

# 搜索算法的逻辑分析<sup>\*</sup>

张玉平

(北京航空航天大学计算机科学与工程系 北京 100083)

**摘要** 搜索算法的初始空间、搜索策略、搜索过程可以用一阶语言描述,搜索算法的逻辑性质由初始状态空间确定.这意味着描述搜索过程的逻辑具有紧致性,初始状态的初等类具有有限封闭性.

**关键词** 搜索算法,一阶逻辑,模型.

**中图分类号** TP18

搜索算法是问题求解时常用的方法.选定初始状态,应用一些搜索策略、算法,经过一段时间的搜索,则有望得到适合需求的解答.在搜索过程中,选取初始状态之后,应用策略生成新的状态,能够判断某个问题是否成立.这是计算机科学中求解实用问题的习惯模式<sup>[1]</sup>,在选定求解策略的基础上,构造出了很多快速的实用算法.本文研究此类算法的模型论性质,指出搜索的初始空间与所求解问题之间的逻辑联系,下面首先分析搜索过程与逻辑推理、逻辑可满足关系之间的联系,指出在搜索过程中其实包含了一阶逻辑中的3个基本概念:可推导关系、可满足关系、模型类.在此基础上,给出了搜索求解的问题与初始空间之间的逻辑联系,在计算机科学中应用搜索技术时,可以根据问题的逻辑性质设计出高效的算法,本文的目的在于分析搜索过程的逻辑性质.

在搜索算法中,判断某个状态是否满足某个问题相当于判定某个模型是否满足( $\models$ )一个语句,这是因为在构造问题模型、初始状态的表述、问题描述时,需要基于一种描述语言.事实上,设计算法的过程中采用的技术都可以用二阶语言描述,而二阶语言所描述的对象在原理上又可以转换为二阶语言  $L$  所描述的一些对象.搜索策略转化为判断状态模型是否满足给定问题的一种技术,搜索时的初始状态成为语言  $L$  的模型,希望通过搜索来求解的问题即是语言  $L$  的语句.

事实上,通过搜索解决问题的过程相当于一种满足关系  $\models$ ,选定初始状态  $v$ ,经过使用搜索策略可以解决问题  $A$ ,即意味着  $v \models A$ .新的满足关系  $\models$ ,不同于经典逻辑中的满足关系  $\models$ ,但是当问题经过适当变换之后,则  $v \models A$  等同于  $\sigma(v) \models \sigma(A)$ ,其中  $\sigma$  是语句的形式转换、同时用一阶逻辑的模型表示初始状态.给定两个初始状态  $v, w$ ,称  $v$  导出  $w$ ,若以  $w$  为初始状态可以搜索到的结论以  $v$  为初始状态也可以搜索到.记此关系为  $v \vdash w$ ,这种标记在以下的讨论中不会导致混淆.  $\sigma$  把每个需要搜索才能得以判定的语句用描述此搜索过程的一阶语句表示.这时,假设每个语句都有固定的搜索路径,因而转换映射是良定义的,它描述的是求解问题  $A$  时的搜索路径.  $\sigma$  与“并且”运算没有可交换性质,即  $\sigma(A \wedge B)$  与  $\sigma(A) \wedge \sigma(B)$  之间一般没有必然的联系,这意味着求解  $A \wedge B$  时的路径与分别求解  $A$  及  $B$  的路径一般没有联系.在第1节中以 GSAT(greedy satisfiability)<sup>[2]</sup> 为例说明如何构造转换映射  $\sigma$ ,为  $\sigma$  的性质提供例证.从给定的初始状态开始,经过使用搜索策略,得出对某个语句真假性的判断.当选定一类初始状态  $\mathcal{S}$ ,则可以依据搜索过程的语义表示( $\models, \vdash$ )定义语法关系  $\vdash$ ,  $\vdash A$  意味着对于  $\mathcal{S}$  中的每个模型  $v, \sigma(v) \models \sigma(A)$  成立.  $\models$  与  $\vdash$  之间的联系来自于经典逻辑中  $\models$  与  $\vdash$  之间的联系.本文基于上述事实,给出了刻画  $\models$  性质的定理,以实现搜索过程的逻辑描述.

