Syntax und Semantik von Programmen 2

Modul 6 (v1.0)

Kanonikvorlesung: Foundations of Computing

Heiko Mantel

MAIS, TU Darmstadt, WS11/12

Motivation

Wie beweist man Aussagen über Programme formal?

- basierend auf der formal modellierten Syntax und Semantik

Unterschiedliche Beweistechniken

- Fallunterscheidung
- Widerspruchsbeweis
- Strukturelle Induktion
- Induktion über Herleitungen

Fokus dieses Moduls: Wie beweist man Eigenschaften?

Übersicht: Modul 6

Äquivalenz zweier Programme

- ❑ Beweis mit Fallunterscheidung

Nichtterminierung eines Programms

- ❑ Widerspruchsbeweis

Induktionsprinzipien

- ❑ Induktion auf den natürlichen Zahlen
- ❑ wohlfundierte Induktion
- ❑ Rechtfertigung der Induktion auf den natürlichen Zahlen
- ❑ strukturelle Induktion auf Aexp
- ❑ Rechtfertigung der strukturellen Induktion auf Aexp

Deterministische Auswertung von Ausdrücken

- ❑ Beweis mit struktureller Induktion

Äquivalenz zweier Programme (1)

Definition (aus Modul 5)

Zwei **Kommandos $c1,c2 \in Com$ sind zueinander semantisch äquivalent** wenn für alle Grundsubstitutionen η, deren Definitionsbereich alle Metavariablen in c1 und c2 einschließt, und für alle Zustände σ,σ' die folgende Bedingung gilt:

- $\langle c1\eta,\sigma\rangle \rightarrow \sigma'$ ist herleitbar genau dann wenn $\langle c2\eta,\sigma\rangle \rightarrow \sigma'$ herleitbar ist.

Notation

Wir schreiben **c1 ~ c2**, um auszudrücken, dass c1 und c2 zueinander semantisch äquivalent sind.

Äquivalenz zweier Programme (2)

Theorem

while B do C od ~ if B then C; while B do C od else skip fi

Beweis

- Seien σ und σ' beliebige Zustände und η eine beliebige Grundsubstitution, deren Definitionsbereich B und C enthält.
- Folgende beiden Teilaussagen sind zu beweisen:
  a) Wenn $\langle$(while B do C od)$\eta,\sigma\rangle\rightarrow\sigma'$ herleitbar ist, dann ist $\langle$(if B then C; while B do C od else skip fi)$\eta,\sigma\rangle\rightarrow\sigma'$ ebenfalls herleitbar.
  b) Wenn $\langle$(if B then C; while B do C od else skip fi)$\eta,\sigma\rangle\rightarrow\sigma'$ herleitbar ist, dann ist $\langle$(while B do C od)$\eta,\sigma\rangle\rightarrow\sigma'$ ebenfalls herleitbar.

Äquivalenz zweier Programme (3)

Beweis (Fortsetzung)

Beweis von Aussage a)

- Angenommen $\langle$(while B do C od)$\eta,\sigma\rangle\rightarrow\sigma'$ sei herleitbar.
- Es gibt zwei Möglichkeiten für die letzte Regel in der Herleitung:
  i. rwhf ist die letzte Regel
  ii. rwht ist die letzte Regel

Wir machen eine Fallunterscheidung über diese Möglichkeiten:

Äquivalenz zweier Programme (4)

Beweis von Fall ai.

- Da rwhf die letzte Regel in der Herleitung ist, muss diese folgende Form haben:

$$\text{rwhf}\ \frac{\begin{array}{c}\vdots\ \mathcal{H}1\\ \langle B\eta,\sigma\rangle \Downarrow \text{false}\end{array}}{\langle(\text{while B do C od})\eta,\sigma\rangle \to \sigma}$$

- Wir haben also eine Herleitung $\mathcal{H}1$ von $\langle B\eta,\sigma\rangle \Downarrow$ false und es gilt $\sigma' = \sigma$.
- Mit Hilfe der Herleitung $\mathcal{H}1$ können wir folgende Herleitung von $\langle(\text{if B then C; while B do C od else skip fi})\eta,\sigma\rangle \to \sigma'$ konstruieren:

$$\text{riff}\ \frac{\begin{array}{c}\vdots\ \mathcal{H}1\\ \langle B\eta,\sigma\rangle \Downarrow \text{false}\end{array} \qquad \text{rsk}\ \frac{}{\langle(\text{skip})\eta,\sigma\rangle \to \sigma}}{\langle(\text{if B then C; while B do C od else skip fi})\eta,\sigma\rangle \to \sigma}$$

Äquivalenz zweier Programme (5)

Beweis von Fall aii.

