

# Informatics 2D Coursework 2 Report

## 1. (5 marks) Task 1.1: Describing The World State

### Constants:

- **cXY** to denote the cell in **X**th column and **Y**th row inside the warehouse, counting starting from 1, from left to right, top to bottom. For example, **c32** would be the cell in column 3 and row 2.

### Predicates:

For each object define an atemporal predicate:

*MailBot(o), Scanner(o), DeliveryBelt(o), Package(o), Switch(o)*

Fluent predicates:

- *Connected(i, j)* to denote the cell *i* and cell *j* are connected.
- *On(o, c)* to denote an object *o* (MailBot/Scanner/Packages) is on cell *c*.
- *BeltOn(b, c)* to denote delivery belt *b* is on cell *c*.
- *SwitchOn(s, c)* to denote a switch *s* is on cell *c*.
- *NextToBelt(c)* to denote a cell *c* is next to the delivery belt.
- *TurnedOn(s)* to denote the switch *s* is turned on.
- *Scanned(p)* to denote the package *p* has been scanned by MailBot.
- *IsHolding(m)* to denote MailBot *m* is holding an object.
- *ObjectHeld(m, o)* to denote an object *o* is held by MailBot *m*.
- *Delivered(p)* to denote a package *p* is delivered.

### Figure 1 Initial State:

*Connected(c11,c12) ∧ Connected(c12,c11) Connected(c11,c21) ∧ Connected(c21,c11)*  
*Connected(c12,c22) ∧ Connected(c22,c12) Connected(c13,c23) ∧ Connected(c23,c13)*  
*Connected(c13,c14) ∧ Connected(c14,c13) Connected(c14,c24) ∧ Connected(c24,c14)*  
*Connected(c14,c15) ∧ Connected(c15,c14) Connected(c15,c25) ∧ Connected(c25,c15)*  
*Connected(c21,c22) ∧ Connected(c22,c21) Connected(c22,c32) ∧ Connected(c32,c22)*  
*Connected(c23,c33) ∧ Connected(c33,c23) Connected(c23,c24) ∧ Connected(c24,c23)*  
*Connected(c24,c34) ∧ Connected(c34,c24) Connected(c24,c25) ∧ Connected(c25,c24)*  
*Connected(c25,c35) ∧ Connected(c35,c25) Connected(c31,c41) ∧ Connected(c41,c31)*  
*Connected(c31,c32) ∧ Connected(c32,c31) Connected(c32,c33) ∧ Connected(c33,c32)*  
*Connected(c33,c43) ∧ Connected(c43,c33) Connected(c33,c34) ∧ Connected(c34,c33)*  
*Connected(c34,c44) ∧ Connected(c44,c34) Connected(c34,c35) ∧ Connected(c35,c34)*  
*Connected(c35,c45) ∧ Connected(c45,c35) Connected(c41,c51) ∧ Connected(c51,c41)*  
*Connected(c42,c52) ∧ Connected(c52,c42) Connected(c43,c53) ∧ Connected(c53,c43)*  
*Connected(c43,c44) ∧ Connected(c44,c43) Connected(c44,c54) ∧ Connected(c54,c44)*  
*Connected(c44,c45) ∧ Connected(c45,c44) Connected(c45,c55) ∧ Connected(c55,c45)*  
*Connected(c53,c54) ∧ Connected(c54,c53) Connected(c54,c55) ∧ Connected(c55,c54) MailBot(M) ∧*  
*Scanner(K) ∧ Switch(S) ∧ DeliveryBelt(B) ∧ Package(001) ∧ Package(010) ∧ Package(011) ∧*  
*On(K, c11) ∧ On(M, c33) ∧ On(S, c53) ∧ On(B, c42) ∧ On(B, c52) ∧ On(001, c15) ∧ On(010, c35)*  
*∧ On(011, c55) ∧ NextToBelt(c31) ∧ NextToBelt(c41) ∧ NextToBelt(c43) ∧ NextToBelt(c51) ∧*  
*NextToBelt(c53)*

2. (5 marks) Task 1.2: Actions

*Action(Move(m, cell1, cell2))*

Precondition:  $\text{MailBot}(m) \wedge \text{On}(m, \text{cell1}) \wedge \text{Connected}(\text{cell1}, \text{cell2})$

