PHIL 331/MATH 281: Week 9

The semantics of QC

1. We give meaning to QC by interpreting its non-logical symbols (i.e. its sentence letters, predicate letters, and individual constants).

2. To do this we do the following:

   - We specify a non-empty set, called the **domain** of the interpretation.
   - To each sentence letter we assign a truth value.
   - To each individual constant we assign a member of the domain.
   - To each *n*-place predicate letter ($n \geq 1$) we assign a set of *n*-tuples of members of the domain.

3. The thing assigned to a symbol σ by an interpretation I is sometimes called its **denotation** on I, written $|\sigma|_I$ (we often drop the subscript).

4. Often we only partially interpret the language. Here is an example of a partial interpretation I:

   - Let the domain of I be the set of natural numbers
   - Let $|\text{‘P’}|_I = T$ and $|\text{‘Q’}|_I = F$
   - Let $|\text{‘a’}|_I = 2$ and $|\text{‘b’}|_I = 3$
   - Let $|\text{‘E’}|_I = \{x\text{: } x \text{ is even}\}$, and let $|\text{‘L’}|_I = \{\langle x, y\rangle\text{: } x \text{ is less than } y\}$

5. Sometimes we treat some non-logical symbols as if they were logical symbols, so that they are always interpreted in the same way. This is how we are treating the 2-place predicate letter ‘=’.

6. Sometimes it is helpful to use special symbols to make it easier to remember how they are being interpreted. On the interpretation above we might use ‘2’ instead of ‘a’ to denote 2, ‘3’ instead of ‘b’ to denote 3, and ‘<’ instead of ‘L’ to denote $\{\langle x, y\rangle\text{: } x \text{ is less than } y\}$. And sometimes it is helpful to use infix notation for predicates: rather than writing ‘$<xy$’ we might write ‘$x < y$’ (this is what we do for ‘=’).

7. Next we define what it is for a wff φ to be true on an interpretation I.

8. For wffs that do not contain variables the idea is simple enough: The atomic wff ‘Pa’ is true on an interpretation iff |‘a’| is a member of |‘P’|; the atomic wff ‘Rab’ is true iff <|‘a’|, |‘b’|> is a member of |‘R’|; and so on. Then the truth values of the complex wffs ‘¬φ’, ‘(φ ∧ ψ)’, ‘(φ ∨ ψ)’, ‘(φ → ψ)’, ‘(φ ↔ ψ)’ are determined as they are in PC.

   So on the interpretation given above we have:
   ‘Ea’ is true (because 2 is even)
   ‘Lba’ is false (because 3 is not less than 2)

‘(Ea ∧ Lba)’ is false (because ‘Lba’ is false)

9. For wffs that contain variables things are a bit more complicated. The problem with the atomic wff ‘P*x*’ is that an interpretation does not assign a value to ‘*x*’. And the problem with the complex wffs ‘∀*x*P*x*’ and ‘∃*x*P*x*’ is that we do not yet have recursive rules to deal with the connectives ‘∀*x*’ and ‘∃*x*’.

10. What we do is first define what it is for a wff to be true on an interpretation *relative to* an assignment of values to the variables. Then we define what it is for a wff to be true on an interpretation (simpliciter).

11. <u>Exercises</u>

    a. Consider an interpretation on which the domain is the natural numbers, ‘a’ denotes the number 2, ‘b’ denotes the number 3, ‘P’ denotes the set of prime numbers, and ‘S’ denotes the set {<*x*, *y*>: the successor of *x* is *y* (i.e. *x* + 1 = *y*)}. Find the truth value of the following on this interpretation:
    i. ‘Pa’
    ii. ‘(Pb → Pb)’
    iii. ‘Sba’
    iv. ‘(Saa ∨ Sab)’
    v. ‘(¬Pb ↔ ¬Sbb)’

    b. Consider an interpretation on which the domain is the set of U.S. states and cities, ‘c’ denotes California, ‘s’ denotes Sacramento, ‘f’ denotes Florida, ‘m’ denotes Miami, ‘C’ denotes the set {<*x*, *y*>: the capital of *x* is *y*}, and ‘W’ denotes the set {<*x*, *y*>: *x* is west of *y*}. Find the truth value of the following on this interpretation:
    i. ‘(Ccs ∧ Cfm)’
    ii. ‘(Ccs → Cfm)’
    iii. ‘Wcf ’
    iv. ‘Wmf’
    v. ‘(Wsm ↔ ¬Wms)’

    c. Find a single interpretation on which the following wffs all come out true: ‘Pcw’, ‘Pmw’, ‘Mc’, ‘Fm’, ‘Mw’, ‘(¬m = c)’.

