Operational Semantics

Hongfei Fu

John Hopcroft Center for Computer Science
Shanghai Jiao Tong University

Mar. 18th

Previous Lecture

- equivalence of commands through rules
- small-step (one-step) operational semantics
- inductive proofs of program properties

Today's Lecture

- rule induction
- inductive definitions

Rule Induction

textbook, Page 42 – 51

Rule Induction

Rule Instances (X/y)

- E: a set of elements
- $X \subseteq E$: premise (a finite set of elements)
- $y \in E$: conclusion (a single element)

Rule Induction

Axiom Instances: $\emptyset/y$

$$\frac{}{y}$$

Non-axiom Rule Instances: $\{x_1, \ldots, x_n\}/y$

$$\frac{x_1, \ldots, x_n}{y}$$

Rule Induction

Recall: Derivation Trees

- E: a set of elements
- R: a set of rule instances
- $y \in E$: an element

An R-derivation of y is

- either a rule instance $(\emptyset / y)$
- or $(\{d_1, \ldots, d_n\} / y)$ such that
  - $(\{x_1, \ldots, x_n\} / y)$ is a rule instance
  - each d_i is a (smaller) R-derivation of x_i

Rule Induction

Recall: Notations

- R: a set of rule instances
- d: an R-derivation
- y: an element

Then we denote

- $d \Vdash_R y$: d is an R-derivation of y.
- $\Vdash_R y$: $d \Vdash_R y$ for some derivation d.
- $d \Vdash y, \Vdash y$: omission of R

Rule Induction

Recall: Properties

- R: a set of rule instances
- d: an R-derivation
- y: an element

Then we have:

- $(\emptyset/y) \Vdash_R y$ if $(\emptyset/y) \in R$;
- $(\{d_1, \ldots, d_n\}/y) \Vdash_R y$ if $(\{x_1, \ldots, x_n\}/y) \in R$ and $d_1 \Vdash_R x_1, \ldots, d_n \Vdash_R x_n$;

Rule Induction

Notation

- E: a set of elements
- R: a set of rule instances

We define $I_R := \{y \in E \mid \Vdash_R y\}$.

Rule Induction

The Principle

- E: a set of elements
- R: a set of rule instances
- $I_R = \{y \in E \mid \Vdash_R y\}$
- P: a predicate over I_R

Then we have that $\forall x \in I_R.P(x)$ iff

$$\forall (X/y) \in R.\,[(X \subseteq I_R \;\&\; \forall x \in X.P(x)) \Rightarrow P(y)] \quad .$$

Rule Induction

The Principle

$$\forall (X/y) \in R.\, [(X \subseteq I_R \;\&\; \forall x \in X.P(x)) \Rightarrow P(y)]$$

- base step: $X = \emptyset$ (axioms)
- inductive step: $X \neq \emptyset$

Rule Induction

Theorem

- E: a set of elements
- R: a set of rule instances
- $I_R = \{y \in E \mid \Vdash_R y\}$
- P: a predicate over I_R

Then we have that $\forall x \in I_R.P(x)$ iff

$$\forall (X/y) \in R.\,[(X \subseteq I_R \;\&\; \forall x \in X.P(x)) \Rightarrow P(y)] \quad .$$

Rule Induction

Closedness

- R: a set of rule instances
- Q: a set of elements

We say that Q is closed under R (or R-closed) if

$$\forall (X/y) \in R.\, (X \subseteq Q \Rightarrow y \in Q) \quad .$$

Rule Induction

Proposition

- R: a set of rule instances
- Q: a set of elements

Then we have:

- I_R is R-closed;
- if Q is R-closed, then $I_R \subseteq Q$.

Rule Induction

Proposition

- R: a set of rule instances
- Q: a set of elements

Then we have:

- I_R is R-closed;
- if Q is R-closed, then $I_R \subseteq Q$.

