1. General AI Knowledge and Application.

[8%] True or False: For each of the statements below, fill in the bubble T if the statement is always and unconditionally true, or fill in the bubble F if it is always false, sometimes false, or just does not make sense.

1). [1%] $\forall x \exists y\ Loves(x, y)$ is the translation of the statement "There is someone who is loved by everyone". F

2). [1%] Every Horn clause is a Definite clause. F (reverse is true)

3). [1%] KB entails α if and only if KB=> α. T

4). [1%] Skolemization for $\exists x \forall y P(x) \vee Q(y)$ is $P(x) \vee Q(f(x))$, where x and y are variables. F, (it is P(X1) ∨ $Q(y)$)

5). [1%] Forward chaining is sound and complete. T (textbook pp.258)

6). [1%] Backward chaining is a form of **goal-directed reasoning** T(textbook pp.258)

7). [1%] Married (Father (x), Mother (y)) is atomic sentence. F

8). [1%] Planning graphs work only for propositional planning problems—ones with no variables. T(textbook pp.379)

9) [1%]Backward chaining is complete. F

10)[1%] Generalized Modus Ponens is a sound inference rule. T pp.326

11)[1%] We could unify K(John,x) and P(x,x) F

Multiple Choice [12%]: Each question has one or more correct choices. Check the boxes of all correct choices and leave the boxes of wrong choices blank.

9) [4%] Which of the following entailments are true?

1) ☒ $(M \wedge P) \Rightarrow S \vdash (M \Rightarrow S) \vee (P \Rightarrow S)$ T
2) ☒ $False \vdash True$ T
3) ☒ $(M \wedge P) \vdash (M \Leftrightarrow P)$ T
4) ☒ $(M \vee P) \wedge (\neg R \vee \neg S \vee D) \vdash (M \vee P)$ T

10) [4%] We have the following predicates In(x,y), Likes(x,y), and State(x). and English sentence**: No person in State A likes any person in State B**. Which of the following logical expressions correctly expresses the English sentence?

1) ☒ ¬[∃c,d In(c,A)∧In(d,B)∧Likes(c,d) , T
2) ☒ ∀c,d [In(c,A)∧In(d,B)] ⇒ ¬Likes(c,d)]. T
3) ☐ ¬∀c In(c,A) ⇒ ∃d In(d,B)∧¬Likes(c,d). F, says there is someone in A that likes everyone in state B,
4) ☒ ∀c In(c,A) ⇒ (∀d In(d,B) ⇒ ¬Likes(c,d)). T

11) [4%] Which of the following pair of atomic sentences have a unifier:

1) ☒ **P(A,B,B), P(x,y,z).** T
2) ☐ **Q(y,G(A,B)),Q(G(x,x),y).** F
3) ☒ **Older(Father(y),y),Older(Father(x),John).** T
4) ☐ **Knows(Father(y),y), Knows(x,x).** F, occurs check prevents

12) [4%] Which of the following is true about Prolog:

1) ☒ Prolog is incomplete T
2) ☐ It has occur check F
3) ☐ Prolog is sound F
4) ☐ It has check for infinite recursion. F

2 First-Order Logic

Unless stated otherwise, give no partial credit. Variable names and order of terms can differ.

$x \neq y$ can be written as $\neg(x = y)$.

2A. Anyone can only be a cook or a chef.

$\forall x \left(IsCook(x) \land \neg IsChef(x)\right) \lor \left(\neg IsCook(x) \land IsChef(x)\right)$

Parentheses for grouping expressions are optional because $\land$ precedes $\lor$.

2B. There is no cook without any assigned work.

$\neg\exists x\ IsCook(x) \land \neg\ \exists y\ AssignsWork(y, x)$

Or

$\neg\exists x\ IsCook(x) \land \forall y\ \neg AssignsWork(y, x)$

Or

$\forall x\ IsCook(x) \rightarrow \exists y\ AssignsWork(y, x)$

Or

$\forall x\ \neg IsCook(x) \lor \exists y\ AssignsWork(y, x)$

2C. Any chef cooks better than any cooks.

