SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2017, Vol. 60 112103:1–112103:9
doi: 10.1007/s11432-015-9050-6

Contrary description logic: Gentzen deduction system

Wei LI[1], Yuefei SUI[2*], Jie LUO[1] & Bo CHEN[2]

[1]*State Key Laboratory of Software Development Environment, Beihang University, Beijing* 100191*, China;*
[2]*Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190*, China*

Received January 10, 2017; accepted February 21, 2017; published online August 23, 2017

Abstract Traditional description logics are based on complementary rather than contrary concepts. This work proposes a contrary description logic based on the contrary concept constructors $\sim, \lhd$ (instead of $\neg$). A Gentzen-type deduction system is applied to make the system sound and complete with the three-valued semantics of contrary description logic.

Keywords description logics, contradictory, contrary, soundness, completeness

Citation Li W, Sui Y F, Luo J, et al. Contrary description logic: Gentzen deduction system. Sci China Inf Sci, 2017, 60(11): 112103, doi: 10.1007/s11432-015-9050-6

1 Introduction

General logic consists of two parts: concepts and deductions. Deductions were formalized into mathematical logic in the 1930s, and concepts were formalized into description logics in the 1990s.

The term "contrary" represents a relationship between two propositions. Two propositions are considered contrary if the truth of one proposition implies the falsity of another, even though the falsity of one proposition may not imply the truth of the other [1].

Traditional description logics are based on contradiction (complementary $\neg$) [2–5], wherein two concepts C, D are contradictory if $D = \neg C$ (equivalently, $C = \neg D$). Contrary concepts, such as "good person" and "bad person", are omnipresent. Generally, contradictory relationships can be expressed in terms of contrary relationships. For example, the complementary concepts "not-good person" and "good person" refer to either a bad person or a not-good and not-bad person.

To express the concepts "good person","bad person", and "not-good and not-bad person", we introduce two concept constructors $\lhd C$ and $\sim C$, where C refers to the general concept "person".

In philosophical logic, the contrary operator $\lhd$ is commuted with logical connectives $\sqcap, \sqcup$ in the following way: for any concepts C and D,

$$\lhd(C \sqcap D) \equiv \lhd C \sqcup \lhd D,$$
$$\lhd(C \sqcup D) \equiv \lhd C \sqcap \lhd D.$$

* Corresponding author (email: yfsui@ict.ac.cn)

For example, “good men” is contrary to “bad men”, and “young men” is contrary to “old men”. Broadly speaking,

• “Good young men” is contrary to “bad old men”, with “not-good and not-bad men” and “not-young and not-old men” as the intermediate concepts;

• “Good or young men” is contrary to “bad or old men”, with “not-good and not-bad men” and “not-young and not-old men” as the intermediate concepts.

In this paper, we present a contrary description logic as described below.

• The contrary concept constructor $\lhd$ and the intermediate concept constructor $\sim$ [6] are introduced, so that $C\sqcup \sim C \sqcup \lhd C \equiv \top$ is an axiom, instead of $C \sqcup \neg C \equiv \top$, and the negation $\neg$ is respectively defined in terms of $\sim$ and $\lhd$ as follows:

$$\neg C \equiv \sim C \sqcup \lhd C,$$
$$\neg \sim C \equiv C \sqcup \lhd C,$$
$$\neg \lhd C \equiv C \sqcup \sim C.$$

• The interpretations of $C \sqcap D$, $C \sqcup D$, $\forall R.C$, $\exists R.C$ are defined in terms of the distributivity and de Morgan laws.

• Three-valued models are used in the system.

• A Gentzen-type deduction system **G** [7–10] is given so that

◇ Soundness theorem is true, which means that for any sequent $\Gamma \Rightarrow \Delta$, if $\Gamma \Rightarrow \Delta$ is provable in **G** then $\Gamma \Rightarrow \Delta$ is valid, that is, $\vdash \Gamma \Rightarrow \Delta$ implies $\models \Gamma \Rightarrow \Delta$; and

◇ Completeness theorem is true, which means that for any sequent $\Gamma \Rightarrow \Delta$, if $\Gamma \Rightarrow \Delta$ is valid then $\Gamma \Rightarrow \Delta$ is provable in **G**, that is, $\models \Gamma \Rightarrow \Delta$ implies $\vdash \Gamma \Rightarrow \Delta$; where Γ, Δ are sets of primitive statements.

The rest of the paper is organized as follows. Section 2 presents the basic definitions in contrary description logic $\mathcal{ALC}^{\lhd}$, in which the unary concept constructors $\sim, \lhd$ are introduced, along with their semantics. Section 3 presents the Gentzen deduction system for $\mathcal{ALC}^{\lhd}$ and proves soundness theorem. Section 4 proves completeness theorem, and the last section concludes the paper.

Our standard notation is previously presented in [11].

2 Description logic $\mathcal{ALC}^{\lhd}$

Let L be a logical language for description logic $\mathcal{ALC}^{\lhd}$, which contains the following symbols:

• Constant symbols: $c_0, c_1, \ldots$;

• Atomic concepts: $A_0, A_1, \ldots$;

• Atomic roles: $R_0, R_1, \ldots$;

• Unary concept constructors: $\sim, \lhd$;

• Binary concept constructors: $\sqcap, \sqcup$;

• Quantifier concept constructors: $\exists, \forall$;

• The subsumption relation: $\sqsubseteq$.

Here, concepts are defined by

$$C ::= A| \sim C| \lhd C|C_1 \sqcap C_2|C_1 \sqcup C_2|\exists R.C|\forall R.C.$$

Primitive statements are expressed as

$$\theta ::= C(c)|R(c,d),$$

and other statements are expressed as

$$\varphi ::= \theta|C \sqsubseteq D.$$

A model M is a pair (U, I), where U is a non-empty set, and I is an interpretation such that

• For any constant $c, I(c) \in U$;

• For any atomic concept $A, I(A) : U \to \{-1, 0, +1\}$;

• For any atomic role $R, I(R) \subseteq U^2$.

