模拟 Boyer–Moore 定理证明器*

马素霞 郑人杰

(清华大学)

A SIMULATE BOYER–MOORE THEOREM PROVER

Ma Suxia and Zheng Renjie

(Qinghua University)

ABSTRACT

SBMTP (Simulate Boyer–Moore Theorem Prover) system is a theorem proving system which is implemented on microcomputer IBM–PC–386 by GCLISP language. The concepts and theoretical basis of the system are computational logic by Boyer and Moore.

SBMTP is made up of three parts mainly: knowledge base management subsystem, theorem proving subsystem and interrupt recovery subsystem.

Knowledge base management subsystem includes a fundamental knowledge base and a series of knowledge base construction tools. User can organize flexibly his knowledge base in accordance with specific requirements. Theorem proving subsystem employs a heuristic method for reasoning to complete proof. Interrupt recovery subsystem provides powerful recovery ability when proof process gets interrupted. This subsystem improves proof efficiency of the system.

摘 要

SBMTP (Simulate Boyer–Moore Theorem Prover)系统是在 IBM–PC–386 微机上用 GCLISP 语言实现的一个定理证明系统. 该系统采用的思想方法和理论基础是 Boyer–Moore 的计算逻辑理论.

SBMTP 主要由三部分组成: 知识库管理部分、定理证明部分及中断恢复部分.

知识库管理部分包括一个基本知识库和一系列知识库构造工具. 用户可根据具体问题

* 1989年7月14日收到.

灵活地组织自己所需要的知识库．定理证明部分采用启发式方法逐步推演，完成证明．中断恢复部分在证明产生中断的情况下提供了较强的恢复能力，提高了证明效率．

§1. 引 言

在计算机领域里，研究定理证明的直接动机是程序验证．程序验证的主要思想是，按照数学的意义精确地指出并证明程序的语义属性．一般来说，程序验证系统由验证条件产生器和定理证明器两部分组成．程序属性由验证条件产生器以定理的形式指出，而定理的证明则由定理证明器来完成．由此可见，程序验证的自动化问题实质上归结为定理的机械化生成问题以及定理的机械化证明问题．现在，程序机械化证明还难以实用，其主要原因是缺乏机械定理证明的能力．设计和实现一个强有力的定理证明器是程序验证的关键．

定理证明的方法很多．一般采用基于一阶谓词逻辑的归结方法．

在定理证明的方法中存在两个极端的情况：证明检查和自动证明．证明检查是指由用户逐步指导如何使用推理规则进行形式证明．虽然对于高级的用户来说，它增加了形式证明的可靠性，对于一般的用户来说，反而因用户不适当的干扰而影响了证明的可靠性．

自动证明是指给定一组公理及一组推理规则后，完全由证明工具进行形式证明．基于一阶谓词逻辑的归结方法就是一种自动证明方法，对于不可满足的公式来说，这种方法是完备的．但因为要试探所有的路径使方法带有盲目性和冗余性，因时空开销过大，及至不能在一个合理和可容忍的时间内构造稍微复杂一点的定理的形式证明．

在证明检查和自动证明两个极端情况之间有一种妥协，就是启发式的证明方法，即用启发式方法实现的系统对待证命题进行分析来寻找一种有效的推理路径．Boyer-Moore 方法就是一种启发式方法，被认为是目前最有前途的定理证明方法．

§2. Boyer-Moore 方法简介

Boyer-Moore 方法来源于 Goodstain 建立的递归数论，即由递归定义的基本思想来构造函数；用数学归纳法来证明涉及递归函数的定理．

Boyer 和 Moore 采用与递归数论一样的处理方法，将全称变元看成是自由变元，存在变元用具体函数来表示．另外，程序验证中的推理往往是关于程序对象的推理，而许多程序对象如自然数、栈、树等都可递归定义．因此，数学归纳法在程序验证中一定占很重要的地位．Boyer 和 Moore 提出的计算逻辑理论使递归定义和数学归纳法的使用更加直接和自然．

在计算逻辑中，公理的引入必须满足递归定义的存在性原理：外壳原理和定义原理．外壳原理是关于递归对象的公理的规范，它通过一个统一的公理模式来产生关于各种递归对象的公理，并保证产生的公理与已有的理论不矛盾．定义原理是关于递归函数的公理的规范，它通过一定的检查来保证定义函数的存在性以及与已有的理论不矛盾．

Boyer 和 Moore 还提出了一组启发式方法．这组启发式方法包括：简化(利用类型信息，函数信息及公、引理来简化)、分元符删除、交融、推广、无关式删除和数学归纳法．每一个方法都由两部分组成，第一部分决定是否选用该方法，第二部分是该方法的实

施.

