Fall 2000

I. Transform the wff below into clause form. For each step required give a brief description of the step and perform the step (if applicable) on the space provided. Failure to follow this format may result in no credit.

<wff>:
A: (\forall x)(P(x) \rightarrow \exists z[\neg \forall y[Q(x,y) \rightarrow P(f(z))] \land \forall y[Q(x,y) \rightarrow P(z)]])

Answer:
{\neg P(x_1,y_1) \lor Q(x_1,y_1) \lor R(x_1,y_1)}
{\neg Q(x_2,y_2) \lor \neg P(f(x_2,y_2))}
{\neg P(x_3,y_3) \lor \neg Q(x_3,y_3) \lor P(g(x_3,y_3))}

II. Resolution Refutation

Bill has been murdered, and AL, Ralph, and George are suspects. AL says he did not do it. He says that Ralph was the victim’s friend but that George hated the victim. Ralph says that he was out of town on the day of the murder, and besides he didn’t even know the guy. George says he is innocent and that he saw AL and Ralph with the victim just before the murder. Assuming that everyone—except possibly for the murderer—is telling the truth, using Resolution Refutation, solve the crime.

(5) Answers Part a:
Let I(x) mean x is innocent; F(x,y) mean x is a friend of y; Hate(x,y) x hates y; Out(x) x is out of town
With(x,y) x is with y; Knows(x,y) x knows y

[1] I(AL) \rightarrow F(Ralph,Bill) \land Hate(George,Bill)
[2] I(Ralph) \rightarrow Out(Ralph) \land \neg Knows(Ralph,Bill)
[3] I(George) \rightarrow With(AL,Bill) \land With(Ralph,Bill)

(3) Answers Part b:
[5] (\forall x)(\forall y)\neg F(x,y) \rightarrow \neg I(x) \rightarrow \neg Hate(x,y)
[6] (\forall x)(\forall y)F(x,y) \rightarrow Knows(x,y)
[7] (\forall x)Out(x) \rightarrow With(x,Bill)
[8] \neg Knows(x,y) \rightarrow With(x,y) \land \neg F(x,y)
[9] \neg Knows(x,Bill) \rightarrow I(x)
[10] I(AL) \lor I(Ralph)
[11] I(George) \lor I(Ralph)
[12] I(AL) \lor I(George)

Fall 2001

I. Transform the wff below into clause form. For each step required give a brief description of the step and perform the step (if applicable) on the space provided. Failure to follow this format may result in no credit.

<wff>:
A: \forall x \forall y[\{ P(x,y) \lor Q(x,y) \} \rightarrow R(x,y)]

Answer:
{ \neg P(x_1,y_1) \land R(x_1,y_1) }
{ \neg Q(x_2,y_2) \land R(x_2,y_2) }
II. Resolution Refutation

If a course is easy, some students are happy. If a course has a final, no students are happy. Use Resolution to show that, if a course has a final, the course is not easy.

(5) Answers Part a:
For every course and every student, if the course has a final and the student is taking the course, then the student is not happy. For every course, if the course is easy, then there is a student taking the course who is not happy.

\[1\] \forall c \forall s \{ F(c) \land T(s,c) \rightarrow \neg H(s) \} \\
\[2\] \forall E(c) \rightarrow \exists s [T(s,c) \land H(s)] \\
\[3\] Goal: (\forall c)[F(c) \rightarrow \neg E(c)]

(2) Answers Part b:
None needed

(5) Answers Part c:
\[1\] \neg F(c) \lor \neg T(s,c) \lor \neg H(s) \\
\[2a\] \neg E(c) \lor T(g(c),c) \\
\[2b\] \neg E(c) \lor H(g(c)) \\
\[3a\] F(Crip_Course) \\
\[3b\] E(Crip_Course)

\text{g(c) designates the Skolem happy student in each course and Crip_Course designates the Skolem course with a final that is hypothesized to be easy.}

III. Adversarial Search

Part (a) A chooses C
Part (b) A chooses D
Part (c) Do Not Evaluate \{O,U,W,X,Y, and K\}
Part (e) Alpha-Beta and Minimax produce the same results

Fall 2002

I. (a) Transform the \text{wff} A below into CNF (\text{clause}) matrix form. For each step required give a brief description of the step and perform the step (if applicable) on the space provided. Failure to follow this format may result in no credit.

(b) Rewrite your answer in part (a) as a single (1 line) \text{<wff>} simplifying if necessary.

(c) Which form is better (matrix form or the 1-line form) and why? \{No explanation, No credit\}

{\text{wff}} A: (\forall x)\{ P(x) \rightarrow [-(\forall y)\{Q(x,y) \rightarrow P(f(z))\}] \land (\forall y)\{Q(x,y) \rightarrow P(x) \} \} 

(5) Part (a) Answer:
\[
\neg P(x_1) \lor Q(x_1, g(x_1)) \\
\neg P(x_2) \lor \neg P(f(B))
\]

(5) Part (b) Answer:
\[ A: (\forall x)\{ \neg P(x) \lor [Q(x, g(x)) \land \neg P(f(B))]) \} \text{ or} \]
A: \((\forall x)(P(x) \rightarrow [Q(x, g(x)) \land \neg P(f(B))])\)

(5) Part (c) Answer:
Part (a) answer is more general than part (b) because if you substitute for x, say x=Obj in part (b) you obtain \(\neg P(\text{Obj}) \lor [Q(\text{Obj}, g(\text{Obj})) \land \neg P(f(B))}\) which is \(\neg P(\text{Obj}) \lor Q(\text{Obj}, g(\text{Obj}))\land \neg P(f(B))\)
But substituting in part (a) yields \(\neg P(\text{Obj}) \lor Q(\text{Obj}, g(\text{Obj}))\land \neg P(x_2) \lor \neg P(f(B))\) which is more general.

II. Resolution Refutation
Sam, Clyde and Oscar are elephants. We know the following facts about them:
1. Sam is pink.
2. Clyde is gray and likes Oscar.
3. Oscar is either pink, or gray (but not both) and likes Sam.
Use resolution refutation to prove that a gray elephant likes a pink elephant; that is prove

\((\exists x)(\exists y)(\text{Gray}(x) \land \text{Pink}(y) \land \text{Likes}(x,y))\)

(5) Answers Part a:
Sam, Clyde and Oscar are elephants. Sam is pink. Clyde is gray and likes Oscar. Oscar is either pink, or gray (but not both) and likes Sam. Prove that a gray elephant likes a pink elephant.

[1] Pink(Sam)
[2] Gray(Clyde)
[3] Likes(Clyde,Oscar)
[4] Pink(Oscar) ∨ Gray(Oscar)
[5] Likes(Oscar,Sam)
[6] \((\exists x)(\exists y)(\text{Gray}(x) \land \text{Pink}(y) \land \text{Likes}(x,y))\) \{given\}

(2) Answers Part b:
Pink(Oscar) implies \neg Gray(Oscar) and Gray(Oscar) implies \neg Pink(Oscar) else none needed

Fall 2003
I. Transform the wff A below into CNF (clause form) matrix form. For each of the steps required give a brief description of the step and perform the step or write N/A{not applicable} on the space provided. Failure to follow this format will result in no credit. In wff A the set \(\{x,y,z\}\) are variables, the set \(\{A,B,C,D,E\}\) are functions and I is a constant.

