CS360 Homework 3– Solution

First Order Logic

1) For each of the following sentences in first-order logic, specify whether it is valid, satisfiable, and/or unsatisfiable:

(a) $P(A) \Rightarrow \forall x P(x)$
Answer: Satisfiable but not valid.

(b) $P(A) \Rightarrow \forall x \neg P(x)$
Answer: Satisfiable but not valid.

(c) $P(A) \Rightarrow \exists x P(x)$
Answer: Valid.

(d) $P(A) \Rightarrow \exists x \neg P(x)$
Answer: Satisfiable but not valid.

2) Solve Problem 9.23 on page 365 of our textbook.

Answer:

(a) Horses are animals:
$\forall x\ (\text{Horse}(x) \Rightarrow \text{Animal}(x))$

The head of a horse is the head of an animal:
$\forall h\ ((\exists y\ (\text{HeadOf}(h, y) \wedge \text{Horse}(y))) \Rightarrow (\exists z\ (\text{HeadOf}(h, z) \wedge \text{Animal}(z))))$

(b) Horses are animals (CNF):
$\forall x\ (\neg\text{Horse}(x) \vee \text{Animal}(x))$
$\neg\text{Horse}(x) \vee \text{Animal}(x)$

The head of a horse is the head of an animal (CNF after negation):
$\neg\forall h\ ((\exists y\ (\text{HeadOf}(h, y) \wedge \text{Horse}(y))) \Rightarrow (\exists z\ (\text{HeadOf}(h, z) \wedge \text{Animal}(z))))$
$\exists h\ \neg((\exists y\ (\text{HeadOf}(h, y) \wedge \text{Horse}(y))) \Rightarrow (\exists z\ (\text{HeadOf}(h, z) \wedge \text{Animal}(z))))$
$\exists h\ \neg(\neg(\exists y\ (\text{HeadOf}(h, y) \wedge \text{Horse}(y))) \vee (\exists z\ (\text{HeadOf}(h, z) \wedge \text{Animal}(z))))$
$\exists h\ ((\exists y\ (\text{HeadOf}(h, y) \wedge \text{Horse}(y))) \wedge \neg(\exists z\ (\text{HeadOf}(h, z) \wedge \text{Animal}(z))))$
$\exists h\ ((\exists y\ (\text{HeadOf}(h, y) \wedge \text{Horse}(y))) \wedge (\forall z\ (\neg\text{HeadOf}(h, z) \vee \neg\text{Animal}(z))))$
$(\text{HeadOf}(H, Y) \wedge \text{Horse}(Y)) \wedge (\neg\text{HeadOf}(H, z) \vee \neg\text{Animal}(z))$
$\text{HeadOf}(H, Y)$, $\text{Horse}(Y)$, $\neg\text{HeadOf}(H, z) \vee \neg\text{Animal}(z)$

(c) We start with the four clauses we have derived in (b):
(1) $\neg$Horse(x) $\vee$ Animal(x)
(2) HeadOf(H, Y)
(3) Horse(Y)
(4) $\neg$HeadOf(H, z) $\vee$ $\neg$Animal(z)

(5) (from 2 and 4, $z = Y$) $\neg$Animal(Y)
(6) (from 1 and 5, $x = Y$) $\neg$Horse(Y)
(7) (from 3 and 6) $\perp$

Rule-Based Systems

3) The knowledge base for a production system is given below:

- If Horse(X) and Offspring (Y,X) then Horse(Y).
- If Parent(X,Y) then Offspring(Y,X).
- If Offspring(X,Y) then Parent(Y,X).
- Horse(Bluebeard).
- Parent(Bluebeard, Charlie).

(a) Use forward chaining to show that Horse(Charlie) is true.

Answer:

- The initial facts in the working memory only unify with the premise of the second rule (X = Bluebeard, Y = Charlie), so we add the conclusion of the second rule, Offspring(Charlie, Bluebeard), to the working memory.
- The new fact, Offspring(Charlie, Bluebeard) unifies with a premise of the first rule (Y = Charlie, X = Bluebeard), and Horse(Bluebeard) is also a fact in the working memory, so we get the conclusion Horse(Charlie).

(b) Use backward chaining to show that Horse(Charlie) is true.

Answer:

- **?-Horse(Charlie).** ?- is used to denote a query that we want to satisfy.
- **?-Horse(X) and ?-Offspring(Charlie, X).** We use the first rule, since it is the only rule that matches Horse(Charlie). We might have needed to branch if there were more rules that matched Horse(Charlie).
- **?-Horse(Bluebeard) and ?-Offspring(Charlie, Bluebeard).** Both the first rule and Horse(Bluebeard) matches Horse(X). We take a guess that X = Bluebeard and try to prove it. If it does not work, we need to backtrack to this point and try the former option.

- **?-Horse(Bluebeard) and ?-Parent(Bluebeard, Charlie)**. Only the second rule matches Offspring(Charlie, Bluebeard).
- **Horse(Charlie) = true.** Since we have reached facts that are in the working memory.