

Aufgabe 11

- (a) Es seien drei Integer-Werte in Dezimaldarstellung gegeben:

$$i_0 = \epsilon_0 \cdot p_l \dots p_2 p_1 p_0$$

$$i_1 = \epsilon_1 \cdot q_m \dots q_2 q_1 q_0$$

$$i_2 = \epsilon_2 \cdot r_n \dots r_2 r_1 r_0$$

Hierbei seien $\epsilon_0, \epsilon_1, \epsilon_2 \in \{-1, 1\}$ die Vorzeichen, $l, n, m \in \mathbb{N}$ die Ziffernlängen und ganz rechts stehen die am wenigsten signifikanten Ziffern. Da es $2^3 = 8$ mögliche Kombinationen von Vorzeichen gibt, können die Vorzeichen in der ersten Dezimalziffer $d_0 \in \{0, 1, \dots, 7\}$ gespeichert werden. Baue nun einen Integer-Wert x zusammen, der als erste Ziffer d_0 hat und danach abwechselnd Ziffern von i, j und k , also

$$x = \dots p_2 r_1 q_1 p_1 r_0 q_0 p_0 d_0$$

Ist die Ziffernlänge von einem der Ausgangswerte länger als die eines anderen, so muss der kürzere mit Nullen aufgefüllt werden. Zum Beispiel werden $i = 12$, $j = 345$, $k = 678$ codiert als

$$x = 8507426310 \quad ,$$

wobei die ganz rechte Null sich aus den Vorzeichen ergibt.

- (b) Für die Implementation in Java bietet es sich an, das oben angegebene Verfahren mit der Binärdarstellung statt der Dezimaldarstellung der Integers durchzuführen. Seien s_0, s_1, s_2 die Vorzeichenbits von i_0, i_1, i_2 , so baue den Integer x so zusammen:

$$x = \dots p_2 r_1 q_1 p_1 r_0 q_0 p_0 s_2 s_1 s_0$$

Diesmal stehen die Buchstaben p, q, r für die Bitfolgen der Ausgangswerte.

Da das Ergebnis x wieder in einem Integer mit 32 Bits gespeichert werden soll und 3 Bits für die Vorzeichen gebraucht werden, bleiben 29 Bits übrig. Das bedeutet bei gleichberechtigter Auteilung, dass von i_0 und i_1 je 10 Bits gespeichert werden können, von i_2 nur 9. Da in Java negative Integers nach dem Zweierkomplement kodiert werden, verbrauchen betraglich kleine negative Integers viele Bitstellen, z.B. wird -1 repräsentiert als einen Bitstring mit 32 Einsen. Bei der Speicherung von x in einem Integer werden die Bitfolgen auf Länge 9 bzw. 10 gekürzt. Um betraglich kleine Werte korrekt zu dekodieren, müssen folglich negative Werte links mit Einsen aufgefüllt werden, positive mit Nullen. Der Code ist an der entsprechenden Stelle kommentiert.

- (c) + (d) Nichts fällt auf, beide Tests sind erfolgreich :-)