\(\{\text{wff } A\}: (\forall x)[(A(x) \land B(x)) \rightarrow [C(x,I) \land (\exists y)(\exists z)[C(y,z) \rightarrow D(x,y)]]) \lor (\forall x)[E(x)]\)

\(A_{10}: [\neg A(x_1) \lor \neg B(x_1) \lor C(x_1,I) \lor E(w_1)]\)
\(\neg A(x_2) \lor \neg B(x_2) \lor C(f(x_2),g(x_2)) \lor D(x_2,f(x_2)) \lor E(w_2)]\)

II. Resolution Refutation

EXCITING LIFE
ALL PEOPLE WHO ARE NOT POOR AND ARE SMART ARE HAPPY. THOSE PEOPLE WHO READ ARE NOT STUPID. JOHN CAN READ AND IS WEALTHY. HAPPY PEOPLE HAVE EXCITING LIVES. CAN ANYONE BE FOUND WITH AN EXCITING LIFE?
5) Answers Part a:

All people who are not poor and are smart are happy. Those people who read are not stupid.

John can read and is wealthy. Happy people have exciting lives. Can anyone be found with an exciting life?

[1] (\forall x)[\neg poor(x) \land smart(x) \rightarrow happy(x)]
[2] (\forall y)[read(y) \rightarrow smart(y)]
[3] read(John) \land \neg poor(John)
[4] (\forall z)[happy(z) \rightarrow exciting(z)]
[5] (\exists w)[exciting(w)] \{goal\}

(2) Answers Part b:

None needed

III. Heuristic Search

The following figure shows a search tree with the state indicated by the tuple inside parentheses. A letter indicates the state name and the integer indicates the estimated cost for finding a solution from that state (a cost of 0 indicates a goal state). Using the Graph-Search algorithm discussed in class, give the solution tree or steps using depth-first search. How many nodes did depth-first expand? Repeat using breadth-first search. How many nodes did breadth-first expand? Repeat using heuristic search. How many nodes did heuristic search expand? Repeat using A* search. How many nodes did A* expand? You must clearly justify your answer(s). "Feelings" or "intuition" are not good/sound reasons. NO JUSTIFICATION \iff NO CREDIT. You must give me the details of the algorithm in order to receive any credit for each case. Can any of these algorithms ever find N as a solution? Explain

[Done in class]
Fall 2000

I. Transform the wff below into clause form. For each step required give a brief description of the step and perform the step (if applicable) on the space provided. Failure to follow this format may result in no credit.

\[ A : (\forall x)(P(x) \rightarrow \exists y[Q(x,y) \rightarrow P(f(y)))] \land \exists y[Q(x,y) \rightarrow P(y)] \]  

Answer:

\[ \neg P(x) \lor Q(x,f(x)) \]
\[ \neg P(x) \lor \neg P(f(g(x,y))) \]
\[ \neg P(x) \lor Q(x,f(x)) \lor P(g(x,y)) \]

II. Resolution Refutation

Bill has been murdered, and AL, Ralph, and George are suspects. AL says he did not do it. He says that Ralph was the victim’s friend but that George hated the victim. Ralph says that he was out of town on the day of the murder, and besides he didn’t even know the guy. George says he is innocent and that he saw AL and Ralph with the victim just before the murder. Assuming that everyone—except possibly for the murderer—is telling the truth, using Resolution Refutation, solve the crime.

(5) Answers Part a:
Let I(x) mean x is innocent; F(x,y) mean x is a friend of y; Hate(x,y) x hates y; Out(x) x is out of town
With(x,y) x is with y; Knows(x,y) x knows y

[1] I(AL) \rightarrow F(Ralph,Bill) \land Hate(George,Bill)
[2] I(Ralph) \rightarrow Out(Ralph) \land \neg Knows(Ralph,Bill)
[3] I(George) \rightarrow With(AL,Bill) \land With(Ralph,Bill)

(3) Answers Part b:

[5] (\forall x)(x)Hate(x,y) \rightarrow \neg F(x,y) If x hates y then x is not a friend of y; also (\forall x)(\forall y)F(x,y) \rightarrow 
\neg Hate(x,y)
[6] (\forall x)(\forall y)F(x,y) \rightarrow Knows(x,y) If x is a friend of y then x knows y.
[7] (\forall x)Out(x) \rightarrow \neg With(x,Bill) If x is out of town then x cannot be with Bill
[8] \neg Knows(x,y) \rightarrow \neg With(x,y) \land F(x,y) If x does not know y then x is not a friend of y nor can x be with y
[9] \neg Knows(x,Bill) \rightarrow I(x) If x does not know Bill, then x must be innocent.
[10] I(AL) \lor I(Ralph) Either AL or Ralph are innocent (i.e., George is not innocent)
[11] I(George) \lor I(Ralph) Either George or Ralph are innocent (i.e., AL is not innocent)
[12] I(AL) \lor I(George) Either AL or George are innocent (i.e., Ralph is not innocent)

Fall 2001

I. Transform the wff below into clause form. For each step required give a brief description of the step and perform the step (if applicable) on the space provided. Failure to follow this format may result in no credit.

\[ A : \forall x \forall y [ (P(x,y) \lor Q(x,y)) \rightarrow R(x,y)] \]  

Answer:

\{ \{ \neg P(x,y), \ R(x,y) \} \}
\{ \{ \neg Q(x,y), \ R(x,y) \} \}
Fall 2004 Exam
(25) Conversion to Clause Form

I. Transform the wff $A$ below into CNF (clause form) matrix form. For each of the steps required give a brief description of the step and perform the step or write N/A (not applicable) on the space provided. Failure to follow this format will result in no credit. In wff A the set \{x, y, z\} are variables, the set \{P, Q, S\} are functions and there are no constants.

