
Programmierung

Aufgabe 1 (AGS 14.1 ★)

Gegeben sei folgendes C₀-Programm *Max*:

```

1  #include <stdio.h>
2
3  int main() {
4      int a, b, max;
5      scanf("%i", &a);
6      scanf("%i", &b);
7      if (a > b)
8          max = a;
9      else max = b;
10     printf("%d", max);
11     return 0;
12 }
```

- (a) Berechnen Sie *schrittweise* das baumstrukturierte Programm $bMax_0 = \text{trans}(Max)$ mit Hilfe der in der Vorlesung angegebenen Übersetzungsfunktionen. Dokumentieren Sie dabei jeden rekursiven Funktionsaufruf.
- (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 jedem ausgeführten Befehl.

Aufgabe 2 (AGS 14.14)

- (a) Gegeben sei folgendes C₀-Programm.

```

1  #include <stdio.h>
2
3  int main() {
4      int x1, x2;
5      scanf("%i", &x1);
6      scanf("%i", &x2);
7      while (x1 > 0){
8          x1 = x2 - x1;
9          if (x2 > x1)
10             x2 = x2 / 2;
11     }
12     printf("%d", x1);
13     return 0;
14 }
```

Übersetzen Sie das Programm mittels *trans* in AM_0 -Code mit linearen Adressen. Geben Sie nur das Endergebnis der Übersetzung (keine Zwischenschritte) an!

- (b) Gegeben sei der folgende Ausschnitt aus einem AM_0 -Programm.

```

3:  LOAD 2;      6:  JMC 14;      9:  LIT 2;      12: STORE 2;
4:  LIT 5;       7:  LOAD 1;     10: MUL;       13: JMP 3;
5:  LT;          8:  LOAD 2;     11: ADD;       14: WRITE 1;
```

Erstellen Sie ein Ablaufprotokoll für dieses Programmfragment, bis die AM_0 terminiert. Die Startkonfiguration ist $(7, \varepsilon, [1/3, 2/1], \varepsilon, \varepsilon)$.

Zusatzaufgabe 1 (AGS 14.7 *)

(a) Gegeben sei folgendes C₀-Programm:

```
1  #include <stdio.h>           8  while (x > y) {
2                               9      z = y * z;
3  int main() {                10     x = x - 1;
4      int x, y, z;            11 }
5      z = 1;                  12 printf("%d", z);
6      scanf("%i", &x);        13 return 0;
7      scanf("%i", &y);        14 }
```

Übersetzen Sie das Programm mittels *trans* in AM₀-Code mit linearen Adressen. Geben Sie nur das Ergebnis der Übersetzung (keine Zwischenschritte) an!

(b) Folgender Ausschnitt aus einem AM₀-Programm sei gegeben:

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

Erstellen Sie ein Ablaufprotokoll für dieses Programmfragment bis der Befehlszähler größer oder gleich 21 ist. Die Startkonfiguration ist $(10, \varepsilon, [1/2, 2/3, 3/0], \varepsilon, \varepsilon)$.

(c) Geben Sie für den im Aufgabenteil (b) gegebenen Ausschnitt eines AM₀-Programms die zugehörigen C₀-Statements an, deren Übersetzung (bis auf eine eventuelle Verschiebung der Befehlsadressen) zu dieser AM₀-Befehlsfolge führt. Vergeben Sie dabei für den Speicherplatz *i* die Variable *xi*.