E06 FF Planner

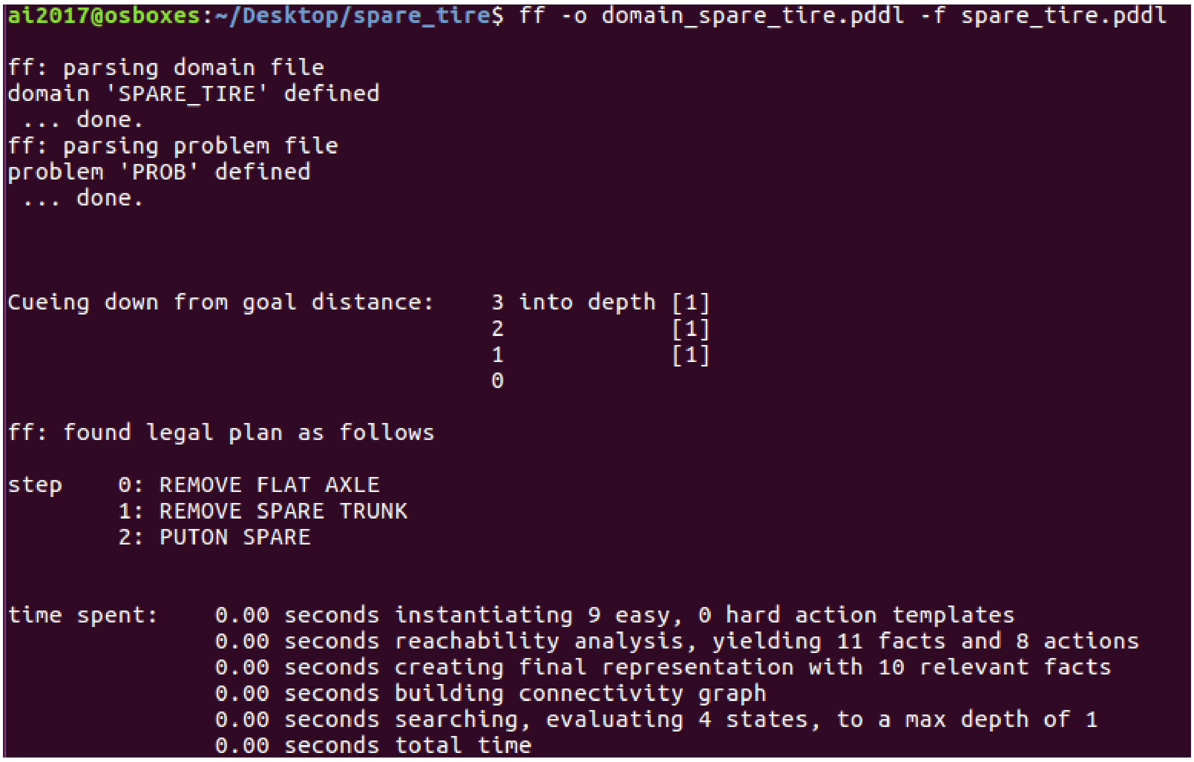
姓名	学号	日期
张天祎	18340215	2020.10.9

1 Examples

1.1 Spare Tire

```
(define (domain spare_tire)
  (:requirements :strips :equality:typing)
  (:types physob location)
  (:predicates  (Tire ?x - physob)
                (at ?x - physob ?y - location))
  (:action Remove
             :parameters (?x - physob ?y - location)
             :precondition (At ?x ?y)
             :effect (and (not (At ?x ?y)) (At ?x Ground)))

  (:action PutOn
             :parameters (?x - physob)
             :precondition (and (Tire ?x) (At ?x Ground) (not (At Flat Axle)))
             :effect (and (not (At ?x Ground)) (At ?x Axle)))
  (:action LeaveOvernight
             :effect (and (not (At Spare Ground))
                          (not (At Spare Axle))
                          (not (At Spare Trunk))
                          (not (At Flat Ground))
                          (not (At Flat Axle))
                          (not (At Flat Trunk)) ))
 )
```

```
(define (problem prob)
 (:domain spare_tire)
 (:objects Flat Spare -physob Axle Trunk Ground - location)

 (:init
  (Tire Flat)
  (Tire Spare)
  (At Flat Axle)
  (At Spare Trunk)

)
 (:goal
  (At Spare Axle)
 ))
```



```
ai2017@osboxes:~/Desktop/spare_tire$ ff -o domain_spare_tire.pddl -f spare_tire.pddl

ff: parsing domain file
domain 'SPARE_TIRE' defined
 ... done.
ff: parsing problem file
problem 'PROB' defined
 ... done.



Cueing down from goal distance:    3 into depth [1]
                                   2            [1]
                                   1            [1]
                                   0

ff: found legal plan as follows

step    0: REMOVE FLAT AXLE
        1: REMOVE SPARE TRUNK
        2: PUTON SPARE


time spent:    0.00 seconds instantiating 9 easy, 0 hard action templates
               0.00 seconds reachability analysis, yielding 11 facts and 8 actions
               0.00 seconds creating final representation with 10 relevant facts
               0.00 seconds building connectivity graph
               0.00 seconds searching, evaluating 4 states, to a max depth of 1
               0.00 seconds total time
```

