归纳程序综合系统 NDIPS 的设计*

徐家福 戴 敏 王志坚

(南京大学计算机软件研究所)

ON THE DESIGN OF AN INDUCTIVE PROGRAM SYNTHESIZER NDIPS

Xu Jiafu, Dai Min, and Wang Zhijian

(*Institute of Computer Software, Nanjing University*)

ABSTRACT

This paper discusses the approaches and problems of inductive program synthesis in software automation. The design background and implementation methods of an inductive program synthesizer NDIPS are proposed and the key techniques of automatic program induction adopted in the system are emphasized.

摘 要

本文讨论了软件自动化归纳途径的现状、症结和解决方法，介绍了归纳程序综合系统 NDIPS 的设计思想和实现技术，强调了系统中自动归纳程序的关键技术.

§1. 软件自动化的归纳途径

归纳途径一般指用实例作为问题的部分描述，通过归纳推理得到问题的完整描述. 它是实现软件自动化的四种主要途径之一[1]. 归纳推理的一般形式是:

$$(\bigwedge_{i=1}^{n} P(x_i)) \models (\forall x P(x))$$

其含义是: 从有限个对象具有某种性质可以归纳出同类对象都具有这种性质. 符号"$\models$"表示归纳推理过程. 与演绎推理不同，归纳推理不是绝对保真的，即使前提为真，也只能得

* 1989年7月24日收到.

© 中国科学院软件研究所 http://www.jos.org.cn

到或然真的结论．在归纳过程中，我们通常无法穷尽所有的个体对象，因此，任何一般归纳过程都是不完备的．症结在于：无法用纯逻辑的方法解释如何从个别过渡到一般．有人把这种过程称作认识上的“飞跃”，这一说法形象地表明这种过渡含有非逻辑的因素．如何解释这种“飞跃”，仍是哲学认识论中悬而未决的问题．

归纳推理一直是计算机科学的重要研究领域．根据其研究的不同侧重点分为两个分支：一种是着重研究归纳推理方法的一般理论性质，如收敛性、计算效率等；另一种则根据人类认知模型发展具体、实用的归纳推理方法．计算机归纳主要用于进行程序综合和语法推理．

归纳程序综合问题可以抽象地用四元组〈P，S，M，C〉表示，其中 P 是程序假设空间，S 是实例集合，M 是推理方法，C 是成功标准．归纳程序综合的任务是：在 S 的约束下，用 M 在 P 中找出满足 C 的某个假设．P，S，M 和 C 均为归纳推理的重要研究内容．

已有的 LISP 归纳程序综合系统以 Summers 的基础性工作 THESYS 系统为代表[2]．该系统利用递归模式和匹配技术，从给它的 I／O 实例中找出其间的递归关系，从而综合出程序．用 LISP 作为其目标语言的主要原因是：用 LISP 所刻划的函数是可构造的，通过将 I／O 实例表示成用 LISP 基本函数(cons, car, cdr 等)刻划的形如 S—表达式的关系式，易从结构的递归性发现程序的递归性质．后继工作大都基于 Summers 的方法[3,4,5,6]，多数集中在设计发现递归关系的专用算法．这类技术的不足是：

(1)仅使用结构匹配技术，没有考虑程序语义．当输入输出之间差异较小时，结构匹配是一种行之有效的方法；但当程序规模较大时，其输入输出之间差异也较大，仅使用结构匹配技术难以识别其内在的关系．

(2)仅使用基本函数构造程序，不支持自底向上的程序综合．简单的基本函数难以表述规模较大的程序，因为二者的含义相差甚远．

(3)所能综合的程序类受限于系统所选定的程序模式，因此，系统的适应性、灵活性较差．

近年来，人们从机器学习的角度设计了各种归纳程序综合系统[7,8,9]，多数方法使用逻辑公式结构，通过构造一般化的逻辑公式产生假设．基于机器学习的归纳方法视概念、关系等为程序的同义语．这样，就能更多地在程序的逻辑意义下考虑程序综合．这类方法基于从认知科学角度提出的各种归纳规则，强调背景知识的作用，其研究重点是如何产生合理的程序假设，而假设的证实则由人或其它已知的演绎系统完成．由于目前机器学习的研究缺少严谨的理论基础，提出的多数归纳程序综合方法都是针对特定领域的．

