归纳证明 自动定理证明 算法 计算机 (73)

计算机科学1998,Vo.25№.5

59-61

关于归纳证明的探讨

Studies in Proof by Induction

钟发荣 孙永强 TP301.6

(浙江师范大学计算机系 金华321004)(上海交通大学计算机系 上海200030)

摘 要 Automatic proofs in theories is a major field of study in computer science and artificial intelligence, and proof by induction is one of the most important and promising automatic proofs in theories. The paper introduces and studies some of the approaches and methods for proof by induction in theories.

关键词 Theorem proving, Induction, Rewriting, Proof by consistency

归纳定理证明是一种难度较大且较有前途的一种自动定理证明方法。较早的归纳定理证明器是Boyer-Moore于1979年提出的BM证明器,BM证明器采用的是传统的结构归纳法,对所证公式中的每个函数项都根据其所谓归纳模板提出相应的归纳方案,然后对这个归纳方案进行分析、比较和整理,得出整个公式适用的归纳方案并根据此方案实施归纳证明。这种方法已能证明一些较困难的定理,如唯一质因子分解定理、快速串搜索算法的正确性和简单优化的编译程序的正确性等。但这种方法有显式的归纳步骤,且不能自动地得出归纳方案,因此不能完全自动化。

基于重写技术的归纳证明方法不同于传统的数学归纳法或结构归纳法,是一种没有显式归纳步骤的隐式归纳法,由于这种方法形式化程度高,很适合于自动实现。基于重写技术的归纳证明方法的基本思想,可形式化为:假设$\sum=(S,F)$是一个标记,其中S是类名的集合,F为算子集合,E为等式集合,X为变元集合,$T(\sum,X)$为项集合,$T(\sum)$为基项(不含变元的项)集合。对$\forall s,t\in T(\sum,X)$,$s=t$称为E的归纳定理,当且仅当对$\forall$代换σ,满足对$\forall x\in X,\sigma(x)\in T(\sum)$,$E\vdash\sigma(s)=\sigma(t)$,即$I(E)\models\sigma(s)=\sigma(t)$。利用Knuth-Bendix完备算法把E变换为收敛的项重写系统(TRS)R,特证命题$s=t$,其中$s,t\in T(\sum,X)$,加入R,再利用Knuth-Bendix完备算法处理,构造了一个收敛TRS而不导致不一致性,即证明了$s=t$是E的归纳定理;如果导致不一致性,即证明了$s=t$不是E的归纳定理。

基于重写技术的归纳证明方法简洁,效率高,搜索准确,应用前景广,它不但可应用于许多难度很高的定理的证明,在人工智能,程序验证和变换,优化的编译程序的正确性证明,及软件工程中,也日益显示出其优越性。

1 证明抽象数据类型的归纳性质

Musser于1980年在文[2]给出了用Knuth-Bendix完备算法实现归纳证明的方法,其基本思想是,等式在初始代数中有效当且仅当把它加入等式公理组不会导致不一致性。具体做法是,E含完备的等式谓词公理,当完备过程产生等式true=false时,就检测出不一致性,即证明了待证等式不是归纳定理。否则,如果未导致不一致性,且完备算法终止,即证明了待证等式为归纳定理。

2 构造子等式理论中的归纳证明

Huet和Hullot于1982年在文[3]用定义原则这一关键性概念,把算子集F划分为二个不变的集合;构造子集C和已定义符号集D且$F=C\cup D$,使得:(1)T(C)中的项是不可归约的;(2)T(F)—T(C)中的项是可归约的,这里T(C)是由构造子构成的项的集合,T(F)—T(C)是含已定义符号的基项的

钟发荣 讲师,研究方向:重写系统、定理证明。孙永强 教授,博士导师,研究方向:计算理论、新型语言。

集合。由此，当待证等式加入重写系统，不能导出C1=C2，其中C1，C2∈T(C)且C1≠C2时，待证等式是归纳定理。

这一方法是基于Knuth-Bendix完备算法的直接修改，无需等式公理化，且易于实现。用这一方法，证明简洁。实验性证据表明，它足以进行诸如整数，链表和树这些数据类型上的简单原始递归程序计算的证明和代数数据类型实现的正确性的证明，也能自动产生标准求和等式的证明。但它有许多局限性。对复杂引理的组合，尽管递归定义十分自然，但也许不可能满足有限终止这一要求。等式公理的排列影响证明过程，算法不一定终止。

