Teil I

Lösungen zu den Aufgaben von Kap. 2

Kapitel 2

Aufgabe 2.1:

1. Überlegen Sie sich: wie viele verschiedene Möglichkeiten gibt es, mit Binärzahlen ein Byte zu füllen?

   $2^8 = 256$ Möglichkeiten

2. Wieviele Ziffern bräuchten Sie im Hexadezimalsystem, um die gleiche Anzahl Möglichkeiten zu erhalten?

   2 Ziffern, da $16^2 = 256$

3. Was folgern Sie daraus bzgl. der Vorteile des Hexadezimalsystems?

   Ein Byte im Binärsystem kann durch 2 Ziffern im Hexadezimalsystem ersetzt werden. Lange binäre Zeichenketten lassen sich also im Hexadezimalsystem sehr kompakt darstellen, wobei jeweils Gruppen mit 4 Bits durch eine Hexadezimal-Ziffer ersetzt werden kann.

   Beispiel 1: 1111 entspricht F, 1110 entspricht E etc.

   Beispiel 2: Farbangaben in HTML-Dateien, z.B. [...] <body bgcolor="#3399FF" [...]>. Die Hex-Ziffernpaare 33, 99 und FF definieren die Farbintensitäten für Rot, Grün und Blau zwichen 0 und 255. Hier: $(33)_{16} = (51)_{10}$, $(99)_{16} = (153)_{10}$, $(FF) = (255)_{10}$

Aufgabe 2.2 [1]:

1. Wie viele Stellen benötigt man, um die folgenden Zahlen als n-stellige Gleitpunktzahlen im Dezimalsystem darzustellen? x_1=0.00010001, x_2=1230001, x_3=$\frac{4}{5}$, x_4=$\frac{1}{3}$

   Man benötigt für x_1 5 Stellen, für x_2 7 Stellen, für x_3 1 Stelle. $x_4 = 0.\bar{3}$ ist für kein n als n-stellige Gleitpunktzahl darstellbar.

2. Bestimmen Sie alle dualen 3-stelligen positiven Gleitpunktzahlen mit einstelligem positiven binären Exponenten sowie ihren dezimalen Wert.

   $0.000 \cdot 2^0 = 0$, $0.100 \cdot 2^0 = 0.5$, $0.101 \cdot 2^0 = 0.625$, $0.110 \cdot 2^0 = 0.75$, $0.111 \cdot 2^0 = 0.875$, $0.100 \cdot 2^1 = 1$, $0.101 \cdot 2^1 = 1.25$, $0.110 \cdot 2^1 = 1.5$, $0.111 \cdot 2^1 = 1.75$.

3. Wie viele verschiedene Maschinenzahlen gibt es auf einem Rechner, der 20-stellige Gleitpunktzahlen mit 4-stelligen binären Exponenten sowie dazugehörige Vorzeichen im Dualsystem verwendet? Wie lautet die kleinste positive und die größte Maschinenzahl?

Für die 20-stellige Mantisse im Dualsystem gibt es 2^{19} verschiedene Möglichkeiten (die erste Nachkommaziffer muss ja 1 sein). Zusammen mit dem Vorzeichen gibt es also 2^{20} Möglichkeiten. Für den 4-stelligen Exponenten im Dualsystem gibt es 2^4 Möglichkeiten, inkl. Vorzeichen also 2^5. Insgesamt gibt es also $2^{20} \cdot 2^5 = 2^{25} = 33554432$ Möglichkeiten. Da wir aber die Zahl 0 noch nicht erfasst haben, sind es insgesamt 33554433 Maschinenzahlen.
Die kleinste positive Maschinenzahl ist dabei $0.1 \cdot 2^{-1111} = 2^{-16}$
$\approx 1.53 \cdot 10^{-5}$, die größte ist $0.11111111111111111111 \cdot 2^{1111}$
$= (1 - 2^{-20}) \cdot 2^{15} = 2^{15} - 2^{-5} = 32767.96875$.

