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. $(p \rightarrow q) \rightarrow (q \rightarrow p)$

p	q	$p \rightarrow q$	$q \rightarrow p$	$(p \rightarrow q) \rightarrow (q \rightarrow p)$
1	**1**	1	1	**1**
1	**0**	0	1	**1**
0	**1**	1	0	**0**
0	**0**	1	1	**1**

   Not a tautology since produces 0

   b. $((p \rightarrow q) \land \neg q) \rightarrow \neg p$

p	q	$p \rightarrow q$	$\neg q$	$(p \rightarrow q) \land \neg q$	$\neg p$	$((p \rightarrow q) \land \neg q) \rightarrow \neg p$
1	**1**	1	0	0	0	**1**
1	**0**	0	1	0	0	**1**
0	**1**	1	0	0	1	**1**
0	**0**	1	1	1	1	**1**

   Is a tautology

   a. $q \rightarrow (\neg p \lor \neg q)$

p	q	$\neg p$	$\neg q$	$\neg p \lor \neg q$	$q \rightarrow (\neg p \lor \neg q)$
1	**1**	0	0	0	**0**
1	**0**	0	1	1	**1**
0	**1**	1	0	1	**1**
0	**0**	1	1	1	**1**

   Not a tautology since produces 0

2. (10 pts) Consider the compound sentence: $(\neg p \land \neg q) \land (\neg r \rightarrow p)$. Find an equivalent expression which uses only $\land$ and $\neg$, and which is as short as possible.

$$(\neg p \land \neg q) \land (\neg r \rightarrow p)$$
$$(\neg p \land \neg q) \land (r \lor p)$$
$$((\neg p \land \neg q) \land r) \lor ((\neg p \land \neg q) \land p)$$
$$(\neg p \land \neg q \land r) \lor (\neg p \land \neg q \land p)$$
$$(\neg p \land \neg q \land r) \lor (\neg p \land p \land \neg q)$$
$$(\neg p \land \neg q \land r) \lor (F \land \neg q)$$
$$(\neg p \land \neg q \land r) \lor F$$
$$\mathbf{\neg p \land \neg q \land r}$$

3. (10 pts) Translate the following sentences of first-order logic into English, where Eat(x,y) means x eats a y, Apple(x) means x is an apple, and SeeDoctor(x) means x sees a doctor.

   $(\forall d\ \exists a\ Eat(d,a) \wedge Apple(a)) \Rightarrow (\neg\exists d\ SeeDoctor(d))$

   For every person d, there exists an object a such that if d eats a and a is an apple, there does not exist a person d such that d sees a doctor. [Literal Translation]

   For any given person if there is an apple he/she eats, then he/she does not see the doctor. [More English Translation]

   $\forall x\ ((\exists a\ Eat(x,a) \wedge Apple(a)) \Rightarrow SeeDoctor(x))$

   For every person x if there exists an object a such that x eats a and a is an apple, then x sees a doctor.

4. (10 pts) Translate the following English sentence into (three statements of) first-order logic, where 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

$\exists x\ \forall t\ CanFool(x,t)$	**First Statement**
$\exists t\ \forall x\ CanFool(x,t)$	**Second Statement**
$\neg\forall x\ \forall t\ CanFool(x,t)$	**Third Statement**

   $\exists x\ \forall t\ CanFool(x,t) \wedge (\exists t\ \forall x\ CanFool(x,t)) \wedge (\neg\forall x\ \forall t\ CanFool(x,t))$

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. foo(x,y) foo(y,x)

      {x/y,y/x}

   b. mother(x,y) mother(y,father(x))

      {x/y,y/father(y)} this does not work since y is being substituted by father(y) which produces an overlap variable y.

   c. p(x, y, z) p(q(y), r(z), foo)

      {x/q(r(foo)),y/r(foo),z/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 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. Less(x, y) => Less(x, s(y))
B. Less(s(x), y) => Less(x, y)
C. Less(s(−1), 1)

a. Prove 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.

Previous Goal	**New Goal**	**Know-Base Rule**	**Substitution**
Less(-1, s(1))	**Less(-1, 1) ∨ ¬Less(-1, 1)**	**Rule A**	**Less(-1, 1) => Less(-1, s(1))**
Less(-1, 1)	**Less(s(-1),1) ∨ ¬Less(s(-1), 1)**	**Rule B**	**Less(s(-1), 1) => Less(-1, 1)**
Less(s(-1),1)	**Less(s(-1), 1)**	**Fact C**	**Less(s(-1), 1)**

Fact C is reached so the starting goal is proved.
Note: Since the previous goals where ∨ (OR), we only had to choose one of the statements in the OR. Statement with a Rule for was chosen.

