

# Übersetzung von $C_0$ -Programmen

$C_0$	$AM_0$
<code>scanf("%i", &amp;x);</code>	READ HS-Adresse von x;
<code>printf("%d", x);</code>	WRITE HS-Adresse von x;
<code>z = 5</code>	LIT 5; STORE HS-Adresse von z;
<code>z = x / y</code>	LOAD HS-Adresse von x; LOAD HS-Adresse von y; DIV; STORE HS-Adresse von z;
<code>if (x &gt; y) then { ... }</code>	LOAD HS-Adresse von x; LOAD HS-Adresse von y; GT; JMC a.1; AM <sub>0</sub> -Code des then-Blocks a.1: weiterer Code

# Übersetzung von $C_0$ -Programmen

$C_0$	$AM_0$
<pre>if (x &gt; y) then { ... } else { ... }</pre>	<pre>LOAD HS-Adresse von x; LOAD HS-Adresse von y; GT; JMC a.1; AM<sub>0</sub>-Code des then-Blocks JMP a.3; a.1: AM<sub>0</sub>-Code des else-Blocks a.3: weiterer Code</pre>
<pre>while (x &lt; y) { ... }</pre>	<pre>a.1: LOAD HS-Adresse von x; LOAD HS-Adresse von y; LT; JMC a.2; AM<sub>0</sub>-Code des while-Blocks JMP a.1; a.2: weiterer Code</pre>

## Übung 2 (a)

READ 1;		LOAD 1;
READ 2;		STORE 3;
LOAD 1;		JMP 1.3.3;
LOAD 2;	1.3.1:	LOAD 2;
GT;		STORE 3;
JMC 1.3.1;	1.3.3:	WRITE 3;

## Übung 2 (b)

```
1:  READ  1;  
2:  READ  2;  
3:  LOAD  1;  
4:  LOAD  2;  
5:  GT;  
6:  JMC 10;
```

```
7:  LOAD  1;  
8:  STORE 3;  
9:  JMP 12;  
10: LOAD  2;  
11: STORE 3;  
12: WRITE 3;
```

## Übung 2 (b)

BZ	DK	HS	Inp	Out
( 1,	$\varepsilon$ ,	[ ],	5 : 7,	$\varepsilon$ )
( 2,	$\varepsilon$ ,	[1/5],	7,	$\varepsilon$ )
( 3,	$\varepsilon$ ,	[1/5, 2/7],	$\varepsilon$ ,	$\varepsilon$ )
( 4,	5,	[1/5, 2/7],	$\varepsilon$ ,	$\varepsilon$ )
( 5,	7 : 5,	[1/5, 2/7],	$\varepsilon$ ,	$\varepsilon$ )
( 6,	0,	[1/5, 2/7],	$\varepsilon$ ,	$\varepsilon$ )
( 10,	$\varepsilon$ ,	[1/5, 2/7],	$\varepsilon$ ,	$\varepsilon$ )
( 11,	7,	[1/5, 2/7],	$\varepsilon$ ,	$\varepsilon$ )
( 12,	$\varepsilon$ ,	[1/5, 2/7, 3/7],	$\varepsilon$ ,	$\varepsilon$ )
( 13,	$\varepsilon$ ,	[1/5, 2/7, 3/7],	$\varepsilon$ ,	7)

## Übung 3 (a)

```
1:  READ 2;  
2:  READ 3;  
3:  LIT 0;  
4:  STORE 1;  
5:  LOAD 1;  
6:  LOAD 3;  
7:  LT;  
8:  JMC 18;  
9:  LOAD 1;
```

```
10: LIT 1;  
11: ADD;  
12: STORE 1;  
13: LOAD 2;  
14: LOAD 2;  
15: MUL;  
16: STORE 2;  
17: JMP 5;  
18: WRITE 2;
```

## Übung 3 (b)

BZ	DK	HS	Inp	Out
( 1,	$\varepsilon$ ,	[ ],	0 : 1,	$\varepsilon$ )
( 2,	$\varepsilon$ ,	[1/0],	1,	$\varepsilon$ )
( 3,	$\varepsilon$ ,	[1/0, 2/1],	$\varepsilon$ ,	$\varepsilon$ )
( 4,	0,	[1/0, 2/1],	$\varepsilon$ ,	$\varepsilon$ )
( 5,	1 : 0,	[1/0, 2/1],	$\varepsilon$ ,	$\varepsilon$ )
( 6,	0 : 1 : 0,	[1/0, 2/1],	$\varepsilon$ ,	$\varepsilon$ )
( 7,	1 : 0,	[1/0, 2/1],	$\varepsilon$ ,	$\varepsilon$ )
( 8,	0,	[1/0, 2/1],	$\varepsilon$ ,	$\varepsilon$ )
( 5,	0,	[1/0, 2/1],	$\varepsilon$ ,	$\varepsilon$ )
( 6,	0 : 0,	[1/0, 2/1],	$\varepsilon$ ,	$\varepsilon$ )
( 7,	0,	[1/0, 2/1],	$\varepsilon$ ,	$\varepsilon$ )
( 9,	$\varepsilon$ ,	[1/0, 2/1],	$\varepsilon$ ,	$\varepsilon$ )
(10,	$\varepsilon$ ,	[1/0, 2/1],	$\varepsilon$ ,	1)

## Befehle der $AM_1$

Die Befehle der  $AM_0$  werden durch folgende Änderungen ergänzt:

Befehl	Auswirkung
LOAD( $b, o$ )	Lädt den Inhalt von Adresse $adr(r, b, o)$ auf den Datenkeller und inkrementiert den Befehlszähler.
STORE( $b, o$ )	Nimmt das oberste Element vom Datenkeller, speichert dieses an Adresse $adr(r, b, o)$ und inkrementiert den Befehlszähler.
WRITE( $b, o$ )	Schreibt den Inhalt an Adresse $adr(r, b, o)$ auf das Ausgabeband und inkrementiert den Befehlszähler.
READ( $b, o$ )	Liest das oberste Element vom Eingabeband, speichert es an Adresse $adr(r, b, o)$ und inkrementiert den Befehlszähler.