4. Verstehen Sie den folgenden 'Witz'?

   Es gibt 10 Gruppen von Menschen: diejenigen, die das Binärsystem verstehen, und die anderen.

   10 im Dualsystem hat den Wert 2 im Dezimalsystem.

Aufgabe 2.3:

1. Schreiben Sie die kleinste und grösste binäre postive Maschinenzahl für die vorhergehende Abbildung ($n = 4$ und $0 \leq e \leq 3$) explizit auf und berechnen Sie deren Wert.

   Lösung: $x_{max} = 0.1111 \cdot 2^3 = 7.5$ und $x_{min} = 0.1000 \cdot 2^0 = 0.5$

2. Stimmt das mit x_{max} und x_{min}überein?

   Lösung: $x_{max} = (1 - B^{-n})B^{e_{max}} = 2^3 - 2^{-1} = 7.5$ und $x_{min} = B^{e_{min}-1} = 2^{-1} = 0.5$. Antwort: ja.

Aufgabe 2.4:

1. Vergewissern Sie sich anhand einfacher Zahlenbeispiele, dass die Rundung ein besseres Verfahren für die Abbildung einer rellen Zahl auf eine Maschinenzahl darstellt als einfaches Abschneiden der überzähligen Ziffern, wie in den früheren Beispielen in Kap. 2.3.2. Was ist der maximale Fehler, der durch das Abschneiden auftreten kann?

   Lösung: wir betrachten z.B. die Zahl 0.739 im Dezimalsystem. Bei zweistelliger Mantisse erhält man durch Abscheiden die Maschinenzahl 0.73 mit dem absoluten Fehler 0.009. Bei Rundung erhält man 0.74 mit dem absoluten Fehler 0.001. Es ist einfach zu sehen, dass der maximale Fehler der Rundung hier 0.005 beträgt (z.B. wird 0.7350 auf 0.74 aufgerundet), während der maximale Fehler beim Abschneiden (fast) doppelt so gross sein kann (z.B. wird 0.7399 auf 0.73 abgeschnitten) mit dem absoluten Fehler von (fast) 0.01. Das lässt sich auf beliebige Mantissenlängen und

Basen erweitern. Es gilt also

$$\begin{aligned} \mid rd(x) - x \mid &\leq \frac{B}{2} \cdot B^{e-n-1} \\ \mid abschneiden(x) - x \mid &< 2 \cdot \frac{B}{2} \cdot B^{e-n-1} = B \cdot B^{e-n-1} = B^{e-n} \end{aligned}$$

2. Wir kennen die (allgemeinen) Rundungsregeln für das Dezimalsystem. Verallgemeinern sie diese für eine beliebige gerade Basis B. Runden Sie anschliessend die folgenden Zahlen auf eine vierstellige Mantisse, berechnen Sie den absoluten Fehler der Rundung und vergewissern Sie sich, dass $\mid rd(x) - x \mid \leq \frac{B}{2} \cdot B^{e-n-1}$. Gilt diese Relation bei gleichen Rundungsregeln auch für ungerade Basen?

   a) $(11.0100)_2$ b) $(11.0110)_2$ c) $(11.111)_2$ d) $(120.212)_3$ e) $(120.222)_3$ f) $(0.FFFFF)_{16}$

   Lösung:

   - Regeln für allgemeines Runden und gerader Basis: Wird eine Gleitpunktzahl mit $n+1$ stelliger Mantisse und gerader Basis B auf n Stellen gerundet, so wird aufgerundet, falls $m_{n+1} \geq \frac{B}{2}$, und abgerundet, falls $m_{n+1} < \frac{B}{2}$.

