

# lehramt-informatik

Hermine Bschlangaul \*

September 4, 2021

## Contents

<b>1</b>	<b>Klassen</b>	<b>4</b>
1.1	Vorlage Theorie-Teil	5
1.2	Vorlage Aufgabensammlung	6
1.3	Vorlage Aufgabe	7
<b>2</b>	<b>Pakete</b>	<b>8</b>
2.1	abmessung.sty	9
2.2	aufgaben-einbinden.sty	10
2.3	aufgaben-metadaten.sty	11
2.4	automaten.sty	12
2.4.1	Endlicher Automat	12
2.4.2	Kellerautomat	14
2.4.3	Turingmaschine	15
2.5	basis.sty	18
2.6	baum.sty	22
2.6.1	Binärbaum	23
2.6.2	AVL-Baum	24
2.6.3	B-Baum	25
2.7	checkbox.sty	26
2.8	chomsky-normalform.sty	27
2.9	cpm.sty	29
2.9.1	Makro-Kürzel	29
2.9.2	Hilfsmakros für Zeitpunkt-Tabelle	30
2.10	cyk-algorithmus.sty	32
2.10.1	Makro-Kürzel	32
2.10.2	TeX-Markup-Beispiel	32
2.11	entwurfsmuster.sty	33
2.11.1	Namensschema der Entwurfsmuster-Makros:	33
2.11.2	Reihenfolge	33
2.11.3	Abstrakte Fabrik (Abstract Factory)	33
2.11.4	Adapter	34
2.11.5	Beobachter (Observer)	36
2.11.6	Dekorierer (Decorator)	38
2.11.7	Einfache Fabrik (Simple Factory)	39
2.11.8	Einzelstück (Singleton)	40
2.11.9	Erbauer (Builder)	40
2.11.10	Fabrikmethode (Factory Method)	42
2.11.11	Kompositum (Composite)	43
2.11.12	Modell-Präsentation-Steuerung (Model-View-Controller)	44
2.11.13	Zustand (State)	45
2.12	er.sty	47
2.12.1	Tex-Markup-Beispiel: Komplettes Diagramm	47

---

\*E-mail: hermine.bschlangaul@gmx.net

2.12.2	Tex-Markup-Beispiel: EER Enhanced Entity-Relation-Modell nach Elmasri/Navante	48
2.12.3	Makro-Kürzel	48
2.13	formale-sprachen.sty	50
2.14	formatierung.sty	53
2.14.1	Schriftarten / Typographie	53
2.14.2	Farben	53
2.14.3	Überschriften	53
2.14.4	Listen	53
2.14.5	Kasten	53
2.14.6	Header	53
2.15	gantt.sty	54
2.16	grafik.sty	55
2.17	graph.sty	56
2.18	hanoi.sty	58
2.19	klassen-konfiguration-aufgabe.sty	59
2.20	klassen-konfiguration-examen.sty	60
2.21	komplexitaetstheorie.sty	61
2.21.1	Makro-Kürzel	61
2.22	kontrollflussgraph.sty	63
2.22.1	Makro-Kürzel	63
2.22.2	TeX-Markup-Beispiel	63
2.22.3	TikZ: pin	63
2.22.4	Umgebungen	64
2.22.5	Makros	65
2.23	kopf-fusszeilen.sty	66
2.24	literatur-dummy.sty	67
2.25	literatur.sty	68
2.26	makros.sty	69
2.27	master-theorem.sty	73
2.27.1	Makro-Kürzel	73
2.28	mathe.sty	77
2.29	minimierung.sty	78
2.30	normalformen.sty	81
2.30.1	Makro-Kürzel	81
2.31	petri.sty	84
2.31.1	Makro-Kürzel	84
2.32	potenzmengen-konstruktion.sty	86
2.33	pseudo.sty	88
2.34	pumping-lemma.sty	89
2.35	quicksort.sty	90
2.36	relationale-algebra.sty	93
2.37	rmodell.sty	94
2.37.1	Makro-Kürzel	94
2.38	sortieren.sty	95
2.39	spalten.sty	97
2.40	sql.sty	98
2.41	struktogramm.sty	99
2.42	syntax.sty	100
2.42.1	Makro-Kürzel	100
2.43	syntaxbaum.sty	102
2.44	synthese-algorithmus.sty	103
2.44.1	Makro-Kürzel	103
2.44.2	TeX-Markup Grundgerüst	103
2.44.3	TeX-Markup Linksreduktion	103
2.44.4	TeX-Markup Rechtsreduktion	103
2.44.5	TeX-Markup Relationen formen	104
2.45	tabelle.sty	107

2.46	typographie.sty . . . . .	108
2.47	uml.sty . . . . .	109
2.48	vollstaendige-induktion.sty . . . . .	111
2.48.1	Makro-Kürzel . . . . .	111
2.49	wasserfall.sty . . . . .	113
2.50	wpkalkuel.sty . . . . .	114
2.50.1	Makro-Kürzel . . . . .	114
<b>3</b>	<b>Index</b>	<b>115</b>

# 1 Klassen

## 1.1 Vorlage Theorie-Teil

```
\documentclass{lehramt-informatik-haupt}

\begin{document}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Theorie-Teil
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\chapter{Thema des Theorie-Teils}

\literatur

\end{document}
```

## 1.2 Vorlage Aufgabensammlung

```
\documentclass{lehramt-informatik-haupt}  
\liLadeAllePakete  
  
\begin{document}  
\liAufgabe{30_AUD/06_Graphen/20_Dijkstra/Aufgabe_Graph-A-I}  
\liAufgabe{30_AUD/06_Graphen/20_Dijkstra/Aufgabe_Graph-M-A-P-R-N}  
\liAufgabe{30_AUD/06_Graphen/20_Dijkstra/Aufgabe_Staedte-A-F}  
\liExamensAufgabe{46114/2008/09/Thema-1/Aufgabe-2}  
\liExamensAufgabe{46115/2013/03/Thema-2/Aufgabe-5}  
\liExamensAufgabe{66112/2004/03/Thema-1/Aufgabe-5}  
\liExamensAufgabe{66115/2013/09/Thema-2/Aufgabe-9}  
\liExamensAufgabe{66115/2015/03/Thema-2/Aufgabe-7}  
\liExamensAufgabe{66115/2016/03/Thema-2/Aufgabe-6}  
\liExamensAufgabe{66115/2017/03/Thema-1/Aufgabe-1}  
\liExamensAufgabe{66115/2018/03/Thema-2/Aufgabe-9}  
\liExamensAufgabe{66115/2020/09/Thema-1/Teilaufgabe-2/Aufgabe-3}  
\end{document}
```

### 1.3 Vorlage Aufgabe

```
\documentclass{lehramt-informatik-aufgabe}
\liLadePakete{}
\begin{document}
\liAufgabenTitel{}
\section{
\index{DB}
\footcite{examen:}
}

\end{document}
```

## 2 Pakete



## 2.1 abmessung.sty

```
1 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2 \ProvidesPackage{lehramt-informatik-baum}[2021/09/04 Einstellung der
3 Seitenabmessung mit Hilfe des geometry-Pakets.]
4 \RequirePackage{geometry}
5 \geometry{
6   a4paper,
7   margin=2cm,
8   includeheadfoot,
9   %showframe,
10  %showcrop,
11  %verbose=true,
12 }
13
```

## 2.2 aufgaben-einbinden.sty

```
14 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
15 \ProvidesPackage{lehramt-informatik-aufgaben-einbinden}[2020/06/13]

\liAufgabe Eine Aufgaben mit den Pfad relativ zu Wurzelverzeichnis dieses Repository einbinden.
16 \def\liAufgabe#1{
17   \input{\LehramtInformatikRepository/#1.tex}
18 }

\liExamensAufgabe Eine Examensaufgaben mit den Pfad relativ zu Staatsexamen einbinden, z. B.
\liExamensAufgabe{46116/2015/09/Thema-2/Teilaufgabe-1/Aufgabe-3}
19 \def\liExamensAufgabe#1{
20   \input{\LehramtInformatikRepository/Staatsexamen/#1.tex}
21 }

\liExamensAufgabeTTA
22 \def\liExamensAufgabeTTA #1 / #2 / #3 : Thema #4 Teilaufgabe #5 Aufgabe #6 {
23   \input{\LehramtInformatikRepository/Staatsexamen/#1/#2/#3/Thema-#4/Teilaufgabe-
      #5/Aufgabe-#6.tex}
24 }

\liExamensAufgabeTA
25 \def\liExamensAufgabeTA #1 / #2 / #3 : Thema #4 Aufgabe #5 {
26   \input{\LehramtInformatikRepository/Staatsexamen/#1/#2/#3/Thema-#4/Aufgabe-#5.tex}
27 }

\liExamensAufgabeA
28 \def\liExamensAufgabeA #1 / #2 / #3 : Aufgabe #4 {
29   \input{\LehramtInformatikRepository/Staatsexamen/#1/#2/#3/Aufgabe-#4.tex}
30 }

31
```

## 2.3 aufgaben-metadaten.sty

```

32 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
33 \ProvidesPackage{lehramt-informatik-aufgaben-metadaten}[2020/07/07 Makros zum
34 Setzen der Aufgaben-Metadaten.]

35 \ExplSyntaxOn

```

`\liAufgabenMetadaten` Setzen der Aufgaben-Metadaten über eine plist bzw. über key-values.

Die Schlüssel-Werte-Paare sind in der Datei `basis.sty` definiert. In der Typescript-Datei `.scripts/nodejs/src/aufgaben.ts` gibt es ein entsprechendes Interface `AufgabenMetadaten`.

```

\liAufgabenMetadaten{
  Titel = {Aufgabe 2},
  Thematik = {Petri-Netz},
  Stichwoerter = {Feld (Array), Implementierung in Java}
  ZitatSchluessel = sosy:pu:4,
  ZitatBeschreibung = {Seite 11},
  BearbeitungsStand = OCR,
  Korrektheit = absolut korrekt,
  RelativerPfad = Staatsexamen/46116/2016/03/Thema-2/Teilaufgabe-1/Aufgabe-2.tex,
  ExamenNummer = 46116,
  ExamenJahr = 2016,
  ExamenMonat = 03,
  ExamenThemaNr = 2,
  ExamenTeilaufgabeNr = 1,
  ExamenAufgabeNr = 2,
}

```

```

36 \def\liAufgabenMetadaten#1{
37
38   \_setze_variablen_zurueck:
39
40   \tl_clear:N \g_auf_examen_telaufgabe_nr_tl
41
42   \keys_set:nn { aufgabenmetadaten } {
43     #1
44   }
45
46   \_setze_relativen_pfad:
47
48   \tl_if_empty:NTF \g_auf_titel_tl
49   {
50     \msg_fatal:nn { aufgabenmetadaten } { kein-titel }
51   }
52   {
53   }
54
55   \_gib_examen_titel: {}
56
57   \section{\_gib_aufgaben_titel:}
58
59   \bigskip
60 }

61 \ExplSyntaxOff

```

`\liAufgabenTitel` Momentan eine dummy Makro das die Thematik enthält.

```

62 \def\liAufgabenTitel#1{}

63

```

## 2.4 automaten.sty

```
64 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
65 \ProvidesPackage{lehramt-informatik-automaten}[2021/02/14 Zum Setzen von Automaten]
```

### 2.4.1 Endlicher Automat

```
66 \liLadePakete{formale-sprachen}

\begin{tikzpicture}[li automat]
\node[state,initial,accepting] (0) {$z_0$};
\node[state,right of=0] (1) {$z_1$};

\path (0) edge[above] node{1} (1);
\path (0) edge[loop,above] node{0} (0);
\path (1) edge[loop,above] node{0} (1);
\end{tikzpicture}
```



```
67 \RequirePackage{tikz}
68 \usetikzlibrary{arrows,automata,positioning}
69 \liLadePakete{mathe}
70 \directlua{
71   automaten = require('lehramt-informatik-automaten')
72 }
```

`\liAutomat` `\liAutomat[automaten-name]{zustaende=Z,alphabet=Σ,delta=δ,ende=E,start=z0}`

- `\liAutomat{}`:  $A = (Z, \Sigma, \delta, E, z_0)$
- `\liAutomat[A_1]{}`:  $A_1 = (Z, \Sigma, \delta, E, z_0)$
- `\liAutomat{zustaende={z_0, z_1, z_2}}`:  $A = (\{z_0, z_1, z_2\}, \Sigma, \delta, E, z_0)$
- `\liAutomat{alphabet={a,b}}`:  $A = (Z, \{a, b\}, \delta, E, z_0)$
- `\liAutomat{delta=d}`:  $A = (Z, \Sigma, d, E, z_0)$
- `\liAutomat{ende={z_0, z_1, z_2}}`:  $A = (Z, \Sigma, \delta, \{z_0, z_1, z_2\}, z_0)$
- `\liAutomat{start=z_1}`:  $A = (Z, \Sigma, \delta, E, z_1)$
- `\liAutomat{dea}`:  $A_{DEA} = (Z, \Sigma, \delta, E, z_0)$
- `\liAutomat{nea}`:  $A_{NEA} = (Z, \Sigma, \delta, E, z_0)$

```
73 \ExplSyntaxOn
74 \NewDocumentCommand {\liAutomat} { 0{A} m } {
75   \tl_set:Nn \l_zustaende_tl {Z}
76   \tl_set:Nn \l_alphabet_tl {\Sigma}
77   \tl_set:Nn \l_delta_tl {\delta}
78   \tl_set:Nn \l_ende_tl {E}
79   \tl_set:Nn \l_start_tl {z\sb{0}}
80   \tl_set:Nn \l_typ_tl {}
81
82   \keys_define:nn { automat } {
83     zustaende .code:n = {\tl_set:Nn \l_zustaende_tl {\liMenge{##1}}},
84     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
85     delta .code:n = {\tl_set:Nn \l_delta_tl {##1}},
86     ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
```

```

87     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
88     dea .value_forbidden:n = true,
89     dea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{DEA}}}},
90     nea .value_forbidden:n = true,
91     nea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{NEA}}}},
92   }
93
94   \keys_set:nn { automat } { #2 }
95
96   $#1 \l_typ_tl = (
97     \l_zustaende_tl,
98     \l_alphabet_tl,
99     \l_delta_tl,
100    \l_ende_tl,
101    \l_start_tl
102  )$
103 }
104 \ExplSyntaxOff

```

`\liAutomatenKante` **Let-Abkürzung:** `\let\k=\liAutomatenKante`

```

105 \def\liAutomatenKante#1#2#3#4{
106   \path (#1) edge[#4] node{#3} (#2);
107 }

108 \tikzset{
109   li automat/.style={
110     ->,
111     node distance=2cm
112   },
113 }

```

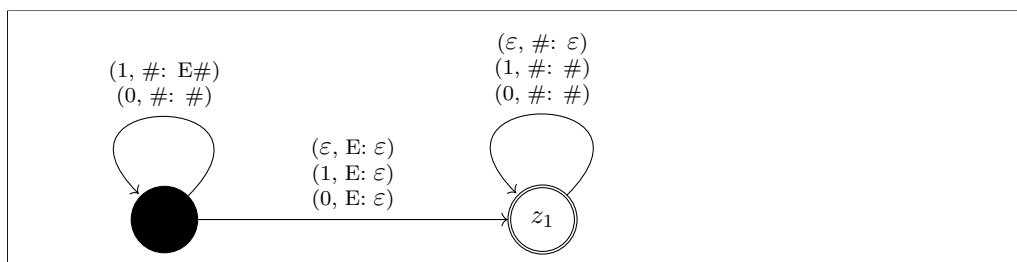
## 2.4.2 Kellerautomat

```
\begin{tikzpicture}[li kellerautomat,node distance=5cm]
\node[state,initial] (0) {\$z_0\$};
\node[state,right of=0,accepting] (1) {\$z_1\$};

\liKellerKante[above,loop]{0}{0}{
  1, KELLERBODEN, E KELLERBODEN;
  0, KELLERBODEN, KELLERBODEN;
}

\liKellerKante[above]{0}{1}{
  EPSILON, E, EPSILON;
  1, E, EPSILON;
  0, E, EPSILON;
}

\liKellerKante[above,loop]{1}{2}{
  EPSILON, KELLERBODEN, EPSILON;
  1, KELLERBODEN, KELLERBODEN;
  0, KELLERBODEN, KELLERBODEN;
}
\end{tikzpicture}
```



```
\liKellerAutomat \liKellerAutomat[⟨automaten-name⟩]
{⟨zustaende=Z,alphabet=Σ,kelleralphabet=Γ,delta=δ,start=z_0,kellerboden=#,ende=E⟩}

\liKellerAutomat{
  zustaeende={z_0, z_1, z_2},
  alphabet={a, b, c},
  kelleralphabet={\#, A},
  ende={z_2},
}
```

$$K = (\{z_0, z_1, z_2\}, \{a, b, c\}, \{\#, A\}, \delta, z_0, \#, \{z_2\})$$

```
114 \ExplSyntaxOn
115 \NewDocumentCommand {\liKellerAutomat} { O{K} m } {
116   \tl_set:Nn \l_zustaende_tl {Z}
117   \tl_set:Nn \l_alphabet_tl {\Sigma}
118   \tl_set:Nn \l_kelleralphabet_tl {\Gamma}
119   \tl_set:Nn \l_delta_tl {\delta}
120   \tl_set:Nn \l_start_tl {z\sb{0}}
121   \tl_set:Nn \l_kellerboden_tl {\#}
122   \tl_set:Nn \l_ende_tl {E}
123
124   \keys_define:nn { kellerautomat } {
125     zustaeende .code:n = {\tl_set:Nn \l_zustaende_tl {\liMenge{##1}}},
126     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
127     kelleralphabet .code:n = {\tl_set:Nn \l_kelleralphabet_tl {\liMenge{##1}}},
128     delta .code:n = {\tl_set:Nn \l_delta_tl {##1}},
129     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
```

```

130     kellerboden .code:n = {\tl_set:Nn \l_kellerboden_tl {##1}},
131     ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
132   }
133
134   \keys_set:nn { kellerautomat } { #2 }
135
136   $#1 = (
137     \l_zustaende_tl,
138     \l_alphabet_tl,
139     \l_kelleralphabet_tl,
140     \l_delta_tl,
141     \l_start_tl,
142     \l_kellerboden_tl,
143     \l_ende_tl
144   )$
145 }
146 \ExplSyntaxOff

```

**\liKellerUebergang** **Let-Abkürzung:** `\let\u=\liKellerUebergang`  
`\liKellerUebergang{a, KELLERBODEN, A; b KELLERBODEN, EPSILON}`

(a, #: A)  
(b, #:  $\varepsilon$ )

Regulärer Ausdruck zum Konvertieren: `\((.*)`, `(.*)`, `(.*)\)` `\u{$1 $2 $3}`

```

147 \ExplSyntaxOn
148 \def\liKellerUebergang#1{
149   \directlua{automaten.drucke_keller_uebergaenge('#1')}
150 }
151 \ExplSyntaxOff

```

**\liKellerKante** `\liKellerKante[\tikz-optionen]{\langle von \rangle{\langle zu \rangle{\langle übergange \rangle`  
**Let-Abkürzung:** `\let\k=\liKellerKante`

```

152 \NewDocumentCommand{\liKellerKante} { 0{above} m m m } {
153   \path (#2) edge[#1] node{\liKellerUebergang{#4}} (#3);
154 }

155 \tikzset{
156   li keller knoten/.style={
157     text width=2cm,
158     align=center,
159     font=\footnotesize,
160   },
161   li kellerautomat/.style={
162     li automat,
163     every edge/.append style={
164       every node/.style={
165         li keller knoten
166       }
167     }
168   }
169 }

```

### 2.4.3 Turingmaschine

```

170 \RequirePackage{amssymb}

```

**\liTuringLeerzeichen**

□

```

171 \def\liTuringLeerzeichen{\Box}

```

```

\liTuringMaschine \liTuringMaschine[\langle automaten-name \rangle]
{\zustaende=Z,alphabet= $\Sigma$ ,bandalphabet= $\Gamma$ ,delta= $\delta$ ,start= $z_0$ ,leerzeichen= $\square$ ,ende=E)}

\liTuringMaschine{
  zustaende={z_0, z_1, z_2},
  alphabet={a, b, c},
  bandalphabet={\liTuringLeerzeichen, A},
  ende={z_2},
}

```

$TM = (\{z_0, z_1, z_2\}, \{a, b, c\}, \{\square, A\}, \delta, z_0, \square, \{z_2\})$
--

```

172 \ExplSyntaxOn
173 \NewDocumentCommand {\liTuringMaschine} { O{TM} m } {
174   \tl_set:Nn \l_zustaende_tl {Z}
175   \tl_set:Nn \l_alphabet_tl {\Sigma}
176   \tl_set:Nn \l_bandalphabet_tl {\Gamma}
177   \tl_set:Nn \l_delta_tl {\delta}
178   \tl_set:Nn \l_start_tl {z\sb{0}}
179   \tl_set:Nn \l_leerzeichen_tl {\liTuringLeerzeichen}
180   \tl_set:Nn \l_ende_tl {E}
181
182   \keys_define:nn { kellerautomat } {
183     zustaende .code:n = {\tl_set:Nn \l_zustaende_tl {\liMenge{##1}}},
184     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
185     bandalphabet .code:n = {\tl_set:Nn \l_bandalphabet_tl {\liMenge{##1}}},
186     delta .code:n = {\tl_set:Nn \l_delta_tl {##1}},
187     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
188     leerzeichen .code:n = {\tl_set:Nn \l_leerzeichen_tl {##1}},
189     ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
190   }
191
192   \keys_set:nn { kellerautomat } { #2 }
193
194   $\text{#1} = (
195     \l_zustaende_tl,
196     \l_alphabet_tl,
197     \l_bandalphabet_tl,
198     \l_delta_tl,
199     \l_start_tl,
200     \l_leerzeichen_tl,
201     \l_ende_tl
202   )$
203 }
204 \ExplSyntaxOff

```

**\liTuringUebergangZelle** Formatiert einen Zustandsübergang für eine Übergangstabelle.

**Let-Abkürzung:** `\let\t=\liTuringUebergangZelle`

`\liTuringUebergangZelle{z_1, LEER, R}:(z_1:  $\square$ , R) \liTuringUebergangZelle{z1, leer, l}:`  
`(z1:  $\square$ , L)`

```

205 \ExplSyntaxOn
206 \def\liTuringUebergangZelle#1{
207   \directlua{tex.print(automaten.gib_einen_turing_uebergang('#1'))}
208 }
209 \ExplSyntaxOff

```

**\liTuringUebergaenge** Nur in den TikZ-Grafiken zu verwenden. Setzt Zeilenumbrüche ans Ende. Nicht für die Tabelle geeignet.

**Let-Abkürzung:** `\let\t=\liTuringUebergaenge`

`(z1:  $\square$ , L)`

`( $\square$ :  $\square$ , R)`



```

210 \ExplSyntaxOn
211 \def\liTuringUebergaenge#1{
212   \directlua{automaten.drucke_turing_uebergaenge('#1')}
213 }
214 \ExplSyntaxOff

\liTuringKante \liTuringKante[\tikz-optionen]{\langle zustand-oder-lese \rangle}{\langle schreibe \rangle}{\langle richtung \rangle}
Let-Abkürzung: \let\t=\liTuringKante

215 \NewDocumentCommand{\liTuringKante} { 0{above} m m m } {
216   \path (#2) edge[#1] node{\liTuringUebergaenge{#4}} (#3);
217 }

\liTuringUeberfuehrung

218 \def\liTuringUeberfuehrung{
219    $\delta : Z \times \Gamma \rightarrow Z \times \Gamma \times \{ L, R, N \}$ 
220 }

221 \tikzset{
222   li turingmaschine/.style={
223     li automat,
224     every edge/.append style={
225       every node/.style={
226         li keller knoten
227       }
228     }
229   }
230 }
231

```

## 2.5 basis.sty

```
232 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
233 \ProvidesPackage{lehramt-informatik-basis}[2020/11/27]

234 \RequirePackage{xparse}

235 \ExplSyntaxOn

\liLadePakete

236 \NewDocumentCommand{\liLadePakete}{ m }
237 {
238   \clist_map_inline:nn { #1 } { \RequirePackage{lehramt-informatik-##1} }
239 }

\liLadeAllePakete

240 \def\liLadeAllePakete{
241   \liLadePakete{
242     aufgaben-einbinden,
243     automaten,
244     baum,
245     checkbox,
246     chomsky-normalform,
247     cpm,
248     cyk-algorithmus,
249     entwurfsmuster,
250     er,
251     formale-sprachen,
252     gantt,
253     grafik,
254     graph,
255     hanoi,
256     kontrollflussgraph,
257     makros,
258     master-theorem,
259     mathe,
260     minimierung,
261     normalformen,
262     petri,
263     potenzmengen-konstruktion,
264     pseudo,
265     quicksort,
266     relationale-algebra,
267     rmodell,
268     sortieren,
269     spalten,
270     struktogramm,
271     syntax,
272     syntaxbaum,
273     synthese-algorithmus,
274     tabelle,
275     typographie,
276     uml,
277     vollstaendige-induktion,
278     wasserfall,
279     wpkalkuel,
280   }
281 }
```

Definition einer Komma-getrennten-Liste mit deren Hilfe die vielen globalen Token-List-Variablen definiert werden können. Die einzelnen Schlüssel sind im Interface AufgabenMetadaten in der Typescript-Datei `.scripts/nodejs/src/aufgaben.ts` erklärt.

```
282 \clist_new:N \g_auf_schluesssel_clist
283 \clist_set:Nn \g_auf_schluesssel_clist {
```

```

284  titel,
285  thematik,
286  stichwoerter,
287  zitat_schluessel,
288  zitat_beschreibung,
289  %
290  bearbeitungs_stand,
291  korrektheit,
292  %
293  relativer_pfad,
294  identische_aufgabe,
295  %
296  examen_nummer,
297  examen_jahr,
298  examen_monat,
299  examen_thema_nr,
300  examen_teilaufgabe_nr,
301  examen_aufgabe_nr,
302 }

```

Initialisierung der globalen Token-List-Variablen `\g_auf_***_tl`. auf steht für Aufgabe.

```

303 \clist_map_inline:Nn \g_auf_schluessel_clist {
304   \tl_new:c {g_auf_#1_tl}
305 }

```

Funktion über alle globalen Token-List-Variablen zurückzusetzen.

```

306 \cs_new:Npn \setze_variablen_zurueck: {
307   \clist_map_inline:Nn \g_auf_schluessel_clist {
308     \tl_clear:c {g_auf_##1_tl}
309   }
310 }

```

Die einzelnen Schlüssel sind im Interface `AufgabenMetadaten` in der Typescript-Datei `scripts/nodejs/src/aufgaben.ts` erklärt.

```

311 \keys_define:nn { aufgabenmetadaten }
312 {
313   Titel .tl_gset:N = \g_auf_titel_tl,
314   Thematik .tl_gset:N = \g_auf_thematik_tl,
315   Stichwoerter .tl_gset:N = \g_auf_stichwoerter_tl,
316   ZitatSchluessel .tl_gset:N = \g_auf_zitat_beschreibung_tl,
317   ZitatBeschreibung .tl_gset:N = \g_auf_zitat_schluessel_tl,
318   %
319   BearbeitungsStand .tl_gset:N = \g_auf_bearbeitungs_stand_tl,
320   Korrektheit .tl_gset:N = \g_auf_korrekttheit_tl,
321   %
322   RelativerPfad .tl_gset:N = \g_auf_relativer_pfad_tl,
323   IdentischeAufgabe .tl_gset:N = \g_auf_identische_aufgabe_tl,
324   %
325   ExamenNummer .tl_gset:N = \g_auf_examen_nummer_tl,
326   ExamenJahr .tl_gset:N = \g_auf_examen_jahr_tl,
327   ExamenMonat .tl_gset:N = \g_auf_examen_monat_tl,
328   ExamenThemaNr .tl_gset:N = \g_auf_examen_thema_nr_tl,
329   ExamenTeilaufgabeNr .tl_gset:N = \g_auf_examen_teilaufgabe_nr_tl,
330   ExamenAufgabeNr .tl_gset:N = \g_auf_examen_aufgabe_nr_tl,
331 }
332 \cs_gset:Npn \setze_relativen_pfad: {
333   \tl_if_empty:NTF \g_auf_relativer_pfad_tl
334   {
335     \bool_if:nTF
336     {
337       ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
338       ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
339       ! \tl_if_empty_p:N \g_auf_examen_monat_tl

```

```

340     }
341     {
342         \tl_gset:Nn \g_auf_relativer_pfad_tl {
343             Staatsexamen /
344             \g_auf_examen_nummer_tl /
345             \g_auf_examen_jahr_tl /
346             \g_auf_examen_monat_tl /
347             \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} { Thema - \g_auf_examen_thema_nr_tl / }
348             \tl_if_empty:NTF \g_auf_examen_teilaufgabe_nr_tl {} { Teilaufgabe - \g_auf_examen_teilaufgabe_nr_tl / }
349             \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} { Aufgabe - \g_auf_examen_aufgabe_nr_tl / }
350         }
351     }
352     {}
353 }
354 {}
355 }

356 \cs_gset:Npn \_gib_examen_titel: {
357     \cs_set:Nn \_trenner: { \, / \, }
358     \bool_if:nTF
359     {
360         ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
361         ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
362         ! \tl_if_empty_p:N \g_auf_examen_monat_tl &&
363         ! \tl_if_empty_p:N \g_auf_examen_aufgabe_nr_tl
364     }
365     {
366         {
367             \footnotesize
368             \par
369             \noindent
370             Staatsexamen ~
371             \g_auf_examen_nummer_tl \_trenner:
372             \g_auf_examen_jahr_tl \_trenner:
373
374             \tl_case:Nn \g_auf_examen_monat_tl
375             {
376                 { 03 } { Frühjahr }
377                 { 09 } { Herbst }
378             } \_trenner:
379
380             \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} {
381                 Thema ~ Nr. ~ \g_auf_examen_thema_nr_tl \_trenner:
382             }
383             \tl_if_empty:NTF \g_auf_examen_teilaufgabe_nr_tl {} {
384                 Teilaufgabe ~ Nr. ~ \g_auf_examen_teilaufgabe_nr_tl \_trenner:
385             }
386             \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} {
387                 Aufgabe ~ Nr. ~ \g_auf_examen_aufgabe_nr_tl
388             }
389             \par
390             \bigskip
391         }
392     }
393 }

394 \cs_new:Npn \_gib_github_url: {
395     \tl_if_empty:NTF \g_auf_relativer_pfad_tl {} {
396         Github :~\href{
397             \LehramtInformatikGithubRawDomain /
398             \LehramtInformatikGithubTexRepo /
399             \LehramtInformatikGitBranch /
400             \g_auf_relativer_pfad_tl
401         }{
402             \nolinkurl{\g_auf_relativer_pfad_tl}

```

```

403     }
404   }
405 }

406 \cs_new:Npn \_gib_aufgaben_titel: {
407   \g_auf_titel_tl
408
409   \tl_if_empty:NTF \g_auf_thematik_tl
410   {}
411   {
412     \, ~ [
413       \g_auf_thematik_tl
414     ]
415   }
416 }

417 \msg_new:nnn { aufgabenmetadaten } { kein-titel }
418   { Der~Schlüssel~ist~zwingend~notwendig }

\def\LehramtInformatikRepository{/pfad/zum/repository}
419 \input /etc/lehramt-informatik.config.tex
    biblatex not working with lualatex and babel
420 % \RequirePackage{polyglossia}
421 % \setmainlanguage{german}
422

```

## 2.6 baum.sty

```
423 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
424 \ProvidesPackage{lehramt-informatik-baum}[2020/06/13 Zum Setzen von
425 Binär- und AVL-Bäumen. Hüll-Paket um TikZ and tikz-qtrees.]

426 \RequirePackage{tikz}

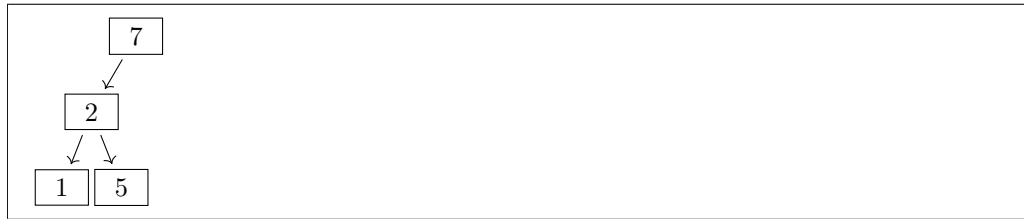
    für li binaer baum
427 \RequirePackage{tikz-qtrees}

    Für b baum
428 \usetikzlibrary{shapes.multipart}
```

### 2.6.1 Binärbaum

TikZ-Stil: li binaer baum: Knoten als Kreise, Kanten als Pfeile.

```
\begin{tikzpicture}[li binaer baum]
\Tree
[.7
  [.2
    [.1 ]
    [.5 ]
  ]
\edge[blank]; \node[blank]{};
]
\end{tikzpicture}
```



```
429 \tikzset{
430   li binaer baum/.style={
431     shorten <=2pt,
432     shorten >=2pt,
433     ->,
434     every tree node/.style={
435       minimum width=2em,
436       draw,
437       rectangle
438     },
439     blank/.style={
440       draw=none
441     },
442     edge from parent/.style={
443       draw,
444       edge from parent path={(\tikzparentnode) -- (\tikzchildnode)}
445     },
446     level distance=1cm,
447     every label/.style={
448       gray,
449       font=\footnotesize,
450       label position=0,
451       label distance=0cm,
452     }
453   },
454 }
```

