha) = (\mathcal{D}^{\alpha}, \cdot^{\alpha})$ for $\mathcal{L}$, such that, for every $\alpha, \beta \in P$ with $\alpha \leq \beta$:
  - (K1) $\mathcal{D}^{\alpha} \subseteq \mathcal{D}^{\beta}$;
  - (K2) for every $c \in \mathtt{NI}$, $c^{\alpha} = c^{\beta}$;
  - (K3) for every $C \in \mathtt{NC}$, $C^{\alpha} \subseteq C^{\beta}$;
  - (K4) for every $R \in \mathtt{NR}$, $R^{\alpha} \subseteq R^{\beta}$.

Conditions (K1)–(K4) state that the information content of any element β accessible from α is wider than the information content of α. Indeed, (K1) says that any element known at α is also known at β. (K2)–(K4) state that every fact known at α is also known at β.

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 $\mathcal{L}$ the elements of $\mathcal{D}^\alpha$ as individual names. We assume that, for every $d \in \mathcal{D}^\alpha$, $d^\alpha = d$. The introduction of a language for every $\alpha \in P$ is a technical machinery needed to treat the interpretation of quantifiers. By Condition (K1), $\alpha \leq \beta$ and $H \in \mathcal{L}_\alpha$ imply $H \in \mathcal{L}_\beta$.

Given $\underline{K} = \langle P, \leq, \rho, \iota \rangle$, $\alpha \in P$ and a closed formula H of $\mathcal{L}_\alpha$, we inductively define the *forcing relation* $\alpha \| \!\!- H$ as follows:

- $\alpha \| \!\!\not- \bot$;
- $\alpha \| \!\!- t : A$ where $A \in \mathtt{NC}$, iff $t^\alpha \in A^\alpha$;
- $\alpha \| \!\!- (s,t) : R$ where $R \in \mathtt{NR}$, iff $(t^\alpha, s^\alpha) \in R^\alpha$;
- $\alpha \| \!\!- t : A \sqcap B$ iff $\alpha \| \!\!- t : A$ and $\alpha \| \!\!- t : B$;
- $\alpha \| \!\!- t : A \sqcup B$ iff either $\alpha \| \!\!- t : A$ or $\alpha \| \!\!- t : B$;
- $\alpha \| \!\!- t : A \to B$ iff, for every $\beta \in P$ such that $\alpha \leq \beta$, either $\beta \| \!\!\not- t : A$ or $\beta \| \!\!- t : B$;
- $\alpha \| \!\!- t : \neg A$ iff, for every $\beta \in P$ such that $\alpha \leq \beta$, $\beta \| \!\!\not- t : A$;
- $\alpha \| \!\!- t : \exists R.A$ iff there exists $d \in \mathcal{D}^\alpha$ such that $\alpha \| \!\!- (t,d) : R$ and $\alpha \| \!\!- d : A$;
- $\alpha \| \!\!- t : \forall R.A$ iff, for every $\beta \in P$ such that $\alpha \leq \beta$ and for every $d \in \mathcal{D}^\beta$, $\beta \| \!\!- (t,d) : R$ implies $\beta \| \!\!- d : A$.
- $\alpha \| \!\!- A$ iff for every $\beta \in P$ such that $\alpha \leq \beta$ and for every $d \in \mathcal{D}^\beta$, $\beta \| \!\!- d : A$.

By the above definition, it is easy to check that $t : \neg A$ is equivalent to $t : A \to \bot$ and hence we could avoid to assume $\neg$ as a primitive connective.

It is easy to prove, using Conditions (K1)–(K4) and the definition of the forcing relation, the following property:

Proposition 1 (Monotonicity property). *Let $\underline{K} = \langle P, \leq, \rho, \iota \rangle$ be a $\mathcal{KALC}$-model, $\alpha \in P$ and H be a closed formula of $\mathcal{L}_\alpha$. If $\alpha \| \!\!- H$ in $\underline{K}$, then $\beta \| \!\!- H$ in $\underline{K}$ for every $\beta \in P$ such that $\alpha \leq \beta$.* □

Given a $\mathcal{KALC}$-model $\underline{K} = \langle P, \leq, \rho, \iota \rangle$ and $\alpha \in P$, an *α-substitution* is a function $\sigma : \mathtt{Var} \to \mathtt{NI} \cup \mathcal{D}^\alpha$. By Condition (K1), we immediately have that an α-substitution is also a β-substitution for every $\beta \in P$ such that $\alpha \leq \beta$. Given a formula H, we denote with $H\sigma$ the formula obtained by simultaneously substituting every occurrence of a variable x in H with $\sigma(x)$. Given a set of formulas Γ, we denote with $\Gamma\sigma$ the set containing $H\sigma$ for every $H \in \Gamma$. If σ is an α-substitution and $d \in \mathcal{D}^\alpha$, $\sigma[d/x]$ is the α-substitution defined as follows:

$$\sigma[d/x](y) = \begin{cases} d, \text{ if } y = x \\ \sigma(y), \text{ otherwise} \end{cases}$$