有关 Boyer-Moore 的计算逻辑理论请参阅文献[1], [2], [7].

§3. SBMTP 的设计与实现

1. SBMTP 的总体结构

SBMTP 系统主要由知识库集、知识库管理工具、证明环境构造工具及推理工具组成. 其基本结构如图 1 所示. 数据流图如图 2 所示.

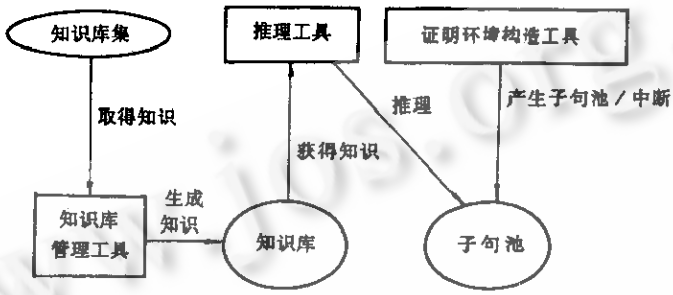

图 1

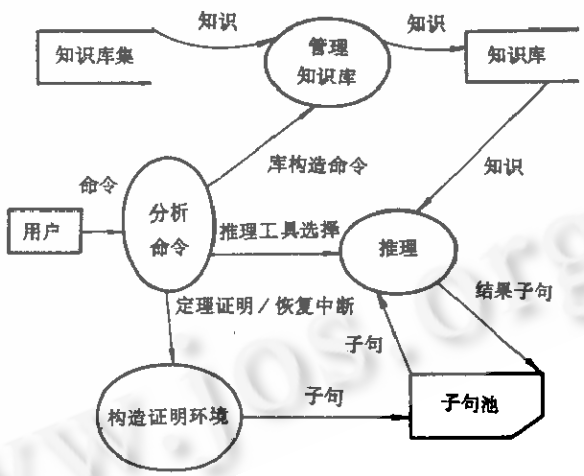

图 2

2 知识库管理工具

知识库管理工具由 KB--show、KB-make、KB-era、KB-rem、KB-del、KB-adds、KB-addf、restore 和 KB-adda 组成.

(1) KB-show

KB-show 是显示知识库内容的工具. KB-show 提示用户指明要显示的知识库, 用户输入选择后, KB-show 将选择的知识库中的知识逐条显示在屏幕上.

(2) KB-make

KB-make 是产生新知识库工具. 用户可以利用这个工具构造新的知识库. 构造的新知识库具有基本知识库的内容. 在此基础上, 用户可利用其它工具来构造自己需要的知识

库.

(3) KB-era

KB-era 是知识库删除工具．用户利用这个工具将不再需要的知识库删除，但要求删除的知识库为空．

(4) KB-rem

KB-rem 用于库间知识移动．KB-rem 提示用户指明源知识库及目标知识库，用户输入选择后，KB-rem 将源知识库中的知识逐条显示，且每显示一条知识都提问用户是否要移动这条知识．KB-rem 将用户要移动的知识加到目标库之中，而源知识库的内容不改变．

(5) KB-del

KB-del 是知识删除工具．KB-del 逐条显示知识，并提问用户是否将其删除．

(6) KB-adds

KB-adds 是增加外壳工具．KB-adds 要求用户输入外壳定义，然后自动检查它是否符合外壳原理、产生有关的外壳信息及一组公理，最后将产生的知识存入知识库．

(7) KB-addf

KB-addf 是增加函数的工具．KB-addf 要求用户输入函数定义，然后自动检查它是否符合定义原理，并产生相应的函数信息，最后将产生的知识存放知识库．

(8) restore

restore 是知识删除工具．它提示用户输入要恢复的步数 n，然后将知识库中最后输入的 n 条知识删除．

(9) KB-adda

KB-adda 是增加公理的工具．

3. 证明环境构造工具

证明环境构造工具由定理转换工具、谓词简化工具及子句池恢复工具组成．其主要功能是构造证明中所需要的子句池．

(1) 定理转换工具

其功能是将定理转换成不含 implies、and、or 的子句形式．

(2) 谓词简化工具

给定一子句集，按如下算法进行简化:

