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

E07 FF Planner

17341111 Xuehai Liu

October 20, 2019

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))
)
```

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)
  (: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`.

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)
```

```
8           (on B D) (ontable D)) )
9  )
```

Please submit a file named E07_YourNumber.pdf, and send it to ai_201901@foxmail.com

3 Codes

puzzleDomain.pddl

```
1  (define (domain puzzle)
2      (:requirements :strips :equality:typing)
3      (:types num loc)
4      (:predicates (adjacent ?x - loc ?y - loc)
5                   (at ?x - num  ?y - loc)
6                   (blank ?x - loc ))
7
8
9  (:action slide
10     :parameters(?T - num ?X - loc ?Y -loc )
11     :precondition(and (blank ?Y ) (at ?T ?X) (adjacent ?X ?Y ) )
12     :effect(and (not (at ?T ?X )) (at ?T ?Y) (blank ?X)  )
13     )
14
15 )
```

puzzleProb.pddl

```
1  (define (problem prob)
2      (:domain puzzle)
3      (:objects  n1 n2 n3 n4 n5 n6 n7 n8 -num
4      l1 l2 l3 l4 l5 l6 l7 l8 l9 - loc )
5      (:init
6      (blank l6)
7      (at n1 l1)
```

```
8       (at n2 l2)
9       (at n3 l3)
10      (at n7 l4)
11      (at n8 l5)
12      (at n6 l7)
13      (at n4 l8)
14      (at n5 l9)
15      (adjacent l1 l2)
16      (adjacent l1 l4)
17      (adjacent l2 l1)
18      (adjacent l2 l3)
19      (adjacent l2 l5)
20      (adjacent l3 l2)
21      (adjacent l3 l6)
22      (adjacent l4 l1)
23      (adjacent l4 l5)
24      (adjacent l4 l7)
25      (adjacent l5 l2)
26      (adjacent l5 l4)
27      (adjacent l5 l6)
28      (adjacent l5 l8)
29      (adjacent l6 l5)
30      (adjacent l6 l3)
31      (adjacent l6 l9)
32      (adjacent l7 l4)
33      (adjacent l7 l8)
34      (adjacent l8 l5)
35      (adjacent l8 l7)
36      (adjacent l8 l9)
37      (adjacent l9 l6)
38      (adjacent l9 l8)
39      )
40      (:goal
```

```
41      (and
42      (at n1 l1)
43      (at n2 l2)
44      (at n3 l3)
45      (at n4 l4)
46      (at n5 l5)
47      (at n6 l6)
48      (at n7 l7)
49      (at n8 l8)
50      (blank l9)
51      )
52      )
53
54  )
```

blocks.pddl

```
1  (define (domain blocks)
2      (:requirements :strips :typing:equality
3                     :universal-preconditions
4                     :conditional-effects)
5      (:types physob)
6      (:predicates
7          (ontable ?x - physob)
8          (clear ?x - physob)
9          (on ?x ?y - physob)
10     )
11     (:action move
12         :parameters (?x ?y - physob)
13         :precondition (and(clear ?x)(clear ?y))
14         :effect (and
15             (forall (?z - physob)
16                 ( when ( on ?x ?z )
```

```
17                      (and(not (on ?x ?z)) (clear ?z) )
18                  )
19              )
20              (not (clear ?y))
21              (on ?x ?y)
22              (not (ontable ?x))
23
24          )
25      )
26
27      (:action moveToTable
28          :parameters (?x - physob)
29          :precondition (and (clear ?x) (not (ontable ?x) ) )
30          :effect (and
31              (forall (?z -physob)
32                  (when (on ?x ?z)
33                      ( and (not (on ?x ?z))(clear ?z) )
34                  )
35              )
36              (ontable ?x)
37          )
38      )
39
40  )
```

blocksprob.pddl

```
1  (define (problem prob)
2      (:domain blocks)
3      (:objects A B C D E F - physob)
4      (:init (clear A)(on A B)(on B C)(ontable C)(ontable D)
5      (ontable F)(on E D)(clear E)(clear F)
6      )
```

```
7       (:goal (and (clear F)(on F A)(on A C)(ontable C)
8       (clear E)(on E B)(on B D)(ontable D))
9       )
10
11  )
```

4 Results

Instruction:

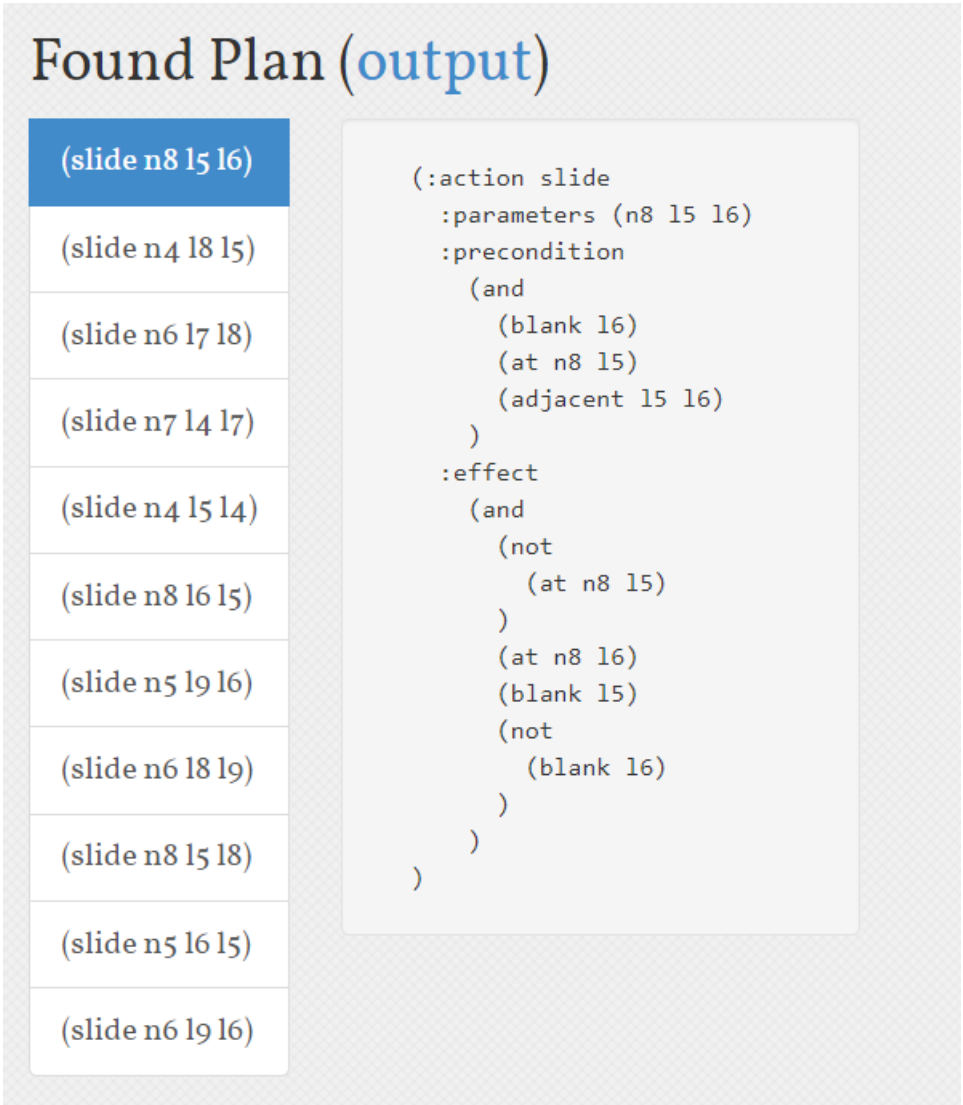

Figure 1: Plan 1

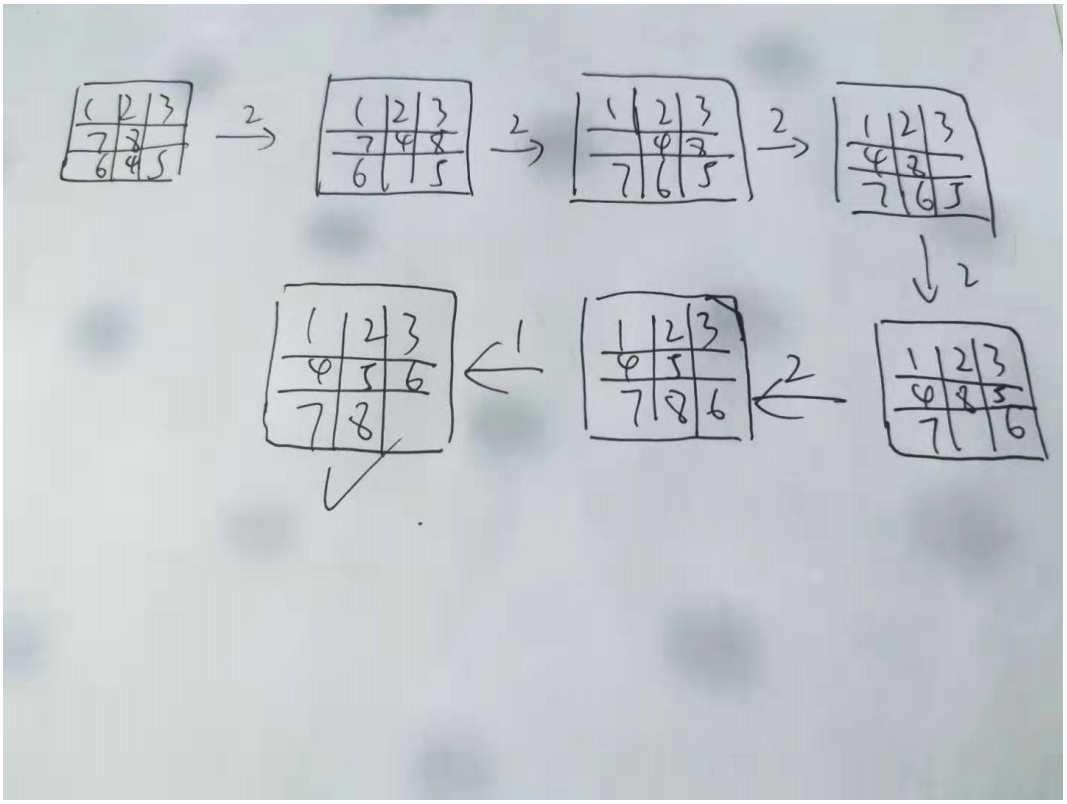

Figure 2: Instruction

Figure 3: Plan 2