Given a $\mathcal{KALC}$-model $\underline{K} = \langle P, \leq, \rho, \iota \rangle$, $\alpha \in P$ and an open formula H of $\mathcal{L}_\alpha$, $\alpha \| \!\!- H$ iff, for every $\beta \in P$ such that $\alpha \leq \beta$ and for every β-substitution σ, $\beta \| \!\!- K\sigma$. Given a formula $H \in \mathcal{L}$ and a $\mathcal{KALC}$-model $\underline{K} = \langle P, \leq, \rho, \iota \rangle$ for $\mathcal{L}$, we write $\underline{K} \| \!\!- H$ (*H is valid in $\underline{K}$*) if, for every $\alpha \in P$ and every α-substitution σ, $\alpha \| \!\!- H\sigma$ in $\underline{K}$. It is easy to check that, given an open formula $x : A$, $\underline{K} \| \!\!- x : A$ iff $\underline{K} \| \!\!- A$. Given a set of formulas $\Gamma \subseteq \mathcal{L}$ and $H \in \mathcal{L}$, H is a *K-logical consequence*

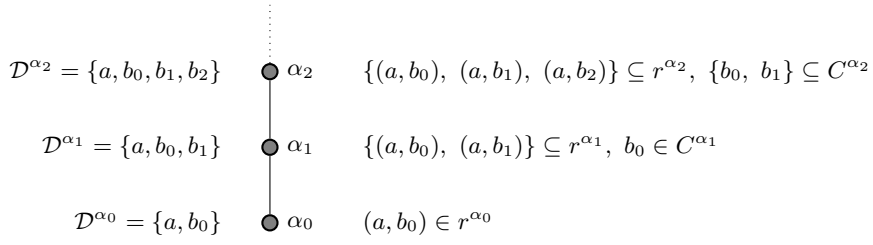

Fig. 1. A counter-model for Kur.

of Γ, denoted $\Gamma \models^{\mathrm{k}} H$, iff for every $\mathcal{KALC}$-model $\underline{K} = \langle P, \leq, \rho, \iota \rangle$ for $\mathcal{L}$, for every $\alpha \in P$ and every α-substitution σ, $\alpha \| \!\!- \Gamma\sigma$ implies $\alpha \| \!\!- H\sigma$. A formula H is *$\mathcal{KALC}$-valid* iff $\emptyset \models^{\mathrm{k}} H$. The logic $\mathcal{KALC}$ is the set of $\mathcal{KALC}$-valid formulas.

The latter definitions allow us to draw some considerations about our semantics and its relations with classical semantics for $\mathcal{ALC}$ and the constructive semantics described in [5, 11, 12]. First of all, given a classical model $\mathcal{M}$ for $\mathcal{L}$, let us consider the $\mathcal{KALC}$-model $\underline{K}_{\mathcal{M}} = \langle \{\rho\}, \{(\rho, \rho)\}, \rho, \iota \rangle$, where $\iota(\rho) = \mathcal{M}$. It is easy to check that $\underline{K}_{\mathcal{M}}$ is equivalent to $\mathcal{M}$ in the following sense: for every closed formula $H \in \mathcal{L}$, $\mathcal{M} \models H$ iff $\rho \| \!\!- H$ in $\underline{K}$. Hence, if H does not hold in a classical model $\mathcal{M}$, $\underline{K}_{\mathcal{M}}$ is a counter-model for H. Thus, if we identify $\mathcal{ALC}$ with the set of formulas valid in every classical model, we immediately get that $\mathcal{KALC} \subseteq \mathcal{ALC}$.

On the other hand, the final elements of a $\mathcal{KALC}$-model essentially coincide with classical interpretations. Indeed, given a $\mathcal{KALC}$-model $\underline{K} = \langle P, \leq, \rho, \iota \rangle$ and a final element $\phi \in P$, let $\mathcal{M}_\phi = \iota(\phi)$. It is easy to check that, for every closed formula H of $\mathcal{L}_\phi$, $\mathcal{M}_\phi \models H$ iff $\phi \| \!\!- H$ in $\underline{K}$. This implies that a formula H is satisfiable in $\mathcal{KALC}$ iff it is classically satisfiable.

Now, let us consider the Kripke structure $\underline{K} = \langle \{\alpha_i\}_{i \in \omega}, \leq, \alpha_0, \iota \rangle$ of Figure 1 where:

- $\alpha_i \leq \alpha_j$ iff $i \leq j$;
- For every α_i:
  - $\mathcal{D}^{\alpha_i} = \{a, b_0, \ldots, b_i\}$;
  - $r^{\alpha_i} = \{(a, b_0), \ldots, (a, b_i)\}$;
  - $C^{\alpha_0} = \emptyset$ and $C^{\alpha_i} = \{b_0, \ldots, b_{i-1}\}$ for $i > 0$.

We note that $\underline{K}$ is not a $\mathcal{KALC}$-model: indeed, $\mathrm{Fin}(\alpha_0) = \emptyset$ and hence its poset is not a K-poset. On the other hand $\underline{K}$ satisfies Conditions (K1)–(K4) and belongs to the class of models obtainable by directly adapting the Kripke semantics for Intuitionistic first order logic to description logics. It is easy to check that $\alpha_0 \| \!\!\not- a : \forall r.\neg\neg C \rightarrow \neg\neg\forall r.C$ in $\underline{K}$.

