分支时态描述逻辑 *ALC*-CTL 及其可满足性判定

李 岫 常 亮 孟 瑜 李凤英
(桂林电子科技大学 广西可信软件重点实验室 桂林 541004)

摘 要 时态描述逻辑是将描述逻辑与时态逻辑相结合后得到的逻辑系统,具有较强的描述能力;但是大部分的时态描述逻辑都是将时态算子同时引入到概念和公式中,使得公式可满足性问题的计算复杂度过高。将描述逻辑 *ALC* 与分支时态逻辑 CTL 相结合,提出新的分支时态描述逻辑 *ALC*-CTL。该逻辑没有将时态算子用于概念的构造过程,而是将时态算子引入到公式的构造中;从分支时态逻辑的角度看,相当于将 CTL 中的原子命题提升为描述逻辑中的个体断言。最终得到的逻辑系统不仅具有较强的刻画能力,还使得公式可满足性问题的复杂度保持在 EXPTIME-完全这个级别。通过将 CTL 的 Tableau 判定算法与描述逻辑 *ALC* 的推理机制有机结合,给出了 *ALC*-CTL 的 Tableau 判定算法并证明了算法的可终止性、可靠性和完备性。

关键词 时态描述逻辑,分支时态逻辑,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

Branching Temporal Description Logic *ALC*-CTL and its Satisfiability Decision

LI Shen CHANG Liang MENG Yu LI Feng-ying
(Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract As a combination of the description logic and the temporal logic, the temporal description logic offers considerable expressive power. However most of the researches on temporal description logic have concentrated on the case where temporal operators occur within concepts and formula. In this setting, reasoning usually becomes quite hard. This paper combined standard description logic *ALC* with standard temporal logic CTL and proposed a new temporal description logic *ALC*-CTL. More precisely, we applied temporal operators only to formula not to concept. From the perspective of the computational tree logic, it is equivalent that atomic proposition in CTL is promoted to individual assertion in *ALC*. This *ALC*-CTL not only offers considerable expressive power going beyond both *ALC* and CTL, but also makes the satisfiability problem of it to preserve EXPTIME-complete. Based on a combination of the Tableau algorithm of CTL and the reasoning mechanism provided by *ALC*, this paper presented a Tableau decision algorithm for the logic *ALC*-CTL and proved that this algorithm is terminating, sound and complete.

Keywords Temporal description logic, Branching temporal tree logic, Satisfiability problem, Tableau algorithm, Complexity

1 引言

描述逻辑(Description Logic, DL)作为一种知识表示的形式化工具,广泛应用于自然语言处理、医学本体的构建等科学研究领域。随着 Web3.0 时代的来临,以描述逻辑 DL 为基础的 OWL 成为了 W3C 推荐的 Web 本体语言标准,这使得描述逻辑得到了更多的关注和更快的发展。描述逻辑 DL 的主要特点在于具有较强的描述能力,同时保证了相关推理问题的可判定性,具有有效的推理算法作为支撑[1];它的主要缺点在于只能表示和推理静态领域的知识。例如网上购物,我们希望描述这样一个性质:当用户搜索到自己心仪的商品后,商品通过网上银行或者支付宝付款后将从待售状态变为支付成功、等待发货状态。该性质中具有时间特点的状态变化过程是无法用描述逻辑来刻画的。

针对这一局限,研究者提出了描述逻辑的多种扩展形式,其中时态描述逻辑得到了较多的关注[2,3]。时态描述逻辑主要是将时态算子加入到描述逻辑中,形成具有二维特征的逻辑系统。目前将描述逻辑与时态逻辑相结合时,绝大部分都是将时态算子既作用于概念的构造又作用于公式的构造,这样不仅可以对具有时态内涵的概念进行描述,而且可以对知识库的时态特征进行刻画。研究者证明了这类时态描述逻辑是可判定的,但是推理复杂度较高,达到了 EXPSPACE 或者

到稿日期:2013-05-03 返修日期:2013-09-19 本文受国家自然科学基金(61363030,61100025,61262030),广西自然科学基金(2012GXNSFBA053169,2012GXNSFAA053220),广西可信软件重点实验室基金(KX201109)资助。

李 岫(1988—),男,硕士生,主要研究方向为形式化验证、语义 Web 服务和信息安全;常 亮(1980—),男,博士,教授,CCF 高级会员,主要研究方向为知识表示与推理、形式化方法、智能规划等,E-mail:changl@guet.edu.cn;孟 瑜(1976—),女,博士生,主要研究方向为知识表示与推理、智能规划等;李凤英(1974—),女,博士,副教授,主要研究方向为形式化方法、智能规划等。

2EXPTIME 的程度。为了降低推理复杂度，Badder 等[4]对线性时态描述逻辑 *ALC*-LTL 进行了限制，即仅允许时态算子作用于公式的构造中；分析并证明了该逻辑有效地降低了推理复杂度，在一般情况下 *ALC*-LTL 可满足性问题为 EXPTIME-完全。但是 Baader 的工作是将时态逻辑 LTL 与描述逻辑 *ALC* 结合，针对分支时态描述逻辑 *ALC*-CTL 的研究，到目前为止研究者都尚未对时态算子进行约束，从而导致推理复杂度较高。所以本文中的一个重点部分就是在原有的 *ALC*-CTL 基础上对时态算子进行限制，即不允许时态算子作用于概念的构造中，给出推理复杂度较低的分支时态描述逻辑 *ALC*-CTL。

