

基于派生谓词的 STRIPS 领域知识提取策略^{*}

边 芮^{1,2}, 姜云飞¹, 吴向军³⁺, 梁瑞仕¹

¹(中山大学 信息科学与技术学院 软件研究所, 广东 广州 510275)

²(广东商学院 公共管理学院, 广东 广州 510320)

³(中山大学 软件学院, 广东 广州 510275)

Strategy of Extracting Domain Knowledge for STRIPS World Based on Derived Predicates

BIAN Rui^{1,2}, JIANG Yun-Fei¹, WU Xiang-Jun³⁺, LIANG Rui-Shi¹

¹(Software Research Institute, School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, China)

²(School of Public Management, Guangdong University of Business Studies, Guangzhou 510320, China)

³(Software School, Sun Yat-Sen University, Guangzhou 510275, China)

+ Corresponding author: E-mail: issxjwu@mail.sysu.edu.cn, http://www.sysu.edu.cn

Bian R, Jiang YF, Wu XJ, Liang RS. Strategy of extracting domain knowledge for STRIPS world based on derived predicates. Journal of Software, 2011, 22(1): 57-70. <http://www.jos.org.cn/1000-9825/3726.htm>

Abstract: Domain knowledge acquisition is essential in the AI planning. Derived rule is a representation to the domain knowledge based on logical reasoning. On the basis of the action model and derived rules analysis, this paper proposes a strategy of extracting domain knowledge for STRIPS world based on derived predicates and the algorithm *GetDomainRule* for the strategy. The domain rules extracted by the algorithm are used to reduce the logical deduction of derived rules, enhancing the efficiency of the planning. For any domain, the mutex properties of any domain can be obtained between predicates from the domain rules, applied in judging the inconsistent planning state. Finally, the strategy proposed in this paper is embedded in the planner StepByStep and experiments are conducted to extract domain rules. Experimental results show the feasibility and validity of the algorithm *GetDomainRule*, and the domain rules extracted by the algorithm express the causal relations between predicates directly, providing reliable domain knowledge for determining the true values of derived predicates and the following planning.

Key words: AI planning; STRIPS; derived predicate; domain knowledge

摘 要: 领域知识的获取是智能规划研究中的重要内容之一.派生规则是一种基于逻辑推理的领域知识表示方法.在对动作模型和派生规则综合分析的基础上提出了基于派生谓词的 STRIPS 领域知识提取策略,并给出了该提取策略的算法描述.在规划求解过程中,利用提取所得的领域规则可减少派生规则的逻辑推导,从而提高规划效率.对任意一个规划领域,利用提取所得的领域规则可以分析出领域谓词之间的互斥关系,这种互斥关系可以对一类矛盾的规划状态进行判定.最后,把所提出的领域规则提取策略嵌入到规划器 StepByStep 之中进行了领域规则提取实验.实验结果验证了该提取算法的可行性和有效性,所提取出的领域规则能够直观地表达谓词间的因果关系,为派生

* 基金项目: 国家自然科学基金(60773201, 60970042)

收稿时间: 2008-12-10; 修改时间: 2009-05-15; 定稿时间: 2009-08-28; jos 在线出版时间: 2010-04-28

谓词的真值判断和后续的规划求解提供了可靠的领域知识.

关键词: 智能规划;STRIPS;派生谓词;领域知识

中图法分类号: TP181 文献标识码: A

在人工智能研究领域,智能规划问题(简称规划问题)实际上是一种关于动作的推理问题,即从某个特定问题的初始状态出发,寻找一个动作序列,使之能够到达目标状态.规划问题(包括规划领域和规划问题实例)主要采用一阶谓词逻辑描述,目前应用最广泛的描述语言是 PDDL(planning domain definition language)^[1].规划器是规划问题的求解系统,它可以运用不同的求解策略来解决规划问题.但是,无论采用何种求解策略,其目的都是为了提高规划求解效率.提高求解效率最有效的方法是在求解过程中运用领域知识,从而避免一些不必要的动作,减少了搜索空间.因此,领域知识的获取成为智能规划研究领域中的重要内容之一.

规划器可以直接或间接地从领域描述中获取领域知识,如 TLPLAN^[2]和 TALplan^[3]的时态模态逻辑规则、DCIPS^[4]的领域约束规则、PDDL 的公理子句和派生规则等,这些是规划器可从领域描述中直接获取的领域知识.相应地,TIM^[5]的状态常量、LPG^[6-8]的动作间互斥关系和谓词间互斥关系、StepByStep^[9-11]的领域规则等,这些是规划器可从领域描述中间接获取的领域知识.其中:TLPLAN,TIM 与 LPG 中的领域知识是与“规划问题实例”相关的,不能反映规划领域中的一般规律,通用性不强;DCIPS 的领域约束规则表示方法比较复杂,应用不够广泛;PDDL 的公理子句和派生规则是一种基于逻辑推理的领域知识表示方法,它增强了 PDDL 的领域描述能力,但在规划求解过程中引入逻辑推导会影响求解效率;StepByStep 的领域规则是一种非常有效的领域知识表示方法,它提取自领域描述中的动作模型,与“规划领域”相关,而与“规划问题实例”无关.如何有效地从领域描述中获取领域知识,提高规划求解效率,这是一个值得研究的问题.

本文研究基于派生谓词的 STRIPS 领域知识提取策略,该策略通过对规划领域中动作模型和派生规则的综合分析,提取领域规则,从而既保留了派生规则对领域知识的强表达能力,又继承了领域规则的与规划问题实例无关的特性.该策略提取的领域规则直观地表达了规划领域描述中所隐含的信息,可用于减少规划求解过程中派生规则的逻辑推导,并指导规划动作的启发式搜索,从而达到提高规划效率的目的.

本文首先简单介绍派生规则的语法和语义,通过对派生规则的分析,说明派生规则对表达领域知识的价值.在此基础上提出从动作模型和派生规则中提取领域规则的方法,利用领域规则可对一类矛盾的规划状态进行判定.然后给出提取领域规则的算法描述,在 StepByStep 规划器中给予实现,并选取一些经典规划领域及其派生规划领域进行实验.经分析,所获取的领域规则直观地表达了谓词间的因果关系,这些领域规则对派生谓词的真值判断和后续的规划求解提供了可靠的领域知识.

1 派生规则

PDDL 是规划领域的描述语言,在其版本 2.1 中增加了时态和数值度量的描述子句^[12],版本 2.2 在前者的基础上增加了派生谓词和时间初始文字两个特性^[13],其中,派生谓词是由表达领域知识的领域公理发展而来的.包含派生谓词的规划领域将谓词分为两类:基本谓词和派生谓词.基本谓词用于表示动作的直接效果,派生谓词则表示动作的间接效果.派生规则表示基本谓词与派生谓词之间的因果关系,是一种人为表达领域知识的方法.

1.1 派生规则的语法

在 PDDL 2.2 中,派生谓词是用“if $\phi(X)$ then $P(X)$ ”规则形式定义的,其语法定义如下:

$$(:\text{derived } P(X) \phi(X)),$$

其中,derived 是派生规则的标识符; $\phi(X)$ 是一阶谓词公式,即派生规则的触发条件; $P(X)$ 是派生规则的逻辑结论,即派生谓词; X 是自由变量集合.

对任意一个领域谓词,如果它是某派生规则的逻辑结论,那么它就是该规划领域的派生谓词,否则,为基本谓词.为了使派生规则的定义有意义且便于规划器的处理,PDDL 2.2 对派生规则和派生谓词进行了限定:

- 动作不影响派生谓词,派生谓词不出现在初始状态和任何动作的效果列表中,但可以出现在动作的前提条件、效果上下文和目标状态中;
- 若谓词 $P(X)$ 由 $\phi(X)$ 派生所得,则 $\phi(X)$ 中的所有自由变量都是 $P(X)$ 中的自由变量;
- 若谓词 $P(X)$ 由 $\phi(X)$ 派生所得,则 $\phi(X)$ 的否定范式 NNF(negation normal form)不包含任何派生谓词的否定形式.

