文章编号:1001 - 9081(2009)11 - 3064 - 04

一阶子句搜索方法

郭远华,曾振柄
(华东师范大学 上海市高可信计算重点实验室,上海 200062)
(yhguo@ sei. ecnu. edu. cn; gyhua2003@ 126. com)

摘 要:子句集的可满足性判定是自动证明领域的热点之一。提出了子句搜索方法判定命题子句集 Φ 的可满足性,该方法查找 Φ 中子句的一个公共不可扩展子句 C,当且仅当找到 C 时 Φ 可满足,此时 C 中各文字的补构成一个模型。结合部分实例化方法将子句搜索方法提升至一阶。一阶子句搜索方法可以判定子句集的 M 可满足性,具备终止性、正确性和完备性,是一种判定子句集可满足性的有效方法。

关键词:一阶逻辑;自动证明;可满足性;子句搜索方法;部分实例化方法
中图分类号: TP181 **文献标志码**:A

Clause searching method in first-order logic

GUO Yuan-hua, ZENG Zhen-bing
(*Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai* 200062, *China*)

Abstract: Deciding satisfiability of clause set is one of the active research topics in the automated reasoning field. A clause searching method of deciding satisfiability of propositional clause set Φ was proposed. This method first searched one clause C which cannot be extended from all clauses in Φ, if and only if C exists Φ was satisfied and the negative of C was one model. The authors updated clause searching method to first-order by partial instantiation method. Clause searching method in first-order logic can decide M satisfiablility of clause set and is of terminating, sound and complete property. It is a valid method for deciding satisfiability of clause set.

Key words: first-order logic; automated reasoning; satisfiability; clause searching method; partial instantiation method

0 引言

自动推理可以分为基于代数的几何定理证明与基于逻辑的自动推理两个范畴。前一个范畴的自动证明方法有我国学者吴文俊的建立的“吴方法”和张景中提出的面积法、消点法[1]。后一个范畴的主要方法有:归结方法[2]、tableau 方法[3]和相继式方法[4]等。归结方法由 Robinson 于 1965 年提出,其本质思想是检查子句集 Φ 是否可推导,如果可推导则 Φ 不可满足。tableau 方法的本质思想是将语义结构中的二元关系显示地表现出来。相继式方法把某些非常简单的相继式作为公理,有一些规则用来从旧的相继式得到新的相继式。文献[5]中提出了扩展规则方法,其思想是计算子句集 Φ 扩展出的极大项的数量 N,若 $N < 2^n$(n 为 Φ 中的原子数量)则 Φ 可满足。但以上逻辑自动证明方法没能给出原命题或子句集的一个模型,本文提出的子句搜索方法在判定子句集 Φ 的可满足性的同时还给出了一个模型,结合部分实例化方法,又将该方法提升至一阶。

1 子句搜索方法

1.1 预备知识

在自动推理中,要证明一个定理是 M 有效的,一般来说,该定理可以看成:

$$C_1 \wedge C_2 \wedge \cdots \wedge C_m \rightarrow B \tag{1}$$

自动推理不去证明式(1)的 M 有效性,而是反证式(2)的 M 不可满足性:

$$C_1 \wedge C_2 \wedge \cdots \wedge C_m \wedge \neg B \tag{2}$$

因此文中的子句搜索方法用于判断式(2)的不可满足性。

定义 1 设 A 是原子,两个文字 A 和 $\neg A$ 都是另外一个的补,集合 $\{A, \neg A\}$ 称为一个互补对。

一个子句说是一个 Tautology,如果此子句中含有一个互补对。

定义 2 给定一个子句 C 和一个原子的集合 AT:$C' = \{C \vee a, C \vee \neg a \mid a \notin AT$ 并且 a 和 $\neg a$ 都不在 C 中出现$\}$,把从 C 到 C' 中元素的推导叫作扩展规则[5],C' 中的元素叫作应用扩展规则的结果。