Proof

- from definition of I_R
- by induction on derivation trees:

$$P(d) := \forall y.\, [d \Vdash_R y \Rightarrow y \in Q]$$

Rule Induction

Theorem

- E: a set of elements
- R: a set of rule instances
- $I_R = \{y \in E \mid \Vdash_R y\}$
- P: a predicate over I_R

Then we have that $\forall x \in I_R.P(x)$ iff

$$\forall (X/y) \in R.\,[(X \subseteq I_R \;\&\; \forall x \in X.P(x)) \Rightarrow P(y)] \quad .$$

Rule Induction

Theorem

- E: a set of elements
- R: a set of rule instances
- $I_R = \{y \in E \mid \Vdash_R y\}$
- P: a predicate over I_R

Then we have that $\forall x \in I_R.P(x)$ iff

$$\forall (X/y) \in R.\,[(X \subseteq I_R \;\&\; \forall x \in X.P(x)) \Rightarrow P(y)] \quad .$$

Proof

Rule Induction

Theorem

- E: a set of elements
- R: a set of rule instances
- $I_R = \{y \in E \mid \Vdash_R y\}$
- P: a predicate over I_R

Then we have that $\forall x \in I_R.P(x)$ iff

$$\forall (X/y) \in R.\,[(X \subseteq I_R \;\&\; \forall x \in X.P(x)) \Rightarrow P(y)] \quad .$$

Proof

- $Q := \{x \in I_R \mid P(x)\}$ and $Q \subseteq I_R$;

Rule Induction

Theorem

- E: a set of elements
- R: a set of rule instances
- $I_R = \{y \in E \mid \Vdash_R y\}$
- P: a predicate over I_R

Then we have that $\forall x \in I_R.P(x)$ iff

$$\forall (X/y) \in R.\,[(X \subseteq I_R \;\&\; \forall x \in X.P(x)) \Rightarrow P(y)] \quad .$$

Proof

- $Q := \{x \in I_R \mid P(x)\}$ and $Q \subseteq I_R$;
- Q is R-closed and $I_R \subseteq Q$;

Rule Induction

Theorem

- E: a set of elements
- R: a set of rule instances
- $I_R = \{y \in E \mid \Vdash_R y\}$
- P: a predicate over I_R

Then we have that $\forall x \in I_R.P(x)$ iff

$$\forall (X/y) \in R.\,[(X \subseteq I_R \;\&\; \forall x \in X.P(x)) \Rightarrow P(y)] \quad .$$

Proof

- $Q := \{x \in I_R \mid P(x)\}$ and $Q \subseteq I_R$;
- Q is R-closed and $I_R \subseteq Q$;
- $I_R = Q$ and $\forall x \in I_R.P(x)$.

Rule Induction

Example: Induction on Derivation Trees

- $\forall c, \sigma, \sigma', \sigma''. [(\langle c, \sigma \rangle \to \sigma' \ \& \ \langle c, \sigma \rangle \to \sigma'') \Rightarrow \sigma' = \sigma'']$.
- $\forall a, \sigma, n', n''. [(\langle a, \sigma \rangle \to n' \ \& \ \langle a, \sigma \rangle \to n'') \Rightarrow n' = n'']$.
- $\forall b, \sigma, t', t''. [(\langle b, \sigma \rangle \to t' \ \& \ \langle b, \sigma \rangle \to t'') \Rightarrow t' = t'']$.

Rule Induction

Example: Induction on Derivation Trees

- $\forall c, \sigma, \sigma', \sigma''. [(\langle c, \sigma \rangle \to \sigma' \ \& \ \langle c, \sigma \rangle \to \sigma'') \Rightarrow \sigma' = \sigma'']$.
- $\forall a, \sigma, n', n''. [(\langle a, \sigma \rangle \to n' \ \& \ \langle a, \sigma \rangle \to n'') \Rightarrow n' = n'']$.
- $\forall b, \sigma, t', t''. [(\langle b, \sigma \rangle \to t' \ \& \ \langle b, \sigma \rangle \to t'') \Rightarrow t' = t'']$.