Effect:  $\neg \text{On}(m, \text{cell1}) \wedge \text{On}(m, \text{cell2})$

*Action(PickUp(m, o, cell))*

Precondition:  $\text{MailBot}(m) \wedge \text{On}(m, \text{cell}) \wedge \text{On}(o, \text{cell}) \wedge \neg \text{IsHolding}(m)$

Effect:  $\neg \text{On}(o, \text{cell}) \wedge \text{IsHolding}(m) \wedge \text{ObjectHeld}(m, o)$

*Action(Drop(m, o, cell))*

Precondition:  $\text{MailBot}(m) \wedge \text{On}(m, \text{cell}) \wedge \text{ObjectHeld}(m, o)$

Effect:  $\text{On}(o, \text{cell}) \wedge \neg \text{ObjectHeld}(m, o) \wedge \neg \text{IsHolding}(m)$

*Action(Deliver(c, m, p, b))*

Precondition:  $\text{MailBot}(m) \wedge \text{Package}(p) \wedge \text{DeliveryBelt}(b) \wedge \text{ObjectHeld}(m, p) \wedge \text{Scanned}(p) \wedge \text{On}(m, c) \wedge \text{NextToBelt}(c)$

Effect:  $\neg \text{ObjectHeld}(m, p) \wedge \neg \text{IsHolding}(m) \wedge \text{Delivered}(p)$

*Action(TurnOnSwitch(m, s, cell))*

Precondition:  $\text{MailBot}(m) \wedge \text{Switch}(s) \wedge \text{On}(m, \text{cell}) \wedge \text{On}(s, \text{cell}) \wedge \neg \text{TurnedOn}(s)$

Effect:  $\text{TurnedOn}(s)$

*Action(Scan(m, p, k, c))*

Precondition:  $\text{MailBot}(m) \wedge \text{Scanner}(k) \wedge \text{Package}(p) \wedge \text{ObjectHeld}(m, k) \wedge \text{On}(m, c) \wedge \text{On}(p, c)$

Effect:  $\text{Scanned}(p)$

3. (10 marks) Task 1.3: Backwards state-space search

- $g_1 = \text{Delivered}(011)$
- Available Actions:  $\text{Deliver}(c41, \text{mailbot}, 011, \text{belt})$
- Choose:  $\text{Deliver}(c41, \text{mailbot}, 011, \text{belt})$
- $g_2 = \text{MailBot}(\text{mailbot}) \wedge \text{Package}(011) \wedge \text{DeliveryBelt}(\text{belt}) \wedge \text{ObjectHeld}(\text{mailbot}, 011) \wedge \text{Scanned}(011) \wedge \text{On}(\text{mailbot}, c41) \wedge \text{NextToBelt}(c41)$
- Available Actions:  $\text{PickUp}(\text{mailbot}, 011, c41), \text{Move}(\text{mailbot}, c41, c51), \text{Move}(\text{mailbot}, c41, c31)$
- Choose:  $\text{PickUp}(\text{mailbot}, 011, c41)$
- $g_3 = \text{MailBot}(\text{mailbot}) \wedge \text{On}(\text{mailbot}, c41) \wedge \text{On}(011, c41) \wedge \neg \text{IsHolding}(\text{mailbot})$
- Available Actions:  $\text{Move}(\text{mailbot}, c51, c41), \text{Move}(\text{mailbot}, c31, c41)$
- Choose:  $\text{Move}(\text{mailbot}, c31, c41)$
- $g_4 = \text{MailBot}(\text{mailbot}) \wedge \text{On}(\text{mailbot}, c31) \wedge \text{Connected}(c31, c41)$
- Available Actions:  $\text{Move}(\text{mailbot}, c41, c31), \text{Move}(\text{mailbot}, c32, c31)$
- Choose:  $\text{Move}(\text{mailbot}, c32, c31)$
- $g_5 = \text{MailBot}(\text{mailbot}) \wedge \text{On}(\text{mailbot}, c32) \wedge \text{Connected}(c32, c31)$
- Available Actions:  $\text{Move}(\text{mailbot}, c22, c32), \text{Move}(\text{mailbot}, c31, c32), \text{Move}(\text{mailbot}, c33, c32)$
- Choose:  $\text{Move}(\text{mailbot}, c33, c32)$
- $g_6 = \text{MailBot}(\text{mailbot}) \wedge \text{On}(\text{mailbot}, c33) \wedge \text{Connected}(c33, c32)$
- $g_6$  satisfies initial world state. The backwards searching terminates.

