

Problem Set 3: Logic

EECS 492 Winter 2017
Due Monday, February 20th, 2017 11:55 p.m.

To help you study for the midterm sample solutions to this problem set will be posted earlier, so

NO LATE SUBMISSIONS WILL BE ACCEPTED.

Notation

There are many variations on logical notation. For the sanity of our graders, please adhere to the standards in the following table. If it is feasible in your word processor, you should use the “preferred” notation, but in some cases it may be necessary to resort to the “acceptable” version.

meaning	preferred	acceptable
logical “not”	$\neg x$	$\sim x$
logical “and”	$x \wedge y$	$x \wedge y$
logical “or”	$x \vee y$	$x \vee y$
logical implication	$x \Rightarrow y$	$x \Rightarrow y$
logical bi-implication	$x \Leftrightarrow y$	$x \Leftrightarrow y$
existential quantifier	$\exists x$	Exists(x), E(x)
universal quantifier	$\forall x$	Forall(x), A(x)

1 Logic Concepts and Manipulation (20 points)

1.1 Validity and Satisfiability

Use model checking via truth tables to determine whether the following logical statements are valid, satisfiable, or unsatisfiable. In your solution file, include both the truth tables and the statements’ validity/satisfiability.

- $(P \Rightarrow Q) \wedge (P \wedge \neg Q)$
- $(P \Rightarrow Q) \vee (Q \vee \neg P)$
- $(Q \Rightarrow (P \vee R)) \wedge \neg[(P \vee R) \vee \neg Q]$

1.2 Conjunctive Normal Form

Convert each of the following first-order logic statements into conjunctive normal form. Show every step of your conversions and proceed carefully, lest early mistakes propagate through later steps!

a. $\forall y, z \ P(y) \Leftrightarrow Q(z)$

b. $\neg \forall x \neg [P(x) \vee Q(x)] \Rightarrow R(x)$

c. $\forall x \forall y \exists z [P(x, z) \Rightarrow Q(y, z)]$

2 Translation (25 points)

2.1 Propositional Logic

Translate the following English sentences into propositional logic statements. You may use the following propositions about planets: BiggerThanEarth, GasPlanet, HasWater, HasRings, SupportsLife, and Terrestrial.

- a. Planets that support life have water and are terrestrial.
- b. Terrestrial planets are not bigger than Earth.
- c. Gas planets with rings are bigger than Earth.
- d. Gas planets have rings but other planets do not.
- e. Planets are either terrestrial or gaseous (but not both).

2.2 Definitions

Consider the following English language sentences:

1. Rob fed Spot before Jim fed Rover.
2. Every dog chases a cat or a dog (or both).
3. There are at least two different cats.
4. Rob likes dogs that are bigger than any cat.
5. Jim fed some cat before Rover was fed.

- a. Define any constants, predicates, and functions that you will use to best represent the five sentences in First-Order Logic. Some examples are Likes(x,y) and Fed(x,y,z) (with feeder x, one being fed y, and time of feeding z).
- b. Translate each of the 5 sentences into First-Order Logic using your symbols from part (a).

3 Inference (25 points)

3.1 Unification

For each of the following pairs of legal first-order logic literals, find the most-general unifier. If none exists, explain why not. You should assume that all variables are universally quantified. Note that your task is to execute the unification algorithm, not to interpret the semantics of the logical statements.

- a. Hates(Jim, x) and Hates(y, Bob)
- b. Hates(Jim, x) and Loves(y, z)
- c. Sees(Owner(x), x, Home(Mittens)) and Sees(Owner(y), SpotHome(y))
- d. Older(Father(x), x) and Older(Father(y), Spot)
- e. Loves(Friend(Spot), x) and Loves(y, y)

3.2 Resolution-Refutation

The following information about University of Michigan classes can be represented in conjunctive normal form.

