

基于IMOM和IBOHM启发式策略的扩展规则算法*

李莹^{1,2}, 孙吉贵^{1,2}, 吴瑕^{1,2}, 朱兴军^{1,2}

¹(吉林大学 计算机科学与技术学院, 吉林 长春 130012)

²(吉林大学 符号计算与知识工程教育部重点实验室, 吉林 长春 130012)

Extension Rule Algorithms Based on IMOM and IBOHM Heuristics Strategies

LI Ying^{1,2}, SUN Ji-Gui^{1,2}, WU Xia^{1,2}, ZHU Xing-Jun^{1,2}

¹(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

²(Key Laboratory of Symbolic Computation and Knowledge Engineering of the Ministry of Education, Jilin University, Changchun 130012, China)

+ Corresponding author: E-mail: liying_21020910@email.jlu.edu.cn

Li Y, Sun JG, Wu X, Zhu XJ. Extension rule algorithms based on IMOM and IBOHM heuristics strategies. Journal of Software, 2009,20(6):1521–1527. <http://www.jos.org.cn/1000-9825/3420.htm>

Abstract: Extension rule-based theorem proving is a splendid reasoning method. Based on the extension rule algorithm IER (improved extension rule), this paper proposes IMOM (improved maximum occurrences on clauses of maximum size) and IBOHM (improved BOHM) heuristic strategies in order to give guidance while choosing the clause that limits the search space. This paper applies these two heuristic strategies to the algorithm IER, designs and implements the algorithms IMOMH_IER and IBOHMH_IER. Since more appropriate search space can be obtained with these two heuristic strategies, the algorithms can decide whether the original question is satisfiable in a fraction of a second. Experimental results show that the speed of the algorithms is 10~200 times that of the existing algorithms DR (directional resolution) and IER.

Key words: theorem proving; propositional logic; extension rule; heuristic strategy; resolution

摘要: 基于扩展规则的方法是一种定理证明方法。在 IER(improved extension rule)扩展规则算法的基础上,提出了 IMOM(improved maximum occurrences on clauses of maximum size)和 IBOHM(improved BOHM)启发式策略,并将两种启发式策略用于 IER 算法中,有指导性地选择限定搜索空间的子句,设计并实现了算法 IMOMH_IER 和 IBOHMH_IER。实验结果表明,由于这两种启发式策略能够选择较为合适的搜索空间,可以尽快地判定出原问题是否可满足,故其速度平均能够达到原有算法 DR(directional resolution)和 IER 的 10~200 倍。

关键词: 定理机器证明;命题逻辑;扩展规则;启发式策略;归结

中图法分类号: TP181 文献标识码: A

自动推理一直是人工智能领域最重要的研究方向之一,现已成功地应用到包括逻辑、数学、计算机科学、

* Supported by the National Natural Science Foundation of China under Grant No.60773097 (国家自然科学基金); the Young Scientific Research Foundation of JiLin Province of China under Grant No.20080107 (吉林省青年科研基金)

Received 2007-11-05; Accepted 2008-07-02

工程和社会科学等领域.很多重要问题一直使用自动定理证明的方法求解,并在一些领域中已经取得了很成功的应用,如数学定理证明、软件生成与验证^[1]、协议验证和硬件验证^[2]等.

基于归结的方法^[3]、基于表的方法^[4]、相继式演算^[5]以及自然演绎法^[6]等是定理机器证明中常用的方法.定理机器证明中通常的思想是通过推出空子句的方法判定子句集是否可满足,基于归结的方法是这种思想的一个典型例子,也一直是定理机器证明中应用最为广泛的方法之一.

我们在文献[7]中提出的基于扩展规则定理的证明方法是一种新的定理机器证明方法,它沿着归结的逆向进行定理机器证明,通过扩展出所有极大项组成的集合判定子句集是否可满足,同时使用容斥原理解决这样做所带来的空间复杂性问题.

作为与基于归结的方法互补的方法,在互补因子较大时,基于扩展规则方法的效率较高.算法IER (improved extension rule)是高效的基于扩展规则的算法,但该算法中用于限定搜索空间的子句的选取是随机的,没有采用任何启发式策略^[7].

