时态描述逻辑 *ALC*-LTL 的 Tableau 判定算法

常 亮 王 娟 古天龙 董荣胜
（桂林电子科技大学计算机科学与工程学院 桂林 541004）

摘 要 时态描述逻辑 *ALC*-LTL 将描述逻辑 *ALC* 的描述能力与线性时态逻辑 LTL 的刻画能力结合起来，在具有较强描述能力的同时还使得可满足性问题保持在 EXPTIME-完全这个级别。针对 *ALC*-LTL 缺少有效的判定算法的现状，将 LTL 的 Tableau 判定算法与描述逻辑 *ALC* 的推理机制有机地结合起来，给出了 *ALC*-LTL 的 Tableau 判定算法并证明了算法的可终止性、可靠性和完备性。该算法具有很好的可扩展性。当 *ALC*-LTL 中的描述逻辑从 *ALC* 改变为任何一个具有可判定性特征的描述逻辑 X 时，只需要对算法进行简单修改，就可以得到相应的时态描述逻辑 X-LTL 的 Tableau 判定算法。

关键词 时态描述逻辑，线性时态逻辑，可满足性问题，Tableau 算法，复杂度

中图法分类号 TP301 **文献标识码** A

Tableau Decision Algorithm for the Temporal Description Logic *ALC*-LTL

CHANG Liang WANG Juan GU Tian-long DONG Rong-sheng
(School of Computer Science and Technology, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract As a combination of the description logic *ALC* and the linear temporal logic LTL, the temporal description logic *ALC*-LTL not only offers considerable expressive power going beyond both *ALC* and LTL, but also makes the satisfiability problem of it preserved to be EXPTIME-complete; however, an efficient decision algorithm for *ALC*-LTL is still absent. Based on a combination of the Tableau algorithm of LTL and the reasoning mechanism provided by *ALC*, this paper presented a Tableau decision algorithm for the logic *ALC*-LTL and proved that this algorithm is terminating, sound and complete. This algorithm enjoys excellent expandability; when the *ALC* component in the logic *ALC*-LTL is substituted by any other description logic X which is still decidable, this algorithm can be easily modified to act as a Tableau decision algorithm for the corresponding temporal description logic X-LTL.

Keywords Temporal description logic, Linear temporal logic, Satisfiability problem, Tableau algorithm, Complexity

描述逻辑在语义 Web 中扮演着关键角色，是 W3C 推荐的 Web 本体语言 OWL 的逻辑基础。描述逻辑的主要特点在于既提供了命题逻辑所无法比拟的刻画能力，又保证了相关推理问题的可判定性，并且有有效的判定算法和推理机制作为支撑[1]。

针对 Web 环境的各种特征和应用需求，研究者提出了描述逻辑的各种扩展形式，以期为语义 Web 提供更为充分的逻辑支持。在描述逻辑众多的扩展形式中，时态描述逻辑获得了较多的关注[2,3]。时态描述逻辑在描述逻辑中引入一个时态维，形成具有二维特征的逻辑系统。在研究者已经提出的各种各样的时态描述逻辑中，绝大部分都将时态算子既用于概念的构造又用于公式的构造，从而既可以对具有时态内涵的概念进行刻画和推理，又可以对知识库的时态特征进行刻画和推理。研究者证明了这些时态描述逻辑中的可满足性问题仍然具有可判定性特征，但推理复杂度都很高，达到了 EXPSPACE 或者 2EXPTIME 的程度。

为了降低推理问题的复杂度，Baader 等[4]在将线性时态逻辑 LTL 与描述逻辑 *ALC* 结合时不允许时态算子出现在概念中，而是仅仅将时态算子应用于公式的构造。在此基础上，Baader 等提出了时态描述逻辑 *ALC*-LTL，然后分别在一般情况下、考虑 rigid concept 的情况下以及同时考虑 rigid concept 和 rigid role 的情况下研究了 *ALC*-LTL 中公式的可满足性问题，证明了该问题在 3 种情况下都是可判定的，并分析了相应的时间复杂度。其中，在一般情况下（即不考虑 rigid concept 和 rigid role 的情况下），*ALC*-LTL 中公式的可满足性问题为 EXPTIME-完全。但到目前为止，研究者尚未针对 *ALC*-LTL 给出有效的判定算法。

有效的判定算法和推理机制是描述逻辑及其扩展形式得到成功应用的关键所在。针对时态描述逻辑 *ALC*-LTL 缺少有效的判定算法的现状，本文首先以线性时态逻辑 LTL 的 Tableau 判定算法为基础，在其中引入描述逻辑 *ALC* 的推理机制，设计出 *ALC*-LTL 在一般情况下的 Tableau 判定算法。接下来，本文依次对算法的可终止性、可靠性和完备性进行证明，并对算法在最坏情况下的复杂度进行考察。最后对相关

到稿日期：2010-09-18 返修日期：2010-12-27 本文受国家自然科学基金（60903079，60963010），广西自然科学基金（0832006Z）资助。
常 亮（1980－），男，博士，副教授，CCF 会员，主要研究方向为知识表示与推理、形式化方法、语义 Web 等，E-mail：changl@guet.edu.cn；**王 娟**（1985－），女，硕士生，主要研究方向为知识推理和语义 Web 服务；**古天龙**（1964－），男，教授，博士生导师，主要研究方向为形式化方法、符号计算；**董荣胜**（1965－），男，教授，主要研究方向为形式化方法、计算思维与计算机方法论。

