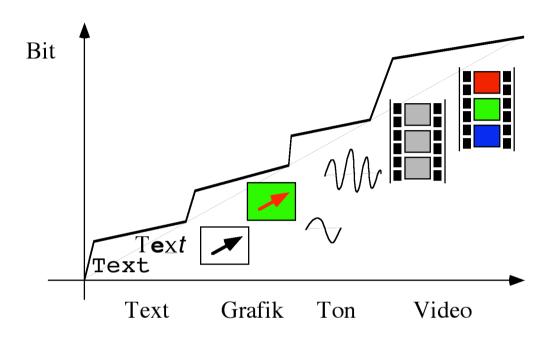
6. Medien

6.1 Medien und Wahrnehmung



• Nutzlast (Bit/bit)

- Information wird in Bit gemessen, bit = Anzahl $\{0,1\}$

```
ASCII-Text 10 byte

Bitmap 1000 Punkte * 1 byte

Telefon 8.000 byte

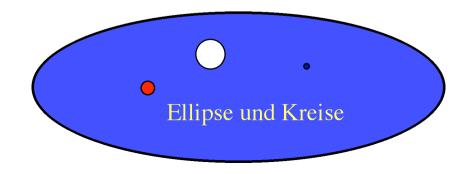
Audio-CD 44.100 Samples * 2 byte * 2

Video 25 Bilder * 256 Spalten * 192 Zeilen * 3 byte/Punkt = 3.686.400 byte

TV 25 Bilder * 704 Spalten * 625 Zeilen * 3 byte/Punkt = 33.000.000 byte
```

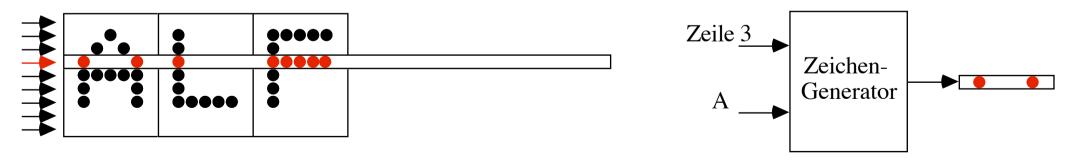
- Aufnahmevermögen und Bitrate
- Dimensionen, räumliche Effekte
 - Menschen haben räumliches Empfinden (Sehen, Hören, Gleichgew.)
 - Raum und Zeit
 - Dimensionen werden vielfältig ausgewertet
- Diskrete und kontinuierliche Medien
 - Klassifikation entsprechend Auflösungsvermögen der Wahrnehmung
 - Im Raum Punkte oder Verläufe: Pixelmaps oder Photographien
 - In der Zeit Stilleben oder Bewegung Grafik oder Animation Bilder oder Video
 - Audiophysikalisch immerPsychisch auch diskret:Sprache oder Musik
- Abschattungseffekte
 - in einem Medium
 - zwischen Medien

kontinuierlich Spracherkennung



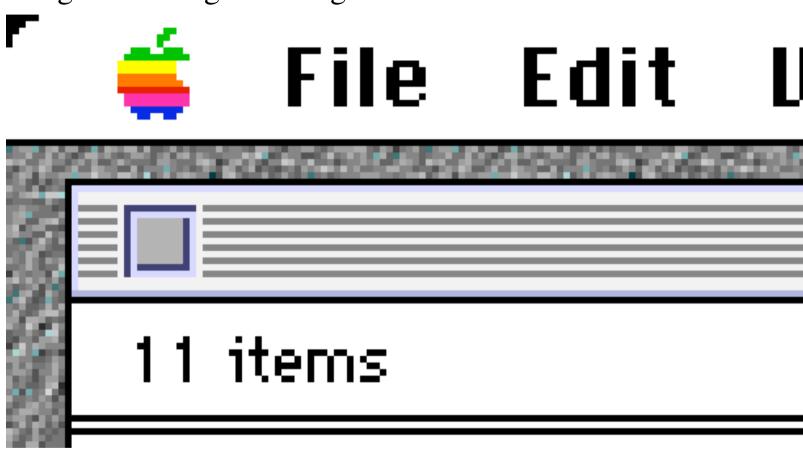
6.2 Computergrafik

- Darstellung visueller Objekte
 - Buchstaben und Zahlen,
 - geometrische Objekte (Gerade, Kreis, Rechteck, ...)
 - Attribute (Farbe, Muster, Font, ...).
- Bildspeicher
 - Hauptspeicher oder im Adapter,
 - eventuell mehrere Ebenen (Farbe, Graustufen, räumliche Position).
- Buchstabenbildschirme
 - nur Buchstaben darstellbar
 - oft als Rasterbildschirm, aber Punkte nur in Gruppen ansprechbar
 - Zeichengenerator: ROM zur Abbildung der Buchstaben auf Raster



• Vektorgrafik ...

- Rasterbildschirm
 - jeder Punkt einzeln ansprechbar
 - uneingeschränkt grafikfähig



- Punkteanzahl typisch 1024*768 bis 2560*1600
- 72, 80 bis 100 Punkte/Zoll (dpi)
- 2560*1600*24 bit für 30" Farbmonitor -> 12.188.000 Byte
- Bildänderungsrate (Framerate, >25 Hz) und Bildwiederholrate (>70Hz)

6.2.1 Text

- Zeichensatz
 - ASCII: <u>A</u>merican <u>S</u>tandard <u>C</u>ode for <u>I</u>nformation <u>I</u>nterchange
 - 0...31 Druckersteuerzeichen
 - 32 .. 127 druckbare Zeichen
 - 128 .. 255 nichtstandardisierte Erweiterungen
 - EBCDIC: Extended Binary Coded Decimal Interchange Code
 - ISO 8859-X

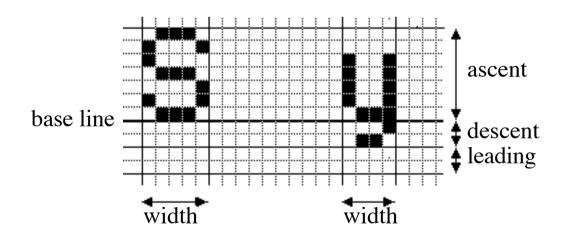
Erweiterung von ASCII um länderspezifische Zeichen

- 1, 2, 3, 4 und 9 für lateinische Zeichensätze
- 5 kyrillisch, 6 arabisch, 7 griechisch und 8 hebräisch
- Unicode

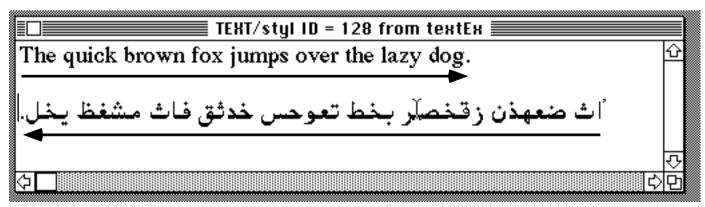
Codes für alle Schriftzeichen der Welt

- 16 Bit/Zeichen
- 28.000 Codes für Ideographen (China, Korea, Japan)
- mehr Zeichen -> mehr Information/Zeichen: ae -> ä, ss -> β
- kompaktes Medium

- Schriftattribute
 - fett, kursiv, Umriss, schattiert, ...
 - Zeichengröße und -breite
 - Kerning und Ligaturen: fl statt fl
- Fontmetrik
 - beschreibt Laufeigenschaften des Textes
 - monospace vs. proportional

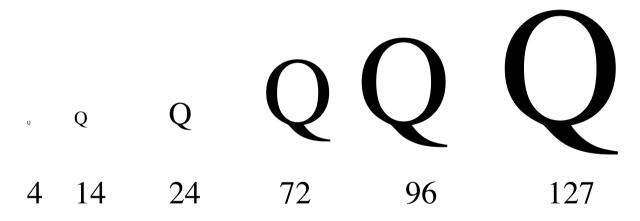


- Fontomania
 - tausende verschiedene Zeichensätze
 - Font-Beschreibungsalgorithmen siehe Kapitel 3
 - Times-Roman
 - Helvetica
 - Palatino
 - Σψμβολ
 - ***□* **■***■*** (Zapf Dingbats)
- nicht-lateinische Schriften
 - andere Fonts
 - Hebräisch, Arabisch, Chinesisch, ...
 - Schreibrichtung rechts -> links, vertikal



6.2.1.1 Zeichendarstellung

- Bitmap-Fonts
 - werden entworfen, gezeichnet, gespeichert und fertig verteilt ...
 - in verschiedenen Größen (z.B. 6 Punkte bis 127 Punkte)



• und Formen

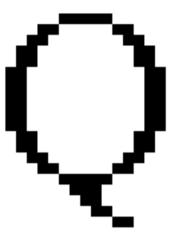
Chicago, Monaco, New York, Geneva

• Zeichensätze als Raster und Outline im System

000								Font – Book Antiqua [/Users/kf/Library/Fonts/Book Antiqua]																	
0000		0002							0009	000A			000D												
						0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F	0030	0031	0032	0033
							!	"	#	\$	%	&	'	()	*	+	,	-		/	0	1	2	3
0034	0035	0036	0037	0038	0039	003A	003B	003C	003D	003E	003F	0040	0041	0042	0043	0044	0045	0046	0047	0048	0049	004A	004B	004C	004D
4	5	6	7	8	9	:	;	<	=	>	?	@	Α	В	C	D	Е	F	G	Н	I	J	K	L	M
004E	004F	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	005A	005B	0050	005D	005E	005F	0060	0061	0062	0063	0064	0065	0066	0067
N	О	Р	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_	`	a	b	С	d	e	f	g
0068	0069	006A	006B	0060	006D	006E	006F	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079	007A	007B	007C	007D	007E		0004	0005
h	i	j	k	1	m	n	О	p	q	r	s	t	u	V	W	X	y	Z	{		}	~		Ä	Å
0007	0009	00D1	00D6	OODC	00E1	00E0	00E2	00E4	00E3	00E5	00E7	00E9	00E8	OOEA	OOEB	OOED	OOEC	OOEE	OOEF	00F1	00F3	00F2	00F4	00F6	00F5
Ç	É	Ñ	Ö	Ü	á	à	â	ä	ã	å	Ç	é	è	ê	ë	í	ì	î	ï	ñ	ó	ò	ô	ö	õ
OOF A	00F9	OOFB	OOFC	2020	00B0	00A2	00 A3	00A7	2022	00B6	OODF	00 AE	00A9	2122	00B4	8A00	2260	0006	0008	221E	00B1	2264	2265	00A5	00B5
ú	ù	û	ü	†	0	¢	£	§	•	\P	ß	®	©	TM		••	≠	Æ	Ø	∞	±	≤	≥	¥	μ
2202	2211	220F	0300	222B	OOAA	OOBA	03A9	00E6	00F8	00BF	00A1	00AC	221 A	0192	2248	2206	00AB	00BB	2026	00A0	0000	0003	00D5	0152	0153
∂	Σ	П	Π	ſ	a	0	Ω	æ	Ø	ż	i	_		f	≈	Δ	**	>>			À	Ã	Õ	Œ	œ
2013	2014	201C	201D	2018	2019	00F7	250A	OOFF	0178	2044	20AC	2039	203A	FB01	FB02	2021	00B7	201 A	201E	2030	0002	00CA	0001	00CB	0008
-	_	"	"	1	,	÷	\Diamond	ÿ	Ÿ	/	€	<	>	fi	fl	‡		,	"	‰	Â	Ê	Á	Ë	È
OOCD	OOCE	OOCF	0000	0003	00D4	F8FF	00D2	OODA	OODB	00D9	0131	0206	02DC	00 AF	02D8	02D9	02DA	00B8	02DD	02DB	0207				
Í	Î	Ϊ	Ì	Ó	Ô		Ò	Ú	Û	Ù	1	^	~	_	J	•	0	3	"	ι	~				π
	00A4	00A6	00 AD	00B2	00B3	00B9	OOBC	OOBD	OOBE	0000	00D7	OODD	OODE	00F0	OOFD	OOFE	0160	0161	0209	2126	2219	F001	F002		
4	¤	-	-	2	3	1	1/4	1/2	$\frac{3}{4}$	Đ	×	Ý	Þ	ð	ý	þ	Š	š	_	Ω		fi	fl		

- Werden bei Bedarf in den Speicher geladen.
- Größe 24 Punkt (Vergrößerung *8)

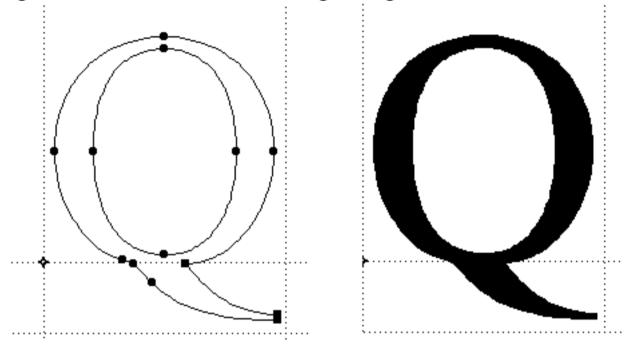
Q



- Auflösungsabhängig, schlecht skalierbar
- Bitmap-Fonts werden bei zunehmender Zeichengröße Speicherfresser
- Bold, Italic, ... müssen separat gespeichert werden

6.2.1.2 Kurven zur Beschreibung von Fonts

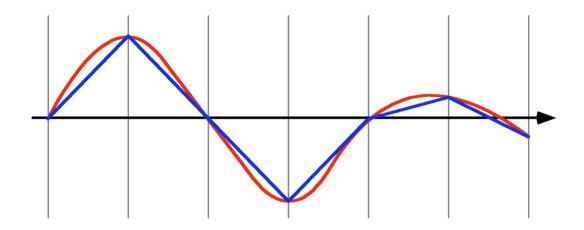
- Die Umrisse der Zeichen werden als Kurvenzug angegeben
- Zur Darstellung wird dieser Kurvenzug ausgefüllt



- unabhängig vom Koordinatensystem
- affine Invarianz
- möglichst einfach berechenbar
- -> Stützpunkte und Interpolation

- Ahnlich Interpolation und Approximation mit Splines
 - stückweise linear: $f_i(x) = a_i x + b_i$

An den Stützpunkten stetig: $f_i(x) = f_{i+1}(x)$



- stückweise kubisch: $f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$

An den Stützpunkten:

- a) stetig: $f_i(x_k) = s_k$, $f_i(x_{k+1}) = s_{k+1} \Rightarrow 2n$ Gleichungen
- b) 'glatt': $f'_i(x) = f'_{i+1}(x) \Rightarrow 2(n-1)$ Gleichungen
- => Gleichungssystem 4n Unbekannte, 2n + 2n 2 Gleichungen je nach Randbedingungen versch. Approximationseigenschaften

• Bézier-Kurven

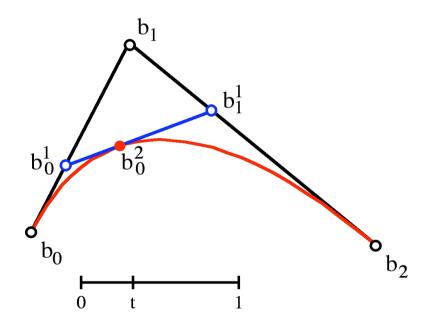
- Beispiel 2. Ordnung

$$b_0^1(t) = (1 - t) b_0 + t b_1$$

$$b_1^1(t) = (1 - t) b_1 + t b_2$$

$$b_1^1(t) = (1-t)b_1 + tb_2$$

 $b_0^2(t) = (1-t)b_0^1(t) + tb_1^1(t)$



- Algorithmus von de Casteljau

gegeben
$$b_0, b_1, \ldots, b_n$$

$$b_i^r(t) = (1-t)b_i^{r-1}(t) + tb_{i+1}^{r-1}(t)$$

$$r = 1, ..., n; i = 0, ..., n-r$$

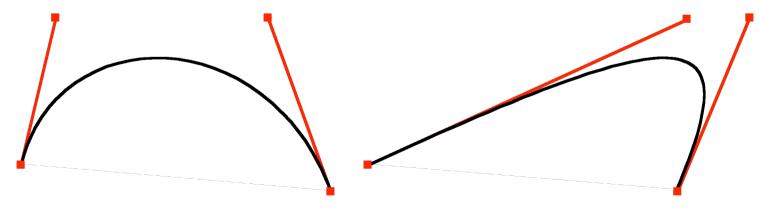
$$B_{i}^{n} = {n \choose i} t^{i} (1-t)^{n-i} \qquad b_{0}^{n} = \sum_{j=0}^{n} b_{j} B_{j}^{n}(t)$$

- Explizite Darstellung mit Bernsteinpolynomen
- Bézier-Kurven 3. Ordnung

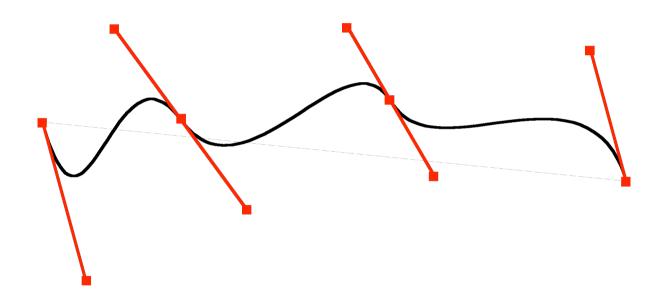
Kontrollpolygon durch vier Punkte:

Anfangspunkt (b₀) und Endpunkt (b₃)

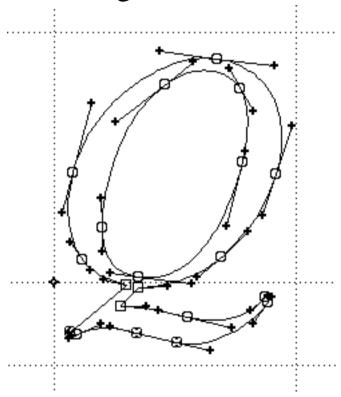
2 Kontrollpunkte (b₁, b₂)



- zusammengesetzte Kurve
 - mehrere Bézier-Splines zur Darstellung einer Kurve
 - Interpolationseigenschaft
 - Kontrollpunkte so legen, daß die Kurve glatt wird



- PostScript Type-1 Fonts
 - Fontparameter
 - Zeichenparameter
 - Bézier Kurven zur Beschreibung des Umrisses
 - 'Hints' zur Detailverbesserung

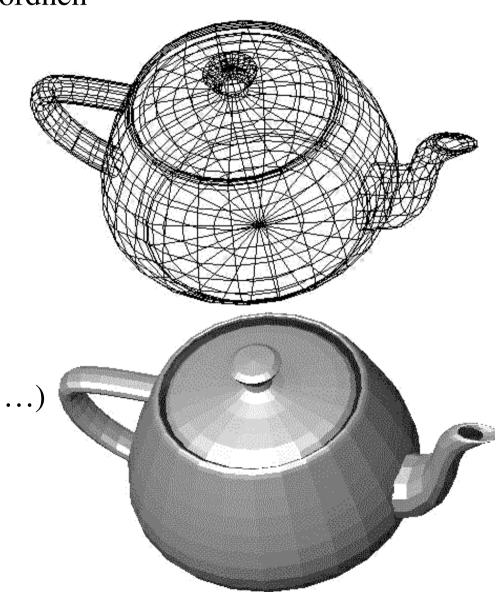


• TrueType oder andere Outline-Fonts benutzen ähnliche Kurven

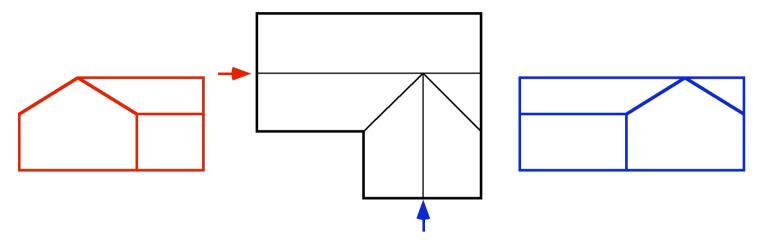
6.2.2 Geometrische Objekte => siehe Kap. 5

6.2.3 3-D Grafik

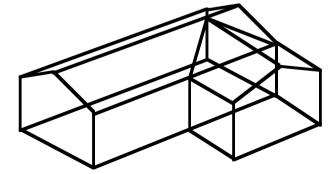
- Modellieren
 - Topologie und Geometrie
 - geometrische Objekte erzeugen und anordnen
 - Attribute festlegen(Glanz, Farbe, Durchsichtigkeit)
 - Texturen bestimmen
 - Lichtquellen anordnen
- Rendering
 - Kameratyp und -position
 - Renderer wählen
 - Abbild berechnen
- Interagieren
 - Zeigemittel (Spacemouse, Handschuh, ...)
 - Auswählen (picking)
 - Navigieren



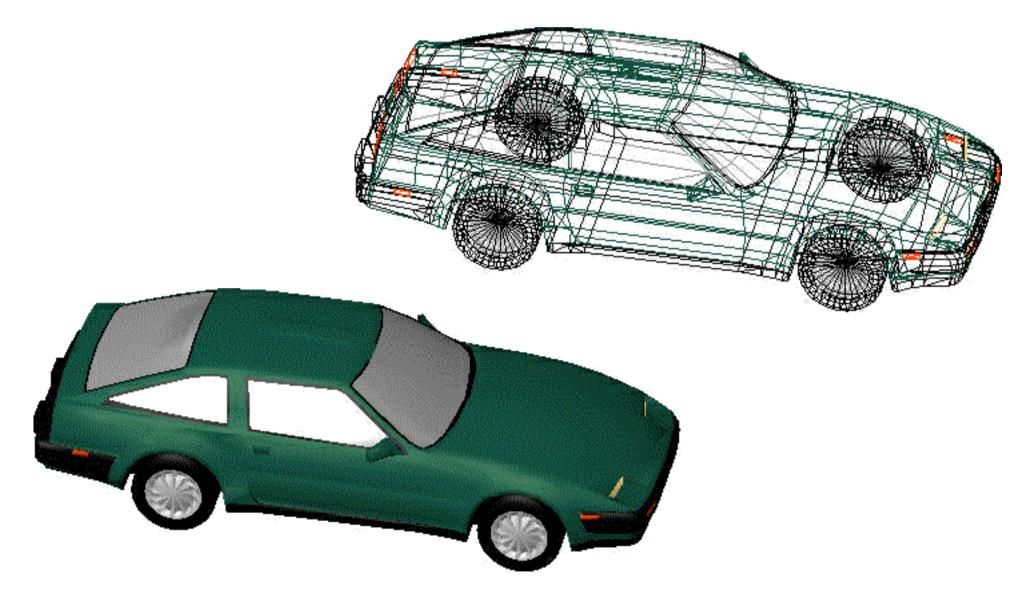
- Präsentation meist zweidimensional
 - Leinwand, Bildschirm, Papier
 - Projektion von 3-D Szenen auf 2-D Ebene
 - Tiefenhinweise gehen teilweise verloren
- Ansicht und Aufsicht



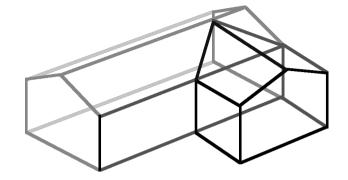
• Projektion und Drahtmodell

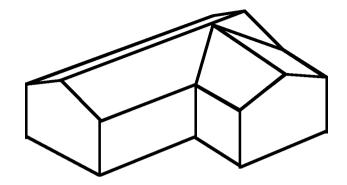


- Drahtmodell (wire frame)
 - (fast) ohne Tiefeneindruck

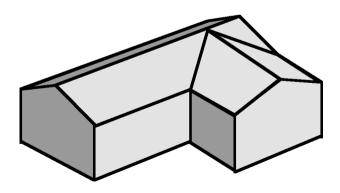


- Depth-Cueing
 - Linien 'vorne' hervorheben



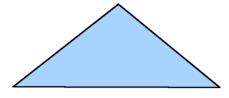


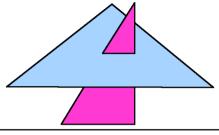
- Animation: Drehen um eine Achse
- Entfernen verdeckter Linien
- Verbesserung der Darstellung
 - Füllen der Flächen



- Entfernen verdeckter Flächen Algorithmus von E. Catmull Tiefenpufferalgorithmus (z-buffer) Pixel = (R, G, B, Z)

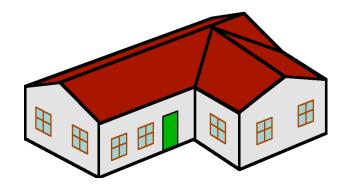
IF newpix.z<pixmap[x,y].z THEN pixmap[x,y]:=</pre>



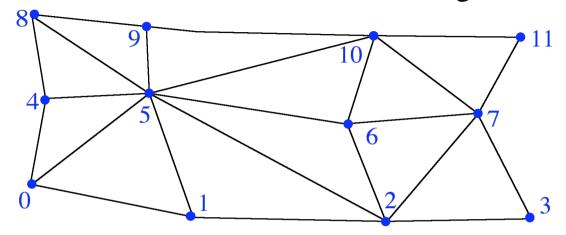


newpix;

- Schattierungen simulieren Lichteinfall
- realistische Farben, Detail

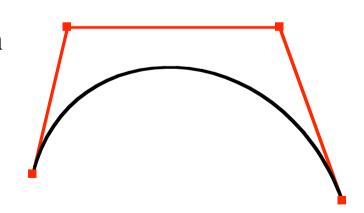


- TriGrid: Gruppe von Dreieck-Facetten
 - vereinfachte Oberflächenbeschreibung

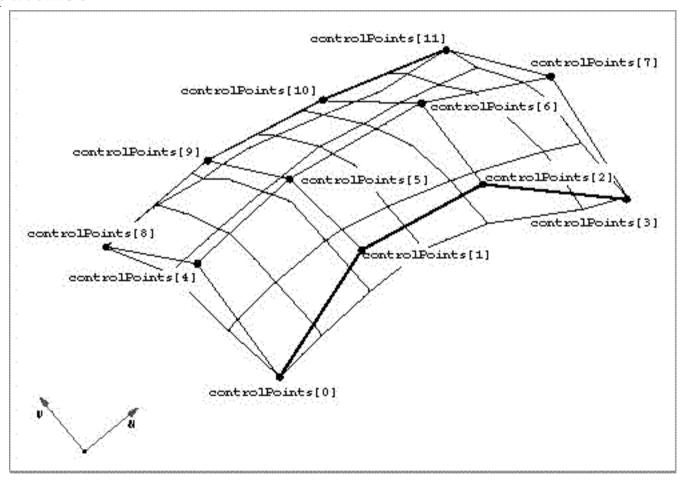


• Splines

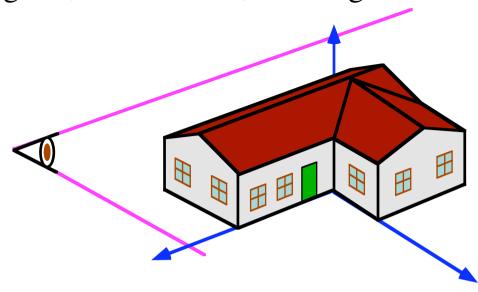
- stückweise definierte Kurve
- Anpassung an vorgegebene Kurve
- viele Spline-Typen mit besonderen Eigenschaften
- Bézier, kubische Splines, deBoor
- NURB: nonuniform rational B-spline



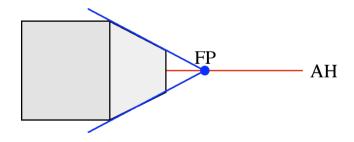
- Spline-Patches
 - stückweise Beschreibung von Oberflächen (patches)
 - Flächen als 3-dimensionales Analogon von Splines
 - Facetten sind Vierecke mit Splines als Kanten
 - NURB-patches

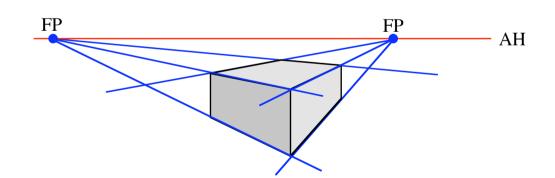


• Kamera: Betrachtungsort, Blickwinkel, Öffnungswinkel



- Perspektivische Projektion
 - Fluchtpunkt(e)





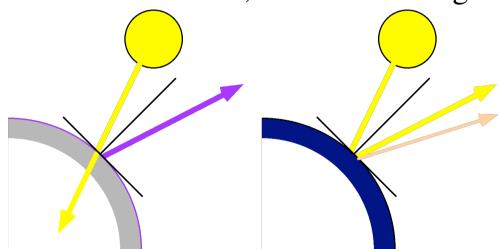
- Maße nicht korrekt ablesbar

• Parallelprojektion

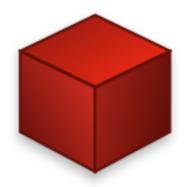
- orthographische Projektion: Grundriß, Aufriß
- schiefe (axonometrische) Projektion
- isometrische Projektion

Beleuchtung

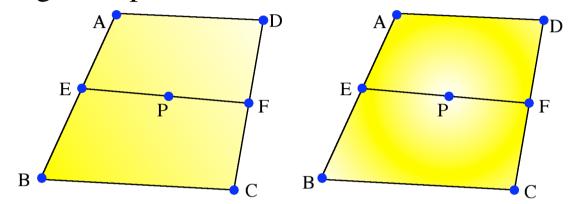
- Umgebungslicht (ambient), Punkt-Licht
- diffuse Reflektion
- Objekte werden von Lichtquelle angestrahlt
- Licht wird teilweise reflektiert, teilweise durchgelassen



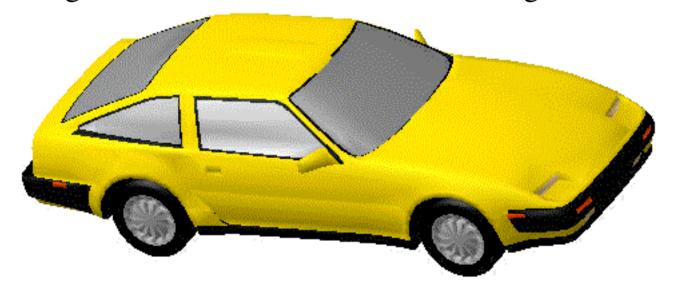
- spiegelnde Reflektion
- imitierendes Modell von Bui-Tung Phong, 1975



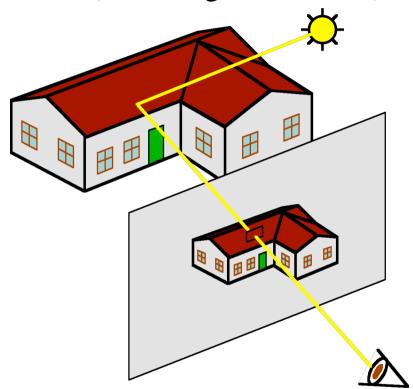
- Flat Shading: ein Farbwert pro Oberflächen-Facette
- Smooth Shading
 - Farbverlauf auf den Facetten
 - Gouraud-Shading: Interpolation zwischen Farbwerten an Eckpunkten



- Phong-Shading: individuelle Intensitätsberechnung für Flächenpixel



- Ray-Tracing
 - Reflektionen von Reflektionen, mehrfache Spiegelung
 - Strahlpfad berechnen
 - vorwärts und rückwärts (vom Auge zum Licht)



- rechenintensiv
- Berechnung blickpunktabhängig

Radiosity

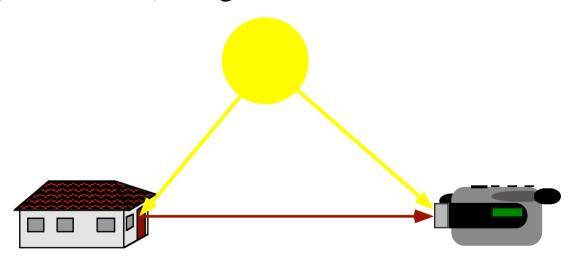
- sichtunabhängige Berechnung
- Einteilung der Oberflächen in patches
- Emitter und Reflektor
- Beleuchtungseinfluß auf alle anderen patches berechnen
- Formeln aus der Wärmelehre
- Abbrechen der Berechnung unter einem Grenzwert

• Textures

- Oberflächenstruktur (Holz: Maserung)
- 'Bekleben' der Oberflächen mit Muster
- Bilder und Filme als Texturen
- VRML: Virtual Reality Markup Language
 - textuelle Beschreibung von 3D-Objekten und Szenen
 - primitive Objekte (cylinder, ...)
 - Transformation, Gruppierung, Oberflächeneigenschaften
 - Texturen (MPEG-Filme)
 - Objekte und Hyperlinks
 - Sensoren erzeugen Events für andere Objekte

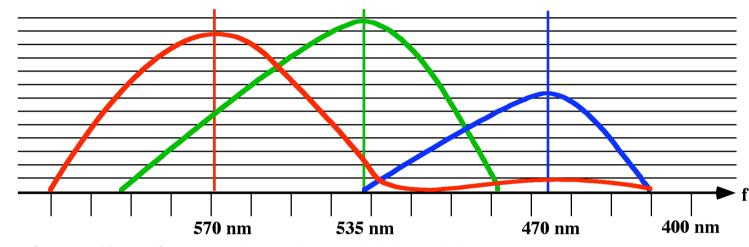
6.3 Standbilder

- Kontinuerliche Verläufe
 - Film hat höhere Auflösung als Auge
 - Abzüge, Bücher
 - Guter Druck typisch 2500 dpi
- Farbe
 - Lichtquelle (, Reflektion), Auge/Kamera/...:



- Reflektiertes Licht = Licht - absorbiertes Licht = Oberflächenfarbe

- Spektrum und Empfindlichkeit des menschliche Sehapparates
 - 120 M Stäbchenzellen für Helligkeit in der Peripherie

