Chapter 36: Functions

- *Motivation*: Ultimately, to formulate simple arithmetic as a formal theory.

Def. 1. A ***function*** $f : D \rightarrow R$ is a map from one set of objects D to another R such that one or more objects in D get mapped to a unique object in R.
- Input (one or more objects in D) is called the argument of f.
- Output (unique object in R) is called the value of f.

- *Math convention*: For function f, write "$f(x) = y$" to mean "f acting on x produces y".
  - *x is the single argument of f (element of D).*
  - *y is the value of f (element of R).*

Examples.
- *Age-in-years* function. g : {people} $\rightarrow$ {integers}
  g(Gwyneth) $= 35$
- *Product* function. h : {integers} $\rightarrow$ {integers}
  $h(5, 7) = 35$
- *Square* function. s : {integers} $\rightarrow$ {integers}
  $s(5) = 25$
- *Material conditional* function. $\supset$: {truth-values} $\rightarrow$ {truth-values}
  $\supset$(T, F) = F

Def. 2. Sense versus Extension

- The ***sense*** of a function f is the rule for correlating objects in D with an object in R that defines f.
- The ***extension*** of an n-place function f is the set of ordered $(n+1)$-tuples $\{\langle argument, value\rangle, \langle argument, value\rangle, ...\}$ that the sense of f determines.

Ex. *Square* function. $s : \{\text{integers}\} \to \{\text{integers}\}$, $s(x) = x^2$
- *Sense*: Take an integer x and multiply it by itself.
- *Extension*: $\{\langle 0, 0\rangle, \langle 1, 1\rangle, \langle 2, 4\rangle, \langle 3, 9\rangle, ...\}$

- *Note*: Two functions can have different senses but same extension.
  - $(x + 1)^2$ and $x^2 + 2x + 1$

Def. 3. Total versus Partial Functions

- A ***total*** function has a value for all arguments from its domain.
- A ***partial*** function does not have a value for all arguments from its domain.

Ex 1. *Square* function. $s : \{\text{integers}\} \to \{\text{integers}\}$, $s(x) = x^2$
- s is a total function. It has a value x^2 for any integer x in its domain.

Ex 2. *Eldest-son-of* function. $f_{esf} : \{\text{humans}\} \to \{\text{humans}\}$
- f_{esf} is a partial function. It does not have a value for all arguments in its domain (some humans don't have an eldest son).

The Language $\mathbf{QL}^f$ (an extension of $\mathbf{QL}^=$)

Alphabet of $\mathbf{QL}^f$

m, n, o, ..., c_k	individual constants $(k \geq 0)$
w, x, y, z, ..., v_k	individual variables $(k \geq 0)$
A, B, C, ..., $\mathsf{P}^0{}_k$	0-place predicates (propositional atoms) $(k \geq 0)$
F, G, H, ..., $\mathsf{P}^1{}_k$	1-place predicates $(k \geq 0)$
L, M, =, ..., $\mathsf{P}^2{}_k$	2-place predicates $(k \geq 0)$
⋮	⋮
$\mathsf{P}^n{}_k$	n-place predicates $(k \geq 0,\ n \geq 0)$
f, g, h, ..., $\mathsf{f}^n{}_k$	n-place functions $(k \geq 0,\ n > 0)$
$\wedge, \vee, \neg, \supset, \forall, \exists$, (,)	connectives, quantifiers, punctuation
$\therefore$, $*$	argument symbols

Term of $\mathbf{QL}^f$

(T1^f) An individual constant or individual variable is a term of $\mathbf{QL}^f$.

(T2^f) If $\mathsf{f}^n{}_k$ is a function symbol, and t_1, ..., t_n are terms of $\mathbf{QL}^f$, then $\mathsf{f}^n{}_k(t_1, ..., t_n)$ is a term of $\mathbf{QL}^f$.

(T3^f) Nothing else is a term.

A ***closed term*** of $\mathbf{QL}^f$ is a term of that does not contain variables.

Ex. For 1-place function f, 2-place function g, the following are terms of $\mathbf{QL}^f$.
f(n), f(y), g(m, n), g(x, f(m)), f(f(y)), g(g(m, n), f(m))

Atomic wff of $\mathbf{QL}^f$

(A1^f) If $\mathsf{P}^n{}_k$ is an n-place predicate symbol, $n \geq 0$, and t_1, ..., t_n are terms of $\mathbf{QL}^f$, then $\mathsf{P}^n{}_k t_1$, ..., t_n is an atomic *wff* of $\mathbf{QL}^f$.

(A2^f) If t_1, t_2 are terms of $\mathbf{QL}^f$, then $t_1 = t_2$ is an atomic *wff* of $\mathbf{QL}^f$.

(A2^f) Nothing else is an atomic *wff* of $\mathbf{QL}^f$.