Now, let us consider the axiom-schema[1]:

$$\text{(Kur)} \quad \forall R.\neg\neg A \rightarrow \neg\neg\forall R.A$$

It is easy to prove the following result:

Theorem 1. *Let K be any instance of the axiom schema* Kur *in $\mathcal{L}$ and let $\underline{K} = \langle P, \leq, \rho, \iota\rangle$ be a $\mathcal{KALC}$-model for $\mathcal{L}$, then $\rho \Vdash K$.* □

By the above theorem we immediately get that every instance of Kur is $\mathcal{KALC}$-valid. On the other hand, let us consider the logic $\mathcal{I}$ consisting of the formulas valid over Kripke models with arbitrary posets. Since the model of Figure 1 is a counter-model for Kur, this principle is not valid in $\mathcal{I}$ and, since it is valid in every finite Kripke model, $\mathcal{I}$ fails to have the finite model property. We remark that, by now, we have not proved that $\mathcal{KALC}$ has the finite model property: however we conjecture that this holds.

According with the above considerations, we think that the Kripke-style semantics for description logics obtained by directly translating the Kripke semantics for Intuitionistic first order logic as described in [5] is not adequate in the description logic context.

Now, let us consider the logic $\mathcal{CALC}^c$ of [11, 12]. This logic is based on a semantics directly inspired by the Kripke style semantics for Intuitionistic modal logics. It is not easy to compare $\mathcal{CALC}^c$ directly with $\mathcal{KALC}$, since the former also addresses paraconsistency issues by defining a paraconsistent negation. Moreover, as the Kur principle is related to the notion of consistency, it is not clear how to transpose it in the paraconsistent context of $\mathcal{CALC}^c$.

To conclude this section, we remark that the Kur axiom schema corresponds to the first order principle $\forall x.\neg\neg H(x) \rightarrow \neg\neg\forall x.H(x)$. This principle is known in the literature of constructive logics as *Kuroda principle* [7, 14]. Adding this schema to Intuitionistic first order logic **Int**, we get a proper extension of **Int**, that satisfies the *disjunction property* (if $A \vee B$ is provable 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). An important property of this logic is that a theory T is classically consistent iff it is consistent w.r.t. Kuroda Logic.

4 Tableau calculus for $\mathcal{KALC}$

In this section we introduce the tableau calculus $\mathcal{T}$ for $\mathcal{KALC}$. The calculus works on *signed formulas*, namely expressions of the kind $\mathbf{T}H$, $\mathbf{F}H$ or $\mathbf{F_c}H$ where H is a closed formula of $\mathcal{L}$. The semantics of formulas extends to signed formulas as follows. Given a $\mathcal{KALC}$-model $\underline{K} = \langle P, \leq, \rho, \iota\rangle$ for $\mathcal{L}$, $\alpha \in P$ and a signed formula W of $\mathcal{L}_\alpha$, *α realizes W in $\underline{K}$*, and we write $\underline{K}, \alpha \rhd W$, iff one of the following conditions hold:

[1] Saying that Kur is an axiom schema, we mean that R and A are meta-variables ranging over role names and concepts respectively. An instance of the axiom schema in $\mathcal{L}$ is obtained by replacing R with a role name of $\mathcal{L}$ and A with concept of $\mathcal{L}$.

– $W = \mathbf{T}H$ and $\alpha \| \!\!- H$ in $\underline{K}$;
– $W = \mathbf{F}H$ and $\alpha \| \!\!\not- H$ in $\underline{K}$;
– $W = \mathbf{F_c}H$ and for every $\beta \in P$ such that $\alpha \leq \beta$, $\beta \| \!\!\not- H$ in $\underline{K}$.

We remark that, for a concept formula H, we have $\underline{K}, \alpha \rhd \mathbf{F_c}H$ iff $\alpha \| \!\!- \neg H$ in $\underline{K}$. By the monotonicity property, we get that $\mathbf{T}$-signed and $\mathbf{F_c}$-signed formulas are *persistent*: namely, if $\alpha, \beta \in P$ and $\alpha \leq \beta$, then $\underline{K}, \alpha \rhd \mathbf{T}A$ implies $\underline{K}, \beta \rhd \mathbf{T}A$, and $\underline{K}, \alpha \rhd \mathbf{F_c}A$ implies $\underline{K}, \beta \rhd \mathbf{F_c}A$. Given a set of signed formulas S, its *persistent part* S_p consists of the persistent signed formulas of S, formally:

$$S_p = \{\mathbf{T}H \mid \mathbf{T}H \in S\} \cup \{\mathbf{F_c}H \mid \mathbf{F_c}H \in S\}$$

The tableau calculus $\mathcal{T}$ consists of the rules in Figures 2 and 3. In the rules we use the notation S, H, with S a set of signed formulas and H a signed formula, to denote the set $S \cup \{H\}$, where we assume that $H \notin S$. We remark that the rules $\mathbf{F}$-*concept*, $\mathbf{T}\exists$, $\mathbf{F}\forall$ and $\mathbf{F_c}\forall$ (marked with *) introduce in the conclusion a *parameter* $p \in \mathtt{NI}$ which must be new. That is, applying a rule, we instantiate p with an individual name not occurring in the premise of the rule. We also notice that some of the rules, e.g. $\mathbf{F}\rightarrow$ and $\mathbf{F}\forall$, restrict the set S occurring in the premise to its persistent part S_p. The rule *At* can be applied when the premise S only contains $\mathbf{F}$-signed formulas.

