

---

 Übungen zur Vorlesung Formale Spezifikations- und Verifikationstechniken
 

---

Prof. Dr. Klaus Madlener

Blatt 10

**32. Aufgabe:** [Vervollständigung modulo  $\sim$ ]

Sei  $>$  eine Knuth-Bendix-Ordnung mit Gewichtsfunktion  $\varphi$  definiert durch  $\varphi(s) = 1$  für  $s \in F \cup V$ .

Seien  $E = \{f(x+y) \rightarrow f(x) * f(y), f(0) \rightarrow 1, x+0 \rightarrow x, 0+x \rightarrow x, x*1 \rightarrow x, 1*x \rightarrow x\}$  und  $G = \{x+y = y+x, (x+y)+z = x+(y+z), x*y = y*x, (x*y)*z = x*(y*z)\}$ .

Vervollständigen Sie  $E$  modulo  $G$  bzgl.  $>$ .

**33. Aufgabe:** [Implementierung durch Gleichungen]

Seien  $g : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ ,  $h : \mathbb{N}^n \rightarrow \mathbb{N}$  primitiv rekursive Funktionen und sei  $f : \mathbb{N}^n \rightarrow \mathbb{N}$  definiert durch:

$$f(x_1, \dots, x_n) = \mu_{z \leq h(x_1, \dots, x_n)} [g(x_1, \dots, x_n, z) = 0]$$

Geben Sie ein Gleichungssystem  $G_{\hat{f}}$  und ein Funktionssymbol  $\hat{f}$  an, so dass  $\hat{f}$  die Funktion  $f$  in  $G_{\hat{f}}$  implementiert.

**34. Aufgabe:** [PL-Semantik]

Es sei folgende Grammatik zur Erzeugung von PL-Programmen gegeben:

$$\Sigma = \{A, B, \dots, Z, 0, 1, \dots, 9, \leftarrow, +, :, ;, \text{loop}, \text{end}, \text{goto}\}$$

$$\langle \text{Buchstabe} \rangle \rightarrow A \mid B \mid \dots \mid Z$$

$$\langle \text{Ziffer} \rangle \rightarrow 0 \mid 1 \mid \dots \mid 9$$

$$\langle \text{Name} \rangle \rightarrow \langle \text{Buchstabe} \rangle \mid \langle \text{Name} \rangle \langle \text{Buchstabe} \rangle \mid \langle \text{Name} \rangle \langle \text{Ziffer} \rangle$$

$$\langle \text{Zuweisung} \rangle \rightarrow \langle \text{Name} \rangle \leftarrow 0 \mid \langle \text{Name} \rangle \leftarrow \langle \text{Name} \rangle + 1 \mid \langle \text{Name} \rangle \leftarrow \langle \text{Name} \rangle$$

$$\langle \text{Befehl} \rangle \rightarrow \langle \text{Zuweisung} \rangle \mid \text{goto } \langle \text{Name} \rangle$$

$$\langle \text{mark. Befehl} \rangle \rightarrow \langle \text{Befehl} \rangle ; \mid \langle \text{Name} \rangle : \langle \text{Befehl} \rangle$$

$$\langle \text{Schleifenanfang} \rangle \rightarrow \text{loop } \langle \text{Name} \rangle ; \mid \langle \text{Name} \rangle : \text{loop } \langle \text{Name} \rangle ;$$

$$\langle \text{Schleifenende} \rangle \rightarrow \text{end}; \mid \langle \text{Name} \rangle : \text{end};$$

$$\langle \text{Programm} \rangle \rightarrow \langle \text{mark. Befehl} \rangle$$

$$\mid \langle \text{Schleifenanfang} \rangle \langle \text{Programm} \rangle \langle \text{Schleifenende} \rangle$$

$$\mid \langle \text{Programm} \rangle \langle \text{Programm} \rangle$$

1. Stellen Sie die operationale Semantik der PL-Programme durch eine Funktion  $M_e : P \times \mathbb{N}^k \rightarrow \mathbb{N}^k$  dar, wobei  $k$  die Anzahl der im Programm vorkommenden Variablen ist.
2. Bestimmen Sie geeignete Termmengen, ein Gleichungssystem  $G$  und eine Funktion EVAL, die  $M_e$  in  $G$  implementiert.

Hinweise: Zur Vereinfachung der Lösung können „sinnvolle“ Einschränkungen oder Erweiterungen der obigen Grammatik vorgenommen werden, beispielsweise kann man fordern, dass jeder Befehl markiert ist, wobei die Marke eine natürliche Zahl ist.

Ein möglicher Ansatz ist es, Befehle als Konstanten (z.B.  $c_{i0} \equiv x_i \leftarrow 0$ ,  $c_{ij} \equiv x_i \leftarrow x_j$ ,  $c'_{ij} \equiv x_i \leftarrow x_j + 1$ ) und Programme als Listen von Befehlen darzustellen.

Untersuchen Sie, inwieweit Sie mit ihrem System „Sonderfälle“, wie z.B. Sprünge in oder aus Schleifen etc. behandeln können.

**Abgabe: bis 02.02.2006, per EMail an Bernd Strieder**