对于任何一种逻辑问题，我们不仅要研究逻辑本身的一些构造方法，更要研究它的模型检测问题与可满足性判定问题。对于模型检测问题：Huang 等[5]在以描述逻辑 *ALC* 为逻辑基础的本体中，利用 LTL 的过去时态算子刻画本体中所具有的特征，通过模型检测技术对多版本本体的演化进行推理；Pietro 等[6]为了解决 Web 服务组合验证问题，用 *ALC*-CTL 对问题的性质进行了描述，通过模型检测技术对性质进行验证。对于可满足性问题：常亮等[7]在文献[4]的基础上给出了 *ALC*-LTL 相应的 Tableau 推理算法；但是对于本文提出的分支时态描述逻辑 *ALC*-CTL 而言，还缺少有效的判定算法，所以本文中的另一个重点就是以 CTL 的 Tableau 判定算法为基础，在其中引入 *ALC* 的推理机制，设计出 *ALC*-CTL 在一般情况下的 Tableau 判定算法。

本文第 1 节给出了研究的背景以及本篇论文的研究出发点；第 2 节给出了分支时态描述逻辑 *ALC*-CTL 的具体构造方法、语法与语义；第 3 节研究了该逻辑 *ALC*-CTL 的 Tableau 判定算法；对于 Tableau 的算法的可靠性、完备性与可终止性在第 4 节中给出了证明；第 5 节对论文及相关工作进行了比较；最后给出了论文的结论。

2 时态描述逻辑 *ALC*-CTL

由于 ALC-CTL 不允许时态算子应用于概念的构造，因此，从时态逻辑的角度看，也可以将 *ALC*-CTL 看作是在命题分支时态逻辑 CTL 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*-CTL 的主要特点在于将 CTL 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*-CTL 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* 中的概念构造符和分支时态逻辑 CTL 中的公式构造符，可以递归地生成 *ALC*-CTL 的概念和公式。

定义 1 *ALC*-CTL 中的概念由如下产生式生成：

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

其中 $C_i\in N_C$，$R\in N_R$。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*-CTL 中的公式由如下产生式生成：

$\varphi,\psi::=C\sqsubseteq D\mid C(p)\mid R(p,q)\mid\neg\varphi\mid\varphi\wedge\psi\mid \mathbf{EX}\varphi\mid \mathbf{AF}\varphi\mid \mathrm{E}(\varphi\mathbf{U}\psi)$

其中 $p,q\in N_I$，$R\in N_R$，C、D 为概念。此外，可以依次引入形如 false、true、$\varphi\vee\psi$、$\varphi\rightarrow\psi$、$\mathbf{AX}\varphi$、$\mathbf{EF}\varphi$、$\mathbf{EG}\varphi$、$\mathbf{AG}\varphi$、$\mathrm{A}(\varphi\mathbf{U}\psi)$、$\mathrm{E}(\varphi\mathbf{R}\psi)$ 以及 $\mathrm{A}(\varphi\mathbf{R}\psi)$ 等公式，由以下恒等公式表示：

$\mathrm{false}\equiv\varphi\wedge\neg\varphi$　$\mathrm{true}\equiv\neg\mathrm{false}$

$\varphi\vee\psi\equiv\neg(\neg\varphi\wedge\neg\psi)$　$\varphi\rightarrow\psi\equiv\neg\varphi\vee\psi$

$\mathbf{AX}\varphi\equiv\neg\mathbf{EX}\neg\varphi$　$\mathbf{EF}\varphi\equiv\mathrm{E}(\mathrm{true}\ \mathrm{U}\ \varphi)$

$\mathbf{EG}\varphi\equiv\neg\mathbf{AF}\neg\varphi$　$\mathbf{AG}\varphi\equiv\neg\mathbf{EF}\neg\varphi$

$\mathrm{E}(\varphi\mathbf{R}\psi)\equiv\neg\mathrm{A}(\neg\varphi\wedge\neg\psi)$　$\mathrm{A}(\varphi\mathbf{R}\psi)\equiv\neg\mathrm{E}(\neg\varphi\wedge\neg\psi)$

$\mathrm{A}(\varphi\mathbf{U}\psi)\equiv\neg\mathrm{E}(\neg\psi\mathbf{U}(\neg\varphi\wedge\neg\psi))\wedge\neg\mathbf{EG}\neg\psi$

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；将每个形如 $\mathrm{E}(\varphi\mathbf{U}\psi)$、$\neg\mathbf{AG}\varphi$ 的公式称为存在可能性断言；将每个形如 $\mathrm{A}(\varphi\mathbf{U}\psi)$、$\neg\mathbf{EG}\varphi$ 的公式称为任意可能性断言。

从语义上看，*ALC*-CTL 的解释结构与 CTL 的解释结构在整体上相似，通过分支将各个状态树形地组织起来；但与 CTL 解释结构不同的是，*ALC*-CTL 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*-CTL 解释结构是一个四元组 $M=(\mathrm{S},T,\Delta,I)$，其中：

(1)S 为所有状态的集合；

(2)$T\subseteq\mathrm{S}\times\mathrm{S}$，表示状态之间的二元关系，即关系之间的相互转换；

(3)Δ 是解释域；

(4)函数 I 对任一 $s\in\mathrm{S}$ 赋予描述逻辑 *ALC* 的一个解释 $I(s)=(\Delta,\cdot^{I(s)})$，其中解释函数 $\cdot^{I(n)}$ 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 $C_i^{I(s)}\subseteq\Delta$；

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

(iii)将每个个体名 $p_i\in N_I$ 解释为 Δ 中的某个元素 $p_i^{I(s)}\in\Delta$，并且对于任一状态 $s'\in\mathrm{S}$ 都有 $p_i^{I(s)}=p_i^{I(s')}$。

定义 4 给定任一 *ALC*-CTL 解释结构 $M=(\mathrm{S},T,\Delta,I)$，对 *ALC*-CTL 中概念和公式的语义递归定义如下：首先，相对于任一状态 $s\in\mathrm{S}$，将每个概念 C 解释为 Δ 的某个子集 $C^{I(s)}$；递归定义为：

