

## Grammatiken

**(24 Punkte)**

### 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/2015/MicroJava.txt>.

- Wie lang (in Terminalsymbolen) ist der kürzeste Satz in *MicroJava*, der mindestens ein *CondFact* enthält? Geben Sie ein Beispiel für einen Satz dieser Länge an.
- Sind die Nonterminalsymbole *Term*, *Statement* und *Relop* rekursiv? Wenn ja, wie (links, rechts, zentral, direkt, indirekt)?
- Wie sieht der konkrete Syntaxbaum für folgenden Satz aus?  

```
program DeepThought final int a=1764; { int getAnswer(){ return 2 \/ a; }}
```

Gibt es mehrere konkrete Syntaxbäume für diesen Satz?
- Welche terminalen Anfänge und Nachfolger haben die Regeln *Statement*, *Condition* und *ConstDecl*?

### 2. Konstruktion einer Grammatik

**(5 Punkte)**

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

- Eine Zahl ist entweder eine Dezimal-, eine Hexadezimal-Zahl oder eine Zeitangabe.
- Hexadezimalzahlen beginnen mit "0x".
- Zeitangaben haben entweder das Format Minuten: Sekunden oder das Format Stunden: Minuten: Sekunden. Der erste Teil (also Minuten bzw. Stunden) darf ein oder zwei Stellen haben, alle anderen müssen zwei Stellen haben. Minuten und Sekunden müssen zwischen 00 und 59 liegen.
- 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. Außerdem können Sie die Terminalklasse *sd* (small digit), die Ziffern zwischen 0 und 5 enthält verwenden.

Beispiele für gültige Zahlen: 0, 1, 42, 0x123, 1:35, 0xAF, 0x0A, 0x23B9, 04:17, 1:00:00, 99:00:53

Beispiele für ungültige Zahlen: 0b012, 1.5, A, 0x, b, 0xG, 0x12F.3, :01:17, 13:1, 1:00:99

### 3. Beseitigung von Linksrekursionen

(5 Punkte)

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

### Abgabe und Hinweise

Die Abgabe der Übungen muss elektronisch erfolgen. Geben Sie folgende Dateien ab:

- Elektronisch in das Repository: PDF-Datei mit der Lösung.
- [svn://svn.ssw.jku.at/2015W/UB/k<MatrNr>/branches/UE1](https://svn.ssw.jku.at/2015W/UB/k<MatrNr>/branches/UE1)