PHIL 331/MATH 281: Week 8

The syntax of QC

1. To specify the syntax of QC we specify a set of symbols, then a set of **terms**, then a set of wffs.

   a. Symbols:

   We have two brackets: '(' and ')'.

   We have denumerably many sentence letters: 'A', 'B', 'C', 'P_1, 'P_2', 'R', …

   We have five truth-functional connectives: '$\neg$', '$\wedge$', '$\vee$', '$\rightarrow$', and '$\leftrightarrow$'.

   (So far this is the same as for PC)

   We have denumerably many **individual constants**: 'a', 'b', 'c', 'a_1', 'a_{35}', …

   We have denumerably many **predicate letters**. These are either **1-place**: 'A^1', 'B^1', 'C^1', '${P_1}^1$', '${P_2}^1$', …; **2-place**: 'A^2', 'P^2', '${Q_1}^2$', '${Q_2}^2$', …; or **3-place**: 'A^3', 'P^3', '${Q_1}^3$', '${Q_2}^3$', …; and so on. The superscripts show the number of places of the predicate letter. They are required, but are sometimes left understood. We sometimes think of the sentence letters as **0-place** predicate letters.

   One 2-place predicate letter is singled out as a logical symbol (i.e. it will always be interpreted in the same way): '='.

   We have denumerably many (**individual**) **variables**: 'w', 'x', 'y', 'z', 'x_1', 'x_{22}', … (note the optional subscripts).

   We have two **quantifiers**: '$\forall$' (universal), '$\exists$' (existential). (Maybe these should be called *determiners*.)

   [Sometimes **function letters** are included.]

   b. Terms:

   The class of terms is defined recursively according to the following formation rules:

   i. Every individual constant is a term
   ii. Every variable is a term
   iii. Nothing else is a term.

   [If we had function letters then there would be an extra formation rule here.]

c. wffs:

The class of wffs is defined recursively according to the following formation rules:

i. Every sentence letter is a wff.
ii. If Π is an *n*-place predicate letter and $\tau_1, \ldots, \tau_n$ are terms then '$\Pi\tau_1\ldots\tau_n$' is a wff. As a special case we have that if τ_1 and τ_2 are terms then '$\tau_1 = \tau_n$' is a wff.

These first two kinds of wff are called **atomic wffs** – they do not contain connectives. Note that they might contain variables. Examples: 'A', 'B_5', 'Ca', 'C^1a', 'C^1x', 'Qxy', 'A^3abc', 'P^4xaab', '$a = a$', '$x = y$', '$d_1 = x_3$',

Next we define non-atomic or **complex wffs**:

iii. If φ and ψ are wffs then so are '$\neg\varphi$', '$(\varphi \wedge \psi)$', '$(\varphi \vee \psi)$', '$(\varphi \rightarrow \psi)$', and '$(\varphi \leftrightarrow \psi)$'.

Examples:
'$\neg A$', '$(A \wedge (P \rightarrow Q))$', '$\neg Fxy$', '$(B^1x \wedge C^2xy)$', '$(\neg B^1a \rightarrow (B^1x \wedge C^2xy))$'.

iv. If φ is a wff and υ is an individual variable then '$\forall\upsilon\varphi$' and '$\exists\upsilon\varphi$' are wffs. Note that φ need not contain the variable υ. We can think of '$\forall\upsilon$' and '$\exists\upsilon$' as being connectives (so we have denumerably many connectives). They are also called quantifiers.

Examples:
'$\forall xAx$', '$\exists yFyy$', '$\exists x\neg Ba$', '$\forall x(Bx \vee Cxy)$', '$\exists y(\exists x\neg B^1a \leftrightarrow \forall x(B^1x \vee C^2xy))$', '$(\forall x(Bx \vee Cxy) \wedge \exists y(\exists x\neg Ba \leftrightarrow \forall x(Bx \vee Cxy)))$'.

v. Nothing else is a wff.

The following are not wffs:
'B^1aa', 'C^2a', 'C^2xC^2xy', '$(\exists xC^2xy)$', 'AB'

2. Note that the last formation rule yields some seemingly odd wffs: '$\exists x(P \wedge Q)$', '$\exists xFa$', '$\forall x(Fy \rightarrow Gz)$'.

3. For any given formula of QC we can use these formation rules to prove that it is or is not a wff, as we did for PC (but we won't do that).

4. Exercises

a. Which of the following formulae of QC are wffs?
   i. 'a'
   ii. 'F^2abc'
   iii. 'F^2ax'

iv. ‘a = x’
v. ‘Fx = Gy’
vi. ‘(A ∧ A^2ab)’
vii. ‘A^2PQ’
viii. ‘(∀xFax → Fax)’
ix. ‘∀x(Fax → ∃xFax)’
x. ‘∀x(∃yFyy → Fxy)’
xi. ‘∀a(∀yFyy ⊨ Faa)’
xii. ‘(∃x∀yFyx ∧ ∀x∃zFzy)’
xiii. ‘∀x(Fab ∧ (Ga → ∃x(Fxb ∧ Gx)))’
xiv. ‘∀x(Fab ∧ ((Ga → ∃xFxb) ∧ Gx))’
xv. ‘∃x∧∃yFxy’
xvi. ‘∀x(P → Fx) ⊨ (P → ∀xFx)’

Unique decomposition

1. Every complex wff has exactly one of the following forms: ‘¬φ’, ‘(φ ∧ ψ)’, ‘(φ ∨ ψ)’, ‘(φ → ψ)’, ‘(φ ↔ ψ)’, ‘∃υφ’, or ‘∀υφ’. We will not prove this (we could modify the PC proof).

2. Each wff thus has a uniquely defined main connective and immediate constituents.

   a. The main connective of ‘∃υφ’ is ‘∃υ’ and its immediate constituent is φ
   b. The main connective of ‘∀υφ’ is ‘∀υ’ and its immediate constituent is φ

   The main connective of ‘(∀xCxb → Bx)’ is ‘→’.
   The main connective of ‘∀x(Cxb → Bx)’ is ‘∀x’.

3. A wff of the form ‘∃υφ’ is called an **existentially quantified** wff. A wff of the form ‘∀υφ’ is called a **universally quantified** wff.

4. We could represent the constituent structure of a wff using a tree, as for PC, but we won’t.

5. Exercises

   a. Identify the main connective and immediate constituent(s) of the following wffs:
   i. ‘∃x(Fx ∧ Gx)’
   ii. ‘(∃xFx ∧ ∃xGx)’
   iii. ‘∃x∀y∃zFxyz’
   iv. ‘¬∀x(Ax → B)’
   v. ‘(A → ∃z(Pz → ∀wPw))’

Scope and bondage

1. We define the **scope** of an occurrence of a connective in a wff φ to be the smallest wff in φ that contains that occurrence of the connective.

   Examples:
   In '∃*x*(F*x* ∧ ∀*y*G*y*)' the scope of the occurrence of '∀*y*' is '∀*y*G*y*', the scope of the occurrence of '∧' is '(F*x* ∧ ∀*y*G*y*)', and the scope of the occurrence of '∃*x*' is '∃*x*(F*x* ∧ ∀*y*G*y*)'.

2. If an occurrence of a connective C_1 occurs within the scope of an occurrence of a connective C_2, then we say that C_2 has **wider scope** than C_1, or that C_1 has **narrower scope** than C_2.

3. An occurrence of a variable υ in a wff is a **bound occurrence** if it occurs within the scope of an occurrence of '∀υ' or '∃υ' in that wff. Otherwise it is a **free occurrence**. Examples:

   a. In '(A*x* → B*x*)', both occurrences of '*x*' are free occurrences.
   b. In '∀*x*(A*x* → B*x*)', all three occurrences of '*x*' are bound occurrences.
   c. In '(∀*x*A*x* → B*x*)', the first two occurrences of '*x*' are bound occurrences, but the third occurrence of '*x*' is a free occurrence.
   d. In '∀*x*(G*x* ∧ F*y*)', the two occurrences of '*x*' are bound occurrences, but the occurrence of '*y*' is a free occurrence.
   e. In '∃*y*(∀*y*G*y* ∧ F*y*)', every occurrence of '*y*' is a bound occurrence. (Which quantifier binds the third occurrence?)

4. A variable υ is said to be **free** in a wff φ just in case it has a free occurrence in φ (note: just one free occurrence will do – it might also have some bound occurrences). Otherwise it is said to be **bound**. So it is bound in φ iff it has no free occurrences in φ.

   Examples:
   a. '(∀*x*(A*xy* ∨ B*x*) ∧ B*x*)'
   b. '∃*x*(A*x* → ∀*x*B*x*)'.

5. A wff that contains at least one free variable is said to be **open**. Otherwise it is said to be **closed**.

   Open: 'F*x*', '∀*y*G*xyz*', '(∃*x*F*x* → (A ∧ G*x*))', '∃*x*G*y*'
   Closed: 'A', 'Fab', '∃*x*Fa*x*', '∀*x*∃*y*F*xy*', '∃*x*Gab'

6. Note:
   a. '¬φ' is closed iff φ is closed
   b. '(φ ∧ ψ)' is closed iff φ is closed and ψ is closed
   c. '(φ ∨ ψ)' is closed iff φ is closed and ψ is closed
   d. '(φ → ψ)' is closed iff φ is closed and ψ is closed
   e. '(φ ↔ ψ)' is closed iff φ is closed and ψ is closed

f. ‘∀υφ’ is closed iff φ is closed or υ is the only unbound variable in φ
g. ‘∃υφ’ is closed iff φ is closed or υ is the only unbound variable in φ

(These all need proof.)

7. Note also that every closed wff has exactly one of the following forms:
   a. ‘¬φ’ for some closed wff φ
   b. ‘(φ ∧ ψ)’ for some closed wffs φ and ψ
   c. ‘(φ ∨ ψ)’ for some closed wffs φ and ψ
   d. ‘(φ → ψ)’ for some closed wffs φ and ψ
   e. ‘(φ ↔ ψ)’ for some closed wffs φ and ψ
   f. ‘∀υφ’ for some φ which is either a closed wff or a 1-place predicate whose only unbound variable is υ
   g. ‘∃υφ’ for some φ which is either a closed wff or a 1-place predicate whose only unbound variable is υ

8. A wff which contains exactly *n* free variables is said to be an *n*-place **predicate**. So a closed wff is a 0-place predicate.

   Note that ‘$\forall x F^3 xxy$’ is a 1-place predicate, but not a 1-place predicate letter, and that ‘F^3’ is a 3-place predicate letter, but not a 3-place predicate. Note that ‘F*xxy*’ is a 2-place predicate.

9. Exercises

   a. Classify each occurrence of a variable as bound or free:
      i. ‘$\forall x_3(\forall x_1 A_1 x_1 x_2 \rightarrow A_1 x_3 a_1)$’
      ii. ‘(∀*y*P*zy* → ∀zP*zy*)’
      iii. ‘(∀*y*∃*x*A*xy* ∨ ¬∀*x*A*yx*)’

   b. List any free variables in the following wffs:
      i. ‘(A → ∃*x*(P*x* ∧ P*y*))’
      ii. ‘∀*y*(F*x* ∧ Gz)’
      iii. ‘(P*x* ∨ ∃*x*(Q*x* ∧ R*x*))’

   c. Classify the following wffs as open or closed:
      i. ‘∀*x*A*x*’
      ii. ‘∃*y*F*yy*’
      iii. ‘∃*x*¬Ba’
      iv. ‘∀*x*(B*x* ∨ C*xy*)’
      v. ‘∃*y*(∃*x*¬Ba ↔ ∀*x*(B*x* ∨ C*xy*))’
      vi. ‘(∀*x*(B*x* ∨ C*xy*) ∧ ∃*y*(∃*x*¬Ba ↔ ∀*x*(B*x* ∨ C*xy*)))’
      vii. ‘(A → ∃z(Pz → ∀*w*P*w*))’