- Da rwht die letzte Regel in der Herleitung ist, muss diese folgende Form haben:

$$\text{rwht}\ \frac{\overset{\vdots\ \mathcal{H}1}{\langle B\eta,\sigma\rangle \Downarrow \text{true}} \quad \overset{\vdots\ \mathcal{H}2}{\langle C\eta,\sigma\rangle \to \sigma''} \quad \overset{\vdots\ \mathcal{H}3}{\langle(\text{while B do C od})\eta,\sigma''\rangle \to \sigma'}}{\langle(\text{while B do C od})\eta,\sigma\rangle \to \sigma'}$$

- Mit Hilfe von $\mathcal{H}1$, $\mathcal{H}2$ und $\mathcal{H}3$ können wir folgende Herleitung von $\langle(\text{if B then C; while B do C od else skip fi})\eta,\sigma\rangle\to\sigma'$ konstruieren:

$$\text{rift}\ \frac{\overset{\vdots\ \mathcal{H}1}{\langle B\eta,\sigma\rangle \Downarrow \text{true}} \quad \text{r;}\ \dfrac{\overset{\vdots\ \mathcal{H}2}{\langle C\eta,\sigma\rangle\to\sigma''} \quad \overset{\vdots\ \mathcal{H}3}{\langle(\text{while B do C od})\eta,\sigma''\rangle\to\sigma'}}{\langle(\text{C; while B do C od})\eta,\sigma\rangle\to\sigma'}}{\langle(\text{if B then C; while B do C od else skip fi})\eta,\sigma\rangle \to \sigma'}$$

Äquivalenz zweier Programme (6)

Beweis (Fortsetzung)

Beweis von Aussage b)

- Angenommen $\langle$(if B then C; while B do C od else skip fi)$\eta,\sigma\rangle\rightarrow\sigma'$ sei herleitbar.
- Es gibt zwei Möglichkeiten für die letzte Regel in der Herleitung:
  i. riff ist die letzte Regel
  ii. rift ist die letzte Regel

Wir machen eine Fallunterscheidung über diese Möglichkeiten:

Äquivalenz zweier Programme (7)

Beweis von Fall bi.

- Da riff die letzte Regel in der Herleitung ist, muss diese folgende Form haben:

$$\text{riff}\ \dfrac{\overset{\vdots\ \mathcal{H}1}{\langle B\eta,\sigma\rangle \Downarrow \text{false}} \qquad \overset{\vdots\ \mathcal{H}2}{\langle(\text{skip})\eta,\sigma\rangle \rightarrow \sigma'}}{\langle(\text{if B then C; while B do C od else skip fi})\eta,\sigma\rangle \rightarrow \sigma'}$$

- Die Herleitung $\mathcal{H}2$ kann nur folgende Form haben:

$$\text{rsk}\ \dfrac{}{\langle(\text{skip})\eta,\sigma\rangle \rightarrow \sigma}$$

- Mit Hilfe von $\sigma' = \sigma$ und $\mathcal{H}1$ können wir folgende Herleitung von $\langle(\text{while B do C od})\eta,\sigma\rangle \rightarrow \sigma'$ konstruieren:

$$\text{rwhf}\ \dfrac{\overset{\vdots\ \mathcal{H}1}{\langle B\eta,\sigma\rangle \Downarrow \text{false}}}{\langle(\text{while B do C od})\eta,\sigma\rangle \rightarrow \sigma}$$

Äquivalenz zweier Programme (8)

Beweis von Fall bii.

…

Übung: Vervollständigen Sie obigen Beweis!

