Homework 4

(due - 23:59 April 28, 2022)

Please submit your homework as a PDF file.

Questions:

[1] Let's assume that "Person a loves person b" can be expressed as "Love(a,b)" and "Person a hates person b" can be expressed as "Hate(a,b)". Translate the following sentences into first-order logic, and then convert them to clause form. (20 points)

a. Sarah loves anyone whom Lia loves. (5 points)
   A. ∀x (Love(Lia, x) => Love(Sarah, x))
      i. ∀x (¬Love(Lia, x) ∨ Love(Sarah, x))
      ii. ¬Love(Lia, x) ∨ Love(Sarah, x)
b. If someone loves Sarah, then David loves Sarah (5 points)
   A. ∀x (Love(x, Sarah) => Love(David, Sarah))
      i. ∀x (¬Love(x, Sarah) ∨ Love(David, Sarah))
      ii. ¬Love(x, Sarah) ∨ Love(David, Sarah)
c. Someone who loves all sports doesn't love all animals. (5 points)
   A. ∃x (Love(x, all sports) ∧ ¬Love(x, all animals))
      i. ∃x (Love(x, all sports) ∧ ¬Love(x, all animals))
      ii. Love(f(x), all sports) ∧ ¬Love(f(x), all animals)
      iii. Love(f(x1), all sports), ¬Love(f(x2), all animals)
d. Everyone who loves Sarah hates everyone who Sarah loves. (5 points)
   A. ∀x(Love(Sarah, x) => ∀y(Love(y, Sarah) => Hate(y, x)))
      i. ∀x(¬Love(Sarah, x) ∨ ∀y(¬Love(y, Sarah) ∨ Hate(y, x)))
      ii. ∀x∀y (¬Love(Sarah, x) ∨ (¬Love(y, Sarah) ∨ Hate(y, x)))
      iii. ¬Love(Sarah, x) ∨ (¬Love(y, Sarah) ∨ Hate(y, x))
      iv. ¬Love(Sarah, x) ∨ ¬Love(y, Sarah) ∨ Hate(y, x)

[2] Translate the following description logic expression into first-order logic, and comment on the result: [20 points]

And(Man, AtLeast(3, Son), AtMost(2, Daughter),

All(Son, And(Unemployed, Married, All(Spouse, Doctor))),
All(Daughter , And(Professor ,Fills(Department,Physics, Math)))) .

Man(x) ∧ LeastOf(x, 3, Sons) ∧ MostOf(x, 2, Daughters)

∧ (SonOf(y, x) => (Unemployed(y) ∧ MarriedWith(y,z) ∧ IsDoctor(z)))

∧ (DaughterOf(z, x) => (IsProfessor(z) ∧ (InMathDepartment(z) ∨ InPhysicsDepartment(z))))

Comment:
LeastOf(a, b, c): a has c, and c's number is at least b.
MostOf(a, b, c): a has c, and c's number is at most b.
FOL is easier to read.

[3] The monkey-and-bananas problem is faced by a monkey in a laboratory with some bananas hanging out of reach from the ceiling. A box is available that will enable the monkey to reach the bananas if he climbs on it. Initially, the monkey is at *A*, the bananas at *B*, and the box at *C*. The monkey and box have height *Low*, but if the monkey climbs onto the box he will have height *High*, the same as the bananas. The actions available to the monkey include *Go* from one place to another, *Push* an object from one place to another, *ClimbUp* onto or *ClimbDown* from an object, and *Grasp* or *Ungrasp* an object. The result of a *Grasp* is that the monkey holds the object if the monkey and object are in the same place at the same height. [30 points]

a. Write down the initial state description. [5 points]
   A. At(Monkey, A) ∧ At(Bananas, B) ∧ At(Box, C) ∧ Height(Monkey, Low) ∧ Height(Box, Low) ∧ Height(Bananas, High) ∧ Pushable(Box) ∧Climbable(Box)
b. Write the six action schemas. [10 points]
   A. Go(x, y)
      i. Precondition: At(Monkey, x)
      ii. Effect: At(Monkey, y), ¬At(Monkey, x)
   B. Push(b, x, y)
      i. Precondition: At(Monkey, x), At(b, x), Pushable(b)
      ii. Effect: At(b, y), At(Monkey, y), ¬At(b, x), ¬At(Monkey, x)
   C. ClimbUp(b)
      i. Precondition: At(Monkey, x), At(b, x), Climbable(b)
      ii. Effect: On(Monkey, b), ¬Height(Monkey, Low), Height(Monkey, High)
   D. ClimbDown(b)

   i. Precondition: On(Monkey, b), Height(Monkey, High)
   ii. Effect: ¬On(Monkey, b), ¬Height(Monkey, High), Height(Monkey, Low)

E. Grasp(o)
   i. Precondition: Height(Monkey, High), Height(o, High), At(Monkey, x), At(o, x)
   ii. Effect: Have(Monkey, o)

