Studies in Logic, Vol. x, No. x (xxxx): 1–10
PII: 1674-3202(xxxx)-0x-0001-10

A Tableau Algorithm for the Generic Extension of Description Logic*

Ma Li
Philosophy Department, Peking University

mq2828@126.com

Abstract. Description Logics (DLs) model concepts, roles, and individuals. Especially, concept is the main modeling object. DLs describe the intension of concepts implicitly only by means of the subsuption relationship between concepts. As it is known, the intension of a concept not only involves the relationship with other concepts, but also some properties of its own. This paper extend the generic sentences to the basic description logic ALC, and get a new logic G-ALC. A kind of intensional semantics is provided to G-ALC, according to which a tableau algorithm is proposed for the reasoning problems of G-ALC. The important properties such as soundness and completeness of this algorithm are strictly proved.
Keywords: Description Logic, Generic Sentence, Extension, Tableau Algorithm, Soundness, Completeness

1 Introduction

This paper backgrounds on the concept descriptions in the description logics (DLs) [4]. DLs mainly model concepts, roles and individuals, and especially concepts are the important modeling elements. Though concepts' extensions are explicitly interpreted by a collection of objects, the intensions are only dealed implicitly by the subsumption relationship between concepts. As a matter of fact, a concept has many intensions besides the relationship with other concepts. For instance, the concept *bird* has the intension being a subconcept of animal, as well as the intensions such as *fly*, *feathered*, *lay eggs*, $\cdots$. Hence, concepts also need to be interpreted intensionally. In order to describe the knowledge more precisely, DLs need to take care of the concepts' intensions.

To enhance the intensional expressive power of DLs, this paper extend a kind of intensional knowledge, generic sentences [1, 13], to the basic description logic ALC, and get a new logic denoted as G-ALC. G-ALC compounds ALC and the reasoning systems GAG and Gaa for generics [16]. We provide an intensional semantics for G-ALC, of which the frames preserve the properties of that of GAG and Gaa. In this

Received xxxx-xx-xx
*This research is supported by the Major Project of National Social Science Foundation of China (No. 12 & ZD119).

semantics, a concept (a terminology precisely) has four-layer interpretation, namely, sense, extension, intension and concept [11]. Based on this intensional semantics, we propose a tableau algorithm for the reasoning problems of G-ALC. Because that all the typical reasoning problems can be reducted to the satisfiability of a formula w.r.t a knowledge base, it's suffice to ensure the tableau algorithm is effective for this reducted problem.

The tableau algorithm provides a reasoning rule for each kind of G-ALC formula and each of the transforming rules holding intuitively. The important properties of this algorithm such as the soundness and completeness are strictly proved with help of induction method and some other technical ways. The termination property is relatively more obvious, therefore a proof in details is omitted in this paper.

The body part of this article comprise five sections range from 2 to 6. In section 2, we introduces the generic extension G-ALC in syntax, semantics and the knowledge base. In section 3, the tableau algorithm is set up and some properties and notations are talked about. In section 4, the relationship between a G-ALC model and a tableau is shown. This is the basis of the proof the tableau algorithm. In section 5 and 6, the soundness and completeness are proved in details.

2 The generic extension of ALC

In this part, we introduce G-ALC in three aspects: syntax, semantics and the knowledge base based on G-ALC. First, let us have a quick look at the syntax part.

2.1 G-ALC syntax

Language $\mathscr{G}-\mathscr{A}\mathscr{L}\mathscr{C}$. 1. symbols: enumerable variables $x_1, x_2, \cdots$; constants $c_1, c_2, \cdots$; concepts $C_1, C_2, \cdots$; roles $R_1, R_2, \cdots$. 2. concept constructors: $\sim, \sqcap, \sqcup, \forall, \exists, N$. 3. connectives: $\neg, \sqsubseteq, \equiv, G, >$. 4. aux Symbols: $(,)$. Variable set is denoteds as Var, constant set is denoted as Con, concept set is denoted as $\mathcal{C}_{\mathcal{A}}$ (means atomic concepts), role set is denoted as $\mathcal{R}$. The term set $Term = Var \cup Con$.

Concept descriptions in $\mathscr{G}-\mathscr{A}\mathscr{L}\mathscr{C}$ are formed according to the following syntax rule:

$$C, D ::= \top \mid \bot \mid A \mid \sim C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C \mid N(C, D)$$

Particularly, $A \in \mathcal{C}_{\mathcal{A}}, R \in \mathcal{R}, \bot = \sim \top, C \sqcup D = \sim (\sim C \sqcap \sim D), \exists R.C = \sim \forall R. \sim C$, $\mathscr{G}-\mathscr{A}\mathscr{L}\mathscr{C}$ concept description set is denoted as $C_{\mathscr{G}-\mathscr{A}\mathscr{L}\mathscr{C}}$。

Formulae in $\mathscr{G}-\mathscr{A}\mathscr{L}\mathscr{C}$ are formed according to hte following syntax rule:

$$\alpha, \beta ::= C(t) \mid R(t_1, t_2) \mid C \sqsubseteq D \mid C \equiv D \mid G(C, D) \mid \alpha > \beta \mid \neg(\alpha > \beta)$$

Particularly, $t, t_1, t_2 \in Term$, $C, D \in \mathcal{C}_{\mathcal{A}}, R \in \mathcal{R}$. The $\mathscr{G}-\mathscr{A}\mathscr{L}\mathscr{C}$ formula set is denoted as $F_{\mathscr{G}-\mathscr{A}\mathscr{L}\mathscr{C}}$.

2.2 G-ALC semantics

Let's look into the G-ALC frame and structure first.

Definition 1 A G-ALC frame is a tuple $F = \langle W, \mathcal{D}, \mathcal{C}, \mathcal{N}, \circledast\rangle$. W and $\mathcal{D}$ are nonempty sets, $\mathcal{C}$ is a map from $\wp(\mathcal{D})^W$ to $\wp(\wp(\mathcal{D})^W)^W$ called concept generating function. $\mathcal{N}$ is a function ranging over W and $\mathcal{D}$ called the normal individual collecting function. $\circledast$ is a set sellect function over W.

Definition 2 A G-ALC structure is a pair $S = \langle F, \varepsilon\rangle$. F is a G-ALC frame, ε is the sense interpretation for G-ALC constants, concepts and roles on frame F:

(1) For any constant c, $\varepsilon(c) \in \mathcal{D}^W$。 $\varepsilon(c)$ is also written as c^ε.

(2) For any concept C, $\varepsilon(C) \in \wp(\mathcal{D})^W$. $\varepsilon(C)$is also written as C^ε. And for any possible world $w \in W$, $\varepsilon(C)(w)$ is defined as follow.

  (a) $\varepsilon(\top)(w) = \mathcal{D}$.
  (b) $\varepsilon(\sim C)(w) = \mathcal{D} - \varepsilon(C)(w)$.
  (c) $\varepsilon(\bot)(w) = \mathcal{D} - \varepsilon(\top)(w) = \emptyset$.
  (d) $\varepsilon(C_1 \sqcap C_2)(w) = \varepsilon(C_1)(w) \bigcap \varepsilon(C_2)(w)$.
  (e) $\varepsilon(C_1 \sqcup C_2)(w) = \varepsilon(\sim (\sim C_1 \sqcap \sim C_2))(w) = \mathcal{D} - \varepsilon(\sim C_1 \sqcap \sim C_2)(w) = \varepsilon(C_1)(w) \bigcup \varepsilon(C_2)(w)$.
  (f) $\varepsilon(\forall R.C)(w) = \{a \in \mathcal{D} \mid \forall b \in \mathcal{D}((a, b) \in \varepsilon(R)(w) \to b \in \varepsilon(C)(w))\}$.
  (g) $\varepsilon(\exists R.C)(w) = \varepsilon(\sim \forall R. \sim C)(w) = \mathcal{D} - \varepsilon(\forall R. \sim C)(w) = \{a \in \mathcal{D} \mid \exists b \in \mathcal{D}((a, b) \in \varepsilon(R)(w) \wedge b \in \varepsilon(C)(w))\}$.
  (h) $\varepsilon(N(C, D))(w) = \mathcal{N}(C, D)(w) \subseteq \varepsilon(C)(w)$

(3) For any role R, $\varepsilon(R) \in (\wp(\mathcal{D} \times \mathcal{D}))^W$. $\varepsilon(R)$ is also written as R^ε.

Here now we go on to the G-ALC assignment and model.

Definition 3 A G-ALC model is a pair $M = \langle S, \sigma\rangle$. S is a G-ALC structure, σ is an assignment, which assigns an individual sense to each variable, $\sigma : Var \to \mathcal{D}^W$.

Let $F = \langle W, \mathcal{D}, \mathcal{C}, \mathcal{N}, \circledast\rangle$ be a G-ALC frame, structure $S = \langle F, \varepsilon\rangle$ can be written as $\langle W, \mathcal{D}, \mathcal{C}, \mathcal{N}, \circledast, \varepsilon\rangle$; model $M = \langle S, \sigma\rangle$ can be written as $\langle W, \mathcal{D}, \mathcal{C}, \mathcal{N}, \circledast, \varepsilon, \sigma\rangle$. Let $M = \langle W, \mathcal{D}, \mathcal{C}, \mathcal{N}, \circledast, \varepsilon, \sigma\rangle$ be a G-ALC model, $W_M, \mathcal{D}_M, \varepsilon_M, \sigma_M$ be the elements of M, F_M, S_M be frame and stucture respectively.

Definition 4 Let M be a model, σ_M be the assignment of M, $\sigma_M(s/x)$ is a variant of σ_M with respect to x, if and only if, $\sigma_M(s/x) : Var \to \mathcal{D}_M^W$ satisfies the following property.

$$\sigma_M(s/x)(y) = \begin{cases} s & : y = x \\ \sigma_M(y) & : y \neq x \end{cases}$$

Let $M =< S, \sigma >$ be a model, $\sigma_M(d/x)$ is a variant of σ_M. Model $< S, \sigma_M(s/x) >$ is denoted as $M_{(s/x)}$.