The following is the plan to reach the goal state:

1. MAILBOT MOVE from C33 to C32
2. MAILBOT MOVE from C32 to C31
3. MAILBOT MOVE from C31 to C41
4. MAILBOT PICKUP PACKAGE011 at C41
5. MAILBOT DELIVER PACKAGE011 at C41

4. (5 marks) Task 2.1 Test Problem #1

The following is the plan ff planner has produced( $w_g = 3$   $w_h = 5$ ):

1. MOVE MAILBOT C33 C43
2. MOVE MAILBOT C43 C53
3. TURNONSWITCH MAILBOT SWITCH C53
4. MOVE MAILBOT C53 C43
5. MOVE MAILBOT C43 C33
6. MOVE MAILBOT C33 C23
7. MOVE MAILBOT C23 C13
8. MOVE MAILBOT C13 C14
9. MOVE MAILBOT C14 C15
10. PICKUP MAILBOT PACKAGE001 C15
11. MOVE MAILBOT C15 C25
12. MOVE MAILBOT C25 C35
13. MOVE MAILBOT C35 C34
14. MOVE MAILBOT C34 C33
15. MOVE MAILBOT C33 C32
16. MOVE MAILBOT C32 C22
17. MOVE MAILBOT C22 C12
18. MOVE MAILBOT C12 C11
19. DROP MAILBOT PACKAGE001 C11
20. PICKUP MAILBOT SCANNER C11
21. SCAN MAILBOT PACKAGE001 SCANNER C11
22. DROP MAILBOT SCANNER C11
23. PICKUP MAILBOT PACKAGE001 C11
24. MOVE MAILBOT C11 C12
25. MOVE MAILBOT C12 C22
26. MOVE MAILBOT C22 C32
27. DELIVER C32 MAILBOT PACKAGE001 BELT SWITCH

5. (5 marks) Task 2.2 Test Problem #2

The following is the plan ff planner has produced( $w_g = 3$   $w_h = 5$ ):

1. MOVE MAILBOT C33 C34
2. MOVE MAILBOT C34 C35
3. MOVE MAILBOT C35 C45
4. MOVE MAILBOT C45 C55
5. PICKUP MAILBOT PACKAGE011 C55
6. MOVE MAILBOT C55 C54
7. MOVE MAILBOT C54 C53
8. TURNONSWITCH MAILBOT SWITCH C53
9. MOVE MAILBOT C53 C43
10. MOVE MAILBOT C43 C33
11. MOVE MAILBOT C33 C32

12. DROP MAILBOT PACKAGE011 C32
13. MOVE MAILBOT C32 C22
14. MOVE MAILBOT C22 C21
15. MOVE MAILBOT C21 C11
16. PICKUP MAILBOT SCANNER C11
17. MOVE MAILBOT C11 C21
18. MOVE MAILBOT C21 C22
19. MOVE MAILBOT C22 C32
20. SCAN MAILBOT PACKAGE011 SCANNER C32
21. MOVE MAILBOT C32 C33
22. MOVE MAILBOT C33 C34
23. MOVE MAILBOT C34 C24
24. MOVE MAILBOT C24 C25
25. MOVE MAILBOT C25 C15
26. SCAN MAILBOT PACKAGE001 SCANNER C15
27. DROP MAILBOT SCANNER C15
28. PICKUP MAILBOT PACKAGE001 C15
29. MOVE MAILBOT C15 C25
30. MOVE MAILBOT C25 C35
31. MOVE MAILBOT C35 C34
32. MOVE MAILBOT C34 C33
33. MOVE MAILBOT C33 C32
34. DELIVER C32 MAILBOT PACKAGE001 BELT SWITCH
35. PICKUP MAILBOT PACKAGE011 C32
36. DELIVER C32 MAILBOT PACKAGE011 BELT SWITCH

