一种命题逻辑的可判定性算法

程 昆,高佳乐
(云南师范大学 信息学院,云南 昆明 650092)

摘 要: 针对命题逻辑的可判定性中真值表法复杂度高的问题,提出了一种基于命题逻辑联结符号完备性和与或树规则的命题逻辑的可判定性算法。算法首先利用常见的等价公式和与或树规则对命题逻辑的公式进行分解,然后参照分解后的树形结构将公式转换成范式形式,最后对照所得的判别式对命题逻辑公式进行判定。理论证明这种算法相比于具有指数级复杂度的真值表法效率高得多。

关键词: 命题逻辑;归纳定义;完备性;与或树;可判定性

中图分类号: TP181　　**文献标识码**: A　　**文章编号**: 1674-7720(2013)24-0076-02

An algorithm of propositional logic′s decision

Cheng Kun, Gao Jiale
(Information School, Yunnan Normal University, Kunming 650092, China)

Abstract: The paper gives a lot of basic knowledge of propositional logic, then proves the completeness of the logic′s join operator. By using the known equivalent formula and AND/OR tree, it translates the formula into normal form and decides the formula of propositional logic. Compared with Truth table, the algorithm largely improves the efficiency in the decision problem.

Key words: propositional logic; inductive definition; completeness; AND/OR tree; decision

所谓命题逻辑的判定问题,即是否存在这样的算法,对于任意命题逻辑的公式,利用这个算法在有限步内判定是否是永真,是否是可满足的,是否是矛盾的[1]。命题逻辑的判定问题是一个已解决的问题。利用真值表法,对于有 n 个命题变元的公式,在不超过 2^n 步内得出命题逻辑的判定问题,指数级的复杂度很难让人们接受。故命题逻辑的判定问题主要是考虑它的效率,在有限的范围内更快地解决这个问题。

1 命题逻辑基础

命题逻辑是数理逻辑的一部分。在命题逻辑中,把简单命题作为基本单位;从简单命题出发,通过使用联结词来构成复合命题。命题逻辑的特征在于,在研究命题的逻辑形式时,只分析复合命题的逻辑形式,把复合命题分析到基本的命题成分即简单命题为止,而不去考虑简单命题自身的成分因素。简单命题被看作是一个整体,它是真的或假的。

1.1 命题逻辑语言£

命题逻辑语言£是命题逻辑使用的形式语言,包含3类符号:第一类为命题符,如 p、q、r 表示任何命题符号;第二类包括5个联结符号:¬、∧、∨、→、↔,它们依次表示非、合取、析取、蕴涵和等值于;第三类包括两个标点符号:“(”和“)”,它们依次称为左括号和右括号。

记 Atom(£)和 Form(£)为命题逻辑语言£的原子公式的集和公式的集,下面用归纳定义法[2]定义命题逻辑中的公式。

定义1 (Atom(£)) £的一个表达式是 Atom(£)的元,当且仅当它是单独的一个命题符号。

定义2 (Form(£)) A∈Form(£),当且仅当它能由(有限次使用)以下条件(1)~(3)生成:

(1)(Atom(£)⊆Form(£);

(2)如果 A∈Form(£),则(¬A)∈Form(£);

(3)如果 A、B∈Form(£),则(A∗B)∈Form(£)。其中符号“∗”表示4个联结符号∧、∨、→、↔中的任意一个。

1.2 常用的逻辑公式和定义

现将本文所需用到的基础逻辑公式和定义罗列如下[3]:设 p、q、r 为原子命题,符号“⇔”的含义为逻辑等值。

(1)分配律:$(p\wedge q)\vee r\Leftrightarrow(p\vee r)\wedge(q\vee r)$

$(p\vee q)\wedge r\Leftrightarrow(p\wedge r)\vee(q\wedge r)$

(2)德·摩根律:$\neg(p\vee q)\Leftrightarrow\neg p\wedge\neg q$

$$\neg(p\wedge q)\Leftrightarrow\neg p\vee\neg q$$

(3)双重否定律：$\neg(\neg p)\Leftrightarrow p$

定义 3 一个命题公式称为合取范式（CNF），当且仅当它具有形式：

$A_1\wedge A_2\wedge\cdots\wedge A_n,(n\geqslant 1)$

其中 $A_1,A_2,\cdots,A_n$ 都是由命题变元或其否定所构成的析取式。例如，$(p\vee q\vee r)\wedge(\neg p\vee q)\wedge\neg q$ 是一个合取范式。

