XV. Übungsblatt "Informatik I" WS 95/96

Prof. Gert Smolka, Christian Schulte

Ausgabe: 9. Februar 1996

Abgabe: 16. Februar 1996 vor der Vorlesung (Programme per E-Mail)

Aufgabe XV.1 (Interpretierer für M (8)) Schreiben Sie einen Interpretierer für die Sprache M. Implementieren Sie diesen als Objekt, das die folgenden Nachrichten versteht:

- `load` lade eine Liste von Zahlen in den Zustand der Maschine, beginnend bei Adresse 0.
- `exec` starte Ausführung bei Adresse 0.
- `trace` starte Ausführung bei Adresse 0, gib dabei bei jeder ausgeführten Instruktion sowohl Programmzähler als auch Instruktionssymbol aus.
- `set-z!` setze an einer Adresse einen Wert.
- `get-z` liefere Wert an einer Adresse zurück.
- `get-pc` liefere Programmzähler zurück.

Stellen Sie dabei den Zustand als ein Scheme-Vektor von Zahlen dar. Denken Sie auch daran, daß die Fehlerbehandlung für M genau zu implementieren ist!
Hinweise: Es ist hilfreich, wenn Sie zwei Hilfsvektoren benutzen: einer soll Instruktionscodes auf ihre Symbole abbilden, der andere Instruktionscodes auf die sie implementierenden Prozeduren. Auf der WWW-Seite

`http://ps-www.dfki.uni-sb.de/courses/info-i/programme.html`

finden Sie das Fakultätsprogramm aus der Vorlesung zum Testen Ihres Interpretierers! Die Spezifikation von M finden Sie auf der Seite

`http://ps-www.dfki.uni-sb.de/courses/info-i/maschine-m.ps`

Aufgabe XV.2 (Fibonacci-Zahlen in M (3)) Schreiben Sie ein M-Programm, das zu gegebenem n die n-te Fibonacci Zahl berechnet.

Aufgabe XV.3 (Indirekte Sprünge in M (2)) Die Maschine M besitzt die Instruktion jump-i a für einen indirekten Sprung nach $Z(a)$. Geben Sie ein M-Programm ohne jump-i Instruktionen an, das auch nach $Z(a)$ springt. Gehen Sie dabei davon aus, daß das Programm an der Adresse sa beginnt.

Aufgabe XV.4 (Übersetzer von A nach M (6)) Auf der WWW-Seite

`http://ps-www.dfki.uni-sb.de/courses/info-i/ma-teile-a.scm`

finden Sie ein fast vollständiges Programm für die Maschine A. Dieses Programm implementiert ein Objekt, das im Zustand einen Interpretierer für M enthält. Beim Laden eines A-Programmes wird dieses Programm in ein M-Programm übersetzt und kann dann von dem M-Interpretierer ausgeführt werden. Vervollständigen Sie das Programm noch um die folgenden Teile:

1. Implementieren Sie die Übersetzung von `push` und `pop` Befehlen.

2. Implementieren Sie die Übersetzung des `goto` Befehls.

3. Implementieren Sie die Übersetzung des `branch` Befehls.

Aufgabe XV.5 (Optimierung der Übersetzung nach M (3)) Sie werden feststellen, daß bei der Übersetzung für einen Befehl `(assign` r i`)`, wobei r ein Register und i eine Zahl ist, zwei M-Instruktionen erzeugt werden.

1. Welche M-Instruktionen werden erzeugt?

2. Wie kann man mit nur einer M-Instruktion auskommen? Schreiben Sie die Übersetzung dementsprechend um.

3. Wo tritt dasselbe Phänomen nochmal auf? Beheben Sie auch diese Ineffizienz.

Aufgabe XV.6 (Münzwechseln in A (3)) Schreiben Sie ein A-Programm zum Wechseln von Münzen. (siehe auch Aufgabe 4 auf dem IV. Übungsblatt).