Universität Karlsruhe (TH)

Lehrstuhl für Programmierparadigmen

Sprachtechnologie und Compiler WS 2009/2010 http://pp.info.uni-karlsruhe.de/
Dozent: Prof. Dr.-Ing. G. Snelting snelting@ipd.info.uni-karlsruhe.de
Übungsleiter: Sebastian Buchwald sebastian.buchwald@kit.edu

Übungsblatt 14 | Ausgabe: 4.2.2010 | Besprechung: 10.2.2010

Aufgabe 1: *Codeerzeugung*

Konstruieren Sie unter der Voraussetzung, dass alle nicht konstanten Operanden im Arbeitsspeicher stehen, Syntaxbäume für die folgenden Anweisungen:

1. $x = a * b + c * d$;
2. $x[i] = y[j] * z[k]$;
3. $x = x + 1$;

Für die Multiplikation und den Arrayzugriff sind zusätzliche Regeln notwendig, stellen Sie diese auf.

Verwenden Sie die Verfahren zum Neuschreiben von Bäumen aus den Zusatzfolien der Vorlesung, um Code für die einzelnen Anweisungen zu erzeugen.

Aufgabe 2: LR-*Parser zur Codegenerierung*

Anstatt Baumersetzungsverfahren kann man auch einen LR-Parser zur Erzeugung von Maschinencode nutzen. Dazu werden die Ausdrücke der Zwischensprache in einer normalisierte Präfixdarstellung als String dargestellt.

Die Codeerzeugung geschieht dann mit Hilfe eines LR-Parsers, dessen Aktionen bei der Komplettierung einer Regel das Ausgeben von Quelltext ist. Passende Regeln zu den Ersetzungen der Vorlesungsfolien sehen dann z.B. so aus:

1	R_i	$\rightarrow$	C_a	{ LD Ri, #a }
2	R_i	$\rightarrow$	M_x	{ LD Ri, x }
3	M	$\rightarrow$	$= M_x\ R_i$	{ ST x, Ri }
4	M	$\rightarrow$	$=$ **ind** $R_i\ R_j$	{ ST *Ri, Rj }
5	R_i	$\rightarrow$	**ind** $+\ C_a\ R_j$	{ LD Ri, a(Rj) }
6	R_i	$\rightarrow$	$+\ R_i$ **ind** $+\ C_a\ R_j$	{ ADD Ri, Ri, a(Rj) }
7	R_i	$\rightarrow$	$+\ R_i\ R_j$	{ ADD Ri, Ri, Rj }
Z1	R	$\rightarrow$	**sp**	
Z2	M	$\rightarrow$	**m**	

Das Terminal **m** steht dabei für einen bestimmten Ort im Arbeitsspeicher (z.B. den Platz einer globalen Variablen). Das Stackregister wird durch das Terminal **SP** gekennzeichnet. Das Terminal **c** steht für Konstanten.

Der Ausdruck $a[i] = b + 1$ sieht damit zum Beispiel so aus:

$$\texttt{= ind + +}\ C_a\ \texttt{sp ind +}\ C_i\ \texttt{sp +}\ M_b\ C_1$$

2.1 *Anwendung*

- Wie sehen die Anweisungen aus der letzten Aufgabe in Präfixform aus?
- Schreiben Sie die Zusatzregel aus der letzten Aufgabe als Grammatikregel auf!
- Wenden Sie das Verfahren auf die Ausdrücke an!

2.2 *Regeln schreiben*

Erweitern Sie das Verfahren auf while-Anweisungen.