

## 1. Ein-/Ausgabegeräte

Datenraten der MMK deutlich unter den „normalen“ Datenraten. (100 - 300.00 KByte /s vs. 0.01 (Tastatur) - 40 (hören) und 20.000 (sehen)(beides nur Input)).

### 1.1. Eingabegeräte

#### 1.1.1 Tastatur

QWERTZ vs. Dvorak (typewriting vs. Ergonomie)

**Row-Scanning** Tasten in einer Matrix angeordnet. z.B.  $6 \times 17$ . Zur Abfrage werden die Spalten seriell auf high gelegt und die horizontale überprüft.

Vorteile:

- weniger Leitungen ( $6 + 17 = 23$ ) statt  $6 * 17 = 102$  (Tastenzahl), wobei nur 6 Stk. auf ihren Pegel geprüft werden müssen.
- Tasten können beliebig belegt werden.

Nachteile:

- Reaktionsgeschwindigkeit. Es müssen nacheinander  $\max(6, 17)$  Leitungen geschaltet werden.
- somit niedrigere mögliche Anschlagrate „typematic-rate“ (typical: 2-30 Hz)
- Verarbeitungsaufwand im Rechner, somit Erhöhung des typematic Delay (Zeit zwischen Tastendruck und Controller-Ausgabe) (typ:  $\delta t = 10ms - 1s$ ).

Dimensionierung:  $\Delta t_{\max} * r * N_E = 1$

**Prelleffekt** Ursache: schneller Pegelwechsel zu Beginn/Ende eines Schaltvorganges.

Lösung: Entprellschaltung durch RS-Flipflop, oder Totzeit (via Controller  $\Rightarrow$  Erhöhung typematic delay)

#### 1.1.2 Maus

**Opto-mechanische Maus** Kontaktkugel, zwei orthogonale, horizontale Walzen an Lochscheiben, Auswertung anhand der Phasen (evtl. Verschiebung  $\Rightarrow$  2 Lichtschranken)

Ortsauflösung:  $r_0 = D \frac{d_{\text{Lochsch.}}}{d_{\text{Sch.achse}}} \left[ \frac{\text{Anz. d Auslöser}}{m} \right]$

**Optische Maus** Bestimmung des optischen Flusses zweier aufeinander folgender Bilder;  $16 \times 16$  Pixel;  $> 1000$  dpi; bis 1500 Hz;

$$V_{\max} = \sqrt{v_{h \max}^2 + v_{v \max}^2}$$

**Trackball** Auf dem Rücken liegende opto-mechanische Maus.

**Spacemouse** 3D Eingabegerät, Puck der sich drehen, neigen, ziehen und schieben, Dehnungsmessstreifen Controller Computer

#### 1.1.3 Joystick

auf Bodenplatte befestigter Stick, analog(poti), digital(taster), isometrisch(DMS) (keine Auslenkung)

#### 1.1.4 Touchscreen - TS

Abstand zwischen Darstellungsebene und Berührungseben  $\Rightarrow$  Parallaxenprobleme, allgemein robust (keine beweglichen Teile)

**optischer TS** Reihe von IR-LED am Displayrand, ggü. Fotosensoren, Gitter unsichtbarer Lichtstrahlen (Opto-Matrix), Lokalisationsgenauigkeit wg. Schattenwurf

**akustischer TS** Piezoelektrische Sender/Empfänger, Signalmuster im MHz Bereich wird vom Sender in gerichtete Ultraschallwellen, x,y Koordinaten aus zeitlicher Lage der Dämpfung, aus Dämpfung kann Andruckkraft ermittelt werden

**Resistiver TS** 2 durchsichtige, gegenüberliegende, leitfähige Schichten; Isolierpunkte; Spannungsteiler; abwechselnd x,y; häufige Spannungswechsel  $\Rightarrow$  elmag. Störfeld;

$$\begin{aligned} x_1 &= \frac{U_{x1} + U_{y2} | x}{U_{x1} - U_{x2}} & y_1 &= \frac{U_{y1} + U_{x2} | y}{U_{y1} - U_{y2}} \\ x_2 &= \frac{U_{x2} + U_{y2} | x}{U_{x2} - U_{x1}} & y_2 &= \frac{U_{y2} + U_{x2} | y}{U_{y2} - U_{y1}} \end{aligned}$$

**Kapazitiver TS** leitend beschichtete Glasplatte; Strom an 4 Ecken; Finger absorbiert Strom; Berührungsposition aus Verhältnissen der Ströme (und R der Glasplatte);

**Piezoelektrischer TS** an 4 Ecken Piezoelement; abfallende Spannungen  $\Rightarrow$  Position; Summe der abfallenden Spannungen  $\Rightarrow$  Druck;

**Grafiktablett** elmag Abtastmechanismus, Gitter von Drähten, Multiplex, Kabel an Griffel / Schwingkreis in Stift (passiv);

#### 1.1.5 Scanner

opt. Abtastung; Orts- und Helligkeitsdiskretisierung; Farbe in 3 Werten; 3\*8Bit  $\Rightarrow$  24 Bit Farbtiefe;