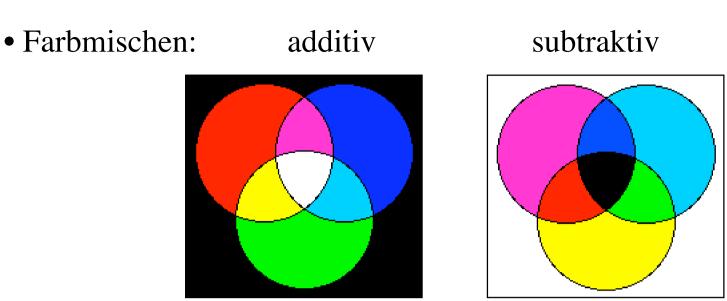


- 7 M Zapfenzellen für Farbe (570, 535, 455 nm)

• Mensch sieht bis zu 350.000 Farbnuancen



Abschattung



Rot

Farbmodelle

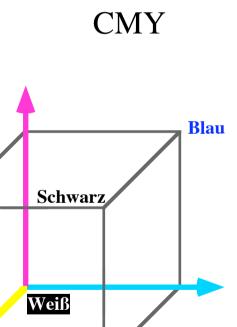
Cyan

RGB

Magenta

Weiß

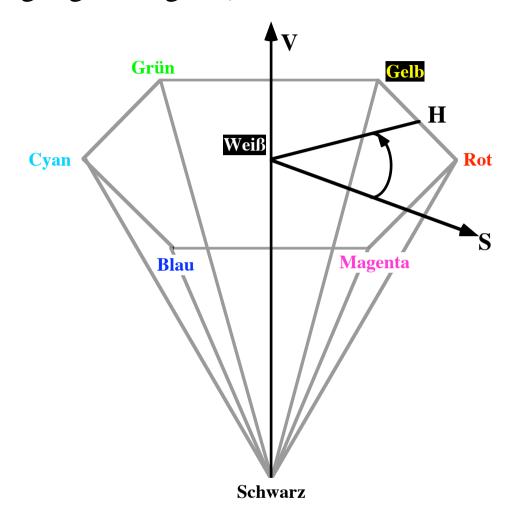
Schwarz



Grün

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} Wei\beta \\ Wei\beta \\ Wei\beta \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

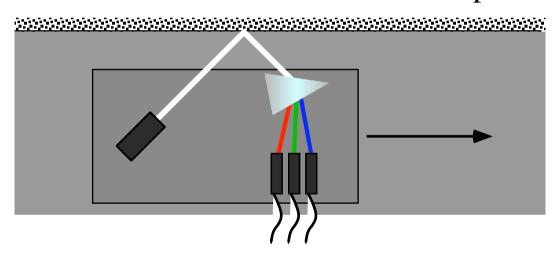
- HSV (Ton, Sättigung, Helligkeit)

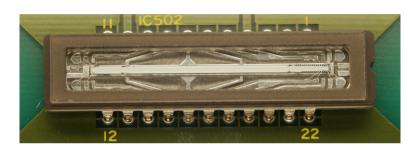


• Digitalisierung

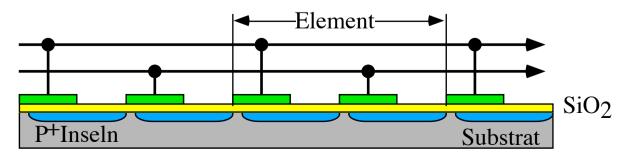
- Horizontale und vertikale Diskretisierung (Zerlegen in Pixel)
- Diskretisierungsschritt entspricht Auflösung: 72 bis 6000 dpi
- Bild wird angeleuchtet und Licht auf Detektor reflektiert
- Quantisierung: 8 oder 12 Bit für Graustufen

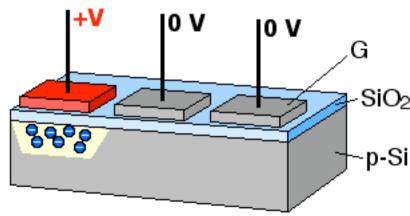
8, 16 oder 24 Bit für Farbe eventuell mit Farbpalette



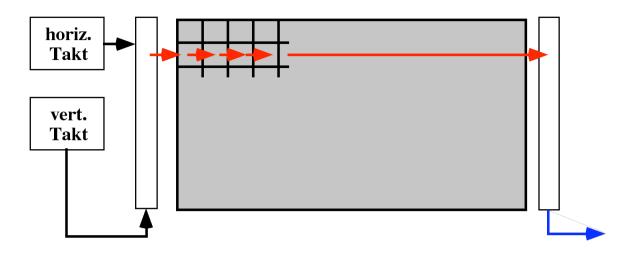


• CCD-Zeile



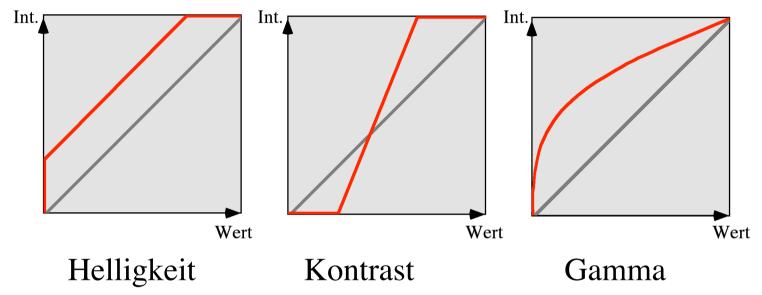


• Digitale Kameras benutzen CCD-Matrix



- ca 3000-4000 ppi
- Unterschiede zwischen Zellen
- nur 50 80% der Chipfläche ist mit aktiven Elementen bedeckt
- Bildkodierung
 - RGB wird meist bei Computermonitoren verwendet
 - CMYK (Cyan, Magenta, Yellow, Schwarz) besonders für Druck
 - HSV für Fernsehen

- Aufbereitung nach der Digitalisierung
 - Kalibrierung der Farbwerte
 - Helligkeitsregelung, Kontrastverstärkung und 'Gamma' pro Farbkanal
 - Vorsicht: Color-Matching verwendet auch γ



- Datenmenge kann groß werden
 - Auflösung für Weiterverarbeitung wichtig (Druckgewerbe)
 - (300 * 4 [inch]) * (300 * 6 [inch]) * 3 Bytes = 6.480.000 Bytes => Kompression

6.4 Video

- S/W Fernsehen (eigentlich Graustufen)
- Auflösung wesentlich geringer als bei Standbildern

	Zeilen Pu	nkte/Zeile	Bilder/s					
CCIR 601	486	720	30	59,94 Hz				
	586	720	25	50 Hz				
CIF	288	352	25	Common Intermediate Format				
QCIF	144	176	25	Quarter CIF				
SIF	240	352	30	Standard Intermediate Format				

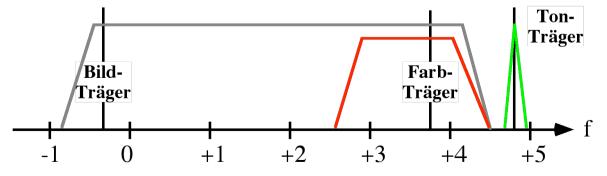
Fernsehnormen

- Halbbilder (Felder, gerade/ungerade Zeilen) mit doppelter Frequenz
- Farbfernsehen: PAL, SECAM: 50 Hz

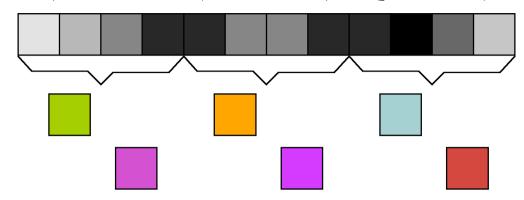
NTSC: 59,94 Hz (in Europa 50 Hz)

- Bildwiederholrate = Übertragungsrate
- HDTV: High Definition TV
 - 1366-1920 * 720-1080; (Bildwiederholrate 50 oder 59,94 Hz)
 - Bildwiederholrate ≠ Übertragungsrate (24 Hz, 36 Hz, 72 Hz)
 - MPEG-2/4 zur Übertragung

- Kameras produzieren RGB
 - drei Übertragungskanäle
 - Synchronisation?
 - => Mischsignal
- Composite
 - NTSC (National Television Systems Committee, ...)
 - PAL (Phase Alternating Line)
 - SECAM (Sequentiel Couleur avec Memoire)
 - Grundidee: SW-Fernsehen + irgendwas = Farbe
 - Farbraum mit Luminance und Chrominanz
 - Luminance := SW-Signal
- Farbraum HSV
 - Chrominanzsignal mit niedrigerer Bandbreite
 - auf Subcarrier (3,58 MHz)



• Farbauflösung des Auges schlechter -> Unterabtastung z.B.: 4:1:1 (YUV, PAL), 15:5:2 (YIQ, NTSC)



- Koeffizienten entsprechen Farbempfindlichkeit des Auges
- NTSC: YIQ (In-phase and Quadrature, I: 1,3 MHz, Q: 0,45 MHz)

$$Y = 0.30 R + 0.59 G + 0.11 B;$$

 $I = 0.60 R - 0.27 G - 0.32 B;$
 $Q = 0.21 R - 0.52 G + 0.31 B;$

• PAL: YUV (U, V: 1,3 Mhz)

$$Y = 0.30 R + 0.59 G + 0.11 B;$$

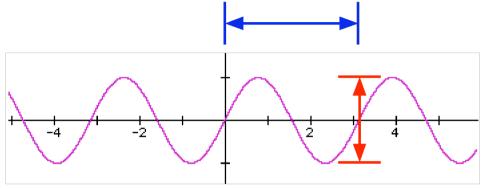
 $U = (B-Y) * 0.493 = -0.15 R - 0.29 G + 0.44 B;$
 $V = (R-Y) * 0.877 = 0.61 R - 0.52 G - 0.10 B;$

VHS noch stärker analog komprimiert

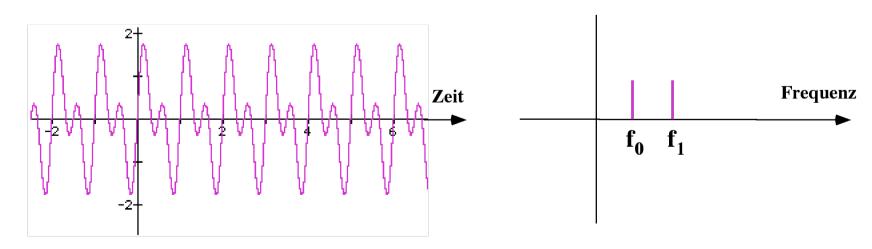
6.5 Audio

6.5.1 Audio-Eigenschaften

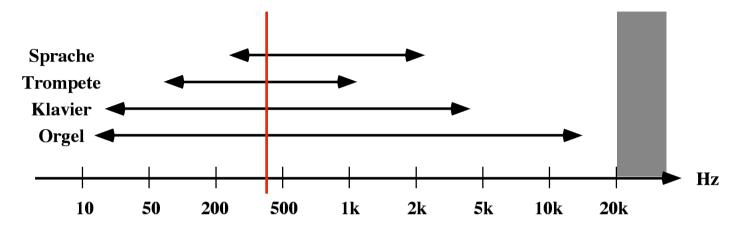
• Frequenz und Amplitude



- Amplitude -> Lautstärke (gemessen in dB)
- Frequenz (1m/Wellenlänge) -> Tonhöhe
- Fourier: Jede Schwingung kann als Summe von Sinusschwingungen dargestellt werden: $f(x) = \sin 2\pi x + \sin 4\pi x$



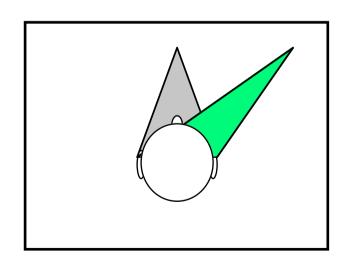
• Typische Frequenzbereiche



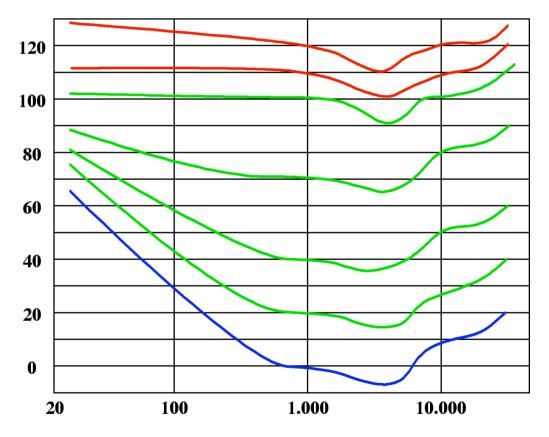
- Telefon 300Hz 3.400 Hz
- Heimstereo 20 Hz 20.000 Hz
- UKW (FM) 20 Hz 15.000 Hz

• Räumliches Hören

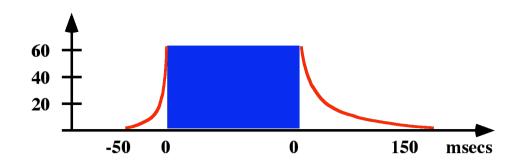
- Lautstärke
- Laufzeitunterschiede zu den Ohren
- Spektrale Analyse nach Ohrposition
- Filterfunktionen durch Außenohr
- Echos



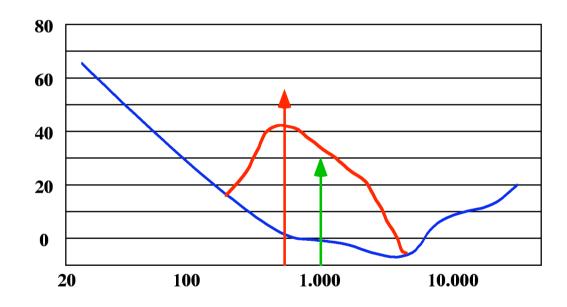
- Menschliches Hörvermögen
 - 20 20.000 Hz
 - hohes zeitliches Auflösungsvermögen
 - logarithmisch bezüglich Amplitude
- Lautstärkeempfinden nach Fletcher und Munson



- Abschattung
 - Zeit



- Frequenz

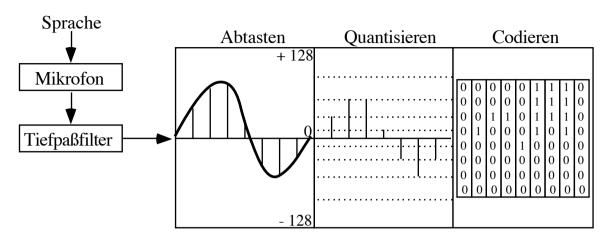


- Phase
- zwei gleiche, phasenversetzte Schwingungen können sich auslöschen:

$$y = \sin x + \sin (x + \pi)$$

6.5.2 Digitale Repräsentationen (PCM, CD-Audio, DAT, ...)

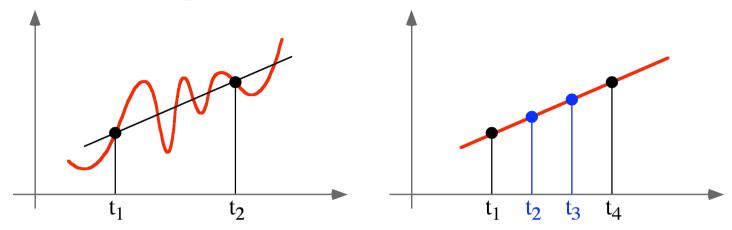
• Digitalisierung am Beispiel Telefon



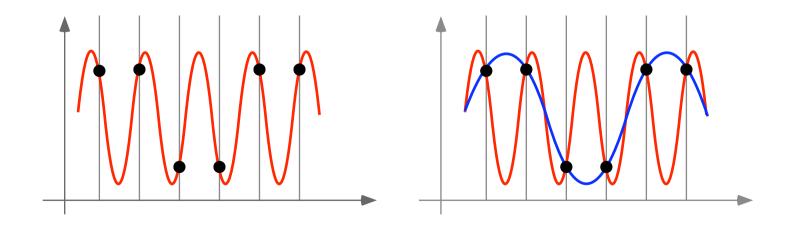
- Allgemein
 - zeitliche Diskretisierung (Abtasten, Sampling) Einteilung der Zeitachse in einzelne Stücke
 - Wert-Diskretisierung (Quantisierung) digitalen Näherungswert finden Reelle Zahl vs. Real/Integer

• Abtasttheorem

- Anzahl Abtastwerte pro Zeiteinheit?

