

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%] $\forall x \exists y \text{ Loves}(x, y)$ 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 $\text{KB} \Rightarrow \alpha$. **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 $P(x) \vee Q(y)$)**
- 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 $K(\text{John}, x)$ 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) ☒ $\neg [\exists c,d In(c,A) \wedge In(d,B) \wedge Likes(c,d)]$, **T**
- 2) ☒ $\forall c,d [In(c,A) \wedge In(d,B)] \Rightarrow \neg Likes(c,d)$. **T**
- 3) ☐ $\neg \forall c In(c,A) \Rightarrow \exists d In(d,B) \wedge \neg Likes(c,d)$. **F**, says there is someone in A that likes everyone in state B,
- 4) ☒ $\forall c In(c,A) \Rightarrow (\forall d In(d,B) \Rightarrow \neg Likes(c,d))$. **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.

$$\forall x (IsCook(x) \wedge \neg IsChef(x)) \vee (\neg IsCook(x) \wedge IsChef(x))$$

Parentheses for grouping expressions are optional because \wedge precedes \vee .

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

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

Or

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

Or

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

Or

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

2C. Any chef cooks better than any cooks.

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

Or

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

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

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

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

Or

$$\begin{aligned} \exists x IsCook(x) \wedge (\forall y IsCook(y) \wedge x \neq y \rightarrow AssignsWork(x, y)) \\ \wedge (\forall y IsCook(y) \wedge (\forall z IsCook(z) \wedge y \neq z \rightarrow AssignsWork(y, z)) \rightarrow x = y) \end{aligned}$$

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

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

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

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

1. Eliminate the implication:

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

2. Reduce the scope of negation:

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

\Rightarrow

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

\Rightarrow

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

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

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:

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

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.

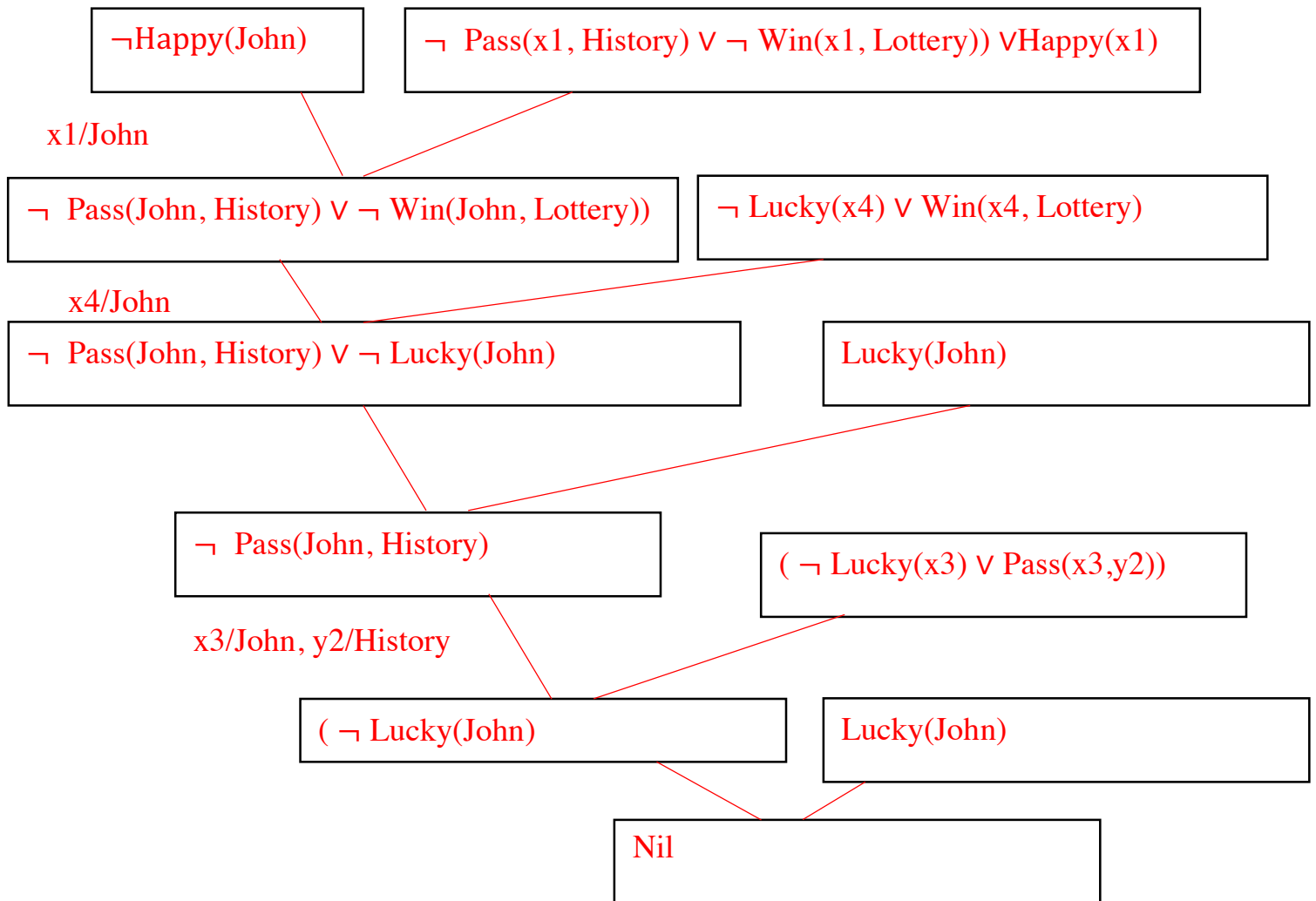
As long as get step 6, give full credit.

