Problem 1  (Problem 8.6 of text)

Not scored.

a. Is valid. The left side and the right sides are both always true – note that \( z \) can be the same thing as \( y \). The sentence is thus \( true \rightarrow true \), which is always true.
b. Is valid. (Making the reasonable assumption that \( P \) is a predicate and not a function, since otherwise the sentence would not have correct syntax.
c. Is valid, since the right hand side of the disjunction is always true.

Problem 2  (Problem 8.23 of text)

2 points: 1 for a and c, 1 for d

a. Good translation.
   (Not good - need to make sure \( x \) and \( y \) are not the same, so add \( \wedge \sim(x=y) \) before implication:
   \[ \neg \exists x,y,n \; Person(x) \wedge Person(y) \wedge \sim(x=y) \Rightarrow \]
   \[ HasSS#(x,n) \wedge HasSS#(y,n) \]

b. Good translation, if we take advantage of “background” knowledge that every person has exactly one ssn. Without using that background knowledge, we could represent the sentence in FOL as
   \[ \exists n \; \neg HasSS#(John, n) \Rightarrow HasSS#(Mary, n) \]

c. Good translation, assuming the background knowledge that everyone has a ssn. Without using that background knowledge, the FOL could be
   \[ \forall x,n \; Person(x) \wedge HasSS#(x,n) \Rightarrow Digits(n,9) \]

d. (i) No two people have the same social security number
   \[ \exists x,y \; Person(x) \wedge Person(y) \wedge SS#(x) = SS#(y) \]

(ii) John’s SSN number is the same as Mary’s
   \[ SS#(John) = SS#(Mary) \]

(iii) Everyone’s SSN has nine digits
   \[ \forall x \; Person(x) \Rightarrow Digits(SS#(x),9) \]
Problem 3 (Problem 8.24 of text)

Total Points: 2 – half a point for each of b, c, d, e.

(Answers for a through k are provided here.)

**Predicates:**
- Takes(s, c, t): Predicate. Student s takes/took course c in term t
- French: Constant. A course teaching the language French
- Pass(s, c): Predicate. Students s passes course c.
- Score(s, c, t): Function. The score of student s in course c in term t
  - Person(x): x is a person
  - Policy(y): y is a policy
  - Buys(x, y): x buys y
  - Smart(x): x is smart
- Expensive(y): y is expensive
- Agent(x): x is an agent
- SellsPolicy(x, y): x sells a policy(insurance) to y
- Insured(x): x is insured
- Barber(x): x is a barber
- Man(y): y is a man in the town
- Shaves(x, y): x shaves y
- Born(x, y): x was born in country y
- Parent(x, y): x is y’s parent
- Citizen(x, y): x is a citizen of country y
- Resident(x, y): x is a resident of country y
- CitizenByBirth(x, y): x is a citizen of country y by birth
- CitizenByDescent(x, y): x is a citizen of country y by descent.
- Politician(x): x is a politician
- CanFool(x, y): x can fool y

a) $\exists s \ \text{Takes}(s, \text{French}, \text{Spring2001})$

b) $\forall s, t \ \text{Takes}(s, \text{French}, t) \rightarrow \text{Pass}(s, \text{French})$

c) $\exists s \ \text{Takes}(s, \text{Greek}, \text{Spring2001}) \land (\forall z \ \text{Takes}(s, \text{Greek}, \text{Spring2001}) \rightarrow s = z)$

d) The easy way is to have a BestScore function, which is OK. The following approach finds the best score using forall:

$$\forall t, s1, s2 \ [(\forall s8 > (\text{Score}(s1, \text{Greek}, t), \text{Score}(s8, \text{Greek}, t))) \land$$
$$\forall s9 > (\text{Score}(s2, \text{Greek}, t), \text{Score}(s9, \text{Greek}, t))]$$
$$\rightarrow > (\text{Score}(s1, \text{Greek}, t), \text{Score}(s2, \text{French}, t))$$
e.) Sentence:
Every person who buys a policy is smart.

