维普资讯 http://www.cqvip.com

基于与状态无关的激活集的包含派生谓词的规划问题求解

蒋志华[1] 姜云飞[2]

(暨南大学计算机系 广州 510632)[1] (中山大学软件研究所 广州 510275)[2]

摘 要 派生谓词是PDDL2.2语言的新特性之一。在2004年的规划大赛IPC-4上,许多规划系统都无法求解包含派生谓词的两个标准竞赛问题。在经典规划中,派生谓词是指不受领域动作直接影响的谓词,它们在当前状态下的真值是在封闭世界假设中由某些基本谓词通过领域公理推导出来的。本文提出一种新的方法来求解包含派生谓词的规划问题,即用与状态无关的激活集来取代派生谓词用于放宽式规划中。

关键词 智能规划,派生谓词,规则图,激活集,放宽式规划

Planning with Derived Predicates Based on Their State-independent Activation Sets

JIANG Zhi-Hua[1] JIANG Yun-Fei[2]

(Dep. of Computer Science, Jinan University, Guang zhou 510632)[1]

(Software Research Institution, Zhongshan University, Guangzhou 510275)[2]

Abstract Derived predicate is one of two new features of PDDL2.2 language, and two domains with derived predicates in IPC-4 are very challenging to many planning system. In classical planning, derived predicates are predicates that are not effected directly by the domain actions, and their truth in the current state is inferred from that of some basic predicates via some domain axioms under the closed world assumption. Similar to rule graph and state-dependent activation set proposed by LPG-td, we propose another approach to planning with derived predicates where state-independent (not state-dependent) activation sets of a derived predicate are used in relax-plan heuristics.

Keywords Intelligent planning, Derived predicate, Rule graph, Activation set, Relax-plan

1 引言

智能规划问题是给出领域描述和问题描述,要求一个动作序列,使得问题的初始状态可以通过应用这个动作序列到达目标状态。两年一届的国际智能规划大赛IPC(International Planning Competition)是这个研究领域最顶级的学术会议和规划系统竞赛。IPC的标准竞赛语言每次都进行一定的扩充和增加新的特性,引导着规划领域研究的主流和热点问题。PDDL2.2(Planning Domain Description Language, Version 2.2)语言[6]是2004年国际规划竞赛IPC-4所采用的标准竞赛语言,它在原来的PDDL2.1的基础上,增加了两个新的特性:派生谓词(Derived Predicate)和时间初始谓词(Time Initial Literal)。在经典规划中,谓词分成两类:基本谓词(Basic Predicate)和派生谓词。二者的差别是,基本谓词可以出现在动作的效果中,而派生谓词不能作为动作的直接效果,只能出现在动作的前提和目标中[1]。因此,派生谓词不受动作的直接影响,它们在当前状态下的真值是由封闭世界假设中某些基本谓词通过领域公理推导出来的。

PDDL语言的最初版本,即PDDL1.0[4],其实也提供了领域公理和派生谓词的表示,但是这些特性从来没有在IPC竞赛系列中使用。PDDL2.1[5]将PDDL1.0扩展到时态规划和数值规划,但是作为2002年国际规划大赛IPC-3的标准竞赛语言,没有一个竞赛问题包含领域公理和派生谓词。在2004年,PDDL2.2正式地将派生谓词引入到确定性规划领域(Deterministic Planning Track)的竞赛问题中,其中包含派生谓词的两个标准问题是:PSR-Middle和PROMELA。在参加决赛的19个规划系统中,只有4个规划系统(LPG-td、SGPlan、Marvin、Downward)能够求解这两种问题,而其中后两种规划系统的求解效果并不理想。

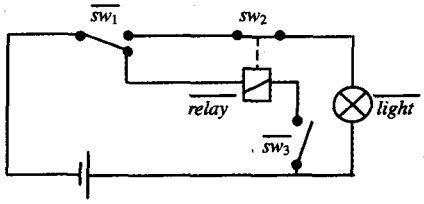


图1 由5个状态谓词所描述的电路

```
(: derived (light) (and sw1 sw2))
(: derived (relay) (and (not_sw1) sw3))
(: derived (not_sw2) (relay))
```