6. (5 marks) Task 3.1 Design

This problem requires the MailBot to collect 3 packages each placed far apart from each other. The warehouse is roughly separated into two sides extending from the initial position of the MailBot with one side extends to two packages and the other extends to the last package, the scanner, and the switch. There is also only one cell adjacent to the delivery belt. The following is a figure of the designed warehouse:

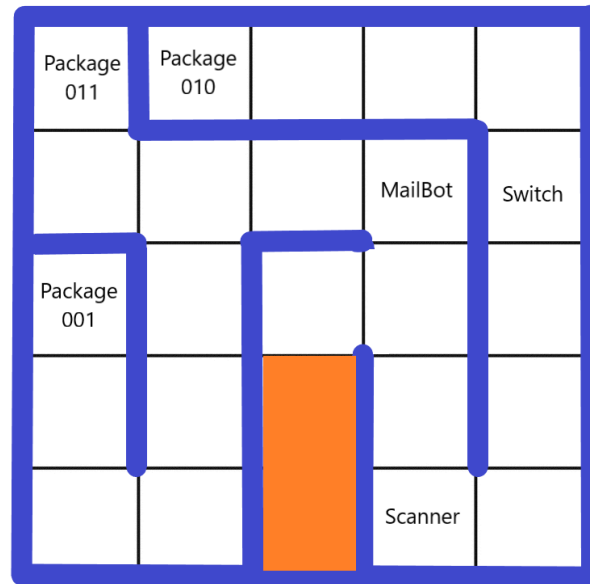


Figure 1: A more challenging problem

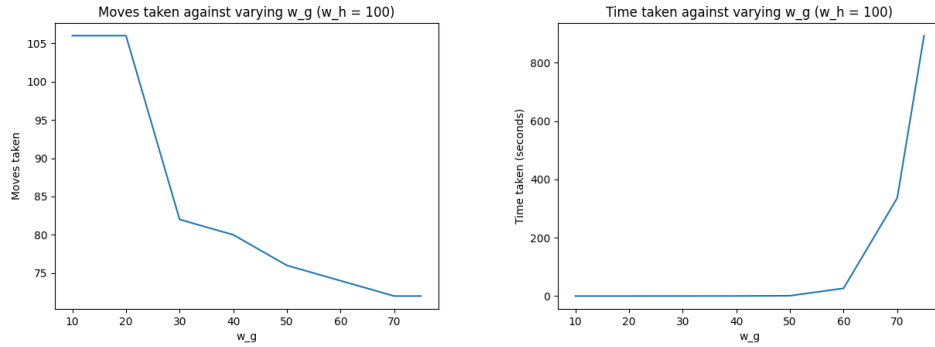
7. (10 marks) Task 3.2 Evaluation

Testing different  $w_g$  values against the same  $w_h$  value have produced result as demonstrated in the following table:

$w_g$	$w_h$	Time taken (Seconds)	Moves taken
1	5	0.01	106
10	100	0.01	106
20	100	0.01	106
30	100	0.16	82
40	100	0.26	80
50	100	1.11	76
60	100	26.55	74
70	100	336.22	72
75	100	892.79	72

Table 1: Performance relative to varying  $w_g$  and  $w_h$  values

Graphs comparing the time and moves taken against the varying  $w_g$  values are also produced using data from the table above:



(a) Moves taken against varying  $w_g$  values where  $w_h$  is held constant at 100 (b) Time taken against varying  $w_g$  values where  $w_h$  is held constant at 100

Figure 2: Graphs produced using data from Table 1

From Figure 2, the data shows a gradual decrease in moves taken and a gradual increase in time taken to reach a valid solution as  $w_g$  increases. This trend indicates that as  $w_g$  increases, the algorithm evaluates the cost to reach a world state( $g(s)$ ) more heavily at each step, hence the plan in the end would need less moves to reach the goal state as lesser the moves taken to reach goal state lesser the  $g(s)$ . Similarly, because the algorithm is more strict on the cost to reach a world state at each step, it would be more difficult to find a plan that would reach goal state, and hence more time would be needed to find a plan.

