Content

→ Foundations & Principles
  • Intro, Lambda calculus, natural deduction [1,2]
  • Higher Order Logic, Isar (part 1) [2,3\textsuperscript{a}]
  • Term rewriting [3,4]

→ Proof & Specification Techniques
  • Inductively defined sets, rule induction, datatype induction, primitive recursion [4,5]
  • General recursive functions, termination proofs [7\textsuperscript{b}]
  • Proof automation, Hoare logic, proofs about programs, invariants [8]
  • C verification [9,10]
  • Practice, questions, examp prep [10\textsuperscript{c}]

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

→ Sets
Last Time

→ Sets
→ Type Definitions
Last Time

- Sets
- Type Definitions
- Inductive Definitions
Inductive Definitions

How They Work
The Nat Example

\[ 0 \in N \quad \frac{n \in N}{n + 1 \in N} \]
The Nat Example

\[ 0 \in \mathbb{N} \quad \frac{n \in \mathbb{N}}{n + 1 \in \mathbb{N}} \]

⇒ \( \mathbb{N} \) is the set of natural numbers \( \mathbb{N} \)
The Nat Example

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

\[ \Rightarrow \quad N \text{ is the set of natural numbers } \mathbb{N} \]

\[ \Rightarrow \quad \text{But why not the set of real numbers? } 0 \in \mathbb{R}, \ n \in \mathbb{R} \implies n + 1 \in \mathbb{R} \]
The Nat Example

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

$\Rightarrow$ \( \mathbb{N} \) is the set of natural numbers \( \mathbb{N} \)

$\Rightarrow$ But why not the set of real numbers? \( 0 \in \mathbb{R}, \ n \in \mathbb{R} \implies n + 1 \in \mathbb{R} \)

$\Rightarrow$ \( \mathbb{N} \) is the **smallest** set that is **consistent** with the rules.
The Nat Example

$0 \in \mathbb{N}$  \hspace{1cm} $n \in \mathbb{N} \Rightarrow n + 1 \in \mathbb{N}$

$\Rightarrow$ \hspace{0.2cm} $\mathbb{N}$ is the set of natural numbers $\mathbb{N}$

$\Rightarrow$ \hspace{0.2cm} But why not the set of real numbers? $0 \in \mathbb{R}$, $n \in \mathbb{R} \Rightarrow n + 1 \in \mathbb{R}$

$\Rightarrow$ \hspace{0.2cm} $\mathbb{N}$ is the **smallest** set that is **consistent** with the rules.

**Why the smallest set?**
The Nat Example

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

→ $\mathbb{N}$ 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 smallest set that is consistent with the rules.

Why the smallest set?
→ Objective: no junk. Only what must be in $X$ shall be in $X$. 
The Nat Example

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

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

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

\[
\begin{align*}
\text{Rules } & \quad a_1 \in X \quad \ldots \quad a_n \in X \\
& \quad a \in X \quad \text{with } a_1, \ldots, a_n, a \in A \\
\end{align*}
\]

define set \( X \subseteq A \)

Formally:
Formally

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

define set $$X \subseteq A$$

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

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

Rules \( \frac{a_1 \in X \ldots a_n \in X}{a \in X} \) with \( a_1, \ldots, a_n, a \in A \)

define set \( X \subseteq A \)

**Formally:** set of rules \( R \subseteq A \) set \( \times \) \( A \) \( (R, X \text{ possibly infinite}) \)

**Applying rules** \( R \) to a set \( B \): \( \hat{R} B \equiv \{ x. \ \exists H. (H, x) \in R \land H \subseteq B \} \)

**Example:**
Formally

Rules \( \frac{a_1 \in X \ldots a_n \in X}{a \in X} \) with \( a_1, \ldots, a_n, a \in A \)

define set \( X \subseteq A \)

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

Applying rules \( R \) to a set \( B \): \( \hat{R} B \equiv \{ x \mid \exists H. (H, x) \in R \land H \subseteq B \} \)

Example:

\[
R \quad \equiv \quad \{(\{\}, 0)\} \cup \{\{n\}, n + 1 \mid n \in \mathbb{N}\}
\]

\[
\hat{R} \{3, 6, 10\} \quad = \quad \{0, 4, 7, 11\}
\]
Formally

Rules \( a_1 \in X \ldots a_n \in X \) with \( a_1, \ldots, a_n, a \in A \)

\( a \in X \)

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 $\hat{R} B \subseteq B$
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:
**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:** $X = \bigcap\{B \subseteq A. \ B \ R-\text{closed}\}$
Generation from Above
Generation from Above

$A$

$R$-closed
Generation from Above

\[ A \]

\[ R\text{-closed} \]
Generation from Above
Generation from Above
Rule Induction

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

induces induction principle

\[
[ \[ P 0; \land n. P n \Rightarrow P (n + 1) \] ] \Rightarrow \forall x \in N. P x
\]
Rule Induction

\[\begin{align*}
0 \in N & \quad n \in N \\
\frac{n + 1 \in N}{\text{induces induction principle}}
\end{align*}\]

\[\begin{align*}
\text{induces induction principle} & \quad \left[ P \ 0; \ \bigwedge n. \ P \ n \implies P \ (n + 1) \right] \implies \forall x \in N. \ P \ x
\end{align*}\]

In general:

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

\[\forall x \in X. \ P \ x\]
Why does this work?