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

	Proof	**Explanation**	
1.	**Less(s(-1), 1)**	**Fact C**	
2.	**Less(s(x), y) => Less(x, y)**	**Rule B**	
3.	**Less(s(-1), 1) => Less(-1, 1)**	**Substitute Fact C into Rule B**	**-> New Fact**
4.	**Less(-1, 1)**	**Modus Ponens (1, 3)**	**-> New Fact**
5.	**Less(x, y) => Less(x, s(y))**	**Rule A**	
6.	**Less(-1, 1) => Less(-1, s(1)**	**Substitute Fact D into Rule A**	**-> New Fact**
7.	**Less(-1, s(1))**	**Modus Ponens (4, 6)**	**-> New Fact**

This is the goal: Less(-1, s(1))

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"

1. **E => F**
2. **(F ∧ R) => J**
3. **¬F => J**
4. **R**

5. **E**

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

	Proof	**Explanation**
1.	**E**	**Fact 5**
2.	**E => F**	**Fact 1**
3.	**F**	**Modus Ponens (1, 2)**
4.	**R**	**Fact 4**
5.	**F ∧ R**	**Modus Ponens (3, 4)**
6.	**(F ∧ R) => J**	**Fact 2**
7.	**J**	**Modus Ponens (5, 6)**

J is defined to be "John will get a good job" and is a fact.

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.

If the last statement did not exist, we could not prove F was True. The second and third statements try to prove "John will get a good job" only if either F was false or if F and R are true. Since forward chaining does not simplify the conditions, it will never prove J.

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 ⇔ s by replacing them with implications
2. Eliminate → (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

1. **No ⇔ s**
2. **Convert statements 1, 2 and 3:**
    a. **E => F** — **Statement 1**
    b. **¬E ∨ F** — **Implication Elimination on Statement 1**
    c. **(F ∧ R) => J** — **Statement 2**
    d. **¬(F ∧ R) ∨ J** — **Implication Elimination on Statement 2**
    e. **¬F => J** — **Statement 3**
    f. **F ∨ J** — **Implication Elimination on Statement 3**
3. **Reducing negations of statement 2**
    a. **¬(F ∧ R) ∨ J** — **Statement 2**
    b. **¬F ∨ ¬R ∨ J** — **De Morgan**
4. **All expressions are in conjunctions of disjunctions:**
    a. **¬E ∨ F** — **Expression 1**
    b. **¬F ∨ ¬R ∨ J** — **Expression 2**
    c. **F ∨ J** — **Expression 3**
    d. **R** — **Expression 4**
5. **All expressions are sperate clauses without conjunctions**

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.

	Proof	**Explanation**
1.	**¬E ∨ F**	**Expression 1**
2.	**¬F ∨ ¬R ∨ J**	**Expression 2**
3.	**F ∨ J**	**Expression 3**
4.	**R**	**Expression 4**
5.	**¬J**	**Negated Conclusion**
6.	**¬F ∨ J**	**Resolution (2, 4)**
7.	**J ∨ J**	**Resolution (3, 6)**
8.	**J**	**Simplify 7**
9.	*****	**Resolution (5, 8)**

Step 8 shows J "John Doe will get a good job" to be true and step 9 concludes the resolution refutation to show by contradiction (empty clause or False = *), ¬J is false. Thus, J is true.

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.

   I. (C(x) => A(x)) ∧ (D(y) => A(y))
   II. ∀x ∃y L(x,y) ∧ (C(y) ∨ D(y))
   III. ∀x ∃y ∃z (L(x,y) ∧ A(y)) => F(x,z)

b. Convert these formulas into conjunctive normal form.

   1. Eliminate ⇔s by replacing them with implications
      a. No ⇔ s

   2. Eliminate à (implication)

a.	**(C(x) => A(x)) ∧ (D(y) => A(y))**	**Statement I**
b.	**(¬C(x) ∨ A(x)) ∧ (¬D(y) ∨ A(y))**	**Implication Elimination on Statement I**
c.	**∀x ∃y ∃z (L(x,y) ∧ A(y)) => F(x,z)**	**Statement III**
d.	**∀x ∃y ∃z ¬(L(x,y) ∧ A(y)) ∨ F(x,z)**	**Implication Elimination on Statement III**

   3. Reduce the scope of negation using De Morgan's rules and double-negation

a.	**∀x ∃y ∃z ¬(L(x,y) ∧ A(y)) ∨ F(x,z)**	**Statement III**
b.	**∀x ∃y ∃z ¬L(x,y) ∨ ¬A(y) ∨ F(x,z)**	**De Morgan**

