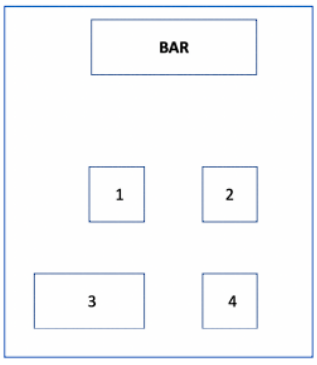
Artificial Intelligence II Assignment

Giacomo Lugano, Luca Sortino, Bauyrzhan Zhakanov May 8, 2023

Robotic Coffee Shop

Abstract

This short report concerns the first Artificial Intelligence II assignment with the purpose of modelling a robotic coffee shop scenario in which customers can order two different kind of drinks, cold and warm, served by two robots, one is a bartender and the other one is a waiter. The bartender is responsible for preparing drinks, while the waiter serves customers and cleans tables. There are four main problems to solve with different number of customers and kind of drinks and four optional extensions that include another waiter or serving also biscuits together with cold drinks.The coffee shop layout is illustrated in Figure 1. Just to focus on the syntax about PDDL+ we took inspiration from the planning.wiki website.



(a) Table arrangement

Figure 1: Layout of the bar

The coffee shop layout consists of a bar counter at the very top and four tables for customers.

Each table is 1 meter apart from any other, and the bar is 2 meters away from tables 1 and 2. Table 3 is the only table of 2 square meters, and all others are of 1 square meter. The barista robot has to prepare drinks ordered by customers. It can be assumed that all the orders are known at the start of the planning problem. The waiter robot is responsible for serving customers and cleaning tables. It can either grasp a single drink using one of its grippers or use a tray to carry up to three drinks at once. When using the tray, the waiter robot moves slower to maintain balance. The robot is not allowed to leave the tray on a table, and it must be returned to the bar after use.

domain.pddl

For doing this assignment we developed a domain.pddl file in which we encoded:

- Types: describing different types of objects including the robots, the drinks and the tables
- Predicates: describing properties of the objects (i.e. properties of the drink or the robot), they can be true or false
- Functions: fluents that return a number (i.e. the distance between tables, the value for the timer or the number of drink the robot is holding)
- Actions: describing the changes of the state of the world (i.e. preparing hot/cold drinks, cleaning, moving the robot, etc...)

Since we have used ENHSP (Expressive Numeric Heuristic Search Planner) that does not support durative actions, to model durative actions we've used a sequence of actions (istantaneus), process and event, the idea is that the action trigger the beginning of the process that simulate the duration of the action at the end of which an event will ne triggered stopping the process and realizing the effect of the action. This also allows to model a multi agent problem, since the process is related to the specific robot, in which each agent can perform different action simultaneously.

All the actions are conditioned by the process Timer based on the value assigned by the action, it will launch different events.

```
(:process Timer
        :parameters (?r -robot)
        :precondition (< (tim ?r) 100)
        :effect (increase (tim ?r) (* #t 1.0))
)
```

In particular

- 0 < t <= 10 Prepare Hot Drink
- 21 < t <= 27 Prepare Cold Drink
- 38 < t <= 42 Move Slow
- 53 < t <= 55 Move Fast
- 66 < t <= 74 Clean

All the conditions are expressed in time units of the planner (considering the maximum time of each action, the time unit we are considering is equal to 2 time units of the planner). Moreover the timer works also as a flag to describe if the robot is busy in performing a durative action ((tim ?r)<100) or it's free ((tim ?r) >100).