Substitution

1. If φ is a wff, τ is a term, and υ is a variable, define $\varphi(\tau/\upsilon)$ to be the wff that results from substituting τ for all free occurrences of υ in φ (there may not be any).

   Examples:
   a. ‘R*xy*’(‘a’/‘*x*’) = ‘Ra*y*’
   b. ‘∃*y*R*xy*’(‘a’/‘*x*’) = ‘∃*y*Ra*y*’
   c. ‘∃*x*R*xy*’(‘a’/‘*x*’) = ‘∃*x*R*xy*’ (‘*x*’ is not free in ‘∃*x*R*xy*’)
   d. ‘∃*y*B*x*’(‘*y*’/‘*x*’) = ‘∃*y*B*y*’ (note that a free variable becomes bound)

2. Sometimes we want to avoid certain substitutions into wffs, as in the last example above. This is a case in which ‘*x*’ is free in ‘∃*y*B*x*’, but it is not **free for** ‘*y*’ (but it is free for ‘z’).

   Why? We’d like it to be the case that ‘$\forall x \exists y \neg(x = y)$’ ⊨ ‘$\exists y \neg(\tau = y)$’, but we need to place a restriction on what τ can be (it cannot be ‘*y*’, but any other term will do).

3. Suppose that φ is a wff, τ is a term, and υ is a variable. If no occurrence of a variable in τ is bound in $\varphi(\tau/\upsilon)$, then we say that υ is **free for** τ in φ (or: τ is *substitutable for* υ).

   Examples:
   a. In ‘∃*y*B*x*’, ‘*x*’ is free for ‘*z*’, but it is not free for ‘*y*’. It is free for any variable other than ‘*y*’. It is also free for any individual constant.
   b. In ‘(∀*x*B*x* → C*y*)’, ‘*x*’ is not free, ‘*y*’ is free, and ‘*y*’ is free for ‘*x*’ (in fact, ‘*y*’ is free for any variable).
   c. In ‘∀*x*(B*x* → C*y*)’, ‘*x*’ is not free, ‘*y*’ is free, but ‘*y*’ is not free for ‘*x*’ (but ‘*y*’ is free for any variable other than ‘*x*’).

   Note that υ is free for τ in φ iff no free occurrence of υ in φ lies within the scope of ‘$\exists\omega$’ or ‘$\forall\omega$’, where ω is a variable that occurs in τ.

4. Note that υ is trivially free for τ in φ if υ has no free occurrences in φ, or if τ has no variables (i.e. it is an individual constant), or if υ is the only variable in τ.

5. Exercises

   a. Find the following, and say whether the variable is free for the substituting term:
      i. ‘A*xcy*’(‘c’/‘*x*’)
      ii. ‘A*xcy*’(‘c’/‘z’)
      iii. ‘(A*xcy* ∧ Bcz)’(‘c’/‘z’)
      iv. ‘(A*xcy* ∧ Bcz)’(‘*y*’/‘z’)
      v. ‘(∀*x*A*xcy* ∧ B*xz*)’(‘a’/‘*x*’)
      vi. ‘∀*x*(A*xcy* ∧ B*xz*)’(‘a’/‘*x*’)
      vii. ‘∀*y*(A*xcy* ∧ B*xz*)’(‘*y*’/‘*x*’)
      viii. ‘∃z(∀*x*A*xcy* ∧ B*xz*)’(‘*y*’/‘*x*’)