Lehrstuhl für Programmiersprachen und Übersetzerkonstruktion
CAU Kiel, Informatik und Prakt. Mathematik
Prof. Dr. M. Hanus, Dr. F.Huch

24118 Kiel
Christan-Albrechts-Platz 4
Tel.: 0431 / 880-7277

4. Übung „Übersetzerbau“

Abgabe am 16. Mai in der Vorlesung

Aufgabe 13 5 Punkte

Implementieren Sie von Hand einen Scanner für die Programmiersprache Simple in Haskell. Definieren Sie zunächst einen algebraischen Datentyp `Token`, der alle möglichen Symbolklassen enthält. Definieren Sie dann eine Funktion

```
scan :: String -> [Token]
```

Aufgabe 14 5 Punkte

In der Vorlesung wurde die LL(k)- und die starke LL(k) (SLL(k))-Eigenschaft definiert. Zeigen Sie anhand folgender Grammatik, dass SLL(2) $\neq$ LL(2) gilt:

$$S \to aAab \mid bAbb \qquad\qquad A \to a \mid \varepsilon$$

Bem.: Für $k = 1$ stimmen beide Eigenschaften überein.

Aufgabe 15 4 Punkte

In der Vorlesung wurde die Linksfaktorisierung am Beispiel des so genannten „dangling if-then-else“ vorgestellt. Überprüfen Sie, ob die resultierende Grammatik die LL(1)-Eigenschaft besitzt. Können Sie diese Grammatik ggf. durch weitere Linksfaktorisierungen bzw. durch Elimination von Linksrekursion in eine LL(1)-Grammatik überführen?

Aufgabe 16 6 Punkte

Die Syntax der Mini–Programmiersprache MPS sei durch die Syntaxdiagramme auf der folgenden Seite gegeben.

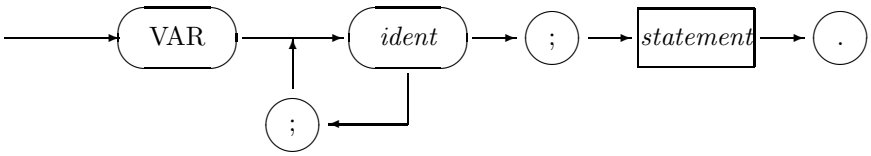
a) Geben Sie die Syntax von MPS durch eine kontextfreie Grammatik G an.

b) Berechnen Sie zu allen Nichtterminalsymbolen aus G die Mengen FIRST und FOLLOW.

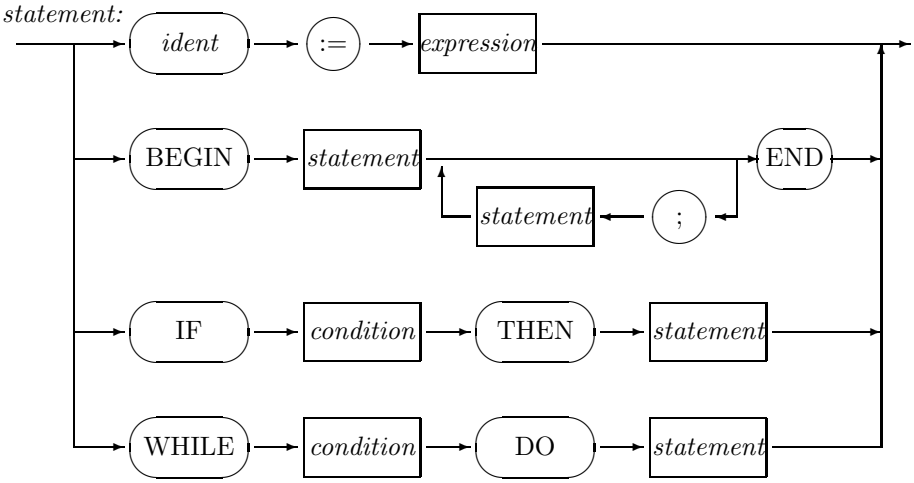
c) Prüfen Sie, ob es sich bei Ihrer Grammatik um eine LL(1)–Grammatik handelt. Wenn nicht, so geben Sie eine LL(1)–Grammatik G' für MPS an.

d) Konstruieren Sie die zugehörige Parsing-Tabelle. Sollten Sie Aufgabenteil c) nicht gelöst haben, geben Sie dennoch eine (mehrdeutige) Parsingtabelle an.

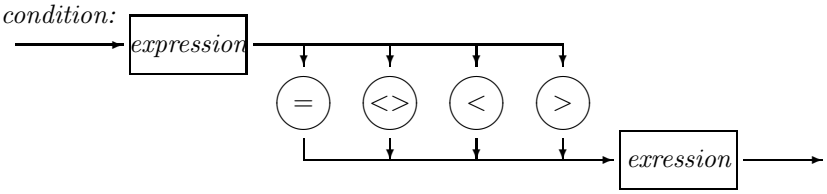
program:



statement:



condition:



expression: