命题时态逻辑定理证明新方法*

贲可荣 陈火旺

(长沙工学院计算机系,长沙 410073)

摘要 本文通过对近10年命题时态逻辑定理证明方法的研究,提出了一种新的证明方法.前人的工作基于对公式的现时部分和后时部分的分解,本文的工作是基于语义反驳树构造.这种新方法为计算机自动证明命题时态逻辑定理,提供了比较好的理论框架.最后还证明了该方法的可靠性和完全性.

关键词 时态逻辑,定理证明,自动推理.

时态逻辑作为一种特殊的模态逻辑,最初是用于研究自然语言的,后来逻辑学家对时态逻辑本身做了大量的研究,一些典型的工作.诸如,时态算子的功能完全性,区间时态逻辑与点时态逻辑的关系等.

最近20多年,时态逻辑在计算机科学特别是AI和软件工程领域发挥了作用.诸如,分析复杂的硬件和软件系统,并发系统的综合,程序验证等.

对时态逻辑的这些应用,大多数都需要时态逻辑的证明系统.这方面,许多人做了有益的工作,例如,M. Abadi 1987年的博士论文题为:"时态逻辑定理证明",G. Venketezh 以及A. R. Cavalli, D. A. Plaisted 等人的判定方法.

我们通过研究和比较,觉得上述诸方法,均基于一种思路,即将时态公式分解成现时及后时两部分,分而治之,其缺陷是,证明复杂,不易学习,复杂度较高.

本文中,我们提出了一种证明方法,基于时态逻辑的语义,而不是前人那种基于语法的思想,利用这种新方法,可以将参考文献中比较容易的典型例子,直观易懂地给予解决.

1 命题时态逻辑

时态逻辑很多,随时间结构,算子的选择而异.本文选用Z. Manna及A. Pnunli在并发程序验证方面提出的一种时态逻辑,这种逻辑既一般又简单,其时间模型为:离散、线性、有头无尾.时态算子选择为□,◇,○,U.

1.1 语法

命题常元: true,false

---

* 本文1992-01-03收到,1992-04-03定稿

本项目得到863计划软件生产自动化课题的部分资助.贲可荣,31岁,讲师,主要研究领域为数理逻辑,定理证明,人工智能.陈火旺,58岁,教授,主要研究领域为计算机软件,人工智能,自然语言理解.

本文通讯联系人:陈火旺,长沙 410073,长沙工学院计算机系

命题符号：p,q,r,…

连接词：¬,→

时态算子：□(总是),◇(可能),○(下一时刻),U(直到)

合适公式：(1) true,false 和命题符号是 wffs,

(2) 如果 u,v 是 wffs,则 ¬u, u→v,○u,□u,◇u, uUv 也是 wffs

(3) 合适公式仅为 (1),(2)所构造.

1.2 语义

定义. 模型是三元组$(M,S,s_0s_1\cdots)$,这里,S 是状态集,M:对每一状态都指派一命题符号子集,该集合中元素在该状态下为真,$s_0s_1\cdots$是 S 中无穷状态序列.

满足关系"⊨"定义如下：

$(M,S,s_0\cdots)\models p$ iff $p\in M(s_0)$ 这里 p 是命题符号

$(M,S,s_0\cdots)\models \neg u$ iff not $(M,S,s_0\cdots)\models u$

$(M,S,s_0\cdots)\models u\rightarrow v$ iff 或者$(M,S,s_0\cdots)\models \neg u$ 或者 $(M,S,s_0\cdots)\models v$

$(M,S,s_0\cdots)\models \bigcirc u$ iff $(M,S,s_1\cdots)\models u$

$(M,S,s_0\cdots)\models \Box u$ iff 对所有 $i\geq 0$, $(M,S,s_i\cdots)\models u$

$(M,S,s_0\cdots)\models \Diamond u$ iff 存在 $i\geq 0$, $(M,S,s_i\cdots)\models u$

$(M,S,s_0\cdots)\models uUv$ iff 存在 $i\geq 0$ $(M,S,s_i\cdots)\models v$

并且对满足 $0\leq j<i$ 的所有 j,$(M,S,s_j\cdots)\models u$

2 反驳树的基本概念

本文所述反驳树方法,限制到古典命题逻辑,为表列(tableaux)方法. 该方法是构造一棵向下生长的反驳树,顶点(树根)为待反驳公式(公式集),结点之间构成偏序,对每一结点,有联系于该结点的有穷公式集,通常将一结点与联系于它的一集公式视为等同.