我们认为，基于机器学习的归纳程序综合是值得注意的研究方向．在理论上，通过研究归纳推理的一般性质，可以使归纳综合具有坚实的理论基础；在实践上，应该发展各种实用技术，提高归纳程序综合系统的实用性．系统的实用性包括执行功效、灵活性和适应性等若干方面，而背景知识的有效使用对提高系统实用性有重要作用，因此，也是我们在 NDIPS 系统设计中考虑的重点．

§2. 系统设计思想

© 中国科学院软件研究所 http://www.jos.org.cn

归纳推理的两个基本概念是"在有限时间内识认(identification in the limit)"和"通过枚举进行识认(identification by enumeration)"[12,13,14]。迄今为止，归纳推理的理论研究都是围绕这两点展开的。Gold 的研究结果表明，不存在通用的归纳推理方法，即不存在这样的算法，它对于某种程序实例给出方式，能够在有限时间内识认该程序。但如果把程序限制为递归可枚举的程序类且处处终止，则存在能在有限时间内识认它们的方法——通过枚举进行识认。枚举识认是通用的归纳推理方法，当给出假设空间的不同结构定义时，它可以适用于许多不同的领域。但其致命弱点是实用性差，因为假设空间的大小随正确假设的描述长度呈指数级增长。

Plotkin[10]和 Shapiro[11]对子句归纳作了大量的基础性研究。Plotkin 提出了利用背景知识对子句进行一般化的若干策略。用逻辑程序表示归纳假设，Shapiro 的模型推理系统 MIS(Model Inference System)表明，归纳过程可分为两个任务：(i)搜索应包括的合适规则；(ii)排除施用不正确的规则。但 MIS 系统实际上是在设有背景知识的情况下对程序进行归纳综合的，因此，搜索空间较大，只能采用模型制导技术来克服分叉过多的问题。

NDIPS 系统是我们在分析、比较归纳程序综合有代表性的工作基础上设计的一个归纳程序综合系统。我们将系统的需求归结为三个方面：用户友善性、自动化程度和系统的实用性。它们反映在系统中就是：能用较少的实例综合出程序；能自动证实程序假设的一致性和完备性；能支持综合较大规模的程序。为此，我们在 NDIPS 系统中的做法是：

(1)选用 PROLOG 作为目标语言。理由是：(i)程序的逻辑成份与控制成份是分离的，使我们能从抽象算法角度考虑程序综合；(ii)综合出的程序是一概念上独立的谓词，可用于构造其它程序。这种逐层构造程序的方法能支持综合较大规模的程序；(iii)根据 Horn 子句的逻辑结构，易于构造程序假设空间；(iv)目标程序可以直接用已有系统解释执行，有利于将调试技术用于归纳程序结合。特别是利用 PROLOG 的推理机制能较方便地处理程序归纳综合所依据的背景知识。

(2)采用交互方式综合程序。程序 P 的实例给出有三种基本方式：给出的实例都是正例；给出的实例既有正例，也有反例；通过询问给出程序的实例(正例或反例)。只给正例的不足是可能得到过于一般化的程序假设。后两种方式在抽象的意义下是等价的，但后者使用的归纳程序综合算法效率较高，它能指导用户提供合适的实例，避免实例给出的盲目性。

(3)将程序综合作为问题求解。归纳程序综合与问题求解有着自然的对应关系：程序的输入域和输出域对应于问题求解的初始状态集和目标状态集；程序综合的背景知识对应于问题求解的操作集；目标程序对应于问题的通解。我们根据问题描述的不同特征，设计了两种基本归纳方法：基于搜索的归纳方法和基于构造的归纳方法。前者适用于能从问题描述确定其假设空间的情形；后者则通过分析问题的特解构造出其通解。