$\{wff A\} = (\forall x \exists y \forall z) \{[P(x,y) \rightarrow Q(y,z)] \lor [Q(y,x) \rightarrow S(x,y)]\} \rightarrow (\exists x \forall z)[P(x,y) \rightarrow S(x,y)]$

(2) Step 0: Eliminate redundant quantifiers and take the existential closure

$A_0: \exists z (\forall x \exists y) \{\sim \exists y \forall z (P(x,y) \lor Q(y,z)) \land \sim [Q(y,x) \lor S(x,y)]\} \lor (\exists x \forall y)[\sim P(x,y) \lor \sim S(x,y)]\}$

(2) Step 1: Remove implications

$A_1: \exists z (\forall x \exists y) \{[\sim P(x,y) \lor Q(y,z)] \land [\sim Q(y,x) \lor S(x,y)]\} \lor (\exists x \forall y)[\sim P(x,y) \lor S(x,y)]\}$

(2) Step 2: Move the Negations down to the Atf's

$A_2: \exists z (\exists x \forall y)\{[P(x,y) \land \sim Q(y,z)] \lor [Q(y,x) \land \sim S(x,y)]\} \lor (\exists x \forall y)[\sim P(x,y) \lor S(x,y)]\}$

(2) Step 3: Standardize Variables Apart

$A_3: \exists z (\exists x \forall y)\{[P(x,y) \land \sim Q(y,z)] \lor [Q(y,x) \land \sim S(x,y)]\} \lor (\exists x \forall y)[\sim P(x,y) \lor S(x,y)]\}$

(2) Step 4: Skolemize: Let $z = C_1, x_1 = C_2; x_2 = C_3$

$A_4: (\forall y)\{P(C_2,y) \land \sim Q(y,C_1)\} \lor [Q(y,x_1) \land \sim S(x_1,y_1)]\} \lor (\exists x \forall y)[\sim P(x_2,y_2) \lor S(x_2,y_2)]\}$

(2) Step 5: Move universal quantifiers to the left

$A_5: (\forall x) (\forall y)\{[P(C_2,x) \land \sim Q(x,C_1)] \lor [Q(x,y) \land \sim S(x,y)]\} \lor (\exists x \forall y)[\sim P(x,y) \lor S(x,y)]\}$

(6) Step 6: Multiply out & distribute $\lor$ over $\land$ using $E_1 \lor (E_2 \land E_3) = (E_1 \lor E_2) \land (E_1 \lor E_3)$

$A_{6a}: (\forall x) (\forall y)\{[P(C_2,x) \lor Q(x,C_1)] \land [P(C_2,x) \lor \sim S(x,y)]\} \lor [Q(x,y) \land \sim S(x,y)]\} \lor [Q(x,y) \land \sim S(x,y)]\}$

$A_{6b}: (\forall x) (\forall y)\{[P(C_2,x) \lor Q(x,C_1)] \land [P(C_2,x) \lor \sim S(x,y)]\} \lor [Q(x,y) \land \sim S(x,y)]\} \lor [Q(x,y) \land \sim S(x,y)]\}$

(2) Step 7: Write in Matrix Form

$A_7: (\forall x) (\forall y)[\sim P(C_3,y) \lor S(C_3,y) \lor P(C_2,x) \lor Q(x,C_2)]$

$A_8: (\forall x) (\forall y)[\sim P(C_3,y) \lor S(C_3,y) \lor P(C_2,x) \lor \sim S(C_2,x)]$

(2) Step 8: Eliminate Universal Quantifiers

$A_9: \sim P(C_3,y) \lor S(C_3,y) \lor P(C_2,x) \lor Q(x,C_2)$

$\sim P(C_3,y) \lor S(C_3,y) \lor P(C_2,x) \lor \sim S(C_2,x)$

$\sim P(C_3,y) \lor S(C_3,y) \lor \sim Q(x,C_1) \lor Q(x,C_2)$

$\sim P(C_3,y) \lor S(C_3,y) \lor \sim Q(x,C_1) \lor \sim S(C_2,x)$
I. Conversion to Clause Form (continued)

(2) Step 9: Rename Variables

\[ A_9: \neg P(C_3, y_1) \lor S(C_3, y_1) \lor P(C_2, x_1) \lor Q(x_1, C_2) \]
\[ \neg P(C_3, y_2) \lor S(C_3, y_2) \lor P(C_2, x_2) \lor \neg S(C_2, x_2) \]
\[ \neg P(C_3, y_3) \lor S(C_3, y_3) \lor \neg Q(x_3, C_1) \lor Q(x_3, C_2) \]
\[ \neg P(C_3, y_4) \lor S(C_3, y_4) \lor \neg Q(x_4, C_1) \lor \neg S(C_2, x_2) \]

(1) Step 10: Remove Tautologies & Simplify (not applicable)

\[ A_{10}: \neg P(C_3, y_1) \lor S(C_3, y_1) \lor P(C_2, x_1) \lor Q(x_1, C_2) \]
\[ \neg P(C_3, y_2) \lor S(C_3, y_2) \lor P(C_2, x_2) \lor \neg S(C_2, x_2) \]
\[ \neg P(C_3, y_3) \lor S(C_3, y_3) \lor \neg Q(x_3, C_1) \lor Q(x_3, C_2) \]
\[ \neg P(C_3, y_4) \lor S(C_3, y_4) \lor \neg Q(x_4, C_1) \lor \neg S(C_2, x_2) \]
II. Resolution Refutation

Consider the following database about zebras:

Zebras are mammals, stripped, and medium size. Mammals are animals and warm-blooded. Striped things are non-solid and non-spotted. Things of medium size are neither small nor large. If Zeke is a zebra, is Zeke non-large?

Solve by drawing a Refutation Graph resulting from the Breadth-First strategy. (Make sure you mark clearly the required substitutions).

[Required: Please note the assigned point values. Each part MUST be answered with something. If left blank, then no credit will be assigned]

(5) a. Represent the axioms/goal in the Predicate Calculus.
(2) b. Represent any commonsense knowledge needed to solve the problem using Predicate Calculus,
(5) c. Convert your axioms, goal and commonsense knowledge (if any) to clause form,
(10) d. Draw your Refutation Graph, show substitutions are consistent.
(3) e. Describe how your graph meets the strategy. What other strategy could you have used and why?

Answers Part a:
Zebras are mammals, stripped, and medium size. Mammals are animals and warm-blooded. Striped things are non-solid and non-spotted. Things of medium size are neither small nor large. If Zeke is a zebra, is Zeke non-large?

\[1 \forall x [\text{zebra}(x) \rightarrow \text{mammal}(x)]\]
\[2 \forall x [\text{zebra}(x) \rightarrow \text{striped}(x)]\]
\[3 \forall x [\text{zebra}(x) \rightarrow \text{medium}(x)]\]
\[4 \forall x [\text{mammal}(x) \rightarrow \text{animal}(x)]\]
\[5 \forall x [\text{mammal}(x) \rightarrow \text{warm}(x)]\]
\[6 \forall x [\text{striped}(x) \rightarrow \text{nonsolid}(x)]\]
\[7 \forall x [\text{striped}(x) \rightarrow \text{nonspotted}(x)]\]
\[8 \forall x [\text{medium}(x) \rightarrow \text{nonsmall}(x)]\]
\[9 \forall x [\text{medium}(x) \rightarrow \text{nonlarge}(x)]\]
\[10 \text{zebra}(\text{zeke}) \rightarrow \text{nonlarge}(\text{zeke})\]

Answers Part b:
None needed

(5) Answers Part c:
\[1 \neg \text{zebra}(x_1) \lor \text{mammal}(x_1)\]
\[2 \neg \text{zebra}(x_2) \lor \text{striped}(x_2)\]
\[3 \neg \text{zebra}(x_3) \lor \text{medium}(x_3)\]
\[4 \neg \text{mammal}(x_4) \lor \text{animal}(x_4)\]
\[5 \neg \text{mammal}(x_5) \lor \text{warm}(x_5)\]
\[6 \neg \text{striped}(x_6) \lor \text{nonsolid}(x_6)\]
\[7 \neg \text{striped}(x_7) \lor \text{nonspotted}(x_7)\]
\[8 \neg \text{medium}(x_8) \lor \text{nonsmall}(x_8)\]
\[9 \neg \text{medium}(x_9) \lor \text{nonlarge}(x_9)\]
\[10a \text{zebra}(\text{zeke})\]
\[10b \neg \text{nonlarge}(\text{zeke})\]
II. Resolution Refutation (continued)