Definition 5 Let M be a G-ALC model, t is a term, the interpretation of t in M is t^M, a map defined as follows:

$$t^M = \begin{cases} \varepsilon_M(a) & : t = c \\ \sigma_M(x) & : t = x \end{cases}$$

Definition 6 Let $M = \langle W, \mathcal{D}, \mathcal{C}, \mathcal{N}, \circledast, \varepsilon, \sigma \rangle$ be a G-ALC model, α be a G-ALC formula, $\| \alpha \|^M$ is a set defined as follows:

1. $\| C(t) \|^M = \{w \in W \mid t^M(w) \in C^\varepsilon(w)\}$
2. $\| R(t_1, t_2) \|^M = \{w \in W \mid (t_1^M(w), t_2^M(w)) \in R^\varepsilon(w)\}$
3. $\| C \sqsubseteq D \|^M = \{w \in W \mid C^\varepsilon(w) \subseteq D^\varepsilon(w)\}$
4. $\| C \equiv D \|^M = \| C \sqsubseteq D \|^M \bigcap \| D \sqsubseteq C \|^M$
5. $\| G(C, D) \|^M = g^*(C^\varepsilon, D^\varepsilon) = \{w \in W \mid D^\varepsilon \in (C^\varepsilon)^{\mathcal{C}}\}$
6. $\| \alpha > \beta \|^M = \cup\{X \subseteq W \mid \circledast(X, \| \alpha \|^M) \subseteq \| \beta \|^M\}$
7. $\| \neg(\alpha > \beta) \|^M = W - \| \alpha > \beta \|^M$

2.3 G-ALC knowledge base

Distinct from the ALC knowledge base, which comprises TBox and ABox, a knowledge base based on G-ALC comprises three sub-bases: TBox, ABox and G-Box. TBox is defined as usual in DL, which is a finite set of formulae with the form such as $C \sqsubseteq D, C \equiv D$. ABox is a finite set of formulae with the form such as $C(t), R(t_1, t_2), N(C, D)(t)$. A GBox is a finite set of formuae with the form such as $G(C, D), \alpha > \beta, \neg(\alpha > \beta)$.

3 A tableau algorithm for G-ALC

As we know, all the ALC reasoning problems can be reducted to the concept satisfiability problem. Similarly, all the G-ALC reasoning problems can be reducted to a kind of problem, that is the satisfiability of a G-ALC formula with respect to a knowledge base KB. Because of the page number restriction, the reduction proof is not given in details. The satisfiability of a G-ALC formula α with respect to a G-ALC knowledge base KB is just the consistance of set $KB \bigcup \{\alpha\}$, i.e., decide whether $KB \bigcup \{\alpha\}$ has a G-ALC model. Let's call such problem ”restricted satisfiability”. So, for G-ALC reasoning problems, it is suffice to provide an algorithm for the ”restricted satisfiability” problem. The tableau algorithm specified below is just deserved.

Before proposing the G-ALC tableau algorithm formally, we first show what's tableau algorithm and some important terminologies and notations of it. Tableau algorithm uses a tableau, which indicate the relationship between the semantics and syntax of the problem domain, to decide whether the problem holds. A tableau algorithm comprises a group of rules to build a tableau, as well as a group rules to decide whether the tableau is open or closed.

A tableau is a pair $\mathcal{T} = < N, S >$, while N is a set of nodes, and S a directed binary relation between the nodes called deriving relation, which is irreflexive, asymmetry and intransitive. For every node, there is a set $\mathcal{K}$ of two kind of node expressions with forms of $[i, \alpha]$ and $r(i, \alpha, j)$. Particularly, α is a G-ALC formula,and i, j are natural numbers. The number i in $[i, \alpha]$ called the **tab**[1] for α.

A tableau for the restriced satisfiability of formula α w.r.t knowledge KB is denoted as $\mathcal{T}_{(KB,\alpha)}$. Each tableau is built using the tableau rules step by step. The tableau $\mathcal{T}_{(KB,\alpha)}$ built by n steps, that is to say, using n rules can be denoted as $\mathcal{T}_{(KB,\alpha)}(n)$. Especially, $\mathcal{T}_{(KB,\alpha)}(0)$ only has one node, and the node-expression set is $\mathcal{K}_{(KB,\alpha)}(0) = \{[0, \alpha] \mid \alpha \in \bigcup KB\} \bigcup \{[0, \alpha]\}$.

For every tableau $\mathcal{T}_{(KB,\alpha)} = < N, S >$, there are two kinds of special nodes named **root-node** and **end-node**. The root-node is the node with $\mathcal{K}_{(KB,\alpha)}(0)$ as node-expression set. For any $x \in N$, if there is no $y \in N$ such that Sxy, then x is a end-node of $\mathcal{T}_{(KB,\alpha)}$.

Let $\mathcal{T} = < N, S >$ be a tableau, $\mathcal{B} = < N', S' >$ a sub-tableau of $\mathcal{T}$ ($N' \subseteq N, S' \subseteq S$). If N' contains the root-node u and an end-node v of $\mathcal{T}$, satisfying that $S'^n uv$ $(n \geq 0)$ and $\mathfrak{B}$ contains no other end-node of $\mathcal{T}$, then $\mathcal{B}$ is a **branch** of $\mathcal{T}$. $S'^n uv$ means that there are $x_1, \cdots, x_{n-1}$, such that $S'ux_1 \wedge S'x_1x_2 \wedge \cdots \wedge S'x_{n-1}v$. n is the **length** of $\mathcal{B}$. In convention, $n = 0$ when $S = \emptyset$.

Let $\mathcal{K}$ be a node-expression set of an end-node. All the formulae in $\mathcal{K}$ are of negative normal form. The **formula-rules** are given below.

1. $[\sqcap]$: If $[i, (C_1 \sqcap C_2)(t)] \in \mathcal{K}$ and $[i, C_1(t)] \notin \mathcal{K}$ or $[i, C_2(t)] \notin \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{[i, C_1(t)], [i, C_2(t)]\}$.
2. $[\sqcup]$: If $[i, (C_1 \sqcup C_2)(t)] \in \mathcal{K}$ and $[i, C_1(t)] \notin \mathcal{K}, [i, C_2(t)] \notin \mathcal{K}$, then this node derives two new nodes such that $\mathcal{K}' = \mathcal{K} \bigcup \{[i, C_1(t)]\}$, and $\mathcal{K}'' = \mathcal{K} \bigcup \{[i, C_2(t)]\}$. (In this case, one branch forks two branches.)
3. $[\forall]$: If $[i, \forall R.C(t)] \in \mathcal{K}$, $[i, R(t, t')] \in \mathcal{K}$, and $[i, C(t')] \notin \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{[i, C(t')]\}$.
4. $[\exists]$: If $[i, \exists R.C(t)] \in \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{[i, R(t, c)], [i, C(c)]\}$. ($c$ is a new constant haven't occurred in $\mathcal{K}$.)
5. $[\sqsubseteq]$: If $[i, C_1 \sqsubseteq C_2] \in \mathcal{K}$ and $[i, C_1(t)] \in \mathcal{K}, [i, C_2(t)] \notin \mathcal{K}$, then this node de-

[1] i, j represent possible worlds, and the intuitionistic meaning of $[i, \alpha]$ is that α is true on possible world i. $r(i, \alpha, j)$ is a ternary relation obtained according to the set collection function $\circledast$ in G-ALC frame. The meaning of $r(i, \alpha, j)$ is $w^j \in \circledast(\{w^i\}, \| \alpha \|^{\mathfrak{M}})$ [16].

rives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{[i, C_2(t)], [i, G(C_2, D) > G(C_1, D)]\}$. ($D$ is any concept occurred in $\mathcal{K}$.)

6. $[\equiv]$: If $[i, C_1 \equiv C_2] \in \mathcal{K}$, and $[i, C_1 \sqsubseteq C_2] \notin \mathcal{K}$ or $[i, C_2 \sqsubseteq C_1] \notin \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{[i, C_1 \sqsubseteq C_2], [i, C_2 \sqsubseteq C_1]\}$.
7. $[N]$: If $[i, N(C, D)(t)] \in \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{[i, C(t)]\}$.
8. $[G]$: If $[i, G(C, D)] \in \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{[i, N(C, D)t > D(t)], [i, C(t) > N(C, D)(t)]\}$ (t is a term occurred in $\mathcal{K}$).
9. $[>]$: If $[i, \alpha > \beta] \in \mathcal{K}$ and $r(i, \alpha, j) \in \mathcal{K}, [j, \beta] \notin \mathcal{K}$then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{[j, \beta]\}$.
10. $[\neg >]$: If $[i, \neg(\alpha > \beta)] \in \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{r(i, \alpha, j), [j, \neg\beta]\}$. ($j = m + 1, m$ is the largest tab in $\mathcal{K}$).

The **r-rules** are given below.

1. $[r_{\sqsubseteq}]$: If $r(i, C_1 \sqsubseteq C_2, j) \in \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{r(i, G(C_2, D) > G(C_1, D), j)\}$. ($D$ is any concept occurred in $\mathcal{K}$).
2. $[r_{>}]$: If $r(i, \alpha, j), r(i, \alpha > \beta, j) \in \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{[j, \beta], r(i, \beta, j)\}$.
3. $[r_{\sqcap}]$: If $r(i, (C \sqcap D)(t), j) \in \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{r(i, C(t), j), r(i, D(t), j)\}$.
4. $[r_{\sqcup}]$: If $r(i, (C \sqcup D)(t), j) \in \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{r(i, C(t), j)\}, \mathcal{K}'' = \mathcal{K} \bigcup \{r(i, D(t), j)\}$.
5. $[r_{\forall}]$: If $r(i, \forall R.C(t), j), r(i, R(t, t'), j) \in \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{r(i, C(t'), j)\}$.
6. $[r_{\exists}]$: If $r(i, \exists R.C(t), j) \in \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{r(i, R(t, c), j), r(i, C(c), j)\}$. ($c$ is a new constant haven't occurred in $\mathcal{K}$).
7. $[r_{GAG}]$: If $\{[i, C_1 \sqsubseteq C_2], [i, G(C_2, D)], r(i, C_1 \sqsubseteq C_2, j), r(i, G(C_2, D), j)\} \subseteq \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{[j, G(C_1, D)]\}$.
8. $[r_{ID}]$: If $\{r(i, \alpha, j)\} \subseteq \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{[j, \alpha]\}$.
9. $[r_{TT}]$: If $\{[i, \alpha > \beta], [i, \beta > \gamma]\} \subseteq \mathcal{K}$, then this node derives a new node such that $\mathcal{K}' = \mathcal{K} \bigcup \{[i, \alpha > \gamma]\}$.
10. $[r_{Gaa}]$: If $\{[i, G(C, D)], [i, C(t)], r(i, G(C, D), j), r(i, C(t), j)\} \subseteq \mathcal{K}$, then this node derives a new node such that$\mathcal{K}' = \mathcal{K} \bigcup \{[j, D(t)]\}$.