**CCD** Matrix, lichtempfindliche SiZellen; Lichtquant; sequentielle Auslesung (Spg an Nachbarzellen) limitiert Auslesegeschwindigkeit; Blooming-Effekt (überbelichtete Zellen  $\Rightarrow$  überb. Streifen in Ldgtransportrichtung);

**PMT** Photokathodenschicht; Dynoden; steigendes Potential;

**Flachbettscanner** gläserne Auflagefläche; CCD; zeilenweise Abtastung; bis 2k4 dpi; kostengünstig; versch. Vorlagedicken;

**Trommelscanner** bis 10k dpi; Vorlage auf Trommel spannen; kollimierter Lichtstrahl; Spiegeleinheit; Linsen-Spiegelsystem; PMT/Lichtstrahl;

#### 1.1.6 Videokamera

Früher Röhren; CCD; 1/3 Chip (Farbfilter/Prisma); kostengünstig/Auflösung & Farbechtheit; progressiv(Vollbild / Abtastzeitpunkt) oder interlace (Halbild / Ab); Zukunft: IC, schneller Auslesung, Preis, hohe Dynamik;

#### 1.1.7 Mikrophon

Luftschall in elek. Signal; versch. Richtcharakteristiken; Tauchspule/Bändchen/ Kondensator/ Kohle/ Piezo;

## 1.2. Ausgabegeräte

### 1.2.1 Video

**Röhrenbildschirm** CRT; Helligkeit Geschw.Variation; farbe: 3 El.Kanonen treffen auf RGB Phosphore; Lochmaske; hohe Farbsättigung & Farbbereich & Kontrast; Flimmern durch raschen Helligkeitsverlust; Interlacing reduziert Flimmern; kugelförmige Wölbung;

**Flüssigkristall-Zelle** LCD; LC zwischen 2 Glasplatten; Glasplatten aussen elek. leitend; Abstand  $5 \mu m$ ;  $90^\circ$  gedrehte Polarisationsfilter; Glasplatten mit Richtungsstruktur; normally white / black; mode; Lichtquelle: LED / Leuchtstoffröhren, bzw. Reflektiv-LCDs;

**Passives Matrix Display** horizontales + vertikales Drahtgitter zwischen Polaris.filter; Rückstellzeit  $\Rightarrow$  schlieren; native Auflösung; Farbe durch benachbarte Monochromzellen; Farbfilter Durchlassfilter mit endliche Bandbreite; schlechterer Kontrast als CRT;

**aktives TFT-LCD** Thin Film Transistor; TF / Pixel; source zeilen, gate spalten verbunden; verschiedene Graustufen; Polarisation rascher als passive;

**Plasma Display** Plasma = fast vollständig ionisiertes Gas; Drahtgitter zur Plasmazellensteuerung; Bild  $\Rightarrow$  Zellen werden gezündet; UV  $\Rightarrow$  Szintillatoren  $\Rightarrow$  sichtbares (monochromatisches) Licht; hohe Farbbrillanz/Sättigung; versch. Helligkeit durch PWM;

### 1.2.2 Audio

**Lautsprecher** Schwingspule, Spinne, berührungsfrei zwischen Dauermagnet;

## 2. Menschliche Sinnesorgane

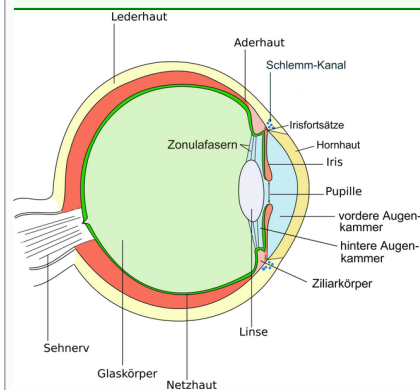
Auge 87% , Ohr 10% , Rest 3%;

### 2.1. Sehen

#### 2.1.1 Das Auge

Hornhaut; Pupille; Iris (hell/dunkel Adaption); Ziliarmuskel; Linse (Fokus); Glaskörper; Retina (optisch aktive Elemente,  $1 - 2 * 10^8$  Rezeptoren) mit Fovea centralis (höchste Zapfendichte) und blindem Fleck; Sehnerv ( $10^6$  Nervenfasern); Aderhaut; Lederhaut;

$$\tan\left(\frac{\beta}{2}\right) = \frac{h}{2l} = \frac{h_z}{x d_{\text{Auge}}} \quad h_z = \sqrt{\frac{l}{D}} \quad D_{\max} \approx 1,4 * 10^{11} m^{-2} \quad D_{\min} \approx 5 * 10^9 m^{-2}$$



**Prinzip des Sehens** ca.  $380-750nm$  ( $4 * 10^{14} Hz - 7,5 * 10^{14} Hz \approx 1$  Oktave;  $\lambda = \frac{c}{f}$ ); 100dB; spektr. Empf. je nach Adaption: Tagsehen / photopische Sehen / Farbempfinden bzw. Nachtsehen / scotopische Sehen; Sakkade  $\Rightarrow$  fovea centralis;

### 2.1.2 Psychooptische und physik. Messgrößen

$l_{cd} \cong$  Lichtstärke eines monochromatischen strahles mit  $f = 5,4 \cdot 10^{14} Hz$  und der Strahlungsstärke von  $\frac{1}{683} \frac{W}{sr}$