每个结点属于唯一层,n+1 层中的每个结点是小于等于 n 层中某结点的后继. 如果没有后继结点,则最底层的那结点称为终止结点.

一有穷分枝是一结点有穷序列：$\Phi_0,\cdots,\Phi_k$,其中：Φ_0 是图的初始结点,对 $i=1,\cdots,k$,Φ_i 是 $\Phi_0,\cdots,\Phi_{i-1}$的后继,Φ_k 是终止结点,该枝称为终止于 Φ_k. 类似定义无穷分枝.

很明显,对每个终止结点,存在唯一枝终点于它.

一公式属于某一枝上的一个结点,称为该枝的公式,一公式属于初始结点,称为初始公式.

3 命题时态逻辑反驳树构造

如果 Φ_0 是一有穷公式集,以 Φ_0 作为唯一结点的树,称为 Φ_0 的命题树. 这里,Φ_0 既是始点,又是终点,且仅有一个分枝. 假设已得到 Φ_0 的命题树 T,我们通过下述规则之一,将它扩充成一新树 T′. T′具备 T 的所有结点,并加一个或二个新的结点,T 中原有的"后继"关系在 T′中仍成立,以 T 中的结点产生出新的结点,作为其后继.

规则¬¬：如果树 T 的一个分枝终止于结点 Φ,Φ 中有公式¬¬u,则增加一新结点{u}

作为 Φ 的后继.

规则→：$u\to v$
$$\begin{array}{cc} \neg u & v \end{array}$$

规则¬→：$\neg(u\to v)$
$$\begin{array}{c} u \\ \neg v \end{array}$$

原子规则：p
$$s_0 : p$$

关于时态算子的规则如下：

规则¬□：$\neg\Box u$
$$\Diamond\neg u$$

规则¬◇：$\neg\Diamond u$
$$\Box\neg u$$

规则□：$\Box u$
$$\forall_{i\geq 0}\, s_i : u$$

规则◇：$\Diamond u$
$$\exists_{i\geq 0}\, s_i : u$$

规则$\bigcirc^m(m\geq 1)$：$\bigcirc^m u$
$$s_m : u$$

规则$\neg\bigcirc^m(m\geq 1)$：$\neg\bigcirc^m u$
$$s_m : \neg u$$

规则□◇：$\Box\Diamond u$
$$\overset{\infty}{\exists}\, s_i : u$$

规则◇□：$\Diamond\Box u$
$$\exists i\forall j\geq i(s_j : u)$$

规则□□：$\Box\Box u$
$$\Box u$$

规则◇◇：$\Diamond\Diamond u$
$$\Diamond u$$

规则□◇□：$\Box\Diamond\Box u$
$$\Diamond\Box u$$

规则◇□◇：$\Diamond\Box\Diamond u$
$$\Box\Diamond u$$

规则$\bigcirc^m\Diamond$：$\bigcirc^m\Diamond u$
$$\exists_{i\geq m}\, s_i : u$$

规则$\bigcirc^m\Box$：$\bigcirc^m\Box u$
$$\forall_{i\geq m}\, s_i : u$$

规则$\Diamond\bigcirc^m$：$\Diamond\bigcirc^m u$
$$\exists_{i\geq m}\, s_i : u$$

规则$\Box\bigcirc^m$：$\Box\bigcirc^m u$
$$\forall_{i\geq m}\, s_i : u$$

规则U：$u\,U\,v$
$$\begin{array}{c} \exists\, j s_j : v \\ s_0\cdots s_{j-1} : u \end{array}$$

规则¬U：$\neg(u\,U\,v)$

如 $s_j : v$ 则$\exists\, k(k<j$ 且 $s_k : \neg u)$

规则○→：$\bigcirc(u\to v)$
$$\bigcirc u\to\bigcirc v$$

规则◇→：$\Diamond(u\to v)$
$$\exists\, i s_i : u\to v$$

规则□→：$\Box(u\to v)$
$$\forall\, i s_i : u\to v$$

这里给出 5 条注释：

(1)古典连接词∨，∧，→可通过¬，→定义，因而可诱导相应的转化规则.