<u>Truth relative to an assignment</u>

1. Suppose that we have an interpretation I of QC. An **assignment** is a function which maps each variable of QC to a member of the domain of I. So if α is an assignment and υ is a variable then α(υ) is the value that α assigns to υ.

2. We can extend an assignment α to include the individual constants of QC as well. If κ is an individual constant then define α(κ) to be the thing that I assigns to κ. We thus get a function that maps every *term* of QC to a member of the domain.

3. We now define the truth a wff on I *relative to an assignment* α recursively, as follows:

- Λ is true relative to α iff |Λ| = T
- ‘$\Pi\tau_1\ldots\tau_n$’ is true relative to α iff <α(τ_1), …, α(τ_n)> ∈ |Π|
- ‘¬φ’ is true relative to α iff φ is not true relative to α
- ‘(φ ∧ ψ)’ is true relative to α iff φ is true relative to α and ψ is true relative to α
- ‘(φ ∨ ψ)’ is true relative to α iff φ is true relative to α or ψ is true relative to α
- ‘(φ → ψ)’ is true relative to α iff φ is not true relative to α or ψ is true relative to α
- ‘(φ ↔ ψ)’ is true relative to α iff either φ is true relative to α and ψ is true relative to α or φ is not true relative to α and ψ is not true relative to α
- ‘∀υφ’ is true relative to α iff φ is true relative to every υ-variant of α
- ‘∃υφ’ is true relative to α iff φ is true relative to some υ-variant of α

A υ-variant of α is any assignment that assigns the same values as α does to every variable other than υ (it may or may not assign the same value to υ). So it differs from υ either not at all or only in the value that it assigns to υ.

4. *Example*. Let I be an interpretation whose domain is the natural numbers, and such that |‘a’| = 2, |‘b’| = 3, |‘E’| = {*x*: *x* is even}, and |‘L’| = {<*x*, *y*>: *x* < *y*}. Let α be some assignment. (Note that no matter which assignment α is, α(‘a’) = 2 and α(‘b’) = 3.)

- **‘Ea’** is true on I relative to α iff α(‘a’) ∈ |‘E’|; that is, iff 2 ∈ {*x*: *x* is even}; that is, iff **2 is even**. Since no matter what α is 2 is even, ‘Ea’ is true on I relative to every assignment.

- **‘Lba’** is true on I relative to α iff <α(‘b’), α(‘a’)> ∈ |‘L’|; that is, iff <3, 2> ∈ {<*x*, *y*>: *x* < *y*}; that is, iff **3 < 2**. Since no matter what α is it is not the case that 3 < 2, ‘Lba’ is false on I relative to every assignment.

- **‘E*x*’** is true on I relative to α iff α(‘*x*’) ∈ |‘E’|; that is, iff α(‘*x*’) ∈ {*x*: *x* is even}; that is, iff **α(‘*x*’) is even**. So ‘E*x*’ is true on I relative to some assignments (ones that assign an even number to ‘*x*’) and false on I relative to some assignments (ones that assign an odd number to ‘*x*’).

- **‘(E*x* → E*x*)’** is true on I relative to α iff either ‘E*x*’ is not true on I relative to α or ‘E*x*’ is true on I relative to α; that is, iff either α(‘*x*’) ∉ |‘E’| or α(‘*x*’) ∈ |‘E’|; that is, iff **either α(‘*x*’) is not even or α(‘*x*’) is even**. Since no matter what α is either α(‘*x*’) is not even or α(‘*x*’) is even, ‘(E*x* → E*x*)’ is true on I relative to every assignment.

- **‘L*xy*’** is true on I relative to α iff <α(‘*x*’), α(‘*y*’)> ∈ |‘L’|; that is, iff <α(‘*x*’), α(‘*y*’)> ∈ {<*x*, *y*>: *x* < *y*}; that is, iff **α(‘*x*’) < α(‘*y*’)**. So ‘L*xy*’ is true on I relative to some assignments and false on I relative to some assignments.