定理 1 子句 C 和它扩展后的结果 C' 是等价的。

证明 C' 一步归结演绎得 C;又 C 蕴涵 C' 中的 2 个子句,因此 C 可演绎得 C'。 证毕。

定义 3 一个非重言式子句是集合 AT 上的极大项当且仅当它包含集合 AT 上的所有原子或其否定。

定理 2 给定一个子句集 Φ,它其中所有原子的集合是 $AT(|AT| = n)$,若 Φ 中的子句都是 AT 上的极大项,则子句集 Φ 不可满足当且仅当 AT 中含有 2^n 个互相不同的子句。

证明 充分性:对 m 使用数学归纳法。当 $m = 1$ 时,不失一般性,假设子句集包含 2 个极大项 $\{A, \neg A\}$,显然子句集不可满足。

假设 $m = i - 1$ 时,定理的充分性成立,我们将证明当 $m = i$ 时,充分性仍成立。对含有 2^m 个极大项的子句集 Φ_m 中任意一原子使用分裂规则,得到含有 $i - 1$ 个原子和 2^{i-1} 个极大项的 2 个子句集合,根据前面的假设这 2 个子句集合都不可满

收稿日期:2009 - 05 - 08;**修回日期**:2009 - 07 - 28。 **基金项目**:国家自然科学基金资助项目(90718041)。
作者简介:郭远华(1978 -),男,湖北天门人,博士研究生,主要研究方向:计算机自动推理; 曾振柄(1963 -),男,甘肃皋兰人,教授,博士生导师,主要研究方向:计算机自动推理。

足,即 Φ_m 不可满足。

必要性:如果 Φ 缺少任一个极大项,极大项中各文字的补是 Φ 的一个模型,即 Φ 缺失极大项,它就是可满足的。

证毕。

定理 3　若子句 C 的形式为 $A_1 \vee A_2$,它不能扩展的一个子句集为 $\{\neg A_1 \vee A_2, \neg A_2\}$。

证明　首先,由前面扩展规则的定义,$\neg A_1 \vee A_2$,$\neg A_2$ 显然不能由 $A_1 \vee A_2$ 扩展;其次,根据归结原理,容易证明集合 $\{A_1 \vee A_2, \neg A_1 \vee A_2, \neg A_2\}$ 是不可满足的,即该集合可以扩展出所有的极大项。　证毕。

根据定理 3,若子句 C 的形式为 $A_1 \vee A_2 \vee \cdots \vee A_n$,则它的一个不可扩展子句集为 $\{\neg A_1 \vee A_2 \vee \cdots \vee A_n, \neg A_2 \vee \cdots \vee A_n, \cdots, \neg A_n\}$。在下文中,用数组元素 $CurIndexofClause[j]$ 指向第 j 个子句 $A_1A_2\cdots A_i\cdots A_n$ 中的某个文字,当指向 A_i 时,表示取当前子句的不可扩展子句为 $\neg A_i\cdots A_n$。

定义 4　两个子句 C_1、C_2 的交子句是指这样的子句 C_3:C_3 扩展的极大项集合就是 C_1、C_2 扩展的极大项集合的交集。

定义 5　两个子句集 Φ_1、Φ_2 的交子句集合是指 Φ_1 中子句与 Φ_2 中子句的交子句的集合。

如子句 $A \vee B$ 与子句 $B \vee C$ 的交子句为 $A \vee B \vee C$。子句集 $\{A \vee B\}$ 与 $\{B \vee C\}$ 的交子句集合为 $\{A \vee B \vee C\}$。根据定义 2、3 和 4 不难得到定理 4。

定理 4　子句 C_1、C_2 的文字组成集合 W,若 W 中存在互补对,则 C_1、C_2 的交子句为空;若 W 不存在互补对,则 W 中文字的析取为 C_1、C_2 的交子句。

定理 4 给出了求两子句交子句的方法。

1.2　数据结构

子句搜索方法要判断子句集 $\Phi = \{C_1, C_2, \cdots, C_m\}$ 是否存在一个不可扩展子句 C,根据定理 2,若存在则 Φ 可满足且 C 中各文字的补构成 Φ 的一个模型;否则不可满足。Φ 的任意子句 $C_i(1 \leqslant i \leqslant m)$ 的不可扩展子句有 $C_{i1}, C_{i2}, \cdots, C_{in_i}$($C_i$ 的文字数记为 n_i),它们构成 C_i 的不可扩展子句集 Φ_{ci}。求解 $\Phi_{c1}, \Phi_{c2}, \cdots, \Phi_{cm}$ 的交子句集合中的一个交子句 C,若 C 存在则 Φ 可满足,否则不可满足。直观上理解,子句搜索算法在每个 C_i 的 n_i 个不可扩展子句中选出一个 $C_{ij}(1 \leqslant j \leqslant n_i)$,在 m 个 C_{ij} 间求解交子句 C。