   (a) $rd\left(0.110100 \cdot 2^2\right) = 0.1101 \cdot 2^2$ mit $\mid rd(x) - x \mid = \mid 3.25 - 3.25 \mid = 0 \leq 1 \cdot 2^{2-4-1} = 2^{-3} = 0.125$

   (b) $rd\left(0.110110 \cdot 2^2\right) = 0.1110 \cdot 2^2$ mit $\mid rd(x) - x \mid = \mid 3.5 - 3.375 \mid = 0.125 \leq 1 \cdot 2^{2-4-1} = 2^{-3} = 0.125$

   (c) $rd\left(0.11111 \cdot 2^2\right) = 0.1000 \cdot 2^3$ mit $\mid rd(x) - x \mid = \mid 4 - 3.875 \mid = 0.125 \leq 1 \cdot 2^{2-4-1} = 2^{-3} = 0.125$

   (d) $rd\left(0.120212 \cdot 3^3\right) = 0.1202 \cdot 3^3$ mit $\mid rd(x) - x \mid = \mid 15,6\overline{66} - 15,\overline{851} \mid = 0.\overline{185} \overset{?}{\leq} 1.5 \cdot 3^{3-4-1} = 1.5 \cdot 3^{-2} = 0.1\overline{66}$. Die Abschätzung gilt nicht. Für ungerade Basen müssten entweder die Abschätzung oder die Rundungsregeln angepasst werden.

   (e) $rd\left(0.120222 \cdot 3^3\right) = 0.1210 \cdot 3^3$ mit $\mid rd(x) - x \mid = \mid 16 - 15,\overline{962} \mid = 0.\overline{037} \leq 1.5 \cdot 3^{3-4-1} = 1.5 \cdot 3^{-2} = 0.1\overline{66}$

   (f) $rd\left(0.FFFFF \cdot 16^0\right) = 0.1000 \cdot 16^1$ mit $\mid rd(x) - x \mid = \mid 1 - 0,999999046325683... \mid = 9.5367 \cdot 10^{-7} \leq 8 \cdot 16^{0-4-1} = 8 \cdot 16^{-5} = 7.6294 \cdot 10^{-6}$

Aufgabe 2.5 [1]:

1. Gesucht ist eine Näherung $\widetilde{x}$ zu $x = \sqrt{2} = 1.414213562...$ mit einem absoluten Fehler von höchstens 0.001.

Lösung: $\tilde{x}_1 = 1.414$ erfüllt das Verlangte, denn $|\tilde{x} - x| = 0.000213562\ldots \leq 0.001$. Andere Möglichkeiten sind $\tilde{x}_2 = 1.4139$. $\tilde{x}_1$ stimmt auf 4 Ziffern mit dem exakten Wert überein, $\tilde{x}_2$ nur auf 3. Eine größere Anzahl an übereinstimmenden Ziffern bedeutet aber keinesfalls immer einen kleineren absoluten Fehler, wie das Beispiel $x = \sqrt{3} = 1.732050808\ldots$ und $\tilde{x}_1 = 2.0$, $\tilde{x}_2 = 1.2$ zeigt: $\tilde{x}_1$ hat keine gültige Ziffer, $\tilde{x}_2$ hat eine gültige Ziffer, trotzdem besitzt $\tilde{x}_1$ den kleineren absoluten Fehler. ∎

2. Es soll $2590 + 4 + 4$ in 3-stelliger Gleitpunktarithmetik gerechnet werden (im Dezimalsystem), einmal von links nach rechts und einmal von rechts nach links. Wie unterscheiden sich die Resultate?

   Lösung: Alle 3 Summanden sind exakt darstellbar. Als Ergebnis erhält man, bei Rechnung von links nach rechts:

   $$2590 + 4 = 2594 \overset{runden}{\rightarrow} 2590, \qquad 2590 + 4 = 2594 \overset{runden}{\rightarrow} 2590.$$

   Die beiden kleinen Summanden gehen damit gar nicht sichtbar in das Ergebnis ein. Rechnet man jedoch in anderer Reihenfolge

   $$4 + 4 = 8 \overset{runden}{\rightarrow} 8, \qquad 8 + 2590 = 2598 \overset{runden}{\rightarrow} 2600$$

   so erhält man einen genaueren Wert, sogar den in 3-stelliger Gleitpunktarithmetik besten Wert (2598 wird bestmöglich durch die Maschinenzahl 2600 dargestellt). ∎

3. Berechnen Sie $s_{300} := \sum_{i=1}^{300} \frac{1}{i^2}$ sowohl auf- als auch absteigend, je einmal mit 3-stelliger und 5-stelliger Gleitpunktarithmetik (in MATLAB können Sie eine Zahl x auf 3 Stellen reduzieren z.B. mit dem Befehl string2num(num2string(x,3))).