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

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

says
Why does this work?

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

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

says

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

but:
Why does this work?

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

\[ \forall x \in X. \ P x \]

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

says

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

but:

\[ X \text{ is the least } R\text{-closed set} \]

hence:
Why does this work?

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

\[ \forall x \in X. \quad P \ x \]

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

says

\{x. \ P \ x\} is R-closed

but: \quad X is the least R-closed set

hence: \quad X \subseteq \{x. \ P \ x\}

which means:
Why does this work?

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

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

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:
\[ \forall x \in X. \ P x \]
Why does this work?

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

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

\[
\frac{a_1 \in X \quad \ldots \quad a_n \in X \quad C_1 \quad \ldots \quad C_m}{a \in X}
\]
Rules with side conditions

\[
\begin{array}{cccccc}
  a_1 \in X & \ldots & a_n \in X & C_1 & \ldots & C_m \\
  \hline
  a \in X
\end{array}
\]

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 $X$?

$X = \bigcap \{ B \subseteq A. B \text{ R} \text{-closed} \}$

hard to work with.

Instead:

view $X$ as least fixpoint, $X$ least set with $\hat{R} X = X$.

Fixpoints can be approximated by iteration:

$X_0 = \hat{R}_0 \{ \}$

$X_1 = \hat{R}_1 \{ \}$

rules without hypotheses...

$X_n = \hat{R}_n \{ \}$

$X_\omega = \bigcup n \in \mathbb{N} (\hat{R}_n \{ \}) = X$.
$X$ as Fixpoint

How to compute $X$?

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

Instead:
$X$ 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 $\hat{R} X = X$. 
**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 as Fixpoint**

**How to compute X?**

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

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

**Fixpoints can be approximated by iteration:**

\[
\begin{align*}
X_0 &= \hat{R}^0 \{} = {} \\
X_1 &= \hat{R}^1 \{} = \text{rules without hypotheses} \\
\vdots
\end{align*}
\]
**X as Fixpoint**

How to compute \( X \)?
\[ X = \bigcap \{ B \subseteq A. B \text{ } R \text{ } \text{closed} \} \text{ hard to work with.} \]

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

**Fixpoints can be approximated by iteration:**

\[
X_0 = \hat{R}^0 \{ \} = \{ \}
\]

\[
X_1 = \hat{R}^1 \{ \} = \text{rules without hypotheses}
\]

\[
\vdots
\]

\[
X_n = \hat{R}^n \{ \}
\]
\(X\) as Fixpoint

How to compute \(X\)?
\(X = \bigcap \{ B \subseteq A. B \ R \textendash closed\}\) hard to work with.

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

Fixpoints can be approximated by iteration:
\[
\begin{align*}
X_0 &= \hat{R}^0 \{\} = \{\} \\
X_1 &= \hat{R}^1 \{\} = \text{rules without hypotheses} \\
&\vdots \\
X_n &= \hat{R}^n \{\} \\
X_\omega &= \bigcup_{n \in \mathbb{N}} (\hat{R}^n \{\}) = X
\end{align*}
\]
Generation from Below

\[ A \hat{R}^0 \{ \} \]
Generation from Below

\[ A \]

\[ \hat{R}^0 \{ \} \cup \hat{R}^1 \{ \} \]
Generation from Below

\[ A = \hat{R}^0 \{ \} \cup \hat{R}^1 \{ \} \cup \hat{R}^2 \{ \} \]
Generation from Below

\[ \hat{R}^0 \{ \} \cup \hat{R}^1 \{ \} \cup \hat{R}^2 \{ \} \cup \ldots \]
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.
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).
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.
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:**

\(\rightarrow\) least and greatest fixpoints exist (complete lattice always non-empty).
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:

- Define `closed f A :: (α set ⇒ α set) ⇒ α set ⇒ bool`
- Show `closed f A ∧ closed f B ⇒ closed f (A \cap B)` if `f` is monotone (`mono` is predefined)
- Define `lfpt f` as the intersection of all `f`-closed sets
- Show that `lfpt f` is a fixpoint of `f` if `f` is monotone
- Show that `lfpt f` is the least fixpoint of `f`
- Declare a constant `R :: (α set × α) set`
- Define `\hat{R} :: α set ⇒ α set` in terms of `R`
- Show soundness of rule induction using `R` and `lfpt \hat{R}`
We have learned today ...

⇒ Formal background of inductive definitions
We have learned today ...

- Formal background of inductive definitions
- Definition by intersection
We have learned today ...

- Formal background of inductive definitions
- Definition by intersection
- Computation by iteration
We have learned today ...

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