(2)规则□→使用后得到$\forall\, i s_i : u\to v$ 不等价于$\forall\, i\, s_i : u\to\forall\, i\, s_i : v$，实际上，$\forall\, i s_i : u\to v$ 相当于无穷多个公式 $s_i : u\to v, i=0,1,\cdots$.

(3)在反驳树构造过程中，将 $\alpha *$ 表示公式 α 已用有关规则转化为别的公式.

(4)在反驳树构造过程中，将□→规则之外的规则先实施完毕.

(5)称一枝是封闭的，如果该枝中同时存在 $s_i : \alpha, s_i : \neg\alpha$，对某个状态 s_i 和某个文字 α.

4 示 例

在构造反驳树时,需要如下原则:

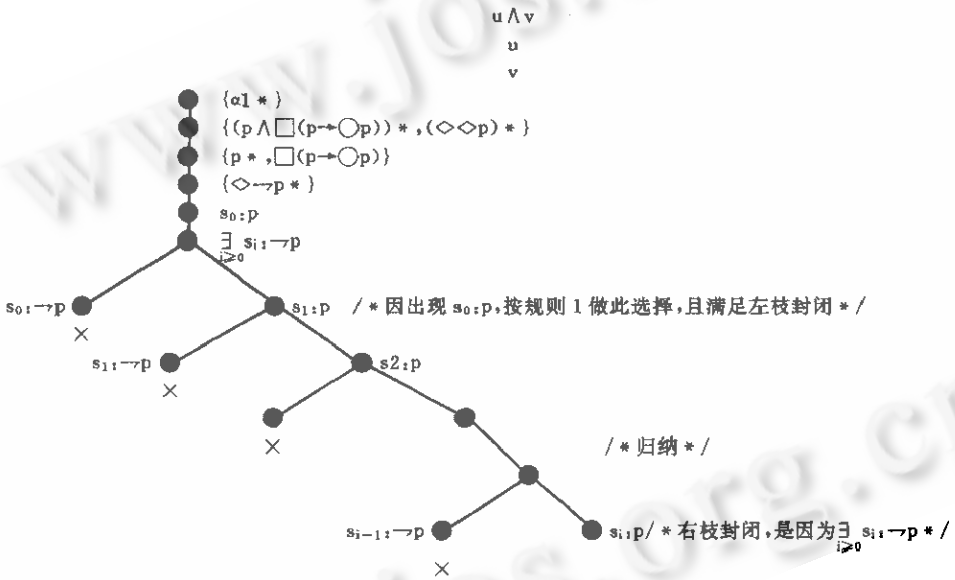
原则 1. 如果出现某个确定的 i,满足 $s_i:\alpha$ 或 $s_i:\alpha \vee \varphi$,其中 α 是文字,则检查形如$\Box(\varphi \rightarrow \psi)$公式中$\neg\varphi$或$\psi$是否为$\neg\alpha$,若有,则取 $s_i:\neg\varphi \vee \psi$.

原则 2. 如果出现某个不确定的 i,亦即形如$\exists\ i \geqslant j\ \alpha(i)$,其中 α 是文字,则设法寻找形如$\forall\ i \geqslant j \rightarrow \alpha(i)$的公式,反过来,如果出现$\forall\ i \geqslant j\alpha(i)$,则寻找$\exists\ i \geqslant j \rightarrow \alpha(i)$的公式.(为方便起见,这里我们用 $\alpha(i)$表示 $s_i:\alpha$)

下面给出四个例子说明我们的方法.