$\forall x\ IsChef(x) \rightarrow \left(\forall y\ IsCook(y) \rightarrow CooksBetter(x, y)\right)$

Or

$\forall x, y\ IsChef(x) \land IsCook(y) \rightarrow CooksBetter(x, y)$

Accept solutions where $A \rightarrow B$ is written as $\neg A \lor B$.

2D. There is exactly one cook who assigns work to all other cooks.

$\exists!\, x\ IsCook(x) \land \left(\forall y\ IsCook(y) \land x \neq y \rightarrow AssignsWork(x, y)\right)$

Or

$\exists x\ IsCook(x) \land \left(\forall y\ IsCook(y) \land x \neq y \rightarrow AssignsWork(x, y)\right)$

$\qquad \land \left(\forall y\ IsCook(y) \land \left(\forall z\ \ IsCook(z) \land y \neq z \rightarrow AssignsWork(y, z)\right) \rightarrow x = y\right)$

Accept solutions where $A \rightarrow B$ is written as $\neg A \lor B$.

If all else is correct but $x \neq y$ is missing, give 1 %.

3 Inference in First Order Logic

3A. Forward Chaining

The order does not matter. Subtract 1% for every unifier {…} that is missing or incorrect. Rules can be written out **or** just referred to by their number. Subtract 1% for every rule that is missing.

$\boldsymbol{MoreFountains(x,y) \Rightarrow NicerCampus(x,y)}$ **unifies with:**

Rule 6 $\mathrm{MoreFountains(USC, UCLA)}$

Rule 7 $\mathrm{MoreFountains(USC, UCLA) \Rightarrow NicerCampus(USC, UCLA)}$ {x/USC, y/UCLA}

$\boldsymbol{AvgGPA(x, xGPA) \wedge AvgGPA(y, yGPA) \wedge GreaterThan(xGPA, yGPA) \Rightarrow SavvierStudents(x, y)}$ **unifies with:**

Rule 3 $AvgGPA(\mathrm{UCLA}, 3.51)$

Rule 4 $AvgGPA(\mathrm{USC}, 3.75)$

Rule 2 $GreaterThan(3.75, 3.51)$

Rule 5 $AvgGPA(USC, 3.75) \wedge AvgGPA(UCLA, 3.51) \wedge GreaterThan(3.75, 3.51) \Rightarrow SavvierStudents(USC, UCLA)$ {x/USC, y/UCLA, xGPA/3.75, yGPA/3.51}

$\boldsymbol{NicerCampus(x, y) \wedge SavvierStudents(x, y) \Rightarrow Better(x, y)}$ **unifies with:**

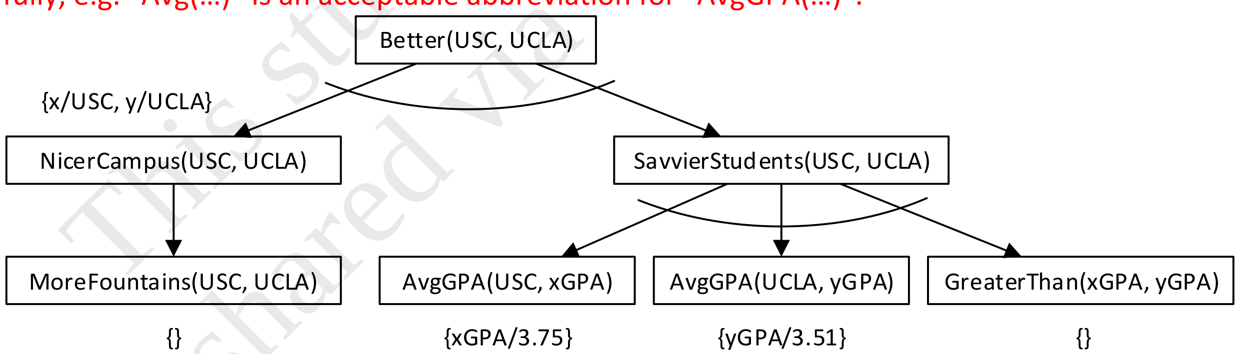
Rule 5 $\mathrm{NicerCampus(USC, UCLA) \wedge SavvierStudents(USC, UCLA) \Rightarrow Better(USC, UCLA)}$ {x/USC, y/UCLA}