在规划领域 BWDP(blocks world-derived predicates)中,谓词(above ?x ?y)是该领域的一个派生谓词,它表示积木块?x 在积木块?y 的上方.与该派生谓词相关的派生规则定义如下:

$$\begin{aligned} (:derived \quad (above \ ?x \ ?y) & \qquad \qquad \qquad //P(X) \\ (or \ (on \ ?x \ ?y) \ (exists \ (?z) \ (and \ (on \ ?x \ ?z) \ (above \ ?z \ ?y)))) & \qquad //\phi(X) \end{aligned}$$

该派生规则的含义是:(on ?x ?y)派生出(above ?x ?y),或 $\exists z((on \ ?x \ ?z) \wedge (above \ ?z \ ?y))$ 派生出(above ?x ?y).

由此可见,派生规则是领域知识的一种客观体现形式,它反映了领域谓词之间的一种逻辑关系.

我们称带有派生规则的规划领域为派生规划领域,其相应的规划问题称为派生规划问题.对于一个派生规划领域,其规划动作描述将有所简化,因为其动作的有些效果不会直接写出,它是用派生规则隐性地表达出来的.附录 A 是规划领域 BWDP 的完整领域描述.

1.2 派生规则的语义

在派生规划领域中,其基本谓词的真值由动作的执行效果来改变,而其派生谓词的真值是在封闭世界假定下,由其派生规则推导所得的.因此,在当前状态下,执行一个动作会引起两方面的变化:基本谓词的变化和派生谓词的变化.

基本谓词的实例称为基本事实(basic fact),派生谓词的实例称为派生事实(derived fact).在规划状态 S_i 下,用符号 B_i 来表示其基本事实集,用符号 D_i 来表示在基本事实集合 B_i 上用派生规则所得到的派生事实集,规划状态 S_i 由 $B_i \cup D_i$ 来确定,即 $S_i = B_i \cup D_i$.

为了描述基本事实集合 B_i 和派生事实集合 D_i 之间的关系,我们用函数 f 描述基本事实集 B_i 在派生规则集 R 下的闭包运算^[14]:

$$f(B_i) = \cap \{S' | B_i \subseteq S', \forall (P(x), \phi(x)) \in R: \forall c, |c| = |x|: (S' = \phi(c) \Rightarrow P(c)) \in S'\} \quad (1)$$

假设在当前状态 S_i 下,执行动作 a 后得到规划次态 S_{i+1} ,那么规划次态 S_{i+1} 由公式(2)可得:

$$S_{i+1} = f((S_i - Del_a) \cup Add_a - D_i) \quad (2)$$

其中, Add_a 和 Del_a 表示动作 a 的增加效果集和删除效果集.

由于派生谓词及其实例都不出现在任何动作的效果列表中,所以 $D_i \cap Del_a = \emptyset, D_i \cap Add_a = \emptyset$.

由 $S_i = B_i \cup D_i, D_i \cap Del_a = \emptyset, D_i \cap Add_a = \emptyset$ 可进行以下推导:

$$((S_i - Del_a) \cup Add_a) - D_i = ((B_i \cup D_i - Del_a) \cup Add_a) - D_i = ((B_i - Del_a) \cup Add_a) \cup D_i - D_i = (B_i - Del_a) \cup Add_a = B_{i+1} \quad (3)$$

用公式(3)可将公式(2)简化为

$$S_{i+1} = f(B_{i+1}) \quad (4)$$

由公式(4)可知,次态 S_{i+1} 仅由基本事实集 B_{i+1} 和派生规则集 R 来确定,它正好反映了在派生规划领域中执行规划动作后状态变化的本质,即:基本事实部分由动作效果直接产生变化,派生事实部分由派生规则逻辑产生.执行一个规划动作的状态变化情况如图 1 所示.

由上可知,假设在状态 S_i 下可并行地执行动作集 A ,那么其后继状态 S_{i+1} 为

$$S_{i+1} = f((B_i - \cup_{a \in A} Del_a) \cup (\cup_{a \in A} Add_a)).$$

在 BWDP 领域中,假设当前状态 S_i 如图 2 所示,

$$B_i = \{(\text{clear } A), (\text{on } A \ B), (\text{on } B \ C), (\text{ontable } C), (\text{handempty})\},$$

$$D_i = \{(\text{above } A \ B), (\text{above } A \ C), (\text{above } B \ C)\}.$$

在执行动作 $\text{unstack}(A \ B)$ 后,

$$B_{i+1} = \{(\text{clear } B), (\text{on } B \ C), (\text{ontable } C)\},$$

$$D_{i+1} = \{(above\ B\ C), (holding\ A)\}.$$

其后继状态 $S_{i+1} = B_{i+1} \cup D_{i+1}$.

用派生谓词来描述动作对状态的间接影响,使领域描述变得简洁、明了.文献[15]证明,若将派生谓词从领域描述中除去,则将造成领域描述或规划解长度的指数级增长.可见,派生谓词是领域描述中的一个重要组成部分,但它也带来一些问题:在执行动作的后继状态中,所有派生谓词的真值需重新推导.因此,本文在派生规则的基础上利用动作模型和逻辑推理来提取相关的领域知识,从而减少在规划求解过程中派生规则的逻辑推导,以便提高派生规划问题的求解效率.

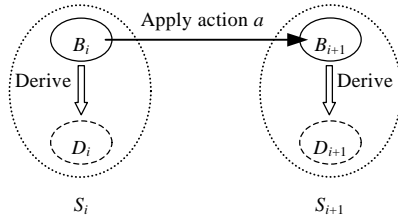


Fig.1 Transition of state after applying action a

图1 执行动作 a 的状态转换示意图



Fig.2 Transition of state S_i in BWDP

图2 BWDP 领域中状态 S_i 的转换示意图

2 派生规划领域的领域知识提取策略

本文对派生规划领域提出一种领域知识提取策略,该策略是在分析动作模型和派生规则的基础上提取出隐含在派生规则中的领域知识.本节首先简单介绍从动作模型中提取领域规则的基本思想,然后重点讨论利用派生规则提取领域规则的提取策略.

2.1 动作模型提取的领域规则

文献[9,10]中对 STRIPS 规划领域提出了相应的领域知识提取策略,并提取出两类领域规则:伴随规则和阻碍规则.所用的符号和定义如下:

用 $PC(a)$ 和 $E(a)$ 分别表示动作 a 的前提条件和动作效果集合,用 $Act(Pd)$ 表示实现谓词 Pd 的动作集合, $CPC(Pd)$ 和 $CE(Pd)$ 表示动作集 $Act(Pd)$ 中所有动作的公共前提条件集合和公共效果谓词集合,即

$$CPC(Pd) = \bigcap_{a \in Act(Pd)} PC(a), \quad CE(Pd) = \bigcap_{a \in Act(Pd)} E(a).$$

定义 2.1. 假设 Pd 和 Pd_j 是动作效果谓词,若 $Pd_j \in CE(Pd) - \{Pd\}$, 则不论采用集合 $Act(Pd)$ 中的哪一个动作,在实现谓词 Pd 的同时, Pd_j 也一定被同时实现.称谓词 Pd_j 是实现谓词 Pd 的直接伴随结果,记为 $Pd_{\{(not\ Pd)\}} \rightarrow Pd_j$, 其中, $\{(not\ Pd)\}$ 为“真”是该伴随规则成立的条件.

定义 2.2. 假设 Pd 和 Pc 分别是动作效果谓词和动作前提条件谓词,若 $Pc \in CPC(Pd) - \{(not\ Pd)\}$, 则不论采用集合 $Act(Pd)$ 中哪一个动作实现谓词 Pd , 都需要前提条件 Pc , 称谓词 Pc 是实现谓词 Pd 的前提条件, 或称谓词 $(not\ Pc)$ 为“真”直接阻碍谓词 Pd 的实现, 记为 $(not\ Pc)_{\{(not\ Pd)\}} \rightarrow Pd$, 其中, $\{(not\ Pd)\}$ 为“真”是该阻碍规则成立的条件.

在不引起混淆的情况下,谓词之间的直接伴随规则和直接阻碍规则可简记为 $Pd \rightarrow Pd_j$ 和 $(not\ Pc) \rightarrow Pd$.

