Computer Knowledge and Technology 电脑知识与技术
Vol.6,No.7,March 2010, pp.1657-1659
http://www.dnzs.net.cn
Tel:+86-551-5690963 5690964

类型一阶逻辑的理论证明探讨

徐 丹

(厦门大学 计算机科学系,福建 厦门 361005)

摘要:类型一阶逻辑在传统的一阶逻辑上,引入了类型,它是多态多类逻辑程序设计语言的理论基础,对编译系统设计与实现的进一步发展具有重要意义。论文在类型一阶逻辑的理论层面进行了探讨,引入了基本简单导出的可靠性定理和等值符号的可替换性定理,并予以证明。通过这两个定理,可以简化类型一阶逻辑理论证明中的工作量,使得将来的理论研究更加方便。

关键词:一阶类型逻辑;B 类型简单导出;原子简单导出;可靠性;可替换性定理

中图分类号:TP311 **文献标识码**:A **文章编号**:1009-3044(2010)07-1657-03

Theoretical Proof of Typed First-order Logic

XU Dan

(Department of Computer Science of Xiamen University, Xiamen 361005, China)

Abstract: Type is introduced into the typed first-order logic, which is based on the traditional first-order logic and the theoretical foundation of polymorphic and polytypic logic programming language. Typed first-order is significant to the further development of the design and implementation of compilation system. This paper is a research at the theoretical level and it introduces reliability theorem of the basic and simple derivation and the exchangeability theorem of the equivalent symbols, and the two theorems are proved in this paper. With the two theorems, the workload of the first-order logic proof can be simplified and the future theoretical research will be easier.

Key words: typed first-order logic; B-typed simple derivation; atomic simple derivation; reliability; exchangeability theorem

自 Prolog 语言[1]诞生以来,逻辑程序设计成为一个新的方向。1970 年代后期和 1980 年代初,在为 Prolog 语言建立了 Horn 子句理论之后,如何扩展 Prolog,将无类型的逻辑程序设计语言改造成带类型的逻辑程序设计语言,同时建立新型逻辑程序设计语言的理论基础,扩展逻辑程序设计的应用范围,就成为一个重要的研究课题。1990 年代中期,J. W. Floyd 等人在系统总结前人工作的基础上,借鉴 Prolog 语言的诸多元素,并从 Prolog 语言的众多变形如 IC-Prolog、NU-Prolog 等以及 ML 语言、Modula-2 语言中吸收新思想,推出了新型逻辑程序设计语言 Gödel[2-4],试图改进 Prolog 语言中存在的不足并解决其中有争议的语义问题。作为一种逻辑程序设计语言,研究与开发 Gödel,首先需要为其建立严格的数学基础,包括其语法和语义理论基础。从 Prolog 语言和 Gödel 语言的关系看,与传统的一阶逻辑[5-8]比较,Gödel 的数学基础应该是多态多类的一阶逻辑,而且,Gödel 语言的语法与多态多类一阶逻辑的类型合式公式之间有着密切的联系,Gödel 语言的语义应该与多态多类一阶逻辑的模型之间有着密切的联系。于是,发展带类型的一阶逻辑理论(简称类型一阶逻辑)成为一种选择。

1 带类型的一阶逻辑的语法

本文继续文献[2-3]的工作。带类型的一阶逻辑来源于对 Gödel 语言的抽象,也来源于对一阶逻辑的扩展。我们约定,一阶类型逻辑中的基类型有:Integer,Real,String,Boolean,分别称为整型、实型、字符串型和布尔型。一阶类型逻辑语法和语义部分的内容见参考文献[2-3]。

2 简单导出的可靠性定理

为了定理说明的方便,以下引入几个定义。

定义 2.1:记所有在解释下与基类型 Boolean 保持一致的类型符为 B。

定义 2.2:将无类型一阶命题逻辑 P* 中的某个合式公式做替换,使得原先的所有命题词替换为一阶类型逻辑中的某个布尔值类型的形式符号,由此得到一个类型合式公式的过程称为简单导出。特别的,如果替换的一阶类型逻辑中的某个布尔值类型的形式符号中的符号全部是类型为 B 的个体词(或者原子),则将此过程称为 B 类型简单导出(或者原子简单导出);如果命题词全部替换,每个命题词都只能使用 B 类型个体词和原子两者之一做替换,则称为基本简单导出。

