Interreg

EUROPÄISCHE UNION

Österreich-Tschechische Republik

Europäischer Fonds für regionale Entwicklung

INFORMATIK

Theorie der Informatik

UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

VŠTE

EUROPÄISCHE UNION

Inhalt

1. Grundlegende mathematische Konzepte und numerische Systeme 2
2. Logik 5
3. Mengen und Beziehungen 9
3.1. Mengen 9
3.2. Beziehungen 11
4. Beziehungsstrukturen 13
5. Abbildung 15
6. Boolesche Algebra 17
7. Wahrscheinlichkeit und Statistiken 20
8. Informationstheorie 23
9. Anwendung der Informationstheorie 27
10. Theorie der Komplexität 31
11. Sprachen und Automaten 34
12. Turingmaschinen 36
13. Berechenbarkeitstheorie 39

1. GRUNDLEGENDE MATHEMATISCHE KONZEPTE UND NUMERISCHE SYSTEME

- Daten oder Angaben, welche wir mit den Sinnen erfassen können.
- Informationen sind Daten die wir verstehen, Sinn ergeben und quantifizierbar (messbar) sind nach Umwandlung in Zahlen.
- Informationen können sein: Text, Bild oder Ton.
- Die Informatik ist eine Wissenschaft der Speicherung, Verarbeitung unter Verwendung von Daten und Informationen. Es ist Teil der Mathematik und dient als Mittel zur Nutzung der Computertechnologie.
- IT-Branchen:
- Computertechnologie - Untersucht Hardware
- Algorithmisierung - Entwurf von Problemlösungsverfahren
- Programmierung - Konvertierung von Algorithmen in eine Programmiersprache
- Software Engineering - Anwendungsentwicklung
- Computergrafik - Lehre davon, wie man 2D- und 3D-Bilder erstellt und verarbeitet.
- Computermodellierung und Simulation - Anwendung von mathematischen Modellen auf reale Situationen
- Formale Logik, Automatisierungstheorie und Formelsprachen - mathematische Maschinenmodelle und Formelsprachen zum Schreiben von Algorithmen und Programmen
- Kybernetik und Robotik - Entwicklung von Maschinen, die zu eigenständigen Tätigkeiten fähig sind.
- Künstliche Intelligenz - Erforschung der Prozesse des menschlichen Denkens, ihrer mathematischen Beschreibung und Anwendung in der Maschinenentwicklung
- Hardware oder Computerhardware - ein Gattungsname für alle physischen Geräte, mit denen ein Computer ausgestattet ist.
- Software - kann in ein System und eine Anwendung unterteilt werden.
- Informationen, die einem computer gespeichert sind, messen wir in Bit - b - und Byte - B Einheiten.
- Das Bit ist die Basis und die kleinste Informationseinheit. Sie hat zwei Werte: 0, 1
- Byte als Informationseinheit hat einen 8-Bit-Wert.

Da die Informatik Teil der Mathematik ist, verwendet sie auch einige ihrer Begriffe, wie z.B.:

- Kartesisches Produkt
- Beziehung
- Morphismus
- Teilbarkeit

- Der größte gemeinsame Teiler, das kleinste gemeinsame Vielfache.
- Primzahlen

Da der Computer Informationen in Bits und Vielfachen speichert, arbeitet er in einem binären numerischen System. Es ist wichtig über numerische Systeme zu sprechen.

Numerische Systeme werden in positionelle und nicht-positionelle Systeme unterteilt.

Positionsnumerische Systeme - Der Wert jeder Ziffer wird durch ihre Position in der Symbolfolge bestimmt.

Ihr Vorteil ist die hohe Elastizität und ein relativ kleiner Ziffernsatz; der Nachteil ist eine sehr einfache Änderung des Wertes einer Zahl durch einfaches Hinzufügen einer Ziffer zur ursprünglichen Zahl.

Das Hauptmerkmal ist die Basis, dann sind die Gewichte der einzelnen Ziffern die Grundlage der Basis.

Beispiele:

- Unär - die Basis 1
- Binär - die Basis 2, zwei Zustände - ja / nein, 1 Bit, die Symbole 0, 1,
- Oktal - die Basis 8, ein Byte (= 8 Bit), die Symbole 0, 1, 2, 3, 4, 5, 6, 7
- Dezimal - die Basis 10, natürlich für Menschen (zehn Finger), die Symbole 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (die Zahl)
- Hexadezimal - die Basis: 16, 2 Bytes (= 16 Bit), Symbole: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Sexagesimal - die Basis 60, Zeitmessung

Nicht-Positions-Numerische Systeme - Ein Verfahren zur Darstellung von Zahlen, bei dem der Wert einer Ziffer nicht durch ihre Position in einer bestimmten Ziffernfolge gegeben ist. Diese Schreibweisen von Zahlen werden heute kaum noch verwendet und gelten als veraltet, haben aber gewisse Vorteile wie einfache Addition und Subtraktion. Die Nachteile sind, dass sie oft kein Symbol für Null- und negative Zahlen sowie eine lange Anzahl von Zahlen enthielten, die den Wert des größten Symbols des Systems deutlich überschreiten.

Beispiel: Römische Ziffern

Transfer zwischen Systemen

Von einem höheren Basissystem aus

Wenn wir von einem höheren Basissystem zu einem niedrigeren Basissystem wechseln, verwenden wir normalerweise einen Partitionsalgorithmus.

73/2 = 36 + 1 - 1 am 1. Platz von hinten
36/2 = 18 + 0 - 0 auf Platz 2 von hinten
18/2 = 9 + 0 - 0 auf Platz 3 von hinten
9/2 = 4 + 1 - 1 an der vierten Position von hinten
4/2 = 2 + 0 - 0 auf dem 5. Platz von hinten
2/2 = 1 + 0 - 0 bis zur 6. Position von hinten
1/2 = 0 + 1 - 1 an der 7. Position von hinten

Aus einem unteren Basissystem

Der Übergang von einem niedrigeren Basissystem zu einem höheren Basissystem ist einfacher. Wir wissen, dass die Zahl an der n-ten Stelle (berechnet aus 0) die Zahl darstellt, die sich aus der Verstärkung der Basis auf der n-ten Stelle ergibt. Diese Nummern werden nummeriert und übertragen.

$1 \cdot 20+0 \cdot 21+0 \cdot 22+1 \cdot 23+0 \cdot 24+0 \cdot 25+1 \cdot 26=73_{10}$

2. LOGIK

Logik ist die Wissenschaft der richtigen Argumentation oder die Kunst der richtigen Argumentation.

Was ist die Schlussfolgerung (Argument)?

Basierend auf der Wahrheit der Prämisse (Annahmen) P1.... Pn kann geschlossen werden, dass die Schlussfolgerung Z wahr ist.

Wir erkennen verschiedene Arten von Argumentationslogik.

Deduktion

Die Schlussfolgerung Z ergibt sich logischerweise aus den Annahmen 1, ..., Pn, wir bezeichnen P1, ..., Pn | = Z, wenn unter keinen Umständen der Fall eintreten kann, dass die Annahmen wahr wären und die Schlussfolgerung unwahr wäre.

Beispiel:

P1: Alle Kaninchen aus dem Hut sind weiß.
P2: Diese Kaninchen sind aus dem Hut.
Z: ⇒ Diese Kaninchen sind weiß.

Induktion

Generalisierung - von spezifisch bis allgemein

Beispiel:

P1: Diese Kaninchen sind aus dem Hut.
P2: Diese Kaninchen sind weiß.
Z: ⇒ (wahrscheinlich) Alle Kaninchen im Hut sind weiß.

Die Schlussfolgerung Z gilt nur mit einer bestimmten Wahrscheinlichkeit.

Abduktion

Wir erstellen Hypothesen für beobachtete Phänomene, Diagnose von "Störungen".

Beispiel:

P1: Alle Kaninchen aus dem Hut sind weiß.
P2: Diese Kaninchen sind weiß.

Z: ⇒ (wahrscheinlich) Diese Kaninchen sind aus dem Hut.

Die Schlussfolgerung Z gilt nur mit einer bestimmten Wahrscheinlichkeit.

Beispiel:

P1: Alle grünen Fliegenpilze sind stark giftig.
P2: Dieser Pilz ist eine grüner Fliegenpilz.
Z: Dieser Pilz ist giftig.

Gültige Deduktion

P1: Alle grünen Fliegenpilze sind stark giftig.
P2: Dieser Bleistift ist Fliegenpilzgrün
Z: Dieser Bleistift ist giftig.

Richtiges Urteil, Schlussfolgerung Falsch ⇒ Mindestens eine Prämisse ist unwahr.
Das Urteil hat die richtige logische Form.
Logische Konnektive - "und", "oder", "wenn", "falls", "dann" und andere haben feste Bedeutungen, interpretieren sie mit elementaren Aussagen oder Teilen davon (Prädikate und funktionale Begriffe).

Logik ist ein Werkzeug, das hilft, die Beziehung des logischen Ergebnisses zu entdecken, Aufgaben wie "Was bedeutet das?" zu lösen. Es hilft unserer Intuition, die manchmal scheitern kann, weil Prämissen komplex formuliert, verflochten und negiert werden können, die Beziehung des Auftretens ist auf den ersten Blick nicht offensichtlich.