Übersicht: Modul 6

Äquivalenz zweier Programme

- ❑ Beweis mit Fallunterscheidung

Nichtterminierung eines Programms

- ❑ Widerspruchsbeweis

Induktionsprinzipien

- ❑ Induktion auf den natürlichen Zahlen
- ❑ wohlfundierte Induktion
- ❑ Rechtfertigung der Induktion auf den natürlichen Zahlen
- ❑ strukturelle Induktion auf Aexp
- ❑ Rechtfertigung der strukturellen Induktion auf Aexp

Deterministische Auswertung von Ausdrücken

- ❑ Beweis mit struktureller Induktion

Nichtterminierung (1)

Theorem

Es gibt keine Zustände σ und σ', so dass

$$\langle \text{while true do skip od},\sigma\rangle \to \sigma' \text{ herleitbar ist.}$$

Die Intuition ist: das Programm while true do skip od terminiert nie.

Beweis (Widerspruchsbeweis)

- Angenommen es gebe σ,σ', so dass $\langle \text{while true do skip od},\sigma\rangle\to\sigma'$ herleitbar ist.
- Sei $\mathcal{H}$ eine Herleitung von $\langle \text{while true do skip od},\sigma\rangle\to\sigma'$, so dass es keine Herleitung von $\langle \text{while true do skip od},\sigma\rangle\to\sigma'$ mit weniger Regelanwendungen gibt, d.h. $\mathcal{H}$ ist eine kleinste Herleitung.
- Die letzte Regel in $\mathcal{H}$ kann nur die Regel rwht sein.
- Daher muss die Herleitung folgende Form haben:

$$\text{rwht}\ \dfrac{\text{rtrue}\ \dfrac{}{\langle \text{true},\sigma\rangle \Downarrow \text{true}} \qquad \begin{array}{c}\vdots\ \mathcal{H}1\\ \langle \text{skip},\sigma\rangle \to \sigma''\end{array} \qquad \begin{array}{c}\vdots\ \mathcal{H}2\\ \langle \text{while true do skip od},\sigma''\rangle \to \sigma'\end{array}}{\langle \text{while true do skip od},\sigma\rangle \to \sigma'}$$

Nichtterminierung (2)

Beweis (Fortsetzung)

- Die Herleitung $\mathcal{H}1$ kann nur folgende Form haben:

$$\text{rsk}\;\frac{}{\langle \text{skip},\sigma\rangle \rightarrow \sigma}$$

- Das heißt, es gilt $\sigma'' = \sigma$.
- Somit hat die Herleitung $\mathcal{H}$ folgende Form:

$$\text{rwht}\;\frac{\text{rtrue}\;\dfrac{}{\langle \text{true},\sigma\rangle \Downarrow \text{true}} \qquad \text{rsk}\;\dfrac{}{\langle \text{skip},\sigma\rangle \rightarrow \sigma} \qquad \begin{array}{c}\vdots\,\mathcal{H}2\\ \langle \text{while true do skip od},\sigma\rangle \rightarrow \sigma'\end{array}}{\langle \text{while true do skip od},\sigma\rangle \rightarrow \sigma'}$$

- $\mathcal{H}2$ ist auch eine Herleitung von $\langle \text{while true do skip od},\sigma\rangle \rightarrow \sigma'$, hat aber weniger Regelanwendungen als $\mathcal{H}$.
- $\mathcal{H}$ ist also keine kleinste Herleitung von $\langle \text{while true do skip od},\sigma\rangle \rightarrow \sigma'$.

Ein Widerspruch!

qed

Übersicht: Modul 6

Äquivalenz zweier Programme

- Beweis mit Fallunterscheidung

Nichtterminierung eines Programms

- Widerspruchsbeweis

Induktionsprinzipien

- Induktion auf den natürlichen Zahlen
- wohlfundierte Induktion
- Rechtfertigung der Induktion auf den natürlichen Zahlen
- strukturelle Induktion auf Aexp
- Rechtfertigung der strukturellen Induktion auf Aexp

Deterministische Auswertung von Ausdrücken

- Beweis mit struktureller Induktion

Induktion

Beweisprinzip der Induktion auf den natürlichen Zahlen

Sei $P \subseteq \mathbb{N}$ eine einstellige Relation über den natürlichen Zahlen.
Wenn folgende Bedingungen gelten:

- $P(0)$
- $\forall n \in \mathbb{N}: \ (P(n) \Rightarrow P(n+1))$

dann gilt auch

- $\forall n \in \mathbb{N}: P(n)$

Die Formel $P(n)$ wird als **Induktionsannahme** bezeichnet, und P wird als **Induktionsformel** bezeichnet.

