Midterm 2 Exam

CSCI 561 Fall 2017: Artificial Intelligence

Student ID: | | | | | | | | | | |

Last Name: ____

First Name: ____

USC email: | | | | | | | | | @usc.edu

Instructions:

1. Date: **10/30/2017 from 8:00 pm – 9:50 pm**
2. Maximum credits/points/percentage for this midterm: 100
3. The percentages for each question are indicated in square brackets [ ] near the question.
4. **No books** (or any other material) are allowed.
5. **Write down your name, student ID and USC email address.**
6. **Your exam will be scanned and uploaded online.**
7. **Write within the boxes provided for your answers.**
8. **Do NOT write on the 2D barcode.**
9. **The back of the pages will not be graded. You may use it for scratch paper.**
10. No questions during the exam. **If something is unclear to you, write that in your exam.**
11. **Be brief: a few words are often enough if they are precise and use the correct vocabulary studied in class.**
12. When finished, raise completed exam sheets until approached by proctor.
13. **Adhere to the Academic Integrity code.**

Problems	**100 Percent total**
1- General AI Knowledge and Application	20
2- FOL to CNF Conversion	10
3- Propositional Proof	15
4- Fuzzy logic	15
5- Block World	20
6- FOL resolution proof	20

1. [20%] General AI Knowledge and Application

True or False [14%]: 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	T	F
2	T	F
3	T	F
4	T	F
5	T	F
6	T	F
7	T	F

1- Closed-world assumption is a solution to the Frame Problem. False

2- If $\beta \Rightarrow \alpha$, then $M(\alpha) \subseteq M(\beta)$. False

3- Backward chaining is sound and complete. False

4- $(\neg x \lor y) \lor (x \land \neg y)$ is valid. True

5- We get $P\ (John, z)$ after resolving $P(x, z) \lor Q(x, z)$ and $\neg Q\ (John, Jack)$. False

6- For any α, False entails α. True

7- Propositional logic does not scale to the environments of unbounded size. True

Multiple Choice [6%]: Each question has <u>one or more</u> correct choices. Check the boxes of all correct choices and leave the boxes of wrong choices blank. Please note that there will be no partial credit and you will receive full credit if and only if you choose all the correct choices and none of the wrong choices.

8- [2%] What is the English equivalent of "$\neg\exists x\ (Human(x) \land Perfect(x))$"?

- [ ] Not every human is perfect.
- [x] There is no perfect human.
- [x] Being a human, implies not being perfect.
- [ ] Some humans are perfect.

9- [2%] Which of the following sentences are necessarily satisfiable?

- [ ] $P(John, x)$
- [x] $A \Leftrightarrow B$
- [x] $(A \land \neg A) \Rightarrow True$
- [ ] $(A \lor \neg A) \Rightarrow False$

9- [2%] What can a planning graph be used for?

- [x] Plan generation
- [ ] Partial-order planning
- [x] Heuristic estimation
- [ ] Determining whether the goal state is reachable from the start state

Note that all students will get credit for this question

2. [10%] FOL to CNF Conversion

Convert the following FOL statement to CNF. Write down all the steps involved.

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

1. Eliminate implications [2%]

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

2. Move $\neg$ inwards [2%]

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

3. Standardize variables [1%]

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

4. Eliminate $\exists$ [2%]

$\forall x\, [(M(F(x)) \wedge \forall z\, \neg N(z,F(x))) \vee (L(x,G(x)) \wedge \neg Q(x,G(x)))]$

5. Drop $\forall$ [1%]

$(M(F(x)) \wedge \neg N(z,F(x))) \vee (L(x,G(x)) \wedge \neg Q(x,G(x)))$

6. Distribute $\wedge$ over $\vee$ [2%]

$M(F(x)) \vee L(x,G(x))$ → clause 1

$\wedge$

$\neg N(z,F(x)) \vee L(x,G(x))$ → clause 2

$\wedge$

$M(F(x)) \vee \neg Q(x,G(x))$ → clause 3

$\wedge$

$\neg N(z,F(x)) \vee \neg Q(x,G(x))$ → clause 4

Number of points for each step is written in front of it;

No Partial Credits.

