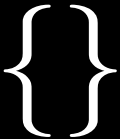




COMP4161: Advanced Topics in Software Verification



Gerwin Klein, June Andronick, Christine Rizkallah, Miki Tanaka
S2/2018

data61.csiro.au



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
 - Inductively defined sets, rule induction [5]
 - Datatypes, recursion, induction [6, 7]
 - Hoare logic, proofs about programs, invariants [8^b, 9]
 - (mid-semester break)
 - C verification [10]
 - CakeML, Isar [11^c]
 - Concurrency [12]

^aa1 due; ^ba2 due; ^ca3 due

Last Time



- Sets
- Type Definitions
- Inductive Definitions



DATA
61



Inductive Definitions

How They Work

The Nat Example



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

- N is the set of natural numbers \mathbb{N}
- But why not the set of real numbers? $0 \in \mathbb{R}, n \in \mathbb{R} \implies n + 1 \in \mathbb{R}$
- \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 .
- Gives rise to a nice proof principle (rule induction)

Formally



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

define set $X \subseteq A$

Formally: set of rules $R \subseteq A \text{ set} \times A$ (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:

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

The Set



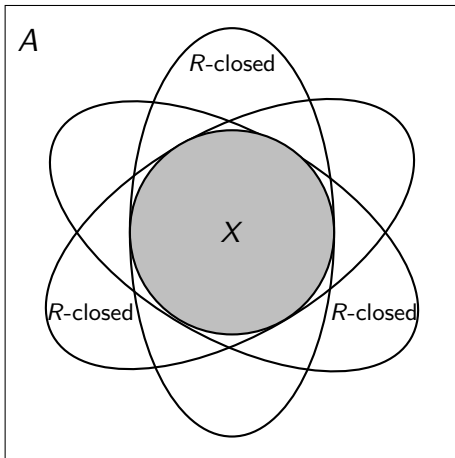
Definition: B is R -closed iff $\hat{R} B \subseteq B$

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

This does always exist:

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

Generation from Above



Rule Induction



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

induces induction principle

$$\llbracket P 0; \bigwedge n. P n \implies P (n + 1) \rrbracket \implies \forall x \in N. P x$$

In general:

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

Why does this work?



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

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

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

Rules with side conditions



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

induction scheme:

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

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:

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

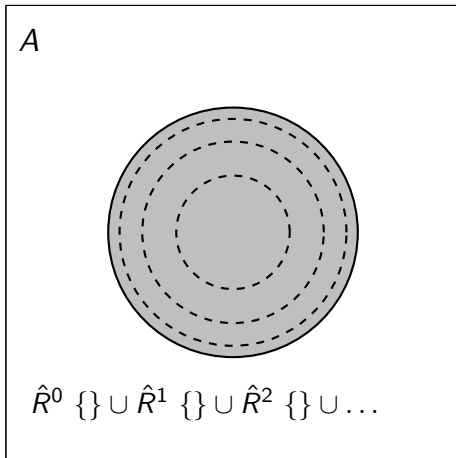
$$X_1 = \hat{R}^1 \{\} = \text{rules without hypotheses}$$

\vdots

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

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

Generation from Below



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** $f A :: (\alpha \text{ set} \Rightarrow \alpha \text{ set}) \Rightarrow \alpha \text{ set} \Rightarrow \text{bool}$
- Show $\text{closed } f A \wedge \text{closed } f B \implies \text{closed } f (A \cap B)$ if f is monotone (**mono** is predefined)
- Define **lfpt** f as the intersection of all f -closed sets
- Show that $\text{lfpt } f$ is a fixpoint of f if f is monotone
- Show that $\text{lfpt } f$ is the least fixpoint of f
- Declare a constant $R :: (\alpha \text{ set} \times \alpha) \text{ set}$
- Define $\hat{R} :: \alpha \text{ set} \Rightarrow \alpha \text{ set}$ in terms of R
- 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