
Programmierung

Aufgabe 1 (AGS 15.17)

(a) Gegeben sei folgendes Fragment eines C_1 -Programms mit den Funktionen f und g :

```
while (*p > i) { f(p); i = i + 1; }
p = &i;
```

Übersetzen Sie die Sequenz dieser Statements in entsprechenden AM_1 -Code mit baumstrukturierten Adressen (mittels *stseqtrans*). Sie müssen keine Zwischenschritte angeben. Nehmen Sie an, die `while`-Anweisung sei das zweite Statement in g , und es sei

$$tab_g = \{f/(proc, 1), g/(proc, 2), i/(var, lokal, 1), p/(var-ref, -2)\} .$$

die Symboltabelle im Aufruf von *stseqtrans* auf die Statements in g .

(b) Gegeben sei folgender AM_1 -Code:

```
1:  INIT 1;           10:  MUL;           19:  READ(global,1);
2:  CALL 18;         11:  STOREI(-3);   20:  LOADA(global,1);
3:  INIT 0;          12:  LOAD(lokal,-2); 21:  PUSH;
4:  LOAD(lokal,-2);  13:  LIT 1;        22:  LOAD(global,1);
5:  LIT 0;           14:  SUB;          23:  PUSH;
6:  GT;              15:  STORE(lokal,-2); 24:  CALL 3;
7:  JMC 17;          16:  JMP 4;        25:  WRITE(global,1);
8:  LIT 2;           17:  RET 2;        26:  JMP 0;
9:  LOADI(-3);       18:  INIT 0;
```

Dokumentieren Sie 12 Schritte der AM_1 mit der Startkonfiguration $\sigma = (22, \varepsilon, 1 : 3 : 0 : 1, 3, \varepsilon, \varepsilon)$.

Aufgabe 2 (AGS 15.16 a, AGS 15.18 b)

(a) Gegeben sei folgendes Fragment eines C_1 -Programms:

```
#include <stdio.h>
int x, y;
void f(...) {...}

void g(int a, int *b) {
    int c;
    c = 3;
    if (c == *b) while (a > 0) f(&a, b);
}

void main() {...}
```

Übersetzen Sie die Sequenz der Statements im Rumpf von g in entsprechenden AM_1 -Code mit baumstrukturierten Adressen (mittels *stseqtrans*). Sie brauchen keine Zwischenschritte anzugeben. Geben Sie zunächst die benötigte Symboltabelle tab_g an.

(b) Gegeben sei folgender AM_1 -Code:

```

1:  INIT 1;           8:  LOADI(-2);       15:  LOADA(global, 1);
2:  CALL 13;         9:  LIT 2;          16:  PUSH;
3:  INIT 0;          10:  DIV;           17:  CALL 3;
4:  LOADI(-2);       11:  STOREI(-2);    18:  WRITE(global, 1);
5:  LIT 2;           12:  RET 1;         19:  JMP 0;
6:  GT;              13:  INIT 0;
7:  JMC 12;          14:  READ(global, 1);

```

Erstellen Sie ein Ablaufprotokoll der AM_1 , indem Sie sie schrittweise ablaufen lassen, bis die Maschine terminiert. Die Anfangskonfiguration sei $(14, \varepsilon, 0 : 0 : 1, 3, 4, \varepsilon)$. Sie müssen nur Zellen ausfüllen, deren Wert sich im Vergleich zur letzten Zeile geändert hat.

Zusatzaufgabe 1 (AGS 15.27)

(a) Gegeben sei folgendes Fragment eines C_1 -Programms.

```

1  #include <stdio.h>           8  void g(int *p) {
2                               9   int l;
3  int x;                       10  if (x == 0) {
4                               11   x=x+1; g(&l);
5  void g(int *p);              12  }
6  void f(...) {...}           13  f(*p);
7  void main() {...}           14  }

```

Übersetzen Sie die Statements im Rumpf von g (Zeilen 10–13) mittels *stseqtrans* in AM_1 -Code mit baumstrukturierten Adressen. Geben Sie keine Zwischenschritte an. *Geben Sie zunächst die dazu benötigte Symboltabelle an.*

(b) Gegeben sei folgender AM_1 -Code:

```

1:  INIT 1;           9:  RET 2;           17:  PUSH;
2:  CALL 10;          10:  INIT 0;         18:  LOADA(global, 1);
3:  JMP 0;            11:  READ(global, 1); 19:  PUSH;
4:  INIT 1;           12:  LOAD(global, 1); 20:  CALL 4;
5:  LOAD(lokal, -3);  13:  LIT 0;          21:  JMP 12;
6:  LIT 1;            14:  GE;             22:  WRITE(global, 1);
7:  SUB;              15:  JMC 22;         23:  RET 0;
8:  STOREI(-2);       16:  LOAD(global, 1);

```

Führen Sie jede der drei AM_1 -Konfigurationen um jeweils vier Schritte weiter:

- $(11, \varepsilon, 0 : 3 : 0, 3, 5, \varepsilon)$,
- $(18, \varepsilon, 5 : 3 : 0 : 5, 3, \varepsilon, \varepsilon)$,
- $(7, 1 : 5, 5 : 3 : 0 : 5 : 1 : 21 : 3 : 0, 7, \varepsilon, \varepsilon)$.

Sie müssen nur Zellen ausfüllen, deren Wert sich im Vergleich zur vorherigen Zeile geändert hat.