B1.1 Logic
Lecture 8

Martin Bays

Oxford, MT 2023

PART II: PREDICATE CALCULUS

So far:

- *Logic of the connectives* $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \ldots$ (as used in mathematics).
- Logical validity in terms of truth tables.
- Found axioms and rule of inference yielding a sound and complete proof system. Deduced compactness.

Now:

- Look *more deeply into* the structure of propositions used in mathematics.
- Analyse grammatically correct use of *functions, relations, constants, variables* and *quantifiers*.
- Define *logical validity* in this refined language.
- Isolate *axioms* and *rules of inference* (beyond those of propositional calculus) used in mathematical arguments.
- Prove: soundness, completeness, compactness.

8. The language of (first-order) predicate calculus

A **countable first-order language** $\mathcal{L}$ consists of the following disjoint sets:

- for each $k \geq 1$, a countable set of k-ary *predicate* (or *relation*) *symbols*;
- for each $k \geq 1$, a countable set of k-ary *function symbols*;
- a countable set of constant symbols.

These symbols are called the **non-logical symbols** of $\mathcal{L}$.

The alphabet of $\mathcal{L}$ consists of its non-logical symbols along with the following disjoint set of **logical symbols**:

- *Connectives:* $\rightarrow, \neg$
- *Quantifier:* $\forall$ ('for all')
- *Variables:* $x_0, x_1, x_2, \ldots$
- *3 punctuation marks:* , ( )
- *Equality symbol:* $\doteq$

8.1 Definition

(a) The **terms** of $\mathcal{L}$ are defined recursively as follows:

  (i) Every variable is a term.

  (ii) Every constant symbol is a term.

  (iii) If f is a k-ary function symbol, and $t_1, \ldots, t_k$ are terms, then so is the string

$$f(t_1, \ldots, t_k).$$

(b) An **atomic formula** of $\mathcal{L}$ is any string of the form

$$P(t_1, \ldots, t_k) \text{ or } t_1 \doteq t_2$$

where $k \geq 1$, $P \in \mathcal{L}$ is a k-ary relation symbol, and all t_i are terms.

(c) The **formulas** of $\mathcal{L}$ are defined recursively as follows:

  (i) Any atomic formula is a formula.

  (ii) If ϕ, ψ are formulas, then so are $\neg\phi$ and $(\phi \rightarrow \psi)$.

  (iii) If ϕ is a formula, then for any variable x_i so is $\forall x_i \phi$.

8.2 Examples The most general countable language has a countably infinite set of symbols of each type:

$$\mathcal{L}_{\text{pred}} := \{(P_i^{(k)})_{i,k>0}, (f_i^{(k)})_{i,k>0}, (c_i)_{i>0}\},$$

where each $P_i^{(k)}$ is a k-ary predicate symbol, each $f_i^{(k)}$ is a k-ary function symbol, and each c_i is a constant symbol.

- The following are all $\mathcal{L}_{\text{pred}}$-terms:

  $$c_3 \quad x_5 \quad f_3^{(1)}(c_2) \quad f_1^{(2)}(x_1, f_1^{(1)}(c_{37}))$$

- $f_2^{(3)}(x_1, x_2)$ is *not* a term (wrong arity).
- $P_2^{(3)}(x_4, c_2, f_3^{(2)}(c_1, x_2))$ and $f_1^{(2)}(c_5, x_2) \doteq x_3$ are atomic formulas.
- $\forall x_1 f_2^{(2)}(x_1, c_7) \doteq x_2$ and $\forall x_2 P_1^{(1)}(x_3)$ are non-atomic formulas.

8.3 Exercise
We have **unique readability** for terms, for atomic formulas, and for formulas.

A more typical example of a language appearing in mathematics is

$$\mathcal{L}_{\text{o.ring}} := \{<, \cdot, +, -, \overline{0}, \overline{1}\},$$

where $<$ is a binary relation symbol,
$\cdot$, $+$, and $-$ are binary function symbols,
and $\overline{0}$ and $\overline{1}$ are constant symbols.
We call this the *language of ordered rings*.

When dealing with binary symbols, we will allow ourselves to use infix notation as an abbreviation, so e.g.

$$\forall x_0 \; x_0 < x_0 + \overline{1}$$

abbreviates the $\mathcal{L}_{\text{o.ring}}$-formula

$$\forall x_0 <(x_0, +(x_0, \overline{1})).$$

8.4 Interpretations and logical validity
(Informal discussion)

- Consider the following $\{f\}$-formula, with f a unary function symbol:

  $$\phi_1 : \forall x_1 \forall x_2 (x_1 \doteq x_2 \to f(x_1) \doteq f(x_2)).$$

  Interpreting $\doteq$ as equality, $\forall$ as 'for all', and f as some unary function,
  ϕ_1 *should always be true*.
  We write

  $$\models \phi_1$$

  and say 'ϕ_1 is **logically valid**'.

- Consider the following $\{g\}$-formula, with g a binary function symbol:

  $$\phi_2 : \forall x_1 \forall x_2 (g(x_1, x_2) \doteq g(x_2, x_1) \to x_1 \doteq x_2)$$

  Then ϕ_2 may be true or false, depending on the situation:

- If we interpret g as $+$ on $\mathbb{N}$,
  then ϕ_2 becomes false,
  since e.g. 1+2=2+1, but $1 \neq 2$.

  So in this *interpretation*, ϕ_2 is false and $\neg\phi_2$ is true. Write

$$\langle \mathbb{N}; + \rangle \models \neg\phi_2$$

- If we interpret g as subtraction on $\mathbb{R}$,
  then ϕ_2 becomes true:
  if $x_1 - x_2 = x_2 - x_1$, then $2x_1 = 2x_2$, and hence $x_1 = x_2$.
  So

$$\langle \mathbb{R}; - \rangle \models \phi_2$$

8.5 Free and bound variables
(Informal discussion)

There is a further complication: Consider the $\{P\}$-formula

$$\phi_3 : \forall x_0 P(x_1, x_0).$$

Specifying the interpretation is not enough to determine whether or not ϕ_3 holds.

For example, in $\langle \mathbb{N}; \leq \rangle$:
- If we put $x_1 = 0$ then ϕ_3 is true;
- if we put $x_1 = 2$ then ϕ_3 is false.

So it depends on the value we assign to x_1 (like in propositional calculus: the truth value of $(p_0 \wedge p_1)$ depends on the valuation).

In ϕ_3 we *can* assign a value to x_1 because x_1 occurs **free** in ϕ_3.

For x_0, however, it makes no sense to assign a particular value; because x_0 is **bound** in ϕ_3 by the quantifier $\forall x_0$.