为了提高扩展规则算法的效率,使之更加高效地用于判定可满足性问题,本文在IER的基础上提出了IMOM(improved maximum occurrences on clauses of maximum size)和IBOHM(improved BOHM)启发式策略,并将两种启发式策略用于IER算法中限定搜索空间的子句C的选择,进而提出了算法IMOMH_IER和IBOHMH_IER.它们利用子句集的信息选择子句C,可以得到更适合该问题的子搜索空间,这样,对可满足的问题可以直接返回结果“SAT”,不可满足的问题可以较早地调用完备的ER(extension rule)算法进行判断,从而减少了问题的判定时间.实验结果表明,算法IMOMH_IER和IBOHMH_IER能够在很大程度上提高原有扩展规则算法的效率,速度平均达到算法DR(directional resolution)和IER的10~200倍.

1 扩展规则方法 ER 和 IER

扩展规则是以所有极大项组成的子句集不可满足为基础的.对于一个子句集,如果能够由它推导出所有极大项组成的子句集,则该子句集不可满足,推导过程中使用的方法称为扩展规则.

给定一个子句 C 和一个原子的集合 M ,

$$D = \{C \vee a, C \vee \neg a \mid a \in M \text{ 并且 } a \text{ 和 } \neg a \text{ 都不在 } C \text{ 中出现}\}.$$

我们把从 C 到 D 中元素的推导过程叫做扩展规则, D 中的元素叫做应用扩展规则的结果^[7,8].

扩展规则的目的是要尽量地把 ϕ 扩展成极大项,然后根据极大项的个数判定该子句集的可满足性.显然,直接应用扩展规则最坏情况下的空间复杂性是呈指数级的.实际上,我们并不需要把所有的极大项都扩展出来,只需计算它们的个数就足够了.如果极大项的个数是 2^m 个,则子句集不可满足;否则,子句集可满足.计算一个子句集可扩展出极大项的个数可以利用容斥原理.

基于扩展规则的算法IER的思想是,在完备的ER算法前先运行一个高效的但并不完备的算法,希望多数的问题能够由这种高效的算法解决,对于解决不了的问题再调用原来完备的算法.具体来说,当我们应用ER算法时,实际上是在搜索所有极大项组成的空间,看其是否可扩展出所有的极大项.然而,我们可以搜索该空间的一个子空间,如果在这个子空间上有极大项不能被扩展,那么显然在整个空间上也应该有极大项不能被扩展,即子句集可满足;否则,我们不能判定该子句集的可满足性,此时再调用原来的ER算法^[7].

2 IMOMH_IER 和 IBOHMH_IER 算法

在IER算法中,通过选取子句C限定搜索空间,尽可能地在该空间上判定出问题是否可满足,从而达到提高效率的目的,所以C的选取十分重要.基于此,我们参考DPLL算法中动态选择下一个分裂变量的启发式策略MOM和BOHM的思想^[9],提出了本节的启发式策略IMOM和IBOHM,并将两种启发式策略用于IER算法中限定搜索空间的子句的选择,提出算法IMOMH_IER和IBOHMH_IER,下面分别介绍这两种算法.

2.1 IMOMH_IER算法

在给出算法 IMOMH_IER 之前,我们先简要介绍一下 MOM 启发式策略.

MOM启发式策略^[9,10]:优先选择最小子句中出现的变量.令 $f(L)$ 为文字 L 出现在最小未被满足的子句中的数目,选择变量 x ,该变量使下列函数值最大:

$$[f(x)+f(\neg x)] \times 2^k + f(x) \times f(\neg x).$$

如果在最短子句集中,变量 x 的正出现比较多,则将变量 x 赋值为 true,否则赋值为 false.该策略倾向于尽可能多地满足最短子句.受该策略思想的启发,我们希望尽量弄假最长子句中的文字,从而提出 IMOM 启发式策略.

IMOM启发式策略:在子句集中,优先选择最长子句中出现的频率高的文字的否定加入子句 C .令 $f(L_i)$ 为文字 L_i 出现在最长的子句集中的数目,选择文字 $\neg L$ 使得函数 $f(L)$ 的值最大.