(10) Refutation Graph Part d:

\[
\begin{array}{cccccccccc}
\mathbb{R}_1 & \mathbb{R}_2 & \mathbb{R}_3 & \mathbb{R}_5 & \mathbb{R}_6 & \mathbb{R}_7 & \mathbb{R}_8 & \mathbb{R}_9 & \mathbb{R}_4 & \mathbb{R}_{10} \\
nil & nil \\
\end{array}
\]

- \mathbb{R}_1 = [10a'] with [1] mammal(zeke) \{zeke/x_1\}
- \mathbb{R}_2 = [10a'] with [2] striped(zeke) \{zeke/x_2\}
- \mathbb{R}_3 = [10a'] with [3] medium(zeke) \{zeke/x_3\}
- \mathbb{R}_4 = [10b'] with [9] ~medium(zeke) \{zeke/x_9\}
- \mathbb{R}_6 = [1] with [5] warm(\{\}) \{\}\
- \mathbb{R}_3 \cap \mathbb{R}_4
- \mathbb{R}_4 \cap \mathbb{R}_{10}

Consistency Check

\[U_1 = \text{zeke, zeke, zeke, zeke} \quad U_2 = \text{x_1, x_2, x_3, x_9}\]

\[U_1 = U_2 \{\text{zeke/x_1, zeke/x_2, zeke/x_3, zeke/x_9}\}\]

Since \(U_1 \) & \(U_2\) unify, then the substitutions are consistent

(3) Answer Part e: My strategy is Breadth First

(1) Since every 1st level resolvent \(\mathbb{R}_1 - \mathbb{R}_{10}\) comes from the base set + negation of the wff to be proved.
Nil came from two first level resolvents \(\mathbb{R}_3\) and \(\mathbb{R}_4\) or from a 1st level resolvent \(\mathbb{R}_{10}\) and from a base set [10b']
(2) Since Nil came from a 1st level resolvent \(\mathbb{R}_{10}\) which came from the negation of the wff
and from it and another member of the negation of the wff [10b'], this represents a set of support strategy
III. Adversarial Search

Consider the following game tree in which the static scores (in parentheses at the tip nodes) are all from the first player’s point of view.

(a) Assuming that the first player is the maximizing player, what move should the first player choose?

(b) Assuming that the first player is the minimizing player, what move should the first player choose?

(c) What nodes would not need to be examined in part (a) using the alpha-beta algorithm—assuming that the nodes are examined in left-to-right order?

(d) What nodes would not need to be examined in part (b) using the alpha-beta algorithm—assuming that the nodes are examined in left-to-right order?

(e) Is the first player’s move in parts (a) and (c) or in parts (b) and (d) different? Explain.

Part (a)

D, E, F, G choose max or D=(H 3); E=(J 0); F=(L 4); G=(P 1)
B, C chooses min or B=E=(J 0); C=(P 1)
A chooses max or A=C=G=(P 1)
A chooses C toward solution A→C→G→P

Part (b)

D, E, F, G choose min or D=(I 2); E=(K-1); F=(M 3); G=(N 0)
B, C chooses max or B=D=(I 2); C=F=(M 3)
A chooses min or A=B=D=(I 2)
A chooses B toward solution A→B→D→I

Part (c)

D chooses max evaluating (H 3) & (I 2) or α_D=3 (H 3); Now B chooses min so β_B ≤ 3 (H 3)
Evaluate (J 0); now α_B≥0 (J 0) and β_B ≤ 3 (H 3) therefore no Beta Cutoff and continue
Evaluate (K –1); now α_B=0 (J 0) now β_B = 0 (J 0)
Now B chooses min so β_B = 0 (J 0), therefore α_C≥0 (J 0)
Evaluate (L 4); now α_C≥4 (L 4) and β_C ≤ 4 (L 4) therefore no Alpha cutoff & continue
Evaluate (M 3); now α_C=4 (L 4) and β_C ≤ 4 (L 4)
Evaluate (N 0); now α_C≥0 (N 0); no cutoff & continue
Evaluate (P 1); now α_C=1 (P 1); β_C =1 (P 1) no cutoff & continue
A chooses C to G to P (P 1)
Alpha-Beta had Pruning resulted in no advantage
III. Adversarial Search. (continued)

Part (d)
D chooses min evaluating (H 3) & (I 2) or $\beta_D = 2$ (I 2); Now B chooses max so $\alpha_E \geq 2$ (I 2)
Evaluate (J 0); now $\beta_F \leq 0$ (J 0) and $\alpha_E \geq 2$ (I 2) Alpha Cutoff at E and continue $\alpha_E = 2$ (I 2); $\beta_A \leq 2$

Evaluate (L 4); now $\beta_D \leq 4$ (L 4) and $\alpha_C \geq 4$ (L 4) therefore no Alpha cutoff & continue
Evaluate (M 3); now $\alpha_I = 4$ (L 4); and $\beta_C \leq 4$ (L 4) and $\beta_A \leq 2$ (I 2) Beta Cutoff at C and $\beta_A = 2$ (I 2)
A chooses B to D to I (I 2)
Do Not Evaluate {K, M, N, P}

Part (e)
A chooses C toward solution A$\rightarrow$C$\rightarrow$G$\rightarrow$P in both parts (a) and (c) because Alpha-Beta and Minimax produce the same results for the same problem.
Similarly, A chooses B toward solution A$\rightarrow$B$\rightarrow$E$\rightarrow$J in both parts (b) and (d) because Alpha-Beta and Minimax produce the same results for the same problem.
In my analysis that was indeed the case.
IV. Computation Deduction.

We wish to replace Ron Zook with Bob Stoops in a short list of ex-Gator coaches. Using Resolution Refutation deduce the following computation to obtain a value for the goal (3 pts) by performing a consistent Refutation Trace (17 pts) for the goal and prove (or provide a good argument) its consistency (5 pts.) Make sure your resolution refutation trace is clearly marked and it follows a complete strategy.

Facts:
F₁: swap(X,Y,nil,nil).

Rules:
R₁: swap(S₁,S₂,Y,Z) → swap(S₁,S₂,cons(S₁,Y),cons(Z,S₂))
R₂: {swap(S₁,S₂,Y,Z) ∧ X≠S₁} → swap(S₁,S₂,cons(X,Y),cons(X,Z))

Goal: (∃z) swap(ron, bob, cons(steve, cons(ron, cons(galen,nil))), z)

{Note: If you prefer, you may use the notation swap(ron, bob, (steve ron galen), z).}

Required: Give the resolution trace, show the substitutions are consistent, and obtain the value of the goal.