工作进行比较。

1 时态描述逻辑 *ALC*-LTL

由于 *ALC*-LTL 中并没有将时态算子应用于概念的构造，因此从时态逻辑的角度来看，也可以将 *ALC*-LTL 看作是在命题线性时态逻辑 LTL 中引入描述逻辑 *ALC* 的刻画成分之后得到的逻辑系统。

从语法上看，*ALC*-LTL 的主要特点在于将 LTL 中的原子命题替换为 *ALC* 中的一般概念，包含公理、概念断言和角色断言。具体来说，*ALC*-LTL 的基本符号包括由概念名组成的集合 N_C、由角色名组成的集合 N_R 以及由个体名组成的集合 N_I；从这些符号出发，通过描述逻辑 *ALC* 中的概念构造符和线性时态逻辑 LTL 中的公式构造符，可以递归地生成 *ALC*-LTL 的概念和公式。

定义 1 *ALC*-LTL 中的概念由如下产生式生成，

$C,D::=C_i \mid \neg C \mid C\sqcup D \mid \forall R.C$

式中，$C_i\in N_C$，$R\in N_R$。此外，可以引入形如 $C\sqcap D$ 和 $\exists R.C$ 的概念，分别作为 $\neg(\neg C\sqcup\neg D)$ 和 $\neg(\forall R.\neg C)$ 的缩写。

令 C、D 为任意两个概念，$R\in N_R$，$p,q\in N_I$，则将 $C\sqsubseteq D$ 称为一般概念包含公理，将 $C(p)$ 称为概念断言，将 $R(p,q)$ 称为角色断言。

定义 2 *ALC*-LTL 中的公式由如下产生式生成，

$\varphi,\psi::=C\sqsubseteq D\mid C(p)\mid R(p,q)\mid\neg\varphi\mid\varphi\wedge\psi\mid X\varphi\mid\varphi u\psi$

式中，$p,q\in N_I$，$R\in N_R$，C、D 为概念。此外，可以依次引入形如 $false$、$true$、$\varphi\vee\psi$、$\varphi\rightarrow\psi$、$F\varphi$ 以及 $G\varphi$ 的公式，分别作为 $\varphi\wedge\neg\varphi$、$\neg false$、$\neg(\neg\varphi\wedge\neg\psi)$、$\neg\varphi\vee\psi$、$true\ u\varphi$ 和 $\neg F\neg\varphi$ 的缩写。

我们将每个一般概念包含公理、概念断言和角色断言都称为一个 *ALC*-断言；将每个 *ALC*-断言及其否定形式都称为一个 *ALC*-文字。此外，将每个形如 $\varphi u\psi$ 的公式称为一个可能性断言。

从语义上看，*ALC*-LTL 的解释结构与 LTL 的解释结构在整体上相似，通过时间的进展将各个状态线性地组织起来；但与 LTL 解释结构不同的是，*ALC*-LTL 解释结构中的每个状态不是简单地映射为由原子命题组成的集合，而是映射为描述逻辑 *ALC* 的一个解释。

定义 3 *ALC*-LTL 解释结构是一个二元组 $M=(\mathbb{N},I)$，其中：

(1)$\mathbb{N}$ 为自然数集合；

(2)函数 I 对每个自然数 $n\in\mathbb{N}$ 赋予描述逻辑 *ALC* 的一个解释 $I(n)=(\Delta,\cdot^{I(n)})$，其中的解释函数 $\cdot^{I(n)}$ 满足以下条件，

(i)将每个概念名 $C_i\in N_C$ 解释为 Δ 的某个子集 $C_i^{I(n)}\subseteq\Delta$；

(ii)将每个角色名 $R_i\in N_R$ 解释为 Δ 上的某个二元关系 $R_i^{I(n)}\subseteq\Delta\times\Delta$；

(iii)将每个个体名 $p_i\in N_I$ 解释为 Δ 中的某个元素 $p_i^{I(n)}\in\Delta$，并且对于任一自然数 $m\in\mathbb{N}$ 都有 $p_i^{I(n)}=p_i^{I(m)}$。

定义 4 给定任一 *ALC*-LTL 解释结构 $M=(\mathbb{N},I)$，对 *ALC*-LTL 中概念和公式的语义递归定义如下。首先，相对于任一自然数 $n\in\mathbb{N}$，将每个概念 C 解释为 Δ 的某个子集 $C^{I(n)}$；其递归定义为：

(1)$\neg(C)^{I(n)}:=\Delta\backslash C^{I(n)}$，其中的“\”为集合差运算；

(2)$(C\sqcup D)^{I(n)}:=C^{I(n)}\cup D^{I(n)}$，其中的“$\cup$”为集合并运算；

(3)$(\forall R.C)^{I(n)}:=\{x\mid$ 对于任一 $y\in\Delta$：如果 $(x,y)\in R^{I(n)}$，则必然有 $y\in C^{I(n)}\}$。

其次，相对于任一自然数 $n\in\mathbb{N}$，用 $(M,n)\models\varphi$ 表示公式 φ 在结构 M 中的时间点 n 下成立，递归定义如下，

(4)$(M,n)\models C\sqsubseteq D$ iff $C^{I(n)}\subseteq D^{I(n)}$；

(5)$(M,n)\models C(p)$ iff $p^{I(n)}\in C^{I(n)}$；

(6)$(M,n)\models R(p,q)$ iff $(p^{I(n)},q^{I(n)})\in R^{I(n)}$；

(7)$(M,n)\models\neg\varphi$ iff $(M,n)\not\models\varphi$(即公式 φ 在结构 M 中的时间点 n 下不成立)；