在非派生规划领域 BlocksWorld 中, $Act((ontable\ ?x)) = \{(put\ down\ ?x)\}$, 动作 $put\ down$ 的描述如下:

```
(:action put-down
:parameters (?x)
:precondition (holding ?x)
:effect (and (clear ?x) (handempty) (ontable ?x) (not (holding ?x))))
```

由定义 2.1 可提取出以下 3 个直接伴随规则:

$$(ontable\ x) \rightarrow (clear\ x), \quad (ontable\ x) \rightarrow (handempty), \quad (ontable\ x) \rightarrow (not\ (holding\ x)).$$

在派生规划领域 BlocksWorld 中,其动作 $put\ down$ 的描述见附录 A.用定义 2.1 只可提取出前两个直接伴随

规则,不能提取出第 3 个直接伴随规则,这是因为派生谓词(holding ?x)及其否定形式都不能出现在动作效果之中。

对 BWDP 领域,在执行动作 put-down 之后的状态中,若要确定谓词(holding x)的真值,则必须判断其触发条件“(not (ontable ?x))^(not (exists (?y) (on ?x ?y)))”是否得到满足。显然,这样的逻辑推导需要付出一定的代价。

由此可见,虽然派生规则提高了领域描述能力,使动作描述变得简洁,但同时也损失了一些可从动作模型中提取的领域知识,并且这些派生谓词的真值还需要通过逻辑推导来得到。

2.2 派生规则提取的间接效果规则

在规划领域中,动作的前提条件和效果之间存在着一定的逻辑关系,通过对实现某一谓词动作集的公共前提条件和公共效果的分析可以提取其基本规则(直接伴随规则和直接阻碍规则)。在派生规划领域中,派生规则的触发条件和派生谓词之间也存在着逻辑关系,通过对派生规则触发条件的逻辑转换和分析也可提取相应的判定规则:间接效果规则和阻碍效果规则。间接效果规则可用于判定派生谓词的真值,阻碍效果规则可用于指导规划动作的启发式搜索。

在派生规划领域中,任意一个派生规则“if $\phi(X)$ then $P(X)$ ”,其派生触发条件 $\phi(X)$ 均可通过否定符号内移的方式转换为否定范式 $\phi(X)_{NNF} \Leftrightarrow \phi(X)$ 。

我们可以将该 $\phi(X)_{NNF}$ 等价地转换为以下两种形式:

$$\phi(X)=F_1 \wedge F_2 \wedge \dots \wedge F_m \Leftrightarrow \phi(X)_{NNF} \quad (5)$$

$$\phi'(X)=G_1 \vee G_2 \vee \dots \vee G_n \Leftrightarrow \phi(X)_{NNF} \quad (6)$$

其中, $F_i(i=1, \dots, m)$ 为若干谓词或其否定的析取式, $G_j(j=1, \dots, n)$ 为若干谓词或其否定的合取式。

下面用 BWDP 领域中的一个派生规则来说明派生规则触发条件的转换方法。

(:derived (holding ?x) (and (not (ontable ?x)) (not (exists (?y) (on ?x ?y))))).

在上述派生规则中,其触发条件 $\phi(X)$ 的否定范式转换如下:

$$\begin{aligned} \phi(X) &= (\text{not (ontable ?x)}) \wedge (\text{not (exists (?y) (on ?x ?y))}) \\ &= (\text{not (ontable ?x)}) \wedge (\text{forall (?y) (not (on ?x ?y))}) = F_1 \wedge F_2 = \phi(X)_{NNF} = \phi'(X) \end{aligned} \quad (7)$$

其中, $F_1 = (\text{not (ontable ?x)})$, $F_2 = (\forall ?y (\text{not (on ?x ?y)}))$ 。

定义 2.3. 假设任意派生规则“if $\phi(X)$ then $P(X)$ ”,其 $\phi(X)_{NNF}$ 的等价公式为 $\phi'(X)$ 和 $\phi''(X)$ 。

(1) 对 $\phi(X)$ 中的 $F_i(i=1, \dots, m)$,当(not F_i)为“真”时,称(not $P(X)$)是(not F_i)的伴随结果。当用动作实现(not F_i)时,称(not $P(X)$)是该动作的间接效果,记为(not F_i) $_{(F_i)} \rightarrow \rightarrow$ (not $P(X)$)。其中,谓词 F_i 为“真”是该间接效果规则成立的条件;

(2) 对 $\phi'(X)$ 中的 $G_j(j=1, \dots, n)$,当 G_j 为“真”时,称 $P(X)$ 是 G_j 的伴随结果。当用动作实现 G_j 时,称 $P(X)$ 是该动作的间接效果,记为 G_j $_{(\text{not } G_j)} \rightarrow \rightarrow P(X)$,其中,谓词(not G_j)为“真”是该间接效果规则成立的条件。

在不引起混淆的情况下,将 $pc \rightarrow \rightarrow$ 简记为 $\rightarrow \rightarrow$,即(not F_i) $\rightarrow \rightarrow$ (not $P(X)$)和 $G_j \rightarrow \rightarrow P(X)$ 。

在 BWDP 领域中有派生谓词(holding ?x),其触发条件 $\phi(X) = F_1 \wedge F_2$ 如公式(7)所示。

由定义 2.3 可以得到间接效果规则(ontable x) $\rightarrow \rightarrow$ (not (holding x))和(on x y) $\rightarrow \rightarrow$ (not (holding x)),其含义是,在(not(ontable x))或(not(on x y))为“真”时,若用动作实现谓词(ontable x)或(on x y),则有(not (holding x))成立。在提取出这两条间接效果规则之后,可将其用于简化派生谓词(holding x)的逻辑推导。

对于递归派生谓词,可对间接效果规则进行扩展,得到其条件间接效果规则。

假设在派生规划领域中存在派生规则 R 为“if $\phi(X)$ then $P(X)$ ”,若 $\phi(X) = G_1 \vee G_2 \vee \dots \vee G_n, \exists G_j (G_j = \phi_j(X) \wedge P(Y))$, $Y \subset X$,那么称规则 R 为递归派生规则。

定义 2.4. 假设递归派生规则“if $\phi(X)$ then $P(X)$ ”, $\phi(X)_{NNF}$ 的等价转换公式形如 $\phi'(X) = G_1 \vee G_2 \vee \dots \vee G_n$,若 $G_j = Pd \wedge P(Y) (j \in \{1, 2, \dots, n\})$,称 $P(Y)$ 是 $P(X)$ 的递归前提。当用动作实现 Pd 时,若 $P(Y)$ 成立,则称 $P(X)$ 是该动作的条件间接效果,记为 Pd $_{(\text{not } Pd) \wedge P(Y)} \rightarrow \rightarrow P(X)$,其中,(not Pd) $\wedge P(Y)$ 为“真”是该条件间接效果规则成立的条件。

BWDP 领域中,其递归派生谓词为(above ?x ?y),其中, $\phi'(X) = (\text{on ?x ?y}) \vee (\text{exists}(?z)((\text{on ?x ?z}) \wedge (\text{above ?z ?y})))$ 。

由定义 2.3 和定义 2.4 可得下面领域规则:

- 间接效果规则 $(\text{on } x \ y)_{(\text{not } (\text{on } x \ y))} \rightarrow \rightarrow (\text{above } x \ y)$, 其含义是, 在 $(\text{not } (\text{on } x \ y))$ 为“真”的前提下实现谓词 $(\text{on } x \ y)$ 时, 派生谓词 $(\text{above } x \ y)$ 成立;
- 条件间接效果规则 $(\text{on } x \ z)_{(\text{not } (\text{on } x \ z)) \wedge (\text{above } z \ y)} \rightarrow \rightarrow (\text{above } x \ y)$, 其含义是, 存在一个 z , 使得 $(\text{not } (\text{on } x \ z)) \wedge (\text{above } z \ y)$ 为“真”, 当实现谓词 $(\text{on } x \ z)$ 时, 派生谓词 $(\text{above } x \ y)$ 也成立.

由此可见, 在(条件)间接效果规则条件成立的情况下, 可以利用该效果规则来判定后继状态中派生谓词的真值. 在(条件)间接效果规则条件不成立时, 则可利用该效果规则判定当前状态中派生谓词的真值, 而无需判定派生规则的触发条件.