施用搜索算法时的初始状态一般按照一定的模式选定,比如在 GSAT 算法<sup>[2]</sup>中,判定语句可满足性时的初始状态是随机选择的,因而所使用的初始状态在一个随机的赋值类中.搜索是一种“机械”的操作过程,同时逻辑

\* 本文研究得到国家自然科学基金和航空高等院校自选科研课题基金资助.作者张玉平,1966年生,博士后,副教授,主要研究领域为应用于计算机科学的逻辑理论.

本文通讯联系人:张玉平,北京 100083,北京航空航天大学计算机科学与工程系

本文 1997-11-10 收到原稿,1998-03-09 收到修改稿

推导也具有类似的性质. 经过转换的搜索过程与逻辑推导过程有一致性, 在此基础上, 本文把搜索过程等同于推导过程, 讨论搜索算法的逻辑性质. 因为初始状态集合  $\mathcal{S}$  是个模型集合, 若它是个初等类<sup>[3]</sup>, 即存在语句集合  $T$ , 使得  $\mathcal{S}$  是  $T$  的所有模型的集合, 则此时对任意的语句  $A$ ,  $\vdash A$  当且仅当  $T \vdash \sigma(A)$ . 因而用搜索得出的结论等同于用一阶逻辑推理得出的结论, 推导关系  $\vdash$  具有紧致性质. 这意味着当一组目标的任意有限子目标集合都可以由某个初始状态搜索得到时, 本组目标可以由一个初始状态搜索得到. 这时, 在搜索过程中采用的策略实质上都将没有必要. 所以, 为了实现搜索的目的, 需要使搜索所采用的初始状态类  $\mathcal{S}$  不能是一阶语句  $T$  的所有模型的集合, 即不能适合由一阶语言描述的规律. 因而, 需要讨论在什么情况下一个初始状态包含于由一阶语句描述的初始状态集合中, 这是本文将要实现的目的.

**定义(初始状态的初等类).** 假设  $\mathcal{S}$  是初始状态的一个集合,  $T$  是语言  $L$  的语句. 称  $\mathcal{S}$  由一阶理论  $T$  描述, 若  $\mathcal{S}$  满足两个条件: (a) 对  $\models$  封闭, 即当  $w \in \mathcal{S}, w \models v$  时,  $v \in \mathcal{S}$ . (b)  $\mathcal{S}$  是  $T$  的所有模型的集合. 即  $v \in \mathcal{S}$  当且仅当  $T \models \sigma(v)$ . 当  $\mathcal{S}$  满足上述条件时, 也称  $\mathcal{S}$  是一个初等类.

在不引起混淆时, 记  $T \models \sigma(v)$  为  $T \models v$ . 对初始状态集合封闭性的约定意味着: 当在  $\mathcal{S}$  中搜索出较多结论的初始状态时, 搜索出较少结论的初始状态也在  $\mathcal{S}$  中. 之所以采用上述约定, 是因为虽然两个初始状态可以相互推导, 但是它们作为语言  $L$  的模型并不一定初等等价<sup>[3]</sup>, 即当  $v \models w, w \models v$  时, 一般情况下  $v \equiv w$  不成立. 上述定义的初等类是逻辑中初等类的推广. 在逻辑中, 要求初等类对初等等价及超积封闭. 上述的性质(a)、(b)即是为了实现同样的目的.

**定理(初始状态初等类的有限封闭性质).** 假设初始状态集合  $\mathcal{S}$  由一阶理论  $T$  描述. 对任意的初始状态  $v$ , 若对此采用策略所得出的任意有限结论集合都由  $\mathcal{S}$  中的某个初始状态搜索得到, 则  $v$  被包含在  $\mathcal{S}$  中.

上述定理给出了判定初始状态被一阶理论描述的条件, 指出了一阶理论对初始状态集合的刻画能力. 当关系  $\models$  等同于关系  $\vdash$  时, 定理对一阶逻辑也成立. 事实上, 经典逻辑中成立更强的结论, 只需要“任意的结论”, 而不需要“任意有限的结论集合”的前提就可以得到相同的结论, 它是一阶逻辑中关于模型的初等类的刻画性质. 定理的证明正是推广了经典逻辑中关于此结论的证明技术, 这将在第 2 节中给出.

