

Systematische Scanner-Implementierung

grober Aufbau des lex-input-files (<name>.l)

Erster Teil: optional, besteht aus:

Vorspann wird im Programm von lex unverändert eingesetzt:
meist #includes, #defines, C-Code

Abkürzungen

Abkürzende Bezeichnungen für reguläre Ausdrücke -> nur für Lex intern benutzt

Zweiter Teil: obligatorisch

Definition der **regulären Ausdrücke** für die Tokens

%%

Dritter Teil: optional

Nachspann, meist C Prozeduren, die 1:1 übertragen werden.

Lexikalische Analyse

Prof. Dr. U. Göhner/HS Kempten 58

Systematische Scanner-Implementierung

Aufbau im Detail:

Erster Teil:

Steuerzeichen innerhalb des ersten Teils:

%{ für Beginn Vorspann

%} für Ende Vorspann

Danach folgen Definitionen für abkürzende Schreibweise in den regulären Ausdrücken des zweiten Teils in folgender Form:

Bezeichner regulärer Ausdruck

z.B.: Ziffer [0-9]

Bezeichner werden mittels { Bezeichner } referenziert.

Zahl {Ziffer}+

Lexikalische Analyse

Prof. Dr. U. Göhner/HS Kempten 59

Übung 8.7:

```
%{
#include <stdio.h>
#include <stdlib.h>
}%
```

ziffer [0-9]

letter [a-zA-Z]

sign [+]

zeichen []

%%

"+" {printf("Operator + gefunden!\n"); }

"*" {printf("Operator * gefunden!\n"); }

"=" {printf("Zuweisung = gefunden!\n"); }

"if" {printf("Schlüsselwort if gefunden!\n"); }

"while" {printf("Schlüsselwort while gefunden!\n"); }

{sign}?{ziffer}+ {printf("Ganzzahl gefunden! Zahlenwert=%i \n",atoi(yytext)); }

%%

yywrap()

{

return 0;

};

void main()

{

do { yylex (); }

while (! feof(stdin)); }

1. Teil

2. Teil

3. Teil

Systematische Scanner-Implementierung

Aufbau im Detail:

Zweiter Teil:

Folge von Zeilen der Form:

regulärer Ausdruck {Aktionen }

Aktionen sind C++-statements, die ausgeführt werden, wenn regulärer

Ausdruck erkannt wird

regulärer Ausdruck |

Aktion

Aktion wird in der nächsten Zeile definiert

Die Aktionen werden unverändert in lex.yy.c übernommen.

Im allgemeinen folgt in der Aktion zum Abschluß ein

"return (<TOKEN>);"

Lexikalische Analyse

Prof. Dr. U. Göhner HS Kempten 60

Unix - Notation:

Bereichs - Angaben [0-9] 12-für 0-9

Außerklammerung: [a,b,c,f] 1 Zeichen, das a,b,c oder f ist

Außerklammerung von Zeichen: [a-zA-Z] 1 Zeichen, das klein- oder Großbuchstabe ist

.,*,+ sind Sonderzeichen, falls benötigt

in "x" oder /*