



Willkommen zum Modul Einführung in die Informatik

Wer bin ich?

- Name: **Julian Bensch** - 25 Jahre
- Beruf: **Softwareentwickler**
- Hobbys: **Programmieren, Lesen, Reisen**
- Jahre in der IT: **11**

Beruflicher Hintergrund

Freelance Software Entwickler seit 2018

Davor BTA (Biologisch technischer Assistent)

Vor dieser Zeit aber auch schon entwickelt und an Computern rumgeschraubt

Inhalte des Moduls 1/5

Einführung Zahlensysteme

- Dezimalsystem (Base-10 System)
- Binäres System (Base-2 System)
- Octales System (Base-8 System)
- Hexadezimals System (Base-16 System)

Rechnen mit binären-, octalen und hexadezimalen Zahlen

- Wo werden sie genutzt und wie rechne ich um?

Einführung in die Logik

- Aussagenlogik
- Boolesche Algebra

Automaten-Theorie

Inhalte des Moduls 2/5

Mengenlehre

- Grundlagen
 - Schnittmenge
 - Vereinigungsmenge
 - Differenzmenge
-

Relationen

- Äquivalenzrelationen
 - Ordnungsrelationen
 - Funktionen
-

Binäre Bäume

- Traversierung
- Suchbäume

Inhalte des Moduls 3/5

Einfache Algorithmen

- Lineare Suche
 - Binäre Suche
 - InsertionSort
 - BubbleSort
 - SelectionSort
 - MergeSort
 - QuickSort
-

Rekursion

- Fakultät
- Türme von Hanoi
- Fibonacci-Zahlen

Inhalte des Moduls 4/5

Laufzeitkomplexität

- Analyse und Vergleich von Algorithmen
 - O-Notation
-

Kosten von Algorithmen

- Speicherplatz
 - Rechenzeit
 - Energieverbrauch
 - Netzwerkverkehr
-

Abstrakte Datentypen

- Was sind Abstrakte Datentypen?
 - Warum braucht man sie?
 - Klassen
-

Inhalte des Moduls 5/5

Datenstrukturen

- Stack
- Liste
- Heap
- Queue
- Hash-Map

Vektor-Rechnung

- Grundlagen
- Skalarprodukt
- Kreuzprodukt

Matrizen-Rechnung

- Addition
- Multiplikation
- Determinante

Übersicht der relevanten Zahlensysteme 1/3

Tabellarische Übersicht der Zahlensysteme

System	Base	Zahlen	Anwendungsbereiche
Dezimal	10	0-9	Alltag, Finanzen
Binär	2	0,1	Informatik
Octal	8	0-7	Ältere Systeme
Hex	16	0-9,A-F	Programmierung

Übersicht der relevanten Zahlensysteme 2/3

Dezimalsystem (base 10 system):

- Anzahl der Ziffern: 10 (0-9)
- Grund: 10 Finger, historisch/biologisch
- Anwendungen: Alltag, Finanzen, Wissenschaftliche Notation
- Beispiel: $123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$

Binäres System (base 2 system):

- Anzahl der Ziffern: 2 (0,1)
- Grund: Elektronik, Schalter (an/aus)
- Anwendungen: Informatik
- Beispiel: $101 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$ (Dezimal: 5)

Übersicht der relevanten Zahlensysteme 3/3

Octales System (base 8 system):

- Anzahl der Ziffern: 8 (0-7)
- Grund: Einfachere Darstellung von Binärdaten & ältere Unix-Systeme
- Anwendungen: Frühe Computersysteme, gelegentlich in der Programmierung
- Beispiel: $123 = 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0$ (Dezimal: 83)

Hexadezimal System (base 16 system):

- Anzahl der Ziffern: 16 (0-9, A-F)
- Grund: Einfachere Darstellung von Binärdaten & Speicheradressen
- Anwendungen: Programmierung, Webfarben
- Beispiel: $7F2 = 7 \times 16^2 + 15 \times 16^1 + 2 \times 16^0$ (Dezimal: 2034)

Umrechnen von Dezimal in andere Zahlensysteme 1/2

Dezimal zu Binär:

- Beispiel: 123
- $123 / 2 = 61 \text{ Rest } 1$
- $61 / 2 = 30 \text{ Rest } 1$
- $30 / 2 = 15 \text{ Rest } 0$
- $15 / 2 = 7 \text{ Rest } 1$
- $7 / 2 = 3 \text{ Rest } 1$
- $3 / 2 = 1 \text{ Rest } 1$
- $1 / 2 = 0 \text{ Rest } 1$
- Ergebnis: 1111011 (von unten nach oben lesen)
- Kontrolle: $1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 123$