The following solution is perfectly valid as well:

6,7	NicerCampus(USC, UCLA)	x/USC, y/UCLA
2,3,4,5	SavvierStudents(USC, UCLA)	x/USC, xGPA/3.75, y/UCLA, yGPA/3.51
1,8,9	Better(USC, UCLA)	x/USC, y/UCLA

3B. Backward Chaining

The node order at every level does not matter. Subtract 1% for every unifier {…} that is missing or incorrect. Subtract 1% for every rule that is missing. Subtract 1% if the "AND arc" around a node's children is missing. Empty unifiers {} can be left out. Rules don't have to be written out fully, e.g. "Avg(…)" is an acceptable abbreviation for "AvgGPA(…)".



4A. [10%] Convert the following FOL formula to CNF showing all the steps involved.

$\forall x[(\forall y(N(y) \Rightarrow \exists z M(z,y))) \Rightarrow \neg\forall y((L(x,y) \vee P(x,y)) \Rightarrow Q(x,y))]$

1.Eliminate the implication:

$\forall x[\neg(\forall y(\neg N(y) \vee \exists z M(z,y))) \vee \neg\forall y(\neg(L(x,y) \vee P(x,y)) \vee Q(x,y))]$

2.Reduce the scope of negation:

$\forall x[(\exists y\neg(\neg N(y) \vee \exists z M(z,y))) \vee \exists y\neg(\neg(L(x,y) \vee P(x,y)) \vee Q(x,y))]$

=>

$\forall x[(\exists y(N(y) \wedge \neg\exists z M(z,y))) \vee \exists y((L(x,y) \vee P(x,y)) \wedge \neg Q(x,y))]$

=>

$\forall x[(\exists y(N(y) \wedge \forall z\neg M(z,y))) \vee \exists y((L(x,y) \vee P(x,y)) \wedge \neg Q(x,y))]$

or $\forall x[(\exists y\forall z(N(y) \wedge \neg M(z,y))) \vee \exists y((L(x,y) \vee P(x,y)) \wedge \neg Q(x,y))]$

3.Standardize variables:

$\forall x[(\exists y(N(y) \wedge \forall z\neg M(z,y))) \vee \exists v((L(x,v) \vee P(x,v)) \wedge \neg Q(x,v))]$

4. Eliminate existential:

$\forall x[(N(f(x)) \wedge \forall z\neg M(z,f(x))) \vee ((L(x,g(x)) \vee P(x,g(x))) \wedge \neg Q(x,g(x)))]$

5.Drop universal quantification symbols

$[(N(f(x)) \wedge \neg M(z,f(x))) \vee ((L(x,g(x)) \vee P(x,g(x))) \wedge \neg Q(x,g(x)))]$

6.Convert to conjunction of disjunction:

$$\begin{gathered}
\big(N(f(x)) \vee L(x,g(x)) \vee P(x,g(x))\big) \wedge \\
(\neg M(z,f(x))) \vee L(x,g(x)) \vee P(x,g(x))) \quad \wedge \\
\big(\neg Q(x,g(x)) \vee N(f(x))\big) \wedge \\
(\neg Q(x,g(x)) \vee \neg M(z,f(x)))
\end{gathered}$$

7.Separate clauses:

$(N(f(x))) \vee L(x,g(x)) \vee P(x,g(x))$

$(\neg M(z,f(x))) \vee L(x,g(x)) \vee P(x,g(x)))$

$\neg Q(x,g(x)) \vee N(f(x))$

$\neg Q(x,g(x)) \vee \neg M(z,f(x))$

step1: 2pt, step2:2pt, step 3:2pt, step4: 2pt, 5&6:2pt

Note that as long as it reaches a step give all points for previous steps. for example, a student reaches step 4 but miss step 2, or 3. Still give 8 pt.

As long as get step 6, give full credit.