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

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

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

其次，相对于任一状态 $s\in\mathrm{S}$，用 $M=(\mathrm{S},T,\Delta,I)\models\varphi$ 表示公式 φ 在结构 M 中的状态 s 下成立，说明 T 是状态的二元关系，$\{s'\in\mathrm{S}\mid(s,s')\in T\}$ 递归定义如下：

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

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

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

(7)$(M,s)\models\neg\varphi$ iff $(M,s)\not\models\varphi$(即公式 φ 在结构 M 中的

状态 s 下不成立)；

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

(9)$(M,s)\models \mathbf{EX}\varphi$ iff 存在一个 s' 有$(s,s')\in T$ 并且$(M,s')\models\varphi$；

(10)$(M,s)\models \mathbf{AF}\varphi$ iff 对任意一条从 s 出发的路径，总有一个状态 s' 满足 $sT*s'$ 并且$(M,s')\models\varphi$；

(11)$(M,s)\models \mathrm{E}(\varphi\mathbf{U}\psi)$ iff 存在一条从 s 出发的路径，并且存在某个整数 $k\geqslant 0$ 使得$(M,s+k)\models\psi$ 并且对于任一 $0\leqslant i<k$ 都有$(M,s+i)\models\varphi$。

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 $I\models\mathscr{B}$。

显然，描述逻辑 *ALC* 中的每个布尔知识库仅仅是 *ALC*-CTL 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*-CTL 中体现为公式的可满足性问题。

定义 5 对于 *ALC*-CTL 中的任一公式 φ，称 φ 是可满足的当且仅当存在某个 *ALC*-CTL 解释结构 $M=(\mathrm{S},T,\Delta,I)$ 使得$(M,s)\models\varphi$，其中 s 为初始状态。

公式的可满足性问题是时态描述逻辑 *ALC*-CTL 中最基本的推理问题。本文将在下一节给出一个具体的 Tableau 判定算法。

3 *ALC*-CTL 的 Tableau 判定算法

在给出算法之前，下面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*-CTL 公式 φ，我们用 neg(φ)表示定义如下的公式：如果 φ 不是以"$\neg$"开头的公式，则令 neg(φ) $:=\neg\varphi$；否则，如果 φ 形如$\neg\varphi_1$，则令 neg(φ) $:=\varphi_1$。

其次，对于任一 *ALC*-LTL 公式 φ，用 sub(φ)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(φ) $:=\{\varphi\}$；

(2)如果 φ 形如$\neg\varphi_1$ 或者 $\mathbf{EX}\varphi_1$ 或者 $\mathbf{AF}\varphi_1$，则 sub(φ) $:=\{\varphi\}\cup$sub(φ_1)；

(3)如果 φ 形如 $\varphi_1\wedge\varphi_2$ 或者 $\mathrm{E}(\varphi_1\mathbf{U}\varphi_2)$，则 sub($\varphi$) $:=\{\varphi\}\cup$sub(φ_1)$\cup$sub(φ_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(ψ)$\subseteq cl(\varphi)$；

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

(4)如果$\neg\mathbf{EX}\varphi_1\in cl(\varphi)$，则 $\mathbf{AX}\neg\varphi_1\in cl(\varphi)$；

(5)如果$\neg\mathbf{AX}\varphi_1\in cl(\varphi)$，则 $\mathbf{EX}\neg\varphi_1\in cl(\varphi)$；

(6)如果 $\mathbf{EG}\varphi_1\in cl(\varphi)$，则 $\mathbf{EXEG}\varphi_1\in cl(\varphi)$；

(7)如果 $\mathbf{AG}\varphi_1\in cl(\varphi)$，则 $\mathbf{AXAG}\varphi_1\in cl(\varphi)$；

(8)如果$\neg\mathrm{E}(\varphi_1\mathbf{U}\varphi_2)\in cl(\varphi)$，则 $\mathbf{AX}\neg\mathrm{E}(\varphi_1\mathbf{U}\varphi_2)\in cl(\varphi)$；

(9)如果$\neg\mathrm{A}(\varphi_1\mathbf{U}\varphi_2)\in cl(\varphi)$，则 $\mathbf{EX}\neg\mathrm{A}(\varphi_1\mathbf{U}\varphi_2)\in cl(\varphi)$；

(10)如果$\neg\mathbf{EG}\varphi_1\in cl(\varphi)$，则 $\mathbf{AX}\neg\mathbf{EG}\varphi_1\in cl(\varphi)$；

(11)如果$\neg\mathbf{AG}\varphi_1\in cl(\varphi)$，则 $\mathbf{EX}\neg\mathbf{AG}\varphi_1\in cl(\varphi)$；

(12)如果 $\mathrm{E}(\varphi_1\mathbf{U}\varphi_2)\in cl(\varphi)$，则 $\mathbf{EXE}(\varphi_1\mathbf{U}\varphi_2)\in cl(\varphi)$；

(13)如果 $\mathrm{A}(\varphi_1\mathbf{U}\varphi_2)\in cl(\varphi)$，则 $\mathbf{AXA}(\varphi_1\mathbf{U}\varphi_2)\in cl(\varphi)$。

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*-CTL 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 $h(s)$ 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 neg(φ_1)$\notin h$；

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

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

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

(6)如果$\neg\mathbf{EX}\varphi_1\in h$，则必然有 $\mathbf{AX}\neg\varphi_1\in h$；

(7)如果$\neg\mathbf{AX}\varphi_1\in h$，则必然有 $\mathbf{EX}\neg\varphi_1\in h$；

(8)如果 $\mathbf{EG}\varphi_1\in h$，则必然有 $\varphi_1\in h$ 并且 $\mathbf{EXEG}\varphi_1\in h$；

(9)如果 $\mathbf{AG}\varphi_1\in h$，则必然有 $\varphi_1\in h$ 并且 $\mathbf{AXAG}\varphi_1\in h$；