图2 因果规则转化为派生谓词

派生谓词提供了一种准确而且自然的方式来表达动作的非直接效果。然而,动作的非直接效果一直属于动作的衍生问题(Ramification Problem)所研究的范畴。文[8]中描述了一个经典的衍生问题的示例(如图1所示):由1个电池、3个开关、1个继电器和1个灯泡所组成的电路。假设在图中所示的初始状态下,我们把开关1打到另一个接触点,那么这个动作的直接效果是状态谓词 sw_1 的真值为true。然而基于常

蒋志华 博士研究生,讲师,主要研究方向为知识工程、智能规划;姜云飞 教授,主要研究方向为模型诊断、智能规划。

维普资讯 http://www.cqvip.com

识，我们知道拨动开关1会使得上面的电路连通，那么灯泡会亮。因此，灯泡亮了是拨动开关1的非直接效果，这个效果来自于因果规则“$sw_1 \wedge sw_2 \Rightarrow light$”。同样地，当我们把开关3闭合，这个动作的直接效果是状态谓词 sw_3 的真值为 true，非直接效果是继电器被激活并且迫使开关2跳开。这些非直接效果分别来自于因果规则“$\neg sw_1 \wedge sw_3 \Rightarrow relay$”和“$\neg sw_1 \wedge sw_3 \Rightarrow relay$”。在 PDDL2.2 语言中，我们可以用派生谓词来表示这些非直接效果。例如，上面3个规则通过引入简单的 STRIPS 谓词可以转化成图2所示的形式。

已经有一些方法来处理派生谓词，但是各自都有一定的局限性。例如，一种方法是把派生谓词编译成谓词公式，但是其复杂性随着领域描述规模的增加而骤然增长[1]。另一种方法是 Gazen 和 Knoblock 提出的预处理算法，把领域公理转化成等价的“演算”操作子，但是这种方法有时会导致无效规划，即转化后的问题与初始问题并不等价[10, 11]。在 IPC-4 上，在能够求解带有派生谓词的规划问题的规划系统中，LPG-td 的表现最好，它使用一种特殊的数据结构——规则动作图(Rule-action Graphs)来描述搜索空间，并且提出了派生谓词的激活集(Activation Set)概念。然而，在 LPG-td 的搜索算法中，在规则图上计算激活集要耗费大量的时间，因为一旦当前状态发生变化，派生谓词的激活集需要重新计算。因此，在本文中，我们提出与状态无关的激活集的概念，可以在预处理阶段一次性地计算它们，或者在放宽式规划中当需要的时候再计算。

在本文下面的内容中，我们先引入规则图来表示派生谓词，并且提出计算与状态无关的激活集的算法。然后，我们在前向状态空间搜索中，扩展放宽式规划算法来应用与状态无关的激活集。最后是结语和今后的工作。

2 规则图和与状态无关的激活集

如前所述，派生谓词不出现在任何领域动作的(增加或者删除)效果中。派生谓词的真值由以下形式的公理来确定：if $\Phi(x)$ then $P(x)$，其中，$P(x)$ 表示派生谓词，x 是变量向量，$\Phi(x)$ 是一阶谓词公式并且满足其否定范式 NNF(negated normal form)不包含任何形式的派生谓词。例如，图3描述了积木块世界中一个经典的派生谓词“above”：积木块 x 处于积木块 y 的上方，当且仅当积木块 x 正好放在积木块 y 的上面，或者积木块 x 放在积木块 z 的上面而积木块 z 位于积木块 y 的上方。

在 IPC-4 上，每个包含派生谓词的问题领域都具有两种形式：“strips＋derived predicates”以及“adl＋derived predicates”。二者的区别是，strips 风格的问题只涉及原子(Ground)派生谓词，而 adl 风格的问题还可以包含量词(Quantifier Variables)和条件效果(Condition Effects)等。但无论是哪一种形式，给定一个规则“if $\Phi(x)$ then $P(x)$”和一组常量 $c(|x|=|c|)$，通过文[2]中的转换机制，我们都可以得到一组等价的原子规则 R，即实例化的规则。