(8)$(M,n)\models\varphi\wedge\psi$ iff $(M,n)\models\varphi$ 并且 $(M,n)\models\psi$；

(9)$(M,n)\models X\varphi$ iff $(M,n+1)\models\varphi$；

(10)$(M,n)\models\varphi u\psi$ iff 存在某个整数 $k\geqslant 0$ 使得 $(M,n+k)\models\psi$ 并且对于任一 $0\leqslant i<k$ 都有 $(M,n+i)\models\varphi$。

描述逻辑中的知识库通常由 TBox 和 ABox 组成，其中的 TBox 是由一般概念包含公理组成的有限集合，ABox 是由概念断言、角色断言以及概念断言和角色断言的否定形式等组成的有限集合。在更一般的情况下，描述逻辑中还允许通过布尔联结符将一般概念包含公理、概念断言和角色断言组织起来，形成布尔知识库[4]。此时，一个最基本的推理问题是判断布尔知识库的一致性，即对于任一布尔知识库 $\mathscr{B}$，判断是否存在描述逻辑的一个解释 $I=(\Delta,\cdot^I)$ 使得 $I\models\mathscr{B}$。针对描述逻辑 *ALC*，文献中已经对布尔知识库的一致性问题进行了研究，并证明了其推理复杂度为 EXPTIME-完全[4]。

显然，描述逻辑 *ALC* 中的每个布尔知识库仅仅是 *ALC*-LTL 中一个不含有时态算子的公式。相应地，描述逻辑 *ALC* 中布尔知识库的一致性问题在 *ALC*-LTL 中体现为公式的可满足性问题。

定义 5 对于 *ALC*-LTL 中的任一公式 φ，称 φ 是可满足的当且仅当存在某个 *ALC*-LTL 解释结构 $M=(\mathbb{N},I)$ 使得 $(M,0)\models\varphi$。

公式的可满足性问题是时态描述逻辑 *ALC*-LTL 中最基本的推理问题。文献[4]已经证明了该问题为 EXPTIME-完全。本文将在下一节给出一个具体的 Tableau 判定算法。

2 *ALC*-LTL 的 Tableau 判定算法

在给出算法之前，先引入一系列符号和术语。

首先，为了避免在判定过程中引入公式的多重否定，对于任一 *ALC*-LTL 公式 φ，我们用 $neg(\varphi)$ 定义如下的公式：如果 φ 不是以“$\neg$”开头的公式，则令 $neg(\varphi):=\neg\varphi$；如果 φ 形如 $\neg\varphi_1$，则令 $neg(\varphi):=\varphi_1$。

其次，对于任一 *ALC*-LTL 公式 φ，用 $sub(\varphi)$ 表示由 φ 的所有子公式组成的集合，递归定义如下，

(1)如果 φ 是一般概念包含公理、概念断言或者角色断言，则 $sub(\varphi):=\{\varphi\}$；

(2)如果 φ 形如 $\neg\varphi_1$ 或者 $X\varphi_1$，则 $sub(\varphi):=\{\varphi\}\cup sub(\varphi_1)$；

(3)如果 φ 形如 $\varphi_1\wedge\varphi_2$ 或者 $\varphi_1 u\varphi_2$，则 $sub(\varphi):=\{\varphi\}\cup sub(\varphi_1)\cup sub(\varphi_2)$。

在此基础上，用 $cl(\varphi)$ 表示满足以下要求的最小集合，

(1)$\varphi\in cl(\varphi)$；

(2)如果 $\psi\in cl(\varphi)$，则 $sub(\psi)\subseteq cl(\varphi)$；

(3)如果 $\psi\in cl(\varphi)$，则 $neg(\psi)\in cl(\varphi)$；

(4)如果 $\neg X\varphi_1\in cl(\varphi)$，则 $X\neg\varphi_1\in cl(\varphi)$；

(5)如果 $\varphi_1 u\varphi_2\in cl(\varphi)$,则 $X(\varphi_1 u\varphi_2)\in cl(\varphi)$。

显然,集合 $cl(\varphi)$ 中元素的个数与公式 φ 的长度成线性关系,并且 $cl(\varphi)$ 中每个公式的长度也与公式 φ 的长度成线性关系。

接下来,对于任一 *ALC*-LTL 公式 φ,将满足以下条件的每个集合 $h\subseteq cl(\varphi)$ 称为公式 φ 的一个 Hintikka 集。

(1)将 h 中所有 *ALC*-文字合取后得到的布尔 *ALC* 知识库是一致的;

(2)如果 $\varphi_1\in h$,则必然有 $\mathrm{neg}(\varphi_1)\notin h$;

(3)如果 $\neg\neg\varphi_1\in h$,则必然有 $\varphi_1\in h$;

(4)如果 $\neg X\varphi_1\in h$,则 $X\neg\varphi_1\in h$;

(5)如果 $\varphi_1\wedge\varphi_2\in h$,则必然有 $\varphi_1\in h$ 并且 $\varphi_2\in h$;

(6)如果 $\neg(\varphi_1\wedge\varphi_2)\in h$,则必然有 $\neg\varphi_1\in h$ 或者 $\neg\varphi_2\in h$;

(7)如果 $\varphi_1 u\varphi_2\in h$,则必然有 $\varphi_2\in h$ 或者 $\{\varphi_1, X(\varphi_1 u\varphi_2)\}\subseteq h$;

