School of Data and Computer Science, Sun Yat-sen University

E07 FF Planner

16337188 Ouyang Runyu

October 19, 2018

Contents

1 Examples

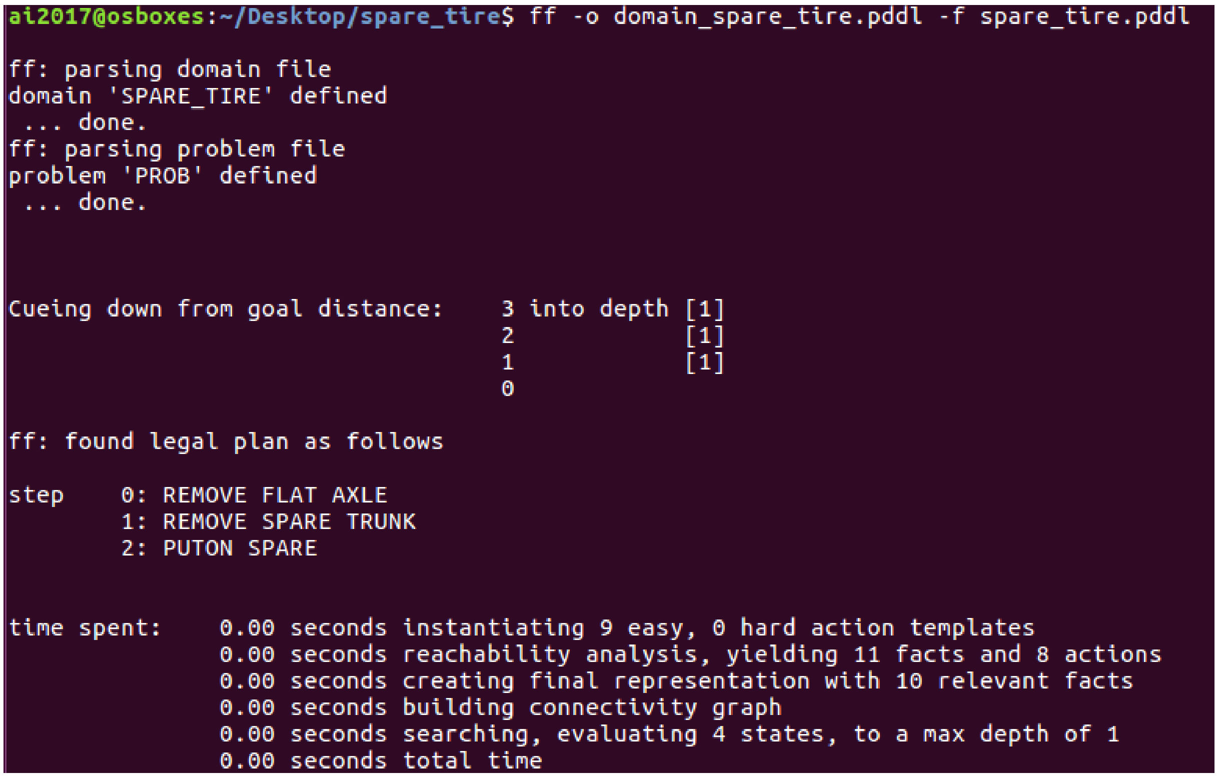
1.1 Spare Tire

domain_spare_tire.pddl

```
(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)) ))
 )
```