3 不完备规范的归纳推导

Kapur和Musser通过建立理论强完备性和归纳完备的基本联系，即无独立的一致公理及一等式成立当且仅当其所有基例成立之间的基本联系，研究了无归纳的归纳的一般形式。它的一些结论，都是在假定无二义性前提下得到的。在二义理论中，用此方法可区别一致性等式和不一致等式，但却无法区别归纳定理和独立公理。

同时二人在文[4]提出了解决这一问题的方法。把不完备系统变换为无二义强完备系统，不完备系统中的定理在无二义强完备系统中仍然是定理，而独立公理经变换后导致不一致性。该方法是一致性证明的扩展，但构造极其复杂。

4 无构造子理论中的自动归纳证明

Jouannaud和Kounalis研究发现，构造子不是必须的，但可用于加速归纳完备过程。他们通过引入归纳可归约性和归纳上可归约性这二个基本概念，给出了关于归纳的主要性质，在探讨了基于归纳可约性和归纳上可归约性的归纳完备性和归纳可归约性与充分完备性的关系的基础上，提出了计算Church-Rosser规范构造子集的算法和改进的归纳完备算法。还给出了如何检查一致扩展的方法，及归纳上可归约性测试化归为归纳可归约性测试的过程，并证明了左线性TRS(包括结合交换重写)的归纳可归约性的可判定性，但还没解决一般情况下的结合交换重写，所以这有待进一步的研究。

下面是归纳可归约性的一个例子。给定整数模2的类Peano规范S(S(0))→0，这里算子是S(后续子)和0(零)，则S(S(X))是归纳可归约的，而S(X)不是归纳可归约的；因为S(0)是S(X)的一个不可归约的实例，S(S(X))的任一基例都含子项S(S(0))，因此是可归约的。

这一方法的能力虽比文[3]强得多，但也有它内在的限制。必须用完备算法，这意味着，要处理终止规则集。尽管这对数据类型的递归定义是很自然的，但在某些情况下，却是难以满足的。另外要通过引入适当的元规则来解决可能的完备发散。本方法提出的测试归纳可归约性的算法效率不高，测试集数目过大，导致本方法效率不高。

5 规范变换归约证明

虽然归纳可归约性是可判定的，但已有的算法都复杂度高且不实用。通过规范变换，把文[5]处理的一般情况化归到文[3]的方法，避免了归纳可归约性测试。对于给定的TRS R，Comon给出了如何计算R不可归约基项集NF的条件语法的方法，还给出了条件文法清理算法，它为判定一般情况下的归纳可归约性提供了手段，也适于计算有序类规范。同时添加了某些新构造子，以保持变换前后的初始代数同构且保持构造子间的某些约束关系，给出了如何在结果有序类代数中完成归纳证明的方法。

但规范变换需要许多前提条件，即要求TRS是基收敛的和NF清理文法是正则的。这导致了对可归约基项语言的正则性检查，这种检查较为复杂。

6 计算基归纳可归约性和归纳完备位置

Bundgen和Kuchlin的测试是文[5]中JK-测试的扩展和改进，适于左线性TRS。如果TRS是左线性的，规则左部仅由构造子和变元构成，该方法总能测试归纳可归约性，但不能测试某些扩展的归纳可归约性。文[5]给出了具有如下性质的有限测试代换集S：对$\forall t\in T(\sum,X)$是归纳可归约的当且仅当对$\forall \sigma\in S$，$\sigma(t)$是可归约的。JK-测试伴随着不可归约基项的top项测试集$GR_R(d)$，其中$d(R)=\max\{depth(r)|r\in R\}$。由于$GR_R(d)$随d按指数增长，即使对那些极易测试的项，也要花费大量的计算。文[7]改进了文[5]的低效率测试。如果F可划分为构造子集C和已定义符号集D，则$d=d(R')$这里$R'\subseteq R$中左部属于T(C,X)(仅由构造子和变元构成的项集)的测试集。由于$GR_R(d)$是随d指数增长的，这样求出的d极大地降低了$GR_R(d)$数目。如果C是自由的，则d=0，因此在构造$GR_R(d)$时含D中符号的项不用考虑。文[7]的核心是给出了一个基于递归问题分解的并行覆盖问题算法和动态规划。这一算