We define the interpretation C^I of C as

$$C^I = \begin{cases} I(A), & \text{if } C = A \\ f_{\sim}C_1^I, & \text{if } C =\sim C_1, \\ f_{\lhd}C_1^I, & \text{if } C = \lhd C_1, \\ C_1^I \cap C_2^I, & \text{if } C = C_1 \sqcap C_2, \\ C_1^I \cup C_2^I, & \text{if } C = C_1 \sqcup C_2, \\ f_{\forall,R}(C^I), & \text{if } C = \exists R.C, \\ f_{\exists,R}(C^I), & \text{if } C = \forall R.C, \end{cases}$$

where[1)]

$$f_{\sim}(x) = \begin{cases} 0, & \text{if } x = +1, \\ +1, & \text{if } x = 0, \\ -1, & \text{if } x = -1, \end{cases} \qquad f_{\lhd}(x) = \begin{cases} -1, & \text{if } x = +1, \\ 0, & \text{if } x = 0, \\ +1, & \text{if } x = -1, \end{cases}$$

$$f_{\forall,R}I(C_1)(a) = \begin{cases} +1, & \text{if } \mathbf{A}b((a,b) \in I(R) \Rightarrow I(C_1)(b) = +1), \\ -1, & \text{if } \mathbf{A}b((a,b) \in I(R) \Rightarrow (I(C_1)(b) = 0 \text{ or } I(C_1)(b) = -1)), \\ 0, & \text{otherwise}, \end{cases}$$

$$f_{\exists,R}I(C_1)(a) = \begin{cases} +1, & \text{if } \mathbf{E}b((a,b) \in I(R) \& I(C_1)(b) = +1), \\ -1, & \text{if } \mathbf{E}b((a,b) \in I(R) \& (I(C_1)(b) = 0 \text{ or } I(C_1)(b) = -1)), \\ 0, & \text{otherwise}; \end{cases}$$

and

$\cap$	+1	0	−1
+1	+1	0	−1
0	0	0	−1
−1	−1	−1	−1

$\cup$	+1	0	−1
+1	+1	+1	+1
0	+1	0	0
−1	+1	0	−1

Hence, we have the following equivalences:

$$\begin{aligned} &\sim\sim C(c) \equiv C(c), && \lhd \sim C(c) \equiv \lhd C(c), \\ &\sim \lhd C(c) \equiv\sim C(c), && \lhd \lhd C(c) \equiv C(c); \end{aligned}$$

and

$$\begin{aligned} &\sim (C \sqcap D)(c) = (C(c) \wedge \sim D(c)) \vee (\sim C(c) \wedge D(c)) \vee (\sim C(c) \wedge \sim D(c)), \\ &\lhd(C \sqcap D)(c) = (\lhd C)(c) \vee (\lhd D)(c), \\ &\sim (C \sqcup D)(c) = (\sim C(c) \wedge \sim D(c)) \vee (\sim C(c) \wedge \lhd D(c)) \vee (\lhd C(c) \wedge \sim D(c)), \\ &\lhd(C \sqcup D)(c) = \lhd C(c) \wedge \lhd D(c), \\ &\sim (\forall R.C)(c) \equiv (\exists R.C)(c) \wedge (\exists R.(\sim C \sqcup \lhd C))(c), \\ &\lhd(\forall R.C)(c) \equiv (\forall R.(\sim C \sqcup \lhd C))(c), \\ &\sim (\exists R.C)(c) \equiv (\forall R.C)(c) \vee (\forall R.(\sim C \sqcup \lhd C))(c), \\ &\lhd(\exists R.C)(c) \equiv (\exists R.(\sim C \sqcup \lhd C))(c), \end{aligned}$$

where $C(c) \equiv D(c)$ if for any interpretation $I, C^I(I(c)) = +1$ if and only if $D^I(I(c)) = +1$.

The satisfaction $M \models \varphi$ of $\varphi : M \models \varphi$ if

$$\begin{cases} C^I(I(c)) = +1, & \text{if } \varphi = C(c), \\ (I(c), I(d)) \in I(R), & \text{if } \varphi = R(c,d), \\ \mathbf{A}a \in U(C^I(a) = +1 \Rightarrow D^I(a) = +1), & \text{if } \varphi = C \sqsubseteq D. \end{cases}$$

1) In syntax, we use $\neg, \wedge, \rightarrow, \forall, \exists$ to denote logical connectives and quantifiers; and in semantics we use $\sim, \&, \Rightarrow, \mathbf{A}, \mathbf{E}$ to denote the corresponding connectives and quantifiers.

A sequent δ is of form $\Gamma \Rightarrow \Delta$, where Γ and Δ are sets of primitive statements.

A sequent δ is satisfied in M, which is denoted by $M \models \delta$, if $M \models \Gamma$ implies $M \models \Delta$, where $M \models \Gamma$ if for each statement $\varphi \in \Gamma, M \models \varphi$. In addition, $M \models \Delta$ if there is a statement $\psi \in \Delta$ such that $M \models \psi$. δ is valid, as denoted by $\models \delta$, if for any model $M, M \models \delta$.

3 Gentzen deduction system G for $\mathcal{ALC}^{\lhd}$

The deduction rules in $\mathcal{ALC}^{\lhd}$ are different from those followed in the traditional Gentzen deduction system. In a traditional Gentzen system, we put A at the right side of the sequence to eliminate $\neg A$ from the left side of a sequence, and dually, we put A at the left side of the sequence in order to eliminate $\neg A$ from the right side of a sequence. Formally,

$$\frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \neg A, \Delta}, \qquad \frac{\Gamma \Rightarrow A, \Delta}{\Gamma, \neg A \Rightarrow \Delta}.$$

In $\mathcal{ALC}^{\lhd}$, we have $C, \sim C, \lhd C$. In this case, we cannot simply move C from the right side to the left side, or vice versa. In addition, we combine the quantifier and binary concept constructors with the unary concept constructors in order to establish the deduction rules..