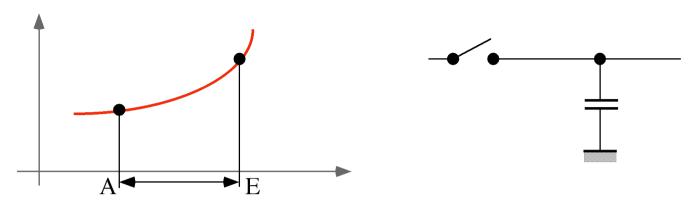


- Abtastfrequenz > 2 * (höchste Frequenz)
- [Whittaker 1915/1929, Borel 1897]

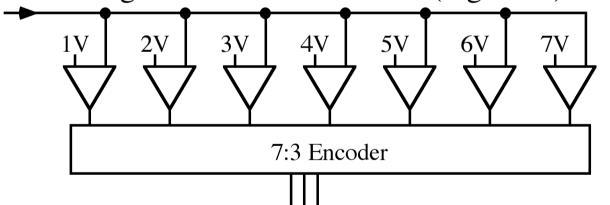


- Aliasing bei zu niedrigen Abtastraten

- Tiefpaß-Filter gegen Aliasing (siehe DSP-Kapitel)
- Sample-and-Hold

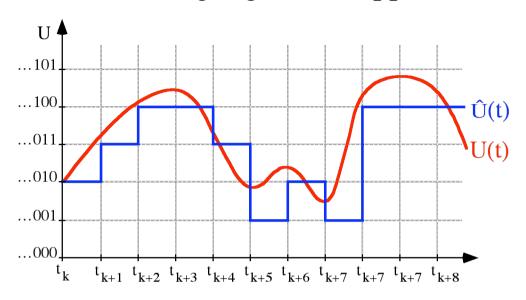


- Quantisierung (ADC)
 - Wandlung des analogen Wertes in diskreten (digitalen) Wert



- Quantisierungsfehler
- 6 dB pro Bit => 96 dB bei 16 bit (CD-A)

• Diskretisierung und Quantisierung ergeben Treppenfunktion



• Codierung

- als Integerzahl (CD: 16 bit)
- als Pseudo-Real (A-law, μ -law: 8 bit)

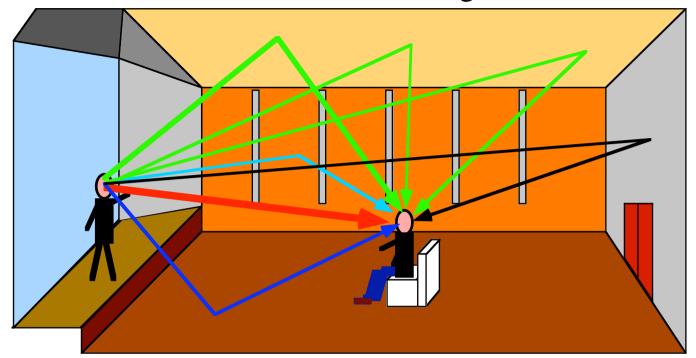
± E x p	M a	n t
---------	-----	-----

- als Differenzen

 typische Verfahren 	bit	samples/sec	Kanäle	Datenstrom
CD-Audio	16	44.100	2	~1.4 Mbit/s
A-law (ISDN)	13 -> 8	8.000	1	64 kbit/s
ADPCM (Telefon)	13 -> 2	8.000	1	16 kbit/s

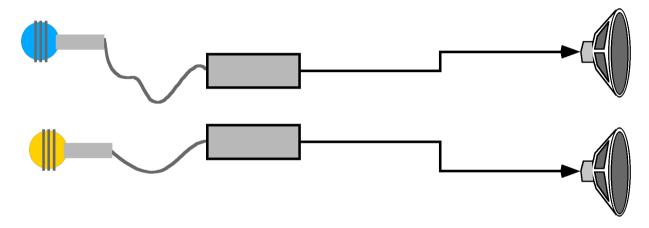
6.5.3 Raumton

• Reflexionen von Wänden, Decke, Boden, Gegenständen



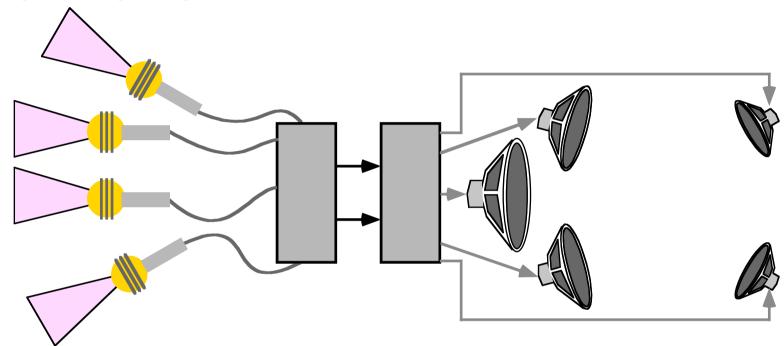
- Wahrnehmung der
 - Signalstärke
 - Richtung der Quelle
 - Dämpfung durch Kopf in höheren Frequenzen
 - Laufzeitunterschiede (650 µsec hörbar)
- Simulation der Reflexionen durch Laufzeitunterschiede

- Kopfhörer
 - kontrollierte Umgebung
 - Bewegungssensor: Kopfdrehung, Ortsveränderung
 - keine Richtungsortung
- Stereo: zwei Kanäle, links und rechts



- Simulierter Raumklang
 - Reflektionen durch Verzögerung simulieren (-> Hall)

- Aufwendige Lautsprecheranordnung
 - Surround Sound (Dolby, DTS)
 - Raumaufnahme oder Simulation
 - Richtungsortung möglich



7. Theoretische Informatik

7.1 Automaten und formale Sprachen

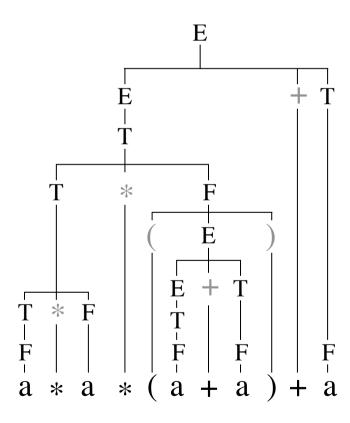
- Formale Sprache
 - Alphabet Σ
 - Σ^* Menge aller Worte über Σ
 - Formale Sprache ist Teilmenge von Σ^*
- Beispiel EXPR: korrekt geklammerte arithmetische Ausdrücke
 - $-\Sigma = \{(,),+,-,*,a\}$; a kann Variable oder Konstante sein
 - $(a-a)*a+a/(a+a)-a \in EXPR$
 - $-((((a+(a))))) \in EXPR$
 - $-((a+)-a) \notin EXPR$
 - wann ist ein Ausdruck w korrekt geklammert, also $w \in EXPR$?
- Grammatik ist 4-Tupel $G=(V, \Sigma, P, S)$
 - V endliche Menge von Variablen
 - Σ endliche Menge von Symbolen (Terminalsymbole)
 - $V \cap \Sigma = \emptyset$
 - P Produktionen (Regeln)
 - $S \subseteq V$ ist die Startvariable

- Produktionen
 - $u, v \in (V \cup \Sigma)^*$
 - Relation $u \Rightarrow_G v$
 - u = xyz, v = xy'z, $x,z \in (V \cup \Sigma)^*$
 - $-y \rightarrow y' \in P$
- Sprache $L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^*_G w \}$
- Ableitung von w_n
 - Folge $(w_0, w_1, w_2, ..., w_n)$
 - $w_0 = S, w_n \in \Sigma^*$
 - $W_0 \Rightarrow W_1 \Rightarrow W_2 \Rightarrow \dots \Rightarrow W_n$
 - nichtdeterministisch
- Ausdruck korrekt geklammert?
 - Ableitung existiert für Grammatik G

$$-G = (\{E,T,F\}, \{(,),a,+,*\}, P, E)$$

$$-P = \{ E \rightarrow T, E \rightarrow E+T, T \rightarrow F, T \rightarrow T*F, F \rightarrow a, F \rightarrow (E) \}$$

$$-a*a*(a+a)+a \in L(G)$$
?



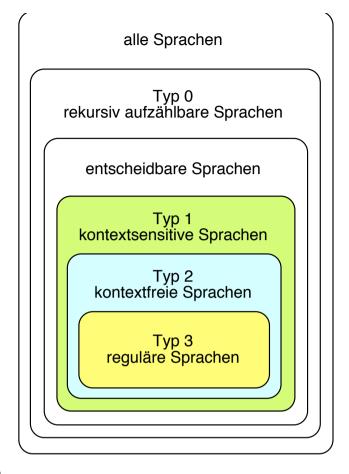
- Grammatik-Typen
 - Definition: jede Grammatik vom Typ 0
 - Typ 1 kontextsensitiv:

$$\forall w_1 \rightarrow w_2 \in P \text{ gilt } |w_1| \leq |w_2|$$

- Typ 2 kontexfrei: $\forall w_1 \rightarrow w_2 \in P$ gilt $w_1 \in V$
- Typ 3 regulär: Typ 2 und $w_2 \in \Sigma \cup \Sigma V$ (w_2 Terminalsymbol(+Variable))
- Chomsky-Hierarchie
 - Sprache L vom Typ x

 \exists Grammatik G vom Typ x mit L(G) = L

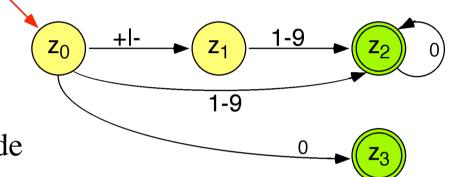
- kontextsensitiv: $uAv \rightarrow uxv$
- kontextfrei: auch ohne 'passenden' Kontext ersetzen
- Syntaxanalyse für Programmiersprachen
- Spezialklassen zwischen Typ 2 und 3: LL(k) und LR(k)
- Entscheidbarkeit
 - \exists Algorithmus, der in endlicher Zeit feststellt, ob $w \in L(G)$
 - Typ 1,2,3 entscheidbar
 - Es gibt Typ 0 Sprachen, die nicht entscheidbar sind

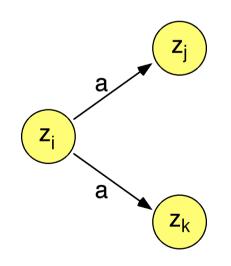


- Automaten
 - akzeptieren ein Wort
 - Menge aller akzeptierten Wörter: Sprache
- Deterministischer endlicher Automat (DFA)
 - $M = (Z, \Sigma, \delta, z_0, E)$
 - Z endliche Menge der Zustände
 - Σ endliches Eingabealphabet, $Z \cap \Sigma = \emptyset$
 - Überführungsfunktion $\delta: Z \times \Sigma \rightarrow Z$
 - $z_0 \in Z$ ist Startzustand, $E \in Z$ Endzustände



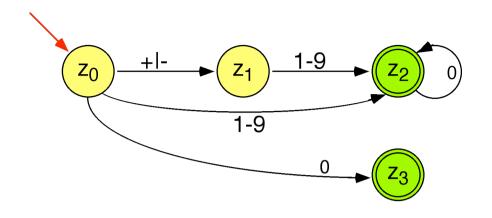
- gerichtet, beschriftet
- Knoten sind Zustände
- Pfeil auf Eingangsknoten
- akzeptierende Endknoten als Doppelkreis
- Kanten von z1 nach z2 mit a beschriftet: $\delta(z_0,a) = z_1$
- Durch DFA erkennbare Sprachen sind regulär (Typ 3)
- Nichtdeterministischer endlicher Automat (NFA)
 - Zustandsübergang in verschiedene Folgezustände bei gleicher Eingabe





- Implementierung von Automaten
 - Tabelle

Z	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'+'	'_'	
0	3	2	2	2	2	2	2	2	2	2	1	1	-1
1	-1	2	2	2	2	2	2	2	2	2	-1	-1	-1
2	2	2	2	2	2	2	2	2	2	2	-1	-1	-1
3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1



- Initialisierung der Tabelle

```
void tabinit(int tab[4][128])
{ int i,j = 0;
  for (i=0; i< 4; i++)
    for (j=0; j< 128; j++) tab [i][j]= -1;
  for (i='0'; i<='9'; i++) tab[0][i] = 2;
  tab[0]['+']= 1; tab[0]['-']= 1;
  tab[0]['0'] = 3;
  for (i='1'; i<='9'; i++) tab[1][i] = 2;
  for (i='0'; i<='9'; i++) tab[2][i] = 2;}</pre>
```

```
#include <stdio.h>
/* function tabinit einsetzen */
int istEndzustand(int zustand)
  {return zustand ==2 | zustand ==3;}
int main(void)
{ int zeichen, zustand = 0;
  int tabelle[4][128];
  tabinit (tabelle);
  while (zeichen=getchar()!='\n')
     if ((zustand = tabelle[zustand][zeichen])<0)</pre>
     break;
  if (istEndzustand(zustand))
     printf("akzeptiert\n");
     else printf("nicht akzeptiert\n");
  return 0;
```

- EBNF: erweiterte Backus-Naur-Form (ISO 14977)
 - kompakte Notation für kontextfrei Grammatiken
 - Regeln mit derselben linken Seite zusammenfassen

$$A \rightarrow \beta_1$$

$$A \rightarrow \beta_2$$
...

$$A \rightarrow \beta_n$$

wird zu: $A ::= \beta_1 \mid \beta_2 \mid ... \mid \beta_n$

- A ::= $\alpha[\beta]\gamma$ steht für

$$A ::= \alpha \gamma \mid \alpha \beta \gamma$$

 β kann einmal zwischen α und γ eingefügt werden

- A ::= $\alpha \{\beta\} \gamma$ steht für

$$A ::= \alpha \gamma \mid \alpha B \gamma$$

$$B ::= \beta \mid \beta B$$

β kann beliebig oft eingefügt werden

- manchmal Minimum und Maximum durch Index

$$A ::= \alpha \{\beta\}^4 \gamma$$

- Gruppierung mit ()

$$P = \{ E \rightarrow T, \\ E \rightarrow E+T, \\ T \rightarrow F, \\ T \rightarrow T*F, \\ F \rightarrow a, \\ F \rightarrow (E) \}$$

$$P = \{ E \rightarrow T \mid E+T, \\ T \rightarrow F \mid T*F, \\ F \rightarrow a \mid (E) \}$$

```
• Beispiel einfache Programmiersprache [Wikipedia]
                                            PROGRAM DEMO1
                                            BEGIN
                                             A0:=3:
program ::= 'PROGRAM' ,
                                             B:=45;
                                             C:=A;
              wsp , ident , wsp,
                                             D123:=B34A;
              'BEGIN' , wsp,
                                             HANS:=WURST;
              {assignment, "; ", wsp} ,
                                            TEXTZEILE:="Moin Moin";
                                            END.
              'END.'
ident ::= alphachar , { alphachar | digit }
num ::= [ "-" ] , digit , { digit }
string ::= '"' , { anychar } , '"'
assignment ::= ident , ":=" , ( num | ident | string )
alphachar ::= "A" | "B" | "C" | "D" | "E" | "F" | "G"
                    "H" | "I" | "J" | "K" | "L" | "M" | "N"
                   | "O" | "P" | "Q" | "R" | "S" | "T" | "U"
                    "V" | "W" | "X" | "Y" | "Z" ;
digit ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
wsp ::= ? ASCII Character 32 ?
anychar ::= ? all visible characters minus " ?
```

- Kontextfreie Sprachen (Typ 2)
 - werden nicht durch endlichen Automaten akzeptiert
 - Beispiel Klammerausrücke: Zählen der Klammern
 - Automat mit mehr 'Gedächtnis' als nur Zustand nötig
- Kellerautomat
 - M = $(Z, \Sigma, \Gamma, \delta, z_0, \#)$
 - Z endliche Menge der Zustände
 - Σ Eingabealphabet
 - Γ Kelleralphabet
 - Überführungsfunktion $\delta: Z \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow P_e (Z \times \Gamma^*)$ (P_e ist Menge aller endlichen Teilmengen)

Χ

- $z_0 \in Z$ ist Startzustand
- # $\in \Gamma$ das unterste Kellerzeichen
- PDA Push Down Automaton
 - Keller ist Zusatzgedächtnis
 - nichtdeterministisch
 - akzeptiert genau die kontextfreien Sprachen
 - können zum Beispiel korrekt geklammerte Ausdrücke erkennen

7.2 Berechenbarkeit

- Intutive Berechenbarkeit
 - formale Definition von 'intuitiv berechenbar' schwer

Eine Funktion f heisst berechenbar, falls ein Algorithmus (Programm) existiert, der (das) ausgehend von der Eingabe $(x_1, ..., x_n)$ in endlich vielen Schritte $f(x_1, ..., x_n)$ berechnet.

