COMP 4161
NICTA Advanced Course

Advanced Topics in Software Verification

Gerwin Klein, June Andronick, Toby Murray, Rafal Kolanski

$\{\}$

Slide 1

Content

➜ Intro & motivation, getting started [1]

➜ Foundations & Principles

- Lambda Calculus, natural deduction [1,2]
- Higher Order Logic [3^a]
- Term rewriting [4]

➜ Proof & Specification Techniques

- Isar [5]
- Inductively defined sets, rule induction [6^b]
- Datatypes, recursion, induction [7^c, 8]
- Calculational reasoning, code generation [9]
- Hoare logic, proofs about programs [10^d,11,12]

^{a}a1 due; ^{b}a2 due; ^{c}session break; ^{d}a3 due

Slide 2

Last Time

➜ Sets
➜ Type Definitions
➜ Inductive Definitions

Slide 3

How Inductive Definitions Work

Slide 4

The Nat Example

$$\frac{}{0 \in N} \qquad \frac{n \in N}{n+1 \in N}$$

- N is the set of natural numbers $\mathbb{N}$
- But why not the set of real numbers? $0 \in \mathbb{R}$, $n \in \mathbb{R} \Longrightarrow n+1 \in \mathbb{R}$
- $\mathbb{N}$ 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)

Slide 5

Formally

Rules $\dfrac{a_1 \in X \quad \dots \quad a_n \in X}{a \in X}$ with $a_1, \dots, a_n, a \in A$

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

Example:

$$\begin{array}{lcl} R & \equiv & \{(\{\}, 0)\} \cup \{(\{n\}, n+1).\ n \in \mathbb{R}\} \\ \hat{R}\ \{3, 6, 10\} & = & \{0, 4, 7, 11\} \end{array}$$

Slide 6

The Set

Definition: B is R-closed iff $\hat{R}\ B \subseteq B$

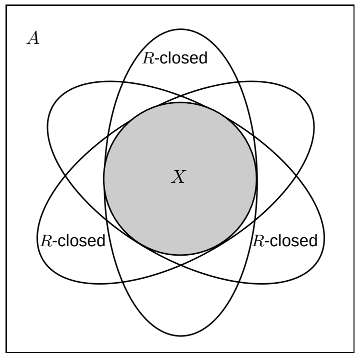
Definition: X is the least R-closed subset of A

This does always exist:

Fact: $X = \bigcap\{B \subseteq A.\ B\ R\text{–closed}\}$

Slide 7

Generation from Above



Slide 8

Rule Induction

NICTA

$$\frac{}{0 \in N} \qquad \frac{n \in N}{n+1 \in N}$$

induces induction principle

$$[\![P\ 0;\ \bigwedge n.\ P\ n \Longrightarrow P\ (n+1)]\!] \Longrightarrow \forall x \in X.\ P\ x$$

In general:

$$\frac{\forall (\{a_1, \ldots a_n\}, a) \in R.\ P\ a_1 \wedge \ldots \wedge P\ a_n \Longrightarrow P\ a}{\forall x \in X.\ P\ x}$$

Slide 9

Why does this work?

NICTA

$$\frac{\forall (\{a_1, \ldots a_n\}, a) \in R.\ P\ a_1 \wedge \ldots \wedge P\ a_n \Longrightarrow P\ a}{\forall x \in X.\ P\ x}$$

$$\forall (\{a_1, \ldots a_n\}, a) \in R.\ P\ a_1 \wedge \ldots \wedge P\ a_n \Longrightarrow P\ a$$

says

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

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

qed

Slide 10

Rules with side conditions

NICTA

$$\frac{a_1 \in X \quad \ldots \quad a_n \in X \qquad C_1 \quad \ldots \quad C_m}{a \in X}$$

induction scheme:

$$\begin{aligned}(\forall (\{a_1, \ldots a_n\}, a) \in R.\ & P\ a_1 \wedge \ldots \wedge P\ a_n \wedge \\ & C_1 \wedge \ldots \wedge C_m \wedge \\ & \{a_1, \ldots, a_n\} \subseteq X \Longrightarrow P\ a)\end{aligned}$$

$$\Longrightarrow$$

$$\forall x \in X.\ P\ x$$

Slide 11

X as Fixpoint

NICTA

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:

$X_0 = \hat{R}^0\ \{\} = \{\}$

$X_1 = \hat{R}^1\ \{\} =$ rules without hypotheses

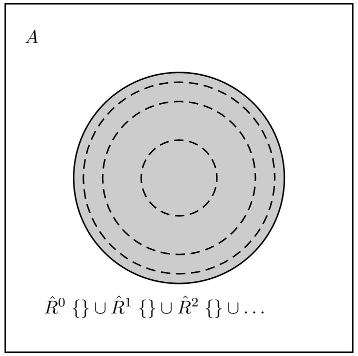
$\vdots$

$X_n = \hat{R}^n\ \{\}$

$X_\omega = \bigcup_{n \in \mathbb{N}} (R^n\ \{\}) = X$

Slide 12

Generation from Below



Slide 13

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?)

Slide 14

Exercise

Formalize the this lecture in Isabelle:

- ➜ Define **closed** $f\ A :: (\alpha$ set $\Rightarrow \alpha$ set$) \Rightarrow \alpha$ set $\Rightarrow$ bool
- ➜ Show closed $f\ A \wedge$ closed $f\ B \Longrightarrow$ 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 $R :: (\alpha$ set $\times\ \alpha)$ set
- ➜ Define $\hat{R} :: \alpha$ set $\Rightarrow \alpha$ set in terms of R
- ➜ Show soundness of rule induction using R and lfpt $\hat{R}$

Slide 15

Rule Induction in Isar

Slide 16

Inductive definition in Isabelle

inductive $X :: \alpha \Rightarrow$ bool
where
rule$_1$: "$[\![X\ s; A]\!] \Longrightarrow X\ s'$"
$\vdots$
| rule$_n$: $\ldots$

Slide 17

Rule induction

show "$X\ x \Longrightarrow P\ x$"
proof (induct rule: X.induct)
  fix s and s' **assume** "$X\ s$" and "A" and "$P\ s$"
  $\ldots$
  show "$P\ s'$"
next
$\vdots$
qed

Slide 18

Abbreviations

show "$X\ x \Longrightarrow P\ x$"
proof (induct rule: X.induct)
  case rule$_1$
  $\ldots$
  show ?case
next
$\vdots$
next
  case rule$_n$
  $\ldots$
  show ?case
qed

Slide 19

Implicit selection of induction rule

assume A: "$X\ x$"
$\vdots$
show "$P\ x$"
using A **proof** induct
$\vdots$
qed

lemma assumes A: "$X\ x$" **shows** "$P\ x$"
using A **proof** induct
$\vdots$
qed

Slide 20

Renaming free variables in rule

case (rule$_i$ $x_1 \ldots x_k$)

Renames first k variables in rule$_i$ to $x_1 \ldots x_k$.

Slide 21

A remark on style

➜ **case** (rule$_i$ x y) ... **show** ?case
is easy to write and maintain

➜ **fix** x y **assume** $formula$ ... **show** $formula'$
is easier to read:

- all information is shown locally
- no contextual references (e.g. ?case)

Slide 22

Demo: Rule Induction in Isar

Slide 23

We have learned today ...

➜ Formal background of inductive definitions
➜ Definition by intersection
➜ Computation by iteration
➜ Formalisation in Isabelle
➜ Rule Induction in Isar

Slide 24