Formale Grundlagen der Informatik 3

Prof. Dr. Christoph Walther / Visar Januzaj, Nathan Wasser
Technische Universität Darmstadt — Wintersemester 2011/12

Übung 7

Version 1 vom 07.02.2011

Vorbemerkung

In dieser Übung werden Aufgaben zu den verschiedensten Bereichen der Vorlesung gestellt. Aus diesem Grund ist diese Übung sehr umfangreich und kann nicht komplett in der Präsenzübung behandelt werden. Trotzdem ist es sinnvoll sich auch nach der Präsenzübung mit den Aufgaben weiter zu beschäftigen, da diese als gute Vorbereitung auf die Klausur dienen.

Aufgabe 7.1 (Quer durch Kapitel 11 und 12)

Hinweis: Es ist immer nur eine Antwortmöglichkeit richtig.

(a) Eine $\mathcal{S}$-Signatur ist sensibel gdw. ...

- ☐ ... die Menge der Variablensymbol der Sorte s nicht leer ist für alle $s \in \mathcal{S}$.
- ☐ ... die Menge der Grundterme der Sorte s nicht leer ist für alle $s \in \mathcal{S}$.
- ☐ ... die Menge der Variablensymbol der Sorte s leer ist für alle $s \in \mathcal{S}$.

(b) Mit Σ-Algebren ...

- ☐ ... können Lemmata bewiesen oder widerlegt werden.
- ☐ ... können Stuck-Terme interpretiert werden.
- ☐ ... bekommen die Worte aus $\mathcal{T}(\Sigma)$ eine Bedeutung.

(c) Eine Σ-Interpretation besteht aus ...

- ☐ ... einer Σ-Algebra sowie einer A-Variablenbelegung.
- ☐ ... einer $\mathcal{S}$-Signatur sowie einer A-Variablenbelegung.
- ☐ ... einer Σ-Algebra sowie einer $\mathcal{S}$-Signatur

(d) Die Trägermenge der Standardalgebra $\mathcal{M}_{\mathcal{P}}$ eines terminierenden $\mathcal{L}$-Programms $\mathcal{P}$ ist ...

- ☐ ... die Menge der Sortensymbole $\mathcal{S}$.
- ☐ ... die Menge der Konstruktorgrundterme $\mathcal{C}(P)$.
- ☐ ... die Menge der Axiome $AX_{\mathcal{P}}$.

(e) Die Sprache $\mathcal{L}^-$ ist definiert wie $\mathcal{L}$ jedoch ...

- ☐ ... ohne Stuck-Terme.
- ☐ ... ohne partiell definierte Prozeduren.
- ☐ ... ohne polymorphe Datentypen.

Aufgabe 7.2 (Substitutionslemma)

Seien A eine Σ-Algebra, $\mathfrak{a}$ eine A-Variablenbelegung, $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ eine Substitution und $t \in \mathcal{T}(\Sigma, \mathcal{V})$. Zeigen Sie, dass dann gilt:

$$\mathfrak{a}(\sigma(t)) = \mathfrak{a}\big[x_1/\mathfrak{a}(t_1), \ldots, x_n/\mathfrak{a}(t_n)\big](t)$$

Aufgabe 7.3 (Theorie einer Σ-Algebra)

Sei $\mathcal{S} := \{s\}, \Sigma_s := \{c\}, \Sigma_{s,s} := \{f, g\}, \Sigma_{s,s,s} := \{h\}$ und $A = (\mathcal{A}, \alpha)$ definiert durch
$\mathcal{A}_s = \mathbb{Z}$
$\alpha_c := 0, \quad \alpha_f(n) := n + 2, \quad \alpha_g(n) := n - 1, \quad \alpha_h(n, m) := 2n + 2m + 1$

Zeigen Sie, dass ϕ_i in der Theorie der Algebra A ist d. h. $A \models \phi_i$:

(a) $\phi_1 := \forall x : s.\ g(g(f(x))) \equiv x$

(b) $\phi_2 := \forall x, y : s.\ h(x, y) \equiv h(y, x)$

(c) $\phi_3 := \forall x, y : s.\ \neg h(x, y) \equiv c$

Aufgabe 7.4 (Kalküle)

(a) Geben Sie einen geeigneten Kalkül an, der die Länge eines Wortes $w \in \Sigma^*$ berechnet.