Beispiel:

P1: Alle Männer mögen Fußball und Bier.
P2: Einige Bierliebhaber mögen Fußball nicht.
P3: Xaver mag nur Liebhaber von Fußball und Bier.
Z: Einige Frauen mag Xaver nicht.

Im Wesentlichen, wenn alle Annahmen wahr sind, dann muss die Schlussfolgerung wahr sein.

Ist das Urteil gültig?

Natürlich, wenn Xaver nur Fußball- und Bierliebhaber mag (3.), mag er einige Bierliebhaber nicht (diejenigen, die Fußball nicht mögen (2.)), mag er (nach 1.) einige "Nicht-Männer", d.h. Frauen nicht.

Sie ist jedoch unter der Definition nicht gültig.

Die Beurteilung ist im Bedarfsfall gültig, d.h. unter allen Umständen (Interpretationen), wenn wahre Annahmen wahr und wahr sind.

Aber: In unserem Beispiel müssten diejenigen, die keine Männer sind, nicht als Frauen interpretiert werden. Es gibt hier keine Prämisse, "wer kein Mann ist, ist eine Frau", ebenso brauchen wir noch die Prämisse "wer ein Liebhaber von etwas ist, das er mag".

Wir müssen alle notwendigen Annahmen treffen, um die Schlussfolgerung zu ziehen.

P1: Alle Männer mögen Fußball und Bier.
P2: Einige Bierliebhaber mögen Fußball nicht.
P3: Xaver mag nur Liebhaber von Fußball und Bier.
P4: Wer kein Mann ist, ist eine Frau.
P5: Wer etwas liebt, dem gefällt es.
Z: Einige Frauen mag Xaver nicht.

Nun ist das Urteil richtig, es hat eine gültige logische Form.
Die Schlussfolgerung ergibt sich logischerweise aus den Annahmen (sie sind "informativ, deduktiv einbezogen").

Logische Eigenschaften

Ein gültiges (richtiges) Urteil kann zu einer falschen Schlussfolgerung führen.

Verwendung: Nachweis AD ABSURDUM

Monotonie

Wenn ein Urteil gültig ist, führt die Ausweitung des Satzes von Annahmen auf eine weitere Annahme nicht zu einer Änderung der Gültigkeit des Urteils.

Aus den umstrittenen Annahmen wird jede Schlussfolgerung gezogen.

Reflexivität, Übergang

Gültige Beurteilungsschemata

$A \supset B, A \mid= B$

modus ponens
P1: Keine Primzahl teilbar 3
P2: 9 ist teilbar 3
Z: 9 ist keine Primzahl.

A⊃¬B,B|=¬A

Modus Ponens + Transposition
P1: Alle Menschen sind vernünftig.
P2: Der Stein ist nicht vernünftig.
Z: Der Stein ist kein Mensch.

A⊃B,¬B|=¬A

Modus Ponens + Transposition
P1: 12 eine Primzahl wenn nicht teilbar durch 3
P2: 12 ist teilbar 3
Z: 12 ist keine Primzahl.

¬A∨¬B,B,B|=¬A

Beseitigung der Disjunktion
P1: 12 ist keine Primzahl oder nicht teilbar 3
P2: 12 ist teilbar 3
Z: 12 ist keine Primzahl.

3. MENGEN UND BEZIEHUNGEN

3.1. Mengen

Definition - eine Zusammenfassung von Objekten, die genau definiert und identifizierbar sind und Teil der Welt unserer Ideen und Gedanken sind; diese Objekte werden Elemente der Menge genannt.

Wir verwenden das folgende Symbol für Mengen:

- Mengen: A, B, C....
- Elemente: a, b, c .....
- Element Sets: $a \in A$
- Für alle x gilt P : $\forall x : P$
- Es gibt x, für die P gilt: $\exists x : P$
- Es gibt nur ein x, das P hält: $\exists! x : P$
- Konjunktionen, Disjunktionen, Negationen: $\wedge$, $\vee$, $\neg$
- Implikationen, Äquivalenzen: $\Rightarrow$, $\Leftrightarrow$
- Summe, Multiplikation: Σ, Π

Die Menge kann wie folgt hergestellt werden

- Aufzählung der Elemente : A=1,2,3,4,5
- Mit Hilfe der charakteristischen Merkmale: $A=a \in N \mid a<6$

Wir definieren eine leere Menge: $\emptyset$, die keine Elemente enthält.

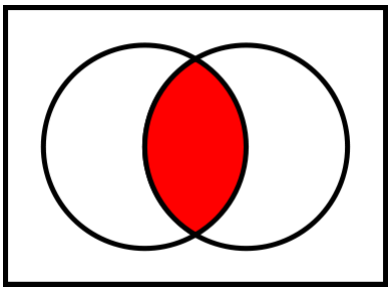
Die Anzahl der Elementelemente bestimmt - Kardinalität - | A | A |, wobei A eine Menge ist.

Wir erkennen Mengen:

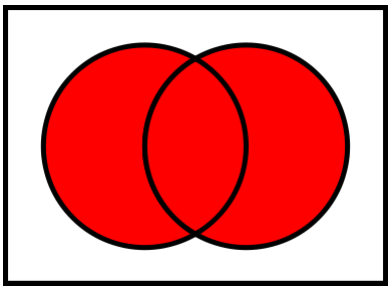
- Endlich
- Unendlich
- Zählbar
- Kontinuum

Grundlegende Operationen mit Mengen

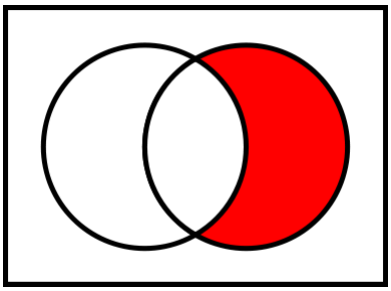
Schnittpunkt: A∩B



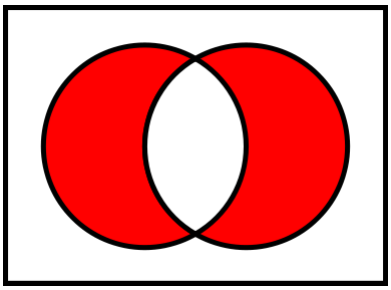
Vereinigung: A∪B



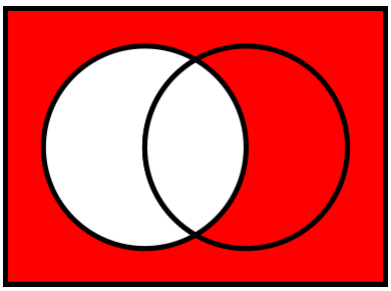
Unterschied der Mengen: B\A



Symmetrische Differenz: A∆B



Komplementär von A in U: A^c=U\A



Teilmenge (Einschließung - das Gegenteil von Ausschluss): A⊆B

(∀x)(x∈A⇒x∈B∈B)

Potentialsatz - Satz aller Teilmengen:

P(∅)=∅
P({a})={∅,{a}}
P({a,b})={∅,{a},{b},{a,b}}

3.2. Beziehungen

Definition: n-te Beziehung zwischen den Mengen A 1, A 2, A 3, ..., A n, wobei n ∈ N, wir meinen jede Untermenge des kartesischen Produkts von n Mengen.

Kartesisches Produkt (diskretes Produkt)

Set X×Y, das alle geordneten Paare enthält, wobei der erste Eintrag von X und der zweite Eintrag z Y ist.

A\B	1	2	3
X	(X,1)	(X,2)	(X,3)
Y	(Y,1)	(Y,2)	(Y,3)
Z	(Z,1)	(Z,2)	(Z,3)

Beziehungen können nach Arität klassifiziert werden zu:

- Unär - jede Teilmenge der Menge N
  Beispiel: Eine positive Zahl ist eine wahr / falsche Aussage.

- Binär - Jeder Satz von geordneten Paaren[x; y]∈M2
  ZB: ist größer als, ist kleiner als, ist eine Teilmenge

- Ternär - Jeder Satz von geordneten Drillingen[x; y; z]∈M3.
  Beispiel: liegt zwischen

- N-te - jeder Satz von geordneten n-Tupletten von Mn.

Operationen mit Beziehungen

Zusätzlich zu den klassischen Mengenoperationen (alle Relationen müssen die gleiche Arithmetik haben), führen wir die umgekehrte Beziehung als: R-1: ∀ a ∈ M1 ∀ b ∈ M2:[a,b] ∈ R ⇔ [b,a] ∈ R-1 und eine zusammengesetzte Beziehung.

Die binäre Beziehung auf dem Set ist:

- Reflexiv, wenn für alle a ∈ M gilt, dass [a, a] ∈ R.

- Symmetrisch, wenn für alle a, b ∈ M gilt, wenn [a, b] ∈ R dann [b, a] ∈ R.

- Antisymmetrisch wenn für alle a, b ∈ M gilt, dass, wenn [a, b] ∈ R und auch [b, a] ∈ R ist, dann a = b.

- Transitiv, wenn für alle a, b, c ∈ M gilt, dass wenn [a, b] ∈ R während [b, c] ∈ R dann [a, c] in R ist.