4B. [10%]

Given the following CNF knowledge base:

$$\neg \text{Pass}(x, \text{History}) \vee \neg \text{Win}(x, \text{Lottery}) \vee \text{Happy}(x)$$
$$(\neg \text{Study}(x) \vee \text{Pass}(x, y))$$
$$(\neg \text{Lucky}(x) \vee \text{Pass}(x, y))$$
$$\neg \text{Study}(\text{John})$$
$$\text{Lucky}(\text{John})$$
$$\neg \text{Lucky}(x) \vee \text{Win}(x, \text{Lottery})$$

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.



$x1/\text{John},$
 $x4/\text{John},$
 $x3/\text{John},$
 $y2/\text{History}$

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 $\text{Color}(b, c)$, $\text{Blank}(b)$, $\text{Wet}(br, c)$, and $\text{Dry}(br)$, 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.)

$\text{Color}(A, \text{red})$ $\text{Blank}(B)$ $\text{Wet}(\text{brush}, \text{blue})$ [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: $\text{Blank}(b) \wedge \text{Wet}(\text{brush}, c)$ [2%, 1% for each literal]

Effect: $\sim \text{Blank}(b) \wedge \text{Color}(b, c)$ [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 $\sim \text{Dry}(\text{brush})$) [1%, No Partial Credit]

Effect: $\sim \text{Wet}(\text{brush}, \text{red}) \wedge \sim \text{Wet}(\text{brush}, \text{blue}) \wedge \text{Dry}(\text{brush})$ [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.]

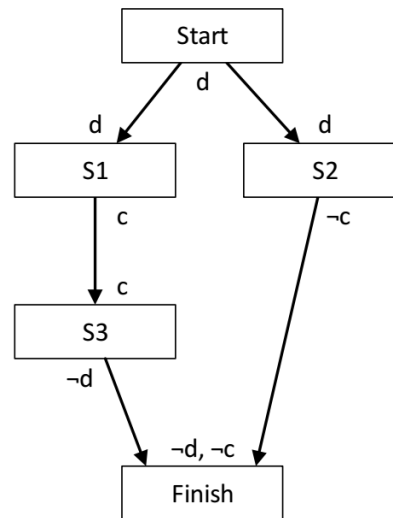
Action(DipBrush(c), [1%, No Partial Credit]

Precond: $\text{Dry}(\text{brush})$ [1%, No Partial Credit]

Effect: $\sim \text{Dry}(\text{brush}) \wedge \text{Wet}(\text{brush}, c)$ [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 , $\neg c$, d and $\neg d$ are the effects/preconditions.



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

"The problem with the plan is that S_2 's effect negates S_3 's precondition and S_3 's effect negates S_2 's precondition. So without any ordering constraints, the plan goes wrong. [4%]"

Because the effects of both S_2 and S_3 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%]"