Ex. For 2-place predicate L, the following are atomic *wffs* of $\mathbf{QL}^f$.
Lxy, Lxm, Lmn, Lmg(g(m, n), f(m))

Wff of $\mathbf{QL}^f$

(W1^f) Any atomic *wff* of $\mathbf{QL}^f$ is a *wff* of $\mathbf{QL}^f$.

(W2^f) If A is a *wff* of $\mathbf{QL}^f$, so is $\neg A$.

(W3^f) If A, B are *wffs* of $\mathbf{QL}^f$, so is $(A \wedge B)$.

(W4^f) If A, B are *wffs* of $\mathbf{QL}^f$, so is $(A \vee B)$.

(W5^f) If A, B are *wffs* of $\mathbf{QL}^f$, so is $(A \supset B)$.

(W6^f) If A is a *wff* of $\mathbf{QL}^f$ and v is a variable which occurs in A (but neither $\forall v$ nor $\exists v$ occurs in A), then $\forall vA$ is a *wff* of $\mathbf{QL}^f$.

(W7^f) If A is a *wff* of $\mathbf{QL}^f$ and v is a variable which occurs in A (but neither $\forall v$ nor $\exists v$ occurs in A), then $\exists vA$ is a *wff* of $\mathbf{QL}^f$.

(W8^f) Nothing else is a *wff* of $\mathbf{QL}^f$.

$\mathbf{QL}^f$ Semantics

- Need to modify the semantics for $\mathbf{QL}^=$ to allow for functions.

> The ***vocabulary*** V of a set of $\mathbf{QL}^f$ *wffs* is the set of constants, predicates *and functions* that appear in those *wffs*.

- *Recall*: The goal of a *q*-valuation is to set up an interpretation of a set of *wffs* with vocabulary V.
- It does this by the following:

(a) It assigns an object in a domain D to each of the *closed terms* in V.

- The closed terms are the names of objects, and now include constants c_k *and* functional expressions of the general form $\mathsf{f}^n{}_k(\mathsf{c}_1, ..., \mathsf{c}_n)$.

(b) It assigns a set of n-tuples of objects in D to each predicate $\mathsf{P}^n{}_k$ in V and a set of $(n+1)$-tuples of objects in D to each function $\mathsf{f}^n{}_k$ in V.

- A set of n-tuples of objects is the extension of an n-place predicate, and a set of $(n+1)$-tuples of objects is the extension of an n-place function.

- *Recall*: A *q*-valuation allows us to set up semantic rules that determine the truth-values, relative to the *q*-valuation, for atomic *wffs* of our language, and thus for *wffs*.

- *So*: For $\mathbf{QL}^f$, need to modify the definition of a *q*-valuation to include the parts of (a) and (b) that refer to functions.

A q-valuation on a vocabulary V of a set of wffs of $\boldsymbol{QL}^f$

(1) specifies a non-empty set of objects as the domain D;

(2) assigns to any constant c_k in V an object in D as its q-value;

(3) assigns a truth-value to any 0-place predicate $\mathsf{P}^0{}_k$ in V as its q-value;

(4) assigns to any n-place predicate $\mathsf{P}^n{}_k$ in V, $n > 0$, a set of n-tuples of objects $\{\langle m_1, ..., m_n\rangle, ...\}$ in D as its q-value;

(5) assigns to any n-place function $\mathsf{f}^n{}_k$ in V a set of $(n+1)$-tuples of objects $\{\langle \mathit{argument}, \mathit{value}\rangle, ... \}$ in D as its q-value, such that

  (i) for each n-tuple of objects $\langle m_1, ..., m_n\rangle$ in D, there is an $(n+1)$-tuple of objects $\langle m_1, ..., m_n, o\rangle$ in D;

  (ii) the set contains no distinct pairs $\langle m_1, ..., m_n, o\rangle$ and $\langle m_1, ..., m_n, o'\rangle$ where o is different from o';

(6) assigns to any term of the form $\mathsf{f}^n{}_k(\mathsf{c}_1, ..., \mathsf{c}_n)$ in V the unique object o as its q-value, such that the $(n+1)$-tuple formed by taking the q-values of $\mathsf{c}_1, ..., \mathsf{c}_n$ followed by o is an element of the q-value of $\mathsf{f}^n{}_k$.

- *Note 1*: (5i) restricts our attention to total functions.
- *Note 2*: (5ii) is required since a function must have a unique value for any given argument.
- *Note 3*: Recall that an *extended q-valuation* is a q-valuation that also assigns objects to one or more *variables*. For $\mathbf{QL}^f$, replace "variables" with "open terms".

The Semantic Rules for $\mathbf{QL}^f$

- Same as for $\mathbf{QL}^=$.

