CSCI 561

Foundations of Artificial Intelligence

Exam #2 – Spring 2014

4:00-5:20pm

Friday, March 28, 2014

Instructor: Prof Tejada

(This exam is closed book, closed notes, closed everything.
No "cheat sheet" allowed.
No calculators, cell phones, or any electronic gadgets.)

Notes. As grading criteria, you will get full marks for each question that has the right answer (with appropriate reasoning or show of steps). You may be assigned a partial credit for question if deemed appropriate. If you face difficulty, or if things are unclear, write down your assumptions and proceed accordingly.

Logical Reasoning (50pts)

Convert from English to FOL (20pts)

Circle True or False. For sentences in English make your judgment of the meaning of the sentence, i.e., you may want to translate it in FOL to conclude.

1. [2pts] [True/False] "Bert and Ernie are brothers" is equivalent to "Bert is a brother and Ernie is a brother"

2. [2pts] [True/False] “ p and q are not both true” is equivalent to “ p and q are both not true”

3. [2pts] [True/False] “Neither p nor q” is equivalent to “both p and q are false”

4. [2pts] [True/False] “Not all A’s are B’s” is equivalent to “$\exists x\ (A(x) \land \Box B(x))$ ”

5. [2pts] [True/False] "MS students and PhD students are welcome to apply." is equivalent to “$\forall x\ [(M(x) \land P(x)) \Rightarrow Apply(x)]$”

Convert from English to FOL continued

Circle True or False. For sentences in English make your judgment of the meaning of the sentence, i.e., you may want to translate it in FOL to conclude.

Questions 6 to 9: Attract is a relation from x to y, i.e., A(x,y) says that x attracts y.

6. [2pts] [True/False] “Everything attracts something”, where “something” means “something or other”, is equivalent to “$\forall x \; \exists y \; A(x, y)$ ”

7. [2pts] [True/False] “Something is attracted by everything”, where “something” means “something in particular”, is equivalent to “$\exists y \; \exists x \; A(x, y)$ ”

8. [3pts] [True/False] “Everything is attracted by something” ”, where “something” means “something or other”, is equivalent to “$\exists x \; \forall y \; A(x, y)$ ”

9. [3pts] [True/False] “Something attracts everything”, where “something” means “something in particular”, is equivalent to “$\exists x \; \exists y \; A(x, y)$ ”

Resolution (20pts)
From the sentence "Heads I win, tails you lose," prove using resolution that "**Iwin**."

1. [10 pts] First build the KB, from the sentence "Heads I win, tails you lose," using the true or false variables **Heads, Tails, IWin, YouLose** and write the sentence in terms of disjunctions clauses. Add to KB the general knowledge that the outcome of a coin toss must be **Head** or **Tails** and the general knowledge that if **YouLose** then **IWin** and, if **IWin** then **YouLose**.

"Heads I win, tails you lose."

+2 (Heads => IWin) or in CNF (¬ Heads ∨ IWin)

+2 (Tails => YouLose) or in CNF (¬ Tails ∨ YouLose)

Add some general knowledge axioms about coins, winning, and losing:

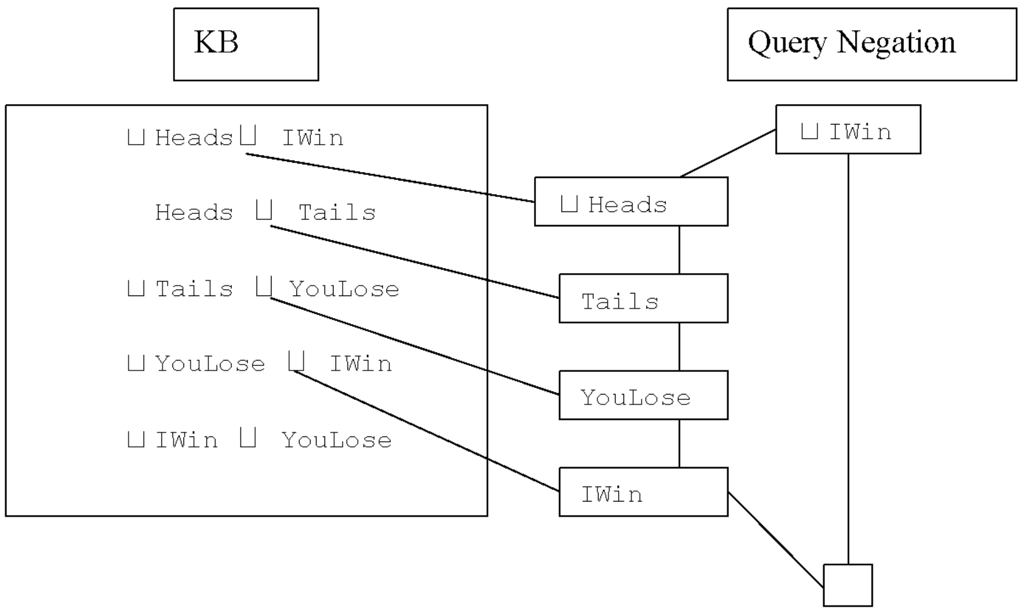
+2 (Heads v Tails)