(10)如果$\neg\mathbf{EG}\varphi_1\in h$，则必然有$\neg\varphi_1\in h$ 或者 $\mathbf{AX}\neg\mathbf{EG}\varphi_1\in h$；

(11)如果$\neg\mathbf{AG}\varphi_1\in h$，则必然有$\neg\varphi_1\in h$ 或者 $\mathbf{EX}\neg\mathbf{AG}\varphi_1\in h$；

(12)如果 $\mathrm{E}(\varphi_1\mathbf{U}\varphi_2)\in h$，则必然有 $\varphi_2\in h$ 或者 $\varphi_1\wedge\mathbf{EXE}(\varphi_1\mathbf{U}\varphi_2)\in h$；

(13)如果 $\mathrm{A}(\varphi_1\mathbf{U}\varphi_2)\in h$，则必然有 $\varphi_2\in h$ 或者 $\varphi_1\wedge\mathbf{AXA}(\varphi_1\mathbf{U}\varphi_2)\in h$；

(14)如果$\neg\mathrm{E}(\varphi_1\mathbf{U}\varphi_2)\in h$，则必然有$\neg\varphi_2\in h$，$\neg\varphi_1\in h$ 或者$\neg\varphi_2\in h$，$\mathbf{AX}\neg\mathrm{E}(\varphi_1\mathbf{U}\varphi_2)\in h$；

(15)如果$\neg\mathrm{A}(\varphi_1\mathbf{U}\varphi_2)\in h$，则必然有$\neg\varphi_2\in h$，$\neg\varphi_1\in h$ 或者$\neg\varphi_2\in h$，$\mathbf{EX}\neg\mathrm{A}(\varphi_1\mathbf{U}\varphi_2)\in h$。

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(Γ) $:=\varnothing$；

(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 neg(φ_1)$\notin\Gamma$，则将 Γ 加入集合 FE(Γ)；

(3)对于任一集合 h$\in$FE(Γ)，如果其不是公式 φ 的 Hintikka 集，则将其从 FE(Γ)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$\wedge$-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$\cup\{\varphi_1,\varphi_2\}$；

③$\neg\wedge$-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$\cup\{\neg\varphi_2\}$；

④$\neg$EX-规则：如果$\neg\mathbf{EX}\varphi_1\in$h 并且 $\mathbf{AX}\neg\varphi_1\notin$h，则令 h_1 $:=$h$\cup\{\mathbf{AX}\neg\varphi_1\}$；

⑤¬AX-规则：如果¬AX$\varphi_1 \in$ h 并且 EX¬$\varphi_1 \notin$ h，则令 h_1 :=h∪{EX¬φ_1}；

⑥EG-规则：如果 EG$\varphi_1 \in$ h 并且 $\varphi_1 \notin$ h，EXEG$\varphi_1 \notin$ h，则 $h_1(s)$:=h∪{φ_1，EXEGφ_1}；

⑦AG-规则：如果 AG$\varphi_1 \in$ h 并且 $\varphi_1 \notin$ h，AXAG$\varphi_1 \notin$ h，则 $h_1(s)$:=h∪{φ_1，AXAGφ_1}；

⑧¬EU-规则：如果¬E(φ_1Uφ_2)∈h 并且¬$\varphi_2 \notin$ h，¬$\varphi_1 \notin$ h，则 h_1 :=h∪{¬φ_2，¬φ_1}；如果¬E(φ_1Uφ_2)∈h 并且¬$\varphi_2 \notin$ h，AX¬E(φ_1Uφ_2)$\notin$ h，则 h_2 :=h∪{¬φ_2，AX¬E(φ_1Uφ_2)}；

⑨¬AU-规则：如果¬A(φ_1Uφ_2)∈h 并且¬$\varphi_2 \notin$ h，¬$\varphi_1 \notin$ h，则 h_1 :=h∪{¬φ_2，¬φ_1}；如果¬A(φ_1Uφ_2)∈h 并且¬$\varphi_2 \notin$ h，EX¬A(φ_1Uφ_2)$\notin$ h，则 h_2 :=h∪{¬φ_2，EX¬A(φ_1Uφ_2)}；

⑩¬EG-规则：如果¬EG$\varphi_1 \in$ h 并且¬$\varphi_1 \notin$ h，则 h_1 :=h∪{¬φ_1}；如果¬EG$\varphi_1 \in$ h 并且 AX¬EG$\varphi_1 \notin$ h，则 h_2 :=h∪{AX¬EGφ_1}；

⑪¬AG-规则：如果¬AG$\varphi_1 \in$ h 并且¬$\varphi_1 \notin$ h，则 h_1 :=h∪{¬φ_1}；如果¬AG$\varphi_1 \in$ h 并且 EX¬AG$\varphi_1 \notin$ h，则 h_2 :=h∪{EX¬AG$\varphi_1 \notin$ h}；

⑫EU-规则：如果 E(φ_1Uφ_2)∈h 并且 $\varphi_2 \notin$ h，则 h_1 :=h∪{φ_2}；如果 E(φ_1Uφ_2)∈h 并且 EXE(φ_1Uφ_2)$\notin$ h，$\varphi_1 \notin$ h，则 h_1 :=h∪{EXE(φ_1Uφ_2)，φ_1}；

⑬AU-规则：如果 A(φ_1Uφ_2)∈h 并且 $\varphi_2 \notin$ h，则 h_1 :=h∪{φ_2}；如果 A(φ_1Uφ_2)∈h 并且 AXA(φ_1Uφ_2)$\notin$ h，$\varphi_1 \notin$ h，则 h_2 :=h∪{AXA(φ_1Uφ_2)，φ_1}；

(4)对于上述扩展后得到的任一集合 h′∈{h_1，h_2}，在 h′∉FE(Γ)的情况下，如果对于任一公式 φ_1∈h′都有 neg(φ_1)∉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(Γ)；

