COMP 4161

NICTA Advanced Course

Advanced Topics in Software Verification

Gerwin Klein
Formal Methods

$\{\}$

CONTENT

LAST TIME

- ➜ Permutative rewriting, AC rules
- ➜ More confluence: critical pairs
- ➜ Knuth-Bendix Algorithm, Waldmeister
- ➜ More Isar: forward, backward, obtain, abbreviations, moreover

ISAR EXERCISE

- Give an Isar proof of the rich grandmother theorem
  (automated methods allowed, but proof must be explaining)

Building up Specification Techniques

Sets in Isabelle

Type **'a set**: sets over type 'a

- $\{\}$, $\{e_1, \ldots, e_n\}$, $\{x.\ P\ x\}$
- $e \in A$, $A \subseteq B$
- $A \cup B$, $A \cap B$, $A - B$, $-A$
- $\bigcup x \in A.\ B\ x$, $\bigcap x \in A.\ B\ x$, $\bigcap\ A$, $\bigcup A$
- $\{i..j\}$
- insert :: $\alpha \Rightarrow \alpha$ set $\Rightarrow \alpha$ set
- $f`A \equiv \{y.\ \exists x \in A.\ y = f\ x\}$
- $\ldots$

Natural deduction proofs:

- equalityI: $[\![A \subseteq B;\ B \subseteq A]\!] \Longrightarrow A = B$
- subsetI: $(\bigwedge x.\ x \in A \Longrightarrow x \in B) \Longrightarrow A \subseteq B$
- ... (see Tutorial)

Bounded Quantifiers

- $\forall x \in A.\ P\ x \equiv \forall x.\ x \in A \longrightarrow P\ x$
- $\exists x \in A.\ P\ x \equiv \exists x.\ x \in A \wedge P\ x$

- ballI: $(\bigwedge x.\ x \in A \Longrightarrow P\ x) \Longrightarrow \forall x \in A.\ P\ x$
- bspec: $[\![\forall x \in A.\ P\ x; x \in A]\!] \Longrightarrow P\ x$

- bexI: $[\![P\ x; x \in A]\!] \Longrightarrow \exists x \in A.\ P\ x$
- bexE: $[\![\exists x \in A.\ P\ x; \bigwedge x.\ [\![x \in A; P\ x]\!] \Longrightarrow Q]\!] \Longrightarrow Q$

DEMO: SETS

Inductive Definitions

Example

$$\frac{}{\langle \mathsf{skip}, \sigma\rangle \longrightarrow \sigma} \qquad \frac{[\![e]\!]\sigma = v}{\langle \mathsf{x := e}, \sigma\rangle \longrightarrow \sigma[x \mapsto v]}$$

$$\frac{\langle c_1, \sigma\rangle \longrightarrow \sigma' \quad \langle c_2, \sigma'\rangle \longrightarrow \sigma''}{\langle c_1; c_2, \sigma\rangle \longrightarrow \sigma''}$$

$$\frac{[\![b]\!]\sigma = \mathsf{False}}{\langle \mathsf{while}\ b\ \mathsf{do}\ c, \sigma\rangle \longrightarrow \sigma}$$

$$\frac{[\![b]\!]\sigma = \mathsf{True} \quad \langle c, \sigma\rangle \longrightarrow \sigma' \quad \langle \mathsf{while}\ b\ \mathsf{do}\ c, \sigma'\rangle \longrightarrow \sigma''}{\langle \mathsf{while}\ b\ \mathsf{do}\ c, \sigma\rangle \longrightarrow \sigma''}$$

What does this mean?

- ➜ $\langle c, \sigma \rangle \longrightarrow \sigma'$ fancy syntax for a relation $(c, \sigma, \sigma') \in E$
- ➜ relations are sets: $E :: (\mathsf{com} \times \mathsf{state} \times \mathsf{state})\ \mathsf{set}$
- ➜ the rules define a set inductively

But which set?