1.2 Briefcase World

Please refer to pddl.pdf at page 2. Please pay More attention to the usages of forall and when. For more examples, please refer to ff-domains.tgz and benchmarksV1.1.zip. For more usages of FF planner, please refer to the documentation pddl.pdf.

2 Tasks

2.1 8-puzzle

Please complete domain puzzle.pddl and puzzle.pddl to solve the 8-puzzle problem.

1	2	3
7	8	
6	4	5

2.2 Blocks World

Planning in the blocks world is a traditional planning exercise, and you can recall what we have introduced in the theory course. There are a collection of blocks: a block can be on the table, or on the top of another block. There are three predicates:

- clear(x): there is no block on top of block x;
- on(x,y): block x is on the top of block y;

- onTable(x): block x is on the table

  There are two actions in this task:

- move(x,y): move block x onto block y, provided that both x and y are clear;
- moveToTable(x): move block x on to the table, provided that x is clear and x is not on the table;

  Give initial state and goal state, find the actions change the initial state to the goal state. In this task, please complete the file domain blocks.pddl to solve the blocks world problem. You should know the usages of forall and when.

```
(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 ()
             :effect ()
             )

  (:action moveToTable
             :parameters (?x - physob)
             :precondition ()
             :effect ( )
 )
```

```
(define (problem prob)
 (:domain blocks)
 (:objects A B C D E F - physob)
 (:init (clear A)(on A B)(on B C)(ontable C) (ontable D)
  (ontable F)(on E D)(clear E)(clear F)
)
 (:goal  (and (clear F) (on F A) (on A C) (ontable C)(clear E) (on E B)
         (on B D) (ontable D)) )
 )
```

Please submit a file named E06 YourNumber.pdf, and send it to ai_2020@foxmail.com