+2 (YouLose => IWin) or in CNF (¬ YouLose ∨ IWin)

+2 (IWin => YouLose) or in CNF (¬ IWin ∨ YouLose)

Resolution continued

2. [10 points] Prove the Goal sentence: "**IWin**" via the Resolution method.



+1: adding ¬ IWin
+2: inferring ¬ Heads in a correct way
+2: inferring Tails in a correct way
+2: inferring YouLose in a correct way
+2: inferring IWin in a correct way
+1: inferring Ø in a correct way

Unification and Prolog (10pts)
Answer the questions below given these two rules in Prolog for appending two lists to produce a third:

append([],Y,Y).
append([X|L],Y,[X|Z]) :- append(L,Y,Z).

1. [4pts] Given the query append(A, B, [[1], [2, [3]]]) list all possible □ for A and B:

A	B	
[]	[[1], [2, [3]]]	+1
[[1]]	[[2, [3]]]	+2
[[1], [2, [3]]]	[]	+1

* No partial points. Must be exactly matched.

* -1: additional wrong pair

Prolog screenshot

```
| ?- append(A,B,[[1],[2,[3]]]).

A = []
B = [[1],[2,[3]]] ? ;

A = [[1]]
B = [[2,[3]]] ? ;

A = [[1],[2,[3]]]
B = []
```

Unification and Prolog continued

Answer the questions below given these two rules in Prolog for appending two lists to produce a third:

append([],Y,Y).
append([X|L],Y,[X|Z]) :- append(L,Y,Z).

To get points for this question, your answer should work in Prolog without any warning message.

2. [6pts] When L=[A], Y=[B, C], X=[[D]] then what are the values for:
   a. [3pts] Z= [A, B, C]

      * No partial points. Must be exactly matched.

      * -1: additional wrong answer

      Prolog screenshot

```
| ?- append([A],[B,C],[A,B,C]).

yes
| ?-
```

   b. [3pts] [X|Z]= [[[D]] , A, B, C] or [[[D]] | [A,B,C]]

      * No partial points. Must be exactly matched.

      * -1: additional wrong answer

      Prolog screenshot

```
| ?- append([[[D]]|[A]],[B,C],[[[D]],A,B,C]).

yes
| ?-
```

Knowledge Representation (20pts)

Circle the letter that corresponds to the best answer for the question:

1. [4pts] This is a form of sound inference:
    a. Resolution
    b. Inheritance
    c. Generalized modus ponens
    d. All of the above
    e. None of the above

2. [4pts] The closed world assumption helps to address this problem:
    a. Frame problem
    b. Ramification problem
    c. Qualification problem
    d. All of the above
    e. None of the above

3. [4pts] This is a type of ontology:
    a. Semantic Network
    b. WordNet
    c. Facebook
    d. All of the above
    e. None of the above

4. [4pts] This is **not** part of Situation Calculus:
    a. Atemporal predicate
    b. Fluent
    c. Frame axiom
    d. All of the above
    e. None of the above

5. [4pts] A PartOf hierarchy allows this type of reasoning:
    a. Individual vs stuff
    b. Inheritance
    c. Transitivity
    d. All of the above
    e. None of the above

Planning (30pts)

Planning representation (10pts)

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.

a. [4pts] Write down the initial state description.

+1 At(Monkey,A) ∧ At(Bananas,B) ∧ At(Box,C) ∧
+2 Height(Monkey,Low) ∧ Height(Box,Low) ∧ Height(Bananas,High) ∧
+1 Other preconditions assumed in the action schemas such as Pushable(Box) ∧ Climbable(Box)

b. [6pts] Write the six action schemas with preconditions and effects (you must define).)