3. [15%] Propositional proof

a. [4%] Please **Write down and indicate** Modus Ponens(Implication-Elimination), And-Introduction, Unit-Resolution and Resolution inference rules. **(Totally 4 inference rules)**

Implication-Elimination:

$$\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta}$$

And-Introduction:

$$\frac{\alpha_1, \ \alpha_2, \ \ldots, \ \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}$$

Unit-Resolution:

$$\frac{\alpha \vee \beta, \quad \neg\beta}{\alpha}$$

Resolution:

$$\frac{\alpha \vee \beta, \quad \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

OR

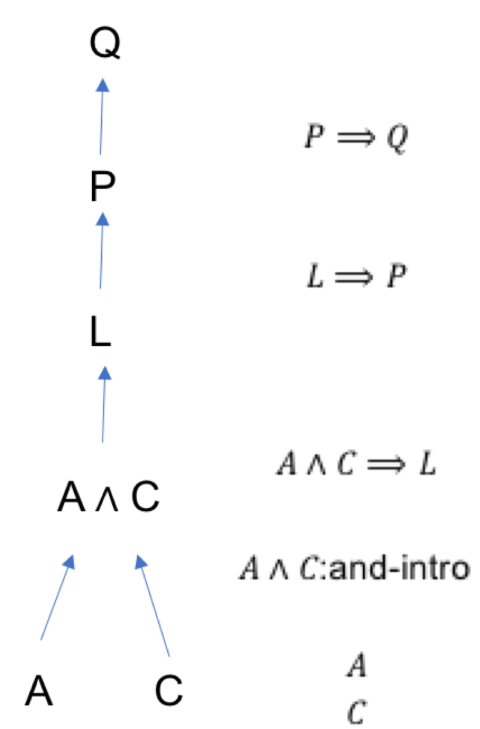
$$\frac{\neg\alpha \Rightarrow \beta, \quad \beta \Rightarrow \gamma}{\neg\alpha \Rightarrow \gamma}$$

Each inference rules is worth one point

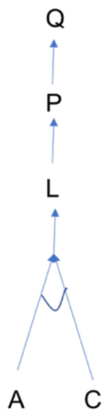
b. [11%] Consider the following propositional logic knowledge base:

$$A \wedge B \implies Q$$
$$L \implies P$$
$$A \wedge C \implies L$$
$$P \implies Q$$
$$A$$
$$C$$

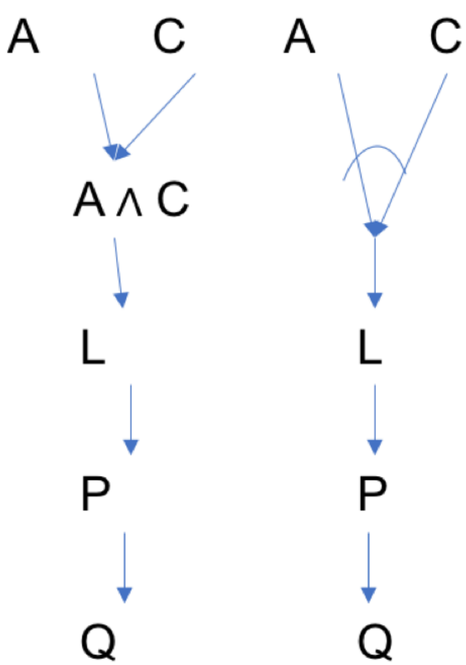
Using **backward chaining**, please prove **Q**. Please **only** use the **Modus Ponens and And-Introduction inference rules**. You will lose points if you use any other rule. Please draw a graph that shows your backward-chaining proof, showing clearly all Modus Ponens and And-Introduction steps, to which sentences they apply, and which sentences result. The query sequence should be from top to bottom in your graph.



If students lose any step or use wrong sentence, then -1
If students lose any inference rules steps, then -1
If students draw graph like below and without mention the and-introduction rules, then -1.
Otherwise one error -1



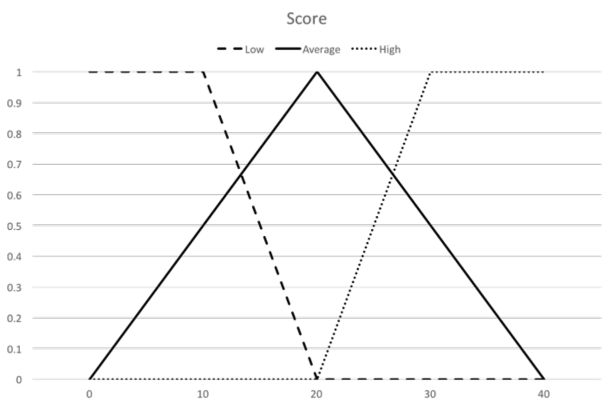
If students draw the graph upside down like two graphs below, then I assume that they used forward chaining(Because I mentioned **The query sequence should be from top to bottom in your graph**)
Only give them half point in this situation, If they write right inference steps