Zwei besondere Beziehungen

- Äquivalenz ist eine Beziehung, die reflexiv, symmetrisch und transitiv ist.
- Ordnung ist eine Beziehung, die reflexiv, antisymmetrisch und transitiv ist.

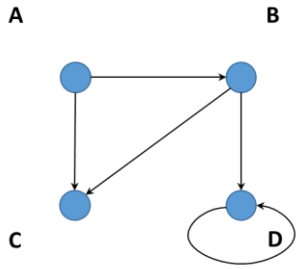
4. BEZIEHUNGSSTRUKTUREN

Unter relationaler Struktur versteht man eine mathematische Struktur, die zusätzlich zum Trägersatz eine oder mehrere Beziehungen enthält (die beliebige und gleichzeitig unterschiedliche arithmetische Eigenschaften aufweisen können). Dazu gehören orientierte Diagramme und geordnete Mengen (einschließlich ihrer Sonderfälle, linear oder gut angeordnete Linien).

Gerichtete Diagramme

Ein gerichteter Graph G ist ein Paar (V, E), wobei E eine Teilmenge des kartesischen Produkts $V \times V$, $E \subseteq V \times V$ ist.

Die Elemente E werden als Pfeile oder orientierte Kanten bezeichnet. Die orientierte Kante e hat die Form (x, y). Wir sagen, dass diese orientierte Kante auf x basiert und in y endet. Gerichtete Diagramme werden durch eine Adjazenzmatrix oder eine Liste von Knoten und Kanten dargestellt.



Dazugehörige Matrix:

		WHERE			
		A	**B**	**C**	**D**
FROM	**A**	0	1	1	0
	B	0	0	1	1
	C	0	0	0	0
	D	0	0	0	1

Für gerichtete Diagramme kann es sein, dass sie nicht symmetrisch sind, 1 ist eine Kante, 0 Kante fehlt.

Der Satz von Eckpunkten und Kanten G: (A, [A, B]), (A,[A,[A, C]) (B,[B, C] D, D])

Geordnete Mengen

Die lineare Anordnung (= Gesamtordnung) ist so definiert, dass:

- Jedes der beiden Elemente der linear geordneten Menge ist vergleichbar, d.h. es besteht eine Beziehung R auf der Menge X und a, b, c ∈ X so dass:
  - Transitivität: $(\forall x,y,z \in A)((xRy \wedge yRz) \Rightarrow xRz)$
  - Schwache Antisymmetrie: $(\forall x,y \in A)((xRy \wedge yRx) \Rightarrow x=y)$
  - Trichotomie: $aRb \vee bRa \vee a=b$

Beispiel: Anordnung auf einer Menge von natürlichen und reellen Zahlen

Eine gute Anordnung ist definiert als:

- Der Satz S ist übersichtlich, wenn jeder nicht leere Teil des bestellten Satzes S das kleinste Element hat.

Beispiel: Natürliche Zahlen

Ein geordneter Satz ist einer, auf dem eine Anordnung definiert ist.

Die Anordnung ist eine binäre Beziehung R, die reflexiv, transitiv und schwach antisymmetrisch ist.

$(\forall x \in A)(xRx)$ Reflexivität - jedes Element befindet sich in einer R - Beziehung zu sich selbst

$(\forall x,y,z \in A)((xRy \wedge yRz) \Rightarrow xRz)$ Transitivität - wenn das Element des Sets in der Anordnung zwischen zwei anderen Elementen liegt, sind die beiden auch vergleichbar.

$(\forall x,y \in A)((xRy \wedge yRx) \Rightarrow x=y)$ schwache Antisymmetrie (keine Zyklen in der Anordnung) = unscharfe Anordnung

Scharfe Anordnung - Reflexivität ersetzt durch Antireflexivität - kein Element steht in Beziehung zu sich selbst $(\forall x \in A)(\neg(xRx))$

5.ABBILDUNG

Ein spezielles Beispiel für eine binäre Beziehung, die ein eindeutiges Bild zu jedem Muster gewährleistet.

Wenn es sich um eine binäre Relation einer numerischen Menge handelt, dann nennen wir sie Funktion.

Definition:

Abbildung f von der Menge A zu der Menge B ist eine solche binäre Beziehung, bei der jedes Element x in A maximal einem Element y in B zuordnet ist, so dass x,y∈f

Bezeichnung: y=fx f:x→y

Wichtiger Hinweis: y=fx∈B, x∈A

y - Das Ziel des Elements x in der Abbildung f, Funktionswert f im Punkt x
x - Quelle des Elements y in der Abbildung f

Eine Menge von Elementen x∈A, für die es nur ein solches Element y∈B gibt, das y=fx, die Domäne von f genannt wird.

B-Elementsatz y∈B, für den es mindestens ein solches Element x∈A gibt, dass fx=y als der Bereich von f

Wir unterscheiden mehrere Grundtypen von Darstellungen:

Darstellungen von Mengen

f:A→B, A=B

Menge in Menge - die gesamte Menge A ist aus einem Bereich definiert.
F
:A→B, D(f)=A

Von Menge zu Menge (Surjektion) - die gesamte Menge B ist ein Wertebereich.

f:A→B, H(f)=B, ∀b∈B:∃a∈A:f(a)=b

Mengen in Mengen

f:A→B, D(f)=A∧H(f)=B,
∀a∈A:∃!b∈B:b=f(a) ∧∀b´∈B:∃!a´∈A:f(a´)=b´

Einfache (injektion)

$\forall b \in B: \exists! a \in A: f(a)=b$

Gegenseitige eindeutige (Bijektion) - Injektionsfunktion von Set A auf Set B (ist injektiv und surjektiv zugleich)

$f: A \leftrightarrow B$

Invers - Existiert nur für einfache Darstellung

Wenn es $f: A \rightarrow B$ einfach ist, dann wird es dargestellt.

$f^{-1}: B \rightarrow A$, které $\forall b \in H(f): f^{-1}(b)=a \in D(f): y=f(x)$, heißt invers.
$D(f^{-1})=H(f)$, $f^{-1}(y)=x \Leftrightarrow f(x)=y$

Abbildungen können auch zusammengesetzt werden:

Mengen A, B, C

Darstellung: $f: A \rightarrow B$ und $g: B \rightarrow C$

Zusammengesetzte Darstellung: $h: A \rightarrow C$, $h=g \circ f$, $h(a)=g(f(a)) \ \forall a \in A$
Sie ist in der Regel nicht kommutativ, sondern assoziativ.

Umkehrung eines komponierten Morphismus: $(g \circ f)^{-1}=f^{-1} \circ g^{-1}$

6.BOOLESCHE ALGEBRA

Boolesche Algebra ist definiert als: Sechste (A, ∧, ∨, -, 0, 1,) wobei A keine Leere Menge ist, 0∈A - das kleinste Element, 1∈A - das größte Element, - die einzige Operation (Komplement), ∧ der Schnitt, das logische Produkt, ∨ die logische Summe.

Es basiert auf Axiomen:

Kommutativität

$$x \vee y = y \vee x$$

$$x \wedge y = y \wedge x$$

Distributivität

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

Neutralität 0 und 1

$$x \vee 0 = x$$

$$x \wedge 1 = x$$

Komplementarität

$$x \vee \text{-} x = 1$$

$$x \wedge \text{-} x = 0$$

A hat die folgenden Eigenschaften:

- Absorption: x ∨ (x ∧ y) = x, x ∧ (x ∨ y) = x
- Aggression von Nullen: x ∧ 0 = 0
- Aggression von einem: x ∨ 1 = 1

- Idempotenz : x v x = x, x ∧ x = x
- Negation der Absorption: x v (- x ∧ y) = x v y, x ∧ (- x v y) = x ∧ y
- Doppelte Negation: - (- x) = x x
- Die Gesetze von De Morgan: - x ∧ - y = - (x v y), - x v - y = - (x ∧ y)
- 0 und 1 ergänzen sich gegenseitig: -0 = 1, -1 = 0

Grundlegende Ein-Eingangs-Operationen:

- ID (Identität)
- NICHT (negieren)
- Zwei grundlegende Funktionen des Eingangs.
- ODER (logische Summe)
- UND (logisches Produkt)

Wahrheitstabelle für grundlegende Funktionen:

A	B	ID(A)	ID(B)	NOT(A)	NOT(B)	A OR B	A AND B
0	0	0	0	1	1	0	0
0	1	0	1	1	0	1	0
1	0	1	0	0	1	1	0
1	1	1	1	0	0	1	1

Abgeleitete Zwei-Eingangs-Operationen:

- NOR (Not OR - Negation der Summe)
- NAND (Nicht UND - negiert das Produkt)
- Implikation ⇒ (bei Erfüllung der Annahme A resultiert B, bei nicht erfüllter Erwartung resultiert irgendetwas und damit 1)
- Äquivalenz von EQ ⇔ (Anpassung der Eingänge)
- XOR (Exklusives ODER - Einzigartige Eingänge)

Wahrheitstabelle für abgeleitete Zwei-Eingangsoperationen:

A	B	NOR	NAND	⇒	⇔	XOR
0	**0**	1	1	1	1	0
0	**1**	0	1	1	0	1
1	**0**	0	1	0	0	1
1	**1**	0	0	1	1	0

Nehmen wir den Ausdruck: A (B + NICHT (C))

Dieser Ausdruck kann beschrieben werden mit Hilfe von:

Wahrheitstabellen:

i	A	B	C	A(B+NOT(C))
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Algebraischer Ausdruck:

UND (NICHT (B) (NICHT (C)) + AB (NICHT (C))) + AB (NICHT (C)) + ABC
Dies ist die Summe aller Ausdrücke, die das Ergebnis 1 haben.