We define:

- $P_1(c, \sigma, \sigma') := \forall \sigma''. [\langle c, \sigma \rangle \to \sigma'' \Rightarrow \sigma' = \sigma'']$.
- $P_2(a, \sigma, n') := \forall n''. [\langle a, \sigma \rangle \to n'' \Rightarrow n' = n'']$.
- $P_3(b, \sigma, t') := \forall t''. [\langle b, \sigma \rangle \to t'' \Rightarrow t' = t'']$.
- $P := (\text{“}\textbf{Aexp}\text{”} \Rightarrow P_1) \ \& \ (\text{“}\textbf{Bexp}\text{”} \Rightarrow P_2) \ \& \ (\text{“}\textbf{Com}\text{”} \Rightarrow P_3)$
  (i.e., we aggregate all the three cases)

Rule Induction

Special Rule Induction

- general rule induction: a property for all elements
- special rule induction: a property for a part of elements

Rule Induction

Special Rule Induction

- R: a set of rule instances
- $A \subseteq I_R$: a subset
- Q: a predicate over I_R

Then we have that $\forall a \in A.Q(a)$ iff

$$\forall (X/y) \in R.\, [(X \subseteq I_R \;\&\; y \in A \;\&\; (\forall x \in X \cap A.Q(x))) \Rightarrow Q(y)] \quad .$$

Rule Induction

Special Rule Induction

We have that $\forall a \in A.Q(a)$ iff

$$\forall (X/y) \in R.[(X \subseteq I_R \ \& \ y \in A \ \& \ (\forall x \in X \cap A.Q(x))) \Rightarrow Q(y)] \quad .$$

Proof

Rule Induction

Special Rule Induction

We have that $\forall a \in A.Q(a)$ iff

$$\forall (X/y) \in R.[(X \subseteq I_R \ \& \ y \in A \ \& \ (\forall x \in X \cap A.Q(x))) \Rightarrow Q(y)] \quad .$$

Proof

- $P(x) := x \in A \Rightarrow Q(x)$ and $\forall a \in I_R.P(a) \Leftrightarrow \forall a \in A.Q(a)$;

Rule Induction

Special Rule Induction

We have that $\forall a \in A.Q(a)$ iff

$$\forall (X/y) \in R.[(X \subseteq I_R \ \& \ y \in A \ \& \ (\forall x \in X \cap A.Q(x))) \Rightarrow Q(y)] \ .$$

Proof

- $P(x) := x \in A \Rightarrow Q(x)$ and $\forall a \in I_R.P(a) \Leftrightarrow \forall a \in A.Q(a)$;
- $\forall x \in I_R.P(x)$ iff $\forall (X/y) \in R.[(X \subseteq I_R \ \& \ \forall x \in X.P(x)) \Rightarrow P(y)]$;

Rule Induction

Special Rule Induction

We have that $\forall a \in A.Q(a)$ iff

$$\forall (X/y) \in R.[(X \subseteq I_R \ \& \ y \in A \ \& \ (\forall x \in X \cap A.Q(x))) \Rightarrow Q(y)] \ .$$

Proof

- $P(x) := x \in A \Rightarrow Q(x)$ and $\forall a \in I_R.P(a) \Leftrightarrow \forall a \in A.Q(a)$;
- $\forall x \in I_R.P(x)$ iff $\forall (X/y) \in R.[(X \subseteq I_R \ \& \ \forall x \in X.P(x)) \Rightarrow P(y)]$;
- $\forall (X/y) \in R.[(X \subseteq I_R \ \& \ \forall x \in X.(x \in A \Rightarrow Q(x))) \Rightarrow (y \in A \Rightarrow Q(y))]$;

Rule Induction

Special Rule Induction

We have that $\forall a \in A.Q(a)$ iff

$$\forall (X/y) \in R.[(X \subseteq I_R \ \& \ y \in A \ \& \ (\forall x \in X \cap A.Q(x))) \Rightarrow Q(y)] \ .$$