(1 pts) I am using Set-of-Support which is a complete strategy
F₁: swap(A,B,nil,nil).
R₁: ~ swap(S₁,S₂,Y₁,Z₁) ∨ swap(S₁,S₂,cons(S₁,Y₁),cons(S₂,Z₁))
R₂: ~ swap(S₁,S₂,Y₂,Z₂) ∨ ~ X≠S₁ ∨ swap(S₁,S₂,cons(X,Y₂),cons(X,Z₂))

~Goal: ~swap(ron, bob, (steve ron galen), z)

(3 pts) ℛ₁=ℛ{~Goal,R₂}: ~swap(S₁,S₂,Y₂,Z₂) ∨ ~X≠S₁ {ron/S₁, bob/S₂, steve/X, (ron galen)/Y₂, cons(X,Z₂)/z}
ℛ₁: ~swap(ron, bob, (ron galen), Z₂) ∨ ~steve≠ron {this evaluates to nil}

(3 pts) ℛ₂=ℛ{ℛ₁,R₁}: ~ swap(S₁,S₂,Y₁,Z₁) {ron/S₁, bob/S₂, (galen)/Y₁, cons(S₂,Z₁)/Z₂}
ℛ₂: ~swap(ron, bob, cons(galen,nil), Z₁)

(3 pts) ℛ₂=ℛ₂-ℛ R₂': ~swap(S₁,S₂,Y₂',Z₂') ∨ ~X'≠S₁ {ron/S₁, bob/S₂, galen/X', nil/Y₂', cons(X',Z₂')/Z₁}
ℛ₂: ~swap(ron, bob, nil, Z₂') ∨ ~galen≠ron {this evaluates to nil}

(3 pts) ℛ₃=ℛ₂-ℛ-F₁: nil {ron/S₁, bob/S₂, nil/Z₂'}

(4 pts) Therefore Z₂'=nil; Z₁=cons(galen,nil)= (galen); Y₂'=nil; Z₂=cons(bob,(galen))=(bob galen);
z=cons(steve, cons(bob, cons(galen,nil)))=(steve bob galen)

(3) Answer: (∃z)swap(ron, bob, cons(steve, cons(ron, cons(galen,nil))), z) is true
with z = cons (steve, cons(bob, cons(galen,nil))) = (steve bob galen)

(5) Substitutions will be consistent because I changed variables every time I re-used any rules and all the
variables were originally standardized apart. To prove consistency we assemble all the numerators in a set, say, $U_1$, and all the denominators in a set, say, $U_2$ and show that $U_1 = U_2 \sigma$ and $\sigma \neq \text{null}$.

**IV. Computation Deduction.** (continued)
Fall 2005 Exam 2 Periods

20. Conversion to Clause Form

I. Transform the wff A below into CNF (clause form) matrix form. For each of the 10 “official steps” required give a brief description of the step and perform the step or write N/A{not applicable} on the space provided. Failure to follow this format will result in no credit. In wff A the set \{w,x,y,t\} are variables, the set \{P,Q,R,A,B\} are functions and there are no constants.

\[
\{\text{wff } A\}: (\forall x)(P(x) \rightarrow (A(x) \land B(x) \lor C(x, w))) \lor (\forall y) (\exists u) [Q(y, t) \lor ((\forall x) R(x) \rightarrow \neg B(y))]
\]

(2) Step 0: Eliminate redundant quantifiers and take the existential closure

\[
A_0: (\exists w) (\exists t) (\forall x) \{P(x) \rightarrow (A(x) \land B(x) \lor \neg C(x, w))\} \lor (\forall y) [Q(y, t) \lor ((\forall x) R(x) \rightarrow \neg B(y))]
\]

(2) Step 1: Remove implications

\[
A_1: (\exists w) (\exists t) (\forall x) \{\neg P(x) \lor (A(x) \land B(x) \lor \neg C(x, w))\} \lor (\forall y) [Q(y, t) \lor ((\forall x) R(x) \rightarrow \neg B(y))]
\]

(2) Step 2: Move the Negations down to the Atfs

\[
A_2: (\exists w) (\exists t) (\forall x) \{\neg P(x) \lor (A(x) \land B(x) \lor \neg C(x, w))\} \lor (\forall y) [Q(y, t) \lor ((\exists z) \neg R(x) \lor \neg B(y))]
\]

(1) Step 3: Standardize Variables Apart

\[
A_3: (\exists w) (\exists t) (\forall x) \{\neg P(x) \lor (A(x) \land B(x) \lor \neg C(x, W))\} \lor (\forall y) [Q(y, T) \lor \neg R(f(y)) \lor \neg B(y)]
\]

(2) Step 4: Skolemize: Let \(w=W; t=T; z=f(y)\)

\[
A_4: (\forall x) \{\neg P(x) \lor (A(x) \land B(x) \lor \neg C(x, W))\} \lor (\forall y) [Q(y, T) \lor \neg R(f(y)) \lor \neg B(y)]
\]

(1) Step 5: Move universal quantifiers to the left

\[
A_5: (\forall x) (\forall y) \{\neg P(x) \lor (A(x) \land B(x) \lor \neg C(x, W))\} \lor [Q(y, T) \lor \neg R(f(y)) \lor \neg B(y)]
\]

(4) Step 6: Multiply out & distribute \lor over \land using \(E_1 \lor (E_1 \land E_2) = (E_1 \lor E_2) \land (E_1 \lor E_2)\)

\[
A_{6a}: (\forall x)(\forall y)((\neg P(x) \lor Q(y, T) \lor \neg R(f(y)) \lor \neg B(y)) \lor ((\neg C(x, W) \lor A(x)) \land (\neg C(x, W) \lor B(x))))
\]

\[
A_{6b}: (\forall x)(\forall y)\{[ \neg P(x) \lor Q(y, T) \lor \neg R(f(y)) \lor \neg B(y) \lor \neg C(x, W) \lor A(x)] \land [ \neg P(x) \lor Q(y, T) \lor \neg R(f(y)) \lor \neg B(y) \lor \neg C(x, W) \lor B(x)] \}
\]
I. Conversion to Clause Form (continued)

(1) Step 7: Write in Matrix Form

\[ A_7: \quad (\forall x)(\forall y)[\sim P(x) \lor Q(y,T) \lor R(f(y)) \lor B(y) \lor C(x,W) \lor A(x)] \]
\[ (\forall x)(\forall y)[\sim P(x) \lor Q(y,T) \lor R(f(y)) \lor B(y) \lor C(x,W) \lor B(x)] \]

(1) Step 8: Eliminate Universal Quantifiers

\[ A_8: \quad [\sim P(x) \lor Q(y,T) \lor R(f(y)) \lor B(y) \lor C(x,W) \lor A(x)] \]
\[ [\sim P(x) \lor Q(y,T) \lor R(f(y)) \lor B(y) \lor C(x,W) \lor B(x)] \]