4B. [10%]

Given the following CNF knowledge base:

$\neg$ Pass(x, History) $\lor$ $\neg$Win(x, Lottery)) $\lor$Happy(x)

($\neg$ Study(x) $\lor$ Pass(x,y))

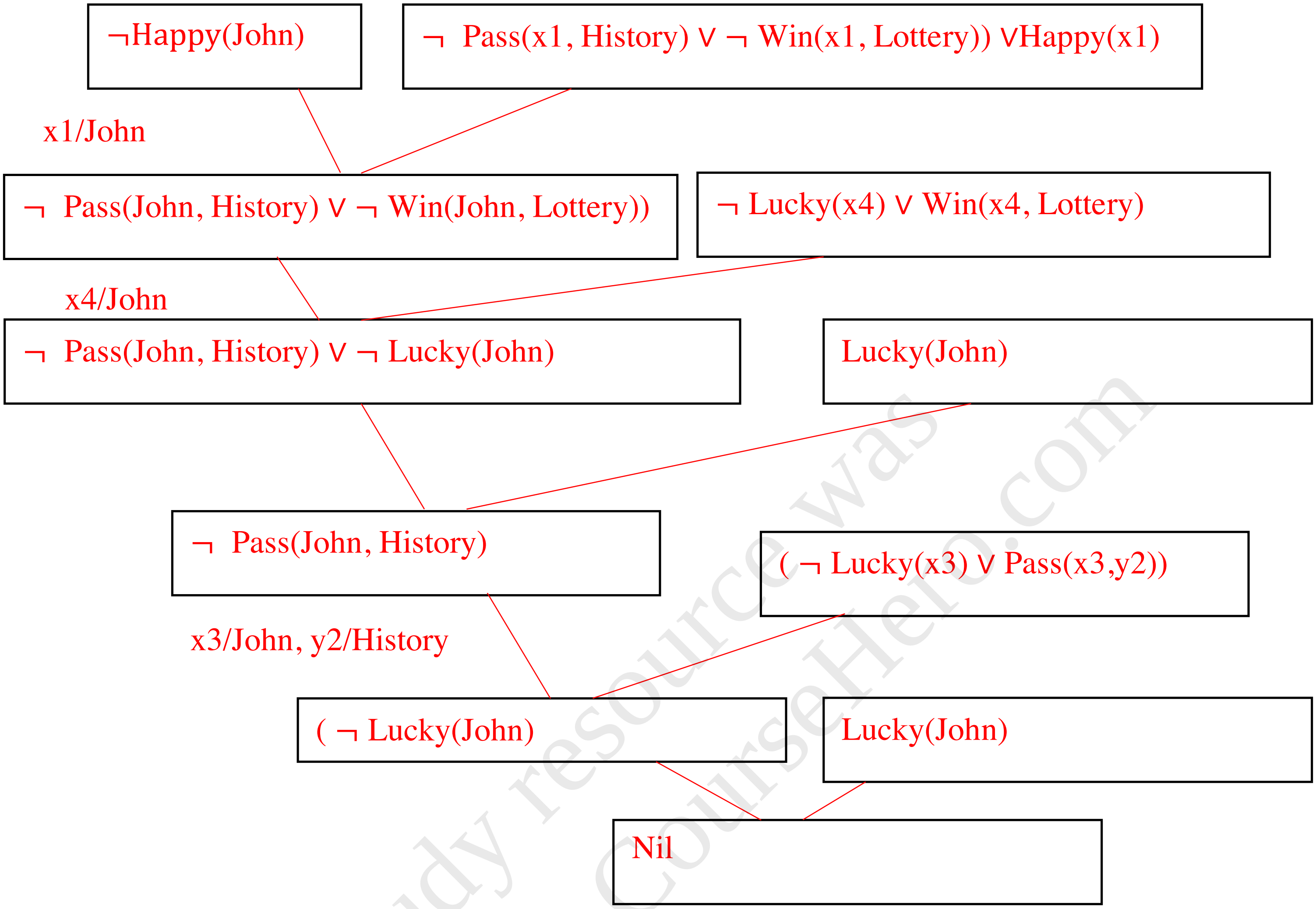
($\neg$ Lucky(x) $\lor$ Pass(x,y))

$\neg$ Study(John)

Lucky(John)

$\neg$ Lucky(x) $\lor$ Win(x, Lottery)

Please demonstrate how one can prove *Happy*(*John*) using resolution as the inference. Show all details of unification needed for each step of the inference process. Assume that *John, History* and *Lottery* are constants, and that *x* , y are variables.