(8)如果 $\neg(\varphi_1 u\varphi_2)\in h$,则必然有 $\neg\varphi_2\in h$,并且有 $\neg\varphi_1\in h$ 或者 $\neg X(\varphi_1 u\varphi_2)\in h$。

对于任一集合 $\Gamma\subseteq cl(\varphi)$,如果存在某个集合 h 使得 $\Gamma\subseteq h$,h 是公式 φ 的一个 Hintikka 集,并且不存在公式 φ 的其它 Hintikka 集 h' 使得 $\Gamma\subseteq h'\subset h$,则称 h 是集合 Γ 的一个极小完全扩展。

算法 1 对于任一集合 $\Gamma\subseteq cl(\varphi)$,可以通过以下步骤求出其所有极小完全扩展:

(1)令 $FE(\Gamma):=\emptyset$;

(2)如果将 Γ 中所有 *ALC*-文字合取后得到的布尔 *ALC* 知识库是一致的,并且对于任一公式 $\varphi_1\in\Gamma$ 都有 $\mathrm{neg}(\varphi_1)\notin\Gamma$,则将 Γ 加入集合 $FE(\Gamma)$;

(3)对于任一集合 $h\in FE(\Gamma)$,如果其不是公式 φ 的 Hintikka 集,则将其从 $FE(\Gamma)$ 中移出,并应用以下规则之一对其进行扩展,

①$\neg\neg$-规则:如果 $\neg\neg\varphi_1\in h$ 并且 $\varphi_1\notin h$,则令 $h_1:=h\cup\{\varphi_1\}$;

②$\neg X$-规则:如果 $\neg X\varphi_1\in h$ 并且 $X\neg\varphi_1\notin h$,则令 $h_1:=h\cup\{X\neg\varphi_1\}$;

③$\wedge$-规则:如果 $\varphi_1\wedge\varphi_2\in h$ 并且 $\{\varphi_1,\varphi_2\}\not\subset h$,则令 $h_1:=h(\{\varphi_1,\varphi_2\}$;

④$\neg\wedge$-规则:如果 $\neg(\varphi_1\wedge\varphi_2)\in h$ 并且 $\neg\varphi_1\notin h$,则令 $h_1:=h\cup\{\neg\varphi_1\}$;如果 $\neg(\varphi_1\wedge\varphi_2)\in h$ 并且 $\neg\varphi_2\notin h$,则令 $h_2:=h\cup\{\neg\varphi_2\}$;

⑤u-规则:如果 $\varphi_1 u\varphi_2\in h$ 并且 $\varphi_2\notin h$,则令 $h_1:=h\cup\{\varphi_2\}$;如果 $\varphi_1 u\varphi_2\in h$ 并且 $\{\varphi_1, X(\varphi_1 u\varphi_2)\}\not\subset h$,则令 $h_2:=h\cup\{\varphi_1, X(\varphi_1 u\varphi_2)\}$;

⑥$\neg u$-规则:如果 $\neg(\varphi_1 u\varphi_2)\in h$ 并且 $\{\neg\varphi_2,\neg\varphi_1\}\not\subset h$,则令 $h_1:=h\cup\{\neg\varphi_2,\neg\varphi_1\}$;如果 $\neg(\varphi_1 u\varphi_2)\in h$ 并且 $\{\neg\varphi_2,\neg X(\varphi_1 u\varphi_2)\}\not\subset h$,则令 $h_2:=h\cup\{\neg\varphi_2,\neg X(\varphi_1 u\varphi_2)\}$。

(4)对于上述扩展后得到的任一集合 $h'\in\{h_1,h_2\}$,在 $h'\notin FE(\Gamma)$ 的情况下,如果对于任一公式 $\varphi_1\in h'$ 都有 $\mathrm{neg}(\varphi_1)\notin h'$,并且将 h' 中所有 *ALC*-文字合取后得到的布尔 *ALC* 知识库是一致的,则将 h' 加入集合 $FE(\Gamma)$。

(5)反复进行上面的步骤(3)和(4),直到集合 $FE(\Gamma)$ 的元素个数不发生变化为止;返回 $FE(\Gamma)$。

显然,上述扩展过程中生成的每个集合都是 $cl(\varphi)$ 的子集。通过逐一考察应用的扩展规则,容易证明以下结论。

引理 1 对于任一集合 $\Gamma\subseteq cl(\varphi)$,算法 1 是可终止的,并且最终返回的 $FE(\Gamma)$ 是由 Γ 的所有极小完全扩展组成的集合。

引理 2 对于任一集合 $\Gamma\subseteq cl(\varphi)$,如果存在某个解释结构 $M=(\mathbb{N}, I)$ 使得对于任一公式 $\psi\in\Gamma$ 都有 $(M,0)\models\psi$,则必然存在 Γ 的一个极小完全扩展 $h\in FE(\Gamma)$,使得对于任一公式 $\gamma\in h$ 都有 $(M,0)\models\gamma$。

Tableau 判定算法的基本思路是判断其能否为给定的公式构造出一个 *Tableau*。对于 ALC-*LTL*,我们首先对其公式的 *Tableau* 定义如下。

定义 6 对于 *ALC*-LTL 中的任一公式 φ,如果存在某个有向图 $T=\langle V,E\rangle$ 满足下面(T1)到(T6)的条件,则将 $T=\langle V,E\rangle$ 称为公式 φ 的一个 Tableau。

(T1)每个顶点 $v\in V$ 都是公式 φ 的一个 Hintikka 集;

(T2)存在某个顶点 $v_0\in V$ 使得 $\varphi\in v_0$;