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

显然，上述扩展过程中生成的每个集合都是 $cl(\varphi)$的子集。通过对扩展规则进行考察，容易证明以下结论：

引理 1 对于任一集合 Γ⊆$cl(\varphi)$，算法 1 是可终止的，并且最终返回的 FE(Γ)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 Γ⊆$cl(\varphi)$，如果存在某个解释结构 $M=(S,T,\Delta,I)$使得对于任一公式 $\psi\in\Gamma$ 都有$(M,s)\models\psi$，s 为初始状态，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$(M,s)\models\gamma$。

Tableau 判定算法的基本思路是判断能否为给定的公式构造出一个 Tableau。其中，对 *ALC*-CTL 中公式的 Tableau 定义如下。

定义 6 对于 *ALC*-CTL 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(T12)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 Γ={ψ| EX$\psi\in v$}或者 Γ={ψ|A**X**$\psi\in v$}，则必然有 Γ⊆u；

(T5)对于任意一个顶点 $v\in V$，如果¬AG$\psi\in v$，则必然存在某条路径 $v_0v_1\cdots$使得：$v=v_0$，¬$\psi\in v_i$，$i\in N$；

(T6)对于任意一个顶点 $v\in V$，如果 E(φ_1Uφ_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 φ_1U$\varphi_2\in v_i$ 和 $\varphi_1\in v_i$，以及有 φ_1U$\varphi_2\in v_k$ 和 $\varphi_2\in v_k$；

(T7)对于任意一个顶点 $v\in V$，如果¬EG$\psi\in v$，则对任意一条路径 $v_0v_1\cdots$使得：$v=v_0$，¬$\psi\in v_i$，$i\in N$；

(T8)对于任意一个顶点 $v\in V$，如果 A(φ_1Uφ_2)$\in v$，则对任意一条路径 $v_0v_1\cdots v_k$(其中 $k\geqslant 0$)使得：$v=v_0$，对于任一 $0\leqslant i<k$都有 φ_1U$\varphi_2\in v_i$ 和 $\varphi_1\in v_i$，以及有 φ_1U$\varphi_2\in v_k$ 和 $\varphi_2\in v_k$；

(T9)对于任意一个顶点 $v\in V$，如果 EG$\psi\in v$，则必然存在某条路径 $v_0v_1\cdots$使得：任意顶点 v_i 有 $\psi\in v_i$，$i\in N$；

(T10)对于任意一个顶点 $v\in V$，如果 AG$\psi\in v$，则对任意一条路径 $v_0v_1\cdots$使得：任意顶点 v_i 有 $\psi\in v_i$，$i\in N$；

(T11)对于任意一个顶点 $v\in V$，如果¬E(φ_1Uφ_2)$\in v$，则对任意一条路径 $v_0v_1\cdots$对任意 $i\in N$，若 $\varphi_2\in v_i$ 则¬$\varphi_1\in v_j$，对 j 有 $0\leqslant j<i$；

(T12)对于任意一个顶点 $v\in V$，如果¬A(φ_1Uφ_2)$\in v$，则必然存在某条路径 $v_0v_1\cdots$对任意 $i\in N$，若 $\varphi_2\in v_i$ 则¬$\varphi_1\in v_j$，对 j 有 $0\leqslant j<i$。

为了表述简便，在下面的算法中，将有向图 $T=\langle V,E\rangle$的顶点集 V 由 $2^{d(\varphi)}$的子集扩充为 $2^{d(\varphi)}\times\{\uparrow,\surd\}$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↑，则将该顶点称为伪状态；如果 v^2 为√，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)对于任意一个顶点 $v\in V$，存在两种类型的可能性断言：第一种是存在的可能性断言¬AGψ 和 E(φ_1Uφ_2)；第二种是任意的可能性断言¬EGψ 和 A(φ_1Uφ_2)。对于任意的可能性断言，在判断其是否在顶点实现的情况下，会出现不确定递归。下面给出如何判断在顶点 v 处可能性断言是否能够出现。

(4.1)存在的可能性断言

①对于公式 ξ=E(φ_1Uφ_2)$\in v$，如果存在一条有限路径 $v=v_0v_1\cdots v_k$(其中 $k\geqslant 0$)使得：$\xi\in v_k$，$\varphi_2\in v_k$ 并且 $\xi\in v_i$，$\varphi_2\in v_i$，对于 $i=0,1,\cdots,k-1$，则称可能性断言 E(φ_1Uφ_2)相对于顶点 v 被实现了，否则称 E(φ_1Uφ_2)相对于顶点 v 没有被实现。如果在公式 ξ=E(φ_1Uφ_2)有 $\varphi_2\in v$，我们认为 ξ在 v 处立即成立。

②对于公式 ξ=¬AG$\psi\in v$，如果存在一条有限路径 $v=v_0v_1\cdots v_k$(其中 $k\geqslant 0$)使得：$\xi\in v_k$，¬$\psi\in v_k$ 并且 $\xi\in v_i$，对于 $i=0,1,\cdots,k-1$，则称可能性断言¬AGψ 相对于顶点 v 被实现了，否则称¬AGψ 相对于顶点 v 没有被实现。如果在公式 ξ=¬AGψ 有¬$\psi\in v$，我们认为 ξ在 v 处立即成立。

(4.2)任意的可能性断言

①对于公式 ξ=A(φ_1Uφ_2)$\in v$，如果任意一条路径 $v=v_0v_1\cdots v_k$(其中 $k\geqslant 0$)使得：对任意一条边$\langle v,v'\rangle\in E$ 都有 $\xi\in$

v'成立，则称可能性断言 $E(\varphi_1 U \varphi_2)$相对于顶点 v 被实现了，否则称 $E(\varphi_1 U \varphi_2)$相对于顶点 v 没有被实现。如果在公式 $\xi = A(\varphi_1 U \varphi_2)$有 $\xi \in v$，我们认为 ξ在 v 处立即成立。