## 1 搜索算法的逻辑描述

搜索的目的是判定一些语句是否成真. 决定搜索对语句成真性的判定效率有两个因素: 搜索策略的选择与初始状态的设定. 当设定了一个初始状态之后, 搜索策略相当于对语句是否成真的一种判定, 这种判断是可操作的. 即给定一个语句, 算法在有限的时间内给出其真假性一个确切的答案. 对算法的这种约定类似于在逻辑中对“可满足”关系( $\models$ )的定义. 在逻辑中, 每个语句都是由一些基本的语法成分、在逻辑联结词的作用下生成的. 语句的目的是为了表示一种事实, 该语句其实是个字符串. 为了解释由基本的语法成分构成的字符串的含义, 需要定义模型这样的概念. 在模型中, 一些基本的语法成分的意义得到了解释, 但这并不意味着语句有了含义. 为了给出语句的含义, 需要指定一种“判定语句真假的算法”. 在逻辑中, 对语句真假性的判定(计算)采用了一种确定的方式, 即对公式长度归约的技术. 根据这种方法, 可以确定地给每个语句赋以真假值. 此方法在很多逻辑理论中被采用, 但是这并不是一种高效的算法. 比如在命题逻辑中, 若实现对语句恒真性的判定, 常规的方法具有指数时间的复杂度.

在定义了语义解释之后, 需要研究语义语法之间的联系: 即在一类语义解释中都成立的语法事实能否通过语法演算的形式得到. 谓词演算可以演算出在所有模型中都成真的语法事实. 其他形式的演算, 比如 Gentzen 证明理论、归结方法等都是为了实现这样的目的. 但是这类演算的缺点在于: 一般意义下, 对语句恒真、恒假的判断是不可判定的. 因而, 还没有一般有效的策略用来判定语句的真假性. 为此, 需要设计其他形式的算法, 以实现对真假性的比较有效的判定. 搜索算法就是为了实现这样的目的, 以达到纯粹用逻辑推导的方法无法快速得到的结论, 即期望  $\vdash$ . 不等同于  $\vdash$ , 以实现对恒真、恒假的有效判定. 从前面的讨论可知, 上述“等同关系”是否成立取决于所选择的初始状态集合.

对可满足性问题随机算法 GSAT<sup>[2]</sup>的讨论可以更加明确上述事实. GSAT 算法是指以下的算法, 它是求解可满足(SAT)问题的一种高效算法.

```

Input: 一个具有  $\wedge \vee$ -标准型的命题语句  $A$  及两个自然数  $\alpha$  和  $\beta$ 
for  $i := 1$  to  $\alpha$ 
   $v :=$  随机生成的赋值
  for  $j := 1$  to  $\beta$ 
    if  $v$  满足  $A$  then return  $v$ 
     $v := v[p]$ . /* 其中  $n(v[p]) = \max\{n(v[p]) : p \text{ 是在 } A \text{ 中出现的命题变元}\}$ .  $v[p]$  是改变  $v$  对命题变元  $p$  的赋值所得到的新赋值. 若  $w$  是一个赋值,  $n[w]$  是由  $w$  满足的  $A$  的子句个数. */
  end for
end for
return “未找到满足  $A$  的赋值”

```

下面讨论 GSAT 的逻辑描述. 为此, 定义仅有一些常量构成的一阶语言  $L = \{t, f, c_i : i \in \omega\}$  及该语言上的理论  $T = \{\forall x(x=t \vee x=f), \forall x \neg(x=t \wedge x=f)\}$ .

理论  $T$  描述关于真假取值的语义事实: 每个模型仅包含两个不同的元素, 它们分别是常量  $t, f$  的解释, 这两个特殊的常量用来表示真值和假值, 常量  $c_i$  用于表示命题变元  $p_i$  的取值. 理论  $T$  意味着每个命题变元或者取真值、或者取假值. 在此定义的基础上, 可以将 GSAT 算法中涉及到的赋值及命题语句分别表示为理论  $T$  的模型及语言  $L$  的语句. 为此, 定义满足以下条件的转换映射  $\sigma$ .

**定义(描述 GSAT 初始状态的转换映射  $\sigma$ ).**

- (1.1) 若  $v = \{v_i : 0 \leq i \leq n\}$  是 GSAT 算法中的一个赋值, 定义理论  $T$  的模型  $\sigma(v)$  满足条件: 若  $v_i = 1$ , 则  $c_i = t$ , 若  $v_i = 0$ , 则  $c_i = f$ .
- (2.1)  $\sigma(p_i) \equiv c_i - t$ ,
- (2.2)  $\sigma(\neg p_i) \equiv \neg(c_i = t)$ ,
- (2.3)  $\sigma(A \vee B) \equiv \sigma(A) \vee \sigma(B)$ ,
- (2.4)  $\sigma(A \wedge B) \equiv \sigma(A) \wedge \sigma(B)$ .