Psychooptik		Physik	
Bezeichnung	Einheit	Bezeichnung	Einheit
Lichtstärke $I_v$	$cd$ (Candela)	Strahl.stärke $I$	$\frac{W}{sr}$
Leuchtdichte $L$	$\frac{cd}{m^2}$	Strahl.dichte $L_\Omega$	$\frac{W}{sr m^2}$
Lichtstrom $\Phi_v$	$lm = cd sr$	Strahl.leistung $P$	$W$
Lichtmenge $Q_e$	$lm \cdot s$	Strahl.energie $E$	$J = W s$
Beleucht.stärke $E_v$	$lx = \frac{lm}{m^2}$	Bestrahl.stärke $E$	$\frac{W}{m^2}$
Belichtung $H$	$lx \cdot s$	Energiedichte $w$	$\frac{J}{m^2}$
Lichtausbeute $\mu = \frac{\text{Lichtstrom}}{\text{Strahlungsleistung}} = 1 \frac{lm}{W}$			

### 2.1.3 Farbsehen

Stäbchen sw, hohe Konz ( $1,2 * 10^8$ ), Nachtsehen; S-Zapfen Blau 430nm, M-Zapfen Grün 530nm, L-Zapfen Rot 560nm, 1:10:10, insg.  $7 * 10^6$ ;

### 2.1.4 Gesichtsfeld

volles Farbempfinden nur im Überlappungsbereich der Farbzonen; primäres Gesichtsfeld horiz.  $-15^\circ < \theta < +15^\circ$  und vert.  $-17^\circ < \phi < +14^\circ$ ; 3D:  $-55^\circ < \theta < 55^\circ$

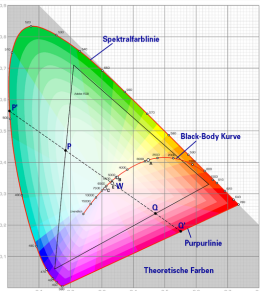
## 2.2. Farbmischung

### Arten der Farbmischung

**Additiv** aktive Primärstrahler; RGB;  
**Subtraktiv** CMY ; Absorption best. Prim.farben; Ausgegangen wird von einer weiß beleuchteten Oberfläche;

**Farbwürfel** Grundfarben, Mischungen, s/w definieren Ecken;  $(R, G, B)^T = (1, 1, 1)^T - (C, M, Y)^T$ ;

## 2.2.1 CIE



**Normfarbtafel nach C.I.E** Ziel: Farbeindruck sämtlicher spektraler Farben durch additive Überlagerung dreier monochromatischer Strahler nachzubilden;  $\lambda_{R,CIE} = 700nm$ ,  $\lambda_{G,CIE} = 546.1nm$ ,  $\lambda_{B,CIE} = 435.8nm$  sog. Normvalenzen; Im Bereich  $350nm < \lambda_R < 540nm$  negativ;  $\Rightarrow$  nachzubildende Farbe mit rot überlagert;  $\Rightarrow$  es ist nicht möglich, alle wahrnehmbaren Farben mit nur drei Primärstrahlern nachzubilden;

**Virtuelle Normvalenzen** Uneigentliche Farbmischung;  $X(r)$ ,  $Y(g)$ ,  $Z(b)$ ; exist nicht real durch add. Farbmischung, können aber jede wahrnehmbare Farbe darstellen;

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.49 & 0.31 & 0.2 \\ 0.177 & 0.813 & 0.01 \\ 0 & 0.01 & 0.99 \end{pmatrix} \begin{pmatrix} R_{CIE} \\ G_{CIE} \\ B_{CIE} \end{pmatrix}$$

Daraus ergibt sich  $z = 1 - (x + y)$ ; Die Farbeindrücke durch elmag Wellen best. F, befinden sich auf Begrenzungsline der Fläche. Im Inneren befinden sich sämtliche Mischfarben, die durch Mischung der x und y Valenzen erzeugen lassen; Weißpunkt im scherpunkt; Luminanznormierte Normkarte:

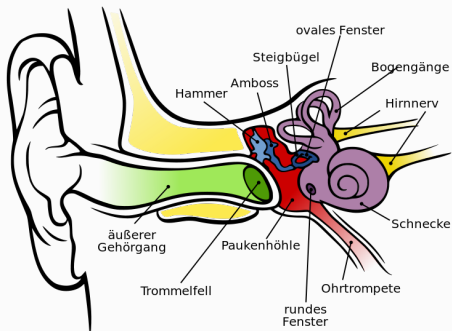
$$x + y + z = 1 \\ \Rightarrow z = 1 - (x + y)$$

$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z} \quad z = \frac{Z}{X+Y+Z}$$

## 2.3. Hören

### 2.3.1 Das Ohr

Äußenohr (Ohrmuschel & Gehörgang); Mittelohr (Trommelfell, Gehörknöchelchen (Hammer, Amboss, Steigbügel) & Eustachische Röhre) - Wandlung von Luftschwingung in mech. Schwingung; Innenohr (Steigbügel über ovale Fenster in mit Flüssigkeit gefüllte Schnecke) Impedanzwandlung von Luft zu Flüssigkeit; Basilarmembran: Haarzellen (25k - 30k Rezeptoren) wandeln Schwingung in el. Nervenimpulse, Frequenz-Ort-Wandlung, Zerlegung in Frequenzanteile  $\Rightarrow$  Hörnerv (30k Nervenfasern)  $\Rightarrow$  Hirn