下面给出 IMOM 策略的系统实现框架.

```
function IMOM()
  S ← getMaxLengthofClause(Φ);
  For each  $C_i \in S$  do
    for each  $L \in literal(C_i)$  do
       $Count_L \leftarrow Count_L + 1$ ;
  Num ← getKthCount();
  For each  $L \in literal(S)$  do
    if  $Count_L > Num$  then  $C \leftarrow C \cup \{\neg L\}$ ;
  return C;
```

算法的基本流程是:首先遍历子句集 Φ 得到其中最长子句的集合 S ,该功能由函数getMaxLengthofClause完成.然后遍历集合 S ,记录每个文字在 S 中出现的次数,用数组Count计数.统计得到每个文字出现的次数后,调用函数getKthCount,得到Count数组中第 K 大的值Num.最后遍历 S 中出现的所有文字 L ,判断其出现次数 $Count_L$ 是否大于Num.若大于,则将其否定 $\neg L$ 加入子句 C .

在该启发式策略中,基于两个优先原则选择文字:首先,最长子句中文字的否定的优先级比较高.因为子句越长,满足该子句的候选真值指派就越多,比短子句的约束要弱.将文字的否定加入子句 C 中,就意味着选取后所限定的空间使得长子句中对应的文字值为 false.由于长子句的候选真值指派比较多,得到的问题在限定的搜索空间中仍可以保持原问题的可满足性,即可满足的问题仍然可以在限定的搜索空间内判定出其可满足性,直接返回结果 SAT.其次,出现次数较多的文字的否定优先级别比较高.因为我们将所有与 C 含有互补对的子句删掉,这样选取文字使得删除子句的个数最多,得到的问题规模最小.由于规模的缩小,先运行的不完备算法可以很快地返回结果.

通过以上的分析可知,该启发式策略很适用于算法 IER 中指导限定搜索空间的子句 C 的选取.应用 IMOM 启发式策略生成的子句 C 使得可满足的问题在限定的空间内即可返回 SAT,不必再调用 ER 算法;对于不可满足的问题,不完备的高效算法可以较早地返回,更早地调用 ER 算法,这就减少了判定所用的时间,提高了算法的效率.下面给出算法的基本实现框架.

```
function IMOMH_IER-framework(CNF: Φ)
  Φ ← Φ
  C ← IMOM();
  For each  $C_i \in \Phi$  do
    if (complement( $C_i$ )) then  $\Phi \leftarrow \Phi / \{C_i\}$ ;
  total ← extend(Φ)
  if total =  $2^{CLAUSENUMBER - |C|}$  then return ER(Φ);
  else return SAT;
```

在IMOMH_IER算法中,首先调用函数IMOM,该函数根据IMOM启发式策略生成子句 C .然后,对于子句集 Φ 中的每个子句 C_i ,判断是否与 C 含有互补对,若含有,则将 C_i 删除.删除后,对子句集 Φ 在缩小后的空间进行扩展,

Φ 可扩展的极大项个数记为 $total$.如果 $total$ 的值小于 C 所限定的空间中所有极大项的个数 $2^{CLAUSENUMBER-|C|}$ (其中, $CLAUSENUMBER$ 是原子句集 Φ 中子句的个数, $|C|$ 是子句 C 的长度),即在 C 限定的子空间内存在子句集不可扩展的极大项,则 Φ 可满足,算法返回SAT;否则,调用算法ER,重新判断原子句集 Φ 的可满足性.应用IMOM启发策略选取的子句 C ,对于可满足的问题,极大地压缩了搜索空间,绝大多数情况下无须再调用ER;对于不可满足的问题,由于搜索空间的缩小,不完备的算法能够尽快地返回,然后调用ER,同样可以提高算法的效率.

定理 1. 在命题逻辑中,算法 IMOMH_IER 是正确且完备的.

证明:由于 IMOM 启发式策略根据文字在子句集中出现的频率对文字进行选取,所以选取的子句 C 对应的原子集必然包含于子句集的原子集.如果在 C 限定的子空间中存在不可扩展的极大项,那么子句集是可满足的;否则,调用 ER 算法重新判定子句集的可满足性.由此可见,算法 IMOMH_IER 是正确且完备的. \square

2.2 IBOHMH_IER算法

BOHM 提示策略^[9,11]:在分裂的每一步,选择一个拥有极大向量($H_1(x), H_2(x), \dots, H_n(x)$)的变量 x 进行分裂,其中,每个 $H_i(x)$ 定义如下:

$$H_i(x) = \alpha \times \max(h_i(x), h_i(-x)) + \beta \times \min(h_i(x), h_i(-x)).$$

$h_i(x)$ 是长度为 i 的未被满足的包含 x 的子句个数, \max 表示取最大, \min 表示取最小.可以看出,该启发式策略偏向于满足长度小的子句(当该文字为真时)或者继续减小短子句的长度(当该文字为假时). α 和 β 按情况选定. BOHM提示策略对于解决随机产生的SAT问题很有效.根据该算法的思想,我们提出适用于IER算法的IBOHM启发式策略,下面给出其详细描述.