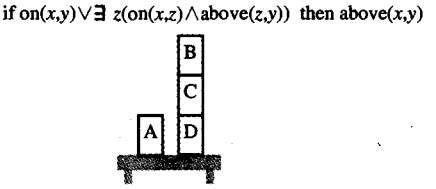


图3 积木块世界的派生谓词“above”

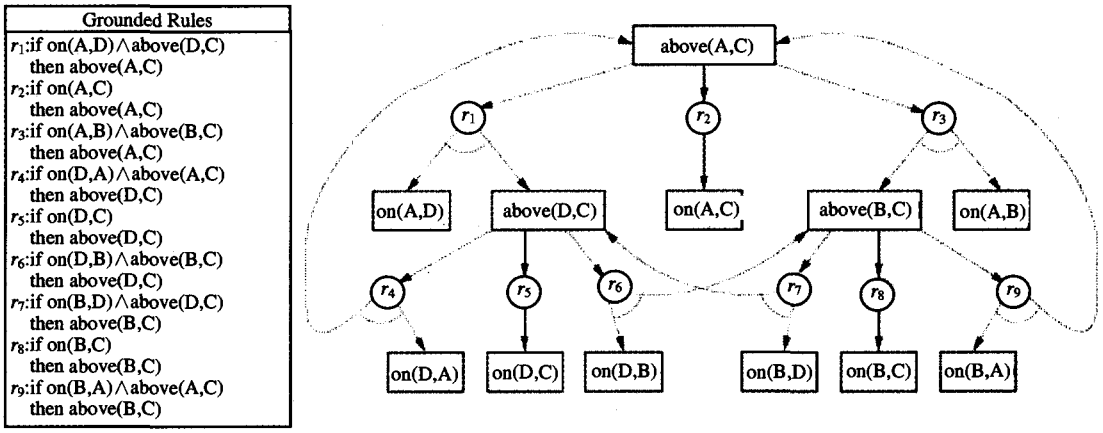


图4 一个规则图实例

在包含变量的规则用领域对象进行实例化之后，规则集 R 只包含基本事实(Basic Facts)和派生事实(Derived Facts)，例如，on(x, y)是基本谓词，因此 on(A,D)是基本事实，而 above(x, y)是派生谓词，因此 above(A,D)是派生事实。为了跟 LPG-td 所定义的激活集进行对比，我们采用了文[2]中的规则图例子(如图4所示)。由实例化规则组成的规则集合 R 可以转化成一张与或图，即规则图(Rule-graph)。一个规则图由两类结点组成：与结点(规则结点)和或结点(规则结点)，与结点是由基本事实标记的叶子结点或者由派生事实标记的内部结点，而或结点是由原子规则标记的内部结点。对于一个规则结点来说，它的入边(只有一条)指向由该规则所推导的派生事实，而它的出边(若干条)则指向该规则的所有触发条件(由基本事实或派生事实组成)。例如，在图4中，派生事实“above(A,C)”可以由规则 r_1、r_2 或者 r_3 来分别推导，而规则 r_1 的触发条件是 on(A,D)和 above(D,C)同时成立。

下面我们先看看 LPG-td 所提出的派生谓词的激活集的概念。在 LPG-td 中，激活集的定义如下：

定义5 在动作图的某一层 L 中，如果存在一个还未支持的派生前提 d，那么 d 的激活集是一个最小的基本事实集合 F，使得 $S(L) \cup F \models^R \Psi_d$，其中 Ψ_d 表示由 d 标志的派生谓词，$S(L)$ 表示组成层 L 的状态的谓词集合”[2]。

