NewPlan例子计算过程

2015年1月19日

**本文档基于杨睿的例子文档作出相关的补充**：

**demo\_p.pddl，包含初始Knowledge Base(下面简称KB)，和goal（即推理目标，下面称goal）**：

(define (problem demo)

(:domain demo)

(:init (and

(K (and (imply (doorOpen) (inRoom))

(imply (not (lightOn)) (not(inRoom)))

)

)

(DK (inRoom))

(DK (not (inRoom)))

(DK (lightOn))

(DK (not (lightOn)))

)

)

(:goal (or

(K (inRoom))

(K (not (inRoom)))

)

)

)

**demo\_domain.pddl，**定义感知(sense)动作和物理(knock)动作：

(define (domain demo)

(:action senseL

:parameters ()

:precondition (and (DK (lightOn)) (DK (not (lightOn))))

:observe (lightOn)

)

(:action senseD

:parameters ()

:precondition (K (hasKnocked))

:observe (doorOpen)

)

(:action knock

:parameters ()

:precondition (K (lightOn))

:effect (((inRoom), (doorOpen, hasKnocked), ),

((not (inRoom)), (hasKnocked), (doorOpen))

)

)

)

计算过程：

1. 首先需要通过parse把输入的KB和goal解释成以下形式：

KB=goal = 

d = doorOpen

i = inRoom

l = lightOn

k = knock

1. 然后需要把KB化简为没有冲突的EpisDNF形式：

在程序的数据结构中EpisDNF由EpisTerm构成，两者的范式如下：

EpisTerm：。（上诉KB就是这个形式）。

EpisDNF：。

理论上若干个EpisTerm的析取式为一个EpisDNF，我们的KB和goal都需要化成EpisDNF形式，不过这里的例子的KB只能构成一个EpisTerm，其同时也是一个EpisDNF。

把一个KB化成EpisDNF的最终状态必须符合以下条件，对于：

* + 1. 其中与都必须为DNF，且；
    2. 所有的都必须符合 ，且；
    3. 任意之间都不能相互推出，即是质的。

杨睿的文档中的Page2中的②对进行处理，就是因为当前KB中的里的存在不满足的，事实上是都不符合，所以对每个都加入，然后进行化简。

1. 化简完KB之后，便使用当前最新的KB进行演进，这时需要联立demo\_domain.pddl中的各个动作了。动作分为感知动作和物理动作两种：

**对于感知动作**：

**senseL**：若其precondition能被KB满足，则对当前节点添加两个子节点，分别代表和。

对：

则把l 放进K算子中的每项中，而中能推出l的项保留，不能的则删除。

对：

则把 放进K算子中的每项中，而中能推出的项保留，不能的则删除。

**对于物理动作**：

**knock**：若当前KB可推出precondition，则看其effect。其中K算子与算子对effect的操作一致。故这里只给出K算子的操作。

假设当前的KB为

这里先关注K算子的内容，需要保证K中每项都能推出effect中的某一个condition，如这里的或，即要推出或。

而其中：

 无法推出 

 可以推出 

那么对进行裂解操作：

，其中





那么就有：



满足K中每项都能推出effect中的某一个condition。

对最新的K中的3个项分别进行演进：

：把其add部分{d, k}合取进去，先检查是否矛盾，若有矛盾，先抹去产生矛盾的部分，如，则抹去。结果变为：

：把其add部分的{k}合取进去，没有矛盾，所以结果为：。

：把其add部分{d, k}合取进去，没有矛盾，结果为：

再用最新的三项结果构造K，并化简，得到：

1. 在明晰如何操作后，接着就是Plan操作，具体看Son的论文。