

## Übungsblatt 6

Besprechungstermin: 24.11.2009

### Aufgabe 1 (freiwillig, ersetzt Aufgaben 2-5)

Besuchen Sie die **Flex**-homepage <http://sourceforge.net> und schreiben Sie ein kleines **Lex**-Programm. Erzeugen Sie den zugehörigen **Lexer** mit dem **Flex**-Generator und testen Sie das Ergebnis an mehreren Eingaben.

### Aufgabe 2

Finden Sie in dem folgenden Pascal-Programm die Lexeme der Symbole. Überlegen Sie sinnvolle Attributwerte für die Symbole.

```
function max (i,j:integer):integer;  
{liefert das Maximum der Zahlen von i und j}  
begin  
    if i > j then max:= i  
    else max := j  
end;
```

### Aufgabe 3

Schreiben Sie ein Programm für die Funktion **nextchar** (). Verwenden Sie dabei ein Pufferschema mit Wächtern.

### Aufgabe 4

- a) Entwerfen Sie eine zweistufige Grammatik  $G$  für arithmetische Ausdrücke über natürlichen Zahlen, positiven Dezimalzahlen und Variablen, die mit einem kleinen, lateinischen Buchstaben beginnen und danach aus einer Folge von Buchstaben einschl. – und schließlich einer Folge von Ziffern bestehen. Als Operationen seien +, \*, \*\*, sin, cos und tan erlaubt.
- b) Entwickeln Sie einen Scanner für die von  $G$  erzeugte Sprache  $L(G)$  durch Implementierung der entsprechenden Übergangsdiagramme. Der Scanner soll für eine natürliche Zahl  $n$  den Attributwert  $n$  liefern und die Werte von Dezimalzahlen in der Symboltabelle festhalten.

**Hinweis:** Sorgen Sie für eine angemessene Behandlung von Trennzeichen.

### Aufgabe 5

Erweitern Sie das LEX-Programm aus der Vorlesung durch

- a) Hinzufügen des Schlüsselwortes `while`.
- b) Ersetzen der Vergleichsoperatoren durch diejenige von Java.
- c) Zulassen des Unterstrichs als zusätzlichen Buchstaben.
- d) Hinzufügen eines neuen Musters für Strings. Strings sollen in eine eigene Tabelle eingetragen werden.