(Q0^f) If A is an atomic *wff* of $\mathbf{QL}^f$ of the form $\mathsf{P}^n{}_k t_1, \ldots, t_n$, where $\mathsf{P}^n{}_k$ is an n-place predicate and $t_1, \ldots, t_n$ are terms of $\mathbf{QL}^f$, then

(a) if $n = 0$, then $A \Rightarrow_q \mathrm{T}$ if the q-value of A is T. Otherwise $A \Rightarrow_q \mathrm{F}$.

(b) If $n > 0$, then $A \Rightarrow_q \mathrm{T}$ if the n-tuple formed by taking the q-values of the terms in A in order is an element of the q-value of A. Otherwise $A \Rightarrow_q \mathrm{F}$.

If A is an atomic *wff* of $\mathbf{QL}^=$ of the form $t_1 = t_2$, where t_1, t_2 are terms of $\mathbf{QL}^=$, then $A \Rightarrow_q \mathrm{T}$ if the q-values of the terms t_1 and t_2 are the same object. Otherwise $A \Rightarrow_q \mathrm{F}$.

(Q1^f) For any *wff* A, $\neg A \Rightarrow_q \mathrm{T}$ if $A \Rightarrow_q \mathrm{F}$; otherwise $\neg A \Rightarrow_q \mathrm{F}$.

(Q2^f) For *wffs* A, B, $(A \wedge B) \Rightarrow_q \mathrm{T}$ if both $A \Rightarrow_q \mathrm{T}$ and $B \Rightarrow_q \mathrm{T}$; otherwise $(A \wedge B) \Rightarrow_q \mathrm{F}$.

(Q3$^=$) For *wffs* A, B, $(A \vee B) \Rightarrow_q \mathrm{F}$ if both $A \Rightarrow_q \mathrm{F}$ and $B \Rightarrow_q \mathrm{F}$; otherwise $(A \vee B) \Rightarrow_q \mathrm{T}$.

(Q4^f) For *wffs* A, B, $(A \supset B) \Rightarrow_q \mathrm{F}$ if $A \Rightarrow_q \mathrm{T}$ and $B \Rightarrow_q \mathrm{F}$; otherwise $(A \supset B) \Rightarrow_q \mathrm{T}$.

(Q5^f) For *wffs* A, B, $(A \equiv B) \Rightarrow_q \mathrm{T}$ if $A \Rightarrow_q \mathrm{T}$ and $B \Rightarrow_q \mathrm{T}$, or if $A \Rightarrow_q \mathrm{F}$ and $B \Rightarrow_q \mathrm{F}$; otherwise $(A \equiv B) \Rightarrow_q \mathrm{F}$.

(Q6^f) For *wff* $C(\ldots v \ldots v \ldots)$ with variable v free, $\forall v C(\ldots v \ldots v \ldots) \Rightarrow_q \mathrm{T}$ if $C(\ldots v \ldots v \ldots) \Rightarrow_{q+} \mathrm{T}$ for every v-variant q^+ of q; otherwise $\forall v C(\ldots v \ldots v \ldots) \Rightarrow_q \mathrm{F}$.

(Q7^f) For *wff* $C(\ldots v \ldots v \ldots)$ with variable v free, $\exists v C(\ldots v \ldots v \ldots) \Rightarrow_q \mathrm{T}$ if $C(\ldots v \ldots v \ldots) \Rightarrow_{q+} \mathrm{T}$ for at least one v-variant q^+ of q; otherwise $\exists v C(\ldots v \ldots v \ldots) \Rightarrow_q \mathrm{F}$.

The Tree Rules for ***QL***^{f}

- Same as for **QL**$^=$, but need to modify $(\forall')$ in order to allow instantiation with closed functional terms

$$
\begin{array}{ll}
(\forall'') & \forall v C(...v...v...) \\
 & \quad\quad | \\
 & C(...c...c...) \qquad [c \text{ old or unprecedented}]
\end{array}
$$

Add $C(...c...c...)$ to an open path containing $\forall v C(...v...v...)$, where c is either a *closed term* on that path that hasn't already been used to instantiate $\forall v C(...v...v...)$, or c is a new *closed term* and there are no other *closed terms* appearing on that path. ***Do not check it off***.

Functions and Functional Relations

Def. An $(n+1)$-place relation $\mathcal{R}$ on a domain for which every n-tuple of objects in D stands in $\mathcal{R}$ to one and only one object is called a ***functional relation***.

- An n-place function and an $(n+1)$place functional relation can have the same extension.

Ex. Domain = {integers}

- *Successor* 1-place function. suc : {integers} → {integers}
  - *Rule*: For any integer x, take its successor. $suc(x)$ is the successor of x.
  - *Extension*: $\{\langle 0, 1\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle, \ldots\}$
- *Precedes* 2-place functional relation. "___precedes___"
  - *Extension*: $\{\langle 0, 1\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle, \ldots\}$

Def. A function f ***corresponds*** to a functional relation $\mathcal{R}$ just when they possess the same extension.