用 $1, \cdots, n$ 分别表示 Φ 中的 n 个原子,用 $-n, \cdots, -1$ 表示原子的否定。Φ 中的任何一个文字都可以用 $-n, \cdots, n$(0 除外)之间的一个整数来表示,互补对中的 2 个文字对应的数字之和为 0。

ClauseList:一维动态数组,存储子句集 Φ,同一子句的文字在数组中是相邻的。

IndexRgofClause:一维动态数组,存储各子句在 ClauseList 中的终止下标,第 i 个子句的终止下标用 IndexRgofClause[i] 表示,IndexRgofClause[0] 赋值为 -1。每个子句 i 的起始下标就是 IndexRgofClause[$i-1$] + 1,终止下标是 IndexRgofClause[i]。

CurIndexofClause:一维动态数组,值为 ClauseList 中的下标。CurIndexofClause[i] 存储第 i 个子句某文字的下标,初值为第 i 个子句的第一个文字在 ClauseList 中的下标,其取值范围是[IndexRgofClause[$i-1$] + 1, IndexRgofClause[i]],显然 CurIndexofClause[1] 为 0。CurIndexofClause[i] 的每一个值对应 C_i 的一个不可扩展子句。如果第 i 个子句为 $A \vee B$,若 CurIndexofClause[i] 为 A 的下标,说明当前取第 i 个子句的不可扩展子句 $\neg A \vee B$,若 CurIndexofClause[i] 为 B 的下标,说明当前取第 i 个子句的不可扩展子句 $\neg B$。CurIndexofClause[0] 为 -1,仅起填充作用。

WordSet:一维动态数组,存储各子句的当前不可扩展子句。数组 IndexofWordSet 指向 WordSet 中各子句的结束位置。

对于子句 i 的不可扩展子句集,算法根据 CurIndexofClaus[i] 依次取其中的一个与其他子句的不可扩展子句集求交子句。所谓子句的当前不可扩展子句是指算法在该子句的不可扩展子句集当前选取的子句。WordSet 记录前面 i 个子句的 i 个不可扩展子句,第 $i+1$ 个子句的不可扩展子句将与 WordSet 中的子句求交子句,若失败则需要将第 i 个子句的当前不可扩展子句从 WordSet 中清除,将第 i 个子句的下一个不可扩展子句加入 WordSet。

1.3　算法描述

算法 1

1) 置 i = 1(代表当前要处理第 i 个子句)。

2) 判断 i 是否大于子句的数量:若大于,返回可满足。

3) 判断 CurIndexofClause[i] 是否大于 IndexRgofClause[i]:若大于,置 CurIndexofClause[i] 为 IndexRgofClause[$i-1$] 加 1,当 i 等于 1,返回不可满足,当 i 大于 1,CurIndexofClause[$i-1$] 加 1,删除 WordSet 中 IndexofWordSet[$i-2$] 指示的下标之后的所有元素,删除 IndexofWordSet[$i-1$],i 减 1,重复 3)。

4) 根据 CurIndexofClause[i],得到对应的第 i 个子句的不可扩展子句 C_i,检查 C_i 中的第一个文字是否与 WordSet 的任一文字互补:若存在互补,则将 CurIndexofClause[i] 加 1,跳转 3)。

5) 依次检查 C_i 中从第二个文字开始的每一个文字 W 是否与 WordSet 的任一文字互补:若存在互补,CurIndexofClause[i] 指向 W 在第 i 个子句中对应的文字,得到以 $\neg W$ 为第一个文字的新的 C_i,重复 5)。