Regulation of the usage of abovementioned tableau rules: each rule can't be used repeatedly, namely, we can't use a used rule for the same formula more than once.

Definition 7 Let $\mathcal{K}$ be a node-expression set. $\mathcal{K}$ is clashed if there is i, t such that

$\{[i, \bot(t)]\} \subseteq \mathcal{K}$, or i, α such that $\{[i, \alpha], [i, \neg\alpha]\} \subseteq \mathcal{K}$. Otherwise, $\mathcal{K}$ is consistent.

Definition 8 Let $\mathcal{T}$ be a tableau, $\mathcal{B}$ is a branch of $\mathcal{T}$. If there exists clashed node-expression set in $\mathcal{B}$, then $\mathcal{B}$ is a closed branch, otherwise it's open. If all the branched of $\mathcal{T}$ are closed, then $\mathcal{T}$ is closed tableau, otherwise, it's open.

Definition 9 A branch which is closed or has no rules can be used is called finished. A tableau with all branches finished is a finished tableau.

Proposition 1 Let $\mathcal{T}$ be a tableau, u is a node of $\mathcal{T}$, v is derived directly from u, $\mathcal{K}, \mathcal{K}'$ are node-expression set of u, v respectively, then $\mathcal{K} \subseteq \mathcal{K}'$.

Proof It can be proved easily according to the tableau rules. Proof in details is omitted here. □

Corollary 1 Let $\mathcal{T}$ be a tableau, $\mathcal{B}$ is a branch of $\mathcal{T}$ with length of n. $\mathcal{K}_0, \mathcal{K}_1, \cdots, \mathcal{K}_n$ are the node-expression sets of $\mathcal{B}$ from the root-node to the end-node. Then $\mathcal{K}_0 \subseteq \mathcal{K}_1 \subseteq \cdots \subseteq \mathcal{K}_n$.

Proof It can be obtained from proposition 1 obviously. Proof in details is omitted here. □

Proposition 2 Let $\mathcal{T}$ be a tableau, $\mathcal{B}$ is a branch of $\mathcal{T}$ with length of n. $\mathcal{K}_0, \mathcal{K}_1, \cdots, \mathcal{K}_n$ are the node-expression sets of $\mathcal{B}$ from the root-node to the end-node. $\mathcal{B}$ is open, iff, the node-expression set of end-node $\mathcal{K}_n$ is consistent.

Proof $\Longrightarrow$: If $\mathcal{B}$ is open, then $\mathcal{K}_0, \mathcal{K}_1, \cdots, \mathcal{K}_n$ are consistent. Therefore, $\mathcal{K}_n$ is consistent holds obviously.

$\Longleftarrow$: If $\mathcal{B}$ is closed, then there is an inconsistent node-expression set $\mathcal{K}_i$, $(0 \leq i \leq n)$. According to the corollary 1, $\mathcal{K}_i \subseteq \mathcal{K}_n$. So $\mathcal{K}_n$ is inconsistent. □

From this proposition, we can infer that whether a branch $\mathcal{B}$ is open correspondence to the consistency of the end-node expression set. So we call the end-node expression set $\mathcal{K}_n$ the **characteristic set** for branch $\mathcal{B}$, denoded as $\mathcal{K}_{\mathcal{B}}$.

G-ALC tableau algorithm comprises the tableau rules, the regulation of rules and the definition of open (closed) tableau.

4 The relationship between the G-ALC model and tableau

As mentioned above, tableau bridges the syntax aspect and the semantics aspect. In the following part, we will show how it bridges the two sides. This question leads to two branches, with (1) is to construct a G-ALC model according to a end-node expression set on a tableau, and (2) is to unfold the fact that the restricted satisfiablility of a formula α w.r.t a knowledge base KB correspondence to the open (closed) property of the tableau $\mathcal{T}_{(KB,\alpha)}$. As for (2), it's in nature the soundness and completeness of the G-ALC tableau algorithm. Soundness: if formula α is satisfiable w.r.t a knowledge base KB, then the tableau $\mathcal{T}_{(KB,\alpha)}$ is open. Completeness is the converse proposition of soundness: if the tableau $\mathcal{T}_{(KB,\alpha)}$ is open, then formula α is satisfiable w.r.t a knowledge base KB. In this section, we will look into the subquestion (1), and the subquesetion (2) will be left to the two coming sections.

Definition 10 Let $\mathcal{T}$ be a finished tableau, $\mathcal{K}$ a consistent end-node expression set on it. $\mathcal{M}_\mathcal{K} = < \mathcal{W}_\mathcal{K}, \mathcal{D}_\mathcal{K}, \mathcal{N}_\mathcal{K}, \circledast_\mathcal{K}, \varepsilon_\mathcal{K}, \mathcal{C}_\mathcal{K}, \sigma_\mathcal{K} >$ ia called a model based on $\mathcal{K}$, for short, $\mathcal{K}$-model, if $\mathcal{M}_\mathcal{K}$ satisfies the following conditions.

1. $\mathcal{W}_\mathcal{K} = I(\mathcal{K})$, i.e. the set of all the tabs in $\mathcal{K}$.

2. $\mathcal{D}_\mathcal{K} = \{t_i \mid [i, \alpha(t)] \in \mathcal{K}, i \in I(\mathcal{K})\}$. t is a term in $\mathcal{K}$.

3. $\circledast_\mathcal{K} : \wp(\mathcal{W}_\mathcal{K}) \times \wp(\mathcal{W}_\mathcal{K}) \to \wp(\mathcal{W}_\mathcal{K})$ is the set sellection function on $\wp(\mathcal{W}_\mathcal{K})$. $r(i, \alpha, j) \in \mathcal{K}$, iff, $j \in \circledast(\{i\}, \parallel \alpha \parallel)$.

4. $\mathcal{N}_\mathcal{K} \in \wp(\mathcal{D}_\mathcal{K}) \times \wp(\mathcal{D}_\mathcal{K}) \to \wp(\mathcal{D}_\mathcal{K})$. For any $C^{\varepsilon_\mathcal{K}}, D^{\varepsilon_\mathcal{K}} \in \wp(\mathcal{D}_\mathcal{K}), i \in \mathcal{W}_\mathcal{K}$, $\mathcal{N}_\mathcal{K}(C^{\varepsilon_\mathcal{K}}, D^{\varepsilon_\mathcal{K}})(i) = \{t_i \mid [i, N(C, D)(t)] \in \mathcal{K}\}$.

5. $\varepsilon_\mathcal{K}$ is the interpretation of constants, concepts and roles on frame $F_\mathcal{K}$: (1) for any constant c, $c^{\varepsilon_\mathcal{K}} \in {\mathcal{D}_\mathcal{K}}^{W_\mathcal{K}}$, for any $i \in \mathcal{W}_\mathcal{K}$, $c^{\varepsilon_\mathcal{K}}(i) = c_i$, iff, there is a formula α such that $[i, \alpha(c)] \in \mathcal{K}$; (2) for any concept C, $C^{\varepsilon_\mathcal{K}} \in \wp(\mathcal{D}_\mathcal{K})^{W_\mathcal{K}}$, for any $i \in \mathcal{W}_\mathcal{K}$, $C^{\varepsilon_\mathcal{K}}(i) = \{t_i \mid [i, C(t)] \in \mathcal{K}\}$; (3) for any role R, $R^{\varepsilon_\mathcal{K}} \in (\wp(\mathcal{D}_\mathcal{K} \times \mathcal{D}_\mathcal{K}))^{W_\mathcal{K}}$, for any $i \in \mathcal{W}_\mathcal{K}$, $R^{\varepsilon_\mathcal{K}}(i) = \{< t^{\mathcal{M}_\mathcal{K}}(i), t'^{\mathcal{M}_\mathcal{K}}(i) > \mid [i, R(t, t')] \in \mathcal{K}\}$.

6. $\mathcal{C}_\mathcal{K}$ is the concept generating function from $\wp(\mathcal{D}_\mathcal{K})^{W_\mathcal{K}}$ to $\wp(\wp(\mathcal{D}_\mathcal{K})^{W_\mathcal{K}})^{W_\mathcal{K}}$. If $[i, G(C, D)] \in \mathcal{K}$, then $(D)^{\varepsilon_\mathcal{K}} \in C^{\varepsilon_\mathcal{K}\mathcal{C}_\mathcal{K}}(i)$, or else $(D)^{\varepsilon_\mathcal{K}} \notin C^{\varepsilon_\mathcal{K}\mathcal{C}_\mathcal{K}}(i)$. For any sense $s, s' \in \wp(\mathcal{D}_\mathcal{K})^{W_\mathcal{K}}$, any $i \in W_\mathcal{K}$, if $s(i) \subseteq s'(i)$, then for any sense s'', $\circledast(\{i\}, g^*(s', s'')) \subseteq g^*(s, s'')$.

7. $\sigma_\mathcal{K}$ is an assignment for the variables. For any variable x, $x^{\sigma_\mathcal{K}} \in {\mathcal{D}_\mathcal{K}}^{W_\mathcal{K}}$. For any $i \in \mathcal{W}_\mathcal{K}$, $x^{\sigma_\mathcal{K}}(i) = x_i$.

Definition 11 For any G-ALC formula α, $\parallel \alpha \parallel^{\mathcal{M}_\mathcal{K}}$ is a set defined as follow.