一般地, 给定  $p_1, \dots, p_n$  是  $n$  个命题变元, 若假定  $\sigma(p_i) \equiv \neg(c_i - t), \sigma(\neg p_i) \equiv (c_i = t)$ , 则可以定义类似的转换映射, 记为  $\sigma_{p_1, \dots, p_n}(\cdot)$ , 其中  $1 \leq i \leq n$ .

根据上述定义, 对任意的赋值  $v$  及语句  $A, v \models A$  当且仅当  $\sigma(v) \models \sigma(A), v[p_1, \dots, p_n] \models A$  当且仅当  $\sigma(v) \models \sigma_{p_1, \dots, p_n}(A)$ . 这里的  $v[p_1, \dots, p_n]$  的定义与 GSAT 中的类似, 表示同时改变赋值  $v$  对命题变元  $p_1, \dots, p_n$  的取值之后得到的赋值. 因而, 可以将赋值及其“翻转”对命题语句的可满足性表示为一阶模型对一阶语句的可满足性. 下面将 GSAT 判定每个语句可满足的过程用语言  $L$  的语句表述. 为此, 首先用  $L$  的语句表示 GSAT 算法中的条件  $n(v[p]) = \max\{n(v[p]) : p \text{ 是在 } A \text{ 中出现的命题变元}\}$ . 假设语句  $A$  具有形式  $C_1 \wedge \dots \wedge C_q$ , 则以下的语句  $\varphi(A, p)$  及  $\varphi_{p_1, \dots, p_n}(A, p)$  表示上述事实:

- (1)  $\forall i_1, \dots, i_k (\sigma_p(C_{i_1}) \wedge \dots \wedge \sigma_p(C_{i_k}) \wedge \bigwedge_{k+1 \leq q} \neg(\sigma_q(C_{i_1}) \wedge \dots \wedge \sigma_q(C_{i_{k+1}})))$ , 其中  $1 \leq i_1, \dots, i_{k+1} \leq n$  互不相同.  $q$  在  $A$  中出现.
- (2) 对任何的变元  $p_1, \dots, p_n$ , 语句  $\varphi_{p_1, \dots, p_n}(A, p)$  的定义方式与  $\varphi(A, p)$  的类似, 只需要用函数  $\sigma_{p_1, \dots, p_n}$  代替函数  $\sigma$ .

给定语句  $A$ , 选取一个赋值  $v$ . 在施用 GSAT 时, 得到满足  $A$  的赋值需要经过  $\beta$  次的测试.

- (a) 判定  $v$  是否可以满足  $A$ .
- (b) 若上一次得到的赋值  $v$  不满足  $A$ , 改变  $v$  对某一个命题变元的取值.

交替地施行上述操作  $\beta$  次, 若能得到可满足的赋值, 则断定语句  $A$  是可满足的. 施行上述的第 2 种操作时, 习惯上根据一些策略选定赋值  $v$ , 以实现高效的搜索.

为了描述 GSAT 的求解过程, 首先对语句的形式作适当的限制. 在 GSAT 中, 习惯上仅考虑具有  $\wedge \vee$ -标准型且每个  $\vee$ -分枝仅包含 3 个子语句的语句. 因为在算法中不考虑语句的等价性, 所以, 在实际操作中可能遇到一些等价的语句, 但它们的语法形式不相同. 形式之间的不一致性在逻辑性质上一般并不会有所影响, 但在

SAT问题的随机算法中,却可能使得算法得出不同的结论.如,可能存在等价的语句  $A$  及  $B$ ,在一次测试中  $A$  被判定为可满足的,但在同一次测试中可能得不出  $B$  的可满足性.为了避免因表示形式不同所导致的矛盾,在以下的讨论中假定所有语句都对应于给定形式的  $\wedge \vee$ -标准型,或者仅考虑具有这种标准型的语句.