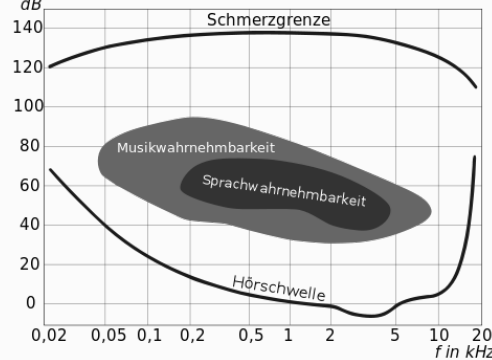


### 2.3.2 Psychoakustik

Empfindlich von etwa 20 Hz - 20 kHz ( $\approx 10$  Oktaven); starke Dämpfung für sehr niedrige und sehr hohe Frequenzen; Resonanzfrequenz des Gehörgangs bei ca.  $3 \dots 3.4 kHz$ ; Lautenheit in [sone] 1 sone  $\triangleq$  Lautheit eines 1kHz Sinustons mit 40 dB Verhältnistönhöhe [mel] 1000 mel  $\triangleq$  1000Hz

Psychoakustik		Physik	
Bezeichnung	Einheit	Bezeichnung	Einheit
Tonheit $Z$	Bark	Frequenz $f$	Hz
Verhältnistönh. $V$	Mel	Schalldruck $p$	$\frac{N}{m^2} = Pa$
		Schallschnelle $v$	$\frac{m}{s}$
		Schallintensität $I$	$\frac{W}{m^2} = \frac{N}{s m}$
Lautstrk.pegel $L_n$	Phon	Schalldruckp. $L$	dB
Lautheit $N$	sone	Schallleist. $P_{ak}$	$W = \frac{N m}{s}$
Bezugsschalldruck $p_0 = 2 \cdot 10^{-5} \frac{N}{m^2} = 20 \mu Pa$			
Bezugsintensität $I_0 = 1.0 \cdot 10^{-12} \frac{W}{m^2}$			

**Hörfäche** Bewertungsfilter mit gleichem Lautstärkeindruck (A, B, C, D - da nichtlinear zur Lautstärke); Lautheit  $Z$  in Sone ist angepasstes Schema;



**Frequenzgruppen** (24) begrenzte Auflösung des Gehörs; jede F.gruppe nimmt gleiche Länge auf Basilarmembran ein (1,3mm - unter 500 Hz = 100Hz, darüber kleine Terz 1,19 der Mittenfrequenz); Bark-Skala; 1.31 Bark = 131 mel = 131 Hz; Blätterrauschen in Ferne L = 10dB, Düsenjäger in 30 m L = 140dB;

**Verdeckungen** Hörschwelle bei Störschall (Maskierung); Spektrale: verbreitet sich mit steigendem Pegel überproportional; Zeitliche: Vorverdeckung; Simultanverdeckung; Nachverdeckung (einige hundert ms); Kompression: Mithörschwelle über Verdeckungen ermitteln; MP3 ab 160 kBit/s;

## 3. Dialogsystem

- fortgeschrittene intuitive Ein-/Ausgabetechniken
- Hohes Maß an Interaktivität durch Benutzerfreundlichkeit und ausgeprägte Dialogfähigkeit
- Intelligentes Systemverhalten, selbstständig logische Schlüsse ziehen;

Teilgebiete der KI: Maschinelles Lernen, Bildverstehende Systeme, Expertensysteme, Robotik, Logik und automatisches Beweisen, Natürlichsprachliche Systeme;

## 3.1. Suchverfahren

Formulierung und Darstellung eines Problems im Zustandsraum; Graphen-Darstellung; Suchbaum;

zyklische Wiederholungen unterbinden (gerichtete Kanten im Baum)

### 3.1.1 Tiefensuche und Breitensuche

1. einelementige Liste mit Wurzelknoten
2. bis Liste leer / Ziel erreicht:
  - prüfe erstes Element auf Zielknoten bzw. max. Suchtiefe
  - wenn ja, fertig
  - wenn nein, entferne dieses Element und füge all seine Nachfolger an gleicher Stelle / am Ende ein.

Vorraussetzung: Elemente der Warteliste werden systematisch erzeugt; Suchtiefe wird geeignet groß festgesetzt / ausgewertete Suchbaum muss gespeichert werden;

### 3.1.2 Heuristische Suche / A-Algorithmus

Verarbeitung zusätzlicher Informationen; Bewertungsmöglichkeit für Erfolgsaussichten eines bestimmten Pfades; Entscheidungen ordnen; Vielversprechende Alternative zuerst, „dem atm billigsten folgen“; Heuristik besteht in Definition einer geeigneten Bewertungs (Kostenfunktion)  $f(n)$ ; z.B.

$$f(n) = g(n) + h(n)$$

Bewertungsfunktion = Bisherige Kosten + Schätzfunktion (hier: falsche Plättchen) Falls  $h(n) \equiv 0$  gewählt wird identisch zur Breitensuche