(b) Berechnen Sie mit Hilfe Ihres Kalküls die Länge der folgenden Wörter aus $\{a, b, c\}^*$:

- cab
- ϵ

Aufgabe 7.5 (Matching)

Bestimmen Sie den jeweils minimalen Matcher für die folgenden Matchingprobleme, falls ein Matcher existiert. Geben Sie dazu jeweils eine Herleitung im Matchingkalkül an, aus der dieser Matcher hervorgeht.
Existiert kein Matcher, geben Sie **eine** scheiternde Herleitung an.
Geben Sie in jeden Schritt die verwendete Regel an.

(a) Pattern: $t_a = f(g(x, y), h(g(y, x)), x)$, Target: $q_a = f(g(h(a), b), h(g(b, h(a))), h(a))$

(b) Pattern: $t_b = f(g(x, y), x)$, Target: $q_b = f(g(a, b), b)$

Aufgabe 7.6 (Fundierte Mengen)

Beweisen oder widerlegen Sie die Fundiertheit der folgenden Mengen:

(a) $(\mathbb{N} \times \mathbb{N}, \succ_a)$ mit $(m_1, n_1) \succ_a (m_2, n_2)$ genau dann, wenn $m_1 > m_2$ oder $n_1 > n_2$

(b) $(\mathbb{N} \times \mathbb{N}, \succ_a)$ mit $(m_1, n_1) \succ_a (m_2, n_2)$ genau dann, wenn $m_1 > m_2$ und $n_1 > n_2$

Aufgabe 7.7 (Relationenbeschreibungen und Induktionsaxiome)

(a) Geben Sie für den folgenden Datentyp die Relationenbeschreibung $R_{S[@A]}$ an.

```
structure S[@A] <=
  null(value : ℕ),
  first(this : @A, that : S[@A]),
  second(left : S[@A], middle : S[@A], right : S[@A])
```

(b) Betrachten Sie die folgenden Prozeduren p:

1.
```
function f(m : ℕ, n : ℕ) : ℕ <=
  if ?0(n)
    then 0
    else m + f(m, ⁻(n))
  end_if
```

   Hierbei sei + folgendermaßen definiert:

```
function [infixr, 10] +(m : ℕ, n : ℕ) : ℕ <=
  if ?0(m)
    then n
    else ⁺(⁻(m) + n)
  end_if
```

2.
```
function g(m : ℕ, n : ℕ) : list[ℕ] <=
  if ?0(m)
    then 0 :: ∅
    else if ?0(n)
           then ∅
           else m :: g(⁻(m), ⁻(n))
    end_if
  end_if
```

- Geben Sie zu jeder Prozedur p die zusammengesetzte Relationenbeschreibung R_p an.
- Geben Sie zu jeder Prozedur p das Induktionsaxiom $IndAx_{p(x,y)=p(^+(x),y),R_p}$ an.

Aufgabe 7.8 (Berechnungskalkül)

Betrachten Sie das Programm P mit folgender Prozedurdefinition:

```
function a(x : ℕ, y : ℕ) : ℕ <=
if ?0(x)
  then y
  else if ?0(y)
         then x
         else ⁺(a(⁻(y), ⁻(x)))
       end_if
end_if
```

Bestimmen den Wert von $eval_P(\mathtt{a}(5,1))$, indem Sie eine Herleitung im Berechnungskalkül angeben. Geben Sie dabei in jedem Schritt die verwendeten Regeln an.

Aufgabe 7.9 (Terminierung)

Beweisen Sie die Terminierung der Prozedur f aus dem nachfolgenden $\mathcal{L}$-Programm, gehen Sie dazu wie folgt vor:

(a) Bestimmen Sie geeignete Maßterme für den Terminierungsbeweis. Überlegen Sie sich dazu, was bei den rekursiven Aufrufen kleiner wird. Für in den Maßtermen verwendete Hilfsprozeduren geben Sie deren Definitionen an oder verweisen Sie auf deren Definitionen im Vorlesungskript oder bisherigen Übungen.