The Gentzen deduction system **G** consists of several axioms and deduction rules, which are described below.

Axioms:

$$\begin{array}{lll}
\Gamma, A(c) \Rightarrow A(c), \Delta; & \Gamma, \sim A(c) \Rightarrow \sim A(c), \Delta; & \Gamma, \lhd A(c) \Rightarrow \lhd A(c), \Delta; \\
\Gamma, A(c), \sim A(c) \Rightarrow \Delta; & \Gamma, A(c), \lhd A(c) \Rightarrow \Delta; & \Gamma, \sim A(c), \lhd A(c) \Rightarrow \Delta; \\
\Gamma \Rightarrow A(c), \sim A(c), \lhd A(c), \Delta; & \Gamma, R(c,d) \Rightarrow R(c,d), \Delta. &
\end{array}$$

The axioms are briefly denoted as

$$\begin{array}{ll}
\Gamma, *A(c) \Rightarrow *A(c), \Delta; & \Gamma, *_1 A(c), *_2 A(c) \Rightarrow \Delta; \\
\Gamma \Rightarrow A(c), \sim A(c), \lhd A(c), \Delta; & \Gamma, R(c,d) \Rightarrow R(c,d), \Delta,
\end{array}$$

where $*, *_1, *_2 \in \{\lambda, \sim, \lhd\}$ and $*_1 \neq *_2$.

Deduction rules for unary concept constructors:

$$\begin{array}{llll}
(\sim\sim^L) & \dfrac{\Gamma, C(c) \Rightarrow \Delta}{\Gamma, \sim\sim C(c) \Rightarrow \Delta} & (\sim\sim^R) & \dfrac{\Gamma \Rightarrow C(c), \Delta}{\Gamma \Rightarrow \sim\sim C(c), \Delta} \\
(\sim \lhd^L) & \dfrac{\Gamma, \sim C(c) \Rightarrow \Delta}{\Gamma, \sim \lhd C(c) \Rightarrow \Delta} & (\sim \lhd^R) & \dfrac{\Gamma \Rightarrow \sim C(c), \Delta}{\Gamma \Rightarrow \sim \lhd C(c), \Delta} \\
(\lhd \sim^L) & \dfrac{\Gamma, \lhd C(c) \Rightarrow \Delta}{\Gamma, \lhd \sim C(c) \Rightarrow \Delta} & (\lhd \sim^R) & \dfrac{\Gamma \Rightarrow \lhd C(c), \Delta}{\Gamma \Rightarrow \lhd \sim C(c), \Delta} \\
(\lhd\lhd^L) & \dfrac{\Gamma, C(c) \Rightarrow \Delta}{\Gamma, \lhd \lhd C(c) \Rightarrow \Delta} & (\lhd\lhd^R) & \dfrac{\Gamma \Rightarrow C(c), \Delta}{\Gamma \Rightarrow \lhd \lhd C(c), \Delta}.
\end{array}$$

Deduction rules for binary and quantifier concept constructors:

$$(\sqcap_1^L)\ \frac{\Gamma, C(c) \Rightarrow \Delta}{\Gamma, (C \sqcap D)(c) \Rightarrow \Delta} \qquad (\sqcap^R)\ \frac{\Gamma \Rightarrow C(c), \Delta \quad \Gamma \Rightarrow D(c), \Delta}{\Gamma \Rightarrow (C \sqcap D)(c), \Delta}$$

$$(\sqcap_2^L)\ \frac{\Gamma, D(c) \Rightarrow \Delta}{\Gamma, (C \sqcap D)(c) \Rightarrow \Delta}$$

$$(\sqcup^L)\ \frac{\Gamma, C(c) \Rightarrow \Delta \quad \Gamma, D(c) \Rightarrow \Delta}{\Gamma, (C \sqcup D)(c) \Rightarrow \Delta} \qquad (\sqcup_1^R)\ \frac{\Gamma \Rightarrow C(c), \Delta}{\Gamma, \Rightarrow (C \sqcup D)(c), \Delta}$$

$$(\sqcup_2^R)\ \frac{\Gamma \Rightarrow D(c), \Delta}{\Gamma, \Rightarrow (C \sqcup D)(c), \Delta}$$

$$(\forall^L)\ \frac{\Gamma \Rightarrow R(c,b), \Delta \quad \Gamma, C(b) \Rightarrow \Delta}{\Gamma, (\forall R.C)(c) \Rightarrow \Delta} \qquad (\forall_1^R)\ \frac{\Gamma, R(c,a) \Rightarrow \Delta}{\Gamma \Rightarrow (\forall R.C)(c), \Delta}$$

$$(\forall_2^R)\ \frac{\Gamma \Rightarrow C(a), \Delta}{\Gamma \Rightarrow (\forall R.C)(c), \Delta}$$

$$(\exists_1^L)\ \frac{\Gamma, R(c,a) \Rightarrow \Delta}{\Gamma, (\exists R.C)(c) \Rightarrow \Delta} \qquad (\exists^R)\ \frac{\Gamma \Rightarrow R(c,b), \Delta \quad \Gamma \Rightarrow C(b), \Delta}{\Gamma \Rightarrow (\exists R.C)(c), \Delta}$$

$$(\exists_2^L)\ \frac{\Gamma, C(a) \Rightarrow \Delta}{\Gamma, (\exists R.C)(c) \Rightarrow \Delta},$$

where b is a constant symbol, and a is a constant symbol that does not occur in both Γ and Δ.