### 3.1.3 A\*-Algorithmus

Schätzfunktion  $h(n)$  monoton, d.h. Kosten werden nicht überschätzt; terminiert wenn Zielknoten gefunden und keine geringere Kostenschätzung existiert; A\* somit optimaler Pfad; wird die optimale Kostenfkt  $h^1(n)$  verwendet, so wird kürzester Pfad auf Anhieb gefunden (sprich: informierte Suche); Liste mit allen Elementen erstellen + sortieren; dem insg. billigsten folgen; nix verwerfen;

## 4. Logik und Theorembeweisen

Wissen algorithmisch darstellen; Fakten ableiten; Behauptungen bestätigen / widerlegen;

### 4.1. Logik

#### 4.1.1 Aussagenlogik

atomare Aussagen; wahr oder falsch; UND, ODER, NICHT; Implikation  $\Rightarrow$ ;

#### 4.1.2 Prädikatenlogik

Analyse und Bewertung von Beziehungen und logischen Verknüpfungen; 1. Ordnung  $\Rightarrow$  nur Veränderung von Objekten, nicht Prädikaten; Prädikate und FUNKtionen, Konstanten, Variablen, Funktionen, Negation, Disjunktion, Konjunktion, Existenz-Quantor, All-Quantor, Implikation, Äquivalenz „In jeder Stadt gibt es einen Bürgermeister“  $(\forall x) \{ \text{Stadt}(x) \Rightarrow (\exists y) [\text{Mensch}(y) \cdot \text{Bgm}(x, y)] \}$  Regeln und Zusammenhänge aufstellen;  $\Rightarrow$  Regelwerk (Axiome); Frage (Theorem); Beweis durch Wahrheitstabelle oder Umformen der Regeln und Schlussfolgern (Resolution, Unifikation - effektiver);

### Umformregeln:

1. Doppelte Negation  $\neg \neg A \equiv A$
2. Idempotenz  $A + A \equiv A$  und  $A \cdot A \equiv A$
3. Kommutativität  $A + B \equiv B + A$
4. Assoziativität  $A + (B + C) \equiv (A + B) + C$
5. Distributivität  $A + (B \cdot C) \equiv (A + B) \cdot (A + C)$
6. De Morgan  $\neg(A \cdot B) \equiv \neg A + \neg B$
7. Kontraposition  $A \Rightarrow B \equiv \neg B \Rightarrow \neg A$
8.  $A \Rightarrow B \equiv \neg A + B$
9.  $A \Leftrightarrow B \equiv (A \Rightarrow B) \cdot (B \Rightarrow A) \equiv (A + B) + (\neg A \cdot \neg B)$
10.  $\neg(\forall x)A(x) \equiv (\exists x)(\neg A(x))$
11.  $\neg(\exists x)A(x) \equiv (\forall x)(\neg A(x))$
12.  $(\forall x)(A(x) \cdot B(x)) \equiv (\forall x)A(x) \cdot (\forall x)B(x)$
13.  $(\exists x)(A(x) + B(x)) \equiv (\exists x)A(x) + (\exists x)B(x)$

### 4.1.3 Standardformen

**Konjunktive Normalform (KNF):**  $(A_1 + A_2 + \dots) \cdot (B_1 + B_2 + \dots) \cdot \dots$

**Disjunktive Normalform:**  $(A_1 \cdot A_2 \cdot \dots) + (B_1 \cdot B_2 \cdot \dots) + \dots$

#### Regeln zur Umformung in Normalform:

1. Eliminierung aller Äquivalenzen (# 9)
2. Eliminierung aller Implikationen (# 8)
3. Einziehung der Negation nach innen (#6, #10, #11)
4. Einführung neuer Variablen für jeden Quantifizierer
5. Eliminierung aller Existenz Quantoren
6. Ausklammern der All-Quantoren und Entfallen dieser
7. Anwendung des Distributivgesetzes zur Transformation in Konjunktive Normalform (#5)
8. Eliminierung der UND-Verknüpfungen durch Auflistung der Klauseln
9. Einführung getrennter Variablen für jede Klausel

## 4.2. Theorembeweis

### 4.2.1 Resolutionsverfahren

Gegeben sind zwei Formel der Form:

$$A_1 + A_2 + \dots + A_n + P$$

$$B_1 + B_2 + \dots + B_n + \neg P$$

wird zu

$$A_1 + \dots + A_n + B_1 + \dots + B_n \equiv R$$

Anwendung beim Theorembeweis:

Geg.: Set von  $n$  existierenden und bewiesenen Axiomen  $S = \{S_1 \dots S_n\}$ ; Es gilt T zu beweisenn

Vorgehen: Erweiterung von  $S$  zu  $S^* = \{S_1 \dots S_n, \neg T\}$  Und Resoluntieren bis leere Klausel erzeugt wird.