   4. Convert expressions into conjunct of disjuncts form

a.	**(¬C(x) ∨ A(x)) ∧ (¬D(y) ∨ A(y))**	**Statement I**
b.	**∀x ∃y L(x,y) ∧ (C(y) ∨ D(y))**	**Statement II**

c. **$\forall x\ \exists y\ \exists z\ \neg L(x,y) \lor \neg A(y) \lor F(x,z)$** **Statement III**

5. **Skolemlize**
   a. **$\forall x\ \exists y\ L(x,y) \land (C(y) \lor D(y))$** **Statement II**
   b. **$\forall x\ L(x, E(x)) \land (C(E(x)) \lor D(E(x)))$** **Skolemize (E(x) = Loved by x)**
   c. **$\forall x\ \exists y\ \exists z\ \neg L(x,y) \lor \neg A(y) \lor F(x,z)$** **Statement III**
   d. **$\forall x\ \exists y\ \neg L(x,y) \lor \neg A(y) \lor F(x,G(x))$** **Skolemize (G(x) = Friend of x)**
   e. **$\forall x\ \neg L(x,E(x)) \lor \neg A(E(x)) \lor F(x,G(x))$** **Skolemize (E(x) = Loved by x)**

6. **Drop universal quantifier**
   a. **$\forall x\ L(x, E(x)) \land (C(E(x)) \lor D(E(x)))$** **Statement II**
   b. **$L(x, E(x)) \land (C(E(x)) \lor D(E(x)))$** **Drop $\forall$**
   c. **$\forall x\ \neg L(x,E(x)) \lor \neg A(E(x))) \lor F(x,G(x))$** **Statement III**
   d. **$\neg L(x,E(x)) \lor \neg A(E(x))) \lor F(x,G(x)$** **Drop $\forall$**

7. **Make each conjunct a separate clause**
   a. **$\neg C(x) \lor A(x)$** **Statement I A** **(1)**
   b. **$\neg D(x) \lor A(x)$** **Statement I B** **(2)**
   c. **$L(x, E(x))$** **Statement II A** **(3)**
   d. **$C(E(x)) \lor D(E(x))$** **Statement II B** **(4)**
   e. **$\neg L(x,E(x)) \lor \neg A(E(x))) \lor F(x,G(x))$** **Statement III** **(5)**

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

Everyone has a friend.

$\forall x\ \exists y\ F(x,y)$

d. Convert the negation of this sentence into CNF:

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

$\neg(\forall x\ \exists y\ F(x,y))$
   a. **$\neg(\forall x\ \exists y\ F(x,y))$** **New Statement**
   b. **$\exists x\ \neg(\exists y\ F(x,y))$** **De Morgan Quantifier**
   c. **$\exists x\ \forall y\ \neg F(x,y)$** **De Morgan Quantifier**

$\exists x\ \forall y\ \neg F(x,y)$ (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.)

	Proof	**Explanation**
1.	**$\neg C(x) \lor A(x)$**	**Statement 1**
2.	**$\neg D(x) \lor A(x)$**	**Statement 2**
3.	**$L(x, E(x))$**	**Statement 3**
4.	**$C(E(x)) \lor D(E(x))$**	**Statement 4**
5.	**$\neg L(x,E(x)) \lor \neg A(E(x))) \lor F(x,G(x))$**	**Statement 5**
6.	**$\exists x\ \forall y\ \neg F(x,y)$**	**Negated Conclusion**

7.	**∀y ¬F(G(y),y)**	**Skolemize (G(y) = Friend of y) (CNF Conversion)**
8.	**¬F(G(x),x)**	**Drop ∀ (CNF Conversion) & Variable Substitution**
9.	**¬C(x) ∨ ¬L(x,E(x)) ∨ F(x,G(x))**	**Resolution (1,5)**
10.	**¬D(x) ∨ ¬L(x,E(x)) ∨ F(x,G(x))**	**Resolution (2,5)**
11.	**D(E(x)) ∨ ¬L(x,E(x)) ∨ F(x,G(x))**	**Resolution (4, 9)**
12.	**¬L(x,E(x)) ∨ F(x,G(x))**	**Resolution (10, 11) & Simplify**
13.	**F(x,G(x))**	**Resolution (3, 16)**
14.	*****	**Resolution (8, 17)**

Step 13 shows F(x,G(x)) "x has a friend that is a friend of x" to be true and step 9 concludes the resolution refutation to show by contradiction (empty clause or False = *), ¬F(G(x),x) … ∃x ∀y ¬F(x,y) is false. Thus, ∀x ∃y F(x,y) is true.