Deduction rules for unary and binary/quantifier concept constructors:

$$(\sim \sqcap^L)\ \frac{\begin{array}{c}\Gamma, (C \sqcap \sim D)(c) \Rightarrow \Delta \\ \Gamma, (\sim C \sqcap D)(c) \Rightarrow \Delta \\ \Gamma, (\sim C \sqcap \sim D)(c) \Rightarrow \Delta\end{array}}{\Gamma, \sim (C \sqcap D)(c) \Rightarrow \Delta} \qquad \begin{array}{l} (\sim \sqcap_1^R)\ \dfrac{\Gamma \Rightarrow (C \sqcap \sim D)(c), \Delta}{\Gamma \Rightarrow \sim (C \sqcap D)(c), \Delta} \\ (\sim \sqcap_2^R)\ \dfrac{\Gamma \Rightarrow (\sim C \sqcap D)(c), \Delta}{\Gamma \Rightarrow \sim (C \sqcap D)(c), \Delta} \\ (\sim \sqcap_3^R)\ \dfrac{\Gamma \Rightarrow (\sim C \sqcap \sim D)(c), \Delta}{\Gamma \Rightarrow \sim (C \sqcap D)(c), \Delta} \end{array}$$

$$(\sim \sqcup^L)\ \frac{\begin{array}{c}\Gamma, (\sim C \sqcap \sim D)(c) \Rightarrow \Delta \\ \Gamma, (\sim C \sqcap \lhd D)(c) \Rightarrow \Delta \\ \Gamma, (\lhd C \sqcap \sim D)(c) \Rightarrow \Delta\end{array}}{\Gamma, \sim (C \sqcup D)(c) \Rightarrow \Delta} \qquad \begin{array}{l} (\sim \sqcup_1^R)\ \dfrac{\Gamma \Rightarrow (\sim C \sqcap \sim D)(c), \Delta}{\Gamma \Rightarrow \sim (C \sqcup D)(c), \Delta} \\ (\sim \sqcup_2^R)\ \dfrac{\Gamma \Rightarrow (\sim C \sqcap \lhd D)(c), \Delta}{\Gamma \Rightarrow \sim (C \sqcup D)(c), \Delta} \\ (\sim \sqcup_3^R)\ \dfrac{\Gamma \Rightarrow (\lhd C \sqcap \sim D)(c), \Delta}{\Gamma \Rightarrow \sim (C \sqcup D)(c), \Delta}, \end{array}$$

and

$$(\lhd \sqcap^L)\ \frac{\Gamma, \lhd C(c) \Rightarrow \Delta \quad \Gamma, \lhd D(c) \Rightarrow \Delta}{\Gamma, \lhd (C \sqcap D) \Rightarrow \Delta} \qquad (\lhd \sqcap_1^R)\ \frac{\Gamma \Rightarrow \lhd C(c), \Delta}{\Gamma \Rightarrow \lhd (C \sqcap D)(c), \Delta}$$

$$(\lhd \sqcap_2^R)\ \frac{\Gamma \Rightarrow \lhd D(c), \Delta}{\Gamma \Rightarrow \lhd (C \sqcap D)(c), \Delta}$$

$$(\lhd \sqcup_1^L)\ \frac{\Gamma, \lhd C(c) \Rightarrow \Delta}{\Gamma, \lhd (C \sqcup D)(c) \Rightarrow \Delta} \qquad (\lhd \sqcup^R)\ \frac{\Gamma \Rightarrow \lhd C(c), \Delta \quad \Gamma \Rightarrow \lhd D(c), \Delta}{\Gamma \Rightarrow \lhd (C \sqcup D)(c), \Delta}$$

$$(\lhd \sqcup_2^L)\ \frac{\Gamma, \lhd C(c) \Rightarrow \Delta}{\Gamma, \lhd (C \sqcup D)(c) \Rightarrow \Delta},$$

and

$$(\sim \forall^L)\ \frac{\Gamma, (\exists R.C \sqcap \exists R.(\sim C \sqcup \lhd C))(c) \Rightarrow \Delta}{\Gamma, (\sim \forall R.C)(c) \Rightarrow \Delta} \qquad (\sim \forall^R)\ \frac{\Gamma \Rightarrow (\exists R.C \sqcap \exists R.(\sim C \sqcup \lhd C))(c), \Delta}{\Gamma \Rightarrow (\sim \forall R.C)(c), \Delta}$$

$$(\sim \exists^L)\ \frac{\Gamma, (\forall R.C \sqcup \forall R.(\sim C \sqcup \lhd C))(c) \Rightarrow \Delta}{\Gamma, (\sim \exists R.C)(c) \Rightarrow \Delta} \qquad (\sim \exists^R)\ \frac{\Gamma \Rightarrow (\forall R.C \sqcup \forall R.(\sim C \sqcup \lhd C))(c), \Delta}{\Gamma \Rightarrow (\sim \exists R.C)(c), \Delta}$$

$$(\lhd \forall^L)\ \frac{\Gamma, (\forall R.(\sim C \sqcup \lhd C))(c) \Rightarrow \Delta}{\Gamma, (\lhd \forall R.C)(c) \Rightarrow \Delta} \qquad (\lhd \forall^R)\ \frac{\Gamma \Rightarrow (\forall R.(\sim C \sqcup \lhd C))(c), \Delta}{\Gamma \Rightarrow (\lhd \forall R.C)(c), \Delta}$$

$$(\lhd \exists^L)\ \frac{\Gamma, (\exists R.(\sim C \sqcup \lhd C))(c) \Rightarrow \Delta}{\Gamma, (\lhd \exists R.C)(c) \Rightarrow \Delta} \qquad (\lhd \exists^R)\ \frac{\Gamma \Rightarrow (\exists R.(\sim C \sqcup \lhd C))(c), \Delta}{\Gamma \Rightarrow (\lhd \exists R.C)(c), \Delta}.$$

Definition 1. A sequent δ is provable in $\mathbf{G}$, and is denoted by $\vdash \delta$, if there exists a sequence $\{\delta_1, \ldots, \delta_n\}$ of sequents, such that $\delta_n = \delta$. Then for each $i \leqslant n$, δ_i is either an axiom or is deduced by using a deduction rule from the previous sequents.

