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

E07 FF Planner

17341175 徐志成

2019 年 10 月 18 日

目录

1 Examples **2**

1.1 Spare Tire 2

1.2 Briefcase World 3

2 Tasks **3**

2.1 8-puzzle 3

2.2 Blocks World 4

3 Codes and Results **6**

3.1 8-puzzle 6

3.2 Blocks World 7

1 Examples

1.1 Spare Tire

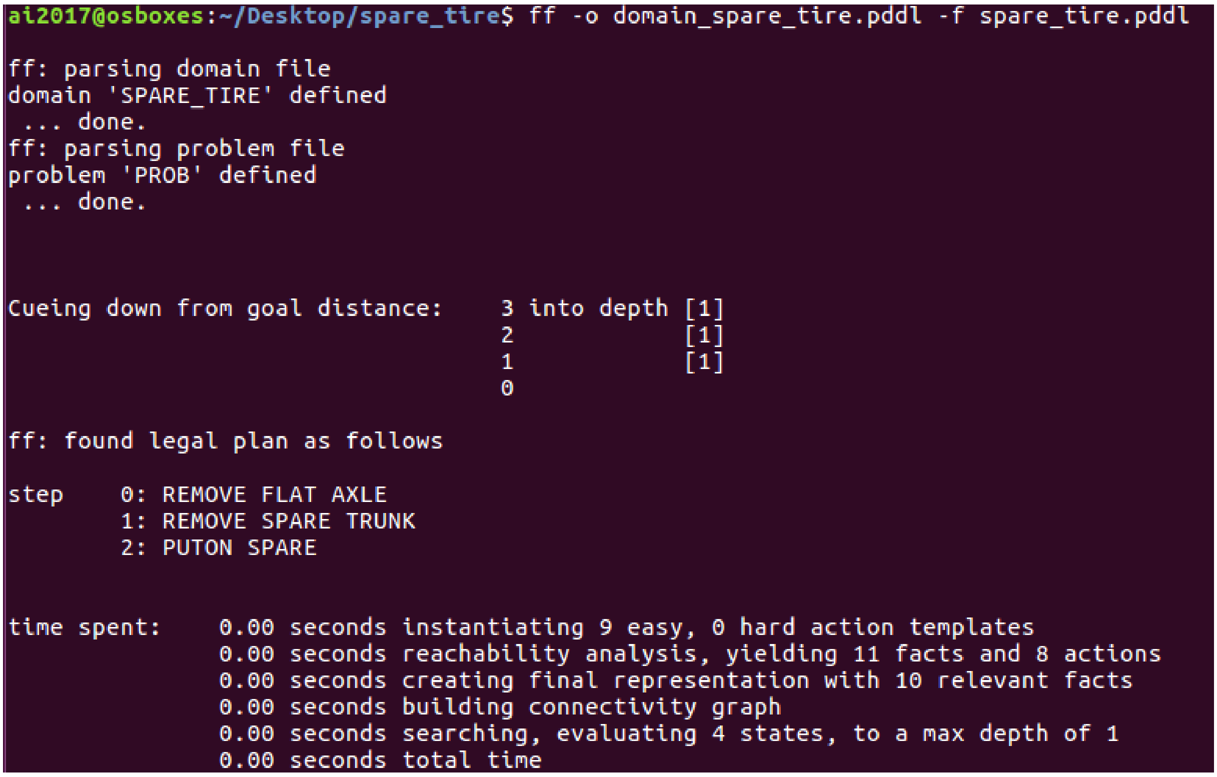
domain_spare_tire.pddl

```
(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

domain_puzzle.pddl

```
(define (domain puzzle)
```

1	2	3
7	8	
6	4	5

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

```
  (: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

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

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。

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_201901@foxmail.com

3 Codes and Results

3.1 8-puzzle

domain_puzzle.pddl

```
(define (domain puzzle)
(:requirements :strips :equality :typing
             )
(:predicates (at ?x - num ?y - loc)
                 (adj ?x - loc ?y - loc))

(:action slide
     :parameters (?x - num ?y - loc ?z - loc)
     :precondition(and (at ?x ?y) (or (adj ?z ?y) (adj ?y ?z )) (at
          T0 ?z))
     :effect(and (at ?x ?z) (at T0 ?y) (not (at ?x ?y)) (not (at T0
          ?z)))
 )
)
```