IBOHM启发式策略:在子句集中,优先选择拥有极大向量($H_1(L_i), H_2(L_i), \dots, H_n(L_i)$)的文字 L_i 加入到子句 C 中,其中, $H_j(L)$ 为子句集中长度为 j 的包含文字 L 的子句个数.下面给出IBOHM策略的系统实现框架.

function IBOHM()

For each $C \in \Phi \wedge L \in C$ do

$H_{counter[L]}(L) \leftarrow H_{counter[L]}(L) + 1;$

$(H_1(L), H_2(L), \dots, H_n(L)) \leftarrow getKthH();$

for each $L \in literal(S)$ do

if $(H_1(L'), H_2(L'), \dots, H_n(L')) \geq (H_1(L), H_2(L), \dots, H_n(L))$ then

$C \leftarrow C \cup \{L'\};$

return $C;$

算法的基本流程是:首先遍历子句集 Φ ,得到对应每个文字的 H 向量的值.然后调用函数 $getKthH$,得到 H 中第 K 大的向量($H_1(L), H_2(L), \dots, H_n(L)$).遍历 Φ 中的所有文字 L' ,若向量($H_1(L'), H_2(L'), \dots, H_n(L')$)大于等于向量($H_1(L), H_2(L), \dots, H_n(L)$),则将 L' 加入子句 C 中.

在该启发式策略中,优先选择较短子句中出现频率较高的文字加入子句 C 中.即优先选择最短子句集中出现频率较高的文字加入子句 C 中,如果 L_1 和 L_2 在最短子句中出现次数相同,但在次短子句集中, L_1 出现次数大于 L_2 出现的次数,则选择 L_1 加入到子句 C 中,依此类推.通过第 2.1 节中的分析我们知道,子句越短,满足该子句的候选真值的指派就越少,进而短子句的约束条件也就越强.优先选择较短子句集中出现次数最多的文字放入子句 C 中,这就意味着选取后所限定的空间满足包含该文字的短子句,而且该启发式选取的子句所限定的空间使得被满足的最短子句最多,因此得到的子问题在限定的搜索空间中可以很好地保持原问题的可满足性,即可满足的问题在限定的搜索空间内可以尽可能正确地判定出其可满足性,直接返回结果SAT.

与IMOMH_IER算法的实现框架类似,IBOHMH_IER算法中首先根据IBOHMH启发式策略生成子句 C .然后,对于子句集 Φ 中的每个子句 C_i ,将与 C 含有互补对的子句删掉,对子句集 Φ 在缩小后的空间进行扩展,后面的操作与IMOMH_IER算法相同,这里不再赘述.

定理 2. 在命题逻辑中,算法 IBOHMH_IER 是正确且完备的.

证明:与算法 IMOMH_IER 类似,容易得出算法 IBOHMH_IER 是正确且完备的. \square

3 实验结果

本节将比较我们提出的 IBOHMH_IER 和 IMOMH_IER 算法与原算法 IER 以及 Dechter 和 Rish 提出的命题逻辑中效率较高的归结方法——有向归结算法^[12]。通过测试可以看出,我们的算法较之原算法在执行效率上有明显的提高。测试环境为:硬件 DELL Intel Pentium 4 2.8GHz CPU/512MB RAM;软件 Windows XP Professional SP2/Visual C++6.0。

随机问题样例由一个随机产生器产生,这个随机产生器有 3 个输入参数 (N, M, K) , 其中, N 代表子句集中子句的个数, M 代表子句集中变量的个数, K 代表子句集中子句的最大长度。每个子句都是通过随机地从 M 个变量中选取小于等于 K 个变量得到的, 并且每个变量为正或为负的概率都等于 0.5。我们用不同规模的问题对改进后的算法进行测试和比较, 选取参数分别为 $(N, 30, 8)$, $(N, 30, 9)$, $(N, 30, 10)$, $(N, 35, 8)$, $(N, 35, 9)$ 和 $(N, 35, 10)$ 这 6 组用例进行测试, 其中, N 介于 100~200 之间。对于每类难度的用例, 随机生成 10 个实例进行求解, 均值作为最后结果, 实验结果如图 1 所示。