Move Action

```
(:action Move
   :parameters (?t1 -table ?r -robot ?t2 -table)
   :precondition (and (waiter ?r)
                      (robot_pos ?r ?t1)
                      (>= (tim ?r) 100))
   :effect (and (to ?t2)
           (when (holding_tray ?r) (assign (tim ?r) 38))
           (when (not (holding_tray ?r)) (assign (tim ?r) 53)))
   )
```

```
(:event Moved
    :parameters (?t1 -table ?r -robot ?t2 -table)
    :precondition (and (or (and (holding_tray ?r)
                  (>= (tim ?r) (+ (* (dist ?t1 ?t2) 2) 38))
                  (<= (tim ?r) (+ (* (dist ?t1 ?t2) 2) 48)))
                  (and (not (holding_tray ?r))
                  (>= (tim ?r) (+ (dist ?t1 ?t2) 53))
                  (<= (tim ?r) (+ (dist ?t1 ?t2) 63))))
                  (to ?t2) (robot_pos ?r ?t1))
    :effect (and (robot_pos ?r ?t2)
                 (not (to ?t2)) (not (robot_pos ?r ?t1))
                 (assign (tim ?r) 150))
)
```

The move action is conditioned by the robot position and the value of the timer, because of, in case of using the tray, the robot must move slowly in order to ensure that everything is balanced.

Drink Prepare

```
(:action Drink_Prepare
    :parameters (?r -robot ?d -drink ?t -table)
    :precondition (and (not (waiter ?r))
                       (robot_pos ?r ?t)
                       (bar ?t) (>= (tim ?r) 100))
    :effect (and (when (= (tim_c ?d) 10)
            (and (assign (tim ?r) 0) (drink_n ?d)))
            (when (= (tim_c ?d) 0)
            (and (assign (tim ?r) 21) (drink_n ?d))))
)
```

```
(:event Drink_Prepared
    :parameters (?r -robot ?d -drink ?t -table)
    :precondition (and (or (and (= (tim_c ?d) 10)
                   (>= (tim ?r) 10) (<= (tim ?r) 20))
                   (and (= (tim_c ?d) 0) (>= (tim ?r) 27)
                   (<= (tim ?r) 37))) (drink_n ?d)
                   (robot_pos ?r ?t) (bar ?t))
    :effect (and (drink_on_table ?d ?t)
            (assign (tim ?r) 150)
            (when (= (tim_c ?d) 10) (assign (tim_c ?d) 8))
            (not (drink_n ?d)))
)

(:process Cooling_down
    :parameters (?d -drink)
    :precondition (and (<= (tim_c ?d) 8) (> (tim_c ?d) 0))
    :effect (decrease (tim_c ?d) (* #t 1.0))
)
```

This function deals with the preparation of the drink that effectively ends when the drink is on the table defined as bar. A second timer like was introduced to simulate the drink temperature (tim c ?d), in detail the hot drinks start with a value of 10 while the cold drink with a value of zero, this difference modify the behavior of the drink prepare action that will take a bigger time for hot drinks and a smaller one for cold ones and will simulate the cooling down of the drinks. Moreover an additional predicate was introduced to exchange the information of which drink is under preparation between the initial action and final event (drink n ?d), this predicates can be easily deleted by using the drink temperature timer (expressed in the process cooling down) as a flag to pass the same information, but after some trials this doesn't seams to introduce big advantages.

Pick Up/Put Down Drink

```
(:action Pick_up_drink
    :parameters (?r -robot ?d -drink ?t -table)
    :precondition (and (waiter ?r) (robot_pos ?r ?t)
                  (or (and (not (holding_tray ?r))
                  (= (holding ?r ) 0))
                  (and (holding_tray ?r) (< (holding ?r) 3)))
                  (drink_on_table ?d ?t) (>= (tim ?r) 100))
    :effect (and (increase (holding ?r) 1)
            (drink_holded ?d ?r)
            (not (drink_on_table ?d ?t)))
    )

(:action Put_down_drink
    :parameters (?r -robot ?d -drink ?t -table ?bis -biscuit)
    :precondition (and (waiter ?r) (robot_pos ?r ?t)
                  (drink_holded ?d ?r) (>= (tim ?r) 100))
    :effect (and (decrease (holding ?r) 1)
            (not (drink_holded ?d ?r)) (drink_on_table ?d ?t)
            (when (and (> (tim_c ?d) 0) (not (bar ?t)))
            (assign (tim_c ?d) 10))
            (when (and (<= (tim_c ?d) 0) (not (bar ?t)))
            (biscuit_deliver ?bis ?t)))
)
```

