**CSci 384: Artificial Intelligence Spring, 2017**

**Instructor: Dr. M. E. Kim Date: March 21st , 2017**

**Due: 3:30 PM, March 30th, 2016 (Thr.)**

**Home Assignment 3: PL & FOL (205/250 points)**

**Q1.** [4/ 10 pt] Consider a vocabulary with only five propositions, *A,B,C*, *D* and *E*. How many models are

there for the following sentences? Explain your answer.

1. [0/ 5] (A ∧ C) ∨ (¬C ∧ D) ∨ (B ∧ ¬E)

we have all the 5 propositions used,(A C, D B E)  
number of models = 25 = 32

Among these 32, how many of them make it true? 20 only.

Since the sentence is a disjunct of 3 conjuncts, the sentence is true if any of conjunct is true.

i.e. when A=C=true OR C=false & D=true OR B=true & E=false.

So, the models of (A, B, C, D, E) are (T, \*, T, \*, \*) | (\*, \*, F, T, \*) | (\*, T, \*, \*, F).

Thus, 3\*23 (for the possible assignments of \* for 3 conjuncts), but the # of overlapped assignments should be deducted.

The overlaps of models happens b/t (T, \*, T, \*, \*) and (\*, T, \*, \*, F) or between (\*, \*, F, T, \*) and (\*, T, \*, \*, F) since C can’t both T and F.

So, the overlapped models of (T, \*, T, \*, \*) and (\*, T, \*, \*, F) is (T, T, T, \*, F) while that of (\*, \*, F, T, \*) and (\*, T, \*, \*, F) is (\*, T, F, T, F), i.e. 2 + 2 = 4 overlapped models.

Therefore, the total number of models is 3\*23 - 4 = 20.

1. [4/ 5] (B ↔ ¬C ↔ D)

we have 3 propositions used, ( B, C, D)  
   number of models = 23 = 8. Coincidently, 8 models, however, not because of 23 = 8. -2

The possible models of sentence happens when (A, B, C, D, E) = (\*, T, F, T, \*) or (\*, F, T, F, \*).

Thus, 22 + 22 = 8 models.

NOTE: It’s **not** suggested to get the number of models by Truth Table Enumeration.

Q2. 12/30

1. 12/ Convert the following set of sentences to clausal form.

a. A ↔ (B ∨ E) (¬A ∨ B ∨ E ) ∧ (¬B ∨ A) ∧ (¬B ∨ A)

b. E → D (¬E ∨ D)

c. (C ∧ F) → ¬B (¬C ∨ ¬F ∨ ¬B)

d. E → B (¬E ∨ B)

e. B → F (¬B ∨ F)

f. B → C (¬B ∨ C)

1. By means of *resolution*, prove ¬A ∧ ¬B from the above clauses.
2. Trace the execution of *DPLL* on the conjunction of the clauses in 1).

**Q3.** [20 pt]According to some political pundits, a person who is radical (R) is electable (E) if (s)he is conservative (C), but otherwise is not electable.

1. [10] Write the above sentence in propositional logic using the given propositions R, E and C.

**R ⇒ ( E ⇔ C ) - if a person is a radical then they are electable if and only if they are conservative**

1. [10] Is your sentence in 1) in ***Horn form***? Justify your answer.

Yes – R ⇒ (E ⇐⇒ C) ≡ R ⇒ ((E ⇒ C) ∧ (C ⇒ E))

≡ ¬R ∨ ((¬E ∨ C) ∧ (¬C ∨ E))

≡ (¬R ∨ ¬E ∨ C) ∧ (¬R ∨ ¬C ∨ E))

**Q4.** [6/ 10] Decide each sentence is valid (necessarily true). Justify your answer.

1. 0/ (∃*x x* = *x*) → (∀*y* ∃ *z* *y* = *z*)

existsx x = x is True. forally exists z y = z This may not be true. So, T => F is false. So, this is not valid.

Valid.  existsx x = x is valid by itself. -- in standard FOL, every model has at least one object;

∀*y* ∃ *z* *y* = *z* is also valid because for every value of y in any given model, there is a z which is y itself.

So, the entire sentence is valid as true 🡪 true is true.

1. 6/ ∀*x* Smart(*x*) ∨ (*x* = *x*)

forallx Smart(x) v (x = x). The second half, (x = x) is true. So, true OR anything will be true. So, this is valid. Good +1

**Q5**.[38/ 40] For the given English sentence, write them in First-Order Logical sentence which is both syntactically valid and express the meaning correct, using the following predicates and function.