Intuition

Wenn die „Eigenschaft P“ für 0 gilt und wenn aus „P gilt für n“ auch „P gilt für n+1“ gefolgert werden kann, dann haben alle natürlichen Zahlen die „Eigenschaft P“.

Wohlfundierte Induktion (1)

Definition

Sei $\prec \subseteq D \times D$ eine binäre Relation auf einer Menge D. Eine unendliche Folge $f: \mathbb{N} \rightarrow D$ (wurde in Modul 3 eingeführt) heißt **unendlich absteigende Kette für $\prec$** genau dann wenn $\forall i \in \mathbb{N}: f(i+1) \prec f(i)$ gilt.

Definition

Eine binäre Relation $\prec \subseteq D \times D$ auf einer Menge D heißt **wohlfundiert** genau dann wenn es keine unendliche absteigende Kette für $\prec$ gibt.

Wohlfundierte Induktion (2)

Definition

Die **transitive Hülle** einer binären Relation $R \subseteq D \times D$ auf einer Menge D ist die kleinste Relation $R^* \subseteq D \times D$, so dass

- $\forall d1,d2 \in D$: [d1 R d2 $\Rightarrow$ d1 R* d2]
- $\forall d1,d2,d3 \in D$: [(d1 R* d2 $\wedge$ d2 R* d3) $\Rightarrow$ d1 R* d3]

Theorem

Sei $\prec \subseteq D \times D$ eine wohlfundierte Relation auf D. Dann gilt:

- $\prec$ ist irreflexiv, d.h. $\nexists d \in D$: $d \prec d$ und
- $\prec^*$ ist eine wohlfundierte Relation.

Beweis

….

Übung: Führen Sie obigen Beweis!

Wohlfundierte Induktion (3)

Beweisprinzip der wohlfundierten Induktion

Sei $P \subseteq D$ eine einstellige Relation auf einer Menge D und $\prec \subseteq D \times D$ eine wohlfundierte Relation auf D. Wenn folgende Bedingung gilt:

- $\forall d \in D$: $[(\forall d' \in D$: $(d' \prec d \Rightarrow P(d'))) \Rightarrow P(d)]$

dann gilt auch

- $\forall d \in D$: $P(d)$

Die Formeln $P(d')$ werden als **Induktionsannahme** bezeichnet, und P wird als **Induktionsformel** bezeichnet.

Intuition

Wenn aus „P gilt für alle d´, die kleiner sind als d“ auch „P gilt für d“ gefolgert werden kann, dann haben alle Elemente von D die „Eigenschaft P“.

Instanziierung des Beweisprinzips (1)

Beispiel (Rechtfertigung der Induktion auf $\mathbb{N}$)

Die Relation $\prec \subseteq \mathbb{N}\times\mathbb{N}$ sei definiert durch

- $m \prec n$ genau dann wenn $m+1=n$

Wir erhalten folgende Spezialisierung des Beweisprinzips der wohlfundierten Induktion:

- „Wenn folgende Bedingung gilt:
  - $\forall n\in\mathbb{N}$: $[\,(\forall n'\in\mathbb{N}:\ (n'+1=n \Rightarrow P(n'))) \Rightarrow P(n)\,]$

  dann gilt auch
  - $\forall n\in\mathbb{N}$: $P(n)$“

Durch ein Fallunterscheidung über „n=0“ erhalten wir

- „Wenn folgende Bedingungen gelten:
  - $(\forall n'\in\mathbb{N}:\ (n'+1=0 \Rightarrow P(n'))\,) \Rightarrow P(0)$
  - $\forall n\in\mathbb{N}$: $[\,n\neq 0 \Rightarrow [\,(\forall n'\in\mathbb{N}:\ (n'+1=n \Rightarrow P(n'))) \Rightarrow P(n)\,]\,]$

  dann gilt auch
  - $\forall n\in\mathbb{N}$: $P(n)$“

Instanziierung des Beweisprinzips (2)

Beispiel (Fortsetzung)

Also gilt auch

- „Wenn folgende Bedingungen gelten:
  - true $\Rightarrow$ P(0)
  - $\forall n'' \in \mathbb{N}$: $[n''+1 \neq 0 \Rightarrow [(\forall n' \in \mathbb{N}:(n'+1=n''+1 \Rightarrow P(n'))) \Rightarrow P(n''+1)]]$

  dann gilt auch
  - $\forall n \in \mathbb{N}$: P(n)“

Also gilt auch

- „Wenn folgende Bedingung gilt:
  - P(0)
  - $\forall n'' \in \mathbb{N}$: $[\, P(n'') \Rightarrow P(n''+1)\,]$

  dann gilt auch
  - $\forall n \in \mathbb{N}$: P(n)“

Das Induktionsprinzip auf den natürlichen Zahlen ist also ein Spezialfall der wohlfundierten Induktion.

