Philosophy 240: Symbolic Logic | Hamilton College
Fall 2009 | Russell Marcus
Mondays, Wednesdays, Fridays: 9am - 9:50am | rmarcus1@hamilton.edu

Class 25 - October 26
Derivations in Predicate Logic II and Changing Quantifiers (§8.2 - §8.3)

**I. More complex derivations**

You may not instantiate a line on which the quantifier is not the main operator.
In this example, line 2 can not be instantiated.

1. (x)(Dx · Ex)
2. (x)Dx ⊃ Fa / (∃x)Fx
3. Dx · Ex 1, UI
4. Dx 3, Simp
5. (x)Dx 4, UG
6. Fa 2, 5, MP
7. (∃x)Fx 6, EG

QED

Similarly, we we can not take off either quantifier in line 1 of the following derivation.

1. (x)(Jx ∨ Kx) ⊃ (∃y)Ly
2. (x)(Jx ∨ Lx)
3. (x)(~Lx ∨ Kx) / (∃x)Lx
4. Jx ∨ Lx 2, UI
5. ~Jx ⊃ Lx 4, DN, Impl
6. ~Lx ∨ Kx 3, UI
7. Lx ⊃ Kx 6, Impl
8. ~Jx ⊃ Kx 5, 7, HS
9. Jx ∨ Kx 8, Impl, DN
10. (x)(Jx ∨ Kx) 9, UG
11. (∃x)Lx 1, 10, MP

QED

You may instantiate the same quantifier twice.
When that quantifier is universal, there are no restrictions.

1. (x)(Mx ⊃ Nx)
2. (x)(Nx ⊃ Ox)
3. Ma · Mb / Na · Ob
4. Ma ⊃ Na 1, UI
5. Ma 3, Simp
6. Na 4, 5, MP
7. Mb ⊃ Nb 1, UI
8. Mb 3, Com, Simp
9. Nb 7, 8, MP
10. Nb ⊃ Ob 2, UI
11. Ob 10, 9, MP
12. Na · Ob 6, 11, Conj

QED

When the quantifier is existential, the second instantiation must go to a new constant.

| | |
|---|---|
| 1. (∃x)(Px • Qx) | |
| 2. (x)(Px ⊃ Rx) | / Ra • Qb |
| 3. Pa • Qa | 1, EI |
| 4. Pa ⊃ Ra | 2, UI |
| 5. Pa | 3, Simp |
| 6. Ra | 4, 5, MP |
| 7. Pb • Qb | 1, EI |
| 8. Qb | 7, Com, Simp |
| 9. Ra • Qb | 6, 8, Conj |

QED

**II. Exercises A**. Derive the conclusions of each of the following arguments.

1.
 1. (x)(Mx ⊃ Nx)
 2. ~Na / ~Ma

2.
 1. (x)(Ox ⊃ ~Px)
 2. (∃x)(Rx · Px) / (∃x)(Rx · ~Ox)

3.
 1. (∃x)Sx ⊃ (x)Tx
 2. (∃x)Ux ⊃ (∃x)Wx
 3. Sb · Ub / (∃x)(Wx · Tx)

4.
 1. (∃x)Gx ⊃(y)(~Hy ∨ Iy)
 2. Gc
 3. ~If / (∃x)~Hx

5.
 1. (∃x)Ax ⊃ (x)(Bx ⊃ Ex)
 2. (∃x)Dx ⊃ (∃x)~Ex
 3. (∃x)(Ax · Dx) / (∃x)~Bx

**III. Exercises B**. Match each sentence in the left column with its equivalent in the right column.

| | |
|---|---|
| 1. Everything is made of atoms. | A. Not everything is made of atoms. |
| 2. Something is made of atoms. | B. It's wrong to say that nothing is made of atoms. |
| 3. Nothing is made of atoms. | C. It's not the case that something is not made of atoms. |
| 4. At least one thing isn't made of atoms. | D. It's false that something is made of atoms. |

Solutions: 1C, 2B, 3D, 4A

## IV. Changing quantifiers

Look at the predicate logic regimentations of each set of equivalent pairs.

| | | |
|---|---|---|
| (x)Ax | is equivalent to | ~(∃x)~Ax |
| (∃x)Ax | is equivalent to | ~(x)~Ax |
| (x)~Ax | is equivalent to | ~(∃x)Ax |
| (∃x)~Ax | is equivalent to | ~(x)Ax |

The rule of **Changing Quantifiers (CQ)**: Any place where you have an expression of one of the above forms, you may replace it with a statement of its logically equivalent form.