NDIPS 系统由人机界面、监控机制、实例归纳模块、轨迹归纳模块、问题求解机制、优化模块和知识库等七部分组成(见结构示意图)。

© 中国科学院软件研究所 http://www.jos.org.cn

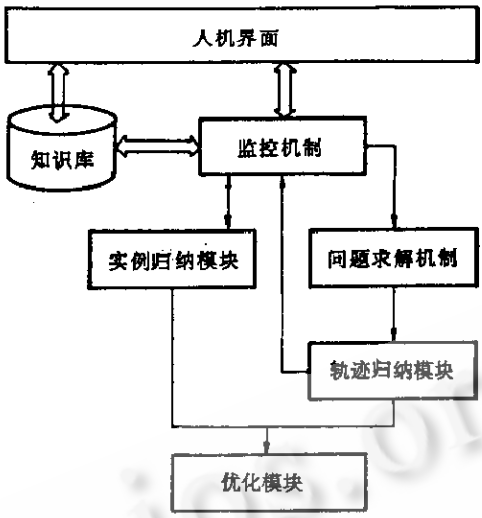

图 1 NDIPS 系统结构示意图

人机界面是用户控制程序综合进程、与系统对话的效互环境．监控机制的任务是分析问题特征，选择合适的归纳方法，并根据模块反馈信息协调各模块的执行．实例归纳模块根据问题定义，建立程序假设空间，利用枚举识认技术搜索满足完备性和一致性的程序假设．其归纳过程包括：搜索可能的程序假设，并利用诊断技术排除错误的程序假设．轨迹归纳模块根据问题求解机制所产生的问题特解路径，按一定的递归模式找出问题的通解．它采用了基于构造的归纳方法，问题定义中的基本操作只用于得到解路径．有时，解路径也可以手工给出．优化模块对产生的目标程序在执行功效等方面进行优化．知识库中存放两类知识：通用知识和领域知识．通用知识包括语言、程序设计、优化和策略知识等；领域知识是与具体归纳问题有关的背景知识．

§3. 背景知识的使用和假设的证实

NDIPS 系统的归纳程序综合方法依赖于背景知识．这类方法的优点是：

(i)不同的背景知识能用来对同一问题产生一算法族，系统可视为一种通用的归纳机制．

(ii)背景知识的级别反映了可综合程序的规模，其内涵和外延决定了问题的可归纳程度．

(iii)领域背景知识拓广了归纳系统的应用范围，使系统具有较好的灵活性和适应性．

在 NDIPS 系统中，背景知识是用事实和规则给出的，解释和使用背景知识是系统综合程序的关键．

由于 Horn 子句逻辑的不确定性，解释使用用 Horn 子句表示的背景知识也是不确定的．背景知识的这种不确定性使搜索空间急剧增大．已有系统的解决办法是：让用户加以指明．我们设计了模式自动推导算法，在例化良好程序的抽象解释的理论模型下，该算法

© 中国科学院软件研究所 http://www.jos.org.cn

能够较精确地确定谓词调用的模式。通过研究谓词模式对归纳程序综合的影响，我们引入了谓词模式的类型特征概念，并总结出的一组按谓词模式类型特征进行枚举的启发式规则，较好地解决了谓词参数的传递问题。

对程序假设空间的有效搜索依赖于程序假设空间的偏序定义。假定 P_i 和 P_j 为程序假设，$\preceq$ 为某种偏序，用 $P_i \preceq P_j$ 表示程序假设 P_i 比 P_j 较为具体，即 P_j 比 P_i 较为一般。在搜索中，当发现某个假设与一个正例不相容时，则无需考虑比它更具体的程序假设；当某个假设包括了一个反例时，则无需考虑比它更一般的程序假设；通常采用置换包含关系定义假设空间的偏序：子句 A 包含子句 B，如果存在某个置换 θ，使得 $A_{head}\theta = B_{head}$ 并且公式 $B_{body} \rightarrow \in (A_{body}\theta)$在任何解释下均为真。这种置换包含关系是一种弱包含关系，在有些情况下，基于置换包含的子句比较是不精确的，因为它没有考虑背景知识。在 NDIPS 系统中，我们利用基于背景知识的泛包含关系组织程序假设空间。

令 θ 是子句 B 的基置换，子句 A 在背景知识 P 下包含了 B，如果存在置换 σ，使得 $A_{head}\sigma = B_{head}$ 并且公式 $P \wedge B_{body}\theta \rightarrow \exists (A_{body}\sigma\theta)$在任何解释下均为真。我们称这种包含关系为泛包含关系。显然，当背景知识为空时，它就退化为置换包含关系。可以证明，泛包含关系下的程序假设空间与置换包含关系下的程序假设空间是等价的。但是，利用泛包含关系可将置换包含关系下的若干程序假设合并为一个等价类，故减少了搜索空间的大小。

归纳推理有两方面内容：产生假设和证实假设。已有的归纳系统能产生假设，但证实通常由人完成，可靠性差。特别当归纳出的程序用于构造其它程序假设时，问题更突出。

假定某个程序假设空间中包含了所需的目标程序 P_j，且能够完全确定它在假设空间中的位置$\cdots \preceq P_i \preceq P_j \preceq P_k \cdots$。为了证实 P_j，在系统进行一般到具体的搜索时，只需证实 P_i 不是目标程序，而当进行具体到一般的搜索以产生 P_j 时，只需证实 P_k 不是目标程序。

