E5_22336216

人工智能实验

中山大学计算机学院

人工智能

本科生实验报告

（2023学年春季学期）

课程名称：Artificial Intelligence

教学班级	DCS315	专业（方向）	计算机科学与技术（系统结构）
学号	22336216	姓名	陶宇卓

1 实验题目

实验五：PDDL求解实际问题

一、算法原理

- **算法原理**

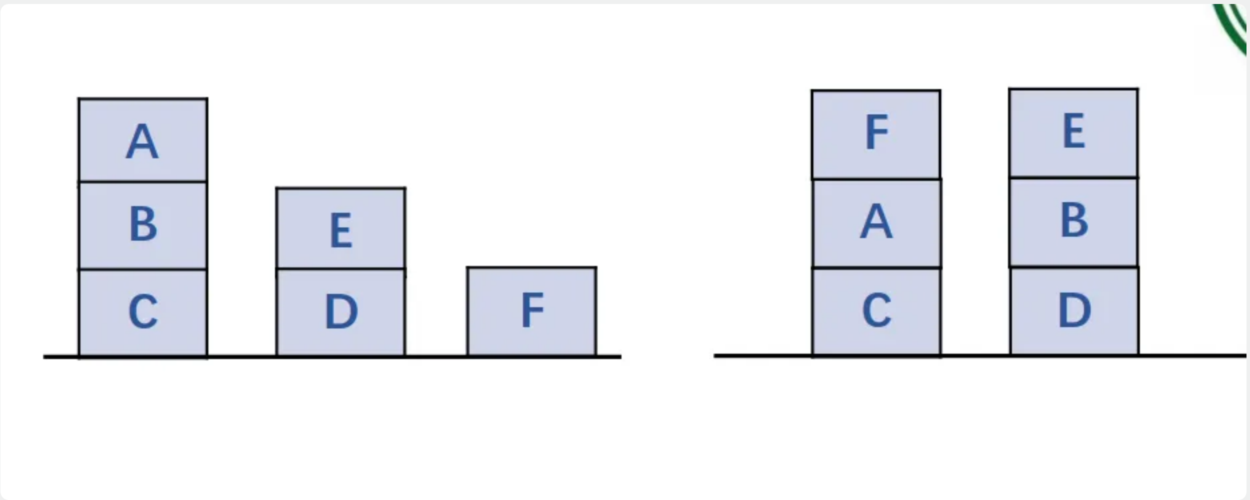
PDDL（Planning Domain Definition Language）是一种用于描述问题领域和规划任务的语言，它被广泛应用于人工智能规划领域。PDDL是一种领域描述语言，它描述了问题的初始状态、目标状态、动作集合以及动作之间的影响关系。PDDL的算法原理通常指的是基于PDDL描述的规划问题的求解算法。

- **滑块问题**

论域（Domain）：

- 物体（Objects）：物体可以是任何代表实体的东西，通常是块，如A、B、C等。
- 动作（Actions）：动作是可以在块世界中执行的操作。
    - move：移动一个块到另一个块的上方
    - moveToTable：将一个块移动到桌面上。
- 谓词（Predicates）：谓词描述了块世界中的状态。
    - (ontable ?x – physob)：x是否在桌面上
    - (clear ?x – physob)：x上方是否清空
    - (on ?x ?y – physob)：x是否在y的上方

问题定义（Problem Definition）：

- 初始状态（Initial State）：初始状态描述了问题开始时块世界的状态，包括各个物体的位置和状态。通常。
- 目标状态（Goal State）：目标状态描述了问题的目标，即我们希望达到的状态。通常，目标状态也由一系列谓词描述，表示某些条件下块世界的期望状态。
- 

- **15-puzzle**

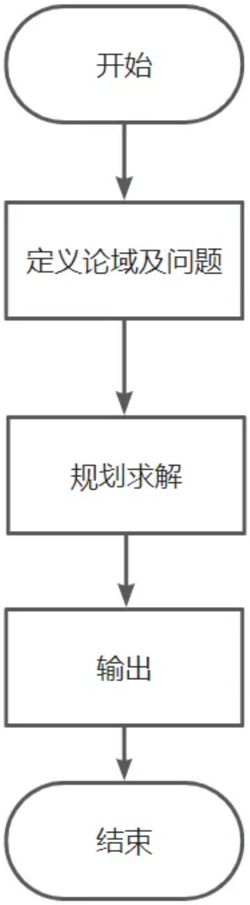
论域（Domain）：

- 类型（Types）：在这个问题中，有两种类型：数字（num）和位置（loc）。
- 常量（Constants）：只有一个常量 B，代表空白空间。
- 谓词（Predicates）：
    - （at ?n – num ?l – loc）：表示数字 ?n 在位置 ?l。
    - （adjacent ?l1 ?l2 – loc）：表示位置 ?l1 和位置 ?l2 是相邻的。
- 动作（Actions）：
        - slide：表示将一个数字滑动到相邻的空位置上。