Umrechnen von Dezimal in andere Zahlensysteme 2/2

Dezimal zu Hexadezimal:

- Beispiel: 123
 - $123 / 16 = 7$ Rest 11 (Notation durch Buchstaben B)
 - $7 / 16 = 0$ Rest 7
 - Ergebnis: 7B (von unten nach oben lesen)
 - Kontrolle: $7 \times 16^1 + 11 \times 16^0 = 123$
-

Dezimal zu Oktal:

- Beispiel: 123
- $123 / 8 = 15$ Rest 3
- $15 / 8 = 1$ Rest 7
- $1 / 8 = 0$ Rest 1
- Ergebnis: 173 (von unten nach oben lesen)
- Kontrolle: $1 \times 8^2 + 7 \times 8^1 + 3 \times 8^0 = 123$

Umrechnen von anderen Zahlensystemen zu Dezimal

Binär zu Dezimal: Beispiel 1111011

- $1111011 = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
- $1111011 = 64 + 32 + 16 + 8 + 0 + 2 + 1$
- $1111011 = 123$

Hexadezimal zu Dezimal: Beispiel 7B

- $7B = 7 \times 16^1 + 11 \times 16^0$
- $7B = 112 + 11$
- $7B = 123$

Oktal zu Dezimal: Beispiel 173

- $173 = 1 \times 8^2 + 7 \times 8^1 + 3 \times 8^0$
- $173 = 64 + 56 + 3$
- $173 = 123$

Aufgabe: Umrechnen von Zahlensystemen

Typ	Aufgabe 1	Aufgabe 2	Aufgabe 3
Binär zu Dezimal	11011011	10011100	10101010
Dezimal zu Binär	173	219	255
Oktal zu Dezimal	357	521	777
Dezimal zu Oktal	317	465	511
Hexadezimal zu Dezimal	9A	FA	B2
Dezimal zu Hexadezimal	154	250	178

Lösung: Umrechnen von Zahlensystemen

Aufgabentyp	Aufgabe 1	Aufgabe 2	Aufgabe 3
Binär zu Dezimal	$11011011 = 219$	$10011100 = 156$	$10101010 = 170$
Dezimal zu Binär	$173 = 10101101$	$219 = 11011011$	$255 = 11111111$
Oktal zu Dezimal	$357 = 239$	$521 = 337$	$777 = 511$
Dezimal zu Oktal	$317 = 475$	$465 = 711$	$511 = 777$
Hexadezimal zu Dezimal	$9A = 154$	$FA = 250$	$B2 = 178$
Dezimal zu Hexadezimal	$154 = 9A$	$250 = FA$	$178 = B2$

Aufgabe: Umrechnen von Zahlensystemen (Mischformen)

Aufgabentyp	Aufgabe
Binär zu Oktal	11010110
Oktal zu Hexadezimal	641
Hexadezimal zu Binär	3FA
Binär zu Hexadezimal	10101011
Hexadezimal zu Oktal	1F
Oktal zu Binär	572

Lösung: Umrechnen von Zahlensystemen (Mischformen)

Aufgabentyp	Aufgabe
Binär zu Oktal	$11010110 = 326$
Oktal zu Hexadezimal	$641 = 1A1$
Hexadezimal zu Binär	$3FA = 111111010$
Binär zu Hexadezimal	$10101011 = AB$
Hexadezimal zu Oktal	$1F = 37$
Oktal zu Binär	$572 = 101111010$

Detaillierte Lösung: Binär zu Oktal 11010110

- $11010110 = 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
- $11010110 = 128 + 64 + 16 + 4 + 2$
- $11010110 = 214$ (Dezimal)
- Nun Dezimal zu Oktal
- $214 / 8 = 26$ Rest 6
- $26 / 8 = 3$ Rest 2
- $3 / 8 = 0$ Rest 3
- Ergebnis: 326 (von unten nach oben lesen)

Detaillierte Lösung: Oktal zu Hexadezimal 641

- $641 = 6 \times 8^2 + 4 \times 8^1 + 1 \times 8^0$
- $641 = 384 + 32 + 1$
- $641 = 417$ (Dezimal)
- Nun Dezimal zu Hexadezimal
- $417 / 16 = 26$ Rest 1
- $26 / 16 = 1$ Rest 10 (Notation durch Buchstaben A)
- $1 / 16 = 0$ Rest 1
- Ergebnis: 1A1 (von unten nach oben lesen)

