Report – Case Study (Coloured Cubes)

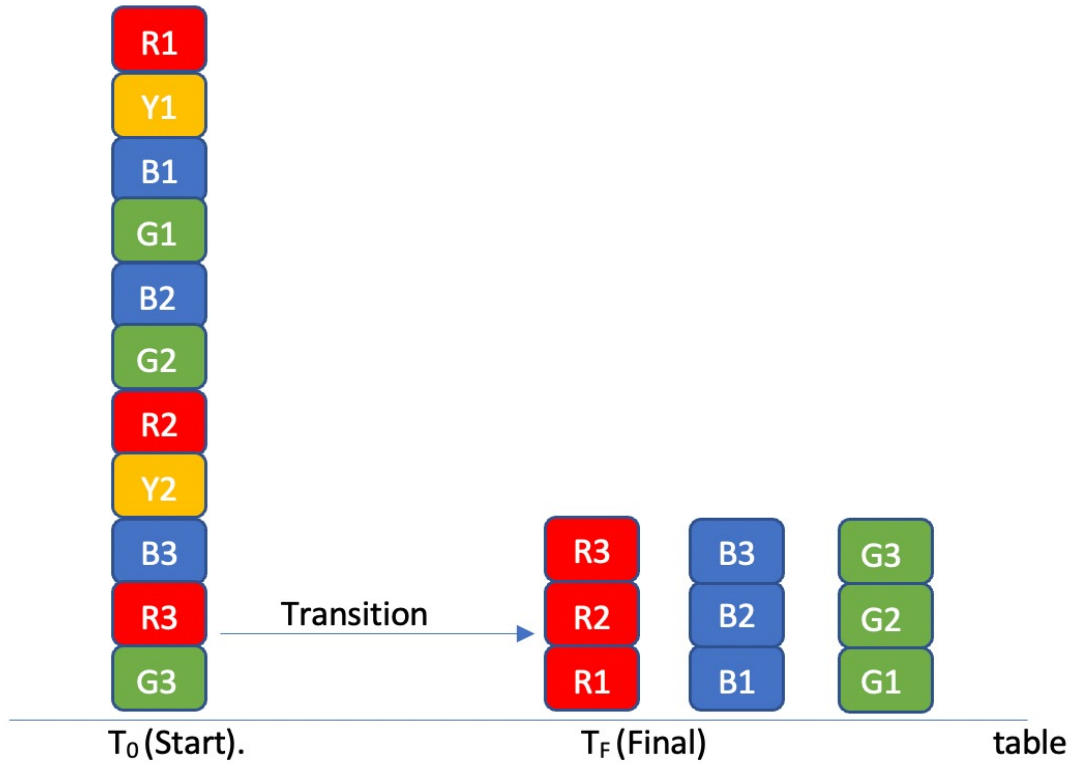
Mohit Mehndiratta
20622275
Master of Data Science

Problem Statement

Consider 11 cubes, out of which 3 cubes are red in color, 3 cubes are blue in color, 3 cubes are green in color and 2 cubes are yellow in color. They are stacked randomly on each other on a table and creating a single stack. We must perform 2 actions. One is "put" in which a robot must put a cube one at a time either on the table if there is an empty space and none of the cube placed matches the color of that cube, otherwise put the cube on the same color of the cube. Another action is "remove", in which a robot must remove the cube which is yellow in color. Thus, in the final step we should have 3 stacks of cubes of colors red, blue, and green.

Transition Diagram

Below is the transition diagram which depicts above scenario:



Please note that we have denoted colours as names like r1 for first red cube, y1 for first Yellow cube etc.