[one point for each of the actions only if its entirely correct and leads to the solution considering all other actions]

+1 Action(ACTION:Go(x,y), PRECOND:At(Monkey,x), EFFECT:At(Monkey,y) ∧¬(At(Monkey,x)))

+1 Action(ACTION:Push(b,x,y), PRECOND:At(Monkey,x) ∧ Pushable(b), EFFECT:At(b,y) ∧ At(Monkey,y) ∧¬At(b,x) ∧¬At(Monkey,x))

+1 Action(ACTION:ClimbUp(b), PRECOND:At(Monkey,x) ∧At(b,x) ∧ Climbable(b), EFFECT:On(Monkey,b) ∧Height(Monkey,High))

+1 Action(ACTION:Grasp(b), PRECOND:Height(Monkey,h) ∧Height(b,h) ∧At(Monkey,x) ∧ At(b,x), EFFECT:Have(Monkey,b))

+1 Action(ACTION:ClimbDown(b), PRECOND:On(Monkey,b) ∧ Height(Monkey,High), EFFECT:¬On(Monkey,b) ∧¬Height(Monkey,High) ∧ Height(Monkey,Low)

+1 Action(ACTION:UnGrasp(b), PRECOND:Have(Monkey,b), EFFECT:¬Have(Monkey,b))

Partial order planning (20pts)

The problem is to plan the transportation of the bird Tweety from the Lincoln Park Zoo in Chicago using the O'hare airport (ORD) to the Bronx Zoo in New York using the JFK airport and the transportation of apples from Los Angeles to New York. Use the PDDL actions given below to create the plan:

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

Figure 10.1 A PDDL description of an air cargo transportation planning problem.

1.[4pts] Given the initial and goal states of this problem as:

Initial State: *At(Tweety, ORD), At(apples, LAX), At(P1, LAX), Cargo(Tweety), Cargo(apples), Plane(P1), Airport(LAX), Airport(JFK), Airport(ORD)*
Goal State: *At(Tweety, JFK), At(apples,JFK)*

Write an efficient and consistent total order plan that starts at the initial state and achieves the goal state.

Load apples +2 efficient total order plan (no extra actions)

Fly LAX ORD +2 consistent total order plan

Load Tweety

Fly ORD JFK

Unload Tweety

Unload apples

Partial order planning continued
2. [4pts] Draw the first action that will be inserted into the initial plan by a partial order planner (POP) shown below. Include solid arrows to represent causal links between action effects and preconditions or dotted arrows to represent ordering constraints.

START

At(Tweety, ORD), At(P1, LAX), At(apples, LAX), Cargo(Tweety), Cargo(Apples) Plane(P1), Airport(LAX), Airport(JFK), Airport(ORD)

+2 for 1 correct action
+2 for 1 correct causal link to satisfy an open goal
-1 for extra/incorrect actions
-1 for extra/incorrect arrows

Unload Tweety *or* *Unload apples*

At(Tweety, JFK), At(apples, JFK)

Finish

Partial order planning continued
3. [4pts] Draw the first and second actions that will be inserted into the initial plan by a partial order planner (POP) shown below. Include solid arrows to represent causal links between action effects and preconditions or dotted arrows to represent ordering constraints.

START

At(Tweety, ORD), At(P1, LAX), At(apples, LAX), Cargo(Tweety), Cargo(Apples) Plane(P1), Airport(LAX), Airport(JFK), Airport(ORD)

+2 for correct Action 1 and correct causal link to satisfy an open goal
+2 for correct Action 2 and correct causal link to satisfy an open goal
-1 for extra/incorrect actions
-1 for extra/incorrect arrows

ACTION 2 = Fly P1 or Load Tweety or Load apples (Unload Tweety/Unload apples)

ACTION 1 = Unload Tweety or Unload apples

At(Tweety, JFK), At(apples,JFK)

Finish

Partial order planning continued

4. [4pts] Draw the first, second, and third actions that will be inserted into the initial plan by a partial order planner (POP) shown below. Include solid arrows to represent causal links between action effects and preconditions or dotted arrows to represent ordering constraints.