- If a function f corresponds to a functional relation $\mathcal{R}$, then $f(n)$ is the unique object x such that n stands in relation $\mathcal{R}$ to x.

Ex. *Successor* 1-place function corresponds to *precedes* 2-place functional relation.
So: $suc(n)$ is the unique integer x such that n stands in the *precedes* relation to x.

- *Note*: "The unique object x such that n stands in relation $\mathcal{R}$ to x" is a Russellian definite description!

Russell's translation scheme for definite descriptions

A sentence of the type "The F is G" is translated into **QL**$^{=}$ by the *wff*

(R) $\exists v((Fv \wedge \forall w(Fw \supset w = v)) \wedge Gv)$

- *Claim*: Any expression in **QL**^{f} with functions corresponds to a more complex expression in **QL**$^{=}$ with functions replaced by functional relations!

Translations

Domain: {integers}		
n ⇒ 0	G ⇒ ___is odd	f ⇒ *successor* function
m ⇒ 1	L ⇒ ___precedes___	

Ex1. The successor of 0 is odd.

(a) In **QL**^{f}: $\mathsf{Gf(n)}$

(b) In **QL**$^{=}$:

The successor of 0 is odd.

There's an integer such that zero precedes it, and there's only one, and that one is odd.

There's an x such that Lnx, and for all y, Lny $\supset$ y = x, and Gx.

$\exists \mathsf{x}((\mathsf{Lnx} \wedge \forall \mathsf{y}(\mathsf{Lny} \supset \mathsf{y} = \mathsf{x})) \wedge \mathsf{Gx})$

Domain: {integers}

$n \Rightarrow 0$	$G \Rightarrow$ ___is odd	$f \Rightarrow$ *successor* function
$m \Rightarrow 1$	$L \Rightarrow$ ___precedes___	

Ex2. There are integers whose successors are odd.

(a) In **QL**^{f}: $\exists z G f(z)$

(b) In **QL**$^=$:

For any integer, there's another that it precedes, and there's only one that it precedes, and that one is odd.

There's a z such that there's an x such that Lzx, and for all y, $Lzy \supset y = x$, and Gx.

$\exists z \exists x((Lzx \wedge \forall y(Lzy \supset y = x)) \wedge Gx)$

Ex3. The successor of 0 is 1.

(a) In **QL**^{f}: $f(n) = m$

(b) In **QL**$^=$:

There's an integer such that 0 precedes it, and there's only one, and that one is 1.

There's an x such that Lnx, and for all y, $Lny \supset y = x$, and $x = m$.

$\exists x((Lnx \wedge \forall y(Lny \supset y = x)) \wedge x = m)$

Domain: {integers}

n ⇒ 0	G ⇒ ___is odd	f ⇒ *successor* function
m ⇒ 1	L ⇒ ___precedes___	

Ex4. Every odd integer has 1 as its successor.

(a) In **QL**^{f}: $\forall z(Gz \supset f(z) = m)$

(b) In **QL**$^=$:

For any integer, if it's odd, then there's another that it precedes, and there's only one, and it is 1.

For all z, if Gz then there's an x such that Lzx and for all y, Lzy ⊃ y = x, and x = m.

$\forall z(Gz \supset \exists x((Lzx \wedge \forall y(Lzy \supset y = x)) \wedge x = m))$

Domain: {integers}		
n ⇒ 0	G ⇒ ___is odd	f ⇒ *successor* function
m ⇒ 1	L ⇒ ___precedes___	h ⇒ *sum* function
	S ⇒ ___+___=___	

Ex5. The sum of any two integers x, y is equal to the sum of the integers y, x. (Addition of integers is symmetric.)

(a) In **QL**^{f}: $\forall x \forall y(h(x, y) = h(y, x))$

(b) In **QL**$^=$:

For any two integers, there's another that is the sum of the first two, and there's only one such, and there's yet another integer that is the sum of the first two in reverse order, and there's only one of these, and these last two integers are the same.

For all x, y, there's a v_1 such that $Sxyv_1$, and <u>for all z, $Sxyz \supset z = v_1$</u>, and there's a v_2 such that $Syxv_2$, and <u>for all z, $Syxz \supset z = v_2$</u>, and <u>$v_2 = v_1$</u>.

$$\forall x \forall y \exists v_1\{(Sxyv_1 \wedge \forall z(Sxyz \supset z = v_1)) \wedge \exists v_2[(Syxv_2 \wedge \forall z(Syxz \supset z = v_2)) \wedge v_2 = v_1]\}$$