(T3)E 是 V 上的一个全关系,即对于任一 $v\in V$ 都存在至少一个顶点 $v'\in V$ 使得 $\langle v,v'\rangle\in E$;

(T4)对于任意一条边 $\langle v,u\rangle\in E$,令集合 $\Gamma=\{\psi\mid X\psi\in v\}$,则必然有 $\Gamma\subseteq u$;

(T5)对于任一顶点 $v\in V$,如果 $\varphi_1 u\varphi_2\in v$,则必然存在某条路径 $v_0v_1\cdots v_k$(其中 $k\geqslant 0$)使得 $v=v_0$,对于任一 $0\leqslant i<k$ 都有 $\varphi_1 u\varphi_2\in v_i$ 和 $\varphi_1\in v_i$,以及有 $\varphi_1 u\varphi_2\in v_k$ 和 $\varphi_2\in v_k$;

(T6)对于任一顶点 $v\in V$,如果 $\varphi_1 u\varphi_2\in v$,则对于任意一条路径 $v_0v_1\cdots v_k$(其中 $k\geqslant 0$)来说,如果 $v=v_0$ 并且对于任一 $0\leqslant i\leqslant k$ 都有 $\varphi_2\notin v_i$,则对于任一 $0\leqslant j\leqslant k$ 都有 $\{\varphi_1 u\varphi_2,\varphi_1, X(\varphi_1 u\varphi_2)\}\subseteq v_j$。

为了表述简便,在下面的算法中,我们将有向图 $T=\langle V,E\rangle$ 的顶点集 V 由 $2^{cl(\varphi)}$ 的子集扩充为 $2^{cl(\varphi)}\times\{\uparrow,\surd\}$ 的子集。在此基础上,我们引入以下符号和术语。

(1)对于每个顶点 $v\in V$,我们用 v^1 和 v^2 分别表示将 v 看作二元组时的第一个元和第二个元;

(2)对于每个顶点 $v\in V$,如果 v^2 为 $\uparrow$,则将该顶点称为伪状态;如果 v^2 为 $\surd$,则将该顶点称为饱和状态;

(3)对于每条边 $\langle v,v'\rangle\in E$,将顶点 v' 称为顶点 v 的后继状态,将顶点 v 称为顶点 v' 的前驱状态,将 $\langle v,v'\rangle$ 分别称为顶点 v 的输出边和顶点 v' 的输入边;

(4)对于任一顶点 $v\in V$ 以及任一可能性断言 $\varphi_1 u\varphi_2\in v^1$,如果存在某条路径 $v_0v_1\cdots v_k$(其中 $k\geqslant 0$)使得 $v=v_0$,对于任一 $0\leqslant i<k$ 都有 $\varphi_1 u\varphi_2\in v_i^1$ 和 $\varphi_1\in v_i^1$,以及有 $\varphi_1 u\varphi_2\in v_k^1$ 和 $\varphi_2\in v_k^1$,则称可能性断言 $\varphi_1 u\varphi_2$ 相对于顶点 v 被实现了,否则称 $\varphi_1 u\varphi_2$ 相对于顶点 v 没有被实现。

最后,对 *ALC*-LTL 的 Tableau 判定算法描述如下。

算法 2 对于任一 *ALC*-LTL 公式 φ,按以下步骤判断其是否可满足:

(1)构建一个初始的有向图 $T=\langle V,E\rangle$,其中 $V\subseteq 2^{cl(\varphi)}\times\{\uparrow,\surd\}$,$E\subseteq V\times V$。构建过程如下,

(1.1)令 $V:=\{(\{\varphi\},\uparrow)\}$,$E:=\emptyset$;

(1.2)对 V 中每个不存在后继状态的伪状态 u 进行以下操作:首先求出 u^1 的所有极小完全扩展 $FE(u^1)=\{h_1,\cdots,h_m\}$;接下来,对于 $FE(u^1)$ 中的每个元素 $h_i(1\leqslant i\leqslant m)$,将顶点 $(h_i,\surd)$ 加入集合 V,将边 $\langle(\Gamma,\uparrow),(h_i,\surd)\rangle$ 加入集合 E;

(1.3)对 V 中每个不存在后继状态的饱和状态 v 进行以下操作:构造集合 $\Gamma=\{\varphi_1\mid X\varphi_1\in v^1\}$,将顶点 $(\Gamma,\uparrow)$ 加入集合 V,将边 $\langle(h,\surd),(\Gamma,\uparrow)\rangle$ 加入集合 E;

(1.4)如果 V 中还含有不存在后继状态的伪状态,则跳

转到步骤(1.2)。

(2)消去图 $T=\langle V,E\rangle$ 中的所有伪状态。具体来说,对于 V 中的每个伪状态 u,令 $Pre_u=\{v_1,\cdots,v_m\}$ 是由 u 的所有前驱状态组成的集合,令 $Post_u=\{w_1,\cdots,w_n\}$ 是由 u 的所有后继状态组成的集合,依次进行以下操作:

(2.1)从 E 中删除顶点 u 的所有输入边和输出边,从 V 中删除顶点 u;

(2.2)对于所有 $1\leqslant i\leqslant m$ 和所有 $1\leqslant j\leqslant n$,在 E 中增加边 $\langle v_i,w_j\rangle$。

(3)消去图 $T=\langle V,E\rangle$ 中不满足要求的饱和状态。具体来说,由以下步骤组成:

(3.1)对于每个顶点 $v\in V$,如果 v 不存在后继状态,则从 E 中删除顶点 v 的所有输入边和输出边,并且从 V 中删除顶点 v;