1. $\parallel C(t) \parallel^{\mathcal{M}_\mathcal{K}} = \{w \in \mathcal{W}_\mathcal{K} \mid t^{\mathcal{M}_\mathcal{K}}(w) \in C^{\varepsilon_\mathcal{K}}(w)\}$
2. $\parallel R(t_1, t_2) \parallel^{\mathcal{M}_\mathcal{K}} = \{w \in \mathcal{W}_\mathcal{K} \mid (t_1^{\mathcal{M}_\mathcal{K}}(w), t_2^{\mathcal{M}_\mathcal{K}}(w)) \in R^{\varepsilon_\mathcal{K}}(w)\}$
3. $\parallel C \sqsubseteq D \parallel^{\mathcal{M}_\mathcal{K}} = \{w \in \mathcal{W}_\mathcal{K} \mid C^{\varepsilon_\mathcal{K}}(w) \subseteq D^{\varepsilon_\mathcal{K}}(w)\}$
4. $\parallel C \equiv D \parallel^{\mathcal{M}_\mathcal{K}} = \parallel C \sqsubseteq D \parallel^{\mathcal{M}_\mathcal{K}} \cap \parallel D \sqsubseteq C \parallel^{\mathcal{M}_\mathcal{K}}$
5. $\parallel G(C, D) \parallel^{\mathcal{M}_\mathcal{K}} = g^*(C^{\varepsilon_\mathcal{K}}, D^{\varepsilon_\mathcal{K}}) = \{w \in \mathcal{W}_\mathcal{K} \mid D^{\varepsilon_\mathcal{K}} \in (C^{\varepsilon_\mathcal{K}})^{\mathcal{C}_\mathcal{K}}\}$
6. $\parallel \alpha > \beta \parallel^{\mathcal{M}_\mathcal{K}} = \cup\{X \subseteq \mathcal{M}_\mathcal{K} \mid \circledast(X, \parallel \alpha \parallel^{\mathcal{M}_\mathcal{K}}) \subseteq \parallel \beta \parallel^{\mathcal{M}_\mathcal{K}}\}$
7. $\parallel \neg(\alpha > \beta) \parallel^{\mathcal{M}_\mathcal{K}} = \mathcal{M}_\mathcal{K} - \parallel \alpha > \beta \parallel^{\mathcal{M}_\mathcal{K}}$

In conclusion, this section provide a method to construct a G-ALC model from a consistent end-node expression set. Then, let's switch to the subquestion (2), the soundeness and completeness of the G-ALC tableau algorithm.

5 The soundness of the G-ALC tableau algorithm

Let $\mathcal{B}$ be a branch, $I(\mathcal{B})$ is the tab set of $\mathcal{B}$. For any $i \in I(\mathcal{B})$, let $F(\mathcal{B}, i) = \{\alpha \mid [i, \alpha]$ occurs on $\mathcal{B}\}$ called the formula set of tab i. Let $C(\mathcal{B}) = \{F(\mathcal{B}, i) \mid i \in I(\mathcal{B})\}$, namely the set of formula set of all the tabs on $\mathcal{B}$, denoted as $\mathcal{B} - characteristics$. Let $\mathcal{K}$ be a node-expression set, $I(\mathcal{K})$ is the tab set of $\mathcal{K}$. For any $i \in I(\mathcal{K})$, let $F(\mathcal{K}, i) = \{\alpha \mid [i, \alpha]$ occurs in $\mathcal{K}\}$ be the formula set of tab i. Let $C(\mathcal{K}) = \{F(\mathcal{K}, i) \mid i \in I(\mathcal{K})\}$ be the set of formula set of all the tabs of $\mathcal{K}$, denoted as $\mathcal{K} - characteristics$. According to the proposition 1, it's obvious that $C(\mathcal{B}) = C(\mathcal{K}_{\mathcal{B}})$.

Proposition 3 Let $\mathcal{K}$ be a node-expression set. $\mathcal{K}$ is consistent, iff, for all $i \in I(\mathcal{K})$, neither there is a formula α such that $\alpha \in F(\mathcal{K}, i)$ and $\neg\alpha \in F(\mathcal{K}, i)$, nor there is a term t such that $\bot(t) \in F(\mathcal{K}, i)$.

Proof It can be proved by the definition of the consistency of node-expression set. Proof in details is omitted here. □

Definition 12 Let α be a G-ALC formula, $\mathcal{K}$ a node-expression set.

（1） α is satisfiable, iff, there is a G-ALC model $\mathfrak{M}$ and a possible world $w \in W_{\mathfrak{M}}$ such that $w \in \parallel \alpha \parallel^{\mathfrak{M}}$.（Also denoted as $\mathfrak{M}, w \models \alpha$.

（2） $C(\mathcal{K})$ is satisfiable, iff, there is a G-ALC model $\mathfrak{M}$, for any $i \in I(\mathcal{K})$ thereis a possible world $w \in W_{\mathfrak{M}}$ such that $\mathfrak{M}, w \models \bigwedge F(\mathcal{K}, i)$. ($\bigwedge F(\mathcal{K}, i)$ means the conjunction of all the formulae in $F(\mathcal{K}, i)$)

Proposition 4 Let $\mathcal{K}$ be a node-expression set. $C(\mathcal{K})$ is satisfiable, iff, $\mathcal{K}$ is consistent.

Proof It can be proved according to proposition 4 and definition 19, 20. Proof in details is omitted here. □

Proposition 5 Let $\mathcal{K}$ be a node-expression set. $C(\mathcal{K}) = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}, i), \cdots, F(\mathcal{K}, k)\}, I(\mathcal{K}) = \{0, \cdots, k\}$. Let K' be the node-expression set obtained by the application of a tableau rule to K.

(1) Except for $[\neg >]$, all the other tableau rules can not generate new tabs, i.e., $I(\mathcal{K}') = I(\mathcal{K}) = \{0, \cdots, k\}$.

(2) Except for $[>], [\neg >], [r_>], [r_{GAG}], [r_{Gaa}, [r_{ID}]$, all the other tableau rules only add new formulae to $F(\mathcal{K}, i)$, i.e., $F(\mathcal{K}, j) = F(\mathcal{K}', j)$, if $j \neq i$.

(3) $[\neg >]$ can generate new tabs. If the rule $[\neg >]$ is used to $[i, \neg(\beta > \gamma)]$ in $\mathcal{K}$, then $C(\mathcal{K}') = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}, k), F(\mathcal{K}', k+1)\}$, and $F(\mathcal{K}', k+1) = \{\neg\gamma\}$, $I(\mathcal{K}') = \{0, \cdots, k, k+1\}$.

(4) Only the rule $[\sqcup]$ derives two new nodes.

Proof It can be easily proved according to tableau rules. Proof in details is omitted here. □

According to proposition 6, all the tableau rules can be devided into three kinds: 1. $[\sqcup], [r_{\sqcup}]$, which result in forks and do not change the formula set of other tabs; 2. $[\sqcap], [\forall], [\exists], [\sqsubseteq], [\equiv], [N], [G], [r_{\sqcap}], [r_{\forall}], [r_{\exists}], [r_{\sqsubseteq}], [r_{TT}]$, which neither result in forks nor change the formula set of other tabs; 3. $[>], [\neg >], [r_{>}], [r_{GAG}], [r_{Gaa}], [r_{ID}]$, which do not result in forks, but change the formula set of other certain tab. Three propositions will be proved below to show all the tableau rules are satisfiable presev-ing.

Proposition 6 Let $\mathcal{K}$ be a node-expression set, $\mathcal{K}'$ and $\mathcal{K}''$ are obtained after the application of $[\sqcup]$. If $C(\mathcal{K})$ is satisfiable, then so is $C(\mathcal{K}')$ or $C(\mathcal{K}'')$.

Proof Let $C(\mathcal{K}) = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}, k)\}$. $\mathcal{K}'$ and $\mathcal{K}''$ are obtained after the application of the rule $[\sqcup]$ to $[i, (C \sqcup D)(t)]$ in $\mathcal{K}$. So $C(\mathcal{K}') = \{F(\mathcal{K}', 0), \cdots, F(\mathcal{K}', k)\}$, and $F(\mathcal{K}', i) = F(\mathcal{K}, i) \cup \{C(t)\}$. $C(\mathcal{K}'') = \{F(\mathcal{K}'', 0), \cdots, F(\mathcal{K}'', k)\}$, and $F(\mathcal{K}'', i) = F(\mathcal{K}, i) \cup \{D(t)\}$. By the proposition 6.(2), for any $j \neq i, F(\mathcal{K}', j) = F(\mathcal{K}'', j) = F(\mathcal{K}, j)$. If $C(\mathcal{K})$ is satisfiable, then there is a model M and possible world w such that $M, w \models (C \sqcup D)(t)$. By the G-ALC semantics, $M, w \models C(t)$ or $M, w \models D(t)$. In addition, the rule $[\sqcup]$ do not change the other tab's formula set, so $C(\mathcal{K}')$ is satisfiable or $C(\mathcal{K}'')$ is satisfiable. □

Proposition 7 Let $\mathcal{K}$ be a node-expression set. $\mathcal{K}'$ is obtained after the application of one of the following rules $[\sqcap], [\forall], [\exists], [\sqsubseteq], [\equiv], [N], [G], [r_{TT}]$. If $C(\mathcal{K})$ is satisfiable, then so is $C(\mathcal{K}')$.

Proof Let $C(\mathcal{K}) = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}, i), \cdots, F(\mathcal{K}, k)\}$, and $I(\mathcal{K}) = \{0, \cdots, k\}$. $\mathcal{K}'$ is the result counterpart after the application of one of the above rules.

As for $[\sqcap]$. Suppose that the rule $[\sqcap]$ is applied to $[i, (C \sqcap D)(t)] \in \mathcal{K}$. Then $C(\mathcal{K}') = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}', i), \cdots, F(\mathcal{K}, k)\}$, and $F(\mathcal{K}', i) = F(\mathcal{K}, i) \cup \{C(t), D(t)\}$. If $C(\mathcal{K})$ is satisfiable, then there is a model $M =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$ and a possible world $w \in W$ such that $M, w \models F(\mathcal{K}, i)$. Because $(C \sqcap D)(t) \in F(\mathcal{K}, i)$, $M, w \models C(t), M, w \models D(t)$. Therefore, $M, w \models F(\mathcal{K}', i)$. For the rule $[\sqcap]$ do not change the formula set of other tabs, thus $C(\mathcal{K}')$ is satisfiable.