3 Codes and Results

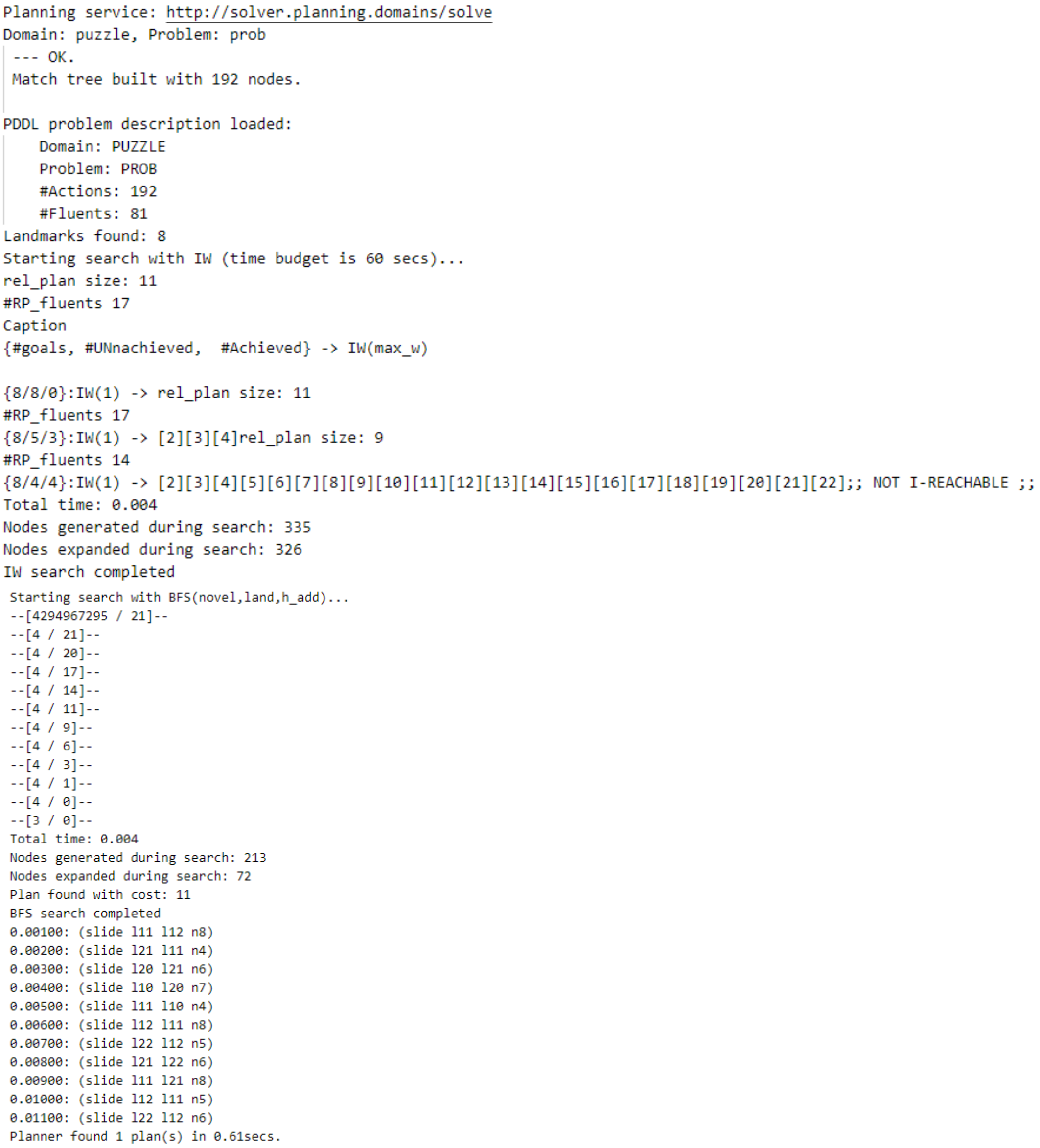
8-puzzle

```
(define (domain puzzle)
(:requirements :strips :typing :equality)
(:types num,local)
(:predicates (empty ?x - local)
            (at ?x - num ?y - local)
            (nexto ?x - local ?y - local)
)

(:action slide
            :parameters (?z - num ?x - local ?y - local )
            :precondition (and (empty ?y) (at ?z ?x) (nexto ?x ?y)
            )
            :effect (and (at ?z ?y) (empty ?x) (not (at ?z ?x)) (not (empty ?y))
            )
)

)
```

```
(define (problem prob) (:domain puzzle)
(:objects n1 n2 n3 n4 n5 n6 n7 n8 -num l00 l01 l02 l10 l11 l12 l20 l21 l22 -
local
)

(:init
    (at n1 l00)
    (at n2 l01)
    (at n3 l02)
    (at n7 l10)
    (at n8 l11)
    (empty l12)
    (at n6 l20)
    (at n4 l21)
    (at n5 l22)
    (nexto l00 l01)
    (nexto l00 l10)
    (nexto l01 l00)
    (nexto l01 l02)
    (nexto l01 l11)
    (nexto l02 l01)
    (nexto l02 l12)
    (nexto l10 l11)
    (nexto l10 l00)
    (nexto l10 l20)
    (nexto l11 l10)
    (nexto l11 l12)
    (nexto l11 l01)
    (nexto l11 l21)
    (nexto l12 l11)
    (nexto l12 l02)
    (nexto l12 l22)
    (nexto l20 l21)
    (nexto l20 l10)
    (nexto l21 l20)
    (nexto l21 l22)
    (nexto l21 l11)
    (nexto l22 l21)
    (nexto l22 l12)
)

(:goal (and
    (at n1 l00)
```

```
      (at n2 l01)
      (at n3 l02)
      (at n4 l10)
      (at n5 l11)
      (at n6 l12)
      (at n7 l20)
      (at n8 l21)
  ))

  )
```