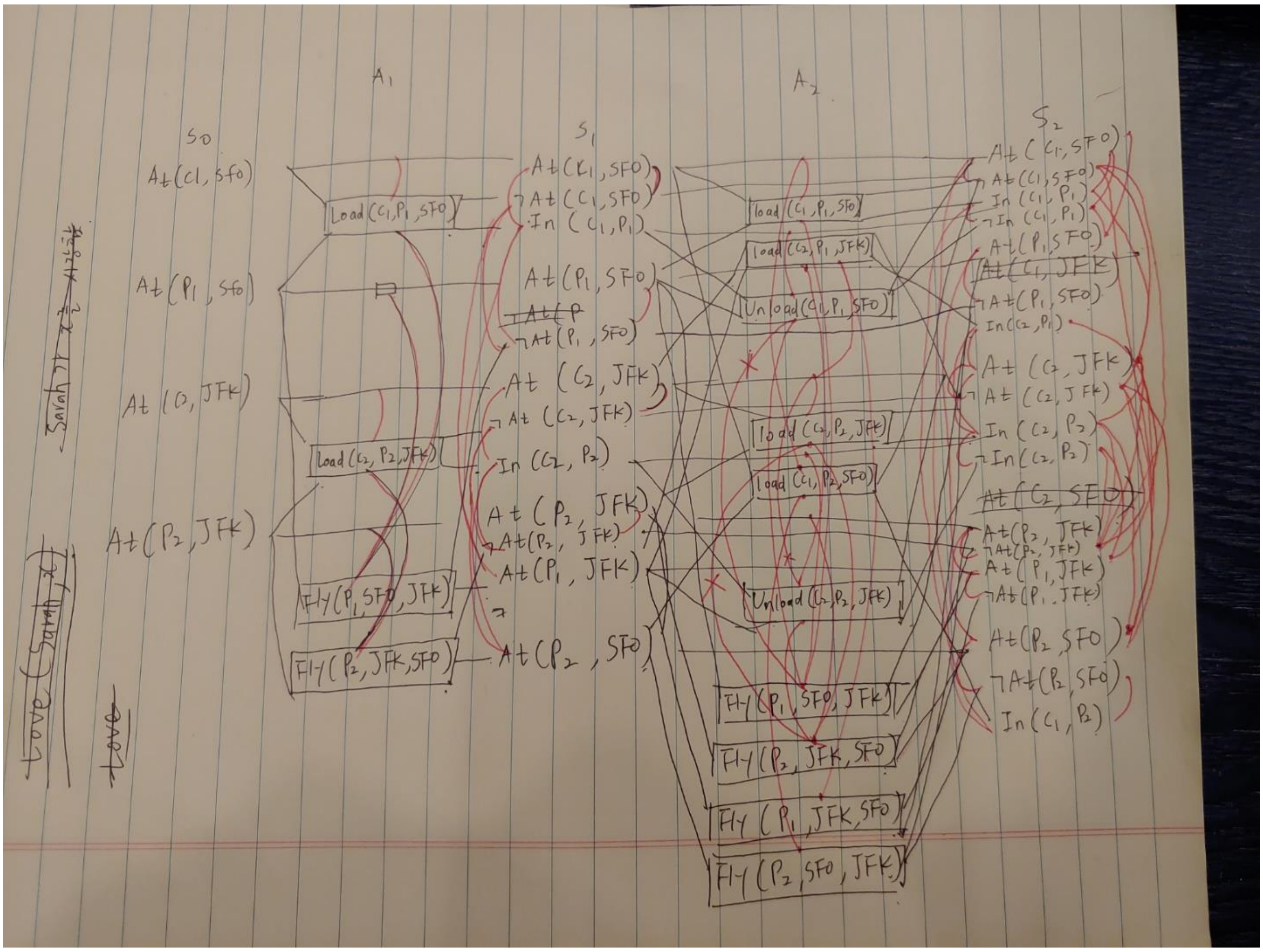
F. Ungrasp(o)
   i. Precondition: Have(Monkey, o)
   ii. Effect: ¬Have(Monkey, o)

c. Suppose the monkey wants to fool the scientists, who are off to tea, by grabbing the bananas, but leaving the box in its original place. Write this as a general goal (i.e., not assuming that the box is necessarily at *C*) in the language of situation calculus. Can this goal be solved by a classical planning system? [5 points]
   A. Have(Monkey, Bananas, s) ∧ (∃x At(Box, x, s0) ∧ At(Box, x, s))
   B. In classical planning system, we could only say about goal state. Therefore, we cannot solve this which have the relationship between two states.

d. Your schema for pushing is probably incorrect, because if the object is too heavy, its position will remain the same when the *Push* schema is applied. Fix your action schema to account for heavy objects. [10 points]
   A. Push(b, x, y)
      i. Precondition: At(Monkey, x), At(b, x), Pushable(b), ¬Heavy(b)
      ii. Effect: At(b, y), At(Monkey, y), ¬At(b, x), ¬At(Monkey, x)

[4] Construct levels 0, 1, and 2 of the planning graph for the air cargo transportation planning problem given by the PDDL description below: [30 points]

$$
\begin{aligned}
&Init(At(C_1,\ SFO) \wedge At(C_2,\ JFK) \wedge At(P_1,\ SFO) \wedge At(P_2,\ JFK)\\
&\quad \wedge Cargo(C_1) \wedge Cargo(C_2) \wedge Plane(P_1) \wedge Plane(P_2)\\
&\quad \wedge Airport(JFK) \wedge Airport(SFO))\\
&Goal(At(C_1,\ JFK) \wedge At(C_2,\ SFO))\\
&Action(Load(c,\ p,\ a),\\
&\quad \text{PRECOND: } At(c,\ a) \wedge At(p,\ a) \wedge Cargo(c) \wedge Plane(p) \wedge Airport(a)\\
&\quad \text{EFFECT: } \neg\, At(c,\ a) \wedge In(c,\ p))\\
&Action(Unload(c,\ p,\ a),\\
&\quad \text{PRECOND: } In(c,\ p) \wedge At(p,\ a) \wedge Cargo(c) \wedge Plane(p) \wedge Airport(a)\\
&\quad \text{EFFECT: } At(c,\ a) \wedge \neg\, In(c,\ p))\\
&Action(Fly(p,\ from,\ to),\\
&\quad \text{PRECOND: } At(p,\ from) \wedge Plane(p) \wedge Airport(from) \wedge Airport(to)\\
&\quad \text{EFFECT: } \neg\, At(p,\ from) \wedge At(p,\ to))
\end{aligned}
$$



(Posted on 2022/04/15)