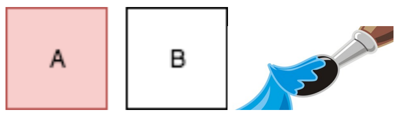
x1/John,
x4/John,
x3/John,
y2/History

each step above 2pt, 5 steps in total.

5. [20%] Planning

Consider the following painting world. We have two blocks (labeled A and B), two colors (red and blue), and a brush. The following rules apply in the world:

- The brush should be dry before getting dipped into the paint.
- When we want to paint a block, the brush should be wet with the right color.
- The block should be blank before getting colored.



5A. [6%] Using predicates Color(b, c), Blank(b), Wet(br, c), and Dry(br), write down the description of the state above under the closed-world assumption (Block A is red, Block B is blank, and the brush is wet with color blue.)

Color(A, red)   Blank(B)   Wet(brush, blue)   [Each Worth 2%][Because of the closed-world assumption, there should be no negative literals; if there is, deduct 1%]

5B. [5%] Write down the schema for action Paint that paints a block with a specific color.

[If the schema is not the same as below, but is reasonably defined and consistent, give credit.]

Action(Paint(b, c),   [1%, No Partial Credit]
    Precond: Blank(b) ∧ Wet(brush, c) [2%, 1% for each literal]
    Effect: ~ Blank(b) ∧ Color(b, c))   [2%, 1% for each literal]

5C. [9%] Write down the schema for actions DryBrush and DipBrush that dries the brush and dips the brush into a specific paint, respectively.

[If the schema is not the same as below, but is reasonably defined and consistent, give credit.]

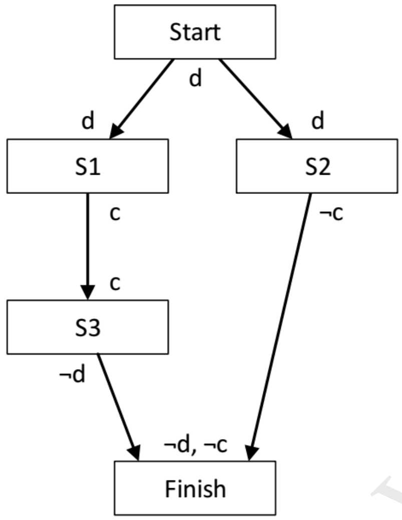
Action(DryBrush(),   [1%, No Partial Credit]
    Precond: -- (or ~ Dry(brush)) [1%, No Partial Credit]
    Effect: ~ Wet(brush, red) ∧ ~ Wet(brush, blue) ∧ Dry(brush)) [3%, 1% for each literal]

[Some might have defined the action as DryBrush(c) (drying the brush from a specific paint); if consistent, give credit.]

Action(DipBrush(c),   [1%, No Partial Credit]
    Precond: Dry(brush)   [1%, No Partial Credit]
    Effect: ~ Dry(brush) ∧ Wet(brush, c))   [2%, 1% for each literal]

6. [5%] Partial Order Planning

Consider the following partial order planning diagram. It currently contains no ordering constraints other than those implied by the partial order of the partial plan. S_1, S_2, and S_3 represent the actions taken, and c, ¬c, d and ¬d are the effects/preconditions.



Describe the flaw in this partial plan. Show how you would resolve it.

"The problem with the plan is that S2's effect negates S3's precondition and S3's effect negates S2's precondition. So without any ordering constraints, the plan goes wrong. [4%]

Because the effects of both S2 and S3 are needed for the finish state, and we have no other states providing them, we cannot resolve the problem with current states, even if we add ordering constraints. We need more states. [1%]"