Proof

- $P(x) := x \in A \Rightarrow Q(x)$ and $\forall a \in I_R.P(a) \Leftrightarrow \forall a \in A.Q(a)$;
- $\forall x \in I_R.P(x)$ iff $\forall (X/y) \in R.[(X \subseteq I_R \ \& \ \forall x \in X.P(x)) \Rightarrow P(y)]$;
- $\forall (X/y) \in R.[(X \subseteq I_R \ \& \ \forall x \in X.(x \in A \Rightarrow Q(x))) \Rightarrow (y \in A \Rightarrow Q(y))]$;
- $\forall (X/y) \in R.[(X \subseteq I_R \ \& \ \forall x \in X \cap A.Q(x)) \Rightarrow (y \in A \Rightarrow Q(y))]$;
- $\forall (X/y) \in R.[(X \subseteq I_R \ \& \ y \in A \ \& \ \forall x \in X \cap A.Q(x)) \Rightarrow Q(y)]$;

Rule Induction

Example

- Y: a location (program variable)

Then $\forall c, \sigma, \sigma'. [(Y \notin \mathrm{loc}(c) \ \& \ \langle c, \sigma \rangle \to \sigma') \Rightarrow \sigma(Y) = \sigma'(Y)]$.

Rule Induction

Example

- Y: a location (program variable)

Then $\forall c, \sigma, \sigma'. [(Y \notin \mathrm{loc}(c) \ \& \ \langle c, \sigma \rangle \to \sigma') \Rightarrow \sigma(Y) = \sigma'(Y)]$.

Proof

Rule Induction

Example

- Y: a location (program variable)

Then $\forall c, \sigma, \sigma'. [(Y \notin \mathrm{loc}(c) \ \& \ \langle c, \sigma \rangle \to \sigma') \Rightarrow \sigma(Y) = \sigma'(Y)]$.

Proof

- $A := \{\langle c, \sigma \rangle \to \sigma' \mid Y \notin \mathrm{loc}(c)\}$.

Rule Induction

Example

- Y: a location (program variable)

Then $\forall c, \sigma, \sigma'. [(Y \notin \mathrm{loc}(c) \ \& \ \langle c, \sigma \rangle \to \sigma') \Rightarrow \sigma(Y) = \sigma'(Y)]$.

Proof

- $A := \{\langle c, \sigma \rangle \to \sigma' \mid Y \notin \mathrm{loc}(c)\}$.
- $Q(c, \sigma, \sigma') := \sigma(Y) = \sigma'(Y)$.

Rule Induction

Example

- Y: a location (program variable)

Then $\forall c, \sigma, \sigma'. [(Y \notin \mathrm{loc}(c) \;\&\; \langle c, \sigma \rangle \to \sigma') \Rightarrow \sigma(Y) = \sigma'(Y)]$.

Proof

- $A := \{\langle c, \sigma \rangle \to \sigma' \mid Y \notin \mathrm{loc}(c)\}$.
- $Q(c, \sigma, \sigma') := \sigma(Y) = \sigma'(Y)$.
- $\forall \langle c, \sigma \rangle \to \sigma' \in A.\ Q(c, \sigma, \sigma')$.

Rule Induction

Example

- Y: a location (program variable)

Then $\forall c, \sigma, \sigma'. [(Y \notin \mathrm{loc}(c) \ \& \ \langle c, \sigma \rangle \to \sigma') \Rightarrow \sigma(Y) = \sigma'(Y)]$.

Proof

- $A := \{\langle c, \sigma \rangle \to \sigma' \mid Y \notin \mathrm{loc}(c)\}$.
- $Q(c, \sigma, \sigma') := \sigma(Y) = \sigma'(Y)$.
- $\forall \langle c, \sigma \rangle \to \sigma' \in A.\ Q(c, \sigma, \sigma')$.
- $\forall (X/y) \in R. [(X \subseteq I_R \ \& \ y \in A \ \& \ (\forall x \in X \cap A. Q(x))) \Rightarrow Q(y)]$