Detaillierte Lösung Hexadezimal zu Binär 3FA

Umrechnen in Dezimal

- $3FA = 3 \times 16^2 + 15 \times 16^1 + 10 \times 16^0$
- $3FA = 768 + 240 + 10$
- $3FA = 1018$ (Dezimal)

Umrechnen in Binär

- $1018 / 2 = 509$ Rest 0
- $509 / 2 = 254$ Rest 1
- $254 / 2 = 127$ Rest 0
- $127 / 2 = 63$ Rest 1
- $63 / 2 = 31$ Rest 1
- $31 / 2 = 15$ Rest 1
- $15 / 2 = 7$ Rest 1
- $7 / 2 = 3$ Rest 1
- $3 / 2 = 1$ Rest 1
- $1 / 2 = 0$ Rest 1
- Ergebnis: 111111010 (von unten nach oben lesen)

Detaillierte Lösung Binär zu Hexadezimal 10101011

- $10101011 = 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
- $10101011 = 128 + 32 + 8 + 2 + 1$
- $10101011 = 171$ (Dezimal)
- Nun Dezimal zu Hexadezimal
- $171 / 16 = 10$ Rest 11 (Notation durch Buchstaben B)
- $10 / 16 = 0$ Rest 10 (Notation durch Buchstaben A)
- Ergebnis: AB (von unten nach oben lesen)

Detaillierte Lösung Hexadezimal zu Oktal 1F

- $1F = 1 \times 16^1 + 15 \times 16^0$
- $1F = 16 + 15$
- $1F = 31$ (Dezimal)
- Nun Dezimal zu Oktal
- $31 / 8 = 3$ Rest 7
- $3 / 8 = 0$ Rest 3
- Ergebnis: 37 (von unten nach oben lesen)

Detaillierte Lösung Oktal zu Binär 572

- $572 = 5 \times 8^2 + 7 \times 8^1 + 2 \times 8^0$
- $572 = 320 + 56 + 2$
- $572 = 378$ (Dezimal)
- Nun Dezimal zu Binär
- $378 / 2 = 189$ Rest 0
- $189 / 2 = 94$ Rest 1
- $94 / 2 = 47$ Rest 0
- $47 / 2 = 23$ Rest 1
- $23 / 2 = 11$ Rest 1
- $11 / 2 = 5$ Rest 1
- $5 / 2 = 2$ Rest 1
- $2 / 2 = 1$ Rest 0
- $1 / 2 = 0$ Rest 1
- Ergebnis: 101111010 (von unten nach oben lesen)

Zusammenfassung:

- Umrechnen von anderen Zahlensystemen zu Dezimal:
 - Binär zu Dezimal: Addition der Potenzen
 - Hexadezimal zu Dezimal: Addition der Potenzen
 - Oktal zu Dezimal: Addition der Potenzen
- Umrechnen von Dezimal in andere Zahlensysteme:
 - Dezimal zu Binär: Division durch 2
 - Dezimal zu Hexadezimal: Division durch 16
 - Dezimal zu Oktal: Division durch 8

Zum umrechnen wird erst in Dezimal umgerechnet, dann in das gewünschte Zahlensystem.

Aussagenlogik 1/5

- Atomare Aussagen:

Grundlegende, nicht weiter zerlegbare Aussagen.

Z.B.,

- p : Es regnet.
- q : Es ist kalt.

- Operatoren:

\wedge : UND (Verbindet zwei Aussagen, die beide wahr sein müssen.)

\vee : ODER (Verbindet zwei Aussagen; mindestens eine muss wahr sein.)

\neg : NICHT (Kehrt den Wahrheitswert einer Aussage um.)

- Merksätze:

Atomare Aussagen sind die Bausteine der Aussagenlogik.

Operatoren: Symbole, die Beziehungen zwischen atomaren Aussagen herstellen.

Außerdem: Operatoren verknüpfen atomare Aussagen zu komplexeren Ausdrücken.

Aussagenlogik 2/5

De Morgan'sche Regel

- Grundsätzlich gilt:
nicht (a und b) ist äquivalent zu ((nicht a) oder (nicht b)), sowie
nicht (a oder b) ist äquivalent zu ((nicht a) und (nicht b)).
- Beispiel:
Es regnet nicht und es ist nicht kalt.
 $\neg(p \wedge q) = \neg p \vee \neg q$
Es regnet nicht oder es ist nicht kalt.