### 2.6.2 AVL-Baum

```
\begin{tikzpicture}[li binaer baum]
\Tree
[.\node[label=-1]{5};
  [.\node[label=-1]{2};
    [.\node[label=0]{1}; ]
    \edge[blank]; \node[blank]{};
  ]
  [.\node[label=0]{7}; ]
]
\end{tikzpicture}
```



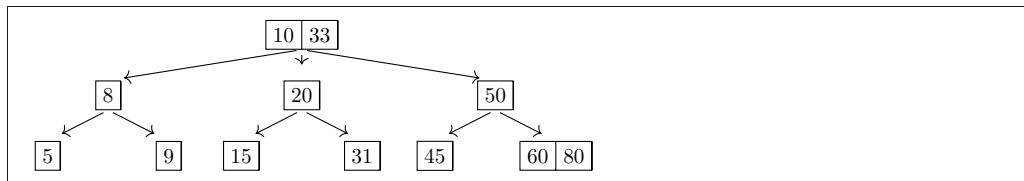


### 2.6.3 B-Baum

```

\begin{tikzpicture}[
  scale=0.8,
  transform shape,
  li bbaum,
  level 1/.style={level distance=10mm,sibling distance=32mm},
  level 2/.style={level distance=10mm,sibling distance=20mm},
]
\node {10 \nodepart{two} 33}
  child {node {8}
    child {node {5}}
    child {node {9}}
  }
  child {node {20}
    child {node {15}}
    child {node {31}}
  }
  child {node {50}
    child {node {45}}
    child {node {60 \nodepart{two} 80}}
  }
;
\end{tikzpicture}

```



```

455 \tikzset{
456   li bbaum knoten/.style={
457     rectangle split parts=10,
458     rectangle split,
459     rectangle split horizontal,
460     rectangle split ignore empty parts,
461     draw,
462     fill=white
463   },
464   li bbaum/.style={
465     every node/.style={
466       li bbaum knoten
467     },
468     level 1/.style={
469       level distance=12mm,
470       sibling distance=25mm,
471     },
472     every child/.style={
473       shorten <= 2pt,
474       shorten >= 6pt,
475       ->,
476     },
477     level 2/.style={
478       level distance=9mm,
479       sibling distance=15mm,
480     },
481   }
482 }
483

```

## 2.7 checkbox.sty

```
484 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
485 \ProvidesPackage{lehramt-informatik-checkbox}[2020/12/14 Zum Setzen von
486 Multiple-Choice-Fragen. Simulation von Kästchen zum ankreuzen]
487 \RequirePackage{amssymb}

\liRichtig Angekreuztes Kästchen (nur innerhalb der itemize-Umgebung zu verwenden.
488 \def\liRichtig{\item[$\boxtimes$]}

\liFalsch Nicht angekreuztes Kästchen (nur innerhalb der itemize-Umgebung zu verwenden.
489 \def\liFalsch{\item[$\square$]}

490
```

## 2.8 chomsky-normalform.sty

```

491 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
492 \ProvidesPackage{lehramt-informatik-chomsky-normalform}[2021/03/26
493 Hilfsmakros zum Setzen des Algorithmus zum Erreichen der Chomsky-Normalform]
494
495 \ExplSyntaxOn
496
497 \liLadePakete{typographie}

\liChomskyUeberschrift \Let-Abkürzung: \let\schritt=\liChomskyUeberschrift

498 \def\liChomskyUeberschrift#1{
499   {
500     \bfseries
501     \sffamily
502     \str_case:nn {#1} {
503       {1} {Elimination~der~$\varepsilon$-Regeln}
504       {2} {Elimination~von~Kettenregeln}
505       {3} {Separation~von~Terminalzeichen}
506       {4} {Elimination~von~mehrelementigen~Nonterminalketten}
507     }
508   }
509 }

\liChomskyErklaerung \Let-Abkürzung: \let\erklaerung=\liChomskyErklaerung
Hoffmann Seite 180

510 \def\li@chomsky@erklaerung@texte#1{
511   \str_case:nn {#1} {
512     %
513     {1} {
514       Alle~Regeln~der~Form~$A\rightarrow\varepsilon$~werden~eliminiert.~
515       Die~Ersetzung~von~$A$~wird~durch~$\varepsilon$~in~allen~anderen~
516       Regeln~vorweggenommen.
517     }
518     {2} {
519       Jede~Produktion~der~Form~$A\rightarrow B$~mit~$A, B\in S$~wird~
520       als~Kettenregel~bezeichnet.~Diese~tragen~nicht~zur~Produktion~
521       von~Terminalzeichen~bei~und~lassen~sich~ebenfalls~eliminieren.
522     }
523     {3} {
524       Jedes~Terminalzeichen~$\sigma$,~das~in~Kombination~mit~anderen~
525       Symbolen~auftaucht,~wird~durch~ein~neues~Nonterminal~
526       $S_{\sigma}$~ersetzt~und~die~Menge~der~Produktionen~durch~die~
527       Regel~$S_{\sigma}\rightarrow\sigma$~ergänzt.
528     }
529     {4} {
530       Alle~Produktionen~der~Form~
531       $A\rightarrow B_{\{1\}}B_{\{2\}}\dots B_{\{n\}}$~
532       werden~in~die~Produktionen~
533       $A\rightarrow
534       A_{\{n-1\}}B_{\{n\}}, A_{\{n-1\}}\rightarrow
535       A_{\{n-2\}}B_{\{n-1\}}, \dots,
536       A_{\{2\}}\rightarrow B_{\{1\}}B_{\{2\}}$~zerteilt.~
537       Nach~der~Ersetzung~sind~alle~längeren~Nonterminalketten~
538       vollständig~heruntergebrochen~und~die~Chomsky-Normalform~erreicht.
539     }
540   }
541 }

542 \def\liChomskyErklaerung#1{
543   {
544     \itshape
545     \footnotesize
546     \liParagraphMitLinien{\li@chomsky@erklaerung@texte{#1}}
547   }

```

548 }

\liChomskyUeberErklaerung **Let-Abkürzung:** \let\schrittE=\liChomskyUeberErklaerung

549 \def\liChomskyUeberErklaerung#1{

550 \liChomskyUeberschrift{#1}\par

551 \liChomskyErklaerung{#1}

552 }

553 \ExplSyntaxOff

554

## 2.9 cpm.sty

```

555 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
556 \ProvidesPackage{lehramt-informatik-cpm}[2020/09/03]
557 \RequirePackage{tikz}
558 \liLadePakete{mathe}

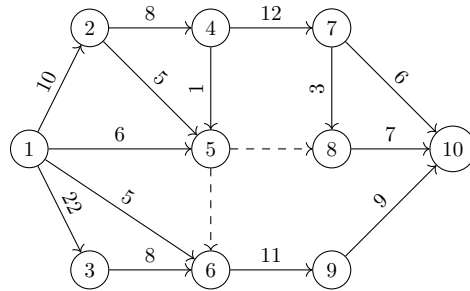
```

### 2.9.1 Makro-Kürzel

```

\let\f=\footnotesize
\let\FZ=\liCpmFruehesterI
\let\SZ=\liCpmSpaetesterI
\let\v=\liCpmVon
\let\ vz=\liCpmVonZu
\let\ z=\liCpmZu

```



```

\begin{tikzpicture}[scale=0.8,transform shape]
\liCpmEreignis{1}{0}{2}
\liCpmEreignis{2}{1}{4}
\liCpmEreignis{3}{1}{0}

\liCpmVorgang{1}{2}{10}
\liCpmVorgang{1}{3}{22}
\liCpmVorgang{1}{5}{6}

\liCpmVorgang[schein]{5}{6}{ }
\liCpmVorgang[schein]{5}{8}{ }
\end{tikzpicture}

```

```

\begin{tabular}{|l|l|l|l|l|l|l|l|}
\hline
\hline
i & a & b & c & d & e & f & g \\
\hline
\FZ & 0 & 2 & 5 & 6 & 10 & 3 & 12 \\
\hline
\SZ & 0 & 2 & 5 & 10 & 10 & 6 & 12 \\
\hline
GP & 0 & 0 & 0 & 3 & 0 & 3 & 0 \\
\hline
\end{tabular}

```

```

\liCpmEreignis \liCpmEreignis\{(.*)\}\{((.*),(.*))\} -> \liCpmEreignis{\$1}{\$2}{\$3}

```

```

559 \ExplSyntaxOn
560 \NewDocumentCommand { \liCpmEreignis } { 0{} m m m } {
561   \tl_set:Nn \l_name_tl {}
562
563   \keys_define:nn { cpmEreignis } {
564     name .code:n = {\tl_set:Nn \l_name_tl {##1}},
565   }
566
567   \keys_set:nn { cpmEreignis } { #1 }
568
569   \tl_if_empty:NT \l_name_tl {
570     \tl_set:Nn \l_name_tl {#2}
571   }

```

```

572
573 \node[circle,draw] (\l_name_tl) at (#3,#4) {#2};
574 }
575 \ExplSyntaxOff

\liCpmVorgang \liCpmVORGANG\((.*)>(.*))\{(.*)\}
576 \ExplSyntaxOn
577 \NewDocumentCommand { \liCpmVorgang } { 0{ } m m m } {
578   \tl_set:Nn \l_schein_tl {}
579   \tl_set:Nn \l_kritisch_tl {}
580
581   \keys_define:nn { cpmVorgang } {
582     schein .code:n = {\tl_set:Nn \l_schein_tl {dashed}},
583     kritisch .code:n = {\tl_set:Nn \l_kritisch_tl {thick}},
584   }
585
586   \keys_set:nn { cpmVorgang } { #1 }
587
588   \path[->,\l_schein_tl,\l_kritisch_tl] (#2) edge node[auto,sloped] {#4} (#3);
589 }
590 \ExplSyntaxOff

```

## 2.9.2 Hilfsmakros für Zeitpunkt-Tabelle

```

\begin{tabular}{|l||l|l|l|l|l|l|l|l|}
\hline
$i$ & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \\
\hline
\FZ & 0 & 5 & 18 & 7 & 19 & 26 & 22 & 30 & \\
\hline
\SZ & 0 & 5 & 18 & 9 & 19 & 26 & 24 & 30 & \\
\hline
GP & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & \\
\hline
\end{tabular}

```

```

\liCpmVonZu Let-Abkürzung: \let\vz=\liCpmVonZu
\liCpmVonZu{1}(2-3):  $l_{(2 \rightarrow 3)}$ 
591 \def\liCpmVonZuOhneMathe#1(#2-#3){#1_{\scriptscriptstyle(#2\rightarrow#3)}}
592 \def\liCpmVonZu#1(#2-#3){%
593   \ifmmode%
594     \liCpmVonZuOhneMathe{#1}(#2-#3)%
595   \else%
596     $\liCpmVonZuOhneMathe{#1}(#2-#3)$%
597   \fi%
598 }

```

```

\liCpmVon Let-Abkürzung: \let\v=\liCpmVon
\liCpmVon{1}(2):  $l_{(\rightarrow 2)}$ 
599 \def\liCpmVonOhneMathe#1(#2){#1_{\scriptscriptstyle(\rightarrow#2)}}
600 \def\liCpmVon#1(#2){%
601   \ifmmode%
602     \liCpmVonOhneMathe{#1}(#2)%
603   \else%
604     $\liCpmVonOhneMathe{#1}(#2)$%
605   \fi%
606 }

```

```

\liCpmZu Let-Abkürzung: \let\z=\liCpmZu
\liCpmZu{1}(2):  $l_{(\leftarrow 2)}$ 
607 \def\liCpmZuOhneMathe#1(#2){#1_{\scriptscriptstyle(\leftarrow#2)}}
608 \def\liCpmZu#1(#2){%
609   \ifmmode%
610     \liCpmZuOhneMathe{#1}(#2)%
611   \else%
612     $\liCpmZuOhneMathe{#1}(#2)$%

```

```

613   \fi%
614 }

\liCpmSpaetesterI Spätester Zeitpunkt, zu dem Ereignis  $i$  eintreten kann
Let-Abkürzung: \let\SZ=\liCpmSpaetesterI
615 \def\liCpmSpaetesterI{$SZ_i$}

\liCpmFruehesterI Frühester Zeitpunkt, zu dem Ereignis  $i$  eintreten kann.
Let-Abkürzung: \let\FZ=\liCpmFruehesterI
616 \def\liCpmFruehesterI{$FZ_i$}

617

```

## 2.10 cyk-algorithmus.sty

```
618 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
619 \ProvidesPackage{lehramt-informatik-cyk-algorithmus}[2021/06/18 Hilfsmakros
620 zum Setzen des CYK-Algorithmus]
```

### 2.10.1 Makro-Kürzel

```
\let\l=\liKurzeTabellenLinie
```

### 2.10.2 TeX-Markup-Beispiel

```
\begin{tabular}{|c|c|c|c|c|c|}
a      & c      & b      & c      & a      & b \\\hline\hline

$R_a$ & $R_c$ & $R_b$ & $R_c$ & $R_a$ & $R_b$ \l6
B      & A      & A      & B      & C      & \l5
S      & -      & S      & S      & \l4
-      & -      & -      & \l3
-      & -      & \l2
S \l1
\end{tabular}
\liWortInSprache{acbcab}
```

`\liKurzeTabellenLinie` **Let-Abkürzung:** `\let\l=\liKurzeTabellenLinie`

```
621 \def\liKurzeTabellenLinie#1{\ccline{1-#1}}
```

`\liWortInSprache` `\liWortInSprache{abc}:  $\Rightarrow abc \in L(G)$`

`\liWortInSprache{abc}[L(Z)]:  $\Rightarrow abc \in L(Z)$`

```
622 \NewDocumentCommand{ \liWortInSprache } { m O{L(G)} } {
623   \bigskip
624   \noindent
625   $\Rrightarrow #1 \in #2$
626 }
```

`\liWortNichtInSprache` `\liWortNichtInSprache{abc}:  $\Rightarrow abc \notin L(G)$`

`\liWortNichtInSprache{abc}[L(Z)]:  $\Rightarrow abc \notin L(Z)$`

```
627 \NewDocumentCommand{ \liWortNichtInSprache } { m O{L(G)} } {
628   \bigskip
629   \noindent
630   $\Rrightarrow #1 \notin #2$
631 }
```

```
632
```



## 2.11 entwurfsmuster.sty

```
633 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
634 \ProvidesPackage{lehramt-informatik-entwurfsmuster}[2021/05/06]
635 Hilfsmakros zum Setzen von Entwurfsmustern / Design Patterns]
```

### 2.11.1 Namensschema der Entwurfsmuster-Makros:

Präfix: \liEntwurfs + Name des Entwurfsmuster DeutscherName + Suffix: (Uml, Akteure, Code, ohne)

### 2.11.2 Reihenfolge

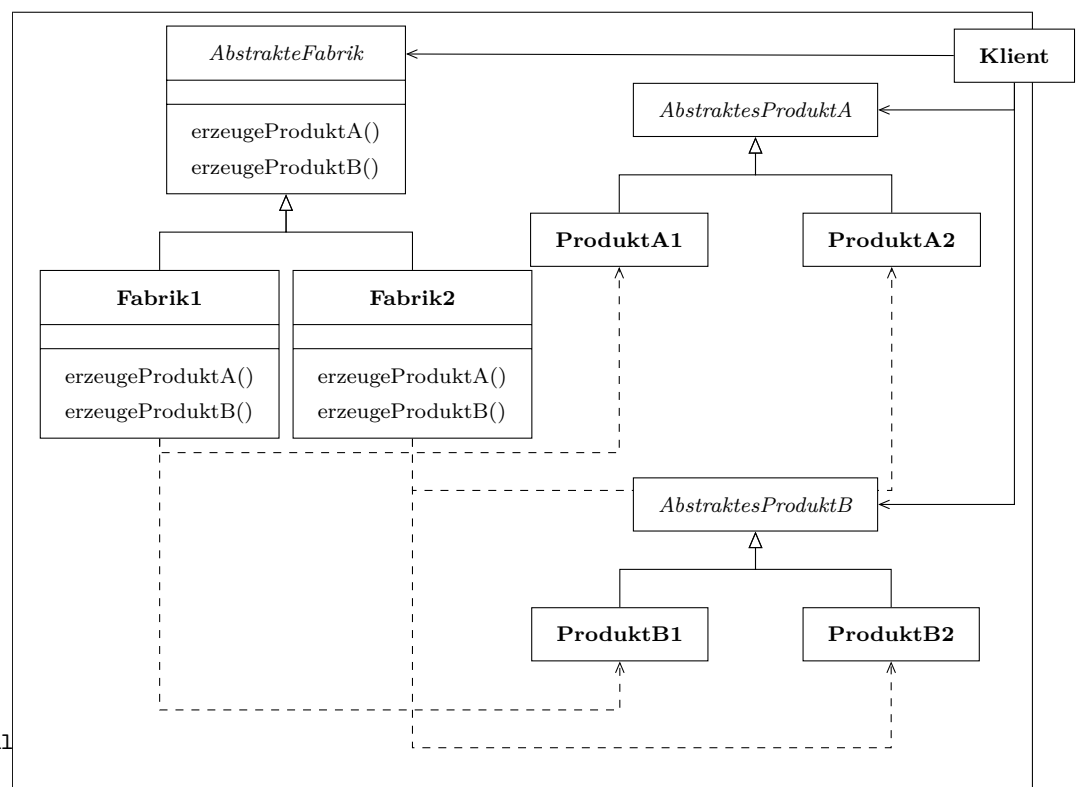
1. Uml: Uml-Klassendiagramm \liEntwurfsEinzelstueckUml
2. Akteure: Akteure, beteiligte Klassen \liEntwurfsEinzelstueckAkteure
3. Code: Allgemeines Code-Beispiel \liEntwurfsEinzelstueckCode
4. ohne: Ohne Suffix, Bündelung der einzelnen Makros eines Entwurfsmusters \liEntwurfsEinzelstueckAkteure

```
636 \RequirePackage{lehramt-informatik-uml}
```

\liEntwurfsCodeAllgemein Allgemeine Code-Beispiele zu den UML-Diagrammen und Stellvertretern

```
637 \RequirePackage{lehramt-informatik-syntax}
638 \def\liEntwurfsCode#1#2{
639   \liJavaDatei{entwurfsmuster/#1/allgemein/#2}
640 }
```

### 2.11.3 Abstrakte Fabrik (Abstract Factory)



```
641 \def\liEntwurfsAbstrakteFabrikUml{
642   \begin{tikzpicture}
643     \umlclass[type=abstract]{AbstrakteFabrik}{-}{-}{
644       erzeugeProduktA()\
645       erzeugeProduktB()\
646     }
```

```

647 \umlclass[below left=1cm and -1.5cm of AbstrakteFabrik]{Fabrik1}{-}{
648     erzeugeProduktA()\
649     erzeugeProduktB()\
650 }
651 \umlclass[below right=1cm and -1.5cm of AbstrakteFabrik]{Fabrik2}{-}{
652     erzeugeProduktA()\
653     erzeugeProduktB()\
654 }
655 \umlVHVinherit{Fabrik1}{AbstrakteFabrik}
656 \umlVHVinherit{Fabrik2}{AbstrakteFabrik}
657
658 \umlsimpleclass[right=3cm of AbstrakteFabrik,type=abstract]{AbstraktesProduktA}
659 \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktA]{ProduktA1}
660 \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktA]{ProduktA2}
661 \umlVHVinherit{ProduktA1}{AbstraktesProduktA}
662 \umlVHVinherit{ProduktA2}{AbstraktesProduktA}
663
664 \umlsimpleclass[above right=0cm and 1cm of AbstraktesProduktA]{Klient}
665
666 \umlsimpleclass[below=4.5cm of AbstraktesProduktA,type=abstract]{AbstraktesProduktB}
667 \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktB]{ProduktB1}
668 \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktB]{ProduktB2}
669 \umlVHVinherit{ProduktB1}{AbstraktesProduktB}
670 \umlVHVinherit{ProduktB2}{AbstraktesProduktB}
671
672 \umlVHVdep[arm1=-1.3cm,arm2=-1cm]{Fabrik1}{ProduktA1}
673 \umlVHVdep[arm2=-1cm]{Fabrik1}{ProduktB1}
674
675 \umlVHVdep[arm1=-1.8cm,arm2=-1.5cm]{Fabrik2}{ProduktA2}
676 \umlVHVdep[arm2=-1.5cm]{Fabrik2}{ProduktB2}
677
678 \umluniassoc[anchor2=25]{Klient}{AbstrakteFabrik}
679 \umlVHuniassoc[arm1=-1cm]{Klient}{AbstraktesProduktA}
680 \umlVHuniassoc{Klient}{AbstraktesProduktB}
681 \end{tikzpicture}
682 }

```

iEntwurfsAbstrakteFabrikCode

```

683 \def\liEntwurfsAbstrakteFabrikCode{
684     \li@EntwurfsCode{abstrakte_fabrik}{Produkte}
685     \li@EntwurfsCode{abstrakte_fabrik}{AbstrakteFabrik}
686     \li@EntwurfsCode{abstrakte_fabrik}{Klient}
687 }

```

\liEntwurfsAbstrakteFabrik

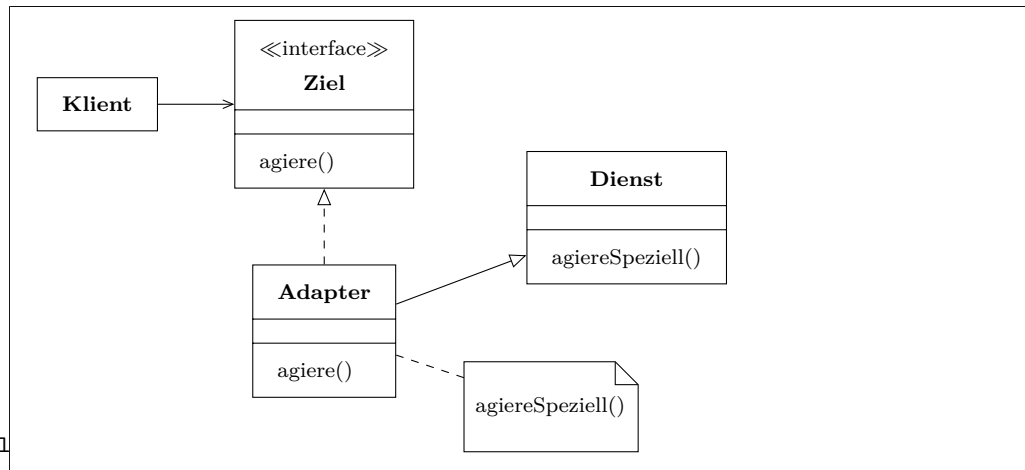
```

688 \def\liEntwurfsAbstrakteFabrik{
689     \liEntwurfsAbstrakteFabrikUml
690     \liEntwurfsAbstrakteFabrikCode
691 }

```

## 2.11.4 Adapter

\liEntwurfsAdapterUml



```

692 \def\liEntwurfsAdapterUml{
693   \begin{tikzpicture}
694     \umlsimpleclass[x=1,y=3]{Klient}{}{}
695     \umlclass[x=4,y=3,type=interface]{Ziel}{}{agiere()}
696     \umlclass[x=4,y=0]{Adapter}{}{agiere()}
697     \umlclass[x=8,y=1.5]{Dienst}{}{agiereSpeziell()}
698
699     \umlreal{Adapter}{Ziel}
700     \umluniassoc{Klient}{Ziel}
701     \umlinherit{Adapter}{Dienst}
702
703     \umlnote[x=7,y=-1,width=2cm]{Adapter}{agiereSpeziell()}
704   \end{tikzpicture}
705   \footcite[so ähnlich wie GoF]{wiki:adapter}
706 }

```

\liEntwurfsAdapterAkteure

**Ziel (Target)** Das Ziel definiert die Schnittstelle, die der Klient nutzen kann.

**Klient (Client)** Der Klient nutzt Dienste über inkompatible Schnittstellen und greift dabei auf adaptierte Schnittstellen zurück.

**Dienst (Adaptee)** Der Dienst bietet wiederzuverwendende Dienstleistungen mit fest definierter Schnittstelle an.

**Adapter** Der Adapter adaptiert die Schnittstelle des Dienstes auf die Schnittstelle zum Klienten.

```

707 \def\liEntwurfsAdapterAkteure{
708   \begin{description}
709
710     \item[Ziel (Target)]
711
712     Das Ziel definiert die Schnittstelle, die der Klient nutzen kann.
713
714     \item[Klient (Client)]
715
716     Der Klient nutzt Dienste über inkompatible Schnittstellen und greift
717     dabei auf adaptierte Schnittstellen zurück.
718
719     \item[Dienst (Adaptee)]
720
721     Der Dienst bietet wiederzuverwendende Dienstleistungen mit fest
722     definierter Schnittstelle an.
723
724     \item[Adapter]
725
726     Der Adapter adaptiert die Schnittstelle des Dienstes auf die
727     Schnittstelle zum Klienten.\footcite{wiki:adapter}

```

```

728
729 \end{description}
730 }

```

\liEntwurfsAdapterCode

```

731 \def\liEntwurfsAdapterCode{
732 \li@EntwurfsCode{adapter}{Dienst}
733 \li@EntwurfsCode{adapter}{Ziel}
734 \li@EntwurfsCode{adapter}{Adapter}
735 \li@EntwurfsCode{adapter}{Klient}
736 }

```

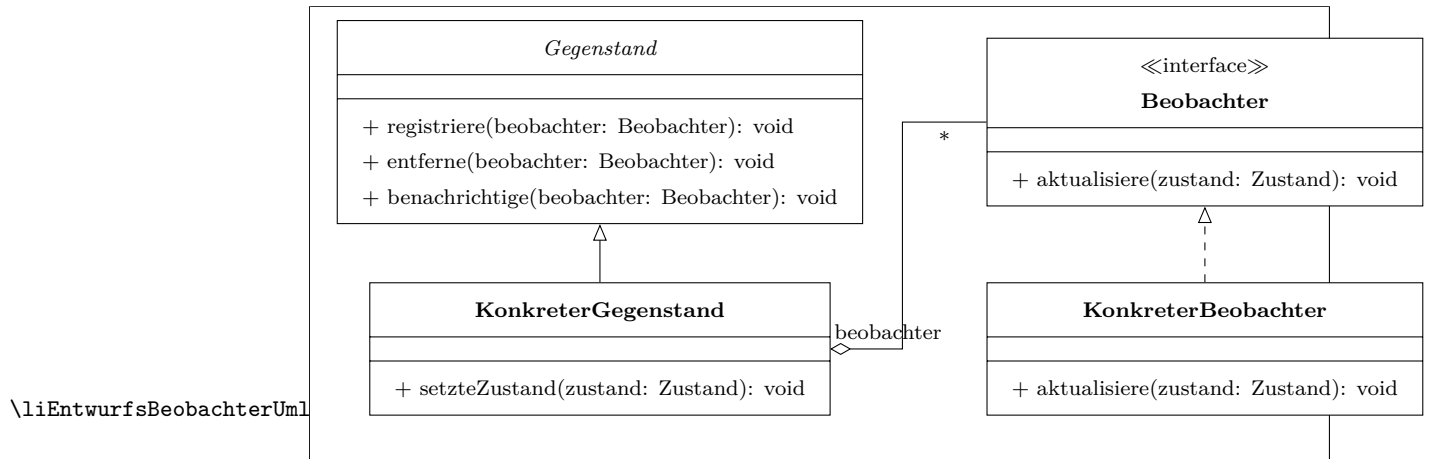
\liEntwurfsAdapter

```

737 \def\liEntwurfsAdapter{
738 \liEntwurfsAdapterUml
739 \liEntwurfsAdapterAkteure
740 \liEntwurfsAdapterCode
741 }

```

### 2.11.5 Beobachter (Observer)



```

742 \def\liEntwurfsBeobachterUml{
743 \begin{tikzpicture}
744 \umlclass[x=0,y=0,type=abstract]{Gegenstand}{-}{
745 + registriere(beobachter: Beobachter): void\\
746 + entferne(beobachter: Beobachter): void\\
747 + benachrichtige(beobachter: Beobachter): void\\
748 }
749 \umlclass[x=0,y=-3]{KonkreterGegenstand}{-}{
750 + setzteZustand(zustand: Zustand): void
751 }
752 \umlinherit{KonkreterGegenstand}{Gegenstand}
753
754 \umlclass[x=8,y=0,type=interface]{Beobachter}{-}{
755 + aktualisiere(zustand: Zustand): void
756 }
757 \umlclass[x=8,y=-3]{KonkreterBeobachter}{-}{
758 + aktualisiere(zustand: Zustand): void
759 }
760 \umlreal{KonkreterBeobachter}{Beobachter}
761
762 \umlHVHagggreg[arg1=beobachter,pos1=0.8,mult2=*,pos2=2.5]
763 {KonkreterGegenstand}{Beobachter}
764 \end{tikzpicture}
765 }

```

\liEntwurfsBeobachterAkteure

**Gegenstand / Subjekt (Subject / Observable)** Ein Subjekt (beobachtbares Objekt, auf Englisch publisher, also „Veröffentlicher“, genannt) hat eine Liste von Beobachtern, ohne deren konkrete Typen zu kennen. Es bietet eine Schnittstelle zur An- und Abmeldung von Beobachtern und eine Schnittstelle zur Benachrichtigung von Beobachtern über Änderungen an.

**Beobachter (Observer)** Die Beobachter (auf Englisch auch subscriber, also „Abonnent“, genannt) definieren eine Aktualisierungsschnittstelle.

**konkreter/s Gegenstand / Subjekt (ConcreteSubject / ConcreteObservable)**

Ein konkretes Subjekt (konkretes, beobachtbares Objekt) speichert den relevanten Zustand und benachrichtigt alle Beobachter bei Zustandsänderungen über deren Aktualisierungsschnittstelle. Es verfügt über eine Schnittstelle zur Erfragung des aktuellen Zustands.