Rule Induction

Proof

- $A := \{\langle c, \sigma\rangle \to \sigma' \mid Y \notin \mathrm{loc}(c)\}$.
- $Q(c, \sigma, \sigma') := \sigma(Y) = \sigma'(Y)$.
- $\forall \langle c, \sigma\rangle \to \sigma' \in A.\ Q(c, \sigma, \sigma')$.
- $\forall (X/y) \in R.\ [(X \subseteq I_R\ \&\ y \in A\ \&\ (\forall x \in X \cap A. Q(x))) \Rightarrow Q(y)]$

$$\frac{}{\langle \mathbf{skip}, \sigma\rangle \to \sigma} \qquad \frac{\dfrac{\vdots}{\langle a, \sigma\rangle \to m}}{\langle X := a, \sigma\rangle \to \sigma\,[m/X]}$$

$$\frac{\dfrac{\vdots}{\langle b, \sigma\rangle \to \mathbf{true}} \qquad \dfrac{\vdots}{\langle c, \sigma\rangle \to \sigma''} \qquad \dfrac{\vdots}{\langle \mathbf{while}\ b\ \mathbf{do}\ c, \sigma''\rangle \to \sigma'}}{\langle \mathbf{while}\ b\ \mathbf{do}\ c, \sigma\rangle \to \sigma'}$$

Rule Induction

Another Example

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Rule Induction

Another Example

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Proof

Rule Induction

Another Example

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Proof

- $A := \{(c, \sigma, \sigma') \mid \langle c, \sigma \rangle \to \sigma' \ \& \ c = w\}$;

Rule Induction

Another Example

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Proof

- $A := \{(c, \sigma, \sigma') \mid \langle c, \sigma \rangle \to \sigma' \ \& \ c = w\}$;
- $Q :=$ **false**;

Rule Induction

Another Example

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Proof

- $A := \{(c, \sigma, \sigma') \mid \langle c, \sigma \rangle \to \sigma' \ \& \ c = w\}$;
- $Q :=$ **false**;
- $A = \emptyset \Leftrightarrow \forall a \in A.\ Q(a)$;

Rule Induction

Another Example

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Proof

- $A := \{(c, \sigma, \sigma') \mid \langle c, \sigma \rangle \to \sigma' \ \& \ c = w\}$;
- $Q :=$ **false**;
- $A = \emptyset \Leftrightarrow \forall a \in A.\ Q(a)$;
- $\forall (X/y) \in R. [(X \subseteq I_R \ \& \ y \in A \ \& \ (\forall x \in X \cap A. Q(x))) \Rightarrow Q(y)]$

Rule Induction

Proof

- $w := \textbf{while true do skip}$
- $A := \{(c, \sigma, \sigma') \mid \langle c, \sigma \rangle \to \sigma' \ \& \ c = w\}$;
- $Q := \textbf{false}$;
- $A = \emptyset \Leftrightarrow \forall a \in A.\ Q(a)$;
- $\forall (X/y) \in R.\ [(X \subseteq I_R \ \& \ y \in A \ \& \ (\forall x \in X \cap A. Q(x))) \Rightarrow Q(y)]$

$$\frac{\vdots \qquad \vdots \qquad \vdots}{\langle b, \sigma \rangle \to \textbf{true} \qquad \langle c, \sigma \rangle \to \sigma'' \qquad \langle \textbf{while } b \textbf{ do } c, \sigma'' \rangle \to \sigma'}{\langle \textbf{while } b \textbf{ do } c, \sigma \rangle \to \sigma'}$$

Inductive Definitions

textbook, Page 39 – 40

Inductive Definitions

Intuition

- A: a nonempty set
- $h : A \to A$: a function
- $a \in A$: an initial element

There is an infinite sequence $a_0, a_1, \ldots$ such that $a_0 = a$ and $a_{n+1} = h(a_n)$.

Inductive Definitions

The Recursion Theorem

- A: a nonempty set
- $h : A \to A$: a function
- $a \in A$: an initial element

There exists a unique function $f : \mathbb{N} \to A$ such that $f(0) = a$ and $f(n+1) = h(f(n))$.

