Einführung in die Informatik
Ausarbeitung Übung 2

Jakob Schulz, Julian Niethammer

31. Oktober 2023

1 Umrechnung zwischen Zahlensystemen

1.1 Problem

Zahlen von verschiedenen Zahlensystemen (Binär, Dezimal, Hexadezimal und Octal) umrechnen in die jeweiligen anderen Zahlensysteme.

1.2 Lösungskonzept

Die anderen Zahlensysteme entsprechen dem Dezimalsystem. Unterschied besteht darin, dass die Stellen der Zahl eine andere Bedeutung haben.
Beim Dezimalsystem steht jede Stelle für ein Vielfaches von 10. Beim Oktalsystem steht wiederrum jede Stelle für ein Vielfaches von 8 und beim Binärsystem dementsprechend jede Stelle für ein Vielfaches von 2
Bsp.:
Dezimalsystem in Oktalsystem:

1000	100	10	1		4096	512	64	8	1
4	2	3	0_{10}	=	1	0	2	0	6_8

Somit muss man, wenn man eine Dezimalzahl in ein beliebiges anderes Zahlensystem umrechnen möchte, nur die Basis (Binär: 2, Oktal: 8, ...) mit Rest teilen. Der vorhandene Rest repräsentiert immer die niederwertigste „freie" Stelle. Das Ergebnis ohne den Rest wird wieder durch die Basis geteilt. Das ganze wiederholt sich solange, bis das Ergebnis ohne den Rest null ist. Das Prinzip besteht somit darin, dass man die Zahl von rechts nach links aufbaut.
Bsp.:
Dezimalzahl 123456 in Hexadezimalzahl umwandeln

Rechnung	ganzzahliges Ergebnis	Rest
$123456 \div 16$	7716	0
$7716 \div 16$	482	4
$482 \div 16$	30	2
$30 \div 16$	1	E
$1 \div 16$	0	1

Hexadezimalzahl: $1E240_{16}$

Will man nun eine Zahl von einem Ausgangssystem in ein Zielsystem umwandeln (keines der Systeme ist Dezimalsystem), bietet es sich an die Zahl vom Ausgangssystem erst in das Dezimalsystem umzuwandeln und dann in das Zielsystem umzuwandeln.
Um eine Zahl von einem anderen System in ein Dezimalsystem umzuwandeln geht man wie folgt vor:
Man nimmt jeden Repräsentanten einer Stelle der Zahl und multipliziert

diesen mit dem Wert der entsprechenden Stelle. Anschließend addiert man alle Ergebnisse zusammen und man hat die Dezimalzahl.
Bsp.: $101011 = 1 \cdot 1 + 1 \cdot 2 + 0 \cdot 4 + 1 \cdot 8 + 0 \cdot 16 + 1 \cdot 32 = 43$

1.3 Beispielaftes Umrechnen von Zahlen

$192_{10} \mathrel{\widehat{=}} 11000000_2 \mathrel{\widehat{=}} C0_{16} \mathrel{\widehat{=}} 300_8$

$0C_{16} \mathrel{\widehat{=}} 12_{10} \mathrel{\widehat{=}} 1100_2 \mathrel{\widehat{=}} 14_8$

$764_8 \mathrel{\widehat{=}} 500_{10} \mathrel{\widehat{=}} 111110100_2 \mathrel{\widehat{=}} 1F4_{16}$

$01111110_2 \mathrel{\widehat{=}} 126_{10} \mathrel{\widehat{=}} 176_8 \mathrel{\widehat{=}} 7E_{16}$

1.4 Tests

Das Ergebnis lässt sich überprüfen, indem man die Zahlen aus jedem Zahlensystem in eine Dezimalzahl umwandelt. Sind alle Dezimalzahlen gleich, geben die Zahlen der anderen Systeme alle die gleiche Zahl wieder.