②对于公式 $\xi = \neg EG\psi \in v$，如果任意一条路径 $v = v_0 v_1 \cdots v_k$（其中 $k \geqslant 0$）使得：对任意一条边$\langle v, v' \rangle \in E$ 都有 $\xi \in v'$成立，则称可能性断言$\neg EG\psi$ 相对于顶点 v 被实现了，否则称 $\neg EG\psi$ 相对于顶点 v 没有被实现。如果在公式 $\xi = \neg EG\psi$ 有 $\neg\psi \in v$，我们认为 ξ在 v 处立即成立。

最后，对 *ALC*-LTL 的 Tableau 判定算法描述如下。

算法 2 对于任一 *ALC*-CTL 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 $V := \{(\{\varphi\}, \uparrow)\}$，$E := \varnothing$；

(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 中饱和状态 v，若有 $EX\psi \in v$，增加后继伪状态 v^1，构造集合 $\Gamma = \{\psi \in v^1\} \cup \{\varphi_1 \mid AX\varphi_1 \in v^1\}$；若没有 $EX\psi \in v$，则增加后继伪状态 v^1，构造集合 $\Gamma = \{\varphi_1 \mid AX\varphi_1 \in v^1\}$，最后将顶点$(\Gamma, \uparrow)$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，$Post_u = \{w_1, \cdots, w_n\}$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 中有公式类似 $EX\psi$ 并且不存在一个后继状态顶点 v'使得$\langle v, v' \rangle$这条边标注有 $EX\psi$ 关系，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 $E(\varphi_1 U \varphi_2) \in v$ 或者$\neg AG\psi \in v$ 或者 $A(\varphi_1 U \varphi_2) \in v$ 或者$\neg EG\psi \in v$ 并且有相对应的 $E(\varphi_1 U \varphi_2)$或者$\neg AG\psi$ 或者 $A(\varphi_1 U \varphi_2)$或者$\neg EG\psi$ 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。

4 判定算法的性质

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

首先证明算法 2 的可靠性，为了证明，我们先引入 Hintikka 结构及相关性质。

定义 7 对于 *ALC*-CTL 中的任一公式 φ，如果存在某个三元组 $H = (S, R, H)$，满足下面(H1)到(H3)的条件，则将三元组 $H = (S, R, H)$称为公式 φ 的一个 Hintikka 结构：

(H1)S 为状态集合，(S, R)是一个状态转移系统，$H: S \to 2^{cl(\varphi)}$为标记函数；

(H2)$\varphi \in H(s_0)$，s_0 为初始状态；

(H3)对于任一状态 $s \in S$都满足以下条件：

①$H(s)$是公式 φ 的一个 Hintikka 集；

②如果 $EX\varphi_1 \in H(s)$，则有 $\varphi_1 \in H(s')$对存在某个状态 s'有 sRs'；

③如果 $AX\varphi_1 \in H(s)$，则有 $\varphi_1 \in H(s')$对任意一个状态 s'有 sRs'；

④如果$\neg AG\varphi_1 \in H(s)$，则有$\neg\varphi_1 \in H(s')$对存在某个状态 s'有 $sR * s'$；

⑤如果 $E(\varphi_1 U \varphi_2) \in H(s)$，则有 $\varphi_2 \in H(s')$对某个状态 s'有一个 R 路径 $s = s_0, s_1, \cdots, s_n = s'$并且 $\varphi_1 \in H(s_i)$对 $i = 0, \cdots, n-1$；

⑥如果$\neg EG\varphi_1 \in H(s)$，则有对任意一条路径 $s = s_0, s_1, \cdots$总有 $n \in N, \neg\varphi_1 \in H(s_n)$；

⑦如果 $A(\varphi_1 U \varphi_2) \in H(s)$，则有对任意一条路径 $s = s_0, s_1, \cdots$，有 $n \in N, \varphi_2 \in H(s_n)$并且 $\varphi_1 \in H(s_i)$对 $i = 0, \cdots, n-1$。

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 $H = (S, R, H)$是公式 φ 的一个 Hintikka 结构。对于初始状态 $s_0 \in S$，由于将 $H(s_0)$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 $I(s_0) = (\Delta s_0, \cdot^{I(s_0)})$使得对于 $H(s_0)$中的每个 ALC-文字 a 都有 $I(s_0) \models a$；对于其它状态 s 有，可以在 $I(s_0)$的基础上构造出描述逻辑 *ALC* 的某个解释 $I(s) = (\Delta, \cdot^{I(s)})$，使得：$\Delta = \Delta s_0$，对于每个个体名 p 都有 $p^{I(s)} = p^{I(s_0)}$，以及对于 $H(s)$中的每个 φ-文字 a 都有 $I(s) \models a$。显然，$M = (S, T, \Delta, I)$是一个 *ALC*-CTL 解释结构。在此基础上，对于任一状态 $s \in S$以及任一公式 $\psi \in H(s)$，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 $(M, s) \models \psi$。因此，当 $s = s_0$ 时得到 $(M, s_0) \models \varphi$，即公式 φ 是可满足的。

定理 1 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 $T' = \langle V', E' \rangle$构造一个三元组 $H = (S, R, H)$，其中的S为状态的集合，并且(S, R)是一个状态转移系统，H 是从S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 $H(s_0) := v_0$；

(2)令 $s := s_0$，令 E 是由满足存在的可能性断言 $E(\varphi_1 U \varphi_2) \in H(s)$，$\varphi_2 \notin H(s)$或者$\neg AG\varphi_1 \in H(s)$，$\neg\varphi_1 \notin H(s)$，任意的可能性断言 $A(\varphi_1 U \varphi_2) \in H(s)$，$\varphi_2 \notin H(s)$或者

