Zur Wahl der Lemmata

Beachte

Die Wahl der „richtigen“ Lemmata ist zum Beweis der Gültigkeit von e in $\mathcal{I}(\text{spec})$ von zentraler Bedeutung.

Frage

Wie findet man sie?

Antwort

1. Das ist i. Allg. sehr schwer!
2. Häufig helfen Heuristiken weiter:
   - Wähle das, was zum Beweis gerade benötigt wird, als Hypothese.
   - Beweise die Hypothese, dann ist sie ein Lemma.

3.3 Induktionsbeweise durch Konsistenzprüfung

Bezeichnung

Dieses Verfahren läuft in der Literatur auch unter den Namen „Induktionslose Induktion“ und „Proof by Consistency“.

Vorteil des Verfahrens

Man benötigt kein Induktionsschema.

Nachteil des Verfahrens

Es hält sehr häufig nicht.

Beschränkung hier

Sowohl die Spezifikation als auch die Beweisziele sind unbedingt.

Konsistenz

Sei spec $=$ (sig, E)
sig $= (S, F, \alpha)$
$E \subseteq \mathsf{Term}(F,V) \times \mathsf{Term}(F,V)$
$G \subseteq \mathsf{Term}(F,V) \times \mathsf{Term}(F,V)$

Ziel

Zeige $G \subseteq ITh(\text{spec})$.

Definition 3.6

G heißt **konsistent** (in spec), falls gilt

$=_E \; = \; =_{E \cup G}$ auf $\mathsf{Term}(F)$.

Beachte

Es gilt stets $=_E \subseteq =_{E \cup G}$.

Also: G ist konsistent, falls für alle $s, t \in \mathsf{Term}(F)$ gilt:

Aus $s =_{E \cup G} t$ folgt $s =_E t$.

Bedeutung der Konsistenz

Lemma 3.2

Es gilt

$G \subseteq ITh(\text{spec})$ gdw G ist konsistent in spec.

Beweis:

„$\curvearrowleft$“: Sei $u = v$ in G, $\sigma \in \mathsf{GSub}(u = v)$.

Es gilt $\sigma(u) =_G \sigma(v)$, also $\sigma(u) =_{E \cup G} \sigma(v)$, also $\sigma(u) =_E \sigma(v)$, da G konsistent ist.

Also $u = v \in ITh(\text{spec})$.

Also $G \subseteq ITh(\text{spec})$.

„$\curvearrowright$“: Sei $s, t \in \mathsf{Term}(F)$ und $s =_{E \cup G} t$.

Jeder G-Schritt $t_1 \vdash\!\dashv_G t_2$ im Beweis zu $s =_{E \cup G} t$ kann durch einen E-Beweis $t_1 =_E t_2$ ersetzt werden, wegen $G \subseteq ITh(\text{spec})$.

Also gilt $s =_E t$. Also ist G konsistent.

Test auf Konsistenz (1)

Sei spec $=$ (sig, E)
sig $= (S, F, \alpha)$, G wie immer
R Regelsystem über sig

Bezeichnung

R heißt **Regelsystem zu** E, falls $=_R \; = \; =_E$ gilt.

$\mathsf{GIrr}(R) = \{t \in \mathsf{Term}(F) \mid t$ ist R-irreduzibel$\}$

Lemma 3.3

Seien R und R' grundkonvergente Regelsysteme zu E bzw. $E \cup G$.

Gilt $\mathsf{GIrr}(R) = \mathsf{GIrr}(R')$, so ist G konsistent.

Beweis: Sei $s, t \in \mathsf{Term}(F)$, $t_R \equiv t\downarrow_R$, $t_{R'} \equiv t\downarrow_{R'}$.

a) Es gilt stets $t_R \equiv t_{R'}$:
Es gilt $t_R =_E t =_{E\cup G} t_{R'}$, also $t_R =_{R'} t_{R'}$, also $t_R \downarrow_{R'} t_{R'}$.
Also gilt $t_R \equiv t_{R'}$ wegen $\mathsf{GIrr}(R) = \mathsf{GIrr}(R')$.

b) Aus $s =_{E\cup G} t$ folgt $s =_E t$:
Aus $s =_{E\cup G} t$ folgt $s_{R'} \equiv t_{R'}$. Nach a) gilt $s_R \equiv t_R$, also $s =_R t$, also $s =_E t$.

c) G ist konsistent nach b).