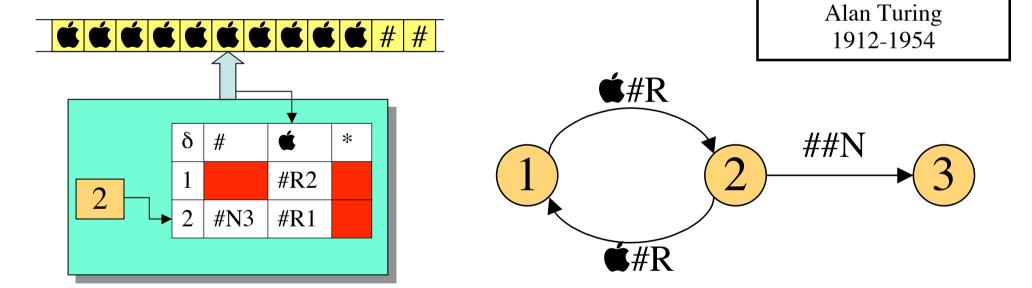
• Beispiele

```
f(n) = 2*n

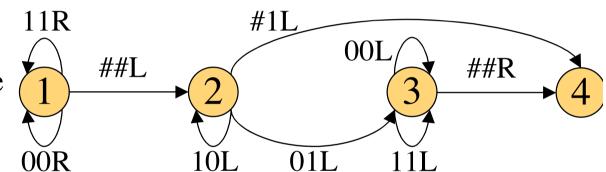
f_{\pi}(n) = 1 falls n Anfang der Dezimalbruchentwicklung von \pi ist sonst 0
```

- Gibt es nicht berechenbare Funktionen?
 - ist für jede reelle Zahl r $f_r(n)$ berechenbar?
 - Nein: überabzählbar viele r (Cantorsches Diagonalschema ...)
 - nur abzählbar viele Algorithmen
 - oder das Cantorsche Abzählschema modifizieren

- Turingmaschine [Alan Turing, 1936]
 - Automat
 - unbegrenztes Band mit Zeichen eines Arbeitsalphabets
 - Schreib/Lesekopf
 - Bandbewegung ein Feld nach rechts oder links
 - Zustand, gelesenes Zeichen und Regel
 - Zustandsübergang, Zeichen schreiben, Band bewegen



- Formale Beschreibung
 - Z endliche Menge von Zuständen
 - Γ Arbeitsalphabet
 - $\Sigma \subset \Gamma$ Eingabealphabet
 - Überführungsfunktion δ
 - δ: Z x Γ → Z x Γ x {L,R,N} (determinsitische TM)
 - δ: Z x Γ → P(Z x Γ x {L,R,N}) (nicht-determinsitische TM)
 - z_o Startzustand
 - $-\# \in \Gamma \Sigma$: Blank
 - E ⊆ Z Menge der Endzustände
- Beispiel: Binäre Addition von 1
- Turing-Vollständigkeit
 - ein Computer, der alle mit Turing-Maschinen berechenbaren Funktionen berechnen kann (Bsp: zellulärer Automat Rule 110)

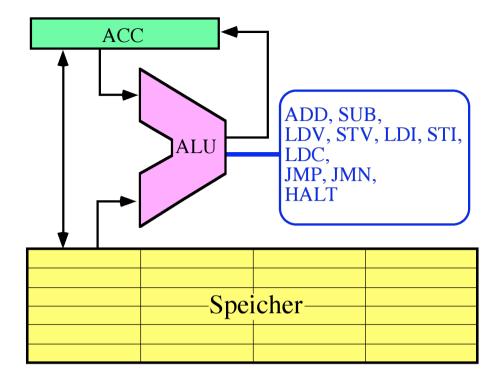


- Turingmaschinen akzeptierten Sprachen
 - $T(M) = \{x \in \Sigma^* | z_0 x \rightarrow^* \alpha z \beta; \alpha, \beta \in \Gamma^*; z \in E\}$
 - allgemeine TM akzeptieren Typ 0 Sprachen
- Linear beschränkte Turing-Maschine: endliches Band
 - akzeptieren kontextsensitive Sprachen (Typ 1)
- Turing-Berechenbarkeit
 - $f: \mathcal{N} \rightarrow \mathcal{N}$ Turing-berechenbar falls \exists TM
 - $\forall n_1, ..., n_k, m \in \mathcal{N}$ gilt
 - $f(n_1, ..., n_k) = m$ genau dann wenn $z_0 n_{1b} n_{2b} ... n_{kb} \rightarrow * #... # z_e m_b # # #$
- Churchsche These [Alonszo Church, 1936]

Die Klasse der Turing-berechenbaren Funktionen ist genau die Klasse der intuitiv berechenbaren Funktionen.

- eine These, von deren Richtigkeit fast alle Informatiker überzeugt sind
- wahrscheinlich, hilfreich, 'funktioniert' gut

- Registermaschinen
 - RM oder auch RAM
 - unendlich viele Registerzellen (~Speicher)
 - Akkumulator und Einadressbefehle
 - Load, Store, Arithmetik, Sprünge
 - Adressierung: Konstante, direkt, indirekt
 - ähnlich MIMA
 - intuitiv: kann alle intuitiv berechenbaren Funktionen berechnen
 - kann Turingmaschinen simulieren



- While Berechenbarkeit
 - kann Touringmaschine simulieren und umgekehrt

• LOOP Berechenbarkeit schwächer

- Primitiv rekursive Funktionen
 - konstante Funktionen sind primitiv rekursiv
 - Identität und Nachfolgerfunktion primitiv rekursiv
 - -f(0,...) = g(...)
 - -f(n+1,...) = h(f(n,...),...)
 - äquivalent zu LOOP-Berechenbarkeit
- Arithmetik als primitiv rerkursive Funktionen

```
-add(0,x) = x /* Identität */
```

- -add(n+1,x) = succ(add(n,x))
- $\operatorname{mult}(0, x) = 0$
- mult(n+1,x) = add(mult(n,x),x)
- μ-Rekursion
 - $-g(x_1,...,x_k) = \min\{n | f(n,x_1,...,x_k) = 0, \forall m < n f(m,x_1,...,x_k) \text{ definient}\}$
 - terminiert evtl. nicht

Die Klasse der μ -rekuriven Funktionen stimmt genau mit der Klasse der While-, RAM- und Touring-berechenbaren Funktionen überein.

• Ackermann-Funktion

- berechenbar, nicht primitiv rekursiv
- ack(0,n) = n+1
- ack(m,0) = a(m-1,1) ; m>0
- ack(m,n) = ack(m-1,a(m,n-1)); m,n>0

m/n	0	1	2	3	4
0	1	2	3	4	5
1	2	3	4	5	6
2	3	5	7	9	11
3	5	13	29	61	125
4	13	65533	2^{65536} -3	a(3,(4,2))	a(3,a(4,3))

Halteproblem

- existiert ein Programm, das für jedes Paar von TM und Eingabe berechnet, ob die TM auf der Eingabe anhält?
- Gegenbeispiel von Turing, 1936
- semi-entscheidbar: TM hält entweder oder läuft endlos
- Gödelscher Unvollständigkeitssatz

Jedes Beweissystem für die Menge der wahren arithmetischen Formeln ist notwendigerweise unvollständig

- es blieben immer wahre arithmetische Formeln, die nicht beweisbar sind
- in jedem logischen System existieren Aussagen, die weder bewiesen noch widerlegt werden können

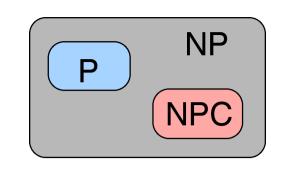
7.3 Komplexitätstheorie

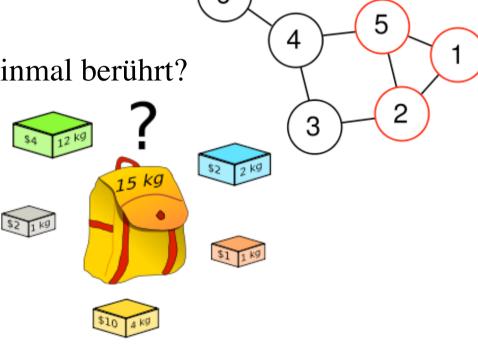
- Berechenbar ist nicht unbedingt 'praktisch' lösbar
 - Ressourcen-Verbrauch: Rechenzeit, Speicher
 - stark wachsende Probleme
- Problem der Klasse P
 - in polynominaler Zeit lösbar
 - Suchen in Listen mit n Elementen O(n)
 - Sortieren O(n log n)
 - Sieb des Erathosthenes O(n²)
- SAT-Problem (Satisfiability)
 - gegeben boolsche Gleichung g
 - SAT findet eine Variablenbesetzung x, so daß g(x) = true
 - $O(2^n)$
- Klasse NP
 - nichtdeterministisch polynomiale Zeit
 - polynomial entscheidbar, ob Kandidat Lösung ist
 - exponentielle Komplexität
 - jedes Problem in NP lässt sich auf SAT-Problem polynomial zurückführen

- NP-vollständig (NPComplete)
 - p ist NPC, falls SAT auf p zurückführbar ist
 - alle 'gleich schwer'
 - Tausende NPC-Problem bekannt
 - P=NP???



- Clique = Teilmenge von Knoten in Graphen paarweise verbunden
- existiert k-Clique in einem Graphen?
- Hamilton-Problem
 - ∃ ein Weg der jeden Knoten genau einmal berührt?
 - Bsp: Springerproblem im Schach
- TSP Travelling Salesman Problem
 - kürzester Hamilton-Kreis?
- Rucksack-Problem
- Anwendungen
 - Kryptographie, Optimierung
 - Stundenplanproblem, Verdrahtung in Chips und Platinen





8. Anwendungen

8.1 Datenbanken

- Datenbanksystem (DBS)
 - Datenbank (DB)
 - Datenbankmanagementsystem (DBMS)
- Datenbank
 - Datenbestände
 - Beschreibung
 - im Jargon: Hardware, Software und Daten
- Datenbankmanagementsystem
 - Softwaresystem
 - Verwaltung und Bereitstellung der Daten
- Datenmodell
 - Diskursbereich
 - Abbildung Diskursbereich auf Datenbank
 - Strukturen, Operationen, Integritätsbedingungen

- Datenbanksprache
 - Definitionssprache (DDL)
 - Manipulation und Abfrage (DML)
- Diskursbereich
 - anwendungsrelevanter Ausschnitt der Realität
 - Handelsunternehmen: Artikel, Lager, Kunden, Lieferanten, Konten ...
 - Universität: Studenten, Assistenten, Professoren, Lehrveranstaltungen, ...
- Entity-Relationsship-Model (ERM)
 - Ableitung des Diskursbereiches
 - Gegebenheiten
 - Beziehungen
- Entwurfsziele ERM
 - schematisch, zeitunabhängiges Modell
 - Unterstützung der frühen Entwurfsphasen
 - Graphische Veranschaulichung von Zusammenhängen
 - Kommunikation: Anwender, Feldingenieur, Entwickler
 - Abbildbarkeit in Datenmodelle (insbesondere in das Relationenmodell)

- Elemente des ERM und graphische Darstellung
- Entity
 - diskrete Abstraktion
 - Gegenstand der Anschauung oder des Denkens
- Entity-Typ
 - Verallgemeinerung (Klassifikation) gleichartiger Entities
 - graphisch: Rechteck mit Namen
 - im Beispiel: Student, Assistant, Professor, Lehrveranstaltung, Prüfung
- Relationships
 - zwischen je einer Entity von mehreren Entity-Typen
 - nicht notwendig verschiedene Entity Typen
- Relationship-Typ
 - Verallgemeinerung (Klassifikation) gleichartiger/ähnlicher Relationen
 - graphisch: Raute mit Namen
 - ungerichtete Kanten zu Entity-Typen
 - im Beispiel: lesen, hören, prüfen, voraussetzen, arbeiten_für

Entity-Typ



• Attribut

- Merkmal zur Beschreibung
- Entity, Relationships
- Name, Merkmalswertemenge (Domäne)
- graphisch: Kreis/Ellipse mit Attributnamen
- ungerichtete Kante zum Entity/Relationship-Typ
- im Beispiel: Namen, Vorname, Adresse, Fachgebiet, Note, ...

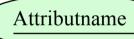
Schlüssel

- Teilmenge von Attributen
- identifizieren eine Ausprägung von Entity/Relationshiptypen
- minimal bzgl. Identifikation
- graphisch: Kreis/Ellipse mit unterstrichenem Attributnamen
- ungerichtete Kante zum Entity/Relationship-Typ
- im Beispiel: Personalnummer, Matrikelnummer, ...

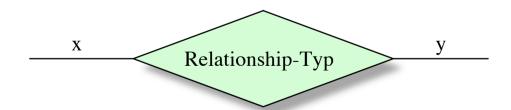
Primärschlüssel

- pragmatisch ausgewählter Schlüssel

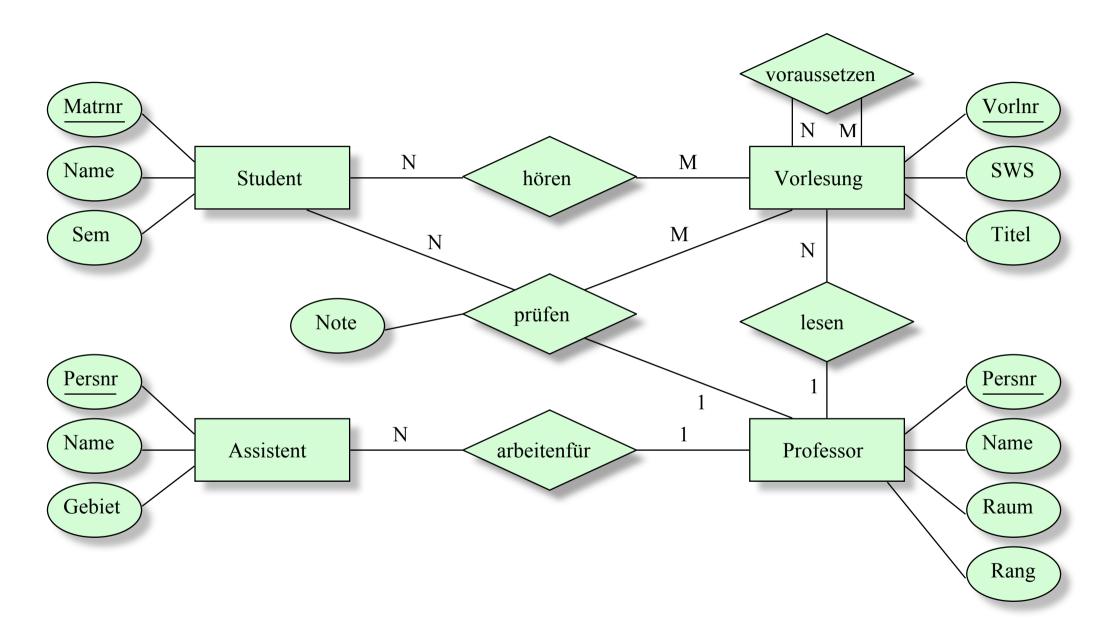




- Art des Relationship-Typs R zwischen Entity-Typen E1 und E2
 - 1:1 In R kann eine Entity aus E2 nur mit einer Entity aus E1 in Bez. stehen In R kann eine Entity aus E1 nur mit einer Entity aus E2 in Bez. stehen
 1:n In R kann eine Entity aus E2 nur mit einer Entity aus E1 in Bez. stehen m:n mehrere Entities aus E1 können zu mehreren in E2 Bez. haben
- Zuordnungscharakteristik
 - Erweiterung auf mehrstellige Relationship-Typen möglich
 - grafisch: Beschriftung der Kanten
 - Beispiel: lesen (1:n), hören (m:n), prüfen (m:n:1)



• ERM-Diagramm des Exkursbereichs Universität



- Relationales Datenmodell [Codd, 1970]
 - Relationenalgebra
 - Tabellen: Spalten (Attribute), Zeilen (Entities)
 - Integritätsbedingungen

• Relation R

- Wertemengen V_i, Attribute A_i
- Teilmenge des kartesischen Produktes der V_i
- $-R = R(A_1, A_2, ..., A_n) \mid V_1 \leftrightarrow V_2 \leftrightarrow ... \leftrightarrow V_n$
- Tupel: $r = (v_1, v_2,...,v_n)$ aus R
- Relation: Tabelle

• Im Beispiel

- Student(Matnr, Name, Semester)
- Professor (Persnr, Name, Grad, Raum)
- Vorlesung (Vorlnr, Titel, SWS, Leser)

• Integrität

- Schlüssel-, Dömänen-, Entity-, referentielle Integrität

- Operationen auf Relationen
- Vereinigung R∪S
- Differenz R-S
 - alle Tupel aus R, die nicht in S sind
- Durchschnitt $R \cap S$
- Projektion π_L
 - alle Spalten zu den Attributen aus Liste L
- Selektion $\sigma_R : R \to R$
 - alle Tupel, die Bedingung B erfüllen
- Produkt RxS
 - kartesisches Produkt aus 2 Tabellen
 - neue Tabelle mit Kombination aller Zeilen
- Verbund ><
- Division /
- Umbenennung ρ

- Überführung ERM -> RDM
- Entity-Typ
 - Relationsschema mit Namen, Attributen, Schlüsseln
- Relationship m:n
 - Relationschema mit Namen des R-typ
 - eigenen Attributen
 - Primärschlüssel der beteiligten Entity-Typen
- Relationship 1:n
 - Integration des 1-Entity-Typs in Relationschema des n-Entity-Typ
 - Primärschlüssels des 1-E-Typs als Fremdschlüssel
 - Aufnahme der Attribute des Relationship-Typs
- Relationship 1:1
 - Integration des 1-Entity-Typs in RS des anderen 1-Entity-Typ
 - Primärschlüssels des einen 1-E-Typs als Fremdschlüssel
 - Aufnahme der Attribute des Relationship-Typs
- Mehrstellige Relationship Typen
 - analog zu Relationship-Typ m:n