(3.2)对于每个顶点 $v\in V$,如果存在某个可能性断言 $\varphi_1 u\varphi_2\in v^1$ 并且 $\varphi_1 u\varphi_2$ 相对于顶点 v 没有被实现,则从 E 中删除顶点 v 的所有输入边和输出边,并且从 V 中删除顶点 v;

(3.3)如果 V 中还含有不存在后继状态的饱和状态,则跳转到步骤(3.1)。

(4)如果最终得到的图 $T=\langle V,E\rangle$ 中存在某个顶点 $v\in V$ 使得 $\varphi\in v^1$,则返回 TRUE,否则返回 FALSE。

3 判定算法的性质

本节证明上文给出的 Tableau 判定算法是可终止的、可靠的和完备的,并对其复杂度进行分析。

定理 1 算法 2 是可终止的,并且在最坏情况下所需要的时间与公式 φ 的长度成指数关系。

证明:由于算法中构建的有向图 $T=\langle V,E\rangle$ 满足 $V\subseteq 2^{cl(\varphi)}\times\{\uparrow,\surd\}$,而集合 $cl(\varphi)$ 中元素的个数又与公式 φ 的长度成线性关系,因此在最坏情况下有向图 T 中的顶点个数与公式 φ 的长度成指数关系。

与步骤(2)和步骤(3)中删除顶点的过程相比,算法 2 的时间开销主要花费在步骤(1)中对每个顶点的构造过程。对于每个顶点 $v\in V$,如果 v 是通过步骤(1.3)生成的,则可以在常数时间内完成。如果 v 是通过步骤(1.2)生成的,则主要的时间开销在于算法 1 中判断将 v^1 中所有 ALC-文字合取后得到的布尔 ALC 知识库是否为一致;由于将 v^1 中所有 ALC-文字合取后得到的布尔 ALC 知识库在长度上与公式 φ 成线性关系,而文献中已经证明了描述逻辑 ALC 中布尔知识库的一致性问题为 EXPTIME-完全[4],因此生成顶点 v 所需要的时间开销与公式 φ 的长度成指数关系。

综上所述,算法 2 是可终止的。并且,将图 T 的顶点个数与生成每个顶点所需要的时间相乘之后可知,算法 2 在最坏情况下所需要的时间与公式 φ 的长度成指数关系。

为了证明算法 2 的可靠性,下面先引入 Hintikka 结构及其性质。

定义 7 对于 ALC-LTL 中的任一公式 φ,如果存在某个二元组 $H=(\mathbb{N},L)$ 满足下面(H1)到(H3)的条件,则将二元组 $H=(\mathbb{N},L)$ 称为公式 φ 的一个 Hintikka 结构。

(H1)$\mathbb{N}$ 为自然数集合,$L:\mathbb{N}\rightarrow 2^{cl(\varphi)}$ 为标记函数;

(H2)$\varphi\in L(0)$;

(H3)对于任一自然数 $n\in\mathbb{N}$ 都满足以下条件:

① $L(n)$ 是公式 φ 的一个 Hintikka 集;

② 如果 $X\varphi_1\in L(n)$,则有 $\varphi_1\in L(n+1)$;

③ 如果 $\varphi_1 u\varphi_2\in L(n)$,则存在某个整数 $k\geqslant 0$ 使得 $\varphi_2\in L(n+k)$ 并且对于任一 $0\leqslant i<k$ 都有 $\varphi_1\in L(n+i)$。

引理 3 对于 ALC-LTL 中的任一公式 φ,如果 φ 存在 Hintikka 结构,则 φ 是可满足的。

证明:令 $H=(\mathbb{N},L)$ 是公式 φ 的一个 Hintikka 结构。对于自然数 $0\in\mathbb{N}$,由于将 $L(0)$ 中所有 ALC-文字合取后得到的布尔 ALC 知识库是一致的,因此可以构造出描述逻辑 ALC 的某个解释 $I(0)=(\Delta_0,\cdot^{I(0)})$ 使得对于 $L(0)$ 中的每个 ALC-文字 α 都有 $I(0)\models\alpha$;对于其它自然数 $n>0$,可以在 $I(0)$ 的基础上构造出描述逻辑 ALC 的某个解释 $I(n)=(\Delta,\cdot^{I(n)})$,使得 $\Delta=\Delta_0$;对于每个个体名 p 都有 $p^{I(n)}=p^{I(0)}$,以及对于 $L(n)$ 中的每个 φ-文字 α 都有 $I(n)\models\alpha$。显然,$M=(N,I)$ 是一个 ALC-LTL 解释结构。在此基础上,对于任一自然数 $n\in\mathbb{N}$ 以及任一公式 $\psi\in L(n)$,根据对 Hintikka 结构和 Hintikka 集的定义,通过对 ψ 的结构进行归纳,容易证明 $(M,n)\models\psi$。因此,当 $n=0$ 时得到 $(M,0)\models\varphi$,即公式 φ 是可满足的。

接下来可以证明算法 2 的可靠性。

定理 2 对于 ALC-LTL 中的任一公式 φ,如果算法 2 返回 TRUE,则 φ 是可满足的。

证明:令 $T=\langle V,E\rangle$ 是算法 2 返回 TRUE 时得到的有向图。由于此时图 T 中的每个顶点都是饱和状态(即对于每个顶点 $v\in V$ 都有 $v^2=\surd$),我们可以将其中的符号"$\surd$"全部去掉,得到有向图 $T'=\langle V',E'\rangle$;其中,顶点集 $V':=\{v^1\mid v\in V\}$,边集 $E':=\{\langle v^1,u^1\rangle\mid\langle v,u\rangle\in E\}$。根据算法 2 中对图 T 的构造过程容易验证,图 $T'=\langle V',E'\rangle$ 是公式 φ 的一个 Tableau。