spare_tire.pddl

```
(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
```

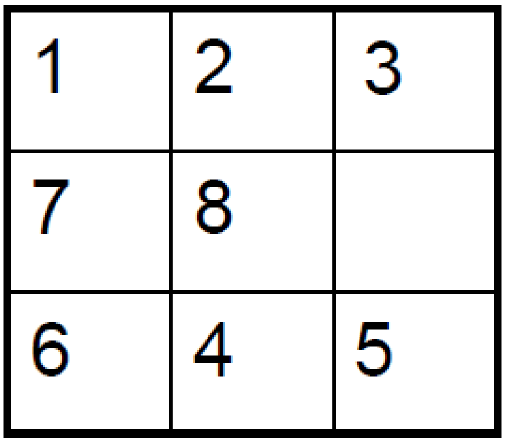
1.2 Briefcase World

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



1	2	3
7	8	
6	4	5

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

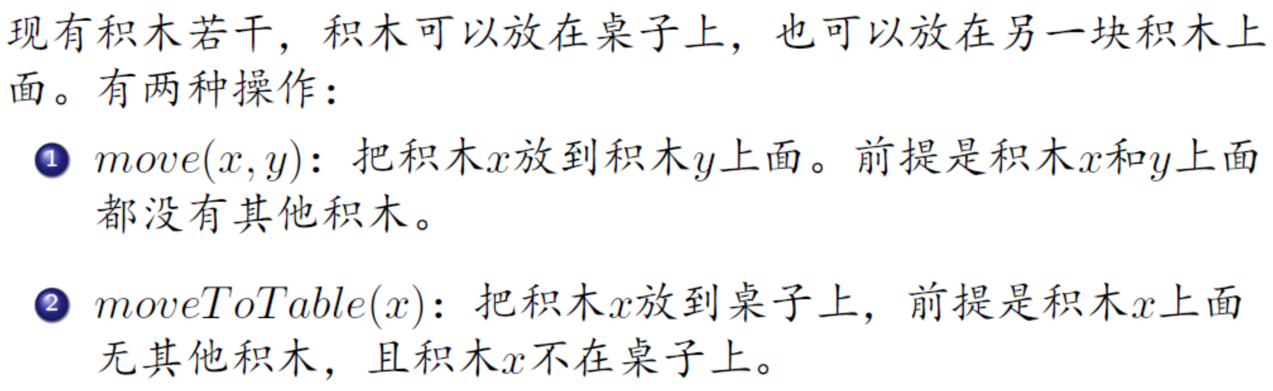
domain_puzzle.pddl

```
(define (domain puzzle)
  (:requirements :strips :equality:typing)
  (:types num loc)
  (:predicates  ())

(:action slide
             :parameters ()
             :precondition ()
             :effect ()
 )
)
```

domain_puzzle.pddl

```
(define (problem prob)
 (:domain puzzle)
 (:objects )
 (:init )
 (:goal ())
)
```

2.2 Blocks World



现有积木若干，积木可以放在桌子上，也可以放在另一块积木上面。有两种操作：

1. $move(x, y)$：把积木x放到积木y上面。前提是积木x和y上面都没有其他积木。
2. $moveToTable(x)$：把积木x放到桌子上，前提是积木x上面无其他积木，且积木x不在桌子上。

Please complete the file `domain_blocks.pddl` to solve the blocks world problem. You should know the usages of `forall` and `when`.

domain_blocks.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 ()
             :effect ()
             )

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

blocks.pddl

```
(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 E07_YourNumber.pdf, and send it to ai_2018@foxmail.com

3 Codes and Results

3.1 8-puzzle

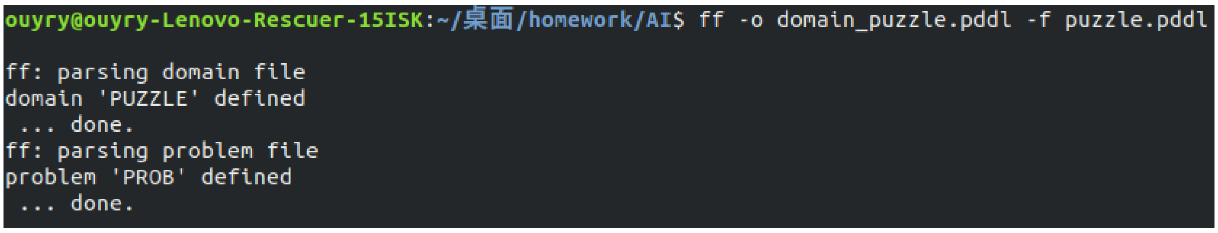
- You can see the code in domain_puzzle.pddl and puzzle.pddl

domain_puzzle.pddl

```
 (define (domain puzzle)
   (:requirements :strips :equality :typing)
   (:types num loc)
   (:predicates (At ?X - num ?Y - loc)
                (Next ?X - loc ?Y - loc))

