Homework 3

CSCI 4511W Spring 2018

March 21, 2018

Instructor: Dr.Maria Gini

Joowon Kim(kimx4342)

---

1. Decide if the following sentences are valid, unsatisfiable, or neither. To do it, use the truth tables and equivalency rules in Chapter 7.

   (a) Big ⇒ Big

   → Valid

Big	Big	$Big \Rightarrow Big$
$True$	$True$	$True$
$False$	$False$	$True$

   (b) Big ⇒ Heavy

   → Neither (satisfiable)

Big	$Heavy$	$Big \Rightarrow Heavy$
$True$	$True$	$True$
$True$	$False$	$False$
$False$	$True$	$True$
$False$	$False$	$True$

   (c) (Big ⇒ Heavy) ⇒ (¬ Big ⇒ ¬ Heavy)

   → Neither (satisfiable)

Big	$Heavy$	$\neg Big$	$\neg Heavy$	$Big \Rightarrow Heavy$	$\neg Big \Rightarrow \neg Heavy$
$True$	$True$	$False$	$False$	$True$	$True$
$True$	$False$	$False$	$True$	$False$	$True$
$False$	$True$	$True$	$False$	$True$	$False$
$False$	$False$	$True$	$True$	$True$	$True$

$(Big \Rightarrow Heavy) \Rightarrow (\neg Big \Rightarrow \neg Heavy)$
$True$
$True$
$False$
$True$

   (d) Big ∨ Heavy ∨ ¬ Heavy

   → Valid

Big	$Heavy$	$\neg Heavy$	$Big \vee Heavy \vee \neg Heavy$
$True$	$True$	$False$	$True$
$True$	$False$	$True$	$True$
$False$	$True$	$False$	$True$
$False$	$False$	$True$	$True$

(e) ((Big ∧ Dense) ⇒ Heavy) ⇔ ((Big ⇒ Dense) ∨ (Heavy ⇒ Dense))

→ Neither (satisfiable)

Big	*Dense*	*Heavy*	*Big* ∧ *Dense*	*Big* ⇒ *Dense*	*Heavy* ⇒ *Dense*	(*Big* ∧ *Dense*) ⇒ *Heavy*
True	*True*	*True*	*True*	*True*	*True*	*True*
True	*True*	*False*	*True*	*True*	*True*	*False*
True	*False*	*True*	*False*	*False*	*False*	*True*
True	*False*	*False*	*False*	*False*	*True*	*True*
False	*True*	*True*	*False*	*True*	*True*	*True*
False	*True*	*False*	*False*	*True*	*True*	*True*
False	*False*	*True*	*False*	*True*	*False*	*True*
False	*False*	*False*	*False*	*True*	*True*	*True*

(*Big* ⇒ *Dense*) ∨ (*Heavy* ⇒ *Dense*)	((*Big* ∧ *Dense*) ⇒ *Heavy*) ⇔ ((*Big* ⇒ *Dense*) ∨ (*Heavy* ⇒ *Dense*))
True	*True*
True	*False*
False	*False*
True	*True*
True	*True*
True	*True*
True	*True*
True	*True*

(f) (Big ⇒ Dense) ⇒ ((Big ∧ Heavy) ⇒ Dense)

→ Valid

Big	*Heavy*	*Dense*	*Big* ⇒ *Dense*	*Big* ∧ *Heavy*	(*Big* ∧ *Heavy*) ⇒ *Dense*
True	*True*	*True*	*True*	*True*	*True*
True	*True*	*False*	*False*	*True*	*False*
True	*False*	*True*	*True*	*False*	*True*
True	*False*	*False*	*False*	*False*	*True*
False	*True*	*True*	*True*	*False*	*True*
False	*True*	*False*	*True*	*False*	*True*
False	*False*	*True*	*True*	*False*	*True*
False	*False*	*False*	*True*	*False*	*True*