(2) Step 9: Rename Variables

\[ A_9: \quad [\sim P(x_1) \lor Q(y_1,T) \lor R(f(y_1)) \lor B(y_1) \lor C(x_1,W) \lor A(x_1)] \]
\[ [\sim P(x_2) \lor Q(y_2,T) \lor R(f(y_2)) \lor B(y_2) \lor C(x_2,W) \lor B(x_2)] \]

(2) Step 10: Remove Tautologies & Simplify: 2nd row drops out since \{\sim B(y_2) \lor B(x_2)\} = True

\[ A_{10}: \quad [\sim P(x_1) \lor Q(y_1,T) \lor R(f(y_1)) \lor B(y_1) \lor C(x_1,W) \lor A(x_1)] \]
(25)

II. Resolution Refutation

THE MEMBERS OF THE ELM ST. BRIDGE CLUB ARE JOE, SALLY, BILL, AND ELLEN. JOE IS MARRIED TO SALLY. BILL IS ELLEN’S BROTHER. THE SPOUSE OF EVERY MARRIED PERSON IN THE CLUB IS ALSO IN THE CLUB. THE LAST MEETING OF THE CLUB WAS AT JOE’S HOUSE. PROVE THAT (1) THE LAST MEETING OF THE CLUB WAS AT SALLY’S HOUSE & (2) ELLEN IS NOT MARRIED.

Solve by drawing a Refutation Graph resulting from your choice of strategy. (Make sure you mark clearly the required substitutions).

[Required: Please note the assigned point values. Each part MUST be answered with something. If left blank, then no credit will be assigned]

(5) a. Represent the axioms/goal in the Predicate Calculus.

(2) b. Represent any commonsense knowledge needed to solve the problem using Predicate Calculus,

(5) c. Convert your axioms, goal and commonsense knowledge (if any) to clause form,

(10) d. Draw your Refutation Graph, show substitutions are consistent.

(3) e. Describe how your graph meets the strategy. What other strategy could you have used and why?

(5) Answers Part a:
[1] is_member(Joe)
[2] is_member(Sally)
[3] is_member(Bill)
[4] is_member( Ellen)
[5] married(Joe,Sally)
[6] sibling(Bill,Ellen)
[7] (∀x)(∀y){married(x,y) ∧ is_member(x)} → is_member(y)}
[8] last_meeting(Joe)
G1:last_meeting(Sally)
G2:¬(∃y)married( Ellen, y)

(2) Answers Part b:
[9] (∀x)(∀y){married(x,y) ∧ last_meeting(x)} → last_meeting(y)}
[10] ( ∀ x)( ∀ y){married(x,y) → married(y,x)}
[11] ( ∀ x)( ∀ y){married(x,y) → ¬sibling(x,y)}
[12] ( ∀ x)( ∀ y)( ∀ z){married(x,y) → ¬married(x,z)}

(5) Answers Part c:
[1] is_member(Joe)
[2] is_member(Sally)
[3] is_member(Bill)
[4] is_member( Ellen)
[5] married(Joe,Sally)
[6] sibling(Bill,Ellen)
[7] ¬married(x,y) ∨ ¬is_member(x) ∨ is_member(y)
[8] last_meeting(Joe)
[9] ¬married(x,y) ∨ ¬last_meeting(x) ∨ last_meeting(y)
\[
\begin{align*}
[10] & \quad \neg\text{married}(x_{10}, y_{10}) \lor \text{married}(y_{10}, x_{10}) \\
[11] & \quad \neg\text{married}(x_{11}, y_{11}) \lor \neg\text{sibling}(x_{11}, y_{11}) \\
[12] & \quad \neg\text{married}(x_{12}, y_{12}) \lor \neg\text{married}(x_{12}, z_{12}) \\
\neg G1: & \quad \neg\text{last}\_\text{meeting}(Sally) \\
\neg G2: & \quad \text{married}(Ellen, Joe) \lor \text{married}(Ellen, Bill)
\end{align*}
\]

II. Resolution Refutation (continued)

(10) Refutation Graph Part d:

\[
\begin{align*}
\mathcal{R}_1 &= [\neg G1] \text{ with } [9] \quad \neg\text{married}(x_9, Sally) \lor \neg\text{last}\_\text{meeting}(x_9) \{Sally/y_9 \} \\
\mathcal{R}_2 &= \mathcal{R}_1 \text{ with } [8] \quad \neg\text{married}(Joe, Sally) \{Joe/x_9 \} \\
\mathcal{R}_3 &= \mathcal{R}_2 \text{ with } 5 \quad \text{nil} \\
\mathcal{R}_4 &= [\neg G2] \text{ with } [10] \quad \text{married}(Bill, Ellen) \lor \text{married}(Ellen, Joe) \{Ellen/x_{10}, Bill/y_{10} \} \\
\mathcal{R}_5 &= \mathcal{R}_4 \text{ with } [11] \quad \text{sibling}(Bill, Ellen) \lor \text{married}(Ellen, Joe) \{Bill/x_{11}, Ellen/y_{11} \} \\
\mathcal{R}_6 &= \mathcal{R}_5 \text{ with } [6] \quad \text{married}(Ellen, Joe) \\
\mathcal{R}_7 &= \mathcal{R}_6 \text{ with } [10] \quad \text{married}(Joe, Ellen) \{Ellen/x_{10}', Joe/y_{10}' \} \\
\mathcal{R}_8 &= \mathcal{R}_7 \text{ with } [12] \quad \text{married}(Joe, z_{12}) \{Joe/x_{12}, Ellen/y_{12} \} \\
\mathcal{R}_9 &= \mathcal{R}_8 \text{ with } [5] \quad \text{nil} \{Sally/z_{12} \}
\end{align*}
\]

Consistency Check

\[
\begin{align*}
U_1 &= [Sally, Joe, Ellen, Bill, Ellen, Ellen, Joe, Ellen, Sally] \\
U_2 &= [y_9, x_9, x_{10}, y_{10}, x_{11}, y_{11}, x_{10}', y_{10}', x_{12}, y_{12}, z_{12}] \\
U_1 &= U_2[y_9, Joe/x_9, Ellen/x_{10}, Bill/y_{10}, Bill/x_{11}, Ellen/y_{11}, \text{Ellen}/x_{10}', \text{Joe}/y_{10}', \text{Joe}/x_{12}, \text{Ellen}/y_{12}, \text{Sally}/z_{12}] \\
\text{Since } U_1 \text{ & } U_2 \text{ unify, then the substitutions are consistent}
\end{align*}
\]
(3) Answer Part e: My strategy is **Set-of-Support**. Every resolvent $\mathcal{R}_1 - \mathcal{R}_9$ comes from the negation of the wff to be proved. Note $\neg G2$ is not Horne. I could have used ancestry-filtered or breadth-first because they are complete strategies.
III. Heuristic Search

The following figure shows a search tree with the state indicated by the tuple inside parentheses. A letter indicates the state name and the integer indicates the estimated cost for finding a solution from that state (a cost of 0 indicates a goal state). Using the Graph-Search algorithm discussed in class, give the algorithm steps using (1) breadth-first search. How many nodes did breadth-first expand? Repeat using (2) depth-first search. How many nodes did depth-first expand? Repeat using (3) heuristic search (you MUST specify a rule to break ties). How many nodes did heuristic search expand? Repeat using (4) A* search. How many nodes did A* expand? You must clearly justify your answer(s). "Feelings" or "intuition" are not good/sound reasons. NO JUSTIFICATION <==> NO CREDIT. You must give me the details of each step of the algorithm in order to receive any credit for each case. Can any of these algorithms ever find N as a solution? Explain

Start: Open={A} Closed={} G={} M={} f(n)=g(n)+h(n) where g(n)=depth(n) & h(n)=heuristic fcn

Breadth First: append M at the end of the open list & f(n)=null.