定义 4 一个命题公式称为析取范式（DNF），当且仅当它具有形式：

$A_1\vee A_2\vee\cdots\vee A_n,(n\geqslant 1)$

其中 $A_1,A_2,\cdots,A_n$ 都是由命题变元或其否定所构成的合取式。例如，$\neg p\vee(p\wedge q)\vee(p\wedge\neg q\wedge r)$是一个析取范式。

定义 5 文字 L[4]或者是原子命题 p 或者是原子命题的否定$\neg p$。

2 命题逻辑联结符号的完备性

定义 6 称联结符号的集为完备的[2]，当且仅当任何 $n(n\geqslant 1)$元的联结符号都能由其中的联结符号定义。

定理：$\{\neg,\wedge,\vee\}$是联结符号的完备集。

证明：由完备性的定义知，需用完备集$\{\neg,\wedge,\vee\}$去定义命题逻辑里所有的联结符号才能证明其完备性。很明显，完备集能定义其自身的联结符号$\neg$、$\wedge$、$\vee$。下面证明利用完备集来定义$\rightarrow$、$\leftrightarrow$，设 p 和 q 为原子命题。

$\rightarrow$：$(p\rightarrow q)\Leftrightarrow\neg p\vee q$

$\leftrightarrow$：$(p\leftrightarrow q)\Leftrightarrow(p\rightarrow q)\wedge(q\rightarrow p)\Leftrightarrow(\neg p\vee q)\wedge(\neg q\vee p)$

利用真值表法，很容易判定这两组逻辑等式是等值的。表 1 给出了“$\rightarrow$”定义的真值表。

表 1 “$\rightarrow$”定义的真值表

p	q	$p\rightarrow q$	$\neg p\vee q$
1	1	1	1
1	0	0	0
0	1	1	1
0	0	1	1

3 命题逻辑的判定算法

对于命题逻辑里任意一个公式 $A\in Form(\pounds)$，利用命题逻辑联结符号的完备性以及所给出的逻辑等值公式，判定算法可以通过以下几个步骤把公式 A 转换为它的合取范式或析取范式 A_0：

(1)输入一个命题公式 A；

(2)若 A 为蕴涵形式，如 $A=B\rightarrow C$，则利用完备集对命题公式进行蕴涵释放，这个过程必须是递归的，因为 B 或 C 也可能是蕴涵；得到一个与公式 A 等价的命题公式 A'；

(3)命题公式 A'利用德·摩根律和双重否定律将否定符号“$\neg$”移到各个命题变元之前，得到与 A'等价的命题公式 A''，显然 A''的组成部分都是文字；

(4)用分配律、结合律将命题公式化为合取范式或析取范式形式的公式 A_0，输出的 A_0 与 A 是逻辑等值的。

需要说明的是，此判定算法满足 3 个特性：(1)对于所有输入的命题公式，该算法是可以终止的，即算法具有可终止性；(2)对于每一个输入，调用该算法所得的输出是输入公式的一个等价公式；(3)所有由该算法计算的输出都是合取范式(CNF)或者析取范式(DNF)形式。

