PHILIPPS-UNIVERSITÄT MARBURG
Fachbereich Mathematik und Informatik
Prof. Dr. R. Loogen J.Berthold

D-35032 Marburg
Hans Meerwein Straße
Lahnberge

17.November 2003

Übungen zu „Grundlagen des Compilerbaus“, WS 2003/04

Nr. 4 , Abgabe und Besprechung: 24. November in der Übung

Mündliche Aufgaben

4 .1 **Scanner für umgangssprachliche Uhrzeitangaben**

Für eine Sprache zur umgangssprachlichen Angabe von Uhrzeiten seien die nebenstehenden Symbolklassen definiert.
Stunden und Minuten werden als Zahlen angegeben, die im gültigen Bereich (0-59) bleiben sollen. Zeitangaben wie z.B. “dreiviertel 12” oder “5 nach 1” lassen sich in diese Token zerlegen.

Haskell Code
```
data Token =
    Nach | Vor
    | Halb | Viertel
    | Dreiviertel
    | Zahl Int
```

(a) Spezifizieren Sie, was (unter Verwendung dieser Token) gültiger Zeitangaben sein sollen. Gehen Sie davon aus, dass Leerzeichen überlesen werden.

(b) Erstellen Sie mit Hilfe des Scanner-Generators `Alex` einen Scanner für diese Zeitangaben.

(c) Schreiben Sie ein Haskell-Programm, das den erzeugten Scanner importiert und aus der gelesenen umgangssprachlichen Angabe die entsprechende Zeit in Minuten seit 0:00 Uhr berechnet. Beispiele:
`viertel nach 7` $= 7 \cdot 60 + 15 = 435$, `dreiviertel 12` $= 12 \cdot 60 - 15 = 705$.

4 .2 **Pragmatik und Syntax von Programmiersprachen**

Diskutieren Sie die Eigenschaften der nebenstehenden Syntaxbeschreibung.
Welche Schwachpunkte sehen Sie? An welchen Stellen würden Sie, aus welchen Gründen, anders vorgehen?

Schriftliche Aufgaben

4 .3 **Aufwand von ALEX-generierten Scannern** 2 Punkte

Stellen Sie fest, mit welchem Aufwand ein von Alex generierter Scanner arbeitet.

4 .4 **Scanner für eine imperative Sprache** 10 Punkte

Für eine kleine WHILE-Programmiersprache sei die Mikrosyntax wie folgt definiert:

- Eine Variable besteht aus einem Buchstaben, gefolgt von beliebig vielen Buchstaben und Ziffern. Schlüsselwörter dürfen nicht als Variablen verwendet werden.

- Eine Zahl (NumVal) besteht entweder aus der Ziffer 0, oder aus einer nicht-leeren Ziffernfolge, die nicht mit einer 0 beginnt. Eine Zahl kann ein Vorzeichen + oder - haben.
- Ein Wahrheitswert (BoolVal) ist entweder True oder False.
- Arithmetische Operatoren (ArithOp): + − * /
- Relationale Operatoren (RelOp): = != < <= > >=

Die Syntax sei durch folgende Regeln beschrieben:

program	→	**program** Variable **{**
		decls
		stmts
		}
decls	→	**var**
		decl **;** (*decl* **;**)*
decl	→	Variable (**,** Variable)* **:** *type*
type	→	**int**
	|	**bool**
stmts	→	*simple* (**;** *simple*)*
simple	→	Variable **:=** *expr*
	|	**print** *expr*
	|	*cond*
	|	*whiledo*
cond	→	**if** *expr* **then** *stmts* (**else** *stmts*)?
whiledo	→	**while (** *expr* **) do** *stmts*
expr	→	Variable
	|	NumVal
	|	BoolVal
	|	*expr* ArithOp *expr*
	|	*expr* RelOp *expr*

(a) Geben Sie sinnvolle Symbolklassen für die lexikalische Beschreibung von FIPS an. Definieren Sie dann einen Haskell-Datentyp `Token` analog zu Ihren Symbolklassen. / 4

(b) Benutzen Sie den Scanner-Generator ALEX, um einen Scanner zu erzeugen, der ein FIPS-Programm in eine Liste von Symbolen umwandelt. / 4

(c) Erweitern Sie den Scanner und die Token-Definition, so dass die Symbole zusätzlich die Zeilennummer enthalten, in der sie stehen. / 2

Hinweise: Der Scanner-Generator ALEX ist nicht unter Windows, sondern nur unter Linux verfügbar. In der Übung wird der Umgang mit ALEX erläutert. Das Programm befindet sich im Verzeichnis `/app/lang/functional/bin/` und kann über ein Terminal-Programm benutzt werden (momentan allerdings nur auf den Linux-Rechnern *tunis, lusaka, douala* und *algiers*). In der Übung wird die Benutzung von Alex an einem Beispiel gezeigt.
Unter *http://www.haskell.org/alex/doc/html/alex.html* befindet sich eine Anleitung für Alex. Beispieldateien zu Alex sind von der Vorlesungsseite aus zu finden.