6) 将 C_i 中各文字加入 WordSet,IndexofWordSet[i] 指向 WordSet 的尾元素,i 加 1,跳转 2)。

以上各步执行过程中,若没有遭遇“跳转”或“重复”,默认的是执行下一步。IndexRgofClause[$i-1$] 指向第 $i-1$ 个子句的尾文字,所以“置 CurIndexofClause[i] 为 IndexRgofClause[$i-1$] 加 1”就初始化了 CurIndexofClause[i],让其指向第 i 个子句的第一个文字。CurIndexofClause[i] 加 1 的目的是取第 i 个子句的下一个不可满足子句。i 的增减控制算法处理下一个子句或上一个子句。

推理算法的目的是找到各子句都不可扩展的子句。若找到,返回子句集可满足;找不到,则不可满足。若算法返回可满足,CurIndexofClause 指向的各文字组成了子句集的一个模型。对于给定的子句集,各子句的不可扩展子句是有限的,因此算法是可终止的。定理 1 和定理 2 确保了算法的正确性和完备性。

设子句集 $\Phi = \{C_1, C_2, \cdots, C_m\}$ 含有 m 个子句,n 个原子。C_i 中的有效文字个数为 NumOfWord[i],C_i 的不可扩展子句集为 Φ_i,显然 Φ_i 中元素数量也为 NumOfWord[i],i = 1,2,$\cdots$,m。根据定理 2,可以逐个判断 2^n 个极大项是否可由 Φ 扩展,若存在一个不能扩展的极大项,则 Φ 可满足,否则不可满足。显然,这种穷举法算法复杂度为 2^n。文中算法有针对性的在 Φ_i 间查找交子句,因此时间复杂度相对穷举法低。

利用乘法原理在 Φ_i 之间查找交子句的时间复杂度为 $\prod_{i=1}^{m}$ NumOfWord[i]。本文算法在查找过程中进行了大量删除

处理,例如 Φ_k 中的当前子句(CurIndexofClause[k] 指向的)与前面的 $\Phi_1,\cdots,\Phi_{k-1}$ 中的当前子句存在文字互补,则挑选 Φ_k 中下一个子句,相对于乘法原理,减少了 $\prod_{i=k+1}^{m}$NumOfWord[i] 次比较。

2 一阶子句搜索方法

利用部分实例化方法[6],本文将子句搜索法提升至一阶。部分实例化方法的基本思想是将一阶问题转化为一系列命题逻辑中的可满足问题。Hooker 将该方法扩展到能处理带函数的一阶逻辑,文献[7]中则修正了 Hooker 的阻塞定义使得该方法具备完备性。下面先给出部分实例化方法的相关基础知识。

定义6 F、F' 是两个谓词公式,称 F 是 F' 的变体,如果存在一个改名替换合一 σ 使得 $F = F'\sigma$。

这里给出的变体定义与标准变体的定义不同,标准变体定义为:公式 F,F' 是变体,如果对于某对替换(σ,θ) 有 $E = F\sigma$ 并且 $F = E\theta$。因此虽然 $P(x,y)$ 和 $P(x,x)$ 不是标准意义上的变体,但在本文中是变体。

定义7 F 是已经消去量词的公式,令 S 是从 F 中的每一个子句 C 到 C 中原子 $S(C)$ 的一个映射。$L(C)$ 是 C 中包含在 $S(C)$ 中的文字,S 是一个满足式映射,若对一个真值解释 ν,子句 C 对应的 $L(C)$ 在 ν 下都为真,则把 $S(C)$ 称作 C 的满足式。如果 $L(C)$ 是 $S(C)$,$S(C)$ 称作真满足式,如果 $L(C)$ 是 $\neg S(C)$,$S(C)$ 称作假满足式。

定义8 给定已消去量词的公式 F 的满足式映射 S,满足式对 $P(t)$,$P(t')$ 被阻塞,如果下列条件成立:

1)$P(t)$ 是真满足式;

2)$P(t')$ 是假满足式;

3)$P(t)$ 和 $P(t')$ 有一个最一般的合一(σ,τ),使得 $P(t)\sigma = P(t')\tau$;