Setzen Sie den Zustandsindex so, dass die Zeilen in der Wahrheitstabelle mit der Nummer 0 versehen sind (siehe Spalte i).

A (B+NOT(C)) = {4, 6, 7}

Um den algebraischen Ausdruck zu minimieren, kann Karnaughs Karte verwendet werden.

Es wird so ausgefüllt, dass die Spalten (Zeilen), über denen die Variable angegeben wird, für sie 1 sind.

Die Tabelle zeigt die Werte nach dem Alphabet.

			B	
		C		
	000	001	011	010
A	100	101	111	110

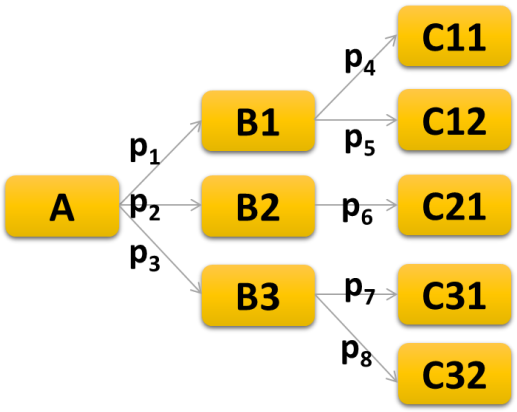
Beispiel: A (NOT (B) (NOT (C)) + A (NOT (B)) C + ABC = A

			B	
		C		
	0	0	0	0
A	1	0	1	1

7. WAHRSCHEINLICHKEIT UND STATISTIKEN

Der Unterschied zwischen Determinismus und Stochastik besteht darin, dass der Determinismus im Voraus bekannt ist. Von A nach B geht mit 100%iger Wahrscheinlichkeit (ich gehe immer diesen Weg, eine andere Option gibt es nicht).

Im Gegensatz dazu geschieht in den stochastischen Phänomenen alles nur mit einer gewissen Wahrscheinlichkeit, von A bis B gehe ich mit der Wahrscheinlichkeit p 1. Aber ich kann auch zu B2 mit der Wahrscheinlichkeit p 2 und zu B3 mit p3 gehen. Das heißt, ich kenne nur die Wahrscheinlichkeit, mit der das Ergebnis passiert. Zufällige Prozesse spielen in diesem Phänomen eine Rolle. Wenn wir uns das Bild ansehen, muss die Summe von p1, p2 und p3 gleich 1 sein, oder von A gehe ich zu irgendeinem B mit 100% Wahrscheinlichkeit. Ebenso ist die Wahrscheinlichkeit p6 (B2-> C21) gleich eins.



Die Wahrscheinlichkeit ist wie folgt definiert:

Sei F das Experiment E, der Satz von Bedingungen des Experiments Π und der Satz aller möglichen Ergebnisse F. F enthält die Ereignisse 0, A, B..... Ω, wobei 0 ein Ereignis ist, das nie eintritt und Ω ein Ereignis, das immer eintritt. Daher kann das Experiment mit E = (Ω, F) geschrieben werden. Die Menge der unter den gleichen Bedingungen wiederholten Versuchsergebnisse ist N, die einzelnen Ergebnisse sind 1... n und I ist eine Teilmenge der Ergebnisse. Teilmenge A(I) ist Teilmenge von I, so dass das Ergebnis des Experiments ein Ereignis A ist. Die Anzahl der Ergebnisse in der Teilmenge A (I) ist n A (I) und die Menge aller Ergebnisse ist Ω (I) . Dann gelten alle Teilmengen, die ich anwende, wobei P (A) die

Wahrscheinlichkeit des Phänomens A ist:

$$P(A) \approx \frac{n_A(I)}{n_\Omega(I)}$$

Die relative Frequenz des A-Phänomens kann definiert werden als: $\frac{n_A(I)}{n_\Omega(I)}$

Wahrscheinlichkeitseigenschaften - die Wahrscheinlichkeit von A liegt immer zwischen 0 und 1, einschließlich beider Extremwerte.

Die Wahrscheinlichkeit, dass das Phänomen A nicht auftritt, oder die Negation des Phänomens A ist 1-P (A).

Wenn die Phänomene A und B nicht gleichzeitig auftreten, dann ist P A∩B=∅

Eintrittswahrscheinlichkeit des Phänomens A oder B: P(A∪B) ist PA∪B=PA+P(B)

Bedingte Wahrscheinlichkeit - Wahrscheinlichkeit eines Phänomens A, wenn Phänomen B auftritt:

$$(A|B) = \frac{P(A \cap B)}{P(B)}; P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Bayes-Satz : PA|B·PB=PB|A·PA

Statistiken sind eine Reihe von Konzepten, Regeln und Verfahren, die uns helfen, numerische Informationen im Hintergrund zu organisieren, um die Techniken zu verstehen und fundierte Entscheidungen zu treffen.

Statistisches Wissen wird auf die Daten angewendet, d.h. die Fakten und Informationen aus Beobachtungen (idealerweise aus Experimenten).

Die Daten werden unterteilt in numerische (quantitative) Daten, die das Ergebnis der Messung sind, und kategorische (qualitative) Daten, die in Gruppen unterteilt sind, die auf gemeinsamen Merkmalen basieren.

Die statistischen Basisvariablen sind der Mittelwert (gewichteter Durchschnitt), der Median, der die zentrale Tendenz, den Mittelwert in der gegebenen Menge und den Modus als den häufigsten Wert darstellt. Als nächstes berechnen wir die Varianz und daraus die Standardabweichung als Wurzel. Wir können auch eine Kovarianz finden, die bestimmt,

wie sehr sich die beiden Werte zusammen ändern. Und dann Schiefe und Kurtosis, die die Verteilung der Werte für eine bestimmte Datei aufdecken.

Die Verteilung der Dateiwerte beschreibt die sogenannte Verteilung. Eine der Grundverteilungen ist die Gaußsche (Normal-)Verteilung. Es ist symmetrisch, mit dem gleichen Mittelwert, Modus, Median, Schiefe und Kurtosis sind Null.
Eine der grundlegenden Statistiken ist das Central Limit Theorem, das die Tatsache ausdrückt, dass die Summe vieler unabhängiger Zufallsvariablen dazu neigt, sich in einem kleinen Satz von Verteilungsfunktionen zu verteilen.

8. INFORMATIONSTHEORIE

Die Informationstheorie beschäftigt sich mit der Messung, Übertragung, Verschlüsselung, Speicherung und Weiterverarbeitung von Informationen in quantitativer Hinsicht. Ihr Gründer ist C.E. Shannon.

Informationen

Was wir mit der Außenwelt austauschen, wenn wir uns an sie anpassen und sie beeinflussen, durch unsere Anpassung.

Inhalt der Nachricht, Kommunikation, Klärung, Erklärung, Anweisung.

Daten, Zahlen, Zeichen, Befehle, Anweisungen, Befehle, Nachrichten und dergleichen. Betrachten Sie zur Information auch die Ideen und Wahrnehmungen, die von lebenden Organismen empfangen und abgegeben werden.

Die Größeninformationen werden als Ordnung (Unsicherheit) der Elemente (System) bereitgestellt.

Die Systeminformationen sind umso größer, je größer die Wahrscheinlichkeit des Auftretens einzelner Zustände ist. Die Informationen sind größer, wenn die Nachricht etwas Neues enthält, das nicht bekannt war oder nicht leicht erraten werden konnte (es ist unwahrscheinlich).

Für die Weitergabe werden die Informationen benötigt.

Jede Methode der Kodierung = Übertragung auf geeignete Signale oder Symbole

Signal - informationstragende physikalische Größe

Nachricht - Ausdruck von Informationen unter Verwendung einer Folge von Symbolen (Zeichen)

Der Bericht als solcher besteht aus drei Teilen:

- Syntax - Track
- Syntaktische Inhalte
- Es kann selbst ohne semantische und pragmatische Inhalte existieren.
- Sie bezieht sich auf die gegenseitige Anordnung von Zeichen als Informationsträger.
- Beschreibt die Seite mit den quantitativen Informationen.
- Semantik - Informationsgehalt
- Semantischer Inhalt

Die Bedeutung der Nachricht kann nicht gemessen werden.

Eine Nachricht mit der gleichen Informationsmenge kann in verschiedenen Sprachen verfasst werden.

Beschreibt die qualitative Informationsseite

Bedeutung - pragmatische Inhalte

Bestimmt die Bedeutung (Nützlichkeit, Wert) der Nachricht und die Priorität.

Qualitative Informationsseite

Der Informationsgehalt der Nachricht wird durch Entropie gemessen.

Informationen sind ein Maß für die Höhe der Unsicherheit oder Unsicherheit über jede zufällige Handlung, die durch die Realisierung dieses Ereignisses eliminiert wird.