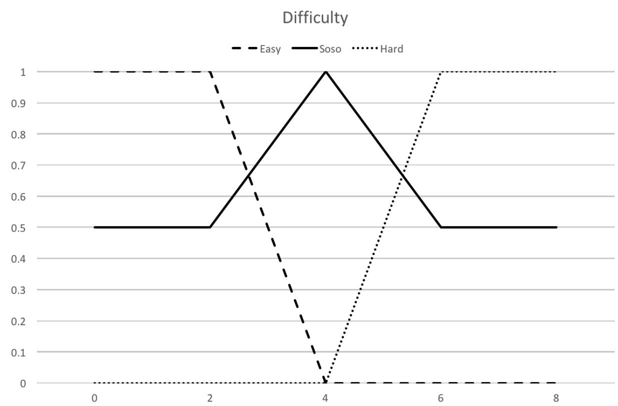


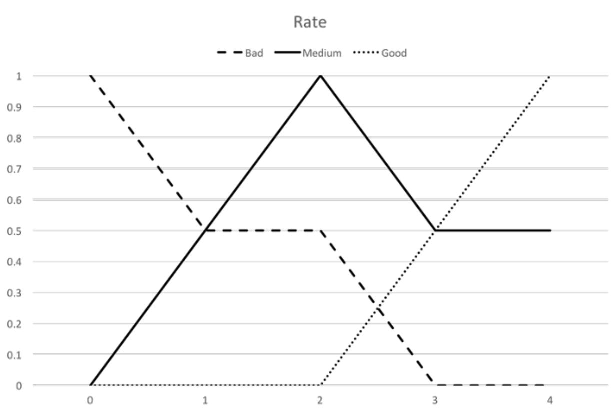
---

4. [15%] Fuzzy logic

A student wants to use fuzzy logic inference system to rate the course he registered this semester. To decide **rate** of a course, he will only consider the **difficulty** of the course content and the midterm exam **score** of the course. (He doesn't know the final exam score before rating ☺). Thus, only consider three variables, x1 for score, x2 for difficulty and y for rate. The membership functions for Low, Average and High **score**; Easy, Soso and Hard **difficulty**; Bad, Medium and Good **rate** are shown below:



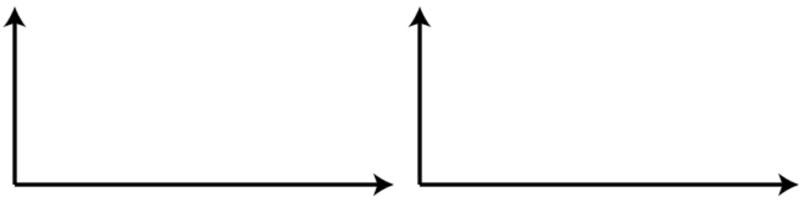




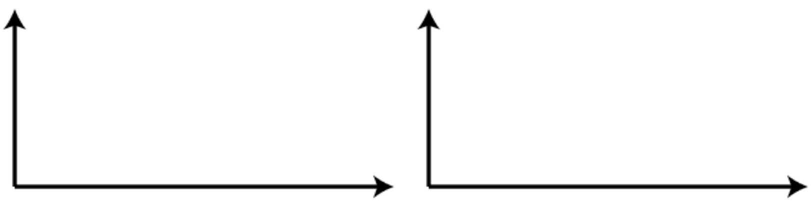
Rule 1: If Score is High, then Rate Good.
Rule 2: If Score is Low, then Rate Bad.
Rule 3: If Score is Average or Difficulty is Easy, then Rate Good.
Rule 4: If Score is Average or Difficulty is Soso, then Rate Medium.
Rule 5: If Score is Average and Difficulty is Hard, then Rate Bad.
Now, given a course experience {x1, x2}= {15, 5}

a. [14%] Draw graphs to show how the fuzzy rules evaluate for the course experience. Show the **clipped values** in your drawings.