(Ⅰ) 任何一个含有一个文字及其否定的子句均可被消去．

(Ⅱ) 如果子句中含有文字(true)，则整个子句都可消去，如果在子句中含有文字(false)，则将此文字从子句中消去．

(Ⅲ) 如果子句中一文字有多次出现，则只保留一次出现．

(3) 子句池恢复工具

此工具在系统接到恢复中断证明的命令(来自用户输入)后，将文件 his-file. lsp 打开，并将其内容加到空子句池中．

文件 his-file.lsp 是证明过程中形成的．证明中断时，his-file.lsp 中记录着子句池最后一次改变的前一状态的内容．

4. 推理工具

推理工具是根据 Boyer-Moore 的一组启发式方法实现的，这组启发式方法包括：

(1) 简化：其主要功能是利用类型信息、函数定义的启发式展开和重写公(引)理对要证公式进行简化。

(2) 分元符删除：其主要功能是删除公式中的分元符(即由整体求分量的逆函数)而将公式等价地转换为一个不含分元符的子句集。

(3) 交融：其主要功能是使用假设中的等式，并在使用之后放弃这个假设。

(4) 推广：其主要功能是将一特殊例证启发式地推广到一个一般的公式，即把一系列表达式的出现替换以变量的出现。

(5) 无关式删除：其主要功能是删除公式中与公式证明无关的部分。

(6) 数学归纳法：其主要功能是启发式地为公式寻找一个归纳匹配模式。

系统按如下规则(图 3)提示用户选择推理工具：

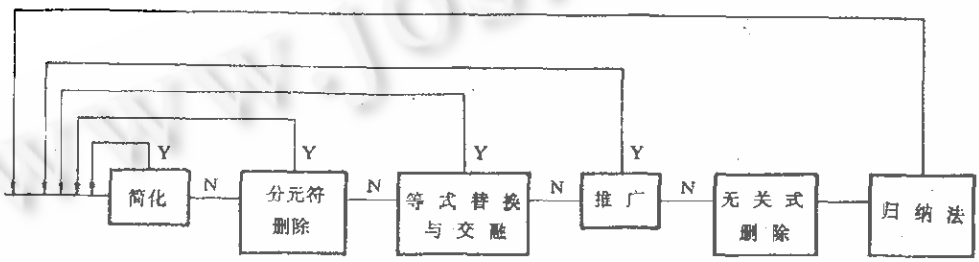

图 3

其中，"Y"是指运用此工具产生的结果不同于只含输入元的孤元集，在这种情况下，将结果加入到子句池中(顶端)，并从子句池中(顶端)取出一子句，重新进入此过程。"N"是指运用此推理工具对其输入不产生任何影响，在这种情况下就使用下一推理工具。

按上述规则，系统使六种工具之一在屏幕上闪烁，以提示用户选择此工具。

对系统使用不太熟的用户可按照系统的提示选择推理工具，当对系统使用熟悉之后，可以不按照系统的提示选择，而是按如下规则来选择：

(1) 对情况"Y"的选择不变。

(2) 对情况"N"，如用户凭经验知道系统提示的工具不会对正在证明的子句产生影响，则可跳过此工具，而选择其后工具。

5. SBMTP 设计的两点说明

(1) 在 SBMTP 中没有采用单一的知识库，而是采用了知识库集。这是由于 SBMTP 是在 IBM-PC-386 微机上实现的，内存容量十分有限，如果将各个方面的知识都放在一个库中，势必造成知识库的日益庞大，到一定限度后使 SBMTP 系统无法运行。另外，将不同方面的知识存于不同的库中有利于提高知识库的检索效率和便于用户管理。

(2) 定理的证明并不都是一帆风顺的，尤其是复杂定理的证明，往往因为各种原因发生中断，其中最常见的是由于 GCLISP 系统中 cons 与 atom 空间的分配不适当造成定理证明的中断，在这种情况下就需要修改 cons 与 atom 空间的比例。我们希望修改后能够从中断点继续运行，这就需要设置中断恢复功能来提高证明效率。

SBMTP′的中断恢复功能是通过在关键的断点处保存子句池信息来实现的.