SIMPLER 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)
- ➜ Alternative (greatest set) occasionally also useful: coinduction

Rules $\dfrac{a_1 \in X \quad \ldots \quad a_n \in X}{a \in X}$ 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 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:

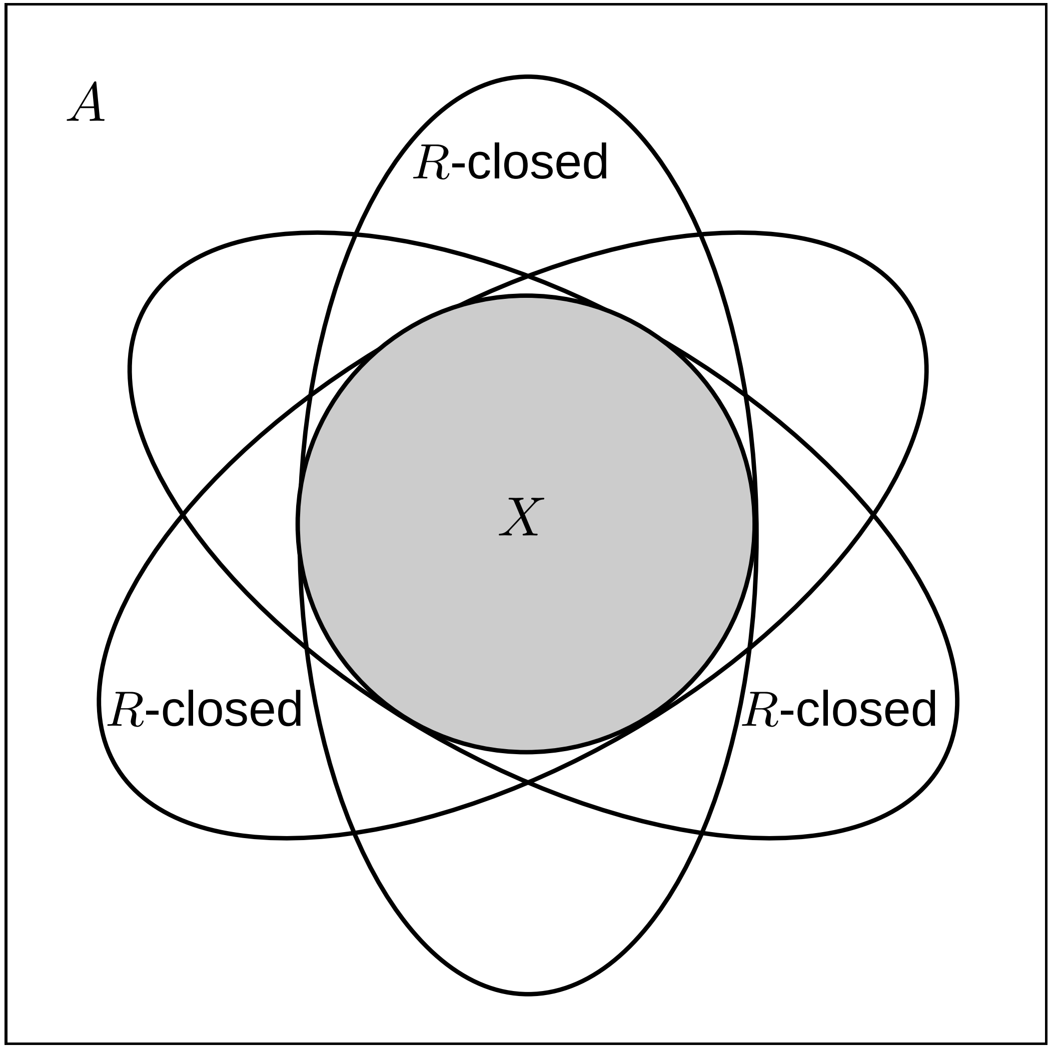
$$
\begin{aligned}
R &\equiv \{(\{\}, 0)\} \cup \{(\{n\}, n+1).\ n \in \mathbb{R}\} \\
\hat{R}\ \{3, 6, 10\} &= \{0, 4, 7, 11\}
\end{aligned}
$$

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



A
R-closed
X
R-closed
R-closed

Rule Induction

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

induces induction principle

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

In general:

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

Why does this work?