(*Big* ⇒ *Dense*) ⇒ ((*Big* ∧ *Heavy*) ⇒ *Dense*)
True
True
True
True
True
True
True
True

(g) Small ∨ Cute ∨ (Small ⇒ Cute)

→ Valid

Small	*Cute*	*Small* ⇒ *Cute*	*Small* ∨ *Cute* ∨ (*Small* ⇒ *Cute*)
True	*True*	*True*	*True*
True	*False*	*False*	*True*
False	*True*	*True*	*True*
False	*False*	*True*	*True*

(h) (Small∧ Cute) ∨ ¬ Cute

→ Neither (satisfiable)

Small	*Cute*	¬*Cute*	*Small* ∧ *Cute*	(*Small* ∧ *Cute*) ∨ ¬*Cute*
True	*True*	*False*	*True*	*True*
True	*False*	*True*	*False*	*True*
False	*True*	*False*	*False*	*False*
False	*False*	*True*	*False*	*True*

(i) ((Rain ⇒ Wet) ∧ (Wet ⇒ Cold)) ⇒ (Rain ⇒ Cold)

→ Valid

$Rain$	Wet	$Cold$	$Rain \Rightarrow Wet$	$Wet \Rightarrow Cold$	$Rain \Rightarrow Cold$	$(Rain \Rightarrow Wet) \wedge (Wet \Rightarrow Cold)$	$((Rain \Rightarrow Wet) \wedge (Wet \Rightarrow Cold)) \Rightarrow (Rain \Rightarrow Cold)$
$True$	$True$	$True$	$True$	$True$	$True$	$True$	$True$
$True$	$True$	$False$	$True$	$False$	$False$	$False$	$True$
$True$	$False$	$True$	$False$	$True$	$True$	$False$	$True$
$True$	$False$	$False$	$False$	$True$	$False$	$False$	$True$
$False$	$True$	$True$	$True$	$True$	$True$	$True$	$True$
$False$	$True$	$False$	$True$	$False$	$True$	$False$	$True$
$False$	$False$	$True$	$True$	$True$	$True$	$True$	$True$
$False$	$False$	$False$	$True$	$True$	$True$	$True$	$True$

(j) ((Rain ∨ Wet) ∧ (¬ Wet ∨ Cold)) ⇒ (Rain ∨ Cold)

→ Valid

$Rain$	Wet	$Cold$	$\neg Wet$	$Rain \vee Wet$	$\neg Wet \vee Cold$	$Rain \vee Cold$	$(Rain \vee Wet) \wedge (\neg Wet \vee Cold)$	$((Rain \vee Wet) \wedge (\neg Wet \vee Cold)) \Rightarrow (Rain \vee Cold)$
$True$	$True$	$True$	$False$	$True$	$True$	$True$	$True$	$True$
$True$	$True$	$False$	$True$	$True$	$True$	$True$	$True$	$True$
$True$	$False$	$True$	$True$	$True$	$True$	$True$	$True$	$True$
$True$	$False$	$False$	$True$	$True$	$True$	$True$	$True$	$True$
$False$	$True$	$True$	$False$	$True$	$True$	$True$	$True$	$True$
$False$	$True$	$False$	$False$	$True$	$False$	$False$	$False$	$True$
$False$	$False$	$True$	$True$	$False$	$True$	$True$	$False$	$True$
$False$	$False$	$False$	$True$	$False$	$True$	$False$	$False$	$True$

2. For each of the following formulas, state briefly if it is a correct representation in propositional calculus of the sentence "If Bill works and his father stays at home, then his mother is happy" or not and explain why. The propositions used in the sentences should have an obvious interpretation.