Aussagenlogik 3/5

Beispiele

- Atomare Aussagen:

p : Es regnet.

q : Es ist kalt.

- komplexe Aussagen:

$p \wedge q$: Es regnet und es ist kalt.

$\neg q$: Es ist nicht kalt.

$\neg(p \wedge q)$: Es regnet nicht und es ist nicht kalt.

- Sehr komplexe Aussage:

$(p \wedge q) \vee (\neg p \wedge \neg q)$: Es regnet und es ist kalt oder es regnet nicht und es ist nicht kalt.

Aufgaben:

Bildet 5 eigene atomare Aussagen und stellt sie vor.

Bildet 5 eigene komplexe Aussagen und stellt sie vor.

Aussagenlogik 4/5

Gesetze der Aussagenlogik

- Kommutativgesetz: $p \wedge q = q \wedge p$, $p \vee q = q \vee p$
- Assoziativgesetz: $p \wedge (q \wedge r) = (p \wedge q) \wedge r$
- Distributivgesetz: $p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$

Aufgabe: Vereinfacht folgende Aussagen

- $(p \wedge q) \vee (\neg p \wedge q)$
- $(p \vee q) \wedge (p \vee \neg q)$
- $(p \wedge q) \vee (p \wedge \neg q)$
- $\neg(p \vee q) \vee (p \wedge q)$
- $(p \vee q \vee r) \wedge (\neg p \vee q \vee r)$

Aussagenlogik 5/5

Lösung: Gesetze der Aussagenlogik

Beispiel 1:

$$(p \wedge q) \vee (\neg p \wedge q)$$

Schritt 1: Distributivgesetz anwenden

Wir können das Distributivgesetz anwenden, um den Ausdruck zu vereinfachen. Dabei nehmen wir q als gemeinsamen Faktor und erhalten:

$$(p \vee \neg p) \wedge q$$

Schritt 3: Tautologie erkennen

Der Ausdruck $(p \vee \neg p)$ ist eine Tautologie, da er immer wahr ist. Wir können ihn also durch 1 (wahr) ersetzen und erhalten:

$$1 \wedge q$$

Schritt 4: Vereinfachen

Da $1 \wedge q$ immer q ist, können wir den Ausdruck vereinfachen zu: q

Lösungen:

- $(p \wedge q) \vee (\neg p \wedge q) = q$
- $(p \vee q) \wedge (p \vee \neg q) = p$
- $(p \wedge q) \vee (p \wedge \neg q) = p$
- $\neg(p \vee q) \vee (p \wedge q) = \neg q$
- $(p \vee q \vee r) \wedge (\neg p \vee q \vee r) = q \vee r$

Boolesche Algebra 1/2

- Bool'sche Variablen:

Grundlegende Variablen, die nur 1 (wahr) oder 0 (falsch) sein können.

Z.B., A: Es ist kalt. oder B: Es regnet.

- Grundoperationen:

AND: Beide Variablen müssen 1 sein.

OR: Mindestens eine Variable muss 1 sein.

NOT: Kehrt den Wert um.

- Merksätze:

Bool'sche Variablen sind die Grundlagen der Booleschen Algebra.

Grundoperationen: Elementare Operationen, die auf Bool'schen Variablen ausgeführt werden können.

Außerdem: Mit Grundoperationen bilden wir logische Ausdrücke und Funktionen.

Tipp: Auch hier gibt es eine De Morgan'sche Regel.

Boolesche Algebra 2/2

Beispiele

- Bool'sche Variablen:
A: Es ist kalt.
B: Es regnet.
- komplexe Ausdrücke:
A AND B: Es ist kalt und es regnet.
NOT A: Es ist nicht kalt.
NOT (A AND B): Es ist nicht kalt oder es regnet nicht. (oder beides nicht - De Morgan'sche Regel)
- Sehr komplexer Ausdruck:
(A AND B) OR (NOT A AND NOT B): Es ist kalt und es regnet oder es ist nicht kalt und es regnet nicht.

Aufgaben:

Bildet 5 eigene Bool'sche Variablen und stellt sie vor.

Bildet 5 eigene komplexe Ausdrücke und stellt sie vor.

Einführung in die Automatentheorie 1/8

Automaten sind Modelle für rechnende Maschinen welche einen Zustand (State) haben können.

