

Zuname _____ Vorname _____ Matr.-Nr. _____

Übungsgruppe

Punkte _____ korr. _____

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

Letzter Abgabetermin

Mittwoch, 11.10.2006, 18⁰⁰ Uhr

Grammatiken

1. Grundbegriffe

(2+3+3+6 Punkte)

Die Grammatik der Sprache *MicroJava* finden Sie im VO-Skriptum im Kapitel 5.2 auf Seite 8.

- a) Geben Sie den Satz mit der minimalen Anzahl von Terminalsymbolen an, den man mit dieser Grammatik erzeugen kann und der mindestens ein "Statement" enthält. Aus wie vielen Terminalsymbolen besteht er?
- b) Betrachten Sie die Nonterminalsymbole *Statement*, *ClassDecl* und *Expr*. 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 Car int speed; { void accelerate(int d) { speed = speed + d; } }
```

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

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 Dezimalzahl oder eine Binärzahl.
- Ein Zahl beginnt mit "#" gefolgt von dem Wert der Zahl.
- Dezimalzahlen bestehen aus Ziffern (Terminalklassen *b* und *z*). Wenn eine Dezimalzahl aus mehr als drei Ziffern besteht, muss nach jeder dritten Ziffer (gerechnet von rechts) ein Punkt (".") stehen (Tausender-Punkte). Eine Dezimalzahl endet mit einem "d".
- Binärzahlen bestehen aus Ziffern (Terminalklasse *b*). Eine Binärzahl endet mit einem "b".

Beispiele für gültige Zahlen: #123d, #1.456d, #1101b

Beispiele für ungültige Zahlen: 1234, 567.8, #5476d, #2b, #121.3d, #1.100b, #5, #101, #b, #d

3. Beseitigung von Linksrekursionen

(5 Punkte)

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

```
Assignment = MultiAss ":" Value.  
MultiAss   = ident | MultiAss "=" ident.  
Value      = Value "+" 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.