Predicates: In(*x, y*) – A country *x* is in the region *y*.

Borders(*x, y*) - *x* borders *y*.

Person(*x*) - *x* is a person.

HasSSN(*x, y*) - *x* has a social security number *y*.

Occupation(*p, o*) – Person *p* has occupation *o*.

Customer(*p, q*) - Person *p* is a customer of person *q*.

Born(*x, y*) - Person *x* was born in the country *y*.

Parent(*x, y*) - Person *x* is a parent of *y*.

Citizen(*x, y, z*) - Person *x* is a citizen of a country y by *z*.

Resident(*x, y*) - Person *x* is a resident of country *y*.

Constant: SouthAmerica, Europe, UK, Birth, Doctor, Lawyer.

1. [10] No region in South America borders any region in Europe.

¬[∃c, d In(c, SouthAmerica) ∧ In(d, Europe) ∧ Borders(c, d)]

1. [10] There is a lawyer all of whose customers are doctors.

∃p Person(p) ∧ Occupation(p, Lawyer ) ∧ ∀q Customer (p, q) ⇒ Occupation(q, Doctor )

1. [8/ 10] No two people have the same social security number.

¬existsx y n Person(x) ∧Person( y) ∧ ￢(x = y) ∧ ~~->~~ (HasSS #(x, n) ∧HasSS #( y, n)) -2

1. [10] A person born in the UK, each of whose parents is a UK citizen or a UK resident, is a UK citizen by birth.

∀x Person(x)∧Born(x,UK)∧(∀y Parent(y, x) ⇒ ((∃ r Citizen(y,UK, r))∨ Resident(y,UK))) ⇒ Citizen(x,UK,Birth).

**Q6.** [10] Translate the given FOL sentence into good and natural English **without using variables *x’*s or *y’*s.**

∀ *x, y, l* SpeaksLanguage(*x, l*) ∧ SpeaksLanguage(*y, l*) ⇒ Understands(*x, y*) ∧ Understands(*y, x*)

If two people speak the same language, then they understand each other.

**Q7.** [7/ 10] Find the **MGU** (most general unifier) in each pair of sentence below or justify why unification is not possible.

1. 5/ Knows(Father(*y*), *y*), Knows(*x, x*)

With the help of these two given sentences, the unifier does not exist because the occurs-check prevent the unification of y with father(y)

1. 2/ Wines(*x, y*) vs. Wines(*y, x*)

θ={x/y} or {y/x} With your θ, Wines(x, x) = Wines(x, x). Incorrect.

Rename Wines (y, x) by Wines (z1, z2). Then, MGU = {x/z1, y/z2}

**Q8.** [14/ 15] Write down the following sentences in the 1st -order logical representations, suitable for their use with ***Generalized Modus Ponens***, i.e. in Horn clauses. Do NOT convert them in CNF.

1. Horses, cows, and pigs are mammals. - x (H(X)C(x)P(x)) => M(x)
2. An offspring of a horse is a horse. - x z (H(z) **∧**O(x,z)) => H(x)
3. Bluebeard is a horse.- H(Bluebeard)
4. Bluebeard is Charlie’s parent. - P(Bluebeard,Charlie)
5. Offspring and parent are inverse relations. - x z O(x,z) <=> P(z,x)
6. Every mammal has a parent. - x Mammal(x)  y Parent(y,x)

Further skolemize the existentially quantified y as a function G of x.

x Mammal(x)  Parent(G(x),x)

**Q9.** [19/ 20 pt] From the sentences you wrote in Q8, answer the following question using a ***backward-chaining algorithm***.

1. [9/ 10] Draw the proof tree generated by a backward chaining algorithm for the query ∃ *h,*  *Horse(h),* where clauses are matched in the order given.

 show the substitution of variables.

1. [5] What do you notice about this domain?

*I noticed that It is infinite*

1. [5] How many solutions for *h* actually follow from your sentences?

*There are two solutions – h/ Bluebeard and h/ Charlie*

**Q10.** [30] From ”Horses are animals”, it follows that ”The head of a horse is the head of an animal.”.

Demonstrate that this inference is valid by carrying out the following steps:

1. [10] Translate the premise and the conclusion into the language of 1st -order logic. Use three predicates:

*HeadOf(h,x)* - meaning *h* is the head of *x*.

*Horse(x)*

*Animal(x)*

x Horse(x) => Animal(x)

h x (Horse(h) HeadOf(x,h)) => a (Animal(a)  HeadOf(x, a))

2) [10] Negate the conclusion, and convert the premise and the negated conclusion into conjunctive normal form (CNF).