**Konkrete Beobachter (ConcreteObserver)** Konkrete Beobachter verwalten die Referenz auf ein konkretes Subjekt, dessen Zustand sie beobachten und speichern und dessen Zustand konsistent ist. Sie implementieren eine Aktualisierungsschnittstelle unter Verwendung der Abfrageschnittstelle des konkreten Subjekts.

```

766 \def\liEntwurfsBeobachterAkteure{
767   \begin{description}
768     \item[Gegenstand / Subjekt (Subject / Observable)]
769
770     Ein Subjekt (beobachtbares Objekt, auf Englisch publisher, also
771     „Veröffentlicher“, genannt) hat eine Liste von Beobachtern, ohne
772     deren konkrete Typen zu kennen. Es bietet eine Schnittstelle zur An-
773     und Abmeldung von Beobachtern und eine Schnittstelle zur
774     Benachrichtigung von Beobachtern über Änderungen an.\footcite[Seite
775     251]{gof}
776
777     \item[Beobachter (Observer)]
778
779     Die Beobachter (auf Englisch auch subscriber, also „Abonnent“,
780     genannt) definieren eine Aktualisierungsschnittstelle.
781
782     \item[konkreter/s Gegenstand / Subjekt (ConcreteSubject / ConcreteObservable)]
783
784     Ein konkretes Subjekt (konkretes, beobachtbares Objekt) speichert
785     den relevanten Zustand und benachrichtigt alle Beobachter bei
786     Zustandsänderungen über deren Aktualisierungsschnittstelle. Es
787     verfügt über eine Schnittstelle zur Erfragung des aktuellen
788     Zustands.
789
790     \item[Konkrete Beobachter (ConcreteObserver)]
791
792     Konkrete Beobachter verwalten die Referenz auf ein konkretes
793     Subjekt, dessen Zustand sie beobachten und speichern und dessen
794     Zustand konsistent ist. Sie implementieren eine
795     Aktualisierungsschnittstelle unter Verwendung der
796     Abfrageschnittstelle des konkreten Subjekts.
797     \footcite{wiki:beobachter}
798   \end{description}
799 }
```

\liEntwurfsBeobachterCode

```

800 \def\liEntwurfsBeobachterCode{
801   \li@EntwurfsCode{beobachter}{Gegenstand}
802   \li@EntwurfsCode{beobachter}{KonkreterGegenstand}
803   \li@EntwurfsCode{beobachter}{Beobachter}
804   \li@EntwurfsCode{beobachter}{KonkreterBeobachterA}
805   \li@EntwurfsCode{beobachter}{KonkreterBeobachterB}
```

```

806 \li@EntwurfsCode{beobachter}{Klient}
807 }

```

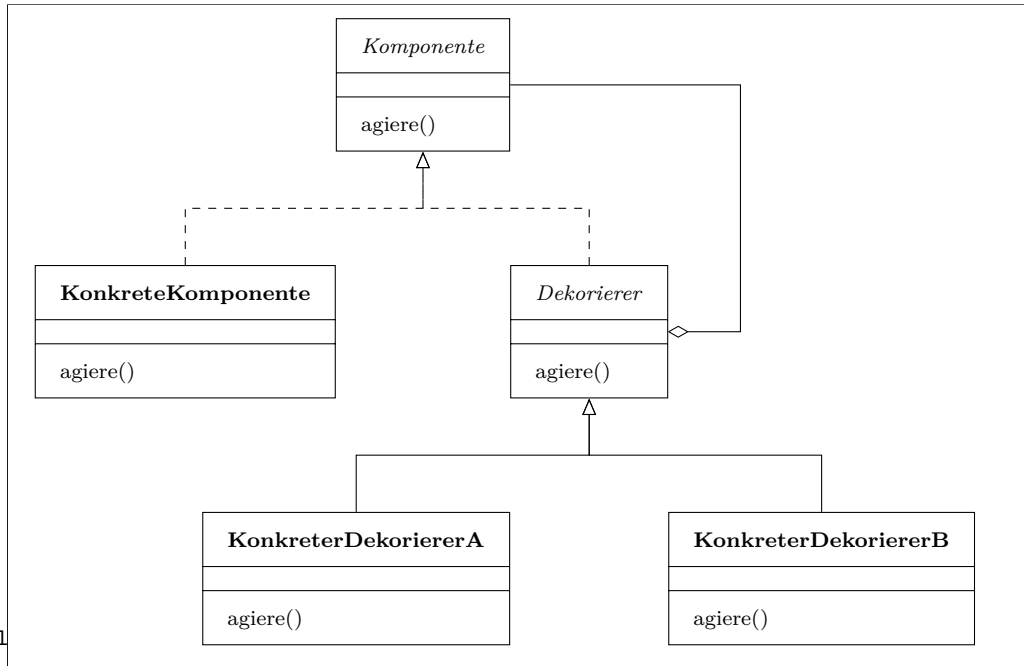
\liEntwurfsBeobachter

```

808 \def\liEntwurfsBeobachter{
809 \liEntwurfsBeobachterUml
810 \liEntwurfsBeobachterAkteure
811 \liEntwurfsBeobachterCode
812 }

```

### 2.11.6 Dekorierer (Decorator)



\liEntwurfsDekoriererUml

```

813 \def\liEntwurfsDekoriererUml{
814 \begin{tikzpicture}
815 \umlclass[type=abstract]{Komponente}{-}{agiere()}
816 \umlclass[below left=1.5cm and 0cm of Komponente]{KonkreteKomponente}{-}{agiere()}
817 \umlclass[below right=1.5cm and 0cm of Komponente,type=abstract]{Dekorierer}{-}{agiere()}
818
819 \umlVHVreal{KonkreteKomponente}{Komponente}
820 \umlVHVreal{Dekorierer}{Komponente}
821
822 \umlclass[below left=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererA}{-}{agiere()}
823 \umlclass[below right=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererB}{-}{agiere()}
824
825 \umlVHVinherit{KonkreterDekoriererA}{Dekorierer}
826 \umlVHVinherit{KonkreterDekoriererB}{Dekorierer}
827
828 \umlHVVHaggreg[arm1=2cm]{Dekorierer}{Komponente}
829 \footcite{wiki:dekorierer}
830 \end{tikzpicture}
831 }

```

\liEntwurfsDekoriererCode

```

832 \def\liEntwurfsDekoriererCode{
833 \li@EntwurfsCode{dekorierer}{Komponente}
834 \li@EntwurfsCode{dekorierer}{KonkreteKomponente}
835 \li@EntwurfsCode{dekorierer}{Dekorierer}
836 \li@EntwurfsCode{dekorierer}{KonkreterDekoriererA}
837 \li@EntwurfsCode{dekorierer}{KonkreterDekoriererB}
838 \li@EntwurfsCode{dekorierer}{Klient}
839 }

```

\liEntwurfsDekorierer

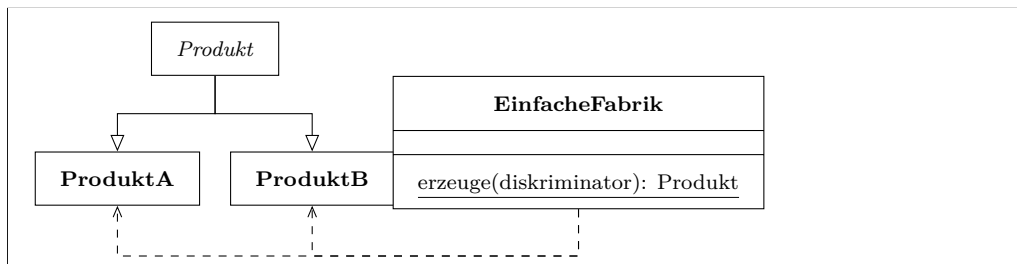
```

840 \def\liEntwurfsDekorierer{
841   \liEntwurfsDekoriererUml
842   \liEntwurfsDekoriererAkteure
843   \liEntwurfsDekoriererCode
844 }

```

## 2.11.7 Einfache Fabrik (Simple Factory)

\liEntwurfsEinfacheFabrikUml Quelle: <https://refactoring.guru/design-patterns/factory-comparison>



```

845 \def\liEntwurfsEinfacheFabrikUml{
846   \begin{tikzpicture}
847     \umlsimpleclass[type=abstract]{Produkt}
848     \umlsimpleclass[below left=1cm and -0.65cm of Produkt]{ProduktA}
849     \umlsimpleclass[below right=1cm and -0.65cm of Produkt]{ProduktB}
850     \umlVHVinherit{Produkt}{ProduktA}
851     \umlVHVinherit{Produkt}{ProduktB}
852     \umlclass[below right=0cm and 1.5cm of Produkt]{EinfacheFabrik}{
853     }{
854       \umlstatic{erzeuge(diskriminator): Produkt}\\
855     }
856     \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktA}
857     \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktB}
858   \end{tikzpicture}
859 }

```

\liEntwurfsEinfacheFabrikAkteure Quelle: frei überstetzt aus GoF

**EinfacheFabrik** Eine Klasse mit einer Erzeugungsmethode, die über eine größere Bedingung verschiedene Objekt instanziiert.

**Produkt** Eine abstrakte Klasse, die von den konkreten Produkten geerbt wird.

**KonkretesProdukt** Ein konkretes Produkt, das von der einfachen Fabrik erzeugt wird.

```

860 \def\liEntwurfsEinfacheFabrikAkteure{
861   \begin{description}
862     \item[EinfacheFabrik]
863
864     Eine Klasse mit einer Erzeugungsmethode, die über eine größere
865     Bedingung verschiedene Objekt instanziiert.
866
867     \item[Produkt]
868
869     Eine abstrakte Klasse, die von den konkreten Produkten geerbt wird.
870
871     \item[KonkretesProdukt]
872
873     Ein konkretes Produkt, das von der einfachen Fabrik erzeugt wird.
874   \end{description}
875 }

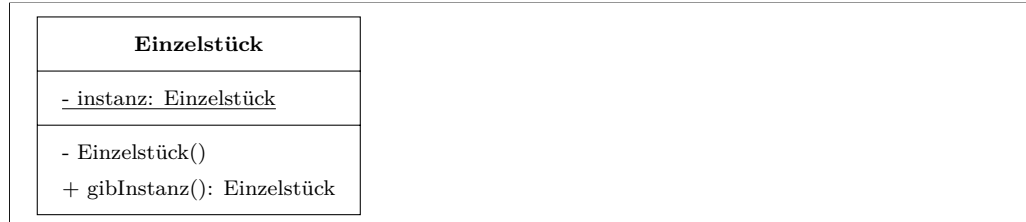
```

\liEntwurfsEinfacheFabrik

```
876 \def\liEntwurfsEinfacheFabrik{
877   \liEntwurfsEinfacheFabrikUml
878   \liEntwurfsEinfacheFabrikAkteure
879 }
```

### 2.11.8 Einzelstück (Singleton)

\liEntwurfsEinzelstueckUml Quelle nach der deutschen Wikipedia



```
880 \def\liEntwurfsEinzelstueckUml{
881   \begin{tikzpicture}
882     \umlclass{Einzelstück}{
883       \umlstatic{- instanz: Einzelstück}\\
884     }{
885       - Einzelstück()\\
886       + gibInstanz(): Einzelstück
887     }
888   \end{tikzpicture}
889 }
```

iEntwurfsEinzelstueckAkteure Quelle: frei überstetzt aus GoF

<b>Einzelstück (Singleton)</b> stellt eine statische Methode bereit, mit deren Hilfe die Klienten nur auf eine einzige Instanz der Klasse zugreifen können.
---

```
890 \def\liEntwurfsEinzelstueckAkteure{
891   \begin{description}
892     \item[Einzelstück (Singleton)]
893
894     stellt eine statische Methode bereit, mit deren Hilfe die Klienten
895     nur auf eine einzige Instanz der Klasse zugreifen können.
896   \end{description}
897 }
```

\liEntwurfsEinzelstueckCode

```
898 \def\liEntwurfsEinzelstueckCode{
899   \li@EntwurfsCode{einzelstueck}{Einzelstueck}
900 }
```

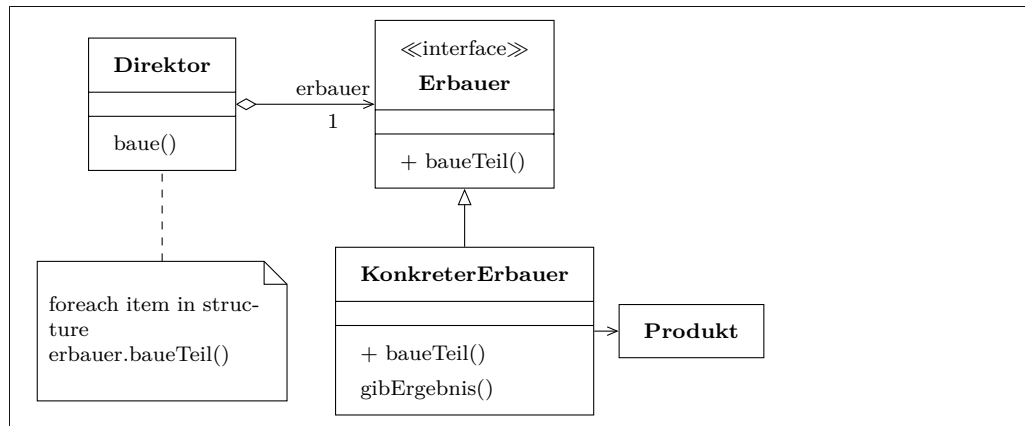
\liEntwurfsEinzelstueck

```
901 \def\liEntwurfsEinzelstueck{
902   \liEntwurfsEinzelstueckUml
903   \liEntwurfsEinzelstueckAkteure
904   \liEntwurfsEinzelstueckCode
905 }
```

### 2.11.9 Erbauer (Builder)

\liEntwurfsErbauerUml Quelle nach der deutschen Wikipedia





```

906 \def\liEntwurfsErbauerUml{
907   \begin{tikzpicture}
908     \umlsimpleclass[x=7,y=0]{Produkt}{}{}
909     \umlclass[x=4,y=3,type=interface]{Erbauer}{}{+ baueTeil()}
910     \umlclass[x=4,y=0]{KonkreterErbauer}{}{
911       + baueTeil()\
912       gibErgebnis()}
913     \umlclass[x=0,y=3]{Direktor}{}{baue()}
914
915     \umluniagg[arg2=erbauer,pos2=0.7,mult2=1]{Direktor}{Erbauer}
916     \umluniassoc{KonkreterErbauer}{Produkt}
917     \umlinherit{KonkreterErbauer}{Erbauer}
918
919     \umlnote[x=0,y=0,width=3cm]{Direktor}{
920       foreach item in structure\
921       erbauer.baueTeil()
922     }
923   \end{tikzpicture}
924   \footcite{wiki:erbauer}
925 }

```

\liEntwurfsErbauerAkteure Quelle: deutsche Wikipedia

**Erbauer** Der Erbauer spezifiziert eine abstrakte Schnittstelle zur Erzeugung der Teile eines komplexen Objektes.

**KonkreterErbauer** Der konkrete Erbauer erzeugt die Teile des komplexen Objekts durch Implementierung der Schnittstelle. Außerdem definiert und verwaltet er die von ihm erzeugte Repräsentation des Produkts. Er bietet auch eine Schnittstelle zum Auslesen des Produkts.

**Direktor** Der Direktor konstruiert ein komplexes Objekt unter Verwendung der Schnittstelle des Erbauers. Der Direktor arbeitet eng mit dem Erbauer zusammen: Er weiß, welche Baureihenfolge der Erbauer verträgt oder benötigt. Der Direktor entkoppelt somit den Konstruktionsablauf vom Klienten.

**Produkt** Das Produkt repräsentiert das zu konstruierende komplexe Objekt.

```

926 \def\liEntwurfsErbauerAkteure{
927   \begin{description}
928     \item[Erbauer]
929
930     Der Erbauer spezifiziert eine abstrakte Schnittstelle zur Erzeugung der
931     Teile eines komplexen Objektes.
932
933     \item[KonkreterErbauer]
934
935     Der konkrete Erbauer erzeugt die Teile des komplexen Objekts durch
936     Implementierung der Schnittstelle. Außerdem definiert und verwaltet er

```

```

937     die von ihm erzeugte Repräsentation des Produkts. Er bietet auch eine
938     Schnittstelle zum Auslesen des Produkts.
939
940     \item[Direktor]
941
942     Der Direktor konstruiert ein komplexes Objekt unter Verwendung der
943     Schnittstelle des Erbauers. Der Direktor arbeitet eng mit dem Erbauer
944     zusammen: Er weiß, welche Baureihenfolge der Erbauer verträgt oder
945     benötigt. Der Direktor entkoppelt somit den Konstruktionsablauf vom
946     Klienten.
947
948     \item[Produkt]
949
950     Das Produkt repräsentiert das zu konstruierende komplexe Objekt.
951     \footcite{wiki:erbauer}
952 \end{description}
953 }

```

\liEntwurfsErbauer

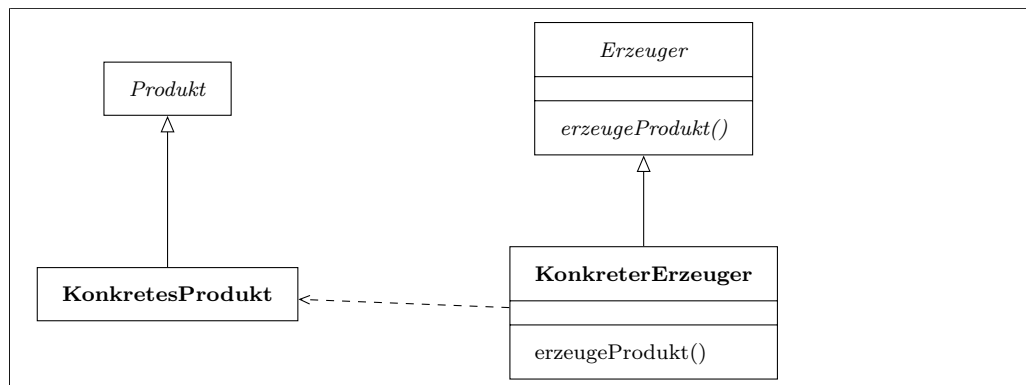
```

954 \def\liEntwurfsErbauer{
955   \liEntwurfsErbauerUml
956   \liEntwurfsErbauerAkteure
957 }

```

#### 2.11.10 Fabrikmethode (Factory Method)

\liEntwurfsFabrikmethodeUml Quelle nach der deutschen Wikipedia



```

958 \def\liEntwurfsFabrikmethodeUml{
959   \begin{tikzpicture}
960     \umlsimpleclass[type=abstract]{Produkt}
961     \umlsimpleclass[below=2cm of Produkt]{KonkretesProdukt}
962     \umlinherit{KonkretesProdukt}{Produkt}
963
964     \umlclass[type=abstract,right=4cm of Produkt]{Erzeuger}{\{
965       \textit{erzeugeProdukt()}\}
966     }
967     \umlclass[below=1.2cm of Erzeuger]{KonkreterErzeuger}{\{
968       erzeugeProdukt()
969     }
970     \umlinherit{KonkreterErzeuger}{Erzeuger}
971
972     \umldep{KonkreterErzeuger}{KonkretesProdukt}
973   \end{tikzpicture}
974 }

```

EntwurfsFabrikmethodeAkteure Quelle: deutsche Wikipedia

**Produkt** Das Produkt ist der Basistyp (Klasse oder Schnittstelle) für das zu erzeugende Produkt.

**KonkretesProdukt** KonkretesProdukt implementiert die Produkt-Schnittstelle.

**Erzeuger** Der Erzeuger deklariert die Fabrikmethode, um ein solches Produkt zu erzeugen und kann eine Default-Implementierung beinhalten.

**KonkreterErzeuger** KonkreterErzeuger überschreibt die Fabrikmethode, um die ihm entsprechenden konkreten Produkte zu erzeugen (z. B. indem er den Konstruktor einer konkreten Produkt-Klasse aufruft).

```

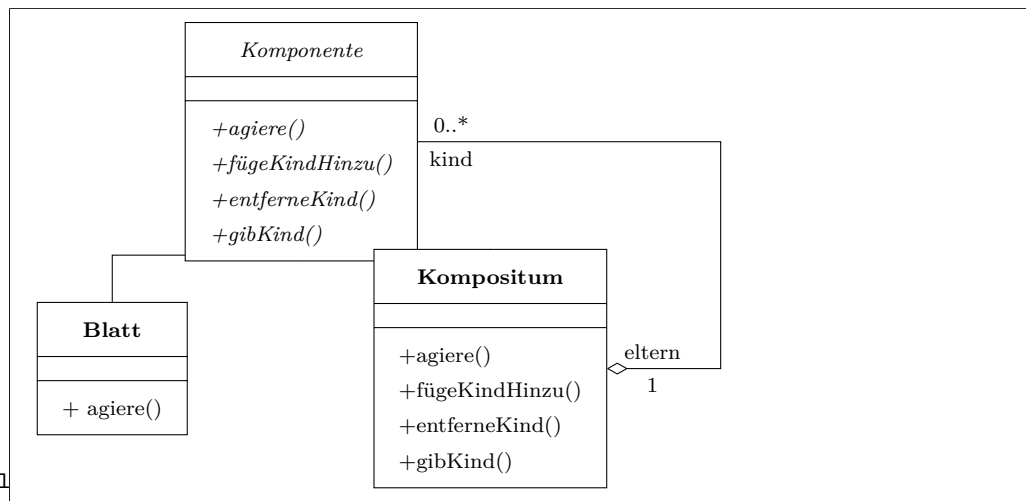
975 \def\liEntwurfsFabrikmethodeAkteure{
976   \begin{description}
977     \item[Produkt]
978
979     Das Produkt ist der Basistyp (Klasse oder Schnittstelle) für das
980     zu erzeugende Produkt.
981
982     \item[KonkretesProdukt]
983
984     KonkretesProdukt implementiert die Produkt-Schnittstelle.
985
986     \item[Erzeuger]
987
988     Der Erzeuger deklariert die Fabrikmethode, um ein solches Produkt
989     zu erzeugen und kann eine Default-Implementierung beinhalten.
990
991     \item[KonkreterErzeuger]
992
993     KonkreterErzeuger überschreibt die Fabrikmethode, um die ihm
994     entsprechenden konkreten Produkte zu erzeugen (z. B. indem er den
995     Konstruktor einer konkreten Produkt-Klasse aufruft).
996
997     \footcite{wiki:fabrikmethode}
998   \end{description}
999 }
```

\liEntwurfsFabrikmethode

```

1000 \def\liEntwurfsFabrikmethode{
1001   \liEntwurfsFabrikmethodeUml
1002   \liEntwurfsFabrikmethodeAkteure
1003 }
```

### 2.11.11 Kompositum (Composite)



\liEntwurfsKompositumUml

```

1004 \def\liEntwurfsKompositumUml{
1005   \begin{tikzpicture}
1006     \umlclass[x=2.5,y=3,type=abstract]{Komponente}{-}{-}{
1007       \textit{+agiere()}\
1008       \textit{+fügeKindHinzu()}\
1009       \textit{+entferneKind()}\
1010       \textit{+gibKind()}
1011     }
1012     \umlclass[x=0]{Blatt}{-}{+ agiere()}
1013     \umlclass[x=5]{Kompositum}{-}{-}{
1014       +agiere()\
1015       +fügeKindHinzu()\
1016       +entferneKind()\
1017       +gibKind()
1018     }
1019
1020     \umlVHVinherit{Kompositum}{Komponente}
1021     \umlVHVinherit{Blatt}{Komponente}
1022     \umlHVHaggreg[anchor1=east,arm1=1.5cm,arg1=eltern,mult1=1,arg2=kind,mult2=0..*,pos2=2.9,]
1023   \end{tikzpicture}
1024 }

```

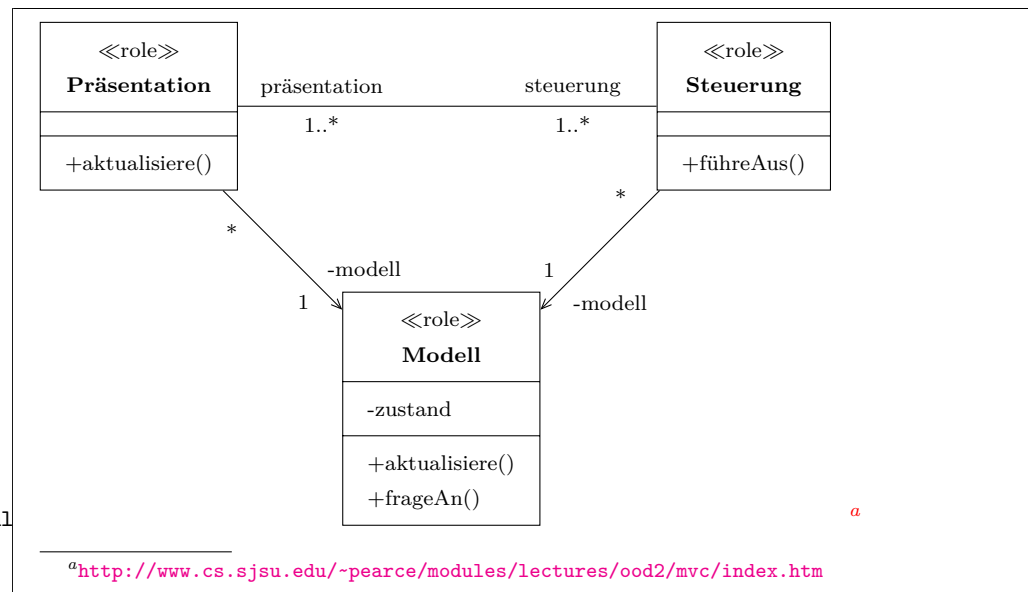
\liEntwurfsFabrikmethode

```

1025 \def\liEntwurfsKompositum{
1026   \liEntwurfsKompositumUml
1027   \liEntwurfsKompositumAkteure
1028 }

```

## 2.11.12 Modell-Präsentation-Steuerung (Model-View-Controller)



```

1029 \def\liEntwurfsModellPraesentationSteuerungUml{
1030   \begin{tikzpicture}
1031     \umlclass[x=-4cm,type=role]{Präsentation}{-}{+aktualisiere()}
1032     \umlclass[x=4cm,type=role]{Steuerung}{-}{+führeAus()}
1033     \umlclass[y=-4cm,type=role]{Modell}{-}{
1034       -zustand
1035     }{
1036       +aktualisiere()\
1037       +frageAn()
1038     }
1039
1040     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Präsentation}{Modell}
1041     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Steuerung}{Modell}

```

```

1042 \umlassoc[arg1=präsentation,mult1=1..*,arg2=steuerung,mult2=1..*]{Präsentation}{Steuerung}
1043 \end{tikzpicture}
1044 \liFussnoteUrl{http://www.cs.sjsu.edu/~pearce/modules/lectures/ood2/mvc/index.htm}
1045 }

```

ModellPraesentationSteuerung

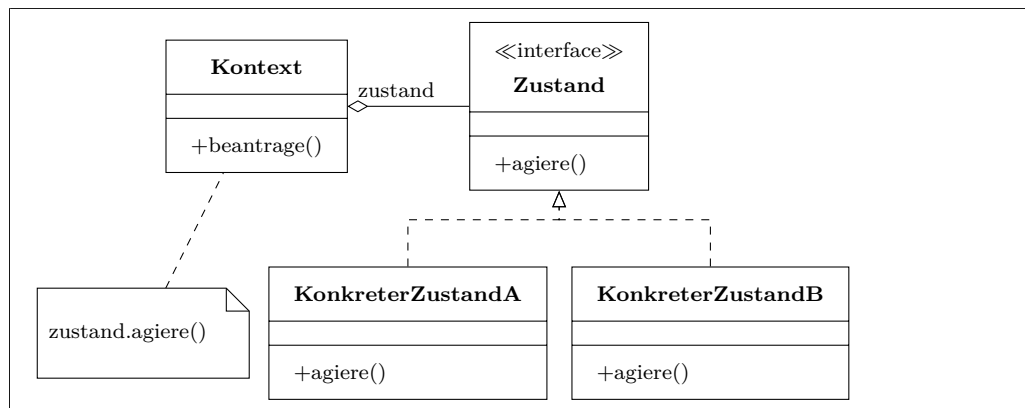
```

1046 \def\liEntwurfs{
1047 \liEntwurfsModellPraesentationSteuerungUml
1048 \liEntwurfsModellPraesentationSteuerungAkteure
1049 }

```

### 2.11.13 Zustand (State)

\liEntwurfsZustandUml Quelle: Englische Wikipedia, so ähnlich wie in GoF



```

1050 \def\liEntwurfsZustandUml{
1051 \begin{tikzpicture}
1052 \umlcclass[x=-1,y=3]{Kontext}{+beantrage()}
1053 \umlcclass[x=3,y=3,type=interface]{Zustand}{+agiere()}
1054 \umlcclass[x=1,y=0]{KonkreterZustandA}{+agiere()}
1055 \umlcclass[x=5,y=0]{KonkreterZustandB}{+agiere()}
1056
1057 \umlVHVreal{KonkreterZustandA}{Zustand}
1058 \umlVHVreal{KonkreterZustandB}{Zustand}
1059
1060 \umlagg[reg[arg=zustand,pos=0.4]{Kontext}{Zustand}
1061
1062 \umlnote[x=-2.5,y=0,width=2.5cm]{Kontext}{zustand.agiere()}
1063 \end{tikzpicture}
1064 }

```

\liEntwurfsZustandAkteure Quelle: Deutsche Wikipedia

**Kontext (Context)** definiert die clientseitige Schnittstelle und verwaltet die separaten Zustandsklassen.

**State (Zustand)** definiert eine einheitliche Schnittstelle aller Zustandsobjekte und implementiert gegebenenfalls ein Standardverhalten.

**KontreterZustand (ConcreteState)** implementiert das Verhalten, das mit dem Zustand des Kontextobjektes verbunden ist.

```

1065 \def\liEntwurfsZustandAkteure{
1066 \begin{description}
1067 \item[Kontext (Context)]
1068
1069 definiert die clientseitige Schnittstelle und verwaltet die separaten
1070 Zustandsklassen.
1071

```

```

1072 \item[State (Zustand)]
1073
1074 definiert eine einheitliche Schnittstelle aller Zustandsobjekte und
1075 implementiert gegebenenfalls ein Standardverhalten.
1076
1077 \item[KontreterZustand (ConcreteState)]
1078
1079 implementiert das Verhalten, das mit dem Zustand des Kontextobjektes
1080 verbunden ist.
1081 \end{description}
1082 }

\liEntwurfsZustand
1083 \def\liEntwurfsZustand{
1084 \liEntwurfsZustandUml
1085 \liEntwurfsZustandAkteure
1086 }

1087

```

## 2.12 er.sty

```
1088 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1089 \ProvidesPackage{lehramt-informatik-er}[2020/06/13 Zum Setzen von
1090 ER-Diagrammen]
```

```
1091 \RequirePackage{tikz-er2}
1092 \usetikzlibrary{positioning}
```

### 2.12.1 Tex-Markup-Beispiel: Komplettes Diagramm

```
\begin{tikzpicture}[er2,scale=0.7,transform shape]
% Person
\node[entity] (Person) {Person};
\node[attribute,right=1cm of Person] {\key{E-Mail}} edge (Person);
\node[multi attribute,above left=1cm of Person] {Vornamen} edge (Person);
\node[attribute,left=1cm of Person] {Nachnamen} edge (Person);
\node[attribute,above right=1cm of Person] {Geburtsdatum} edge (Person);

% Kunde
\node[entity,below left=1cm of Person] (Kunde) {Kunde};

% Händler
\node[entity,below right=1cm of Person] (Händler) {Händler};

\node[specialization,below=0.2cm of Person]{is-a}
  edge (Kunde) edge (Händler) edge (Person);

% Transaktion
\node[relationship,below=2cm of Person] (Transaktion) {Transaktion}
  edge node[auto]{1} (Kunde)
  edge node[auto]{1} (Händler);
\node[attribute,below=1cm of Transaktion] {Preis} edge (Transaktion);
\node[attribute,left=1cm of Transaktion,text width=2cm] {Abschlussdatum} edge (Transaktion);
\node[attribute,right=1cm of Transaktion] {Lieferadresse} edge (Transaktion);

% Zahlungsmittel
\node[entity,below=4cm of Händler] (Zahlungsmittel) {Zahlungsmittel}
  edge node[auto]{1} (Transaktion);
\node[attribute,right=1cm of Zahlungsmittel] {Inhaber} edge (Zahlungsmittel);

% Bankverbindung
\node[entity,below left=1cm and 0cm of Zahlungsmittel] (Bankverbindung) {Bankverbindung};
\node[attribute,below left=1cm of Bankverbindung] {\key{IBAN}} edge (Bankverbindung);

% Kreditkarte
\node[entity,below right=3cm and -2cm of Zahlungsmittel]
  (Kreditkarte) {Kreditkarte};

\node[attribute,below left=1cm of Kreditkarte]
  {\key{Nummer}} edge (Kreditkarte);

\node[attribute,below right=1cm of Kreditkarte,text width=2cm]
  {Ablaufdatum} edge (Kreditkarte);

\node[attribute,right=1cm of Kreditkarte]
  {Anbieter} edge (Kreditkarte);

\node[generalization,below=0.8cm of Zahlungsmittel]{is-a}
  edge (Zahlungsmittel) edge (Bankverbindung) edge (Kreditkarte);

% Produkt
\node[entity,below=4cm of Kunde] (Produkt) {Produkt}
  edge node[auto]{1} (Transaktion);
\node[attribute,left=1cm of Produkt] {\key{Bezeichnung}} edge (Produkt);
\node[attribute,below left=1cm of Produkt,text width=2cm] {Beschreibungstext} edge (Produkt);
```

```

\node[attribute,above left=1cm of Produkt] {Bewertung} edge (Produkt);
\end{tikzpicture}

```

### 2.12.2 Tex-Markup-Beispiel: EER Enhanced Entity-Relation-Modell nach Elmasri/Navante

```

\node[below=1cm of Forscher,circle,draw] (union) {u}
  edge (Sekretär) edge (Forscher) edge (Techniker);

\node[below=1cm of union,entity] (AktiveMitarbeiter) {Aktive Mitarbeiter}
  edge node {$\bigcup$} (union);

```

```
1093 \RequirePackage{soul}
```

```
1094 \RequirePackage{fontawesome}
```

### 2.12.3 Makro-Kürzel

```

\let\a=\liErMpAttribute
\let\d=\liErDatenbankName
\let\e=\liErMpEntity
\let\r=\liErMpRelationship

```

```
1095 \ExplSyntaxOn
```

```
\liErEntity
```

```
1096 \def\liErEntity#1{\textbf{#1}}
```

```
\liErRelationship
```

```
1097 \def\liErRelationship#1{\ul{#1}}
```

```
\liErAttribute
```

```
1098 \def\liErAttribute#1{\emph{#1}}
```

```
\liErMpEntity mp = marginpar
```

**Let-Abkürzung:** \let\e=\liErMpEntity

```

1099 \def\liErMpEntity#1{
1100   \liErEntity{#1}
1101   \marginpar{
1102     \liErEntity{\tiny\faSquareO{~E:~#1}
1103   }
1104 }

```

□

```
\liErMpRelationship Let-Abkürzung: \let\r=\liErMpRelationship
```

```

1105 \def\liErMpRelationship#1{
1106   \liErRelationship{#1}
1107   \marginpar{
1108     \liErRelationship{\tiny\faGg{~R:~#1}
1109   }
1110 }

```

```
\liErMpAttribute Let-Abkürzung: \let\a=\liErMpAttribute
```

```

1111 \def\liErMpAttribute#1{
1112   \liErAttribute{#1}
1113   \marginpar{
1114     \liErAttribute{\tiny\faCircleThin{~A:~#1}
1115   }
1116 }

```



```

\liErDatenbankName Let-Abkürzung: \let\d=\liErDatenbankName
                    datenbank name
1117 \def\liErDatenbankName#1{
1118   {
1119     \footnotesize\texttt{(#1)}
1120   }
1121 }

1122 \ExplSyntaxOff
1123