As for $[\forall]$. Suppose that the rule $[\forall]$ is allpied to $[i, \forall R.C(t)] \in \mathcal{K}$. Then $C(\mathcal{K}') = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}', i), \cdots, F(\mathcal{K}, k)\}$. For any term t', if $R(t, t') \in F(\mathcal{K}, i)$, then $C(t') \in F(\mathcal{K}', i)$. If $C(\mathcal{K})$ is satisfiable, then there is a model $M =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$, and a possible world $w \in W$ such that $M, w \models F(\mathcal{K}, i)$. Because $\forall R.C(t) \in F(\mathcal{K}, i)$, $M, w \models \forall R.C(t)$, i.e., $t^M(w) \in (\forall R.C)^{\varepsilon}(w) = \{d \in \mathcal{D} \mid \forall d' \in \mathcal{D}(< d, d' >\in R^{\varepsilon}(w) \to d' \in C^{\varepsilon}(w))\}$. For any term t', if $R(t, t') \in F(\mathcal{K}, i)$, then $M, w \models R(t, t')$, namely, $< t, t' >\in R^{\varepsilon}(w)$. Thus $t' \in C^{\varepsilon}(w)$, i.e.,

$M, w \models C(t')$. As a result, $M, w \models F(\mathcal{K}', i)$. For the rule $[\forall]$ do not change the formula set of other tabs, thus $C(\mathcal{K}')$ is satisfiable.

As for $[\exists]$. Suppose that the rule $[\exists]$ is allpied to $[i, \exists R.C(t)] \in \mathcal{K}$. Then $C(\mathcal{K}') = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}', i), \cdots, F(\mathcal{K}, k)\}$, and $F(\mathcal{K}', i) = F(\mathcal{K}, i) \cup \{R(t, c), C(c)\}$. If $C(\mathcal{K})$ is satisfiable, then there is a model $M =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$, and a possible world $w \in W$ such that $M, w \models F(\mathcal{K}, i)$. Because $\exists R.C(t) \in F(\mathcal{K}, i)$, $M, w \models \exists R.C(t)$, i.e., $t^M(w) \in (\exists R.C)^\varepsilon(w) = \{d \in \mathcal{D} \mid \exists d' \in \mathcal{D}(< d, d' > \in R^\varepsilon(w) \wedge d' \in C^\varepsilon(w))\}$. Let $M' =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon', \sigma >$, and $c^{\varepsilon'} \in \{d \in \mathcal{D} \mid < t^\varepsilon(w), d^{\varepsilon'}(w) > \in R^\varepsilon(w) \wedge d^{\varepsilon'}(w) \in C^\varepsilon(w))\}$, all the other interpretations coincide with ε. So $M', w \models R(t, c)$, and $M', w \models C(c)$, i.e., $M', w \models F(\mathcal{K}', i)$. For the rule $[\exists]$ do not change the formula set of other tabs, thus $C(\mathcal{K}')$ is satisfiable.

As for $[\sqsubseteq]$. Suppose that the rule $[\sqsubseteq]$ is allpied to $[i, C \sqsubseteq D] \in \mathcal{K}$. Then $C(\mathcal{K}') = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}', i), \cdots, F(\mathcal{K}, k)\}$. For any term t, if $C(t) \in F(\mathcal{K}, i)$, then $F(\mathcal{K}', i) = F(\mathcal{K}, i) \cup \{D(t)\}$. If $C(\mathcal{K})$ is satisfiable, then there is a model $M =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$, and a possible world $w \in W$ such that $M, w \models F(\mathcal{K}, i)$. Because $C \sqsubseteq D \in F(\mathcal{K}, i)$, $M, w \models C \sqsubseteq D$, i.e., $C^\varepsilon(w) \subseteq D^\varepsilon(w)$. For any term t, if $C(t) \in F(\mathcal{K}, i)$, then $M, w \models C(t)$, i.e., $t^\varepsilon(w) \in C^\varepsilon(w) \subseteq D^\varepsilon(w)$. Therefore, $M, w \models D(t)$, and $M, w \models F(\mathcal{K}', i)$. For the rule $[\sqsubseteq]$ do not change the formula set of other tabs, thus $C(\mathcal{K}')$ is satisfiable.

As for $[\equiv]$. Suppose that the rule $[\equiv]$ is allpied to $[i, C \equiv D] \in \mathcal{K}$. Then $C(\mathcal{K}') = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}', i), \cdots, F(\mathcal{K}, k)\}$, and $F(\mathcal{K}', i) = F(\mathcal{K}, i) \cup \{C \sqsubseteq D, D \sqsubseteq C\}$. If $C(\mathcal{K})$ is satisfiable, then there is a model $M =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$, and a possible world $w \in W$ such that $M, w \models F(\mathcal{K}, i)$. Because $C \equiv D \in F(\mathcal{K}, i)$, $M, w \models C \equiv D$, i.e., $C^\varepsilon(w) \subseteq D^\varepsilon(w)$ and $D^\varepsilon(w) \subseteq C^\varepsilon(w)$. Thus $M, w \models C \sqsubseteq D$ and $M, w \models D \sqsubseteq C$. So $M, w \models F(\mathcal{K}', i)$. For the rule $[\equiv]$ do not change the formula set of other tabs, thus $C(\mathcal{K}')$ is satisfiable.

As for $[N]$. Suppose that the rule $[N]$ is allpied to $[i, N(C, D)(t)] \in \mathcal{K}$. Then $C(\mathcal{K}') = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}', i), \cdots, F(\mathcal{K}, k)\}$, and $F(\mathcal{K}', i) = F(\mathcal{K}, i) \cup \{C(t)\}$. If $C(\mathcal{K})$ is satisfiable, then there is a model $M =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$, and a possible world $w \in W$ such that $M, w \models F(\mathcal{K}, i)$. Because $N(C, D)(t) \in F(\mathcal{K}, i)$, $M, w \models N(C, D)(t)$, i.e., $t^\varepsilon(w) \in \mathcal{N}(C^\varepsilon, D^\varepsilon)(w) \subseteq C^\varepsilon(w)$. Therefore $M, w \models C(t)$, and $M, w \models F(\mathcal{K}', i)$. For the rule $[N]$ do not change the formula set of other tabs, thus $C(\mathcal{K}')$ is satisfiable.

As for $[G]$. Suppose that the rule $[G]$ is allpied to $[i, G(C, D)] \in \mathcal{K}$. Then $C(\mathcal{K}') = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}', i), \cdots, F(\mathcal{K}, k)\}$. For any term t, if t occurs in $F(\mathcal{K}', i)$, then $N(C, D)t > D(t), C(t) > N(C, D)(t) \in F(\mathcal{K}', i)$. If $C(\mathcal{K})$ is satisfiable, then there is a model $M =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$, and a possible world $w \in W$ such that $M, w \models F(\mathcal{K}, i)$. Because $G(C, D) \in F(\mathcal{K}, i)$, $M, w \models G(C, D)$, i.e., $w \in \| G(C, D) \|^M = g^*(C^\varepsilon, D^\varepsilon)$. According to the G-ALC frame condition $N_G : g^*(C^\varepsilon, D^\varepsilon) \subseteq g(C^\varepsilon, D^\varepsilon) = \{w \in W \mid \forall t, \circledast(\{w\}, \| N(C, D)(t) \|^M) \subseteq \| D(t) \|^M\}$, $M, w \models N(C, D)(t) > D(t)$. In addition for the G-ALC frame condition $D_N æg^*(C^\varepsilon, D^\varepsilon) \subseteq \{w \in W \mid \forall t, \circledast(\{w\}, \| C(t) \|^M) \subseteq \| N(C,D)(t) \|^M\}$, $M, w \models C(t) > N(C,D)(t)$. Thus $M, w \models F(\mathcal{K}', i)$. For the rule $[G]$ do not change the formula set of other tabs, thus $C(\mathcal{K}')$ is satisfiable.

As for $[r_{TT}]$. Suppose that the rule $[r_{TT}]$ is allpied to $[i, \alpha > \beta], [i, \beta > \gamma] \in \mathcal{K}$. Then $C(\mathcal{K}') = \{F(\mathcal{K}', 0), \cdots, F(\mathcal{K}', k)\}$, and $F(\mathcal{K}', i) = F(\mathcal{K}, i) \cup \{[i, \alpha > \gamma]\}$. According to the proposition 6.(2), for any $j \neq i, F(\mathcal{K}', j) = F(\mathcal{K}, j)$. If $C(\mathcal{K})$ is satisfiable, then there is a model $M =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$, and a possible world $w \in W$ such that $M, w \models F(\mathcal{K}, i)$. Because $\alpha > \beta, \beta > \gamma \in F(\mathcal{K}, i)$, $M, w \models \alpha > \beta, M, w \models \beta > \gamma$. In addition for transitivity of the G-ALC frame, $M, w \models \alpha > \gamma$. Thus $\mathfrak{M} \models C(\mathcal{K}')$. For the rule $[r_{TT}]$ do not change the formula set of other tabs, thus $C(\mathcal{K}')$ is satisfiable. □

Definition 13 Let $\mathcal{K}$ be a node-expression set, $\mathfrak{M}$ is a model of $C(\mathcal{K})$. $\mathfrak{M}$ is called a **RS model**, if $R(i, \alpha, j) \in \mathcal{K}$, then there are possible worlds $w_i, w_j \in W_{\mathcal{K}}$, such that $w_j \in \circledast(\{w_i\}, \| \alpha \|^{\mathfrak{M}})$. If a RS model models $C(\mathcal{K})$, then $C(\mathcal{K})$ is called RS-satisfiable. $\mathcal{K} - models$ are such RS-models.

Proposition 8 Let $\mathcal{K}$ be a node-expression set, $\mathcal{K}'$ is obtained by applying one of the following rules $[>], [\neg >], [r_\sqcup], [r_\sqcap], [r_\forall], [r_\exists], [r_>], [r_\sqsubseteq], [r_{GAG}], [r_{ID}], [r_{Gaa}]$. If $C(\mathcal{K})$ is RS-satisfiable, then so is $C(\mathcal{K}')$.