Grundreduzierbarkeit

Sei

R Regelsystem über sig $= (S, F, \alpha)$.

Definition 3.7

$t \in \mathsf{Term}(F, V)$ heißt **grundreduzierbar** in R, falls für jede Grundinstanz $\sigma(t)$ (d. h. jedes $\sigma \in \mathsf{GSub}(t)$) der Term $\sigma(t)$ R-reduzierbar ist.

Beispiel

R: $x + 0 \to x$ $\quad$ $\mathsf{app}(\mathsf{nil}, l_2) \to l_2$
$x + s(y) \to s(x+y)$ $\quad$ $\mathsf{app}(x.l_1, l_2) \to x.\mathsf{app}(l_1, l_2)$

Es sind $t_1 \equiv x + y$ und $t_2 \equiv \mathsf{app}(l_1, l_2)$ irreduzibel. Sie sind aber grundreduzierbar.

Beachte

Ist t reduzierbar, so ist t auch grundreduzierbar.

Bemerkung

Wir benötigen den Begriff „grundreduzierbar“ für den Test, ob $\mathsf{GIrr}(R) = \mathsf{GIrr}(R')$ aus Lemma 3.3 gilt.

Ein Lemma zur Grundreduzierbarkeit

Lemma 3.4

Sei R terminierend und C_R die durch R definierte Konstruktormenge. Enthält $t \in \mathsf{Term}(F, V)$ ein $f \in F - C_R$, so ist t grundreduzierbar.

Beweis: Übung.

Beispiel

Sei R das Regelsystem für $+$, app wie oben und

$F = \{0, s, +, ., \mathsf{nil}, \mathsf{app}\}$.

Es ist $C_R = \{0, s, \mathsf{nil}, .\}$.

Also sind $t_1 \equiv x + y$ und $t_2 \equiv \mathsf{app}(l_1, l_2)$ grundreduzierbar.

Komplexität der Grundreduzierbarkeit

Problem

Eingabe: R über sig $= (S, F, \alpha)$ und $t \in \mathsf{Term}(F, V)$.
Frage: Ist t in R grundreduzierbar?

Resultate

1. Für unbedingte R ist das Problem entscheidbar. Es ist schwer im allgemeinen Fall: Zeit $O(2^{2^n})$, falls $|t| = n$. Es ist viel leichter, falls R links-linear ist.
2. Für bedingte R ist das Problem unentscheidbar.

Beachte

a) Lemma 3.4 liefert in der Praxis häufig ein hinreichend starkes Kriterium für die Grundreduzierbarkeit und für den Test, ob $\mathsf{GIrr}(R_1) = \mathsf{GIrr}(R_2)$ gilt.

b) Wir geben noch ein Kriterium an, das noch etwas schärfer ist.

Test auf $\mathbf{GIrr}(R_1) = \mathbf{GIrr}(R_2)$

Seien

R_1, R_2 Regelsysteme über sig $= (S, F, \alpha)$.

Lemma 3.5

Für jede Regel $l \longrightarrow r$ in R_2 sei l in R_1 grundreduzierbar.

Dann gilt GIrr$(R_1) \subseteq$ GIrr(R_2).

Gilt zusätzlich $R_1 \subseteq R_2$, so gilt GIrr$(R_1) =$ GIrr(R_2).

Beweis:

Sei $t \in$ GIrr(R_1). Annahme: $t \notin$ GIrr(R_2).

t ist mit $l \longrightarrow r$ aus R_2 reduzierbar.

Es gibt σ mit $t >_{TT} \sigma(l)$.

t ist in R_1 reduzierbar, da $\sigma(l)$ in R_1 reduzierbar. $\lightning$

Sei zusätzlich $R_1 \subseteq R_2$.

Dann gilt GIrr$(R_2) \subseteq$ GIrr(R). Also gilt GIrr$(R_1) =$ GIrr(R_2).

Das Hauptergebnis

Sei spec $=$ (sig, R)
sig $= (S, F, \alpha)$
G Menge von sig-Gleichungen

Satz 3.2.