Theorem 1 (Soundness theorem)**.** For any sequent δ, if $\vdash \delta$ then $\models \delta$.

Proof. We prove that each axiom is valid and that each deduction rule is capable of preserving the validity.

In order to verify the validity of the axioms, assume that for any model M, $M \models \Gamma, \sim A(c)$. Then, for any model M, $M \models \sim A(c)$, and $M \models \sim A(c), \Delta$. Hence, $\sim A(c), \Delta$ is valid. The same process is followed for other rules covering unary concept constructors.

In order to verify that $(\sim\sim^L)$ preserves the validity, assume that for any model M, $M \models \Gamma, C(c)$ implies $M \models \Delta$. Because $M \models \sim\sim C(c)$ is equivalent to $M \models C(c)$, for any model M, $M \models \Gamma, \sim\sim C(c)$ implies $M \models \Delta$. The same process is followed for other rules covering unary concept constructors.

In order to verify that $(\sim \sqcap_1^R)$ preserves the validity, assume that for any model M, $M \models \Gamma$ implies $M \models (C \sqcap \sim D)(c), \Delta$. Because $M \models (C \sqcap \sim D)(c)$ implies $M \models \sim (C \sqcap D)(c)$, for any model $M, M \models \Gamma$ implies $M \models \sim (C \sqcap D)(c), \Delta$. The same process is followed for other rules covering unary concept constructors.

In order to verify that $(\lhd\forall^R)$ preserves the validity, assume that for any model $M, M \models \Gamma, (\forall R.(\sim C \sqcup \lhd C))(c)$ implies $M \models \Delta$. Because $M \models (\forall R.(\sim C \sqcup \lhd C))(c)$ is equivalent to $M \models (\lhd\forall R.C)(c)$, for any model $M, M \models \Gamma, (\lhd\forall R.C)(c)$ implies $M \models \Delta$. The same process is followed for other rules covering unary concept constructors.

4 Completeness theorem

Theorem 2 (Completeness theorem)**.** For any sequent δ, if $\models \delta$ then $\vdash \delta$.

Proof. Let $\delta = \Gamma \Rightarrow \Delta$. We construct a tree T, wherein

(i) For each branch ξ of T, there exists a sequent $\Gamma' \Rightarrow \Delta'$ at the leaf of ξ, such that $\Gamma' \Rightarrow \Delta'$ is an axiom; or

(ii) There is a model M, such that $M \not\models \Gamma \Rightarrow \Delta$.

Here, T is constructed as follows:

• The root of T is $\Gamma \Rightarrow \Delta$;

• For a node ξ, if for each sequent $\Gamma' \Rightarrow \Delta'$ at ξ, Γ' and Δ' are sets of pseudo-atomic statements, where pseudo-atomic statements are $A(c)|R(c,d)| \sim A(c)| \lhd A(c)$, then the node is a leaf;

• Otherwise, ξ has the direct child node containing the following sequents:

$$\begin{cases} \Gamma_1, C(c) \Rightarrow \Delta_1, & \text{if } \Gamma_1, \sim\sim C(c) \Rightarrow \Delta_1 \in \xi, \\ \Gamma_1 \Rightarrow C(c), \Delta_1, & \text{if } \Gamma_1 \Rightarrow \sim\sim C(c), \Delta_1 \in \xi, \\ \Gamma_1, \sim C(c) \Rightarrow \Delta_1, & \text{if } \Gamma_1, \sim \lhd C(c) \Rightarrow \Delta_1 \in \xi, \\ \Gamma_1 \Rightarrow \sim C(c), \Delta_1, & \text{if } \Gamma_1 \Rightarrow \sim \lhd C(c), \Delta_1 \in \xi, \\ \Gamma_1, \lhd C(c) \Rightarrow \Delta_1, & \text{if } \Gamma_1, \lhd \sim C(c) \Rightarrow \Delta_1 \in \xi, \\ \Gamma_1 \Rightarrow \lhd C(c), \Delta_1, & \text{if } \Gamma_1 \Rightarrow \lhd \sim C(c), \Delta_1 \in \xi, \\ \Gamma_1, C(c) \Rightarrow \Delta_1, & \text{if } \Gamma_1, \lhd \lhd C(c) \Rightarrow \Delta_1 \in \xi, \\ \Gamma_1 \Rightarrow C(c), \Delta_1, & \text{if } \Gamma_1 \Rightarrow \lhd \lhd C(c), \Delta_1 \in \xi, \end{cases}$$

$$\begin{cases} \left[\begin{array}{l} \Gamma_1, C_1(c) \Rightarrow \Delta_1, \\ \Gamma_1, C_2(c) \Rightarrow \Delta_1, \end{array}\right. & \text{if } \Gamma_1, (C_1 \sqcap C_2)(c) \Rightarrow \Delta_1 \in \xi, \\ \left\{\begin{array}{l} \Gamma_1 \Rightarrow C_1(c), \Delta_1, \\ \Gamma_1 \Rightarrow C_2(c), \Delta_1, \end{array}\right. & \text{if } \Gamma_1 \Rightarrow (C_1 \sqcap C_2)(c), \Delta_1 \in \xi, \\ \left\{\begin{array}{l} \Gamma_1, C_1(c) \Rightarrow \Delta_1, \\ \Gamma_1, C_2(c) \Rightarrow \Delta_1, \end{array}\right. & \text{if } \Gamma_1, (C_1 \sqcup C_2)(c) \Rightarrow \Delta_1 \in \xi, \\ \left[\begin{array}{l} \Gamma_1 \Rightarrow C_1(c), \Delta_1, \\ \Gamma_1 \Rightarrow C_2(c), \Delta_1, \end{array}\right. & \text{if } \Gamma_1 \Rightarrow (C_1 \sqcup C_2)(c), \Delta_1 \in \xi, \end{cases}$$