These functions concern the waiter robot and are for the specific for delivering drinks. As shown in the put down drink action the condition of the drink to still be hot ((tim c ?d)>0) must be respected for serving a hot drink, this is required since the drink from its preparation to its service cool down and we want to force the delivery of hot drinks in the time window in which they are still hot. Moreover this action trigger the predicates (biscuits deliver ?bis ?t) used to signal the robot that it can move to the bar and pick the biscuits to serve together with the cold drink.

Pick Up/Put Down Tray

```
(:action Pick_up_tray
    :parameters (?r -robot ?t -table)
    :precondition (and (waiter ?r) (robot_pos ?r ?t)
                  (bar ?t) (not (holding_tray ?r))
                  (= (holding ?r) 0) (>= (tim ?r) 100))
    :effect (holding_tray ?r)
    )

(:action Put_down_tray
    :parameters (?r -robot ?t -table)
    :precondition (and (waiter ?r) (robot_pos ?r ?t)
                  (bar ?t) (holding_tray ?r)
                  (= (holding ?r) 0) (>= (tim ?r) 100))
    :effect (not (holding_tray ?r))
)
```

These functions are used for giving the possibility to the waiter to use a tray for holding up to three drinks at once. The robot can hold a tray if and only if is not holding a drink yet.

Pick Up/Put Down Biscuits

```
(:action Pick_up_biscuit
    :parameters (?r -robot ?bis -biscuit ?t -table ?t1 -table)
    :precondition (and (waiter ?r) (robot_pos ?r ?t)
                  (bar ?t) (or (and (not (holding_tray ?r))
                  (= (holding ?r) 0)) (and (holding_tray ?r)
                  (< (holding ?r) 3))) (>= (tim ?r) 100)
                  (biscuit_deliver ?bis ?t1))
    :effect (and (increase (holding ?r) 1)
            (biscuit_holded ?bis ?r))
)
```

```
(:action Put_down_biscuit
    :parameters (?r -robot ?bis -biscuit ?t -table)
    :precondition (and (waiter ?r) (robot_pos ?r ?t)
                  (biscuit_holded ?bis ?r)
                  (biscuit_deliver ?bis ?t)
                  (>= (tim ?r) 100))
    :effect (and (decrease (holding ?r) 1)
            (not (biscuit_holded ?bis ?r))
            (biscuit_on_table ?bis ?t)
            (not (biscuit_deliver ?bis ?t)))
)
```

These functions deals with delivering food described by the fourth optional problem in which all the customers with cold drinks will also receive a biscuit but the waiter can not bring at the same time the drink and the biscuit

Clean Action

```
(:action Clean
    :parameters (?r -robot ?t -table)
    :precondition (and (waiter ?r) (robot_pos ?r ?t)
                  (not (holding_tray ?r)) (= (holding ?r) 0)
                  (not (clean ?t)))
    :effect (and (assign (tim ?r) 66))
)

(:event Cleaned
    :parameters (?r -robot ?t -table)
    :precondition (and (robot_pos ?r ?t)
                  (and (>= (tim ?r) (+ (* (t_dim ?t) 4) 66))
                  (<= (tim ?r) (+ (* (t_dim ?t) 4) 76))))
    :effect (and (clean ?t) (assign (tim ?r) 150))
)
```

These functions are responsible of the cleaning action that is constrained by the use of the tray.