**Erklärung: Statt Beweis wird Unerfüllbarkeit seines Gegenteils gezeigt.**

#### Tautologie beweisen:

1. Wahrheit auf KNF bringen
2. Gegenteil auf KNF bringen
3. Zeige, dass Gegenteil  $\{ \}$  ist.

## 5. Wissensrepräsentation

effizient speichern; strukturiert darstellen; Menge von Fakten, Regeln, Prozeduren, Modellen, Daten, Heuristiken; interpretierbar mit Hilfe von Repräsentationsmechanismen;

### 5.0.2 Prädikatenlogik

Aufteilung in Fakten und Regeln; Standardisiert durch KNF; Resolution als Inferenzmechanismus; Formulierung aufwändig und unnatürlich; zwingend Umformung in KNF;

**5.1. Produktionsregeln**  
keine Umformung in KNF; Wenn-Dann bleibt erhalten; Vorwärts-Rückwärtsverkettung als Inferenzmechanismus; Darstellung im UND/ODER-Graphen; Fakten als Blatt, Regeln als Verzweigung;

Vorwärtsverkettung
<div>1. Gültige Fakten einkreisen</div> <div>2. Suchen nach Regeln, in denen diese Fakten im Bedingungsteil der Regeln vorkommen</div> <div>3. Überprüfen ob Aktionsteil der Regeln eingeleitet werden kann</div> <div>4. Back to #2</div> <div>5. Wenn keine neuen Regeln mehr feuern, überprüfen ob ein Ziel erfüllt wurde</div>
Rückwärtsverkettung
<div>1. Vorgabe eines möglichen Ziels</div> <div>2. Untersuchen der Bedingungen die zum Erreichen dieses Ziels erfüllt sein müssen</div> <div>3. Formulierung dieser Bedingungen als neue Teilziele, back to # 2</div> <div>4. Falls Ziel wg. Bedingungen nicht erreicht werden kann, back to #1 mit anderem Ziel</div> <div>5. Wurden für ein Ziel alle Bedingungen erfüllt ⇒ Finish</div>

**5.2. Semantische Netze**  
Graphische Modelle zur Darstellung von Wissen über Beziehungen zw. Objekten; entsprechen etwa Fakten der Prädikatenlogik; Knoten = Objekte; Kanten = Prädikate; Verwendung bei natürlichssprachigen Systemen; keine 2 Knoten gleicher Beschriftung; Richtung der Kanten von Bedeutung;

**5.3. Rahmen**  
Darstellung der Zerlegung von Objekten oder Situationen in ihre Bestandteile; Ähnlichkeit zu semantischen Netzen, wesentlich mächtiger und flexibler; FrameName - zentraler Knoten, Slots - Kanten, Filler - Knoten; 1. Suchverfahren zur Ermittlung von Beziehungen; 2. „Rahmen-Abgleich“; Fakten als Fragezeichen markiert; mit aktuellen Daten auffüllen;

6. Grammatiken

natürlichssprachige Systeme; Modellierung von Dialogen;

**6.1. Kontextfreie Grammatiken**  
CFG;  $\mathcal{G} = \{V, T, P, S\}$  mit Variable (Großbuchstaben), Terminale (Kleinbuchstaben), Produktionsregel ( $A \rightarrow \alpha$  mit  $A \in \{V\}$  und  $\alpha \in \{V \cup T\}$ ), Startsymbol;

**6.2. Chomsky-NormalForm**  
CNF; Enthält nur Produktionsregeln, bei denen auf der rechten Seite nur zwei Variablen oder nur ein terminaler Ausdruck steht:

$A \rightarrow BC$  oder  $A \rightarrow a$

**6.3. Backus-Naur-Form (BNF)**  
formale exakte Definition von Programmiersprachen; Nichtterminalsymbole werden syntaktische Variablen genannt und durch <, > gekennzeichnet; Darst. von Wdh. durch Rekursion;

**6.4. EBNF**  
Erweiterte BNF; Optionen [...]; abgezählte Wdh. 4\*;

**6.5. Parsing**  
Satzgenerierung: Produktionsregeln solange anwenden, bis alle Variablen V durch terminale Symbole T ersetzt sind; Parse-Tree; Ambiguitäten;

**6.6. Anwendung von Grammatiken in KI**  
Sprache; Mustererkennung;

**6.7. Beispiele**  
Palindrom-String:  
 $S \rightarrow aSa|bSb|a * |b*$   
Doppelte Anzahl a wie b:  
 $S \rightarrow A|SA|AS|aSC|CSa|aSD|DSa|bSB|BSb$   
 $A \rightarrow Bb|Ca|Da$   
 $B \rightarrow aa \quad C \rightarrow ab \quad D \rightarrow ba$

Grammatik-Grammatik:  
S (Satz), NP (Nominalphrase), VP (Verbalphrase), PP (Präpositionalphrase), DET (Determinator, Artikel), ADJ (Adjektiv), AUX (Hilfswort), V (Verb), PRE (Präposition) und N (Nomen)  
 $S \rightarrow NP \text{ VP} | VP \text{ NP}$   
 $NP \rightarrow DET \text{ N} | ADJ \text{ N} | DET \text{ NP} | NP \text{ PP}$   
 $VP \rightarrow V \text{ NP} | AUX \text{ V} | V \text{ PP} | V \text{ NP} | VP \text{ PP} | AUX \text{ VP}$   
 $PP \rightarrow PRE \text{ NP}$   
 $DET \rightarrow \text{„der“}, \text{„die“}, \text{„das“}, \dots$   
 $ADJ \rightarrow \text{„klein“}, \text{„groß“}, \dots$   
 $AUX \rightarrow \text{„wird“}, \dots$   
 $V \rightarrow \text{„streicheln“}, \dots$   
 $PRE \rightarrow \text{„in“}, \text{„mit“}, \dots$   
 $N \rightarrow \text{„Junge“}, \text{„Hund“}, \text{„Hand“}, \dots$