NDIPS 系统所使用的基于搜索的归纳方法按两种方式搜索程序的假设空间：一般到具体的搜索和具体到一般的搜索。这两种搜索方式具有不同的特点，分别适用于不同的归纳问题。当满足一定条件时，这两种搜索可以同时进行：

最一般的假设——→……——→目标程序←——……←——最具体的假设

对于可以双向搜索的归纳问题，我们可以在一定意义下对程序假设进行自动证实，即当双向搜索汇聚为一点时，该点的程序假设就是归纳出的目标程序。我们借鉴了 Mitchell 的工作[15]，放宽其中不允许析取概念描述的限制。而对于单向搜索，系统则通过构造"临界实例"，让用户进行假设的证实。令 S(P)表示假设 P 的实例集，则 $S(P_j)-S(P_i)$称为程序假设 P_i 和 $P_j(P_i \preceq P_j)$的临界实例集。如果某个临界实例是正例，则表明 P_i 过于具体；当某个临界实例是反例，则表明 P_j 过于一般。

§4. 结　　语

南京大学计算机软件研究所对软件自动化的转换途径、过程化途径和演绎机制进行了一些研究。为了进一步探讨软件自动化的各种途径，我们开展了归纳途径的研究，在此基础上设计出实验性归纳程序综合系统 NDPIS，并在 SON-3 工作站上实现。系统的主要特点是：(1)采用逐层构造程序的归纳综合方法，能支持综合较大规模的程序；(2)利用泛包含关系定义程序假设空间的偏序结构，减少了搜索空间的大小；(3)设计了解释使用背

© 中国科学院软件研究所 http://www.jos.org.cn

景知识的模式自动推导等算法，提高了系统综合程序的功效；(4)在一定条件下可以对程序假设进行自动证实，提高了系统的自动化程度。

NDIPS 系统目前已通过的实例包括表处理谓词(如 append, insert 和 reverse 等)，集合运算(如 member, subset, union 和 intersection 等)，分类程序(如 insert-sort 和 merge-sort 等)和概念定义(如 ancestor, arch)等。

由于归纳途径研究本身的难度，尚有许多理论问题和技术问题有待于进一步讨论，程序综合只是归纳推理的一个具体应用领域，我们将在更多的领域中探讨归纳推理的应用，以求得对归纳机制在软件自动化中的作用和地位有较为全面和深入的认识。

致 谢：感谢 NDIPS 系统研制组全体同志的同心协力，特别是王军、陆朝俊、章骏和丁琴等。

参考文献

[1] 徐家福，"软件自动化"，计算机研究与发展，1988，11。

[2] P. D. Summers, "A methodology for LISP program construction from examples", J.ACM 24, 1, 1977.

[3] J. P. Jouannaud, G. Guiho, "Inference of functions with an interactive system", Machine Intelligence 9, 1979.

[4] J. P. Jouannaud, Y. Kodratoff, "Characterization of a class of functions synthesized by a Summers-like method using a B. M. W. matching technique", IJCAIS, 1979, pp.440-447.

[5] Y. Kodratoff, "A class of functions synthesized from a finite scheme", Int.J.Comput.Inf.Sci.8, 1979, pp.489-521.

[6] M. A. Bauer, "Programming by examples", Art. Intell. 12, 1979, pp.1-12.

[7] B. Cohn, C. Sammut, "Program Synthesis Through Concept Learning", Automatic Program Construction Techniques, MacMillan Pub. Co., 1984.

[8] C. Sammut, R. B. Banerji, "Learning Concept by Asking Questions", Machine Learning, An Artificial Intelligence Approach(2), 1981, pp.149-191.

[9] S. Amarel, "Program Synthesis as a Theory Formation Task, Problem Representation and Solution Methods", Artificial Intelligence Approach(2), 1986, pp.499-570.

[10] G. D. Plotkin, "A further note on inductive generalization", Machine Intelligence 6, 1971, pp.101-124.

[11] E. Y. Shapiro, "A general incremental algorithm that infers theories from facts", IJCAIS, 1977.

[12] E. M. Gold, "Language identification in the limit", Inf. Control 10, 1976, pp.447-474.

[13] E. M. Gold, "System identification via state characterization", Automatica 8, 1972, pp.621-636.

[14] E. M. Gold, "Complexity of automation identification from given data", Inf. Control 37, 1978, pp.302-320.

[15] T. M. Mitchell, "Version space, A candidate elimination approach to rule learning", IJCAIS, 1977, pp.305-310.

© 中国科学院软件研究所 http://www.jos.org.cn