Programmieren für Mathematiker WS2017/18

Dozent: Prof. Dr. Wolfgang Walter

12. August 2018

Inhaltsverzeichnis

Kapitel I

allgemeine Informationen

Eine Programmiersprache ist lexikalisch, syntaktisch und semantisch eindeutig definiert. Eine Compiler übersetzt die Programmiersprache in Maschinensprache . Ein Interpreter arbeitet das Programm dann ab. Ein Laufzeitsystem stellt grundlegende Operationen und Funktionen zur Verfügung.

Bereiche der Informatik

Die Informatik untergliedert sich in 4 Bereiche:

- Technische Informatik
- Praktische Informatik
- Theoretische Informatik
- Angewandte Informatik

Die Technische Informatik beschäftigt sich mit der Konstruktion der Hardware, zum Beispiel der Datenleitungen, um Informationen durch das Internet zu transportieren. Wichtige Firmen sind hier: Intel, Globalfoundries und Infineon.

Die Praktische Informatik beschäftigt sich mit der Software, also Betriebssystem, Compiler, Interpreter und so weiter. In alltäglicher Software findet sich rund 1 Fehler in 100 Zeilen Quelltext. In wichtiger Software, also Raketen, Betriebssysteme, ..., ist es nur 1 Fehler pro 10.000 Zeilen Code.

Die Theoretische Informatik beschäftigt sich mit Logik, formalen Sprachen, der Automatentheorie, Komplexität von Algorithmen, ...

Die Angewandte Informatik beschäftigt sich mit der Praxis, dem Nutzer, der Interaktion zwischen Mensch und Maschine, ...

Maßeinheiten und Größenordnungen

Ein bit ist ein Kunstwort aus “binary“ und “digit“. Es kann nur 2 Werte speichern: 0 und 1

Ein nibble ist eine Hexadezimalziffer, bündelt also 4 bits und kann damit 16 Werte annehmen: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E und F.

Ein byte bündelt 2 nibble, also 8 bit. Er ist die gebräuchlichste, direkt addressierbare, kleinste Speichereinheit. Weitere Speichergrößen sind:

Name	**Anzahl byte**	**Name**	**Anzahl byte**
1 KB	10^3	1 KiB	$2^{10} = 1.024$
1 MB	10^6	1 MiB	$2^{20} = 1.048.576$
1 GB	10^9	1 GiB	$2^{30} = 1.073.741.824$
1 TB	10^{12}	1 TiB	2^{40}
1 PB	10^{15}	1 PiB	2^{50}
1 EB	10^{18}	1 EiB	2^{60}

Der ROM ("read-only-memory") speichert wichtige Informationen auch ohne Strom, wie zum Beispiel die Uhrzeit, Informationen über die Festplatte, ... Er ist nicht mehr änderbar, außer durch Belichtung.

Der RAM ("random-access-memory") ermöglicht den Zugriff auf alle Adressen, insbesondere im Hauptspeicher.

Kapitel II

Zahldarstellungen

Basis-Konvertierung ganzer Zahlen

Die Notation $[9]_{10}$ bedeutet, dass man die Zahl 9 im Zehner-System betrachtet. Es gilt also $[9]_{10} = [1001]_2$ und $[10]_{10} = [1010]_2$.

Um eine Zahl von einer gegebenen Basis in eine Zielbasis b zu konvertieren, so teilt man immer wieder durch b und notiert den Rest als nächste Ziffer von hinten nach vorne. Am Beispiel von $[57]_{10}$ ins Zweier-System sieht das so aus:

$$\begin{aligned}
\frac{57}{2} &= 28 \text{ Rest } 1 \Rightarrow \text{ letzte Ziffer der Binärdarstellung} \\
\frac{28}{2} &= 14 \text{ Rest } 0 \Rightarrow \text{ vorletzte Ziffer der Binärdarstellung} \\
\frac{14}{2} &= 7 \text{ Rest } 0 \\
\frac{7}{2} &= 3 \text{ Rest } 1 \\
\frac{3}{2} &= 1 \text{ Rest } 1 \\
\frac{1}{2} &= 0 \text{ Rest } 1
\end{aligned}$$

Also gilt: $[57]_{10} = [111001]_2$.

Die umgekehrte Richtung verläuft ähnlich:

```
 111001  :  1010  =  101  R  111
-1010
 ------
 01000
-00000
 ------
  10001
 -01010
 ------
    111
```

Also $[111001]_2$ durch $[10]_{10} = [1010]_2$ gleich $[101 \text{ Rest } 111]_2 = [5 \text{ Rest } 7]_{10} \Rightarrow [57]_{10}$.

Von Basis 2 in Basis 4, 8 oder 16 ist dann ganz einfach: $[111100101]_2$

- Zweiergruppen von hinten nach vorne zusammenzählen: $[13211]_4$
- Dreiergruppen von hinten nach vorne zusammenzählen: $[745]_8$
- Vierergruppen von hinten nach vorne zusammenzählen: $[1E5]_{16}$

Basis-Konversion gebrochener Zahlen

Festkommadarstellung (nur Betrag der Zahl, ohne Vorzeichen):

Gewichte	B^k	B^{k-1}	...	B^1	B^0	.	B^{-1}	B^{-2}	...	B^{-l}
Ziffern	m_k	m_{k+1}	...	m_{-1}	m_0	.	m_1	m_2	...	m_l

Also: $\sum\limits_{i=k}^{l} m_i \cdot B^{-i}$.

Die Konvertierung des ganzzahligen Anteils vor dem “.“ läuft wie gehabt. Um den gebrochenen Anteil zu konvertieren, multipliziert man wiederholt mit der Zielbasis b und nimmt den jeweiligen ganzzahligen Anteil als Nachkommaziffern (von links nach rechts). Mit dem gebrochenen Anteil macht man weiter. Wir wollen die Zahl $[0.625]_{10}$ ins Zweiersystem konvertieren:

$$0.625 \cdot 2 = \mathbf{1}.25$$
$$0.25 \cdot 2 = \mathbf{0}.5$$
$$0.5 \cdot 2 = \mathbf{1}$$

Also gilt: $[0.625]_{10} = [0.101]_2$.

Wieder anders herum:

$$0.101 \cdot 1010 = \mathbf{110}.010$$
$$0.010 \cdot 1010 = \mathbf{10}.100$$
$$0.100 \cdot 1010 = \mathbf{101}.0$$

Also gilt $[0.101]_2 = [0.110|10|101]_2 = [0.625]_{10}$.

Jetzt wollen wir $[0.1]_{10}$ ins Zweiersystem konvertieren:

$$0.1 \cdot 2 = \mathbf{0}.2$$
$$0.2 \cdot 2 = \mathbf{0}.4 \tag{1}$$
$$0.4 \cdot 2 = \mathbf{0}.8$$
$$0.8 \cdot 2 = \mathbf{1}.6$$
$$0.6 \cdot 2 = \mathbf{1}.2$$
$$0.2 \cdot 2 = \mathbf{0}.4 \tag{2}$$

Wie man sieht, sind die Zeilen (1) und (2) gleich, das heißt, diese Konvertierung wird unendlich lange laufen. Also: $[0.1]_{10} = [0.0\overline{0011}]_2$. Aber es muss gelten: $[0.1]_{10} \cdot [10]_{10} = [1]_{10}$. Aber es stimmt: $[0.0\overline{0011}]_2 \cdot [1010]_2 = [0.\overline{1}]_2 = [1]_2$.

Entsprechend gilt:

$$
\begin{aligned}
[0.2]_{10} &= [0.\overline{0011}]_2 \\
[0.3]_{10} &= [0.01\overline{0011}]_2 \\
[0.4]_{10} &= [0.011\overline{0011}]_2 \\
[0.5]_{10} &= [0.1]_2 \\
[0.6]_{10} &= [0.1\overline{0011}]_2 \\
[0.7]_{10} &= [0.1011\overline{0011}]_2 \\
[0.8]_{10} &= [0.11\overline{0011}]_2 \\
[0.9]_{10} &= [0.111\overline{0011}]_2
\end{aligned}
$$

Problem: Rundungen schon bei $\frac{1}{10} \Rightarrow$ falsche Nachkommastellen. Die Lösung sind hier Gleitkommazahlen.

Kapitel III

Grundstrukturen von Algorithmen

Kapitel IV

Ein- und Ausgabe

Anhang