```

## 2.13 formale-sprachen.sty

```

1124 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1125 \ProvidesPackage{lehramt-informatik-formale-sprachen}[2021/02/21 Hilfsmakros
1126 zum Setzen von mathematischen Formeln bei dem Thema Formale Sprachen]

1127 \directlua{
1128   formale_sprachen = require('lehramt-informatik-formale-sprachen')
1129 }

1130 \RequirePackage{hyperref}

1131 \liLadePakete{mathe,typographie}

\liMenge   $\liMenge{a, b, c}$:  $\{a, b, c\}$ 
Let-Abkürzung: \let\m=\liMenge

1132 \def\liMengeOhneMathe#1{\{ #1 \}}
1133 \def\liMenge#1{%
1134   \ifmode%
1135     \liMengeOhneMathe{#1}%
1136   \else%
1137     $\liMengeOhneMathe{#1}$%
1138   \fi%
1139 }

\liEpsilon \liEpsilon:  $\varepsilon$ 
Let-Abkürzung: \let\e=\liEpsilon

1140 \def\liEpsilon{$\varepsilon$}

\liPotenzmenge Umgeben mit geschweiften Klammern in einer Mathematik-Umgebung

1141 \def\erzeuge@tiefgestellt#1{\directlua{formale_sprachen.erzeuge_tiefgestellt('#1')}}
1142 \def\liPotenzmengeOhneMathe#1{\{ \erzeuge@tiefgestellt{#1} \}}
1143 \def\liPotenzmenge#1{$\liPotenzmengeOhneMathe{#1}$}

\liZustandsmenge \liZustandsmenge{z1, z2}:  $\{z_1, z_2\}$ 

1144 \let\liZustandsmengeOhneMathe=\liPotenzmengeOhneMathe
1145 \let\liZustandsmenge=\liPotenzmenge

\liUeberfuehrungsFunktion \liUeberfuehrungsFunktion{z0, a}:  $\delta(z_0, a)$ 
Let-Abkürzung: \let\d=\liUeberfuehrungsFunktion

1146 \def\liUeberfuehrungsFunktionOhneMathe#1{\delta(\erzeuge@tiefgestellt{#1})}
1147 \def\liUeberfuehrungsFunktion#1{
1148   \ifmode
1149     \liUeberfuehrungsFunktionOhneMathe{#1}
1150   \else
1151     $\liUeberfuehrungsFunktionOhneMathe{#1}$
1152   \fi
1153 }

\liAlphabet \liAlphabet{a, b} ( $\Sigma = \{a, b\}$ ):  $\Sigma = \{a, b\}$ 

1154 \def\liAlphabet#1{$\Sigma = \{ #1 \}$}

\liBandAlphabet \liBandAlphabet{\liTuringLeerzeichen}:  $\Gamma = \Sigma \cup \{\square\}$ 

1155 \def\liBandAlphabet#1{$\Gamma = \Sigma \cup \{ #1 \}$}

\liZustandsBuchstabe

1156 \def\liZustandsBuchstabe{z}

\liZustandsBuchstabeGross

1157 \def\liZustandsBuchstabeGross{Z}

```

```

\liZustandsmengeNr
1158 \def\zustandsnamens@liste#1#2{
1159   $
1160   \{
1161     \directlua{formale_sprachen.erzeuge_zustandsnamens_liste('#1', '#2')}
1162   \}
1163   $
1164 }
1165 \def\liZustandsmengeNr#1{\zustandsnamens@liste{\liZustandsBuchstabe}{#1}}

\liZustandsmengeNrGross
1166 \def\liZustandsmengeNrGross#1{\zustandsnamens@liste{\liZustandsBuchstabeGross}{#1}}

\liZustandsname \liZustandsname{1}: $z_1$
1167 \def\liZustandsname#1{\liZustandsBuchstabe_#1$}

\liZustandsnameGross \liZustandsnameGross{1}: $Z_1$
1168 \def\liZustandsnameGross#1{\liZustandsBuchstabeGross_#1$}

\liAbleitung \liAbleitung{S -> aB -> ab}:  $S \vdash aB \vdash ab$ 
1169 \def\liAbleitung#1{\directlua{formale_sprachen.formatiere_ableitung('#1')}}$}

liProduktionsRegeln \begin{liProduktionsRegeln}[P_1]
    S -> S A B | EPSILON,
    B A -> A B,
    A A -> a a,
    B B -> b b
\end{liProduktionsRegeln}

1170 \NewDocumentEnvironment { liProduktionsRegeln }
1171 { 0{P} +b }
1172 {
1173   \liGeschweifteKlammern{#1}
1174   {
1175     \begin{align*}
1176       \directlua{formale_sprachen.produktions_regeln('#2')}
1177     \end{align*}
1178   }{-0.2cm}{-1.5cm}
1179 } {}

\liProduktionen \liProduktionen{S -> A, A -> a}:  $\{S \rightarrow A, A \rightarrow a\}$ 
1180 \def\liProduktionen#1{
1181   \liMenge{\directlua{formale_sprachen.produktions_regeln('#1', true)}}
1182 }

\liZustandsnameTiefgestelltt Automatisch tiefgestellte Nummerierung \z1
Let-Abkürzung: \let\z=\liZustandsnameTiefgestelltt
1183 \def\liZustandsnameTiefgestelltt#1{
1184   \ifmmode
1185     \liZustandsBuchstabe\sb{#1}
1186   \else
1187     $\liZustandsBuchstabe\sb{#1}$
1188   \fi
1189 }

1190 \ExplSyntaxOn

\liAusdruck \liAusdruck[L_2]{a_1,a_2,\dots,a_n}{n \in N}:  $L_2 = \{a_1, a_2, \dots, a_n \mid n \in N\}$ 
    Ohne „=:“: \liAusdruck[] {x}{y}:  $\{x \mid y\}$ 
    Regulärer Ausdruck zum Konvertieren:
    \$(.*) += +\\{ *(.*) ( *\\, *)?\\ ( *\\, *)?(.*) *\\}\\$
    \liAusdruck[$1]{$2}{$5}

```

```

1191 \NewDocumentCommand{ \liAusdruck } { O{L} m m } {
1192   $
1193   \tl_if_empty:nTF {#1} {} {#1 =}
1194   \{
1195     \, #2 \,
1196     |
1197     \, #3 \,
1198   \}$
1199 }
1200 \ExplSyntaxOff

\liFlaci   Link zur flaci.com Website: \liFlaci{Grxk1oczg}:
           Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter)
           Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Grxk1oczg

1201 \def\liFlaci#1{%
1202   \par
1203   {%
1204     \scriptsize
1205     Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte
1206     Automaten, Compiler und Interpreter) Ein Projekt der Hochschule
1207     Zittau/Görlitz und der Pädagogischen Hochschule Schwyz:
1208     \href{https://flaci.com/#1}{flaci.com/#1}%
1209   }%
1210   \par
1211 }

\liGrammatik \liGrammatik[\langle grammatik-name \rangle]{\langle variablen=V,alphabet=\Sigma,produktionen=P,start=S \rangle}
              \liGrammatik{variablen={}, alphabet={}}

              • \liGrammatik{}:  $G = (V, \Sigma, P, S)$ 

              • \liGrammatik[G_1]{}:  $G_1 = (V, \Sigma, P, S)$ 

              • \liGrammatik{variablen={S,A,B}}:  $G = (\{S, A, B\}, \Sigma, P, S)$ 

              • \liGrammatik{alphabet={a,b}}:  $G = (V, \{a, b\}, P, S)$ 

              • \liGrammatik{start=X}:  $G = (V, \Sigma, P, X)$ 

1212 \ExplSyntaxOn
1213 \NewDocumentCommand {\liGrammatik} { O{G} m } {
1214   \tl_set:Nn \l_variablen_tl {V}
1215   \tl_set:Nn \l_alphabet_tl {\Sigma}
1216   \tl_set:Nn \l_produktionen_tl {P}
1217   \tl_set:Nn \l_start_tl {S}
1218
1219   \keys_define:nn { grammatik } {
1220     variablen .code:n = {\tl_set:Nn \l_variablen_tl {\liMenge{##1}}},
1221     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
1222     produktionen .code:n = {\tl_set:Nn \l_produktionen_tl {\liProduktionen{##1}}},
1223     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
1224   }
1225
1226   \keys_set:nn { grammatik } { #2 }
1227
1228   $#1 = (
1229     \l_variablen_tl,
1230     \l_alphabet_tl,
1231     \l_produktionen_tl,
1232     \l_start_tl
1233   )$
1234 }
1235 \ExplSyntaxOff

1236

```

## 2.14 formatierung.sty

```
1237 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1238 \ProvidesPackage{lehramt-informatik-formatierung}[2020/11/27]
```

### 2.14.1 Schriftarten / Typographie

```
1239 \RequirePackage{mathpazo}
1240 \RequirePackage[no-math]{fontspec}
1241 \setmainfont{texgyrepagella}
1242 \setsansfont{QTAncientOlive}
1243 \RequirePackage{sectsty}
1244 \allsectionsfont{\fontspec{QTAncientOlive}}
```

### 2.14.2 Farben

```
1245 \RequirePackage{xcolor}
1246 \definecolor{infogray}{rgb}{0.97,0.97,0.97}
```

### 2.14.3 Überschriften

```
1247 \RequirePackage{titlesec}
1248 \titleformat{\chapter}[display]{\sffamily\bfseries}{\Opt}{\LARGE}
1249 \titlespacing{\chapter}{0pt}{0pt}{*1}
1250 \titleformat{\paragraph}[hang]{\sffamily\normalsize\bfseries}{\theparagraph}{1em}{}
1251 \setcounter{secnumdepth}{0}
```

### 2.14.4 Listen

```
1252 \RequirePackage{paralist}
1253 \renewcommand\labelitemi{-}
1254 \renewcommand\labelitemii{-}
1255 \renewcommand\labelitemiii{-}
1256 \renewcommand\labelitemiv{-}
1257 % Counter: enumi enumii enumiii enumiv
1258 % Styles: \arabic{counter} \alph{counter} \Alph{counter} \roman{counter} \Roman{counter}
1259 \renewcommand{\labelenumi}{(\alph{enumi})}
1260 \renewcommand{\labelenumii}{(\roman{enumii})}
```

### 2.14.5 Kasten

```
1261 \RequirePackage{mdframed}
1262 \mdfsetup{backgroundcolor=infogray}
```

liKasten

```
1263 \NewDocumentEnvironment { liKasten }{ } {
1264   \begin{mdframed}[backgroundcolor=white!0]
1265 } {
1266   \end{mdframed}
1267 }
```

### 2.14.6 Header

```
1268 \RequirePackage{fancyhdr}
1269 \fancyhead[L,C,R]{}
1270 \fancyfoot[L]{}
1271 \fancyfoot[C]{}
1272 \fancyfoot[R]{\thepage}
1273 \pagestyle{fancy}
1274 \renewcommand{\headrulewidth}{0pt}
1275 \renewcommand{\footrulewidth}{0pt}

1276
```

## 2.15 gantt.sty

```

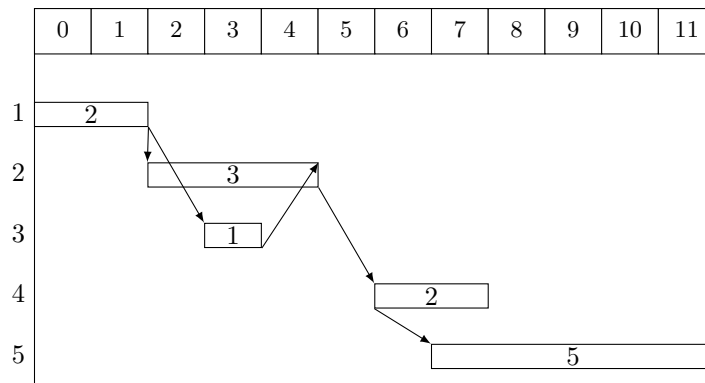
1277 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1278 \ProvidesPackage{lehramt-informatik-gantt}[2020/09/05]

\begin{ganttchart}[x unit=0.75cm, y unit chart=0.8cm]{0}{11}
\gantttitlelist{0,...,11}{1} \\
\ganttbar[name=1]{1}{0}{1} \\
\ganttbar[name=2]{2}{2}{4} \\
\ganttbar[name=3]{3}{3}{3} \\
\ganttbar[name=4]{4}{6}{7} \\
\ganttbar[name=5]{5}{7}{11}

\node at (1) {2};
\node at (2) {3};
\node at (3) {1};
\node at (4) {2};
\node at (5) {5};

\ganttlink[link type=f-f]{3}{2}
\ganttlink[link type=f-s]{1}{2}
\ganttlink[link type=f-s]{1}{3}
\ganttlink[link type=f-s]{2}{4}
\ganttlink[link type=s-s]{4}{5}
\end{ganttchart}

```



```

1279 \RequirePackage{tikz-uml}
1280 \RequirePackage{pgfgantt}
1281 \setganttlinklabel{f-s}{}
1282 \setganttlinklabel{s-s}{}
1283 \setganttlinklabel{f-f}{}
1284 \setganttlinklabel{s-f}{}

1285

```

## 2.16 grafik.sty

```
1286 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1287 \ProvidesPackage{lehramt-informatik-grafik}[2020/12/27 Lädt das Paket
1288 TikZ, um Grafiken zeichnen zu können.]
1289 \RequirePackage{tikz}
1290
```

## 2.17 graph.sty

```

1291 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1292 \ProvidesPackage{lehramt-informatik-graph}[2020/06/09]
1293 \RequirePackage{tikz}

```

Für die die Adjazenzliste (xrightarrow)

```

1294 \RequirePackage{amsmath}

```

Für Adjazenz-Matrix

```

\[
\begin{blockarray}{ccccc}
& a & b & c & d & e \\
\begin{block}{c(ccccc)}
a & 0 & 1 & 0 & 4 & 0 \\
b & 0 & 0 & 0 & 1 & 3 \\
c & 0 & 0 & 0 & 1 & 0 \\
d & 4 & 1 & 1 & 0 & 2 \\
e & 0 & 0 & 0 & 0 & 0
\end{block}
\end{blockarray}
\]

```

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

```

1295 \RequirePackage{blkarray}
1296 \usetikzlibrary{arrows.meta}

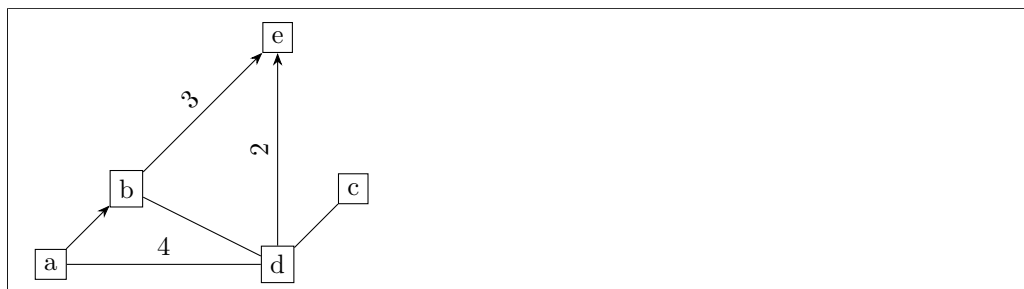
```

```

\begin{tikzpicture}[li graph]
\node (a) at (0,0) {a};
\node (b) at (1,1) {b};
\node (c) at (4,1) {c};
\node (d) at (3,0) {d};
\node (e) at (3,3) {e};

\path[->] (a) edge (b);
\path (b) edge (d);
\path[->] (b) edge node {3} (e);
\path (c) edge (d);
\path (d) edge node {4} (a);
\path[->] (d) edge node {2} (e);
\end{tikzpicture}

```





```

1297 \tikzset{
1298   li graph/.style={
1299     every node/.style={
1300       rectangle,
1301       draw,
1302     },
1303     every edge/.style={
1304       >={Stealth[black]},
1305       draw,
1306     },
1307     every edge/.append style={
1308       every node/.style={
1309         sloped,
1310         auto,
1311       }
1312     }
1313   },
1314   li markierung/.style={
1315     ultra thick,
1316   }
1317 }

```

**liGraphenFormat** Dummy-Umgebung, zum Parsen durch Java gedacht.

```

\begin{liGraphenFormat}
a: 0 0
b: 1 1
c: 4 1
d: 3 0
e: 2 2
a -> b
b -- d
b -> e: 3
c -- d
d -> e: 2
d -- a: 4
\end{liGraphenFormat}

```

```

1318 \NewDocumentEnvironment { liGraphenFormat }{ +b } {} {}

```

```

1319

```

## 2.18 hanoi.sty

```

1320 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1321 \ProvidesPackage{lehramt-informatik-hanoi}[2020/12/19 Paket zum Setzen
1322 von Hanoi-Grafiken]

    Quelle: https://kleinco.de/latex-snippets/7/tikz-towers-of-hanoi-illustration-for-lat
1323 \RequirePackage{tikz}
1324 \RequirePackage{xcolor}

\liHanoi \liHanoi{anzahl-scheiben}{gewicht-scheibe/turm-nr,gewicht-scheibe/turm-nr},
z.B. : \liHanoi{4}{4/1,3/1,2/3,1/2}

1325 \def\li@mset #1[#2]=#3{%
1326 \expandafter\xdef\csname #1#2\endcsname{#3}
1327 }
1328 \def\li@mget #1[#2]{%
1329 \csname #1#2\endcsname
1330 }
1331 \def\li@minc #1[#2]+=#3{%
1332 \pgfmathparse{\li@mget #1[#2]+#3}%
1333 \li@mset #1[#2]=\pgfmathresult
1334 }
1335
1336 \def\liHanoi#1#2{
1337   \edef\li@numdiscs{#1}
1338   \def\li@sequence{#2}
1339   \begin{tikzpicture}[line width=4mm,brown!40,line cap=round,xscale=3]
1340     % init colors
1341     \foreach[count=\j] \c in {red,green!80!black,blue,orange,violet,gray,yellow!80!black,purp
1342     \li@mset col[\j]={\c};
1343     % draw poles and init pole counters
1344     \foreach \j in {1,2,3}{
1345       \li@mset pos[\j]=0
1346       \draw (\j,-.5) -- +(0, .5 + .5*\li@numdiscs);
1347     }
1348     % draw base
1349     \draw (.5,-.5) -- +(3,0);
1350     % draw discs
1351     \foreach[count=\k] \i/\j in \li@sequence{
1352       \draw[draw={\li@mget col[\i]}] (\j,\li@mget pos[\j]) +(-.4*\i/\li@numdiscs,0) -- +(.4*
1353       \li@minc pos[\j]+=.5}
1354     }
1355   \end{tikzpicture}
1356 }

1357

```

## 2.19 klassen-konfiguration-aufgabe.sty

```
1358 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1359 \ProvidesPackage{lehramt-informatik-klassen-konfiguration-aufgabe}[2021/09/01 Die
1360 Klasse Aufgabe konfigurieren, d. h. Laden von einigen Paketen, Registrierung von Hooks]

    Formatierung muss vor literatur sein, sonst Option clash
1361 \liLadePakete{
1362   formatierung,
1363   abmessung,
1364   literatur-dummy,
1365   makros,
1366   aufgaben-metadaten,
1367   kopf-fusszeilen,
1368   mathe
1369 }
1370
1371 % Komischer Option-Clash deshalb ganz am Ende, für die Silbentrennung
1372 \RequirePackage[ngerman]{babel}
1373 \ExplSyntaxOn
1374 \AddToHook{enddocument}{
1375   \_gib_github_url:
1376 }
1377 \ExplSyntaxOff
1378
```

## 2.20 klassen-konfiguration-examen.sty

```
1379 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1380 \ProvidesPackage{lehramt-informatik-klassen-konfiguration-examen}[2021/09/04 Die
1381 Klasse liSetzeExamen konfigurieren, d. h. Laden von einigen Paketen, Registrierung von Hooks]
```

Lade die wichtigsten Pakete. Formatierung muss vor literatur sein, sonst Option clash

```
1382 \liLadePakete{
1383   formatierung,
1384   literatur-dummy,
1385   makros,
1386   aufgaben-metadaten,
1387   abmessung
1388 }
1389 \liLadeAllePakete
```

Komischer Option-Clash deshalb ganz am Ende, für die Silbentrennung

```
1390 \RequirePackage[ngerman]{babel}
1391 \RequirePackage{standalone}
1392 \ExplSyntaxOn
```

\liSetzeExamen

```
1393 \def\liSetzeExamen#1#2#3{
1394   \tl_gset:Nn \g_auf_examen_nummer_tl { #1 }
1395   \tl_gset:Nn \g_auf_examen_jahr_tl { #2 }
1396   \tl_gset:Nn \g_auf_examen_monat_tl { #3 }
1397 }
```

\liSetzeExamenThemaNr

```
1398 \def\liSetzeExamenThemaNr#1{
1399   \tl_gset:Nn \g_auf_examen_thema_nr_tl { #1 }
1400   \section{Thema-Nr.~#1}
1401 }
```

\liSetzeExamenTeilaufgabeNr

```
1402 \def\liSetzeExamenTeilaufgabeNr#1{
1403   \tl_gset:Nn \g_auf_examen_teilaufgabe_nr_tl { #1 }
1404   \subsection{Teilaufgabe-Nr.~#1}
1405 }
```

\liBindeAufgabeEin

```
1406 \def\liBindeAufgabeEin#1{
1407   \tl_gset:Nn \g_auf_examen_aufgabe_nr_tl { #1 }
1408   \input{
1409     \LehramtInformatikRepository /
1410     Staatsexamen /
1411     \g_auf_examen_nummer_tl /
1412     \g_auf_examen_jahr_tl /
1413     \g_auf_examen_monat_tl /
1414     \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} {
1415       Thema - \g_auf_examen_thema_nr_tl /
1416     }
1417     \tl_if_empty:NTF \g_auf_examen_teilaufgabe_nr_tl {} {
1418       Teilaufgabe - \g_auf_examen_teilaufgabe_nr_tl /
1419     }
1420     Aufgabe - \g_auf_examen_aufgabe_nr_tl .tex
1421   }
1422 }
```

```
1423 \ExplSyntaxOff
```

```
1424
```

## 2.21 komplexitaetstheorie.sty

```
1425 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1426 \ProvidesPackage{lehramt-informatik-komplexitaetstheorie}[2021/07/08 Zum
1427 Setzen von Karp's NP-vollständigen Problemen, Hilfsmakros für die
1428 Polynomialzeitreduktion.]
```

### 2.21.1 Makro-Kürzel

```
\let\n=\liProblemName
\let\r=\liPolynomiellReduzierbar
\let\b=\liProblemBeschreibung
```

```
1429 \liLadePakete{mathe}
```

Für das Makro `\liProblemBeschreibung{ }{ }` benötigt.

```
1430 \RequirePackage{mdframed}
```

```
\liStrich $L, \liStrich{L}$:  $L, L'$ 
```

```
1431 \def\liStrich#1{#1^{\prime}}
```

`\liProblemName` Zu Setzen von Problemnamen wie zum Beispiel SAT, COL, VERTEX COVER

**Let-Abkürzung:** `\let\n=\liProblemName`

`\liProblemName: SAT VERTEX COVER`

```
1432 \def\liProblemName#1{\texttt{\textsc{#1}}}
```

`\liProblemBeschreibung` Zu setzen von Problem-Beschreibungen:

```
\liProblemBeschreibung
```

```
{ }
```

```
{ }
```

```
{ }
```

#### CLIQUE

**Gegeben:** Ein ungerichteter Graph  $G = (V, E)$ , eine Zahl  $k \in \mathcal{N}$

**Frage:** Gibt es eine Menge  $S \subseteq V$  mit  $|S| = k$ , sodass für alle Knoten  $u \neq v \in V$  gilt, dass  $\{u, v\}$  eine Kante in  $E$  ist?

**Let-Abkürzung:** `\let\b=\liProblemBeschreibung`

```
1433 \def\liProblemBeschreibung#1#2#3{
1434   \begin{mdframed}[
1435     userdefinedwidth=9cm,
1436     align=center,
1437     backgroundcolor=white!0,
1438   ]
1439   \centerline{\large\liProblemName{#1}}
1440
1441   \medskip
1442
1443   \begin{description}
1444     \item[Gegeben:] #2
1445     \item[Frage:] #3
1446   \end{description}
1447 \end{mdframed}
1448 }
```

```

\liPolynomiellReduzierbar Let-Abkürzung: \let\r=\liPolynomiellReduzierbar

1449 \NewDocumentCommand{ \liPolynomiellReduzierbar } { m O{p} m } {
1450 \begin{displaymath}
1451   \liProblemName{#1}
1452   \preceq_{#2}
1453   \liProblemName{#3}
1454 \end{displaymath}
1455 }

\liProblemVertexCover

1456 \def\liProblemClique{%
1457 Das \textbf{Cliquesproblem} fragt nach der Existenz einer Clique der
1458 Mindestgröße  $n$  in einem gegebenen Graphen.
1459 \footcite[Seite 76]{theo:fs:4}
1460 Eine Clique ist eine Teilmenge von Knoten in einem ungerichteten
1461 Graphen, bei der \emph{jedes Knotenpaar durch eine Kante} verbunden ist.
1462 \footcite{wiki:cliquenproblem}
1463 }

\liProblemVertexCover

1464 \def\liProblemVertexCover{%
1465 %
1466 Das \textbf{Knotenüberdeckungsproblem} (\liProblemName{Vertex Cover})
1467 fragt, ob zu einem gegebenen einfachen Graphen und einer natürlichen
1468 Zahl  $k$  eine Knotenüberdeckung der Größe von höchstens  $k$  existiert.
1469 \footcite{wiki:knotenueberdeckung}
1470
1471 Das heißt, ob es eine aus maximal  $k$  Knoten bestehende Teilmenge  $U$ 
1472 der Knotenmenge gibt, so dass jede Kante des Graphen mit
1473 mindestens einem Knoten aus  $U$  verbunden ist.
1474 \footcite[Seite 78]{theo:fs:4}%
1475 }

\liProblemSubsetSum Kein Karp-21-Problem

1476 \def\liProblemSubsetSum{%
1477 Das \textbf{Teilsommenproblem} (\liProblemName{Subset Sum} oder
1478 \liProblemName{SSP}) ist ein spezielles Rucksackproblem.
1479 \footcite{wiki:teilsommenproblem}
1480 Gegeben sei eine Menge von ganzen Zahlen  $I = \{w_1, w_2, \dots, w_n\}$ .
1481 Gesucht ist eine Untermenge, deren Elementsumme maximal, aber nicht
1482 größer als eine gegebene obere Schranke  $c$  ist.
1483 \footcite[Seite 74]{theo:fs:4}
1484 }

\liProblemSubsetSum Kein Karp-21-Problem

1485 \def\liProblemSat{%
1486 Das \textbf{Erfüllbarkeitsproblem der Aussagenlogik} \liProblemName{Sat}
1487 und \liProblemName{k-SAT} mit  $k \geq 3$ ,  $k \in \mathbb{N}$  (Satz von
1488 Cook) fragt, ob eine aussagenlogische Formel erfüllbar
1489 ist.\footcite{wiki:sat} Das Erfüllbarkeitsproblem der
1490 \emph{Aussagenlogik} ist in exponentieller Zeit in Abhängigkeit der
1491 Anzahl der Variablen mit Hilfe einer Wahrheitstabelle entscheidbar.
1492 Diese \emph{Wahrheitstabelle} kann nicht in polynomieller Zeit
1493 aufgestellt werden.
1494 \footcite[Seite 71]{theo:fs:4}
1495 }

1496

```

## 2.22 kontrollflussgraph.sty

1497 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

1498 \ProvidesPackage{lehramt-informatik-kontrollflussgraph}[2020/11/07]

### 2.22.1 Makro-Kürzel

```
\let\b=\liBedingung
\let\c=\liKontrollCode
\let\f=\liBedingungFalsch
\let\k=\liKontrollTextzeileKnoten
\let\p=\liKontrollKnotenPfad
\let\w=\liBedingungWahr
```

### 2.22.2 TeX-Markup-Beispiel

```
\begin{liKontrollflussgraph}[xscale=1,yscale=-1.6]
\node[knoten] at (0,0) (S) {S};

\node[pin=\c{boolean resultat = false;}] at (0,1) (2) {2};
\node[pin=\c{if (wort != null)}] at (0,2) (3) {3};
\node[pin=\c{int laenge = wort.length;}] at (1,3) (4) {4};
\node[pin=\c{if (laenge >= 2)}] at (1,4) (5) {5};
\node[pin=\c{resultat = true; int i = 0;}] at (2,5) (6) {6};
\node[pin=\c{for (i < laenge / 2;)}] at (2,6) (7) {7};
\node[pin=\c{char c1; char c2 ...}] at (3,7) (8) {8};
\node[pin=\c{if (Char...) }] at (2,8) (10) {10};
\node[pin=\c{resultat = false; break;}] at (2,9) (12) {12};
\node[pin=\c{if (laenge == 1)}] at (1,10) (17) {17};
\node[pin=\c{resultat = true;}] at (1,11) (18) {18};
\node[pin=180:\c{return resultat;}] at (-1,9) (22) {22};
\node[knoten] at (-1,10) (E) {E};

\path (S) -- (2);
\path (2) -- (3);
\path[wahr] (3) -- (4) \b{right}{wort != null};
\path[falsch] (3) -- (22) \b{left}{wort == null};
\path (4) -- (5);
\path[wahr] (5) -- (6) \b{right}{laenge >= 2};
\path[falsch] (5) -- (17) \b{left,rotate=70,pos=0.2}{laenge < 2};
\path (6) -- (7);
\path[wahr] (7) -- (8) \b{right}{i < laenge / 2};
\path[falsch] (7) -- (22) \b{left}{i >= laenge / 2};
\path (8) -- (10);
\path[wahr] (10) -- (12) \b{right}{c != c};
\path[falsch] (10) -- (7) \b{left,rotate=70,pos=0.8}{c == c};
\path (12) -- (22);
\path[wahr] (17) -- (18) \b{right}{laenge == 1};
\path[falsch] (17) -- (22) \b{right,rotate=-20,pos=0.99}{laenge != 1};
\path (18) -- (22);
\path (22) -- (E);
\end{liKontrollflussgraph}
```

### 2.22.3 TikZ: pin

Die Knoten erhalten sogenannte Pins, in denen Code-Ausschnitte der entsprechenden Anweisung zu sehen ist.

```
pin=[options]angle:text
```

```
\node[pin=\c{resultat = true;}] at (1,11) {18};
\node[pin=180:\c{return resultat;}] at (-1,9) {22};
\node[pin=[pin distance=3cm]\c{resultat = ...}]
```

```

1499 \RequirePackage{tikz}
1500 \usetikzlibrary{positioning}
1501 \tikzset{
1502   li kontrollfluss/.style={
1503     knoten/.style={
1504       circle,
1505       draw
1506     },
1507     usebox/.style={
1508       draw,
1509       rectangle,
1510       font=\scriptsize,
1511       anchor=west,
1512       align=left,
1513     },
1514     bedingung/.style={
1515       midway,
1516       draw=none,
1517       font=\scriptsize
1518     },
1519     knotenbeschriftung/.style={
1520       draw,
1521       rectangle,
1522       midway,
1523       font=\scriptsize
1524     },
1525     wahr/.style={
1526       thick
1527     },
1528     falsch/.style={
1529       dashed
1530     },
1531     every node/.style={
1532       circle,
1533       draw,
1534     },
1535     every edge/.append style={
1536       every node/.style={
1537         draw=none,
1538         bedingung,
1539       }
1540     },
1541     every path/.style={
1542       draw,
1543       ->,
1544     },
1545     every pin/.style={
1546       draw,
1547       dotted,
1548       rectangle,
1549       pin position=right
1550     },
1551     every pin edge/.style={
1552       dotted,
1553       arrows=-,
1554     }
1555   }
1556 }

```

#### 2.22.4 Umgebungen

liKontrollflussgraph

```

1557 \NewDocumentEnvironment { liKontrollflussgraph } { 0{} } {

```



```

1558 \begin{tikzpicture}[
1559     li kontrollfluss,
1560     #1
1561 ]
1562 } {
1563 \end{tikzpicture}
1564 }

```

## 2.22.5 Makros

`\liAnweisung`

```
1565 \def\liAnweisung#1(#2,#3){\node[knoten] at (#2,#3) (#1) {#1};}
```

`\liBedingung` **Let-Abkürzung:** `\let\b=\liBedingung`

```
1566 \def\liBedingung#1#2{\node[bedingung,#1]{\texttt{#2}}}
```

`\liBedingungWahr` **Let-Abkürzung:** `\let\w=\liBedingungWahr`

```
1567 \def\liBedingungWahr#1{\node[bedingung,#1]{\texttt{[true]}}}
```

`\liBedingungFalsch` **Let-Abkürzung:** `\let\f=\liBedingungFalsch`

```
1568 \def\liBedingungFalsch#1{\node[bedingung,#1]{\texttt{[false]}}}
```

`\liKontrollCode` **Let-Abkürzung:** `\let\c=\liKontrollCode`

```
1569 \def\liKontrollCode#1{{\tiny\texttt{\textcolor{gray}{#1}}}}
```

`\liKontrollTextzeileKnoten` **Let-Abkürzung:** `\let\k=\liKontrollTextzeileKnoten`

```
1570 \def\liKontrollTextzeileKnoten#1{\raisebox{-2pt}{\tikz[scale=0.5,transform shape] \node[draw
```

`\liKontrollKnotenPfad` **Let-Abkürzung:** `\let\p=\liKontrollKnotenPfad`

```
1571 \ExplSyntaxOn
```

```
1572 \NewDocumentCommand { \liKontrollKnotenPfad } { m }
```

```
1573 {
```

```
1574   \seq_set_split:Nnn \l_tmpa_seq { - } { #1 }
```

```
1575   \seq_set_map:NNn \l_tmpa_seq \l_tmpa_seq {\liKontrollTextzeileKnoten{##1}}
```

```
1576   \seq_use:Nn \l_tmpa_seq { ~~~~ }
```

```
1577 }
```

```
1578 \ExplSyntaxOff
```

```
1579
```