Like rules of replacement, CQ is based on logical equivalence, rather than validity, and thus may be used on part of a line.

Another way to look at these four rules.
There are three spaces around each quantifier:

1. Directly before the quantifier
2. The quantifier itself
3. Directly following the quantifier

CQ says that to change a quantifier, you change each of the three spaces.

Add or remove a negation directly before the quantifier.
Switch quantifiers: existential to universal or vice versa.
Add or remove a negation directly after the quantifier.

## V. Some transformations permitted by CQ

'It's not the case that every P is Q' is equivalent to 'something is P and not Q'.

| | |
|---|---|
| ~(x)(Px ⊃ Qx) | |
| (∃x)~(Px ⊃ Qx) | CQ |
| (∃x)~(~Px ∨ Qx) | Impl |
| (∃x)(Px · ~Qx) | Dm, DN |

'It's not the case that something is both P and Q' is equivalent to 'everything that's P is not Q,' and to 'everything that's Q is not P'.

| | |
|---|---|
| ~(∃x)(Px · Qx) | |
| (x)~(Px · Qx) | CQ |
| (x)(~Px ∨ ~Qx) | DM |
| (x)(Px ⊃ ~Qx) | Impl |
| (x)(Qx ⊃ ~Px) | Trans, DN |

## VI. Sample derivations using CQ

1.

| | | |
|---|---|---|
| 1. (∃x)Lx ⊃ (∃y)My | | |
| 2. (y)~My | /~La | |
| 3. ~(∃y)My | 2, CQ | |
| 4. ~(∃x)Lx | 1, 3, MT | |
| 5. (x)~Lx | 4, CQ | |
| 6. ~La | 5, UI | Note: You may not use EI to get this conclusion! |

QED

2.

| | | |
|---|---|---|
| 1. (x)[(Ax · Bx) ⊃ Ex] | | |
| 2. ~(x)(Ax ⊃ Ex) | / ~(x)Bx | |
| 3. (∃x)~(Ax ⊃ Ex) | | 2, CQ |
| 4. (∃x)~(~Ax ∨ Ex) | | 3, Impl |
| 5. (∃x)(Ax · ~Ex) | | 4, DM, DN |
| 6. Aa · ~Ea | | 5, EI |
| 7. (Aa · Ba) ⊃ Ea | | 1, UI |
| 8. ~Ea | | 6, Com, Simp |
| 9. ~(Aa · Ba) | | 7, 8, MT |
| 10. ~Aa ∨ ~Ba | | 9, DM |
| 11. Aa | | 6, Simp |
| 12. ~Ba | | 10, 11, DN, DS |
| 13. (∃x)~Ba | | 12, EG |
| 14. ~(x)Bx | | 13, CQ |

QED

3.

| | | |
|---|---|---|
| 1. (x)~Dx ⊃ (x)Ex | | |
| 2. (∃x)~Ex /(∃x)Dx | | |
| 3. ~(x)Ex | 2, CQ | |
| 4. ~(x)~Dx | 1, 3, MT | |
| 5. (∃x)Dx | 4, CQ | |

QED Note: No instantiation!

**Exercises B**. Derive the conclusions of each of the following arguments.

1.
1. ~(∃x)Hx
2. (x)~Hx ⊃ (z)Iz / Ia

2.
1. (∃x)(Hx · Gx) ⊃ (x)Ix
2. ~Ia / (x)(Hx ⊃ ~Gx)

3.
1. (∃x)(Ax ∨ Bx) ⊃ (x)Dx
2. (∃x)~Dx / ~(∃x)Ax

4.
1. (x)~Fx ⊃ (x)~Gx / (∃x)Gx ⊃ (∃x)Fx

5. 1. (∃x)~Ax ⊃ (x)~Bx
2. (∃x)~Ax ⊃ (∃x)Bx
3. (x)(Ax ⊃ Fx) / (x)Fx

Solutions may vary.