Programmierung

Aufgabe 1 (AGS 14.1 ⋆)

Gegeben sei folgendes C_0-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 C_0-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 Ablaufsprotokoll für dieses Programmfragment, bis die AM_0 terminiert. Die Startkonfiguration ist $(7, \varepsilon, [1/3, 2/1], \varepsilon, \varepsilon)$.

Zusatzaufgabe 1 (AGS 14.6)

(a) Gegeben sei folgendes C_0-Programm:

```
1  #include <stdio.h>
2
3  int main() {
4    int a, b, h;
5    scanf("%i", &a);
6    scanf("%i", &b);
7    if (b > 0) {
8      while (b > 0) {
9        h = a % b;
10       a = b;
11       b = h;
12     }
13     printf("%d", a);
14   }
15   return 0;
16 }
```

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

(b) Folgendes AM_0-Programm sei gegeben:

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

Erstellen Sie ein Ablaufsprotokoll für dieses Programmfragment, bis die AM_0 terminiert. Die Startkonfiguration ist $(10, 2 : 2, [1/2, 2/0], \varepsilon, \varepsilon)$.