

CS-480 Homework on Logic and Knowledge Representation

1. (10 pts) Which of the following compound propositional sentences are tautologies? You may use a truth table, but are not required to. To prove a sentence is NOT a tautology you only need to give one truth setting for p and q for which the compound sentence is false.

- a. $(p \rightarrow q) \rightarrow (q \rightarrow p)$
- b. $((p \rightarrow q) \wedge \neg q) \rightarrow \neg p$
- c. $q \rightarrow (\neg p \vee \neg q)$

2. (10 pts) Consider the compound sentence: $(\neg p \wedge \neg q) \wedge (\neg r \rightarrow p)$. Find an equivalent expression which uses only \wedge and \neg , and which is as short as possible.
3. (10 pts) Translate the following sentences of first-order logic into English, where $\text{Eat}(x,y)$ means x eats a y, $\text{Apple}(x)$ means x is an apple, and $\text{SeeDoctor}(x)$ means x sees a doctor.

$$(\forall d \exists a \text{ Eat}(d,a) \wedge \text{Apple}(a)) \Rightarrow (\neg \exists d \text{ SeeDoctor}(d))$$
$$\forall x ((\exists a \text{ Eat}(x,a) \wedge \text{Apple}(a)) \Rightarrow \text{SeeDoctor}(x))$$

4. (10 pts) Translate the following English sentence into (three statements of) first-order logic, where $\text{CanFool}(x,t)$ means that you can fool x at time t:

You can fool some of the people all the time, and you can fool all the people some of the time, but you can't fool all the people all the time.

5. (10 pts) Unification

Task: Give the most general unifier for each of the following pairs of expressions, or state why no unifier exists:

- (a) $\text{foo}(x,y)$ $\text{foo}(y,x)$
- (b) $\text{mother}(x,y)$ $\text{mother}(y,\text{father}(x))$
- (c) $\text{p}(x, y, z)$ $\text{p}(\text{q}(y), \text{r}(z), \text{foo})$

6. (20 pts) Backwards and Forwards Chaining

Use a first-order language with two constants “-1” and “1”, the unary function $s(x)$, denoting the “successor of x ”, and the binary predicate $\text{Less}(x, y)$, representing “ x is less than y ”.

Consider the following Horn database, with two rules labeled A and B and fact C:

- A. $\text{Less}(x, y) \Rightarrow \text{Less}(x, s(y))$
- B. $\text{Less}(s(x), y) \Rightarrow \text{Less}(x, y)$
- C. $\text{Less}(s(-1), 1)$

(a) Prove $\text{Less}(-1, s(1))$ by backward chaining (with substitution) on rules A and B and fact C. In each row of the proof, show the new subgoal, which previous goal and knowledge-base rule gave rise to it, and the substitution needed.

(b) Using forward chaining (and substitution), begin enumerating all facts implied by fact C, given rules A and B, until you prove $\text{Less}(-1, s(1))$. Show all the facts you generated, starting with fact C.

7. (30 pts) Consider the following statements:

1. If there is an economic downturn, there will be fewer jobs.
2. If there are fewer jobs and John Doe has a good resume, he will get a good job.
3. If there are not fewer jobs, John Doe will get a good job.
4. John Doe has a good resume.
5. There is an economic downturn.

Part a. Convert the above statements into propositional logic by assigning a propositional literal to each basic proposition:

- E = “there is an economic downturn”
- F = “there will be fewer jobs”
- R = “John Doe has a good resume”
- J = “John Doe will get a good job”

Part b. Use forward chaining to prove that John Doe will get a good job.

Part c. Explain why, if the knowledge base did not contain the last statement (there is an economic downturn), it would not be possible to use forward chaining to prove that John Doe will get a good job.

EXTRA CREDIT (10 pts):

Part d. Convert the first four statements (not including statement 5) into conjunctive normal form (CNF).

The steps are the following:

1. Eliminate \Leftrightarrow s by replacing them with implications
2. Eliminate \rightarrow (implication)
3. Reduce the scope of negation using De Morgan’s rules and double-negation
4. Convert expressions into conjunct of disjuncts form
5. Make each conjunct a separate clause

Part e. Add to the result of part (d) the negation of “John Doe will get a good job”, and use resolution refutation to prove that he will.

9. You've Got A Friend

Consider a first-order logical language that contains the predicates, $A(x)$, $C(x)$, $D(x)$, to say x is an animal, cat, and dog, respectively, and $L(x, y)$ and $F(x, y)$, to say x loves y and y is a friend of x , respectively.

(a) Translate the following knowledge base sentences into the first-order language.

- I. Cats and dogs are animals.
- II. Everyone loves either a cat or a dog.
- III. Anyone who loves an animal has a friend.

(b) Convert these formulas into conjunctive normal form.

(c) Translate the following query sentence into first-order logic:

Everyone has a friend.

(d) Convert the negation of this sentence into CNF:

NOT Everyone has a friend. (= Someone does not have a friend.)

(e) Prove the query statement “everyone has a friend” from the first three statements, using resolution and proof-by-refutation. (Strategic Hint: Resolve with the negated query last.)