Results

With the purpose of solving the problems different planning engines within ENHS were used, unfortunately the optimal planners fail to give back a solution forcing the usage of satisfiability planner that don't optimize the final outcome. Among them sat-hmrph worked at best on all the problems obtaining the solution hereafter reported:

For the first problem we have two customers at table 2 that ordered two cold drinks and table 3 and 4 need to be cleaned. The planner found a solution with the following results:

```
0: (Move b wr t4)
2.0: (Moved b wr t4)
2.0: (Move t4 wr t4)
2.0: (Clean wr t4)
5.0: (Cleaned wr t4)
5.0: (Move t4 wr b)
7.0: (Moved t4 wr b)
7.0: (Move b wr b)
9.0: (Moved b wr t4)
9.0: (Move t4 wr b)
11.0: (Moved t4 wr b)
11.0: (Move b wr b)
11.0: (Clean wr b)
20.0: (Drink_Prepare br d1 b)
25.0: (Drink_Prepared br d1 b)
40.0: (Drink_Prepare br d2 b)
44.0: (Move b wr t3)
45.0: (Drink_Prepared br d2 b)
45.0: (Drink_Prepare br d2 b)
46.0: (Moved b wr t3)
```

46.0: (Move t3 wr t2)
46.0: (Clean wr t3)
46.0: (Moved t3 wr b)
47.0: (Pick_up_drink wr d1 b)
47.0: (Move b wr t3)
48.0: (Moved b wr t2)
48.0: (Put_down_drink wr d1 t2 bis1)
48.0: (Pick_up_drink wr d1 t2)
48.0: (Put_down_drink wr d1 t2 bis2)
48.0: (Move t2 wr b)
48.0: (Moved t2 wr t3)
49.0: (Move t3 wr t2)
49.0: (Clean wr t3)
49.0: (Moved t3 wr b)
50.0: (Drink_Prepared br d2 b)
50.0: (Pick_up_biscuit wr bis1 b t2)
50.0: (Drink_Prepare br d2 b)
50.0: (Move b wr t3)
51.0: (Moved b wr t2)
51.0: (Put_down_biscuit wr bis1 t2)
51.0: (Move t2 wr b)
51.0: (Moved t2 wr t3)
52.0: (Move t3 wr t2)
52.0: (Clean wr t3)
52.0: (Moved t3 wr b)
53.0: (Pick_up_drink wr d2 b)
53.0: (Pick_up_biscuit wr bis2 b t2)
53.0: (Move b wr t3)
54.0: (Moved b wr t2)
54.0: (Put_down_drink wr d2 t2 bis2)
54.0: (Put_down_biscuit wr bis2 t2)
54.0: (Move t2 wr t3)
54.0: (Moved t2 wr t3)

```
55.0: (Drink_Prepared br d2 b)
55.0: (Clean wr t3)
62.0: (Cleaned wr t3)
62.0: @PlanEND
```

Plan-Length:118
Elapsed Time: 62.0
Metric (Search):98.0
Planning Time (*msec*): 984
Heuristic Time (*msec*): 443
Search Time (*msec*): 531
Expanded Nodes: 739
States Evaluated: 1771
Fixed constraint violations during search (zero-crossing): 0
Number of Dead-Ends detected: 3
Number of Duplicates detected: 740

It is not the optimal solution because there are repeated actions that could be avoided

For the second problem we have to deal with four customers at table 3 that ordered two cold drinks and two warm drinks and then table 4 need to be cleaned. The planner found a solution with the following results:

```
0: (Move b wr t1)
1.0: (Moved b wr t1)
1.0: (Clean wr t1)
4.0: (Cleaned wr t1)
4.0: (Move t1 wr b)
5.0: (Moved t1 wr b)
5.0: (Clean wr b)
13.0: (Drink_Prepare br d1 b)
18.0: (Drink_Prepared br d1 b)
38.0: (Pick_up_drink wr d1 b)
```

