

Zur Erinnerung: H_0

- ▶ In H_0 dürfen nur tail-rekursive Funktionen über Int definiert werden. Zudem gelten gewisse syntaktische Einschränkungen. So müssen beispielsweise die formalen Parameter jeder Funktion der Reihe nach von x_1 bis x_k benannt sein.
- ▶ Tail-rekursive Funktionen enthalten auf der rechten Seite entweder
 - (i) keinen Funktionsaufruf,
 - (ii) genau einen Funktionsaufruf an der “äußersten Position” (d.h. nicht eingeschachtelt) oder
 - (iii) eine Fallunterscheidung, deren Zweige wiederum nach (i), (ii) oder (iii) aufgebaut sind.
- ▶ Dadurch kann auf die Verwendung eines Laufzeitkellers zur Verwaltung rekursiver Funktionsaufrufe verzichtet werden und damit das Programm in AM_0 übersetzt werden.
- ▶ Grundidee der Übersetzung: Parameter x_1 bis x_k werden im HS auf den Adressen 1, ..., k abgespeichert. Vor einem Funktionsaufruf werden die Parameter der aufzurufenden Funktion erst auf dem Datenkeller in der Reihenfolge 1, ..., k ausgerechnet, um dann mit `STORE k; ... STORE 1`; sukzessive im HS gespeichert zu werden.

Übung 3 (a)

```
h:  LOAD 3;
    LOAD 1;
    GT;
    JMC h.3;
    LOAD 2;
    LIT 1;
    SUB;
    STORE 1;
    WRITE 1;
    JMP 0;
```

```
h.3: LOAD 2;
      LOAD 1;
      LOAD 3;
      SUB;
      LOAD 2;
      STORE 3;
      STORE 2;
      STORE 1;
      JMP h;
```

Zusatzaufgabe 1 (a)

```
f:  LOAD 1;  
    LIT 0;  
    LE;  
    JMC f.3;  
    LOAD 2;  
    STORE 1;  
    WRITE 1;  
    JMP 0;
```

```
f.3: LOAD 1;  
     LIT 2;  
     DIV;  
     LOAD 2;  
     LOAD 1;  
     ADD;  
     STORE 2;  
     STORE 1;  
     JMP f;
```