

12. Übung - Programmierung

C_0, H_0, AM_0

SS 18

„C₀₀“

- ▶ keine Konstanten
- ▶ Variablendeklaration: $\text{int } x_1, \dots, x_m, m \geq 1$
- ▶ Leseanweisung nur am Anfang des Blocks
- ▶ Schreiben genau ein mal vor „return 0;“

$C_{00} - H_0$ Übersetzung

1. Funktionsblöcken von C_{00} (geschachtelt) Ablaufpunkte zuordnen
2. Jedem Ablaufpunkt wird eine H_0 - Funktion mit m Parametern
3. Der Funktionswert (rechts vom „=“) jeder Funktion ist der nächste erreichbare Ablaufpunkt mit entsprechend veränderten Parametern
4. Die übliche „main = do“ dranschrauben

$C_{00} - H_0$ Übersetzung

1. Funktionsblöcken von C_{00} (geschachtelt) Ablaufpunkte zuordnen

```
 $C_{00}$   
f1      x2 = 1;  
f2      while (x1 > 0)  
f21     {  
f211    x2 = x2 * x1;  
f212    x1 = x1 - 1;  
        }  
f3      printf („%d“, x2 );
```

$C_{00} - H_0$ Übersetzung

2. Jedem Ablaufpunkt wird eine H_0 - Funktion mit m Parametern

C_{00}	H_0
f1 $x2 = 1;$	f1 x1 x2 =
f2 while ($x1 > 0$)	f2 x1 x2 =
f21 {	f21 x1 x2 =
f211 $x2 = x2 * x1;$	f211 x1 x2 =
f212 $x1 = x1 - 1;$	f212 x1 x2 =
}	
f3 printf („%d“, x2);	f3 x1 x2 =

$C_{00} - H_0$ Übersetzung

3. der Funktionswert (rechts vom „=“) jeder Funktion ist der nächste erreichbare Ablaufpunkt mit entsprechend veränderten Parametern

C_{00}	H_0
f1 $x2 = 1;$	f1 x1 x2 = f2 x1 1
f2 while ($x1 > 0$)	f2 x1 x2 = if ($x1 > 0$) then f21 x1 x2 else f3 x1 x2
f21 {	f21 x1 x2 = f211 x1 x2
f211 $x2 = x2 * x1;$	f211 x1 x2 = f212 x1 ($x2 * x1$)
f212 $x1 = x1 - 1;$	f212 x1 x2 = f2 ($x1 - 1$) x2
}	
f3 printf („%d“, x2);	f3 x1 x2 = x2

$C_{00} - H_0$ Übersetzung

4. die übliche „main = do “dranschrauben

H_0

```
f1 x1 x2 = f2 x1 1
f2 x1 x2 = if (x1 > 0)
            then f21 x1 x2
            else f3 x1 x2
f21 x1 x2 = f211 x1 x2
f211 x1 x2 = f212 x1 (x2*x1)
f212 x1 x2 = f1 (x1-1) x2

f3 x1 x2 = x2
```

```
main = do x1 ← readLn
          print(f1 x1 0)
```