38.0: (Move b wr t3)
40.0: (Moved b wr t3)
40.0: (Put_down_drink wr d1 t3 bis2)
40.0: (Pick_up_drink wr d1 t3)
40.0: (Put_down_drink wr d1 t3 bis1)
40.0: (Move t3 wr b)
42.0: (Moved t3 wr b)
42.0: (Pick_up_biscuit wr bis2 b t3)
42.0: (Move b wr b)
42.0: (Clean wr b)
75.0: (Pick_up_biscuit wr bis1 b t3)
75.0: (Move b wr t3)
77.0: (Moved b wr t3)
77.0: (Put_down_biscuit wr bis1 t3)
77.0: (Put_down_biscuit wr bis2 t3)
77.0: (Drink_Prepare br d2 b)
80.0: (Move t3 wr b)
82.0: (Drink_Prepared br d2 b)
82.0: (Moved t3 wr b)
82.0: (Pick_up_drink wr d2 b)
82.0: (Move b wr b)
82.0: (Clean wr b)
115.0: (Move b wr t3)
117.0: (Moved b wr t3)
117.0: (Put_down_drink wr d2 t3 bis2)
117.0: (Move t3 wr t4)
119.0: (Moved t3 wr b)
119.0: (Move b wr t3)
120.0: (Drink_Prepare br d4 b)
121.0: (Moved b wr t4)
126.0: (Move t4 wr b)
126.0: (Moved t4 wr t3)
127.0: (Move t3 wr b)

```
129.0: (Drink_Prepared br d4 b)
129.0: (Moved t3 wr b)
129.0: (Pick_up_drink wr d4 b)
129.0: (Move b wr t3)
129.0: (Drink_Prepare br d1 b)
131.0: (Moved b wr t3)
131.0: (Put_down_drink wr d4 t3 bis1)
131.0: (Move t3 wr b)
133.0: (Moved t3 wr b)
133.0: (Move b wr b)
133.0: (Clean wr b)
134.0: (Drink_Prepared br d1 b)
166.0: (Move b wr t3)
167.0: (Drink_Prepare br d3 b)
168.0: (Moved b wr t3)
176.0: (Drink_Prepared br d3 b)
177.0: (Move t3 wr t4)
179.0: (Moved t3 wr b)
179.0: (Pick_up_drink wr d3 b)
179.0: (Move b wr t3)
181.0: (Moved b wr t4)
181.0: (Move t4 wr b)
181.0: (Moved t4 wr t3)
182.0: (Put_down_drink wr d3 t3 bis2)
182.0: @PlanEND
```

Plan-Length: 249

Elapsed Time: 182.0

Metric (Search): 226.0

Planning Time (*msec*): 48805

Heuristic Time (*msec*): 45893

Search Time (*msec*): 48240

Expanded Nodes: 139689

States Evaluated: 294620

Fixed constraint violations during search (zero-crossing): 0
Number of Dead-Ends detected: 3841
Number of Duplicates detected: 187082