• Beispielrelationen

Assistent:			
Persnr	Name	Fachgebiet	Boss
3002 2125	Platon	Ideenlehre	
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005 2127	Rhetikus	Planetenbewegung	
3006	Newton	Keplersche Gesetze	

Professor:			
Persnr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36

Student:			
Matrnr	Name	Sem	
24002	Xenokrates	18	
25403	Jonas	12	
26120	Fichte	10	
26830	Aristoxenos	8	
27550	Schopenhauer	6	
28106	Carnap	3	
29120	Theophrastos	2	
29555	Feuerbach	2	

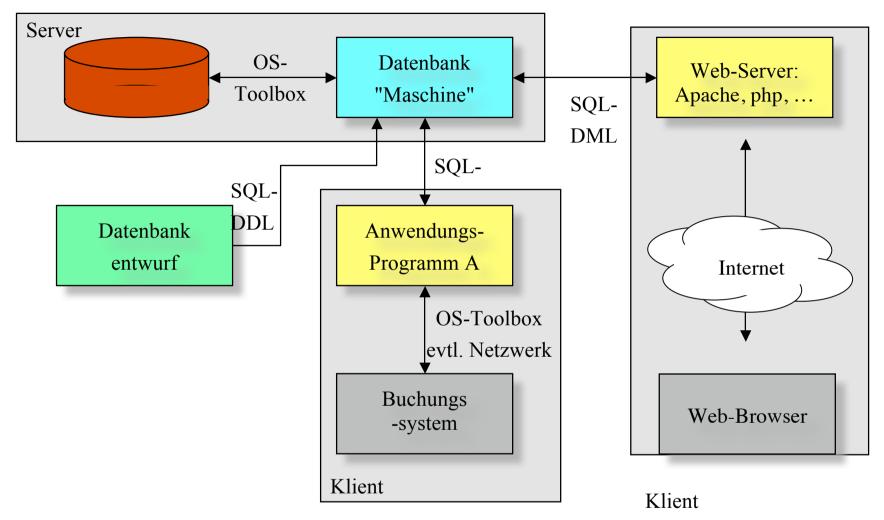
Vorlesung:			
Vorlnr	Titel	SWS	Leser
5001 2137	Grundzüge		4
5041	Ethik	4	2125
5043 2126	Erkenntnistheorie		3
5049	Gewäsch	2	2125
4052 2125	Logik		4
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126

voraussetzen:		
Vorgänger Nachfolger		
5001	5041	
5001	5043	
5001	5049	
5041	5216	
5043	5052	
5041	5052	

hören:		
Matrnr Vorlnr		
26120	5001	
27550	5001	
27550	4052	
28106	5041	
28106	5052	
28106	5216	
28106	5259	
29120	5001	
29120	5041	
29120	5049	
29555	5022	
25403	5022	

prüfen:			
Matrnr	<u>Vorlnr</u>	<u>Persnr</u>	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Datenbanksystem



• Beispiele

- Microsoft Access, dbase, ...
- Oracle, Ingres, Informix, Adabas, Sybase
- mySQL, PostgreSQL, DB2

- SQL eine relationale Datenbanksprache
 - Datendefinitionssprache DDL
 - Datenmanipulationssprache DML
 - Änderungssprache
 - Abfragesprache
- Datenmanipulation / Abfrage

<u>Standardanfragen</u>

```
select ... Projektion
from ... Relation, Produkt, Join
where ... Selektion, Differenz, Durchschnitt
union ... Vereinigung
all/any/in/exists ... Divison, Quantifizierung
group by ... Aggregation
order by ... Sortierung
```

- Einfache Anfrage
 - select Spalte(n) from Tabelle(n) where Bedingung

• Abfrage aus einer Tabelle

select Matrnr, Name, Sem select * select distinct Sws

from Student from Assistent; from Vorlesung

where Sem < 5; where Leser = 2125 and not Titel = 'Ethik';

• Abfrage über mehrere Tabellen

select Name, Titel select s.Name, v.Titel select s.Name, v.Titel

from Professor, Vorlesung from Student, hören, Vorlesung from Student s, hören h, Vorlesung v

where Persnr = Leser where Student.Matrnr = hören.Matrnr where s.Matrnr = h.Matrnr

and Titel = 'Logik'; and hören.Vorlnr = Vorlesung.Vorlnr; and h.Vorlnr = v.Vorlnr;

Spaltenauswahl

- Spaltenname, Konstante
- * alle Spalten, distinct Duplikatebeseitigung

Zeilenauswahl

- Spaltenname, Konstante
- Vergleichsoperatoren: =, !=, <>, >, >=, <, <=
- logische Verknüpfungen and, or, not
- Bereichsangaben between ... and, Muster like '...%...'
- Leertest is null, is not null

• Geschachtelte Abfragen

```
select * from Assistent

where Boss = where Note = (select avg (Note) from prüfen);

(select Persnr from Professor where Name = 'Sokrates');
```

Mengenoperationen

- Vereinigung: union

- Durchschnitt: intersect

- Differenz: except

```
(select Name from Assistent)<br/>Vorlesung(select Vorlnr from Vorlesung)select Vorlnr fromunionexceptwhere Vorlnr not in(select Name from Professor);(select Vorlnr from hören);(select Vorlnr from hören);
```

Quantifizierung

- all, any, none (für alle, ...)
- in, not in (Element von)
- exists, not exists

```
select Name select Name select *

from Student from Professor from Vorlesung

where Sem >= all where not exists where Vorlnr in

(select Sem (select * (5001, 5041);

from Student); from Vorlesung

where Leser = Persnr);
```

- Gruppierung (ähnlich Sortieren)
 - group by Spalte(n) Zusammenfassung in Ergebnistabelle
 - having Bedingung Kriterium

select Leser, sum (Sws) select Leser, sum (Sws) select Leser, Name, sum (Sws)

from Vorlesung from Vorlesung, Professor

group by Leser; group by Leser where Leser = Persnr and Rang = 'C4'

having avg (Sws) > 3; group by Leser, Name

having avg (Sws) > 3;

• Elementfunktionen

- arithmetische Funktionen: +, -, *, /, ...
- Strings: char_length, substring, II, ...
- Datum: current_time, current_date, +,-,*,...

select Matrnr, Note - 1 select Rang || '-Professur' select Matrnr, Sem, current_date from prüfen from Professor; from Student; where Vorlnr = 5001;

• Sortieren

- order by Spalte(n)
- asc/desc

select Persnr, Name, Raum select Name
from Professor from Student
where Rang = 'C4' order by Matrnr;
order by Rang desc. Name asc:

- Datenmanipulationssprache
- Einfügen
 - insert into Tabelle values Tupel;
 - insert into Tabelle Anfrage;

```
insert into Professor insert into Student (Matrnr, Name) insert into Assistent (Persnr, Name) values (2136, 'Curie', 'C4', null); values (25000, 'Goethe'); select Matrnr, Name from Student where Sem > 15;
```

• Ändern

- update Tabelle set Werte where Bedingung;

```
update Vorlesungupdate Studentupdate Professorupdate Vorlesungset Sws = 2;set Sem = Sem + 1;set Raum = 213set Titel = 'Grundlagen', Sws = 3, Leser = 2125where name = 'Russel';where Vorlnr = 5001;
```

• Löschen

- delete from Tabelle where Bedingung;

```
delete from Vorlesung delete from Student delete from voraussetzen where Vorlnr = 5001; where Sem >= 15; where Vorgänger in (select Nachfolger from voraussetzen);
```

- Datendefinitionssprache
 - create, alter, drop
 - in table, view
 - mit Rechte Vergabe (grant) und Entzug (revoke)
- Datentypen
 - char(n), varchar(n)
 - number (d), number (d,s)
 - date, long, blob
 - Namen: alphachar, { alphachar | digit }
- Tabellen
 - create definiert Tabelle
 - Spalten und Integritätsbedingungen

create table Tabellen-Name (Spalten-Name Typ, Spalten-Name Typ not null, Spalten-Name Typ not null unique,

```
create table Professor

(persnr number (5) not null unique primary key,

Name varchar (20) not null,

Rang char (2),

Raum number (4));

create table voraussetzen

(Vorgänger number (5),

Nachfolger number (5));
```

8.2 Bürosoftware

- Typische Büroarbeiten
 - Briefe schreiben
 - einfache und mittlere Berechnungen (Angebote, einfache Modelle)
 - Präsentation (Vorträge)
- Dokumentenverarbeitung
 - Integration verschiedener Medien
 - Text, Grafik, Bild, Tabellen
 - Unterstützungsfunktionen: Rechtschreibung, ...
 - Gruppenarbeit und Versionskontrolle
 - dynamische Komposition
- Office-Pakete
 - MS-Office
 - Staroffice ... OpenOffice
 - WordPerfect Office

- ...

8.2.1 Text

- vi, WordStar, Edit, Edlin
 - konzeptuell eine lange Zeichenkette mit Space zwischen Wörtern
 - 'File' = Magnetbandkonzept
- Microsoft Word, Star..., WordPerfect, Words ...
 - raffinierte Speicherstrukturen

8.2.1.1 Text

- Zeichen
 - Buchstaben, Ziffern, Sonderzeichen

```
- A, B, C, D, ..., Z, Ä, Ö, Ü
```

- -a,b,c,d,...,z,ä,ö,ü,ß
- 0 ... 9
- ·,#+*!"§\$%&/()=?'^`<>^
- Steuerzeichen für Ausgabegerät: LF, CR, FF, NL, VT, HT, ...
- Tastatur als Eingabegerät
 - Tasten für die wichtigsten Buchstaben
 - Umschalttasten
 - evtl. nicht alle Zeichen eingebbar

• Zeilen

- Einheit des Darstellungsmediums (Papier, Bildschirm)
- Breite bestimmt Anzahl Zeichen in der Zeile

Der durchschlagende Erfolg von Internet, WWW und der modernen digitalen Kommunikationstechnologien hat einen neuen Wirtschaftszweig rund um Kommunikation, Information und elektronischen Handel (E-Commerce) geschaffen. Diese neue Industrie hat sehr hohen Bedarf an einem neuen Typus Ingenieur, dessen Spezialität Softwareanwendungen in Netzwerken (Network Computing) sind. Die starke betriebswirtschaftliche Orientierung eines solchen Fachmanns ist eine weitere unverzichtbare Oualifikation.

• Absatz

- logische Struktur
- prinzipiell beliebig lange Zeichenkette
- Objekt vieler Funktionen der Verarbeitung

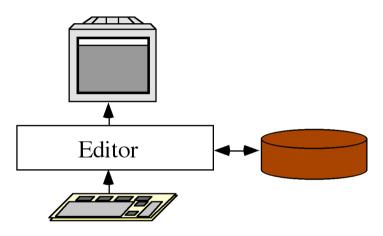
Der durchschlagende Erfolg von Internet, WWW und der modernen digitalen Kommunikationstechnologien hat einen neuen Wirtschaftszweig rund um Kommunikation, Information und elektronischen Handel (E-Commerce) geschaffen. Diese neue Industrie hat sehr hohen Bedarf an einem neuen Typus Ingenieur, dessen Spezialität Softwareanwendungen in Netzwerken (Network Computing) sind. Die starke betriebswirtschaftliche Orientierung eines solchen Fachmanns ist eine weitere unverzichtbare Qualifikation.

Die Fakultät für Mathematik und Informatik der Technischen Universität - Bergakademie Freiberg hat als erste deutsche Universität einen Bachelor-Studiengang, der genau solche Ingenieure schnell und gründlich ausbildet, entwickelt. Wir hoffen, so unseren Studenten einen sicheren und zukunftsorientierten Beruf zu eröffnen.

• Umbruch

- Abbildung von Absätzen auf Zeilen
- Abbildung der Zeilenmenge auf Seiten
- Zeilenwechsel <> Absatzwechsel

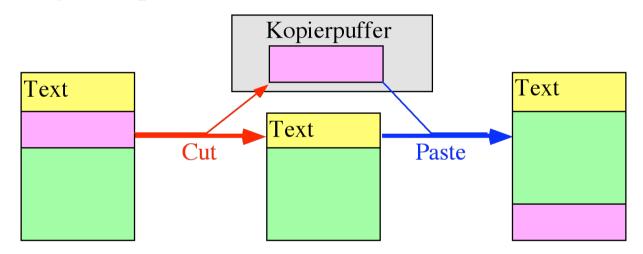
- Editor: einfaches Textverarbeitungsprogramm
 - Zeichen eintippen
 - Zeichenkette anzeigen
 - Cursor: Bearbeitungsposition: einfügen, löschen
 - Cursor positionieren mit Spezialtasten oder Maus
 - Speichern, Drucken, ...



Markieren

- Cursor setzen, Umschalttaste + Cursor neu setzen
- Markieranfangs-Zeichen, Cursor bewegen, Markierende-Zeichen
- z.B. Wordstar Cntl-K + Cntl-B
- evtl. mit Maus

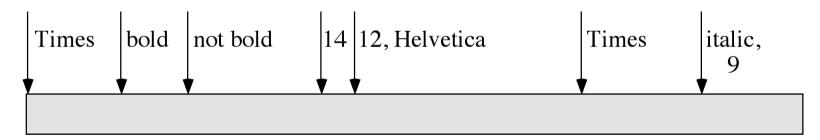
- Kopierpuffer
 - Text markieren
 - Kommando: ausschneiden oder kopieren
 - Zielposition setzen
 - Kommando: einsetzen
 - Zwischenablage, Clipboard



- Einfache Satzmethoden
 - Tabulator
 - Leertaste (Space, Blank)
 - aber Vorsicht!

8.2.1.2 Formatierung

- Satzregeln aus der Buchdruckkunst
 - Ziel: optimale Lesbarkeit
 - "Schönheit"
 - widows and orphans
 - unterstreichen ist unschön
 - kein Fontsalat
 - ...
- Text bekommt Attribute



- Text immer noch sequentiell
 - Attribute gelten für alle folgenden Zeichen
 - bis anderes Attribut der selben Kategorie

- Buchstabenbild
 - fett, kursiv, fett+kursiv
 - unterstrichen, outline, schattiert, ...
 - hoch^{gestellt}, sub_{script}
 - Farbe, ...
- Zeichensatz (Font)
 - monospace
 - proportional
 - Serifen: Times, Antiqua, ...
 - sans-serif: Helvetica, Frutiger, ...
 - Σψμβολ
- Markieren
 - Cursor setzen, Umschalttaste + Cursor neu setzen
 - Cursor setzen, Maustaste gedrückt lassen+Maus bewegen
 - Wort: Doppelclick
 - Absatz: Dreifachclick

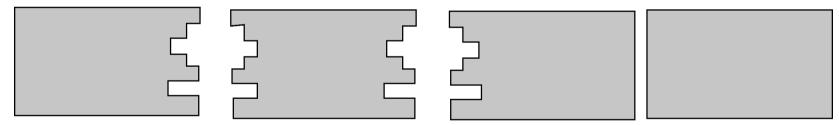
- Zeichenformat bezieht sich auf einzelne Zeichen
 - Zeichenmenge auswählen (=> markieren)
 - ohne Auswahl: nächste getippte Zeichen

Der durchschlagende Erfolg von *Internet*, WWW und der modernen digitalen <u>Kommunikationstechnologien</u> hat einen neuen <u>Wirtschaftszweig</u> rund um Kommunikation, Information und elektronischen <u>Handel</u> (E-Commerce) geschaffen. Diese neue Industrie hat sehr hohen Bedarf an einem neuen Typus Ingenieur, dessen Spezialität Softwareanwendungen in Netzwerken (Network

Computing) sind. Die starke betriebswirtschaftliche Orientierung eines solchen Fachmanns ist eine weitere unverzichtbare Qualifikation.