7. Automatentheorie

Verarbeitung von Symbolfolgen; Modellierung von Dialogen;

7.1. Automatentypen

**7.1.1 Zustandsautomat**  
Graphenform; bestimmte Anzahl von Knoten (Zustände) und Verbindungen (Transitionen);  
 $Z = (S, X, T, s_0, \mathcal{F})$   
Set mit endlicher Anzahl Zustände, x zulässiges Alphabet für die zu verarbeitende Symbolfolge X, T Transitionsfunktionen,  $s_0$  Anfangszustand,  $\mathcal{F}$  ein Set von festgelegten Endzuständen; deterministisch / nicht-d.;

**7.1.2 Kellerautomaten**  
komplexere Grammatiken; Erweiterung mit Stack (LIFO); Transition abhängig von Stack und Eingang; Stack leer ⇒ Folge akzeptiert;  
 $Z = (S, X, Y, T, s_0, y_0, \mathcal{F})$   
Y - zulässiges Alphabet fürn Stack,  $y_0$  Start für Stack,  $\mathcal{F}$  leer wenn Endzustand über leeren Stack definiert ist;

8. Dialoggestaltung

Ein-/Ausgabe; Fehlerbehandlung; Fehlertoleranz; Kenntnis der Aufgabe; Benutzergruppen;

**8.1. Expertensysteme**  
komplexes, wissensbasiertes Softwarepaket; Wissensbasis statt Datenbasis; Komponenten zur Pflege und Erweiterung dieser Basis; Schließeregeln können neues Wissen produzieren;

**8.2. Wissen**  
informelles, technisches (Algorithmen, Formeln, fixe Formeln, variable Daten), formales (wenn-dann, variable formeln + daten);

**8.3. Einsatzgebiete**  
komplexe Aufgabenstellungen; Diagnoseaufgaben; Konfigurationsaufgaben; Beratungsaufgaben;

**8.4. Aufbau**  
Wissensbasis: Fakten, Regeln, Prozeduren; wichtigste Komponente; Inferenzkomponente: Verarbeitung; Such- Verkettungsmechanismus; Erklärungskomponente: Lösungsweg; graphisch; Debugging; Dialogkomponente: Interface; Wissenserwerbkomponente: effiziente Entlastung des Programmierers; Experten, Entwickler, Anwender;

**8.5. Dialogformen**  
Frage-Antwort; Menüauswahl; Formular; Kommandosprachen; Natürlichsprachlich; Direkte Manipulation; Multimediadialog;

9. Sprachkommunikation

eine der natürlichsten Kommunikationsformen; größtes Potential; bedeutendste & komplexeste Teil: Spracherkennung;

**9.1. Klassifizierung**  
Zuordnung zu Bedeutungseinheiten; Merkmalsextraktion; Merkmalsvektor; Merkmalsraum; Klassen; Training;

**9.2. Abstandsklassifikatoren**  
Distanz eines Mustervektors zu Klasse;

$$m_k = \frac{1}{M_k} \sum_{i=1}^{M_k} r_{k,i}$$
$$d_k(x, m_k) = (x - m_k)^T * W_k * (x - m_k)$$

Trennfunktion:

$$d_1(x, m_1) - d_2(x, m_2) = 0$$

Gewichtsmatrix  $W_k$  entscheidend;  $m_k$  wird im Training ermittelt; x gehört zur Klass mit min. Abstand; Quadratischer Abstand:  $W_k$  ist Einheitsmatrix; Trennfunktion ist eine Gerade; Mahalanobis Abstand: **Inverse** der Kovarianzmatrix; Abhängig von Klasse; Bestandteil des Trainings; Trennfunktion ist Kegelschnitt (Gerad, Ellipse, Parabel, Hyperbel)

$$W_{K,k} = \frac{1}{M_k} \sum_{i=1}^{M_k} r_{k,i} \cdot r_{k,i}^T - m_k \cdot m_k^T$$
$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

10. HMM und Algorithmen

**10.0.1 Markov-Modelle**  
Abbildung stochastischer Prozesse, deren aktueller Zustand nur vom vorausgegangenen Zustand abhängt; Matrixdarstellung  
 $A = p \{q_{t+1} = s_j | q_t = s_i\}$   
Startzustand  $q_1$ ; Vektor  $e = (p(q_1 = s_1), \dots, p(q_1 = s_N))^T$  der Einsprungswahrscheinlichkeit

**10.1. HMM**  
Hidden-Markov-Modelle; statistischer Klassifikator; liefert p dass eine Beobachtung einer best. Klasse zugeordnet werden kann; klassifizieren ganze Sequenzen (dynamische Folgen); „Finde diejenige Klasse, die die Beobachtung  $o = (o_1, o_2, \dots, o_t)$  am besten nachbilden kann.“;