Seien R und R' grundkonvergent, R' ein Regelsystem zu $R \cup G$ und $R \subseteq R'$. Für jedes $l \longrightarrow r$ in $R' - R$ sei l in R grundreduzierbar.

Dann gilt $G \subseteq ITh$(spec).

Beweis: Dies folgt aus den letzten Lemmata.

Beachte

Es reicht, dass R und R' terminierend und grundkonfluent sind, man benötigt nicht die Konfluenz.

Literatur

Jouannaud + Kounalis, LICS 1986

Beispiel: Listen (1)

Sei

$F = \{\mathsf{nil}, ., \mathsf{app}, \mathsf{rev}\},\ C = \{\mathsf{nil}, .\}$.

R: $\mathsf{app}(\mathsf{nil}, l_2) \longrightarrow l_2$
$\mathsf{app}(x.l_1, l_2) \longrightarrow x.\mathsf{app}(l_1, l_2)$
$\mathsf{rev}(\mathsf{nil}) \longrightarrow \mathsf{nil}$
$\mathsf{rev}(x.l) \longrightarrow \mathsf{app}(\mathsf{rev}(l), x.\mathsf{nil})$

G: $\mathsf{app}(\mathsf{app}(l_1, l_2), l_3) = \mathsf{app}(l_1, \mathsf{app}(l_2, l_3))$ (1)
$\mathsf{rev}(\mathsf{rev}(l)) = l$ (2)
$\mathsf{rev}(\mathsf{app}(l_1, l_2)) = \mathsf{app}(\mathsf{rev}(l_2), \mathsf{rev}(l_1))$ (3)

Es ist $C = C_R$ eine Konstruktormenge.

Vervollständigung von R und (2) liefert $R \cup R_1$ mit

R_1: $\mathsf{rev}(\mathsf{rev}(l)) \longrightarrow l$
$\mathsf{rev}(\mathsf{app}(l, x.\mathsf{nil})) \longrightarrow x.\mathsf{rev}(l)$

Also

$R_1 \subseteq ITh(\mathsf{spec})$.

Beispiel: Listen (2)

Vervollständigung von R und (1) liefert $R \cup R_2$ mit

$R_2 : \mathsf{app}(\mathsf{app}(l_1, l_2), l_3) \longrightarrow \mathsf{app}(l_1, \mathsf{app}(l_2, l_3))$

Also

(1) $\in ITh(\mathsf{spec})$.

Vervollständigung von $R \cup \{(1), (2)\}$ und von $R \cup \{(1), (3)\}$ divergiert.

Also

Vorsicht bei simultanen Beweisen mehrerer Hypothesen.

Vervollständigung von $R \cup G$ terminiert und liefert

$R = R_1 \cup R_2 \cup R_3$ mit

$R_3 : \mathsf{rev}(\mathsf{app}(l_1, l_2)) \longrightarrow \mathsf{app}(\mathsf{rev}(l_2), \mathsf{rev}(l_1))$.

Also

$G \subseteq ITh(\mathsf{spec})$.

Simultanes Beweisen mehrerer Hypothesen kann hilfreich sein.

Beispiel: Ganze Zahlen

Besonderheit

Nicht-freie Konstruktoren.

Sei

$F = \{0, s, p, +, -\},\ C = \{0, s, p\}$.

$$
\begin{array}{lll}
R: & x + 0 \to x & x - 0 \to x \\
 & x + s(y) \to s(x + y) & x - s(y) \to p(x - y) \\
 & s + p(y) \to p(x + y) & x - p(y) \to s(x - y) \\
 & s(p(x)) \to x & p(s(x)) \to x
\end{array}
$$

Es ist $C = C_R$ eine Konstruktormenge, aber nicht frei.

$$
\begin{array}{lll}
G: & s(x) + y = s(x + y) & s(x) - y = s(x - y) \\
 & p(x) + y = p(x + y) & p(x) - y = p(x - y) \\
 & 0 + y = y & x - x = 0 \\
 & (x + y) - y = x & (x - y) + y = x
\end{array}
$$

Vervollständigung von $R \cup G$ liefert $R' = R \cup R_1$ mit $R_1 = \vec{G}$.

Also

$G \subseteq ITh(\text{spec})$.