$$
\left\{\begin{array}{ll}
\left\{\begin{array}{l}
\Gamma_1 \Rightarrow R(c,a), \Delta_1,\\
\Gamma_1, C_1(a) \Rightarrow \Delta_1, \ {\scriptstyle a \text{ does not occur in current } T,}\\
\left[\begin{array}{l}\Gamma_2, R(c',a) \Rightarrow \Delta_2,\\ \Gamma_2 \Rightarrow C(a), \Delta_2,\end{array}\right. \ {\scriptstyle \text{for each } \Gamma_2 \Rightarrow (\forall R.C)(c'), \Delta_2 \subseteq \xi,}\\
\left[\begin{array}{l}\Gamma_2, R(c',a) \Rightarrow \Delta_2,\\ \Gamma_2, C(a) \Rightarrow \Delta_2,\end{array}\right. \ {\scriptstyle \text{for each } \Gamma_2, (\exists R.C)(c') \Rightarrow \Delta_2 \subseteq \xi,}
\end{array}\right. & \text{if } \Gamma_1, (\forall R.C_1)(c) \Rightarrow \Delta_1 \in \xi,\\
\left\{\begin{array}{l}
\left[\begin{array}{l}\Gamma_1 \Rightarrow R(c,a), \Delta_1,\\ \Gamma_1 \Rightarrow C_1(a), \Delta_1,\end{array}\right. \ {\scriptstyle a \text{ does not occur in current } T,}\\
\left[\begin{array}{l}\Gamma_2, R(c',a) \Rightarrow \Delta_2,\\ \Gamma_2 \Rightarrow C(a), \Delta_2,\end{array}\right. \ {\scriptstyle \text{for each } \Gamma_2 \Rightarrow (\forall R.C)(c'), \Delta_2 \subseteq \xi,}\\
\left[\begin{array}{l}\Gamma_2, R(c',a) \Rightarrow \Delta_2,\\ \Gamma_2, C(a) \Rightarrow \Delta_2,\end{array}\right. \ {\scriptstyle \text{for each } \Gamma_2, (\exists R.C)(c') \Rightarrow \Delta_2 \subseteq \xi,}
\end{array}\right. & \text{if } \Gamma_1 \Rightarrow (\exists R.C_1)(c), \Delta_1 \in \xi,
\end{array}\right.
$$

$$
\left\{\begin{array}{ll}
\left\{\begin{array}{l}
\Gamma_1, (C_1 \sqcap \sim C_2)(c) \Rightarrow \Delta_1,\\
\Gamma_1, (\sim C_1 \sqcap C_2)(c) \Rightarrow \Delta_1,\\
\Gamma_1, (\sim C_1 \sqcap \sim C_2)(c) \Rightarrow \Delta_1,
\end{array}\right. & \text{if } \Gamma_1, \sim (C_1 \sqcap C_2)(c) \Rightarrow \Delta_1 \in \xi,\\
\left[\begin{array}{l}
\Gamma_1 \Rightarrow (C_1 \sqcap \sim C_2)(c), \Delta_1,\\
\Gamma_1 \Rightarrow (\sim C_1 \sqcap C_2)(c), \Delta_1,\\
\Gamma_1 \Rightarrow (\sim C_1 \sqcap \sim C_2)(c), \Delta_1,
\end{array}\right. & \text{if } \Gamma_1 \Rightarrow \sim (C_1 \sqcap C_2)(c), \Delta_1 \in \xi,\\
\left\{\begin{array}{l}
\Gamma_1, (\sim C_1 \sqcap \sim C_2)(c) \Rightarrow \Delta_1,\\
\Gamma_1, (\sim C_1 \sqcap \lhd C_2)(c) \Rightarrow \Delta_1,\\
\Gamma_1, (\lhd C_1 \sqcap \sim C_2)(c) \Rightarrow \Delta_1,
\end{array}\right. & \text{if } \Gamma_1, \sim (C_1 \sqcup C_2)(c) \Rightarrow \Delta_1 \in \xi,\\
\left[\begin{array}{l}
\Gamma_1 \Rightarrow (\sim C_1 \sqcap \sim C_2)(c), \Delta_1,\\
\Gamma_1 \Rightarrow (\sim C_1 \sqcap \lhd C_2)(c), \Delta_1,\\
\Gamma_1 \Rightarrow (\lhd C_1 \sqcap \sim C_2)(c), \Delta_1,
\end{array}\right. & \text{if } \Gamma_1 \Rightarrow \sim (C_1 \sqcup C_2)(c), \Delta_1 \in \xi,\\
\Gamma_1, (\exists R.C_1 \sqcap \exists R.(\sim C_1 \sqcup \lhd C_1))(c) \Rightarrow \Delta_1, & \text{if } \Gamma_1, (\sim \forall R.C_1)(c) \Rightarrow \Delta_1 \in \xi,\\
\Gamma_1 \Rightarrow (\exists R.C_1 \sqcap \exists R.(\sim C_1 \sqcup \lhd C_1))(c), \Delta_1, & \text{if } \Gamma_1 \Rightarrow (\sim \forall R.C_1)(c), \Delta_1 \in \xi,\\
\Gamma_1, (\forall R.C_1 \sqcup \forall R.(\sim C_1 \sqcup \lhd C_1))(c) \Rightarrow \Delta_1, & \text{if } \Gamma_1, (\sim \exists R.C_1)(c) \Rightarrow \Delta_1 \in \xi,\\
\Gamma_1 \Rightarrow (\forall R.C_1 \sqcup \forall R.(\sim C_1 \sqcup \lhd C_1))(c), \Delta_1, & \text{if } \Gamma_1 \Rightarrow (\sim \exists R.C_1)(c), \Delta_1 \in \xi,
\end{array}\right.
$$

