

Inhaltsverzeichnis

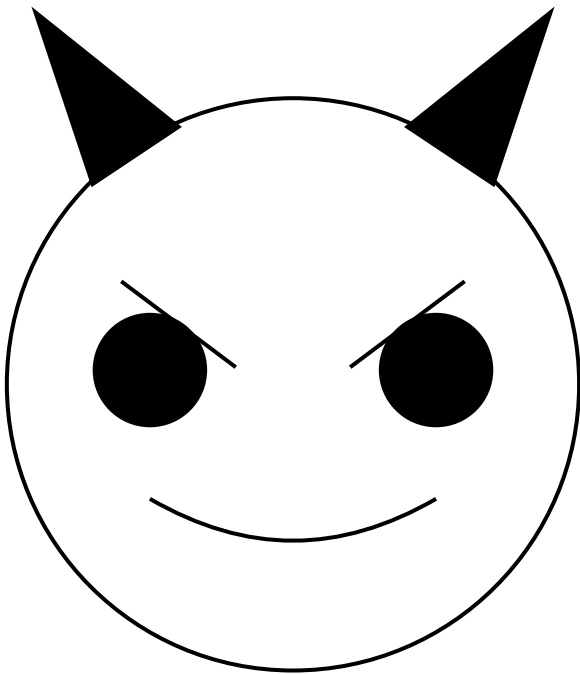
Allerlei	2
Teufel	2
Automat	2
AutomatDFA	2
AutomatNFA	3
CYKAlgorithmus	3
Demo-1	3
Demo-2	3
Demo	4
Header	4
MealyAutomat	4
MinimalautomatBeispiel/MinimalautomatBeispiel1	4
MinimalautomatBeispiel/MinimalautomatBeispiel2	5
MinimalautomatBeispiel/MinimalautomatBeispiel3	5
MinimalautomatBeispiel/MinimalautomatBeispiel4	6
MinimalautomatBeispiel/MinimalautomatBeispiel5	6
MinimalautomatBeispiel/MinimalautomatBeispiel6	7
MooreAutomat	7
Datenbanken	7
ERMExample	7
Eigene	8
Proseminar/Cluster/en-circles	8
Proseminar/Cluster/en-clusters	8
Proseminar/Cluster/en-moons	8
Proseminar/Cluster/en-special	8
Proseminar/Cluster/km-circles	9
Proseminar/Cluster/km-clusters	9
Proseminar/Cluster/km-moons	9
Proseminar/Cluster/km-special	9
Proseminar/Cluster/kn-circles	10
Proseminar/Cluster/kn-clusters	10
Proseminar/Cluster/kn-moons	10
Proseminar/Cluster/kn-special	10
Proseminar/Cluster/rolf-circles	11

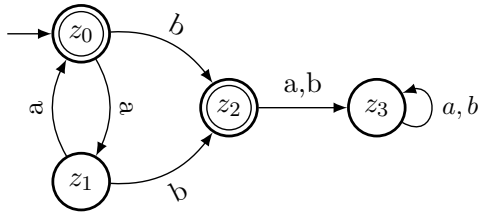
Proseminar/Cluster/rolf-clusters	11
Proseminar/Cluster/rolf-moons	11
Proseminar/Cluster/rolf-special	11
Proseminar/Cluster/thumb-circles	12
Proseminar/Cluster/thumb-clusters	12
Proseminar/Cluster/thumb-moons	12
Proseminar/Cluster/thumb-special	12
Graphen	13
GraphNachbarschaftGrad	13
GraphNichtPlanarK33	13
GraphNichtPlanarK5	13
GraphTopologie	14
GraphWegPfad	14
GraphZyklus	14
Haskell	14
HaskellTypen	14
Listenoperationen	15
Java	15
ODBCSchematisch	15
StreamDemo	15
Logik	15
KVDiagramm	15
KVVuerfel	16
QuineMCCluskeyTabelle	16
QuineMCCluskeyZusammenfassen	16
Mengen	17
FunktionBijektiv	17
FunktionInjektiv	17
FunktionSurjektiv	17
Mengenmultiplikation/Mengenmultiplikation1	18
Mengenmultiplikation/Mengenmultiplikation2	18
Mengenmultiplikation/Mengenmultiplikation3	18
Mengenmultiplikation/Mengenmultiplikation4	18
VennDifferenz	18
VennSchnitt	19
VennVereinigung	19
Prozesse	19
FCFS-WorstCase	19
FCFS	19
Prozesszustaende	19
Rechner	20
ALU	20

AmpelPLA	20
BarrelShifter	20
Beispielprozessor	20
CLA	20
CPLD	21
CSA	21
DreiTorRegister	21
Eintorspeicher	22
GALPAL	22
Geraeteverwaltung	22
HardwareSkizze	22
ISA	23
LUT	23
LUTOder	23
MIPS	23
MuxDemuxKommunikation	24
MuxShannon	24
NAdressmaschine	24
PLA	24
PLAAmpel	25
PROM	25
Pages	25
Physik/DiodenStromstaerke	26
Physik/Metastabil	26
Physik/TransistorStoertoleranz	26
RegisterParallel	27
RegisterSeriell	27
Shiftregister	27
Speicherhierarchie	27
StackExample	28
Stackmaschine	28
StackmaschineSimpler	28
Schaltkreis	28
Addier-Subtrahierer	28
Demultiplexer	28
KomplexerSchaltkreis	29
SynchronzaehlerDFF	29
SynchronzaehlerTFF	29
Volladdierer	29
Software	30
DreiSchichtenArchitektur	30
Meta/ProgrammierparadigmenUeberblick	30
ModelViewController	30
RegexExample	30
SQL/SQLFields	31
SQL/SQLFieldsDCL	31

SQL/SQLFieldsDDL	32
SQL/SQLFieldsDML	32
SQL/SQLFieldsDQL	33
ThreadStates	33
UML/UMLCompositePattern	33
UML/UMLDecoratorPattern	34
UML/UMLExample	34
UML/UMLFactoryPattern	34
UML/UMLObserverPattern	34
UML/UMLSEQObserverPattern	35
UML/UMLSEQObserverPatternAdapted	35
UML/UMLStateDiagramExample	35
UML/UMLThread	35
XML/XMLUebersicht	36
Sprachen	36
CYKAlgorithmus	36
ChomskyHierarchie	36
Grammatik	36

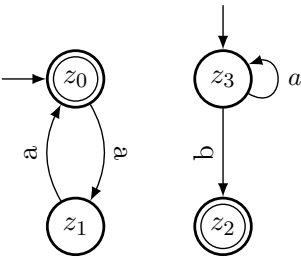
```
%%Einbindung erfolgt über:
\getGraphics{\Pfad}
```

Pfad	Ergebnis
Allerlei/Teufel	

Automat/AutomatDFA	<p>Deterministic Finite Automaton</p>  $D = \left(\begin{array}{ccccc} \text{Zustände} & \text{Eingabe} & \text{Übergangsfkt.} & \text{Start(s)} & \text{Endzust.} \\ Z & \Sigma & \delta & S & E \\ \text{Menge} & \text{Alphabet} & Z \times \Sigma \rightarrow Z & \in Z & \subseteq Z \end{array} \right)$ $T(D) = \{x \in \Sigma^* \mid \hat{\delta}(S, x) \cap E \neq \emptyset\}$
--------------------	---

Nondeterministic Finite Automaton

Automat/AutomatNFA

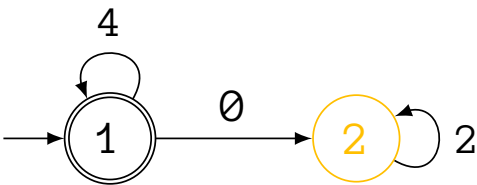


$$N = \left(\begin{matrix} \text{Zustände} & \text{Eingabe} & \text{Übergangsfkt.} & \text{Start(s)} & \text{Endzust.} \\ Z & \Sigma & \delta & S & E \\ \text{Menge} & \text{Alphabet} & Z \times \Sigma \rightarrow Z^* & \subseteq Z & \subseteq Z \end{matrix} \right)$$
$$T(N) = \{x \in \Sigma^* \mid \hat{\delta}(S, x) \cap E \neq \emptyset\}$$

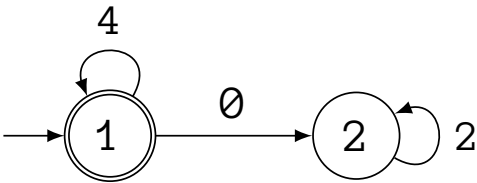
Automat/CYKAlgorithmus

		1 a	2 b	3 c	4 c
T[1,j]	1	A	B	CE	CE
T[2,j]	2	D	—	E	
T[3,j]	3	S	—		
T[4,j]	4	S			

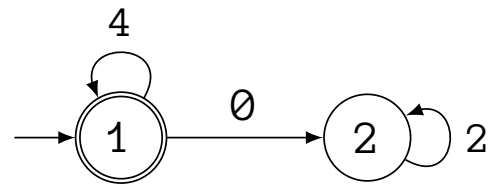
Automat/Demo-1



Automat/Demo-2



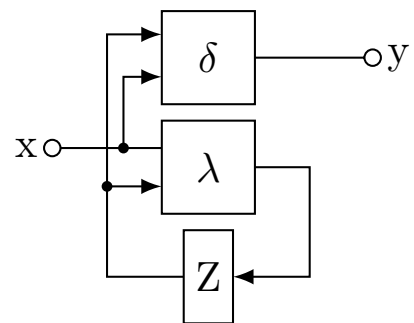
Automat/Demo



Automat/Header

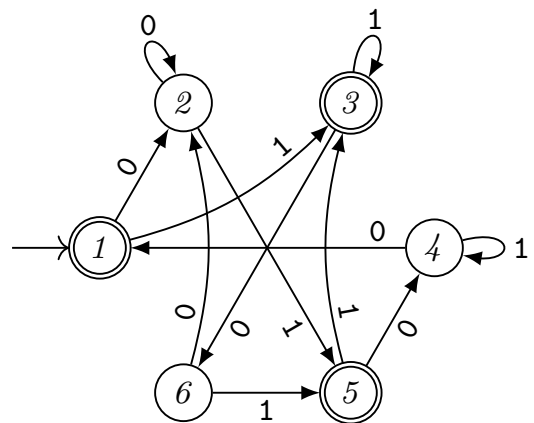
Isch bin a Hädder!