START

At(Tweety, ORD), At(P1, LAX), At(apples, LAX), Cargo(Tweety), Cargo(Apples) Plane(P1), Airport(LAX), Airport(JFK), Airport(ORD)

+2 for correct Action 1 & 2 and correct causal link to satisfy an open goal
+2 for correct Action 3 w/correct causal link to an open goal /ordering constraint
-1 for extra/incorrect actions
-1 for extra/incorrect arrows

ACTION 3 = Fly P1 or Load Tweety or Load apples (Unload Tweety/Unload apples)

ACTION 2 = Fly P1 or Load Tweety or Load apples (Unload Tweety/Unload apples)

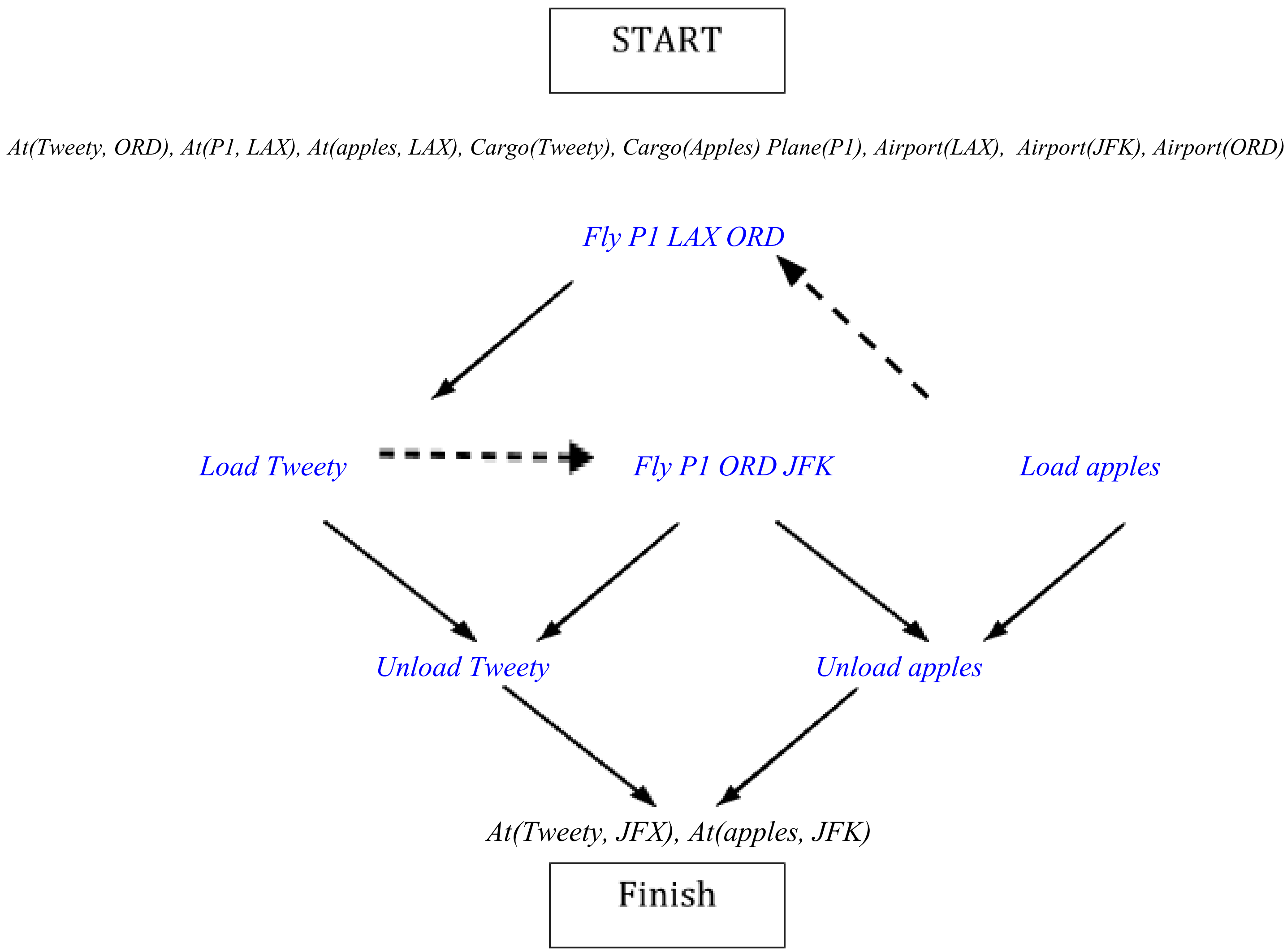
ACTION 1 = Unload Tweety or Unload apples

At(Tweety, JFK), At(apples, JFK)

Finish

Partial order planning continued

5. [4pts] Draw all actions that will be inserted into the initial plan by a partial order planner (POP) shown below. Include solid arrows to represent causal links between action effects and preconditions or dotted arrows to represent ordering constraints.



+1 for correct Action 1-3 and correct causal links to satisfy an open goal
+3 for correct Action 4=6 w/correct causal link to an open goal/ordering constraints
-1 for extra/incorrect actions
-1 for extra/incorrect arrows