为了使以上算法过程更加清晰明了、准确性更高，并基于命题逻辑联结符号完备集$\{\neg,\wedge,\vee\}$的特征，引入人工智能中状态空间搜索中常用的与或树的概念来对公式进行深度分解[5]。这里需要注意联结符号的优先级问题，联结符号$\neg$、$\wedge$、$\vee$、$\rightarrow$、$\leftrightarrow$是按优先级从高到低排列。用与或树对公式进行分解，在其最底层的是命题变元或其否定形式，只需从底往上按照与、或规则组合命题变元和其否定形式，再经过简单的变换处理就能得到公式的合取范式或析取范式。以公式 $A=P\wedge(P\rightarrow Q)\rightarrow Q$ 为例来说明与或树的分解过程，如图 1 所示。

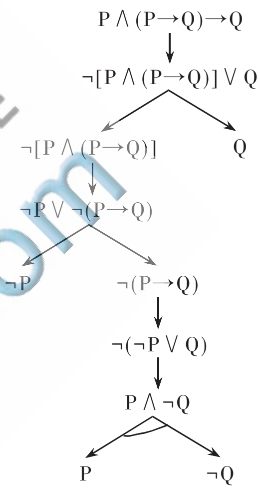

图 1 与或树的分解过程

由底向上按与、或规则组合，公式 $A=[\neg P\vee(P\wedge\neg Q)]\vee Q$ 经变形得 $A=\neg P\vee\neg Q\vee Q$，显然，A 是一个永真式。

假设公式 $A\in Form(\pounds)$为永真，且 A'是它的某个合取范式。根据合取范式的定义，它具有 $A_1'\wedge A_2'\wedge\cdots\wedge A_n',(n\geqslant 1)$，其中 A_i'是由文字所构成的析取式。公式 A 为永真，当且仅当它的每一个因子为永真，又 A_i'为永真当且仅当在其中必存在互补对（p 和$\neg p$）。于是得到关于永真性判定问题的判别式：命题逻辑的公式为永真，当且仅当它的合取范式中的每一个因子至少含有一个互补对。

同理可以得到，命题逻辑公式是矛盾的当且仅当它的析取范式中的每一个因子至少含有一个互补对。

由于算法只涉及到对公式进行等值代换，其主要的时间消耗是在命题公式的范式求解过程中，故其复杂度大致为与或树的深度 h。假设在公式 A 中出现原子命题的数目为 n，最理想的状态是利用与或树规则分解完后形成一个满二叉树，最后一层为 n 个原子命题或者原子命题的否定，这样得到二叉树的层数为$\lfloor \lg n\rfloor+1$，算法的复杂度下界大致为$\lfloor \lg n\rfloor+1$；最坏的情况为，公式 A 中出现的是 n 个均不同的原子命题，这样只能通过真值表法去判定其为永真式或矛盾式，其算法复杂度为 2^n。显然，相比于真值表法的指数级复杂度，该算法的复杂度有所降低，提高了在判定过程中的效率。

命题逻辑的判定问题是已解决的问题，但是本文对命题逻辑公式先进行等值替换构造互补对来进行判定的算法思路，对设计命题逻辑公式判定的推理机提供了

理论基础[6]，同时对一阶谓词逻辑片段(如二元一阶谓词逻辑 FO^2)的可判定性证明有启发意义。

参考文献

[1] 张会凌.命题逻辑判定系统中基本真值矩阵的生成算法[J].甘肃联合大学学报，2005，19(1)：16-19.

[2] 陆钟万.面向计算机科学的数理逻辑[M].北京：科学出版社，2002.

[3] 屈婉玲，耿素云.离散数学[M].北京：高等教育出版社，2008.

[4] HUTH M，RYAN M.面向计算机科学的数理逻辑系统建模与推理[M].何伟，樊磊，译.北京：机械工业出版社，2007.

[5] NILSSON N J.人工智能[M].郑扣根，译.北京：机械工业出版社，2007.

[6] 郭远华.启发式方法生成命题逻辑可读证明[J].计算机应用研究，2011，28(12)：4429-4432.

(收稿日期：2013-08-22)

作者简介：

程昆，男，1988 年生，硕士研究生，主要研究方向：描述逻辑。

高佳乐，女，1989 年生，硕士研究生，主要研究方向：现代教育技术。

《电子技术应用》 www.ChinaAET.com