

# Homework 6

## A Logic Sandwich

*Due: 6:00pm on 11/04/09*

### Problem 6.1

Read Chapter 14 from Russel and Norvig.

### Problem 6.2

You can represent a queue as an abstract data structure in predicate logic. The following axioms define the basic operations enqueue, dequeue, and first on a queue:

- $Queue(\epsilon)$
- $\forall x : Queue(x) \Rightarrow x = \epsilon \vee \exists y, z : Queue(y) \wedge x = Enqueue(y, z)$
- $\forall x, y : Queue(x) \Rightarrow Queue(Enqueue(x, y))$
- $\forall y : Front(Enqueue(\epsilon, y)) = y$
- $\forall x, y : Queue(x) \Rightarrow x = \epsilon \vee Front(x) = Front(Enqueue(x, y))$
- $\forall y : Dequeue(Enqueue(\epsilon, y)) = \epsilon$
- $\forall x, y : Queue(x) \Rightarrow x = \epsilon \vee Dequeue(Enqueue(x, y)) = Enqueue(Dequeue(x), y)$

Using the above as the basis for your queue:

- Give the axioms for the function,  $Append(x,y)$  that takes the contents of one queue and appends it onto queue in predicate logic.

*Solution:*

$$\forall x, y : Queue(x) \wedge Queue(y) \Rightarrow Append(x, y) = Append(Enqueue(x, Front(y)), Dequeue(y))$$

$$Append(x, \epsilon) = x$$

- Give the axioms for  $Reverse(x)$ , which reverses the order of the elements in a queue, in predicate logic using  $Append$  from part a.

*Solution:*

$$Reverse(\epsilon) = \epsilon$$

$$\forall x, y : Queue(x) \Rightarrow Reverse(Enqueue(x, y)) = Append(Enqueue(\epsilon, y), Reverse(x))$$

**Problem 6.3**

- a. Transform the following into Skolem normal form:

$$\alpha = \forall x \forall y \forall z [f(f(x, y), z) = f(x, f(y, z)) \wedge \exists e \forall x (f(x, e) = x \wedge \exists x' (f(x, x') = e))]$$

*Solution:*

$$\alpha = \forall x \forall y \forall z [f(f(x, y), z) = f(x, f(y, z)) \wedge \exists e \forall x (f(x, e) = x \wedge \exists x' (f(x, x') = e))] \quad (1)$$

The  $x$  inside the brackets is different from the one outside, so for clarity we rename  $x$  and  $x'$  to  $p$  and  $p'$  accordingly:

$$\alpha = \forall x \forall y \forall z [f(f(x, y), z) = f(x, f(y, z)) \wedge \exists e \forall p (f(p, e) = p \wedge \exists p' (f(p, p') = e))] \quad (2)$$

We convert the problem to prenex normal form:

$$\alpha = \forall x \forall y \forall z \exists e \forall p \exists p' [f(f(x, y), z) = f(x, f(y, z)) \wedge (f(p, e) = p \wedge (f(p, p') = e))] \quad (3)$$

We replace  $e$  with Skolem function  $g(x, y, z)$ . The Skolem function  $g$  does not depend on  $p$  because the quantifier to be removed  $\exists e$  is in the scope of  $\forall x, \forall y,$  and  $\forall z$  but not in the scope of  $\forall p$ .

$$\alpha = \forall x \forall y \forall z \forall p \exists p' [f(f(x, y), z) = f(x, f(y, z)) \wedge (f(p, g(x, y, z)) = p \wedge (f(p, p') = g(x, y, z)))] \quad (4)$$

We replace  $p'$  with Skolem function  $h(x, y, z, p)$ . The Skolem function  $h$  depends on variables  $x, y, z,$  and  $p$  because it is in the scope of  $\forall x, \forall y, \forall z,$  and  $\forall p$ .

$$\alpha = \forall x \forall y \forall z \forall p [f(f(x, y), z) = f(x, f(y, z)) \wedge (f(p, g(x, y, z)) = p \wedge (f(p, h(x, y, z, p)) = g(x, y, z)))] \quad (5)$$

- b. Unify the following atomic sentences or give an argument why this is not possible.  $P$  is a predicate,  $a$  and  $b$  are constants,  $f, g$  and  $h$  are functions and  $x, y, z, x_0, x_1, x_2$  are variables. Assume that standardizing apart has already been performed, i.e., unify the pair of sentences without performing any variable renaming.

- (a)  $\{p(f(f(f(a, x_0), x_1), x_2)), p(f(x_2, f(x_1, f(x_0, a))))\}$

*Solution:*

$$x_2/f(f(a, x_0), x_1), x_1/f(a, x_0), x_0/a$$

- (b)  $\{p(x), p(f(x))\}$

*Solution:*

Occurs check.

- (c)  $\{p(x, h(x, y), y), p(x, z, b)\}$

*Solution:*

$$z/h(x, y), y/b$$

- (d)  $\{p(x, x, y, y), p(x, f(y), y, f(x))\}$

*Solution:*

Occurs check.

### Problem 6.4

Matt is attempting to cook mushrooms for the TAs to enjoy during a grading meeting. This might have some unexpected consequences.

Here is what we know in predicate logic:

Let Matt, Yuri and Ilke be constants. Let  $x$  and  $y$  be variables. Let Meet, MustEatMushroom, HatesMatt, and LikesEachOther be functions.

$$\begin{aligned} & \text{Meet}(\text{Yuri}, \text{Matt}) \wedge \\ & \text{Meet}(\text{Ilke}, \text{Matt}) \wedge \\ & (\forall x, y (\text{Meet}(x, \text{Matt}) \Rightarrow \text{MustEatMushroom}(x)) \wedge \\ & (\text{MustEatMushroom}(x) \Rightarrow \text{HatesMatt}(x)) \wedge \\ & (\text{HatesMatt}(x) \wedge \text{HatesMatt}(y) \Rightarrow \text{LikesEachOther}(x, y))) \end{aligned}$$

- a. Bring the above to Prenex normal form

*Solution:* To bring something into Prenex normal form, we must eliminate implications, move negation inwards, then bring qualifiers to the front.

$$\begin{aligned} & \forall x, y (Meet(Yuri, Matt) \wedge \\ & \quad Meet(Ilke, Matt) \wedge \\ & \quad \neg Meet(x, Matt) \vee MustEatMushroom(x) \wedge \\ & \quad \neg MustEatMushroom(x) \vee HatesMatt(x) \wedge \\ & \quad \neg HatesMatt(x) \vee \neg HatesMatt(y) \vee LikesEachOther(x, y)) \end{aligned}$$

- b. Show  $LikesEachOther(Yuri, Ilke)$

*Solution:*

1,3 / [x = Yuri] MustEatMushroom(Yuri)

2,3 / [x = Ilke] MustEatMushroom(Ilke)

4 / [x = Yuri] HatesMatt(Yuri)

4 / [x = Ilke] HatesMatt(Ilke)