问题定义（Problem Definition）：

- 问题名称（Problem Name）：PPT1
- 论域（Domain）：puzzle
- 对象（Objects）：
    - num1 到 num15：代表15个数字。
    - loc–1–1 到 loc–4–4：代表4x4网格中的位置。
- 初始状态（Initial State）：描述了数字的初始位置，以及空白空间的位置，以及位置之间的相邻关系。
- 目标状态（Goal State）：描述了我们希望达到的状态，即数字按照特定顺序排列在网格中，而空白空间在右下角位置。

- **流程图**



二、伪代码

block

```
定义 (domain blocks)
  需求 (:strips :typing :equality
         :universal-preconditions
         :conditional-effects)
  类型 (physob)
  谓词
     (ontable ?x - physob) ; 物体 x 在桌面上
     (clear ?x - physob) ; 物体 x 上方没有其他物体
     (on ?x ?y - physob) ; 物体 x 在物体 y 上方

  动作 move
    参数 (?x ?y - physob) ; 物体 x 和物体 y
    先决条件  (and (clear ?x) (clear ?y) (not (= ?x ?y))) ; 物体 x 和物体 y 都是清空的，并且它们不相同
    效果  (and (on ?x ?y)
               (not (clear ?y))
               (forall (?z - physob) (when (on ?x ?z) (and (not (on ?x ?z)) (clear ?z)))))
    )

  动作 moveToTable
    参数 (?x - physob) ; 物体 x
    先决条件  (and (clear ?x) (exists (?y - physob) (on ?x ?y))) ; 物体 x 是清空的，并且存在一个物体 y，在物体 x 下方
    效果  (and (ontable ?x)
               (forall (?z - physob) (when (on ?x ?z) (and (not (on ?x ?z)) (clear ?z))))) ; x在桌子上，x下面的z变成不在x下面，并且clear
  )

```

puzzle

```
定义 (domain puzzle)
  需求 (:strips :equality :typing)
  类型 (num loc)
  常量 (B - num) ; B 表示空白位置
  谓词
     (at ?n - num ?l - loc) ; 数字 n 在位置 l
     (adjacent ?l1 ?l2 - loc) ; 位置 l1 和 l2 是相邻的

  动作 slide
    参数 (?n - num ?from - loc ?to - loc) ; 数字 n 从位置 from 滑动到位置 to
    先决条件  (and (at ?n ?from) (at B ?to) (adjacent ?from ?to)) ; 数字 n 在位置 from，空白位置在位置 to，并且位置 from 和位置 to 是相邻的
    效果  (and
              (not (at ?n ?from))
              (at ?n ?to)
              (not (at B ?to))
              (at B ?from)
            )
  )
```

三、关键代码展示

> 完整代码见../code，此处只展示修改的部分

- **block.pddl**

```
(define (domain blocks)
  (:requirements :strips :typing :equality
                 :universal-preconditions
                 :conditional-effects)
  (:types physob)
  (:predicates
       (ontable ?x - physob)
       (clear ?x - physob)
       (on ?x ?y - physob))

  (:action move
           :parameters (?x ?y - physob)
           :precondition (and (clear ?x) (clear ?y) (not (= ?x ?y)))
           :effect (and (on ?x ?y)
                        (not (clear ?y))
                        (forall (?z - physob) (when (on ?x ?z) (and (not (on ?x ?z)) (clear ?z)))))
           )

  (:action moveToTable
           :parameters (?x - physob)
           :precondition (and (clear ?x) (exists (?y - physob) (on ?x ?y)))
           :effect (and (ontable ?x)
                        (forall (?z - physob) (when (on ?x ?z) (and (not (on ?x ?z)) (clear ?z)))))
  )
)
```