   (:action slide
        :parameters (?ZeroLoc - loc ?NextNum - num ?NextLoc - loc)
        :precondition (and (At n0 ?ZeroLoc) (At ?NextNum ?NextLoc)
                           (Next ?ZeroLoc ?NextLoc)
                      )
        :effect (and (At n0 ?NextLoc) (not (At n0 ?ZeroLoc))
                     (At ?NextNum ?ZeroLoc) (not (At ?NextNum ?NextLoc)
                )
        )
   )
)
```

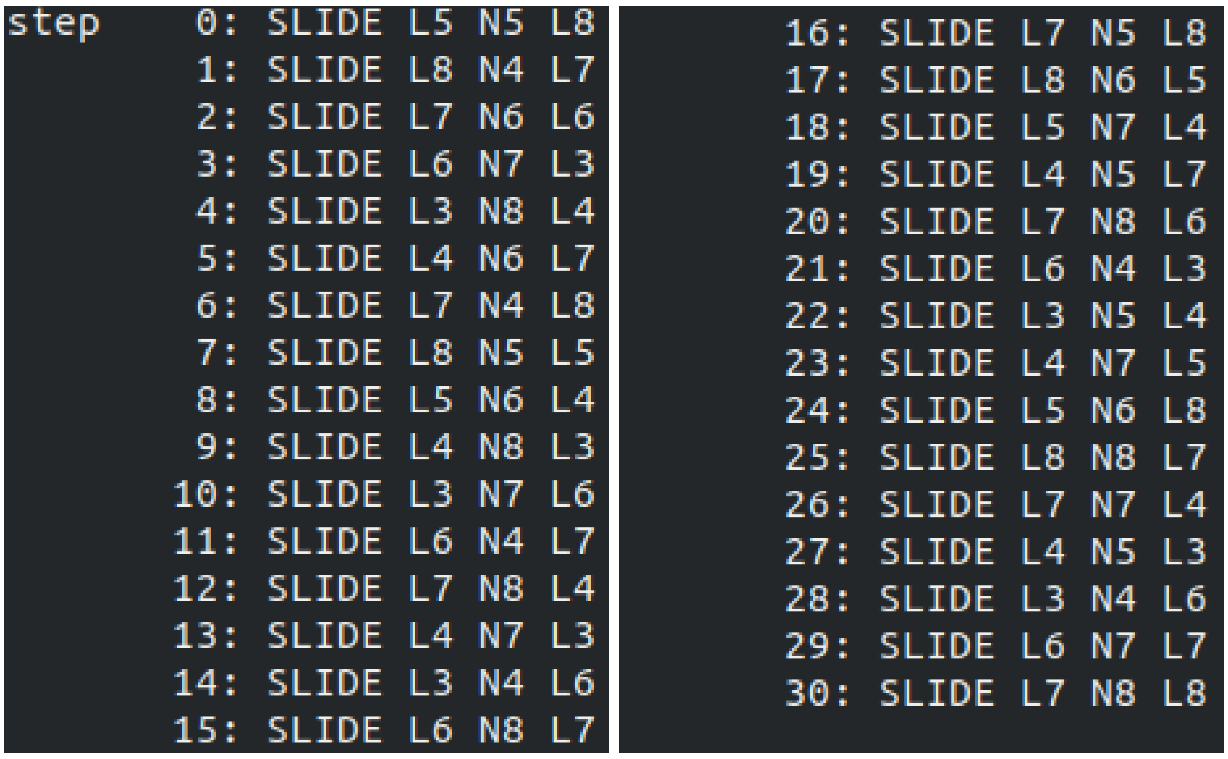
puzzle.pddl

```
  (define (problem prob)
    (:domain puzzle)

    (:objects n0 n1 n2 n3 n4 n5 n6 n7 n8 –num
              l0 l1 l2 l3 l4 l5 l6 l7 l8 −loc)

    (:init (At n1 l0) (At n2 l1) (At n3 l2) (At n7 l3)
           (At n8 l4) (At n0 l5) (At n6 l6) (At n4 l7)
           (At n5 l8) (Next l0, l1) (Next l1 l0) (Next l0 l3)
           (Next l3 l0) (Next l1 l2) (Next l2 l1) (Next l1 l4)
           (Next l4 l1) (Next l2 l5) (Next l5 l2) (Next l3 l4)
           (Next l4 l3) (Next l3 l6) (Next l6 l3) (Next l4 l5)
           (Next l5 l4) (Next l4 l7) (Next l7 l4) (Next l5 l8)
           (Next l8 l5) (Next l6 l7) (Next l7 l6) (Next l7 l8)
           (Next l8 l7)
    )

    (:goal (and (At n1 l0) (At n2 l1) (At n3 l2) (At n4 l3)
                (At n5 l4) (At n6 l5) (At n7 l6) (At n8 l7)
                (At n0 l8)
           )
    )
)
```



```
ouyry@ouyry-Lenovo-Rescuer-15ISK:~/桌面/homework/AI$ ff -o domain_puzzle.pddl -f puzzle.pddl

ff: parsing domain file
domain 'PUZZLE' defined
 ... done.
ff: parsing problem file
problem 'PROB' defined
 ... done.
```

Figure 1: The result of 8-puzzle.



```
step     0: SLIDE L5 N5 L8
         1: SLIDE L8 N4 L7
         2: SLIDE L7 N6 L6
         3: SLIDE L6 N7 L3
         4: SLIDE L3 N8 L4
         5: SLIDE L4 N6 L7
         6: SLIDE L7 N4 L8
         7: SLIDE L8 N5 L5
         8: SLIDE L5 N6 L4
         9: SLIDE L4 N8 L3
        10: SLIDE L3 N7 L6
        11: SLIDE L6 N4 L7
        12: SLIDE L7 N8 L4
        13: SLIDE L4 N7 L3
        14: SLIDE L3 N4 L6
        15: SLIDE L6 N8 L7
```