A *proof table* for a finite set of signed formulas S is a tree τ such that:

– the root of τ is S;
– given a node S' of τ, the successors $S'_1, \ldots, S'_n$ of S' in τ are the consequences of an instance of a rule having S' as premise.

A set S of signed formulas is *contradictory* if either $\mathbf{T}\bot \in S$, $\{\mathbf{T}H, \mathbf{F}H\} \subseteq S$, or $\{\mathbf{T}H, \mathbf{F_c}H\} \subseteq S$ for some formula H. Clearly, contradictory sets are not realizable. When all the leaves of a proof table τ are contradictory, we say that τ is *closed*. A formula H is *provable* in $\mathcal{T}$ iff there exists a closed proof table for $\{\mathbf{F}H\}$.

We remark that the calculus $\mathcal{T}$ is essentially inspired by the calculus for Kuroda logic of [10] and by the tableau calculi for Intuitionistic logic [1, 8], with an efficient treatment of duplications. In particular the rules of Figure 3 are related to the treatment of $\mathbf{T}$-signed implications avoiding duplications on the purely propositional part of the logic. This treatment of implication is a key point in the development of an "efficient" decision procedure for the logic. Finally, we remark that the rule *At* could be reformulated without the side condition without affecting the soundness and completeness of the calculus. However the restricted rule is better if we are interested in an efficient proof search strategy since it reduces the non-determinism of the calculus.

It can be proved that the calculus $\mathcal{T}$ is sound and complete w.r.t. $\mathcal{KALC}$ semantics: the proof can be developed in a way similar to the one of [10] for *Kuroda Logic*.

Theorem 2 (Completeness). *Let Γ be a finite set of closed formulas of $\mathcal{L}$ and let H be a closed formula of $\mathcal{L}$. $\Gamma \models^{\mathrm{k}} H$ iff there exists a closed proof table for $\{\mathbf{T}A \mid A \in \Gamma\} \cup \{\mathbf{F}H\}$.* □

$$\frac{S, \mathbf{T}(A)}{S, \mathbf{T}(t:A), \mathbf{T}(A)}\,\mathbf{T}\text{-}concept \qquad \frac{S, \mathbf{F}(A)}{S, \mathbf{F}(p:A)}\,\mathbf{F}\text{-}concept^{*} \qquad \frac{S}{S_p}\,At \quad \text{if the only } \mathbf{F}\text{-signed formulas of } S \text{ are atomic.}$$

$$\frac{S, \mathbf{T}(t:A \sqcap B)}{S, \mathbf{T}(t:A), \mathbf{T}(t:B)}\,\mathbf{T}\sqcap \qquad \frac{S, \mathbf{F}(t:A \sqcap B)}{S, \mathbf{F}(t:A) \mid S, \mathbf{F}(t:B)}\,\mathbf{F}\sqcap \qquad \frac{S, \mathbf{F_c}(t:A \sqcap B)}{S_p, \mathbf{F_c}(t:A) \mid S_p, \mathbf{F_c}(t:B)}\,\mathbf{F_c}\sqcap$$

$$\frac{S, \mathbf{T}(t:A \sqcup B)}{S, \mathbf{T}(t:A) \mid S, \mathbf{T}(t:B)}\,\mathbf{T}\sqcup \qquad \frac{S, \mathbf{F}(t:A \sqcup B)}{S, \mathbf{F}(t:A), \mathbf{F}(t:B)}\,\mathbf{F}\sqcup \qquad \frac{S, \mathbf{F_c}(t:A \sqcup B)}{S, \mathbf{F_c}(t:A), \mathbf{F_c}(t:B)}\,\mathbf{F_c}\sqcup$$

