

Zuname _____ Vorname _____ Matr.-Nr. _____

Übungsgruppe

Punkte _____ korr. _____

- 1 (Löberbauer) Do 10¹⁵-11⁴⁵
- 2 (Löberbauer) Do 13⁴⁵-15¹⁵
- 3 (Würthinger) Do 10¹⁵-11⁴⁵

Letzter Abgabetermin

Mittwoch, 15.10.2008, 18⁰⁰ Uhr

Grammatiken

1. Grundbegriffe

(2+3+3+6 Punkte)

Die Grammatik der Sprache *MicroJava* finden Sie im VO-Skriptum im Kapitel 4.2 auf Seite 7.

- a) Geben Sie den Satz mit der minimalen Anzahl von Terminalsymbolen an, den man mit dieser Grammatik erzeugen kann und der mindestens einen *Type* enthält. Aus wie vielen Terminalsymbolen besteht er?
- b) Betrachten Sie die Nonterminalsymbole *Statement*, *Factor* und *ActPars*. Geben Sie für jedes dieser Nonterminalsymbole an, ob es links-, zentral- oder rechtsrekursiv ist, und ob es direkt oder indirekt rekursiv ist.
- c) Zeichnen Sie den Syntaxbaum für folgenden Satz:

```
program Adder int sum; { void add(int x) { sum = sum + 1; } }
```

 Gibt es mehrere Syntaxbäume für diesen Satz?
- d) Bestimmen Sie alle terminalen Anfänge und Nachfolger für die Regeln *ConstDecl*, *Statement*, *Relop* und *Expr*.

2. Konstruktion einer Grammatik

(5 Punkte)

Geben Sie eine Grammatik (in EBNF) für die Zahlen einer fiktiven Programmiersprache gemäß den folgenden Bedingungen an:

- Eine Zahl ist entweder eine Fließkomma- oder eine Hexadezimalzahl.
- Hexadezimalzahlen beginnen mit "h".
- Fließkommazahlen enthalten Tausender-Punkte, d.h. wenn eine Zahl aus mehr als drei Ziffern besteht, muss nach jeder dritten Ziffer (gerechnet von rechts) ein Punkt (".") stehen. Vorkomma- und Nachkommastellen werden mit einem Beistrich getrennt (",").
- Die Terminalklassen *d* (digit) und *h* (hex digit) sind gegeben, *d* deckt die Ziffern zwischen 0 und 9, und *h* die Ziffern A bis F ab.

Beispiele für gültige Zahlen: 123, 1.456, hAF, h23, h1A, h23B9, 432.212,12, 12,2, 1.243,2

Beispiele für ungültige Zahlen: 1234, 567.8, 5432,1, h5.476, h, h2,A

3. Beseitigung von Linksrekursionen

(5 Punkte)

Gegeben sei folgender Auszug aus einer fiktiven Grammatik, der Zuweisungen beschreibt:

```
ConstDecl = "const" Type ident "=" Expression
           | ConstDecl "," ident "=" Expression .
Type       = "int" | "float" .
Expression = ( Expression ( "+" | "-" ) number ) | number .
```

Beseitigen Sie alle Linksrekursionen und geben Sie die transformierte Grammatik in EBNF an. *ident* und *number* sind Terminalklassen, die einen Namen (Buchstabe gefolgt von Ziffern und Buchstaben) bzw. eine Zahl (bestehend aus Ziffern) definieren.