下面根据图 $T'=\langle V',E'\rangle$ 构造一个二元组 $H=(\mathbb{N},L)$,其中 $\mathbb{N}$ 为自然数集合,L 是从 $\mathbb{N}$ 到 V' 的函数。构造过程由以下步骤组成。

(1)在 T' 中找出某个满足 $\varphi\in v_0$ 的顶点 v_0;令 $L(0):=v_0$;

(2)令 $i:=0$,令 E 是由满足 $\varphi_1 u\varphi_2\in L(i)$ 和 $\varphi_2\notin L(i)$ 的所有可能性断言组成的集合;

(3)如果 E 为空集,则从图 T' 中找出 $L(i)$ 的任一后继状态 u,令 $L(i+1):=u,i:=i+1$;否则依次进行以下操作:

① 从图 T' 中找出一条路径 $v_0v_1\cdots v_k$(其中 $k\geqslant 0$),使得 $L(i)=v_0$,对于任一 $0\leqslant j<k$ 和任一可能性断言 $\varphi_1 u\varphi_2\in E$ 都有 $\varphi_2\notin L(j)$,并且存在某个可能性断言 $\psi_1 u\psi_2\in E$ 使得 $\psi_2\in L(k)$;

② 对于每个 $1\leqslant j\leqslant k$,令 $L(i+j):=v_j$;

③ 对于每个可能性断言 $\varphi_1 u\varphi_2\in E$,如果 $\varphi_2\in L(k)$,则将 $\varphi_1 u\varphi_2$ 从集合 E 中删除;

④ 对于每个可能性断言 $\varphi_1 u\varphi_2\in L(k)$,如果 $\varphi_2\notin L(k)$ 并且 $\varphi_1 u\varphi_2\notin E$,则将 $\varphi_1 u\varphi_2$ 加入集合 E 中;

⑤ 令 $i:=i+k$;跳转到步骤(3)。

根据对 Tableau 和 Hintikka 结构的定义容易验证,上面构造的二元组 $H=(\mathbb{N},L)$ 是公式 φ 的一个 Hintikka 结构。因此,根据引理 3,公式 φ 是可满足的。

最后我们证明算法 2 的完备性。在证明过程中,对于算法 2 构造的图 $T=\langle V,E\rangle$ 中的每个顶点 $v\in V$ 来说,如果存在某个 ALC-LTL 解释结构 $M=(\mathbb{N},I)$ 使得对于任一 $\psi\in v^1$ 都有 $(M,0)\models\psi$,则称顶点 v 是可满足的。

定理 3 对于 ALC-LTL 中的任一公式 φ,如果 φ 是可满

足的，则算法 2 将返回 TRUE。

证明：令存在某个解释结构 $M=(\mathbb{N}, I)$ 使得 $(M,0)\models\varphi$。

首先，令 $T_1=\langle V_1,E_1\rangle$ 是算法 2 在执行完步骤(1)之后得到的有向图，则根据引理 1 和引理 2，必然存在某个饱和状态 $u\in V_1$ 使得 $\varphi\in u^1$，并且顶点 u 是可满足的。

接下来，令 $T_2=\langle V_2,E_2\rangle$ 是算法 2 通过步骤(2)消去所有伪状态之后得到的有向图，则对于每个饱和状态 $v\in V_2$，当其是一个可满足的顶点时，容易验证有以下结论成立。

①必然存在某个饱和状态 $v'\in V_2$ 使得 $\langle v,v'\rangle\in E_2$，并且 v' 是一个可满足的顶点；

②如果 $\varphi_1 u\varphi_2\in v$，则必然存在某条路径 $v_0v_1\cdots v_k$（其中 $k\geqslant 0$）使得 $v=v_0$，每个顶点 v_i（其中 $0\leqslant i\leqslant k$）都是一个可满足的顶点，对于任一 $0\leqslant i<k$ 都有 $\varphi_1 u\varphi_2\in v_i^1$ 和 $\varphi_1\in v_i^1$，以及有 $\varphi_1 u\varphi_2\in v_k^1$ 和 $\varphi_2\in v_k^1$。

最后，通过对算法 2 执行步骤(3)时的循环次数进行归纳，容易证明如下结论：算法 2 在执行步骤(3)的过程中不会删除任何一个可满足的顶点。

综上所述，执行完步骤(3)之后得到的图中仍然存在某个顶点 u 使得 $\varphi\in u^1$，因而算法 2 将返回 TRUE。

4 讨论及相关工作比较

时态描述逻辑 *ALC*-LTL 实际上代表了一类时态描述逻辑的构建方法。将 *ALC*-LTL 中的描述逻辑 *ALC* 替换为其它任何一个描述逻辑 X 之后，可以类似地构建出相应的时态描述逻辑 X-LTL。与之相对应，本文给出的算法中将描述逻辑 *ALC* 的推理机制作为一个模块进行调用，因而具有很好的可扩展性。当 *ALC*-LTL 中的描述逻辑从 *ALC* 改变为任何一个具有可判定性特征的描述逻辑 X 时，只需要将该算法中用到的"判断布尔 *ALC* 知识库是否一致"的模块替换为由描述逻辑 X 提供的"判断布尔 X 知识库是否一致"的模块，就可以得到相应的时态描述逻辑 X-LTL 的 Tableau 判定算法。

高效的 Tableau 判定算法是描述逻辑得到成功应用的关键因素。在描述逻辑的发展历程中，每当一个描述逻辑系统被构建出来之后，研究者都会相应地为其开发一个 Tableau 判定算法[5]。对于描述逻辑的各种扩展形式来说，Tableau 判定算法也是研究者在研究判定算法时的首要选择。针对将模态逻辑 K 与描述逻辑 *ALC* 结合后得到的模态描述逻辑 K_{ALC}，文献[6]给出了相应的 Tableau 判定算法。针对将命题动态逻辑 PDL、描述逻辑 *ALC* 以及动作理论结合后得到的动态描述逻辑 DDL，文献[7]给出了相应的 Tableau 判定算法。与这些工作相比，本文研究的是时态描述逻辑的 Tableau 判定算法。

针对将命题时态逻辑 PTL 与描述逻辑 *ALC* 结合后得到的时态描述逻辑 PTL_{ALC}，Lutz 等[8]，借助 quasimodel 技术给出了相应的 Tableau 判定算法，并证明了该算法的可终止性、可靠性和完备性。与本文研究的 *ALC*-LTL 相比，PTL_{ALC} 的主要特点在于将时态算子 X 和 u 既用于公式的构造又用于概念的构造，因而应用 PTL_{ALC} 可以对具有时态内涵的概念进行刻画和推理。然而正如本文在引言部分所述，这样的时态描述逻辑通常具有很高的推理复杂度。在文献[8]中，Lutz 等分析了其 Tableau 判定算法具有 2EXPTIME 的复杂度。

为了降低推理复杂度，Sturm 等[9]将 PTL_{ALC} 中的解释域从通常情况下的恒定解释域限制为特殊情况下的扩展解释域，然后再次借助 quasimodel 技术给出了相应的 Tableau 判定算法；Sturm 等对该算法的可终止性、可靠性和完备性进行了证明，并分析了该算法的复杂度为 EXPSPACE。在 Sturm 等给出的 Tableau 判定算法的基础上，Guensel 开发了相应的时序描述逻辑推理机[10]。与这些工作相比，由于 *ALC*-LTL 中仅仅将时态算子 X 和 u 用于公式的构造，使得我们可以在不引入 quasimodel 技术的情况下设计出相应的 Tableau 判定算法，并且算法的复杂度降低到了 EXPTIME。

结束语 时态描述逻辑 *ALC*-LTL 将描述逻辑 *ALC* 与时态逻辑 LTL 的描述能力结合了起来。一方面将 LTL 中使用的命题公式提升为描述逻辑 *ALC* 中的一般概念，包含公理、概念断言和角色断言，从而在很大程度上超越了 LTL 的刻画能力；另一方面将 *ALC* 对静态领域的刻画能力扩展到了动态领域，以将 *ALC* 中的知识库通过时间的进展很好地联系和组织起来。

本文给出了 *ALC*-LTL 的 Tableau 判定算法并证明了算法的可终止性、可靠性和完备性。该算法在最坏情况下的复杂度不超过 *ALC*-LTL 中公式可满足性问题本身具有的复杂度。因此，本文给出的判定算法为 *ALC*-LTL 及其代表的一类时态描述逻辑提供了算法支持，为这些逻辑系统的进一步应用打下了基础。我们下一步的工作是对 *ALC*-LTL 的 Tableau 判定算法进行优化，进而开发相应的时态描述逻辑推理机。

参 考 文 献

[1] Baader F, Horrocks I, Sattler U. Description logics as ontology languages for the semantic Web[C]// Hutter D, Stephan W, eds. Mechanizing Mathematical Reasoning: Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday. Berlin: Springer-Verlag, 2005: 228-248

[2] Artale A, Franconi E. A survey of temporal extensions of description logics [J]. Annals of Mathematics and Artificial Intelligence, 2000, 30(1-4): 171-210

[3] Lutz C, Wolter F, Zakharyaschev M. Temporal description logics: a survey[C]//Demri S, Jensen C S, eds. Proceedings of the 15th International Symposium on Temporal Representation and Reasoning. Los Alamitos: IEEE Computer Society Press, 2008: 3-14

[4] Baader F, Ghilardi S, Lutz C. LTL over description logic axioms [C]// Brewka G, Lang J. eds. Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning. Cambridge: AAAI Press, 2008: 684-694

[5] 梅婧，林作铨. 从 ALC 到 SHOQ(D)：描述逻辑及其 Tableau 算法 [J]. 计算机科学, 2005, 32(3): 1-11

[6] Lutz C, Sturm H, Wolter F, et al. A tableau decision algorithm for modalized ALC with constant domains [J]. Studia Logica, 2002, 72(2): 199-232

[7] 常亮，史忠植，邱莉榕，等. 动态描述逻辑的 Tableau 判定算法 [J]. 计算机学报, 2008, 31(6): 896-909

[8] Lutz C, Sturm H, Wolter F, et al. Tableau calculus for temporal description logic: the constant domain case[C]// Gore A, Leitsch A, Nipkow T, eds. Proceedings of the First International Joint Conference on Automated Reasoning. Berlin: Springer-Verlag, 2001: 121-136

[9] Sturm H, Wolter F. A tableau calculus for temporal description logic: the expanding domain case [J]. Journal of Logic and Computation, 2002, 12(5): 809-838

[10] Guensel C. A Tableaux-based Reasoner for Temporalised Description Logics [D]. UK: University of Liverpool, 2005