- **‘∃*x*E*x*’** is true on I relative to α iff ‘E*x*’ is true on I relative to α', for some α' which is an ‘*x*’-variant of α; that is, iff α'(‘*x*’) ∈ |‘E’|, for some α' which is an ‘*x*’-variant of

α; that is, iff α'('*x*') is even, for some α' which is an '*x*'-variant of α; that is, iff **some natural number is even**. Since no matter what α is some natural is even, '∃*x*E*x*' is true on I relative to every assignment.

- **'∀*x*E*x*'** is true on I relative to α iff 'E*x*' is true on I relative to α', for every α' which is an '*x*'-variant of α; that is, iff α'('*x*') ∈ |'E'|, for every α' which is an '*x*'-variant of α; that is, iff α'('*x*') is even, for every α' which is an '*x*'-variant of α; that is, iff **every natural number is even**. Since no matter what α is it is not the case that every natural number is even, '∀*x*E*x*' is false on I relative to every assignment.

- **'∃*x*∃*y*L*xy*'** is true on I relative to α iff '∃*y*L*xy*' is true on I relative to α', for some α' which is an '*x*'-variant of α; that is, iff 'L*xy*' is true on I relative to α'', for some α'' which is a '*y*'-variant of α', for some α' which is an '*x*'-variant of α; that is, iff α''('*x*') < α''('*y*'), for some α'' which is a '*y*'-variant of α', for some α' which is an '*x*'-variant of α; that is, iff **there is some natural number that is less than some natural number**. Since no matter what α is there is some natural number that is less than some natural number, '∃*x*∃*y*L*xy*' is true on I relative to every assignment.

Some other results that can be derived:

- **'∃*x*L*x*b'** is true on I relative to α iff **some natural number is less than 3**.
- **'∀*x*La*x*'** is true on I relative to α iff **2 is less than every natural number**.
- **'∀*x*∃*y*L*xy*'** is true on I relative to α iff **for every natural number there is some natural number that it is less than**.
- **'∃*x*∀*y*L*xy*'** is true on I relative to α iff **some natural number is less than every natural number**.
- **'∃*x*∀*y*L*yx*'** is true on I relative to α iff **some natural number is such that every natural number is less than it**.
- **'∃*x*∃*y*(L*xy* ∧ L*yx*)'** is true on I relative to α iff **some pair of natural numbers are less than each other**.
- **'∀*x*∀*y*(L*xy* ∨ L*yx*)'** is true on I relative to α iff **for every pair of natural numbers at least one is less than the other**.

5. Note that for every interpretation I, every wff φ (open or closed) and every assignment α, either φ is true on I relative to α, or φ is false on I relative to α (i.e. for truth on an interpretation relative to an assignment there are no truth value gaps).

6. Result: for every interpretation I and every <u>closed</u> wff φ, either φ is true relative to every assignment or φ is false relative to every assignment.

   Proof. By induction on the number of connectives in φ (details omitted). ■

7. The corresponding result does not obtain for <u>open</u> wffs. On the interpretation above, the open wff 'E*x*' is true relative to some assignments and false relative to others.

8. Note that some open wffs are true on some interpretations relative to every assignment (e.g. '(E*x* → E*x*)' on the interpretation above), and some are false on some interpretations relative to every assignment (e.g. '(E*x* ∧ ¬E*x*)' on the interpretation above).

9. <u>Exercises</u>

   Let I be the following interpretation: the domain is the English alphabet, 'a' denotes 'a', 'b' denotes 'b', etc., |'V'| = {*x*: *x* is a vowel}, |'P'| = {<*x*, *y*>: *x* precedes *y* in alphabetical order}. Suppose that α is an assignment such that α('*x*') = 'd' and α('*y*') = 'e'. Find the truth values of the following wffs on I relative to α:

   a. 'Va'
   b. 'Pgf'
   c. 'V*x*'
   d. '(V*y* ∧ ¬V*x*)'
   e. '∃*x*V*x*'
   f. '∃*x*(V*x* ∧ V*y*)'
   g. '∃*x*∀*y*P*xy*'
   h. '∃*x*∀*y*(¬*x* = *y* → P*xy*)'

<u>Sequences and satisfaction</u>

1. Sometimes rather than talking about assignments and truth relative to an assignment logicians talk about sequences and satisfaction by sequences. This is just a terminological alternative.