Beispiel: Natürliche Zahlen

Besonderheit: AC-Symbole.

Sei

$F = \{0, s, +\},\ C = \{0, s\}$.

$$
\begin{array}{lll}
R: & x + 0 \to x & x + s(y) \to s(x + y) \\
 & & \\
G: & 0 + x = x & s(x) + y = s(x + y) \\
 & x + y = y + x & (x + y) + z = x + (y + z)
\end{array}
$$

Es ist $C = C_R$ eine Konstruktormenge.

Vervollständigung von $R \cup G$ liefert $R' = R \cup R_1 \cup E^{\succ}$ mit

$$
\begin{array}{lll}
R_1: & 0 + y \to y & s(x) + y \to s(x + y) \\
 & & (x + y) + z \to x + (y + z) \\
 & & \\
E: & x + y = y + x & x + (y + z) = y + (x + z)
\end{array}
$$

$E^{\succ} = \{\sigma(u) \to \sigma(v) \mid u \doteq v \text{ in } E, \sigma(u) \succ \sigma(v)\}$

$\succ$ die LPO zu $+ \rhd s \rhd 0$.

Die linken Seiten der Regeln in $R_1 \cup E^{\succ}$ sind in R grundreduzierbar. R und R' sind grundkonvergent.

Also

$G \subseteq ITh(\text{spec})$.

Ein sparsames Verfahren: Motivation

Sei spec $=$ (sig, R) sig $= (S, F, \alpha)$
R unbedingtes Regelsystem
R grundkonfluent
$R \subseteq \succ$, $\succ$ eine Reduktionsordnung, total auf Term(F)
$G \subseteq$ Term$(F, V) \times$ Term(F, V)

Aufgabe: Überprüfe, ob $G \subseteq ITh$(spec) gilt.

Ziel

a) Die Vervollständigung von $R \cup G$ vermeiden: Sie kann divergieren oder gar abbrechen.
b) Weniger Inferenzschritte machen.

Ansatz

Die möglichen Inkonsistenzen verkleinern bis

i) es klar ist, dass keine Inkonsistenz existiert oder
ii) eine Inkonsistenz offensichtlich wird.

Literatur

Bachmair, LICS 1988

Nachweisbare Inkonsistenzen

Erinnerung

$s = t$ heißt konsistent (mit R), falls für alle $\sigma \in$ GSub$(s = t)$ gilt $\sigma(s) =_R \sigma(t)$, also auch $\sigma(s) \downarrow_R \sigma(t)$.

Definition 3.8

Die Gleichung $s = t$ heißt **nachweisbar inkonsistent** mit R, falls i), ii) oder iii) gilt

i) $s \succ t$ und s ist in R nicht grundreduzierbar.
ii) $t \succ s$ und t ist in R nicht grundreduzierbar.
iii) s, t sind $\succ$-unvergleichbar und $\#(s, t)$ ist nicht grundreduzierbar in $R \cup \{\#(x, x) \to x\}$. Dabei ist $\# \notin F$ neues Funktionssymbol.

G heißt **nachweisbar inkonsistent**, falls ein $s = t$ aus G nachweisbar inkonsistent ist.

Grundlagen des Verfahrens (1)

Idee Verbinde Suche nach Lösungen (mit Narrowing) und Suche nach Nicht-Lösungen (also Inkonsistenzen)

Ausführung Sei H Menge von Hypothesen

- Saturieren von H mit Narrowing:
  $s = t \in H$ simplifizieren.
  $s = t$ löschen, falls $s = t$ „offensichtlich konsistent", sonst ersetzen durch $\mathrm{CP}(R, s = t)$.
  (Dies verkleinert Inkonsistenzen!)
- Stop, falls
  - $H = \emptyset \leadsto$ Antwort: JA
  - $s = t \in H$ ist „offensichtlich inkonsistent" $\leadsto$ Antwort: NEIN

Ergebnis Verfahren mit Start $H = \{s = t\}$ liefert NEIN gdw $s = t \notin ITh(\text{spec})$

ohne Beweis

Grundlagen des Verfahrens (2)

Fragen

a) Wann ist $s = t$ „offensichtlich konsistent"?

b) Wann ist $s = t$ „offensichtlich inkonsistent"?