#### 8. (15 marks) Task 4.1 Energy Station

The following is the plan ff planner has produced( $w_g = 3$   $w_h = 5$ ):

1. MOVE MAILBOT C33 C43
2. MOVE MAILBOT C43 C53
3. TURNONSWITCH MAILBOT SWITCH C53
4. MOVE MAILBOT C53 C43
5. MOVE MAILBOT C43 C33
6. MOVE MAILBOT C33 C23
7. MOVE MAILBOT C23 C13
8. RECHARGE MAILBOT C13 STATION
9. MOVE MAILBOT C13 C14
10. MOVE MAILBOT C14 C15
11. MOVE MAILBOT C15 C25
12. MOVE MAILBOT C25 C35
13. PICKUP MAILBOT PACKAGE010 C35
14. MOVEWITHOBJECT MAILBOT C35 C34
15. MOVEWITHOBJECT MAILBOT C34 C33
16. MOVEWITHOBJECT MAILBOT C33 C32
17. DROP MAILBOT PACKAGE010 C32
18. MOVE MAILBOT C32 C33
19. MOVE MAILBOT C33 C23
20. MOVE MAILBOT C23 C13
21. RECHARGE MAILBOT C13 STATION
22. MOVE MAILBOT C13 C14

23. MOVE MAILBOT C14 C15
24. PICKUP MAILBOT PACKAGE001 C15
25. MOVEWITHOBJECT MAILBOT C15 C14
26. MOVEWITHOBJECT MAILBOT C14 C13
27. RECHARGE MAILBOT C13 STATION
28. MOVEWITHOBJECT MAILBOT C13 C23
29. MOVEWITHOBJECT MAILBOT C23 C33
30. MOVEWITHOBJECT MAILBOT C33 C32
31. DROP MAILBOT PACKAGE001 C32
32. MOVE MAILBOT C32 C22
33. MOVE MAILBOT C22 C21
34. MOVE MAILBOT C21 C11
35. PICKUP MAILBOT SCANNER C11
36. MOVEWITHOBJECT MAILBOT C11 C21
37. MOVEWITHOBJECT MAILBOT C21 C22
38. MOVEWITHOBJECT MAILBOT C22 C32
39. SCAN MAILBOT PACKAGE010 SCANNER C32
40. SCAN MAILBOT PACKAGE001 SCANNER C32
41. DROP MAILBOT SCANNER C32
42. PICKUP MAILBOT PACKAGE001 C32
43. DELIVER C32 MAILBOT PACKAGE001 BELT SWITCH
44. PICKUP MAILBOT PACKAGE010 C32
45. DELIVER C32 MAILBOT PACKAGE010 BELT SWITCH

9. (15 marks) Task 4.2 Limited Power

The following is the plan ff planner has produced( $w_g = 3$   $w_h = 5$ ):

1. DELIVERYBOT C51 C41
2. DELIVERYBOT C41 C31
3. DBMOVE DELIVERYBOT C31 C32
4. DBMOVE DELIVERYBOT C32 C33
5. MBMOVE MAILBOT C33 C43
6. MBMOVE MAILBOT C43 C53
7. MBTURNONSWITCH MAILBOT SWITCH C53
8. DBMOVE DELIVERYBOT C33 C23
9. DBMOVE DELIVERYBOT C23 C13
10. DBRECHARGE DELIVERYBOT C13 STATION
11. DBMOVE DELIVERYBOT C13 C23
12. DBMOVE DELIVERYBOT C23 C33
13. DBMOVE DELIVERYBOT C33 C32
14. DBMOVE DELIVERYBOT C32 C22
15. DBMOVE DELIVERYBOT C22 C21
16. DBMOVE DELIVERYBOT C21 C11
17. DBPICKUPSCANNER DELIVERYBOT SCANNER C11