4)$P(t)$ 和 $P(t')$ 分别是 F 中两个子句 C,C' 中的满足式,它们满足:①$C\sigma$ 和 F 中任意子句不为变体;或②$C'\tau$ 和 F 中任意子句不为变体。

称一个满足式被阻塞如果它在某个被阻塞的满足式对中,满足式映射被阻塞如果它的某个满足式被阻塞。

例如,给定一个子句集 $\Phi = \{P(x) \vee Q(a),\neg P(b) \vee R(c)\}$,模型 $\{P(x),\neg P(b),\neg Q(a),\neg R(c)\}$ 对应的满足式映射被阻塞,因为 $P(x)$,$\neg P(b)$ 被阻塞。

定义9 给定一个消去量词的公式 F 和一个满足式映射 S,满足式对 $P(t)$ 和 $P(t')$ 被 M-阻塞,如果它们被阻塞并且其最一般合一(σ,τ),使得 $P(t)\sigma$ 和 $P(t')\tau$ 中不含深度大于 M 的项。一个满足式被 M-阻塞,如果它在某个被 M-阻塞的满足式对中。满足式映射 S 被 M-阻塞,如果它其中的某个满足式是 M-阻塞的。

定理4 给定公式 $F = \forall x_1 C_1 \wedge \cdots \wedge \forall x_m C_m$,$S$ 是消去量词的公式 $F' = C_1 \wedge \cdots \wedge C_m$ 的一个满足式映射,则:

1) 如果 S 不被 M 阻塞,则 F 一定是 M 可满足的;

2) 如果 S 不被阻塞,则 F 一定可满足。

该定理的证明可以参见文献[6],下面给出部分实例化算法。

算法2 部分实例化算法 PPI

令公式 $F = \forall x_1 C_1 \wedge \cdots \wedge \forall x_m C_m$ 为一个一阶逻辑公式。

1) 初始化:令 $F_0 = C_1 \wedge \cdots \wedge C_m$,$k = 0$,$M = 0$。

2) 基可满足:找出 F_k 的满足式映射 S,在 F_k 中把和同一个原子存在改名替换的所有原子都看成是相同原子。

3)终止性检查:

If S 不存在, return 0

Else If S 不被阻塞, return 1

Else If S 不是 M 阻塞, F 是 M 可满足的. 令 $M = M + 1$, 转3)

4) 实例化:(S 被 M-阻塞) 令 C_h 和 C_i 是 F_k 中一对被 M-阻塞的子句对,(σ,τ) 是 $P(t)$ 和 $P(t')$ 的 MGU。令 $F_{k+1} = F_k \wedge C_h\sigma \wedge C_i\tau$,并进行标准化处理,$k = k + 1$,转2)。

PPI 算法的终止性、正确性、完备性参见文献[6]。PPI 算法的步骤2)采用子句搜索方法判定 F_k 的可满足性,并给出满足式映射得到的算法就是一阶子句搜索(First-Order Clause Searching, FOCS)算法。

3 应用算例

下文中形如 $m@n$ 的字符串表示第 m 个子句中的第 n 个变量,把简单析取范式 $A_1 \vee A_2 \vee \cdots$ 写成 $\{A_1,A_2,\cdots\}$ 的形式。

例1 人都是要吃饭的,秦始皇是人。因此,秦始皇是要吃饭的。

用如下谓词:

$H(x)$:x 是人;$E(x)$:x 要吃饭。其中 x 为变量,a 为常量,代表秦始皇。

则有:

人都是要吃饭的:$\forall x(H(x) \rightarrow E(x))$

秦始皇是人:$H(a)$

秦始皇是要吃饭的:$E(a)$

将上述逻辑公式转化为子句形式:

$\forall x(H(x) \rightarrow E(x)) = \forall x(\neg H(x) \vee E(x))$

即等价于子句 $\neg H(x) \vee E(x)$。

用反驳(或否证)法来证明结论,即假设秦始皇不吃饭:$\neg E(a)$。对此问题有如下三个子句:1)$\{\neg H(x),E(x)\}$;2)$\{H(a)\}$;3)$\{\neg E(a)\}$。当阻塞深度为0时,FOCS 算法返回 false,此时的子句集为:$\{\neg H(x),E(x)\}$,$\{H(a)\}$,$\{\neg E(a)\}$,$\{\neg H(a),E(a)\}$。第4个子句由部分实例化算法添加。因此我们证明了结论:秦始皇是要吃饭的。