Strukturelle Induktion für Aexp (1)

Definition (aus Modul 5)

Die Menge Aexp ist durch folgende BNF definiert:

- a ::= n | X | a+a | a-a | a*a

Beweisprinzip der strukturellen Induktion für Aexp

Sei $P \subseteq Aexp$ eine einstellige Relation über Aexp. Wenn folgende Bedingungen gelten:

- $\forall n \in N: P(n)$
- $\forall x \in Loc: P(x)$
- $\forall a1,a2 \in Aexp: (P(a1) \wedge P(a2)) \Rightarrow P(a1+a2)$
- $\forall a1,a2 \in Aexp: (P(a1) \wedge P(a2)) \Rightarrow P(a1-a2)$
- $\forall a1,a2 \in Aexp: (P(a1) \wedge P(a2)) \Rightarrow P(a1*a2)$

dann gilt auch

- $\forall a \in Aexp: P(a)$

Die Formeln P(a1) und P(a2) werden als **Induktionsannahmen** bezeichnet, und P wird als **Induktionsformel** bezeichnet.

Strukturelle Induktion für Aexp (2)

Definition

Die Ausdrücke a1 und a2 sind die **direkten Teilausdrücke** der Ausdrücke a1+a2, a1-a2 und a1*a2. Die Ausdrücke n und x haben keine direkten Teilausdrücke.

Rechtfertigung der strukturellen Induktion für Aexp

Die Relation $\prec \subseteq$ Aexp$\times$Aexp sei definiert durch

- a´ $\prec$ a genau dann wenn a´ ein direkter Teilausdruck von a ist.

Wir erhalten folgende Spezialisierung des Beweisprinzips der wohlfundierten Induktion:

- „Wenn folgende Bedingung gilt:
  - $\forall a \in$ Aexp: [($\forall a' \in$ Aexp:
    (a´ ist ein direkter Teilausdruck von a $\Rightarrow$ P(a´)))
    $\Rightarrow$ P(a)]

  dann gilt auch
  - $\forall a \in$ Aexp: P(a)“

Strukturelle Induktion für Aexp (3)

Rechtfertigung (Fortsetzung)

Durch ein Fallunterscheidung über die Struktur der Ausdrücke in Aexp erhalten wir

- „Wenn folgende Bedingung gilt:
  - $\forall n \in N$: P(n)
  - $\forall x \in Loc$: P(x)
  - $\forall a1,a2 \in Aexp$: [(P(a1) $\wedge$ P(a2)) $\Rightarrow$ P(a1+a2)]
  - $\forall a1,a2 \in Aexp$: [(P(a1) $\wedge$ P(a2)) $\Rightarrow$ P(a1-a2)]
  - $\forall a1,a2 \in Aexp$: [(P(a1) $\wedge$ P(a2)) $\Rightarrow$ P(a1*a2)]

  dann gilt auch
  - $\forall a \in Aexp$: P(a)“

Das Beweisprinzip der strukturellen Induktion auf Aexp ist also ein Spezialfall der wohlfundierten Induktion.

Übersicht: Modul 6

Äquivalenz zweier Programme

- Beweis mit Fallunterscheidung

Nichtterminierung eines Programms

- Widerspruchsbeweis

Induktionsprinzipien

- Induktion auf den natürlichen Zahlen
- wohlfundierte Induktion
- Rechtfertigung der Induktion auf den natürlichen Zahlen
- strukturelle Induktion auf Aexp
- Rechtfertigung der strukturellen Induktion auf Aexp

Deterministische Auswertung von Ausdrücken

- Beweis mit struktureller Induktion

Deterministische Auswertung (1)

Theorem

Für alle $a \in Aexp$, $m, m' \in N$ und alle Zustände σ gilt:
wenn $\langle a,\sigma\rangle \Downarrow m$ und $\langle a,\sigma\rangle \Downarrow m'$ herleitbar sind, dann gilt $m' = m$.