The Proof Sketch

Inductive Definitions

The Proof Sketch

- T: the set of all functions $g : \{0, \ldots, n\} \to A$ such that
  - $g(0) = a$;
  - $g(k+1) = h(g(k))$ for all $0 \leq k < n$;

Inductive Definitions

The Proof Sketch

- T: the set of all functions $g : \{0, \ldots, n\} \to A$ such that
  - $g(0) = a$;
  - $g(k+1) = h(g(k))$ for all $0 \leq k < n$;
- existence: $f := \{(n, a') \in \mathbb{N} \times A \mid \exists g \in T.g(n) = a'\}$
  - for all n, there is $g \in T$ such that $g(n) = a$ for some $a \in A$;
  - for all n, there exists a unique $a \in A$ such that $(n, a) \in f$;

Inductive Definitions

The Proof Sketch

- T: the set of all functions $g : \{0, \ldots, n\} \to A$ such that
  - $g(0) = a$;
  - $g(k+1) = h(g(k))$ for all $0 \leq k < n$;
- existence: $f := \{(n, a') \in \mathbb{N} \times A \mid \exists g \in T.g(n) = a'\}$
  - for all n, there is $g \in T$ such that $g(n) = a$ for some $a \in A$;
  - for all n, there exists a unique $a \in A$ such that $(n, a) \in f$;
- uniqueness: for $f, g : \mathbb{N} \to A$, if $f(0) = g(0) = a$, $f(n+1) = h(f(n))$ and $g(n+1) = h(g(n))$, then we have that $f = g$.

Inductive Definition

Application

- the set of all **IMP** programs
- the set of all derivation trees

Inductive Definitions

Derivation Trees

- R: a set of rule instances

Then we have

Inductive Definitions

Derivation Trees

- R: a set of rule instances

Then we have

- $D_0 := \{(X/y) \in R \mid X = \emptyset\}$

Inductive Definitions

Derivation Trees

- R: a set of rule instances

Then we have

- $D_0 := \{(X/y) \in R \mid X = \emptyset\}$
- $d \in D_{n+1}$ iff
  - either $d \in D_0$,
  - or $d = \{d_1, \ldots, d_n\}/y$ for some $d_1, \ldots, d_n \in D_n$ and $(x_1, \ldots, x_n)/y \in R$ such that d_i is rooted at x_i $(1 \leq i \leq n)$

Inductive Definitions

Derivation Trees

- R: a set of rule instances

Then we have

- $D_0 := \{(X/y) \in R \mid X = \emptyset\}$
- $d \in D_{n+1}$ iff
  - either $d \in D_0$,
  - or $d = \{d_1, \ldots, d_n\}/y$ for some $d_1, \ldots, d_n \in D_n$ and $(x_1, \ldots, x_n)/y \in R$ such that d_i is rooted at x_i $(1 \leq i \leq n)$
- $D := \bigcup_n D_n$

Inductive Definitions

Definition of I_R

- E: a set of elements
- R: a set of rule instances where all elements are from E

Then we have

Inductive Definitions

Definition of I_R

- E: a set of elements
- R: a set of rule instances where all elements are from E

Then we have

- $\widehat{R} : 2^E \to 2^E$: $\widehat{R}(B) := \{y \in E \mid \exists X \subseteq B.\, (X/y) \in R\}$

Inductive Definitions

Definition of I_R

- E: a set of elements
- R: a set of rule instances where all elements are from E

Then we have

- $\widehat{R} : 2^E \to 2^E$: $\widehat{R}(B) := \{y \in E \mid \exists X \subseteq B.\, (X/y) \in R\}$
- $A \subseteq B \Rightarrow \widehat{R}(A) \subseteq \widehat{R}(B)$

Inductive Definitions

Definition of I_R

- E: a set of elements
- R: a set of rule instances where all elements are from E

Then we have

- $\widehat{R} : 2^E \to 2^E$: $\widehat{R}(B) := \{y \in E \mid \exists X \subseteq B.\, (X/y) \in R\}$
- $A \subseteq B \Rightarrow \widehat{R}(A) \subseteq \widehat{R}(B)$
- $A_0 := \emptyset$, $A_{n+1} := \widehat{R}(A_n)$

Inductive Definitions

Definition of I_R

- E: a set of elements
- R: a set of rule instances where all elements are from E

Then we have

- $\widehat{R} : 2^E \to 2^E$: $\widehat{R}(B) := \{y \in E \mid \exists X \subseteq B.\, (X/y) \in R\}$
- $A \subseteq B \Rightarrow \widehat{R}(A) \subseteq \widehat{R}(B)$
- $A_0 := \emptyset$, $A_{n+1} := \widehat{R}(A_n)$
- $A_0 \subseteq A_1 \subseteq \cdots \subseteq A_n \subseteq \ldots$