6. 证明例示

证明定理(implies (plistp x_1)(equal (reverse (reverse x_1))x_1))

其中(plisty x_1) = (if (listp x_1) (plistp (cdr x_1)) (equal x_1("nil")))

(reverse x_1) = (if (listp x_1) (append (reverse (cdr x_1)) (cons (car x_1)("nil"))) ("nil"))

(append x_1x_2) = (if (listp x_1) (cons (car x_1) (append (cdr x_1) x_2)) x_2)

证明过程:

(1) 选归纳法得下一归纳法模式:

(and (implies (not (listp x_1)) (p x_1)) (implies (and (listp x_1) (p (cdr x_1)))(p x_1)))

上述归纳法模式导出下列三个新推测:

(Ⅰ) (implies (and (not (listp x_1)) (plistp x_1)) (equal (reverse (reverse x_1)) x_1))

(Ⅱ) (implies (and (listp x_1) (not (plistp (cdr x_1))) (plistp x_1)) (equal (reverse (reverse x_1)) x_1))

(Ⅲ) (implies (and (listp x_1) (equal (reverse (reverse (cdr x_1))) (cdr x_1)) (plistp x_1)) (equal (reverse (reverse x_1)) x_1))

(2) 选简化工具把(Ⅰ)和(Ⅱ)简化为真，把(Ⅲ)简化为:

(implies (and (listp x_1) (equal (reverse (reverse (cdr x_1)))(cdr x_1)) (plistp (cdr x_1))) (equal (reverse (append (reverse (cdr x_1)) (cons (car x_1)("nil")))) x_1))

(3) 选择分元符删除工具，用(cons x_2 x_3)替代 x_1，将(cdr x_1)和 (car x_1)删除，并选简化工具后得

(implies (and (equal (reverse (reverse x_3)) x_3) (plistp x_3)) (equal (reverse (append (reverse x_3) (cons x_2("nil")))) (cons x_2 x_3)))

(4) 选交融工具将 x_3 交融成(reverse (reverse x_3))，并抛弃该假设后得:

(implies (plistp x_3) (equal (reverse (append (reverse x_3) (cons x_2("nil")))) (cons x_2(reverse (reverse x_3)))))

(5) 选推广工具用 x_1 替代(reverse x_3)得:

(implies (plistp x_3) (equal (reverse (append x_1(cons x_2("nil")))) (cons x_2(reverse x_1))))

(6) 选无关式删除工具得

(equal (reverse (appoend x_1(cons x_2("nil")))) (cons x_2(reverse x_1)))

(7) 选归纳法得归纳模式:

(and (implies (not (listp x_1)) (p x_1x_2)) (implies (and (listp x_1) (p (cdr x_1) x_2)) (p x_1x_2)))

上述模式产生两个新目标:

(Ⅰ) (implies (not (listp x_1)) (equal (reverse (append x_1(cons x_2("nil")))) (cons x_2(reverse x_1)))),

(Ⅱ) (implies (and (listp x_1) (equal (reverse (append (cdr x_1)(cons x_2("nil"))) (cons x_2(reverse (cdr x_1))))) (equal (reverse (append x_1(cons x_2("nil")))) (cons x_2(reverse x_1))))

(8) 选简化工具把(Ⅰ)简化为真，把(Ⅱ)简化为:

(implies (and (listp x_1) (equal (reverse (append (cdr x_1) (cons x_2("nil")))) (cons x_2(reverse (cdr x_1))))) (equal (append (reverse (append (cdr x_1) (cons x_2("nil")))) (cons (car

x_1) ("nil"))) (cons x_2 (append (reverse (cdr x_1)) (cons (car x_1) ("nil"))))

(9) 选分元符删除工具，用(cons x_3 x_4)替代 x_1、将(car x_1)和(cdr x_1)删除，并选简化工具后得:

(implies (equal (reverse (append x_4(cons x_2("nil")))) (cons x_2(reverse x_4))) (equal (append (reverse (append x_4(cons x_2(" nil")))) (cons x_3(" nil"))) (cons x_2 (append (reverse x_4) (cons x_3("nil"))))))

(10) 选交融工具，将(reverse (append x_4(cons x_2(" nil"))))交融成(conx x_2(reverse x_4))、抛弃该等式，再选简化工具后得真．由于子句池中所有子句都为真，所以原定理为真．

§4. 结 论

SBMTP 系统具有如下特性:

1. SBMTP 中用一系列启发式方法来实现推理工具，在证明过程中经常释放不需要的内存，在很大程度上降低了时空开销．

2. 知识库的构造灵活．

3. SBMTP 支持从顶向下推理和从底向上推理两种方式．

4. SBMTP 提供了较强的证明中断恢复能力，提高了证明效率．

5. 系统自动记录定理证明过程，以供用户查阅．

6. 推理工具是可扩充的，可以增加新的推理工具，也可修改旧的推理工具．

SBMTP 的实现目标之一是计划在将来实现的"软件重用系统"中应用．

目前，SBMTP 只是一个独立的定理证明系统，要想将其用在将来实现的"软件重用系统"中，还需要许多改进，尤其是与整体的连接以及证明效率的提高，应着重考虑．

最后，SBMTP 系统的实现得益于武汉大学李卫华老师和上海交通大学孙永强、陆汝占老师的热情帮助和指导，他们在定理证明方面的工作给了我们很大启示，在此致谢．

参 考 文 献

[1] 李卫华译，"计算逻辑"，《计算机工程与应用》，1982，4-5，1983，7．

[2] 李卫华，"定理证明与程序验证"，《计算机科学》，1982.1．

[3] D. I. Good, "Mechanical Proof About Computer Programs", Institute of Computing Science, The University of Texas at Austin, Austin, Texas 78712, U.S.A., 1984.

[4] R. S. Boyer and J.S.Moore, "Proving Theorems About Lisp Functions", Journal of the Association for Computing Machinery, 1975.

[5] Shmuel Kate, "PARIS: A System for Reusing Partially-Interpreted Schemas", MCC Software Technology Program," Austin, Texas, 1987.

[6] D. I. Good, "An Interactive Program Verification System", IEEE Transaction on Software Engineering, Vol. se-1, No.1, March 1975.

[7] 李卫华译，"计算机科学中的正确性问题"，《计算机工程与应用》，1984，10-11．

第一届中国人工智能联合学术会议征文通知

中国人工智能学会、中国自动化学会模式识别与机器智能专业委员会、中国计算机学会人工智能与模式识别专业委员会、中国计算机学会软件专业委员会智能软件学组、中国软件行业协会人工智能协会、全国高校人工智能研究会等六个有关人工智能学术组织负责人，于1989年7月8日在清华大学进行了协商，决定：于1990年7月在长春吉林大学召开第一届中国人工智能联合学术会议，会议将正式出版论文集向国内外发行，会议征集未发表的论文和专题报告：

1.人工智能的理论与方法.
2.智能搜索与方法.
3.知识获取与机器学习.
4.专家系统与知识工程.
5.自动推理与定理机器证明.
6.自然语言理解和生成.
7.人工智能的工具和技术.
8.并行和分布式处理
9.智能计算机体系结构.
10.模式识别与图象处理.
11.计算机视觉与智能机器人.
12.智能辅助系统(ICAD，ICAM，ICAI等).
13.智能控制与智能管理.
14.人工智能的其它应用.

征文(不退稿)截止日期：1989年11月30日(以邮戳为准)

通知接收日期：1989年11月31日以前

录用论文标准格式返回日期：1990年3月10日以前

征文联系人：长春市吉林大学计算机系李志林

会议设学术顾问、主席团、程序委员会、组织委员会、秘书长等，名单如下(以姓氏笔划为序)：

学术顾问：王湘浩，李家治，秦元勋，曾宪昌，慈云桂

主席团：王树林，石纯一，石青云，陆汝钤，何志钧，涂序彦

程序委员会：

主席：刘叙华　**副主席**：马希文，孙怀民，何华灿，张钹，庞云阶，宣国荣，戴汝为

委员：万嘉若，史忠植，刘开瑛，邢汉承，刘椿年，刘锡荟，李介谷，李太航，李祖枢，宋国宁，吴立德，余瑞钊，张一立，张运祯，林尧瑞，周国栋，周祥和，胡运发，洪家荣，黄厚宽，裴珉，熊范纶，蔡庆生

组织委员会：

主席：鞠九滨　**副主席**：刘大有，李志林，童天湘

委员：辛跃权，曲学楼，孟凡二，武艳茹，姜山青

会议秘书长：刘大有，袁萌

中国人工智能学会
中国自动化学会模式识别与机器智能专业委员会
中国计算机学会人工智能与模式识别专业委员会
中国计算机学会软件专业委员会智能软件学组
中国软件行业协会人工智能协会
全国高校人工智能研究会

1989.7.25