Proof As for $[>]$. Suppose that the rule $[>]$ is allpied to $[i, \alpha > \beta], r(i, \alpha, j) \in \mathcal{K}$. Then $C(\mathcal{K}') = \{F(\mathcal{K}', 0), \cdots, F(\mathcal{K}', i), \cdots, F(\mathcal{K}', k)\}$. Particularly, $F(\mathcal{K}', j) = F(\mathcal{K}, j) \cup \{\beta\}$, $0 \leq j \leq k$, and $F(\mathcal{K}', l) = F(\mathcal{K}, l)$, $l \neq j$. If $C(\mathcal{K})$ is RS-satisfiable, for $\alpha > \beta \in F(\mathcal{K}, i)$, so there is a RS-model $\mathfrak{M}$ and a possible world w_i such that $\mathfrak{M}, w_i \models \alpha > \beta$. Thus $\circledast(\{w_i\}, \| \alpha \|^{\mathfrak{M}}) \subseteq \| \beta \|^{\mathfrak{M}}$. Because $\mathfrak{M}$ is a RS-model, and $R(i, \alpha, j) \in \mathcal{K}$, $w_j \in \circledast(\{w_i\}, \| \alpha \|^{\mathfrak{M}})$. Therefore $w_j \in \| \beta \|^{\mathfrak{M}}$, i.e., $\mathfrak{M}_{\mathcal{K}}, w_j \models \beta$. So $\mathfrak{M}_{\mathcal{K}}$ is a RS-model of $C(\mathcal{K}')$. According to the construcure of $\mathfrak{M}_{\mathcal{K}}$, in fact, w_i and w_j are i and j respectively. Let i and j replace w_i and w_j respectively in below.

As for $[\neg >]$. Suppose that the rule $[\neg >]$ is allpied to $[i, \neg(\alpha > \beta)] \in \mathcal{K}$. Then $R(i, \alpha, j), [j, \neg\beta] \in \mathcal{K}'$, and $j = k+1$ is a new tab. Because $C(\mathcal{K}) = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}, i), \cdots, F(\mathcal{K}, k)\}$, $C(\mathcal{K}') = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}, i), \cdots, F(\mathcal{K}, k), F(\mathcal{K}', k+1)\}$, and $F(\mathcal{K}', k+1) = \{\neg\beta\}$. If $C(\mathcal{K})$ is RS-satisfiable, so there is a RS-model $\mathfrak{M}$, for $i, (0 \leq i \leq k)$, $\mathfrak{M} \models F(\mathcal{K}, i)$. For $\neg(\alpha > \beta) \in F(\mathcal{K}, i)$, so there is an i such that $\mathfrak{M}, i \models \neg(\alpha > \beta)$. So $\circledast(\{i\}, \| \alpha \|^{\mathfrak{M}}) \nsubseteq \| \beta \|^{\mathfrak{M}}$, i.e., there is a $w' \in \circledast(\{i\}, \| \alpha \|^{\mathfrak{M}})$, and $w' \notin \| \beta \|^{\mathfrak{M}}$. Thus $w' \in \| \neg\beta \|^{\mathfrak{M}}$. Let $w' = w_{k+1}$, so $\mathfrak{M} \models F(\mathcal{K}', k+1)$. Therefore $\mathfrak{M} \models C(\mathcal{K}')$. Now that $j = k+1$, so there are $i, j \in W_{\mathcal{K}}$ such that $j \in \circledast(\{i\}, \| \alpha \|^{\mathfrak{M}})$. So $C(\mathcal{K}')$ is RS-satisfiable.

As for $[r_\sqcup]$. Suppose that the rule $[r_\sqcup]$ is allpied to $r(i, (C \sqcup D)(t), j) \in \mathcal{K}$. Let $C(\mathcal{K}) = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}, k)\}$, $\mathcal{K}$ß and $\mathcal{K}''$ are obtained after the application of rule $[r_\sqcup]$. So $C(\mathcal{K}') = \{F(\mathcal{K}', 0), \cdots, F(\mathcal{K}', k)\}$, and $F(\mathcal{K}', i) = F(\mathcal{K}, i) \cup$

$\{r(i, C(t), j)\}$, $C(\mathcal{K}'') = \{F(\mathcal{K}'', 0), \cdots, F(\mathcal{K}'', k)\}$, $F(\mathcal{K}'', i) = F(\mathcal{K}, i) \cup \{r(i, D(t), j)\}$. According to the proposition 6.(2), for any $j \neq i, F(\mathcal{K}', j) = F(\mathcal{K}'', j) = F(\mathcal{K}, j)$. If $C(\mathcal{K})$ is RS-satisfiable, for $r(i, (C \sqcup D)(t), j) \in F(\mathcal{K}, i)$, so there is a RS-model $\mathfrak{M}$ and a possible worlds i, j such that $j \in \circledast(\{i\}, \| (C \sqcup D)(t) \|^{\mathfrak{M}}) = \circledast(\{i\}, \| C(t) \|^{\mathfrak{M}}) \cup \circledast(\{i\}, \| D(t) \|^{\mathfrak{M}})$. Therefore, $j \in \circledast(\{i\}, \| C(t) \|^{\mathfrak{M}})$ or $j \in \circledast(\{i\}, \| D(t) \|^{\mathfrak{M}})$. Thus $C(\mathcal{K}')$ is RS-satisfiable, or $C(\mathcal{K}'')$ is RS-satisfiable.

As for $[r_\sqcap]$. Suppose that the rule $[r_\sqcap]$ is allpied to $r(i, (C \sqcap D)(t), j) \in \mathcal{K}$. Let $C(\mathcal{K}) = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}, k)\}$, then $C(\mathcal{K}') = \{F(\mathcal{K}', 0), \cdots, F(\mathcal{K}', k)\}$, and $F(\mathcal{K}', i) = F(\mathcal{K}, i) \cup \{r(i, C(t), j), r(i, D(t), j)\}$. According to the proposition 6.(2), for any $j \neq i, F(\mathcal{K}', j) = F(\mathcal{K}, j)$. If $C(\mathcal{K})$ is RS-satisfiable, for $r(i, (C \sqcap D)(t), j) \in C(\mathcal{K})$, then there is a RS-model $\mathfrak{M}$ and possible worlds i, j such that $j \in \circledast(\{i\}, \| (C \sqcap D)(t) \|^{\mathfrak{M}}) = \circledast(\{i\}, \| C(t) \|^{\mathfrak{M}}) \cap \circledast(\{i\}, \| D(t) \|^{\mathfrak{M}})$. Thus $j \in \circledast(\{i\}, \| C(t) \|^{\mathfrak{M}})$, and $j \in \circledast(\{i\}, \| D(t) \|^{\mathfrak{M}})$. So $C(\mathcal{K}')$ is RS-satisfiable.

As for $[r_\forall]$. Suppose that the rule $[r_\forall]$ is allpied to $r(i, \forall R.C(t), j), r(i, R(t, t'), j) \in \mathcal{K}$. Let $C(\mathcal{K}) = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}, k)\}$, then $C(\mathcal{K}') = \{F(\mathcal{K}', 0), \cdots, F(\mathcal{K}', k)\}$, and $F(\mathcal{K}', i) = F(\mathcal{K}, i) \cup \{r(i, C(t'), j)\}$. According to the proposition 6.(2), for any $j \neq i, F(\mathcal{K}', j) = F(\mathcal{K}, j)$. If $C(\mathcal{K})$ is RS-satisfiable, for $r(i, \forall R.C(t), j), r(i, R(t, t'), j) \in C(\mathcal{K})$, so there is a RS-model $\mathfrak{M}$ and possible worlds i, j such that $j \in (\circledast(\{i\}, \| \forall R.C(t) \|^{\mathfrak{M}}) \cap \circledast(\{i\}, \| R(t, t') \|^{\mathfrak{M}})) \subseteq \circledast(\{i\}, \| C(t') \|^{\mathfrak{M}})$. Thus $j \in \circledast(\{i\}, \| C(t') \|^{\mathfrak{M}})$, and $C(\mathcal{K}')$ is RS-satisfiable.

As for $[r_\exists]$. Suppose that the rule $[r_\exists]$ is allpied to $r(i, \exists R.C(t), j) \in \mathcal{K}$. Let $C(\mathcal{K}) = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}, k)\}$, then $C(\mathcal{K}') = \{F(\mathcal{K}', 0), \cdots, F(\mathcal{K}', k)\}$, and $F(\mathcal{K}', i) = F(\mathcal{K}, i) \cup \{r(i, R(t, c), j), C(c)\}$. According to the proposition 6.(2), for any $j \neq i, F(\mathcal{K}', j) = F(\mathcal{K}, j)$. If $C(\mathcal{K})$ is RS-satisfiable, for $r(i, \exists R.C(t), j) \in C(\mathcal{K})$, so there is a RS-model $\mathfrak{M} =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$ and possible worlds i, j such that $j \in (\circledast(\{i\}, \| \exists R.C(t) \|^{\mathfrak{M}})$. Let $\mathfrak{M}' =< W, \mathcal{D}, \mathcal{N}, \circledast', \mathcal{C}, \varepsilon, \sigma >$, and $\circledast'$ conincides with $\circledast$ except for $j \in \circledast'(\{i\}, R(t, c)), j \in \circledast'(\{i\}, C(c))$. Therefore $\mathfrak{M}'$ is also a RS-model, and $\mathfrak{M}' \models C(\mathcal{K}')$. So $C(\mathcal{K}')$ is RS-satisfiable.