   $\Rightarrow$ Possible propositional calculus of "If Bill works and his father stays at home, then his mother is happy" is

   (BillWork $\wedge$ DadHome) $\Rightarrow$ MomHappy

   $\equiv \neg$ (BillWork $\wedge$ DadHome) $\vee$ MomHappy [Implication Elimination]

   $\equiv (\neg$ BillWork $\vee \neg$ DadHome) $\vee$ MomHappy [De Morgan]

   $\equiv \neg$ MomHappy $\Rightarrow \neg$ (BillWork $\wedge$ DadHome) [Contraposition]

   (a) BillWork $\wedge$ DadHome $\wedge$ MomHappy

   $\rightarrow$ This is an $\boxed{\text{incorrect}}$ representation. It says BillWork and DadHome and MomHappy and implies the conjunction of MomHappy, which should not.

   (b) (BillWork $\wedge$ DadHome) $\Rightarrow$ MomHappy

   $\rightarrow$ This is a $\boxed{\text{correct}}$ representation by the Implication Elimination I wrote above.

   (c) (BillWork $\vee$ DadHome) $\Rightarrow$ MomHappy

   $\rightarrow$ This is an $\boxed{\text{incorrect}}$ representation. It does not represent the conjunction of BillWork and DadHome.

   (d) MomHappy $\Rightarrow$ (BillWork $\wedge$ DadHome)

   $\rightarrow$ This is an $\boxed{\text{incorrect}}$ representation. If both sides are negated, than it would be a correct sentence by contraposition.

   (e) $\neg$ BillWork $\vee$ ($\neg$ MomHappy $\vee$ DadHome)

   $\rightarrow$ This is an $\boxed{\text{incorrect}}$ representation. If MomHappy is not negated and instead, DadHome is negated, than it would be a correct sentence by De Morgan.

3. Convert the following set of propositional clauses to CNF and prove by resolution with refutation that it is Pleasant.

   (a) Cold $\wedge$ Dry $\Rightarrow$ Pleasant
   $\equiv \neg$ (Cold $\wedge$ Dry) $\vee$ Pleasant [Implication Elimination]
   $\equiv \neg$ Cold $\vee \neg$ Dry $\vee$ Pleasant [De Morgan]

   (b) January $\Rightarrow$ Winter $\wedge$ Wet
   $\equiv \neg$ January $\vee$ (Winter $\wedge$ Wet) [Implication Elimination]
   $\equiv (\neg$ January $\vee$ Winter) $\wedge$ ($\neg$ January $\vee$ Wet) [Distributivity]

   (c) Winter $\Rightarrow$ Dry
   $\equiv \neg$ Winter $\vee$ Dry [Implication Elimination]

   (d) Wet $\Rightarrow$ Cold
   $\equiv \neg$ Wet $\vee$ Cold [Implication Elimination]

   (e) January

   $\rightarrow$

   January, $\neg$ January $\vee$ (Winter $\wedge$ Wet)

   (Winter $\wedge$ Wet) $\equiv \neg$ **Winter** $\vee \neg$ **Wet**

   $\neg$ Winter $\vee \neg$ Wet , $\neg$ Wet $\vee$ Cold

   $\neg$ **Winter** $\vee \neg$ **Wet** $\vee$ **Cold**

   $\neg$ Cold $\vee \neg$ Dry $\vee$ Pleasant , $\neg$ Winter $\vee$ Dry

   ---

   $\neg$ **Winter** $\vee \neg$ **Cold** $\vee$ **Pleasant**

   $\neg$ Winter $\vee \neg$ Cold $\vee$ Pleasant , $\neg$ Winter $\vee \neg$ Wet $\vee$ Cold

   ---

   $\neg$ **Winter** $\vee \neg$ **Wet** $\vee$ **pleasant**

   $\equiv \neg$ **(Winter** $\wedge$ **Wet)** $\vee$ **Pleasant** [De Morgan]

   $\equiv$ **(Winter** $\wedge$ **Wet)** $\Rightarrow$ **Pleasant** [Implication Elimination]

4. Programming questions.

   (a) Create a propositional data base, include in it the clauses from problem 3 above and prove it is Pleasant using the program.

   (b) Create a propositional data base and represent the following: "If the unicorn is mythical, then it is immortal but if it is not mythical then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical, if it is horned."