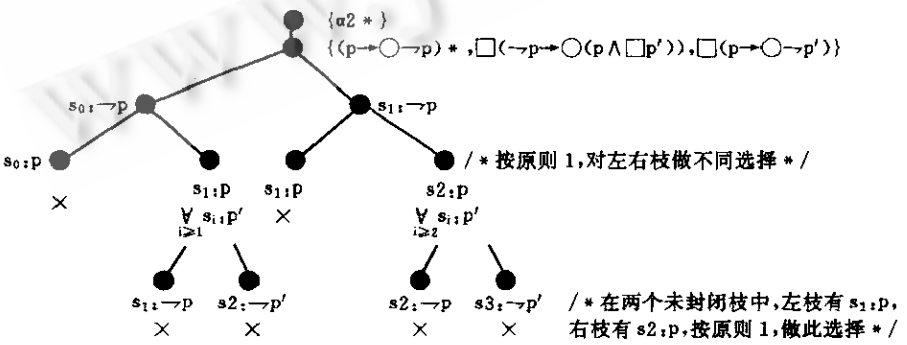
(1)反驳 $\alpha 1 = = (p \wedge \Box(p \rightarrow \bigcirc p)) \wedge \Diamond\Diamond\neg p$

解:逐条使用所给规则,这里,我们还使用两次诱导 $\wedge$ 规则:



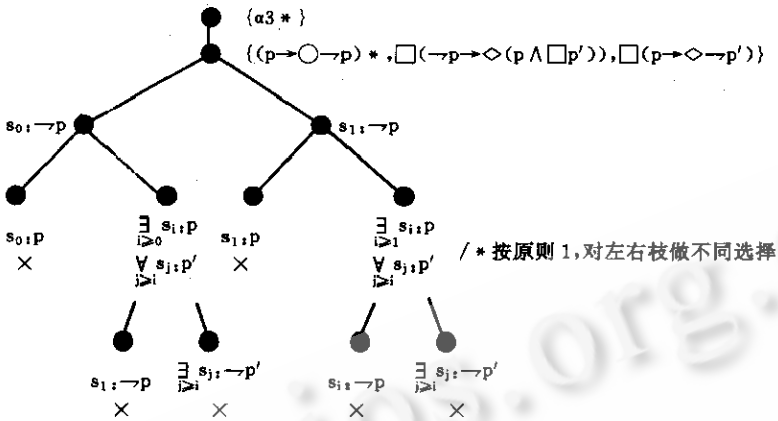
(2)反驳 $\alpha 2 = = (p \rightarrow \bigcirc\neg p) \wedge \Box(\neg p \rightarrow \bigcirc(p \wedge \Box p')) \wedge \Box(p \rightarrow \bigcirc\neg p')$

解:



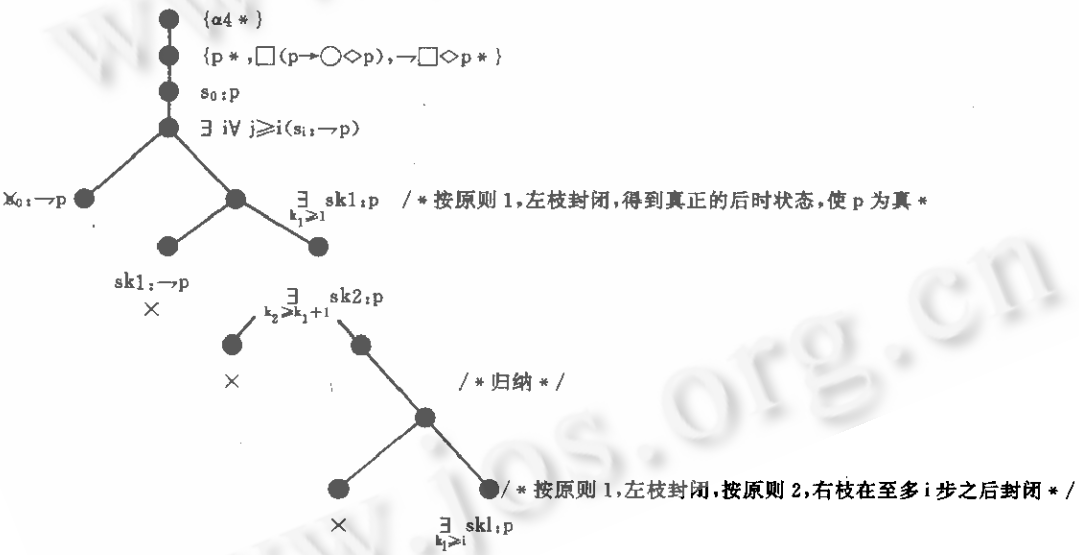
(3)反驳 $\alpha 3 = = (p \rightarrow \bigcirc \neg p) \wedge \Box(\neg p \rightarrow \Diamond(p \wedge \Box p')) \wedge \Box(p \rightarrow \Diamond \neg p')$