## 2.23 kopf-fusszeilen.sty

```
1580 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1581 \ProvidesPackage{lehramt-informatik-kopf-fusszeilen}[2021/08/20 Kopf-
1582 und Fußzeilen mit Hilfe des Pakets fancyhdr.]

1583 \ExplSyntaxOn

1584 \fancyhead{}
1585 \fancyhead[R0,LE]{\scriptsize\LehramtInformatikTitel}
1586 \fancyhead[L0,RE]{\scriptsize\today}
1587 \fancyfoot{}
1588 \fancyfoot[LE,R0]{\thepage}
1589 \fancyfoot[L0,CE]{\scriptsize\LehramtInformatikAutorName}
1590 \fancyfoot[CO,RE]{\scriptsize\LehramtInformatikAutorEmail}
1591 \renewcommand{\headrulewidth}{0.4pt}
1592 \renewcommand{\footrulewidth}{0.4pt}
1593 \setlength{\headwidth}{\textwidth}

1594 \ExplSyntaxOff

1595
```

## 2.24 literatur-dummy.sty

```
1596 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1597 \ProvidesPackage{lehramt-informatik-literatur-dummy}[2020/11/27]

\literatur

1598 \def\literatur{}

\footcite

1599 % \RequirePackage[stable,multiple]{footmisc}
1600 \NewDocumentCommand{ \footcite } { o m }{}

1601
```

## 2.25 literatur.sty

```
1602 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1603 \ProvidesPackage{lehramt-informatik-literatur}[2020/11/27]

1604 \RequirePackage{csquotes}
1605 \RequirePackage[
1606   bibencoding=utf8,
1607   citestyle=authortitle,
1608   backend=biber,
1609 ]{biblatex}
1610 \addbibresource{\LehramtInformatikRepository/Quellen/10_DB.bib}
1611 \addbibresource{\LehramtInformatikRepository/Quellen/20_OOMUP.bib}
1612 \addbibresource{\LehramtInformatikRepository/Quellen/30_AUD.bib}
1613 \addbibresource{\LehramtInformatikRepository/Quellen/40_SOSY.bib}
1614 \addbibresource{\LehramtInformatikRepository/Quellen/60_FUMUP.bib}
1615 \addbibresource{\LehramtInformatikRepository/Quellen/70_THEO.bib}
1616 \addbibresource{\LehramtInformatikRepository/Quellen/80_DDI.bib}
1617 \addbibresource{\LehramtInformatikRepository/Quellen/Allgemein.bib}
1618 \addbibresource{\LehramtInformatikRepository/Quellen/Examen.bib}
1619 \addbibresource{\LehramtInformatikRepository/Quellen/Wikipedia.bib}
1620 % To allow footnotes in the heading
1621 \RequirePackage[stable,multiple]{footmisc}

\literatur

1622 \def\literatur{\printbibliography[heading=subbibliography]}

1623
```

## 2.26 makros.sty

```

1624 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1625 \ProvidesPackage{lehramt-informatik-makros}[2020/11/27 Eine Sammlung von
1626 häufig verwendeten Makros und Umgebungen, die thematisch zu keinem
1627 anderen Paket passen]
1628 \RequirePackage{hyperref}
1629 \RequirePackage{graphicx}

Für die Umgebung liQuellen benötigt.
1630 \RequirePackage{paralist}

\inhaltsverzeichnis
1631 \def\inhaltsverzeichnis {
1632   \begin{mdframed}
1633     \begin{group}
1634       \let\clearpage\relax
1635       \tableofcontents
1636     \end{group}
1637   \end{mdframed}
1638 }

\memph \mephm (\marginpar and \emph)
1639 \newcommand{\memph}[1]{\emph{#1}\marginpar{\tiny#1}}

\SLASH
1640 \newcommand\SLASH{\char`\\}

\liPseudoUeberschrift Text, der sich wie eine Überschrift verhält.
1641 \newcommand{\liPseudoUeberschrift}[1]{
1642   \bigskip
1643   \noindent
1644   \textsf{\textbf{#1}}
1645   \noindent
1646 }

\liBeschriftung Ähnlich dem Makro \liPseudoUeberschrift{}. Am Ende des Textes wird ein Doppelpunktzeichen angehängt.
1647 \newcommand{\liBeschriftung}[1]{
1648   \par
1649   \noindent
1650   \medskip
1651   \textbf{#1}:
1652   \medskip
1653   \noindent
1654 }

\hinweis
1655 \def\hinweis#1{{\footnotesize#1}}

liProjektSprache \begin{liProjektSprache}{NameProjektSprache} \end{liProjektSprache}: Zum Einbetten von projekteigenen Minisprachen/DSLs (Domain-specific language) (z. B. RelationenSchema). Der Inhalt der Umgebung wird nicht von TeX kompiliert, sondern von dem Java-Kommandozeilen-Programm didaktik.java verarbeitet.
1656 \NewDocumentEnvironment { liProjektSprache }{ o +b } {} {}

liEinbettung
1657 \NewDocumentEnvironment { liEinbettung }{ o +b } {#2} {}

liAntwort Zum Setzen von Antworten. Sie werden mit einem Kasten umgeben. Könnten eventuell ausgeblendet werden, wenn man nur die Lösung sehen will.
1658 \RequirePackage{xparse}
1659 \ExplSyntaxOn

```

```

1660 \NewDocumentEnvironment{ liAntwort } { 0{standard} }
1661 {
1662   \str_case:nn {#1} {
1663     {standard} {
1664       \def\beschriftung{}
1665       \mdfsetup{backgroundcolor=gray!1,linecolor=gray}
1666     }
1667     {richtig} {
1668       \def\beschriftung{richtig}
1669       \mdfsetup{backgroundcolor=green!5,linecolor=green}
1670     }
1671     {falsch} {
1672       \def\beschriftung{falsch}
1673       \mdfsetup{backgroundcolor=red!3,linecolor=red}
1674     }
1675     {muster} {
1676       \def\beschriftung{Musterlösung}
1677       \mdfsetup{backgroundcolor=green!12,linecolor=green}
1678     }
1679   }
1680   \ifx\beschriftung\empty\else
1681     \noindent
1682     \textbf{\beschriftung{:}}
1683     \fi
1684     \begin{mdframed}
1685   }
1686 {\end{mdframed}}

```

**liAdditum** Zusätzliches Material bei Aufgaben, das zum Lösen der Aufgaben nicht unbedingt nötig ist, z. b. Hintergrundinformation, Test-Dateien, komplette Code-Dateien.

```

1687 \NewDocumentEnvironment{ liAdditum } { o }
1688 {
1689   \begin{mdframed}[backgroundcolor=yellow!5]
1690     \IfNoValueTF {#1}
1691     { \liPseudoUeberschrift{Additum} }
1692     { \liPseudoUeberschrift{Additum:~#1} }
1693   }
1694 {\end{mdframed}}

```

**liExkurs** \begin{liExkurs}[Linear rekursiv]  
 Die häufigste Rekursionsform ist die lineare Rekursion, bei der in jedem Fall der rekursiven Definition höchstens ein rekursiver Aufruf vorkommen darf.  
 \end{liExkurs}

#### **Exkurs: Linear rekursiv**

Die häufigste Rekursionsform ist die lineare Rekursion, bei der in jedem Fall der rekursiven Definition höchstens ein rekursiver Aufruf vorkommen darf.

```

1695 \NewDocumentEnvironment{ liExkurs }{o +b}{
1696   \vspace{0.2cm}%
1697   \begin{mdframed}[
1698     backgroundcolor=white,
1699     bottomline=false,
1700     innermargin=1cm,
1701     leftline=true,
1702     linecolor=black,
1703     linewidth=0.1cm,
1704     outermargin=1cm,
1705     rightline=false,
1706     topline=false,
1707   ]

```

```

1708 \footnotesize
1709 \noindent%
1710 \textbf{Exkurs:~#1}\par%
1711 \noindent%
1712 #2
1713 \end{mdframed}
1714 \vspace{0.2cm}
1715 }{}

```

liQuellen <https://tex.stackexchange.com/a/229004>

```

\begin{liQuellen}
\item Quelle 1
\item Quelle 2
\end{liQuellen}

```

**Weiterführende Literatur:**

- Quelle 1
- Quelle 2

```

1716 \cs_new:Npn \listen@punkt #1 {\item #1}
1717 \NewDocumentEnvironment { liQuellen }{ +b }
1718 {
1719   \seq_clear_new:N \l_quellen
1720   \seq_set_split:Nnn \l_quellen {\item} {#1}
1721   \seq_remove_all:Nn \l_quellen {}
1722   \begin{mdframed}[backgroundcolor=white!0]
1723     \footnotesize
1724     \noindent
1725     \textsf{\textbf{Weiterführende~Literatur:}}
1726     \medskip
1727     \begin{compactitem}
1728       \seq_map_function:NN \l_quellen {\listen@punkt}
1729     \end{compactitem}
1730   \end{mdframed}
1731   %
1732   \makeatletter
1733   \par\@afterindentfalse\@afterheading
1734   \makeatother
1735 } {}

```

liLernkartei

```

1736 \NewDocumentEnvironment { liLernkartei }{ m +b }
1737 {
1738   \begin{mdframed}
1739     \footnotesize
1740     \noindent%
1741     \textbf{Lernkarteikarte:~#1}\par%
1742     \noindent%
1743     #2
1744   \end{mdframed}
1745 } {}

```

liDiagramm `\begin{liDiagramm}{beschriftung}\end{liDiagramm}`: Zu setzen einer Graphik bzw eines Diagramms.

```

1746 \NewDocumentEnvironment { liDiagramm }{ m +b }
1747 {
1748   \begin{mdframed}[backgroundcolor=white!0]
1749     \small
1750     \noindent%
1751     \textit{#1}:
1752     \begin{center}

```

```

1753   #2
1754   \medskip
1755   \end{center}
1756   \end{mdframed}
1757 } {}
1758 \ExplSyntaxOff

\liFussnoteUrl \liFussnoteUrl[\langle zusätzlicher-text \rangle]{\langle url \rangle} \liFussnoteUrl[zusätzlicher Text]{url}:
Eine HTTP-URL als Fußnote setzen.
1759 \NewDocumentCommand{\liFussnoteUrl} { o m } {
1760   \footnote{\url{#2}\IfNoValueTF{#1}{}{ ( #1 ) }}
1761 }
1762

\liFussnoteLink \liFussnoteLink[\langle zusätzlicher-text \rangle]{\langle link-text \rangle}{\langle url \rangle} \liFussnoteLink[zusätzlicher Text]{tex
Einen Link, d. h. einen Link-Text und eine URL als Fußnote setzen.
1763 \NewDocumentCommand{\liFussnoteLink} { o m m } {
1764   \footnote{\href{#3}{#2}\IfNoValueTF{#1}{}{ ( #1 ) }}
1765 }

\zB
1766 \def\zB{z.\.,B. }

\ZB
1767 \def\ZB{Z.\.,B. }

\dh
1768 \def\dh{d.\.,h. }

1769

```



## 2.27 master-theorem.sty

1770 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

1771 \ProvidesPackage{lehramt-informatik-master-theorem}[2021/04/13]

### 2.27.1 Makro-Kürzel

\let\O=\liO

\let\o=\liOmega

\let\T=\liT

\let\t=\liTheta

\liMasterVariablenDeklaration

{3} % a

{3} % b

{\mathcal{O}(1)} % f(n)

\liMasterFallRechnung

% 1. Fall

{für  $\varepsilon = 4$ : \}

$f(n) = 5n^2 \in \mathcal{O}(n^{\log_2 8 - 4}) = \mathcal{O}(n^{\log_2 4}) = \mathcal{O}(n^2)$

% 2. Fall

$f(n) = 5n^2 \notin \mathcal{T}(n^{\log_2 8}) = \mathcal{T}(n^3)$

% 3. Fall

$f(n) = 5n^2 \notin \mathcal{O}(n^{\log_2 8 + \varepsilon})$

\liMasterWolframLink{T[n]=9T[n/3]\%2B5n^2}

\liMasterVariablenDeklaration

{ } % a

{ } % b

{ } % f(n) ohne \$mathe\$

\liMasterFallRechnung

% 1. Fall

{ }

% 2. Fall

{ }

% 3. Fall

{ }

$\rightarrow T(n) \in \mathcal{T}(n^2 \cdot \log n)$

\liMasterWolframLink{T[n]=9T[n/3]\%2B5n^2}

1772 \ExplSyntaxOn

1773 \RequirePackage{amsmath}

\liRundeKlammer

1774 \def\liRundeKlammer#1{

1775 \negthinspace \left( #1 \right)

1776 }

\liTheta \liTheta{n^2}:  $\Theta(n^2)$

1777 \def\liThetaOhneMathe#1{

1778 \Theta \liRundeKlammer{#1}

1779 }

1780 \def\liTheta#1{

1781 \ifmmode

1782 \liThetaOhneMathe{#1}

1783 \else

1784 \$\liThetaOhneMathe{#1}\$

1785 \fi

1786 }

```

\liOmega \liOmega{n^2}:  $\Omega(n^2)$ 
1787 \def\liOmegaOhneMathe#1{
1788 \Omega \liRundeKlammer{#1}
1789 }
1790 \def\liOmega#1{
1791 \ifmmode
1792 \liOmegaOhneMathe{#1}
1793 \else
1794 $\liOmegaOhneMathe{#1}$
1795 \fi
1796 }

\liO \liO{n^2}:  $\mathcal{O}(n^2)$ 
1797 \def\liOOhneMathe#1{
1798 \mathcal{O} \liRundeKlammer{#1}
1799 }
1800 \def\liO#1{
1801 \ifmmode
1802 \liOOhneMathe{#1}
1803 \else
1804 $\liOOhneMathe{#1}$
1805 \fi
1806 }

\liT Let-Abkürzung: \let\T=\liT
\liT{16}{2}:  $16 \cdot T\left(\frac{n}{2}\right)$  \liT{}{2}:  $T\left(\frac{n}{2}\right)$ 
1807 \def\liTOhneMathe#1#2{
1808 \tl_if_blank:nTF {#1}
1809 {}
1810 {#1 \cdot }
1811 T
1812 \liRundeKlammer{\frac{n}{#2}}
1813 }
1814 \def\liT#1#2{
1815 \ifmmode
1816 \liTOhneMathe{#1}{#2}
1817 \else
1818 $\liTOhneMathe{#1}{#2}$
1819 \fi
1820 }

\liRekursionsGleichung \liRekursionsGleichung:  $T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$ 
1821 \def\liRekursionsGleichung{
1822 $T(n) = \liT{a}{b} + f(n)$
1823 }

\liBedingungEins \liBedingungEins:  $f(n) \in \mathcal{O}(n^{\log_b a - \varepsilon})$ 
1824 \def\liBedingungEins{
1825 $f(n) \in \liO{n^{\log\sb{b}a - \varepsilon}}$
1826 }

\liBedingungZwei \liBedingungZwei:  $f(n) \in \Theta(n^{\log_b a})$ 
1827 \def\liBedingungZwei{
1828 $f(n) \in \liTheta{n^{\log\sb{b}a}}$
1829 }

\liBedingungDrei \liBedingungDrei:  $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ 
1830 \def\liBedingungDrei{
1831 $f(n) \in \liOmega{n^{\log\sb{b}a + \varepsilon}}$
1832 }

1833 \ExplSyntaxOff

```

## \liMasterVariablen

```

1834 \def\liMasterVariablen{
1835   \begin{displaymath}
1836     T(n) = \liT{a}{b} + f(n)
1837   \end{displaymath}
1838
1839   \begin{itemize}
1840     \item[$a = $]
1841       Anzahl der Unterprobleme in der Rekursion  $a \geq 1$ 
1842
1843     \item[$\textstyle{\frac{1}{b}} = $]
1844       Teil des Originalproblems, welches wiederum durch alle Unterprobleme
1845       repräsentiert wird  $b > 1$ 
1846
1847     \item[$f(n) = $]
1848       Kosten (Aufwand, Nebenkosten), die durch die Division des Problems und
1849       die Kombination der Teillösungen entstehen. Eine von  $T(n)$ 
1850       unabhängige und nicht negative Funktion.
1851   \end{itemize}
1852   \footcite{wiki:master-theorem}
1853   \footcite[Seite 19-35]{aud:fs:2}
1854 }
```

## \liMasterFaelle

```

1855 \def\liMasterFaelle{
1856   \begin{description}
1857     \item[1. Fall:]
1858        $T(n) \in \liTheta{n^{\log_{sb{b}}a}}$ 
1859
1860       \hfill falls \liBedingungEins
1861       für  $\varepsilon > 0$ 
1862
1863     \item[2. Fall:]
1864        $T(n) \in \liTheta{n^{\log_{sb{b}}a} \cdot \log n}$ 
1865
1866       \hfill falls \liBedingungZwei
1867
1868     \item[3. Fall:]
1869        $T(n) \in \liTheta{f(n)}$ 
1870
1871       \hfill falls \liBedingungDrei
1872       für  $\varepsilon > 0$ 
1873       und ebenfalls für ein  $c$  mit  $0 < c < 1$  und alle hinreichend großen  $n$ 
1874       gilt:
1875        $a \cdot f(\textstyle{\frac{n}{b}}) \leq c \cdot f(n)$ 
1876   \end{description}
1877 }
```

## \liMasterVariablenDeklaration

```

1878 \def\liMasterVariablenDeklaration#1#2#3{
1879   \begin{description}
1880     \item[Allgemeine Rekursionsgleichung:] \strut
1881
1882     \liRekursionsGleichung
1883
1884     \item[Anzahl der rekursiven Aufrufe ( $a$ ):] \strut
1885
1886     #1
1887
1888     \item[Anteil Verkleinerung des Problems ( $b$ ):] \strut
1889
1890     um  $\frac{1}{\#2}$  also  $b = \#2$ 
1891   \end{description}
```

```

1892 \item[Laufzeit der rekursiven Funktion ( $f(n)$ ):] \strut
1893
1894  $\#3$ 
1895
1896 \item[Ergibt folgende Rekursionsgleichung:] \strut
1897
1898  $T(n) = T_{\#1}(\#2) + \#3$ 
1899 \end{description}
1900 }

```

\liMasterFallRechnung

```

1901 \def\liMasterFallRechnung#1#2#3{
1902 \begin{description}
1903 \item[1. Fall:] \liBedingungEins:
1904
1905 #1
1906
1907 \item[2. Fall:] \liBedingungZwei:
1908
1909 #2
1910
1911 \item[3. Fall:] \liBedingungDrei:
1912
1913 #3
1914 \end{description}
1915 }

```

\liMasterExkurs

```

1916 \def\liMasterExkurs{
1917 \begin{liExkurs}[Master-Theorem]
1918 \liMasterVariablen
1919
1920 \noindent
1921 Dann gilt:
1922
1923 \liMasterFaelle
1924 \end{liExkurs}
1925 }

```

\liMasterWolframLink Link zu Wolframalpha (+ durch Prozent 2B ersetzen)

```

1926 \def\liMasterWolframLink#1{
1927 Berechne die Rekursionsgleichung auf WolframAlpha:
1928 \href{https://www.wolframalpha.com/input/?i=\#1}{WolframAlpha}
1929 }

```

```

1930

```

## 2.28 mathe.sty

```
1931 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1932 \ProvidesPackage{lehramt-informatik-mathe}[2020/06/10]
1933
1934 % for example \ltimes \rtimes
1935 %\RequirePackage{amssymb}
1936 \RequirePackage{amsmath}
1937
1938 %%
1939 % \mlq \mrq
1940 %%
1941 \DeclareMathSymbol{\mlq}{\mathord}{operators}{``}
1942 \DeclareMathSymbol{\mrq}{\mathord}{operators}{`}
1943
```

## 2.29 minimierung.sty

```

1944 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1945 \ProvidesPackage{lehramt-informatik-mathe}[2021/03/13 Für den
1946 Minimierungsalgorithmus von einem NEA zu einem DEA]

1947 \liLadePakete{typographie}

\let\z=\liZustandsnameTiefgestellt
\let\f=\liFussnote
\let\l=\liLeereZelle
\let\Z=\liZustandsPaar
\let\verklaerung=\liMinimierungErklaerung


\begin{tabular}{|c||c|c|c|c|c|c|c|c|c|}
\hline
\z0 & \l & & \l & & \l & & \l & & \l & & \\
\z1 & & & \l & & \l & & \l & & \l & & \\
\z2 & & & & & \l & & \l & & \l & & \\
\z3 & & & & & & & \l & & \l & & \\
\z4 & & & & & & & & & \l & & \\
\z5 & & & & & & & & & & & \\
\z6 & & & & & & & & & & & \\
\z7 & & & & & & & & & & & \\
\z8 & & & & & & & & & & & \\
& \z0 & \z1 & \z2 & \z3 & \z4 & \z5 & \z6 & \z7 & \z8 & & 
\end{tabular}

\liFussnoten

\begin{liUebergangsTabelle}{0}{1}
\Z01 & \Z10 & \Z23 & \\
\Z05 & \Z15 & \Z25 & \f2 \\
\Z15 & \Z05 & \Z35 & \f2 \\
\Z23 & \Z44 & \Z55 & \\
\Z24 & \Z44 & \Z55 & \\
\Z34 & \Z44 & \Z55 & 
\end{liUebergangsTabelle}

1948 \def\liFussnote#1{$x_{\text{\scriptsize #1}}$}

1949 \def\li@fussnote@text#1#2{
1950   \liFussnote{\scriptsize #1}
1951   \quad
1952   {\footnotesize #2}
1953 }

1954 \def\liFussnoteEinsText{
1955   \li@fussnote@text{1}
1956   {Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.}
1957 }

1958 \def\liFussnoteZweiText{
1959   \li@fussnote@text{2}
1960   {Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt.}
1961 }

1962 \def\liFussnoteDreiText{
1963   \li@fussnote@text{3}

```

```

1964 {In weiteren Iterationen markierte Zustände.}
1965 }

\liFussnoteVierText
1966 \def\liFussnoteVierText{
1967   \li@fussnote@text{4}
1968   {...}
1969 }

\liFussnoten


|       |                                                                      |
|-------|----------------------------------------------------------------------|
| $x_1$ | Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.          |
| $x_2$ | Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt. |
| $x_3$ | In weiteren Iterationen markierte Zustände.                          |
| $x_4$ | ...                                                                  |



1970 \def\liFussnoten{
1971   \bigskip
1972
1973   \noindent
1974   \liFussnoteEinsText
1975
1976   \noindent
1977   \liFussnoteZweiText
1978
1979   \noindent
1980   \liFussnoteDreiText
1981
1982   \noindent
1983   \liFussnoteVierText
1984 }

\liLeereZelle \liLeereZelle:  $\emptyset$ 
Let-Abkürzung: \let\l=\liLeereZelle
1985 \def\liLeereZelle{\$\emptyset\$}

\liZustandsPaarVariablenName
1986 \def\liZustandsPaarVariablenName{z}

\liZustandsPaar
1987 \def\liZustandsPaar#1#2{
1988   $(
1989     \liZustandsPaarVariablenName_#1,
1990     \liZustandsPaarVariablenName_#2
1991   )$
1992 }

liUebergangsTabelle
1993 \renewcommand{\arraystretch}{1.4}
1994 \NewDocumentEnvironment{ liUebergangsTabelle } { m m } {
1995   \liPseudoUeberschrift{Übergangstabelle}
1996   \begin{center}
1997     \begin{tabular}{r|l|l}
1998       \textbf{Zustandspaar} & \textbf{#1} & \textbf{#2} \\ \hline
1999     \end{tabular}
2000   \end{center}
2001
2002 }

liUeberschriftDreiecksTabelle \liUeberschriftDreiecksTabelle: Minimierungstabelle (Table filling)

2003 \ExplSyntaxOn
2004 \def\liUeberschriftDreiecksTabelle{
2005   \liPseudoUeberschrift{Minimierungstabelle~(Table~filling)}
2006 }

```

**\liMinimierungErklaerung** **Let-Abkürzung:** \let\verklaerung=\liMinimierungErklaerung

— Der Minimierungs-Algorithmus (auch Table-Filling-Algorithmus genannt) trägt in seinem Verlauf eine Markierung in alle diejenigen Zellen der Tabelle ein, die zueinander nicht äquivalente Zustände bezeichnen. Die Markierung „ $x_n$ “ in einer Tabellenzelle  $(i, j)$  bedeutet dabei, dass das Zustandspaar  $(i, j)$  in der  $k$ -ten Iteration des Algorithmus markiert wurde und die Zustände  $i$  und  $j$  somit zueinander  $(k - 1)$ -äquivalent, aber nicht  $k$ -äquivalent und somit insbesondere nicht äquivalent sind. Bleibt eine Zelle bis zum Ende unmarkiert, sind die entsprechenden Zustände zueinander äquivalent. —————

```

2007 \def\liMinimierungErklaerung{
2008   %\footcite[Seite~19]{koenig}
2009   \liParagraphMitLinien{
2010     Der~Minimierungs-Algorithmus~(auch~Table-Filling-Algorithmus-genannt)~
2011     trägt~in~seinem~Verlauf~eine~Markierung~in~alle~
2012     diejenigen~Zellen~der~Tabelle~ein,~die~zueinander~nicht~äquivalente~
2013     Zustände~bezeichnen.~Die~Markierung~„ $x_{\sb{n}}$ “~in~einer~Tabellenzelle~( $i$ ,~
2014      $j$ )~bedeutet~dabei,~dass~das~Zustandspaar~( $i$ ,~ $j$ )~in~der~ $k$ -ten~
2015     Iteration~des~Algorithmus~markiert~wurde~und~die~Zustände~ $i$ ~und~ $j$ ~
2016     somit~zueinander~( $k - 1$ )~äquivalent,~aber~nicht~ $k$ -äquivalent~und~
2017     somit~insbesondere~nicht~äquivalent~sind.~Bleibt~eine~Zelle~bis~zum~Ende~
2018     unmarkiert,~sind~die~entsprechenden~Zustände~zueinander~äquivalent.
2019   }
2020 }
2021 \ExplSyntaxOff
2022

```



## 2.30 normalformen.sty

```
2023 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2024 \ProvidesPackage{lehramt-informatik-normalformen}[2020/12/10
2025 Hilfsmakros zum Setzen von Normalformen, Funktionalen Abhängigkeiten,
2026 Attributhülle]
```

Mathe für textit tex etc, typographie für geschweifte Klammer

```
2027 \liLadePakete{mathe,typographie}
```

```
2028 \directlua{
2029   helfer = require('lehramt-informatik-helfer')
2030   normalformen = require('lehramt-informatik-normalformen')
2031 }
```

### 2.30.1 Makro-Kürzel

```
\let\ah=\liAttributHuelle
\let\ahL=\liLinksReduktion
\let\ahl=\liLinksReduktionInline
\let\ahr=\liRechtsReduktionInline
\let\fa=\liFunktionaleAbhaengigkeit
\let\FA=\liFunktionaleAbhaengigkeiten
\let\m=\liAttributMenge
\let\r=\liRelation
\let\u=\underline
```

```
2032 \def\liTeilen#1{
2033   \directlua{tex.print(normalformen.teilen('#1'))}
2034 }
```

**\liAttributHuelle Let-Abkürzung:**  $\text{let}\ \text{ah}=\text{liAttributHuelle}$   
 $\text{ah}\{F, \text{m}\{A, B\}\}$  AttrHülle( $F, \{A, B\}$ ) Regulärer Ausdruck zum Konvertieren  
 $\text{AttrHülle}((.*)\backslash)\ \backslash\text{ah}\{\$1\}$

```
2035 \def\liAttributHuelleOhneMathe#1{\text{AttrHülle}(\#1)}
2036 \def\liAttributHuelle#1{
2037   \ifmmode
2038     \liAttributHuelleOhneMathe{\#1}
2039   \else
2040     $\liAttributHuelleOhneMathe{\#1}$
2041   \fi
2042 }
```

**\liAttributMenge Let-Abkürzung:**  $\text{let}\ \text{m}=\text{liAttributMenge}$   
2043  $\backslash\text{def}\ \backslash\text{liAttributMenge}\#1\{\backslash\{\ \backslash\text{textit}\{\#1\}\ \backslash\}$

**liAHuelle**

```
2044 \NewDocumentEnvironment{ liAHuelle } { +b } {
2045   \begingroup
2046   \footnotesize
2047   \begin{multline*}
2048     \#1
2049   \end{multline*}
2050   \endgroup
2051 } { }
```

**\liLinksReduktion** Nur innerhalb von liAHuelle zu verwenden bzw. multline

**Let-Abkürzung:**  $\text{let}\ \text{ahL}=\text{liLinksReduktion}$   
 $\text{ahL}\{\text{ursprüngliche linke Attributmenge}\{\text{ohne dieses Attribut}\}\{\text{Ergebnis}\}$   
2052  $\backslash\text{def}\ \backslash\text{liLinksReduktion}\#1\#2\#3\{$   
2053  $\backslash\text{shoveleft}\{$   
2054  $\backslash\text{liAttributHuelleOhneMathe}\{\text{FA},$   
2055  $\backslash\text{liAttributMenge}\{\#1\ \backslash\text{string}\ \#2\}\} =$   
2056  $\}\ \backslash\backslash$

```

2057 \shoveright{
2058   \liAttributMenge{#3}
2059 } \\\
2060 }

```

```

\liLinksReduktionInline Let-Abkürzung: \let\ahL=\liLinksReduktionInline
    \ahl{ursprüngliche linke Attributmenge}{ohne dieses Attribut}{Ergebnis}
\ahl{A, E}{E}{A, E, F, B, \textbf{D}}: AttrHülle( $F, \{A, E \setminus E\}$ ) =  $\{A, E, F, B, D\}$ 
2061 \def\liLinksReduktionInline#1#2#3{%
2062   {%
2063     \footnotesize%
2064     $\liAttributHuelleOhneMathe{F,
2065       \liAttributMenge{#1} \string\ #2}} =
2066     \liAttributMenge{#3}$
2067   }
2068 }

```

```

\liLinksReduktionInline Let-Abkürzung: \let\ahr=\liLinksReduktionInline
    \ahr{gelöschte FA}{neue FA ohne rechts Attribut}{gegebene Attribute}{Ergebnis}
2069 \def\liRechtsReduktionInline#1#2#3#4{%
2070   {%
2071     \footnotesize%
2072     $\liAttributHuelleOhneMathe{
2073       F \setminus
2074       \liFunktionaleAbhaengigkeit{#1}
2075       \def\tmp{#2}\ifx\tmp\empty
2076       \else
2077         \cup \liFunktionaleAbhaengigkeit{#2}
2078       \fi
2079       ,
2080       \liAttributMenge{#3}
2081     } =
2082     \liAttributMenge{#4}$
2083   }
2084 }

```

```

\liFunktionaleAbhaengigkeit Let-Abkürzung: \let\fa=\liFunktionaleAbhaengigkeit
    \fa{A, B -> C, D}:  $\{A, B\} \rightarrow \{C, D\}$  Regulärer Ausdruck zum Konvertieren:
    \$(.*?) \\\rightarrow (.*)\$ \fa{$1 -> $2}
2085 \def\liFunktionaleAbhaengigkeit#1{%
2086   \directlua{normalformen.drucke_funk_abhaengigkeit('#1', false)}%
2087 }

```

```

\liFunktionaleAbhaengigkeiten \FA[F]{
    M -> M;
    M -> N;
    V -> T, P, PN;
    P -> PN;
}

```

$$F = \left\{ \begin{array}{l} \{M\} \rightarrow \{M\}, \\ \{M\} \rightarrow \{N\}, \\ \{V\} \rightarrow \{T, P, PN\}, \\ \{P\} \rightarrow \{PN\}, \end{array} \right\}$$

```

Let-Abkürzung: \let\FA=\liFunktionaleAbhaengigkeiten
    Regulärer Ausdruck zum Konvertieren: \item \$(.*) \\\rightarrow (.*)\$ \fa{(.*) \\\rightarrow
$1 -> $2;

```

```

2088 \NewDocumentCommand {\liFunktionaleAbhaengigkeiten} { O{FA} m } {
2089   \liGeschweifteKlammern
2090   {#1}
2091   {
2092     \begin{align*}
2093       \directlua{normalformen.drucke_funk_abhaengigkeiten('#2')}
2094     \end{align*}
2095   }
2096   {-0.5cm}
2097   {-1.7cm}
2098 }

\liRelation Let-Abkürzung: \let\r=\liRelation
      \r[R3]{\u{A}, B, C}:  $R_3(\underline{A}, B, C)$  Regulärer Ausdruck zum Konvertieren:
      \$(R.*)\((.*)\)\$ \liRelation[$1]{$2}

2099 \NewDocumentCommand {\liRelation} { O{R} m } {
2100   $\directlua{
2101     local name = helper.konvertiere_tiefgestellt('#1')
2102     tex.print(name)
2103   }$(\textit{\, #2\,})
2104 }

2105

```

## 2.31 petri.sty

2106 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2107 \ProvidesPackage{lehramt-informatik-petri}[2020/12/03]

