CS 440 - Homework 2

Matthew Liu

1. Suppose we could destroy any block in the otherwise traditional blocks world. The destruction is gentle with any supported blocks settling down on the support under the destroyed block.

a) Operators defined below

DestroyBlkUnderBlk(?x)

- Preconditions: Blk(?x), Blk(?y), On(?y,?x), On(?x,?z)
- Delete: Blk(?x), On(?y,?x), On(?x,?z)
- Add: On(?y, ?z)

DestroyClrBlkOnTable(?x)

- Preconditions: Blk(?x), Clr(?x), On(?x,?y), Tbl(?y)
- Delete: Blk(?x), Clr(?x), On(?x,?y)
- Add: Nothing

DestroyClrBlkOnBlock(?x)

- Preconditions: Blk(?x), Clr(?x), On(?x,?y), Blk(?y)
- Delete: Blk(?x), Clr(?x), On(?x,?y)
- Add: Clr(?y)

b) Examples listed below

Example with Destroy: Moving a block on a block on the table to the table. Using destroy would cost of 2, whereas MoveToTable would cost 3.

- Initial State: Blk(?x), Blk(?y), Tbl(?z), On(?x,?y), On(?y,?z)
- Actions: Destroy(?y)
- Goal State: Blk(?x), Tbl(?z), On(?x,?z)

Example without Destroy: Moving a block on the top of a stack of 3 blocks to the table. Using destroy would cost of $3x2 = 6$, whereas MoveToTable just costs 3.

- Initial State: Blk(?a), Blk(?b), Blk(?c), Blk(?d), Tbl(?e), On(?a,?b), On(?b,?c), On(?c,?d), On(?d,?e)
- Actions: MoveToTable(?a)
- Goal State: Blk(?a), Blk(?b), Blk(?c), Blk(?d), Tbl(?e), On(?a,?e)

c) Examples listed below

Example with Destroy: Moving a block with 2 blocks on top and the block itself is on a block on the table to the table. Using destroy would evoke a cost of 4, whereas MoveToTable then MoveToBlock would cost $3 + 2 = 5$.

- Initial State: Blk(?a), Blk(?b), Blk(?c), Blk(?d), Tbl(?e), On(?a,?b), On(?b,?c), On(?c,?d), On(?d,?e)
- Actions: Destroy(?d)
- Goal State: Blk(?a), Blk(?b), Blk(?c), Tbl(?e), On(?c,?e), On(?b, ?a)

Example without Destroy: Moving a block on a block on the table to the table. Using destroy would cost of 4, whereas MoveToTable would cost 3.

- Initial State: Blk(?x), Blk(?y), Tbl(?z), On(?x,?y), On(?y,?z)
- Actions: MoveToTable(?y)
- Goal State: Blk(?x), Tbl(?z), On(?x,?z)

2. Give the most natural English sentence you can for the following first-order logic sentences. If it is ambiguous, give all meanings. If it is not possible to translate into English, explain why.

a) Every human that drinks tea owns a cup.

b) Ambigious as the variable y is not defined, thus making it impossible to translate. It could be a human, in which case the statement is saying that for all people who can't shave themselves there exists a barber who shaves them and if there exists a barber who shaves a person, then that person cannot shave themselves.

c) Similar to b, y could be a human, in which case for all people who cannot shave themselves there exists a barber who will shave them, and the barber shaves himself or herself.

3. Give a first-order logic sentence that best captures the meaning of the following English sentences. If the meaning of a predicate is not obvious, be sure to explain your intended semantics.

a) Not everyone likes onions

$$\exists x(\text{Human}(x) \wedge \forall y(\text{Onion}(y) \wedge \neg\text{Like}(x, y)))$$

b) Some cats would do anything to cause trouble

$$\exists x(\text{Cat}(x) \wedge \forall y(\text{Action}(y) \Rightarrow \exists z(\text{Trouble}(z) \wedge \text{Cause}(z))))$$

c) The road to hell is paved with good intentions

$$\exists x(\text{RoadToHell}(x) \wedge \exists y(\text{GoodIntentions}(y) \Rightarrow \text{Paved}(x, y)))$$

4. Give the most general unifier and a corresponding unification instance for each set of expressions below. If unification is not possible, write "No Unification" and briefly explain why.

a) No Unification. There are too many circular dependencies, always leading to the repeat definition of a variable (as in a variable shows up on the left side of equal multiple times).

b) Unification below

- MGU: $\{?z = ?y, ?y = ?w, ?w = ?v, ?v = ?x, ?u = ?x, ?x = \text{Bob}\}$
- Unification Instance: $\text{Parents}(Bob, Bob, Bob, Bob)$

c) No Unification, Same problem as a, but one step further. Lets assume for one second that the MGU $\{?x = \text{Sister}(?u), \text{Sister}(?v) = ?x, \text{Sister}(?u) = \text{Sister}(?v), ?v = ?uSon(?v) = Son(?r), ?v = ?r, ?r = \text{John}\}$, This leads to part of a unification instance ($?x = \text{Sister-of(John)}$). This however is impossible, as the questions notes that John has no sisters at all

d) No Unification. Dogs and Like are different functions, hence they can not be equated.