转换映射  $\sigma$  具有用来描述 GSAT 求解算法的第 1 步的作用:给定一个赋值,判定它是否满足一个语句.因而当  $\beta=1$  时,GSAT 所定义的可满足性能够由语言描述.一般地,为了描述 GSAT 求解的全过程,需要引进一些二阶术语.这是因为 GSAT 依赖于语句  $\wedge \vee$ -标准型的结构,它需要应用二阶语言或者特定的一阶语言描述.比如,在语言  $L$  上添加一些描述算术的谓词,通过编号(Gödel 编号)的方法表述语句的结构.但此技术影响到对搜索深度的描述,所以,下面用语言  $L$  的语句描述 GSAT 求解的全过程,定义一个扩充的转换映射  $\tau$ ,它具有与  $\sigma$  类似的性质: $v \models A$  当且仅当  $\sigma(v) \models \tau(A)$ .  $\tau$  的定义意味着用性质比较弱的语言描述给定的自然数,所以需要假定参数  $\beta$  是确定的.  $\tau$  与  $\sigma$  之间的差异表现在它不再是通过对话句的结构归纳而定义的,而是针对语句结构定义的.为了定义转换映射  $\tau$ ,先定义映射  $\tau_i, 0 \leq i \leq \beta$ ,这里的  $\beta$  是 GSAT 算法中的参数.

**定义(描述 GSAT 求解过程的转换映射  $\tau$ ).**

假定  $p_1, \dots, p_n$  是出现在  $A$  中的所有命题变元.

$$\tau_0(A) = \sigma(A).$$

$$\tau_1(A) = \bigvee_{i,j \text{ 互不相同}} \sigma_{p_i, p_j}(A).$$

$$\tau_2(A) = \bigvee_{i,j \text{ 互不相同}} (\sigma_{p_i, p_j}(A) \wedge \varphi(A, p_j)).$$

$$\tau_3(A) = \bigvee_{i,j,k \text{ 互不相同}} (\sigma_{p_i, p_j, p_k}(A) \wedge \varphi(A, p_j) \wedge \varphi_p(A, p_k)).$$

...

$$\tau(A) = \bigvee_{i < \beta} \tau_i(A).$$

在上述定义的基础上,针对 GSAT,已经赋予  $\vdash, \models$  以确切的含义.用  $\sigma, \tau$  可以构造出前面所说的转换映射.这使得  $\models$  与  $\models$  之间的关系变得比较明确.虽然以  $v$  为初值能够搜索到语句  $A$  及  $B$  的可满足赋值,但是根据上述的两个搜索过程并不能直接地构造出  $A \wedge B$  的可满足赋值的搜索过程.另一方面,虽然两个赋值可以搜索出相同的结论,但是并不能保证两个赋值是完全相同的(模型结构相同).更细致的观察可以发现,针对 GSAT 的转换映射可以由命题逻辑的语句构造出来.这是因为在构造  $\sigma, \tau$  时,并没有用到一阶语言  $L$  中的量词.把相关的技术推广到一阶逻辑的目的在于为了用模型论的技术证明本文的定理.

对搜索算法的期望是它在实质上不等同于逻辑推导,即  $\vdash \neq \models$ ,以保证搜索算法确实能够得到较逻辑方法更多的结论.在研究逻辑系统时,定义了很多具有可靠性、完全性的逻辑,它们都对应于一个模型集合  $\mathcal{M}$ .由  $\mathcal{M}$  确定成真的结论能够用逻辑演算的方法得到.采用扩充模型集合  $\mathcal{M}$  的方式,添加模型  $v$  以扩充集合  $\mathcal{M}$ ,以得到更多的事实,由此获得新的推导关系,从而更新一个逻辑理论.通过选定特殊的模型集合以得到有别于经典逻辑的演算的技术,在直觉主义逻辑、模态逻辑、时态逻辑、多值逻辑、非单调逻辑等理论中被采用.因而,模型集合决定了逻辑理论的性质,在一定程度上也决定了搜索算法的性质.

在逻辑理论中,假定  $\mathcal{M}$  是理论  $T$  的所有模型的集合,当  $v$  所能满足的语句都可以在  $T$  的某个模型中满足时,模型  $v$  其实已经被包含在  $T$  的模型类中.因而,这时所添加的模型并不能改进模型类所确定的推导演算.在经典逻辑中,一个模型是否已被包含在一个模型集合中,可以根据它所满足的语句是否都能被给定模型集合中的某个模型满足来确定.这是逻辑中对初等类性质的描述<sup>[3]</sup>,本文正是推广了相应的技术<sup>[3]</sup>,证明了所述的定理.因为满足关系  $\models$  不具备对“并且”的交换性,所以在定理中要求“有限的结论集合”,其必要性是第 2 节的证明所要求的.在搜索算法中,希望通过选择更广泛的初始状态以得到更多的结论.因而需要保证,所选择的初始状态不被包含在一个给定的初始状态集合中.当所选择的初始状态都包含在一个初始状态集合中,同时这个集合又可以用一阶语言描述时,搜索算法所得到的结论实质上等同于用一阶逻辑的推理所得出的结论.本文的定理刻画了初始状态与由一阶性质描述的初始状态集合之间的关系.