- **puzzle.pddl**

```
(define (domain puzzle)
  (:requirements :strips :equality :typing)
  (:types num loc)
  (:constants B - num) ; B represents the empty space
  (:predicates
    (at ?n - num ?l - loc) ; ?n is at location ?l
    (adjacent ?l1 ?l2 - loc) ; ?l1 and ?l2 are adjacent locations
  )

  (:action slide
    :parameters (?n - num ?from - loc ?to - loc)
    :precondition (and (at ?n ?from) (at B ?to) (adjacent ?from ?to))
    :effect (and
              (not (at ?n ?from))
              (at ?n ?to)
              (not (at B ?to))
              (at B ?from)
            )
  )
)
```

- PPT1.pddl

```
(define (problem PPT1)
  (:domain puzzle)
  (:objects
    num1 num2 num3 num4 num5 num6 num7 num8 num9 num10 num11 num12 num13 num14 num15 - num
    loc-1-1 loc-1-2 loc-1-3 loc-1-4
    loc-2-1 loc-2-2 loc-2-3 loc-2-4
    loc-3-1 loc-3-2 loc-3-3 loc-3-4
    loc-4-1 loc-4-2 loc-4-3 loc-4-4 - loc
  )
  (:init
    (at num14 loc-1-1) (at num10 loc-1-2) (at num6 loc-1-3) (at B loc-1-4)
    (at num4 loc-2-1) (at num9 loc-2-2) (at num1 loc-2-3) (at num8 loc-2-4)
    (at num2 loc-3-1) (at num3 loc-3-2) (at num5 loc-3-3) (at num11 loc-3-4)
    (at num12 loc-4-1) (at num13 loc-4-2) (at num7 loc-4-3) (at num15 loc-4-4)
    ; Add adjacency relations here
    (adjacent loc-1-1 loc-1-2) (adjacent loc-1-1 loc-2-1)
    (adjacent loc-1-2 loc-1-1) (adjacent loc-1-2 loc-1-3) (adjacent loc-1-2 loc-2-2)
    (adjacent loc-1-3 loc-1-2) (adjacent loc-1-3 loc-1-4) (adjacent loc-1-3 loc-2-3)
    (adjacent loc-1-4 loc-1-3) (adjacent loc-1-4 loc-2-4)
    (adjacent loc-2-1 loc-1-1) (adjacent loc-2-1 loc-2-2) (adjacent loc-2-1 loc-3-1)
    (adjacent loc-2-2 loc-1-2) (adjacent loc-2-2 loc-2-1) (adjacent loc-2-2 loc-2-3) (adjacent loc-2-2 loc-3-2)
    (adjacent loc-2-3 loc-1-3) (adjacent loc-2-3 loc-2-2) (adjacent loc-2-3 loc-2-4) (adjacent loc-2-3 loc-3-3)
    (adjacent loc-2-4 loc-1-4) (adjacent loc-2-4 loc-2-3) (adjacent loc-2-4 loc-3-4)
    (adjacent loc-3-1 loc-2-1) (adjacent loc-3-1 loc-3-2) (adjacent loc-3-1 loc-4-1)
    (adjacent loc-3-2 loc-2-2) (adjacent loc-3-2 loc-3-1) (adjacent loc-3-2 loc-3-3) (adjacent loc-3-2 loc-4-2)
    (adjacent loc-3-3 loc-2-3) (adjacent loc-3-3 loc-3-2) (adjacent loc-3-3 loc-3-4) (adjacent loc-3-3 loc-4-3)
    (adjacent loc-3-4 loc-2-4) (adjacent loc-3-4 loc-3-3) (adjacent loc-3-4 loc-4-4)
    (adjacent loc-4-1 loc-3-1) (adjacent loc-4-1 loc-4-2)
    (adjacent loc-4-2 loc-3-2) (adjacent loc-4-2 loc-4-1) (adjacent loc-4-2 loc-4-3)
    (adjacent loc-4-3 loc-3-3) (adjacent loc-4-3 loc-4-2) (adjacent loc-4-3 loc-4-4)
    (adjacent loc-4-4 loc-3-4) (adjacent loc-4-4 loc-4-3)
  )
  (:goal (and
    (at num1 loc-1-1) (at num2 loc-1-2) (at num3 loc-1-3) (at num4 loc-1-4)
    (at num5 loc-2-1) (at num6 loc-2-2) (at num7 loc-2-3) (at num8 loc-2-4)
    (at num9 loc-3-1) (at num10 loc-3-2) (at num11 loc-3-3) (at num12 loc-3-4)
    (at num13 loc-4-1) (at num14 loc-4-2) (at num15 loc-4-3) (at B loc-4-4)
  ))
)
; 14 10 6 0
; 4 9 1 8
; 2 3 5 11
; 12 13 7 15
```