解:



(4)反驳 $\alpha 4 = = (p \wedge \Box(p \rightarrow \bigcirc \Diamond p)) \wedge \neg \Box \Diamond p$

解:



5　可靠性定理

定理 1. 上述建立反驳树的方法是语义可靠的,即,如果一有穷公式集 Φ 能够被反驳,则 Φ 不可满足. 特别,如果 $\Phi = \{\neg\varphi\}$,则 φ 是逻辑真的.

证明:我们证明,在构造反驳树的过程中,如果一枝可满足,则应用给出的规则,做了扩展之后,所得新枝(如果有两个新枝,则必有其一)必然是可满足的.

对于命题连接词的规则,这里省略证明. 下面仅考虑部分时态算子规则的情形.

(1)规则□. 设 Ψ 是待扩展分枝上的所有公式所成集合,$\Box\alpha \in \Psi$,Ψ 可满足,因而存在模型$(M,S,s_0\cdots)$使得$(M,S,s_0\cdots) \models \Psi$,当然有$(M,S,s_0\cdots) \models \Box\alpha$. 据定义,对所有 $i \geqslant 0$,$(M,S,s_i\cdots) \models \alpha$,从而,扩展之后的公式 $s_i:\alpha$(对所有 i),在$(M,S,s_0\cdots)$下仍被满足.

值得指出的是:$s_i:\alpha$ 并不是我们语法定义中的合适公式,但是,它的意义是可以通过模型而获得,即 $s_i:\alpha$ 为真,指$(M,S,s_i\cdots)\vdash\alpha$.说 α 为真,实际上指 $s_0:\alpha$ 为真,即$(M,S,s_0\cdots)\vdash\alpha$.

(2)规则$\Box\Diamond$.如果$(M,S,s_0\cdots)\vdash\Box\Diamond\alpha$,则对每个 i,$(M,S,s_i\cdots)\vdash\Diamond\alpha$,从而,对每个 i,存在 $j(i)\geqslant i$,$(M,S,s_{j(i)}\cdots)\vdash\alpha$.不难证明,$\{j(i):i\geqslant 0\}$中有无穷上长序列,从而存在无穷多个 s_i,满足$(M,S,s_i\cdots)\vdash\alpha$.

(3)规则$\Diamond\Box$.如果$(M,S,s_0\cdots)\vdash\Diamond\Box\alpha$,则存在 i,$(M,S,s_i\cdots)\vdash\Box\alpha$,于是,存在 i,对每个 $j\geqslant i$$(M,S,s_j\cdots)\vdash\alpha$.

(4)规则$\Diamond\Box\Diamond$.如果$(M,S,s_0\cdots)\vdash\Diamond\Box\Diamond\alpha$,则存在 i,$(M,S,s_i\cdots)\vdash\Box\Diamond\alpha$,从而,存在 i,对所有 $j\geqslant i$$(M,S,s_j\cdots)\vdash\Diamond\alpha$,从而,存在 i,对所有 $j\geqslant i$,存在 $k(j)\geqslant j$$(M,S,sk(j)\cdots)\vdash\alpha$.如果我们证明了对每个满足 $0\leqslant m<i$ 的 m,都有 $k(m)\geqslant m$ 使得$(M,S,sk(m)\cdots)\vdash\alpha$,则我们必有$(M,S,s_0\cdots)\vdash\Box\Diamond\alpha$.我们取 $k(m)=k(i)$,这里 $0\leqslant m<i$,则 $k(i)\geqslant i>m$,且$(M,S,sk(m)\cdots)\vdash\alpha$.从而得证.

(5)规则 U.如果$(M,S,s_0\cdots)\vdash\alpha U\beta$,则按定义存在 j,$(M,S,s_j\cdots)\vdash\beta$ 且对满足 $0\leqslant i<j$ 的所有 i,$(M,S,s_i\cdots)\vdash\alpha$.