and

$$
\left\{\begin{array}{ll}
\left\{\begin{array}{l}
\Gamma_1, \lhd C_1(c) \Rightarrow \Delta_1,\\
\Gamma_1, \lhd C_2(c) \Rightarrow \Delta_1,
\end{array}\right. & \text{if } \Gamma_1, \lhd (C_1 \sqcap C_2)(c) \Rightarrow \Delta_1 \in \xi,\\
\left[\begin{array}{l}
\Gamma_1 \Rightarrow \lhd C_1(c), \Delta_1,\\
\Gamma_1 \Rightarrow \lhd C_2(c), \Delta_1,
\end{array}\right. & \text{if } \Gamma_1 \Rightarrow \lhd (C_1 \sqcap C_2)(c), \Delta_1 \in \xi,\\
\left[\begin{array}{l}
\Gamma_1, \lhd C_1(c) \Rightarrow \Delta_1,\\
\Gamma_1, \lhd C_2(c) \Rightarrow \Delta_1,
\end{array}\right. & \text{if } \Gamma_1, \lhd (C_1 \sqcup C_2)(c) \Rightarrow \Delta_1 \in \xi,\\
\left\{\begin{array}{l}
\Gamma_1 \Rightarrow \lhd C_1(c), \Delta_1,\\
\Gamma_1 \Rightarrow \lhd C_2(c), \Delta_1,
\end{array}\right. & \text{if } \Gamma_1 \Rightarrow \lhd (C_1 \sqcup C_2)(c), \Delta_1 \in \xi,\\
\Gamma_1, (\forall R.(\sim C_1 \sqcup \lhd C_1))(c) \Rightarrow \Delta_1, & \text{if } \Gamma_1, (\lhd \forall R.C_1)(c) \Rightarrow \Delta_1 \in \xi,\\
\Gamma_1 \Rightarrow (\forall R.(\sim C_1 \sqcup \lhd C_1))(c), \Delta_1, & \text{if } \Gamma_1 \Rightarrow (\lhd \forall R.C_1)(c), \Delta_1 \in \xi,\\
\Gamma_1, (\exists R.(\sim C_1 \sqcup \lhd C_1))(c) \Rightarrow \Delta_1, & \text{if } \Gamma_1, (\lhd \exists R.C_1)(c) \Rightarrow \Delta_1 \in \xi,\\
\Gamma_1 \Rightarrow (\exists R.(\sim C_1 \sqcup \lhd C_1))(c), \Delta_1, & \text{if } \Gamma_1 \Rightarrow (\lhd \exists R.C_1)(c), \Delta_1 \in \xi,
\end{array}\right.
$$

where $\left[\begin{smallmatrix}\delta_1\\ \delta_2\end{smallmatrix}\right.$ indicates that δ_1 and δ_2 are at a same child node, and $\left\{\begin{smallmatrix}\delta_1\\ \delta_2\end{smallmatrix}\right.$ indicates that δ_1 and δ_2 are at different direct child nodes.

Theorem 3. If for each branch $\xi \subseteq T$, there exists a sequent $\Gamma' \Rightarrow \Delta' \in \xi$ at the leaf node of ξ, which is an axiom, then T is a proof tree of $\Gamma \Rightarrow \Delta$.

Proof. By the definition of T, T is a proof tree of $\Gamma \Rightarrow \Delta$.

Theorem 4. If there exists a branch $\xi \subseteq T$, such that each sequent $\Gamma' \Rightarrow \Delta' \in \xi$ at the leaf node of ξ, which is not an axiom, then we find a model M, such that $M \not\models \Gamma \Rightarrow \Delta$.

Proof. Let η be the leaf node of ξ, such that each sequent $\Gamma' \Rightarrow \Delta' \in \eta$ is not an axiom. Let

$$\Theta^L = \bigcup_{\Gamma' \Rightarrow \Delta' \in \xi} \Gamma', \quad \Theta^R = \bigcup_{\Gamma' \Rightarrow \Delta' \in \xi} \Delta'.$$

Then, let U be a set of all the constant symbols occurring in $\Theta^L \cup \Theta^R$. We thus define an interpretation I, such that for any atomic concept symbol A and any $a, b \in U$,

$$I(A)(a) = \begin{cases} +1, & \text{if } A(a) \in \Theta^L, \\ 0, & \text{if } \sim A(a) \in \Theta^L, \\ -1, & \text{if } \lhd A(a) \in \Theta^L, \end{cases}$$
$$(a,b) \in I(R) \text{ iff } R(a,b) \in \Theta^L.$$

Hence, we proved by induction on branch ξ that, for any sequent $\delta' = \Gamma' \Rightarrow \Delta' \in \xi, M \models \Gamma'$ and $M \not\models \Delta'$.

Case 1. $\delta' = \Gamma_2, \sim\sim C_1(c) \Rightarrow \Delta_2 \in \eta$. Then, δ' has a direct child node containing $\Gamma_2, C_1(c) \Rightarrow \Delta_2$. By induction assumption, $M \models \Gamma_2, C_1(c)$ and $M \not\models \Delta_2$, i.e., $M \models \Gamma_2, \sim\sim C_1(c)$ and $M \not\models \Delta_2$.

Similar for cases $\sim\sim^R, \sim\lhd^L, \sim\lhd^R, \lhd\sim^L, \lhd\sim^R, \lhd\lhd^L, \lhd\lhd^R$.

Case 2. $\delta' = \Gamma_2, (C_1 \sqcap C_2)(c) \Rightarrow \Delta_2 \in \eta$. Then, δ' has a direct child node containing $\Gamma_2, C_1(c) \Rightarrow \Delta_2$ and $\Gamma_2, C_2(c) \Rightarrow \Delta_2$. By induction assumption, $M \models \Gamma_2, C_1(c); M \models \Gamma_2, C_2(c)$ and $M \not\models \Delta_2$, i.e., $M \models \Gamma_2, (C_1 \sqcap C_2)(c)$ and $M \not\models \Delta_2$.