从这个定义我们可以看到，由于最小化的要求，该层中的所有基本事实即使是某个派生事实的触发条件，但却不属于该派生事实的激活集。我们用 Σ 来表示所有这样的 F 的集

维普资讯 http://www.cqvip.com

合。例如，在图 4 中，派生事实“above(A,C)”在图 3 所示的状态下，其激活集集合 Σ 是{{on(A,D),on(D,C)}，{on(A,D),on(D,B),on(B,C)}，{on(A,C)}，{on(A,B)}}。其中，{on(A,B)}是派生事实“above(A,C)”的一个激活集，但是包含该激活集的完整的触发条件是{on(A,B),on(B,C)}(通过应用规则 r_3 和 r_8)，基本事实“on(B,C)”由于属于当前状态因而没有包含在激活集中。然而，如果当前状态发生变化，例如我们交换一下木块 C 和木块 D 的位置，当前状态变为{table(A)，table(C)，on(B,D)，on(D,C)}，则派生事实“above(A,C)”的激活集集合 Σ 变为{{on(A,D)}，{on(A,C)}，{on(A,B)}}。因此，LPG-td 所定义的派生谓词的激活集是与当前状态紧密相关的，如果状态发生了变化，那么激活集集合需要重新进行计算。实际上，在 LPG-td 的放宽式规划中，存在着大量这样耗费时间的重新计算，因此，我们需要定义与状态无关的派生谓词的激活集。

定义 1 给定一个由原子规则集合 R 生成的规则图，派生谓词 d 的与状态无关的激活集 SIAS(State-Independent Activation Set)是一个最小的基本事实集合 F，使得 $F \models^R \Psi_d$，其中 Ψ_d 表示由 d 标志的派生谓词。

接下来，我们提出一个深度优先算法 SIAS-search(如图 5 所示)，来计算在一个给定的规则图中某个派生谓词的所有激活集组成的集合 Σ。在该算法的四个参数中，变量 d 表示一个派生事实结点，集合 A 记录在生成一个激活集时所有可能的成员，Path 表记录已访问的派生事实结点，Open 表保存待访问的结点。该算法的输出是集合 Σ。

Algorithm 1

```
Algorithm 1 SIAS-search (d, A, Path, Open)
1.  For each successor r of d do
2.  { Open ← Open ∪ {r} ;
3.  While Open ≠ ∅ do
4.  { x = first_element (Open) ;
5.  If x is a rule node Then Open ← Open ∪ {first_successor (x) };
6.  Else If x is a basic fact Then
7.  { A ← A ∪ {x} ;
8.  If the antecedent of x has other unvisited successor x'
9.  Then Open ← Open ∪ {x'}
10. Else  Σ ← Σ ∪ {A};
11. }
12. Else If x is a derived fact Then
13. If x ∉ Path Then
14. {  SIAS-search (x , A, Path ∪ {x} , Open);
15. If the antecedent of x has other unvisited successor x'
16. Then Open ← Open ∪ {x'};
17. }
18. }
19. For each x' in A which is the successor of r Do
20. A = A - {x'};
21. }
```

图 5 计算所有与状态无关的激活集的深度优先算法

算法 SIAS-search 可以对规则图进行全面的搜索，寻找所有可能的激活集，而与具体的状态无关。其中，函数 first_element 是取 Open 表的第一个元素 x。如果 x 是一个规则结点，那么 x 的一个触发条件首先进入到 Open 表中，剩余的触发条件可以由语句 8～9、15～16 也逐一进入 Open 表。如果 x 是一个基本事实结点，那么它称为集合 A 中的元素。直到 A 中所有的元素构成一个完整的触发条件，A 才成为一个新的激活集，添加到集合 Σ 中。如果 x 是一个派生事实结点并且不出现在 Path 表中，那么搜索过程通过语句 14 进行递归，反之，如果 x 出现在 Path 表中，则它应该被剪枝以避免无限循环。最后，在找到每一个激活集的时候，集合 A 应该保留还有可能构成新的激活集的元素而去掉无用的元素(语句 19～20)。

下面，我们回过头来看看前面所举的例子。例如在图 4 所示的规则图中，通过使用算法 SIAS-search，可以得到派生事实“above(A, C)”的与状态无关的激活集集合；见表 1(1)，LPG-td 所求出的激活集也列举如下，见表 1(2)和(3)。