Beweis

- Wir verwenden das Beweisprinzip der strukturellen Induktion für $P(a) = \forall\sigma{:}\ \forall m,m' \in N{:}\ [(„\langle a,\sigma\rangle \Downarrow m$ ist herleitbar“ $\wedge$ „$\langle a,\sigma\rangle \Downarrow m'$ ist herleitbar“) $\Rightarrow m = m']$
- Wir müssen 5 Fälle beweisen:

Fall $a = n$ für ein $n \in N$

- Die Herleitung von $\langle n,\sigma\rangle \Downarrow n'$ kann nur folgende Form haben:

$$\text{rN}\ \frac{}{\langle n,\sigma\rangle \Downarrow n}$$

Daher muss $m = n = m'$ gelten.

Deterministische Auswertung (2)

Beweis (Fortsetzung)

Fall a=x für ein x∈Loc

- Die Herleitung von $\langle x,\sigma\rangle \Downarrow n'$ kann nur folgende Form haben:

$$\text{rLoc}\quad \frac{}{\langle x,\sigma\rangle \Downarrow n'}\ \sigma(x)=n'$$

Daher muss $m = \sigma(x) = m'$ gelten.

Fall a=a1+a2 für a1,a2∈Aexp wobei P(a1) und P(a2) gelten

- Die Herleitungen von $\langle a,\sigma\rangle \Downarrow m$ und $\langle a,\sigma\rangle \Downarrow m'$ können nur folgende Formen haben:

$$\text{r+}\quad \frac{\begin{array}{c}\vdots\ \mathcal{H}1\\ \langle a1,\sigma\rangle \Downarrow m1\end{array}\quad \begin{array}{c}\vdots\ \mathcal{H}2\\ \langle a2,\sigma\rangle \Downarrow m2\end{array}}{\langle a1+a2,\sigma\rangle \Downarrow m}\ m=m1+m2$$

$$\text{r+}\quad \frac{\begin{array}{c}\vdots\ \mathcal{H}1'\\ \langle a1,\sigma\rangle \Downarrow m1'\end{array}\quad \begin{array}{c}\vdots\ \mathcal{H}2'\\ \langle a2,\sigma\rangle \Downarrow m2'\end{array}}{\langle a1+a2,\sigma\rangle \Downarrow m'}\ m'=m1'+m2'$$

- Aus P(a1) und P(a2) folgen $m1=m1'$ und $m2=m2'$.

Daher muss $m = m1+m2 = m1'+m2' = m'$ gelten.

Deterministische Auswertung (3)

Beweis (Fortsetzung)

Fall a=a1-a2 für a1,a2∈Aexp wobei P(a1) und P(a2) gelten

- …

Fall a=a1*a2 für a1,a2∈Aexp wobei P(a1) und P(a2) gelten

- …

Siehe Übungsblatt und Musterlösung

Deterministische Auswertung (4)

Theorem

Für alle $b \in Bexp$, $t,t' \in T$ und alle Zustände σ gilt:
wenn $\langle b,\sigma\rangle \Downarrow t$ und $\langle b,\sigma\rangle \Downarrow t'$ herleitbar sind, dann gilt $t'=t$.

Beweis

- …

Siehe Übungsblatt und Musterlösung

Übersicht: Modul 6

Äquivalenz zweier Programme

- Beweis mit Fallunterscheidung

Nichtterminierung eines Programms

- Widerspruchsbeweis

Induktionsprinzipien

- Induktion auf den natürlichen Zahlen
- wohlfundierte Induktion
- Rechtfertigung der Induktion auf den natürlichen Zahlen
- strukturelle Induktion auf Aexp
- Rechtfertigung der strukturellen Induktion auf Aexp

Deterministische Auswertung von Ausdrücken

- Beweis mit struktureller Induktion

Rückblick

Einige wesentliche Lernziele dieses Moduls

- Wie kann ich eine operationelle Semantik zur Verifikation nutzen?
- Beherrschung elementarer Verifikationstechniken:
  - Fallunterscheidung
  - Widerspruchsbeweis
  - wohlfundierte Induktion
  - strukturelle Induktion
  - … wird in Modul 7 vervollständigt …
- Fähigkeit, ein Induktionsprinzip zu rechtfertigen

Literatur

Glynn Winskel

The Formal Semantics of Programming Languages; Kapitel 3
The MIT Press, 1993.