Inductive Definitions

Definition of I_R

- E: a set of elements
- R: a set of rule instances where all elements are from E

Then we have

- $\widehat{R} : 2^E \to 2^E$: $\widehat{R}(B) := \{y \in E \mid \exists X \subseteq B.\, (X/y) \in R\}$
- $A \subseteq B \Rightarrow \widehat{R}(A) \subseteq \widehat{R}(B)$
- $A_0 := \emptyset$, $A_{n+1} := \widehat{R}(A_n)$
- $A_0 \subseteq A_1 \subseteq \cdots \subseteq A_n \subseteq \ldots$
- $I_R = \bigcup_n A_n$.

Inductive Definitions

Definition of I_R

- E: a set of elements
- R: a set of rule instances where all elements are from E

Then we have

- $\widehat{R} : 2^E \to 2^E$: $\widehat{R}(B) := \{y \in E \mid \exists X \subseteq B.\, (X/y) \in R\}$
- $A \subseteq B \Rightarrow \widehat{R}(A) \subseteq \widehat{R}(B)$
- $A_0 := \emptyset$, $A_{n+1} := \widehat{R}(A_n)$
- $A_0 \subseteq A_1 \subseteq \cdots \subseteq A_n \subseteq \ldots$
- $I_R = \bigcup_n A_n$.

Proposition

- I_R is R-closed;
- if Q is R-closed, then $I_R \subseteq Q$.

Inductive Definitions

Proposition

- I_R is R-closed;
- if Q is R-closed, then $I_R \subseteq Q$.

Proof

Inductive Definitions

Proposition

- I_R is R-closed;
- if Q is R-closed, then $I_R \subseteq Q$.

Proof

- our goal: I_R is R-closed.

Inductive Definitions

Proposition

- I_R is R-closed;
- if Q is R-closed, then $I_R \subseteq Q$.

Proof

- our goal: I_R is R-closed.
- $(X/y) \in R$ and $X \subseteq I_R$

Inductive Definitions

Proposition

- I_R is R-closed;
- if Q is R-closed, then $I_R \subseteq Q$.

Proof

- our goal: I_R is R-closed.
- $(X/y) \in R$ and $X \subseteq I_R$
- $X \subseteq A_n$ for some n

Inductive Definitions

Proposition

- I_R is R-closed;
- if Q is R-closed, then $I_R \subseteq Q$.

Proof

- our goal: I_R is R-closed.
- $(X/y) \in R$ and $X \subseteq I_R$
- $X \subseteq A_n$ for some n
- $y \in A_{n+1} \subseteq I_R$

Inductive Definitions

Proposition

- I_R is R-closed;
- if Q is R-closed, then $I_R \subseteq Q$.

Proof

Inductive Definitions

Proposition

- I_R is R-closed;
- if Q is R-closed, then $I_R \subseteq Q$.

Proof

- our goal: if Q is R-closed, then $I_R \subseteq Q$.

Inductive Definitions

Proposition

- I_R is R-closed;
- if Q is R-closed, then $I_R \subseteq Q$.

Proof

- our goal: if Q is R-closed, then $I_R \subseteq Q$.
- proof by induction on n: $A_n \subseteq Q$

Inductive Definitions

The Example Again (textbook, Page 39)

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Inductive Definitions

The Example Again (textbook, Page 39)

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Proof (by Contradiction)

Inductive Definitions

The Example Again (textbook, Page 39)

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Proof (by Contradiction)

Inductive Definitions

The Example Again (textbook, Page 39)

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Proof (by Contradiction)

- Suppose that $\exists \sigma, \sigma'. \langle w, \sigma \rangle \to \sigma'$.

Inductive Definitions

The Example Again (textbook, Page 39)

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Proof (by Contradiction)

- Suppose that $\exists \sigma, \sigma'. \langle w, \sigma \rangle \to \sigma'$.
- $(w, \sigma, \sigma') \in A_n$ for some n.

Inductive Definitions

The Example Again (textbook, Page 39)

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Proof (by Contradiction)

- Suppose that $\exists \sigma, \sigma'. \langle w, \sigma \rangle \to \sigma'$.
- $(w, \sigma, \sigma') \in A_n$ for some n.
- Let n^* be the least such that $(w, \sigma, \sigma') \in A_{n^*}$ for some w, σ, σ'.

Inductive Definitions

The Example Again (textbook, Page 39)

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Proof (by Contradiction)

- Suppose that $\exists \sigma, \sigma'. \langle w, \sigma \rangle \to \sigma'$.
- $(w, \sigma, \sigma') \in A_n$ for some n.
- Let n^* be the least such that $(w, \sigma, \sigma') \in A_{n^*}$ for some w, σ, σ'.
- $n^* > 0$ and $(w, \sigma, \sigma') \in A_{n^*-1}$.

Inductive Definitions

The Example Again (textbook, Page 39)

- $w :=$ **while true do skip**

We prove that $\forall \sigma, \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.

Proof (by Contradiction)

- Suppose that $\exists \sigma, \sigma'. \langle w, \sigma \rangle \to \sigma'$.
- $(w, \sigma, \sigma') \in A_n$ for some n.
- Let n^* be the least such that $(w, \sigma, \sigma') \in A_{n^*}$ for some w, σ, σ'.
- $n^* > 0$ and $(w, \sigma, \sigma') \in A_{n^*-1}$.
- Contradiction to the minimality of n^*.

Inductive Definitions

Well-Founded Recursion (Chapter 10.4)

- B: a set
- $\prec$: a well-founded binary relation on B

Inductive Definitions

Well-Founded Recursion (Chapter 10.4)

- B: a set
- $\prec$: a well-founded binary relation on B
- for $b \in B$: $\prec^{-1}\{b\} := \{b' \in B \mid b' \prec b\}$

Inductive Definitions

Well-Founded Recursion (Chapter 10.4)

- B: a set
- $\prec$: a well-founded binary relation on B
- for $b \in B$: $\prec^{-1}\{b\} := \{b' \in B \mid b' \prec b\}$
- for $B' \subseteq B$ and $f : B \to C$: $f \restriction B' : B' \to C$ is defined by

$$f \restriction B' := \{(b, f(b)) \mid b \in B'\}$$

Inductive Definitions

Well-Founded Recursion (Chapter 10.4)

- B, C: sets
- $\prec$: a well-founded binary relation on B
- $\prec^{-1}\{b\} := \{b' \in B \mid b' \prec b\}$
- $f \restriction B' := \{(b, f(b)) \mid b \in B'\}$

Then for any function

$$F : \{(b, h) \mid b \in B, h : \prec^{-1}\{b\} \to C\} \to C$$

there exists a unique function $f : B \to C$ such that

$$\forall b \in B. f(b) = F(b, f \restriction \prec^{-1}\{b\}).$$

Inductive Definitions

The loc Function

- $\mathrm{loc}(\textbf{skip}) := \emptyset$
- $\mathrm{loc}(X := a) := \{X\}$
- $\mathrm{loc}(c_0; c_1) := \mathrm{loc}(c_0) \cup \mathrm{loc}(c_1)$
- $\mathrm{loc}(\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1) := \mathrm{loc}(c_0) \cup \mathrm{loc}(c_1)$
- $\mathrm{loc}(\textbf{while } b \textbf{ do } c) := \mathrm{loc}(c)$

Summary

- rule induction
- inductive definitions
- end of operational semantics (Chapter 2 to Chapter 4)

Questions?

Homework 3

Problem

Consider $w := \textbf{while}\ X \leq 1000\ \textbf{do}\ X := (2 \times X) + 1$. Determine the set M of all states σ such that $\exists \sigma'. \langle w, \sigma \rangle \to \sigma'$, and prove that

- $\forall \sigma \in M. \exists \sigma'. \langle w, \sigma \rangle \to \sigma'$;
- $\forall \sigma \in \Sigma \backslash M. \forall \sigma'. \langle w, \sigma \rangle \not\to \sigma'$.