Translation:
\((\forall x)(\forall y) \text{Person}(x) \land \text{Policy}(y) \land \text{Buys}(x,y) \Rightarrow \text{Smart}(x)\)

f.) Sentence:
No person buys an expensive policy.

Translation:
\((\forall x)(\forall y) \text{Person}(x) \land \text{Policy}(y) \land \text{Buys}(x,y) \Rightarrow \neg \text{Expensive}(y)\)

g.) Sentence:
There is an agent who sells policies only to people who are not insured.

Translation:
\((\exists x)(\text{Agent}(x) \land (\forall y) \text{SellsPolicy}(x,y) \Rightarrow \neg \text{Insured}(y))\)

Note: Here the students might choose to use a more general predicate, \(\text{Sells}(x,y,z)\), i.e. \(x\) sells \(y\) to \(z\) and have a separate predicate called \(\text{Policy}(y)\).

h.) Sentence:
There is a barber who shaves all men in town who do not shave themselves.

Translation:
\((\exists x)(\text{Barber}(x) \land (\forall y) \text{Man}(y) \land \text{Shaves}(x,y) \Rightarrow \neg \text{Shaves}(y,y))\)

i.) Sentence:
A person born in the UK, each of whose parents is a UK citizen or a UK resident, is a UK citizen by birth.

Translation:
\((\forall x)((\text{Born}(x,UK) \land (\forall y)(\text{Parent}(y,x) \land (\text{Citizen}(y,UK) \lor \text{Resident}(y,UK)))) \Rightarrow \text{CitizenByBirth}(x,UK))\)

j.) Sentence:
A person born outside the UK, one of whose parents is a UK citizen by birth, is a UK citizen by descent.

Translation:
\((\forall x)((\neg \text{Born}(x,UK) \land (\exists y)(\text{Parent}(y,x) \land (\text{CitizenByBirth}(y,UK)) \Rightarrow \text{CitizenByDescent}(x,UK)))\)

k.) Sentence:
Politicians can fool some of the people all of the time, and they can fool all of the people some of the time, but they can’t fool all of the people all of the time.

Translation:

\[ (\forall x)(\text{Politician}(x) \implies (\exists y)((\forall t)\text{CanFool}(x,y,t)) \land ((\forall y)(\exists t)\text{CanFool}(x,y,t)) \land \neg((\forall y)(\forall t)\text{CanFool}(x,y,t))) \]
**Problem 4** (Problem 9.6 of text)

**Total Points : 1 – half a point for each of d and f.**

Horse(x): x is a horse  
Cow(x): x is a cow  
Pig(x): x is a pig  
Mammal(x): x is a mammal  
Parent(x,y): x is a parent of y  
Offspring(x, y): x is an offspring of y

a.) Horses, cows, and pigs are mammals

\( \forall x \text{ Horse}(x) \Rightarrow \text{Mammal}(x) \)  
\( \forall x \text{ Cow}(x) \Rightarrow \text{Mammal}(x) \)  
\( \forall x \text{ Pig}(x) \Rightarrow \text{Mammal}(x) \)

b.) An offspring of a horse is a horse.

\( \forall x, y \text{ Horse}(x) \land \text{Offspring}(y, x) \Rightarrow \text{Horse}(y) \)

c.) Bluebeard is a horse

\( \text{Horse(Bluebeard)} \)

d.) Bluebeard is Charlie’s parent.

\( \text{Parent(Bluebeard, Charlie)} \)

e.) Offspring and parent are inverse relations

\( \forall x,y \text{ Offspring}(x, y) \Rightarrow \text{Parent}(y, x) \)  
\( \forall x,y \text{ Parent}(x, y) \Rightarrow \text{Offspring}(y, x) \)

f.) Every mammal has a parent

\( \forall y \text{ Mammal}(y) \Rightarrow \exists x \text{ Parent}(x, y) \)

**Problem 5**

*not scored.*