

9. Übung Programmierung

Dominic Deckert

16. Juni 2017

Previously on ...

- ▶ C_1
- ▶ AM_1
 - ▶ Aktivierungsblöcke
 - ▶ Adressierungsarten
 - ▶ PUSH, CALL, RETURN

Übersetzung

Symboltabelle: verwaltet lokale Adressen der Variablen & “Adressen” der Programme

Baumstrukturierte Adressen: Adresse der Struktureinheit . Subadresse

Struktureinheit: Programm / while / if / atomarer Befehl

trans

Genaue Übersetzung siehe Extrablatt

Zu beachten: - Erstellung mehrerer (lokaler) Symboltabellen

- getrennte Übersetzung der Unterprogramme
- Parameter mit PUSH übergeben, Anzahl lokaler Variablen mit INIT festlegen
- mit RETURN lokale Variablen löschen

Formale Übersetzung

Formale Übersetzung siehe Extra

8.2

INIT 0; CALL 2; JMP 0;

1 :INIT 1; LIT 2; STORE(lo, 1);

LOAD(lo, -2); LIT 1; GT; JMC 1.2.1;

1.2.2.1.1 :LOAD(lo, -2); LOAD(lo, 1);

MOD; LIT 0; ADD; NE; JMC 1.2.2.1.2;

LOAD(lo, 1); LIT 1; ADD; STORE(lo, 1); JMP 1.2.2.1.1

8.2

```
1.2.2.1.2 : LOAD(lo, -2); LOAD(lo, 1); DIV;  
            STORE(lo, -2); WRITE(lo, 1); LOAD(lo, -2);  
            PUSH; CALL 1;  
1.2.1 : RET 1;
```

```
2 : INIT 1; READ(lo, 1); LOAD(lo, 1);  
    PUSH; CALL 1; RET 0;
```

8.3 a)

Übersetzung:

8.3 a)

Übersetzung:

```
(1.3)  1.3.3  LOAD (lokal, 1), LOADI -2,  
          LT, JC 1.3.1  
      1.3.2  LOAD (lokal, 2), LIT 2,  
          MULT, STORE (lokal, 2),  
          LOAD (lokal, 2), PUSH,  
          LOAD (lokal, -2), PUSH,  
          CALL 2  
          JMP 1.3.3  
(1.4)  1.3.1  LOAD (lokal, 2), STOREI -3
```

Fragen?

Fragen zur 8. Übung?

3 a)

Übersetzung:

3 a)

Übersetzung:

(2.2) **2.2.3** LOADI -2, LOAD(lokal, 1)
GT, JC **2.2.1**

2.2.2 LOAD (lokal, -2), PUSH,
CALL 1,
LOAD (lokal, 1), LIT 1,
ADD, STORE (lokal, 1),
JMP **2.2.3**

(2.3) **2.2.1** LOADA (lokal, 1), STORE (lokal, -2)

3 b)

BZ	DK	LK	REF	In	Out
22	ε	1:3:0:1	3	ε	ε
23	1				
24	ε	1:3:0:1:1			
3		1:3:0:1:1:25:3	7		
4					

3 b)

BZ	DK	LK	REF	In	Out
5	1	1:3:0:1:1:25:3	7	ε	ε
6	0:1				
7	1				
8	ε				
9	2				

3 b)

BZ	DK	LK	REF	In	Out
10	1:2	1:3:0:1:1:25:3	7	ε	ε
11	2				
12	ε	2:3:0:1:1:25:3			
13	1	2:3:0:1:1:25:3	7	ε	ε
...					

Kalkül

Rechenvorschriften zum Nachweis von Programmeigenschaften
durch “Protokollieren” von Programm-Zuständen vor/nach Befehlen

Rechenvorschriften zum Nachweis von Programmeigenschaften
durch “Protokollieren” von Programm-Zuständen vor/nach Befehlen

- ▶ Sequenzregel
- ▶ Alternativregel
- ▶ Iterationsregel
- ▶ Schwächere Nach-/Stärkere Vorbedingung

Hinweise

Schleifeninvariante?

Hinweise

Schleifeninvariante? - Programmeigenschaften, die **vor** und **nach** Durchlauf durch eine Schleife gelten

Zusammensetzung (bei unseren Aufgaben): $SI = X \wedge Y$

X : relevanter Zusammenhang der Variablen

Y : Einschränkung der Variablen (bzgl. der Abbruchbedingung)

1a)

$X =$

1a)

$$X = (z = y \cdot (x - x_1))$$

$$Y = (x_1 \geq 0)$$

$$SI = (z = y \cdot (x - x_1)) \wedge (x_1 \geq 0)$$

1 b)

$$A = SI, B = SI \wedge \neg(x1 > 0)$$

$$C = SI, D = SI$$

$$E = SI \wedge (x1 > 0), G = SI$$

1c)

siehe Tafel

$SI =$

$A =, B =$

$C =, D =$

$E =$

$$SI = (b = a^{n-i}) \wedge (i \geq 0)$$

$$A = SI, B = SI \wedge \neg(i > 0)$$

$$C = SI, D = Si \wedge (i > 0)$$

$$E = \{b = b * a; i = i - 1; \}$$