2.3 派生规则提取的阻碍效果规则

对于基本谓词, 可以从动作模型中得到实现它的必要前提条件, 在规划求解过程中, 可以利用其必要的前提条件来确定候选动作集. 对于派生谓词, 它是派生规则触发条件的逻辑结论, 阻碍派生规则触发条件的实现也就间接地阻碍了派生谓词的实现. 派生规则描述了其触发条件与派生谓词之间的直接因果关系, 通过对派生规则和动作模型的分析, 可以提取出基本谓词和派生谓词之间的间接因果关系. 在规划求解过程中, 可以利用该间接因果关系来确定候选动作集.

下面介绍利用派生规则和动作模型来提取实现谓词的公共前提条件, 并由此得到阻碍效果规则的方法.

定义 2.5. 假设任意谓词 Pd^* , $CPC(Pd)$ 是使谓词 Pd 为“真”的必要前提条件:

- (1) 若 Pd 为基本谓词, $CPC(Pd)$ 是动作集 $Act(Pd)$ 中所有动作的公共前提条件集合;
- (2) 若 Pd 为派生谓词, $CPC(Pd)$ 是谓词 Pd 的公共前提条件.

在派生规划领域中, 若 Pd 是基本谓词, $CPC(Pd)$ 可以通过动作模型求得. 若 Pd 是派生谓词, $CPC(Pd)$ 则需通过派生规则的触发条件的否定范式求得.

定义 2.6. 假设有派生规则“if $\phi(X)$ then $P(X)$ ”, $\phi'(X) = G_1 \vee G_2 \vee \dots \vee G_n$, 谓词 $P(X)$ 的公共前提条件 $CPC(P(X)) = \bigcap_{j=1, \dots, n} CPC(G_j)$, 且有:

- (1) 若 $Pd_u = \forall x Pd_v(x, X)$, 则 $CPC(Pd_u) = \bigcap_{\forall x} (CPC(Pd_v(x, X)) - \{(\text{not } Pd_v(x, X))\})$;
- (2) 若 $Pd_u = \exists x Pd_v(x, X)$, 则 $CPC(Pd_u) = \bigcap_{\forall x} (CPC(Pd_v(x, X)) - \{(\text{not } Pd_v(x, X))\})$;
- (3) 若 $G_j = Pd_1 \wedge Pd_2 \wedge \dots \wedge Pd_k$, 则 $CPC(G_j) = \bigcap_{u=1, \dots, k} (CPC(Pd_u) - \{(\text{not } Pd_u)\})$;
- (4) 若 $G_j = Pd_1 \wedge Pd_2 \wedge \dots \wedge Pd_k \wedge P(Y)$, 则 $CPC(G_j) = \bigcap_{u=1, \dots, k} (CPC(Pd_u) - \{(\text{not } Pd_u)\})$.

其中, Pd_u 是出现在 $\phi'(X)$ 中的谓词.

在 BWDP 领域中, 对其派生谓词(holding ?x)来说, 其触发条件 $\phi(X) = F_1 \wedge F_2$ 如公式(7)所示.

$$\begin{aligned} CPC(\text{holding } ?x) &= (CPC(\text{not } (\text{ontable } ?x)) - \{(\text{ontable } ?x)\}) \cap (CPC(\forall ?y (\text{not } (\text{on } ?x \ ?y)))) \\ &= \{(\text{clear } ?x), (\text{handempty})\} \cap \{(\text{clear } ?x), (\text{handempty})\} \\ &= \{(\text{clear } ?x), (\text{handempty})\}, \end{aligned}$$

其含义是, 要派生出谓词(holding x), 则必须先使谓词(clear x)和(handempty)成立.

定义 2.7. 假设任意派生规则“if $\phi(X)$ then $P(X)$ ”, 若谓词 $Pc \in CPC(P(X))$, 则不论用哪些动作实现派生规则的触发条件, 都需要前提条件 Pc , 称谓词 Pc 是触发谓词 $P(X)$ 的前提条件, 或称谓词 $(\text{not } Pc)$ 为“真”直接阻碍谓词 $P(X)$ 的派生, 记为 $(\text{not } Pc)_{(\text{not } P(X))} - \rightarrow P(X)$, 其中, $(\text{not } P(X))$ 为“真”是该阻碍效果规则成立的条件.

在不引起混淆的情况下, 将 $pc - \rightarrow$ 简记为 $- \rightarrow$, 即 $(\text{not } Pc) - \rightarrow P(X)$.

关于定义 2.7 的说明如下:

- (1) 当 $P(X)$ 为“真”时, 显然无需用动作实现派生规则的触发条件, 当然也就不存在其他谓词阻碍谓词 $P(X)$ 的派生;

** 本文仅以“正谓词”形式来论述, 但对“负谓词”形式也同样有效, 这里不再一一叙述.

(2) 当 $P(X)$ 为“假”时,则说明其触发条件 G_1, G_2, \dots, G_n 都为“假”. $\forall Pc \in CPC(P(X))$, 由定义 2.6 可知:

$$Pc \in \bigcap_{i=1, \dots, n} CPC(G_i).$$

即无论实现哪个 G_i 都需要谓词 Pc 为“真”. 所以,若要派生出 $P(X)$,则必须先用动作(或动作序列)使谓词 Pc 为“真”,即谓词(not Pc)为“真”一定阻碍谓词 $P(X)$ 的派生.

在不引起混淆的前提下,我们把间接效果规则和阻碍效果规则简称为伴随规则和阻碍规则,这两种规则与文献[9]所提取的知识统称为领域规则.

在 BWDP 领域中, $CPC(\text{holding } ?x) = \{(\text{clear } ?x), (\text{handempty})\}$, 由定义 2.7 可以得到下列阻碍效果规则:

$$(\text{not } (\text{clear } x)) \rightarrow (\text{holding } x) \quad (\text{not } (\text{handempty})) \rightarrow (\text{holding } x).$$

这两条阻碍效果规则的含义是,在(not(holding x))为“真”的前提下,谓词(not (clear x))和(not (handempty))阻碍了谓词(holding x)的派生.若要派生出谓词(holding x),则必须先用动作(或动作序列)使(clear x)和(handempty)为“真”,即要抓起积木块 x ,则必先使 x 上为空且当前机械手臂为空.

同理,我们还可以得到阻碍效果规则(not (holding x)) \rightarrow (above $x y$),其含义是,若要派生出谓词(above $x y$),必须先用动作(或动作序列)使(holding x)为“真”.

由此可见,在规划求解过程中,阻碍规则可以确定谓词的实现次序,并可有效地指导规划动作的启发式选择.

2.4 领域规则的对比分析

由上一节的叙述可知,文献[9,10]的领域知识提取策略是从动作模型中提取领域规则.本文的领域知识提取策略是从动作模型和派生规则中提取领域规则.

在派生规划领域中,其谓词可分为基本谓词和派生谓词.本文在设计领域知识提取策略时,对基本谓词采用文献[9,10]中的方法,从动作模型中提取领域规则;对派生谓词,从派生规则中提取领域规则.

对派生规划领域中不同领域规则的分析对比如下:

1) 间接效果规则和直接伴随规则的比较

假设从规划领域中可以提取出间接效果规则 $R_1 = "Pd_u \rightarrow Pd_v"$,直接伴随规则 $R_2 = "Pd_i \rightarrow Pd_j"$.

① 规则含义的不同

由规则 R_1 可知,在用动作实现谓词 Pd_u 后,可由派生规则直接派生出谓词 Pd_v ,也为“真”.由此可见,谓词 Pd_v 是动作的间接效果,这也是将规则 R_1 命名为间接效果规则的缘由.

由规则 R_2 可知,在用动作实现谓词 Pd_i 的同时,谓词 Pd_j 也被实现了.

② 规则时效性的不同

由规则 R_1 可知,在用动作实现谓词 Pd_u 后,谓词 Pd_v 是谓词 Pd_u 的逻辑结论.在后续状态中,只要谓词 Pd_u 成立,谓词 Pd_v 也一定成立.因此,其时效性可一直延续到谓词 Pd_u 为“假”.

由规则 R_2 可知,在用动作实现谓词 Pd_i 时,谓词 Pd_j 也一定在规划次态中.但在后续状态中,在不改变谓词 Pd_i 真值的情况下,谓词 Pd_j 还可被其他动作删掉.因此,其时效性仅限于规划次态.