See Figure 3 $\qquad \frac{S, \mathbf{F}(t:A \to B)}{S_p, \mathbf{T}(t:A), \mathbf{F}(t:B)}\,\mathbf{F}\to \qquad \frac{S, \mathbf{F_c}(t:A \to B)}{S_p, \mathbf{T}(t:A), \mathbf{F_c}(t:B)}\,\mathbf{F_c}\to$

$$\frac{S, \mathbf{T}(t:\neg A)}{S, \mathbf{F_c}(t:A)}\,\mathbf{T}\neg \qquad \frac{S, \mathbf{F}(t:\neg A)}{S_p, \mathbf{T}(t:A)}\,\mathbf{F}\neg \qquad \frac{S, \mathbf{F_c}(t:\neg A)}{S_p, \mathbf{T}(t:A)}\,\mathbf{F_c}\neg$$

$$\frac{S, \mathbf{T}(t:\exists R.A)}{S, \mathbf{T}((t,p):R), \mathbf{T}(p:A)}\,\mathbf{T}\exists^{*}$$

$$\frac{S, \mathbf{F}(t:\exists R.A), \mathbf{T}((t,s):R)}{S, \mathbf{T}((t,s):R), \mathbf{F}(s:A)}\,\mathbf{F}\exists \qquad \frac{S, \mathbf{F_c}(t:\exists R.A), \mathbf{T}((t,s):R)}{S, \mathbf{T}((t,s):R), \mathbf{F_c}(s:A), \mathbf{F_c}(t:\exists R.A)}\,\mathbf{F_c}\exists$$

$$\frac{S, \mathbf{T}(t:\forall R.A), \mathbf{T}((t,s):R)}{S, \mathbf{T}((t,s):R), \mathbf{T}(s:A), \mathbf{T}(t:\forall R.A)}\,\mathbf{T}\forall$$

$$\frac{S, \mathbf{F}(t:\forall R.A)}{S_p, \mathbf{T}((t,p):R), \mathbf{F}(p:A)}\,\mathbf{F}\forall^{*} \qquad \frac{S, \mathbf{F_c}(t:\forall R.A)}{S_p, \mathbf{T}((t,p):R), \mathbf{F_c}(p:A)}\,\mathbf{F_c}\forall^{*}$$

Fig. 2. The rules of calculus $\mathcal{T}$

The classical problems on description logics can be restated in $\mathcal{T}$ as follows:

- *Concept validity*
  Given a concept A and a TBox Γ, determine if $\Gamma \models^{\mathrm{k}} A$. This holds iff there exists a closed proof table for $\{\mathbf{T}(K) \mid K \in \Gamma\} \cup \{\mathbf{F}(A)\}$.
- *Subsumption*
  Given a TBox Γ, determine if $\Gamma \models^{\mathrm{k}} A \sqsubseteq B$ holds. This holds iff there exists a closed proof table for $\{\mathbf{T}(K) \mid K \in \Gamma\} \cup \{\mathbf{F}(A \to B)\}$.
- *Instance checking*
  Given a knowledge base Γ and an ABox assertion H, check if $\Gamma \models^{\mathrm{k}} H$. This holds iff there exists a closed proof table for $\{\mathbf{T}(K) \mid K \in \Gamma\} \cup \{\mathbf{F}(H)\}$.

As for the problem of *concept satisfiability*, that is the problem to determine if there exists a $\mathcal{KALC}$-model for a concept A given a TBox Γ, by the considerations made at the end of the previous section, it can be reduced to the analogous problem for $\mathcal{ALC}$.

$$\frac{S_p, \mathbf{T}(t : A \to B)}{S_p, \mathbf{F_c}(t : A) \mid S_p, \mathbf{T}(t : B)}\,\mathbf{T}\to certain$$

$$\frac{S, \mathbf{T}(t : A \to B)}{S, \mathbf{F}(t : A) \mid S, \mathbf{T}(t : B)}\,\mathbf{T}\to At \quad \text{if } A \text{ is atomic or negated}$$

$$\frac{S, \mathbf{T}(t : (A \sqcap B) \to C)}{S, \mathbf{T}(t : A \to (B \to C))}\,\mathbf{T}\to\sqcap \qquad \frac{S, \mathbf{T}(t : (A \sqcup B) \to C)}{S, \mathbf{T}(t : A \to C), \mathbf{T}(t : B \to C)}\,\mathbf{T}\to\sqcup$$

$$\frac{S, \mathbf{T}(t : (A \to B) \to C)}{S, \mathbf{F}(t : A \to B), \mathbf{T}(t : B \to C) \mid S, \mathbf{T}(t : C)}\,\mathbf{T}\to\to$$

$$\frac{S, \mathbf{T}(t : \exists R.A \to B), \mathbf{T}((t, s) : R)}{S, \mathbf{T}(s : A), \mathbf{T}(t : B), \mathbf{T}((t, s) : R) \mid S, \mathbf{T}((t, s) : R), \mathbf{F}(s : A), \mathbf{T}(t : \exists R.A \to B)}\,\mathbf{T}\to\exists$$

$$\frac{S, \mathbf{T}(t : \forall R.A \to B)}{S, \mathbf{F}(t : \forall R.A), \mathbf{T}(t : \forall R.A \to B) \mid S, \mathbf{T}(t : B)}\,\mathbf{T}\to\forall$$

Fig. 3. Rules for $\mathbf{T}\to$

Example 1. We consider the following knowledge base describing associations between food and wines, inspired by the classical example of [4]. The TBox T_w introduces the basic properties of food and wines:

$$(Ax_1) \quad \texttt{FOOD} \sqsubseteq \exists \texttt{goesWith}.\texttt{COLOR}$$
$$(Ax_2) \quad \texttt{COLOR} \sqsubseteq \exists \texttt{isColorOf}.\texttt{WINE}$$