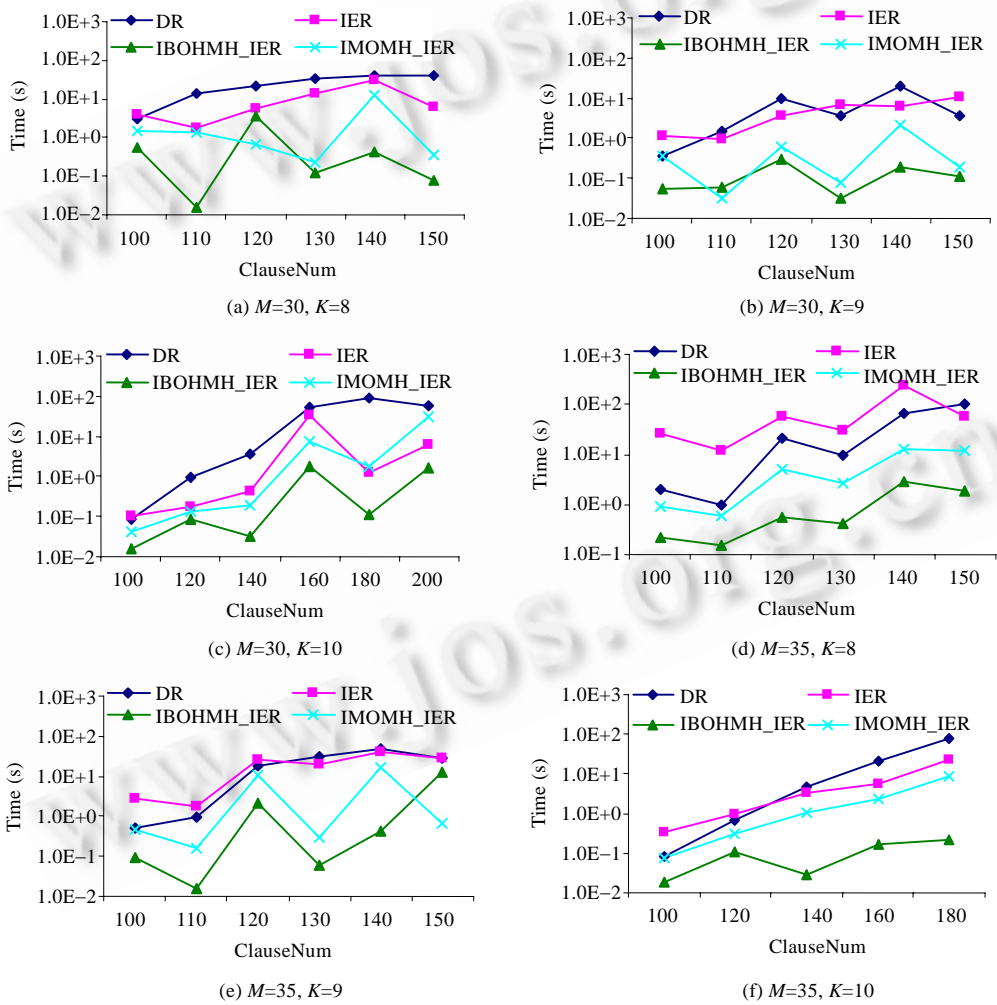


Fig.1 Experimental results of random problem instances

图 1 随机问题用例测试结果

此外,我们对处于相变区域的 Uniform Random-3-SAT 问题和 AIM 类问题(<http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html> 2007)进行测试,实验结果见表 1.表中“—”表示时间超时,超时时间为 600s.

Table 1 Experimental results of benchmark instances

表 1 标准用例测试结果

Instances	Algorithms			
	DR (s)	IER (s)	IBOHMH_IER (s)	IMIMH_IER (s)
uf20_07	0.781	2.875	0.015	0.015
uf20_08	0.710	0.265	0.015	0.014
uf20_09	3.671	6.171	0.140	0.141
uf20_010	3.703	3.600	0.031	0.312
uf20_013	3.828	1.437	0.234	0.515
uf20_016	1.140	0.218	0.015	0.010
uf20_018	1.312	15.218	0.014	0.015
aim-50-1_6-yes1-1	—	315.570	103.520	47.406
aim-50-1_6-yes1-2	3.031	20.462	6.297	2.750