As for $[r_\sqsubseteq]$. Suppose that the rule $[r_\sqsubseteq]$ is allpied to $r(i, C_1 \sqsubseteq C_2, j) \in \mathcal{K}$. Then $C(\mathcal{K}') = \{F(\mathcal{K}, 0), \cdots, F(\mathcal{K}', i), \cdots, F(\mathcal{K}, k)\}$, and $F(\mathcal{K}', i) = F(\mathcal{K}, i) \cup \{G(C_2, D) > G(C_1, D)\}$ with D being any concept occurs in $\mathcal{K}$. If $C(\mathcal{K})$ is RS-satisfiable, then there is a RS-model $\mathfrak{M} =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$, and possible worlds $i, j \in W$ such that $j \in \circledast(\{i\}, \| C_1 \sqsubseteq C_2 \|^{\mathfrak{M}})$. Because $C_1 \sqsubseteq C_2 \in F(\mathcal{K}, i)$, $\mathfrak{M}, w \models C_1 \sqsubseteq C_2$, i.e., $C_1^\varepsilon(w) \sqsubseteq C_2^\varepsilon(w)$. By the inverse proportion between the concept extension and intension in the G-ALC semantics, we know that $(C_2^\varepsilon)^{\mathcal{C}}(w) \sqsubseteq (C_1^\varepsilon)^{\mathcal{C}}(w)$. So for any concept $D, \| G(C_2, D) \|^{\mathfrak{M}} \subseteq \| G(C_1, D) \|^{\mathfrak{M}}$. Thanks to the I_D condition of the G-ALC frame, $\{w \in W \mid \circledast(\{w\}, \| G(C_2, D) \|^{\mathfrak{M}})\} \subseteq \| G(C_2, D) \|^{\mathfrak{M}} \subseteq \| G(C_1, D) \|^{\mathfrak{M}}$. So $\| C_1 \sqsubseteq C_2 \|^{\mathfrak{M}} \subseteq \| G(C_2, D) > G(C_1, D) \|^{\mathfrak{M}}$. In addition to the monotonicity of the set sellection function, we get that $\circledast(\{w\}, \| C_1 \sqsubseteq C_2 \|^{\mathfrak{M}}) \subseteq \circledast(\{w\}, \| G(C_2, D) > G(C_1, D) \|^{\mathfrak{M}})$. Because that $j \in \circledast(\{i\}, \| C_1 \sqsubseteq C_2 \|^{\mathfrak{M}})$, $w_j \in \circledast(\{i\}, \| G(C_2, D) > G(C_1, D) \|^{\mathfrak{M}})$. So $C(\mathcal{K}')$ is RS-satisfiable.

As for $[r_>]$. Suppose that the rule $[r_>]$ is allpied to $r(i,\alpha,j), r(i,\alpha > \beta, j) \in \mathcal{K}$. Let $C(\mathcal{K}) = \{F(\mathcal{K},0),\cdots, F(\mathcal{K},k)\}$, then $C(\mathcal{K}') = \{F(\mathcal{K}',0),\cdots, F(\mathcal{K}',k)\}$, and $F(\mathcal{K}',i) = F(\mathcal{K},i) \cup \{[j,\beta], r(i,\beta,j)\}$. Accordint to the proposition 6.(2), for any $j \neq i, F(\mathcal{K}',j) = F(\mathcal{K},j)$. If $C(\mathcal{K})$ is RS-satisfiable, because $r(i,\alpha,j), r(i,\alpha > \beta, j) \in C(\mathcal{K})$, there is a RS-model $\mathfrak{M} =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$ and possible worlds i, j such that $j \in (\circledast(\{i\}, \| \alpha \|^{\mathfrak{M}}) \cap \circledast(\{i\}, \| \alpha > \beta \|^{\mathfrak{M}}) \subseteq \| \alpha \|^{\mathfrak{M}} \cap \| \alpha > \beta \|^{\mathfrak{M}})$. By the property of the set sellection, we know that $\| \alpha \|^{\mathfrak{M}} \cap \| \alpha > \beta \|^{\mathfrak{M}}) \subseteq \| \beta \|^{\mathfrak{M}}$. So $\mathfrak{M} \models C(\mathcal{K}')$, i.e., $C(\mathcal{K}')$ is RS-satisfiable.

As for $[r_{ID}]$. Suppose that the rule $[r_{ID}]$ is allpied to $r(i,\alpha,j) \in \mathcal{K}$. Let $C(\mathcal{K}) = \{F(\mathcal{K},0),\cdots, F(\mathcal{K},k)\}$, then $C(\mathcal{K}') = \{F(\mathcal{K}',0),\cdots, F(\mathcal{K}',k)\}$, and $F(\mathcal{K}',i) = F(\mathcal{K},i) \cup \{[j,\alpha]\}$. According to the proposition 6.(2), for any $j \neq i, F(\mathcal{K}',j) = F(\mathcal{K},j)$. If $C(\mathcal{K})$is RS-satisfiable, and $r(i,\alpha,j) \in C(\mathcal{K})$, so there is a RS-model $\mathfrak{M} =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$ and possible worlds i, j such that $j \in \circledast(\{i\}, \| \alpha \|^{\mathfrak{M}}$. By the ID property of the G-ALC frame, we know that $\circledast(\{i\}, \| \alpha \|^{\mathfrak{M}} \subseteq \| \alpha \|^{\mathfrak{M}}$. So $j \in \| \alpha \|^{\mathfrak{M}}$, therefore, $\mathfrak{M} \models C(\mathcal{K}')$. Consequently $C(\mathcal{K}')$ is RS-satisfiable.

As for $[r_{Gaa}]$. Suppose that the rule $[r_{Gaa}]$ is allpied to $[i, G(C,D)], [i, C(t)], r(i, G(C,D), j), r(i, C(t), j) \in \mathcal{K}$. Let $C(\mathcal{K}) = \{F(\mathcal{K},0),\cdots, F(\mathcal{K},k)\}$, then $C(\mathcal{K}') = \{F(\mathcal{K}',0),\cdots, F(\mathcal{K}',k)\}$, and $F(\mathcal{K}',i) = F(\mathcal{K},i) \cup \{[j, D(t)]\}$. For the reason of proposition 6.(2), for any $j \neq i, F(\mathcal{K}',j) = F(\mathcal{K},j)$. If $C(\mathcal{K})$ is RS-satisfiable, as $G(C,D), C(t), r(i, G(C,D), j), r(i, C(t), j) \in C(\mathcal{K})$, so there is a RS-model $\mathfrak{M} =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$ and possible worlds i, j such that $\mathfrak{M}, i \models G(C,D), \mathfrak{M}, i \models C(t)$, $j \in (\circledast(\{i\}, \| G(C,D) \|^{\mathfrak{M}} \cap \circledast (\{i\}, \| C(t) \|^{\mathfrak{M}})$. On the grounds of the Gaa property of G-ALC frame, $(\circledast(\{i\}, \| G(C,D) \|^{\mathfrak{M}} \cap \circledast (\{i\}, \| C(t) \|^{\mathfrak{M}}) \subseteq \| \alpha \|^{\mathfrak{M}}$. Hence $j \in \| D(t) \|^{\mathfrak{M}}$, and $\mathfrak{M} \models C(\mathcal{K}')$. Consequenctly $C(\mathcal{K}')$ is RS-satisfiable.

As for $[r_{GAG}]$. Suppose that the rule $[r_{GAG}]$ is allpied to $[i, C_1 \sqsubseteq C_2], [i, G(C_2, D)], r(i, C_1 \sqsubseteq C_2, j), r(i, G(C_2, D), j) \in \mathcal{K}$. Let $C(\mathcal{K}) = \{F(\mathcal{K},0),\cdots, F(\mathcal{K},k)\}$, then $C(\mathcal{K}') = \{F(\mathcal{K}',0),\cdots, F(\mathcal{K}',k)\}$, and $F(\mathcal{K}',i) = F(\mathcal{K},i) \cup \{[j, G(C_1, D)]\}$. According to 6.(2), for any $j \neq i, F(\mathcal{K}',j) = F(\mathcal{K},j)$. If $C(\mathcal{K})$ is RS-satisfiable, for the reason that $[i, C_1 \sqsubseteq C_2], [i, G(C_2, D)], r(i, C_1 \sqsubseteq C_2, j), r(i, G(C_2, D), j) \in C(\mathcal{K})$, so there is a RS-model $\mathfrak{M} =< W, \mathcal{D}, \mathcal{N}, \circledast, \mathcal{C}, \varepsilon, \sigma >$ and possible worlds i, j such that $\mathfrak{M}, i \models C_1 \sqsubseteq C_2, \mathfrak{M}, i \models G(C_2, D)$, $j \in (\circledast(\{i\}, \| C_1 \sqsubseteq C_2 \|^{\mathfrak{M}} \cap \circledast (\{i\}, \| G(C_2, D) \|^{\mathfrak{M}})$. In addition to the GAG property of the G-ALC frame, $(\circledast(\{i\}, \| C_1 \sqsubseteq C_2 \|^{\mathfrak{M}} \cap \circledast (\{i\}, \| G(C_2, D) \|^{\mathfrak{M}}) \subseteq \| \alpha \|^{\mathfrak{M}}$. Hence $j \in \| G(C_1, D) \|^{\mathfrak{M}}$ and $\mathfrak{M} \models C(\mathcal{K}')$. Consequently $C(\mathcal{K}')$ is RS-satisfiable. □

Proposition 9 Let α be a G-ALC formula. If α is satisfiable w.r.t the knowledge base KB, then the tableau $\mathcal{T}_{(KB,\alpha)}$ is open.

Proof If the formula α is satisfiable w.r.t the knowledge base KB, namely the root-node expression set $\mathcal{K}_0$ of the tableau $\mathcal{T}_{(KB,\alpha)}$ is satisfiable. For the reason of the proposition 7,8,9, we know that there must be a branch $\mathcal{B}$ on tableau $\mathcal{T}_{(KB,\alpha)}$ such that its end-node expression set $\mathcal{K}_{\mathcal{B}}$ is satisfiable. Thanks to the proposition 2, $\mathcal{B}$ is open, iff, $\mathcal{K}_{\mathcal{B}}$ is satisfiable. Hence $\mathcal{T}_{(KB,\alpha)}$ must have an open branch, that is to say, $\mathcal{T}_{(KB,\alpha)}$ is open. □

6 The completeness of the G-ALC tableau algorithm

Lemma 1 Let α be a G-ALC formula, $\mathcal{T}_{(KB,\alpha)}$ a finished open tableau, $\mathcal{B}$ a open branch of $\mathcal{T}_{(KB,\alpha)}$, and $\mathcal{K}_B$ the end-node expression set of $\mathcal{B}$. For any G-ALC formula β, if there is $i \in I(\mathcal{K}_B)$ such that $\beta \in F(\mathcal{K}_{\mathcal{B}}, i)$, then $\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}, i \models \beta$.

Proof To prove by the induction on the formula structure.