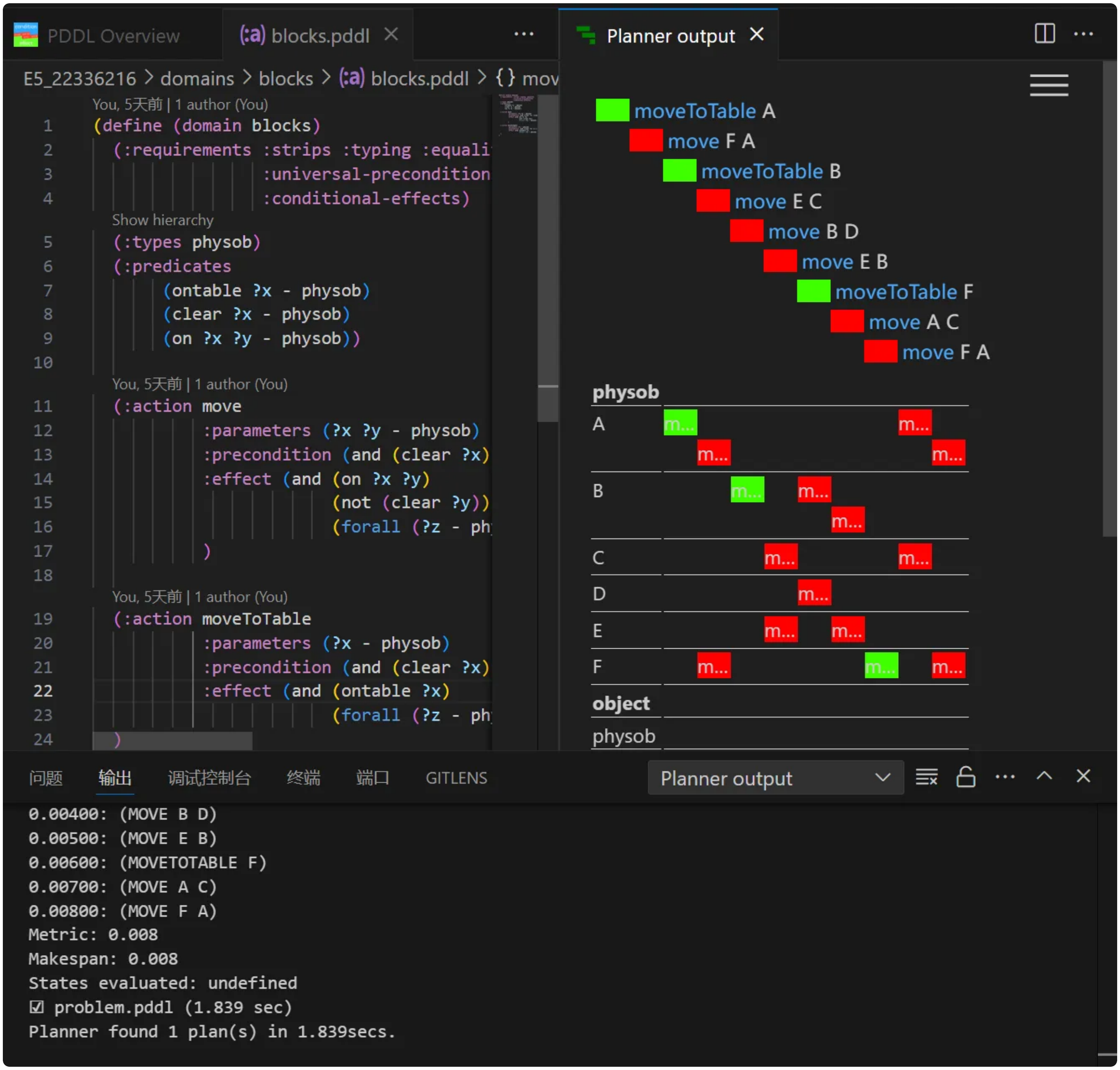
四、创新点&优化

无

3 实验结果及分析

1. 实验结果展示示例（可图可表可文字，尽量可视化）

由于输出行数过多，在实验报告中只展示截图，具体数据请见/result



You, 5天前 | 1 author (You)