Rule 1:

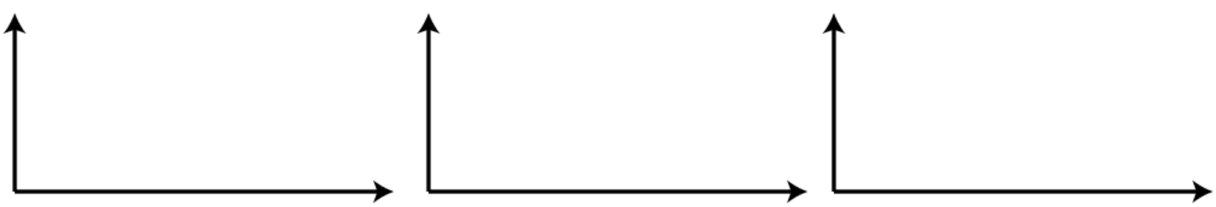


Rule 2:

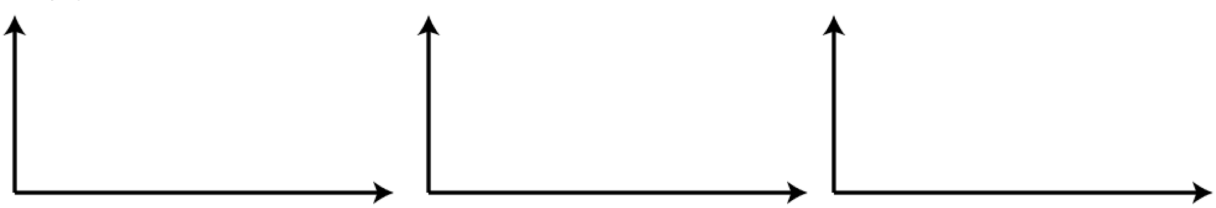


Rule 3:

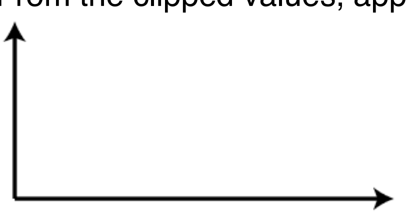
Rule 4:

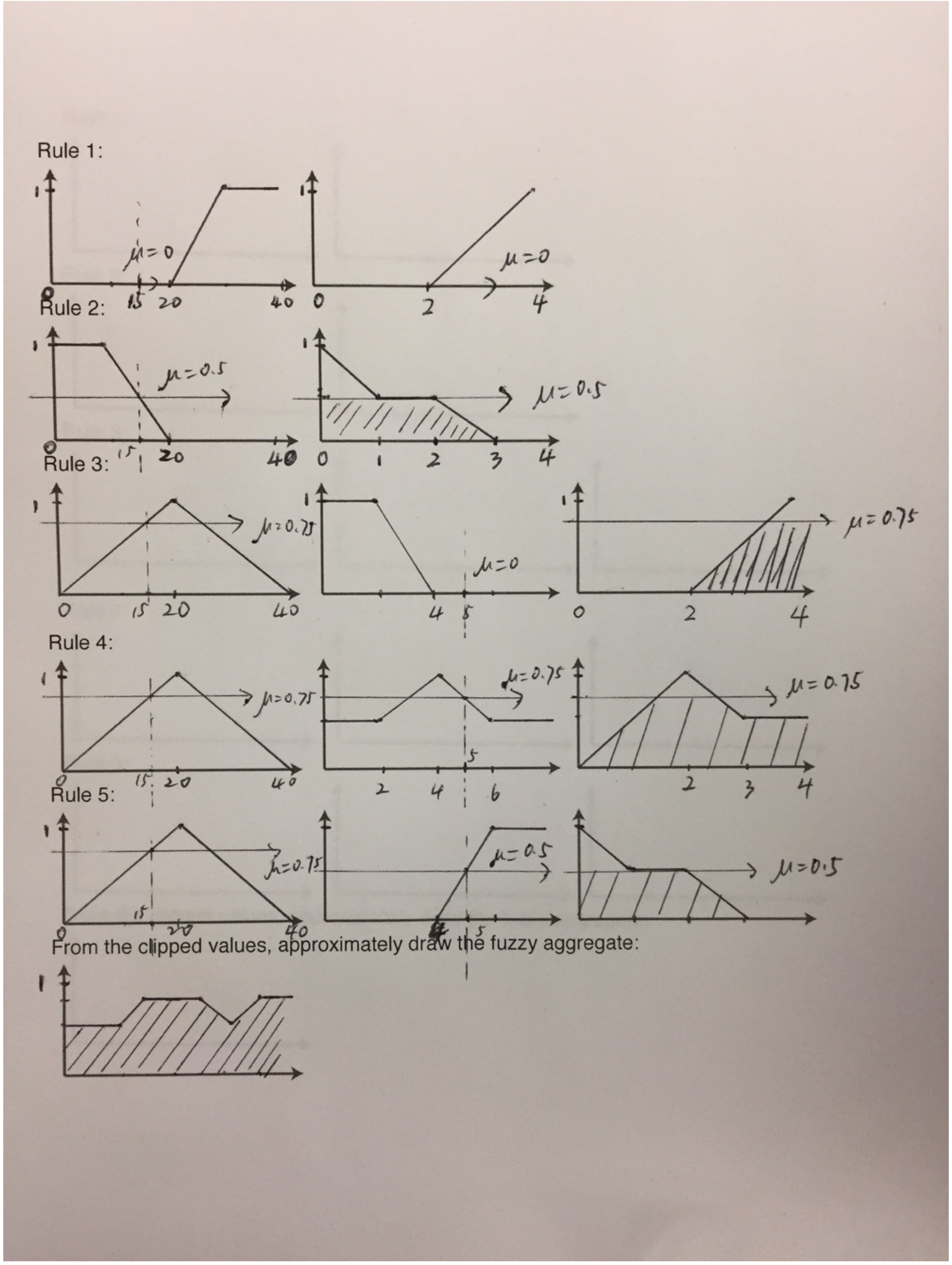


Rule 5:



From the clipped values, approximately draw the fuzzy aggregate:





Student should show right clipped value and right graph(right function lines), it's ok if they don't indicate the function lines' name.
One graph worth 1 points. Without showing the correct clipped value or function line, -1

b. [1%] Using the center of mass, what's the rate of this course mostly like? (Please check one of the following boxes):

☐ Lower than 2.
■ Higher than 2.
☐ Equal to 2.

5. [20%] Block World

In a block world, we usually use On(x, y) and Clear(x) to describe basic states. On(x, y) means block x is placed on y. Clear(x) means we can put something on x. Alice and Bob have 4 blocks on the table. The initial state can be described as:

Clear(B), Clear(D), On(B, A), On(D, Table), On(C, Table), On(A, C)

They want to build a tower in this state:

On(A, D), On(B, A), On(C, B)

Alice designs two operators to move them:

• ***PutOn(x, y)***: Put block x onto y.
• ***PutOnTable(y)*** Put block x onto the table.

a. [6%] Please write the STRIPS form of the operators.

- Action: $PutOn(x, y)$
- Preconditions: $Clear(x), Clear(y), On(x, z)$
- Effect: $On(x, y), Clear(z), \sim Clear(y), \sim On(x, z)$
- Action: $PutOnTable(x)$
- Preconditions: $Clear(x), On(x, z)$
- Effect: $On(x, Table), Clear(z), \sim On(x, z)$

Each Preconditions or Effect is worth 1.5 point.

b. [8%] Alice tries to use forward search starting from the initial state. She takes a step PutOnTable(B). Bob chooses to use backward search starting from the goal state. He takes a step PutOn(C, B) back. Write down the states Alice and Bob have respectively.

- Alice:

  $On(A, C), On(B, table), On(C, table), On(D, table), Clear(A), Clear(B), Clear(D)$

- Bob:

  $Clear(B), Clear(C), On(B, A), On(A, D), On(C, table), On(D, table)$

Each error -1, until they got 0.

c. [6%] Bob thinks the state describers On and Clear are not enough. There might exist some inconsistency in operators. Please find out why he claims that then fix it by adding one new state describer and modifying the STRIPS operators you presented in Question (a).

When we apply $PutOn(A, table)$, $On(A, table)$ will be deleted and added at the same time. It may bring inconsistency. We should make sure table is a constant and will not be involved in operators. We can define $Block(x)$ to identify a block then add $Block(x), Block(y)$ into preconditions.

*If the answer mentions that table is a constant and should not be any parameter in $PutOn$ and $PutOnTable$, it can be considered as correct. But only saying "cannot put the table onto a block" is incomplete because using table as the second parameter in $PutOn$ should also be prevented.

This reason is worth 3 points and the way to fixing it is worth 3 points.