通过对比表 1(2)和(3)，可以看到对于同一派生事实，LPG-td 所求出的激活集是与当前状态相关的。而通过对比表 1(1)和(2)，可以看到 SIAS-search 算法所求出的激活集是与任何状态无关的。仔细比较二者的区别，我们可以发现基本事实“on(B,C)”已存在图 3 所示的状态中，因此出现在表 1(1)的激活集{on(A,B),on(B,C)}缩减为表 1(2)中的激活集{on(A,B)}。另外，由于最小化的需求，表 1(1)的激活集{on(A,B),on(B,D)，on(D,C)}也没有出现在表 1(2)中。通过这个例子，我们可以看到，一般地，与状态无关的激活集集合要比与状态相关的激活集集合大一些，但是通过计算与状态无关的激活集，我们可以避免当状态经常发生变化的时候需要重新计算激活集。而在规划求解的过程中，当前状态无时无刻不在发生变化。此外，关于算法的复杂性方面，在最坏情况下，激活集集合 Σ 的规模会随着原子规则集合中所涉及的对象 n 的数目指数级地增长。但是幸运的是，在 IPC-4 的大多数竞赛问题中，激活集的数目小于 $O(n^2)$[2]。

3 在放宽式规划中使用与状态无关的激活集

在 IPC-4 中，许多规划系统由于不支持 PDDL 语言的新特性，因此求解的竞赛问题只限于 STRIPS 问题。例如，YAHSP 规划系统(“Yet Another Heuristic Search Planner”)[12]在尝试的 279 个问题中，能够解出 255 个问题，成功率高达 91%，但是它不能处理数值规划(Metric Planning)、时态规划(Temporal Planning)、派生谓词以及时态初始谓词等等，然而，像 LPG-td 等规划系统支持上述所有的特性，虽然成功率只有 77%左右，但是能够解出的问题接近 1000 个，求解问题的规模远远大于 YAHSP。在另一方面，LPG-td 虽然提出新的方法来处理派生谓词——计算激活集，但是并非在所有竞赛问题中都是有利的，在某些问题上，LPG-td 的表现比 SGPlan、DOWNWARD 等要稍逊一筹，因为它的启发式有时会特别地低效，而后两个规划系统也可以处理派生谓词，均采用编译的方法。因此，PDDL2.2 语言的新特性对于大多数规划系统来说，仍然很有挑战性。

启发式规划(Heuristic Planning)一直是求解 IPC 系列竞赛中标准问题的非常有效的方法，至今为止，大多数规划系统都采用了基于启发式的放宽式规划(Relaxed-plan)技术，即 FF 规划系统[13]最先引入的，通过计算后继状态的启发值，从中选择最接近目标的来作为下一状态。放宽式的意思是通过忽略动作的所有删除效果来从规划图中抽取出规划解，这样可以避免动作之间的互斥关系而快速地获得后继状态的启发值。基于前向状态空间搜索的启发式规划采用放宽式技术可以使得搜索过程从初始状态到达目标状态。

接下来，我们提出一种算法 DP-relax-plan *(如图 6 所示)，将与状态无关的激活集应用到包含派生谓词的规划领域的放宽式规划中。其中，每个状态不仅包含基本事实，而且还包含大量的由领域规则所推导的派生事实。该算法基于 FF 规划系统的放宽式算法，可以用于任何前向状态空间搜索的规划系统。

维普资讯 http://www.cqvip.com

表 1

	派生事实"above(A, C)"的激活集集合 Σ	
SIAS-search	(1){{on(A,D), on(D,C)}, {on(A,D), on(D,B), on(B,C)}, {on(A,C)}, {on(A,B), on(B,D), on(D,C)}, {on(A,B), on(B,C)}}	与任何状态无关
LPG-td	(2){{on(A,D),on(D,C)}, {on(A,D),on(D,B),on(B,C)}, {on(A,C)}, {on(A,B)}}	与状态{table(A), table(D), on(B,C), on(C,D)}相关
	(3) {{on(A,D)}, {on(A,C)}, {on(A,B)}}	与状态{table(A), table(C), on(B,D), on(D,C)}相关