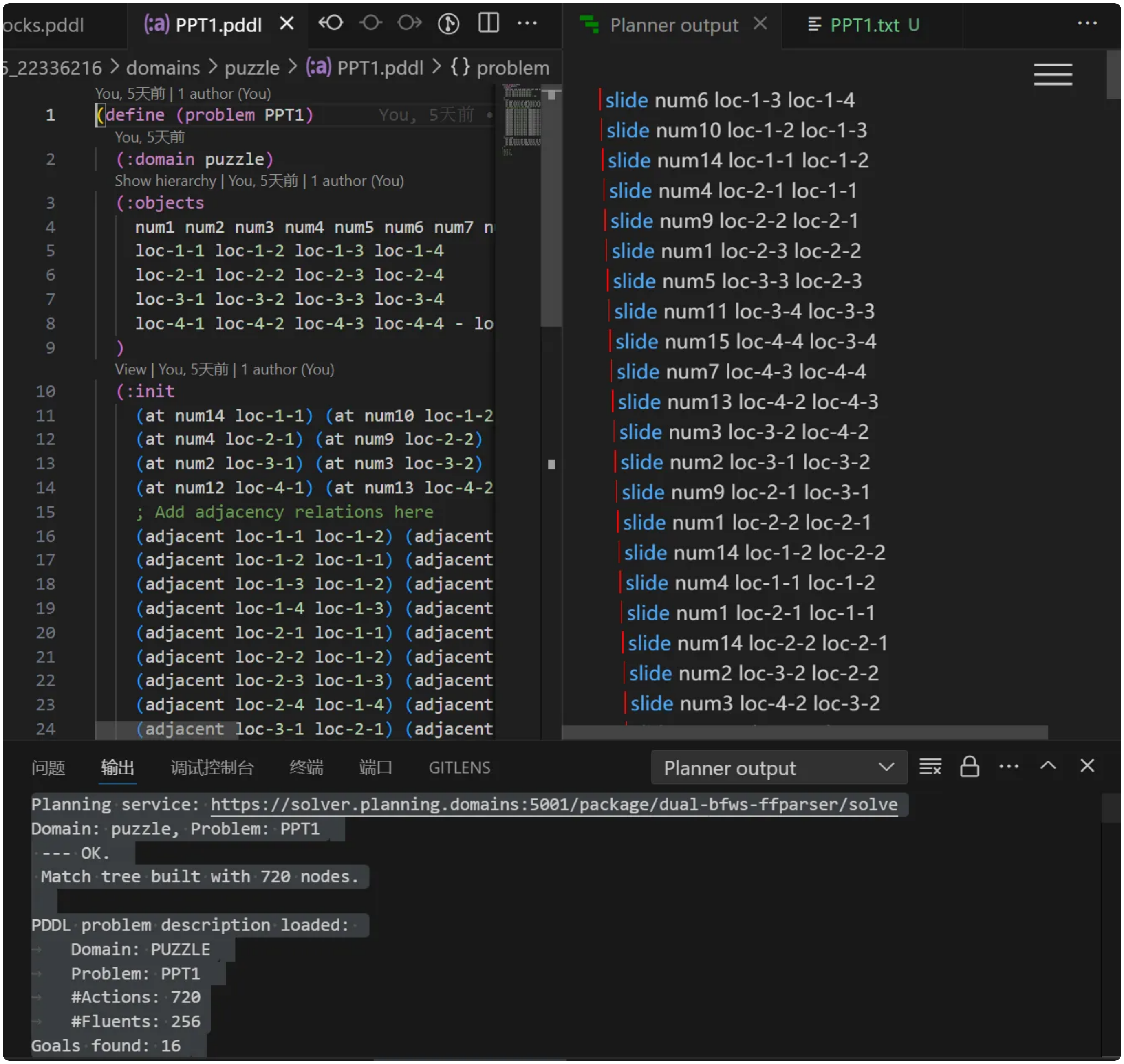
```
(define (domain blocks)
  (:requirements :strips :typing :equali
                 :universal-precondition
                 :conditional-effects)
  Show hierarchy
  (:types physob)
  (:predicates
     (ontable ?x - physob)
     (clear ?x - physob)
     (on ?x ?y - physob))

  You, 5天前 | 1 author (You)
  (:action move
      :parameters (?x ?y - physob)
      :precondition (and (clear ?x)
      :effect (and (on ?x ?y)
                   (not (clear ?y))
                   (forall (?z - ph
      )

  You, 5天前 | 1 author (You)
  (:action moveToTable
      :parameters (?x - physob)
      :precondition (and (clear ?x)
      :effect (and (ontable ?x)
                   (forall (?z - ph
  )
```

