Prof. Dr. R. Loogen, M. Dieterle
Fachbereich Mathematik und Informatik
Hans-Meerwein-Straße
D-35032 Marburg

3. Februar 2010

Philipps Universität Marburg

Übungen zu „Grundlagen des Compilerbau“, Winter 2009/10

Nr. 11, Abgabe der Aufgaben: 10. Februar 2010 vor der Vorlesung

---

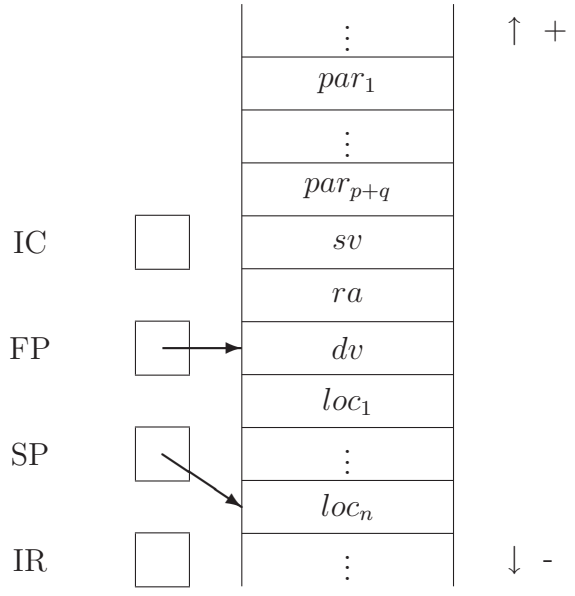
Hinweis: Dies ist das letzte Übungsblatt. Es waren 144 + 2 Punkte zu erreichen, die 50% Grenze liegt bei 72 Punkten.

Aufgaben

11.1 MPP-Maschinenzustände

4 Punkte

Die untenstehende Abbildung skizziert den Zustandsraum der Maschine MPP. Überprüfen und begründen Sie, ob die angegebenen Maschinenzustände $Z_1 - Z_4$ bei der Programmausführung eines von PSPP nach MPP übersetzten Programms entstehen können. Dabei sei 200 die feste Adresse des Kellerboden.



	Z_1	Z_2	Z_3	Z_4
IC				
FP	184	183	179	180
SP	183	184	179	179
IR				
Stack 200	0	0	0	0
199	0	0	0	0
198	0	0	0	0
197	4	4	4	4
196	198	198	198	198
195	3	3	3	3
194	198	198	198	198
193	8	8	8	8
192	2	2	2	2
191	194	194	194	194
190	100	100	100	100
189	194	194	194	194
188	1	1	1	1
187	1004	1004	1004	1004
186	189	194	194	189
185	55	55	55	55
184	194	189	189	189
183	20	20	20	1304
182			1304	194
181			189	25
180			75	184
179			184	1055

Bitte wenden!

11.2 **Display-Technik** 5 Punkte

(a) Beschreiben Sie, wie die Maschine MP konzeptionell erweitert werden kann um mit lokalen Displays zu arbeiten. Welche Konsequenzen hat dies für die Übersetzung von PSP-Programmen.

Im Skript auf Seite 98 ist ein PSP-Beispielprogramm und der durch die Abarbeitung des übersetzten MP-Programms bis zum zweiten Aufruf von A entstehende Prozedurkeller abgebildet.

(b) Skizzieren Sie den entsprechenden Prozedurkeller, wie er bei Benutzung lokaler Displays entstehen würde.

(c) Globale Displays können mit Hilfe eines Static Link Arrays realisiert werden. Beim Aufruf einer Prozedur muss ein Display-Eintrag gesichert werden. Schreiben Sie die Zustände des Prozedurkellers und des SLA (jeweils nach Funktionsaufrufen) bis nach dem zweiten Aufruf von A auf.

11.3 **Basisblockdarstellung** 3 Punkte

Bestimmen Sie eine Basisblockdarstellung zu folgender Drei-Adress-Codesequenz:

```
    i  = m-1
    j  = n
    t1 = 4*n
    v  = a[t1]
L1: i  = i+1
    t2 = 4*i
    t3 = a[t2]
    if t3<v goto L1
L2: j  = j-1
    t4 = 4*j
    t5 = a[t4]
    if t5>v goto L2
    if i>=j goto L3
    t6 = 4*i
    x  = a[t6]
    t7 = 4*i
    t8 = 4*j
    t9 = a[t8]
    a[t7] = t9
    t10 = 4*j
    a[t10] = x
    goto L1
L3: t11 = 4*i
    x = a[t11]
    t12 = 4*i
    t13 = 4*n
    t14 = a[t13]
    a[t12] = t14
    t15 = 4*n
    a[t15] = x
```