2. Suppose we have an interpretation I and an assignment α. Since there are denumerably many individual variables, we can specify an ordering of them. Fix such an ordering, say <'*x*', '*z*', 'y_{23}', 'x_3', …>. Given this ordering of the variables, an assignment α determines a **sequence** of members of the domain: <α('*x*'), α('*z*'), α('y_{23}'), α('x_3'), …>. Conversely, a sequence <s_1, s_2, s_3, …> of members of the domain determines an assignment α: α('*x*') = s_1, α('*z*') = s_2, α('y_{23}') = s_3, and so on. So given an ordering of the variables, there is a 1-1 correspondence between assignments and sequences – we can think of them as being the same thing.

3. So rather than talking about a wff being true relative to an assignment, we can talk about it being true relative to a sequence, or being **true of** a sequence, or being **satisfied by** a sequence.

4. We will talk about assignments instead of sequences.

5. On either approach we can think of a wff φ on an interpretation I as having an **extension**: either the set of assignments {α: φ is true on I relative to α}, or the set of sequences {σ: σ satisfies φ on I}.

<u>Truth</u>

1. We now define what it is for a wff φ to be true on an interpretation I (simpliciter):

   - φ is said to be **true** on I if it is true on I relative to every assignment.
   - φ is said to be **false** on I if it is false on I relative to every assignment.

2. We saw above that if φ is a <u>closed</u> wff then on every interpretation I either φ is true on I relative to every assignment, or φ is false on I relative to every assignment. So either φ is true on I (simpliciter) or φ is false on I (simpliciter).

3. We also saw that for some <u>open</u> wffs φ (e.g. ‘E*x*’) there are some interpretations I such that φ is true on I relative to some assignments and φ is false on I relative to some assignments. So φ is neither true on I nor false on I.

   This makes dealing with open wffs tricky – they can lack truth values on an interpretation. We will often restrict our attention to just closed wffs.

4. Note that some open wffs can be true (simpliciter) on an interpretation. ‘(E*x* → E*x*)’ is true on the interpretation above, and is in fact true on every interpretation. And some open wffs can be false (simpliciter) on an interpretation. ‘(E*x* ∧ O*x*)’ is false on the interpretation above (but on other interpretations it is neither true nor false).

   This makes dealing with open wffs also kind of counterintuitive.

5. *Example*. Let I be an interpretation whose domain is the natural numbers, and on which ‘a’ denotes 2, ‘b’ denotes 3, and ‘E’ denotes the set of even numbers. Then:

   - ‘Ea’ is true on I
   - ‘Eb’ is false on I
   - ‘E*x*’ is neither true nor false on I
   - ‘(E*x* → E*x*)’ is true on I
   - ‘(E*x* ∧ ¬E*x*)’ is false on I
   - ‘∃*x*E*x*’ is true on I
   - ‘∀*x*E*x*’ is false on I

6. Note the following:

   - Whether or not ‘Ea’ is true or false on an interpretation depends on the interpretation. But on every interpretation it is either true or false.
   - ‘(Ea → Ea)’ is true on every interpretation.
   - ‘(Ea ∧ ¬Ea)’ is false on every interpretation.
   - Whether or not ‘Ex’ is true or false or neither on an interpretation depends on the interpretation. But there is an interpretation on which it is each of these three things.
   - ‘(E*x* → E*x*)’ is true on every interpretation.

- ‘$(\mathrm{E}x \wedge \neg\mathrm{E}x)$’ is false on every interpretation.

7. In PC we have the following results, for all wffs φ and ψ and interpretations I:

   - ‘$\neg\varphi$’ is true on I iff φ is not true on I
   - ‘$(\varphi \wedge \psi)$’ is true on I iff φ is true on I and ψ is true on I
   - ‘$(\varphi \vee \psi)$’ is true on I iff φ is true on I or ψ is true on I
   - ‘$(\varphi \rightarrow \psi)$’ is true on I iff φ is not true on I or ψ is true on I
   - ‘$(\varphi \leftrightarrow \psi)$’ is true on I iff either φ is true on I and ψ is true on I or φ is not true on I and ψ is not true on I

   In QC these results still hold, but only if we restrict ‘φ’ and ‘ψ’ to ranging over closed wffs. If we include open wffs then these results do not hold. On the interpretation above: ‘Ex’ is not true, but it is not the case that ‘$\neg\mathrm{E}x$’ is true; ‘$(\mathrm{E}x \vee \neg\mathrm{E}x)$’ is true, but neither ‘Ex’ is true nor ‘$\neg\mathrm{E}x$’ is true.