(b) Geben Sie die zusammengesetzte Relationenbeschreibung der Prozedur an.

(c) Geben Sie die aus Maßtermen und Relationenbeschreibung gebildeten Terminierungshypothesen an.

(d) Belegen Sie die Gültigkeit der Terminierungshypothesen durch Anwendung bekannter arithmetischer Gleichungen.

```
structure bool <=
  true,
  false

structure ℕ <=
  0,
  ⁺(⁻ : ℕ)

function f(x : ℕ, y : ℕ, z : ℕ) : ℕ <=
  if x > z
    then f(⁺(z), ⁺(y), x)
    else if z > x
           then z
           else if ?0(y)
                  then x
                  else f(⁺(x), ⁻(y), ⁺(z))
                end_if
         end_if
  end_if
```

Aufgabe 7.10 (Minimale Elemente und Vorgänger)

Bestimmen Sie für jede der folgenden Mengen $(M, \succ)$ die minimalen Elemente $min_{\succ}(M)$. Geben Sie außerdem für jedes Element $m \in M$ die Anzahl der Vorgänger $|pre_{\succ}(m)|$ an.

(a) $(\mathbb{N} \times \mathbb{N}, \succ_a)$ mit $(m_1, n_1) \succ_a (m_2, n_2)$ genau dann, wenn $m_1 > m_2$ und $n_1 > n_2$
Hinweis: $>$ bezeichnet die übliche (transitive) Ordnung auf den natürlichen Zahlen

(b) $(C_{\texttt{list}[\mathbb{N}]}, \succ_b)$ mit $k \succ_b l$ genau dann, wenn $\texttt{but\_last}(k) = \texttt{but\_last}(l)$ und $k \neq \emptyset$ und $l \neq \emptyset$ und $\texttt{last}(k) > \texttt{last}(l)$
Hinweise:

- die Funktion `but_last`(k) schneidet das letzte Element einer Liste ab (siehe Übung 2 Aufgabe 2.3)
- die Funktion `last`(k) gibt das letzte Element einer Liste zurück (siehe Kapitel 5, Folie 19)

Aufgabe 7.11 (Induktion)

Betrachten Sie das folgende $\mathcal{L}$-Programm P:

```
structure bool <=
        true,
        false

structure ℕ <=
        0,
        ⁺(⁻ : ℕ)

function [infixr, 10] +(m : ℕ, n : ℕ) : ℕ <=
  if ?0(m)
    then n
    else ⁺(⁻(m) + n)
  end_if

function f(m : ℕ, n : ℕ) : ℕ <=
  if ?0(n)
    then 0
    else m + f(m, ⁻(n))
  end_if

function zero(x : ℕ) : ℕ <=
  if ?0(x)
    then 0
    else zero(pred(x))
  end_if

lemma null <= ∀ x : ℕ
        f(0, x) = zero(x)
```

Beweisen Sie $AX_{\texttt{null}}$ durch Induktion über x mittels der Relationenbeschreibung $R_{\mathbb{N}}$.

Gehen Sie dazu wie folgt vor:

(a) Geben Sie die Menge AX_p der Axiome zum Programm P an.

(b) Bilden Sie die Induktionsformeln (nach Kapitel 6, Folie 19f.) in Form von HPL-Sequenzen seq_{I_i} für den Rumpf b des Lemmas `null` und eine geeignete Umbenennung $R'_{\mathbb{N}}$ von $R_{\mathbb{N}}$.

(c) Beweisen Sie die Induktionsformeln I_i, zeigen Sie also für jede Sequenz $seq_{I_i} = \langle H_i, IH_i \Vdash b\rangle$:

$$AX_p \cup \mathcal{E}_P \quad \models \quad \forall x : \mathbb{N} \left(\bigwedge_{h \in H} h \equiv \texttt{true} \wedge \bigwedge_{ih \in IH_i} ih \equiv \texttt{true} \right) \to b \equiv \texttt{true}$$

Geben Sie in jedem Schritt an, welche Gleichung Sie anwenden.