

## 5. Übung „Übersetzerbau“

Abgabe am 23. Mai in der Vorlesung

---

### Aufgabe 17

5 Punkte

In der 4. Übung Aufgabe 16 haben Sie einen Parsing-Tabelle für die Sprache MPS erstellt. Nun soll ein Parser für MPS implementiert werden.

Zur Vereinfachung des Parsers können Sie von folgender Vereinfachung von MPS ausgehen: Bezeichner werden bei ihrer Deklaration durch Kommas statt durch Semikolons getrennt. Implementieren Sie den rekursiven Abstiegsparser für MPS in Haskell unter Berücksichtigung der (angepassten) Parsing-Tabelle.

Kombinieren Sie Ihren Parser mit dem (ggf. leicht modifizierten) Scanner aus Aufgabe 13. Erweitern Sie Ihren Parser, so dass er auch einen abstrakten Syntaxbaum erzeugt.

### Aufgabe 18

5 Punkte

- Geben Sie eine zur Definition äquivalente Charakterisierung der Menge der LL(0)-Grammatiken sowie der Menge der LL(0)-Sprachen (d.h. der durch LL(0)-Grammatiken erzeugbaren Sprachen) an. Diese Definition soll natürlich einfacher sein, als die in der Vorlesung gegebene.
- Zeigen Sie, dass jede reguläre Sprache durch eine LL(1)-Grammatik erzeugt wird.

### Aufgabe 19

4 Punkte

Die kontextfreie Grammatik  $G$  sei gegeben durch:

$$\begin{aligned} S' &\rightarrow S\$ \\ S &\rightarrow AS \mid b \\ A &\rightarrow SA \mid a \end{aligned}$$

- Konstruieren Sie (analog zur Vorlesung) einen erfolgreichen Lauf des shift-reduce-Parsers für die Eingabe  $abab$ .
- Berechnen Sie den LR(0)-Automaten zu  $G$ . Ist  $G \in LR(0)$ ?

### Aufgabe 20

6 Punkte

Die Grammatik  $G_{\text{bool}}$  sei gegeben durch

$$\begin{aligned} S &\rightarrow B\$ \\ B &\rightarrow (B \wedge B) \mid \neg B \mid t \mid f \end{aligned}$$

- Berechnen Sie die LR(0)-Automaten von  $G_{\text{bool}}$ .
- Geben Sie die LR(0)-Analysetabelle von  $G_{\text{bool}}$  an.
- Bestimmen Sie die Konfigurationsfolge, die der LR(0)-Analyseautomat bei Eingabe des Wortes  $((\neg t \wedge f) \wedge (f \wedge t))$  durchläuft.