Wir bestimmen den Grad der Unsicherheit als die Entropie des Systems.

$$H(X) = -\sum_{i=1}^{n} p(x_i)p(x_i)$$

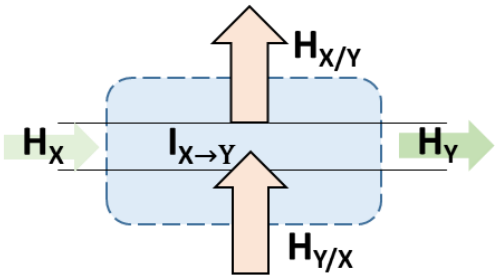
Informationen über System X erhalten Sie bei:

- Direkter Beobachtung
- Indirekt über ein System von Y (X-System ist für uns nicht verfügbar)

Der Systemstatus Y ist nicht unbedingt identisch mit dem Zustand von X (der Text des Telegramms in einer Stadt X, Y in der zweiten Stadt).

Unterschiede zweierlei Art

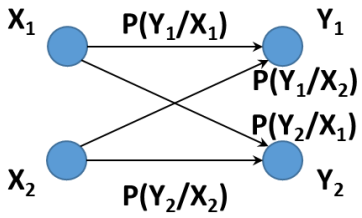
- Einige Zustände des X-Systems erscheinen in einem Zustand Y (Y kann die Feinheiten in X nicht unterscheiden, Y ist gröber als X).
- Übertragungsfehler zwischen X und Y - z.B.: Rauschen
- Verwendung eines Kanals



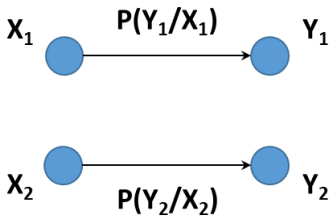
H X / Y - durchschnittlich verlorene Informationen
H Y / X - durchschnittliche Störinformation
I X X → Y - gegenseitige Information

Für Kanalinformationen unterscheiden wir **zwei Arten von Kanälen**:

Laut (störend):



Lautlos:



P (Y 1 / X 1) Wahrscheinlichkeit beim Senden des Elements X 1 empfangen wir Y 1
P (Y 2 / X 2) Wahrscheinlichkeit beim Senden des Elements X 2 empfangen wir Y 2
P (Y 1 / X 2) Wahrscheinlichkeit beim Senden von Element X2 empfangen wir Y 1
P (Y 2 / X 1) Wahrscheinlichkeit beim Senden des Elements X 1 empfangen wir Y 2

Kanalpermeabilität (Kanalkapazität) - Kanalfähigkeit zur Übertragung von Informationen. Maximale Informationen, die pro Zeiteinheit übertragen werden können.

$C = B\left(1 + \frac{S}{N}\right)$, wobei B die Kanalbreite, S das Signal und N das Rauschen ist.

Abtasttheorem

Eine genaue Rekonstruktion eines kontinuierlichen, frequenzbegrenzten Signals aus seinen Abtastwerten ist möglich, wenn die Abtastrate größer als das Doppelte der höchsten harmonischen Komponente des abgetasteten Signals ist.

Die minimal mögliche Länge der Signalelemente

$\tau_0 = \frac{1}{2F_m}$, F ist die Grenzfrequenz, Bandbreite.

9. ANWENDUNG DER INFORMATIONS-THEORIE

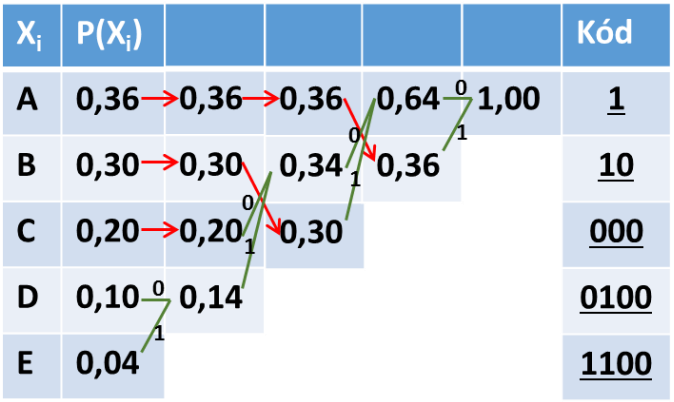
Beispiele sind Huffman-Kodierung, arithmetische Kodierung, LZW, JPEG, MP3, TIFF, Fehlererkennungs- und Fehlerkorrekturcodes, statistische Anwendungen und minimal description length (MDL).

Die Huffman-Kodierung ist ein Algorithmus zur verlustfreien Datenkompression. Konvertiert die Zeichen der Eingabedatei in Bitstrings unterschiedlicher Länge. Die häufigsten Symbole in Bitstrings mit der kürzesten Länge (sogar 1bit). Weniger häufige in längere Ketten (kann länger als 8 Bit sein).

Es besteht aus zwei Phasen

- Übergibt die Datei und erstellt Statistiken.
- Erstellt einen binären Baum, um Daten zu komprimieren.
- Der Code wird von den Blättern bis zur Wurzel erzeugt.

Beispiel:



Die Shannon-Fano-Kodierung ist die statistische Methode der verlustfreien Kompression. Huffman's Kodierung unterscheidet sich nur im Binärbaum-Design. Der Zeichensatz wird rekursiv in zwei etwa gleiche Teilmengen unterteilt. Einem Subcode wird dann eine binäre 1 und der zweite 0 zugewiesen, so dass der Code von der Wurzel bis zu den Blättern aufgebaut ist, im Gegensatz zur Huffman-Codierung, welcher von den Blättern bis zur Wurzel gebildet wird, und deshalb nicht optimal sein muss.

Beispiel:

Char	p(x)	s	Groups				Code
A	0,36	1	0				0
B	0,3	0,64	1	0			10
C	0,2	0,34		1	0		110
D	0,1	0,14			1	0	1110
E	0,04	0,04				1	1111

Die arithmetische Kodierung ist eine Los-Längenkompression von Daten variabler Länge eines Codeworts. Es kodiert den gesamten Text in eine einzige Zahl, einen Bruchteil n∈<0;1). Es ist anstrengend, mit realen Zahlen zu berechnen. Für einen längeren Bericht wären wir nicht mehr in der Lage, die erforderliche Präzision zu erreichen, und einige Subintervalle könnten beginnen zu verschmelzen. Es verwendet eine Blockkompression, bei der die Blöcke groß genug sind, um eine ausreichende Genauigkeit bei der Verteilung zu gewährleisten.

Informationen können während der Übertragung gesichert werden, um Fehler zu erkennen und zu korrigieren und unbefugtes Lesen zu verhindern.

Codes zur Fehlererkennung und –behebung

Parität - Daten Für jedes Segment wird ein anderes Bit verwendet, dessen Wert der Anzahl der binären Komplemente der Zahl ungerade oder gerade ist (ungerade / gerade Parität).
Gerade: 10011011 10010101 → 100110111 1001010101010 100101010
Ungerade: 10011011 10010101 → 100110110 1001010 100101011

CRC - Prüfsumme - Die Daten werden in Abschnitte der erforderlichen Länge (8, 16, 32 Bit) unterteilt und diese Segmente werden ohne Übertragung nach den Bits hinzugefügt. Das resultierende Datensegment verbindet sich mit den übertragenen Daten.

00001101	Data
00010000	
01000100	
10010000	
11110001	**CRC**

Hamming-Code ist ein linearer Code, der in der Telekommunikation verwendet wird, um bis zu zwei fehlerhafte Bits zu erkennen oder ein schlechtes Bit zu reparieren.

Algorithmus

Alle Bitpositionen, die ein Vielfaches von 2 sind, werden für das Paritätsbit (1, 2, 4, 8, 16, 32,....) verwendet.

Alle anderen Bitpositionen gehören zum kodierten Informationswort (3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17,....).

Jedes Paritätsbit wird aus einigen der Informationswortbits berechnet. Die Position des Paritätsbits gibt die Reihenfolge der Bits an, die im Codewort gescannt und übersprungen werden.

Für das Paritätsbit p 1 (Position 1) wird im Rest des Codeworts 1 Bit übersprungen, 1 Check, 1 Skip, 1 Check,

Für Paritätsbit p 2 (Position 2) wird das erste Bit übersprungen, 2 Checks, 2 Skips, 2 Checks, etc.

Für p 3 (Position 4) werden die ersten 3 Bits übersprungen, 4 werden überprüft, 4 werden übersprungen, 4 werden überprüft,

Gegen unbefugtes Lesen

- Verschlüsselung – Kryptographie; Historisch
- Stenographie - Verstecken der Botschaft
- Ersetzungschiffre - ersetzt jedes Zeichen durch ein anderes durch eine Regel.
- Zeichenverschiebung - Caesars Chiffre - jeder Buchstabe des Alphabets wurde auf eine feste Anzahl von Positionen verschoben.
- Tabellen der Ersetzungen - Ersetzen eines Zeichens durch ein anderes ohne internen Kontext oder Kenntnis des Schlüssels.
- Vigener Cipher - verwendet Passwörter, deren Zeichen den Offset des offenen Textes bestimmen, so dass der offene Text in Zeichenblöcke unterteilt wird, solange das Passwort gültig ist und jedes Zeichen mit dem entsprechenden Passwort-Zeichen hinzugefügt wird.
- Verman's Chiffre - die einzige bekannte Chiffre, die sich bisher als nicht nachweisbar erwiesen hat, basiert auf der Hinzufügung von offenen Buchstaben und Passwörtern, aber das Passwort ist ein Block von zufällig ausgewählten Daten der gleichen Größe wie der offene Text.

Transpositionsraster

- Skytalé - eine Art von Verschlüsselung, die aus einem Zylinger und einem Wickelpapier oder Pergament darauf besteht, auf das eine Nachricht geschrieben wird. – Modern

- Symmetrische Verschlüsselung - eine herkömmliche Verschlüsselung, verwendet einen einzigen Schlüssel zum Ver- und Entschlüsseln, DES (Data Encryption Standard), das aktuelle AES (Advanced Encryption Standard).
- Asymmetrische Verschlüsselung - Verwenden Sie verschiedene Schlüssel (privater und öffentlicher Schlüssel), RSA, PGP (OpenPGP) zur Ver- und Entschlüsselung, verwendet für die elektronische Signatur.
- Hash-Funktion - eine einseitige mathematische Funktion, die beispielsweise zur Sicherstellung der Datenintegrität verwendet wird.

Unterzeichnung

- Elektronische Signatur - Kennzeichnung bestimmter Daten, die eine klassische handschriftliche Unterschrift oder eine beglaubigte Unterschrift am Computer ersetzen. Es ist mit einer Datennachricht versehen oder logisch mit ihr verbunden, es ermöglicht die Authentifizierung der signierten Person in Bezug auf die Datennachricht. Eine elektronische Signatur ist ein Mittel zur Überprüfung der Identität des Absenders.

10. THEORIE DER KOMPLEXITÄT

Die Komplexitätstheorie konzentriert sich auf die Klassifizierung von Rechenproblemen nach ihrer eigenen Komplexität und der Beziehung zwischen den Klassen. Das Problem ist eine Aufgabe, die auf einem Computer gelöst werden kann. Das Problem gilt als schwierig, wenn seine Lösung erhebliche Ressourcen erfordert, unabhängig davon, welcher Algorithmus verwendet wird. Es formalisiert diesen Ansatz und führt Computermodelle zur Untersuchung und Quantifizierung der Menge der Ressourcen durch, die zur Lösung von Problemen benötigt werden (Zeit, Speicher). Eine weitere Komplexitätsstufe ist viel Kommunikation, Gate-Zähler der Schaltung, die Anzahl der Zugriffe auf den Cache und die Anzahl der Prozessoren. Eines der Ziele ist es, die praktischen Grenzen dessen zu bestimmen, was Computer berechnen können und was nicht.

Analyse von Algorithmen vs. Theorie der Komplexität vs. Theorie der Berechenbarkeit

- Die Algorithmusanalyse beschäftigt sich mit der Menge der Ressourcen, die von einem bestimmten Algorithmus benötigt werden.
- Die Komplexitätstheorie identifiziert allgemeinere Fragen zu allen Algorithmen, die zur Lösung eines bestimmten Problems verwendet werden können.
- Der Versuch, Probleme nach den Einschränkungen der verfügbaren Ressourcen zu klassifizieren, führt zu Einschränkungen der verfügbaren Ressourcen.
- Die Quantifizierungstheorie fragt, welche Probleme grundsätzlich algorithmisch gelöst werden können.

Die Grundlage ist ein Rechenproblem

Problem: Unendliche Sammlung von Beispielen und Lösungen für die Situation, Eingabe von Input, d.h. Eine Instanz des Problems kann nicht mit dem Problem selbst verwechselt werden. Das Problem muss unabhängig von seiner Zuordnung angegangen werden.

Beispiel: Prime Number Test - Input: Anzahl; Ausgabe: Ja / Nein

Sie können das Problem auf verschiedene Arten erfassen: Der grundlegendste Weg ist die Form eines ausgesprochenen Satzes. Für die Bearbeitung per Computer ist eine Übersetzung in die Sprache der Computer notwendig. Mathematische Aufgaben in Graphen werden z.B. durch Matrizen festgelegt.

Das Entscheidungsproblem ist die Grundart der Probleme der Komplexitätstheorie. Sie hat nur zwei Ausgaben Ja / Nein. In der Sprache, in der die Mitglieder der Sprache die Fälle

mit der Antwort JA und die anderen Mitglieder mit der Antwort NEIN sind, sollten Sie einen Algorithmus verwenden, um zu entscheiden, wie die angegebene Eingabe mit dieser Sprache übereinstimmt. Wenn JA, dann gibt der Algorithmus die Eingabe zurück und wird akzeptiert, andernfalls abgelehnt. Der Algorithmus berechnet ein charakteristisches Merkmal der Sprache.

Formale Sprachen (und deren Probleme) sind einige der Alphabete, die nicht unbedingt binär sind.

Die Problemfunktion ist ein Rechenproblem, bei dem ein Ausgang der Gesamtfunktion für alle möglichen Eingaben gilt, aber komplexer ist als der Ausgang der Entscheidungsfunktion. Das Problem der Funktion ist umfangreicher als das Entscheidungsproblem. Es kann auch wie folgt auf das Entscheidungsproblem umgestellt werden: Wir wollen zwei Zahlen multiplizieren, die Antwort auf das Entscheidungsproblem ist ein Tripel (a, b, c) und die Antwort JA wird nur ausgegeben wenn gilt: a * b = c.

Für die Komplexitätstheorie müssen Sie die Eingabegröße messen. Die Größe der Eingabe hängt davon ab, wie lange es dauert, den Algorithmus zu verarbeiten. Diese Teilprobleme, die der für die Lösung benötigte Platz sein können, verarbeiten einen separaten Algorithmus und beziehen sich alle auf die Größe der Bit-Eingabe. Die Komplexitätstheorie beschäftigt sich damit, wie der Algorithmus mit der Eingabegröße umgehen wird.

Beispiel: Lösen des verbundenen Graphen von n Kanten im Vergleich zum Graphen von 2 n Kanten. Wenn der Eingang n ist, dann ist die Zeit, die zur Berechnung der Funktion (n) benötigt wird. Wenn die Funktion τ(n) polynomial ist, sprechen wir von einem polynomialen Algorithmus.

Cobham These - das Problem kann in polynomialer Zeit gelöst werden, wenn es für den Algorithmus existiert, der die n-Bit-Eingabe in der Zeit verarbeitet, wobei c eine Konstante ist, die vom Problem abhängt, nicht von seiner Eingabe.

Wir messen aber nicht nur die Eingaben, sondern auch die Ressourcen, sei es ein bestimmter Algorithmus oder ein Problem.

Im Allgemeinen ist die Datengröße n, und nicht für einen bestimmten Eingabewert auf die (Größe n), in der Regel für alle möglichen (unendliche Zahl) → Größe und Komplexität schätzen in der Regel asymptotisch. Die Abhängigkeit von IN-Ressourcen wird gemessen an:

- Verstrichene Zeit (in Schritten)
- Speicher (in Bits / Bytes / Zelle)
- Pakete (in der Regel in Frames)
- Cache (z.B. die Anzahl der Zugriffe)

Complexity Theory Class definiert die Komplexität der Probleme:

- Klasse P
  Das Entscheidungsproblem der U liegt in der Klasse P, wenn es eine Turingmaschine gibt, die die Sprache LU v in Polynomzeit bestimmt.
  Ex ..: Finden des kürzesten Pfades, minimaler Spannweitenbaum

- Klasse NP
  Die Entscheidungsrolle der U liegt in der Klasse NP, wenn und nur wenn es eine nicht-deterministische Turingmaschine gibt, die die Sprache LU v in Polynomzeit entscheidet.
  Ex ..: K-Farbe (Farbpalette kann bis zu den Farben aufgegeben werden ?) Clique des Graphen (existiert Clique im Graphen mindestens von k Knoten?)

- NPC-Klasse - NP-komplettes Problem
  Das Problem ist in der Klasse NP und es ist auch wahr, dass das Polynom die Rolle der einzelnen Klassen NP reduziert. NP - vollständige Aufgaben sind das "schwierigste" aller NP-Probleme.

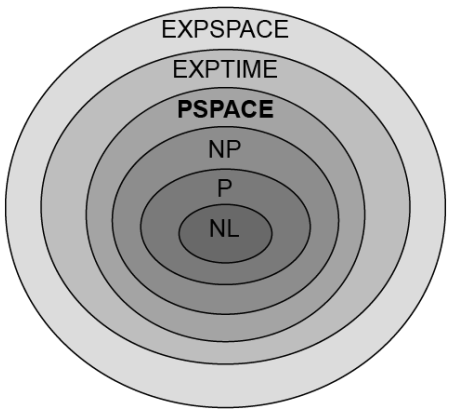
Beispiel: Reisender Verkäufer Problem, Rucksack Problem

Klassen PSPACE und NPSPACE

Die Sprache L ist in der Klasse PSPACE, wenn und nur wenn es eine deterministische Turingmaschine gibt, die mit der Komplexität des Speicherpolynoms arbeitet (d.h. keinen Speicherzellenindex größer als p (n) verwenden) und die Sprache L übernimmt.