   $s_{300} = 1.6416062828976228698\ldots$ bei exakter Rechnung

   $s_{141} = s_{142} = \ldots = s_{300} = 1.6390$ 5-stellig gerechnet, addiert von 1 bis 300

   $s_{300} = 1.6416$ 5-stellig gerechnet, addiert von 300 bis 1

   $s_{14} = s_{15} = \ldots = s_{300} = 1.59$ 3-stellig gerechnet, addiert von 1 bis 300

   $s_{300} = 1.64$ 3-stellig gerechnet, addiert von 300 bis 1.

   Bei 3- bzw. 5-stelliger Rechnung und geeigneter Wahl der Summationsreihenfolge wird also das auf 3 bzw. 5 Stellen genaue exakte Ergebnis erzielt. Dagegen wird bei 3- bzw. 5-stelliger Rechnung und ungeschickter Wahl der Summationsreihenfolge das exakte Ergebnis nur auf 1 bzw. 2 Stellen genau erreicht. Dies macht den Einfluss deutlich, den die Summationsreihenfolge bei der Rechnung auf einem Computer besitzt. ∎

4. Es ist $\lim_{n\to\infty}(1+\frac{1}{n})^n = \mathrm{e}$. Erstellen Sie eine Tabelle mit ihrem Rechner für n = 1, 10, 100, ... für den Ausdruck $(1+\frac{1}{n})^n$sowie den absoluten und relativen Fehler. Erklären Sie Ihre Beobachtungen.

n	(1+1/n)^n	abs. Fehler	rel. Fehler
1.E+00	2.0000000E+00	7.1828183E-01	2.6424112E-01
1.E+02	2.7048138E+00	1.3467999E-02	4.9546000E-03
1.E+03	2.7169239E+00	1.3578962E-03	4.9954210E-04
1.E+04	2.7181459E+00	1.3590163E-04	4.9995417E-05
1.E+05	2.7182682E+00	1.3591262E-05	4.9999457E-06
1.E+06	2.7182805E+00	1.3593026E-06	5.0005949E-07
1.E+08	2.7182818E+00	4.2063248E-08	1.5474204E-08
1.E+09	2.7182820E+00	2.0235546E-07	7.4442415E-08
1.E+10	2.7182821E+00	2.2477574E-07	8.2690375E-08
1.E+15	3.0350352E+00	3.1675338E-01	1.1652706E-01
1.E+16	1.0000000E+00	1.7182818E+00	6.3212056E-01

5. Überlegen Sie sich einen kurzen iterativen Algorithmus, der die Maschinengenauigkeit Ihres Rechners prüft. Schliessen Sie aus dem Ergebnis, ob Ihr Rechner im Dual- oder Dezimalsystem rechnet und mit welcher Stellenzahl er operiert.

   $eps := 1$; while $1. + eps \neq 1.$ do $eps := eps/2$; $eps := eps \cdot 2$; write eps. Das Ergebnis, das dieses Programm liefert, hängt natürlich von dem Rechner ab, auf dem es läuft. Auf einem Taschenrechner könnte man z. B. $eps = 5 \cdot 10^{-10}$ erwarten. Auf einem PC werden Sie auch einen Unterschied feststellen, wenn Sie in Ihrem Programm *double precision* anstelle *single precision* verwenden.

Aufgabe 2.6 [1]:

1. Untersuchen Sie, ob die Multiplikation und die Division zweier Zahlen gut oder schlecht konditionierte Funktionsauswertungen sind.

   Lösung: Multiplikation: $f(x) = c \cdot x$ $(c \in \mathbb{R})$, $K := \frac{|f'(x)| \cdot |x|}{|f(x)|} = \frac{|c| \cdot |x|}{|cx|} = 1$, daraus folgt gute Konditionierung. Analog für Division.