Formal Description - Signature

- Cube = {r1,y1,b1,g1,b2,g2,r2,y2,b3,r3,g3}
- Loc = {r1,y1,b1,g1,b2,g2,r2,y2,b3,r3,g3,t}
- Inertial fluent = on(C1,L1)
- Defined fluent = above(C1,L1)
- Actions = put(C1,L1), remove(C1)
- n = 11
- Step = 0..n

System Description (SD)

Causal Law:

put(C1, L1) causes on(C1, L1)

remove(C1) causes removed(C1)

State Constraints:

-on(C1,L1) if remove(C1)

-on(C1,L2) if on(C1,L1), L1 != L2 % A cube cannot be at 2 locations

-on(C2,C) if on(C1,C), C1 != C2 % one cube can be at top of one another

above(C1,L1) if on(C1,L1)

above(C1,L1) if on(C1,C2), on(C2,L1)

Executability Conditions:

impossible put(C1,L1) if on(C2,C1), loc(L1) % impossible to move an occupied cube

impossible put(C1,C2) if on(C3,C1), on(C2,C4)

impossible put(C1,L1) if on(C1,L2), on(C2,L1), C1 != C2

Encoding SD into ASP

Signature	Values
Relations/Predicates	holds, occurs, removed, fluent
Functions/action	put, on, above, remove
Objects	cube, loc

Inertial Rules:

%% Inertia rule 1: Anything that holds at step I, will also hold at step I+1 if no evidence shows its opposite.

```
holds(F,I+1) :- fluent(inertial,F), holds(F,I), not -holds(F,I+1),
                                                 I<n.
```

%% Inertia rule 2: Anything that does not hold at step I, will also not hold at step I+1, as long as no evidence shows its opposite.

```
-holds(F,I+1) :- fluent(inertial,F), -holds(F,I), not holds(F,I+1),
                                                  I<n.
```

Initial State vs Final State

Initial State

%% Initial Configuration

holds(on(g3,t), 0).

holds(on(r3,g3), 0).

holds(on(b3,r3), 0).

holds(on(y2,b3), 0).

holds(on(r2,y2), 0).

holds(on(g2,r2), 0).

holds(on(b2,g2), 0).

holds(on(g1,b2), 0).

holds(on(b1,g1), 0).

holds(on(y1,b1), 0).

holds(on(r1,y1), 0).

final state

%% Encoding action

occurs(put(r1,t),0).

occurs(remove(y1),1).

occurs(put(b1,t),2).

occurs(put(g1,t),3).

occurs(put(b2,b1),4).

occurs(put(g2,g1),5).

occurs(put(r2,r1),6).

occurs(remove(y2),7).

occurs(put(b3,b2),8).

occurs(put(r3,r2),9).

occurs(put(g3,g2),10).

Query Evaluation.. (I)

Show all the outcomes of holds for each transition

The output contains information like:

holds(on(r3,r2),11), holds(on(g3,g2),11), holds(on(b3,b2),11),
holds(on(r2,r1),11), holds(on(g2,g1),11), holds(on(b2,b1),11),
holds(on(r1,t),11), holds(on(g1,t),11), holds(on(b1,t),11).

This exactly, is the 11th and final stage of our desired outcome.

Query Evaluation.. (II)

Show the cubes which are removed:

Output:

```
mohitmehndiratta@x86_64-apple-darwin13 Self_Study % clingo practice.lp 0
clingo version 5.4.0
Reading from practice.lp
Solving...
Answer: 1
removed(y1) removed(y2)
SATISFIABLE

Models       : 1
Calls        : 1
Time         : 0.035s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.029s
mohitmehndiratta@x86_64-apple-darwin13 Self_Study %
```

From this, we can see that both the yellow cubes are removed.

Query Evaluation.. (III)

Show the cubes which are not removed:

Output:

```
mohitmehndiratta@x86_64-apple-darwin13 Self_Study % clingo practice.lp 0
clingo version 5.4.0
Reading from practice.lp
Solving...
Answer: 1
-removed(r1) -removed(b1) -removed(g1) -removed(b2) -removed(g2) -removed(r2) -removed(b3) -removed(r3) -removed(g3)
SATISFIABLE

Models       : 1
Calls        : 1
Time         : 0.033s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.026s
mohitmehndiratta@x86_64-apple-darwin13 Self_Study %
```

We can see that all cubes except yellow ones are not removed.

Thank You.