(6)规则$\Box\rightarrow$.如果$(M,S,s_0\cdots)\vdash\Box(\alpha\rightarrow\beta)$,则对每个 i,$(M,S,s_i\cdots)\vdash\alpha\rightarrow\beta$.

我们没有列出所有时态算子规则,因为余下的证明与已有的证明类似.

因为 Φ 的反驳树的构造是从单结点 Φ 不断使用如上规则所得,如果 Φ 是可满足的,则 Φ 必有一可满足分枝,但一封闭树显然是不可满足的.因而,如果 Φ 已被反驳,则它是不可满足的,特别,如果 $\Phi=\{\neg\varphi\}$,则 $\neg\varphi$ 是不可满足的,即 φ 是逻辑真的.

6 反驳方法的完全性定理

定理 2. 如果有穷公式集 Φ 不可反驳,即 Φ 的反驳树存在开分枝,则 Φ 是可满足的.

证明:设 Φ 的反驳树中一开分枝上未被标记的公式集为 Ψ.如果 Ψ 中仍存在公式,可以使用除$\Box\rightarrow$规则之外的某些规则,则先用尽这些规则,所得的公式集仍记为 Ψ(如果扩展之后封闭了,则我们就不必证了).

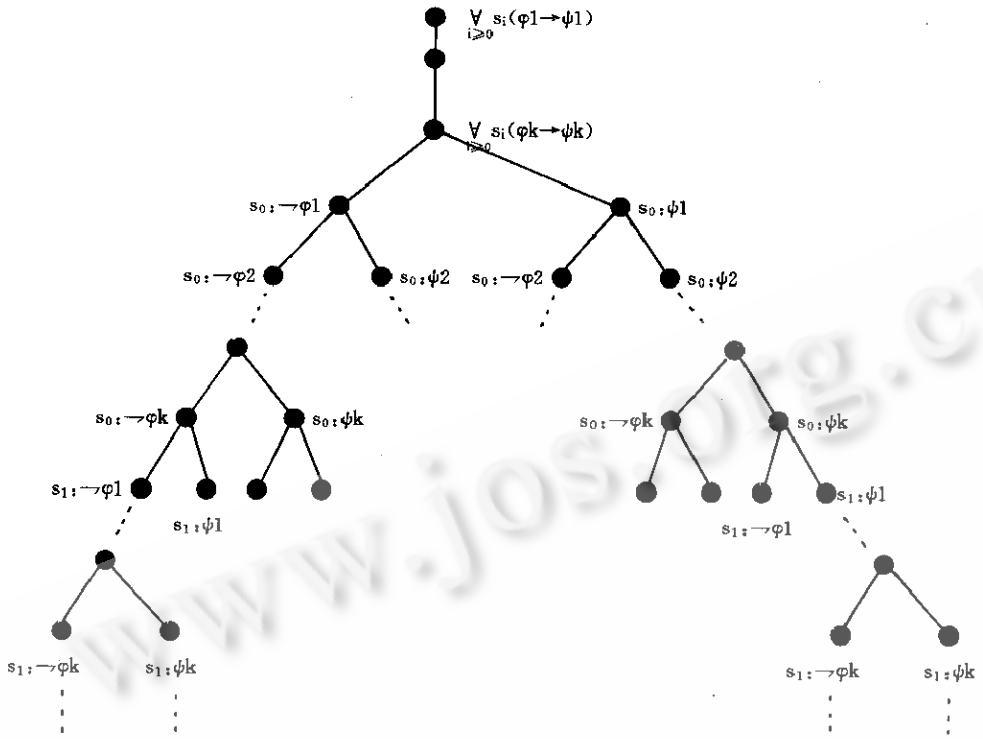
下面考虑使用$\Box\rightarrow$规则的情况.不妨设 Ψ 中有 k0 个形如$\Box(\varphi\rightarrow\psi)$的公式,记为$\Box(\varphi 1\rightarrow\psi 1),\cdots,\Box(\varphi k0\rightarrow\psi k0)$.如果 φi 或 ψi 中嵌套含蕴函式,

① φi 形如 $\alpha\rightarrow\beta$,则$\Box((\alpha\rightarrow\beta)\rightarrow\psi i)$可转化为$\Box(\neg\alpha\rightarrow\psi i)\wedge\Box(\beta\rightarrow\psi i)$.