2) 条件间接效果规则和条件伴随规则^[10]的比较

假设可提取出条件间接效果规则 $R_1 = "Pd_{(\text{not } Pd_i \wedge P(Y))} \rightarrow P(X)"$,条件伴随规则 $R_2 = "Pd_{i(\text{not } Pd_i, Pc)} \rightarrow Pd_j"$.

① 规则含义的不同

由规则 R_1 可知,若 $P(Y)$ 为“真”,则在用动作实现谓词 Pd 后,可逻辑派生出谓词 $P(X)$ 也为“真”,其为动作的间接效果.

由规则 R_2 可知,若 Pc 为“真”,则在用动作实现谓词 Pd_i 时,谓词 Pd_j 为“真”,其为动作的直接效果.

② 规则时效性的不同

由规则 R_1 可知,若 $P(Y)$ 为“真”,在用动作实现谓词 Pd 后,谓词 $P(X)$ 是 $P(Y) \wedge Pd$ 的逻辑结论.在后续状态中,若不改变条件 $P(Y) \wedge Pd$ 的真值,谓词 $P(X)$ 也一定成立.其时效性可一直延续到条件 $P(Y) \wedge Pd$ 为“假”.

由规则 R_2 可知,若 Pc 为“真”,在动作实现谓词 Pd_i 时,谓词 Pd_j 也一定在规划次态之中.但在后续状态中,在

不改变谓词 Pd , 真值的情况下, 谓词 Pd 还可被其他动作删除, 其时效性也仅限于规划次态.

3) 阻碍效果规则和直接阻碍规则的比较

① 规则含义的不同

阻碍效果规则“(not Pc) $\neg \rightarrow Pd$ ”描述的是直接阻碍了派生规则的触发条件从而阻碍了派生谓词 Pd 的派生. 直接阻碍规则“(not Pc) $\neg \rightarrow Pd$ ”描述的是直接阻碍了实现谓词 Pd 的所有动作从而阻碍了谓词 Pd 的实现.

② 规则时效性的相同

在任意规划状态中, 只要谓词 Pd 为“假”, (not Pc) 就会阻碍谓词 Pd 的成立, 所以, 这两类规则的时效性相同.

3 领域谓词的互斥性

在派生规划领域中, 通过对派生规则的研究可以提取出领域规则, 通过对领域规则的进一步分析, 可以得出领域谓词之间的相互阻碍关系, 利用这种相互阻碍关系可以直接判断一类派生规划问题的可解性.

定义 3.1. 假设有派生规则 R : “if $\phi(X)$ then $P(X)$ ”, $\phi(X) = F_1 \wedge \dots \wedge (\text{not } F_i) \wedge \dots \wedge (\text{not } F_j) \wedge \dots \wedge F_m$, 对基本谓词 F_i 和 F_j , 若 $(\text{not } P(X)) \neg \rightarrow F_i, (\text{not } P(X)) \neg \rightarrow F_j, F_j \notin \cup_{a \in \text{Act}(F_i)} E(a), F_i \notin \cup_{a \in \text{Act}(F_j)} E(a)$, 则称 F_i 和 F_j 为互阻谓词对.

在 BWDP 领域中, 取派生规则 “if (not (ontable ?x)) \wedge (not (exists (?y) (on ?x ?y))) then (holding ?x)”, 其触发条件 $\phi(X) = (\text{not } F_1) \wedge (\text{not } F_2), F_1 = (\text{ontable } ?x), F_2 = (\text{on } ?x ?y)$, 有以下条件成立:

(1) $(\text{not } P(X)) \neg \rightarrow F_1, (\text{not } P(X)) \neg \rightarrow F_2$;

(2) $F_1 \notin \cup_{a \in \text{Act}(F_2)} E(a), F_2 \notin \cup_{a \in \text{Act}(F_1)} E(a)$.

由定义 3.1 可知, F_1 和 F_2 即 (ontable ?x) 和 (on ?x ?y) 是互阻谓词对.

互阻谓词对的含义是: 若用动作实现了谓词 F_i , 那么在实现谓词 F_j 时一定使谓词 F_i 为“假”; 反之亦然.

因为在用动作实现谓词 F_i 时, 由定义 3.1 可知 $F_j \notin \cup_{a \in \text{Act}(F_i)} E(a)$, 即无论用哪个动作 a 实现谓词 F_i , 在次态中谓词 F_j 都不会同时被实现, 且 $P(X)$ 一定为“假”.

这时, 若要实现谓词 F_j , 则必须先使 $P(X)$ 为“真”.

由派生规则 R 可知, (not F_i) 必须为“真”, 即又需用动作使谓词 F_i 为“假”.

同理, 若用动作实现了谓词 F_j , 那么在用动作实现谓词 F_i 时又使谓词 F_j 为“假”.

定理 3.1. 假设在派生规划领域中存在派生规则 R : “if $\phi(X)$ then $P(X)$ ”, $\phi(X) = F_1 \wedge \dots \wedge (\text{not } F_i) \wedge \dots \wedge (\text{not } F_j) \wedge \dots \wedge F_m$, F_i 和 F_j 是互阻谓词对, 若目标状态含有 F_i 和 F_j , 且 F_i 或 F_j 需用动作来实现, 那么该目标状态无法实现.

证明: 假设可用动作实现谓词 F_i 和 F_j , 其中, 规划状态序列为 $S_0(\text{Init}), S_1, S_2, \dots, S_g(\text{Goal})$, 且 $F_i, F_j \in S_g$.

对规划状态序列 $S_0, S_1, S_2, \dots, S_g$, 逆序查找第 1 个状态 S_t , 使得 $F_i \notin S_t$ 或 $F_j \notin S_t$, 且 $F_i, F_j \in S_{t+1}$.

(1) 在规划状态序列 $S_0, S_1, S_2, \dots, S_g$ 中, 不存在满足上述条件的状态 S_t , 即: 对于任意状态 $S_t, i=0, \dots, g, F_i, F_j \in S_t$, 则谓词 F_i 和 F_j 在整个规划求解的过程中一直存在, 无需用动作来实现, 这显然与前提条件“谓词 F_i 或 F_j 需用动作实现”相矛盾;

(2) 在规划状态序列 $S_0, S_1, S_2, \dots, S_g$ 中, 存在满足上述条件的状态 S_t :

(2.1) $F_i \notin S_t, F_j \notin S_t, F_i, F_j \in S_{t+1}$

不妨假设在状态 S_t 下执行动作 a_t , 使得 $F_i, F_j \in S_{t+1}$.

由“ F_i 和 F_j 是互阻谓词对”可知, $F_j \notin \cup_{a \in \text{Act}(F_i)} E(a), F_i \notin \cup_{a \in \text{Act}(F_j)} E(a)$, 即没有动作 a_t 可同时实现谓词 F_i 和 F_j .

这就与“ $F_i, F_j \in S_{t+1}$ ”相矛盾.

(2.2) $F_i \notin S_t$ 或 $F_j \notin S_t, F_i, F_j \in S_{t+1}$

不妨假设 $F_i \notin S_t, F_j \in S_t$, 在状态 S_t 下执行动作 a_t , 使得 $F_i \in S_{t+1}$.

由“在状态 S_t 下执行动作 a_t ”可知, $PC(a_t) \subseteq S_t$.

由“ F_i 和 F_j 是互阻谓词对”可知, $P(X) \in PC(a_t)$, 即 $P(X) \in S_t$.

由“ $P(X)$ 的触发条件”可知, (not F_i) $\in S_t, (\text{not } F_j) \in S_t$, 即 $F_j \notin S_t$.

这就与“ $F_j \in S_t$ ”相矛盾.

综合情况(1)和情形(2)可知,不论在规划状态序列 $S_0, S_1, S_2, \dots, S_g$ 中是否存在满足条件的状态 S_t ,命题都可得出矛盾.因此,若目标状态中含有互阻谓词对,且互阻谓词对中存在需要动作实现的谓词,那么该目标状态无法实现. □

由定理 3.1 证明过程的第 1 种情况可知,若初始状态和目标状态都包含互阻谓词对,且规划求解过程不改变它们的真值,则此目标仍是可实现的.

在 BWDP 领域中,由定义 3.1 可知,(ontable ?x)和(on ?x ?y)是互阻谓词对.