Erklärung

- Zustand: Wo der Automat gerade ist, z.B. "an" oder "aus".

Themen:

- Endliche Automaten
- Nichtdeterministische Automaten
- Übergangsfunktionen

Einführung in die Automatentheorie 2/8

Endliche Automaten (DFA)

Deterministic Finite Automaton (Deterministischer Endlicher Automat)

Eindeutiger nächster Zustand für jede Eingabe und jeden aktuellen Zustand.

Definition:

- Deterministisch: Eindeutiger nächster Zustand
 - Formale Definition: $(Q, \Sigma, \delta, q_0, F)$
-

Erklärung

- Q : Menge aller Zustände
 - Σ : Menge der Symbole (Eingabe)
 - δ : Übergangsfunktion
 - q_0 : Startzustand
 - F : Menge der Endzustände
-

Merksatz

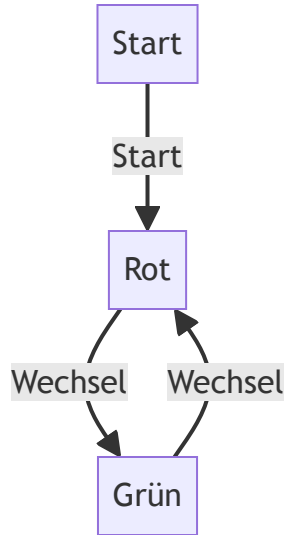
- DFA ist der einfachste Automatentyp.

Einführung in die Automatentheorie 3/8

Beispiel Aufgabe

- Zeichnet einen DFA für eine Ampel, die nur "rot" und "grün" kennt.

Beispiel:



Einführung in die Automatentheorie 4/8

Nichtdeterministische Automaten (NFA)

- Mehrere mögliche nächste Zustände
- Formale Definition: $(Q, \Sigma, \Delta, q_0, F)$

Merksatz

- NFA erlauben mehr Freiheit, aber gleiche Ausdruckskraft wie DFA.

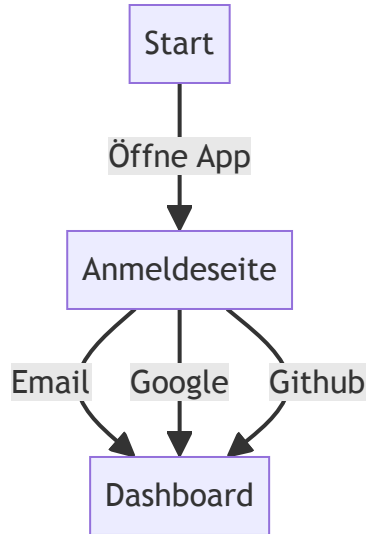
Erklärung

- 2^Q : Menge der möglichen Zustandskombinationen

Einführung in die Automatentheorie 5/8

Beispiel Aufgabe

- Zeichnet einen NFA für eine App in der man sich nach dem Start mit Email, Google oder Github anmelden kann und danach zu einem Dashboard kommt.



Einführung in die Automatentheorie 6/8

Übergangsfunktionen

- DFA: $\delta : Q \times \Sigma \rightarrow Q$
- NFA: $\Delta : Q \times \Sigma \rightarrow 2^Q$

Merksatz

- Übergangsfunktionen bestimmen die Dynamik des Automaten.

Erklärung

- \rightarrow : "führt zu" oder "wird zu"

Einführung in die Automatentheorie 7/8

Anwendungen der Automatentheorie

- Textverarbeitung: z.B. Rechtschreibprüfung
- Netzwerkprotokolle: Regeln für Datenübertragung
- Compilerbau: Umwandlung von Quellcode in Maschinencode

Aufgabe

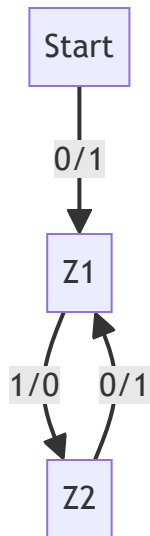
- Nennt 5 Anwendungsbeispiele für Automaten in der Praxis und zeichnet einen DFA oder NFA für jedes Beispiel.

Nutzt gerne: <https://mermaid.live/>

Einführung in die Automatentheorie 8/8

Mealy- und Moore-Automat

Beide Automaten sind deterministisch und sind in der Lage Ausgaben zu erzeugen.



Turingmaschine

Turingmaschinen sind nicht deterministisch und lassen nur mit nichtdeterministischen Automaten simulieren.

