

## Aufgabe 1

Die RAM soll um einen Befehl

$\text{cmpbs } [*n_1] [*n_2]$

zum Vergleich von Speicherbereichen erweitert werden. Der Beginn des ersten Bereichs ist an Adresse  $n_1$  abgelegt, der Beginn des zweiten an Adresse  $n_2$ . Die Länge der zu vergleichenden Bereiche steht im Akkumulator. Am Ende der Operation soll der Akkumulator den Wert 0 enthalten, falls sich die Bereiche unterscheiden; an Speicheradresse  $n_1$  und  $n_2$  stehen dann die Adressen, an denen der erste Unterschied auftritt. Enthalten beide Bereiche identische Werte, soll der Akkumulator den Wert 1 enthalten. Hat der Akkumulator zu Beginn den Wert 0 (die Bereiche sind leer), soll der Vergleich als erfolgreich gelten - der Akkumulator ist dann also auf 1 zu setzen.

## Lösung 1

Solange der Akkumulator größer 0 ist, dekrementiere den Akkumulator und inkrementiere die Speicheradressen:

$$\frac{s_\pi = \text{cmpbs } [*n_1] [*n_2] \quad \sigma(0) > 0 \quad \sigma(\sigma(n_1)) = \sigma(\sigma(n_2))}{(\pi, \alpha, \beta, \sigma) \vdash (\pi, \alpha, \beta, \sigma[0 \mapsto \sigma(0) - 1][n_1 \mapsto \sigma(n_1) + 1][n_2 \mapsto \sigma(n_2) + 1])}$$

Wenn die Speicheradressen  $n_1$  und  $n_2$  unterschiedliche Werte enthalten, der Akkumulator jedoch noch nicht bei 0 angekommen ist, soll der Akkumulator den Wert 0 bekommen:

$$\frac{s_\pi = \text{cmpbs } n_1 \ n_2 \quad \sigma(0) > 0 \quad \sigma(\sigma(n_1)) \neq \sigma(\sigma(n_2))}{(\pi, \alpha, \beta, \sigma) \vdash (\pi + 1, \alpha, \beta, \sigma[0 \mapsto 0])}$$

Hatte der Akkumulator zu Beginn den Wert 0 oder ist der Akkumulator bei 0 angekommen (also war der letzte Vergleich  $\sigma(\sigma(n_1)) = \sigma(\sigma(n_2))$  erfolgreich) soll der Akkumulator den Wert 1 bekommen:

$$\frac{s_\pi = \text{cmpbs } n_1 \ n_2 \quad \sigma(0) = 0}{(\pi, \alpha, \beta, \sigma) \vdash (\pi + 1, \alpha, \beta, \sigma[0 \mapsto 1])}$$

## Aufgabe 2

Schreiben Sie ein RAM-Programm, welches das Maximum zweier Zahlen  $A$  und  $B$  bestimmt.

Können Sie mithilfe der Schrittrelation  $\vdash$  beweisen, dass Ihr Programm korrekt ist?

## Lösung 2

Die folgende RAM bestimmt das Maximum von  $A$  und  $B$  durch abwechselndes Dekrementieren der beiden Werte. Sie springt zu einer Ausgabe, sowie einer der geladenen Werte 0 ist.

Dies wäre notwendig für eine RAM, dessen Hauptspeicher nur natürliche Zahlen speichern könnte  $N \rightarrow \mathbb{N}_0$ .

```
1 READ 1      : Lese A ein
2 READ 2      : Lese B ein
3 LOAD [1]
4 JZ 12       : Wenn Akkumulator 0 ist, ist B größer oder gleichgroß
5 SUB 1
6 STORE [1]
7 LOAD [2]
8 JZ 14       : Wenn Akkumulator 0 ist, ist A größer
9 SUB 1
10 STORE [2]
11 GOTO 3
12 WRITE 2
13 HALT
14 WRITE 1
15 HALT
```

Listing 1: Maximum von  $A$  und  $B$  wenn  $N \rightarrow \mathbb{N}_0$

Es sei darauf hingewiesen, dass eine RAM, wie wir sie definiert haben, auch negative Zahlen speichern und Gebrauch von dem Befehl **JGTZ** machen kann. Das macht die Bestimmung des Maximums deutlich einfacher:

```
1 READ 1      : Lese A ein
2 READ 2      : Lese B ein
3 LOAD [1]
4 SUB [2]     :  $\sigma(0) = A - B$ 
5 JGTZ 8      :  $\sigma(0) > 0 \Rightarrow A$  ist max
6 WRITE 2     :  $\sigma(0) < 0 \Rightarrow B$  ist max
7 HALT
8 WRITE 1
9 HALT
```

Listing 2: Maximum von  $A$  und  $B$  bei  $N \rightarrow \mathbb{Z}$

Die Korrektheit der RAM 2 lässt sich durch die Angabe der Menge aller Konfigurationen  $\text{Conf}(\mathcal{R}_{am})$  zeigen, wobei jede Konfiguration als Quadrupel der Form  $(\pi, \alpha, \beta, \sigma)$  angegeben ist.

```

-1      :  $(\pi, \alpha, \beta, \sigma)$ 
0      :  $(1, (A, B), (), \sigma_0)$  // Startkonfiguration
1  READ 1      :  $\vdash (2, (B), (), \sigma_0[1 \mapsto A])$ 
2  READ 2      :  $\vdash (3, (), (), \sigma_0[1 \mapsto A][2 \mapsto B])$ 
3  LOAD [1]    :  $\vdash (4, (), (), \sigma_0[0 \mapsto A][1 \mapsto A][2 \mapsto B])$ 
4  SUB [2]     :  $\vdash (5, (), (), \sigma_0[0 \mapsto A - B][1 \mapsto A][2 \mapsto B])$ 
5  JGTZ 8      :  $A - B > 0 \ ? \vdash (8, (), (), \sigma_0[\dots]) \ : \vdash (6, (), (), \sigma_0[\dots])$ 
6  WRITE 2     :  $\vdash (7, (), (2), \sigma_0[0 \mapsto A - B][1 \mapsto A][2 \mapsto B])$ 
7  HALT       :  $\vdash (0, (), (2), \sigma_0[0 \mapsto A - B][1 \mapsto A][2 \mapsto B])$ 
8  WRITE 1     :  $\vdash (8, (), (1), \sigma_0[0 \mapsto A - B][1 \mapsto A][2 \mapsto B])$ 
9  HALT       :  $\vdash (0, (), (1), \sigma_0[0 \mapsto A - B][1 \mapsto A][2 \mapsto B])$ 

```

Listing 3: Beweis von RAM 2

Für den Fall  $A - B > 0$  wird der Programmzähler auf 8 gesetzt und 1 auf das Ausgabeband geschrieben, für den Fall  $A - B \leq 0$  wird der Programmzähler auf 6 gesetzt und 2 auf das Ausgabeband geschrieben.