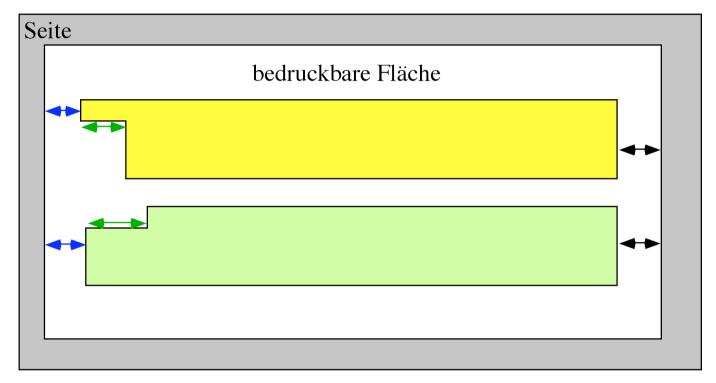
- Dialogbox Zeichenformat
 - einstellen und ausprobieren
- Kerning und Ligaturen
- Spezialzeichen
 - Währungssymbole
 - Länderbuchstaben, ©®™ §¶""
 - Dialog im Menü Einfügen
- Tabulatoren: horizontale Marken
 - stabile Formatierung
 - Sprung zur Marke
 - linksbündig, rechtsbündig, zentiert, Zahl, Währung

- Absatz
 - inhaltlich eng zusammengehörender Text
 - semantische Einheit
 - von anderen Absätzen abgesetzt
- Absatzformat
 - automatische Absatzformatierung siehe Formatvorlage (Stile)
 - Dialogbox Absatz
- Form des Absatzes
 - Flattersatz: linksbündig, rechtbündig, zentriert



- Blocksatz
- Absatzzeilen zusammenhalten
- Mit nächstem Absatz zusammenhalten

- Einzug = Abstand vom horizontalen Rand
 - links, rechts
 - hängend (= Sonderfall erste Zeile)



- Zeilenabstand (Durchschuß, leading)
- Abstand zu andern Absätzen
 - vorher, hinterher
 - Sonderfall oberer Rand bzw. unterer Rand
 - Leerzeilen sind "schlechter Stil" bzw. gefährlich

• Maßeinheiten

- 1 Inch = 2,54 Zentimeter
- 1 Zeile = 1 Pica = 1/6 Inch
- -1 Punkt = 1 pt = 0.035 cm
- 12 Punkt = 1 Zeile = 1/6 Inch

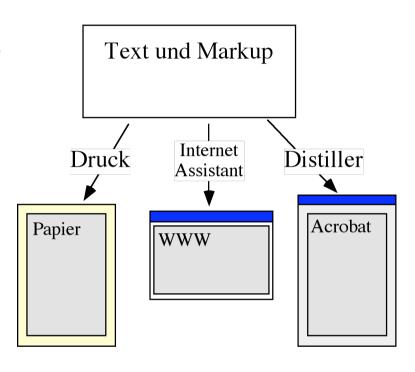
14

• Seitenmaße

- bedruckbare Fläche kleiner als Papier => Satzspiegel
- Abstände: links, rechts, oben, unten
- Bücher: gespiegelt für gerade und ungerade Seiten
- besondere Seitenbereiche: Kopfzeilen, Fußzeilen
- Seitennummern

8.2.1.3 Text und Struktur

- Markup
 - logische Struktur für Text
 - Überschrift, normaler Paragraph, Zitat, ...
 - Fußnote, Literaturverweis, Bildunterschrift, ...
- Zuordnung der Attribute beim Satz
 - Autor produziert Inhalt und Struktur
 - Drucker setzt
- Tex
- Formatvorlagen (Stile, Styles)
 - Mehr als 2 Seiten => Formatvorlagen
 - Menge von Attributen für Zeichen
 - Menge von Attributen für Absatz
 - Analogie zur Programmiersprache: Prozeduren für Satz
- Besondere Formatvorlagen
 - Überschriften => automatisches Inhaltsverzeichnis
 - Abbildungsverzeichnis, ...

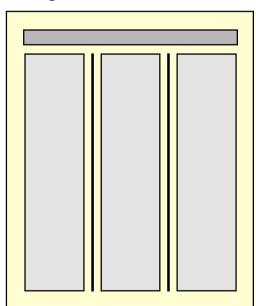


8.2.1.4 Spezialfunktionen für Text

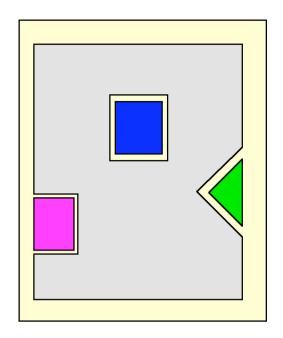
- Trennung
 - manuell mit Sonderzeichen oder am Bindestrich
 - automatische Silbentrennung
 - semi-manuell
- Rechtschreibprüfung
 - manuell oder automatisch
 - Wörterbücher für Sprachen
 - neues Textattribut: Sprache
 - persönliche Wörterbücher
- Grammatik
 - einfache Regeln
 - stereotype Ratschläge
 - funktioniert in Englisch besser
- Thesaurus
 - Synonyme und Antinyme finden
- Zählfunktionen: Wörter, Buchstaben, ...
- Tabellen, Listen, grafische Rahmen, ...

8.2.2 Dokumente

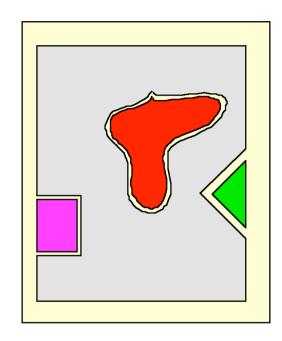
- Dokumentenverarbeitung
 - Word, Wordperfect, (StarlOpen)Office
 - zusätzliche Funktionen zur Integration
- Komponenten
 - Text
 - Grafik
 - Bild
 - Objekte von anderen Programmen



- Mischen der Komponenten
- Textfluß
- Objekt läuft mit Text
- Objekt fixiert, Text läuft herum
- Spalten
- Positionsrahmen
- Dynamisches Einbetten von Obkjekten (OLE)

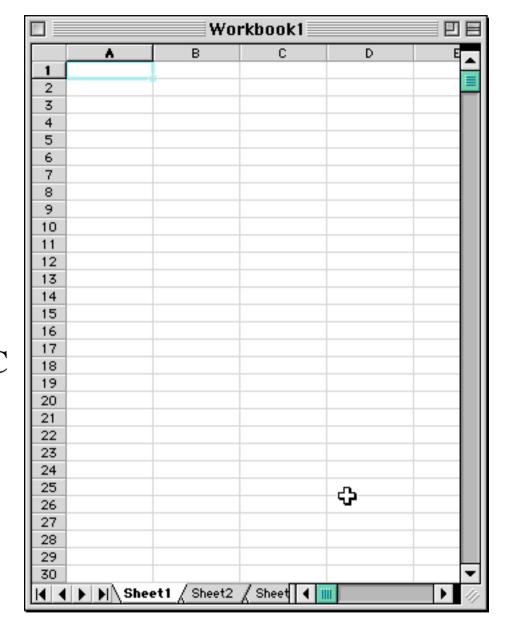


- Besondere Programme für Computersatz
 - InDesign, Quark Xpress
 - Schnittstelle zu Druckmaschinen (Farbseparation, ...)
- Seitenorientiert
- Textfluß zentrales Element
 - präzise Angabe der Textfläche
 - Text fließt um Objekte herum
- Objekte
 - Position auf der Seite fest
 - beliebige Kontur der Objekte
 - Text folgt der Kontur
- Viele weitere Eigenschaften
 - Inhaltsverwaltung, Bearbeitungshilfen
 - Farbkontrolle
 - hairlines ...



8.2.3 Tabellenkalkulation

- Berechnungen
 - Zahlenkolonnen addieren
 - Angebote: (10 * 1,53 + 12*12,5 +...+15,73)*1,16
 - einfache Modelle: Gehalt berechnen, Steuerformel, ...
 - Kontostand projezieren
- Visicalc [Bricklin, 1978] für Apple II
- Lotus 1-2-3 [Kapor, 1982] für IBM-PC
- Aktuell: Excel, Lotus, Quattro
- Grundidee: Rechenblatt
 - Spreadsheet
 - Zeilen und Spalten
 - Formeln
 - Variablen
 - einfache Programmierkonstrukte



8.2.2.1 Rechenblatt

- Felder
 - enhalten ein 'Datum'
 - Typ
 - Adresse = (Spalte, Zeile)

Datentypen

- Zeichenketten
- Zahlen
- Währung
- Datum, Zeit
- Postleitzahl, Telefonnummer, Social Security Number

Debitor

Müller

Meier

Schmidt

Hartmann

Meier-Möllend

Rechnung vom Betrag

1713,12

1800,67

1190.78

5312.5

782

12-Jan-99

13-Jan-99

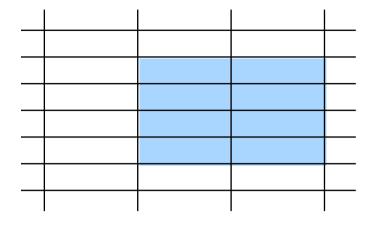
17-Jan-99

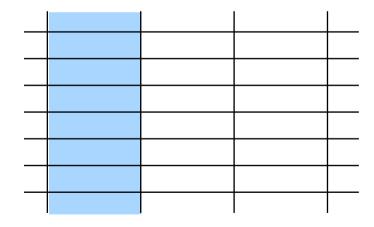
03-Feb-99

09-Feb-99

- Formatieren
 - Zeichenattribute ähnlich Textprogramm
 - Datentyp-Formate (Währung, Zeit, ...)
 - Dezimalstellen
 - -0.13 = 13%
- Positionieren mit Maus oder Cursortasten

- Eingabe
 - Zelle selektieren + tippen
 - Eingabezeile
 - automatische Formatierung
- Eingabe größerer Datenmengen
 - Return => selbe Spalte, Zeile n+1
 - Tab => selbe Zeile, Spalte n+1
- Selektion
 - Zelle
 - Zellenrechteck mit Maus oder Shift-Klick
 - Zellmenge mit Cntl-Taste
 - Spalte
 - Zeile
- Editieren
 - Cut, Copy, Paste
 - Füllen nach links, rechts, oben, unten
 - Füllen mit Werten oder Datum



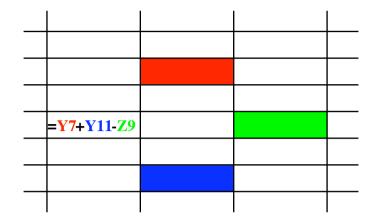


- Einfügen und Löschen
 - Zelle oder Zellgruppe
 - Selektion + Einfügen/Löschen
 - Zellen verrücken
 - Zeilen/Spalten einfügen -> verschieben
 - Zeilen/Spalten löschen -> verschieben
 - Achtung: Bezüge können leiden!
- Drag und Drop
- Viele Editiertricks
 - Excel 'rät' die gewünschte Aktion
- Layout
 - Zeilenbreite
 - Spaltenhöhe
 - Zellen: Zeichenattribute
 - Zellen: bedingte Formatierung
 - Hintergrund, Rand, ...
 - Formatvorlagen

8.2.2.2 Formeln

- Elemente
 - Konstante
 - Zellinhalte
 - Operatoren
 - Funktionen
- Arithmetik
 - -+,-,*,/
 - Punkt-vor-Strich, Klammern
 - auch mit Datum rechnen
- Bezüge
 - Tupel (Spalte, Zeile)
 - ähnlich Adressen in der Programmiersprache
 - auch Zellbereiche A1:B6 (Rechteck!)
- Konstante
 - Zahlen, Datum, Prozent
 - Text

- Relative Bezüge
 - Normalfall, automatisch
 - (Spalte-n, Zeile-n)
 - im neuen Excel absolut angezeigt
 - A1, B17, AA123, ...
- Absolute Bezüge
 - Ausnahme
 - Adresse (Spalte, Zeile) fest
 - Kennzeichen in neueren Excels: \$
 - \$A\$3
 - absolute Zeilenadresse: A\$3
 - absolute Spaltenadresse \$A3
- Bezüge und Editieren
 - automatische Anpassung relativer Bezüge
 - beim Kopieren, nicht beim Verschieben
 - auch beim Einfügen/Löschen 'dazwischen'
 - auch bei Bereich füllen



- Funktionen
 - <name>(<argument1>;<argument2>;...;<argumentn>)
 - Summe(), Max(), Min(), Anzahl(), ...
 - Zellmenge ist ein Argument: Summe(A3:B10)
 - Schachtelung möglich
- Funktionsgebiete
 - Mathematik, Trigonometrie, Statistik
 - Finanzmathematik (Annuität, ...)
 - Datum und Zeit, Text
 - Logische Funktionen, Feldinformation
- WENN(Bedingung;<then-val>;<else-val>)
 - WENN(B17<>0;B16/B17;0)
 - Bedingung mit UND(), ODER(), NICHT()
- Funktionsassistent
 - kennt Funktionsnamen
 - hilft bei Argumenten

8.2.2.3 Namen

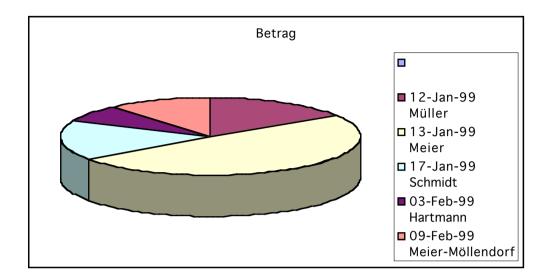
- Tabellenkalkulation ist Programm!
 - dokumentieren der Formeln
 - Sicherheit bei späteren Änderungen
- Zellbezüge unübersichtlich
 - Bezug auf Prozentsatz etc.?
 - dokumentiert nicht
- Zellen können benannt werden
 - einzeln oder Bereich
 - Namensregeln ähnlich Programmiersprachen
 - automatischer Namensvorschlag
 - benannte absolute Bezüge
 - Indirektion: Name -> Zelle
- Namen in Formeln einsetzen
 - verändern sich nicht beim Kopieren
 - Auswahl aus Bereich

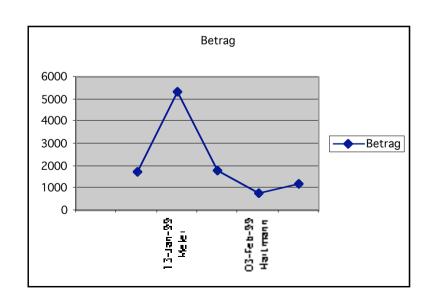
8.2.2.4 Diagramme

- Visualisierung einer Tabelle
 - Anteile
 - Trends
- Tabelle selektieren
 - Achsen
 - Werte
- Diagramm-Art wählen
 - Kuchen
 - Linie, Punkte
 - Pseudo 3D

Rechnung vom	Betrag
12-Jan-99	1713,12
13-Jan-99	5312,5
17-Jan-99	1800,67
03-Feb-99	782
09-Feb-99	1190,78
	12-Jan-99 13-Jan-99 17-Jan-99 03-Feb-99

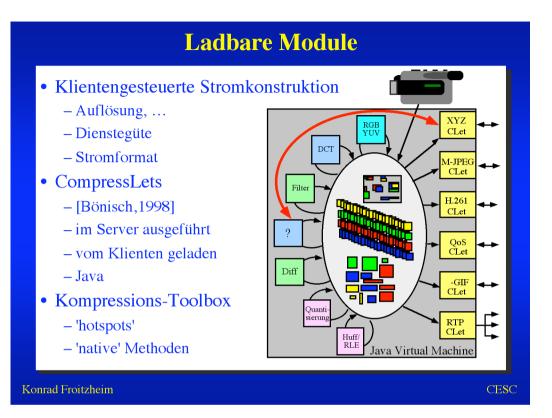
- Formatieren
 - Achsen, Beschriftung
 - Legende



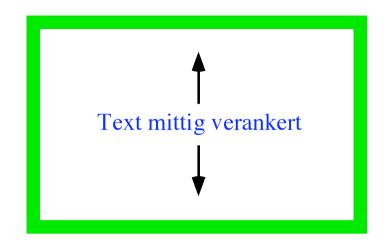


8.2.4 Präsentationen und Vorträge

- Elektronische Folien
 - Text (Punkte)
 - Grafiken
 - Animation
- Didaktische Aufbereitung
 - sequentielle Präsentation
 - Hauptpunkte Details
 - Präsentationskontrolle
- Dramaturgie
 - Übergänge zwischen Kapiteln
 - Einzelpunkte animieren
 - Überblendungen
- Powerpoint, Impact, Keynote, ...