For the third problem we have two customers at table 4 and two customers at table 1, all of them ordered warm drinks. After that table 3 need to be cleaned. The planner found a solution with the following results:

```
0: (Move b wr t3)
2.0: (Moved b wr t3)
2.0: (Clean wr t3)
9.0: (Cleaned wr t3)
9.0: (Move t3 wr b)
11.0: (Moved t3 wr b)
11.0: (Move b wr b)
11.0: (Clean wr b)
44.0: (Move b wr t1)
45.0: (Moved b wr t1)
45.0: (Move t1 wr t3)
45.0: (Drink_Prepare br d2 b)
45.0: (Moved t1 wr t3)
47.0: (Move t3 wr t4)
49.0: (Moved t3 wr b)
50.0: (Drink_Prepared br d2 b)
50.0: (Pick_up_drink wr d2 b)
50.0: (Move b wr t1)
51.0: (Moved b wr t1)
51.0: (Put_down_drink wr d2 t1 bis2)
51.0: (Pick_up_drink wr d2 t1)
51.0: (Put_down_drink wr d2 t1 bis1)
51.0: (Move t1 wr b)
51.0: (Drink_Prepare br d1 b)
51.0: (Moved t1 wr t4)
```

```
56.0: (Drink_Prepared br d1 b)
56.0: (Move t4 wr b)
58.0: (Moved t4 wr b)
58.0: (Pick_up_biscuit wr bis1 b t1)
58.0: (Move b wr t1)
59.0: (Moved b wr t1)
59.0: (Put_down_biscuit wr bis1 t1)
59.0: (Move t1 wr b)
60.0: (Moved t1 wr b)
60.0: (Pick_up_drink wr d1 b)
60.0: (Move b wr t1)
61.0: (Moved b wr t1)
61.0: (Put_down_drink wr d1 t1 bis1)
61.0: (Move t1 wr b)
62.0: (Moved t1 wr b)
62.0: (Pick_up_biscuit wr bis2 b t1)
62.0: (Move b wr b)
62.0: (Clean wr b)
95.0: (Move b wr t1)
96.0: (Moved b wr t1)
96.0: (Put_down_biscuit wr bis2 t1)
96.0: (Move t1 wr t4)
96.0: (Drink_Prepare br d2 b)
96.0: (Moved t1 wr t4)
99.0: (Move t4 wr t4)
101.0: (Drink_Prepared br d2 b)
101.0: (Moved t4 wr b)
101.0: (Drink_Prepare br d3 b)
110.0: (Drink_Prepared br d3 b)
110.0: (Pick_up_drink wr d3 b)
110.0: (Drink_Prepare br d1 b)
114.0: (Move b wr t2)
114.0: (Clean wr b)
```

```
114.0: (Moved b wr t4)
115.0: (Drink_Prepared br d1 b)
115.0: (Put_down_drink wr d3 t4 bis1)
115.0: (Drink_Prepare br d4 b)
124.0: (Drink_Prepared br d4 b)
124.0: (Move t4 wr t4)
124.0: (Moved t4 wr t2)
125.0: (Move t2 wr b)
125.0: (Moved t2 wr t4)
127.0: (Move t4 wr t4)
129.0: (Moved t4 wr b)
129.0: (Pick_up_drink wr d4 b)
129.0: (Clean wr b)
129.0: (Moved b wr t4)
130.0: (Put_down_drink wr d4 t4 bis2)
130.0: @PlanEND
```

Plan-Length: 203
Elapsed Time: 130.0
Metric (Search): 176.0
Planning Time (*msec*): 31052
Heuristic Time (*msec*): 28716
Search Time (*msec*): 30549
Expanded Nodes: 98749
States Evaluated: 191859
Fixed constraint violations during search (zero-crossing): 0
Number of Dead-Ends detected: 1455
Number of Duplicates detected: 134353