由定理 3.1 可知,如果在规划求解过程中需用动作实现谓词(ontable ?x)或(on ?x ?y),那么在目标状态中(ontable ?x)和(on ?x ?y)不可能同时存在.

定义 3.2. 假设有派生规则“if $\phi(X)$ then $P(X)$ ”, $\phi(X)=F_1 \wedge \dots \wedge F_i \wedge \dots \wedge F_j \wedge \dots \wedge F_m$,对基本谓词 F_i 和 F_j ,

$$F_j, (\text{not } F_j) \notin \cup_{a \in \text{Act}(F_i)} E(a), F_i, (\text{not } F_i) \notin \cup_{a \in \text{Act}(F_j)} E(a).$$

若存在谓词 F' ,使得 $(\text{not } F') \rightarrow F_i, (\text{not } F') \rightarrow F_j, F_i \rightarrow (\text{not } F'), F_j \rightarrow (\text{not } F'), (\text{not } P(X)) \rightarrow F'$,那么称谓词(not F_i)和(not F_j)为互斥谓词对.

在 BWDP 领域中,存在派生规则“if (not (ontable ?x)) \wedge (not (exists (?y) (on ?x ?y))) then (holding ?x)”,其触发条件 $\phi(X)=F_1 \wedge F_2, F_1=(\text{not } (\text{ontable } ?x)), F_2=(\text{not } (\text{on } ?x ?y)), F_1, (\text{not } F_1) \notin \cup_{a \in \text{Act}(F_2)} E(a), F_2, (\text{not } F_2) \notin \cup_{a \in \text{Act}(F_1)} E(a)$,且存在 $F'=(\text{handempty})$,有:

$$(1) (\text{not } F') \rightarrow F_1, (\text{not } F') \rightarrow F_2, F_1 \rightarrow (\text{not } F'), F_2 \rightarrow (\text{not } F')$$

$$(2) (\text{not } P(X)) \rightarrow F'$$

由定义 3.2 可知,(not F_1)和(not F_2)即(ontable ?x)和(on ?x ?y)是互斥谓词对.

互斥谓词对的含义是:当 F_i 和 F_j 为“假”时,若用动作使谓词 F_i 为“真”,那么谓词 F_j 为“真”一定不能实现;反之亦然.

因为若先用动作使谓词 F_i 为“真”,由定义 3.2 可知, $F_j \notin \cup_{a \in \text{Act}(F_i)} E(a)$,即无论用哪个动作 a 使谓词 F_i 为“真”,在次态中谓词 F_j 都不会被实现,且 $P(X)$ 和 F' 一定为“假”.

这时,若用动作使 F_j 为“真”,则必须先使 F' 为“真”.

若要实现谓词 F' ,则必须先使 $P(X)$ 为“真”.

由派生规则 R 可知, $\phi(X)$ 必须为“真”,即又需谓词 F_j 为“真”.

此时形成一个前提条件的循环链,因此无法用动作实现谓词 F_j .

定义 3.2 中,在先实现谓词 F_i 的情况下,实现谓词 F_j 所产生的前提条件循环链如图 3 所示.

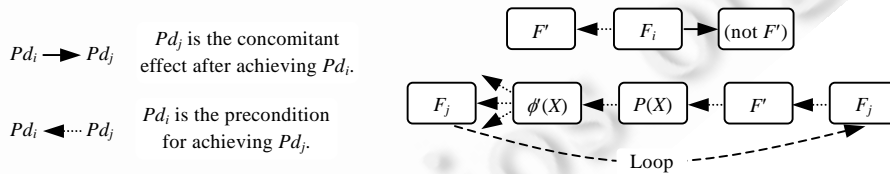


Fig.3 Loop chain of preconditions while achieving F_j
图 3 实现谓词 F_j 所产生的前提条件循环链示意图

同理,若先用动作使 F_j 为“真”,则也无法用动作实现谓词 F_i .

定理 3.2. 假设派生规划领域中存在派生规则 R :“if $\phi(X)$ then $P(X)$ ”, $\phi(X)=F_1 \wedge \dots \wedge F_i \wedge \dots \wedge F_j \wedge \dots \wedge F_m, (\text{not } F_i)$ 和 $(\text{not } F_j)$ 是互斥谓词对,如果需用动作使谓词 F_i 和 F_j 都为“真”,那么在任意状态中, F_i 和 F_j 一定不能都为“假”.

证明:假设可用动作使谓词 F_i 和 F_j 都为“真”,其中,规划状态序列为 $S_0(\text{Init}), S_1, S_2, \dots, S_g(\text{Goal})$,且存在状态 S_t ,使得 $F_i, F_j \in S_t$.

$$(1) F_i, F_j \in S_{t+1}$$

由“(not F_i)和(not F_j)是互斥谓词对”可知, $F_j \notin \cup_{a \in Act(F_i)} E(a), F_i \notin \cup_{a \in Act(F_j)} E(a)$.

所以,一定不存在动作 a_i , 执行 a_i 使谓词 F_i 和 F_j 同时为“真”.这就与“ $F_i, F_j \in S_{t+1}$ ”相矛盾.

(2) $F_i \in S_{t+1}$ 或 $F_j \in S_{t+1}$

不妨假设在状态 S_t 下执行动作 a_i , 使得 $F_i \in S_{t+1}, F_j \notin S_{t+1}$.

在实现谓词 F_i 的情况下, 由图 3.1 的分析可知, 在状态 S_{t+1} 之后都不可能用动作实现谓词 F_j .

这显然与“可用动作使谓词 F_i 和 F_j 都为“真””相矛盾.

所以,若在规划求解过程中需用动作使谓词 F_i 和 F_j 为“真”,则不存在状态 S_t , 使得 $F_i, F_j \notin S_t$, 即在任意状态中, F_i 和 F_j 一定不能同时为“假”. \square

在 BWDP 领域中, 由定义 3.2 可知, (ontable ?x) 和 (on ?x ?y) 是互斥谓词对.

由定理 3.2 可知, 如果用动作使谓词 (not (ontable ?x)) 和 (not (on ?x ?y)) 都为“真”, 那么在初始状态中 (not (on-table ?x)) 和 (not (on ?x ?y)) 不能都为“假”, 即在初始状态中, (ontable ?x) 和 (on ?x ?y) 不能都为“真”.

推论 3.1. 假设在派生规划领域中存在派生规则 R : “if $\phi(X)$ then $P(X)$ ”, $\phi(X) = F_1 \wedge \dots \wedge F_i \wedge \dots \wedge F_j \wedge \dots \wedge F_m$, (not F_i) 和 (not F_j) 是互斥谓词对, 如果在规划求解过程中需使谓词 $P(X)$ 为“真”, 那么在任意状态中 F_i 和 F_j 一定不能都为“假”.

证明: 由规划求解过程中需使谓词 $P(X)$ 为“真”和派生规则 R 可知, 需 F_k 都为“真” ($k=1, 2, \dots, m$).

由条件“(not F_i) 和 (not F_j) 是互斥谓词对”和定理 3.2 可知, 在任意状态中, F_i 和 F_j 一定不能都为“假”. \square

定义 3.3. 假设任意派生规则 “if $\phi(X)$ then $P(X)$ ”, $\phi(X) = F_1 \wedge F_2 \wedge \dots \wedge F_m$, 如果 $\exists F_u \in \{F_1, F_2, \dots, F_m\}, \forall F_j \in \{F_1, F_2, \dots, F_m\} - \{F_u\}$, 都有 (not F_u) 和 (not F_j) 是互斥谓词对, 那么称 F_u 为该派生规则的特征谓词.

定理 3.3. 假设派生规则 R : “if $\phi(X)$ then $P(X)$ ”, $\phi(X) = F_1 \wedge F_2 \wedge \dots \wedge F_m$, 若 F_u 为该派生规则的特征谓词, 则一定存在间接效果规则 $F_u \rightarrow P(X)$.

证明: 假设在状态 S_t 下有 $F_u \notin S_t$, 且需用动作来实现之.

由“ F_u 为特征谓词和定义 3.3 可知, $\forall F_j \in \{F_1, F_2, \dots, F_m\} - \{F_u\}$, 都有 (not F_u) 和 (not F_j) 是互斥谓词对.