```
        16: SLIDE L7 N5 L8
        17: SLIDE L8 N6 L5
        18: SLIDE L5 N7 L4
        19: SLIDE L4 N5 L7
        20: SLIDE L7 N8 L6
        21: SLIDE L6 N4 L3
        22: SLIDE L3 N5 L4
        23: SLIDE L4 N7 L5
        24: SLIDE L5 N6 L8
        25: SLIDE L8 N8 L7
        26: SLIDE L7 N7 L4
        27: SLIDE L4 N5 L3
        28: SLIDE L3 N4 L6
        29: SLIDE L6 N7 L7
        30: SLIDE L7 N8 L8
```

Figure 2: The result of 8-puzzle.

3.2 Blocks World

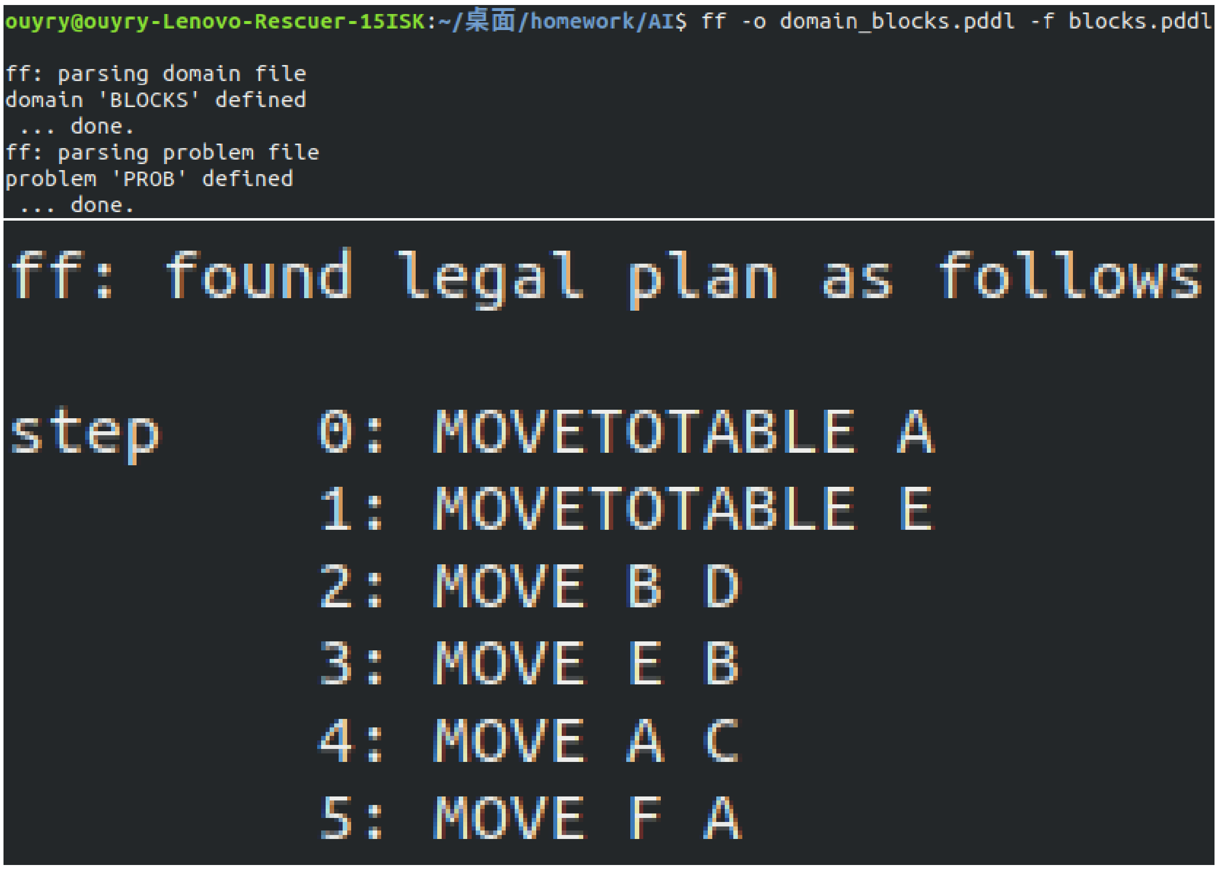
- You can see the code in domain_blocks.pddl and blocks.pddl

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

blocks.pddl

```
(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)
      )
  )
)
```



```
ouyry@ouyry-Lenovo-Rescuer-15ISK:~/桌面/homework/AI$ ff -o domain_blocks.pddl -f blocks.pddl

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

ff: found legal plan as follows

step    0: MOVETOTABLE A
        1: MOVETOTABLE E
        2: MOVE B D
        3: MOVE E B
        4: MOVE A C
        5: MOVE F A
```

Figure 3: The result of Blocks World.