在VScode的PDLL插件中运行（实际上也是发送数据到在线运行网站）结果如下，找到了一条11步的路径：

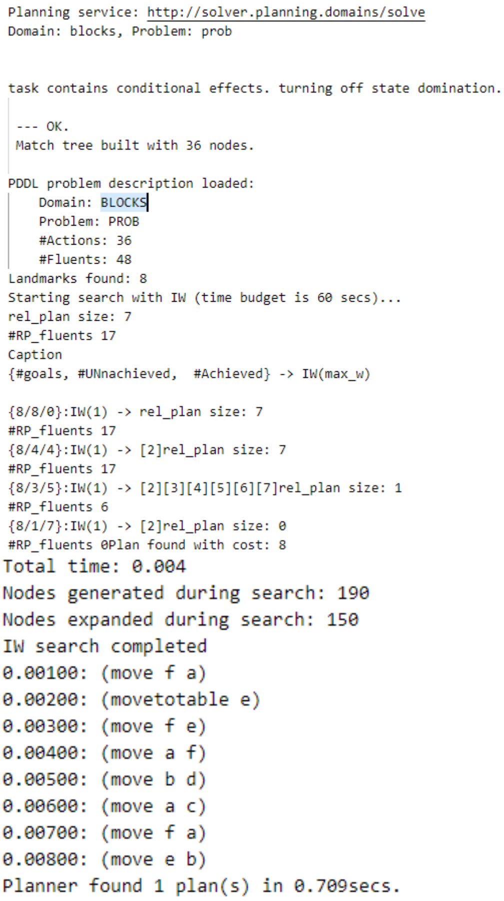


```
Planning service: http://solver.planning.domains/solve
Domain: puzzle, Problem: prob
 --- OK.
 Match tree built with 192 nodes.

PDDL problem description loaded:
    Domain: PUZZLE
    Problem: PROB
    #Actions: 192
    #Fluents: 81
Landmarks found: 8
Starting search with IW (time budget is 60 secs)...
rel_plan size: 11
#RP_fluents 17
Caption
{#goals, #UNnachieved,  #Achieved} -> IW(max_w)

{8/8/0}:IW(1) -> rel_plan size: 11
#RP_fluents 17
{8/5/3}:IW(1) -> [2][3][4]rel_plan size: 9
#RP_fluents 14
{8/4/4}:IW(1) -> [2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22];; NOT I-REACHABLE ;;
Total time: 0.004
Nodes generated during search: 335
Nodes expanded during search: 326
IW search completed
 Starting search with BFS(novel,land,h_add)...
 --[4294967295 / 21]--
 --[4 / 21]--
 --[4 / 20]--
 --[4 / 17]--
 --[4 / 14]--
 --[4 / 11]--
 --[4 / 9]--
 --[4 / 6]--
 --[4 / 3]--
 --[4 / 1]--
 --[4 / 0]--
 --[3 / 0]--
 Total time: 0.004
 Nodes generated during search: 213
 Nodes expanded during search: 72
 Plan found with cost: 11
 BFS search completed
 0.00100: (slide l11 l12 n8)
 0.00200: (slide l21 l11 n4)
 0.00300: (slide l20 l21 n6)
 0.00400: (slide l10 l20 n7)
 0.00500: (slide l11 l10 n4)
 0.00600: (slide l12 l11 n8)
 0.00700: (slide l22 l12 n5)
 0.00800: (slide l21 l22 n6)
 0.00900: (slide l11 l21 n8)
 0.01000: (slide l12 l11 n5)
 0.01100: (slide l22 l12 n6)
 Planner found 1 plan(s) in 0.61secs.
```

有一个有趣的现象，一开始我在goal处多加了一个冗余的条件(empty l22)，这样运行的结果是65步的一条路径，显然比11步的路径差多了。猜测是goal的不同会影响FF规划器的启发式函数的效果，导致出现很差的结果。

Blocks World

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



```
Planning service: http://solver.planning.domains/solve
Domain: blocks, Problem: prob


task contains conditional effects. turning off state domination.

 --- OK.
 Match tree built with 36 nodes.

PDDL problem description loaded:
    Domain: BLOCKS
    Problem: PROB
    #Actions: 36
    #Fluents: 48
Landmarks found: 8
Starting search with IW (time budget is 60 secs)...
rel_plan size: 7
#RP_fluents 17
Caption
{#goals, #UNnachieved,  #Achieved} -> IW(max_w)

{8/8/0}:IW(1) -> rel_plan size: 7
#RP_fluents 17
{8/4/4}:IW(1) -> [2]rel_plan size: 7
#RP_fluents 17
{8/3/5}:IW(1) -> [2][3][4][5][6][7]rel_plan size: 1
#RP_fluents 6
{8/1/7}:IW(1) -> [2]rel_plan size: 0
#RP_fluents 0Plan found with cost: 8
Total time: 0.004
Nodes generated during search: 190
Nodes expanded during search: 150
IW search completed
0.00100: (move f a)
0.00200: (movetotable e)
0.00300: (move f e)
0.00400: (move a f)
0.00500: (move b d)
0.00600: (move a c)
0.00700: (move f a)
0.00800: (move e b)
Planner found 1 plan(s) in 0.709secs.
```

仅比上一问多使用一些forall&when，其实比上一问还简单，成功找到了一条8步的路径。