### 2.31.1 Makro-Kürzel

\let\t=\liPetriTransitionsName

\let\tp=\liPetriTransPfeile

\let\k=\liPetriErreichKnotenDrei

2108 \RequirePackage{tikz}

2109 \usetikzlibrary{petri,arrows.meta}

Für die Darstellungsmatrix

2110 \RequirePackage{blkarray}

```
\def\TmpA#1{
  \liPetriSetzeSchluessel%
  \pgfkeys{/petri/.cd,#1}%
  \begin{tikzpicture}[li petri]
    \node at (-0.25,-0.25) {};
    \node at (\TmpX,\TmpY) {};

    \begin{scope}[transform canvas={scale=\TmpScale},x=2cm,y=2cm,]
      \node[place,tokens=\TmpPlaceOne,label=$p_1$] at (0,1) (p1) {};
      \node[place,tokens=\TmpPlaceTwo,label=$p_2$] at (2,2) (p2) {};
      \node[place,tokens=\TmpPlaceThree,label=east:$p_3$] at (2,0) (p3) {};

      \node[transition,label=east:$t_1$, \TmpTransitionOne] at (2,1) {}
        edge[pre] (p2)
        edge[post] (p3);
      \node[transition,label=$t_2$, \TmpTransitionTwo] at (1,1.5) {}
        edge[pre] (p1)
        edge[post] (p2);
      \node[transition,label=$t_3$, \TmpTransitionThree] at (1,0.5) {}
        edge[pre] (p3)
        edge[post] (p1);
      \node[transition,label=$t_4$, \TmpTransitionFour] at (1,1) {}
        edge[pre] (p2)
        edge[pre] (p3)
        edge[post] (p1);
    \end{scope}
  \end{tikzpicture}
}
```

\liPetriSetzeSchluessel

```
2111 \def\liPetriSetzeSchluessel{%
2112   \def\TmpTransitionOne{}%
2113   \def\TmpTransitionTwo{}%
2114   \def\TmpTransitionThree{}%
2115   \def\TmpTransitionFour{}%
2116   \def\TmpTransitionFive{}%
2117   \def\TmpTransitionSix{}%
2118   \def\TmpTransitionSeven{}%
2119   \def\TmpTransitionEight{}%
2120   \def\TmpTransitionNine{}%
2121   \def\TmpTransitionTen{}%
2122   \pgfkeys{/petri/.cd,
2123     p1/.store in=\TmpPlaceOne,p1/.default=0,p1,
2124     p2/.store in=\TmpPlaceTwo,p2/.default=0,p2,
2125     p3/.store in=\TmpPlaceThree,p3/.default=0,p3,
2126     p4/.store in=\TmpPlaceFour,p4/.default=0,p4,
```

```

2127 p5/.store in=\TmpPlaceFive,p5/.default=0,p5,
2128 p6/.store in=\TmpPlaceSix,p6/.default=0,p6,
2129 p7/.store in=\TmpPlaceSeven,p7/.default=0,p7,
2130 p8/.store in=\TmpPlaceEight,p8/.default=0,p8,
2131 p9/.store in=\TmpPlaceNine,p9/.default=0,p9,
2132 p10/.store in=\TmpPlaceTen,p10/.default=0,p10,
2133 t1/.store in=\TmpTransitionOne,t1/.default=activated,
2134 t2/.store in=\TmpTransitionTwo,t2/.default=activated,
2135 t3/.store in=\TmpTransitionThree,t3/.default=activated,
2136 t4/.store in=\TmpTransitionFour,t4/.default=activated,
2137 t5/.store in=\TmpTransitionFive,t5/.default=activated,
2138 t6/.store in=\TmpTransitionSix,t6/.default=activated,
2139 t7/.store in=\TmpTransitionSeven,t7/.default=activated,
2140 t8/.store in=\TmpTransitionEight,t8/.default=activated,
2141 t9/.store in=\TmpTransitionNine,t9/.default=activated,
2142 t10/.store in=\TmpTransitionTen,t10/.default=activated,
2143 scale/.store in=\TmpScale,scale/.default=0.5,
2144 x/.store in=\TmpX,x/.default=5,
2145 y/.store in=\TmpY,y/.default=5,
2146 }%
2147 }

```

```

2148 \tikzset{
2149   li petri/.style={
2150     activated/.style={
2151       very thick
2152     },
2153     inhibitor/.style={
2154       {Circle[open,length=2mm,fill=white]}-
2155     }
2156   }
2157 }

```

**\liPetriTransitionsName** **Let-Abkürzung:** \let\t=\liPetriTransitionsName  
 \t\_(\d+)\\$ \t\$1

```

2158 \def\liPetriTransitionsNameOhneMathe#1{t\sb{#1}}
2159 \def\liPetriTransitionsName#1{
2160   \ifmmode
2161     \liPetriTransitionsNameOhneMathe{#1}
2162   \else
2163     $\liPetriTransitionsNameOhneMathe{#1}$
2164   \fi
2165 }

```

**\liPetriErreichTransition** **Let-Abkürzung:** \let\t=\liPetriErreichTransition

```

2166 \NewDocumentCommand{ \liPetriErreichTransition } { m m m O{} O{} } {
2167   \draw[->] (#1) edge[#4] node[pos=0.5,auto,sloped,#5]{t\sb{#3}$} (#2);
2168 }

```

**\liPetriErreichKnotenDrei** **Let-Abkürzung:** \let\k=\liPetriErreichKnotenDrei

```

2169 \def\liPetriErreichKnotenDrei#1#2#3{(#1,#2,#3)}

```

**\liPetriTransPfeile** **Let-Abkürzung:** \let\tp=\liPetriTransPfeile

```

2170 \def\liPetriTransPfeile#1{ $\rightarrow \hspace{0.4cm} \liPetriTransitionsName{#1} \hspace{0.3cm}
2171 }

```

## 2.32 potenzmengen-konstruktion.sty

```
2172 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2173 \ProvidesPackage{lehramt-informatik-potenzmengen-konstruktion}[2021/02/21 Hilfsmakros
2174 einzusetzen bei der Potenzmengen-Konstruktion (Potenzmengen-Algorithmus)]
```

```
2175 \liLadePakete{formale-sprachen}
2176 \ExplSyntaxOn
```

```
\def\z#1{
  \liZustandsMengenSammlungNr{#1}{
    {
      {0} {0}
      {1} {0,1}
      {2} {0,2}
      {3} {0,1,3}
      {4} {0,2,3}
      {5} {0,3}
    }
  }
}

\let\s=\liZustandsnameGross

\begin{tabular}{l|l|l}
Zustandsmenge & Eingabe $a$ & Eingabe $b$ \\ \hline
\z0 & \z0 & \z1 \\
\z1 & \z2 & \z1 \\
\z2 & \z0 & \z3 \\
\z3 & \z4 & \z3 \\
\z4 & \z5 & \z3 \\
\z5 & \z5 & \z3 \\
\end{tabular}
```

```
\liZustandsMengenSammlung \liZustandsMengenSammlung{nummer}{latex3 str_case:nn}
```

```
\def\z#1{
  \liZustandsMengenSammlung{#1}{
    {
      {0} {z0}
      {1} {z0, z1}
      {2} {z0, z1, z2}
      {3} {z0, z2}
      {4} {z0, z1, z2, z3}
      {5} {z0, z3}
      {6} {z0, z2, z3}
      {7} {z0, z1, z3}
    }
  }
}
```

```
2177 \def\liZustandsMengenSammlung#1#2{
2178   \liZustandsnameGross{#1}
2179   {
2180     \footnotesize
2181     \liPotenzmenge{
2182       \str_case:nn {#1} {#2
2183       }
2184     }
2185 }
```

```
\liZustandsMengenSammlungNr
```

```
2186 \def\liZustandsMengenSammlungNr#1#2{
2187   \liZustandsnameGross{#1}
2188   {
```

```
2189     \footnotesize
2190     \liZustandsmengeNr{
2191         \str_case:nn {#1} #2
2192     }
2193 }
2194 }

2195 \ExplSyntaxOff
2196
```

## 2.33 pseudo.sty

2197 \NeedsTeXFormat{LaTeX2e}[1995/12/01]  
 2198 \ProvidesPackage{lehramt-informatik-pseudo}[2020/12/30 Paket zum Setzen  
 2199 von Pseudo-Code, Hüll-Paket um algorithm2e]

```
\begin{algorithm}[H]
\KwData{$G = (V, E, w)$: ein zusammenhängender, ungerichteter,
kantengewichteter Graph kruskal(G)}
$E' \leftarrow \emptyset$;
$L \leftarrow E$;
Sortiere die Kanten in $L$ aufsteigend nach ihrem Kantengewicht.;
\While{$L \neq \emptyset$}{
  wähle eine Kante $e \in L$ mit kleinstem Kantengewicht;
  entferne die Kante $e$ aus $L$;
  \If{der Graph $(V, E' \cup \{e\})$ keinen Kreis enthält}{
    $E' \leftarrow E' \cup \{e\}$;
  }
}
\KwResult{$M = (V, E')$ ist ein minimaler Spannbaum von $G$.}
\caption{Minimaler Spannbaum nach Kruskal\footcite{wiki:kruskal}}
\end{algorithm}
```

Algorithmus 1: Minimaler Spannbaum nach Kruskal
---

<p><b>Data:</b> <math>G = (V, E, w)</math>: ein zusammenhängender, ungerichteter, kantengewichteter Graph kruskal(<math>G</math>)</p> <p><math>E' \leftarrow \emptyset</math>;  <math>L \leftarrow E</math>;          Sortiere die Kanten in <math>L</math> aufsteigend nach ihrem Kantengewicht.;</p> <p><b>while</b> <math>L \neq \emptyset</math> <b>do</b></p> <div style="margin-left: 20px;"> <p>wähle eine Kante <math>e \in L</math> mit kleinstem Kantengewicht;</p> <p>entferne die Kante <math>e</math> aus <math>L</math>;</p> <p><b>if</b> der Graph <math>(V, E' \cup \{e\})</math> keinen Kreis enthält <b>then</b></p> <div style="margin-left: 20px;"> <p><math>E' \leftarrow E' \cup \{e\}</math>;</p> </div> <p><b>end</b></p> </div> <p><b>end</b></p> <p><b>Result:</b> <math>M = (V, E')</math> ist ein minimaler Spannbaum von <math>G</math>.</p>
---

2200 \RequirePackage[german,boxruled]{algorithm2e}

2201



## 2.34 pumping-lemma.sty

2202 \NeedsTeXFormat{LaTeX2e}[1995/12/01]  
 2203 \ProvidesPackage{lehramt-informatik-pumping-lemma}[2021/08/11 Enthält die  
 2204 Definitionen für das Pumping-Lemma in der Regulären Sprache und  
 2205 in der Kontextfreien Sprache]

\liPumpingRegulaer

```
2206 \def\liPumpingRegulaer{%
2207   Es sei  $L$  eine reguläre Sprache. Dann gibt es eine Zahl  $j$ , sodass für
2208   alle Wörter  $\omega$  in  $L$  mit  $|\omega| \geq j$  (jedes Wort  $\omega$  in
2209    $L$  mit Mindestlänge  $j$ ) jeweils eine Zerlegung  $\omega = uvw$ 
2210   existiert, sodass die folgenden Eigenschaften erfüllt sind:
2211
2212   \begin{enumerate}
2213     \item  $|v| \geq 1$ 
2214     (Das Wort  $v$  ist nicht leer.)
2215
2216     \item  $|uv| \leq j$ 
2217     (Die beiden Wörter  $u$  und  $v$  haben zusammen höchstens die Länge  $j$ .)
2218
2219     \item Für alle  $i = 0, 1, 2, \dots$  gilt  $uv^i w$  in  $L$ 
2220     (Für jede natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^i w$  in der
2221     Sprache  $L$ )
2222   \end{enumerate}
2223
2224   Die kleinste Zahl  $j$ , die diese Eigenschaften erfüllt, wird
2225   Pumping-Zahl der Sprache  $L$  genannt.\footcite{wiki:pumping-lemma}
2226 }
```

\liPumpingKontextfrei

```
2227 \def\liPumpingKontextfrei{%
2228   Es sei  $L$  eine kontextfreie Sprache. Dann gibt es eine Zahl  $j$ , sodass
2229   sich alle Wörter  $\omega$  in  $L$  mit  $|\omega| \geq j$  zerlegen lassen in
2230    $\omega = uvwxy$ , sodass die folgenden Eigenschaften erfüllt sind:
2231
2232   \begin{enumerate}
2233     \item  $|vx| \geq 1$ 
2234     (Die Wörter  $v$  und  $x$  sind nicht leer.)
2235
2236     \item  $|vwx| \leq j$ 
2237     (Die Wörter  $v$ ,  $w$  und  $x$  haben zusammen höchstens die Länge  $j$ .)
2238
2239     \item Für alle  $i \in \mathbb{N}_0$  gilt  $uv^iwx^iy$  in  $L$  (Für jede
2240     natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^iwx^iy$  in der
2241     Sprache  $L$ )
2242   \end{enumerate}
2243 }
2244
```

## 2.35 quicksort.sty

```

2245 % https://tex.stackexchange.com/a/142634
2246 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2247 \ProvidesPackage{lehramt-informatik-quicksort}[2020/06/12]
2248
2249 %-----
2250 % USAGE:
2251 % \QSinitialize{comma, separated, numerical, values}
2252 % \loop
2253 % \QSpivotStep
2254 % \ifnum\value{pivotcount}>0
2255 %   \QSSortStep
2256 % \repeat
2257 %-----
2258
2259 % xintfrac does not load xinttools, this must be done explicitly if needed as here.
2260 \RequirePackage{xintfrac, xinttools}
2261
2262 \RequirePackage{tikz}
2263
2264 %-----
2265 % FIRST PART: TikZ styles and macros for the actual drawing
2266 \newcounter{cellcount}% used for coordinates of the node
2267 \newcounter{pivotcount}% when it will remain at zero, will signal the sort is finished.
2268
2269 % Styles defined by Tom Bombaldi. (modified: all share the same size)
2270 % (re-modified \bf -> \bfseries due to extremely annoying warnings from
2271 % KOMA-script which are truly a pain and do not make any sense regarding \bf:
2272 % if I want to use \bf, and know what I am doing, why should I get HARASSED
2273 % by police of LaTeX good conduct ? )
2274 \tikzset{l/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=lime!70!gray},
2275          o/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=olive!50},
2276          r/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2277 % this is the "b" style as used in the image below
2278          b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2279 % nicer:
2280          b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=white, text=magenta},
2281          g/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=gray, text=white, text=white},
2282
2283 % NOTE the b style was originally the same as the r(aised) style apart from
2284 % not being raised, but I find it nicer with a somewhat different
2285 % specification. I have not updated the images though.
2286
2287 % How the nodes are drawn depending on whether on the left of the pivot value
2288 % or on the right, or is a pivot value, or a raised pivot during selection phase.
2289
2290 \def\DecoLEFT #1{%
2291   \xintFor* ##1 in {#1} \do
2292     {\stepcounter{cellcount}\node[o] at (\arabic{cellcount},0) {##1};}%
2293 }
2294
2295 \def\DecoINERT #1{%
2296   \xintFor* ##1 in {#1} \do
2297     {\stepcounter{cellcount}\node[g] at (\arabic{cellcount},0) {##1};}%
2298 }
2299
2300 \def\DecoRIGHT #1{%
2301   \xintFor* ##1 in {#1} \do
2302     {\stepcounter{cellcount}\node[l] at (\arabic{cellcount},0) {##1};}%
2303 }
2304
2305 \def\DecoLEFTwithPivot #1{\stepcounter{pivotcount}%
2306   \xintFor* ##1 in {#1} \do

```

```

2307     {\stepcounter{cellcount}}%
2308     \xintifForLast {\node[r]}\node[o]} at (\arabic{cellcount},0) {##1};}%
2309 }
2310
2311 \def\DecoINERTwithPivot #1{\stepcounter{pivotcount}%
2312     \xintFor* ##1 in {#1} \do
2313     {\stepcounter{cellcount}}%
2314     \xintifForLast {\node[b]}\node[g]} at (\arabic{cellcount},0) {##1};}%
2315 }
2316
2317 \def\DecoRIGHTwithPivot #1{\stepcounter{pivotcount}%
2318     \xintFor* ##1 in {#1} \do
2319     {\stepcounter{cellcount}}%
2320     \xintifForLast {\node[r]}\node[l]} at (\arabic{cellcount},0) {##1};}%
2321 }
2322
2323 %-----
2324 % SECOND PART: the actual sorting routines.
2325
2326 \def\QS@sort@a #1{\expandafter \QS@sort@b \expandafter {\xintLength {#1}}{#1}}
2327 \def\QS@sort@b #1{\ifcase #1
2328     \expandafter\QS@sort@empty
2329     \or\expandafter\QS@sort@single
2330     \else\expandafter\QS@sort@c
2331     \fi
2332 }%
2333 \def\QS@sort@empty #1{}
2334 \def\QS@sort@single #1{\QSIr {#1}}
2335
2336 % This step is to pick the last as pivot.
2337 \def\QS@sort@c #1%
2338     {\expandafter\QS@sort@d\expandafter {\romannumeral0\xintnthelt {-1}{#1}}{#1}}%
2339
2340 % Here \QSLr, \QSIr, \QSR have been let to \relax.
2341 % The trick with \xintApplyUnbraced is that for example when selecting
2342 % the elements smaller than pivot, if we had been using \xintApply we
2343 % would have had at the minimum an empty brace pair. Thus we use the
2344 % "unbraced" variant, but then the \QS@select@smaller has added in
2345 % anticipation a level of braces.
2346 \def\QS@sort@d #1#2{%
2347     \QSLr {\xintApplyUnbraced {\QS@select@smaller {#1}}{#2}}}%
2348     \QSIr {\xintApplyUnbraced {\QS@select@equal {#1}}{#2}}}%
2349     \QSRr {\xintApplyUnbraced {\QS@select@greater {#1}}{#2}}}%
2350 }%
2351 \def\QS@select@smaller #1#2{\xintifLt {#2}{#1}{#{#2}}{ }}% space will stop a f-
    expansion
2352 \def\QS@select@equal #1#2{\xintifEq {#2}{#1}{#{#2}}{ }}% space will stop a f-
    expansion
2353 \def\QS@select@greater #1#2{\xintifGt {#2}{#1}{#{#2}}{ }}% space will stop a f-
    expansion
2354
2355 %
2356 % NOTE 1: thus, each comparison with the pivot is done three (!) times.
2357 %
2358 % NOTE 2: we may well end up with \QSLr {<empty>} situations. This is handled
2359 % silently by the \xintFor loops, and also when \QSLr becomes \QS@sort@a, the
2360 % latter must handle correctly an empty argument.
2361
2362 %-----
2363 % THIRD PART: the main macros \QSpivotStep, \QSsortStep and \QSinitialize.
2364
2365 % This draws all with suitable highlighting for the newly chosen pivots
2366 % (which will be shown raised)

```

```

2367 \def\QSpivotStep {\let\QSLr\DecoLEFTwithPivot
2368             \let\QSIr\DecoINERT
2369             \let\QSIrr\DecoINERT
2370             \let\QSRr\DecoRIGHTwithPivot
2371 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2372             \setcounter{cellcount}{0}\setcounter{pivotcount}{0}%
2373             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2374 }
2375
2376 % This sorts and then draws, showing where the pivot chosen in the previous
2377 % step go. Next time they will have become "inert". If pivotcount is still at
2378 % zero on exit from \QSpivotStep, then this is the signal to stop before
2379 % executing \QSSortStep.
2380 \def\QSSortStep {\def\QSLr {\noexpand\QS@sort@a}%
2381             \def\QSRr {\noexpand\QS@sort@a}%
2382             \def\QSIr {\noexpand\QSIrr}%
2383             \let\QSIrr\relax
2384             \edef\QS@list{\QS@list}%
2385             \let\QSLr\relax
2386             \let\QSRr\relax
2387             \let\QSIr\relax
2388             \edef\QS@list{\QS@list}%
2389             \let\QSLr\DecoLEFT
2390             \let\QSIr\DecoINERTwithPivot
2391             \let\QSIrr\DecoINERT
2392             \let\QSRr\DecoRIGHT
2393 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2394             \setcounter{cellcount}{0}%
2395             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2396 }
2397
2398 \def\QSinitialize #1{%
2399     % first, we convert the comma separated values into a list of braced items
2400     % we use an \edef, and anyhow many \edef's will be used later
2401     \edef\QS@list {\noexpand\QSRr {\xintCSVtoList {#1}}}%
2402     \let\QSRr\DecoRIGHT
2403     % The \QSRr marker mutated to draw the last element as
2404     % pivot and the earlier ones with the suitable style.
2405     %
2406     % The list of marked braced items \QS@list is used both for drawing
2407     % (as here) and for doing the exchange of elements during sort.
2408     \par\centerline{\rule[1.5mm]{0pt}{8mm}\setcounter{cellcount}{0}%
2409             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2410 }
2411

```

## 2.36 relationale-algebra.sty

```

2412 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2413 \ProvidesPackage{lehramt-informatik-relationale-algebra}[2020/12/11]
2414 \RequirePackage{amsmath}
2415 \RequirePackage{amssymb}

```

```

    Zum Zeichnen von Operatorenbäumen verwenden wir TikZ
\begin{tikzpicture}
  \node
    (pi) {\pi_{\text{Kunde.Name, Kunde.Geburtsdatum}}};

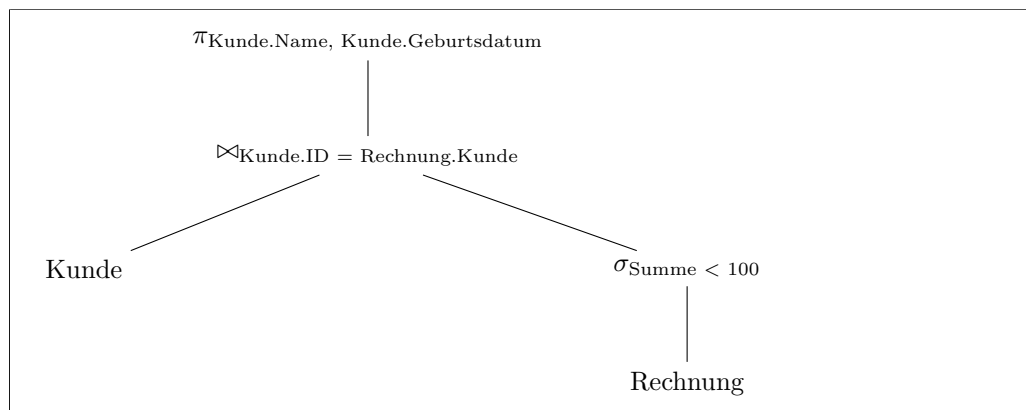
  \node[below=of pi]
    (theta join) {\bowtie_{\text{Kunde.ID = Rechnung.Kunde}}};
    edge (pi);

  \node[below left=of theta join]
    {Kunde}
    edge(theta join);

  \node[below right=of theta join]
    (sigma rechnung) {\sigma_{\text{Summe < 100}}};
    edge (theta join);

  \node[below=of sigma rechnung]
    {Rechnung}
    edge(sigma rechnung);
\end{tikzpicture}

```



```

2416 \RequirePackage{tikz}
2417 \usetikzlibrary{positioning}

    Privates Makros, das zwei Querstriche erzeugt.
2418 \def\o@join{\setbox0=\hbox{\bowtie}}%
2419   \rule[-.02ex]{.25em}{.4pt}\llap{\rule[\ht0]{.25em}{.4pt}}%
2420 }

\leftouterjoin A \leftouterjoin B: A ⋈ B
2421 \def\leftouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie}}

\rightouterjoin A \rightouterjoin B: A ⋉ B
2422 \def\rightouterjoin{\mathbin{\bowtie\mkern-5.8mu\o@join}}

\fullouterjoin A \fullouterjoin B: A ⋈⋉ B
2423 \def\fullouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie\mkern-5.8mu\o@join}}
2424

```

## 2.37 rmodell.sty

```
2425 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2426 \ProvidesPackage{lehramt-informatik-rmodell}[2020/09/01]
2427 Makros und Umgebungen zum Setzen des Relationenmodells beim Thema
2428 Datenbanken.]
2429 \RequirePackage{soul}
```

### 2.37.1 Makro-Kürzel

```
\let\a=\liAttribut
\let\f=\liFremd
\let\p=\liPrimaer
\let\r=\liRelationMenge
```

`\liPrimaer` `\liPrimaer{text}`: Unterstreichung für den Primärschlüssel

```
2430 \def\liPrimaer#1{\ul{#1}}
```

`\liFremd` `\liFremd{text}`: Überstreichung für den Fremdschlüssel

```
2431 \def\liFremd#1{{\setul{-0.9em}{}\ul{#1}}}
```

`liRmodell` `\begin{liRmodell}` `\end{liRmodell}`: Kleinere Schrift und Schreibmaschinenschrift.

```
2432 \def\li@Rmodell@Schrift{\footnotesize\ttfamily}
2433 \ExplSyntaxOn
2434 \NewDocumentEnvironment { liRmodell }
2435 { +b }
2436 {
2437   \medskip
2438   {
2439     \linespread{2}
2440     \setlength{\parindent}{0pt}
2441     \li@Rmodell@Schrift#1
2442   }
2443   \medskip
2444 } {}
2445 \ExplSyntaxOff
```

`\liRelationMenge` **Let-Abkürzung:** `\let\r=\liRelationMenge`

`\liRelationMenge{name}{attribut, attribut}`: Umhüllen der Attribute mit geschweiften und dann eckigen Klammern.

```
2446 \def\liRelationMenge#1#2{
2447 \noindent
2448 #1 : \[ #2 ]\}
2449 \par
2450 }
```

`\liAttribut` **Let-Abkürzung:** `\let\a=\liAttribut`

`\liAttribut{text}`: Gleiche Schrift wie Umgebung `liRmodell`

```
2451 \def\liAttribut#1{{\li@Rmodell@Schrift#1}}
```

`liRelationenSchemaFormat` Dummy-Umgebung, zum Parsen durch Java gedacht.

```
\begin{liRelationenSchemaFormat}
Springer(Startnummer*, Nachname, Vorname, Geburtsdatum, Körpergröße)
Sprung(SID*, Beschreibung, Schwierigkeit)
springt(SID[Sprung], Startnummer[Springer], Durchgang)
\end{liRelationenSchemaFormat}
```

```
2452 \NewDocumentEnvironment { liRelationenSchemaFormat }{ +b } {} {}
```

```
2453
```

## 2.38 sortieren.sty

```
2454 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2455 \ProvidesPackage{lehramt-informatik-sortieren}[2020/06/10 Schaubilder
2456 für händisches Sortieren bzw. einen Schreibtischlauf setzen]
```

```
\tikz[
  rectangle split parts=5,
]{
  \node[li sortierung zahlenreihe] (reihe) {\nodepart{one} 2 \nodepart{two} 1 \nodepart{three} 3 \nodepart{four} 4 \nodepart{five} 5}
  \liSortierPfeil{one}{two}
  \liSortierPfeil{two}{three}
  \liSortierMarkierung{two split south}{three split north}
  \liSortierPfeilUnten{three}{one}
}
```



```
2457 \RequirePackage{tikz}
2458 \usetikzlibrary{shapes.multipart,positioning}
```

`\liVertauschen` `\liVertauschen{1 2 >4 <3 5}`: Setze ein Schaubild mit Hilfe von TikZ. < und > werden dazu verwendet, um den Vertauschprozess zu visualisieren.

```
2459 \def\liVertauschen#1{
2460   \directlua{
2461     local sortieren = require('lehramt-informatik-sortieren')
2462     sortieren('#1')
2463   }
2464 }
```

`\liSortierPfeil`

```
2465 \def\liSortierPfeil#1#2{
2466   \draw[-latex] ([xshift=1mm]reihe.#1 north) -- ++(0,0.25) -| ([xshift=-1mm]reihe.#2 north);
2467 }
```

`\liSortierPfeilUnten`

```
2468 \def\liSortierPfeilUnten#1#2{
2469   \draw[-latex] ([xshift=1mm]reihe.#1 south) -- ++(0,-0.25) -| ([xshift=-1mm]reihe.#2 south);
2470 }
```

`\liSortierMarkierung`

```
2471 \def\liSortierMarkierung#1#2{\node[
2472   draw,
2473   very thick,
2474   fit=(reihe.#1) (reihe.#2),
2475   inner sep=0pt
2476 ] {}
2477 }

2478 \tikzset{
2479   li sortierung zahlenreihe/.style={
2480     draw,
2481     thin,
2482     font=\large,
2483     rectangle split horizontal,
2484     rectangle split,
2485   }
2486 }
```

```

2487 % https://tex.stackexchange.com/a/140895
2488 \RequirePackage{forest,xstring}
2489 \usetikzlibrary{calc}
2490
2491 \makeatletter
2492 \pgfmathdeclarefunction{strrepeat}{2}{%
2493   \begingroup\pgfmathint{#2}\pgfmath@count\pgfmathresult
2494   \let\pgfmathresult\pgfutil@empty
2495   \pgfutil@loop\ifnum\pgfmath@count>0\relax
2496     \expandafter\def\expandafter\pgfmathresult\expandafter{\pgfmathresult#1}%
2497     \advance\pgfmath@count-1\relax
2498   \pgfutil@repeat\pgfmath@smuggleone\pgfmathresult\endgroup}
2499 \makeatother
2500
2501 \def\myNodes{}
2502
2503 \ExplSyntaxOn
2504 \newcommand*\sortList[1]{%
2505   \clist_sort:Nn#1{\int_compare:nNnTF{##1}>{##2}\sort_return_swapped:\sort_return_same:}}
2506 \ExplSyntaxOff
2507
2508 \forestset{
2509   sort/.code={%
2510     \pgfmathparse{level()}>\forestSortLevel}%
2511     \ifnum\pgfmathresult=0
2512       \StrSubstitute{\forestov{content}}{ }{,}{\myList}%
2513       \sortList\myList
2514       \StrSubstitute{\myList}{,}{ }{\myList}%
2515       \pgfmathparse{strrepeat("1",level())}%
2516       \xappto\myNodes{\noexpand\node at ({\forestov{name}}|-m)!-!({\forestov{name}}$)
2517         (m\forestov{name}) {\myList}}%
2518       \pgfmathparse{level()}==\forestSortLevel}%
2519       \ifnum\pgfmathresult=1
2520         \forestOget{\forestov{@first}}{name}\forestFirst
2521         \forestOget{\forestov{@last}}{name}\forestLast
2522         \xappto\myNodes{{[<-]edge (\forestOv{\forestov{@first}}{name})
2523           \ifx\forestFirst\forestLast\else edge (\forestOv{\forestov{@last}}{name})\fi}}%
2524       \fi
2525       \ifnum\forestov{@parent}=0\else
2526         \xappto\myNodes{edge (m\forestOv{\forestov{@parent}}{name})}%
2527       \fi
2528       \gappto\myNodes{;}%
2529     \fi}}
2530
2531 \forestset{sort level/.code=%
2532   \pgfmathparse{#1}\let\forestSortLevel\pgfmathresult
2533   \pgfmathparse{strrepeat("1",\forestSortLevel+1)}\let\forestOnes\pgfmathresult}
2534
2535

```



## 2.39 spalten.sty

```
2535 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2536 \ProvidesPackage{lehramt-informatik-spalten}[2020/12/07 Lädt das Paket
2537 „multicol“, damit mehrspaltiger Satz mit Hilfe der Umgebung „multicols“
2538 realisiert werden kann.]
2539 \RequirePackage{multicol}
```

`\liSpaltenUmbruch` `\liSpaltenUmbruch`: Spezieller Spaltenumbruch, der den Inhalt mit Hilfe von `\vfill\strut` nach oben schiebt.

```
2540 \def\liSpaltenUmbruch{\vfill\strut\columnbreak}
```

```
2541
```

## 2.40 sql.sty

```
2542 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2543 \ProvidesPackage{lehramt-informatik-tabelle}[2021/09/02 Zu Setzen von SQL]
2544 \liLadePakete{syntax}
2545 \RequirePackage{fancyvrb}
2546 \DefineVerbatimEnvironment{liSqlErgebnis}{Verbatim}
2547 {fontsize=\footnotesize}
2548
```

## 2.41 struktogramm.sty

```
2549 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2550 \ProvidesPackage{lehramt-informatik-struktogramm}[2021/01/31 Lädt das
2551 Paket struktex zum Setzen von Struktogrammen]
2552 \RequirePackage{struktex}
2553
```

## 2.42 syntax.sty

```
2554 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2555 \ProvidesPackage{lehramt-informatik-syntax}[2020/06/10 Ein Hüll-Paket um
2556 `minted`, das sich um die Syntax-Hervorhebung des Java-Codes kümmert.]
2557 \RequirePackage{xparse}
```

Um das Paket `minted` benutzen zu können, muss `pygmentize` installiert sein. Außerdem müssen die TeX-Dateien mit der Kommandozeilen-Option `--shell-escape` kompiliert werden.

### 2.42.1 Makro-Kürzel

```
\let\j=\liJavaCode
\let\s=\liSqlCode
```

```
2558 \ExplSyntaxOn
2559 \directlua{
2560   syntax = require('lehramt-informatik-syntax')
2561   syntax.importiere_konfiguration('tex_repo_lokaler_pfad', '\LehramtInformatikRepository')
2562   syntax.importiere_konfiguration('github_domain', '\LehramtInformatikGithubDomain')
2563   syntax.importiere_konfiguration('github_raw_domain', '\LehramtInformatikGithubRawDomain')
2564   syntax.importiere_konfiguration('github_tex_repo', '\LehramtInformatikGithubTexRepo')
2565   syntax.importiere_konfiguration('github_code_repo', '\LehramtInformatikGithubCodeRepo')
2566   syntax.importiere_konfiguration('git_branch', '\LehramtInformatikGitBranch')
2567 }
2568 \RequirePackage{hyperref}
2569 \RequirePackage{minted}
2570 % pygmentize -L styles
2571 \usemintedstyle{colorful}
2572 %\BeforeBeginEnvironment{minted}{\begin{mdframed}}
2573 %\AfterEndEnvironment{minted}{\end{mdframed}}
2574 %\setminted{breaklines=true,linenos}
2575 \setminted{
2576   breaklines=true,
2577   linenos,
2578   fontsize=\footnotesize,
2579 }
```

`\liJavaCode` Im Zeilenfluss einen kurzen Java-Code-Ausschnitt setzen.