$\neg EG\varphi_1 \in H(s)$，$\neg\varphi_1 \notin H(s)$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$H(s)$的任一后继顶点s'，令$H(s'):=u$，$(s,s')\in R$；否则依次进行以下操作：

对于存在的可能性断言：

①从图T'中找出一条路径$v_0v_1\cdots v_k$(其中$k\geqslant 0$)，使得：$H(s)=v_0$，存在某个状态转移序列$s=s_0\cdots s_j\cdots s_k$和任一存在的可能性断言分别得到：$E(\varphi_1 U\varphi_2)\in E$都有$\varphi_2\notin H(s_j)$或者$\neg AG\varphi_1\in E$都有$\neg\varphi_1\notin H(s_j)$，并且有某个存在的可能性断言$E(\varphi_1 U\varphi_2)\in E$使得$\varphi_2\in H(s_k)$或者$\neg AG\varphi_1\in E$使得$\neg\varphi_1\in H(s_k)$；

②对于状态转移序列$s_1\cdots s_j\cdots s_k$，令$H(s_j):=v_j$；

③对于每个存在的可能性断言$E(\varphi_1 U\varphi_2)\in E$或者$\neg AG\varphi_1\in E$，分别有$\varphi_2\in H(s_k)$或者$\neg\varphi_1\in H(s_k)$，则将公式$E(\varphi_1 U\varphi_2)$或者$\neg AG\varphi_1$从集合$E$中删除；

④对于每个存在的可能性断言$E(\varphi_1 U\varphi_2)\in H(s_k)$，如果$\varphi_2\notin H(s_k)$并且$E(\varphi_1 U\varphi_2)\notin E$或者$\neg AG\varphi_1\in H(s_k)$，如果$\neg\varphi_1\notin H(s_k)$并且$\neg AG\varphi_1\notin E$，则将$E(\varphi_1 U\varphi_2)$或者$\neg AG\varphi_1$加入集合$E$中；

⑤令$s=s_k$，跳转到步骤(3)。

对于任意的可能性断言：

①从图T'中任意一条路径$v_0v_1\cdots v_k$(其中$k\geqslant 0$)，使得：$H(s)=v_0$，对于任一状态转移序列$s=s_0\cdots s_j\cdots s_k$和任一存在的可能性断言分别得到：$A(\varphi_1 U\varphi_2)\in E$都有$\varphi_2\notin H(s_j)$或者$\neg EG\varphi_1\in E$都有$\neg\varphi_1\notin H(s_j)$，并且有某个存在的可能性断言$A(\varphi_1 U\varphi_2)\in E$使得$\varphi_2\in H(s_k)$或者$\neg EG\varphi_1\in E$使得$\neg\varphi_1\in H(s_k)$；

②对于状态转移序列$s_1\cdots s_j\cdots s_k$，令$H(s_j):=v_j$；

③对于每个存在的可能性断言$A(\varphi_1 U\varphi_2)\in E$或者$\neg EG\varphi_1\in E$，分别有$\varphi_2\in H(s_k)$或者$\neg\varphi_1\in H(s_k)$，则将公式$A(\varphi_1 U\varphi_2)$或者$\neg EG\varphi_1$从集合$E$中删除；

④对于每个存在的可能性断言$A(\varphi_1 U\varphi_2)\in H(s_k)$，如果$\varphi_2\notin H(s_k)$并且$A(\varphi_1 U\varphi_2)\notin E$或者$\neg EG\varphi_1\in H(s_k)$，如果$\neg\varphi_1\notin H(s_k)$并且$\neg EG\varphi_1\notin E$，则将$A(\varphi_1 U\varphi_2)$或者$\neg EG\varphi_1$加入集合$E$中；

⑤令$s=s_k$，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三元组$H=(S,R,H)$是公式φ的一个Hintikka结构。又根据引理3，公式φ是可满足的，由此得证算法2是可靠的。

接下来证明算法2的完备性。在证明过程中，对于算法2构造的有向图$T=\langle V,E\rangle$中的每个顶点$v\in V$来说，如果存在某个*ALC*-CTL解释结构$M=(S,T,\Delta,I)$使得对于任一$\psi\in v^1$都有$(M,s)\models\psi$，则称顶点v是可满足的。

定理2 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$M=(S,T,\Delta,I)$使得$(M,s)\models\varphi$。

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$E(\varphi_1 U\varphi_2)\in v$，则必然存在某条有限路径$v_0v_1\cdots v_k$(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$E(\varphi_1 U\varphi_2)\in v_i^1$和$\varphi_1\in v_i^1$，以及有$E(\varphi_1 U\varphi_2)\in v_k^1$和$\varphi_2\in v_k^1$。

③如果$\neg AG\varphi_1\in v$，则必然存在某条有限路径$v_0v_1\cdots v_k$(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$\neg AG\varphi_1\in v_i^1$以及有$\neg AG\varphi_1\in v_k^1$和$\neg\varphi_1\in v_k^1$。

④如果$A(\varphi_1 U\varphi_2)\in v$，则必然对任意一条有限路径$v_0v_1\cdots v_k$(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$A(\varphi_1 U\varphi_2)\in v_i^1$和$\varphi_1\in v_i^1$，以及有$A(\varphi_1 U\varphi_2)\in v_k^1$和$\varphi_2\in v_k^1$。

⑤如果$\neg EG\varphi_1\in v$，则必然对任意一条有限路径$v_0v_1\cdots v_k$(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$\neg EG\varphi_1\in v_i^1$以及有$\neg EG\varphi_1\in v_k^1$和$\neg\varphi_1\in v_k^1$。

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$\varphi\in u^1$，算法2将返回TRUE。

最后我们证明算法2是可终止的，并讨论分析其算法复杂度。

定理3 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的长度呈指数关系。

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$\varphi$的长度呈线性关系，因此在最坏情况下有向图$T$中的顶点个数与公式$\varphi$的长度呈指数关系。