定理 2.1 B 类型简单导出的可靠性定理

如果在无类型一阶命题逻辑 P* 中,有合式公式 A 是恒真的,则 A 经过 B 类型简单导出,得到的一阶类型逻辑下的合式公式也是恒真的。

原来的无类型逻辑中的命题词在某个赋值之下只有真假两种情况,和我们替换的个体词刚好一致,直观上看,这个公式是成立的。对此,我们做证明如下:

[数学归纳法]

收稿日期:2010-01-08

作者简介:徐丹(1984-),女,贵州安顺人,厦门大学计算机科学系学生,硕士,主要研究方向为带类型的逻辑程序设计语言。

设 A 中所有的命题词的集合为 N={p1,p2,…,pn},生成 C 的替换函数(或者置换)为 g={<p1,d1>,<p2,d2>,…,<pn,dn>}(或者为 σ={d1/p1,d2/p2,…,dn/pn}),其中的 di 是一阶类型逻辑中的某个类型为 B 的个体词。记 A 中命题词替换为某个个体词生成的一阶类型逻辑中的合式公式为 g(A)=C(或者 Aσ=C)。

任取一阶类型逻辑中的赋值 φ,构造无类型一阶命题逻辑下的赋值 φ,使得

φ(pi)=t 当且仅当 φ(g(pi))=φ(ej)=<true,Boolean>

φ(pi)=f 当且仅当 φ(g(pi))=φ(ej)=<false,Boolean>,其中 ej 是 pi 替换的目标。

显然,A 和 C 的层数是相等的。由于 A 是恒真的,所以一定有 φ(A)=t。施归纳于它们的层数 t,需要证明的命题是 φ(A)=t 时,φ(C)=<true,Boolean>(为了证明的方便,以下同时证明 φ(A)=f 时,φ(C) =<false,Boolean>)。

基始:t=0 时,A 是一个命题词,C 是一个个体词。根据 φ 的定义,若 φ(A)=t 则必有 φ(C) =<true,Boolean>,若 φ(A)=f 则必有 φ(C) =<false,Boolean>。

归纳:t=k(k>0)时。由于 A 来此于无类型一阶命题逻辑 P*,故 A 和 C 只可能有以下五种形式(为了将无类型一阶命题逻辑中的逻辑连接词与一阶类型逻辑中的逻辑连接词区分开来,在类型一阶命题逻辑的逻辑连接词上添加下划线)。

1) A=$\underline{\neg}$ A1,C= ¬ C1,其中 C1=g(A1)。若 φ(A)=t,则 φ(A1)=f,依据基始,有 φ(C1) =<false,Boolean>。据 ¬ 的定义,有 φ(C) =<true,Boolean>。同理,φ(A)=f 时,有 φ(C) =< false,Boolean>。

2) A=A1 $\underline{\wedge}$ A2,C=C1 ∧ C2,其中 C1=g(A1),C2=g(A2)。若 φ(A)=t,则 φ(A1)=t 且 φ(A2)=t,依据基始,有 φ(C1) =<true,Boolean>且 φ(C2) =<true,Boolean>。据 ∧ 的定义,有 φ(C) =<true,Boolean>。若 φ(A)=f,则 φ(A1)=f 或 φ(A2)=f。不妨设 φ(A1)=f,依据基始,有 φ(C1) =<false,Boolean>。据 ∧ 的定义,有 φ(C) =<false,Boolean>。

3) A=A1 $\underline{\vee}$ A2,C=C1?C2,其中 C1=g(A1),C2=g(A2)。若 φ(A)=t,则 φ(A1)=t 或者 φ(A2)=t,不妨设 φ(A1)=t,则依据基始,有 φ(C1) =<true,Boolean>。据 ∨ 的定义,有 φ(C) =<true,Boolean>。若 φ(A)=f,则 φ(A1)=f 且 φ(A2)=f,依据基始,有 φ(C1) =<false,Boolean>且 φ(C2) =<false,Boolean>。据 ∨ 的定义,有 φ(C) =<false,Boolean>。

4) A=A1 $\underline{\rightarrow}$ A2,C=C1→C2,其中 C1=g(A1),C2=g(A2)。若 φ(A)=t,则 φ(A1)=f 或者 φ(A2)=t。当 φ(A1)=f 时,依据基始,有 φ(C1) =<false,Boolean>,据→的定义,有 φ(C) =<true,Boolean>;当 φ(A2)=t 时,依据基始,有 φ(C2) =<true,Boolean>,据→的定义,有 φ(C) =<true,Boolean>。若 φ(A)=f,则 φ(A1)=t 且 φ(A2)=f,依据基始,有 φ(C1) =<true,Boolean>且 φ(A2)= <false,Boolean>,据→的定义,有 φ(C) =<false,Boolean>。