Ax_1, which is equivalent to $\texttt{FOOD} \to \exists \texttt{goesWith}.\texttt{COLOR}$, expresses the fact that every food has associated a correct wine color and Ax_2, equivalent to $\texttt{COLOR} \to \exists \texttt{isColorOf}.\texttt{WINE}$, expresses the fact that there exists at least one wine for each wine color. The ABox A_w specifies the following instances and associations:

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

Now, let $\overline{\mathsf{T}}_w = \{\mathbf{T}H \mid H \in \mathsf{T}_w\}$ and $\overline{\mathsf{A}}_w = \{\mathbf{T}H \mid H \in \mathsf{A}_w\}$. Figure 4 provides an example of instance checking for the formula:

$$\texttt{meat} : \texttt{FOOD} \to \exists \texttt{goesWith}.(\texttt{COLOR} \sqcap \exists \texttt{isColorOf}.\texttt{WINE})$$

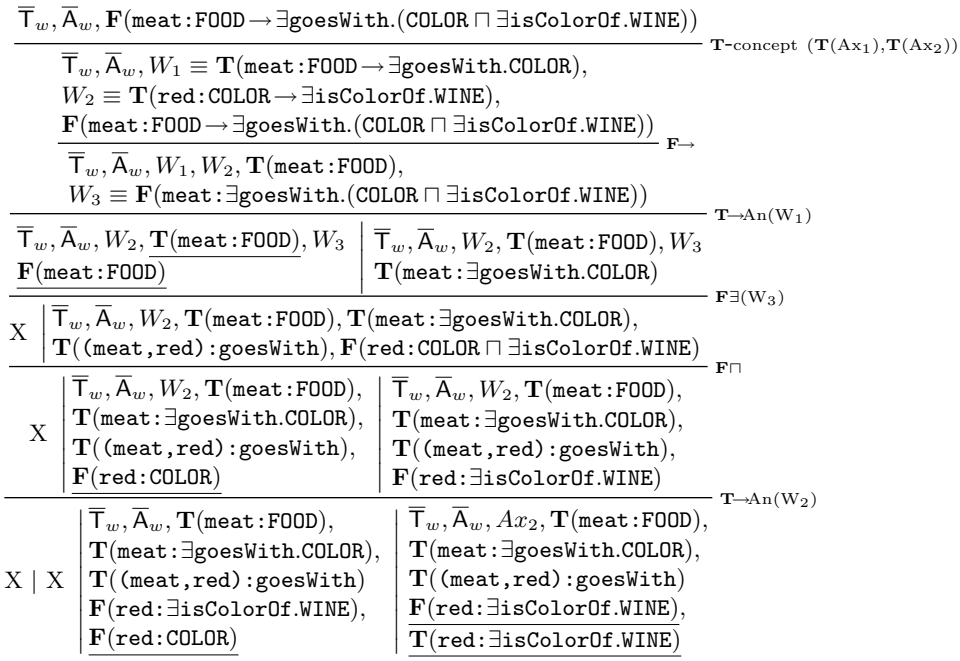

Fig. 4.

Another example is provided by the closed proof table of Figure 5. This proof table proves the subsumption problem for:

$$\texttt{FOOD} \sqsubseteq \exists \texttt{goesWith}.(\texttt{COLOR} \sqcap \exists \texttt{isColorOf}.\texttt{WINE})$$

We remark that, for the sake of conciseness, in the first line of the proof in Figure 4 we simultaneously apply the rules $\mathbf{T}$-*concept* to $\mathbf{T}(Ax_1)$ and $\mathbf{T}(Ax_2)$. In the proofs, we specify with $r(H)$ the fact that a rule r is applied to the signed formula H whenever this is not obvious from the context. We denote with X the closed branches of the proof and we underline the signed formulas causing clashes. $\diamond$

As shown in Figure 6, our calculus allows to exhibit a proof of general instances of the Kuroda principle. We remark that the application of the $\mathbf{F_c}\forall$ rule is essential in order to get a closed proof table. In fact, the $\mathbf{F_c}\forall$ rule directly corresponds to the condition on the final states. That is, without the condition on the partial order, this rule would not be sound, whereas this would be the case for its intuitionistic counterpart:

$$\frac{S, \mathbf{F_c}(t : \forall R.A)}{S_p, \mathbf{T}((t,p) : R), \mathbf{F}(p : A)}\,\mathbf{F_c}\forall'$$