Algorithm 2

```
Algorithm 2 DP-relax-plan (I, G)
1.  S ← I ∪ D(I, R);
2.  level ← 0;
3.  For each action a which is applicable in S Do
4.  S' ← S ∪ Add(a);
5.  level ← level + 1;
6.  S'' ← S' ∪ D(S', R);
7.  If S'' doesn't contain G , Then S ← S'' , GOTO 3
8.  Else extract-relax-plan(I, G, ∅) ;
    Extract-relax-plan (I, G, Act)
1.  If G contains derived predicates Then G← Remove-derived(G) ;
2.  select an action set A in the (level -1) layer in the relaxed plan graph ,
    and A is a minimal set whose members can support furthest the basic facts of G ;
3.  G ← (G- {add(a) | a ∈A}) ∪{pre(a) | a ∈A};
4.  Act ← Act ∪ {A};
5.  level ← level - 1;
6.  If the level > 0 Then GOTO 1;
7.  If G contains I Then return Act.
    Remove-derived(S)
1. For each derived fact d in S Do
2. {If d doesn't be marked in Lookup table Then
3.    {SIAS-search (d,∅,{d}, ∅) ;
4.       Mark d in Lookup;
5.       Lookup ← Lookup ∪{(d, Σ) };
6.    }
7.  S'← (S - {d})∪ best-SISA(d);
8.}
9. Return S'.
```

图 6 将与状态无关的激活集用于放宽式规划的算法

算法 DP-relax-plan 由两部分组成:①以放宽式的方式扩展规划图;②在放宽规划图上抽取解。$D(I, R)$是集合 I 通过规则集合 R 推导出来的所有派生事实的集合(详细过程请见文献[1])。函数"Remove-derived"是用派生事实的最佳激活集来替换派生事实,因为在解提取阶段,需要利用动作的增加效果来提取动作,而派生事实不出现在任何动作的效果中,所以需要用它的激活集来进行替换。派生事实的最佳激活集可以简单的定义为包含基本事实的数目最少的激活集。此外,因为我们所定义的激活集是与状态无关的,所以我们构造一个查找表来存储已经计算过的派生事实的激活集,在需要的时候进行查找以避免重复计算。当然,当选择最佳激活集并不能导致最终构成一个合法的动作序列,我们可以回到函数"Remove-derived"的第 7 步来选择另一个激活集来重新进行解提取。与我们的方法不同的是,在放宽式规划中,LPG-td 通过调用"And-Search(g,∅,∅,∅, I∪F)"[2,3]不断地计算激活集,其中,g 是派生事实,$I \cup F$ 是当前状态。

最后,我们通过一个例子来说明以上算法。假设我们有一个如表 2 所示的规划问题,动作模型在 Domain. pddl 中描述,问题模型在 Problem. pddl 中描述。其中,以"P"开头的是基本事实,以"D"开头的是派生事实,共有 4 个动作 A、B、C、D,"NOT"表示删除效果。因为要计算的是后继状态的启发值,因此初始状态在下表中不需给出。

现在我们要计算后继状态 $I=\{P1, P2, P3\}$的启发值,其放宽式规划图如图 7 所示。在该图中,命题层和动作层交替扩展,命题层包含基本事实和派生事实,每个派生事实用带箭头的虚线指向它的与状态无关的激活集(也在图 7 右下角的查找表给出)。由于是放宽式规划,因此动作的删除效果不出现图 7 中。

表 2

Domain. pddl		Problem. pddl
Action A	Action B	Initial:……
Precondition: P1, D1	Precondition: P2	Goal: P7, D3
Effect: P4, NOT P1	Effect: P5, P6	
Action C	Action E	
Precondition: P4	Precondition: P4, D2	
Effect: P7	Effect: P8, NOT P4	
(: derived (D1) (and P2 P3))	(: derived (D2)(P6))	
	(: derived (D3)(P8))	

在图 7 中,因为最后一层命题层包含目标状态中的所有命题,所以解提取过程开始。派生事实 $D3$ 首先被其激活集替换,因此目标集合变为$\{P7, P8\}$。在选择动作集合$\{C, E\}$之后,通过添加动作的前提,目标集合变为$\{P4, D2\}$。之后派生事实 $D2$ 由其激活集进行替换,目标集合变为$\{P4, P6\}$,此时可以选择动作集合$\{A, B\}$,因此当到达第 0 层命题层时,目标集合为$\{P1, P2, D1\}$。在用激活集替换到派生事实 $D1$ 之后,目标集合$\{P1, P2, P3\}$包含待估值状态的所有命题,因此解提取过程结束。待估值状态(也即后继状态)I 的启发值为 2(因为其经过两层动作能够到达目标状态)。