1. Classes with thicker books are harder.

$$S1 = \neg ThickerBook(x, y) \vee HarderClass(x, y)$$

2. Hardness of classes is transitive.

$$S2 = \neg HarderClass(u, v) \vee \neg HarderClass(v, w) \vee HarderClass(u, w)$$

3. Programming classes have thicker books than some other classes.

$$S3 = \neg ProgrammingClass(z) \vee ThickerBook(z, SomeOtherClass(z))$$

4. EECS 492 has a thicker book than EECS 376.

$$S4 = ThickerBook(EECS492, EECS376)$$

5. EECS 376 is a harder class than ENGR 101.

$$S5 = HarderClass(EECS376, ENGR101)$$

6. ENGR 101 is a programming class.

$$S6 = ProgrammingClass(ENGR101)$$

7. ECON 101 is not a programming class.

$$S7 = \neg ProgrammingClass(ECON101)$$

a. Use resolution-refutation (p.345) with an answer literal to find some class that EECS 492 is harder than. In other words, find a c such that $HarderClass(EECS492, c)$ is entailed. Show all steps in your work, including which sentences get resolved and how variables are bound at each step.

b. Continue this resolution-refutation process to find the other two values of c for which $HarderClass(EECS492, c)$ is entailed. Show all steps in your work.

4 Prolog (30 points)

For this task you will need access to the Prolog programming language. If your system does not have Prolog installed, you can download SWI Prolog at www.swi-prolog.org. If you're having trouble getting started, a good Prolog tutorial can be found at www.learnprolognow.org.

Consider the following First-Order Logic expressions

1. $\forall x \text{ University}(x) \wedge \text{Wealthy}(x) \wedge \text{GoodAt}(x, \text{Engr}) \Rightarrow \text{GoodAt}(x, \text{CS})$
 2. $\forall x \text{ University}(x) \wedge \text{focusOn}(x, \text{Engr}) \Rightarrow \text{GoodAt}(x, \text{Engr})$
 3. $\forall x \text{ University}(x) \wedge \text{focusOn}(x, \text{Football}) \Rightarrow \text{GoodAt}(x, \text{Football})$
 4. $\forall x \text{ University}(x) \wedge \text{GoodAt}(x, \text{Football}) \wedge \text{focusOn}(x, \text{Engr}) \Rightarrow \text{Wealthy}(x)$
 5. $\forall x \forall y \text{ University}(x) \wedge \text{Beat}(x, y, \text{Football}) \wedge \text{University}(y) \wedge \text{GoodAt}(y, \text{Football}) \Rightarrow \text{GoodAt}(x, \text{Football})$
 6. $\exists x \text{ University}(x) \wedge \text{Beat}(x, \text{Michigan}, \text{Football})$
 7. $\text{University}(\text{Michigan}) \wedge \text{University}(\text{MIT}) \wedge \text{University}(\text{OSU}) \wedge \text{University}(\text{Alabama})$
 8. $\text{focusOn}(\text{Alabama}, \text{Football}) \wedge \text{focusOn}(\text{MIT}, \text{Engr}) \wedge \text{focusOn}(\text{OSU}, \text{Football})$
- $\wedge \text{focusOn}(\text{Michigan}, \text{Football})$
9. $\text{beat}(\text{Michigan}, \text{OSU}, \text{Football})$

- a. Create a file where you convert the preceding expressions as exactly as possible into Prolog syntax. You can create one or more functions as necessary. Include this file (or a paste of it) with your assignment submission.
- b. Use Prolog to answer the queries: Who is good at football? and Who is good at Engr? Extract all answers. Include the output of Prolog with your assignment submission (cut and paste is fine).
- c. Now say we want to correct or change a statement in our knowledge base using **assert** and **retract**. Retract the statement $\text{focusOn}(\text{Michigan}, \text{Football})$ add the statement $\text{focusOn}(\text{Michigan}, \text{Engr})$ to your KB using **assert** and **retract**. Now extract all of the answers of asking rerunning the commands from (b). What new answers did you get and (briefly) why? What answers from part b did you no longer get and (briefly) why? Provide your prolog statements as well.
- d. What happens when you invert the order of premises for expression 5? Change (a copy of) your file from part (a), load it into a new instance of Prolog and run the same queries as in part b. Explain how (if at all) this changes what Prolog tells you, and why. Include the output of Prolog with your assignment submission (you don't need to include the slightly modified Prolog code).