domain_puzzle.pddl

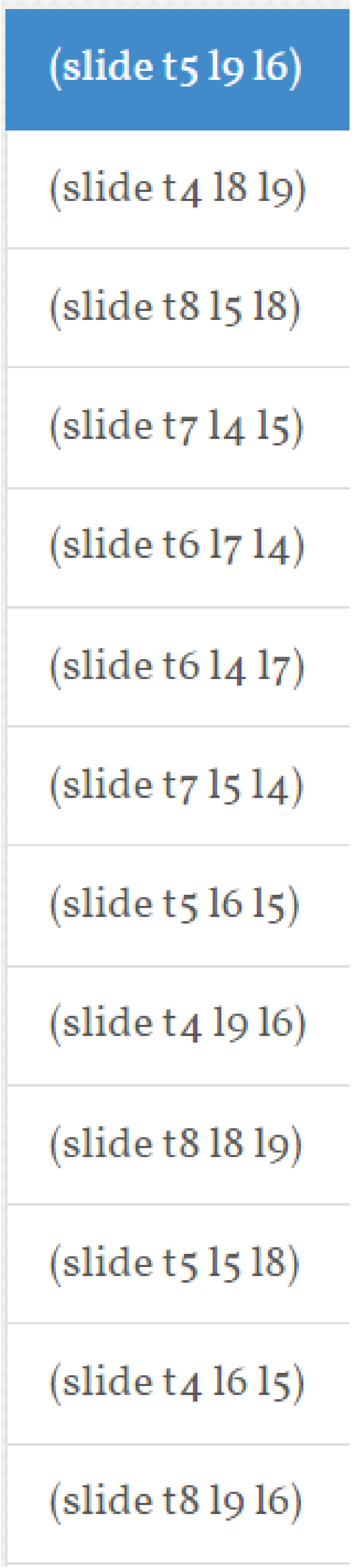
```
(define (problem prob)
(:domain puzzle)
(:objects T0 T1 T2 T3 T4 T5 T6 T7 T8 -num L1 L2 L3 L4 L5 L6 L7 L8 L9 -loc)
(:init (at T1 L1) (at T2 L2)(at T3 L3) (at T7 L4) (at T8 L5)(at T0 L6) (at T6 L7)(at T4 L8) (at T5 L9)
  (adj L5 L2)(adj L5 L4) (adj L5 L6) (adj L5 L8) (adj L1 L2) (adj L2 L3)
(adj L1 L4) (adj L3 L6) (adj L7 L8) (adj L8 L9) (adj L4 L7)(adj L6 L9))
(:goal (and(at T1 L1) (at T2 L2) (at T3 L3) (at T4 L4) (at T5 L5) (at T6 L6) (at T7 L7) (at T8 L8) (at T0 L9)))
)
```

经过验证，图一图二的动作集合可以使8数码问题达到目标状态

3.2 Blocks World

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 (on ?x ?y) (not (clear ?y)) (not (ontable ?x))
                         (forall (?z -physob)
```



(slide t5 l9 l6)

(slide t4 l8 l9)

(slide t8 l5 l8)

(slide t7 l4 l5)

(slide t6 l7 l4)

(slide t6 l4 l7)

(slide t7 l5 l4)

(slide t5 l6 l5)

(slide t4 l9 l6)

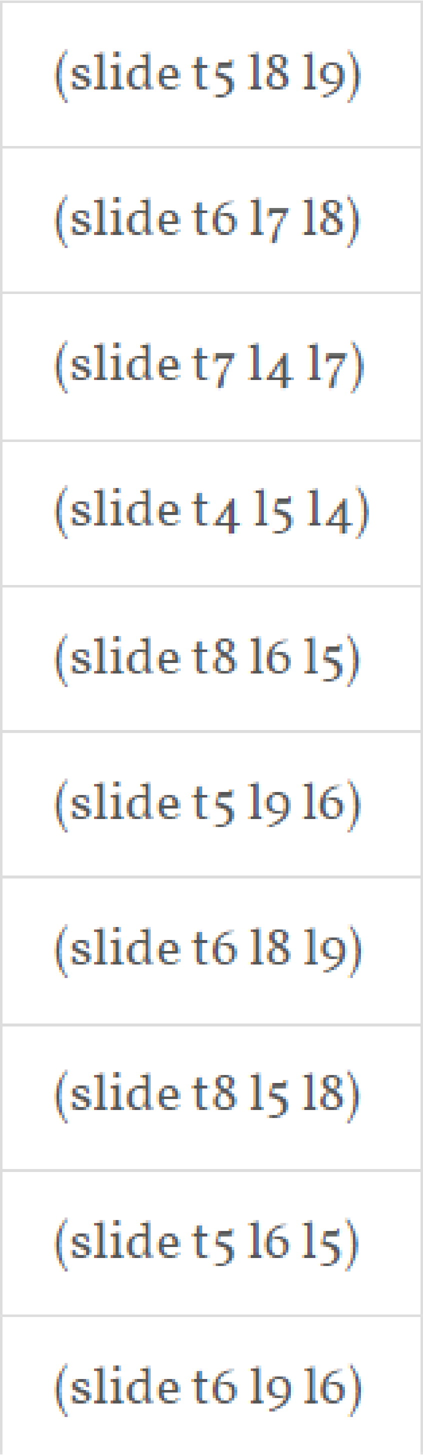
(slide t8 l8 l9)

(slide t5 l5 l8)

(slide t4 l6 l5)

(slide t8 l9 l6)

图 1: actions1



(slide t5 l8 l9)

(slide t6 l7 l8)

(slide t7 l4 l7)

(slide t4 l5 l4)

(slide t8 l6 l5)

(slide t5 l9 l6)

(slide t6 l8 l9)

(slide t8 l5 l8)

(slide t5 l6 l5)

(slide t6 l9 l6)

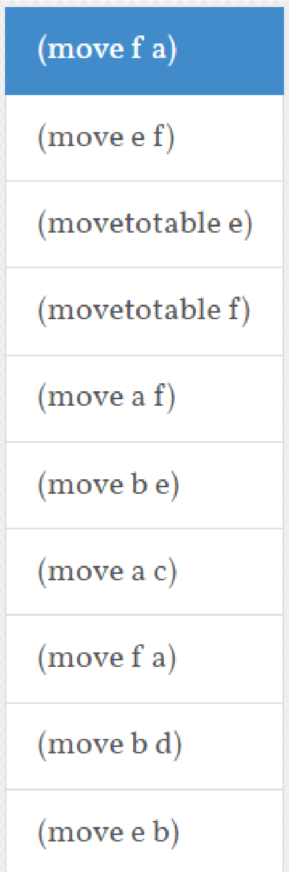
图 2: actions2

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

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)) )
)
```

经过验证，图三的动作集合可以使积木世界问题达到目标状态



```
(move f a)
(move e f)
(movetotable e)
(movetotable f)
(move a f)
(move b e)
(move a c)
(move f a)
(move b d)
(move e b)
```

图 3: blocks world results