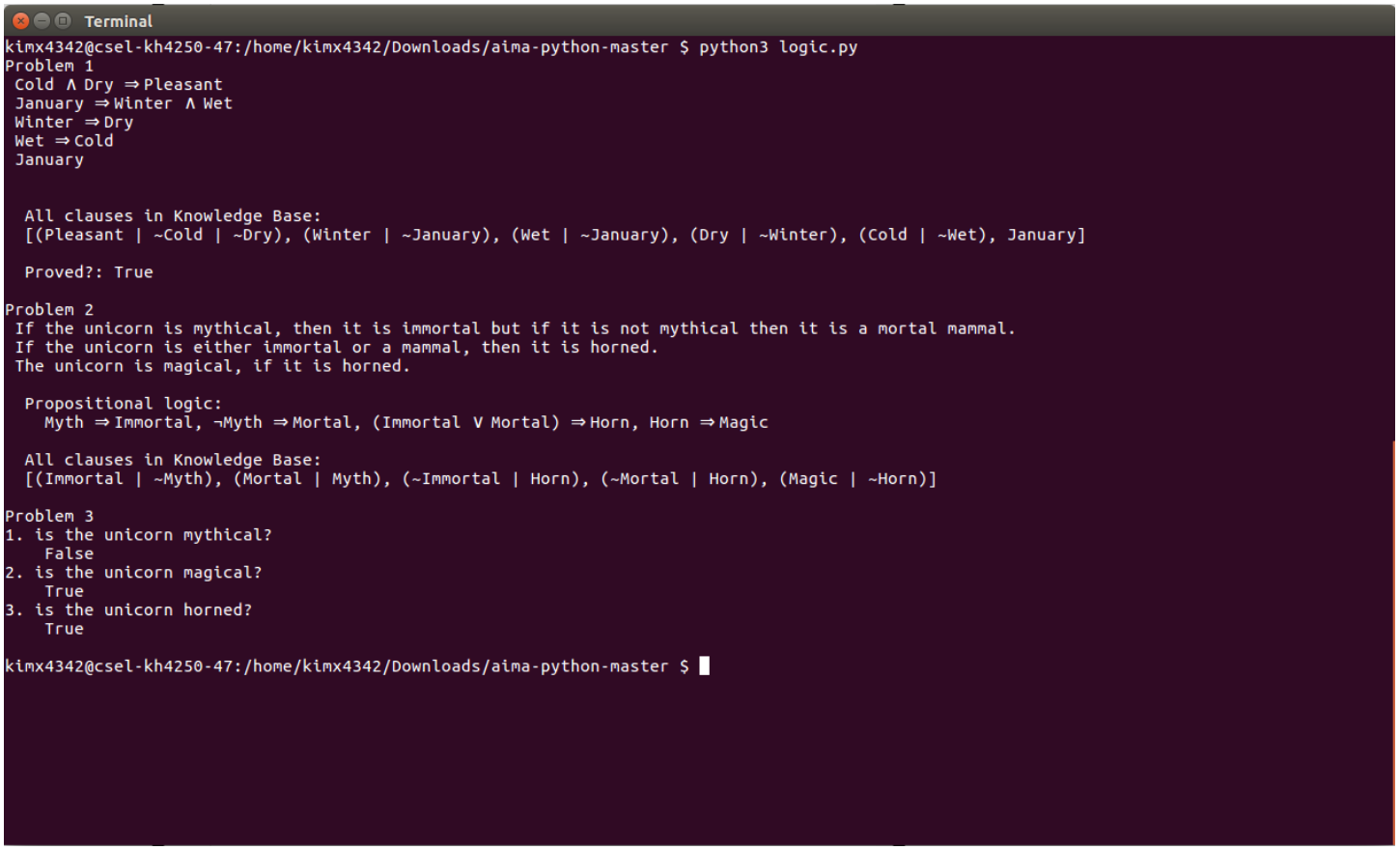
   (c) Use ask to answer the following questions from the data base you created for the previous question:

      i. is the unicorn mythical?

      ii. is the unicorn magical?

      iii. is the unicorn horned?