18. DBMOVEWITHOBJECT DELIVERYBOT C11 C21 SCANNER
19. DBMOVEWITHOBJECT DELIVERYBOT C21 C22 SCANNER
20. DBMOVEWITHOBJECT DELIVERYBOT C22 C32 SCANNER
21. DBDROP DELIVERYBOT SCANNER C32
22. DBMOVE DELIVERYBOT C32 C33
23. DBMOVE DELIVERYBOT C33 C23
24. DBMOVE DELIVERYBOT C23 C13
25. DBRECHARGE DELIVERYBOT C13 STATION
26. DBMOVE DELIVERYBOT C13 C23
27. DBMOVE DELIVERYBOT C23 C33
28. DBMOVE DELIVERYBOT C33 C32
29. DBPICKUPSCANNER DELIVERYBOT SCANNER C32
30. DBMOVEWITHOBJECT DELIVERYBOT C32 C33 SCANNER
31. DBMOVEWITHOBJECT DELIVERYBOT C33 C43 SCANNER
32. DBMOVEWITHOBJECT DELIVERYBOT C43 C53 SCANNER
33. DBDROP DELIVERYBOT SCANNER C53
34. MBPICKUPSCANNER MAILBOT SCANNER C53
35. MBMOVEWITHOBJECT MAILBOT C53 C54 SCANNER
36. MBMOVEWITHOBJECT MAILBOT C54 C55 SCANNER
37. SCANLARGEPACKAGE MAILBOT PACKAGE011 SCANNER C55
38. MBDROP MAILBOT SCANNER C55
39. DBMOVE DELIVERYBOT C53 C54
40. DBMOVE DELIVERYBOT C54 C55
41. PICKUPTOGETHER MAILBOT DELIVERYBOT PACKAGE011 C55
42. MOVEOBJECTTOGETHER MAILBOT DELIVERYBOT C55 C54 PACKAGE011
43. MOVEOBJECTTOGETHER MAILBOT DELIVERYBOT C54 C53 PACKAGE011
44. DELIVERTOGETHER C53 MAILBOT DELIVERYBOT PACKAGE011 BELT SWITCH

#### 10. (25 marks) Task 4.3 Your Extension

### **Real-world Factor**

A lot of times, the packages are categorized into different types and needed to be delivered onto different types of delivery belts. A good example can be found in airports where packages of a specific flight need to be sent to the flight via a specific belt. This massively affects the planning as there are different types of packages of different sizes and they need to be delivered to separate belts of their corresponding types.

### **Extend Formalisation**

To reflect this real-world factor, a new warehouse problem is constructed with two delivery belts of type 1 and type 2, and packages where package001 is type 1, package010 and package011 are both type 2 and large packages. Everything else stays the same as in Task 4.2. The initial states are shown below:





Figure 3: Initial states for a problem reflecting the real-world factor. The orange delivery belt is the type 1 delivery belt, and the green delivery belt is the type 2 delivery belt. The goal is to deliver both package001 and package010.

- **Predicates:**

The *NextToBelt ?b* predicate are separated into two: *NextToBelt1 ?b* and *NextToBelt2 ?b* to denote the cell adjacent to delivery belt 1 and delivery belt 2.

- **Functions:**

Three additional functions are added to represent the different types of packages and belts.

*PackageType ?p* and *LargePackageType ?p* to denote the type of normal and large packages *p*. The initial states of packages and be set in the problem file.

*BeltType ?b* to denote the types of delivery belts *b*.

- **Actions:**

The original deliver actions are extended to 6 different actions.

*DELIVERTOGETHERTYPE1(cell, mailbot, deliverybot, package, deliverybelt, switch)* and

*DELIVERTOGETHERTYPE2(cell, mailbot, deliverybot, package, deliverybelt, switch)* are actions which allows MailBot and DeliveryBot to deliver a large package of a specific type to a specific delivery belt.

Similarly, the delivery actions for MailBot and DeliveryBot to deliver normal packages are separated into two:

*MBDELIVERTYPE1(cell, mailbot, package, belt, switch)* and

*MBDELIVERTYPE2(cell, mailbot, package, belt, switch)* are actions for MailBot to deliver the package of different types.

*DBDELIVERTYPE1(cell, deliverybot, package, belt, switch)* and

*DBDELIVERTYPE2(cell, deliverybot, package, belt, switch)* are actions for DeliveryBot to deliver the package of different types.

