Knowledge Representation

a. Why is Knowledge Representation important?

Knowledge Representation determines how domains can be modeled.

b. How does it affect computation?

The choice of knowledge representation determines how domains can be modeled, which influences how problems can be solved. For example if a problem is modeled in propositional logic, this may require more sentences than using FOL and therefore more time for inference.

c. What are 3 types of Knowledge Representation systems/languages discussed in class?

Choose any 3 (propositional logic, FOL, prolog, etc)

d. What are their advantages/disadvantages?

e. Match the keyword with the letter of its definition below:

1. Frame problem:___H___
2. Ramification problem:___C___
3. Qualification problem:___J___
4. Fluent:___G___
5. Atemporal predicate:__D____
6. Situation calculus:___F___
7. Ontology:__A____
8. Subclass:___I___
9. Inheritance:__B____
10. Semantic network:___E___

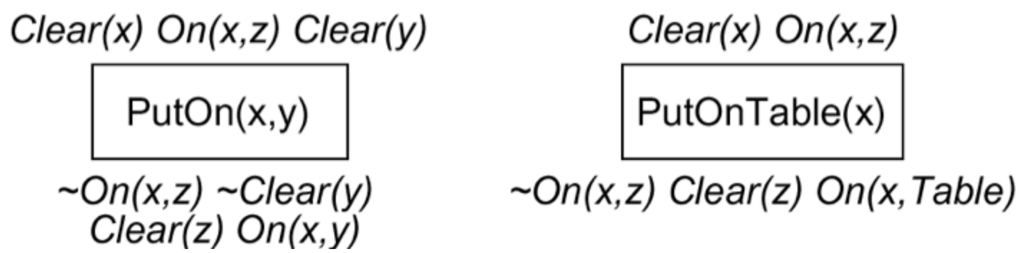
Definitions:

A. A rigorous and exhaustive organization of some knowledge domain that is usually hierarchical and contains all the relevant entities and their relations
B. A form of inference that determines if an object is a member of a class, and all members of the class share some property, then the object will have that property
C. Handling secondary (implicit) effects of actions
D. Predicates that never change over time
E. Graphical network representation of knowledge
F. Approach to modeling actions in logic for use in problem solving and planning
G. Predicates that change over time
H. Representing all things that stay the same from one situation to the next
I. A subdivision of a set or class.

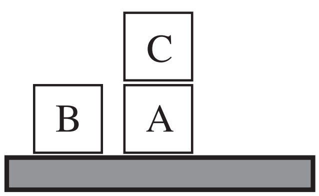
J. Defining the circumstances under which an action is guaranteed to work

Planning

Given these actions and predicates for the block world domain, answer the questions below.



a. Describe the start state given in the diagram below using the given predicates above:



Start State

See lecture 16 slides

b. The goal state is On(A,B)^On(B,C). Draw the block world diagram for the goal state.

c. Draw the partial order plan for solving this block world problem:

d. In what order will a partial order planner **choose** the actions to add to the plan? (You may label your drawing above with the order in which the actions were added to the plan)

"2 jugs problem" as planning problem
You have a well, and two water jugs of size 3 gallons and 5 gallons. The goal is to get 4 gallons in one of the jugs. In doing so, you are allowed to:

Fill up either jug completely Empty either jug completely.

Pour water from one jug to the other, until the poured jug is empty, or the other jug is full.

Specify

a. The initial state

00

b. The operators. For each operator, give the preconditions and effects.

(x,y) is the state (x,y) to (3,y) (x,y) to (x,5) (x,y) to (0,y) (x,y) to (x,0) (x,y) to (0,x+y) if (x+y)<5 (x,y) to (x-5+y, 5) if (x+y)>5 (x,y) to (x+y, 0) if (x+y)<3 (x,y) to (3, x+y-3) if (x+y)>3

c. what is the correct plan?

1) Fill up Jug A with 5 litres and Fill jug B from A - there are 2 litres left in Jug A.

2) Empty Jug B and then tip the remaining 2 litres from Jug A into Jug B.

3) Re-Fill Jug A with 5 litres and then fill Jug B again from it. As Jug B already has 2 litres in it,

only 1 litre is taken out of Jug A. i.e. Jug A now has 4 litres left in it.

Logical Reasoning

a. Circle the sentence that is the CNF form of this sentence $A \Leftrightarrow (B \vee E)$

1) $(\neg A \vee B \vee E) \wedge (B \vee A) \wedge (E \vee A)$
2) $(\neg A \vee B \vee E) \wedge (\neg B \vee A) \wedge (\neg E \vee A)$
3) $(\neg A \wedge B \wedge E) \vee (\neg B \wedge A) \vee (\neg E \wedge A)$
4) $(B \vee A) \wedge (E \vee A)$
5) $(B \wedge A) \wedge (E \wedge A)$

b. To unify α: <(x, +(y, x)), and β: <(10, +(a, b)), circle the most general unifier θ that makes α and β identical:

1) $\theta = \{x/a, y/b\}$
2) $\theta = \{x/10, y/a, b/10\}$
3) $\theta = \{x/10, y/b, a/10\}$
4) $\theta = \{x/a, y/b, b/10\}$
5) $\theta = \{x/10, y/a, a/b\}$

c. Translate each sentence to first order logic, using predicate Smart(x), At(x,USC), Loves(x,y):

1) Everyone at USC is smart

$\forall x\ At(x,USC) \wedge Person(x) \Rightarrow Smart(x)$

2) Someone at USC is smart

$\exists x\ At(x, USC) \wedge Person(x) \wedge Smart(x)$

3) There is someone who loves everyone

$\exists x\ \forall y\ Loves(x,y)$

4) Everyone is loved by someone

$\forall y\ \exists x\ Loves(x,y)$

d. Given the Prolog program (where tweety is a constant, X a variable, and NOT is Prolog's standard form of negation):

bird(tweety).
fly(X): bird(X), NOT ostrich(X).

Is the fact that tweety flies inferred? Why or why not?

Yes, Prolog will find Fly(Tweety) is true.

- bird(X) will unify with bird(tweety) X/tweety
- ostrich(tweety) will fail because it is not found in the kb
- so NOT ostrich(tweety) is true
- then fly(tweety) will be true.

e. What is the most general unifier of [a, [a], [A, b]] and [c, [d], [a, B]], where a, b, c and d are variables and A and B are constants? You need not show your work, just the answer.

a/A, c/A, d/A, b/B