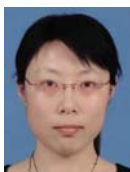
4 结 语

基于扩展规则的定理机器证明方法是与归结原理对偶的方法,现已成功地应用于知识编译^[13,14].IER算法是基于扩展规则的高效算法,但算法中用于限定搜索空间的子句C的选取是随机的,这使得算法没有很好地发挥扩展规则方法的优越性.为了进一步提高算法的效率,使得扩展规则算法更好地用于定理机器证明,本文在算法IER选取子句C的过程中分别加入启发式策略IMOM和IBOHM,得到算法IMOMH_IER和IBOHMH_IER.其中,IMOM策略中优先选择较长子句文字的否定文字,IBOHM策略优先满足较短子句,根据这两个策略选择的子句C比IER算法中随机选择子句C所限定的空间更适合扩展规则方法.实验结果表明,对不同规模和难度的问题,算法IMOMH_IER和IBOHMH_IER在效率上约是算法DR和IER的 10~200 倍.

References:

- [1] Fenkam P, Jazayeri M, Reif G. On methodologies for constructing correct event-based applications. In: Carzaniga A, Fenkam P, eds. Proc. of the 3rd Int'l Workshop on Distributed Event-Based Systems. Oakland: IEEE Computer Press, 2004. 38–43.
- [2] Kubica J, Rieffel EG. Collaborating with a genetic programming system to generate modular robotic code. In: Jangdon WB, et al. eds. Proc. of the Genetic and Evolutionary Computation Conf. New York: Morgan Kaufmann, 2002. 804–811.
- [3] Robinson JA. A machine oriented logic based on the resolution principle. Journal of the ACM, 1965,12(1):23–41.
- [4] Moser M, Ibens O, Ietz R, Steinbach J, Goller C, Schumann J, Mayr K. SETHEO and E-SETHEO-the CADE-13 systems. Journal of Automated Reasoning, 1997,18(2):237–246.
- [5] Dyckhoff R, Pinto L. Cut-Elimination and a permutation-free sequent calculus for intuitionistic logic. Studia Logica, 1998,60(12): 107–118.
- [6] Polakow J, Pfenning F. Relating natural deduction and sequent calculus for intuitionistic non-commutative linear logic. Electronic Notes in Theoretical Computer Science, 2004,20(18):1–18.
- [7] Lin H, Sun JG, Zhang YM. Theorem proving based on extension rule. Journal of Automated Reasoning, 2003,31(1):11–21.
- [8] Wu X, Sun JG, Lu S, Li Y, Meng W. Improved propositional extension rule. In: Wang GY, Peters JF, Skowron A, eds. Proc. of the 1st Int'l Conf. on Rough Sets and Knowledge Technology. New York: Springer-Verlag, 2006. 592–597.
- [9] Silva JPM. The impact of branching heuristics in propositional satisfiability algorithms. In: Proc. of the 9th Portuguese Conf. on Artificial Intelligence. New York: Springer-Verlag, 1999. 62–74.
- [10] Freeman JW. Improvements to propositional satisfiability search algorithms [Ph.D. Thesis]. Philadelphia: University of Pennsylvania, 1995.
- [11] Buro M, Kleine-Buning H. Report on a SAT competition. Technical Report, University of Paderborn, 1992.
- [12] Dechter R, Rish I. Directional resolution: The Davis-Putnam procedure, revisited. In: Doyle J, Sandewall E, Torasso P, eds. Proc. of the 4th Int'l Conf. on Principles of KR&R. Bonn: Morgan Kaufmann Publishers, 1994. 134–145.
- [13] Lin H, Sun JG. Knowledge compilation using the extension rule. Journal of Automated Reasoning, 2004,32(2):93–102.

[14] Murray NV, Rosenthal E. Duality in knowledge compilation techniques. LNAI 3488, 2005. 182–190.



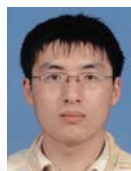
李莹(1985—),女,内蒙古兴安盟人,硕士生,主要研究领域为自动推理.



吴瑕(1976—),女,博士,讲师,主要研究领域为自动推理.



孙吉贵(1962—2008),男,博士,教授,博士生导师,主要研究领域为人工智能.



朱兴军(1983—),男,硕士生,主要研究领域为约束程序.

www.jos.org.cn

www.jos.org.cn