      What can you say from the answers you get? Do the statements logically follow from the knowledge base? or not?
      → The output of the program is attached, as well the code I modified.



logic.py

```
1129  SPACE = '  '
1130
1131  def prob1():
1132      p1_kb = PropKB()
1133      Cold, Dry, Pleasant, January, Winter, Wet = expr('Cold, Dry, Pleasant, January, Winter, Wet')
1134      p1_kb.tell((Cold & Dry) | '==>' | Pleasant)
1135      p1_kb.tell(January | '==>' | (Winter & Wet))
1136      p1_kb.tell(Winter | '==>' | Dry)
1137      p1_kb.tell(Wet | '==>' | Cold)
1138      p1_kb.tell(January)
1139
1140      print('Problem 1')
1141      print(' Cold ∧ Dry → Pleasant \n January → Winter ∧ Wet \n Winter → Dry \n Wet → Cold \n January \n'.format(SPACE))
1142      print()
1143      print('{}All clauses in Knowledge Base: '.format(SPACE))
1144      print('{}{}'.format(SPACE,p1_kb.clauses))
1145      print()
1146
1147      check = pl_resolution(p1_kb, Pleasant)
1148      print('{}Proved?: {}'.format(SPACE,check))
1149      print()
1150
1151  def prob2():
1152      p2_kb = PropKB()
1153      Myth, Immortal, Mortal, Horn, Magic = expr('Myth, Immortal, Mortal, Horn, Magic')
1154      p2_kb.tell(Myth | '==>' | Immortal)
1155      p2_kb.tell(~Myth | '==>' | Mortal)
1156      p2_kb.tell((Immortal | Mortal) | '==>' | Horn)
1157      p2_kb.tell(Horn | '==>' | Magic)
1158
1159      print('Problem 2')
1160      print(' If the unicorn is mythical, then it is immortal but if it is not mythical then it is a mortal mammal.'.format(SPACE))
1161      print(' If the unicorn is either immortal or a mammal, then it is horned.'.format(SPACE))
1162      print(' The unicorn is magical, if it is horned.{}'.format(SPACE,'\n'))
1163      print()
1164      print('{}Propositional logic: '.format(SPACE))
1165      print('{}{}Myth → Immortal, ¬Myth → Mortal, (Immortal ∨ Mortal) → Horn, Horn → Magic{}'.format(SPACE,SPACE,'\n'))
1166      print('{}All clauses in Knowledge Base: '.format(SPACE))
1167      print('{}{}'.format(SPACE,p2_kb.clauses))
1168      print()
1169
1170  def prob3():
1171      p3_kb = PropKB()
1172      Myth, Immortal, Mortal, Horn, Magic = expr('Myth, Immortal, Mortal, Horn, Magic')
1173      p3_kb.tell(Myth | '==>' | Immortal)
1174      p3_kb.tell(~Myth | '==>' | Mortal)
1175      p3_kb.tell((Immortal | Mortal) | '==>' | Horn)
1176      p3_kb.tell(Horn | '==>' | Magic)
1177
1178      print('Problem 3')
1179      print('1. is the unicorn mythical?'.format(SPACE))
1180      print('{}{}{}'.format(SPACE,SPACE,p3_kb.ask_if_true(Myth)))
1181      print('2. is the unicorn magical?'.format(SPACE))
1182      print('{}{}{}'.format(SPACE,SPACE,p3_kb.ask_if_true(Magic)))
1183      print('3. is the unicorn horned?'.format(SPACE))
1184      print('{}{}{}'.format(SPACE,SPACE,p3_kb.ask_if_true(Horn)))
1185
1186  if __name__ == "__main__":
1187      prob1()
1188      prob2()
1189      prob3()
1190      print()
1191
```