

Homework 2

Kaizhao Liang

1. a. **Here defined three DESTROY operators:**

Name & Variable: DESTROY1(?y)

Preconditions: Blk(?y), Blk(?x), Tbl(?z), Clr(?y), On(?y,?z), On(?x,?z), Diff(?x,?y)

Delete: Blk(?y), On(?y,?z), Clr(?y)

Add:

Name & Variable: DESTROY2(?y)

Preconditions: Blk(?y), Blk(?z), Clr(?y), On(?y,?z)

Delete: Blk(?y), On(?y,?z), Clr(?y)

Add: Clr(?z)

Name & Variable: DESTROY3(?y)

Preconditions: Blk(?y), Blk(?x), On(?x,?y), On(?y,?z)

Delete: Blk(?y), On(?y,?z), On(?x,?y)

Add: On(?x,?z)

Explanation:

DESTROY1 is instantiated when there is no block over block ?y and there exists a block ?x that is on the table. After ?y is destroyed, the table ?z is still not clear.

DESTROY2 is instantiated when there is no block over block ?y and block ?y is on another block ?z. After ?y is destroyed, the supporting block ?z is clear.

DESTROY3 is instantiated when there is a block ?x over block ?y. After ?y is destroyed, the ?x gently settles on ?z and it does not matter if ?z is the table or a block in this case.

There is a case which is not covered in these operators — the targeted block is the last block on table. However, if there is only one block on the table, there is no point doing anything to it because it's already in the correct order.

- b. Since DESTROY costs less than MOVE does, an admissible planner should always choose DESTROY over MOVE. It ends up destroying any block that is not in order in one pile. For example:

Problem: Blk(A), Blk(B), On(B,A), On(A,Tbl), Clr(B)

Solution: DESTROY(B)

If there are multiple sorted piles on the table, since DESTROY costs less than Move does, an admissible planner should always favour DESTROY, and end up destroying all the piles except the one with most

blocks. For example:

Problem: Blk(A), Blk(B, Blk(C), Blk(D), Blk(E), Blk(F), On(A, B), On(B, C), On(C, Tbl), Clr(A), On(D, E), On(E, Tbl), Clr(D), On(F, Tbl), Clr(F)

Solution: DESTROY(F), DESTROY(D), DESTROY(E)

- c. If it's only one block away from a goal state, while it requires at least two moves to reach one of the goal states, the admissible planner should destroy it. For example:

Problem: Blk(A), Blk(B), On(B, A), On(A, Tbl), Clr(B)

Solution: DESTROY(B)

If one block has to be moved to table and it's not the base of all the other blocks on the table, which means that it needs to be moved to block later or destroyed, the admissible planner may destroy it right away. For example:

Problem: Blk(A), Blk(B), Blk(C), Blk(D), On(A, D), On(D, Tbl), On(B, Tbl), On(C, Tbl), Clr(A), Clr(B), Clr(C)

Solution: DESTROY(A), MoveToBlock(C, D), MoveToBlock(B, C)

2.
 - a. Every human who drinks tea owns a cup.
 - b. There is a barber who only shaves those who do not shave themselves.
 - c. There is a barber who shaves himself and those who do not shave themselves.

3.
 - a. **Possible interpretation:** Some people do not like every kind of onions.

FLO statement: $\exists x \text{ Human}(x) \wedge \forall y [\text{Onion}(y) \Rightarrow \neg \text{likes}(x, y)]$

- b. **Possible interpretation:** Some cat would do something to cause certain kind of trouble

FLO statement: $\exists x \text{ Cat}(x) \wedge \exists z \text{ Trouble}(z) \wedge \exists y [\text{Does}(x, y) \wedge \text{Causes}(x, z)]$

- c. **Possible Interpretation:** there exists one road to the only hell in the world and the road is paved with some kinds of good intentions.

FLO statement: $\exists x \text{ Road}(x) \wedge \exists z [\text{Intention}(z) \wedge \text{Good}(z)] \wedge \forall y ([\text{Hell}(y) \wedge \text{To}(x, y)] \Rightarrow \text{Pave}(x, z))$

4.
 - a. i) MGU: $\{?z=?x, ?x=F(?y), ?u=?z, ?t=?y, ?v=?u, ?w=?v\}$
 ii) Unification: $G(F(?y), F(?y), F(?y), F(?y))$

- b. i)MGU: $\{?x=Bob, ?w=?x, ?y=?w, ?z=?y, ?u=?w, ?v=?w\}$
 ii)Unification: Parents(Bob,Bob,Bob,Bob)
- c. i)MGU: $\{?r=John, ?x=sister-of(?u), ?u=?v, ?v=?r\}$
 ii)Unification: Knows(Sister-of(John),William,Sister-of(John),John)
- d. No Unification. Dog and Likes can never be matched.