For the fourth problem there are two customers at table 4 and two customers at table 1, all of them ordered cold drinks. We also have four customers at table 3 that ordered warm drinks. Then table table 4 need to be cleaned. The planner found a solution with the following results:

```
0: (Move b wr t4)
2.0: (Moved b wr t4)
2.0: (Move t4 wr t1)
2.0: (Clean wr t4)
5.0: (Cleaned wr t4)
5.0: (Move t4 wr b)
5.0: (Drink_Prepare br d2 b)
5.0: (Moved t4 wr t1)
9.0: (Move t1 wr t4)
9.0: (Moved t1 wr t4)
10.0: (Drink_Prepared br d2 b)
10.0: (Move t4 wr t4)
12.0: (Moved t4 wr b)
12.0: (Pick_up_drink wr d2 b)
12.0: (Move b wr b)
14.0: (Moved b wr t4)
14.0: (Put_down_drink wr d2 t4 bis2)
14.0: (Pick_up_drink wr d2 t4)
14.0: (Put_down_drink wr d2 t4 bis3)
14.0: (Pick_up_drink wr d2 t4)
14.0: (Put_down_drink wr d2 t4 bis1)
14.0: (Pick_up_drink wr d2 t4)
14.0: (Put_down_drink wr d2 t4 bis4)
14.0: (Move t4 wr b)
16.0: (Moved t4 wr b)
16.0: (Pick_up_biscuit wr bis4 b t4)
16.0: (Move b wr b)
16.0: (Clean wr b)
49.0: (Move b wr t4)
51.0: (Moved b wr t4)
51.0: (Put_down_biscuit wr bis4 t4)
51.0: (Move t4 wr b)
53.0: (Moved t4 wr b)
```

```
53.0: (Pick_up_biscuit wr bis1 b t4)
53.0: (Move b wr b)
53.0: (Clean wr b)
86.0: (Move b wr t4)
88.0: (Moved b wr t4)
88.0: (Put_down_biscuit wr bis1 t4)
88.0: (Move t4 wr b)
90.0: (Moved t4 wr b)
90.0: (Pick_up_biscuit wr bis2 b t4)
90.0: (Move b wr b)
90.0: (Clean wr b)
123.0: (Move b wr t4)
125.0: (Moved b wr t4)
125.0: (Put_down_biscuit wr bis2 t4)
125.0: (Pick_up_drink wr d2 t4)
125.0: (Put_down_drink wr d2 t4 bis2)
125.0: (Move t4 wr t4)
127.0: (Moved t4 wr b)
127.0: (Pick_up_biscuit wr bis3 b t4)
127.0: (Move b wr b)
129.0: (Moved b wr t4)
129.0: (Pick_up_drink wr d2 t4)
129.0: (Put_down_biscuit wr bis3 t4)
129.0: (Put_down_drink wr d2 t4 bis3)
129.0: (Move t4 wr b)
131.0: (Moved t4 wr b)
131.0: (Pick_up_biscuit wr bis3 b t4)
131.0: (Move b wr b)
131.0: (Clean wr b)
164.0: (Move b wr t1)
165.0: (Moved b wr t1)
165.0: (Move t1 wr t1)
166.0: (Moved t1 wr b)
```