结语 派生谓词是 PDDL2.2 语言的新特性之一,大多数规划系统还不能很好地对其进行处理。基于规划系统 LPG-td 的工作,本文对派生谓词的激活集提出了新的定义,该定义下的激活集与具体状态无关,可以避免在状态频繁变化的情况下需要耗费大量时间计算派生谓词的激活集。本文还提出了两个算法:SIAS-search 和 DP-relax-plan,前者用来计算在给定规则图的情况下派生谓词的与状态无关的激活集集合,后者用来将这些激活集用于放宽式规划中以处理带有派生谓词的规划问题。在今后的工作中,要进一步地扩展这些算法以支持 PDDL 语言的其他特性以及提高其效率。另一项正在进行中的工作是将本文提出的算法在 FF 和 HAHSP 等启发式规划系统上实现。

参 考 文 献

1 Thiebaux S, Hoffmann J, Nebel B. In Defense of PDDL Axioms. Artificial Intelligence, 2005, 168(1-2): 38~69

2 Gerevini A, Saetti A, Serina I, et al. Fast Planning in Domains with Derived Predicates. An: Approach Based on Rule-Action Graphs and Local Search. In: Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05), AAAI-Press, Pittsburgh, USA, 2005

3 Gerevini A, Saetti A, Serina I, et al. Planning with Derived Predicates through Rule-Action Graphs and Relaxed-Plan Heuristics: [R. T. 2005-01-40]. Università degli Studi di Brescia, Dipartimento di Elettronica per l Automazione. Brescia, Italy, 2005

维普资讯 http://www.cqvip.com

4 McDermott D. PDDL-the planning domain definition language: [Technical Report]. CVC TR-98-003/DCS TR-1165. Yale Center for Computational Vision and Control, 1998

5 Fox M, Long D. PDDL2.1: An extension to PDDL for expressing temporal planning domains. J Artificial Intelligence Res, 2003, 20:61～124

6 Edelkamp S, Hoffmann J. PDDL2.2: The language for the Classic Part of the 4th International Planning Competition: [T. R. no. 195]. Institut f·ur Informatik, Freiburg, Germany, 2004

7 Hoffmann J, Edelkamp S, Englert R, et al. Towards Realistic Benchmarks for Planning: the Domains Used in the Classical Part of IPC-4 - Extended Abstract. In: Proceedings of the 4th International Planning Competition (IPC-40), June, Whistler, Canada, 2004, 7～14

8 Thielscher M. Ramification and causality. Artificial Intelligence, 1997, 89: 317～364

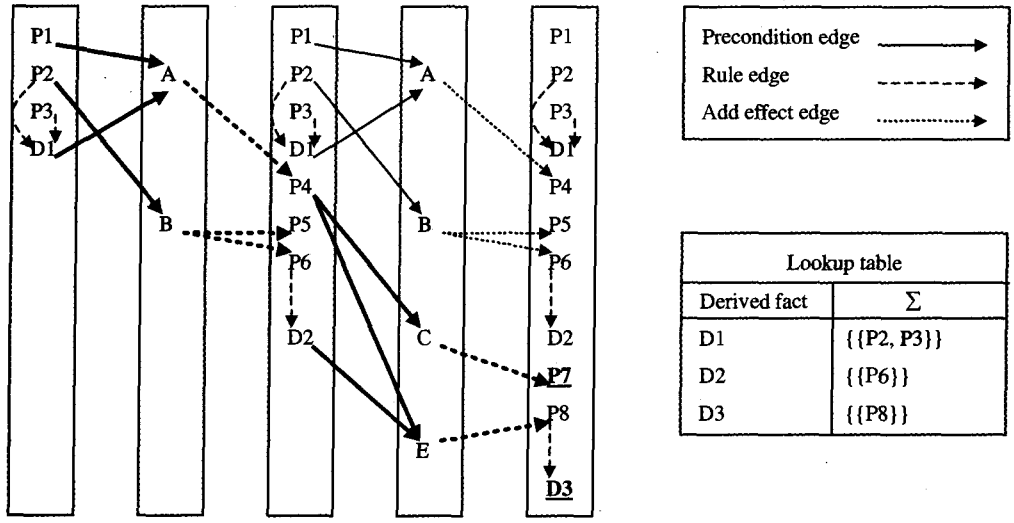
9 McCain N, Turner H. A causal theory of ramifications and qualifications. In: Proceedings IJCAI-95, Montreal, Que, 1995. 1978～1984