例2 著名 Steamroller 问题[8] 的简化版本。1) 鸟与蛇都是动物;2) 有一些鸟与蛇;3) 有一些植物;4) 每一种动物都喜欢吃所有那些不喜欢吃所有植物的动物;5) 鸟不喜欢吃蛇;6) 因此,一定有一种喜欢吃所有植物的动物。

用如下谓词:

$A(x)$:x 是动物;$S(x)$:x 是蛇;$B(x)$:x 是鸟;$P(x)$:x 是植物;$E(x,y)$:x 喜欢吃 y。

则有:

鸟是动物:$\forall x(B(x) \rightarrow A(x))$

蛇是动物:$\forall x(S(x) \rightarrow A(x))$

有一些鸟与蛇:$B(b)$,$S(s)$

有一些植物:$P(p)$

每一种动物都喜欢吃所有那些不喜欢吃所有植物的动物:

$(\forall x)(\forall y)(\forall z)((A(x) \wedge A(y) \wedge P(z) \wedge \neg E(y,z)) \rightarrow E(x,y))$

鸟不喜欢吃蛇:

$(\forall x)((\forall y)(((B(x) \wedge S(y)) \rightarrow (\neg E(x,y)))))$

一定有一种喜欢吃所有植物的动物:

$(\exists x)(A(x) \wedge (\forall y)(P(y) \rightarrow E(x,y)))$

上述逻辑公式中,x,y,z 是变量符号,b,s,p 是常量符号。这个问题的经典一阶逻辑的子句集如下(经过变量重命名):

$\{\neg S(x_1), A(x_1)\}$

$\{\neg B(x_2), A(x_2)\}$

$\{B(b)\}$

$\{S(s)\}$

$\{P(p)\}$

$\{\neg B(x_3), \neg S(y_3), \neg E(x_3, y_3)\}$

$\{\neg A(x_4), \neg A(y_4), \neg P(z_4), E(x_4, y_4)\}$

$\{\neg A(x_5), P(f(x_5))\}$

$\{\neg A(x_6), \neg E(x_6, f(x_6))\}$

当阻塞深度为 1 时,FOCS 算法返回 false,此时增加的子句为:

$\{A(b), \neg B(b)\}$

$\{\neg S(s), A(s)\}$

$\{\neg S(12@1), \neg B(b), \neg E(b, 12@1)\}$

$\{\neg A(b), P(f(b))\}$

$\{\neg A(b), \neg E(b, f(b))\}$

$\{\neg A(s), P(f(s))\}$

$\{\neg A(s), \neg E(s, f(s))\}$

$\{\neg S(s), \neg B(b), \neg E(b, s)\}$

$\{E(f(b), 18@1), \neg A(b), \neg P(18@1), E(b, f(b)), \neg A(f(b))\}$

$\{E(f(b), 19@1), \neg A(s), \neg P(19@1), E(s, f(s)), \neg A(f(s))\}$

$\{\neg A(b), \neg A(s), \neg P(20@1), E(s, 20@1), E(b, s)\}$

$\{\neg A(b), \neg A(s), E(b, s), \neg P(f(s)), E(s, f(s))\}$

所以结论是:一定有一种喜欢吃所有植物的动物。

4　结语

本文提出的子句搜索方法具有可终止性、正确性和完备性,在查找子句的过程中排除了大量的子句,其效率远优于运用穷举法和乘法原理查找子句。与常用的归结方法相比,该方法不仅判定了子句集的可满足性,还给出了一个模型。结合部分实例化方法将子句搜索方法提升至一阶,一阶子句搜索方法也具备可终止性、正确性和完备性。本文实现的一阶子句搜索算法(FOCS)较好地给出了部分一阶逻辑实例的证明。FOCS 算法还可以利用 DP 算法做一些前期的优化工作,进一步提高自动证明效率。

参考文献:

[1] 罗慧敏. 基于消点法的几何自动推理系统实现[J]. 计算机应用, 2008, 28(11): 2984 - 2986.