法把项序列覆盖问题化归为项集合覆盖问题。然后，为测试扩展的归纳可归约性，把它化归为覆盖测试：项 t 在某处归纳可归约的依赖于与 t 在该处重叠的那些规则，

7 等式理论中的一致性证明

'Deshowitz 在文[8]提出等式 s=t 在由 R 定义的初始模型中有效当且仅当从 R 不能导出二个互异的不可归约基项且 s=t。这样，对任 l→r，一旦存在基例 σ(l)是不可归约的，就意味着不一致性。这一归纳可归约性的概念是文[5]中归纳完备算法的基础。归纳完备过程主要问题之一是，它要求为它提供用于把等式定向为重写规则的项上的序。对给定的序，一旦不能对产生的等式定向，归纳完备过程就失败。还有为构造收敛重写系统，归纳完备过程经常要演译大量的无用等式。文[9]提及这一问题，并给出了极大地限制演绎等式的数目的线性证明方法。

文[10]提出了一种概括了 Fribourg 线性方法的一致性证明方法，它不仅能处理重写规则，也能处理不可定向等式。这一方法适用于与给定重写系统相容的项上的任意序。在一定意义上，它是反驳完备的，它反驳不是归纳定理的任何等式。与归纳完备过程相比，这一方法不会失败。这一方法是以某种途径把三个基本概念：重叠、归纳可归约性、覆盖集结合起来产生的。等式推导规则和证明简化的框架在概念上是很简单的，且很适于研究进一步的扩展。

讨论 在重写系统的归纳证明研究过程中，都是用不导致不一致性来证明归纳定理的。其一致性也可表述为：R 和 R∪{s=t}都定义了同一个初始代数。文[3]虽概念清晰且效率高，但它要求构造子是自由的。文[2]方法简单，却要求含等式谓词公理、且效率不高。文[5]虽极大地放宽了文[3]的限制，但由于测试归纳可归约性费时，效率不高，而且以 Knuth-Bendix 完备算法为基础，对一般情况的归纳可归约性测试，还没有有效的算法。文[6]虽可把等式理论中的归纳证明问题化归为文[3]的一致性证明，却以引入新构造子为代价，且要求更多的前提条件。文[7]也只是在左线性 TRS 中改进了归纳可归约性测试。我们参照文[6]的方法用 C 语言构造了一个定理证明器，并在算法上对可归约性测试作了改进，并结合文[11]，推广到条件等式理论中，取得了较为满意的结果。由于归纳证明还处于初级阶段，从目前情况看，归纳证明还有许多工作要做，如何放宽限制条件，最终达到能进行一般的归纳证明，提高证明效率，其中主要是测试效率，还有待进一步深入的研究。

参 考 文 献

[1] R. Boyer and J. M. Moore, A Computational Logic, Academic Press, 1979

[2] D. R. Musser, On Proving Inductive Properties of Abstract Data Types, Proc. 7th ACM Symp. On POPL, 1980

[3] G. Huet and J. M. Hullot, Proofs by Induction in Equational Theories with Constructors, J. of Computer and System Sciences 25(2)1982

[4] D. Kapur and D. R. Musser, Inductive Reasoning with Incomplete Specifications, 1st IEEE Symp. Logic in Computer Science, 1986

[5] J. P. Jouannaud and Kounalis, Automatic Proofs by Induction in Theories without Constructors, Information and Computation 82, 1989

[6] H. Comon, Inductive Proofs by Specification Treansformation, LNCS 355, 1989

[7] R. Bundgen and W. Kuchlin, Computing Ground Reducibility and Inductively Complete Positions, LNCS 355, 1989

[8] N. Dershowitz, Applications of the Knuth-Bendix Completion Procedure, Proc. of the Seminaire d'Informatique, 1982

[9] L. Fribourg, A Strong Restriction of the Inductive Completion Procedure, LNCS 226, 1986

[10] L. Bachmair, Proof by Consistency in Equational Theories, Proc. 3rd IEEE Symp. Logic in Computer Science, 1988

[11] E. Bevers and J. Lewi, Proof by Consistency in Conditional Equational Theories, LNCS 516, 1990