Recap

- Last time: First-Order Logic
- This time:
  - Representing mathematical concepts
  - Introduction to Higher-Order Logic
Representing Knowledge

So far, we have:

▶ Seen the primitive rules of (first-order) logic
▶ Reasoned about abstract Ps, Qs, and Rs

But we usually want to reason in some mathematical theory. For example: number theory, real analysis, automata theory, euclidean geometry, ...

How do we represent this theory so we can prove theorems about it?

▶ Which logic do we use? — Propositional, FOL, Temporal, Hoare Logic, HOL?
▶ Do we axiomatise our theory, or define it in terms of more primitive concepts?
▶ What style do we use? e.g. functions vs. relations
Further Issues

What are the important theorems in our theory?

- Which formalisation is most useful?
- Is it easy to understand?
- Is it natural?
- How easy is it to reason with?

Often a matter of taste, or experience, or tradition, or efficiency of implementation, or following the idioms of the people you are working with. No one right way!

Granularity of the representation

- What primitives do we need? Consider geometry:
  - Define lines in terms of points? (Tarski)
  - Or take points and lines as primitive? (Hilbert)

- Or computing; should we treat programs as:
  - State transition systems? (operational)
  - Functions mapping inputs to outputs? (∼ denotational)
Axioms vs. Definitions

Let’s say we want to reason using the natural numbers \{0, 1, 2, 3, \ldots\}

Axiomatise? Assume a collection of function symbols and unproven axioms. For instance, the Peano axioms:

\[
\begin{align*}
\forall x. \neg(0 = S(x)) \\
\forall x. x + 0 &= x \\
\forall x. x + S(y) &= S(x + y)
\end{align*}
\]

…”

Define? If our logic has sets as a primitive (or are definable), then we can define the natural numbers via the von Neumann ordinals:


0 = \emptyset, 1 = \{\emptyset\}, 2 = \{\emptyset, \{\emptyset\}\}, \ldots

Then we can prove the Peano axioms for this definition.
Axioms vs. Definitions

Axiomatisation:

▶ (+) Sometimes less work – finding a good definition, and (formally) working with it can be hard.
▶ (-) How do we know that our axiomatisation is adequate for our purposes, or is complete?
▶ (-) How do we know that our axiomatisation is consistent? Can we prove $\bot$ from our axioms (and hence everything)?

Definition:

▶ (-) Can be a lot of work, sometimes needing some ingenuity.
▶ (+++) If the underlying logic is consistent, then we are guaranteed to be consistent (c.f., “Why should you believe Isabelle” from Lecture 4). We have relative consistency.
Axiomatisation, an example: Set Theory

Let’s take the FOL + a binary atomic predicate $\in$ + the following axiom for every formula $P$ with one free variable $x$:

$$\exists y. \forall x. x \in y \leftrightarrow P(x)$$

“For every predicate $P$ there is a set $y$ such that its members are exactly those that satisfy $P$”

We can now define empty set, pairing, union, intersection...
Axiomatisation, an example: Set Theory

Let’s take the FOL + a binary atomic predicate $\in$ + the following axiom for every formula $P$ with one free variable $x$:

$$\exists y. \forall x. x \in y \leftrightarrow P(x)$$

“For every predicate $P$ there is a set $y$ such that its members are exactly those that satisfy $P$”

We can now define empty set, pairing, union, intersection...

But it is too powerful! Let $P(x) \equiv \neg(x \in x)$. Then by the axiom there is a $y$ such that:

$$y \in y \leftrightarrow y \notin y$$

This is Russell’s paradox.
Axiomatisation, an example: Set Theory

Let’s take the FOL + a binary atomic predicate ∈ + the following axiom for every formula $P$ with one free variable $x$:

$$\exists y. \forall x. x \in y \leftrightarrow P(x)$$

“For every predicate $P$ there is a set $y$ such that its members are exactly those that satisfy $P$”

We can now define empty set, pairing, union, intersection...

But it is too powerful! Let $P(x) \equiv \neg(x \in x)$. Then by the axiom there is a $y$ such that:

$$y \in y \leftrightarrow y \notin y$$

This is Russell’s paradox.