**Let-Abkürzung:** `\let\j=\liJavaCode`

```
2580 \def\liJavaCode#1{\,\mintinline{java}|#1|\,}
```

`\liLatexCode` Im Zeilenfluss einen kurzen L<sup>A</sup>T<sub>E</sub>X-Code-Ausschnitt setzen.

```
2581 \def\liLatexCode#1{\mintinline{latex}|#1|}
```

```
2582 \def\li@GithubLink#1#2{
2583   \begin{flushright}
2584     \tiny
2585     Code-Beispiel~auf~Github~ansehen:~
2586     \href{#1}{\nolinkurl{#2}}
2587   \end{flushright}
2588 }
```

`\liJavaDatei` Eine komplette Java-Datei einbinden, die Verzeichnis `./Code/src/main/java/org/bschlangaul` liegt.

```
2589 \NewDocumentCommand{\liJavaDatei}{0{firstline=3} m }{
2590   \inputminted[#1]{java}{
2591     \directlua{
2592       syntax.drucke_absoluten_pfad('#2', false)
2593     }
2594   }
2595   \li@GithubLink
```

```

2596     {\directlua{syntax.drucke_github_url('#2', false)}}
2597     {\directlua{syntax.drucke_relativen_repo_pfad('#2', false)}}
2598 }

\liJavaTestDatei Eine komplette Java-Test-Datei einbinden, die Verzeichnis ./Code/src/test/java/org/bschlangaul
liegt.
2599 \NewDocumentCommand{\liJavaTestDatei}{ 0{firstline=3} m }{
2600   \inputminted[#1]{java}{
2601     \directlua{
2602       syntax.drucke_absoluten_pfad('#2', true)
2603     }
2604   }
2605   \li@GithubLink
2606   {\directlua{syntax.drucke_github_url('#2', true)}}
2607   {\directlua{syntax.drucke_relativen_repo_pfad('#2', true)}}
2608 }

\liJavaExamen \liJavaExamen{66116}{2015}{03}{Kunde} Regulärer Ausdruck zum Konvertieren: \liJavaDatei([.
\liJavaExamen$1{$2}{$3}{$4}{$5}
2609 \NewDocumentCommand{\liJavaExamen}{ 0{firstline=3} m m m m }{
2610   \inputminted[#1]{java}{
2611     \directlua{
2612       syntax.drucke_absoluten_examens_pfad('#2', '#3', '#4', '#5')
2613     }
2614   }
2615   \li@GithubLink
2616   {\directlua{syntax.drucke_github_examens_url('#2', '#3', '#4', '#5')}}
2617   {\directlua{syntax.drucke_relativen_examens_repo_pfad('#2', '#3', '#4', '#5')}}
2618   {\directlua{syntax.drucke_relativen_examens_repo_pfad('#2', '#3', '#4', '#5')}}
2619 }

\liAssemblerCode
2620 \def\liAssemblerCode#1{\mintinline{asm}|#1|}

\liAssemblerDatei \liAssemblerDatei{relativer-pfad}: Relativer Pfad zu einer Assembler-Datei.
2621 \NewDocumentCommand{\liAssemblerDatei}{ m }{
2622   \inputminted{asm}{#1}
2623 }

\liMinispracheDatei \liMinispracheDatei{relativer-pfad}: Relativer Pfad zu einer Minisprachen-Datei
(Hochsprache für die Minimaschine von Albert Wiedemann).
2624 \NewDocumentCommand{\liMinispracheDatei}{ m }{
2625   \inputminted{componentpascal}{#1}
2626 }

\liHaskellCode \liHaskellCode{haskell}: Zum Setzen von Haskell-Code.
2627 \def\liHaskellCode#1{\mintinline{haskell}|#1|}

\liHaskellDatei \liHaskellDatei{relativer-pfad}: Relativer Pfad zu einer Haskell-Datei.
2628 \NewDocumentCommand{\liHaskellDatei}{ m }{
2629   \inputminted{haskell}{#1}
2630 }

2631 \ExplSyntaxOff

\liSqlCode \liHaskellCode{sql}: Zum Setzen von SQL-Code.
Let-Abkürzung: \let\s=\liSqlCode
2632 \def\liSqlCode#1{\mintinline{sql}|#1|}
2633

```

## 2.43 syntaxbaum.sty

```
2634 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2635 \ProvidesPackage{lehramt-informatik-syntaxbaum}[2021/02/14 Zum Setzen von
2636 Syntaxbäumen mit Hilfe des Pakets tikz-qtrees]
2637 \RequirePackage{tikz-qtrees}
2638
2639 \tikzset{li parsetree/.style={
2640     every internal node/.style={
2641         draw,circle
2642     },
2643     every leaf node/.style={
2644         draw,rectangle
2645     },
2646 }
2647 }
2648
```

## 2.44 synthese-algorithmus.sty

```
2649 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2650 \ProvidesPackage{lehramt-informatik-synthese-algorithmus}[2021/03/19
2651 Hilfsmakros zum Setzen des Synthese-Algorithmuses zur Umformung einer
2652 Relation in die 3. Normalform]

2653 \liLadePakete{normalformen,mathe,typographie}
2654 \ExplSyntaxOn
```

### 2.44.1 Makro-Kürzel

```
\let\schrittE=\liSyntheseUeberErklaerung
```

### 2.44.2 TeX-Markup Grundgerüst

```
\let\schrittE=\liSyntheseUeberErklaerung
\begin{enumerate}
\item \schrittE{1}

\begin{enumerate}
\item \schrittE{1-1}
\item \schrittE{1-2}
\item \schrittE{1-3}
\item \schrittE{1-4}
\end{enumerate}

\item \schrittE{2}
\item \schrittE{3}
\item \schrittE{4}
\end{enumerate}
```

### 2.44.3 TeX-Markup Linksreduktion

```
\let\ahl=\liLinksReduktionInline
\let\fa=\liFunktionaleAbhaengigkeit
\let\m=\liAttributMenge
\let\b=\textbf

\liPseudoUeberschrift{\fa{C, E -> D, X}}

 $\m{D, X} \in \ahl{C, E}{E}{A, C, B, \b{D, X}} \setminus$ 
 $\m{D, X} \notin \ahl{C, E}{C}{E, F}$ 

\liPseudoUeberschrift{\fa{C, E -> F}}

 $F \notin \ahl{C, E}{E}{A, C, B} \setminus$ 
 $F \in \ahl{C, E}{C}{E, \b{F}}$ 
```

### 2.44.4 TeX-Markup Rechtsreduktion

```
\let\ahr=\liRechtsReduktionInline
\let\fa=\liFunktionaleAbhaengigkeit

\liPseudoUeberschrift{F}

 $F \in \ahr{E -> F, X}{E -> F}{E, \b{F}}$ 

\liPseudoUeberschrift{A}

 $A \notin \ahr{B -> A}{B}{B} \setminus$ 
 $A \in \ahr{C -> A}{C}{\b{A}, B, C}$ 
```

### 2.44.5 TeX-Markup Relationen formen

```
\let\r=\liRelation
\let\u=\underline

\r[R1]{\u{A, D}, E}\
\r[R2]{\u{B, C}, A, E}\
\r[R3]{\u{D}, B}
```

#### 1. Kanonische Überdeckung

— Die kanonische Überdeckung - also die kleinst mögliche noch äquivalente Menge von funktionalen Abhängigkeiten kann in vier Schritten erreicht werden. —

##### (a) Linksreduktion

— Führe für jede funktionale Abhängigkeit  $\alpha \rightarrow \beta \in F$  die Linksreduktion durch, überprüfe also für alle  $A \in \alpha$ , ob  $A$  überflüssig ist, d. h. ob  $\beta \subseteq \text{AttrHülle}(F, \alpha - A)$ .

##### (b) Rechtsreduktion

— Führe für jede (verbliebene) funktionale Abhängigkeit  $\alpha \rightarrow \beta$  die Rechtsreduktion durch, überprüfe also für alle  $B \in \beta$ , ob  $B \in \text{AttrHülle}(F - (\alpha \rightarrow \beta) \cup (\alpha \rightarrow (\beta - B)), \alpha)$  gilt. In diesem Fall ist  $B$  auf der rechten Seite überflüssig und kann eliminiert werden, d. h.  $\alpha \rightarrow \beta$  wird durch  $\alpha \rightarrow (\beta - B)$  ersetzt. —

##### (c) Löschen leerer Klauseln

— Entferne die funktionalen Abhängigkeiten der Form  $\alpha \rightarrow \emptyset$ , die im 2. Schritt möglicherweise entstanden sind. —

##### (d) Vereinigung

— Fasse mittels der Vereinigungsregel funktionale Abhängigkeiten der Form  $\alpha \rightarrow \beta_1, \dots, \alpha \rightarrow \beta_n$ , so dass  $\alpha \rightarrow \beta_1 \cup \dots \cup \beta_n$  verbleibt. —

#### 2. Relationsschemata formen

— Erzeuge für jede funktionale Abhängigkeit  $\alpha \rightarrow \beta \in F_c$  ein Relationenschema  $\mathcal{R}_\alpha := \alpha \cup \beta$ . —

#### 3. Schlüssel hinzufügen

— Falls eines der in Schritt 2. erzeugten Schemata  $\mathcal{R}_\alpha$  einen Schlüsselkandidaten von  $\mathcal{R}$  bezüglich  $F_c$  enthält, sind wir fertig, sonst wähle einen Schlüsselkandidaten  $\mathcal{K} \subseteq \mathcal{R}$  aus und definiere folgendes zusätzliche Schema:  $\mathcal{R}_\mathcal{K} := \mathcal{K}$  und  $\mathcal{F}_\mathcal{K} := \emptyset$  —

#### 4. Entfernung überflüssiger Teilschemata

— Eliminiere diejenigen Schemata  $\mathcal{R}_\alpha$ , die in einem anderen Relationenschema  $\mathcal{R}_{\alpha'}$  enthalten sind, d. h.  $\mathcal{R}_\alpha \subseteq \mathcal{R}_{\alpha'}$ . —

\liSyntheseUeberschrift Let-Abkürzung: \let\schritt=\liSyntheseUeberschrift

```
2655 \def\liSyntheseUeberschrift#1{
2656   {
2657     \bfseries
2658     \sffamily
2659     \str_case:nn {#1} {
2660       {1} {Kanonische-Überdeckung}
2661       {1-1} {Linksreduktion}
2662       {1-2} {Rechtsreduktion}
2663       {1-3} {Löschen-leerer-Klauseln}
2664       {1-4} {Vereinigung}
2665       {2} {Relationsschemata-formen}
2666       {3} {Schlüssel-hinzufügen}
2667       {4} {Entfernung-überflüssiger-Teilschemata}
2668     }
```



```
2669 }
2670 }
```

**\liSyntheseErklaerung** **Let-Abkürzung:** \let\erklaerung=\liSyntheseErklaerung

```
2671 \def\li@synthese@erklaerung@texte#1{
2672   \str_case:nn {#1} {
2673     {1} {
2674       Die~kanonische~Überdeckung~---also~die~kleinst~mögliche~noch~
2675       äquivalente~Menge~von~funktionalen~Abhängigkeiten~kann~in~vier~
2676       Schritten~erreicht~werden.
2677     }
2678     {1-1} {
2679       Führe~für~jede~funktionale~Anhängigkeit~
2680        $\alpha \rightarrow \beta$ ~in~F~die~Linksreduktion~durch,~
2681       überprüfe~also~für~alle~
2682        $A \in \alpha$ ,~ob~ $A$ ~überflüssig~ist,~d.h.~ob~
2683        $\beta \subseteqq \text{liAttributHuelle}\{F, \alpha \cup A\}$ .
2684     }
2685     {1-2} {
2686       Führe~für~jede~(verbliebene)~funktionale~Abhängigkeit~ $\alpha \rightarrow \beta$ ~die~Rechtsreduktion~durch,~überprüfe~also~für~
2687       alle~ $B \in \beta$ ,~ob~ $B \in \text{liAttributHuelle}\{F \cup (\alpha \rightarrow \beta) \cup (\alpha \rightarrow B)\}$ ,~
2688        $\alpha$ ~gilt.~In~diesem~Fall~ist~B~auf~der~rechten~Seite~
2689       überflüssig~und~kann~eliminiert~werden,~d.h.~ $\alpha \rightarrow \beta$ ~wird~durch~ $\alpha \rightarrow (\beta \cup B)$ ~
2690       ersetzt.
2691     }
2692     {1-3} {
2693       Entferne~die~funktionalen~Abhängigkeiten~der~Form~ $\alpha \rightarrow \emptyset$ ,~die~im~2.~Schritt~möglicherweise~
2694       entstanden~sind.
2695     }
2696     {1-4} {
2697       Fasse~mittels~der~Vereinigungsregel~funktionale~Abhängigkeiten~
2698       der~Form~ $\alpha \rightarrow \beta_{\{1\}}, \dots, \alpha \rightarrow \beta_{\{n\}}$ ,~so~dass~ $\alpha \rightarrow \beta_{\{1\}} \cup \dots \cup \beta_{\{n\}}$ ~
2699       verbleibt.
2700     }
2701     % Kemper Seite 197
2702     {2} {
2703       Erzeuge~für~jede~funktionale~Abhängigkeit~ $\alpha \rightarrow \beta$ ~in~F~ein~Relationenschema~ $\mathcal{R}_{\alpha}$ ~
2704        $:= \alpha \cup \beta$ .
2705     }
2706     {3} {
2707       Falls~eines~der~in~Schritt~2.~erzeugten~Schemata~ $\mathcal{R}_{\alpha}$ ~
2708       einen~Schlüsselkandidaten~von~ $\mathcal{R}$ ~bezüglich~ $F$ ~
2709       enthält,~sind~wir~fertig,~sonst~wähle~einen~Schlüsselkandidaten~
2710        $\mathcal{K} \subseteqq \mathcal{R}$ ~aus~und~definiere~folgendes~
2711       zusätzliche~Schema:~ $\mathcal{R}_{\mathcal{K}} := \mathcal{K} \cup F$ ~
2712       und~ $\mathcal{F}_{\mathcal{K}} := \emptyset$ 
2713     }
2714     {4} {
2715       Eliminiere~diejenigen~Schemata~ $\mathcal{R}_{\alpha}$ ,~die~in~einem~
2716       anderen~Relationenschema~ $\mathcal{R}_{\alpha'}$ ~enthalten~sind,~d.h.~
2717        $\mathcal{R}_{\alpha} \subseteqq \mathcal{R}_{\alpha'}$ .
2718     }
2719   }
2720 }
2721 \def\liSyntheseErklaerung#1{
2722 {
2723   \itshape
2724   \footnotesize
2725 }
```

```

2731     \liParagraphMitLinien{\li@synthese@erklaerung@texte{#1}}
2732   }
2733 }

```

\liSyntheseUeberErklaerung **Let-Abkürzung:** \let\schrittE=\liSyntheseUeberErklaerung

```

2734 \def\liSyntheseUeberErklaerung#1{
2735   \liSyntheseUeberschrift{#1}\par
2736   \liSyntheseErklaerung{#1}
2737 }

```

```

2738 \ExplSyntaxOff
2739

```

## 2.45 tabelle.sty

2740 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2741 \ProvidesPackage{lehramt-informatik-tabelle}[2020/12/05 Lädt das Paket tabluarx]

2742 \RequirePackage{tabularx}

2743

## 2.46 typographie.sty

```
2744 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2745 \ProvidesPackage{lehramt-informatik-typographie}[2021/03/20 Typographische Makros,
2746 die das Erscheinungsbild verändern. Die Schriftdefinition sind in
2747 formatierung.sty definiert.]
```

```
2748 \ExplSyntaxOn
```

Mit dem Packet wasysym gab es Unverträglichkeiten, deshalb verwenden wir fontawesome.

```
2749 \RequirePackage{fontawesome}
```

`\liErledigt` `\liErledigt:` ☑

```
2750 \let\liErledigt=\faCheckSquareO
```

`\liNichtsZuTun` `\liNichtsZuTun:` ∅ Nichts zu tun

```
2751 \def\liNichtsZuTun{${\emptyset}$-Nichts-zu-tun}
```

`\liParagraphMitLinien` `\liParagraphMitLinien{Lorem ipsum...}:`

— Lorem ipsum dolor sit, ipsum dolor sit, ipsum dolor sit, ipsum dolor sit, ipsum dolor sit, ipsum dolor sit —

```
2752 \def\liParagraphMitLinien#1{
```

```
2753   \noindent
```

```
2754   \vrule height 2pt depth -1.6pt width 0.4cm
```

```
2755   \enspace
```

```
2756   #1
```

```
2757   \enspace
```

```
2758   \leaders\vrule height 2pt depth -1.6pt \hfill \null
```

```
2759   \par
```

```
2760   \medskip
```

```
2761 }
```

`\liGeschweifteKlammern` Große geschweifte Klammer mit Istgleich-Zeichen.

$\text{Variable} = \left\{ \begin{array}{c} \text{Inhalt} \end{array} \right\}$
---

```
2762 \def\liGeschweifteKlammern#1#2#3#4{
```

```
2763   \par
```

```
2764   \medskip
```

```
2765   \noindent
```

```
2766   #1 \, $= \Bigl\{
```

```
2767   \vspace{#3}
```

```
2768   #2
```

```
2769   \vspace{#4}
```

```
2770   \begin{flushright}$\Bigr\}$\end{flushright}
```

```
2771   \par
```

```
2772 }
```

```
2773 \ExplSyntaxOff
```

```
2774
```

## 2.47 uml.sty

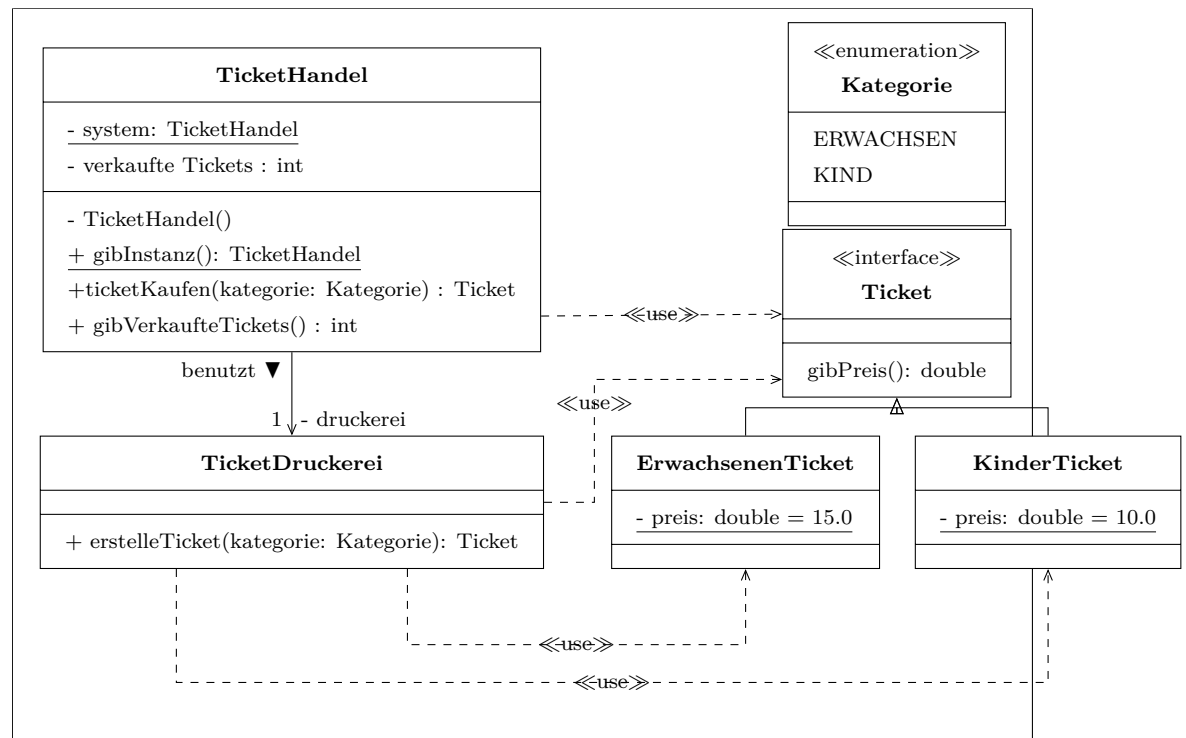
```

2775 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2776 \ProvidesPackage{lehramt-informatik-uml}[2020/06/13 Ein Hüll-Paket um
2777 `tikz-uml', das einige Design-Einstellungen vornimmt und manche
2778 Erweiterung bereitstellt]

2779 \RequirePackage{tikz-uml}
2780 \RequirePackage{tikz-uml-activity}
2781 % Not compatible with wasysym
2782 %\RequirePackage{mathabx}
2783 \RequirePackage{wasysym}
2784 \usetikzlibrary{positioning}

2785 \tikzumlset{
2786   fill class=white!0,
2787   font=\footnotesize,
2788   fill object=white!0,
2789   fill note=white!0,
2790   fill state=white!0,
2791   % Use case
2792   fill usecase=white!0,
2793   fill system=white!0,
2794 }

```



```

\liUmlLeserichtung \umluniassoc[arg1=,mult2=1,arg2=- druckerei,name=benutzt]{TicketHandel}{TicketDruckerei}
\liUmlLeserichtung[pos=below left,dir=down,distance=0cm]{benutzt}

```

```

2795 \NewDocumentCommand{ \liUmlLeserichtung } { 0{dir=right} m } {
2796   \def\@liDirLeft{}
2797   \def\@liDirRight{ \RIGHTarrow}
2798   \pgfkeys{/lese/dir/.is choice}
2799   \pgfkeys{/lese/dir/up/.code={\def\@liDirRight{ \UParrow}}}
2800   \pgfkeys{/lese/dir/down/.code={\def\@liDirRight{ \DOWNarrow}}}
2801   \pgfkeys{/lese/dir/left/.code={\def\@liDirRight{}\def\@liDirLeft{\LEFTarrow }}}
2802   \pgfkeys{/lese/dir/right/.code={\def\@liDirRight{ \RIGHTarrow}}}
2803
2804   \def\@liPos{above}
2805   \pgfkeys{/lese/pos/.code={\def\@liPos{##1}}}
2806

```

```

2807 \def\@liDistance{0cm}
2808 \pgfkeys{/lese/distance/.code={\def\@liDistance{##1}}}}
2809
2810 \pgfkeys{/lese/.cd,#1}
2811
2812 \node[\@liPos = \@liDistance of #2-middle] {
2813   \@liDirLeft{\footnotesize#2}\@liDirRight
2814 };
2815 }
2816

```

## 2.48 vollstaendige-induktion.sty

2817 \NeedsTeXFormat{LaTeX2e}[1995/12/01]  
 2818 \ProvidesPackage{lehramt-informatik-vollstaendige-induktion}[2021/07/01  
 2819 Hilfsmakros zum Setzen der Vollständigen Induktion, vor allem die  
 2820 Überschriften für die einzelnen Schritte]

### 2.48.1 Makro-Kürzel

\let\m=\liInduktionMarkierung  
 \let\e=\liInduktionErklaerung

```
\begin{align*}
C_{n+1}
&= \frac{
  \{(4 \cdot (\{m\{n + 1\} - 1) + 2) \cdot \text{cn}(\{m\{n + 1\} - 1)\}
  \{m\{n + 1\} + 1\}
}{
  \{e\{Java nach Mathe}\} \\
\%
&= \frac{
  \{(4\{m\{n\} + 2) \cdot \text{cn}(\{m\{n\}\}
  \{m\{n + 2\}\}
}{
  \{e\{addiert, subtrahiert}\} \\
\%
&= \frac{
  \{(4n + 2) \cdot m\{(2n)!\}
  \{(n + 2) \cdot m\{(n + 1)! \cdot n!\}
}{
  \{e\{für cn(n) Formel eingesetzt}\} \\
\%
&= \frac{
  \{(4n + 2) \cdot (2n)! \cdot m\{\cdot (n + 1)\}\}
  \{(n + 2) \cdot (n + 1)! \cdot n! \cdot m\{\cdot (n + 1)\}\}
}{
  \{e\{(n + 1)\$ multipliziert}\} \\
\%
&= \frac{
  \{(4n + 2) \cdot m\{(n + 1) \cdot (2n)!\}
  \{(n + 2) \cdot (n + 1)! \cdot m\{(n + 1) \cdot n!\}
}{
  \{e\{umsortiert}\} \\
\%
&= \frac{
  \{m\{(2(n + 1))!\}
  \{m\{(n + 2)! \cdot (n + 1)!\}
}{
  \{e\{Hilfsgleichungen verwendet}\} \\
\%
&= \frac{
  \{(2(\{m\{n + 1\}\}!\}
  \{((\{m\{n + 1\}\} + 1)! \cdot (\{m\{n + 1\}\}!\}
}{
  \{e\{(n + 1)\$ verdeutlicht}\} \\
\end{align*}
```

Lade häufig benötigte Pakete

2821 \RequirePackage{lehramt-informatik-typographie}  
 2822 \RequirePackage{lehramt-informatik-mathe}  
 2823 \RequirePackage{lehramt-informatik-syntax}  
 2824 \ExplSyntaxOn

\liInduktionMarkierung Hilfsmakro um Teile von mathematischen Formeln markieren zu können.

**Let-Abkürzung:** \let\m=\liInduktionMarkierung

2825 \def\liInduktionMarkierung#1{\textcolor{violet}{#1}}

\liInduktionErklaerung Gedacht für die rechte Spalte in der align-Umgebung. Das text-Makro ist dann nicht mehr nötig.

**Let-Abkürzung:** \let\e=\liInduktionErklaerung

```
2826 \def\liInduktionErklaerung#1{\scriptsize\text{#1}}
```

\liInduktionAnfang

```
2827 \def\liInduktionAnfang{
2828   \liPseudoUeberschrift{Induktionsanfang}
2829
2830   % https://de.wikibooks.org/wiki/Mathe_für_Nicht-Freaks:_Vollständige_Induktion
2831   \liParagraphMitLinien{
2832     Beweise,~dass~ $A(1)$ ~eine~wahre~Aussage~ist.
2833   }
2834 }
```

\liInduktionVoraussetzung

```
2835 \def\liInduktionVoraussetzung{
2836   \liPseudoUeberschrift{Induktionsvoraussetzung}
2837
2838   % https://de.wikibooks.org/wiki/Mathe_für_Nicht-Freaks:_Vollständige_Induktion
2839   \liParagraphMitLinien{
2840     Die~Aussage~ $A(k)$ ~ist~wahr~für~ein~beliebiges~ $k \in \mathbb{N}$ $.
2841   }
2842 }
```

\liInduktionSchritt

```
2843 \def\liInduktionSchritt{
2844   \liPseudoUeberschrift{Induktionsschritt}
2845
2846   % https://de.wikibooks.org/wiki/Mathe_für_Nicht-Freaks:_Vollständige_Induktion
2847   \liParagraphMitLinien{
2848     Beweise,~dass~wenn~ $A(n=k)$ ~wahr~ist,~
2849     auch~ $A(n=k+1)$ ~wahr~sein~muss.
2850   }
2851 }
```

```
2852 \ExplSyntaxOff
```

```
2853
```



## 2.49 wasserfall.sty

```
2854 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2855 \ProvidesPackage{lehramt-informatik-wasserfall}[2020/06/10]
2856 \RequirePackage{tikz}
2857 \tikzset{wasserfall/.style={
2858   >=stealth,
2859   node distance = 2mm and -8mm,
2860   start chain = A going below right,
2861   every node/.style = {
2862     draw,
2863     text width=24mm,
2864     minimum height=12mm,
2865     align=center,
2866     inner sep=1mm,
2867     fill=white,
2868     drop shadow={fill=black},
2869     on chain=A
2870   },
2871 }}
2872 \usetikzlibrary{chains,positioning,shadows}
2873
```

## 2.50 wpkalkuel.sty

2874 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2875 \ProvidesPackage{lehramt-informatik-wpkalkuel}[2020/06/13]

### 2.50.1 Makro-Kürzel

\let\wp=\liWpKalkuel

\let\equivalent=\liWpEquivalent

\let\erklaerung=\liWpErklaerung

2876 \RequirePackage{amsmath}

2877 \ExplSyntaxOn

\liWpKalkuel **Let-Abkürzung:** \let\wp=\liWpKalkuel

2878 \def\liWpKalkuelOhneMathe#1#2{

2879 \text{wp}(\texttt{\scriptsize"#1"},\thinspace #2)

2880 }

2881 \def\liWpKalkuel#1#2{

2882 \ifmmode

2883 \liWpKalkuelOhneMathe{#1}{#2}

2884 \else

2885 \$\liWpKalkuelOhneMathe{#1}{#2}\$

2886 \fi

2887 }

\MatheEnv

2888 \def\MatheEnv#1{

2889 \medskip

2890

2891 \hspace{1em}#1

2892

2893 \medskip

2894 }

\Mathe

2895 \def\Mathe#1{

2896 \MatheEnv{\${#1}\$}

2897 }

\liWpEquivalent **Let-Abkürzung:** \let\equivalent=\liWpEquivalent

2898 \def\liWpEquivalent#1{

2899 \MatheEnv{\${\equiv}\hspace{1em}\${#1}\$}

2900 }

\liWpErklaerung **Let-Abkürzung:** \let\erklaerung=\liWpErklaerung

2901 \newlength{\@Skip@Erklaerung@Reset}

2902 \def\liWpErklaerung#1{

2903 \setlength{\@Skip@Erklaerung@Reset}{\leftskip}

2904 \setlength{\leftskip}{0.5cm}

2905

2906 \par

2907 \noindent

2908 {

2909 \scriptsize

2910 #1

2911 }

2912 \par

2913

2914 \setlength{\leftskip}{\@Skip@Erklaerung@Reset}

2915 }

\liWpErklaerungVerzweigung

```

2916 \def\liWpErklaerungVerzweigung{
2917   $\liWpKalkuelOhneMathe{if~\{-b~\}~then~\{-a1~\}~else~\{-a2~\}}{Q}
2918   \equiv
2919   (b \land \liWpKalkuelOhneMathe{a1}{Q})
2920   \lor
2921   (\neg b \land \liWpKalkuelOhneMathe{a2}{Q})$
2922 }

2923 \ExplSyntaxOff

2924

```

### 3 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols		
\# ..... 121	\AddToHook ..... 1374	\bfseries .. 500, 1248,
\, .... 357, 412, 1195,	\advance ..... 2497	1250, 2270, 2276,
1197, 1766, 1767,	\AfterEndEnvironment 2573	2278, 2280, 2281, 2657
1768, 2103, 2580, 2766	\allsectionsfont ... 1244	\Bigl ..... 2766
\@Skip@Erklaerung@Reset	\Alph ..... 1258	\Bigr ..... 2770
... 2901, 2903, 2914	\alph ..... 1258, 1259	\bigskip ..... 59, 390,
\@afterheading ..... 1733	\alpha 2680, 2682, 2683,	623, 628, 1642, 1971
\@afterindentfalse . 1733	2686, 2688, 2689,	\bool ..... 335, 358
\@liDirLeft 2796, 2801, 2813	2690, 2691, 2692,	\bowtie ..... 2418, 2421, 2422, 2423
\@liDirRight 2797, 2799,	2696, 2702, 2703,	\Box ..... 171
2800, 2801, 2802, 2813	2708, 2709, 2710,	\boxtimes ..... 488
\@liDistance ..... 2713, 2721, 2722, 2723	\arabic 1258, 2292, 2297,	
... 2807, 2808, 2812	2302, 2308, 2314, 2320	C
\@liPos .. 2804, 2805, 2812	\arraystretch ..... 1993	\c ..... 1341, 1342
\\ ..... 621, 644,		\cdot .... 1810, 1864, 1875
645, 648, 649, 652,	B	\centerline ..... 1439, 2371, 2393, 2408
653, 745, 746, 747,	\BeforeBeginEnvironment	\chapter ..... 1248, 1249
854, 883, 885, 911,	..... 2572	\char ..... 1640
920, 965, 1007,	\begin 642, 693, 708, 743,	\clearpage ..... 1634
1008, 1009, 1014,	767, 814, 846, 861,	\cline ..... 621
1015, 1016, 1036,	881, 891, 907, 927,	\clist ..... 238, 282,
1640, 1998, 2056, 2059	959, 976, 1005,	283, 303, 307, 2505
\{ ..... 219, 1132,	1030, 1051, 1066,	\columnbreak ..... 2540
1142, 1154, 1155,	1175, 1264, 1339,	\cs .... 306, 332, 356,
1160, 1194, 1480,	1434, 1443, 1450,	357, 394, 406, 1716
2043, 2448, 2766, 2917	1558, 1632, 1684,	\csname ..... 1326, 1329
\} ..... 219, 1132,	1689, 1697, 1722,	\cup ..... 1155,
1142, 1154, 1155,	1727, 1738, 1748,	2077, 2689, 2703, 2710
1162, 1198, 1481,	1752, 1835, 1839,	
2043, 2448, 2770, 2917	1856, 1879, 1902,	D
\_ ..... 38, 46, 55, 57,	1917, 1996, 1997,	\DeclareMathSymbol ..
306, 332, 356, 357,	2047, 2092, 2212,	..... 1941, 1942
371, 372, 378, 381,	2232, 2373, 2395,	\DecoINERT ..... 2295, 2368, 2369, 2391
384, 394, 406, 1375	2409, 2572, 2583, 2770	\DecoINERTwithPivot .
	\begingroup 1633, 2045, 2493	..... 2311, 2390
\_ ..... 2055, 2065	\beschriftung ..... 1664, 1668,	\DecoLEFT ..... 2290, 2389
	1672, 1676, 1680, 1682	\DecoLEFTwithPivot ..
A	\beta ..... 2680,	..... 2305, 2367
\addbibresource ..... 2683, 2687, 2688,	2689, 2692, 2702,	\DecoRIGHT 2300, 2392, 2402
1610, 1611, 1612,	2703, 2704, 2709, 2710	\DecoRIGHTwithPivot .
1613, 1614, 1615,	\bf ..... 2270, 2271, 2272	..... 2317, 2370
1616, 1617, 1618, 1619		