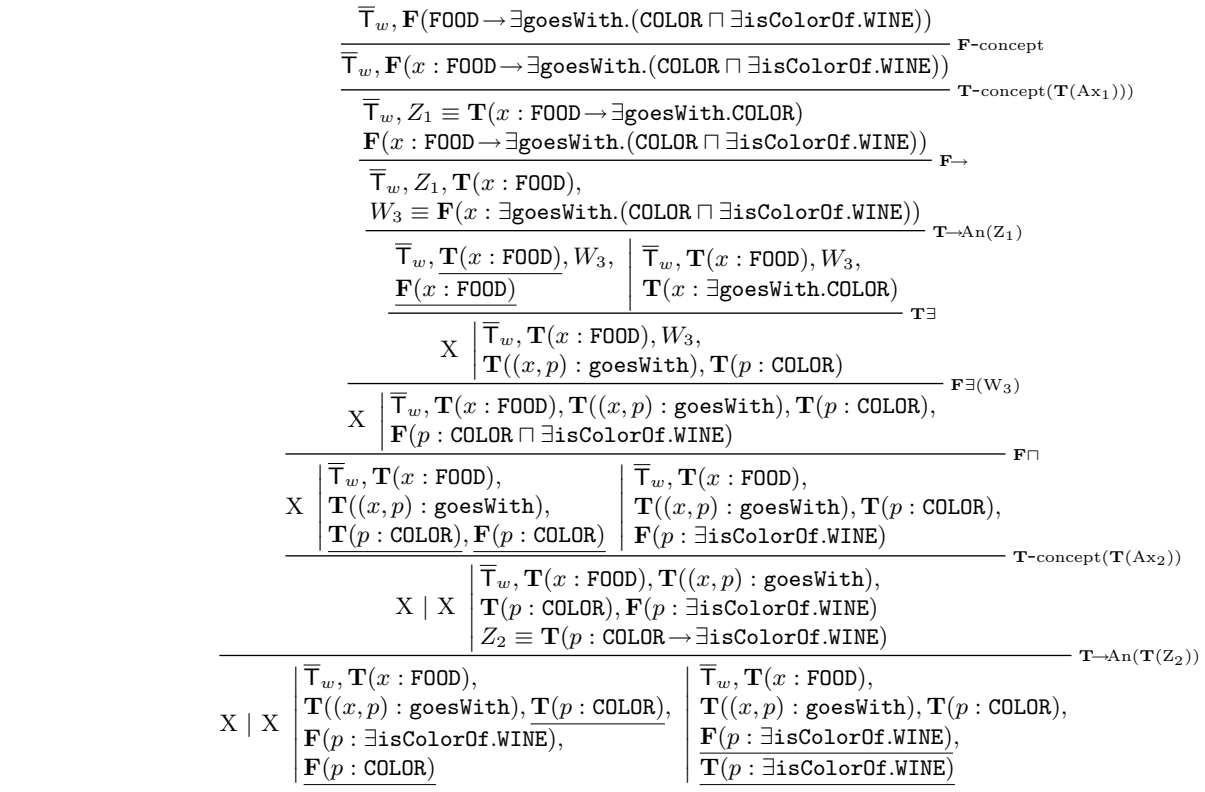

Fig. 5.

However, it can be shown that this intuitionistic version of the rule would not allow to get a closed proof table for the above Kuroda instance.

To conclude this section let us discuss the constructivity of the logic $\mathcal{KALC}$. First of all, we remark that the rule $\mathbf{F}\sqcup$ of $\mathcal{T}$ can be replaced by the following rules:

$$\frac{S, \mathbf{F}(t : A \sqcup B)}{S, \mathbf{F}(t : A)}\mathbf{F}\sqcup_1 \qquad \frac{S, \mathbf{F}(t : A \sqcup B)}{S, \mathbf{F}(t : B)}\mathbf{F}\sqcup_2$$

Namely, the calculus $\mathcal{T}'$ consisting of the rules of $\mathcal{T}$, where the rule $\mathbf{F}\sqcup$ is replaced by the rules $\mathbf{F}\sqcup_1$ and $\mathbf{F}\sqcup_2$, is still sound and complete with respect to the constructive consequence relation $\overset{\mathrm{k}}{\models}$. However, we have not included these rules in $\mathcal{T}$ as they would increase the non-determinism in the proof search. Using $\mathcal{T}'$, we can directly prove that $\mathcal{KALC}$ meets the disjunction property. Indeed, if $A \sqcup B \in \mathcal{KALC}$, by the completeness theorem, there exists a closed proof table for $\mathbf{F}(A \sqcup B)$. By the rules of the calculus $\mathcal{T}'$, we deduce that such a proof has

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathbf{F}(\forall R.\neg\neg A \rightarrow \neg\neg\forall R.A)}{\mathbf{F}(p : \forall R.\neg\neg A \rightarrow \neg\neg\forall R.A)}\,\mathbf{F}\text{-concept}}{\mathbf{T}(p : \forall R.\neg\neg A), \mathbf{F}(p : \neg\neg\forall R.A)}\,\mathbf{F}\rightarrow}{\mathbf{T}(p : \forall R.\neg\neg A), \mathbf{T}(p : \neg\forall R.A)}\,\mathbf{F}\neg}{\mathbf{T}(p : \forall R.\neg\neg A), \mathbf{F_c}(p : \forall R.A)}\,\mathbf{T}\neg}{\mathbf{T}(p : \forall R.\neg\neg A), \mathbf{T}((p,q) : R), \mathbf{F_c}(q : A)}\,\mathbf{F_c}\forall}{\mathbf{T}((p,q) : R), \mathbf{F_c}(q : A), \mathbf{T}(q : \neg\neg A)}\,\mathbf{T}\forall}{\mathbf{T}((p,q) : R), \mathbf{F_c}(q : A), \mathbf{F_c}(q : \neg A)}\,\mathbf{T}\neg}{\mathbf{T}((p,q) : R), \underline{\mathbf{F_c}(q : A)}, \underline{\mathbf{T}(q : A)}}\,\mathbf{F_c}\neg$$