**2 定理的证明**

为了证明本文的定理,需要引用模型论中有关超滤和超积的结论.<sup>[3]</sup>集合  $U$  上的一个超滤  $u$  是  $U$  的一些子

集 $\mathcal{U}$ 的集合,该集合满足条件: $\emptyset \in \mathcal{U}$ ;若  $x \in \mathcal{U}, y \in \mathcal{U}$ , 则  $x \cap y \in \mathcal{U}$ ;若  $x \in \mathcal{U}, x \subseteq y \subseteq U$ , 则  $y \in \mathcal{U}$ ;当  $x \in U$  时,或者  $x \in \mathcal{U}$  或者  $\neg x \in \mathcal{U}$ . 一个集合上的超滤的存在性不是显然的,但是集合论<sup>[3]</sup>中的定理表明:若  $\mathcal{v}$  是  $U$  的子集的一个集合、不包含空集并且对  $\cap$  封闭,即当  $x, y \in \mathcal{v}$  时,  $x \cap y \in \mathcal{v}$ , 则存在超滤  $\mathcal{u}$  包含  $\mathcal{v}$ <sup>[3]</sup>.

给定集合  $U$  及其上的超滤  $\mathcal{u}$ , 模型集合  $\{v_x | x \in U\}$ . 超积  $\Pi_{x \in U} v_x$  也是一个模型,它具有性质:对任意的语句  $A$ ,  $\Pi_{x \in U} v_x \models A$  当且仅当  $\{x \in U | v_x \models A\} \in \mathcal{u}$ . 这就是模型论中的超积基本定理<sup>[3]</sup>. 因而当模型集中的每个模型都满足语句  $A$  时,它的超积也满足此语句. 采用定理中的约定,下面依据模型论中的上述事实证明本文的定理.

假设  $S$  是由  $v$  可以搜索出的所有结论的集合,即  $A \in S$  当且仅当  $\sigma(v) \models \sigma(A)$ . 根据对搜索算法的一般约定,假定  $S$  非空. 根据定理的前题可知,对任意的有限集合  $s \subseteq S$ , 存在  $w \in \mathcal{C}$ , 使得  $w \models s$ , 且  $T \models w$ . 以  $U$  表示集合  $S$  的所有有限集合组成的集合. 对于任意的  $s \in U$ , 定义  $\langle s \rangle$  是  $U$  中所有包含  $s$  的有限集合的集合, 定义  $\mathcal{v} = \{\langle s \rangle; s \in U\}$ .

$S$  是非空的,因而  $\mathcal{v}$  也是非空的. 同时,  $\mathcal{v}$  对  $\cap$  封闭,这是因为对任意的  $s, t \in U$ , 等式  $\langle s \rangle \cap \langle t \rangle = \langle s \cap t \rangle$  成立. 因而  $\mathcal{v}$  包含在  $U$  上的一个超滤  $\mathcal{u}$  中. 根据定理中的约定可知,对任意的  $x \in U$  (即  $x \subseteq S$ , 因而它是由  $v$  搜索出的一个有限结论集合), 存在初始状态  $v_x \in \mathcal{C}$ , 使得  $\sigma(v_x) \models \sigma(x)$ . 这里采用的是集合论中的约定,  $\sigma(v_x) \models \sigma(x)$  的含义是对任何的  $A \in x, \sigma(v_x) \models \sigma(A)$ . 根据模型论中的超积基本定理,  $w = \Pi_{x \in U} v_x$  是  $T$  的模型. 这是因为每个  $v_x$  满足  $T$ . 此时,  $w \models v$ , 这是因为对任意的语句  $A$ , 从  $v \models A$  可知  $\sigma(v) \models \sigma(A)$ . 因为  $\sigma(A)$  是语言  $L$  的语句, 模型集合  $\{v_x; x \in U\}$  中满足  $\sigma(A)$  的模型的标号  $x$  的集合属于超滤  $\mathcal{u}$ . 这是因为  $\{x; v_x \models A, x \in U\} \supseteq \langle \{A\} \rangle$ . 模型论中的超积基本定理保证  $\sigma(w) \models \sigma(A)$ , 从而  $w \models A$ , 因而  $w \models v$ . 根据初始状态集合  $\mathcal{C}$  对  $\models$  的封闭性可知,  $v \in \mathcal{C}$ , 这就是定理中需要证明的.