166.0: (Drink_Prepare br d4 b)
171.0: (Drink_Prepared br d4 b)
171.0: (Pick_up_drink wr d4 b)
171.0: (Move b wr t4)
172.0: (Moved b wr t1)
172.0: (Put_down_drink wr d4 t1 bis4)
172.0: (Pick_up_drink wr d4 t1)
172.0: (Put_down_drink wr d4 t1 bis1)
172.0: (Pick_up_drink wr d4 t1)
172.0: (Put_down_drink wr d4 t1 bis3)
172.0: (Put_down_biscuit wr bis3 t1)
172.0: (Move t1 wr b)
172.0: (Drink_Prepare br d1 b)
172.0: (Moved t1 wr t4)
173.0: (Move t4 wr t1)
173.0: (Moved t4 wr t1)
177.0: (Drink_Prepared br d1 b)
177.0: (Move t1 wr t1)
178.0: (Moved t1 wr b)
178.0: (Pick_up_drink wr d1 b)
178.0: (Move b wr b)
179.0: (Moved b wr t1)
179.0: (Move t1 wr b)
180.0: (Moved t1 wr b)
180.0: (Move b wr t4)
182.0: (Moved b wr t4)
182.0: (Put_down_drink wr d1 t4 bis3)
182.0: (Move t4 wr b)
184.0: (Moved t4 wr b)
184.0: (Move b wr b)
184.0: (Clean wr b)
217.0: (Move b wr t3)
219.0: (Moved b wr t3)

```
219.0: (Move t3 wr t1)
219.0: (Drink_Prepare br d3 b)
219.0: (Moved t3 wr t1)
224.0: (Drink_Prepared br d3 b)
224.0: (Move t1 wr t1)
225.0: (Moved t1 wr b)
225.0: (Pick_up_drink wr d3 b)
225.0: (Move b wr b)
226.0: (Moved b wr t1)
226.0: (Put_down_drink wr d3 t1 bis3)
226.0: (Move t1 wr t3)
226.0: (Drink_Prepare br d2 b)
226.0: (Moved t1 wr t3)
231.0: (Drink_Prepared br d2 b)
231.0: (Drink_Prepare br d4 b)
234.0: (Move t3 wr t4)
236.0: (Drink_Prepared br d4 b)
236.0: (Moved t3 wr b)
236.0: (Move b wr t3)
236.0: (Drink_Prepare br d6 b)
238.0: (Moved b wr t4)
238.0: (Move t4 wr b)
238.0: (Moved t4 wr t3)
243.0: (Move t3 wr t4)
245.0: (Drink_Prepared br d6 b)
245.0: (Moved t3 wr b)
245.0: (Pick_up_drink wr d6 b)
245.0: (Move b wr t3)
247.0: (Moved b wr t4)
247.0: (Move t4 wr b)
247.0: (Moved t4 wr t3)
248.0: (Put_down_drink wr d6 t3 bis3)
248.0: (Drink_Prepare br d5 b)
```

```
255.0: (Move t3 wr t4)
257.0: (Drink_Prepared br d5 b)
257.0: (Moved t3 wr b)
257.0: (Pick_up_drink wr d5 b)
257.0: (Move b wr t3)
257.0: (Drink_Prepare br d4 b)
259.0: (Moved b wr t4)
259.0: (Move t4 wr b)
259.0: (Moved t4 wr t3)
261.0: (Put_down_drink wr d5 t3 bis4)
261.0: (Move t3 wr t2)
261.0: (Moved t3 wr t2)
262.0: (Drink_Prepared br d4 b)
262.0: (Move t2 wr t3)
262.0: (Drink_Prepare br d7 b)
262.0: (Moved t2 wr t3)
269.0: (Move t3 wr t4)
271.0: (Drink_Prepared br d7 b)
271.0: (Moved t3 wr b)
271.0: (Pick_up_drink wr d7 b)
271.0: (Move b wr t3)
271.0: (Drink_Prepare br d3 b)
273.0: (Moved b wr t4)
273.0: (Move t4 wr b)
273.0: (Moved t4 wr t3)
276.0: (Drink_Prepared br d3 b)
276.0: (Put_down_drink wr d7 t3 bis4)
276.0: (Drink_Prepare br d1 b)
281.0: (Drink_Prepared br d1 b)
281.0: (Move t3 wr t3)
281.0: (Drink_Prepare br d8 b)
281.0: (Moved t3 wr t3)
290.0: (Drink_Prepared br d8 b)
```

```
292.0: (Move t3 wr b)
294.0: (Moved t3 wr b)
294.0: (Pick_up_drink wr d8 b)
294.0: (Move b wr t3)
296.0: (Moved b wr t3)
296.0: (Put_down_drink wr d8 t3 bis3)
296.0: @PlanEND
```

Plan-Length: 467
Elapsed Time: 296.0
Metric (Search): 407.0
Planning Time (*msec*): 577337
Heuristic Time (*msec*): 555111
Search Time (*msec*): 576736
Expanded Nodes: 880099
States Evaluated: 2049570
Fixed constraint violations during search (zero-crossing): 0
Number of Dead-Ends detected: 14065
Number of Duplicates detected: 1204999

Conclusions

For this work we used EHNSP, it represents a good choice for planning problems involving numerical aspects, thanks also to its simplicity, but it doesn't support durative actions requiring the action-process-event trick explained before, this introduce a bigger number of predicates and functions leading to a bigger complexity of the planning problem and a consequent slow down in the performance together with a bigger memory space requirements, indeed to obtain a solution it was needed to set the tolerance of the planner to 1, reducing the memory requirements. For additional information and the source code please refer to the github repository: https://github.com/jek97/Robotic-coffee-shop.git