Prof. Dr. M. Jäger
FB MNI

Compilerbau - Praktikumsaufgabe 6 „Code-Generator für SPL“

Zunächst ein Hinweis: Der Code, den Ihr Compiler erzeugt, muss in keiner Hinsicht so aussehen, wie der Code der Referenzimplementierung. Sie dürfen sich zwar gerne diesen Code ansehen; versuchen Sie aber nicht, unter allen Umständen gleichen Code zu erzeugen. Wichtigstes Ziel in dieser Phase ist korrekter Code, d.h. Code, der genau das Verhalten zeigt, das laut Sprachdefinition von ihm gefordert wird.

Schritt 1 (Informieren)

Studieren Sie gründlich das Kapitel zum Assembler-Code-Generator für SPL im Praktikumsskript.

Schritt 2 (Code-Erzeugung anhand des AST)

Programmieren Sie einen rekursiven Durchgang durch die abstrakte Syntax, so dass zu jedem Knotentyp der passende Puck-Assemblercode ausgegeben wird.

Entwerfen Sie zunächst jeweils ein ganz kurzes SPL-Programm, das den ins Auge gefassten Knotentyp beim Übersetzen produziert (so etwas sollte eigentlich aus Lösungen zu früher gestellten Aufgaben schon zur Verfügung stehen). Programmieren Sie dann die Codegenerierung für diesen Knotentyp. Prüfen Sie zuletzt, ob der erzeugte Code für Ihr SPL-Programm korrekt ist.

Hier ist ein Vorschlag für die Reihenfolge der Knotentypen:

- *StmList* (außer dem rekursiven Abstieg ist hier nichts zu tun)
- *IntExp*
- *OpExp* (nur Arithmetik, keine Vergleiche)
- *SimpleVar* (ohne Berücksichtigung von Referenzvariablen)
- *VarExp*
- *AssignStm*
- *ArrayVar* (nicht vergessen: Indexgrenzen-Überprüfung, s.u.)
- *WhileStm*
- *OpExp* (nur Vergleiche, keine Arithmetik)
- *IfStm*
- *CallStm* (ohne Berücksichtigung von Referenzvariablen)
- *ProcDec* mit folgenden Teilaufgaben:
  - Framegröße berechnen
  - Prozedur-Prolog ausgeben
  - Code für Prozedurkörper erzeugen
  - Prozedur-Epilog ausgeben

Hinweise:

- Falls die Indexgrenzen-Überprüfung fehlschlägt, lassen Sie den Code zum Label `_indexError` verzweigen. Das ist eine Bibliotheksfunktion, die das Programm mit einer Fehlermeldung abbricht.
- Denken Sie an die Überprüfung, ob genügend Register für Hilfsvariable zur Verfügung stehen. Brechen Sie die Übersetzung ggf. mit einer geeigneten Fehlermeldung ab.

Schritt 3 (Fehler bei Referenzparametern beheben)

Untersuchen Sie, an welcher Stelle ein Programm, das Referenzparameter benutzt, noch fehlerhaft ist. Ergänzen Sie den Code-Generator so, dass auch SPL-Programme mit Referenzparametern richtig übersetzt werden.

Schritt 4 (Testen)

Überzeugen Sie sich von der Korrektheit des erzeugten Codes für die etwas größeren Testprogramme (Queens, Sierpinski, Sortieren eines Arrays).