- **Implementation and Plan Found:** After the describe implementation are completed and ff planner has found a valid plan where  $w_g = 3$  and  $w_h = 5$ . The found plan is shown below:

1. MBMOVE MAILBOT C33 C43
2. MBMOVE MAILBOT C43 C53
3. MBTURNONSWITCH MAILBOT SWITCH C53
4. MBMOVE MAILBOT C53 C43
5. MBMOVE MAILBOT C43 C33
6. MBMOVE MAILBOT C33 C23
7. MBMOVE MAILBOT C23 C13

8. MBRECHARGE MAILBOT C13 STATION
9. MBMOVE MAILBOT C13 C23
10. MBMOVE MAILBOT C23 C33
11. DBMOVE DELIVERYBOT C51 C41
12. MBMOVE MAILBOT C33 C32
13. MBMOVE MAILBOT C32 C22
14. MBMOVE MAILBOT C22 C21
15. MBMOVE MAILBOT C21 C11
16. MBPICKUPSCANNER MAILBOT SCANNER C11
17. MBMOVEWITHOBJECT MAILBOT C11 C21 SCANNER
18. MBMOVEWITHOBJECT MAILBOT C21 C22 SCANNER
19. MBMOVEWITHOBJECT MAILBOT C22 C32 SCANNER
20. MBDROP MAILBOT SCANNER C32
21. MBMOVE MAILBOT C32 C33
22. MBMOVE MAILBOT C33 C23
23. MBMOVE MAILBOT C23 C13
24. MBRECHARGE MAILBOT C13 STATION
25. DBMOVE DELIVERYBOT C41 C31
26. DBMOVE DELIVERYBOT C31 C32
27. DBPICKUPSCANNER DELIVERYBOT SCANNER C32
28. MBMOVE MAILBOT C13 C14
29. DBMOVEWITHOBJECT DELIVERYBOT C32 C33 SCANNER
30. DBMOVEWITHOBJECT DELIVERYBOT C33 C34 SCANNER
31. DBDROP DELIVERYBOT SCANNER C34
32. DBMOVE DELIVERYBOT C34 C35
33. MBMOVE MAILBOT C14 C24
34. MBMOVE MAILBOT C24 C34
35. MBPICKUPSCANNER MAILBOT SCANNER C34
36. MBMOVEWITHOBJECT MAILBOT C34 C35 SCANNER
37. SCANLARGEPACKAGE MAILBOT PACKAGE010 SCANNER C35
38. MBDROP MAILBOT SCANNER C35
39. PICKUPTOGETHER MAILBOT DELIVERYBOT PACKAGE010 C35
40. MOVEOBJECTTOGETHER MAILBOT DELIVERYBOT C35 C34 PACKAGE010
41. DELIVERTOGETHERTYPE2 C34 MAILBOT DELIVERYBOT PACKAGE010 BELT2 SWITCH
42. MBMOVE MAILBOT C34 C35
43. MBPICKUPSCANNER MAILBOT SCANNER C35
44. MBMOVEWITHOBJECT MAILBOT C35 C25 SCANNER
45. MBMOVEWITHOBJECT MAILBOT C25 C15 SCANNER
46. SCANPACKAGE MAILBOT PACKAGE001 SCANNER C15
47. MBDROP MAILBOT SCANNER C15
48. MBPICKUP MAILBOT PACKAGE001 C15
49. MBMOVEWITHOBJECT MAILBOT C15 C14 PACKAGE001
50. MBDROP MAILBOT PACKAGE001 C14
51. MBMOVE MAILBOT C14 C13
52. MBRECHARGE MAILBOT C13 STATION
53. MBMOVE MAILBOT C13 C14
54. MBPICKUP MAILBOT PACKAGE001 C14
55. MBMOVEWITHOBJECT MAILBOT C14 C24 PACKAGE001
56. MBMOVEWITHOBJECT MAILBOT C24 C34 PACKAGE001
57. MBMOVEWITHOBJECT MAILBOT C34 C33 PACKAGE001
58. MBMOVEWITHOBJECT MAILBOT C33 C43 PACKAGE001
59. MBDELIVERTYPE1 C43 MAILBOT PACKAGE001 BELT1 SWITCH