

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, 13.10.2010, 18⁰⁰ Uhr

Grammatiken

1. Grundbegriffe

(2+3+3+6 Punkte)

Die Grammatik der Sprache *MicroJava* finden Sie auf der Übungshomepage:
<http://ssw.jku.at/Teaching/Lectures/UB/UE/2010/MicroJava.txt>.

- a) Wie lang ist der kürzeste Satz in *MicroJava*, der mindestens einen *Type* und ein *Statement* enthält? Geben Sie Ihren Satz an.
- b) Sind die Nonterminalsymbole *Statement*, *Factor*, *CondTerm* rekursiv? Wenn ja wie (links, rechts, zentral, direkt, indirekt)?
- c) Wie sieht der Syntaxbaum für folgenden Satz aus?

```
program DeepThought final int a=100; { int getAnswer() { return a-58; } }
```

 Gibt es mehrere Syntaxbäume für diesen Satz?
- d) Welche terminalen Anfänge und Nachfolger haben die Regeln *ClassDecl*, *Statement*, *AssignOp* und *Condition*?

2. Konstruktion einer Grammatik

(5 Punkte)

Wie könnte man, die folgenden Bedingungen mit einer EBNF-Grammatik beschreiben? Geben Sie eine Grammatik.

- Eine Zahl ist entweder eine Dezimal-, eine Hexadezimal-Zahl oder eine Zeitangabe.
- Hexadezimalzahlen beginnen mit "0x".
- Zeitangaben enden mit M (Minuten) oder S (Sekunden)
- Zeitangaben können Kommastellen haben (durch Punkt getrennt, zB 1.25s).
- 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: 0, 123, 1.456S, 0xAF, 0x23, 0x1A, 0x23B9, 432.212S, 1.S, .5M

Beispiele für ungültige Zahlen: 1.5, A, 0xG, 0x12F.3, 0x3AS

3. Beseitigung von Linksrekursionen

(5 Punkte)

Wo befinden sich in der folgenden Grammatik Linksrekursionen? Wie könnte man die Linksrekursionen entfernen? Geben Sie eine umgeformte Grammatik ohne Linksrekursionen an.

```
VarDecl
=
  ( Type ident [ "=" Expression ] | VarDecl "," ident [ "=" Expression ] )
.
Type
=
  "int" | "double"
.
Expression
=
  ( Expression ( "+" | "-" ) number ) | number
.
```

ident und *number* sind Terminalklassen, die einen Namen (Buchstabe gefolgt von Ziffern und Buchstaben) bzw. eine Zahl (bestehend aus Ziffern) definieren.