Die Sprache L ist in einer Klasse NPSPACE nur dann, wenn es eine nicht-deterministische Turingmaschine gibt, die mit polynomaler Komplexität des Gedächtnisses arbeitet und die Sprache L akzeptiert.

Es hat sich gezeigt, dass NP⊆NPSPACE und des Weiteren entsprechend des Savitch-Satzes gilt PSPACE=NPSPACE.



11. SPRACHEN UND AUTOMATEN

Sprachen und Automaten sind der Grundstein der theoretischen Informatik.

Die Sprache bedeutet jede nicht leere Menge V des Alphabets, ihre Elemente sind Zeichen oder Symbole.

Wir definieren:

- Das Wort über dem Alphabet V ist die letzte Folge von Zeichen aus V, w = a1 a2...an
- Wortlänge w - Länge der Sequenz w, | w | w | = n
- Leeres Wort ε - Reihenfolge der Länge 0
- Eine Menge aller Wörter über dem Alphabet V ^ *
- Eine Menge aller nicht leeren Wörter über dem Alphabet V ^ +
- Die Grammatik G ist vierfach (N, Σ, P, S),
- N ist der endgültige Satz von nicht-terminalen Symbolen (Non-Terminals).
- Σ ist der letzte Satz von Terminalsymbolen, so dass kein Symbol gleichzeitig zu N und Σ gehört.
- P ist der letzte Satz von Ableitungsregeln. Jede Regel ist das Formular (ΣUN) ^ * N (ΣUN) ^ * → (ΣUN) ^ *
- S ist ein Element von N, das als Initialsymbol bezeichnet wird.

Konvention

- Wir bezeichnen Terminals - a, b, c, ...
- Kettenklemmen bezeichnet - u, v, w, ...
- Einzelne Nicht-Terminale - A, B, C ..... X, Y, Z
- Zeichenketten von Nicht-Terminals und Terminals - α, β, γ, ...
- Eine leere Zeichenkette wird durch das Symbol e oder sogar ε gekennzeichnet.

Die Grammatik ist nach der **Chomsky-Hierarchie** in mehrere Typen unterteilt.

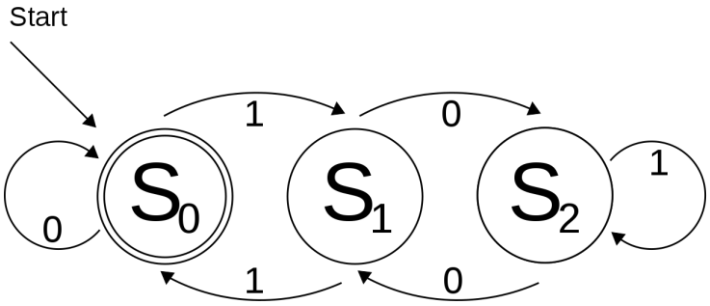
- Typ 0 - Alle formalen Grammatiken (uneingeschränkte Grammatiken)
- Typ 1 - Kontext-Grammatiken
- Typ 2 - Kontextfreie Grammatiken
- Typ 3 - Normale Grammatik

Der endliche Automat ist das theoretische Rechenmodell, das in der Informatik zum Studium formaler Sprachen verwendet wird. Es beschreibt einen sehr einfachen Computer, der sich in einem von mehreren Zuständen befinden kann, zwischen denen er Symbole einschaltet, die aus der Eingabe lesen. Der Satz von Zuständen ist endlich (daher der Name), der endliche Automat hat keinen zusätzlichen Speicher, zusätzlich zum aktuellen Status. Wir unterscheiden deterministische EA - an jedem Punkt in der Tabelle hat nur einen Zielzustand - und nicht deterministische EA - an jedem Punkt der Tabelle ist nicht

ein Zielzustand, sondern der gesamte Satz von Zuständen. Außerdem gibt es in der Übergabetabelle eine leere Eingabespalte, genannt e. Jede nicht-deterministische Maschine kann in deterministisch umgewandelt werden. Der ursprüngliche Satz von Zuständen muss durch seinen potenziellen Satz ersetzt werden. Jeder Zustand der so erzeugten Maschine entspricht einem Satz von Zuständen der ursprünglichen nicht-deterministischen Maschine und es gibt klare Übergänge zwischen ihnen.

Formal ist der endliche Automat definiert als geordnete fünf (S, Σ, σ, s, A) wobei:

- S ist ein endlicher, nicht leerer Satz von Zuständen.
- Σ ist ein endlicher, nicht leerer Satz von Eingabesymbolen, genannt Alphabet.
- σ ist die so genannte Übergangsfunktion (auch eine Übergangs-Tabelle), die die Regeln für den Übergang zwischen Zuständen beschreibt. Kann entweder $S \times \Sigma \rightarrow S$ (deterministischer Automat) oder $S \times \{\Sigma \cup \varepsilon\}$ haben. $\rightarrow P(S)$ (nicht deterministischer Automat), siehe unten.
- s ist der Ausgangszustand, $s \in S$.
- A ist eine Reihe von Empfangszuständen, $A \subseteq S$.



Tätigkeiten an der Maschine

Die Maschine befindet sich zunächst in einem definierten Ausgangszustand. Liest bei jedem Schritt ein Symbol aus dem Eingang und tritt in einen Zustand ein, dem ein Wert zugewiesen wird, der in der Übergabetabelle dem aktuellen Zustand entspricht und das Symbol zurückliest. Es fährt mit dem Lesen des nächsten Symbols des Eingangs, eines weiteren Übergangs durch die Übergangs-Tabelle usw. fort. Je nachdem, ob das Gerät nach dem Lesen des Eingangs in einem Zustand, der zum Satz von Empfängerzuständen gehört, stoppt, akzeptiert der Automat entweder den Eingang oder nicht. Die Menge aller Zeichenketten, die der Automat akzeptiert, bildet eine reguläre Sprache.

12. TURINGMASCHINEN

Church (Church-Touring-)Arbeit: Turingmaschinen (und ihre entsprechenden Systeme) definieren ihre Rechenleistung, die intuitiv als effizient berechenbar angesehen wird. Für jeden Algorithmus gibt es eine gleichwertige Turingmaschine.

Das Church-Argument(Church-Turing) kann nicht formal nachgewiesen werden, wird aber durch eine Reihe von Argumenten gestützt:

Turingmaschinen sind sehr robust - verschiedene Modifikationen verändern ihre Rechenleistung nicht.

Es schlug eine Reihe von verschiedenen Berechnungsmodellen vor, deren Stärke den Turingmaschinen entspricht.

Es gibt keinen bekannten Berechnungsprozess, den wir als effektiv quantifizieren würden und der nicht möglich wäre, mit Hilfe der Turingmaschine zu implementieren.

Die Turingmaschine ist das theoretische Modell eines Computers. Sie besteht aus mehreren Teilen:

- Prozessoreinheit - endlicher Automat
- Programm - Regeln für Übergangsfunktionen
- Rechtsseitiges Endlosband - Wird zur Aufzeichnung von Zwischenergebnissen verwendet.

Es wird für die Modellierung von Algorithmen in der Theorie der Berechenbarkeit verwendet.

Turing's Programmiersprachen und Computer, haben die gleiche Rechenleistung wie Turing's Maschine.

Definition:

$M=(Q,\Gamma,b,\Sigma,s,\delta,F)$	Turingmaschine
Q	Endlicher Satz von internen Zuständen
Γ	Endliches Alphabet von Symbolen auf dem Band
$b\in\Gamma$	Ein leeres Symbol ist nicht Teil des Alphabets der Eingabezeichenkette.
$\Sigma\subseteq\Gamma\setminus b$	Endlicher Satz von Eingangssymbolen

$s \in Q$	Ausgangszustand
$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$	Übergangsfunktion, L nach links nach links, R Kopf nach rechts bewegen
$F \subseteq QQ$	Satz von Endzuständen
$\langle q,s,n \rangle \in Q \times \{yb\hat{}\omega \mid y \in \Gamma\hat{}* \} \times N_0$	Konfiguration
q	Aktueller Status
s	Der kleinste kontinuierliche Teil des Bandes, der die nicht leeren Symbole enthält.
n	Lesekopfposition (Zellzahl)

Wenn Q,Γ, disjunkte Sätze sind, kann eine kompakte Form verwendet werden.

Band: 1234
Zustand der Maschine: q
Position des Lesekopfes: 2
Konfiguration: 1q234
Erstkonfiguration des TM-Eingangs w∈*
s,wb,0, Kompakte Form: sw

Berechnungsmethode für die Turingmaschine

Wenn der aktuelle Status der Endzustand ist, beendet er die Berechnung.

Der Lesekopf liest ein Symbol aus der Zelle, auf der er sich gerade befindet.