Automat/MealyAutomat



Ausgabe von Zustand & Eingabe abhängig

Automat/MinimalautomatBeispiel/MinimalautomatBeispiel1



Automat/MinimalautomatBeispiel/MinimalautomatBeispiel2

2					
3					
4					
5					
6					
	1	2	3	4	5

Automat/MinimalautomatBeispiel/MinimalautomatBeispiel3

2	0				
3		0			
4	0		0		
5		0		0	
6	0		0		0
	1	2	3	4	5

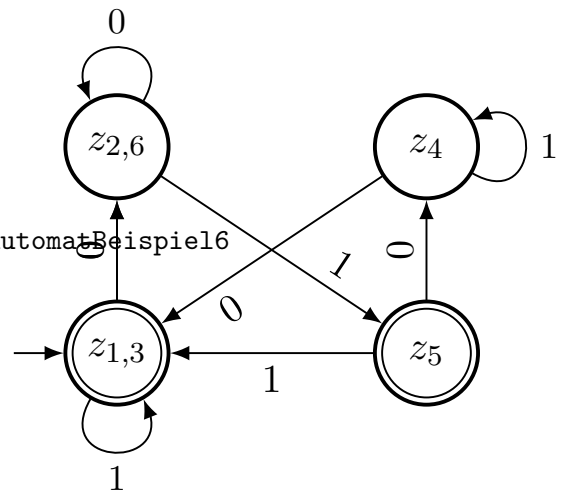
Automat/MinimalautomatBeispiel/MinimalautomatBeispiel4

2	0				
3		0			
4	0	1	0		
5		0		0	
6	0		0	1	0
	1	2	3	4	5

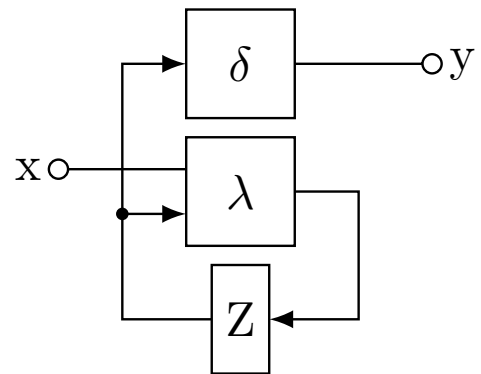
Automat/MinimalautomatBeispiel/MinimalautomatBeispiel5

2	0				
3		0			
4	0	1	0		
5	2	0	2	0	
6	0		0	1	0
	1	2	3	4	5

Automat/MinimalautomatBeispiel/MinimalautomatBeispiel6

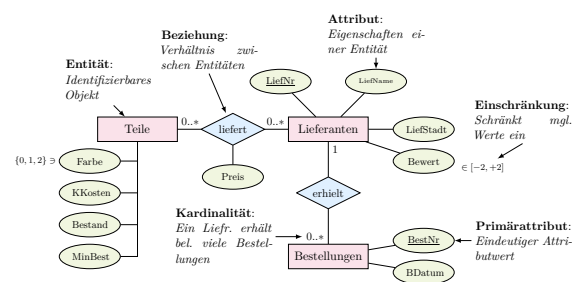


Automat/MooreAutomat

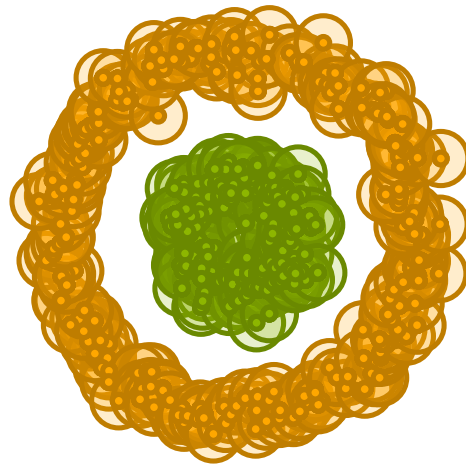


Ausgabe nur vom Zustand abhängig

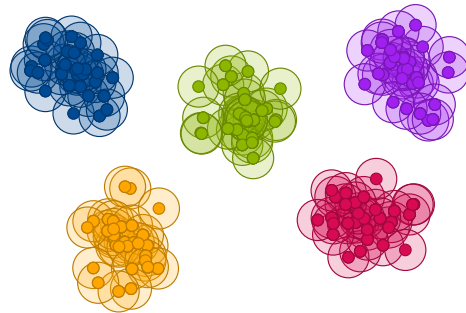
Datenbanken/ERMEExample



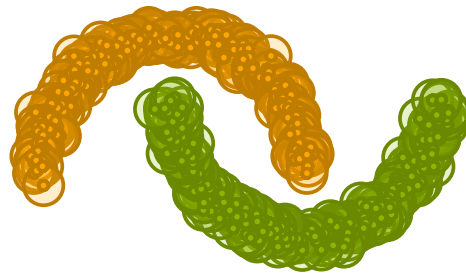
Eigene/Proseminar/Cluster/en-circles
pdf



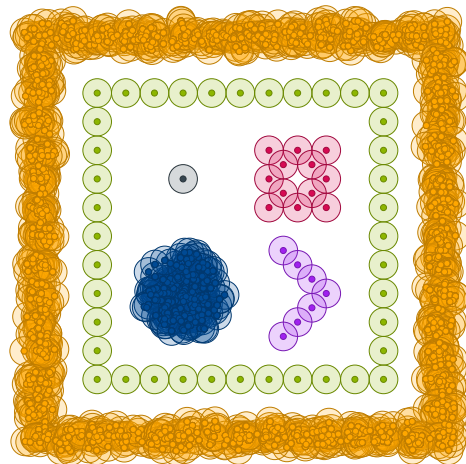
Eigene/Proseminar/Cluster/en-clusters
pdf



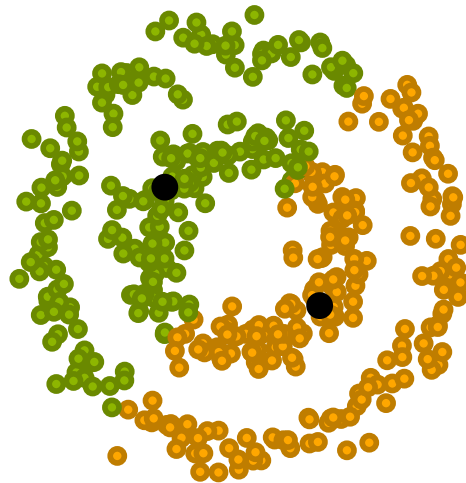
Eigene/Proseminar/Cluster/en-moons
pdf



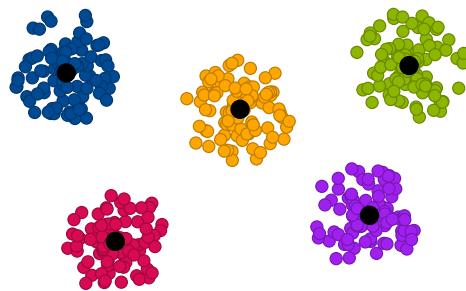
Eigene/Proseminar/Cluster/en-special
pdf



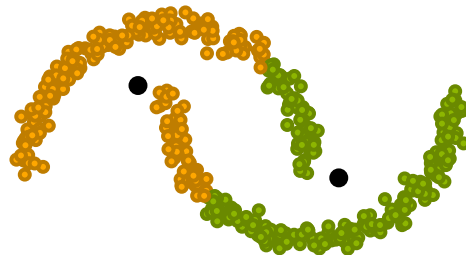
Eigene/Proseminar/Cluster/km-circles
pdf



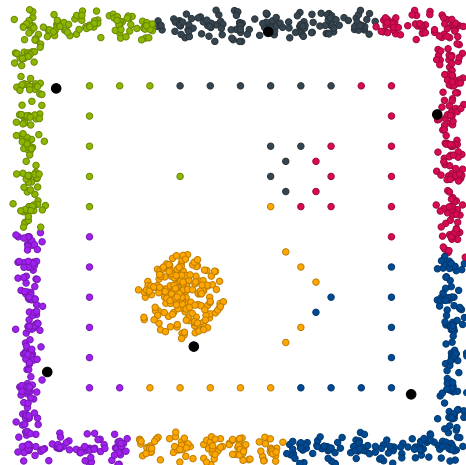
Eigene/Proseminar/Cluster/km-clusters
pdf



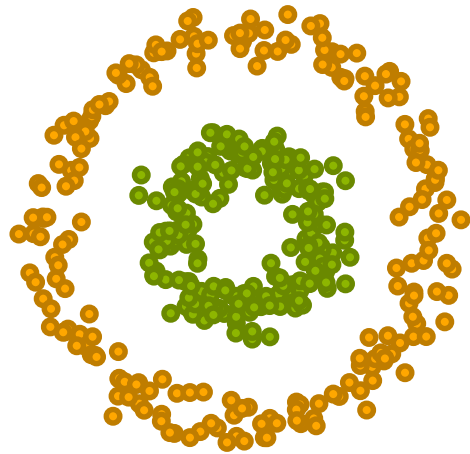
Eigene/Proseminar/Cluster/km-moons
pdf



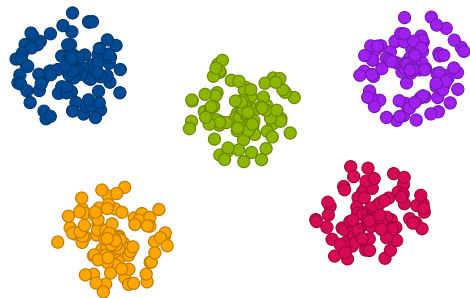
Eigene/Proseminar/Cluster/km-special
pdf



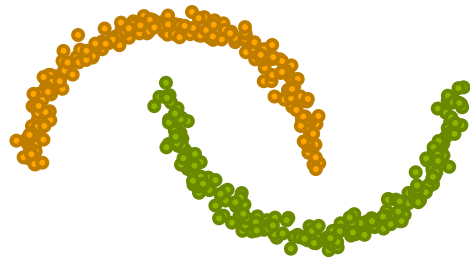
Eigene/Proseminar/Cluster/kn-circles
pdf



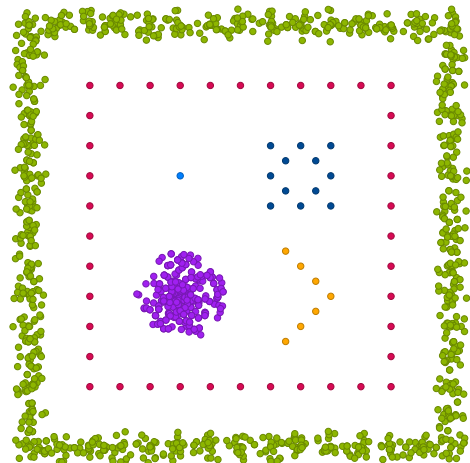
Eigene/Proseminar/Cluster/kn-clusters
pdf



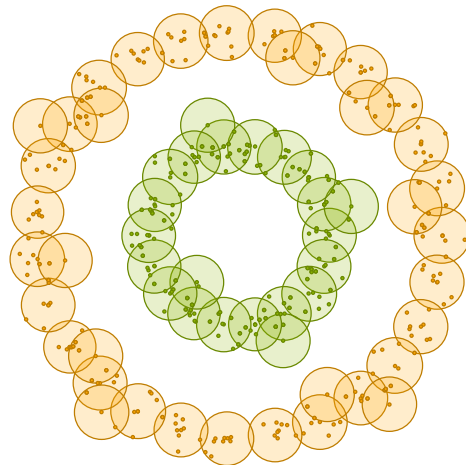
Eigene/Proseminar/Cluster/kn-moons
pdf



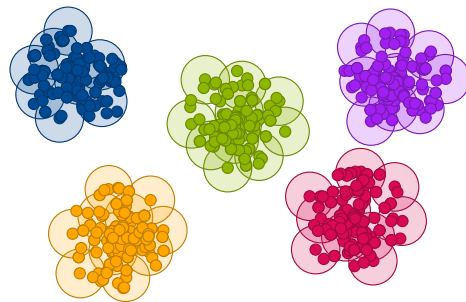
Eigene/Proseminar/Cluster/kn-special
pdf



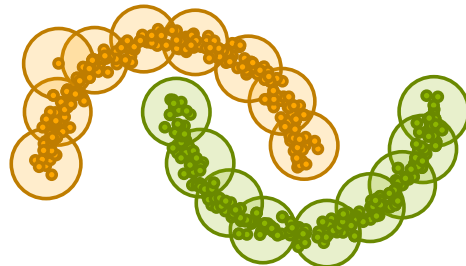
Eigene/Proseminar/Cluster/rolf-circles
pdf



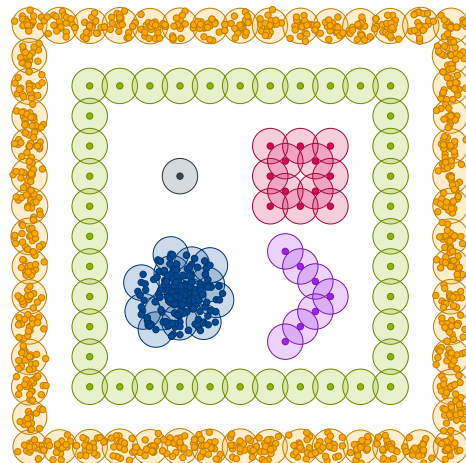
Eigene/Proseminar/Cluster/rolf-clusters
pdf



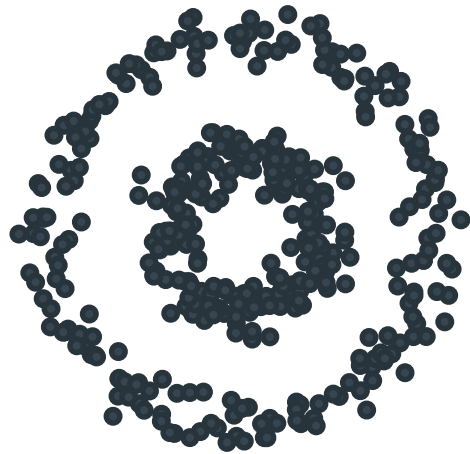
Eigene/Proseminar/Cluster/rolf-moons
pdf



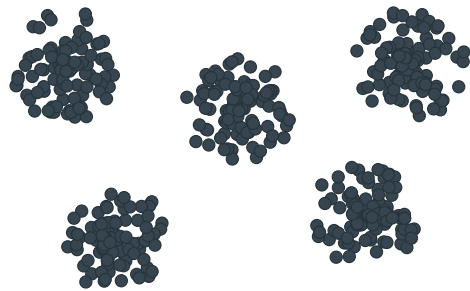
Eigene/Proseminar/Cluster/rolf-special
pdf



Eigene/Proseminar/Cluster/thumb-circles
pdf



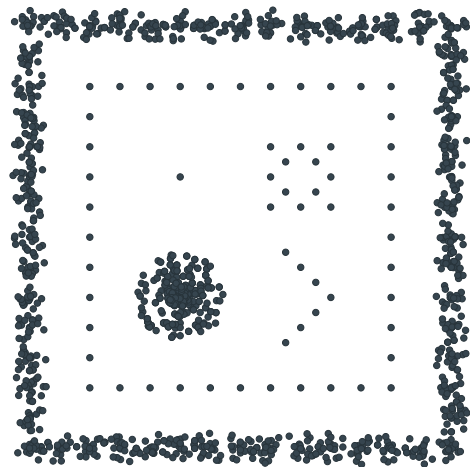
Eigene/Proseminar/Cluster/thumb-clusters
pdf



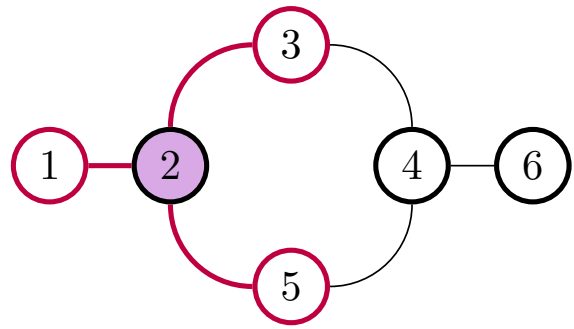
Eigene/Proseminar/Cluster/thumb-moons
pdf



Eigene/Proseminar/Cluster/thumb-special
pdf

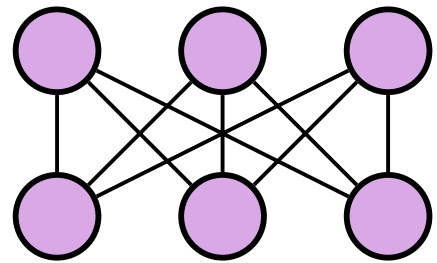


Graphen/GraphNachbarschaftGrad



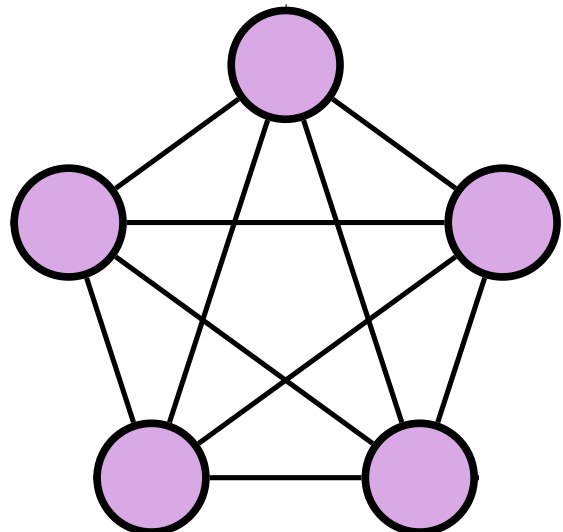
$K_{3,3}$

Graphen/GraphNichtPlanarK33



K_5

Graphen/GraphNichtPlanarK5



A graph with 6 nodes labeled 1 through 6. Nodes 1, 2, 3, 4, and 6 are red, while node 5 is black. The edges are: (1,2), (2,3), (3,4), (4,6), (2,5), and (4,5).

A graph with 6 nodes labeled 1 through 6. Nodes 1, 2, 3, 4, and 5 are arranged in a cycle, connected by thick red edges. Node 6 is connected to node 4 by a thin black edge.

The diagram illustrates the type hierarchy in glibc, showing the relationships between various data types and their associated operations. The types are organized into a hierarchical structure, with arrows indicating the flow of information or the relationship between types.

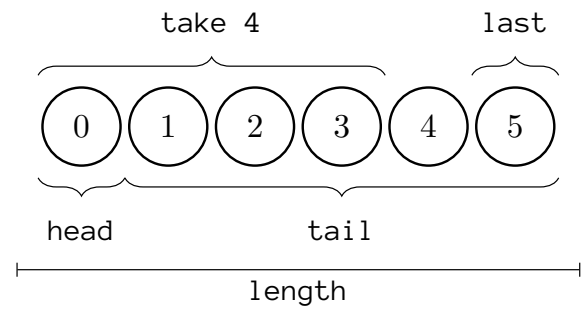
Types and their associated operations:

- Eq** (Patternmatching): `(eq), (ne)`
- Show** (Anzeigen im glibc): `show, ...`
- Read** (Einlesen von Werten): `read, ...`
- Ord** (Ermöglicht Listengenerierung getreu: [a..b]): `(lt), (le), (gt), (ge), (lt), (le), min, max`
- Num** (Anzeigen im glibc): `(lt), (le), (gt), (ge), (minimum), ...`
- Bounded** (Es existieren minimum und Maximum): `...`
- Enum** (Ermöglicht Listengenerierung getreu: [a..b]): `succ, pred`
- Real** (Anzeigen im glibc): `...`
- Fractional** (Anzeigen im glibc): `(lt), ...`
- Integral** (Ermöglicht Listengenerierung getreu: [a..b]): `div, mod, ...`
- RealFrac** (Anzeigen im glibc): `round, ceiling, floor, ...`
- Floating** (Anzeigen im glibc): `pi, exp, log, sqrt`
- RealFloat** (Anzeigen im glibc): `...`

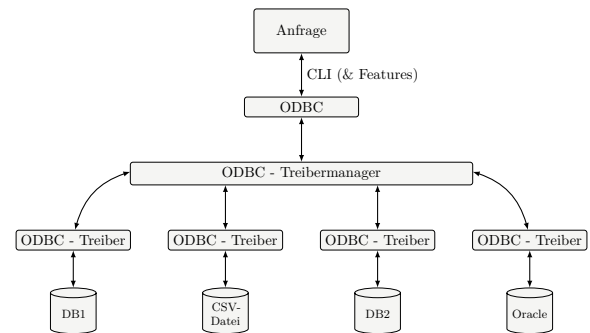
The diagram shows the following relationships (arrows):

- Eq** is derived from **Ord**.
- Show** is derived from **Eq**.
- Read** is derived from **Show**.
- Ord** is derived from **Enum**, **Real**, and **Fractional**.
- Num** is derived from **Real** and **Fractional**.
- Bounded** is derived from **Num**.
- Enum** is derived from **Integral**.
- Real** is derived from **Integral**, **RealFrac**, and **Floating**.
- Fractional** is derived from **RealFrac** and **Floating**.
- RealFrac** is derived from **RealFloat**.
- Floating** is derived from **RealFloat**.

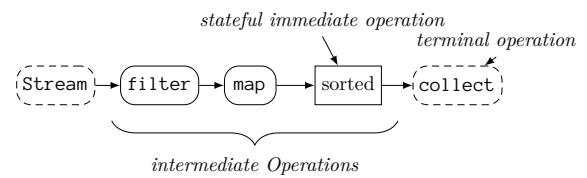
Haskell/Listenoperationen



Java/ODBCSchematisch



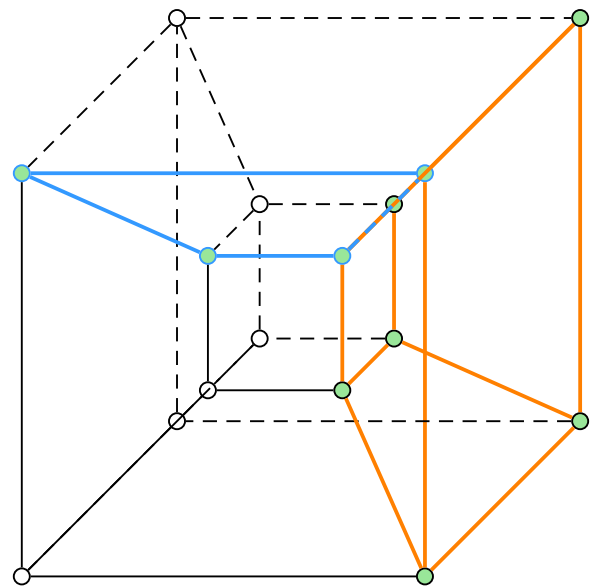
Java/StreamDemo



Logik/KVDiagramm

	\bar{a}	a	a	\bar{a}	
\bar{b}	0	0	0	0	\bar{d}
b	1	1	1	1	\bar{d}
b	1	1	1	1	d
\bar{b}	0	0	1	1	d
	\bar{c}	\bar{c}	c	c	

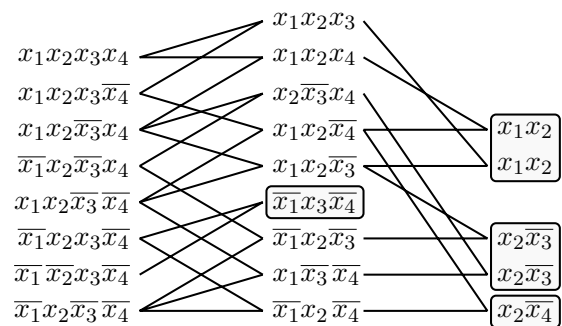
Logik/KVWuerfel



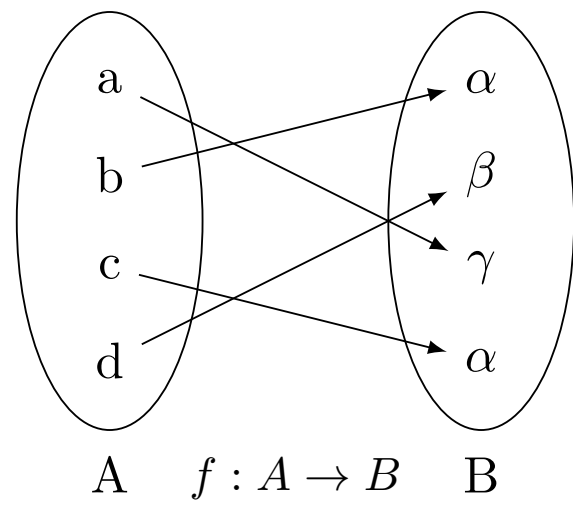
Logik/QuineMCcluskeyTabelle

	1	2	3	4	5	6	7	8
x_1x_2	+	+	+		+			
$x_2\bar{x}_3$			+	+	+			+
$x_2\bar{x}_4$		+			+	+		+
$\bar{x}_1x_3\bar{x}_4$						+	+	

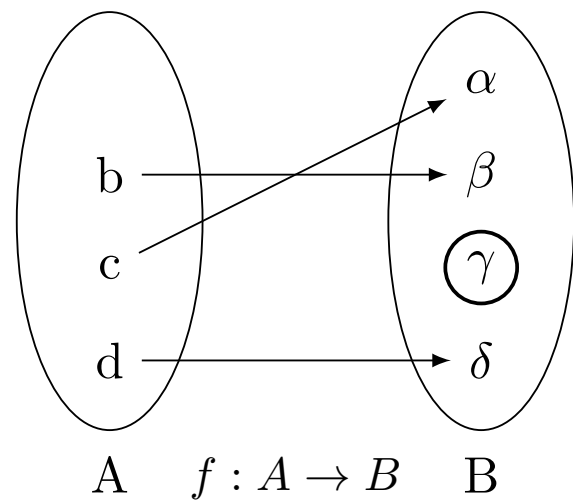
Logik/QuineMCcluskeyZusammenfassen



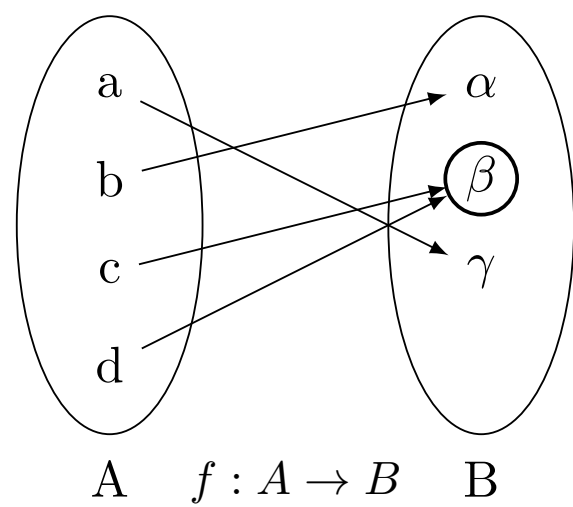
Mengen/FunktionBijektiv



Mengen/FunktionInjektiv



Mengen/FunktionSurjektiv



Mengen/Mengenmultiplikation/Mengenmultiplikation1

$$\begin{pmatrix} 1 & 3 & 4 & -1 \\ 1 & 1 & 3 & 0 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 1 & 2 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Für a ergibt sich also:
 $a = 1 * 3 + 3 * 1 + 4 * 1 + (-1) * 1$
 $= 9$

Mengen/Mengenmultiplikation/Mengenmultiplikation2

$$\begin{pmatrix} 1 & 3 & 4 & -1 \\ 1 & 1 & 3 & 0 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 1 & 2 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 9 & b \\ c & d \end{pmatrix}$$

Für b ergibt sich also:
 $b = 1 * 0 + 3 * 2 + 4 * 1 + (-1) * 2$
 $= 8$

Mengen/Mengenmultiplikation/Mengenmultiplikation3

$$\begin{pmatrix} 1 & 3 & 4 & -1 \\ 1 & 1 & 3 & 0 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 1 & 2 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 9 & 8 \\ c & d \end{pmatrix}$$

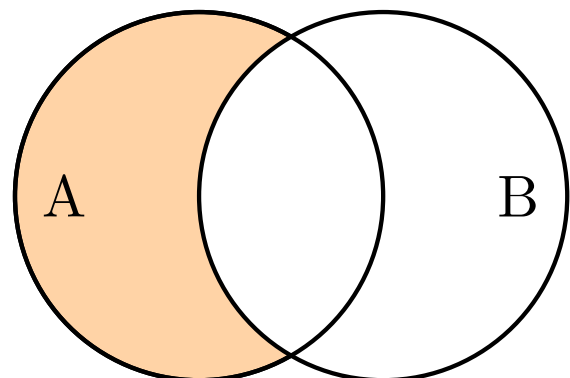
Für c ergibt sich also:
 $c = 1 * 3 + 1 * 1 + 3 * 1 + 0 * 1$
 $= 7$

Mengen/Mengenmultiplikation/Mengenmultiplikation4

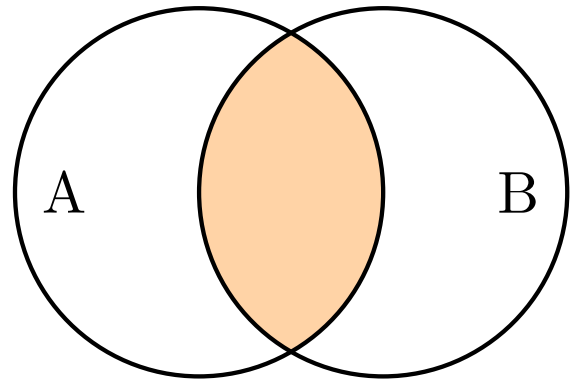
$$\begin{pmatrix} 1 & 3 & 4 & -1 \\ 1 & 1 & 3 & 0 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 1 & 2 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 9 & 8 \\ 7 & d \end{pmatrix}$$

Für d ergibt sich also:
 $d = 1 * 0 + 1 * 2 + 3 * 1 + 0 * 2$
 $= 5$

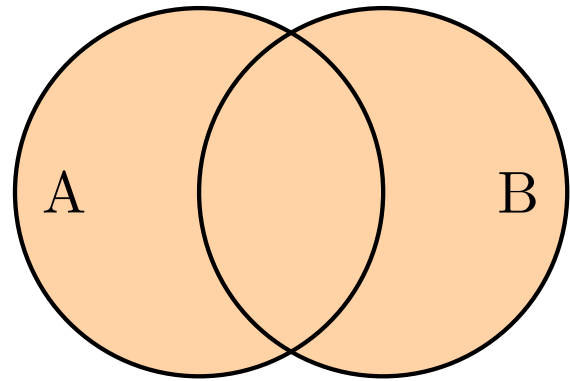
Mengen/VennDifferenz



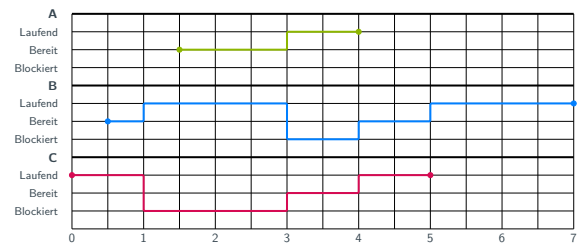
Mengen/VennSchnitt



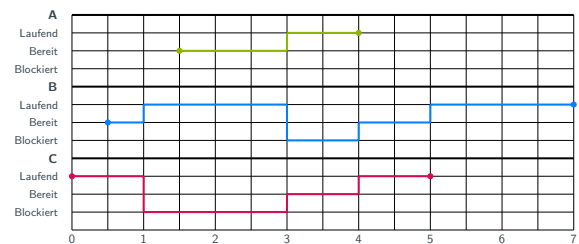
Mengen/VennVereinigung



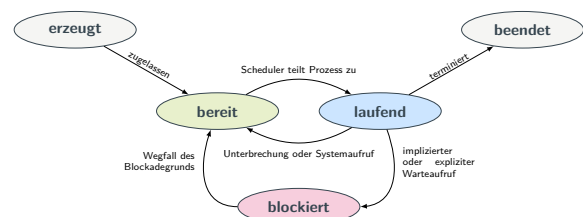
Prozesse/FCFS-WorstCase



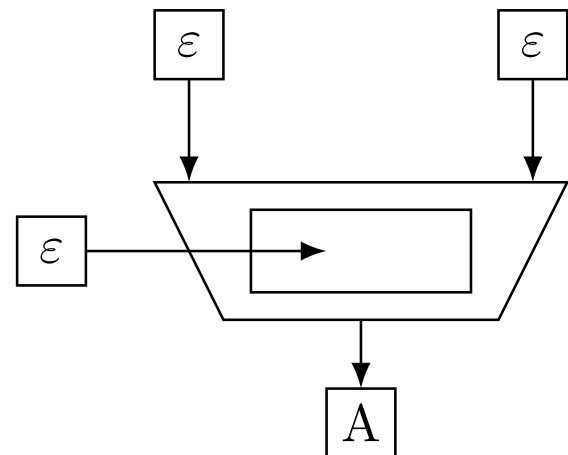
Prozesse/FCFS



Prozesse/Prozesszustände



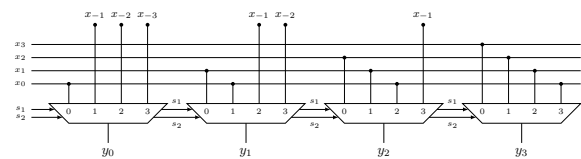
Rechner/ALU



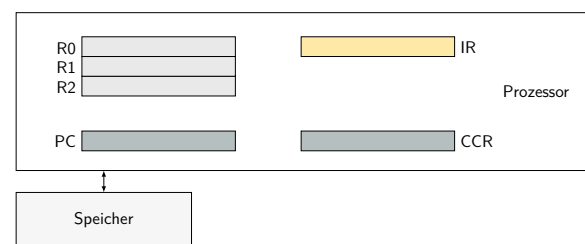
Rechner/AmpelPLA

δ	Zustand			Folgez.		NS			WO			T
	h_0	z_0	z_1	z'_0	z'_1	\bullet	\circ	\circ	\bullet	\circ	\circ	
δ_0	0	0	0	0	1	\bullet	\circ	\circ	\circ	\circ	\bullet	$7 \Rightarrow h'_0 = 1$
δ_1	1	0	1	0	1	\bullet	\circ	\circ	\circ	\bullet	\circ	$T - 1$
δ_3	0	0	1	1	0	\circ	\bullet	\circ	\circ	\circ	\bullet	$2 \Rightarrow h'_0 = 1$
δ_4	1	1	0	1	0	\circ	\bullet	\circ	\circ	\bullet	\circ	$T - 1$
δ_6	0	1	0	1	1	\circ	\circ	\bullet	\bullet	\circ	\circ	$7 \Rightarrow h'_0 = 1$
δ_7	1	1	1	1	1	\circ	\circ	\bullet	\bullet	\circ	\circ	$T - 1$
δ_9	0	1	1	0	0	\circ	\bullet	\bullet	\circ	\bullet	\circ	$2 \Rightarrow h'_0 = 1$
δ_{10}	1	0	0	0	0	\circ	\bullet	\bullet	\circ	\bullet	\circ	$T - 1$

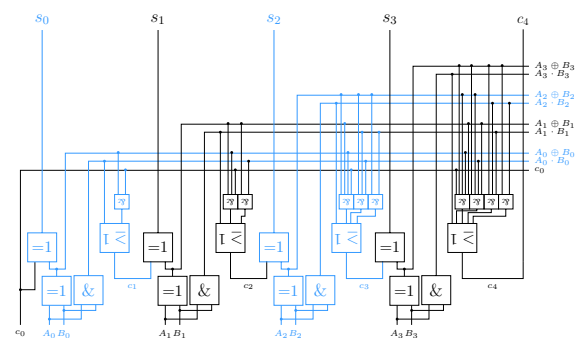
Rechner/BarrelShifter



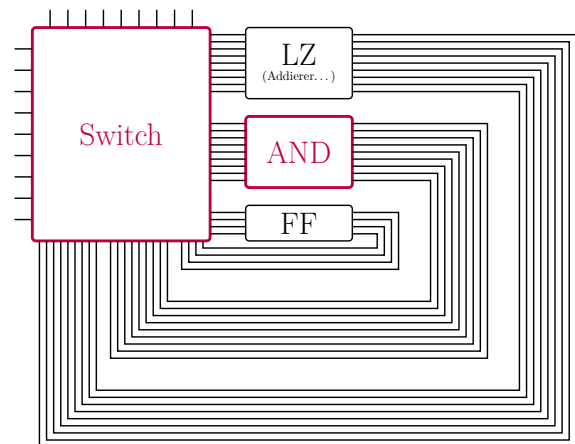
Rechner/Beispielprozessor



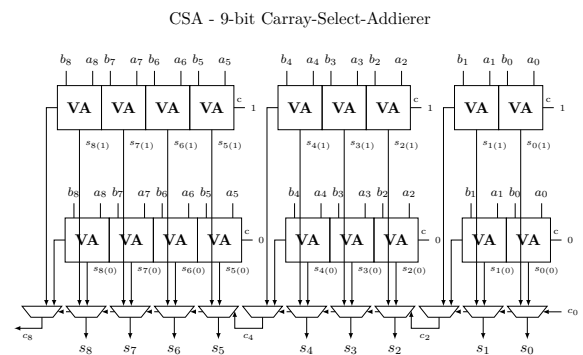
Rechner/CLA



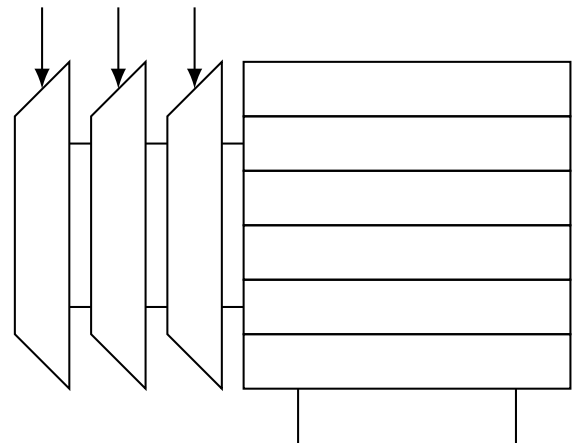
Rechner/CPLD



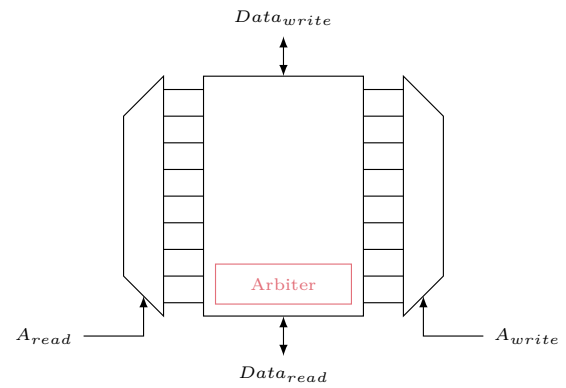
Rechner/CSA



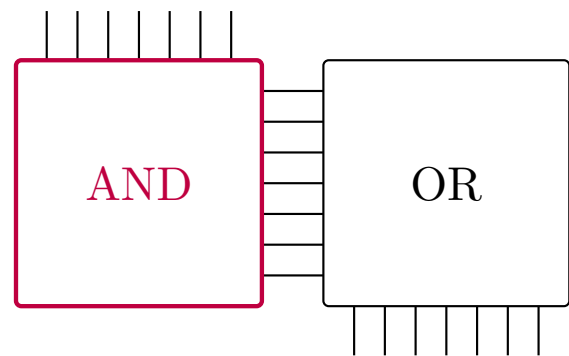
Rechner/DreiTorRegister



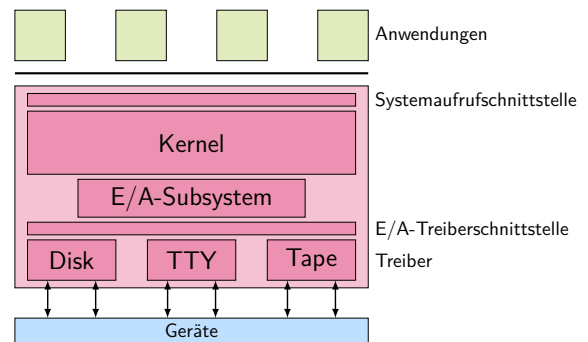
Rechner/Eintorspeicher



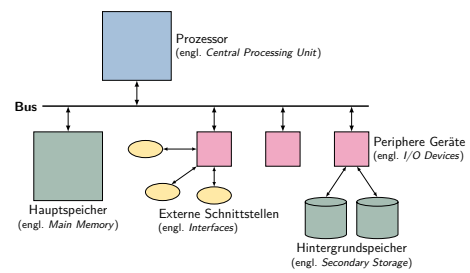
Rechner/GALPAL



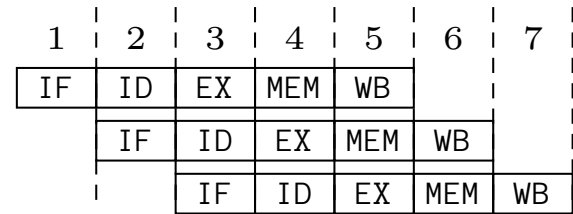
Rechner/Geraeteverwaltung



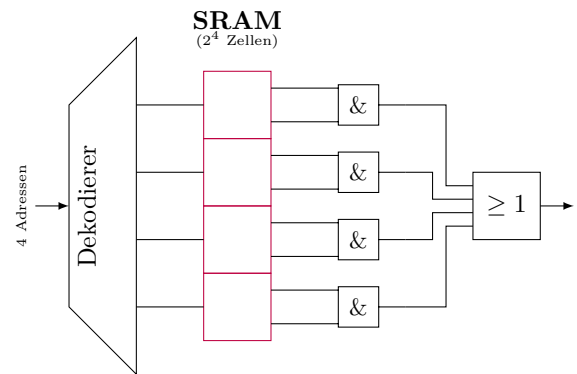
Rechner/HardwareSkizze



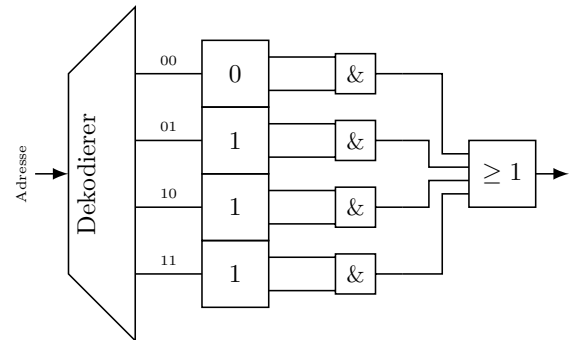
Rechner/ISA



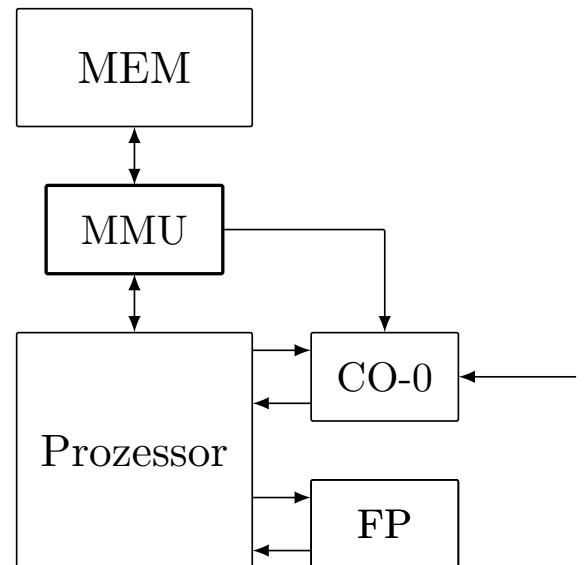
Rechner/LUT



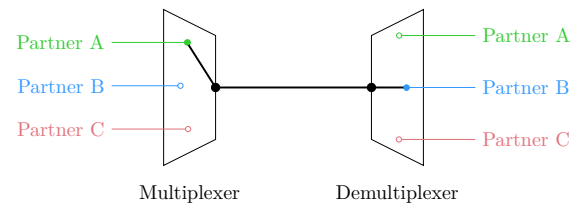
Rechner/LUT0der



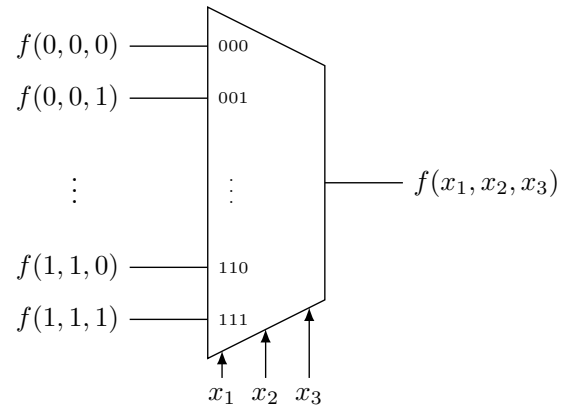
Rechner/MIPS



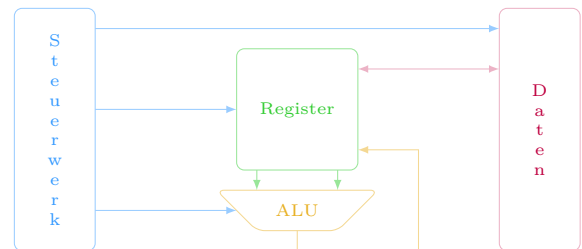
Rechner/MuxDemuxKommunikation



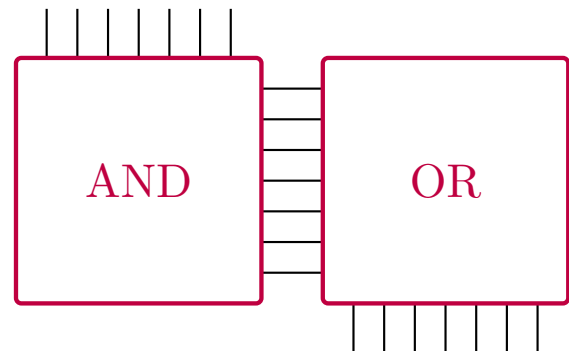
Rechner/MuxShannon



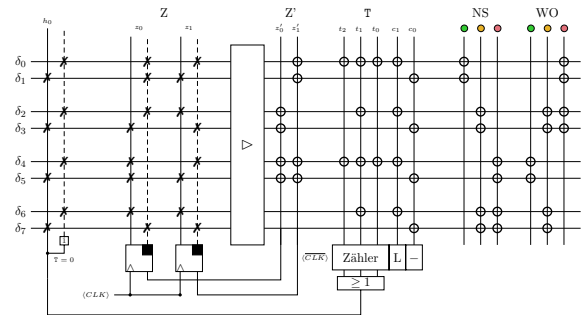
Rechner/NAdressmaschine



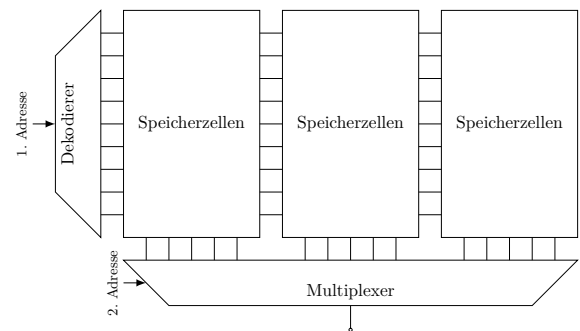
Rechner/PLA



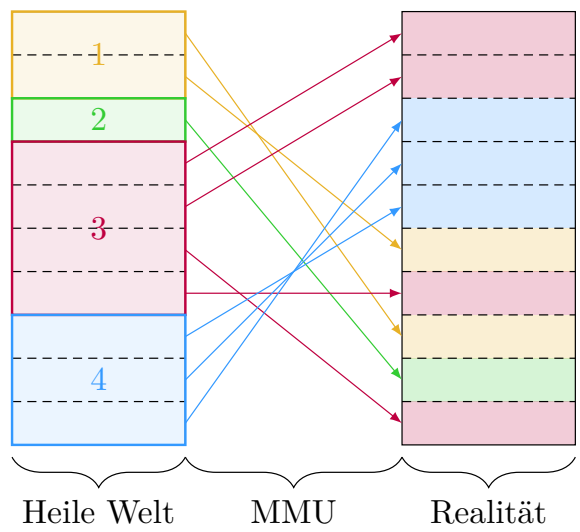
Rechner/PLA Ampel



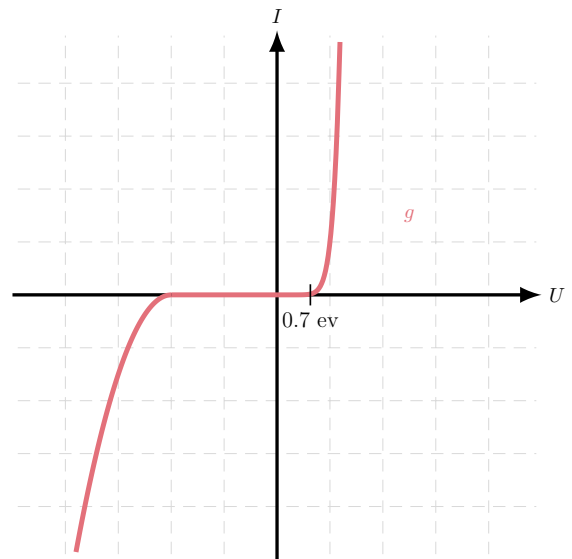
Rechner/PROM



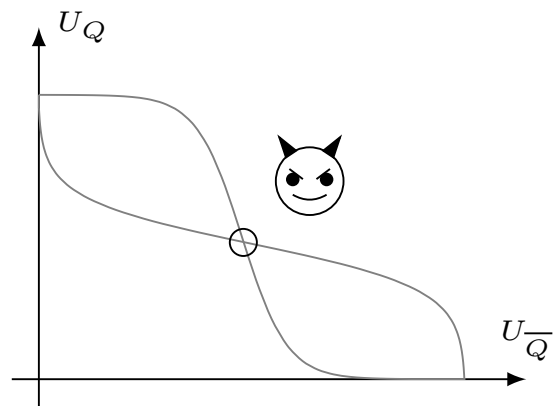
Rechner/Pages



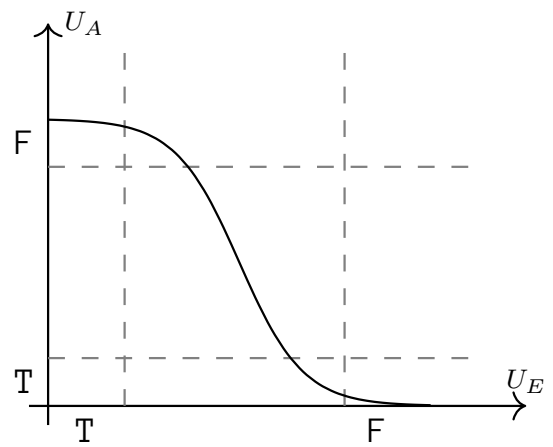
Rechner/Physik/DiodenStromstaerke



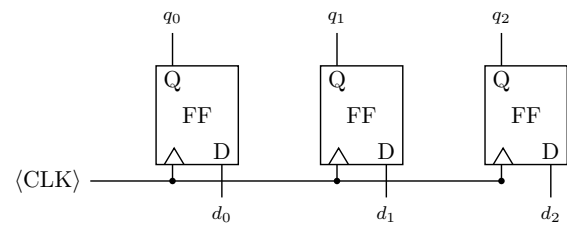
Rechner/Physik/Metastabil



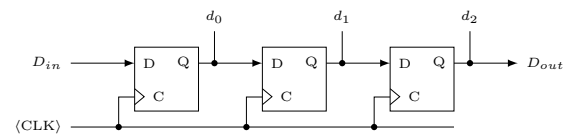
Rechner/Physik/TransistorStoertoleranz



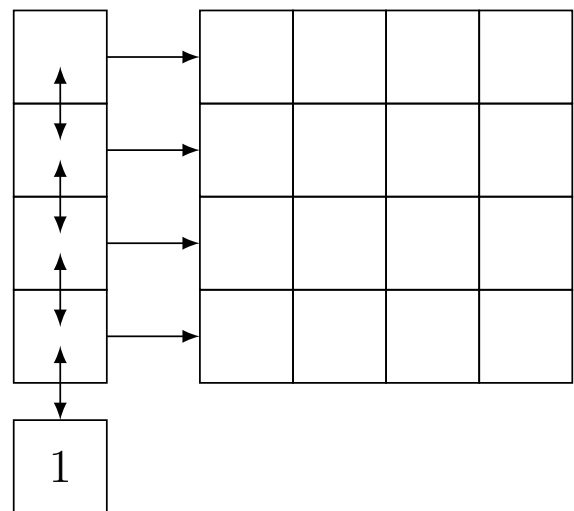
Rechner/RegisterParallel



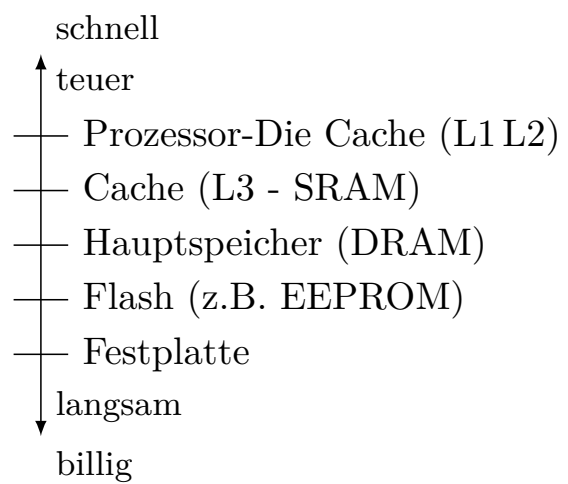
Rechner/RegisterSeriell



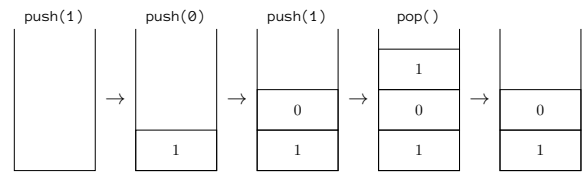
Rechner/Shiftregister



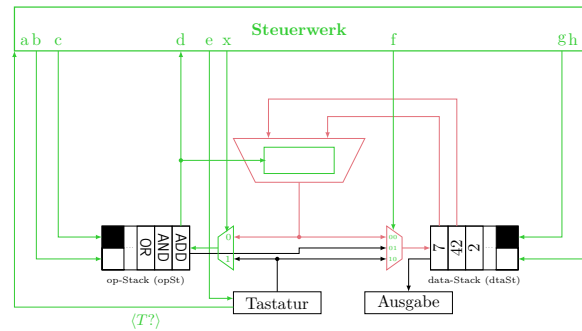
Rechner/Speicherhierarchie



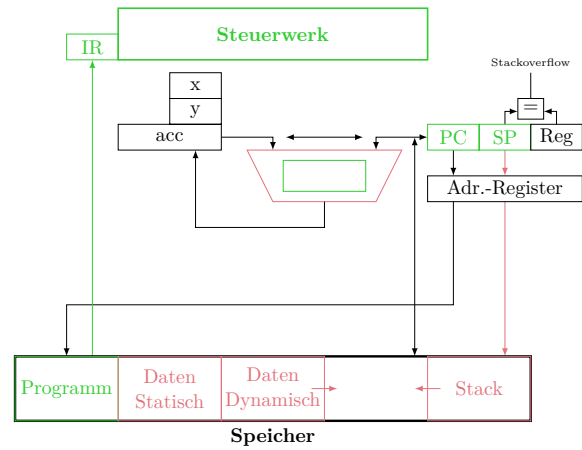
Rechner/StackExample



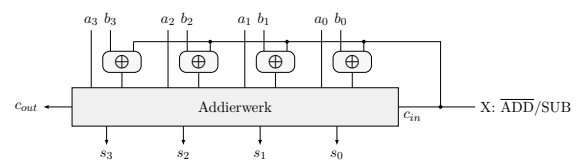
Rechner/Stackmaschine



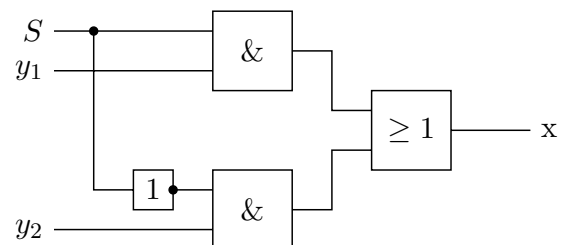
Rechner/StackmaschineSimpler



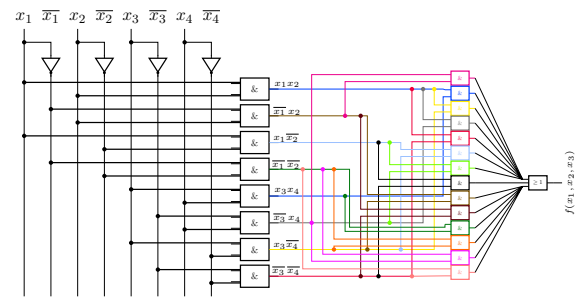
Schaltkreis/Addier-Subtrahierer



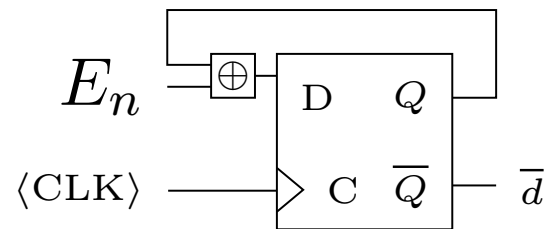
Schaltkreis/Demultiplexer



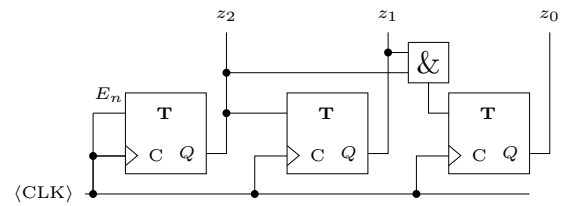
Schaltkreis/KomplexerSchaltkreis



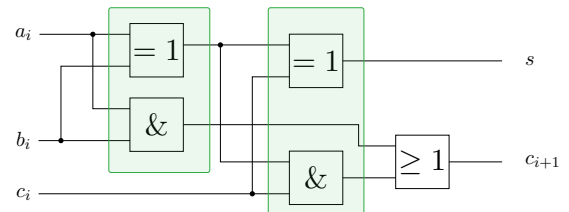
Schaltkreis/SynchroneZaehlerDFF



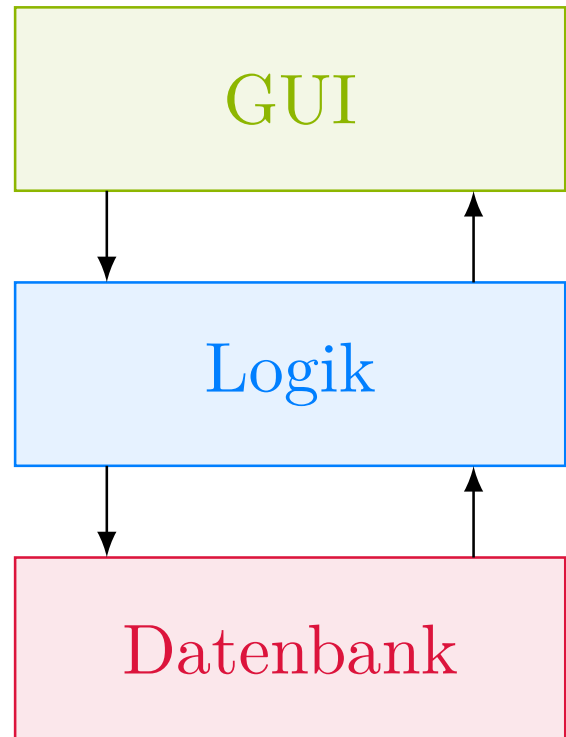
Schaltkreis/SynchroneZaehlerTFF



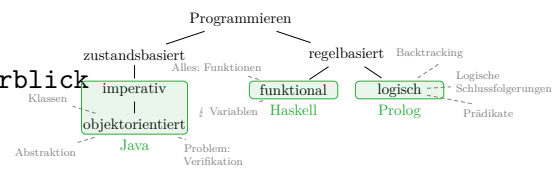
Schaltkreis/Volladdierer



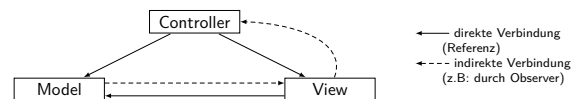
Software/DreiSchichtenArchitektur



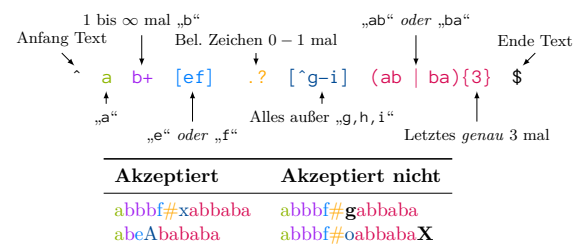
Software/Meta/ProgrammierparadigmenUeberblick



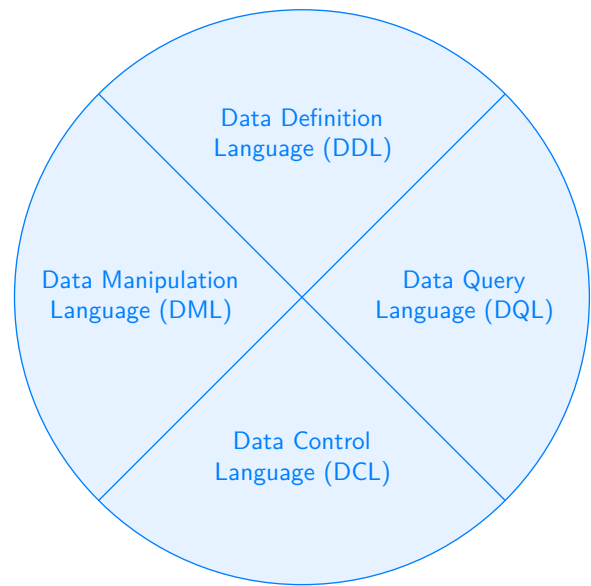
Software/ModelViewController



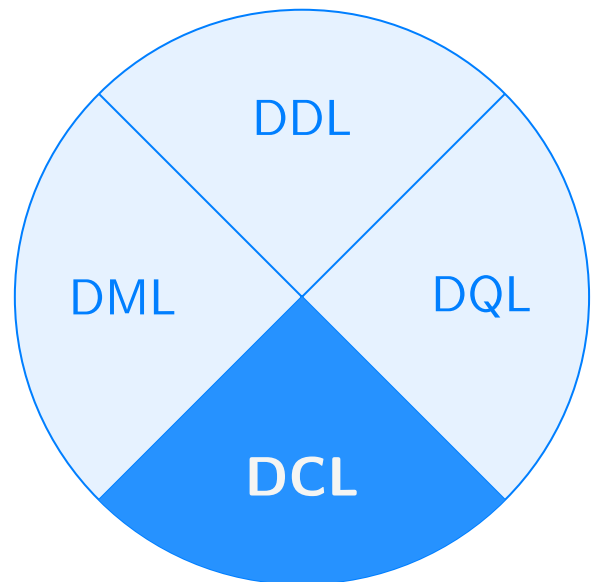
Software/RegexExample



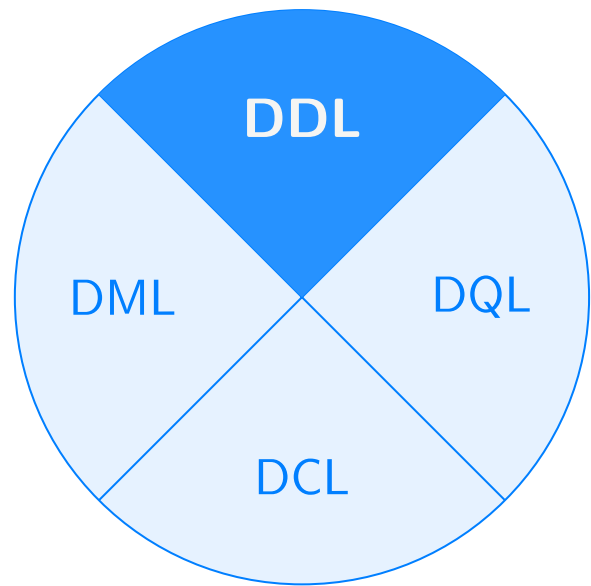
Software/SQL/SQLFields



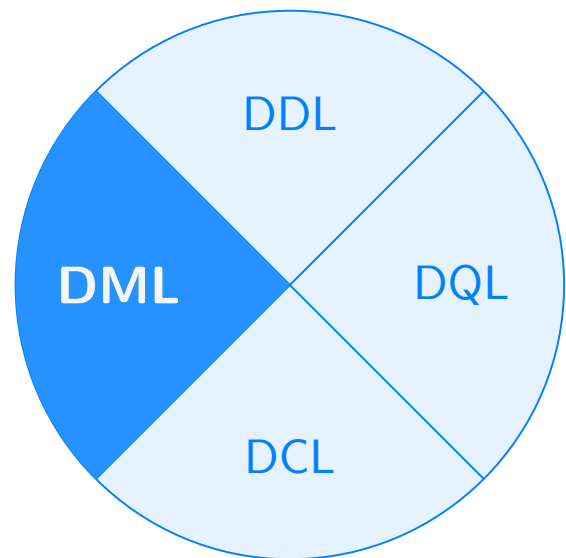
Software/SQL/SQLFieldsDCL



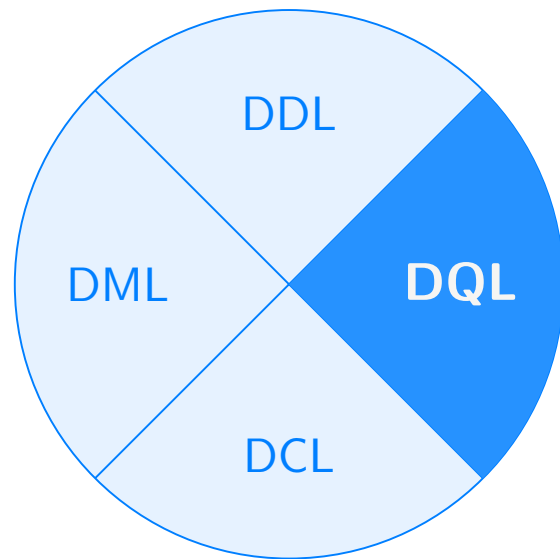
Software/SQL/SQLFieldsDDL



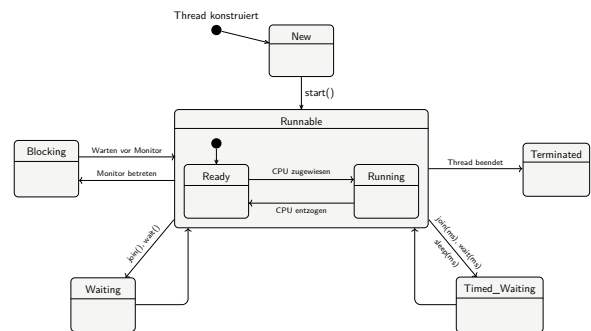
Software/SQL/SQLFieldsDML



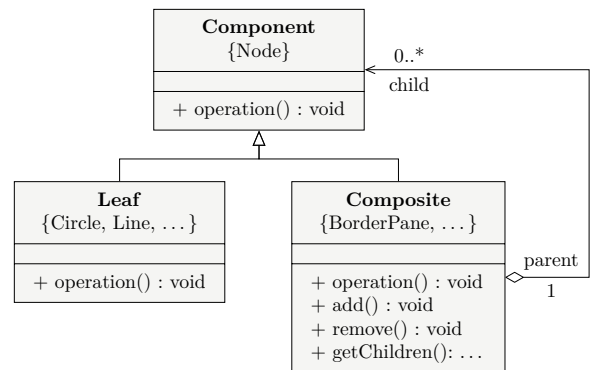
Software/SQL/SQLFieldsDQL



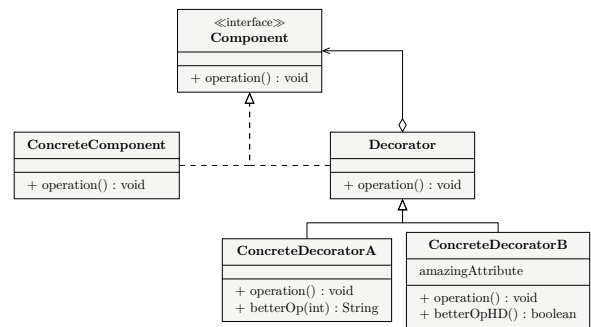
Software/ThreadStates



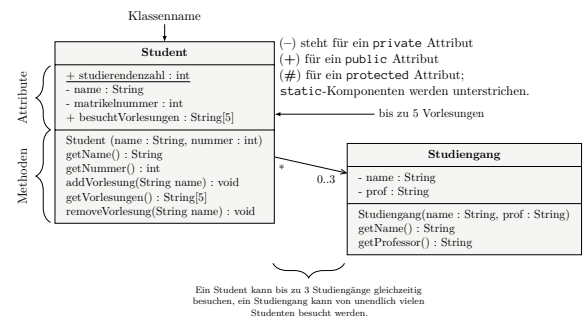
Software/UML/UMLCompositePattern



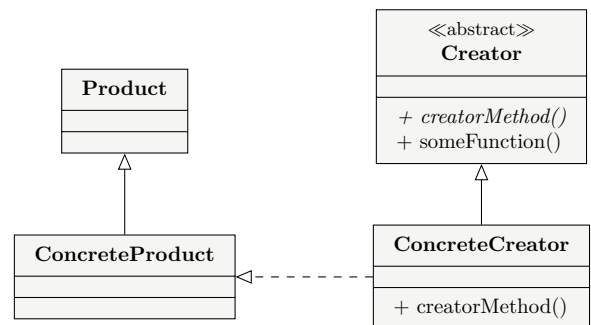
Software/UML/UMLDecoratorPattern



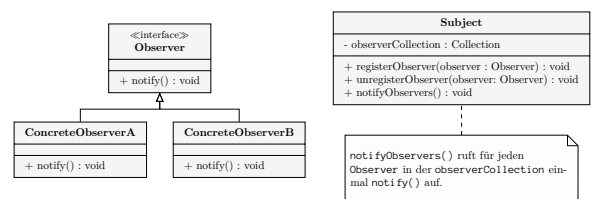
Software/UML/UMLExample



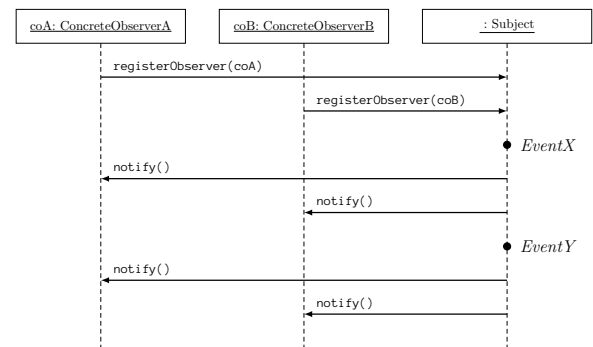
Software/UML/UMLFactoryPattern



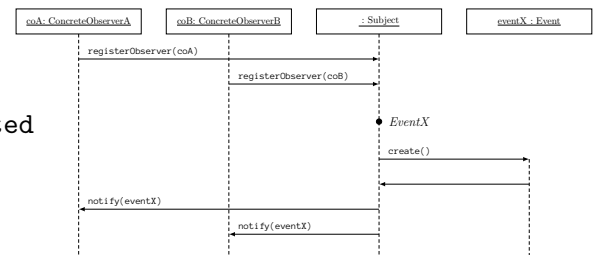
Software/UML/UMLObserverPattern



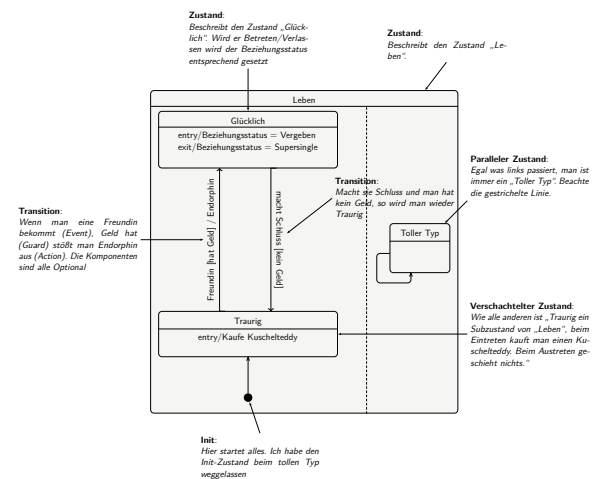
Software/UML/UMLSEQObserverPattern



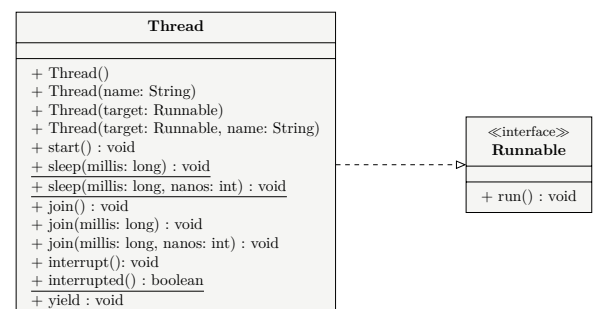
Software/UML/UMLSEQObserverPatternAdapted



Software/UML/UMLStateDiagramExample



Software/UML/UMLThread



Software/XML/XMLUebersicht

Liefert Metadaten zur Datei, xml ist das

Wurzelement → `<?xml version="1.0" encoding="UTF-8" />` (Kommentar)

Kind des „blaetter“-Elements → `<blaetter>` (Kommentar: `<!-- Alle Blaetter -->`)

Kind des „blaetter“-Elements → `<blatt num="42" abgabe="10.10" />` (Attribut: `num="42"`, `abgabe="10.10"`)

Kind des „blatt“-Elements → `<aufgabe topic="Java_NIO" />` (Attribut: `topic="Java_NIO"`)

Kind des „aufgabe“-Elements → `<unteraufgabe nr="1.1">` (Attribut: `nr="1.1"`)

Kind des „unteraufgabe“-Elements → `<punkte>4</punkte>` (Text: `4`)

Kind des „punkte“-Tags → `<text>Bitte ... & />` (Text: `Bitte ... &`)

Kind des „text“-Tags → `<grafik />` (Text: `Leerer „Grafik“`)

Kind des „grafik“-Tags → `</unteraufgabe>` (Schluss-Tag)

Kind des „unteraufgabe“-Tags → `</aufgabe>` (Schluss-Tag)

Kind des „aufgabe“-Tags → `</blatt>` (Schluss-Tag)

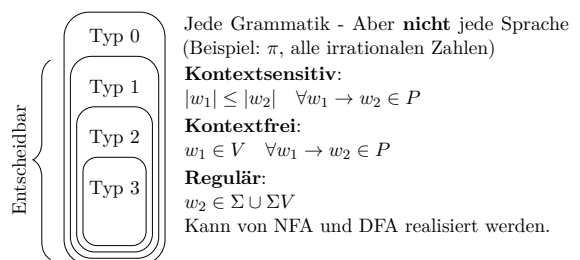
Kind des „blatt“-Tags → `</blaetter>` (Schluss-Tag)

Verbleibende Elemente: `</xml>`

Sprachen/CYKAlgorithmus

		1 a	2 b	3 c	4 c
T[1,j]	1	A	B	CE	CE
T[2,j]	2	D	—	E	
T[3,j]	3	S	—		
T[4,j]	4	S			

Sprachen/ChomskyHierarchie



Sprachen/Grammatik

