Aufgabenblatt zur 7. Übung

Zeitraum: 23.05. bis 27.05.2011

Bitte beachten: Am Montag, dem 30.05.11, findet in der 1. DS im HSZ/Audimax die Ersatzvorlesung für Freitag, dem 03.06.11, statt.

1. Aufgabe: (AGS 12.1)*

Gegeben sei folgendes C_0-Programm Max:

```
/* Max */
#include <stdio.h>
int main() {
  int a, b, max;
  scanf("%i", &a);
  scanf("%i", &b);
  if (a > b) max = a;
  else max = b;
  printf("%d", max);
  return 0;
}
```

(a) Berechnen Sie schrittweise das baumstrukturierte Programm $bMax_0 = \underline{trans}(Max)$ mit Hilfe der in der Vorlesung/Skript angegebenen Übersetzungsfunktionen.

(b) Wandeln Sie $bMax_0$ in ein Programm Max_0 mit linearisierten Adressen um und berechnen Sie $\mathcal{P}[\![Max_0]\!](5 : 7)$. Dokumentieren Sie den Zustand der AM_0 nach Ausführung jedes Befehls.

2. Aufgabe: (AGS 12.6*)

(a) Gegeben sei folgendes C_0–Programm:

```
/*ggT*/
#include <stdio.h>

int main()
{ int a,b,h;
  scanf("%i",&a);
  scanf("%i",&b);
  if (b>0) {
   while (b>0)
    {h=a % b; a=b; b=h;}
   printf("%d",a);
  }
  return 0;
}
```

Übersetzen Sie obiges C_0–Programm in ein AM_0–Programm ggT_0 mit linearisierten Adressen. Zwischenschritte brauchen Sie nicht anzugeben.

(b) Folgendes AM_0–Programm sei gegeben:

```
1:  LIT 0;
2:  STORE 2;
3:  READ 1;
4:  LOAD 1;
5:  LIT 0;
6:  GT;
7:  JMC 17;
8:  LOAD 1;
9:  LIT 2;
10: SUB;
11: STORE 1;
12: LOAD 2;
13: LIT 1;
14: ADD;
15: STORE 2;
16: JMP 4;
17: WRITE 2;
```

Lassen Sie dieses Programm auf der AM_0 mit der Startkonfiguration $(10, 2:2, [\,1/2, 2/0\,], \varepsilon, \varepsilon)$ schrittweise ablaufen.

(c) Geben Sie für das unter **(b)** gegebene AM_0–Programm für den Befehlsbereich 4 bis 17 die zugehörigen C_0–Statements an, deren Übersetzungen zu dieser AM_0–Befehlsfolge geführt haben. Nehmen Sie dabei an, dass der Speicherplatz i für die Variable `xi` vergeben wurde.

Zusatzaufgabe: (AGS 12.7*)

```
/* prog */
#include <stdio.h>
int main() {
  int x, y, z;
  z = 1;
  scanf("%i", &x);
  scanf("%i", &y);
  while (x > y) {
    z = y * z;
    x = x - 1;
  }
  printf("%d", z);
  return 0;
}
```

Übersetzen Sie obiges C_0–Programm in ein AM_0–Programm $prog_0$ mit linearisierten Adressen. Zwischenschritte brauchen Sie keine anzugeben.

(b) Folgender Ausschnitt aus einem AM_0–Programm sei gegeben:

```
9:  ...
10: LOAD 1;
11: LOAD 2;
12: LT;
13: JMC 19;
14: LOAD 2;
15: LOAD 1;
16: SUB;
17: STORE 1;
18: JMP 21;
19: LOAD 2;
20: STORE 1;
21: ...
```

Lassen Sie dieses Programm auf der AM_0 mit der Ausgangskonfiguration
$(10, \varepsilon, [1/2, 2/3, 3/0], \varepsilon, \varepsilon)$ schrittweise ablaufen bis der Befehlszähler größer bzw. gleich 21 ist.

(c) Geben Sie für den unter (b) gegebenen Ausschnitt aus einem AM_0–Programm die zugehörigen C_0–Statements an, deren Übersetzung (bis auf eine eventuelle Verschiebung der Befehlsadressen) zu dieser AM_0–Befehlsfolge führt. Vergeben Sie dabei für den Speicherplatz i die Variable `xi`.