6. [20%] FOL Resolution Proof

a. [8%] Given the following statements:

1. Marcus was a man.
2. Marcus was a Pompeian.
3. All pompeians were Romans.
4. Caesar was a ruler.
5. All Romans were either loyal to Caesar or hated him.
6. Everyone is loyal to someone.
7. People only try to assassinate rulers they are not loyal to.
8. Marcus tried to assassinate Caesar.

Hints:
Here we provide FOL for statement 7 and 8 as the followings:

7. FOL: $\neg man(x) \vee \neg ruler(y) \vee \neg tryAssassinate(x, y) \vee \neg loyalto(x, y)$
8. FOL: $\neg tryAssassinate(Marcus, Caesar)$

Please translate the following statements into First Order Logic. You also need to convert it to CNF if it's convertible.

1. [1%] Marcus was a man.

man(Marcus)

2. [1%] Marcus was a Pompeian.

Pompeian(Marcus)

3. [1%] All pompeians were Romans.

Pompeian(x) =>Roman(x)
CNF: ~Pompeian(x) V Roman(x)

4. [1%] Caesar was a ruler.

ruler(caesar)

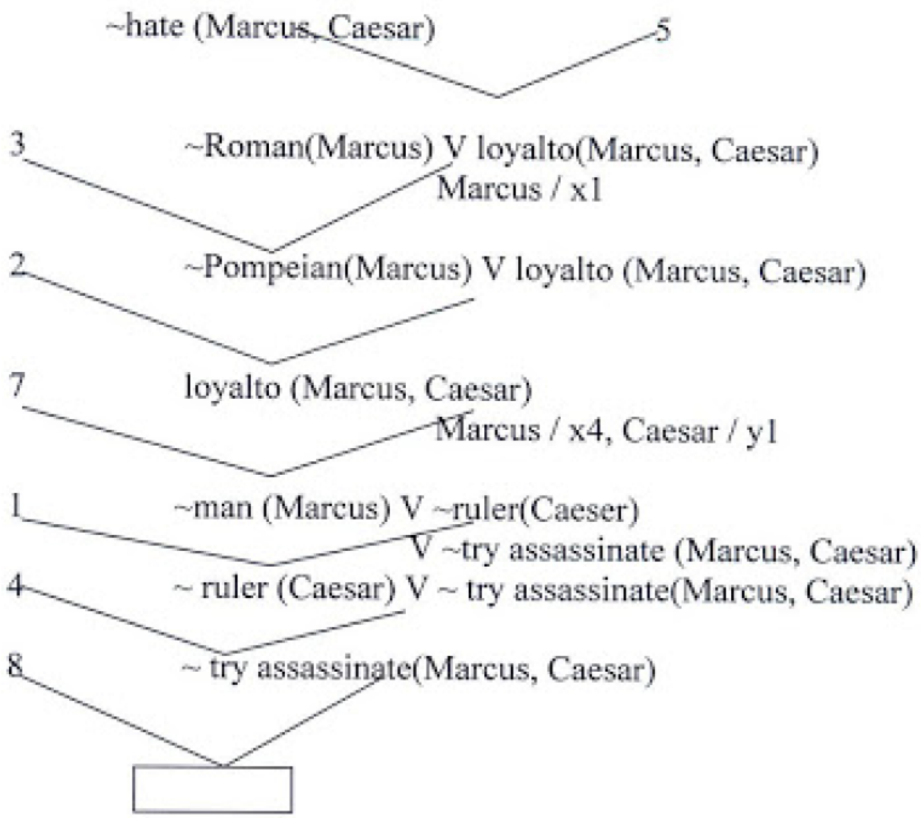
5. [2%] All Romans were either loyal to Caesar or hated him.

~Roman(x) V loyalto(x, Caesar) V hate(x, Caesar)
Each error -1, until they got 0.
.

6. [2%] Everyone is loyal to someone.

loyalto(x, f(x))
Each error -1, until they got 0.

b. [12%] With the hints and results from 6a, please Only use the resolution inference rule to prove **hate(Marcus, Caesar).** You may or may not use all of the clauses. You will lose points if you use any other rule. Please clearly show which sentences are resolved and what results. If unification is used at any step, please show the substitution, or you will lose points for each missing substitution.



The empty clause shows that ~hate (Marcus, Caesar) produces a contradiction or hate(Marcus, Caesar) will not produce a contradiction with the known statements.

Each error -1, until they got 0.