10 Davidson M, Garagnani M. Pre-processing planning domains containing Language Axioms. In: Grant T, Witteveen, C, eds. Proc of the 21st Workshop of the UK Planning and Scheduling SIG (PlanSIG-02), 2002. 23～34

11 Garagnani M. A correct algorithm for efficient planning with preprocessed domain axioms. In: Bramer M, Preece A, Coenen F, eds. Research and Development in Intelligent Systems XVII (Proc of ES-2000). 2000. 363～374

12 Vidal V. A lookahead strategy for solving large planning problems. IJCAI 2003. 1524～1525

13 Hoffmann J, Nebel B. The FF Planning System: Fast Plan Generation Through Heuristic Search. Journal of Artificial Intelligence Research, 2001, 14: 253～302



Lookup table

Derived fact	Σ
D1	{{P2, P3}}
D2	{{P6}}
D3	{{P8}}

图 7 放宽式规划的一个示例

带下划线的粗体部分表示目标状态的命题。蓝色粗线表示解路径。

(上接第 161 页)

参 考 文 献

1 Aggarwal CC, Han J, Wang J, et al. A framework for clustering evolving data streams. In: Proc. of VLDB, 2003

2 Aggarwal C C, Han J, Wang J, et al. A framework for projected clustering of high dimensional data streams. In: Proc. of VLDB, 2004

3 Beringer J, Hullermeier E. Online Clustering of Parallel Data Streams. Data & Knowledge Engineering, 2005

4 Guha S, Meyerson A, Mishra N, et al. Clustering Data Streams: Theory and Practice. In: TKDE special issue on clustering, Vol. 15, 2003

5 Cao F, Estery M, Qian W, et al. Density-based Clustering over an Evolving Data Stream with Noise. In: Proceedings of the 2006 SIAM Conference on Data Mining (SDM'2006)

6 Ester M, Kriegel H-P, Sander J, et al. Incremental clustering for mining in a data warehousing environment. In: Gupta A, Shmueli O, Widom J, eds. Proceedings of the 24th International Conference on Very Large Data Bases. New York: Morgan Kaufmann Publishers Inc, 1998. 323～333

7 Liu Qing-Bao, Deng Su, Lu Changhui, et al. Relative Density Based K-nearest Neighbors Clustering Algorithm. In: the Second International Conference on Machine Learning and Cybernetics, China, 2003

8 Ester M, Kriegel H-P, Sander J, et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proc. 2nd Int Conf on Knowledge Discovery and Data Mining, Portland, OR, 1996. 226～231

9 Han Jiawei, Kamber M, Fan Ming, et al. Data Mining: Conceots and Techniques. Beijing: China Mechine Press, 2001(in Chinese)

10 Ankerst M, Breunig M, Kriegel H-P, et al. OPTICS: Ordering Points To Identify the Clustering Structure. In: Proc. ACM SIGMOD '99, Int Conf. on Management of Data, Philadelphia, PA, 1999

11 Zhou Yong-Feng, Liu Qing-Bao, Deng Su, et al. An Incremental Outlier Factor Based Clustering Algorithm. In: the First International Conference on Machine Learning and Cybernetics Nov China, 2002

12 金澈,清卫宁,周傲英.流数据分析与管理综述.软件学报,2004, 15(8)

13 朱蔚恒,印鉴,谢益煌.基于数据流的任意形状聚类算法.软件学报,2006,17(3)