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

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

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

RULES WITH SIDE CONDITIONS

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

induction scheme:

$$(\forall (\{a_1, \dots a_n\}, a) \in R.\ P\ a_1 \wedge \dots \wedge P\ a_n \wedge$$
$$C_1 \wedge \dots \wedge C_m \wedge$$
$$\{a_1, \dots, a_n\} \subseteq X \Longrightarrow P\ a)$$

$$\Longrightarrow$$

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

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:

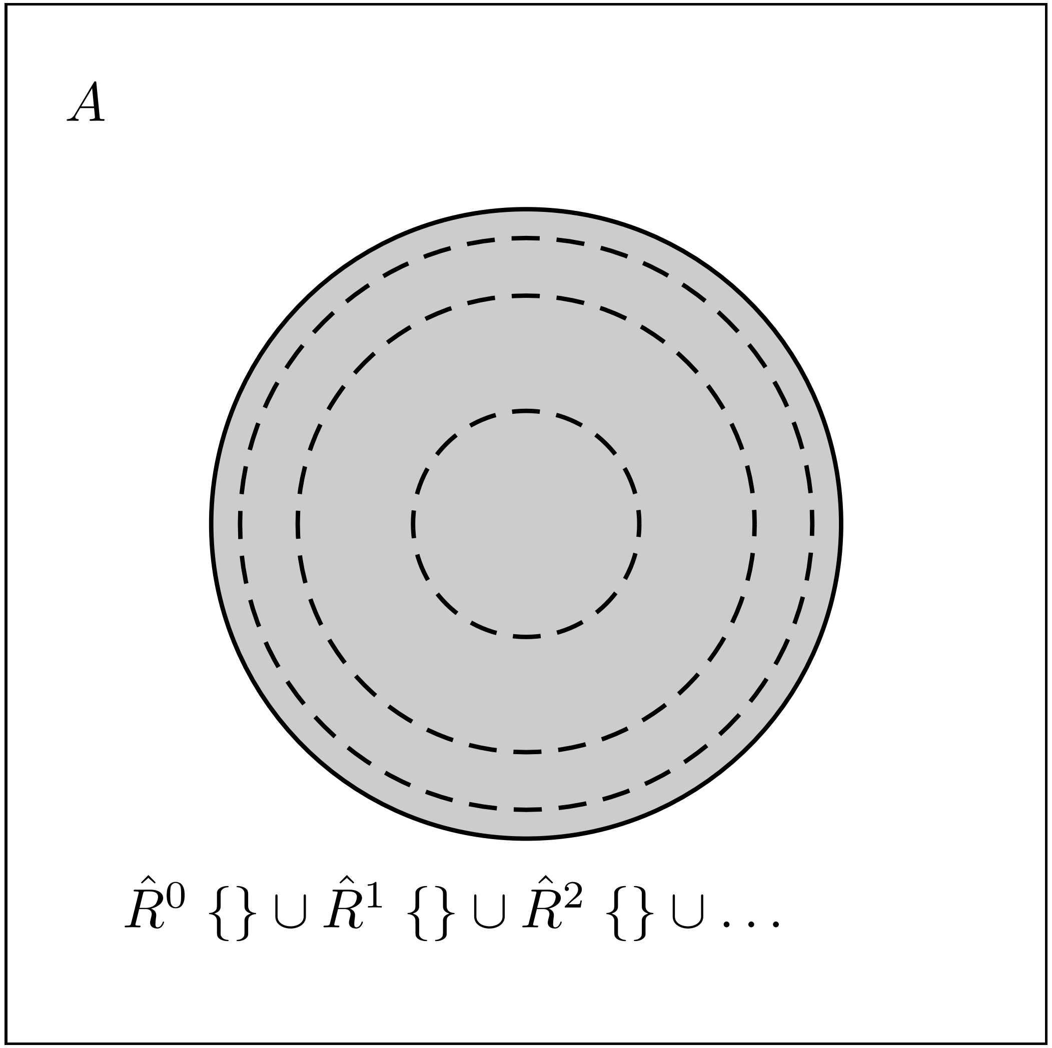
$$X_0 = \hat{R}^0\ \{\} = \{\}$$

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

$$\vdots$$

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

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


A
$\hat{R}^0\,\{\} \cup \hat{R}^1\,\{\} \cup \hat{R}^2\,\{\} \cup \ldots$

DEMO: INDUCTIVE DEFINITONS

We have seen today ...

- Sets in Isabelle
- Inductive Definitions
- Rule induction
- Fixpoints

Formalize this lecture in Isabelle:

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