② ψi 形如 $\alpha\rightarrow\beta$,则$\Box((\varphi i\rightarrow(\alpha\rightarrow\beta))$可转化为$\Box((\varphi i\wedge\alpha)\rightarrow\beta)$,

③ $\alpha\rightarrow\beta$ 做为 φi 或 ψi 的真子公式.则引进新命题符号 q,增加公式$\Box((\alpha\rightarrow\beta)\rightarrow q))$,$\Box(q\rightarrow(\alpha\rightarrow\beta))$.

做了如上变化之后,设有 k 个形如$\Box(\varphi i\rightarrow\psi i)$的公式,使用$\Box\rightarrow$规则,分别得到 $\vee s_i(\varphi 1\rightarrow\psi 1),\cdots,\vee s_i(\varphi k\rightarrow\psi k)$.为了遍历所有可能,我们构造子树如下:



构造的具体方法是,先遍历 s_0状态的所有可能情形,再遍历 s_1状态的所有可能情形…,将该子树的根接到一开始考虑的开分枝末端.

设 Ψ 中出现的原子命题为 p1,…,pn(包括新列入的命题符号). 我们不妨假设,扩展无穷树中仍有开分枝,否则,我们就不必证了. 不妨设此枝为 Ψ1,如果在 Ψ1 中有 s_i:pj,则构造模型使得 pj 在 s_i 状态为真,如果有 s_i:¬pj,则构造模型使用 pj 在 s_i 状态为假,因为该枝不封闭,所以这是可以办到的. 如果在该枝中,s_i:pj 与 s_i:¬pj 均不出现,则任意给出真假定义,这样,可以构造一模型(M,S,s_0…)使得对任何 φ∈Ψ1,(M,S,s_0…)⊨φ 从而证明了 Ψ1 可满足. 因而 Ψ 可满足,进一步 Φ 可满足.

7　结束语

目前,我们正在使用386微机,Turbo PROLOG 语言. 实现此方法. 在我们的方法中,用到有关自然数性质的归纳法. 我们准备用启发式方法来解决有关归纳问题.

作者曾用例子来比较我们的方法与别人的方法的优劣. 因为文献[1,7,9,10]的方法冗长,限于篇幅,这里不予给出.

致谢　王兵山教授仔细阅读了全文,并提出了有益的建议,作者在此表示衷心感谢.

参考文献

1　Abadi M, Manna Z. Noncausal deduction in first-order temporal logic. J. ACM, 1990, **37**(2): 279-317.

2　Abadi M. The power of temporal proofs. Theoretical Computer Science 65, 1989: 35-83.

3 Abadi M, Manna Z. Noncausal temporal deduction. LNCS 193, 1985:1—15.
4 Bell J L, Machover M. A course in mathematical logic. 1977.
5 Cavalli A, Farinas Del Cerro L. A decision method for linear temporal logic. LNCS 170, 1984:113—127.
6 Chang C, Lee R. Symbolic logic and mechanical theorem proving. 1973.
7 Manna Z, Wolper P. Synthesis of communicating processes from temporal logic specifications. ACM Transitions on Programming Languages and Systems, 1984,6(1):68—93.
8 Plaisted D. Mechanical theorem proving. Formal Techniques in Artificial Intelligence, 1990.
9 Venkatesh G. A decision method for temporal logic based on resolution. LNCS 206,1985:272—289.
10 Wolper P. Temporal logic can be more expressive. Information and Control 56, 1983:72—99.
11 侍晖,陈火旺. 时态命题逻辑语义消解算法. 中国计算机学会,第七届年会论文集,1986.

A NEW METHOD FOR THEOREM PROVING OF PTL

Ben Kerong and Chen Huowang

(*Department of Computer Science, Changsha Institute of Technology, Changsha* 410073)

Abstract In this paper, a new method for theorem proving of PTL (propositional temporal logic) based on constructing semantic refutation tree is presented. This method, which is different from the other existed methods all based on decomposing a temporal formula into now—part and next—part, provides a well theory framework for automatic theorem proving of PTL. The soundness and completeness of this method are also proved.

Key words Temporal logic, theorem proving, automated reasoning.