$b \in \{\text{global, lokal}\}$

$r$  : aktueller REF

$$adr(r, b, o) = \begin{cases} r + o & \text{wenn } b = \text{lokal,} \\ o & \text{wenn } b = \text{global.} \end{cases}$$



## Befehle der $AM_1$

Die Befehle der  $AM_0$  werden durch folgende Änderungen ergänzt:

Befehl	Auswirkung
LOADI( $o$ )	Ermittelt Wert ( $= b$ ) an Adresse $r + o$ , lädt den Inhalt von Adresse $b$ auf den Datenkeller und inkrementiert den Befehlszähler.
STOREI( $o$ )	Ermittelt Wert ( $= b$ ) an Adresse $r + o$ , nimmt das oberste Element vom Datenkeller, speichert dieses an Adresse $b$ und inkrementiert den Befehlszähler.
WRITEI( $o$ )	Ermittelt Wert ( $= b$ ) an Adresse $r + o$ , schreibt den Inhalt an Adresse $b$ auf das Ausgabeband und inkrementiert den Befehlszähler.
READI( $o$ )	Ermittelt Wert ( $= b$ ) an Adresse $r + o$ , liest das oberste Element vom Eingabeband, speichert es an Adresse $b$ und inkrementiert den Befehlszähler.
LOADA( $b, o$ )	Legt $adr(r, b, o)$ auf den Datenkeller und inkrementiert den Befehlszähler.

## Befehle der $AM_1$

Die Befehle der  $AM_0$  werden durch folgende Änderungen ergänzt:

Befehl	Auswirkung
PUSH	Legt oberstes Element vom Datenkeller auf den Laufzeitkeller und inkrementiert den Befehlszähler.
CALL $adr$	Ermittelt Befehlszählerwert, inkrementiert diesen und legt ihn auf den Laufzeitkeller. Setzt den Befehlszähler auf $adr$ . Legt den REF auf den Laufzeitkeller. Ändert REF auf die Länge des Laufzeitkellers.
INIT $n$	Legt $n$ -mal 0 auf den Laufzeitkeller.
RET $n$	Auf Laufzeitkeller alles nach REF-Zeiger löschen. Oberstes Element vom Laufzeitkeller nehmen und als REF setzen. Oberstes Element vom Laufzeitkeller nehmen und als Befehlszähler setzen. Weitere $n$ Elemente vom Laufzeitkeller löschen.

## Übung 4

BZ	DK	LK	REF	Inp	Out
( 14,	$\varepsilon$ ,	0 : 0 : 1,	3,	4,	$\varepsilon$ )
( 15,	$\varepsilon$ ,	4 : 0 : 1,	3,	$\varepsilon$ ,	$\varepsilon$ )
( 16,	1,	4 : 0 : 1,	3,	$\varepsilon$ ,	$\varepsilon$ )
( 17,	$\varepsilon$ ,	4 : 0 : 1 : 1,	3,	$\varepsilon$ ,	$\varepsilon$ )
( 3,	$\varepsilon$ ,	4 : 0 : 1 : 1 : 18 : 3,	6,	$\varepsilon$ ,	$\varepsilon$ )
( 4,	$\varepsilon$ ,	4 : 0 : 1 : 1 : 18 : 3,	6,	$\varepsilon$ ,	$\varepsilon$ )
( 5,	4,	4 : 0 : 1 : 1 : 18 : 3,	6,	$\varepsilon$ ,	$\varepsilon$ )
( 6,	2 : 4,	4 : 0 : 1 : 1 : 18 : 3,	6,	$\varepsilon$ ,	$\varepsilon$ )
( 7,	1,	4 : 0 : 1 : 1 : 18 : 3,	6,	$\varepsilon$ ,	$\varepsilon$ )
( 8,	$\varepsilon$ ,	4 : 0 : 1 : 1 : 18 : 3,	6,	$\varepsilon$ ,	$\varepsilon$ )
( 9,	4,	4 : 0 : 1 : 1 : 18 : 3,	6,	$\varepsilon$ ,	$\varepsilon$ )

## Übung 4

BZ	DK	LK	REF	Inp	Out
(10,	2 : 4,	4 : 0 : 1 : 1 : 18 : 3,	6,	$\epsilon$ ,	$\epsilon$ )
(11,	2,	4 : 0 : 1 : 1 : 18 : 3,	6,	$\epsilon$ ,	$\epsilon$ )
(12,	$\epsilon$ ,	2 : 0 : 1 : 1 : 18 : 3,	6,	$\epsilon$ ,	$\epsilon$ )
(18,	$\epsilon$ ,	2 : 0 : 1,	3,	$\epsilon$ ,	$\epsilon$ )
(19,	$\epsilon$ ,	2 : 0 : 1,	3,	$\epsilon$ ,	2)
( 0,	$\epsilon$ ,	2 : 0 : 1,	3,	$\epsilon$ ,	2)