由定理 3.2 和 “ $F_u \notin S_t$ ” 可知, $\forall F_j \in \{F_1, F_2, \dots, F_m\} - \{F_u\}$, 都有 $F_j \in S_t$.

由定义 3.2 可知, (not F_j) $\notin \cup_{a \in Act(F_u)} E(a), F_j \in S_{t+1}$. 即实现 F_u 的动作都不会使 F_j 为“假”.

由“ $F_u \in S_{t+1}, F_j \in S_{t+1}$ ”和派生规则 R 可知, 有间接效果规则 $F_u \rightarrow P(X)$ 成立. \square

在 BWDP 领域中, 派生规则 “if (not (ontable ?x)) \wedge (not (exists (?y) (on ?x ?y))) then (holding ?x)”.

由定义 3.3 可知, (not (ontable x)) 和 (not (on x y)) 都是该派生规则的特征谓词.

由定理 3.3 可得下列间接效果规则:

$$(\text{not } (\text{ontable } x)) \rightarrow (\text{holding } x), \quad (\text{not } (\text{on } x \ y)) \rightarrow (\text{holding } x).$$

由此可见, 通过对派生规则和领域规则的进一步分析, 可以利用定理 3.1 来判断目标状态中的矛盾之处, 也可利用定理 3.2 来判断初始状态中的矛盾之处. 这些领域知识对提高规划求解效率有着同样重要的作用.

4 派生规划领域的领域规则提取算法

在第 2 节中, 我们定义了基本谓词与派生谓词之间的间接效果规则和阻碍效果规则, 它们是通过派生规则提取的领域规则, 与通过动作模型提取的领域规则一起共同构成了派生规划领域的基本规则. 这种仅通过领域描述提取领域知识的提取策略具有领域无关性.

在派生规划领域中, 其谓词可分为基本谓词和派生谓词. 本文在设计领域知识提前策略时, 对这两类不同的谓词作不同的处理. 具体策略如下: 对基本谓词提取基本规则 (直接伴随规则和直接阻碍规则), 对派生谓词提取效果规则 (间接效果规则和阻碍效果规则).

根据上述领域知识提取策略的设计思想, 其领域规则提取算法 *GetDomainRule* 描述如下:

输入: 派生规划领域描述 *Dom*;

输出: 伴随规则集 *SC* (set of concomitant rules), 阻碍规则集 *SO* (set of obstructive rules).

```

GetDomainRule(Dom)
begin
    SC ← ∅, SO ← ∅;
    foreach P in Dom do
        if P is basic predict then
            begin
                //提取与基本谓词 P 有关的领域知识
                ActP ← {a | a ∈ Dom, P ∈ E(a)};
                CPCP ← ∩a ∈ ActP PC(a);
                CEP ← ∩a ∈ ActP E(a);
                SC ← SC ∪ {P(not P) → Pdi | Pdi ∈ CEP - {P}}; //直接伴随规则
                SO ← SO ∪ {(not Pc)(not P) - \ → P | Pc ∈ CPCP - {(not P)}}; //直接阻碍规则
            end
        else begin
            //提取与派生谓词 P 有关的领域知识
            ϕ(X)NNF ← Transform_NNF(P);
            ϕ'(X) ← Transform_CF(ϕ(X)NNF); //ϕ'(X) = F1 ∧ F2 ∧ ... ∧ Fm
            ϕ''(X) ← Transform_DF(ϕ(X)NNF); //ϕ''(X) = G1 ∨ G2 ∨ ... ∨ Gn
            SC ← SC ∪ {(not Fi)(Fi) → \ → (not P) | Fi ∈ ϕ'(X)}; //提取间接效果规则(定义 2.3)
            SC ← SC ∪ {Gj(not Gj) → \ → P | Gj ∈ ϕ''(X)};
            CPCP ← GetCPC(ϕ''(X), Dom); //计算谓词 P 的公共前提(定义 2.6)
            SO ← SO ∪ {(not Pc)(not P) - \ → P | Pc ∈ CPCP}; //提取阻碍效果规则(定义 2.7)
            foreach Fi in ϕ'(X) do //提取间接效果规则(定义 3.3)
                SC ← SC ∪ {Fi(not Fi) → \ → P | Fi is a characteristic predict};
            end
        end
    end
end
    
```

本算法已嵌入到规划器 StepByStep 之中,它增加了该规划器的处理能力.目前对一些经典的规划领域,如 BlocksWorld, Gripper 和 Monlake 等进行了领域规则提取实验.实验验证了该提取算法的可行性和有效性,所提取出的领域规则能够直观地表达谓词间的因果关系.

对一些规划领域所提取得到的领域规则个数及算法运行时间见表 1,其中,BlocksWorld 领域的具体领域规则形式见文献[9,10]和附录 B.

Table 1 Numbers of domain rules for classic domains and their derived domains

表 1 经典规划领域及其派生规划领域的领域规则个数对照表

Type	Domain	Action model		Derived rule			Time (s)
		Concomitant rule	Obstructive rule	Indirect effect rule	Indirect cond-effect rule	Obstructive effect rule	
Normal domain	BlocksWorld	22	11				0.008 5
	Gripper	14	13				0.008 9
	Monlake	10	35				0.011 2
	Ferry	14	13				0.008 6
Derived domain	BWDP	12	9	5	1	3	0.007 7
	GripperDP	6	9	2	0	2	0.008 1
	MonlakeDP	10	29	4	0	0	0.009 9
	FerryDP	6	9	2	0	1	0.007 5

(1) 在领域规则数量方面

与非派生规划领域相比,本算法从派生规划领域的动作模型中提取出的领域规则个数都有所减少.这是因为,在派生规划领域中,动作模型只描述动作的直接效果,动作的间接效果用派生规则来描述,动作模型所隐含

的信息有所减少.

(2) 在算法运行时间方面

与非派生规划领域相比,本算法对派生规划领域的运行时间略有下降.本文算法在提取领域知识时需从动作模型和派生规则中提取相应的领域知识.表面上看,对派生规划领域所花时间应略高于非派生规划领域所花时间,但实际情况正好相反.这是因为派生规则简化了动作模型的定义,从而使从动作模型提取领域知识所花费的时间有所减少.在从派生规则提取领域知识时,虽然需要对其触发条件进行逻辑变换,但其触发条件的规模都非常有限,这部分所增加的时间非常少.因此,总体上从派生规划领域提取领域知识所花费的时间会有所下降.

实验环境:

(1) 硬件环境:Dell Latitude D630, Intel Core™ 2 Dou 2.20GHz,内存 1.99G.

(2) 模拟环境:VMware Workstation 5.5, Red Hat Linux 7.32.

5 结 论

本文以含派生规则的 STRIPS 规划领域为研究对象,在分析动作模型和派生规则的基础上给出了一种自动提取领域知识的提取策略,利用该策略可以提取出隐含在领域描述中的领域知识.本文的领域知识提取策略已嵌入在规划器 StepByStep 之中,并对一些含派生规则的规划领域进行了实验.实验所得的领域知识能够直观地反映领域描述中所隐含的领域知识.这些领域知识可用于减少规划求解过程中派生规则的逻辑推导、指导规划动作的启发式搜索等.在规划问题求解过程中,运用所得到的领域知识来提高规划效率将是我们后续的研究工作.