### 3 总 结

本文用一阶逻辑描述搜索算法求解问题的过程,以 GSAT<sup>[2]</sup>为例说明构造上述描述的技术及逻辑特性. 搜索过程的可描述性决定了搜索算法的逻辑性质由初始状态空间决定,搜索过程与逻辑推导之间的差异表现为初始状态集合与模型集合的差异性. 更一般地,各种逻辑理论之间的联系表现在所讨论的模型集合上,这是因为它们的语法推导关系具有可靠性与完全性,这在逻辑的意义下是可定义的. 因而,本文在讨论搜索过程的逻辑性质时,关注于初始状态集合,给出的定理刻画了初始状态与由一阶语句描述的状态集合之间的关系. 搜索算法所有可能的初始状态集合一般可以由一阶逻辑刻画,所以搜索过程具有紧致性. 本文进一步讨论了一般的初始状态类的封闭性质.

对搜索算法的研究集中于设计高效的求解算法、寻找快速的求解策略;对算法的评价习惯上用随机生成的问题模型,并观察算法在这类模型上的表现. 这些技术在 GSAT 的相关研究中有集中的体现.<sup>[3]</sup> 本文用逻辑方法研究了算法的逻辑特征,即讨论算法与谓词逻辑之间的联系. 算法求解过程是一种“机械的操作”过程,因而由一个初始状态搜索出的结论集合是递归的,由所有初始状态搜索出的结论集合也是递归的. 在这种意义下,转换映射的存在性是显然的. 但是,当算法采用非递归的初始状态空间时,结论集合的递归性质则是非显然的. 所以,本文的讨论有别于对算法的递归性质的讨论,事实上,由递归论技术得到的转换映射不便于讨论  $\models$  的逻辑性质,因为需要引进自然数对求解过程编号.

搜索算法中对一个状态是否满足给定的问题的判断自然地表示为可满足关系( $\models$ ). 本文把搜索过程用满足关系  $\models$  表示. 假设  $v$  是一个初始状态,  $A$  是通过搜索欲得到判定的结论, 则  $v \models A$  表示以  $v$  作为初始状态可以得到结论  $A$ . 正如经典逻辑的性质由满足关系  $\models$  决定,对搜索算法的逻辑考察也出于对  $\models$  的讨论. 满足关系  $\models$  具有与经典逻辑的满足关系( $\models$ )类似的性质. 两种满足关系之间的差异性表现在两个方面:(1)  $v \models A$  及  $v \models B$  时,  $v \models A \wedge B$  是否一定成立.(2) 当  $v \models w, w \models v$  时,  $v$  与  $w$  的结构之间是否有必然联系. 虽然  $\models$  与  $\models$  的性质有所不同,模型论的技术仍不失为分析  $\models$  的性质的有用工具. 本文用模型论技术分析了初始状态之间的语义联系,证明了初始状态的初等类具有有限封闭性质. 作者将在进一步的工作中讨论满足上述条件的搜索算法的性质.

### 参考文献

- 1 陆汝钤. 人工智能. 北京: 科学出版社, 1996  
(Lu Ru-qian. Artificial Intelligence. Beijing: Science Press, 1996)
- 2 Selman B, Kautz H. An empirical study of greedy local search for satisfiability problem. In: Proceedings of the 11th National Conference on Artificial Intelligence. Cambridge, MA: MIT Press, 1993. 46~51
- 3 Chang C.C, Keisler H J. Model Theory. Amsterdam: North-Holland Publishing Company, 1973

## The Logical Analysis of Search Algorithms

ZHANG Yu-ping

(Department of Computer Science and Engineering Beijing University of Aeronautics and Astronautics Beijing 100083)

**Abstract** The state space, search heuristics, and move generator in searching solutions to a given problem are characterized by first order language, which implies that the logical properties of a search algorithm are determined by its state space, that is, the logic describing the searching process has compactness and the class of initial states has finite close property.

**Key words** Search algorithm, first order logic, model.