

## Übungsblatt 01

Termin: 2007/03/13

---

**Ü 1.1 Grammatiken und formale Sprachen**

---

Gegeben sei die Grammatik  $G = (V_N, V_T, P, S)$  mit  $V_N = \{ A, B, S \}$ ,  $V_T = \{ a, b \}$  und  $P$ :

$$S \rightarrow A B a,$$
$$A \rightarrow a A, A \rightarrow a,$$
$$B \rightarrow bB, B \rightarrow \epsilon.$$

- Zeigen Sie, dass in der Grammatik  $G$  folgende Worte aus  $S$  ableitbar sind:  $aaa$ ,  $abba$ .
- Geben Sie die von  $G$  erzeugte Sprache  $L(G)$  in Mengendarstellung an. Finden Sie auch einen regulären Ausdruck für  $L(G)$ .
- Begründen Sie, warum  $G$  kontextfrei, aber nicht regulär ist. Finden Sie eine reguläre Grammatik  $G_R$  mit  $L(G_R) = L(G)$ .

---

**Ü 1.2 EBNF**

---

In der VO wurde die Syntax der Sprache PL0 durch eine Variante der Extended Backus-Naur Form (EBNF) definiert, die von Niklaus Wirth stammt.

- Stellen Sie die EBNF-Produktion für die Variable `ident` durch entsprechende Produktionen einer kontextfreien Grammatik dar (sodass die Menge der ableitbaren Worte für beide Darstellungen übereinstimmt).
- Es gibt einen Internationalen Standard ISO-14977 für die EBNF-Notation, der unter <http://www.iso.org/> frei erhältlich ist. Stellen Sie die Produktionen `ident` und `SimpleExpression` der Sprache PL0 in der EBNF-Notation gemäß ISO-14977 dar.

---

**Ü 1.3 Lexikalische Analyse – MiniCompiler**

---

Unter <http://minicompiler.itec.uni-klu.ac.at/> finden Sie den Source Code des in der VO vorgestellten MiniCompilers für die Sprache PL0. Laden Sie die Version *Stage 1 Lexer* herunter und studieren Sie den Source Code der Klasse `Scanner`.

- Welche Lexeme sind im HashSet `singles` enthalten? Welche gemeinsame Eigenschaft unterscheidet sie von den übrigen Lexemen der Sprache PL0?
- Warum ist das Lexem `<` nicht im HashSet `singles` enthalten? Wie wird dieses Lexem vom Scanner des MiniCompilers erkannt?
- In welche Lexeme bzw. Token würde der MiniCompiler folgende Eingabezeichenkette zerlegen? Geben Sie auch die Stellen im Source Code an, wo die entsprechenden Tokenobjekte erzeugt werden.

```
IFa<=bTHEN (* ??? *)
```