1. The algorithm selects A and expands A (applies Γ) in order to obtain M={B,C}
   \( n_1=B \); \( n_2=C \); Open={B,C}, Closed={A}, G={A,B,C}, \( f(n_1)=1 \); \( f(n_2)=1 \)
2. The algorithm expands B in order to obtain M={D,E}
   \( n_3=D \); \( n_4=E \); Open={C,D,E}, Closed={A,B}, G={A,B,C,D,E}, \( f(n_3)=1 \); \( f(n_4)=1 \)
3. The algorithm expands C in order to obtain M={F,G}
   \( n_5=F \); \( n_6=G \); Open={D,E,F,G}, Closed={A,B,C}, G={A,B,C,D,E,F,G}, \( f(n_5)=1 \); \( f(n_6)=1 \)
4. The algorithm expands D in order to obtain M={H,I}
   \( n_7=H \); \( n_8=I \); Open={E,F,G,H,I}, Closed={A,B,C,D}, G={A,B,C,D,E,F,G,H,I}, \( f(n_7)=1 \); \( f(n_8)=1 \)
5. The algorithm expands E in order to obtain M={J,K}, \( n_9=J \); \( n_{10}=K \)
   Open={F,G,H,I,J,K}, Closed={A,B,C,D,E}, G={A,B,C,D,E,F,G,H,I,J,K}, \( f(n_9)=1 \); \( f(n_{10})=1 \)
6. The algorithm expands F in order to obtain M={L,M}, \( n_{11}=L \); \( n_{12}=M \), \( f(n_{11})=1 \); \( f(n_{12})=1 \)
7. The algorithm expands G in order to obtain M={N,P}, \( n_{13}=N \); \( n_{14}=P \), \( f(n_{13})=1 \); \( f(n_{14})=1 \)
8. The algorithm expands H in order to obtain M={}
9. The algorithm expands I in order to obtain M={}

Algorithm Details: You can use algorithm graphsearch for everything
Closed={A,B,C,D,E,F,G,H,I}  
10. The algorithm expands J in order to obtain M={}  
Closed={A,B,C,D,E,F,G,H,I,J}  
11. The algorithm expands K in order to obtain M={},  
G={A,B,C,D,E,F,G,H,I,J,K,L,M,P}  
K is a solution exit w/ success. BFS expands Closed={A,B,C,D,E,F,G,H,I,J,K} 11 nodes
III. Heuristic Search. (continued)

**Depth-First:** Append $M$ at the front of the open list with $f(n) = \text{null}$ (alternatively use $f(n) = \text{depth}(n)$).

1. The algorithm selects $A$ and expands $A$ (applies $\Gamma$) in order to obtain $M = \{B, C\}$
   
   $n_1 = B; n_2 = C$; Open = \{B, C\}, Closed = \{A\}, $G = \{A, B, C\}$, $f(n_1) = 1$; $f(n_2) = 1$

2. The algorithm expands $B$ in order to obtain $M = \{D, E\}$
   
   $n_3 = D; n_4 = E$; Open = \{D, E\}, Closed = \{A, B\}, $G = \{A, B, C, D, E\}$, $f(n_3) = 2$; $f(n_4) = 2$

3. The algorithm expands $D$ in order to obtain $M = \{H, I\}$
   
   $n_5 = H; n_6 = I$; Open = \{H, I, E, C\}, Closed = \{A, B, D\}, $G = \{A, B, C, D, E, H, I\}$, $f(n_5) = 3$; $f(n_6) = 3$

4. The algorithm expands $H$ in order to obtain $M = \{\}$, $G = \{A, B, C, D, E, H, I\}$
   
   Open = \{I, E, C\}, $G = $Closed = \{A, B, D, H\}

5. The algorithm expands $I$ in order to obtain $M = \{\}$, $G = \{A, B, C, D, E, H, I, E, J\}$

6. The algorithm expands $E$ in order to obtain $M = \{J, K\}$
   
   $n_7 = J; n_8 = K$; Open = \{J, K, C\}, Closed = \{A, B, D, H, I, E\}, $G = \{A, B, C, D, E, H, I, J, K\}$, $f(n_7) = 3$; $f(n_8) = 3$

7. The algorithm expands $J$ in order to obtain $M = \{\}$
   
   Open = \{K, C\}, Closed = \{A, B, D, H, I, E, J\}, $G = \{A, B, C, D, E, H, I, J, K\}$

8. The algorithm expands $K$ in order to obtain $M = \{\}$

K is a solution and the algorithm terminates. DFS expands Closed = \{A, B, D, H, I, E, J, K\} 8 nodes

**Heuristic-Search:** Use the function $f(n) = h(n)$ and sort the open list using $f$ values, FIFO.

1. The algorithm selects $A$ and expands $A$ (applies $\Gamma$) in order to obtain $M = \{B, C\}$
   
   $n_1 = B; n_2 = C$; Open = \{B, C\}, Closed = \{A\}, $G = \{A, B, C\}$, $f(n_1) = 20$; $f(n_2) = 10$, Open = \{C, B\}

2. The algorithm expands $C$ in order to obtain $M = \{F, G\}$
   
   $n_3 = F; n_4 = G$; Open = \{F, G\}, Closed = \{A, C\}, $G = \{A, B, C, F, G\}$, $f(n_3) = 8$; $f(n_4) = 20$, Open = \{F, B, G\}

3. The algorithm expands $F$ in order to obtain $M = \{L, M\}$, $G = \{A, B, C, F, G, L, M\}$
   
   $n_5 = L; n_6 = M$; Open = \{L, M, G, B\}, Closed = \{A, C, F\}, $f(n_5) = 27$; $f(n_6) = 22$, Open = \{B, G, M\}

4. The algorithm expands $B$ in order to obtain $M = \{D, E\}$, $G = \{A, B, C, F, G, L, M, D, E\}$
   
   $n_7 = D; n_8 = E$; Open = \{D, G, M, L\}, Closed = \{A, C, F, B\}, $f(n_7) = 13$; $f(n_8) = 12$, Open = \{E, D, G, M, L\}

5. The algorithm expands $E$ in order to obtain $M = \{J, K\}$, $G = \{A, B, C, F, G, L, M, D, E, J, K\}$
   
   $n_9 = J; n_{10} = K$; Open = \{J, D, G, M\}, Closed = \{A, C, F, B, E\}, $f(n_9) = 2$; $f(n_{10}) = 0$

