



Übungsblatt 10

21.12.2001

Aufgabe 1 While-Sprache

(Bierwald)

Eine Maschine ist nicht dann perfekt wenn man nichts mehr hinzufügen kann, sondern erst wenn es nichts mehr gibt, das man weglassen könnte.

ANTOINE DE SAINT-EXUPERY

Eine Maschine ist die While-Sprache (ohne Ergänzungen) nicht. Aber ist sie perfekt?

Hinweis (Rot13). Mrvtra Fvr, qnff zna nhs qvr Ibetnratre-Shaxgvba iremvpugra xnaa haq ahe Mhjrxfhatra qre Sbez "K orpbzrf 0" bqre "K orpbzrf Anpus(K)" orabrgvtg.

Aufgabe 2 Berechenbare Funktionen

(Bierwald)

Sei m Ihre Matrikelnummer. Ist die Funktion $f(x) = \begin{cases} 1, & \text{falls } \varphi_m(m) \downarrow \\ 0, & \text{sonst} \end{cases}$ berechenbar?

Aufgabe 3 While-Programme

(Bierwald)

Zeigen Sie, daß kein While-Programm (ohne Ergänzungen) mit genau einer Variablen die Funktion $f(x) = 2x$ berechnen kann.

Aufgabe 4 Gödelisierungen

(Bierwald, Käußl)

a) Ein fauler Mensch sagt, man könne die Gödelisierung dadurch vereinfachen, daß man jedes Zeichen der While-Sprache als Ziffer eines geeigneten Zahlensystems auffaßt, wodurch jedes While-Programm zu seiner eigenen Gödelzahl wird. Ein anderer meint, man könne die Zahl der Zeichen, aus denen das Programm besteht, als Gödelzahl ansehen. Was meinen Sie dazu?

b) Jede natürliche Zahl ist eindeutig als Produkt von Primzahlpotenzen darstellbar. Konstruieren Sie unter Benutzung dieses Sachverhalts eine Gödelisierung der While-Programme (ohne Ergänzungen).

Aufgabe 5 Übersetzungstechniken für Programme der ergänzten While-Sprache

(Käußl)

a) Geben Sie eine allgemeine Methode an, die erlaubt, Zuweisungen aus der While-Sprache mit Ergänzungen durch Folgen von Zuweisungen der Form $x := T$ äquivalent zu ersetzen. T ist dabei Term der While-Sprache, von der Form $y \text{ op } z$, mit op als Operationssymbol der ergänzten While-Sprache oder Variable.

b) Geben Sie ein allgemeines Verfahren an, das erlaubt, Zuweisungen der Form $x := y \text{ op } z$ durch Einsetzen des Programms für op in eine äquivalente Folge von Anweisungen der While-Sprache umzuwandeln. (Hinweis: Verwenden Sie als Definition: Ein While-Programm

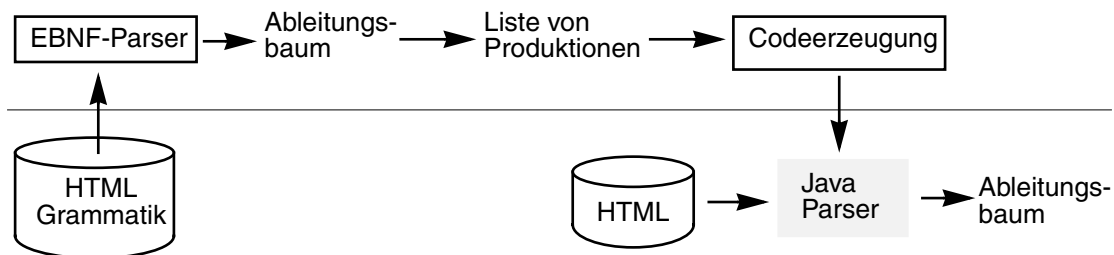
ist ein Paar $(\{X_1, \dots, X_n\}, P)$, wobei P der Programmtext in der While-Sprache ist und $\{X_1, \dots, X_n\}$ die in P vorkommenden Variablen.)

c) Geben Sie die Schritte an, die notwendig sind, um ein Programm in der While-Sprache mit Ergänzungen in ein äquivalentes überzuführen, das die Bedingungen der Teilaufgabe a erfüllt.

Aufgabe 6 Die fast überall verschwindende Funktion (Käufel)
 Zeigen Sie, daß jede Funktion mit N als Definitionsbereich, die nur für endlich viele Argumente einen von Null verschiedenen Wert hat, berechenbar ist.

Aufgabe 7 Parser Generator – Abschluß (Schuster)
 Schreiben Sie einen Parser Generator, welcher eine Datei mit Produktionen in *EBNF* Form einliest und einen Java-Parser erzeugt.

Für einen vollständigen Parser Generator muß zum einen eine Grammatik eingelesen werden und zum anderen der Code für einen Parser generiert werden. Der erste Schritt wird mit einem handgeschriebenen rekursiven SLL(1) Parser für *EBNF* durchgeführt. Jede Methode erzeugt dabei einen Knoten des Ableitungsbaums und fügt erzeugte Teilbäume ein. Für den Schritt der Codeerzeugung werden zunächst die $\text{First}_1(r\text{Follow}(A))$ Mengen bestimmt. Anschließend können die rekursiven Methoden erzeugt werden. Beispiel:



Der Benutzer erstellt eine SLL(1)-Grammatik für HTML, speichert diese in eine Datei und übergibt diese dem Parser Generator. Der Generator liest die Datei mit dem EBNF-Parser aus und erzeugt einen Ableitungsbaum. Im Baum sind alle HTML-Produktion der Datei aufgeführt und werden in eine Liste von Produktionen überführt. Abschließend wird diese Liste an die Codeerzeugung weitergegeben, welche einen Akzeptor – einen Java-Parser – erzeugt. Der Java-Parser kann jetzt vom Benutzer in eigene Programme eingebunden werden, um HTML-Dokumente einzulesen.

Wir wünschen ein frohes Weihnachtsfest und einen

guten Rutsch ins nächste Jahr!