8. Exercises

   Let I be the following interpretation: the domain is the natural numbers, ‘a’ denotes 2, ‘E’ denotes the set of even numbers, and ‘R’ denotes $\{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 3\rangle\}$. Find the truth value on I (if any) of the following wffs:

   a. ‘$(\mathrm{Raa} \rightarrow \mathrm{Ea})$’
   b. ‘$\exists x\mathrm{E}x$’
   c. ‘$\forall x\forall y(\mathrm{R}xy \rightarrow \mathrm{E}x)$’
   d. ‘$\forall x(\mathrm{E}x \rightarrow \neg x = \mathrm{a})$’
   e. ‘$\forall x\exists y\neg\mathrm{R}xy$’
   f. ‘Ex’
   g. ‘Rxy’
   h. ‘$\exists x(\mathrm{E}x \wedge \exists y\mathrm{R}xy)$’

Truth tables

1. If φ is a wff of PC then it contains finitely many sentence letters, say n, and there are only finitely many ways of interpreting these n sentence letters (2^n ways). We can make a list of these 2^n interpretations and calculate the truth value of φ on each. That’s what we do when we construct the truth table for φ.

2. But for any wff of QC, no matter how simple, there are infinitely many ways of interpreting its symbols, because there are infinitely many ways of specifying a domain for the interpretation. Worse still, there are *uncountably* many ways, so we cannot even in principle make a list of them.

3. But for any given finite domain and any given wff of QC there are only finitely many ways of interpreting the symbols of the wff.

Example. Take the domain {1} which contains just one thing – the number 1. Take the atomic wff 'Fa'. To interpret the symbols of 'Fa' we need to assign a denotation to 'a' and a denotation to 'F'. Since there is only one member of the domain, there is only one way of assigning a denotation to 'a'. Since there is only one member of the domain, there are only 2 sets of members of the domain ({} and {1}), and so only two ways of assigning a denotation to 'F'. So altogether there are 1 × 2 = 2 ways of interpreting the symbols of 'Fa'. We can thus make the following truth table for 'Fa' relative to this domain:

Domain	|'a'|	|'F'|	|'Fa'|
{1}	1	{}	F
	1	{1}	T

4. Let's do the same thing with a domain that contains exactly two things, {1, 2}. Then there are two ways of interpreting 'a', and four ways of interpreting 'F', and thus 2 × 4 = 8 ways of interpreting 'a' and 'F':

Domain	|'a'|	|'F'|	|'Fa'|
{1, 2}	1	{}	F
	1	{1}	T
	1	{2}	F
	1	{1, 2}	T
	2	{}	F
	2	{1}	F
	2	{2}	T
	2	{1, 2}	T

5. For any given *n* there are uncountably many domains with exactly *n* members. But in every case we will end up with the same final column under 'Fa'. So 'Fa' is true (or false) on some interpretation whose domain has exactly *n* members iff it is true (or false) on some interpretation whose domain is {1, 2, …, *n*}. So too for any wff φ.

6. So if a wff φ is true (or false) on some interpretation with a finite domain then by stepping through the domains {1}, {1, 2}, {1, 2, 3}, … and constructing a truth table for φ relative to each of these domains and we will eventually find an interpretation on which it is true (or false).

7. But note that some wffs are true only on interpretations with an infinite domain, so this procedure will not find an interpretation on which they are true. Example: '$((\forall x\exists y Rxy \wedge \forall x\neg Rxx) \wedge \forall x\forall y\forall x((Rxy \wedge Ryz) \rightarrow Rxz))$'

8. <u>Exercises</u>

On the domain {1, 2, 3}, how many ways are there of interpreting the individual constant 'a'? How many ways are there of interpreting the 1-place predicate letter 'F'? So how many ways are there of interpreting the symbols in the wff 'Fa'?