Philosophy 220

Overlapping Quantifiers

When order matters:

- When a string of quantifiers are all universal, it does not matter in what order the variables are listed.
  - $(\forall x)(\forall y)$…
  - $(\forall y)(\forall x)$…
- When a string of quantifiers are all existential, it does not matter in what order the variables are listed.
  - $(\exists x)(\exists y)$…
  - $(\exists y)(\exists x)$…
- When quantifiers are mixed, the order DOES matter because it matters which variable goes with which quantifier.
  - $(\forall x)(\exists y)Pxy \neq (\exists x)(\forall y)Pxy$

Phrasing into English:

- $(\forall x)(\forall y)$
  - For all of x and all of y…
  - For every pair x and y…
- $(\exists x)(\exists y)$
  - There is an x and there is a y such that…
  - There is a pair x and y such that…
- $(\forall x)(\exists y)$
  - For all of x there is a y such that…
- $(\exists x)(\forall y)$
  - There is an x such that for each y…

Flashback!

- Remember the argument from early in this unit that looked valid in English but was clearly not valid in SL?
  - None of David's friends support Republicans.
  - Sarah Supports Breitlow, and Breitlow is a Republican.
  - ___
  - Sarah is no friend of David's
- We now, at last, have the machinery in PL to symbolize that argument:
  - $(\forall x)[Fxd \supset \sim(\exists y)(Ry \;\&\; Sxy)]$
  - $Ssb \;\&\; Rb$
  - ___
  - $\sim Fsd$

To broaden the scope of a quantifier:

- Each sentence in the left column is equivalent to the sentence to its right (so long as **x** does not occur in **P**), and it is often desirable to make the sentences in the right column so that one can make substitution instances of them.
- For conditional sentences:

1	$(\exists x)Ax \supset P$	$(\forall x)(Ax \supset P)$
2	$(\forall x)Ax \supset P$	$(\exists x)(Ax \supset P)$
3	$P \supset (\exists x)Ax$	$(\exists x)(P \supset Ax)$
4	$P \supset (\forall x)Ax$	$(\forall x)(P \supset Ax)$

Broadening scope with v, &

1	(∃x)Ax v P	(∃x)(Ax v P)
2	(∀x)Ax v P	(∀x)(Ax v P)
3	P v (∃x)Ax	(∃x)(P v Ax)
4	P v (∀x)Ax	(∀x)(P v Ax)
1	(∃x)Ax & P	(∃x)(Ax & P)
2	(∀x)Ax & P	(∀x)(Ax & P)
3	P & (∃x)Ax	(∃x)(P & Ax)
4	P & (∀x)Ax	(∀x)(P & Ax)

Note that such a procedure <u>does not</u> work for biconditional (≡) sentences.