

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/2017/MicroJava-Grammatik.pdf>.

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

```
program PI { void nonsense() int pi; { pi = 3 * 4;}}
```


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

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 **natürliche positive Zahl (inklusive 0)**, eine **Zeitangabe** oder eine **Datumsangabe**.

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.

Datumsangaben folgen dem Format Tag.Monat.Jahr. Der Punkt zwischen Monat und Jahr ist **nur dann** erlaubt, wenn Monat eine Zahl ist. Ansonsten soll nach dem Monat ein Leerzeichen erfolgen. Tag darf ein oder zwei Stellen haben, mit einem maximalen Wert von 31. Monat kann als Wort oder Zahl (mit 1 oder 2 Stellen, maximaler Wert 12) geschrieben werden und Jahr **muss** immer eine vierstellige Zahl sein. Es ist **nicht** erforderlich die Kalender Semantik einzuhalten, nach der bestimmte Monate 28 / 30 oder 31 Tage haben. Jeder Monat kann 31 Tage haben.

Die Terminalklasse *d* (digit) ist gegeben, sie deckt die Ziffern zwischen 0 und 9 ab. Außerdem können Sie die Terminalklasse *sd* (small digit), die Ziffern zwischen 0 und 5 enthält verwenden.

Leerzeichen können entweder direkt angegeben werden (" ") oder mit der Terminalklasse *blank*.

Weiters ist die Terminalklasse *monthString* (Month string) gegeben, welche die Monate in textueller Form enthält ("Jänner" - "Dezember").

Beispiele für gültige Zahlen: 0, 1, 42, 1:35, ~~0e2345~~, 04:17, 1:00:00, 99:00:53, 1.März 2017, 1.3.2017, 01.03.2017, 31.Februar 2018

Beispiele für ungültige Zahlen: 1.5, A, :01:17, 13:1, 1:00:99, 32. Februar. 2018, 31. Februar.201, 111.Februar 20, 31.Februar2042

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" | "float"
.
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://ssw.jku.at/2017W/UB/k<MatrNr>/branches/UE1