**10.1.1 HMM**  
stochastische Version eines endlichen Zustandsautomaten; Zustandsübergänge und Symbolemissionen nicht deterministisch; Beobachtungswahrscheinlichkeitsmatrix;  $v = (V_1, \dots, v_M)$  Menge der möglichen Beobachtungen;

$$B = \begin{bmatrix} p(v_1 | s_1) & \dots & p(v_1 | s_N) \\ \vdots & \ddots & \vdots \\ p(v_M | s_1) & \dots & p(v_M | s_N) \end{bmatrix}$$
$$\lambda = (e, A, B)$$
$$p(o | \lambda)$$

Von Beobachtungsfolge o kann i.A. nicht auf durchlaufene Zustandsfolge q geschlossen werden (hidden)

**HMM - Eigenschaften**  
**Ergodisches HMM** Es kann aus jedem Zustand in jeder andere Zustand erreicht werden; A ist voll besetzt  
**Links-Rechts-HMM** keine Rücksprünge; kausal; A hat rechte obere Dreiecksform; Graphisch nach rechts aufsteigend ⇒ Name

**10.1.2 Klassifizierung mit HMM**  
Pro Klasse ein HMM; das HMM welches die größte Produktionswahrscheinlichkeit  $p(o | \lambda_k)$  liefert repräsentiert die gesuchte Klasse  $k_x$ ;

**10.1.3 Training von HMM**  
Kompensation von Störungen; Bed.: geeignete Parameter  $\lambda_k$ ; Training mit iterativen Verfahren; ⇒ Baum-Welch-Algorithmus

**10.1.4 Trellis**  
Zeitabfolge in Diagramm; Berechnung sehr rechenintensiv ( $OPS \ 2T + N^T$ ); Weg q;

$$p(o | \lambda_k) = \sum_{q \in Q} e_{q1} b_{q1}(o_1) \prod_{t=2}^T a_{q_{t-1} q_t} b_{q_t}(o_t)$$

**10.2. HMM in der Spracherkennung**  
Cepstrum; Merkmalsextrahierung; 12D Merkmalsvektor;

**10.2.1 Modelle**  
Einzelworterkenner vs. fließende Sprache; Phoneme, kleinste bedeutungsunterscheidenden Lauteinheiten; HMM pro Phonem; Pausen;

### 10.2.2 Training

Zusammenfassung der Phonem HMM zu einem HMM;

### 10.2.3 Erkennung

Wörterbücher, Grammatiken, Wahrscheinlichkeiten bestimmter Phonemkombinationen, Sprachmodelle für Wortkombinationen;

## 10.3. HMM-Algorithmen

### 10.3.1 Vorwärts-Algorithmus

Vorwärts-Wahrscheinlichkeit:

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = s_i | \lambda_k)$$

d.h. die Wahrscheinlichkeit, dass die Teilbeobachtung  $o_i$  emittiert werden und das sich das HMM zu t im Zustand  $s_i$  befindet;

#### Vorwärts-Algorithmus (Rekursiv)

1. Initialisierung:  
 $\alpha_1(i) = e_i b_i(o_1), \quad 1 \leq i \leq N$
2. Induktion:  
$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1})$$
 $1 \leq t \leq T-1; \quad 1 \leq j \leq N;$
3. Terminierung  
$$P(o | \lambda_k) = \sum_{i=1}^N \alpha_T(i)$$

Benötigte OPS :  $T * N^2$ ;

### 10.3.3 Viterbi-Algo

meist reicht Kenntnis des wahrscheinlichsten Pfades;

#### Viterbi-Algorithmus

1. Initialisierung:  
 $\delta_1(i) = e_i b_i(o_1) \quad 1 \leq i \leq N$   
 $\psi_1(i) = 0$
2. Induktion:  
$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t)$$
$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$$
 $2 \leq t \leq T; \quad 1 \leq j \leq N$
3. Terminierung:  
$$P^* = \max_{1 \leq i \leq n} [\delta_t(i)]$$
$$q_T^* = \max_{1 \leq i \leq n} [\delta_t(i)]$$
4. Ermittlung der wahrsch. Zustandsfolge:  
 $q_t^* = \psi_{t+1}(q_{t+1}^*)$   
 $t = T-1, T-2, \dots, 1$

### 10.3.2 Baum-Welch-Algorithmus

Rückwärtswahrscheinlichkeit:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | q_t = s_i, \lambda_k);$$

d.h. Wahrscheinlichkeit, die restlichen Teilbeob. zu emittieren;

#### Baum-Welch-Algorithmus (Rekursiv)

1. Initialisierung  
 $\beta_T(i) = 1 \quad 1 \leq i \leq N$
2. Induktion  
$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$
 $t = T-1, T-2, \dots, 1 \quad 1 \leq i \leq N$

Wahrscheinlichkeit, dass sich das HMM zu t im Zustand  $s_i$  befindet und o emittiert wird; Summe drüber  $\Rightarrow$  „alle Aufenthalte im Zustand  $s_i$ “

$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)}$$

Wahrscheinlichkeit, dass sich das HMM zu t in  $s_i$  und zu t+1 in  $s_j$  befindet; Summe drüber  $\Rightarrow$  „aller Übergänge von  $s_i$  zu  $s_j$ “

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)}$$

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j)$$