Background: the axiom is called “unrestricted comprehension”, it was replaced by:

$$\forall z. \exists y. \forall x. (x \in y \leftrightarrow (x \in z \land P(x)))$$

+ some other axioms to give ZF set theory.
Building up Definitions: Integers

Starting from the natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$, we can define:

► each integer $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ as an **equivalence class** of pairs of natural numbers under the relation $
(a, b) \sim (c, d) \iff a + d = b + c$;

► For example, $-2$ is represented by the equivalence class $[(0, 2)] = [(1, 3)] = [(100, 102)] = \ldots$.

► we define the sum and product of two integers as

\[
[(a, b)] + [(c, d)] = [(a + c, b + d)] \\
[(a, b)] \times [(c, d)] = [(ac + bd, ad + bc)]
\]

► we define the set of **negative** integers as the set

\[
\{[(a, b)] \mid b > a\}.
\]

► Exercise: show that the product of negative integers is non-negative.
Other Representation Examples

- The rationals $\mathbb{Q}$ can be defined as pairs of integers. Reasoning about the rationals therefore reduces to reasoning about the integers.
- The reals $\mathbb{R}$ can be defined as sets of rationals. Reasoning about the reals therefore reduces to reasoning about the rationals.
- The complex numbers $\mathbb{C}$ can be defined as pairs of reals. Reasoning about the complex numbers therefore reduces to reasoning about the reals.
- In this way, we have relative consistency.
  - If the theory of natural numbers is consistent, so is the theory of complex numbers.
Functions or Predicates?

We can represent some property $r$ holding between two objects $x$ and $y$ as:

- a function with equality $r(x) = y$
- a predicate $r(x, y)$

Is it better to use functions or predicates to represent properties?

It is not always clear which is best!
For example, suppose we represent division of real numbers (/) by a function $\text{div} : \text{real} \times \text{real} \Rightarrow \text{real}$.

- We define $\text{div}(x, y)$ when $y \neq 0$ in normal way
- What about division-by-zero? What is the value of $\text{div}(x, 0)$?
- In first-order logic, functions are assumed to be total, so we have to pick a value!
- We could choose a convenient element: say 0. That way:

$$0 \leq x \rightarrow 0 \leq 1/x.$$
Q) Can we represent division of real numbers (/) by a relation $\text{Div}: \text{real} \times \text{real} \times \text{real} \Rightarrow \text{bool}$ such that $\text{Div}(x, y, z)$ is

- $x/y = z$ when $y \neq 0$, and
- $\perp$ when $y = 0$?

A) Yes: $\text{Div}(x, y, z) \equiv x = y \times z \land \forall w. x = y \times w \rightarrow z = w$

That is, $z$ is that unique value such that $x = y \times z$.

But now formulas are more complicated.

$$x, y \neq 0 \rightarrow \frac{1}{(x/y)/x} = y$$

becomes

$$\text{Div}(x, y, u) \land \text{Div}(u, x, v) \land \text{Div}(1, v, w) \land x, y \neq 0 \rightarrow w = y$$
Can we represent the concept of square roots with a function $\sqrt{} : \text{real} \Rightarrow \text{real}$?

- All positive real numbers have two square roots, and yet a function maps points to single values.

- We can pick one of the values arbitrarily: say, the positive (principal) square root.

- Or we can have the function map every real to a set $\sqrt{} : \text{real} \Rightarrow \text{real set}$:

  $$\sqrt{x} \equiv \{ y \mid x = y^2 \}. $$

- But now we have two kinds of object: reals and sets of reals, and we cannot conveniently express:

  $$(\sqrt{x})^2 = x$$

- Our representation of reals is no longer self-contained.
Q) Can we represent the concept of square roots with a relation $Sqrt : real \times real \Rightarrow bool$?

A) Yes. E.g. $Sqrt(x, y) \equiv x = y^2$.

Again drawback of formulas being more complicated
Functions, Predicates and Sets

We can translate back and forth. But too much translation makes a formalisation hard to use!

Any function $f: \alpha \rightarrow \beta$ can be represented as a relation $R: \alpha \times \beta \rightarrow bool$ or a set $S: (\alpha \times \beta) set$ by defining:

$$R(x, y) \equiv f(x) = y$$
$$S \equiv \{(x, y) \mid f(x) = y\}.$$

Any predicate $P$ can be represented by a function $f$ or a set $S$ by defining:

$$f(x) \equiv \begin{cases} True & : P(x) \\ False & : \text{otherwise} \end{cases}$$
$$S \equiv \{x \mid P(x)\}.$$

Any set $S$ can be represented by a function $f$ or a predicate $P$ by defining:

$$f(x) \equiv \begin{cases} True & : x \in S \\ False & : \text{otherwise} \end{cases}$$
$$P(x) \equiv x \in S$$
In pure (without axioms) FOL, we cannot directly represent the statement:

there is a function that is larger on all arguments than the log function.