¬∀x ∀h {[Horse(x)  HeadOf(h, x)]  ∃y [Animal(y)  HeadOf(h, y)]}

= ∃x ∃h ¬{[Horse(x)  HeadOf(h, x)] ∃y [Animal(y)  HeadOf(h, y)]}

= ∃x ∃h ¬{¬[Horse(x)  HeadOf(h, x)]  ∃y [Animal(y)  HeadOf(h, y)]}

= ∃x ∃h {[Horse(x)  HeadOf(h, x)]  ¬∃y [Animal(y)  HeadOf(h, y)]}

= ∃x ∃h {[Horse(x)  HeadOf(h, x)]  ∀y [¬Animal(y)  ¬HeadOf(h, y)]}

≡ Horse(G)  HeadOf(H, G)  [¬Animal(y)  ¬HeadOf(H, y)]

3. [10] Use ***resolution*** to show that the conclusion follows from the premise. Draw the *proof tree of resolution,* showing the *substitutions*; see Figure 9.11-12 in the textbook.

HeadOf(H,y), HeadOf(H,G) == ******Animal(G)

**Animal(G),******Horse(x)  Animal(x) == ******Horse(G)

Horse(G), [¬Horse(x)  Animal(x)] - Animal(G) ={x/G}

Animal(G), [¬Animal(y)  ¬HeadOf(H, y)] - ¬ HeadOf(H, G) ={y/G}

¬ HeadOf(H, G), HeadOf(H, G) |-- null So, the claim is true.

={x/G, y/G

**Q11.** [39/ 40] Given sentences (A – E) below,

* 1. All great chefs are French.
  2. All Frenches enjoy good food.
  3. Paul or Sophie is a great chef.
  4. Paul is not a great chef.
  5. Query: Who enjoys a good food?

1. [10] write them in their 1st -order logical representations using the following predicates and constants:

x GC(x) -> I(x)

x I(x) -> EF(x) name a variable in the different name from x in the 1st sentence.

GC(Paul)  GC(Sophie)

******GC(Paul)

 z EF(z)

Predicates: GC – great chef(s), F – French, EF – enjoy good food,

Constants: Paul, Sophie

1. [10] Answer the query by ***forward chaining*** method. Draw the proof trees showing the substitutions step by step. Refer to the slide #23 - #25.

Start from GC(Paul) or GC(Sophie), i.e. in the reverse direction.

****This is forward chaining. Show the substitution, {x/sophie}.

1. [10] Convert the sentences in 1) to the definite clauses in CNF, suitable for Knowledge\_Base through Skolemization, etc. if necessary. Refer to the slide #42.

******GC(x)  I(x)

******I(x)  EF(x)

GC(Paul)  GC(Sophie)

1. [9/ 10] By applying ***resolution****,* answer the query. Show the steps of proof by drawing the resolution tree. Refer to the figure in slide #43 of Chap. 9.

***~~~~***~~GC(Sophie)~~, ******GC(Paul), GC(Sophie)  GC(Paul) == GC(Sophie)

GC(~~Paul~~), GC(Sophie), ****** GC(x)  I(x) == I(~~Paul~~ Sophie) θ={x/ ~~Paul~~ Sophie }

I(Sophie) , ******I(x)  EF(x) == EF(~~Paul~~  Sophie) θ={x/ Sophie }

EF(Sophie), ****** EF(z) == empty clause θ={ z/~~Louis~~ Sophie, x/ Sophie }

z/ Sophie

Thus, z = Sophie enjoys a good food.

**Q12.** [6/ 15] Suppose that the sentence A in Q11 is changed to:

A1. *Some* great chefs are French.

1. [1/ 5] Write it in the FOL sentence.

x GC(x) ~~=>~~ **∧** I(x) ∃x [GC(x) ∧ I(x)]

1. [5] Convert 1) to the the definite clause in CNF, suitable for Knowledge\_Base through Skolemization, etc. if necessary.

GC(K), I(K)

1. [0/ 5] Prove how the same query can be answered (or not). Justify your answer step by step.

******GC(Michael), GC(Michael)  GC(Louis) == GC(Louis)

In the given sentences, there is neither Michael nor Louis.

Resolve ~EF(x) with ~I(y) V EF(y) 🡺 ~I(y) with θ= {x/y}

Resolve ~I(y) with I(K) 🡺 null with {y/K} but K is a skolem constant, so it doen’t answer the query because Skolem constant K is a value which doesn’t exist in KB.

Thus, the query is not answered.