5) A=A1 $\underline{\leftrightarrow}$ A2,C=C1↔C2,其中 C1=g(A1),C2=g(A2)。若 φ(A)=t,则 φ(A1→A2)=t 且 φ(A2→A1)=t。依据第 4 步的结论,有 φ(C1→C2)=<true,Boolean>且 φ(C2→C1)=<true,Boolean>。据↔的定义,有 φ(C) =<true,Boolean>。若 φ(A)=f,则 φ(A1 $\underline{\leftrightarrow}$ A2)=f 或者 φ(A2 $\underline{\leftrightarrow}$ A1)=f。此处不妨设 φ(A1 $\underline{\leftrightarrow}$ A2)=f,则依据第 4 步的结论,有 φ(C1→C2)=<false,Boolean>。据↔的定义,有 φ(C) =<false,Boolean>。

综上所述,φ(A)=t 时,φ(C) =<true,Boolean>,φ(A)=f 时,φ(C) =<false,Boolean>。

由于 A 是恒真的,所以一定有 φ(A)=t。又因为 φ 是任取的一阶类型逻辑上的赋值,所以对于任意的一阶类型逻辑上的赋值?,都能通过构造无类型一阶逻辑上的赋值 φ 来证明 φ(C) =<true,Boolean>,所以 C 是恒真的。

定理 2.2 原子简单导出的可靠性定理

如果在无类型一阶命题逻辑 P* 中,有合式公式 A 是恒真的,则将 A 中所有的命题词一一替换为一阶类型逻辑中的某个原子,则新得到的一阶类型逻辑下的合式公式也是恒真的。

要证明这个定理,只要按照定理 2.1 的证明思路,构造赋值即可,此处不再证明,仅提供一个例子作为验证。

我们知道德·摩尔根律在 P* 下一定是成立的,由之,我们有以下恒真合式公式:

H1= ¬ (A∧B)↔ ¬ A∨ ¬ B 和 H2= ¬ (A∨B)↔ ¬ A∧ ¬ B

构造替换函数 g1={<A,C1>,<B,C2>},g2={<A,D>,<B,D>},其中,C1、C2 和 D 都是一阶类型逻辑中的原子,则得到一阶类型逻辑中的 g1(H1)= ¬ (C1∧C2)↔ ¬ C1∨ ¬ C2,g2(H2)= ¬ (D∧D)↔ ¬ D∨ ¬ D。不论是哪一种赋值,C1、C2 和 D 都只有<true,Boolean>或者<false,Boolean>两种值,所以以下我们分别对各种情况作讨论(假设赋值为 φ):

对于 g1(H1),

1) φ(C1)=<true,Boolean>,φ(C2)=<true,Boolean>,则 φ(¬ (C1∧C2))=<false,Boolean>,φ(¬ C1∨¬ C2)=<false,Boolean>,故 φ(g1(H1)) = <true,Boolean>。

2) φ(C1)=<true,Boolean>,φ(C2)=<false,Boolean>,则 φ(¬ (C1∧C2))=<true,Boolean>,φ(¬ C1∨¬ C2)=<true,Boolean>,故 φ(g1(H1)) = <true,Boolean>。

3) φ(C1)=<false,Boolean>,φ(C2)=<true,Boolean>,同 2 理,有 φ(g1(H1))= <true,Boolean>。

4) φ(C1)=<false,Boolean>,φ(C2)=<false,Boolean>,则 φ(¬ (C1∧C2))=<true,Boolean>,φ(¬ C1∨¬ C2)=<true,Boolean>,故 φ(g1(H1)) = <true,Boolean>。

因此 g1(H1)必是恒真合式公式。

对于 g2(H2),它的情况比 g1(H1)简单,显然也是恒真的。

定理 2.3 基本简单导出的可靠性定理

如果在无类型一阶命题逻辑 P* 中,有合式公式 A 是恒真的,则 A 经过基本简单导出,得到的一阶类型逻辑下的合式公式也是恒真的。证明方法同定理 2.1 及定理 2.2。

3 等值符号的可替换性定理

再次引入定义:

定义 3.1:所有由基本简单导出得到的类型合式公式的集合称为基本简单导出集合。

等值符号的可替换性定理:设 A、B、C 是基本简单导出集合中的类型合式公式,Γ 是基本简单导出集合的子集,如果 B 和 C 等值(即 B↔C),由 Γ 和 A 把 B 在其中的某些出现替换为 C 而得到 Γ′ 和 A′。那么

[1] A↔ A′

[2] (Γ→A) ↔(Γ′→A′)

分析:我们可以把式子[1]描述为 M=(B→C)↔(A→A′),于是只要证明(B→C)↔(A→A′)为恒真,原命题则得证。同理对于[2]式,只需证明 N=(B↔C)→((Γ→A)↔(Γ′→A′))为恒真即可。

证明:由于 A、B、C 是基本简单导出集合中的类型合式公式,Γ 是基本简单导出集合的子集,故一定存在无类型一阶命题逻辑 P* 中的合式公式 A0、B0、C0 和合式公式集合 Γ0,使得 A、B、C 分别由 A0、B0、C0 基本简单导出,Γ 中的所有合式公式都是由 Γ0 中的某一个基本简单导出。

令 GA(X)表示将 A 中某些 B 的出现替换为 X,使得 B 的一个出现被替换当且仅当在构造 A′ 时,该位置的 B 被替换。则 A=GA(B),A′=GA(C)。显然一定有无类型一阶命题逻辑 P* 中的公式 GA0(),使得 GA()是由 GA0()将其中的命题词替换为带类型一阶逻辑中的 B 类型个体词和原子得到的。于是对 GA0(C0)做基本简单导出可以得到 GA(C)= A′。记 GA0(C0)= A0′,即证明 A′ 可以由导出 A 的 A0 替换部分 B0 的出现(即 A0′)再做基本简单导出得到。同理可证,有 Γ0′ 使得对 Γ0′ 中的所有合式公式做基本简单导出得到的集合是 Γ′。

构造无类型一阶命题逻辑 P* 中的合适公式 M0=(B0↔C0)→(A0↔A0′),N0=(B0↔C0)→((Γ0→A0) /(Γ0′→A0′)),由 A、B、C、Γ 和 A0、B0、C0、Γ0 之间的被导出和导出的关系,显然可知,M0 和 N0 做基本简单导出可以相应得到 M 和 N。

已知在无类型一阶命题逻辑 P* 中,有等值公式的可替换性定理,则任意的 M0 和 N0 形式的公式都是恒真的。又因为 M0 和 N0 是由 M 和 N 分别作基本简单导出得到,根据基本简单导出的可靠性定理,M 和 N 为恒真公式,于是定理得证。

4 结论

本文探讨了带类型一阶逻辑的理论证明方法,通过证明基本简单导出的可靠性定理,方便了带类型一阶逻辑的理论证明。比如,我们可以利用该可靠性定理,直接证明($\rightarrow_+$),如果 $\Gamma, <A,\tau\rightarrow\sigma> \vdash <B,\sigma_1\rightarrow\sigma>$则 $\Gamma \vdash <A,\tau\rightarrow\sigma> \rightarrow <B,\sigma\rightarrow\sigma>$。

另外,等值符号的可替换性定理的证明,也一定程度上减轻了证明过程中的工作量。但是,基本简单导出的可靠性定理和等值符号的可替换性定理的应用范围只局限在基本简单导出集合之内。我们之后的任务是通过更多的理论探讨,拓展这两个定理的应用范围,从而更好的简化理论证明,构造完备的带类型一阶逻辑理论体系。

参考文献:

[1] 刘椿年,曹德和.Prolog 语言、它的应用与实现[M].北京:科学出版社,1990.

[2] Hill P M, Lloyd J W. The Godel programming Language [M]. London:MIT Press,1994.

[3] 高伟.逻辑程序设计语言 Godel 的说明性语义[D].厦门:厦门大学计算机科学系,2009.

[4] 昌杰.逻辑程序设计语言 Godel 的过程性语义[D].厦门:厦门大学计算机科学系,2009.

[5] 胡世华,陆钟万.数理逻辑基础(上,下)[M].北京:科学出版社,1980.

[6] 耿素云,屈婉云,王捍贫.离散数学教程[M].北京:北京大学出版社,2003.

[7] 陆钟万.面向计算机的数理逻辑[M].北京:科学出版社, 1998.

[8] 孙明湘.数理逻辑[M].长沙:中南大学出版社,2004.