To formalise it, we would need to quantify over functions:

\[ \exists f. \forall x. f(x) > \log x. \]

Likewise we cannot quantify over predicates.

**Solutions in FOL:**

- Represent all functions and predicates by *sets*, and quantify over these. This is the approach of first-order set theories such as *ZF*.

- Introduce sorts for predicates and functions. Not so elegant now having 2 kinds of each.
Higher-Order Logic (HOL)

Alternatively...
In HOL, we represent sets and predicates by functions, often denoted by lambda abstractions.

**Definition (Lambda Abstraction)**
Lambda abstractions are terms which denote functions directly by the rules which define them, e.g. the square function is $\lambda x. x \cdot x$.

This is a way of defining a function without giving it a name:

\[
f(x) \equiv x \cdot x \quad \text{vs} \quad f \equiv \lambda x. x \cdot x
\]

We can use lambda abstractions exactly as we use ordinary function symbols. E.g. $(\lambda x. x \cdot x) \ 3 = 9$. 

Using $\lambda$-notation, we can think about functions as individual objects. E.g., we can define functions which map from and to other functions.

**Example**
The $K$-combinator maps some $x$ to a function which sends any $y$ to $x$.

$$\lambda x. \lambda y. x.$$

**Example**
The composition function maps two functions to their composition:

$$\lambda f. \lambda g. \lambda x. f(g \, x).$$
Types \textit{bool}, \textit{ind} (individuals) and $\alpha \Rightarrow \beta$ primitive. All others defined from these.

Two primitive (families of) functions:

\begin{align*}
\text{equality} & : \alpha \Rightarrow \alpha \Rightarrow \text{bool} \\
\text{implication} & : \text{bool} \Rightarrow \text{bool} \Rightarrow \text{bool}
\end{align*}

All other functions defined using this, lambda abstraction and application.

Distinction between formulas and terms is dispensed with: formulas are just terms of type \textit{bool}.

Predicates are represented by functions $\alpha \Rightarrow \text{bool}$. Sets are represented as predicates.
True is defined as:
\[ \top \equiv (\lambda x. x) = (\lambda x. x) \]

Universal quantification as function equality:
\[ \forall x. \phi \equiv (\lambda x. \phi) = (\lambda x. \top) \]

This works for \( x \) of any type: \( \text{bool}, \text{ind} \Rightarrow \text{bool}, \ldots \)

Therefore, we can quantify over functions, predicates and sets.

Conjunction and disjunction are defined:
\[
\begin{align*}
P \land Q & \equiv \forall R. (P \rightarrow Q \rightarrow R) \rightarrow R \\
P \lor Q & \equiv \forall R. (P \rightarrow R) \rightarrow (Q \rightarrow R) \rightarrow R
\end{align*}
\]

Define natural numbers (\( \mathbb{N} \)), integers (\( \mathbb{Z} \)), rationals (\( \mathbb{Q} \)), reals (\( \mathbb{R} \)), complex numbers (\( \mathbb{C} \)), vector spaces, manifolds, …
Higher-Order Logic is the underlying logic of Isabelle/HOL, the theorem prover we are using.

(below is for interest only!)

The axiomatisation is slightly different to the one described on the previous slides, and a bit more powerful.

If you are really keen, look at the chapter “Higher-Order Logic” in the “logics” document in the Isabelle documentation.

Or the file Isabelle2013-2/src/HOL/HOL.thy in the Isabelle installation.

Exercise (only if you are interested!): why can’t Russell’s paradox happen in HOL?
Summary

- This time:
  - Issues involved in representing mathematical theories
  - Axioms vs. Definitions
  - Functions vs. Predicates
  - Introduction to Higher-Order Logic
  - Reading: Bundy, Chapter 4 (contains further discussion of issues in representation, e.g. variadic functions).

- On the course web-page: some more exercises, asking you to prove False from the axioms of Naive Set Theory.

- Next time:
  - Coursework 1: Proving in Isabelle/HOL.