1. In case $\beta = C(t)$. In this case, $[i, C(t)] \in \mathcal{K}_{\mathcal{B}}$, by the definition 18, $t^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}(i) = t_i$ and $C^{\varepsilon_{\mathcal{K}_{\mathcal{B}}}}(i) = \{t_i \mid [i, C(t)] \in \mathcal{K}_{\mathcal{B}}\}$. Hence $t^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}(i) \in C^{\varepsilon_{\mathcal{K}_{\mathcal{B}}}}(i)$. In addition to the definition 19, $\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}, i \models C(t)$.
2. In case $\beta = R(t, t')$. In this case, $[i, R(t, t')] \in \mathcal{K}_{\mathcal{B}}$, by the definition 18, $R^{\varepsilon_{\mathcal{K}_{\mathcal{B}}}}(i) = \{< t^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}(i), t'^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}(i) > \mid [i, R(t, t')] \in \mathcal{K}_{\mathcal{B}}\}$. According to the definition 19, $i \in \parallel R(t, t') \parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$, i.e., $\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}, i \models R(t, t')$.
3. In case $\beta = C \sqsubseteq D$. In this case, $[i, C \sqsubseteq D] \in \mathcal{K}_{\mathcal{B}}$, and $\mathcal{T}_{(KB,\alpha)}$ is a finished open tableau, so there is a term t such that $[i, C(t)] \in \mathcal{K}_{\mathcal{B}}$. And the rule $[\sqsubseteq]$ is used to $[i, C \sqsubseteq D]$. Hence $[i, D(t)] \in \mathcal{K}_{\mathcal{B}}$. According to the definition 18, $C^{\varepsilon_{\mathcal{K}_{\mathcal{B}}}}(i) \subseteq D^{\varepsilon_{\mathcal{K}_{\mathcal{B}}}}(i)$. Again by the definition 19, we get $i \in \parallel C \sqsubseteq D \parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$, i.e., $\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}, i \models C \sqsubseteq D$.
4. In case $\beta = C \equiv D$. In this case, $[i, C \equiv D] \in \mathcal{K}_{\mathcal{B}}$, and $\mathcal{T}_{(KB,\alpha)}$ is a finished open tableau, so the rule $[\sqsubseteq]$ is used to $[i, C \sqsubseteq D]$. Hence $\{[i, C \sqsubseteq D], [i, D \sqsubseteq C]\} \in \mathcal{K}_{\mathcal{B}}$. According to the definition 18, we get that $C^{\varepsilon_{\mathcal{K}_{\mathcal{B}}}}(i) \equiv D^{\varepsilon_{\mathcal{K}_{\mathcal{B}}}}(i)$. In addition to the definition 19, there goes that $i \in \parallel C \equiv D \parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$, i.e., $\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}, i \models C \equiv D$.
5. In case $\beta = G(C, D)$. For $[i, G(C, D)] \in \mathcal{K}_{\mathcal{B}}$. According to the definition 18, we get that $D^{\varepsilon_{\mathcal{K}_{\mathcal{B}}}} \in (C^{\varepsilon_{\mathcal{K}_{\mathcal{B}}}})^{\mathcal{C}_{\mathcal{K}_{\mathcal{B}}}}(i)$. In additon to the definition 19, we get that $i \in \parallel G(C, D) \parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$. Hence, $\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}, i \models G(C, D)$.
6. In case $\beta = \alpha > \gamma$. In this case, $[i, \alpha > \gamma] \in \mathcal{K}_{\mathcal{B}}$, and $\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}$is a RS-model, then $r(i, \alpha, j) \in \mathcal{K}_{\mathcal{B}}$, iff, $j \in \circledast(\{i\}, \parallel \alpha \parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}})$. Because $\mathcal{T}_{(KB,\alpha)}$ is a finished open tableau, if $r(i, \alpha, j) \in \mathcal{K}_{\mathcal{B}}$, then $[i, \alpha > \gamma]$ must use the rule $[>]$. Hence $[j, \gamma] \in \mathcal{K}_{\mathcal{B}}$, namely $j \in \parallel \gamma \parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$. So if $j \in \circledast(\{i\}, \parallel \alpha \parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}})$, then $j \in \parallel \gamma \parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$. Therefore $\circledast(\{i\}, \parallel \alpha \parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}) \subseteq \parallel \gamma \parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$. According to the definition 19, $i \in \parallel \alpha > \gamma \parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$, i.e., $\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}, i \models \alpha > \gamma$.
7. In case $\beta = \neg(\alpha > \gamma)$. In this case, $[i, \neg(\alpha > \gamma)] \in \mathcal{K}_{\mathcal{B}}$. Because $\mathcal{T}_{(KB,\alpha)}$ is finished and open, $[i, \neg(\alpha > \gamma)]$ must use the rule $[\neg >]$. Hence

$\{r(i,\alpha,j),[j,\neg\gamma]\}\subseteq\mathcal{K}_{\mathcal{B}}$. By the induction hypothesis, we know that $j\in\parallel\neg\gamma\parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$. By the definition 19, $j\notin\parallel\gamma\parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$. Because $\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}$ is a RS-model, $r(i,\alpha,j)\in\mathcal{K}_{\mathcal{B}}$, iff, $j\in\circledast(\{i\},\parallel\alpha\parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}})$. Hence, $j\in\circledast(\{i\},\parallel\alpha\parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}})$, $j\notin\parallel\gamma\parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$, and $\circledast(\{i\},\parallel\alpha\parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}})\nsubseteq\parallel\gamma\parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$.By the definition 19, $i\in\parallel\neg(\alpha>\gamma)\parallel^{\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}}}$, consequently $\mathfrak{M}_{\mathcal{K}_{\mathcal{B}}},i\models\neg(\alpha>\gamma)$.

□

Proposition 10 Let α be a G-ALC formula. If $\mathcal{T}_{(KB,\alpha)}$is a finished opem tableau, then α is satisfiable w.r.t. the knowledge base KB.

Proof If $\mathcal{T}_{(KB,\alpha)}$ is open, then there is an open branch $\mathcal{B}$. If $\mathcal{T}_{(KB,\alpha)}$ is finished, then according to the lemma 1, for the reason that $\alpha\in F(\mathcal{K}_{\mathcal{B}},0)$, so $\mathfrak{M}_{\mathfrak{B}},0\models\alpha$. For $\mathfrak{M}_{\mathfrak{B}}$ is a G-ALC model, so α is satisfiable w.r.t. the knowledge base KB. □

7 Conclusion

This paper extends generic sentences to the basic description logic ALC, and get a new logic G-ALC. As a matter of fact, we compound ALC and the term logic systems GAG and Gaa, for the reasoning of generic sentences, in syntax and semantics. Then a tableau agorithm is given for the reasoning problems of G-ALC, and the important properties such as the soundness and completeness of it are strictly proved. The future work following this paper is to research the algorithm's computational complexity and the application of G-ALC in some certaim knowledge fields.

References

[1] Ariel Cohen, *Think Generic! The Meaning and Use of Gerneric Sentences*,CSLI Publications, 1996

[2] Asher, N. and Morreau, M., 1991, "Commonsense entailment: A modal theory of non-monotonic reasoning", *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, J. Mylopoulos and R. Reiter (eds), Morgan Kauffman, Los Altos, California, pp. 387-392.

[3] Boutilier, C., 1994. "Conditional logics of normality: A modal approach", *Artificial Intelligence*, **VOl** (68): 87-154.

[4] Franz Baader, Deborah L. McGuinness,Daniele Nardi, Peter F.Patel-Schneider, *The description logic handbook: Theory, implementation and application*, Cambridge University Press, 2007

[5] Mao, Yi., 2003, *A Formalism for Non-monotonic Reasoning Encoded Generics*, Ph.D. dissertation, The University of Texas at Austin.

[6] McCarthy, J., 1980, "Circumscription-a form of non-monotonic reasoning", *Artificial Intelligence*, **VOl** (13): 27-39

[7] McCarthy, J., 1986, "Applications of circumscription to formalizing commonsense knowledge", *Artificial Intelligence* , **VOl** (28), 89-116.

[8] Priest, G., 2001, *An Introduction to Non-Classical Logic*, Cambridge University Press, 2001.

[9] Reiter, R., 1980, "A logic for default reasoning", *Artificial Intelligence*, **VOl** (13 (1, 2)): 81-132.

[10] Reiter, R., 1987, “Non-monotonic Reasoning”, *Annual Review of Computer Science*, **VOl** (2): 147-186.

[11] Zhou, Beihai, and Mao, Yi, 2010, “Four semantic layers of common nouns”, *Synthese*，**VOl** (175(1)): 47-68.

[12] 周北海, 毛翊, 2003, “一个关于常识推理的基础逻辑”, 《哲学研究》, 2003 年增刊: 1-10。

[13] 周北海, 概称句的本质与概念, 北京大学学报（哲学社会科学版）, 2004.

[14] 张立英，周北海，基于主谓项涵义联系的概称句推理的几个逻辑，《哲学动态》2004年增刊

[15] 张立英，一种类型的概称句推理，《哲学动态》，增刊，2005.

[16] 周北海, 2008, “涵义语义与关于概称句推理的词项逻辑”, 《逻辑学研究》, 2008 年第 1 期: 37-48。

描述逻辑的概称句扩张及其Tableau判定算法

马丽
北京大学哲学系
mq2828@126.com

摘　要

描述逻辑主要刻画概念、角色和个体，特别地，概念是描述逻辑主要的刻画对象。描述逻辑对概念的外延有确定的解释，是个体域的一个子集。描述逻辑对概念的内涵也有处理，但是只是通过概念与其上下位概念之间的外延包含关系来体现。事实上，一个概念的内涵不仅是它与上下位概念之间的关系，而且还有它自己独特的性质。比如“鸟”这个概念不仅是“动物”的子概念，而且它还有“会飞、有羽毛、生蛋”等等一些内涵项。既然描述逻辑的初衷是为了更好地刻画知识，那么就要更好地刻画知识的内涵。本文把一种表达概念内涵的句子概称句扩张到基本描述逻辑 ALC 中去，扩张后的逻辑记为 G-ALC。把概称句推理的系统 GAG 和 Gaa 与 ALC 相结合，为 G-ALC 提供了内涵语义，并且基于此语义为其推理问题提供了 tabeau（树图）算法。算法的的可靠性和完全性在本文中得到了细致的、严谨的技术化证明。本文的工作对描述逻辑处理自然语言、构建语义本体都有理论意义，希望这一研究能够在语义网即相关研究中能够得到应用。

关键词: 描述逻辑，概称句，扩张，树图算法，可靠性，完全性