Beachte (Intuition)

a) $s = t$ ist „offensichtlich konsistent" $\leadsto$ $s = t$ ist konsistent
$\leadsto$ $s = t$ kann aus H gestrichen werden

b) $s = t$ ist „offensichtlich inkonsistent"
$\leadsto$ $s = t$ ist inkonsistent $\leadsto$ Es gilt nicht $H \subseteq ITh(\text{spec})$

Antworten

zu a) Es gilt $s =_{I(E)} t$ (z.B. $s =_E t$)

zu b) viele Kriterien möglich, z.B.

b1) $s = t$ ist nachweisbar inkonsistent (Test ist aber schwer)

b2) $\mathrm{top}(s) = f \neq g = \mathrm{top}(t)$, $f, g \in C$ und C ist Menge von freien Konstruktoren

4 Positiv/negativ bedingte Spezifikationen

4.1 Motivation

Bisher (z. B. bei dem Cover-Set-Ansatz)

spec ist eine Spezifikation mit Konstruktoren in C, so dass

a) Jedes $f \in F - C$ ist total definiert.

b) Jede definierende Gleichung ist nur positiv bedingt.

Jetzt

Aufgabe der Voraussetzungen a) und b). Als Beweisziele sind beliebige (all-quantifizierten) Klauseln zugelassen.

Beachte

Zu a): Partialität ist schwer zu modellieren. Es gibt dazu verschiedene Ansätze. Unser Ansatz ist mit „totalen Algebren $\mathcal{A}$ mit Fehlerelementen". Die $f \in F - C$ werden zu partiellen Funktionen auf dem C-Redukt $\mathcal{A}^C$ von $\mathcal{A}$ interpretiert.

Zu b): Eine definierende Gleichung ist im Sinn der Logik eine Klausel mit ausgezeichnetem Kopf-Atom: $u = v \wedge u' \neq v' \Rightarrow s = t$. Wir werten sie so aus, dass wir „im wesentlichen" wieder in der Horn-Logik sind. Es gibt dann in den „Datenmodellen" initiale Modelle von spec.

Beispiel zur Partialität (1)

Sei $F = \{0, s, +, -\}$, $C = \{0, s\}$ und

$$E:\quad \begin{array}{ll} x + 0 = x & x - 0 = x \\ x + s(y) = s(x + y) & s(x) - s(y) = x - y \end{array}$$

Es ist $0 - s^{i+1}(0)$ nicht E-gleich zu einem C-Term, also ist $0 - s^{i+1}(0)$ ein Fehlerelement. Allgemein ist ein Fehlerelement ein $t \in \text{Term}(F)$, der ein $0 - s^{i+1}(0)$ als Teilterm enthält. E-gleiche Grundterme werden identifiziert:

Sei $[t]$ die E-Äquivalenzklasse von $t \in \text{Term}(F)$ und $\mathcal{A} = (A, (f^{\mathcal{A}})_{f \in F})$ mit $A = \{[t] \mid t \in \text{Term}(F)\}$ und

$$\begin{array}{ll} 0^{\mathcal{A}} = [0] & [t_1] +^{\mathcal{A}} [t_2] = [t_1 + t_2] \\ s^{\mathcal{A}}(t) = [s(t)] & [t_1] -^{\mathcal{A}} [t_2] = [t_1 - t_2] \end{array}$$

Dann ist das erweiterte C-Redukt von $\mathcal{A}$ die partielle Algebra

$$\mathcal{A}_e^C = (A^C, (f^{\mathcal{A}_e^C})_{f \in F}) \qquad A^C = \{[t] \mid t \in \text{Term}(C)\}$$

Es ist z. B.

$$[s^i(0)] +^{\mathcal{A}^C_e} [s^j(0)] = [s^{i+j}(0)]$$

$$[s^i(0)] -^{\mathcal{A}^C_e} [s^j(0)] = \begin{cases} [s^{i-j}(0)] & i \geq j \\ \text{nicht definiert} & \text{sonst} \end{cases}$$

Offenbar ist $\mathcal{A}_e^C$ isomorph zu $\mathbb{N}$ mit der partiellen Differenz als $-^{\mathbb{N}}$.