6. The algorithm expands $K$ in order to obtain $M = \{\}$, $G = \{A, B, C, F, G, L, M, D, E, J, K\}$

K is a solution and the algorithm terminates. Heuristic search expands Closed = \{A, C, F, B, E, K\} 6 nodes

**Heuristic-Search:** Use the function $f(n) = h(n)$ and sort the open list using $f$ values, LIFO.

1. The algorithm selects $A$ and expands $A$ (applies $\Gamma$) in order to obtain $M = \{B, C\}$
   
   $n_1 = B; n_2 = C$; Open = \{B, C\}, Closed = \{A\}, $G = \{A, B, C\}$, $f(n_1) = 20$; $f(n_2) = 10$, Open = \{C, B\}

2. The algorithm expands $C$ in order to obtain $M = \{F, G\}$
   
   $n_3 = F; n_4 = G$; Open = \{F, G\}, Closed = \{A, C\}, $G = \{A, B, C, F, G\}$, $f(n_3) = 8$; $f(n_4) = 20$, Open = \{F, B, G\}

3. The algorithm expands $F$ in order to obtain $M = \{L, M\}$, $G = \{A, B, C, F, G, L, M\}$; $n_5 = L; n_6 = M$
   
   Open = \{M, L, G, B\}, Closed = \{A, C, F\}, $f(n_5) = 27$; $f(n_6) = 22$, Open = \{B, G\}

4. The algorithm expands $G$ in order to obtain $M = \{N, P\}$, $G = \{A, B, C, F, G, L, M, N, P\}$; $n_7 = N; n_8 = P$
   
   Open = \{N, P\}, Closed = \{A, C, F, G\}, $f(n_7) = 0$; $f(n_8) = 9$, Open = \{N, P\}

5. The algorithm expands $N$ in order to obtain $M = \{\}$, $G = \{A, B, C, F, G, L, M, N, P\}$
N is a solution and the algorithm terminates. Heuristic search expands \( \text{Closed} = \{A,C,F,G,N\} \) 5 nodes.

**A* Search:** Uses \( f(n) = g(n) + h(n) \) where \( g(n) = \text{depth}(n) \) & \( h(n) = \text{cost} \) and sort the open list using \( f \)

1. The algorithm selects \( A \) and expands \( A \) (applies \( \Gamma \)) in order to obtain \( M = \{B,C\} \)
   
   \( n_1 = B; \ n_2 = C; \ \text{Open} = \{B,C\}, \ \text{Closed} = \{A\}, \ \text{G} = \{A,B,C\}, \ f(n_1) = 1+20; \ f(n_2) = 1+10, \ \text{Open} = \{C_{11}, B_{21}\} \)

2. The algorithm expands \( C \) in order to obtain \( M = \{F,G\} \)
   
   \( n_3 = F; \ n_4 = G; \ \text{Open} = \{F,B,G\}, \ \text{Closed} = \{A,C\}, \ G = \{A,B,C,F,G\}, \ f(n_3) = 2+8; \ f(n_4) = 2+20, \ \text{Open} = \{F_{11}, B_{21}, G_{22}\} \)

3. The algorithm expands \( F \) in order to obtain \( M = \{L,M\} \)
   
   \( n_5 = L; \ n_6 = M; \ \text{Open} = \{B,G,M,L\}, \ \text{Closed} = \{A,C,F\}, \ f(n_5) = 3+27; \ f(n_6) = 3+22, \ \text{Open} = \{B_{21}, G_{22}, M_{25}, L_{30}\} \)

4. The algorithm expands \( B \) in order to obtain \( M = \{D,E\} \)
   
   \( n_7 = D; \ n_8 = E; \ \text{Open} = \{E,D,G,M,L\}, \ \text{Closed} = \{A,C,F,B\}, \ f(n_7) = 2+13; \ f(n_8) = 2+12, \ \text{Open} = \{E_{14}, D_{15}, G_{22}, M_{25}, L_{30}\} \)

5. The algorithm expands \( E \) in order to obtain \( M = \{J,K\} \)
   
   \( n_9 = J; \ n_{10} = K; \ \text{Open} = \{J,K,D,G,M,L\}, \ \text{Closed} = \{A,C,F,B,E\}, \ f(n_9) = 3+2; \ f(n_{10}) = 3+0, \ \text{Open} = \{K_{35}, J_{15}, D_{15}, G_{22}, M_{25}, L_{30}\} \)

6. The algorithm expands \( K \) in order to obtain \( M = \{\} \)
   
   \( K \) is a solution and the algorithm terminates. Heuristic search expands \( \text{Closed} = \{A,C,F,B,E,K\} \) 6 nodes.

\( N \) is found by heuristic search with LIFO: \( \{A,C,F,G\} \) comes before \( \{A,C,F,B\} \) & \( h(B) = h(G) = 20 \) & LIFO orders \( G \) before \( B \).

We wish to find the last coach in a short list of UF coaches. Using Resolution Refutation deduce the following computation to obtain a value for the goal (3 pts) by performing a consistent Refutation Trace (17 pts) for the goal and prove (or provide a good argument) its consistency (5 pts.) Make sure your resolution refutation trace is clearly marked and it follows a complete strategy.

Facts:
F₁: last(cons(U,nil),U).

Rules:
R₁: last(X,Y) → last(cons(W,X),Y)

Goal: (∃z) last(cons(steve, cons(ron, cons(urban,nil))), z)

{Note: If you prefer, you may use the notation last( (steve ron urban), z).}

Required: Give the resolution trace (17 pts), show the substitutions are consistent (5pts), and obtain the value of the goal (3 pts).

(1 pts) I am using Set-of-Support which is a complete strategy
F₁: last(cons(U,nil),U).
R₁: ~ last(X,Y) v last(cons(W,X),Y)
~Goal: ~last( (steve ron urban), z)

(4 pts) ℜ₁=ℜ{~Goal,R₁}: ~last(X,Y) {steve/W, (ron urban)/X, z/Y}
ℜ₁: ~last( (ron urban), z)

(4 pts) ℜ₂=ℜ{ℜ₁,R₁}: ~last(X’,Y’) {ron/W’, (urban)/X’, z/Y’}
ℜ₂: ~last((urban), z)

(4 pts) ℜ₃=ℜ{ℜ₂,F₁}: nil {urban/U, U/z}

(4 pts) Therefore z=U=Y’=Y=urban; X’=(urban); W’=ron; X=(ron urban); W=steve

(3) Answer: (∃z)last(cons(steve, cons(ron,cons(urban,nil))), z) is true with z = urban

(5) Substitutions will be consistent because I changed variables every time I re-used any rules and all the variables were originally standardized apart. To prove consistency we assemble all the numerators in a set, say, U₁, and all the denominators in a set, say, U₂ and show that U₁=U₂σ and σ≠null. U₁=[steve,(ron,urban),z,ron,(urban),z,urban,U], U₂=[W,X,Y,W’,X’,Y’,U,z] and U₁=U₂σ σ={steve/W, (ron urban)/X, z/Y, ron/W’, (urban)/X’, z/Y’, urban/U, U/z} and σ≠null.