[2] ROBINSON J A. A machine-oriented logic based on the resolution principle [J]. Journal of the ACM, 1965, 12(1): 23 - 41.

[3] SMULLYAN R M. First-order logic [M]. Berlin: Springer-Verlag, 1994.

[4] FITTING M. First-order logic and automated theorem proving [M]. 2nd ed. Berlin: Springer-Verlag, 1996.

[5] LIN HAI, SUN JI-GUI, ZHANG YI-MIN. Theorem proving based on extension rule [J]. Journal of Automated Reasoning, 2003, 31(1): 11 - 21.

[6] HOOKER J N, RAGO G, CHANDRU V, *et al.* Partial instantiation methods for inference in first-order logic [J]. Journal of Automated Reasoning, 2002, 28(4): 371 - 396.

[7] 吴霞. 基于扩展规则的定理证明的研究[D]. 长春: 吉林大学, 2006.

[8] WALTHER C. A mechanical solution of Schubert's streamroller by many-sorted resolution [J]. Aritificial Intelligence, 1986, 26(2): 217 - 224.

(上接第 3063 页)

[5] MONTEMERLO M, THRUN S, KOLLER D, *et al.* FastSLAM: A factored solution to the simultaneous localization and mapping problem [C]// Proceeding of the 18th National Conference on Artificial Intelligence. Menlo Park, CA, USA: AAAI Press, 2002: 593 - 598.

[6] MONTEMERLO M, THRUN S, KOLLER D, *et al.* FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges [C]// IJCAI-2003: Proceedings of the International Joint Conference on Artificial Intelligence. Acapulco, Mexico: Morgan Kaufmann, 2003: 1151 - 1156.

[7] JULIER S J, UHLMANN J K. Unscented filtering and nonlinear estimation [J]. Proceedings of the IEEE, 2004, 92(3): 401 - 422.

[8] BAILEY T, NIETO J, NEBOT E. Consistency of the FastSLAM algorithm [C]// Proeeedings of the IEEE International Conference on Robotics and Automation. Washington, DC: IEEE Press, 2006: 424 - 429.

[9] 张莉莉, 蔡自兴, 陈白帆. 基于相对观测量的机器人合作 FastSLAM 算法[J]. 华中科技大学学报: 自然科学版, 2008, 36(S1): 171 - 173.

[10] KIM C, SAKTHIVEL R, CHUNG W K. Unscented FastSLAM: A robust and efficient solution to the SLAM problem [J]. IEEE Transactions on Robotics, 2008, 24(4): 808 - 820.

[11] van der MERWE R, de FREITAS N, DOUCET A. The unscented particle filter [EB/OL]. [2009 - 04 - 12]. http://cslu.cse.ogi.edu/publications/ps/UPF_CSLU_talk.pdf.

[12] JULIER S J, UHLMANN J K. A new extension of the Kalman filter to nonlinear systems [C]// Proceedings of Signal Processing, Sensor Fusion, and Target Recognition, SPIE 3068. Bellingham, WA: Society of Photo-Optical Instrumentation Engineers, 1997: 182 - 193.

[13] 周武, 赵春霞. 一种基于遗传算法的 FastSLAM2.0 算法[J]. 机器人, 2009, 31(1): 25 - 32.

[14] SHI YONG, HAN CHONG-ZHAO. The divided difference particle filter [C]// 10th International Conference on Information Fusion. Washington, DC: IEEE Press, 2007: 1 - 7.

[15] NORGAARD M, POULSEN N K, RAVN O. New developments in state estimation for nonlinear system [J]. Automatica, 2000, 36(11): 1627 - 1638.

[16] SAULSON B G, CHANG K-C. Nonlinear estimation comparison for ballistic missile tracking [J]. Optical Engineering, 2004, 43(6): 1424 - 1438.

[17] LIU J S. Metropolized independent sampling with comparisions to rejection sampling and importance sampling [J]. Statistics and Computing, 1996, 6(2): 113 - 119.

[18] THRUN S, BURGARD W, FOX D. Probabilistic robotics [M]. London: MIT Press, 2005: 189 - 279.