Content

→ Intro & motivation, getting started

→ Foundations & Principles
  • Lambda Calculus, natural deduction [1,2]
  • Higher Order Logic [3\textsuperscript{a}]
  • Term rewriting [4]

→ Proof & Specification Techniques
  • Inductively defined sets, rule induction [5]
  • Datatypes, recursion, induction [6, 7]
  • Hoare logic, proofs about programs, C verification [8\textsuperscript{b}, 9]
  • (mid-semester break)
  • Writing Automated Proof Methods [10]
  • Isar, codegen, typeclasses, locales [11\textsuperscript{c}, 12]

\textsuperscript{a}a1 due; \textsuperscript{b}a2 due; \textsuperscript{c}a3 due
Last Time

- Sets
- Type Definitions
- Inductive Definitions
Inductive Definitions

How They Work
The Nat Example

\[
\begin{align*}
0 & \in N \\
n & \in N \\
n + 1 & \in N
\end{align*}
\]

→ \( N \) is the set of natural numbers \( \mathbb{N} \)
→ But why not the set of real numbers? \( 0 \in \mathbb{R}, \ n \in \mathbb{R} \implies n + 1 \in \mathbb{R} \)
→ \( \mathbb{N} \) is the \textit{smallest} set that is \textit{consistent} with the rules.

Why the smallest set?
→ Objective: \textit{no junk}. Only what must be in \( X \) shall be in \( X \).
→ Gives rise to a nice proof principle (rule induction)
Formally:

Rules $a_1 \in X \ldots a_n \in X$ with $a_1, \ldots, a_n, a \in A$

define set $X \subseteq A$

Formally: set of rules $R \subseteq A$ set $\times A$  ($R$, $X$ possibly infinite)

Applying rules $R$ to a set $B$:

$\hat{R} B \equiv \{x. \exists H. (H, x) \in R \land H \subseteq B\}$

Example:

$R \equiv \{({\{\}}, 0)\} \cup \{({\{n\}}, n + 1). \ n \in \mathbb{N}\}$

$\hat{R} \{3, 6, 10\} = \{0, 4, 7, 11\}$
The Set

**Definition:** \( B \) is \( R \)-closed iff \( \hat{R} B \subseteq B \)

**Definition:** \( X \) is the least \( R \)-closed subset of \( A \)

This does always exist:

**Fact:** \[ X = \bigcap \{ B \subseteq A. B \text{ \( R \)-closed} \} \]
Generation from Above

\[ A \]
Rule Induction

\[
\begin{align*}
0 & \in N \\
n & \in N \\
n + 1 & \in N
\end{align*}
\]

induces induction principle

\[
[P \; 0; \; \land \; n. \; P \; n \implies P \; (n + 1)] \implies \forall x \in X. \; P \; x
\]

In general:

\[
\forall(\{a_1, \ldots a_n\}, \; a) \in R. \; P \; a_1 \land \ldots \land P \; a_n \implies P \; a \\
\forall x \in X. \; P \; x
\]
Why does this work?

\[ \forall\left(\{a_1, \ldots, a_n\}, a\right) \in R. \ P a_1 \land \ldots \land P a_n \implies P a \]
\[ \forall x \in X. \ P x \]

\[ \forall\left(\{a_1, \ldots, a_n\}, a\right) \in R. \ P a_1 \land \ldots \land P a_n \implies P a \]

says
\[ \{x. \ P x\} \text{ is } R\text{-closed} \]

but:
\[ X \text{ is the least } R\text{-closed set} \]
hence:
\[ X \subseteq \{x. \ P x\} \]
which means:
\[ \forall x \in X. \ P x \]

qed
Rules with side conditions

\[
\begin{align*}
& a_1 \in X \quad \ldots \quad a_n \in X \quad C_1 \quad \ldots \quad C_m \\
\hline
& a \in X
\end{align*}
\]

induction scheme:

\[
(\forall (\{a_1, \ldots a_n\}, a) \in R. ~ P ~ a_1 \land \ldots \land P ~ a_n \land C_1 \land \ldots \land C_m \land \{a_1, \ldots, a_n\} \subseteq X \implies P ~ a) \\
\implies \\
\forall x \in X. ~ P ~ x
\]
X as Fixpoint

How to compute X?

\( X = \bigcap \{ B \subseteq A. B \ R - \text{closed} \} \) hard to work with.

**Instead:** view \( X \) as least fixpoint, \( X \) least set with \( \hat{R} \ X = X \).

Fixpoints can be approximated by iteration:

\[
X_0 = \hat{R}^0 \ \{\} = \{\} \\
X_1 = \hat{R}^1 \ \{\} = \text{rules without hypotheses} \\
\vdots \\
X_n = \hat{R}^n \ \{\} \\
X_\omega = \bigcup_{n \in \mathbb{N}} (R^n \ \{\}) = X
\]
Generation from Below

\[
A = \hat{R}^0 \{\} \cup \hat{R}^1 \{\} \cup \hat{R}^2 \{\} \cup \ldots
\]
Does this always work?

**Knaster-Tarski Fixpoint Theorem:**
Let \((A, \leq)\) be a complete lattice, and \(f :: A \Rightarrow A\) a monotone function.
Then the fixpoints of \(f\) again form a complete lattice.

**Lattice:**
Finite subsets have a greatest lower bound (meet) and least upper bound (join).

**Complete Lattice:**
*All* subsets have a greatest lower bound and least upper bound.

**Implications:**
- least and greatest fixpoints exist (complete lattice always non-empty).
- can be reached by (possibly infinite) iteration. (Why?)
Exercise

Formalize this lecture in Isabelle:

1. Define \texttt{closed} \( f \ A :: (\alpha \text{ set} \Rightarrow \alpha \text{ set}) \Rightarrow \alpha \text{ set} \Rightarrow \text{ bool} \)
2. Show \( \text{closed} \ f \ A \land \text{closed} \ f \ B \implies \text{closed} \ f \ (A \cap B) \) if \( f \) is monotone (\texttt{mono} is predefined)
3. Define \texttt{lfpt} \( f \) as the intersection of all \( f \)-closed sets
4. Show that \( \text{lfpt} \ f \) is a fixpoint of \( f \) if \( f \) is monotone
5. Show that \( \text{lfpt} \ f \) is the least fixpoint of \( f \)
6. Declare a constant \( R :: (\alpha \text{ set} \times \alpha) \text{ set} \)
7. Define \( \hat{R} :: \alpha \text{ set} \Rightarrow \alpha \text{ set} \) in terms of \( R \)
8. Show soundness of rule induction using \( R \) and \( \text{lfpt} \ \hat{R} \)
We have learned today ...

- Formal background of inductive definitions
- Definition by intersection
- Computation by iteration
- Formalisation in Isabelle