Fig. 6.

one of the following forms:

$$\dfrac{\dfrac{\dfrac{\mathbf{F}(A \sqcup B)}{\mathbf{F}(p : A \sqcup B)}\,\mathbf{F}\text{-}concept}{\mathbf{F}(p : A)}\,\mathbf{F}\sqcup_1}{\Pi_1} \qquad \dfrac{\dfrac{\dfrac{\mathbf{F}(A \sqcup B)}{\mathbf{F}(p : A \sqcup B)}\,\mathbf{F}\text{-}concept}{\mathbf{F}(p : B)}\,\mathbf{F}\sqcup_2}{\Pi_2}$$

where Π_1 and Π_2 stand for closed proof tables. Now, let us suppose that the first case holds. This means that we can build a closed proof table for $\mathbf{F}(A)$ as:

$$\dfrac{\dfrac{\mathbf{F}(A)}{\mathbf{F}(p : A)}\,\mathbf{F}\text{-}concept}{\Pi_1}$$

Since we can build a similar closed proof table even in the second case, the disjunction property holds.

Theorem 3 (Disjunction property). *Let $A \sqcup B \in \mathcal{L}$. If $A \sqcup B \in \mathcal{KALC}$, then either $A \in \mathcal{KALC}$ or $B \in \mathcal{KALC}$.* □

5 Conclusions and future work

We have presented a constructive semantics for description logics inspired by the Kripke semantics for Intuitionistic first order logic. $\mathcal{KALC}$ semantics differs from the direct translation of the Kripke semantics for Intuitionistic first order logic described in [5], since it restricts to Kripke models that force any instance of the axiom schema Kur. This semantics has a close relationship with $\mathcal{BCDL}$ [6],

since the latter can be compared to the fragment of $\mathcal{KALC}$ without implication. Moreover, it can be shown that Kur also holds in $\mathcal{BCDL}$.

We have also introduced a tableau calculus that is sound and complete w.r.t. $\mathcal{KALC}$ semantics. As for the future work, we plan to prove that this logic has the finite model property, in order to use this calculus to give rise to an algorithm for $\mathcal{KALC}$ decidability. Furthermore, we are interested in a closer examination of $\mathcal{KALC}$ relationship with $\mathcal{BCDL}$, in order to define a tableau calculus for this logic.

References

1. A. Avellone, G. Fiorino, and U. Moscato. Optimization techniques for propositional intuitionistic logic and their implementation. *Theoretical Computer Science*, 409(1):41–58, 2008.
2. 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.
3. L. Bozzato, M. Ferrari, C. Fiorentini, and G. Fiorino. A constructive semantics for $\mathcal{ALC}$. In D. Calvanese, E. Franconi, V. Haarslev, D. Lembo, B. Motik, A.-Y. Turhan, and S. Tessaris, editors, *2007 International Workshop on Description Logics*, volume 250 of *CEUR Proceedings*, pages 219–226, 2007.
4. 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. In J. Sowa, editor, *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, pages 401–456. Morgan-Kaufmann, 1991.
5. V. de Paiva. Constructive description logics: what, why and how. Technical report, Xerox Parc, 2003.
6. M. Ferrari, C. Fiorentini, and G. Fiorino. Bcdl: Basic constructive description logic. Submitted to *Journal of Automated Reasoning*, 2009.
7. D.M. Gabbay. *Semantical Investigations in Heyting's Intuitionistic Logic*. Reidel, Dordrecht, 1981.
8. J. Hudelmaier. An $O(n \log n)$-SPACE decision procedure for intuitionistic propositional logic. *Journal of Logic and Computation*, 3(1):63–75, 1993.
9. K. Kaneiwa. Negations in description logic - contraries, contradictories, and subcontraries. In *In Proceedings of the 13th International Conference on Conceptual Structures (ICCS '05)*, pages 66–79. Kassel University Press, 2005.
10. P. Miglioli, U. Moscato, and M. Ornaghi. Avoiding duplications in tableau systems for intuitionistic logic and Kuroda logic. *Logic Journal of the IGPL*, 5(1):145–167, 1997.
11. S.P. Odintsov and H. Wansing. Inconsistency-tolerant description logic. Motivation and basic systems. In V. Hendricks and J. Malinowski, editors, *Trends in Logic. 50 Years of Studia Logica*, pages 301–335. Kluwer Academic Publishers, Dordrecht, 2003.
12. S.P. Odintsov and H. Wansing. Inconsistency-tolerant description logic. Part II: A tableau algorithm for $CALC^C$. *J. Applied Logic*, 6(3):343–360, 2008.
13. M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.
14. A.S. Troelstra, editor. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*. Springer-Verlag, 1973.