Case 3. $\delta' = \Gamma_2 \Rightarrow (C_1 \sqcap C_2)(c), \Delta_2 \in \eta$. Then, δ' has a direct child node containing $\Gamma_2 \Rightarrow C_i(c), \Delta_2$. By induction assumption, $M \models \Gamma_2$ and $M \not\models \Delta_2, C_i(c)$. Hence, $M \models \Gamma_2$ and $M \not\models \Delta_2, (C_1 \sqcap C_2)(c)$.

Case 4. $\delta' = \Gamma_2, (C_1 \sqcup C_2)(c) \Rightarrow \Delta_2 \in \eta$. Then, δ' has a direct child node containing $\Gamma_2, C_i(c) \Rightarrow \Delta_2$. By induction assumption, $M \models \Gamma_2, C_i(c)$ and $M \not\models \Delta_2$. Hence, $M \models \Gamma_2, (C_1 \sqcup C_2)(c)$ and $M \not\models \Delta_2$.

Case 5. $\delta' = \Gamma_2 \Rightarrow (C_1 \sqcup C_2)(c), \Delta_2 \in \eta$. Then, δ' has a direct child node containing $\Gamma_2 \Rightarrow C_1(c), \Delta_2$ and $\Gamma_2 \Rightarrow C_2(c), \Delta_2$. By induction assumption, $M \models \Gamma_2$ and $M \not\models \Delta_2, C_1(c); M \not\models \Delta_2, C_2(c)$. Hence, $M \models \Gamma_2$ and $M \not\models \Delta_2, (C_1 \sqcup C_2)(c)$.

Case 6. $\delta' = \Gamma_2, (\forall R.C_1)(c) \Rightarrow \Delta_2 \in \eta$. Then, δ' has a direct child node containing $\Gamma_2 \Rightarrow R(c,a), \Delta_2$ and $\Gamma_2, C_1(a) \Rightarrow \Delta_2$ for some $a \in U$ not occurring in Γ_2 and Δ_2. By induction assumption, $M \models \Gamma_2$ and $M \not\models R(c,a), \Delta_2$, and $M \models \Gamma_2, C_1(a)$ and $M \not\models \Delta_2$, which imply $M \models \Gamma_2, (\forall R.C_1)(c)$ and $M \not\models \Delta_2$.

Case 7. $\delta' = \Gamma_2 \Rightarrow (\forall R.C_1)(c), \Delta_2 \in \eta$. Then, δ' has a direct child node containing either $\Gamma_2, R(c,b) \Rightarrow \Delta_2$ or $\Gamma_2 \Rightarrow C_1(b), \Delta_2$, where b is a constant symbol in U. By induction assumption, either $M \models \Gamma_2, R(c,b)$ and $M \not\models \Delta_2$, or $M \models \Gamma_2$ and $M \not\models C_1(b), \Delta_2$. For any $b' \in U$, either $\Gamma_2, R(c,b') \Rightarrow \Delta_2 \in \xi$ or $\Gamma_2 \Rightarrow C_1(b'), \Delta_2 \in \xi$. Hence, we find either expression to be true.

$$M \models \Gamma_2, R(c,b') \text{ and } M \not\models \Delta_2,$$

or

$$M \models \Gamma_2 \text{ and } M \not\models C_1(b'), \Delta_2.$$

Thus, $M \models \Gamma_2$ and $M \not\models (\forall R.C_1)(c), \Delta$.

The same is true for other cases. This completes the proof of the theorem.

5 Conclusion

In this paper, a sound and complete Gentzen deduction system is provided for contrary description logic, which is different from traditional Gentzen deduction system (TGDS) in terms of several points discussed below.

• In TGDS, the rules for $\neg$ is moving $\neg A$ in Γ to A in Δ, or vice versa, that is, moving $\neg A$ in Δ to A in Γ. Meanwhile, in **G**, we cannot move $\sim A$ or $\lhd A$ in Γ to A in Δ, or vice versa, because such an action would prevent us from distinguishing $\sim A$ from $\lhd A$, which in turn, induces the following second difference.

• In **G**, there exist rules for two consecutive unary concept constructors (e.g., $\sim\lhd$).

• In TGDS, the rules are given according to the logical connectives or quantifiers and the positions of the logical connectives or quantifiers. Moreover, in **G**, except for these rules, there exist deduction rules for combining the unary contrary concept constructors and binary concept constructors (e.g., $\sim\sqcap^L$)/quantifier concept constructors (e.g., $\sim\forall^R$).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Parsons T. The traditional square of opposition. Stanford Encyclopedia of Philosophy, http://plato.stanford.edu/entries/square
2 Artale A, Franconi E. Temporal description logics. In: Handbook of Temporal Reasoning in Artificial Intelligence. Oxford: Elsevier, 2005, 1: 375–388
3 Baader F, Calvanese D, McGuinness D L, et al. The Description Logic Handbook: Theory, Implementation, Applications. Cambridge: Cambridge University Press, 2003
4 Baader F, Horrocks I, Sattler U. Chapter 3 description logics. In: Handbook of Knowledge Representation. Oxford: Elsevier, 2007
5 Fensel D, van Harmelen F, Horrocks I, et al. OIL: an ontology infrastructure for the semantic web. IEEE Intell Syst, 2001, 16: 38–45
6 Zhu W, Li W, Sui Y, et al. Sound and complete Gentzen deduction system for propositional intermediate logic (in Chinese). J Softw, 2016, 27: 209–218
7 Avron A. Classical Gentzen-type methods in propositional many-valued logics. In: Proceedings of the 31st IEEE International Symposium on Multiple-Valued Logic (ISMVL'01). Washington: IEEE Computer Society, 2001. 287
8 Avron A. On the expressive power of three-valued and four-valued languages. J Logic Comput 1999, 9: 977–994
9 Rademaker A. A Proof Theory for Description Logics. Berlin: Springer, 2012
10 Takeuti G. Proof Theory. 2nd ed. Amsterdam: North-Holland Publishing Company, 1987
11 Li W. Mathematical Logic, Foundations for Information Science. Basel: Birkhäuser, 2010