Programming assignment - Adrián Gómez Lamuedra

Definition of the problem:

- States: $(x, d_{01}, d_{12}, d_{13}, d_{23}, g)$ where:

  x: position of the robot $x \in \{0,1,2,3\}$

  0: dining room
  1: family room
  2: living room
  3: kitchen

  d_{ij}: door from room i to room j, $d_{ij} \in \{0,1\}$ 0: closed, 1: open

  g: apple grabbed (1) or not (0) $g \in \{0,1\}$

- Initial state: (depends of the input of the program)

  In our case: $(0,1,1,0,1,0)$

- Goal state: $(3, d_{01}, d_{12}, d_{13}, d_{23}, 1)$

- Operators:

  Pick-up. 1- $(3, d_{01}, d_{12}, d_{13}, d_{23}, 0) \rightarrow (3, d_{01}, d_{12}, d_{13}, d_{23}, 1)$

  Go: d→f 2- $(0,1, d_{12}, d_{13}, d_{23}, 0) \rightarrow (1,1, d_{12}, d_{13}, d_{23}, 0)$

  f→d 3- $(1,1, d_{12}, d_{13}, d_{23}, 0) \rightarrow (0,1, d_{12}, d_{13}, d_{23}, 0)$

  f→l 4- $(1, d_{01}, 1, d_{13}, d_{23}, 0) \rightarrow (2, d_{01}, 1, d_{13}, d_{23}, 0)$

  l→f 5- $(2, d_{01}, 1, d_{13}, d_{23}, 0) \rightarrow (1, d_{01}, 1, d_{13}, d_{23}, 0)$

  f→K 6- $(1, d_{01}, d_{12}, 1, d_{23}, 0) \rightarrow (3, d_{01}, d_{12}, 1, d_{23}, 0)$

  K→f 7- $(3, d_{01}, d_{12}, 1, d_{23}, 0) \rightarrow (1, d_{01}, d_{12}, 1, d_{23}, 0)$

  l→K 8- $(2, d_{01}, d_{12}, d_{13}, 1, 0) \rightarrow (3, d_{01}, d_{12}, d_{13}, 1, 0)$

  K→l 9- $(3, d_{01}, d_{12}, d_{13}, 1, 0) \rightarrow (2, d_{01}, d_{12}, d_{13}, 1, 0)$

Open-door:

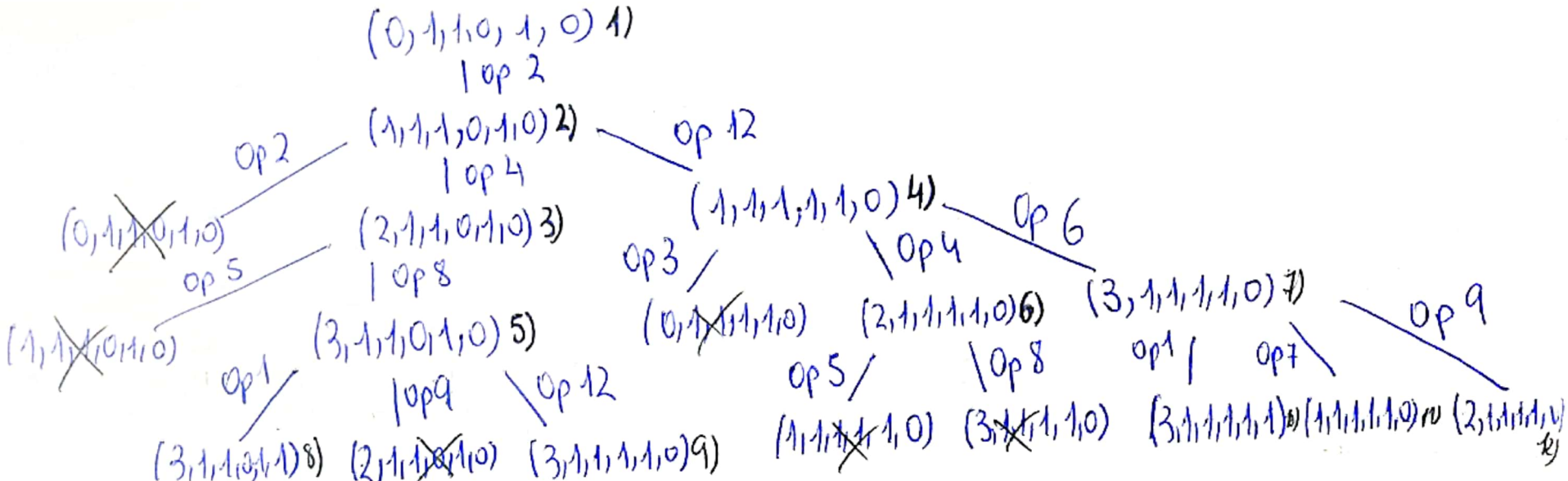
Open d_{01}: 10- $(x, 0, d_{12}, d_{13}, d_{23}, 0) \rightarrow (x, 1, d_{12}, d_{13}, d_{23}, 0)$ if $x \in \{0,1\}$

Open d_{12}: 11- $(x, d_{01}, 0, d_{13}, d_{23}, 0) \rightarrow (x, d_{01}, 1, d_{13}, d_{23}, 0)$ if $x \in \{1,2\}$

Open d_{13}: 12- $(x, d_{01}, d_{12}, 0, d_{23}, 0) \rightarrow (x, d_{01}, d_{12}, 1, d_{23}, 0)$ if $x \in \{1,3\}$

Open d_{23}: 13- $(x, d_{01}, d_{12}, d_{13}, 0, 0) \rightarrow (x, d_{01}, d_{12}, d_{13}, 1, 0)$ if $x \in \{2,3\}$

1- BFS:



Order of expension: 1-2-3-4-5-6-7-8 3-Nodes expanded: 8

Final path: 1-2-3-5-8; → Move to family room, Move to living room, Move to kitchen, Grab the apple

Cost: 4 moves

2- Open and closed lists at each iteration:

Iteration	OPEN	CLOSED
	1	[]
1	2	1
2	3,4	1,2
3	4,5	1,2,3
4	5,6,7	1,2,3,4
5	6,7,8,9	1,2,3,4,5
6	7,8,9	1,2,3,4,5,6
7	8,9,10,11,12	1,2,3,4,5,6,7
8		

4- Data structures used in the implementation:

- For the nodes we use a struct with 3 fields: the robot position (int), the doors configuration (boolean vector), and the grabbed or not apple (boolean). This uniquely determines every state.
- For the list open we use a simple-linked list.
- For the lists parents and closed we use a double-linked list, so that we can also have information about the father of a node, and backtrack to a solution. Linked-lists can have unordered info, which is of our interest.