Ling 130 Guide to PS 2

James Pustejovsky

March 1, 2010

General Comments

Many of you have asked about the interaction of quantifiers and the λ-operator. The book doesn't mention anything about this, so I'll mention a few things. First, remember that a quantifier ($\forall$ or $\exists$) can be added to any expression ϕ that is already propositional, i.e., of type t. Of course, any free variables occurring in that expression, ϕ, will be bound by the quantifier, if it refers to that named variable.

(1) a. Given ϕ, we can form $\forall x \phi[x]$;
b. Given ϕ, we can form $\exists x \phi[x]$.

A λ-expression can contain any number of quantifiers, as long as the expression is well-formed. So, now take some specific examples, in (2).

(2) a. Given $\exists x[Love(x, a) \wedge happy(a)]$, we can λ-abstract to get:
b. $\lambda y[\exists x[Love(x, y) \wedge happy(y)]]$:
"the set of all things such that it is happy and there is something that loves it"

Also, the book is a bit unclear about the effect that quantifiers and λ-operators have on the arity of a predicative expression. Arity is the term given to identify how many parameters or arguments an expression has. Normally, this is done by inspection of the argument list:

(3) a. $R(x, y)$: arity is 2;
b. $P(x, y, z)$: arity is 3;
c. . . .

But they sometimes introduce predicates by their type, so that the arity may not be obvious from inspection:

(4) a. R: $e \to (e \to t)$, arity is 2;
b. P: $e \to (e \to (e \to t))$, arity is 3;
c. . . .

So, if you do have a quantifier binding a variable in the predicate P, as typed above, then it will reduce the arity by the value of that quantified term:

(5) a P: $e \to (e \to (e \to t))$, arity is 3;
c. $\exists x[P(x)] : (e \to (e \to t)$, arity is 2.

I will try to clarify this as much as possible throughout the course, and in the literature, it is most common to be able to determine the arity and type "by inspection" of the arguments.

3.3

Some questions have emerged regarding a possible ambiguity in the expression in Problem (3.3a) on Page 81 of the book, repeated below:

(6) $\forall x[\exists y[M(x,y)]]$

Let me try to clarify without giving away the answer, which would not be fair.

(7) $\forall x[\exists y[P(x) \rightarrow Q(x,y)]]$

Now let $P = Human$, and $Q = HaveLiver$. We have:

(8) $\forall x[\exists y[Human(x) \rightarrow HaveLiver(x,y)]]$

Imagine what this is intended to mean: "If you are human, you have a liver." How many livers are there if there are 20 people? Well, 20 most likely. But it's not really stated formally, is it? It is logically conceivable for Mary and Bill to have the same liver. In reality, this is unlikely if not impossible of course. Is it possible for all 20 people to have the same liver? Logically, yes, but not in reality.

Now take the same expression, and let $P = American$ and $Q = LiveInState$.

(9) $\forall x[\exists y[American(x) \rightarrow LiveInState(x,y)]]$

This says something like: "If you are American, then you live in some State." So, Jill lives in California, and Russ lives in Maine, etc. But, there is a possibility that everyone lives in the same state, say Oklahoma. It's allowed by this reading, but not forced. OK?

3.5

Use type deduction to show if the expressions are well-formed. If they are, give the reduced form.

Example 1.

$$\lambda x[\forall y[[Student(x) \wedge Attend(x,y)] \rightarrow Like(x,y)]](j)$$

(10) a. $[Student(x) \wedge Attend(x,y) \rightarrow Like(x,y)] : t$
b. $\forall y[Student(x) \wedge Attend(x,y) \rightarrow Like(x,y)] : t$
c. $\lambda x[\forall y[Student(x) \wedge Attend(x,y) \rightarrow Like(x,y)]] : e \rightarrow t$
d. FA: $\lambda x[\forall y[Student(x) \wedge Attend(x,y) \rightarrow Like(x,y)]] : e \rightarrow t, j : e \Longrightarrow$
$\forall y[Student(j) \wedge Attend(j,y) \rightarrow Like(j,y)] : t$

Example 2.

$$\exists x[\lambda y[Happy(x) \wedge Love(y,x)](j)](b)$$

(11) a. $[Happy(x) \wedge Love(y,x)] : t$
b. $\lambda y[Happy(x) \wedge Love(y,x)] : e \rightarrow t$
c. FA: $\lambda y[Happy(x) \wedge Love(y,x)] : e \rightarrow t, j : e \Longrightarrow$
$[Happy(x) \wedge Love(j,x)] : t$
d. $\exists x[Happy(x) \wedge Love(j,x)] : t$
e. $\exists x[Happy(x) \wedge Love(j,x)](b)$, Ill-formed, because there is a type mismatch due to the "stranded entity", (b).

Example 3.

Type deduction with propositional connectives is done analogously to normal predicative deductions. Consider the connective *and*.

(12) a. *and*: $\lambda p \lambda q[q \wedge p]$
b. *and*: $t \rightarrow (t \rightarrow t)$
c. Frodo lives and Sam lives.
d. $[Lives(f) \wedge Lives(s)]$
e. FA: $\lambda p \lambda q[q \wedge p] : t \rightarrow (t \rightarrow t), Lives(s) : t \Longrightarrow$
$\lambda q[q \wedge Lives(s)] : t \rightarrow t$
f. . . .

3.6

In the book, they make a mistake in the λ-expression for **detest**, if we assume VP composition, with the direct object composing with the verb first. It should read as below instead.

(13) **detest**. $\lambda y \lambda x[Detest(x, y)]$