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的长度呈线性关系，而且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[5]，因此，生成顶点v所需要的时间开销与公式φ的长度呈指数关系。

综上所述，算法2是可终止的。并且，将有向图T中的顶点个数与生成每个顶点所需要的时间乘积起来之后可知，算法2在最坏情况下所需要的时间与公式φ的长度呈指数关系。

5 讨论及相关工作比较

描述逻辑*ALC*作为一种逻辑语言，为了解决不同的应用需求而提出了多种扩展方式：文献[8]提出动作描述逻辑

DDL，即将命题动态逻辑、描述逻辑 *ALC* 以及动作理论相结合，并且在该逻辑基础上做了事件检测方面的有关研究。针对描述逻辑 *ALC* 与时态逻辑 LTL 相结合，文献[4]给出了多种不同的结合方式，又针对可满足性问题分析了相应的时间复杂度，文献[9]将该逻辑应用到运行时验证当中。文献[10]给出了描述逻辑 *ALC* 与时态逻辑 CTL 相结合(时态算子作用于概念构造与公式构造当中，推理时间复杂度较高)，并且在该逻辑的基础上做了生成反例方面的研究。与这些工作相比，本文研究的是描述逻辑 *ALC* 与时态逻辑 CTL 相结合，将时态算子仅仅作用于公式的构造中，有效地降低了时间复杂度。

任何逻辑都需要高效的 Tableau 判定算法作为支撑。在描述逻辑的研究过程中，任何一个描述逻辑系统的提出都会有相应的 Tableau 判定算法[11]。对于描述逻辑的扩展形式，Tableau 算法也成为研究的重点。针对模态逻辑 K 与描述逻辑 *ALC* 结合后得到的模态描述逻辑 K_{ALC}，文献[12]给出了相应的 Tableau 判定算法。又如上文提到的动态描述逻辑 DDL，文献[13]给出了相应的 Tableau 算法；文献[7]给出了时态描述逻辑 *ALC*-LTL 的 Tableau 算法。而本文针对所研究的时态描述逻辑 *ALC*-CTL 给出了具体的 Tableau 算法。

任何一种 Tableau 算法都需要证明算法的可终止性、可靠性和完备性，并且算法的时间复杂度是其重要的衡量标准。文献[14]将命题时态逻辑 PTL 与描述逻辑 *ALC* 结合得到时态描述逻辑 PTL_{ALC}，并借助 quasimodel 技术给出了相应的 Tableau 判定算法，随后对算法进行了相关的证明。与本文研究的 *ALC*-CTL 相比，PTL_{ALC} 的主要特点是将时态算子 X 和 U 既用于概念构造又用于公式构造当中，这样导致推理复杂度较高，Tableau 算法达到了 2EXPTIME，进而又改进算法使复杂度降低到 EXPSPACE[15]。本文给出了 *ALC*-CTL 的 Tableau 算法，并且指出算法复杂度为 EXPTIME。

将 *ALC*-CTL 中的描述逻辑替换为其它任何一个描述逻辑 X 之后，可以类似地构建出一类相应的时态描述逻辑 X-CTL。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*-CTL 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-CTL 的 Tableau 判定算法。

结束语 本文首先提出新的时态描述逻辑 *ALC*-CTL 结合方法，即将描述逻辑 *ALC* 与分支时态逻辑 CTL 相结合，并且限定了时态算子只允许应用于公式构造当中，这样不仅使得描述逻辑 *ALC* 扩展到了时间维度，提高了刻画能力；并且保证了可判定性，其推理判断的复杂度相对于其他结合方式有所降低。

针对 *ALC*-CTL，本文给出其 Tableau 判定算法并证明了算法的可靠性、完备性和可终止性。该算法为 *ALC*-CTL 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。我们进一步的工作是对算法进行优化，并且开发相应的时态描述逻辑推理机，将其应用到实际问题当中。

参 考 文 献

[1] Baader F, Calvanese D, McGuinness D, et al. The Description Logic Handbook: Theory, Implementation and Applications [M]. Cambridge: Cambridge University Press, 2002

[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. 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] Huang Z, Stuckenschmidt H. Reasoning with Multiversion Ontologies: A Temporal Logic Approach [C]// Gil Y, Motta E, eds. Proceedings of the 4th International Semantic Web Conference. Lecture Notes in Computer Science, 2005, 3729: 398-412

[6] Pietro I, Pagliarecci F, Spalazzi L. Model Checking Semantically Annotated Services [J]. IEEE Transaction on Software Engineering, 2011, 38(3): 592-608

[7] 常亮，王娟，古天龙，等. 时态描述逻辑 ALC-LTL 的 Tableau 判定算法[J]. 计算机科学，2011，38(8)：150-154

[8] Xiaofeng W, Liang C, Zhixin L, et al. A dynamic description logic based system for video event detection [J]. Frontiers of Electrical and Electronic Engineering in China, 2010, 5(2): 137-142

[9] Badder F, Bauer A, Lippmann M. Runtime verification using a temporal description logic [C]// Ghilardi S, Sebastiani R, eds. Proceedings of the 7th International Symposium on Frontiers of Combining Systems. Berlin: Springer-Verlag, 2009: 149-164

[10] Weitl F, Nakajima S, Freitag B. Structured counter-examples for the temporal description logic ALCCTL [C]// Fiadeiro J, Gnesi S, eds. Proceedings of the 2010 8th IEEE International Conference on Software Engineering and Formal Methods. IEEE Computer Society, 2010: 232-243

[11] 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

[12] 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

[13] 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

[14] 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

[15] Sturm H, Wolter F. A tableau calculus for temporal description logic: the expanding domain case [J]. Journal of Logic and Computation, 2002, 12(5): 809-83