Wenn in der Übergangsfunktion für den aktuellen Zustand ein Übergang definiert ist und ein Übergang für das zu lesende Symbol definiert ist, dann (bei mehreren möglichen Übergängen in nicht-deterministischen Maschinen wird einer zufällig ausgewählt):

Ändert es den Status.
An der aktuellen Kopfposition wird das entsprechende Symbol eingegeben.
Der Kopf bewegt sich entsprechend (verschiebt sich nicht).
Es gibt verschiedene Modifikationen von TM.
TM mit der Möglichkeit, die Berechnung ohne Kopfverschiebung durchzuführen.
Übergangsfunktion um N erweitert - N – $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R,N\}$

Die N-Verschiebung (kein verschieben des Lesekopfes) kann auf einem herkömmlichen TS so angeordnet werden, dass der Lesekopf das Eingangssymbol aus der Zelle liest, auf der er sich befindet, den entsprechenden Vorgang durchführt und nach rechts fährt (R). Liest nun das Symbol aus der Zelle, schreibt das gleiche Symbol (d.h. ändert nicht das Symbol, auf dem sich der Kopf befindet) und bewegt sich nach links (L). So haben wir den Lesekopf an die vorherige Position gebracht und kein anderes Symbol wurde geändert.

TM mit zweiseitigem Endlosband
Das Band ist nicht linksbündig begrenzt.
Mögliche Verschiebung des Lesekopfes nach links von jeder Konfiguration möglich
N-Band TM
Liest von und schreibt auf mehrere Bänder gleichzeitig.
Die einzige Änderung betrifft die Übergangsfunktion:δ:Q×Γ^n→Q×(Γ×{L,R,N})^n
Nicht-deterministisches TM (NTM)

Ermöglicht "Mehrfachauswahl".

$$\delta: Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R,N\}}$$

Subroutinen werden verwendet, um die Erstellung komplexer TM zu erleichtern. Das Unterprogramm ist der Satz von Zuständen, der den Anfangs- und Endzustand enthält. Normalerweise löst es ein Teilproblem in TM. In der normalen Programmierung ist es das Äquivalent einer Funktion.

Der universelle TM- U als Eingang akzeptiert den Code eines anderen TM T und das Maschineneingangswort T. Er dekodiert die Übergangsfunktion der Maschine T und simuliert die Berechnung der Maschine. Es kann jede partiell rekursive Funktion berechnen (oder entspricht der universellen partiell rekursiven Funktion), entscheidet jede rekursive Sprache und akzeptiert jede rekursive Sprache.

13. BERECHENBARKEITSTHEORIE

Die grundlegende Theorie der Frage der Berechenbarkeit lautet: Was ist und was ist nicht algorithmisch berechenbar (lösbar)? Für welche Probleme gibt es Algorithmen, die sie lösen?

Das Problem wird definiert durch:

- Name: XY
- Instanz: Was kann eingegeben werden?
- Ergebnis: Output, Input zu Output Beziehung

Das Problem ist die teilweise Anzeigeart**, gültig für die Eingabe gibt eine gültige Ausgabe für ungültige Eingabe Sonderergebnis "Falsche Eingabe".

Probleme haben die folgenden Eigenschaften:

- Algorithmische Lösbarkeit
  Für ein bestimmtes Problem sagen wir, dass es algorithmisch lösbar ist (oder die jeweilige (Teil-)Ansicht ** ist algorithmisch berechenbar), wenn es einen Algorithmus gibt, der als Eingabe in der Lage ist, jede Instanz des Problems zu akzeptieren und seine Berechnung für einen solchen Eintrag endet immer in der Ausgabe des gewünschten Ergebnisses.

- Algorithmische Entscheidbarkeit
  Über das Problem von Ja / Nein sagen wir, dass der Algorithmus entscheidbar sein kann, wenn es einen Algorithmus gibt, der in der Lage ist, jede Instanz des Problems als Eingabe zu akzeptieren und seine Berechnung für jeden solchen Eintrag endet immer und es wird die Antwort Ja / Nein ausgegeben.

  Das Problem ist algorithmisch lösbar, wenn es einen Algorithmus gibt, der für jedes bestimmte aufgegebene Problem in einer bestimmten Zeit ein Ergebnis eindeutig bestimmt. Entscheidungen für Ja / Nein Fragen.

- Partielle Entscheidbarkeit
  Über das Problem Ja / Nein werden wir sagen, dass das teilweise entscheidbar ist, wenn es einen Algorithmus gibt, dessen Berechnung nur für diejenigen Einträge beendet ist, die mit den Instanzen des Problems mit der Antwort Ja übereinstimmen.

  Das Problem ist teilweise entscheidbar, wenn Sie wissen, dass der Algorithmus stoppt und nur dann ein Ergebnis liefert, wenn die Antwort auf die Frage ja ist. Wenn die Antwort Nein lautet, werden Sie es nie erfahren, denn der Algorithmus hört nie auf.

Die Komplexitätstheorie liefert uns interessante Ergebnisse. Einer von ihnen nennt sich Halting Problem. Wir können keinen Algorithmus konstruieren, der die Endgültigkeit des allgemeinen Programms seines Ablaufs überprüfen würde.

Die Komplexitätstheorie liefert uns eine interessante Hypothese. Einer von ihnen ist die Church-Turing-Thesis. Für jeden Algorithmus gibt es eine gleichwertige Turingmaschine.

- Anhaltendes Problem
  Zuordnung: Wenn Sie den Quellcode des Programms und seine Eingabe kennen, entscheiden Sie, ob das Programm stoppt oder ob es für immer ohne Unterbrechung ausgeführt werden soll.

Im Jahr 1936 bewies Alan Turing, dass es keinen allgemeinen Algorithmus gibt, der das Halteproblem für alle Eingaben aller Programme dort lösen würde. Das Halteproblem wird daher als algorithmisch unentscheidbares Problem bezeichnet.

Beweis:

Die Unentschlossenheit des stoppenden Problems kann durch Widerspruch nachgewiesen werden.

Ursprüngliche Annahme: Nehmen wir an, dass das Halteproblem gelöst werden kann. Das bedeutet, dass es ein Programm stoppt (Programm, Eingabe), welches eine universelle Lösung für das Problem des Stoppens ist - wir gehen daher davon aus, dass, wenn dieses Programm und seine Eingabe übergeben wird, das Programm nach einer endlichen Zeit stoppt, die Antwort so zurückgibt, dass, wenn das aufrufende Programm (Eingabe) nach einer endlichen Anzahl von Schritten vorbei ist, dann stoppt (Programm, Eingabe) wenn JA zurückkehrt, andernfalls (wenn in Managing Calls Programm (Eingabe) in einer Schleife gefangen) NEIN zurückgibt.

Anschließend Konstruieren Sie das Programm Paradox (Programm), das den Aufruf stoppt (Softwareprogramm), und wenn der Aufruf JA zurückkehrt, wissen wir dass er in einer Schleife gefangen ist. Und wenn NEIN zurückgegeben wird, endet er sofort.

Wir fragen, was das Ergebnis eines Aufrufs Paradox (Paradoxon) ist.

Nehmen wir an, dass für einen Moment, der stoppt (Paradox, Paradox), JA ausgegeben wird. Per Definition wissen wir, dass Paradox (Paradox) eine Endlosschleife ist. Per Definition stoppt das Programm, aber wenn die Paradox (Paradox) in einer Schleife gefangen ist, dann muss es stoppen und ein NEIN zurückzugeben. Wir sind also zu einem Widerspruch gekommen.

Andererseits nehmen wir an, dass für einen Moment, der stoppt (Paradox, Paradox), ein NEIN zurückgibt. Per Definition, Programm Paradox dann wissen wir, dass Paradox (Paradox) aufhört. Per Definition stoppt das Programm, aber wenn das Paradoxon (Paradoxon) stoppt, muss es stoppen (Paradoxon, Paradoxon) und JA zurückgeben. Wiederum stoßen wir auf einen Widerspruch.

Da wir alle Möglichkeiten ausgeschöpft haben und immer an Wiedersprüche gelang sind, wird das Problem eine falsche Ausgangsprämisse sein. Folglich existiert das Programm Stopp (Programm, Eingabe), wie definiert, nicht.

Church - Turing-These

Die Hypothese besagt, dass jede mögliche Berechnung erfolgreich durchgeführt werden kann, wenn genügend Zeit und Speicher vorhanden ist.

Der Algorithmus muss die folgenden Anforderungen erfüllen:

- Der Algorithmus besteht aus einer endlichen Anzahl von Anweisungen, die durch die Verwendung einer endlichen Anzahl von Symbolen genau definiert werden.
- Der Algorithmus gibt das Ergebnis immer nach einer endlichen Anzahl von Schritten zurück.
- Den Algorithmus kann sogar einen Mensch mit Bleistift und Papier ausführen.
- Die Ausführung des Algorithmus erfordert keine menschliche Intelligenz, außer dem, was notwendig ist, um Anweisungen zu verstehen und auszuführen.

Da jedes Computerprogramm es in die Sprache einer Turingmaschine übersetzen kann und umgekehrt, kann man eine These in jeder gängigen Programmiersprache formuliert sein.

Die Programmiersprache erfordert eines der folgenden Konstrukte s (unter anderem), dass eine Turing-Komplettlösung (d.h. gleichwertige Turing-Maschine)

- Zyklus während -bis.
- Unbegrenzte (zumindest in der Theorie eklige) Rekursionen,
- Bedingter Sprung.

Gängige Programmiersprachen haben in der Regel alle drei Strukturen. Unter den Sprachen, die komplett sind, ist SQL nicht enthalten (d.h. keine Stored Procedures).