🟩 moveToTable A
🟥 move F A
🟩 moveToTable B
🟥 move E C
🟥 move B D
🟥 move E B
🟩 moveToTable F
🟥 move A C
🟥 move F A

```
0.00400: (MOVE B D)
0.00500: (MOVE E B)
0.00600: (MOVETOTABLE F)
0.00700: (MOVE A C)
0.00800: (MOVE F A)
Metric: 0.008
Makespan: 0.008
States evaluated: undefined
☑ problem.pddl (1.839 sec)
Planner found 1 plan(s) in 1.839secs.
```


ocks.pddl
PPT1.pddl
Planner output
PPT1.txt U
5_22336216 > domains > puzzle > PPT1.pddl > { } problem
You, 5天前 | 1 author (You)
You, 5天前
Show hierarchy | You, 5天前 | 1 author (You)
View | You, 5天前 | 1 author (You)
问题
输出
调试控制台
终端
端口
GITLENS
Planner output

```
(define (problem PPT1)
  (:domain puzzle)
  (:objects
    num1 num2 num3 num4 num5 num6 num7 n
    loc-1-1 loc-1-2 loc-1-3 loc-1-4
    loc-2-1 loc-2-2 loc-2-3 loc-2-4
    loc-3-1 loc-3-2 loc-3-3 loc-3-4
    loc-4-1 loc-4-2 loc-4-3 loc-4-4 - lo
  )
  (:init
    (at num14 loc-1-1) (at num10 loc-1-2
    (at num4 loc-2-1) (at num9 loc-2-2)
    (at num2 loc-3-1) (at num3 loc-3-2)
    (at num12 loc-4-1) (at num13 loc-4-2
    ; Add adjacency relations here
    (adjacent loc-1-1 loc-1-2) (adjacent
    (adjacent loc-1-2 loc-1-1) (adjacent
    (adjacent loc-1-3 loc-1-2) (adjacent
    (adjacent loc-1-4 loc-1-3) (adjacent
    (adjacent loc-2-1 loc-1-1) (adjacent
    (adjacent loc-2-2 loc-1-2) (adjacent
    (adjacent loc-2-3 loc-1-3) (adjacent
    (adjacent loc-2-4 loc-1-4) (adjacent
    (adjacent loc-3-1 loc-2-1) (adjacent
```

```
slide num6 loc-1-3 loc-1-4
slide num10 loc-1-2 loc-1-3
slide num14 loc-1-1 loc-1-2
slide num4 loc-2-1 loc-1-1
slide num9 loc-2-2 loc-2-1
slide num1 loc-2-3 loc-2-2
slide num5 loc-3-3 loc-2-3
slide num11 loc-3-4 loc-3-3
slide num15 loc-4-4 loc-3-4
slide num7 loc-4-3 loc-4-4
slide num13 loc-4-2 loc-4-3
slide num3 loc-3-2 loc-4-2
slide num2 loc-3-1 loc-3-2
slide num9 loc-2-1 loc-3-1
slide num1 loc-2-2 loc-2-1
slide num14 loc-1-2 loc-2-2
slide num4 loc-1-1 loc-1-2
slide num1 loc-2-1 loc-1-1
slide num14 loc-2-2 loc-2-1
slide num2 loc-3-2 loc-2-2
slide num3 loc-4-2 loc-3-2
```

```
Planning service: https://solver.planning.domains:5001/package/dual-bfws-ffparser/solve
Domain: puzzle, Problem: PPT1
 --- OK.
 Match tree built with 720 nodes.

PDDL problem description loaded:
    Domain: PUZZLE
    Problem: PPT1
    #Actions: 720
    #Fluents: 256
Goals found: 16
```

2. 评测指标展示及分析

block.pddl运行1.839s

PPT1.pddl运行5.781s

4 思考题

无

5 参考资料

无