- Textformat
 - vertikaler Text-Anker
 - Zeichenformate wie gewohnt
- Zeichenprogramm eingebaut
 - fertige Elemente
 - Clip-Art
 - geometrische Grafik
- Animierte Grafik
 - Komponenten einblenden lassen
 - Art der Einblendung wählbar
 - Anlaß wählbar
 - Geräusch möglich
 - Bewegung = ausblenden, verschoben einblenden

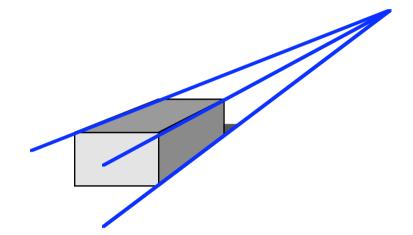


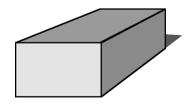
- Entwurfshilfen
 - Grunddesign mit 'Wizard'
 - Outline mit Überschriften, Hauptpunkten, Unterpunkten
 - Einzelfolie
 - Notiz-Seiten
 - 'Lichtkasten' zum Sortieren
- Ausgabeoptionen
 - elektronische (Präsentation, Arbeitsplatz, Kiosk)
 - Folien
 - Papier: 'Handouts', Notiz-Seiten
 - WWW leider nur Bitmap
- Präsentationshilfen
 - Generalprobe mit Zeitnahme
 - zeitgesteuerte Präsentation

8.2.5 Grafik

8.2.5.1 Objektgrafik

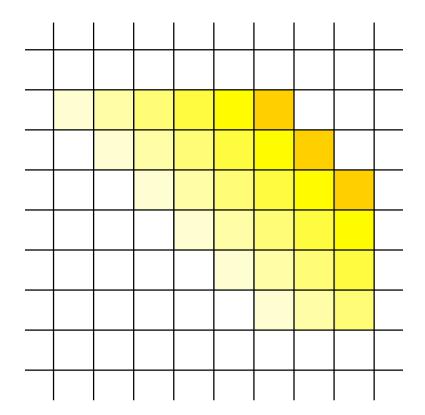
- MacDraw, Corel, Canvas, Illustrator...
- Graphische Objekte
 - Strecken, Linienzüge
 - Rechtecke, Polygone
 - Kreise, Ellipsen
- Attribute
 - Farbe, Muster, Stiftform, ...
 - Linien, Fläche
- Anordnung der Objekte
 - zusätzliche geometrische Figuren
 - Verdecken von Hilfskonstruktionen
- Pseudo-3D
 - Perspektive
 - Schattenwurf





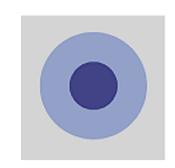
8.2.5.2 Pixelgrafik

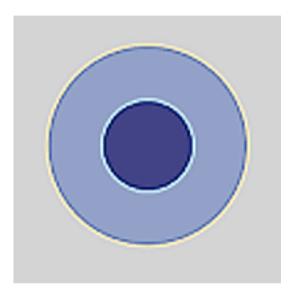
- MacPaint, PaintShop
- Photos und Bilder
 - Pixel
 - 'glatte' Übergänge
 - Farben
- Werkzeuge
 - Stift: Punkte setzen
 - Radiergummi: Punkte entfernen
 - Farbeimer: ausfüllen
 - Spraydose
- Attribute
 - Farbe, Muster, Stiftform, ...
 - Linien, Fläche



- Photoshop
- Bildverbesserung
 - Retouching
 - Kontrast, Helligkeit, Farbsättigung, Histogramm-Equalization
 - Kanten bearbeiten (Schärfe, ...)
- Selektion
 - Rechteck, Ellipse und Lasso
 - 'magnetisches' Lasso
 - Zauberstab
 - farbgesteuert
- Filter
 - Störungen entfernen (z.B. Durchschnitt mit Nachbarn)
 - Effekte
- Zeichenwerkzeuge
 - Stift, Pinsel, Farbeimer, Radiergummi

- Farbkalibrierung
- Digitalisierte Bilder sind verfälscht
 - CCD-Rauschen, unterschiedlich stark in den Farbkanälen
 - Helligkeit und Kontrast
 - Farben schlecht (Farbtemperatur des Lichtes ...)
- Rauschen entfernen
 - Lab Farbraum
 - blauer Kanal besonders verrauscht
- Nachschärfen
- Kurven (Helligkeit, Kontrast, Farbkanäle)
 - Histogramme
 - Farbkorrektur
- Effekte
 - als Filter implementiert
 - Verwischen (blur, PSF), Kanten schärfen, ...
- Kai's Powertools
 - lokale und globale Transformationen





0.01	0.1	0.25	0.1	0.01
0.1	0.35	0.5	0.35	0.1
0.25	0.5	1.0	0.5	0.25
0.1	0.35	0.5	0.35	0.1
0.01	0.1	0.25	0.1	0.01

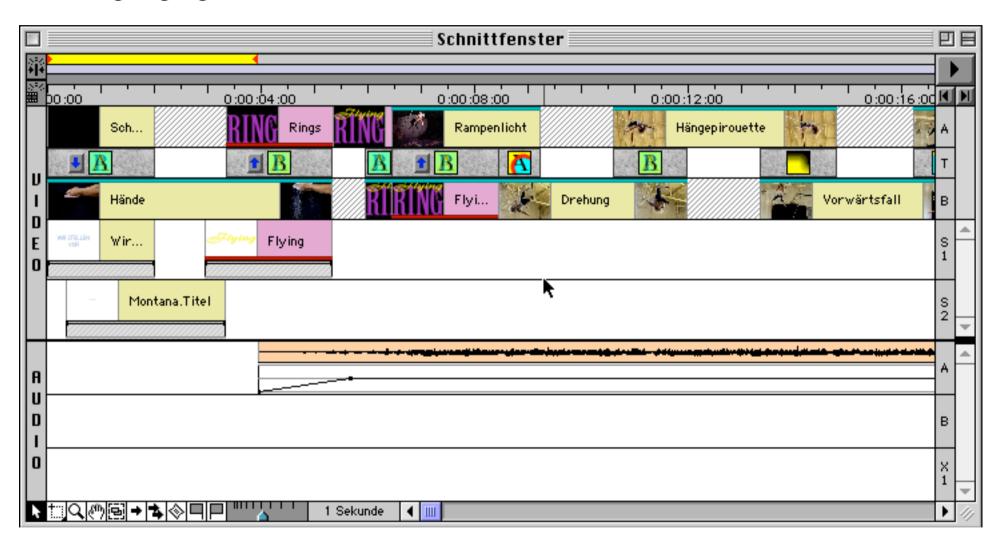
- Räumliche Anordnung der Bildelemente
 - Layer
 - Organisation des Bildes
 - wichtige Elemente in Layern
 - Transparenz
 - arithmetische Verknüpfung
- Druck-Ausgabe
 - Farbanpassung an Drucker
 - Farbseparationen



8.2.6 Digitaler Videoschnitt

- Adobe Premiere, Apple Final Cut Pro
- Projekt
 - technische Parameter
 - Menge von Komponenten: Audio, Video, Titel
 - Digitalisierung
 - Import
- Medienströme bearbeiten
 - editieren
 - Präsentationseigenschaften ändern
 - Filter
 - einfache Effekte: Bewegung, Zoom, Drehung
- Produktion des Filmes
 - Movie-Datei in vielen Formaten
 - Edit Decision List für Schnittsysteme

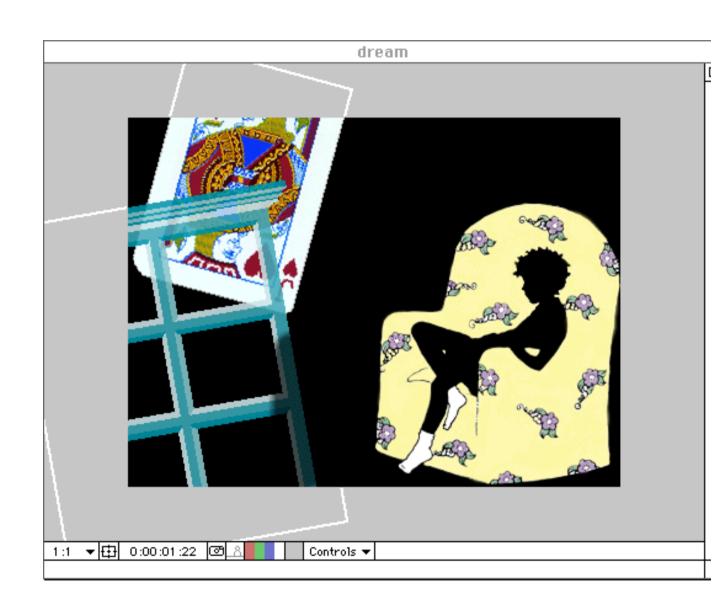
- Mehrere Spuren
 - zeitliche Anordnung
 - Übergänge gestalten



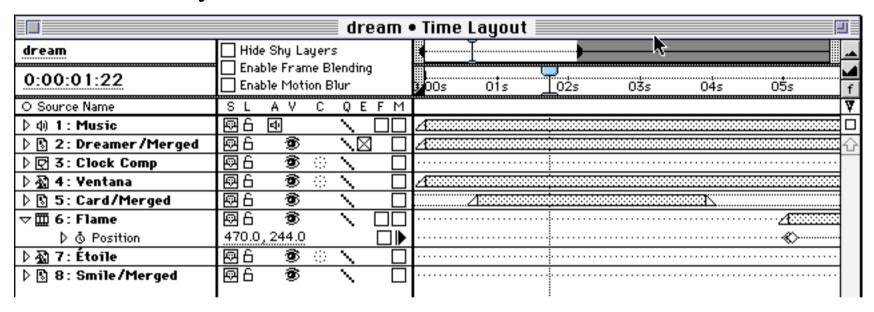
8.3 Kombination (After Effects, Director)

8.3.1 After Effects

- Bilder lernen Laufen
 - Animationen
 - Zeichnungen, Text
- Filmformat
 - Quicktime
 - GIF
 - ...
- Objekte anordnen
- Änderungen in der Zeit
 - Eigenschaften
 - Anordnung
- f(Objekt, t)



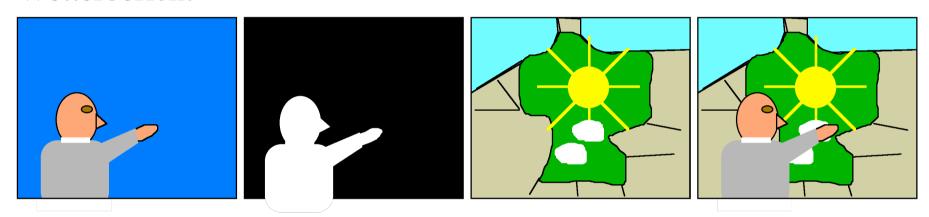
- Schichten (Layer) überlagern
 - Darstellung zeitlich begrenzt (In, Out)
 - Maske
- Zeitachsen und Keyframes



- Keyframes haben Schichteigenschaften als Attribute
 - Übergang zwischen Keyframes
 - => gradueller Übergang von A1 nach A2
 - lineare oder komplexe Übergangsfunktion

- Geometrische Attribute der Schichten in der Zeit
 - Größe
 - Bewegung entlang eines Pfades (Gerade, Bezier-Spline)
 - Rotation, Ankerpunkt
 - Bewegungsverzerrung
 - Durchsichtigkeit (Ein- und Ausblenden)
- Farb-Attribute
- Filter in der Zeit
 - Verschwimmen
 - Schatten
 - Textur
 - Erhebung, 3D, ...
- Überblenden zwischen Schichten
 - Transparenz
 - Überblendeffekte (Dissolve, Wipe, Vorhang, ...)

- Motion Pack
 - Identifizieren bewegter Elemente
 - Bewegungen glätten
 - Bewegungen 2. Ordnung
- Keying Pack
 - Wetterbericht



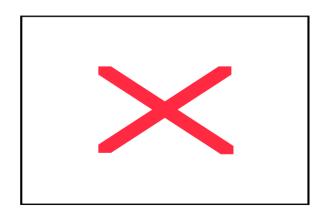
- Teile des Orginalbildes werden 'transparent'
- einfache Keys: Farbe, Helligkeit
- scharfe Grenzen
- Wertebereich der Transparenz
- Verzerrung von Objekten
 - perspektivisch
 - wellenförmig, Polarkoordinaten, Wirbel, ...

• La Reine des Neiges



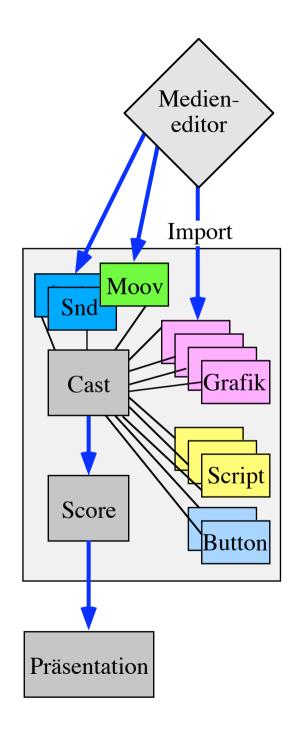


- Film mit Kind und gezeichneter Raum
- Chroma-Key, Bewegungsfilter
- Komposition

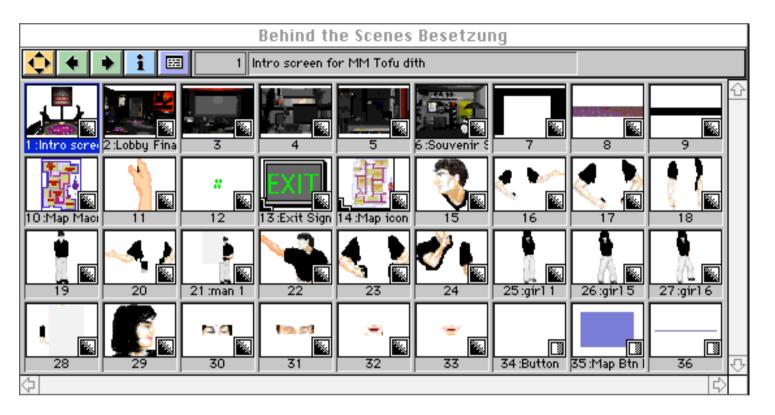


8.3.2 MacroMedia Director

- Präsentationen
 - Animationen (nicht extrem anspruchsvoll)
 - Abspielen von Video und Sound
 - ShockWave
- Grafische Elemente
 - Text, Grafik, Video, Sound
 - 16/32 bit Farben
 - 1,2,4,8 bit Farben mit verschiedenen Paletten
- Benutzungsschnittstelle
 - simple Navigation mit Knöpfen
- Begriffe
 - Darsteller (actors) und Besetzung (cast)
 - Drehbuch (score)
 - Programmiersprache Lingo (scripts)
 - Sprite Behälter für actor

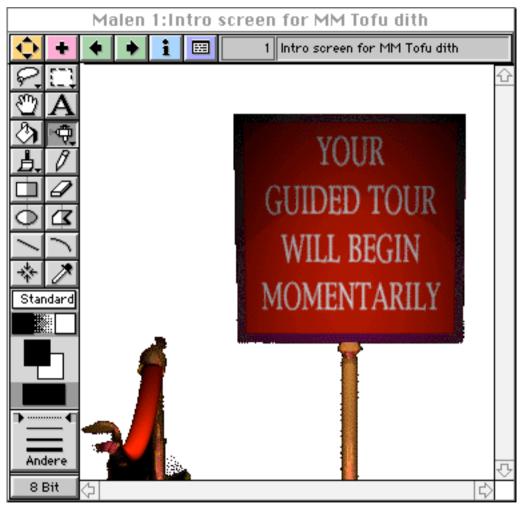


• Darsteller (actors) und Besetzung (cast)

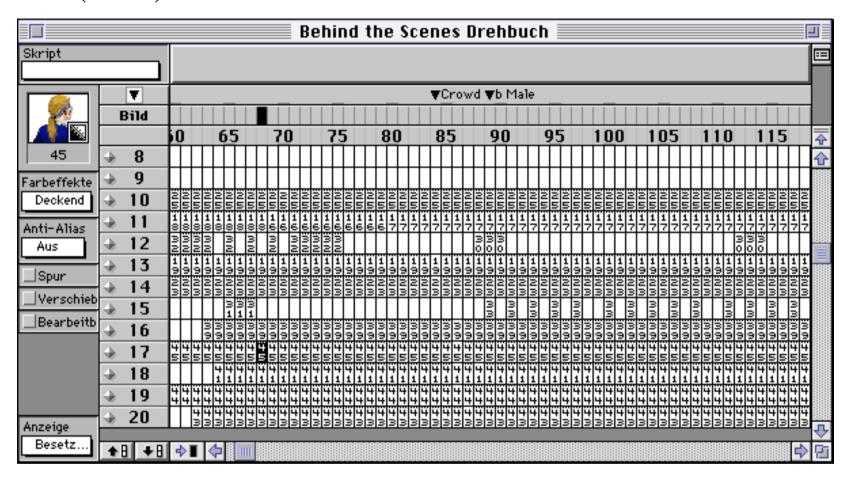


- Grafische Objekte (PICT, PICT-Sequenz, Bitmap)
- QuickTime Movie
- Sound
- Button, Menu, ...
- Script

- Zeichenwerkzeug integriert
 - Nachbearbeitung von importierten Bildern
 - Grafiken können auch Skripts haben



• Drehbuch (score)



- Zeit horizontal (Frames)
- vertikal 'Spuren'
- Nummern aus cast

- Programmiersprache Lingo (scripts)
 - erweitertes HyperTalk
 - Objektorientierung als Nachgedanke
- Basis-Lingo
 - on <Event> ... end <Event>
 - set <Variable> of <Objekt> to <Wert>
 - set <Variable> = <Wert>
 - if <condition> then <clause> else <clause> end if
 - go <Identifier>
 - repeat with <Variable>=<Wert> to <Wert>

• • •

end repeat

- <Identifier> ruft Prozedur
- globale Variable
- Handler
 - mouseUp, mouseDown, startMovie, stopMovie
 - selbstdefiniert

```
on mouseUp

go to frame "start"
```

end mouseUp

- Objektorientierung
 - Darstellerscript = Klasse
 - property-variable = Instanzvariable
 - ancestor-variable = Inheritance
 - birth ist Konstruktor (on birth)

```
MECH , 76
global gMyPegBoard, gConstrainSprite, gIsRunning, gHandCursor, gSpeedCntrSprite
property PEG columns, PEG rows, myHPegs, myVPegs, myAnimator, myButtons
on birth me
    set PEG columns = 16
                                           -- we have 16 columns of pegs
    -- build the vert peg list
    set myVPeqs = []
                                           -- empty list
    set count = 2
    setAt myVPegs, 1,50
                                           -- set first array elt
    repeat while count < PEG rows
    set pos = 50 + ((count - 1) * 30)
    setAT myVPeqs , count, pos
    set count = count + 1
    end repeat
    return me
end birth
```