References:

- [1] McDermott D. The 1998 AI planning systems competition. *AI Magazine*, 2000,21(2):35–55.
- [2] Bacchus F, Kabanza F. Using temporal logics to express search control knowledge for planning. *Artificial Intelligence*, 2000, 116(1-2):123–191. [doi: 10.1016/S0004-3702(99)00071-5]
- [3] Kvarnström J, Doherty P. TALplanner: A temporal logic based forward chaining planner. *Annals of Mathematics and Artificial Intelligence (AMAI)*, 2001,30(1-4):119–169.
- [4] Wu KH, Jiang YF. Planning with domain constraints based on model-checking. *Journal of Software*, 2004,15(11):1629–1640 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/15/1629.htm>
- [5] Fox M, Long D. The automatic inference of state invariants in TIM. *Journal of AI Research*, 1998,9:367–421. <http://www.jair.org/contents.htm>
- [6] Gerevini A, Serina I. LPG: A planner based on local search for planning graphs with action costs. In: *Proc. of the 6th Int'l Conf. on Artificial Intelligence Planning and Scheduling Systems (AIPS 2002)*. AAAI Press, 2002. 13–22.
- [7] Gerevini A, Saetti A, Serina I. An empirical analysis of some heuristic features for local search in LPG. In: *Proc. of the 14th Int'l Conf. on Automated Planning and Scheduling (ICAPS 2004)*. AAAI Press, 2004. 171–180.
- [8] Gerevini A, Saetti A, Serina I, Toninelli P. Fast planning in domains with derived predicates: An approach based on rule-action graphs and local search. In: *Proc. of the 20th National Conf. on Artificial Intelligence (AAAI 2005)*. AAAI Press, 2005. 1157–1162.
- [9] Wu XJ, Jiang YF, Ling YB. Strategy of extracting domain knowledge for STRIPS world. *Journal of Software*, 2007,18(3):490–504 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/490.htm> [doi: 10.1360/jos180490]
- [10] Wu XJ, Jiang YF, Ling YB. Researches on relations of effect of action for STRIPS domain. *Journal of Software*, 2007,18(6): 1328–1349 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/1328.htm> [doi: 10.1360/jos181328]
- [11] Wu XJ, Jiang YF, Ling YB. Research and development of StepByStep planner. *Journal of Software*, 2008,19(9):2243–2264 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/2243.htm> [doi: 10.3724/SP.J.1001.2008.02243]
- [12] Fox M, Long D. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of AI Research*, 2003,20: 61–124. <http://www.jair.org/contents.htm>

- [13] Edelkamp S, Hoffmann J. PDDL2.2: The language for the classical part of the 4th int'l planning competition. Technical Report, No.195, Freiburg: Albert Ludwigs Universitat, Institut fur Informatik, 2004.
- [14] Jiang ZH, Jiang YF. An improved method for calculating activation sets of action derived preconditions. Chinese Journal of Computers, 2007,30(12):2061–2073 (in Chinese with English abstract).
- [15] Thiébaux S, Hoffmann J, Nebel B. In defense of PDDL axioms. Artificial Intelligence, 2005,168(1-2):38–69. [doi: 10.1016/j.artint.2005.05.004]

附中文参考文献:

- [4] 吴康恒,姜云飞.基于模型检测的领域约束规划.软件学报,2004,15(11):1629–1640. <http://www.jos.org.cn/1000-9825/15/1629.htm>
- [9] 吴向军,姜云飞,凌应标.基于 STRIPS 的领域知识提取策略.软件学报,2007,18(3):490–504. <http://www.jos.org.cn/1000-9825/18/490.htm> [doi: 10.1360/jos180490]
- [10] 吴向军,姜云飞,凌应标.STRIPS 规划领域中动作效果关系的研究.软件学报,2007,18(6):1328–1349. <http://www.jos.org.cn/1000-9825/18/1328.htm> [doi: 10.1360/jos181328]
- [11] 吴向军,姜云飞,凌应标.智能规划器 StepByStep 的研究和开发.软件学报,2008,19(9):2243–2264. <http://www.jos.org.cn/1000-9825/19/2243.htm> [doi: 10.3724/SP.J.1001.2008.02243]
- [14] 蒋志华,姜云飞.一种计算动作派生前提的激活集的改进方法.计算机学报,2007,30(12):2061–2073.

附录 A. BWDP 规划领域的描述

```
(define (domain BlocksWorld-DerivedPredicates)
  (:requirements :strips :derived-predicates)
  (:predicates (on ?x ?y) (ontable ?x) (handempty) (clear ?x) //Basic Predicates
               (above ?x ?y) (holding ?x)) //Derived Predicates
  (:derived (above ?x ?y) (or (on ?x ?y) (exists (?z) (and (on ?x ?z) (above ?z ?y))))))
  (:derived (holding ?x) (and (not (ontable ?x)) (not (exists (?y) (on ?x ?y))))))
  (:action pick-up
  :parameters (?x)
  :precondition (and (clear ?x) (ontable ?x) (handempty))
  :effect (and (not (ontable ?x)) (not (clear ?x)) (not (handempty))))
  (:action put-down
  :parameters (?x)
  :precondition (holding ?x)
  :effect (and (clear ?x) (handempty) (ontable ?x)))
  (:action stack
  :parameters (?x ?y)
  :precondition (and (holding ?x) (clear ?y))
  :effect (and (not (clear ?y)) (clear ?x) (handempty) (on ?x ?y)))
  (:action unstack
  :parameters (?x ?y)
  :precondition (and (on ?x ?y) (clear ?x) (handempty))
  :effect (and (clear ?y) (not(clear ?x)) (not (handempty)) (not (on ?x ?y))))))
```

附录 B. BWDP 规划领域的领域规则

应用领域知识提取算法可得如下领域规则:

(1) 12 条直接伴随规则

$(on\ x\ y) \rightarrow (\text{not}\ (clear\ y))$	$(on\ x\ y) \rightarrow (clear\ x)$
$(on\ x\ y) \rightarrow (\text{handempty})$	$(\text{not}\ (on\ x\ y)) \rightarrow (clear\ y)$
$(\text{not}\ (on\ x\ y)) \rightarrow (\text{not}\ (clear\ x))$	$(\text{not}\ (on\ x\ y)) \rightarrow (\text{not}\ (\text{handempty}))$
$(ontable\ x) \rightarrow (clear\ x)$	$(ontable\ x) \rightarrow (\text{handempty})$
$(\text{not}\ (ontable\ x)) \rightarrow (\text{not}\ (clear\ x))$	$(\text{not}\ (ontable\ x)) \rightarrow (\text{not}\ (\text{handempty}))$
$(\text{handempty}) \rightarrow (clear\ x)$	$(\text{not}\ (\text{handempty})) \rightarrow (\text{not}\ (clear\ x))$

(2) 9 条直接阻碍规则

$(\text{not}\ (\text{holding}\ x)) \rightarrow (\text{not}\ (on\ x\ y))$	$(\text{not}\ (clear\ y)) \rightarrow (\text{not}\ (on\ x\ y))$
$(\text{not}\ (clear\ x)) \rightarrow (\text{not}\ (on\ x\ y))$	$(\text{not}\ (\text{handempty})) \rightarrow (\text{not}\ (on\ x\ y))$
$(\text{not}\ (\text{holding}\ x)) \rightarrow (\text{not}\ (ontable\ x))$	$(\text{not}\ (clear\ x)) \rightarrow (\text{not}\ (ontable\ x))$
$(\text{not}\ (\text{handempty})) \rightarrow (\text{not}\ (ontable\ x))$	$(\text{not}\ (\text{holding}\ x)) \rightarrow (\text{not}\ (\text{handempty}))$
$(\text{not}\ (clear\ x)) \rightarrow (\text{not}\ (\text{handempty}))$	

(3) 5 条间接效果规则

$(on\ x\ y) \rightarrow (\text{not}\ (\text{holding}\ x))$	$(ontable\ x) \rightarrow (\text{not}\ (\text{holding}\ x))$
$(\text{not}\ (on\ x\ y)) \rightarrow (\text{holding}\ x)$	$(\text{not}\ (ontable\ x)) \rightarrow (\text{holding}\ x)$
$(on\ x\ y) \rightarrow (\text{above}\ x\ y)$	

(4) 1 条条件间接效果规则

$(on\ x\ z) \wedge (\text{not}\ (on\ x\ z)) \wedge (\text{above}\ z\ y) \rightarrow (\text{above}\ x\ y)$

(5) 3 条阻碍效果规则

$(\text{not}\ (clear\ x)) \rightarrow (\text{not}\ (\text{holding}\ x))$	$(\text{not}\ (\text{handempty})) \rightarrow (\text{not}\ (\text{holding}\ x))$
$(\text{not}\ (\text{holding}\ x)) \rightarrow (\text{not}\ (\text{above}\ x\ y))$	



边茵(1982-),女,吉林通化人,博士,主要研究领域为知识工程与应用,智能规划和调度.



吴向军(1965-),男,博士,副教授,主要研究领域为人工智能,算法设计,网络应用.



姜云飞(1945-),男,博士,教授,博士生导师,主要研究领域为自动推理,智能规划,基于模型诊断.



梁瑞仕(1982-),男,博士生,主要研究领域为智能规划,启发式搜索.