1.5 Beispiel aus der Thematik IPv4 Adressen

1.5.1 Aufgabe

Den unteren und oberen darstellbaren Wert im Dezimalsystem und im Hexadezimalsystem bestimmen

1.5.2 Ausarbeitung

- Sie haben 8 Bit zur Informationsdarstellung: Die Wertedarstellung geht von $0_{10} = 00_{16}$ bis $255_{10} = FF_{16}$
- Das höchstwertige Bit muss 0 sein: Die Wertedarstellung geht von $0_{10} = 00_{16}$ bis $127_{10} = 7F_{16}$
- Jetzt muss das höchstwertige Bit immer 1 sein, das zweithöchste Bit muss 0 sein: Die Wertedarstellung geht von: $128_{10} = 80_{16}$ bis $191_{10} = BF_{16}$
- Jetzt mussen das höchste und das zweithöchstes Bit 1 gesetzt sein, das dritthöchste Bit muss 0 sein: Die Wertedarstellung geht von $192_{10} = C0_{16}$ bis $223_{10} = DF_{16}$

2 Gebrochenrationale Zahlen

2.1 Größtmögliche Zahl berechnen

2.1.1 Aufgabe

Größt mögliche Dezimalzahl mit 4 Bit Vor- und Nachkommastellen berechnen.

2.1.2 Anastz

$1111.1111_2 = 1 \cdot 1 + 1 \cdot 2 + 1 \cdot 4 + 1 \cdot 8 + 1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + 1 \cdot \frac{1}{8} + 1 \cdot \frac{1}{16} = 15.9375$

2.2 Lücken in der Tabelle füllen

Dualsystem	Oktalsystem	Hexadezimalsystem
101101.101	55.5	2D.A
10101011.11001101	253.632	AB.CD

Hier wäre Nachvollziehung/Entstehung der eingesetzten Werte noch wichtig gewesen. Frage, wie setzen sich Ergebnisse zusammen.
Eingesetzte Werte sind blau.

3 Binäre Addition/Subtraktion

3.1 Addition im Dualsystem

3.1.1 Aufgaben

Aufgaben lösen, indem man Dezimalzahlen zuerst in Dualsystem umwandelt und in diesem dann rechnet

3.1.2 Ansatz

$$\begin{array}{lcc l} 125_{10} + 199_{10} = & & 01111101 & \\ & + & 11000111 & \\ \cline{3-3} & = & 101000100_2 & = 324_{10} \end{array}$$

$27_{10} + 30_{10} = 11011_2 + 11110_2 = 111001_2 = 57_{10}$
$115_{10} + 21_{10} = 1110011_2 + 10101_2 = 10001000_2 = 136_{10}$

3.2 Subtraktion im Dualsystem

3.2.1 Aufgabe

8 stellige Binärzahlen unter Verwendung des Zweierkomplements subtrahieren.

3.2.2 Ansatz

$$
\begin{array}{lrl}
55_{10} - 120_{10} = & & 00110111 \\
& + & 10001000 \\
\cline{3-3}
& = & 10111111_Z \quad = -65_{10}
\end{array}
$$

$42_{10} - 12_{10} = 00101010 + 11110100 = 00011110_Z = 30_{10}$
$18_{10} - 105_{10} = 00010010 + 10010111 = 10101001_Z = -87_{10}$
Z steht für Zweierkomplement

3.3 Warum ist die Begrenzung auf 8 Binärstellen zwingend notwendig?

Ohne die Bergenzung auf 8 Binärstellen kann beim höchstwertigen Bit aufgrund des Übertrages ein neues h"öchstwertiges Bit entstehen. Dadrch würde eine falsche Zahl dargestellt werden.
Bsp für unbegrenzte Binärstellen:

$$
\begin{array}{lrl}
42_{10} - 12_{10} = & & 00101010 \\
& + & 11110100 \\
& & {\scriptstyle 111} \\
\cline{3-3}
& = & 100011110_Z \quad \neq 30_{10}
\end{array}
$$

4 Resumee zur dieser Übungsaufgabe

Dauer für

- Durchführung: ca. 1 Stunde
- Dokumentation: ca. 3 Stunden

Welche großen Probleme waren zu lösen?
Einarbeitung in die Umgebung von LaTex und anschauliche Darstellung des Lösungsansatzes