<code>\IfNoValueTF</code> .....	1217, 1220, 1221,	<code>\li@GithubLink</code> .....
... 1690, 1760, 1764	1222, 1223, 1229,	2582, 2595, 2605, 2616
<code>\ifnum</code> .....	1230, 1231, 1232,	<code>\li@mgget</code> . 1328, 1332, 1352
2495, 2511, 2519, 2525	1574, 1575, 1576,	<code>\li@minc</code> .....
<code>\ifx</code> .... 1680, 2075, 2523	1719, 1720, 1721, 1728	1331, 1353
<code>\in</code> .....	<code>\labelenumi</code> .....	<code>\li@msset</code> .....
519, 625,	1259	1325, 1333, 1342, 1345
1487, 1825, 1828,	<code>\labelenumii</code> .....	<code>\li@numdiscs</code> .....
1831, 1858, 1864,	1260	... 1337, 1346, 1352
1869, 2208, 2219,	<code>\labelitemi</code> .....	<code>\li@Rmodell@Schrift</code> .
2229, 2239, 2680,	1253	... 2432, 2441, 2451
2682, 2688, 2709, 2840	<code>\labelitemii</code> .....	<code>\li@sequence</code> .. 1338, 1351
<code>\inhaltsverzeichnis</code> <u>1631</u>	<code>\labelitemiii</code> .....	<code>\li@synthese@erklaerung@texte</code>
<code>\input</code> .....	<code>\labelitemiv</code> .....	2671, 2731
17, 20,	<code>\land</code> .....	<code>\li@Ableitung</code> .....
23, 26, 29, 419, 1408	<code>\LARGE</code> .....	<u>1169</u>
<code>\inputminted</code> 2590, 2600,	<code>\large</code> .....	<code>liAdditum</code> (environment)
2610, 2622, 2625, 2629	<code>\leaders</code> .....	1687
<code>\int</code> .....	<code>\left</code> .....	<code>liAHuelle</code> (environment)
2505	<code>\LEFTarrow</code> .....	2044
<code>\item</code> .....	<code>\leftarrow</code> .....	<code>\li@Alphabet</code> .....
488,	<code>\leftarrow</code> .....	<u>1154</u>
489, 710, 714, 719,	<code>\leftouterjoin</code> ....	<code>liAntwort</code> (environment)
724, 768, 777, 782,	2421	1658
790, 862, 867, 871,	<code>\leftskip</code> 2903, 2904, 2914	<code>\li@Anweisung</code> .....
892, 928, 933, 940,	<code>\LehramtInformatikAutorEmail</code>	<u>1565</u>
948, 977, 982, 986,	..... 1590	<code>\li@AssemblerCode</code> ...
991, 1067, 1072,	<code>\LehramtInformatikAutorName</code>	<u>2620</u>
1077, 1444, 1445,	..... 1589	<code>\li@AssemblerDatei</code> ..
1716, 1720, 1840,	<code>\LehramtInformatikGitBranch</code>	<u>2621</u>
1843, 1847, 1857,	..... 399, 2566	<code>\li@Attribut</code> .....
1863, 1868, 1880,	<code>\LehramtInformatikGithubCodeRepo</code>	<u>2451</u>
1884, 1888, 1892,	..... 2565	<code>\li@AttributHuelle</code> ...
1896, 1903, 1907,	<code>\LehramtInformatikGithubDomain</code>	... 2035, 2683, 2688
1911, 2213, 2216,	..... 2562	<code>\li@AttributMenge</code> ....
2219, 2233, 2236, 2239	<code>\LehramtInformatikGithubRawDomain</code>	2043, 2055, 2058,
<code>\itshape</code> .....	397, 2563	2065, 2066, 2080, 2082
544, 2729	<code>\LehramtInformatikGithubTexRepo</code>	<code>\li@Aufgabe</code> .....
	..... 398, 2564	<u>16</u>
<b>J</b>	<code>\LehramtInformatikRepository</code>	<code>\li@AufgabenMetadaten</code> .
<code>\j</code> 1341, 1342, 1344, 1345,	..... 17, 20, 23,	<u>36</u>
1346, 1351, 1352, 1353	26, 29, 1409, 1610,	<code>\li@Ausdruck</code> .....
	1611, 1612, 1613,	<u>1191</u>
<b>K</b>	1614, 1615, 1616,	<code>\li@Automat</code> .....
<code>\k</code> .....	1617, 1618, 1619, 2561	<u>73</u>
1351	<code>\LehramtInformatikTitel</code>	<code>\li@AutomatenKante</code> ...
<code>\keys</code> .....	..... 1585	<u>105</u>
42, 82,	<code>\leq</code> .... 1875, 2216, 2236	<code>\li@BandAlphabet</code> ....
94, 124, 134, 182,	<code>\let</code> .....	<u>1155</u>
192, 311, 563, 567,	1144,	<code>\li@Bedingung</code> .....
581, 586, 1219, 1226	1145, 1634, 2367,	<u>1566</u>
	2368, 2369, 2370,	<code>\li@BedingungDrei</code> ....
<b>L</b>	2383, 2385, 2386,	... 1830, 1871, 1911
<code>\l</code> .. 75, 76, 77, 78, 79,	2387, 2389, 2390,	<code>\li@BedingungEins</code> ....
80, 83, 84, 85, 86,	2391, 2392, 2402,	... 1824, 1860, 1903
87, 89, 91, 96, 97,	2494, 2532, 2533, 2750	<code>\li@BedingungFalsch</code> .
98, 99, 100, 101,	<code>\li@chomsky@erklaerung@texte</code>	<u>1568</u>
116, 117, 118, 119,	..... 510, 546	<code>\li@BedingungWahr</code> ...
120, 121, 122, 125,	<code>\li@EntwurfsCode</code> ....	<u>1567</u>
126, 127, 128, 129,	638, 684, 685, 686,	<code>\li@BedingungZwei</code> ....
130, 131, 137, 138,	732, 733, 734, 735,	... 1827, 1866, 1907
139, 140, 141, 142,	801, 802, 803, 804,	<code>\li@Beschriftung</code> ....
143, 174, 175, 176,	805, 806, 833, 834,	<u>1647</u>
177, 178, 179, 180,	835, 836, 837, 838, 899	<code>\li@BindeAufgabeEin</code> ..
183, 184, 185, 186,	<code>\li@EntwurfsCodeAllgemein</code>	<u>1406</u>
187, 188, 189, 195,	..... 637	<code>\li@ChomskyErklaerung</code>
196, 197, 198, 199,	<code>\li@fussnote@text</code> 1949,	..... 510, 551
200, 201, 561, 564,	1955, 1959, 1963, 1967	<code>\li@ChomskyUeberErklaerung</code>
569, 570, 573, 578,		..... 549
579, 582, 583, 588,		<code>\li@ChomskyUeberschrift</code>
1214, 1215, 1216,		..... 498, 550
		<code>\li@CpmEreignis</code> .....
		<u>559</u>
		<code>\li@CpmFruehesterI</code> ...
		<u>616</u>
		<code>\li@CpmSpaetesterI</code> ...
		<u>615</u>
		<code>\li@CpmVon</code> .....
		<u>599</u>
		<code>\li@CpmVonOhneMathe</code> ..
		..... 599, 602, 604

\liCpmVonZu .....	591	\liEntwurfsFabrikmethodeUml .....	958, 1001	\liInduktionErklaerung .....	2826
\liCpmVonZuOhneMathe .....	591, 594, 596	\liEntwurfsKompositum .....	1025	\liInduktionMarkierung .....	2825
\liCpmVorgang .....	576	\liEntwurfsKompositumAkteure .....	1027	\liInduktionSchritt .....	2843
\liCpmZu .....	607	\liEntwurfsKompositumUml .....	1004, 1026	\liInduktionVoraussetzung .....	2835
\liCpmZuOhneMathe .....	607, 610, 612	\liEntwurfsModellPraesentation .....	1046	\liJavaCode .....	2580
liDiagramm (environment) .....	1746	\liEntwurfsModellPraesentationAkteure .....	1048	\liJavaDatei .....	639, 2589
liEinbettung (environment) .....	1657	\liEntwurfsModellPraesentationSteuerung .....	1029, 1047	\liJavaExamen .....	2609
\liEntwurfs .....	1046	\liEntwurfsModellPraesentationSteuerungUml .....	1083	\liJavaResourceter .....	2599
\liEntwurfsAbstrakteFabrik .....	688	\liEntwurfsZustand .....	1065, 1085	liKasten (environment) .....	1263
\liEntwurfsAbstrakteFabrikCode .....	683, 690	\liEntwurfsZustandAkteure .....	1050, 1084	liKellerAutomat .....	114
\liEntwurfsAbstrakteFabrikUml .....	641, 689	\liEntwurfsZustandUml .....	1140	liKellerKante .....	152
\liEntwurfsAdapter .....	737	\liEpsilon .....	1140	liKellerUebergang .....	147, 153
\liEntwurfsAdapterAkteure .....	707, 739	\liErAttribute .....	1098, 1112, 1114	\liKontrollCode .....	1569
\liEntwurfsAdapterCode .....	731, 740	\liErDatenbankName .....	1117	liKontrollflussgraph (environment) .....	1557
\liEntwurfsAdapterUml .....	692, 738	\liErEntity .....	1096, 1100, 1102	\liKontrollKnotenPfad .....	1571
\liEntwurfsBeobachter .....	808	\liErledigt .....	2750	\liKontrollTextzeileKnoten .....	1570, 1575
\liEntwurfsBeobachterAkteure .....	766, 810	\liErMpAttribute .....	1111	\liKurzeTabellenLinie .....	621
\liEntwurfsBeobachterCode .....	800, 811	\liErMpEntity .....	1099	\liLadeAllePakete .....	240, 1389
\liEntwurfsBeobachterUml .....	742, 809	\liErMpRelationship .....	1105	\liLadePakete .....	66, 69, 236, 241, 497, 558, 1131, 1361, 1382, 1429, 1947, 2027, 2175, 2544, 2653
\liEntwurfsDekorierer .....	840	\liErRelationship .....	1097, 1106, 1108	\liLatexCode .....	2581
\liEntwurfsDekoriererAkteure .....	842	\liExamensAufgabe .....	19	\liLeereZelle .....	1985
\liEntwurfsDekoriererCode .....	832, 843	\liExamensAufgabeA .....	28	liLernkartei (environment) .....	1736
\liEntwurfsDekoriererUml .....	813, 841	\liExamensAufgabeTA .....	25	\liLinksReduktion .....	2052
\liEntwurfsEinfacheFabrik .....	876	\liExamensAufgabeTTA .....	22	\liLinksReduktionInline .....	2061, 2069
\liEntwurfsEinfacheFabrikAkteure .....	860, 878	liExkurs (environment) .....	1695	\liMasterExkurs .....	1916
\liEntwurfsEinfacheFabrikUml .....	845, 877	\liFalsch .....	489	\liMasterFaelle .....	1855, 1923
\liEntwurfsEinzelstueck .....	901	\liFlaci .....	1201	\liMasterFallRechnung .....	1901
\liEntwurfsEinzelstueckAkteure .....	890, 903	\liFremd .....	2431	\liMasterVariablen .....	1834, 1918
\liEntwurfsEinzelstueckCode .....	898, 904	\liFunktionaleAbhaengigkeit .....	2074, 2077, 2085	\liMasterVariablenDeklaration .....	1878
\liEntwurfsEinzelstueckUml .....	880, 902	\liFunktionaleAbhaengigkeiten .....	2088	\liMasterWolframLink .....	1926
\liEntwurfsErbauer .....	954	\liFussnote .....	1948, 1950	\liMenge .....	83, 84, 86, 125, 126, 127, 131, 183, 184, 185, 189, 1132, 1181, 1220, 1221
\liEntwurfsErbauerAkteure .....	926, 956	\liFussnoteDreiText .....	1962, 1980	\liMengeOhneMathe .....	1132, 1135, 1137
\liEntwurfsErbauerUml .....	906, 955	\liFussnoteEinsText .....	1954, 1974	\liMinimierungErklaerung .....	2007
\liEntwurfsFabrikmethode .....	1000, 1025	\liFussnoteLink .....	1763	\liMinispracheDatei .....	2624
\liEntwurfsFabrikmethodeAkteure .....	975, 1002	\liFussnoten .....	1970	\liLinespread .....	2439
		\liFussnoteUrl .....	1044, 1759	\liNichtsZuTun .....	2751
		\liFussnoteVierText .....	1966, 1983	liO .....	1797, 1825
		\liFussnoteZweiText .....	1958, 1977	liOmega .....	1787, 1831
		\liGeschweifteKlammern .....	1173, 2089, 2762	liOmegaOhneMathe .....	1787, 1792, 1794
		\liGrammatik .....	1212		
		liGraphenFormat (environment) .....	1318		
		\liHanoi .....	1325		
		\liHaskellCode .....	2627		
		\liHaskellDatei .....	2628		
		\liInduktionAnfang .....	2827		

<code>\li0OhneMathe</code> . . . . .	<code>\liSetzeExamenTeilaufgabeNr</code> . . . . .	<code>\liZustandsmengeNr</code> . . . . .
. . . 1797, 1802, 1804	. . . . . 1402	. . . . . 1158, 2190
<code>\liParagraphMitLinien</code> . . . . .	<code>\liSetzeExamenThemaNr</code> . . . . .	<code>\liZustandsmengeNrGross</code> . . . . .
. . . 546, 2009, 2731,	. . . . . 1398	. . . . . 1166
2752, 2831, 2839, 2847	<code>\liSortierMarkierung</code> 2471	<code>\liZustandsMengenSammlung</code> . . . . .
<code>\liPetriErreichKnotenDrei</code> . . . . .	<code>\liSortierPfeil</code> . . . . .	. . . . . 2177
. . . . . 2169	<code>\liSortierPfeilUnten</code> 2468	<code>\liZustandsMengenSammlungNr</code> . . . . .
<code>\liPetriErreichTransition</code> . . . . .	<code>\liSpaltenUmbruch</code> . . . . .	. . . . . 2186
. . . . . 2166	<code>\liSqlCode</code> . . . . .	<code>\liZustandsmengeOhneMathe</code> . . . . .
<code>\liPetriSetzeSchluessel</code> . . . . .	<code>\listen@punkt</code> . . . . .	. . . . . 1144
. . . . . 2111	<code>\liStrich</code> . . . . .	<code>\liZustandsname</code> . . . . .
<code>\liPetriTransitionsName</code> . . . . .	<code>\liSyntheseErklaerung</code> . . . . .	<code>\liZustandsnameGross</code> . . . . .
. . . . . 2158, 2170	. . . . . 2671, 2736	. . . . . 1168, 2178, 2187
<code>\liPetriTransitionsNameOhneMathe</code> . . . . .	<code>\liSyntheseUeberErklaerung</code> . . . . .	<code>\liZustandsnameTiefgestellt</code> . . . . .
. . . . . 2158, 2161, 2163	. . . . . 2734	. . . . . 1183
<code>\liPetriTransPfeile</code> 2170	<code>\liSyntheseUeberschrift</code> . . . . .	<code>\liZustandsPaar</code> . . . . .
<code>\liPolynomiellReduzierbar</code> . . . . .	. . . . . 2655, 2735	<code>\liZustandsPaarVariablenName</code> . . . . .
. . . . . 1449	<code>\liT</code> 1807, 1822, 1836, 1898	. . . . . 1986, 1989, 1990
<code>\liPotenzmenge</code> . . . . .	<code>\liTeilen</code> . . . . .	<code>\llap</code> . . . . .
. . . . . 1141, 1145, 2181	. . . . . 2032	. . . . . 2419
<code>\liPotenzmengeOhneMathe</code> . . . . .	<code>\literatur</code> . . . . .	<code>\log</code> . . . . .
. . . . . 1142, 1143, 1144	. . . . . 1598, 1622	. . . . . 1825,
<code>\liPrimaer</code> . . . . .	<code>\liTheta</code> . . . . .	1828, 1831, 1858, 1864
<code>\liProblemBeschreibung</code> . . . . .	. . . . . 1777,	<code>\loop</code> . . . . .
. . . . . 1433	1828, 1858, 1864, 1869	. . . . . 2252
<code>\liProblemClique</code> . . . . .	<code>\liThetaOhneMathe</code> . . . . .	<code>\lor</code> . . . . .
. . . . . 1456	. . . . . 1777, 1782, 1784	. . . . . 2920
<code>\liProblemName</code> . . . . .	<code>\liTOhneMathe</code> . . . . .	<code>\ltimes</code> . . . . .
. . . . . 1432, 1439,	. . . . . 1807, 1816, 1818	. . . . . 1934
1451, 1453, 1466,	<code>\liTuringKante</code> . . . . .	
1477, 1478, 1486, 1487	. . . . . 215	
<code>\liProblemSat</code> . . . . .	<code>\liTuringLeerzeichen</code> . . . . .	
<code>\liProblemSubsetSum</code> . . . . .	. . . . . 171, 179	
. . . . . 1476, 1485	<code>\liTuringMaschine</code> . . . . .	
<code>\liProblemVertexCover</code> . . . . .	. . . . . 172	
. . . . . 1456, 1464	<code>\liTuringUeberfuehrung</code> . . . . .	
<code>\liProduktionen</code> 1180, 1222	. . . . . 218	
<code>liProduktionsRegeln</code> (environment) . . . . .	<code>\liTuringUebergaenge</code> . . . . .	
. . . . . 1170	. . . . . 210, 216	
<code>liProjektSprache</code> (environment) . . . . .	<code>\liTuringUebergangZelle</code> . . . . .	
. . . . . 1656	. . . . . 205	
<code>\liPseudoUeberschrift</code> . . . . .	<code>\liUeberfuehrungsFunktion</code> . . . . .	
. . . . . 1641,	. . . . . 1146	
1691, 1692, 1995,	<code>\liUeberfuehrungsFunktionOhneMathe</code> . . . . .	
2005, 2828, 2836, 2844	. . . . . 1146, 1149, 1151	
<code>\liPumpingKontextfrei</code> . . . . .	<code>liUebergangsTabelle</code> (environment) . . . . .	
. . . . . 2227	. . . . . 1993	
<code>\liPumpingRegulaer</code> . . . . .	<code>\liUeberschriftDreiecksTabelle</code> . . . . .	
<code>liQuellen</code> (environment) . . . . .	. . . . . 2003	
. . . . . 1716	<code>\liUmlLeserichtung</code> . . . . .	
<code>\liRechtsReduktionInline</code> . . . . .	. . . . . 2795	
. . . . . 2069	<code>\liVertauschen</code> . . . . .	
<code>\liRekursionsGleichung</code> . . . . .	. . . . . 2459	
. . . . . 1821, 1882	<code>\liWortInSprache</code> . . . . .	
<code>\liRelation</code> . . . . .	. . . . . 622	
. . . . . 2099	<code>\liWortNichtInSprache</code> 627	
<code>liRelationenSchemaFormat</code> (environment) . . . . .	<code>\liWpEquivalent</code> . . . . .	
. . . . . 2452	. . . . . 2898	
<code>\liRelationMenge</code> . . . . .	<code>\liWpErklaerung</code> . . . . .	
. . . . . 2446	. . . . . 2901	
<code>\liRichtig</code> . . . . .	<code>\liWpErklaerungVerzweigung</code> . . . . .	
. . . . . 488	. . . . . 2916	
<code>liRmodell</code> (environment) . . . . .	<code>\liWpKalkuel</code> . . . . .	
. . . . . 2432	. . . . . 2878	
<code>\liRundeKlammer</code> . . . . .	<code>\liWpKalkuelOhneMathe</code> . . . . .	
. . . . . 1774,	. . . . . 2878, 2883,	
1778, 1788, 1798, 1812	2885, 2917, 2919, 2921	
<code>\liSetzeExamen</code> . . . . .	<code>\liZustandsBuchstabe</code> . . . . .	
. . . . . 1393	. . . . . 1156,	
	1165, 1167, 1185, 1187	
	<code>\liZustandsBuchstabeGross</code> . . . . .	
	. . . . . 1157, 1166, 1168	
	<code>\liZustandsmenge</code> . . . . .	
	. . . . . 1144	

## M

<code>\makeatletter</code> . . . . .	1732, 2491
<code>\makeatother</code> . . . . .	1734, 2499
<code>\marginpar</code> . . . . .	1101, 1107, 1113, 1639
<code>\mathbb</code> . . . . .	1487, 2239, 2840
<code>\mathbin</code> . . . . .	2421, 2422, 2423
<code>\mathcal</code> . . . . .	1798, 2709,
. . . . .	2714, 2716, 2717, 2718
<code>\Mathe</code> . . . . .	2895
<code>\MatheEnv</code> . . . . .	2888, 2896, 2899
<code>\mathord</code> . . . . .	1941, 1942
<code>\mdfsetup</code> . . . . .	1262,
. . . . .	1665, 1669, 1673, 1677
<code>\medskip</code> . . . . .	1441,
. . . . .	1650, 1652, 1726,
. . . . .	1754, 2437, 2443,
. . . . .	2760, 2764, 2889, 2893
<code>\emph</code> . . . . .	1639
<code>\mintinline</code> . . . . .	2580,
. . . . .	2581, 2620, 2627, 2632
<code>\mkern</code> . . . . .	2421, 2422, 2423
<code>\mlq</code> . . . . .	1939, 1941
<code>\mrq</code> . . . . .	1939, 1942
<code>\msg</code> . . . . .	50, 417
<code>\myList</code> . . . . .	2512, 2513, 2514, 2517
<code>\myNodes</code> . . . . .	2501,
. . . . .	2516, 2522, 2526, 2528

## N

<code>\NeedsTeXFormat</code> . . . . .	1, 14,
. . . . .	32, 64, 232, 423,
. . . . .	484, 491, 555, 618,
. . . . .	633, 1088, 1124,
. . . . .	1237, 1277, 1286,
. . . . .	1291, 1320, 1358,
. . . . .	1379, 1425, 1497,
. . . . .	1580, 1596, 1602,



1624, 1770, 1931, 1944, 2023, 2106, 2172, 2197, 2202, 2246, 2412, 2425, 2454, 2535, 2542, 2549, 2554, 2634, 2649, 2740, 2744, 2775, 2817, 2854, 2874	\neg ..... 2921	\pgfkeys .. 2122, 2798, 2799, 2800, 2801, 2802, 2805, 2808, 2810	\QSpivotStep ..... 2253, 2363, 2367, 2378
\negthinspace ..... 1775	\NewDocumentCommand . 74, 115, 152, 173, 215, 236, 560, 577, 622, 627, 1191, 1213, 1449, 1572, 1600, 1759, 1763, 2088, 2099, 2166, 2589, 2599, 2609, 2621, 2624, 2628, 2795	\pgfmath@count ..... ... 2493, 2495, 2497	\QSr ..... 2340
\NewDocumentEnvironment 1170, 1263, 1318, 1557, 1656, 1657, 1660, 1687, 1695, 1717, 1736, 1746, 1994, 2044, 2434, 2452	\newlength ..... 2901	\pgfmath@smuggleone 2498	\QSRr ..... 2349, 2370, 2381, 2386, 2392, 2401, 2402, 2403
\newlength ..... 2901	\node ..... 573, 1565, 1570, 2292, 2297, 2302, 2308, 2314, 2320, 2471, 2516, 2812	\pgfmathdeclarefunction ..... 2492	\QSSortStep ..... 2255, 2363, 2379, 2380
\noexpand ..... 2380, 2381, 2382, 2401, 2516	\noindent ..... 369, 624, 629, 1643, 1645, 1649, 1653, 1681, 1709, 1711, 1724, 1740, 1742, 1750, 1920, 1973, 1976, 1979, 1982, 2447, 2753, 2765, 2907	\pgfmathhint ..... 2493	\quad ..... 1951
\nolinkurl .... 402, 2586	\normalize ..... 1250	\pgfmathparse ..... ..... 1332, 2510, 2515, 2518, 2532, 2533	R
\notin ..... 630	\null ..... 2758	\pgfmathresult ..... ..... 1333, 2493, 2494, 2496, 2498, 2511, 2519, 2532, 2533	\raisebox ..... 1570
O	\o@join ..... 2418, 2421, 2422, 2423	\pgfutil@empty ..... 2494	\relax ..... 1634, 2340, 2383, 2385, 2386, 2387, 2495, 2497
\Omega ..... 1788	\omega 2208, 2209, 2229, 2230	\pgfutil@loop ..... 2495	\renewcommand ... 1253, 1254, 1255, 1256, 1259, 1260, 1274, 1275, 1591, 1592, 1993
\or ..... 2329	P	\pgfutil@repeat .... 2498	\repeat ..... 2256
\pagestyle ..... 1273	\par ... 368, 389, 550, 1202, 1210, 1648, 1710, 1733, 1741, 2371, 2393, 2408, 2449, 2735, 2759, 2763, 2771, 2906, 2912	\preceq ..... 1452	\RequirePackage . 4, 67, 170, 234, 238, 420, 426, 427, 487, 557, 636, 637, 1091, 1093, 1094, 1130, 1239, 1240, 1243, 1245, 1247, 1252, 1261, 1268, 1279, 1280, 1289, 1293, 1294, 1295, 1323, 1324, 1372, 1390, 1391, 1430, 1499, 1599, 1604, 1605, 1621, 1628, 1629, 1630, 1658, 1773, 1935, 1936, 2108, 2110, 2200, 2260, 2262, 2414, 2415, 2416, 2429, 2457, 2488, 2539, 2545, 2552, 2557, 2568, 2569, 2637, 2742, 2749, 2779, 2780, 2782, 2783, 2821, 2822, 2823, 2856, 2876
\paragraph ..... 1250	\parindent ..... 2440	\prime ..... 1431	\right ..... 1775
\path ... 106, 153, 216, 588		\printbibliography . 1622	\RIGHTarrow ... 2797, 2802
		\ProvidesPackage 2, 15, 33, 65, 233, 424, 485, 492, 556, 619, 634, 1089, 1125, 1238, 1278, 1287, 1292, 1321, 1359, 1380, 1426, 1498, 1581, 1597, 1603, 1625, 1771, 1932, 1945, 2024, 2107, 2173, 2198, 2203, 2247, 2413, 2426, 2455, 2536, 2543, 2550, 2555, 2635, 2650, 2741, 2745, 2776, 2818, 2855, 2875	\Rightarrow .... 625, 630
		Q	\rightarrow ..... 219, 514, 519, 527, 531, 533, 534, 536, 591, 599, 2170, 2680, 2687, 2689, 2692, 2697, 2702, 2703, 2708
		\QS@list ..... 2373, 2384, 2388, 2395, 2401, 2406, 2409	\rightouterjoin ... 2422
		\QS@select@equal .... ..... 2348, 2352	\Roman ..... 1258
		\QS@select@greater .. ..... 2349, 2353	\roman ..... 1258, 1260
		\QS@select@smaller .. ... 2344, 2347, 2351	\romannumeral ..... 2338
		\QS@sort@a ..... 2326, 2359, 2380, 2381	\rtimes ..... 1934
		\QS@sort@b .... 2326, 2327	\rule 2371, 2393, 2408, 2419
		\QS@sort@c .... 2330, 2337	S
		\QS@sort@d .... 2338, 2346	\sb ..... 79, 89, 91, 120, 178, 526,
		\QS@sort@empty . 2328, 2333	
		\QS@sort@single 2329, 2334	
		\QSinitialize ..... ... 2251, 2363, 2398	
		\QSIr . 2334, 2340, 2348, 2368, 2382, 2387, 2390	
		\QSIrr 2369, 2382, 2383, 2391	
		\QSLr ..... 2340, 2347, 2358, 2359, 2367, 2380, 2385, 2389	



527, 531, 534, 535, 536, 1185, 1187, 1825, 1828, 1831, 1858, 1864, 2013, 2158, 2167, 2702, 2703, 2704, 2709, 2713, 2714, 2717, 2718, 2721, 2722, 2723	\scriptscriptstyle .. ..... 591, 599, 607	\textit ..... 965, 1007, 1008, 1009, 1010, 1751, 2043, 2103	\TmpTransitionEight .. ..... 2119, 2140
\scriptsize .... 1204, 1510, 1517, 1523, 1585, 1586, 1589, 1590, 2826, 2879, 2909	\theop ..... 1272, 1588	\textsc ..... 1432	\TmpTransitionFive .. ..... 2116, 2137
\section ..... 57, 1400	\theparagraph ..... 1250	\textsf ..... 1644, 1725	\TmpTransitionFour .. ..... 2115, 2136
\seq . 1574, 1575, 1576, 1719, 1720, 1721, 1728	\Theta ..... 1778	\textstyle .... 1843, 1875	\TmpTransitionNine .. ..... 2120, 2141
\setbox ..... 2418	\thinspace ..... 2879	\texttt 1119, 1432, 1566, 1567, 1568, 1569, 2879	\TmpTransitionOne ... ..... 2112, 2133
\setcounter ..... 1251, 2372, 2394, 2408	\tikz ..... 1570	\textwidth ..... 1593	\TmpTransitionSeven . ..... 2118, 2139
\setganttlinklabel .. 1281, 1282, 1283, 1284	tikz: bbaum ..... 25	\theop ..... 1272, 1588	\TmpTransitionSix ... ..... 2117, 2138
\setlength ..... 1593, 2440, 2903, 2904, 2914	tikz: li binaer baum .... 23	\theparagraph ..... 1250	\TmpTransitionTen ... ..... 2121, 2142
\setmainfont ..... 1241	\tikzchildnode ..... 444	\Theta ..... 1778	\TmpTransitionThree . ..... 2114, 2135
\setmainlanguage .... 421	\tikzparentnode ..... 444	\thinspace ..... 2879	\TmpTransitionTwo ... ..... 2113, 2134
\setminted .... 2574, 2575	\tikzset ..... 108, 155, 221, 429, 455, 1297, 1501, 2148, 2274, 2478, 2639, 2857	\tikz ..... 1570	\TmpX ..... 2144
\setminus ..... 2073	\tikzumset ..... 2785	tikz: bbaum ..... 25	\TmpY ..... 2145
\setsansfont ..... 1242	\times ..... 219	tikz: li binaer baum .... 23	\today ..... 1586
\setul ..... 2431	\tiny ..... 1102, 1108, 1114, 1569, 1639, 2584	\tikzchildnode ..... 444	\ttfamily ..... 2432
\sffamily ..... 501, 1248, 1250, 1352, 2658	\titleformat .. 1248, 1250	\tikzparentnode ..... 444	
\shoveleft ..... 2053	\titlespacing ..... 1249	\tikzset ..... 108, 155, 221, 429, 455, 1297, 1501, 2148, 2274, 2478, 2639, 2857	U
\shoveright ..... 2057	\tl .... 40, 48, 75, 76, 77, 78, 79, 80, 83, 84, 85, 86, 87, 89, 91, 116, 117, 118, 119, 120, 121, 122, 125, 126, 127, 128, 129, 130, 131, 174, 175, 176, 177, 178, 179, 180, 183, 184, 185, 186, 187, 188, 189, 304, 308, 333, 337, 338, 339, 342, 347, 348, 349, 360, 361, 362, 363, 374, 380, 383, 386, 395, 409, 561, 564, 569, 570, 578, 579, 582, 583, 1193, 1214, 1215, 1216, 1217, 1220, 1221, 1222, 1223, 1394, 1395, 1396, 1399, 1403, 1407, 1414, 1417, 1808	\Sigma ..... 76, 117, 175, 1154, 1155, 1215	\ul ..... 1097, 2430, 2431
\sigma ..... 524, 526, 527	\tmp ..... 2075	\sigma ..... 524, 526, 527	\umlaggreg ..... 1060
\SLASH ..... 1640	\TmpPlaceEight ..... 2130	\SLASH ..... 1640	\umlassoc ..... 1042
\small ..... 1749	\TmpPlaceFive ..... 2127	\small ..... 1749	\umlclass ... 643, 647, 651, 695, 696, 697, 744, 749, 754, 757, 815, 816, 817, 822, 823, 852, 882, 909, 910, 913, 964, 967, 1006, 1012, 1013, 1031, 1032, 1033, 1052, 1053, 1054, 1055
\sort ..... 2505	\TmpPlaceFour ..... 2126	\sort ..... 2505	\umldep ..... 972
\sortList ..... 2504, 2513	\TmpPlaceNine ..... 2131	\sortList ..... 2504, 2513	\umlHVVHaggreg ..... ..... 762, 828, 1022
\square ..... 489	\TmpPlaceOne ..... 2123	\square ..... 489	\umlinherit ..... 701, 752, 917, 962, 970
\stepcounter 2292, 2297, 2302, 2305, 2307, 2311, 2313, 2317, 2319	\TmpPlaceSeven ..... 2129	\stepcounter 2292, 2297, 2302, 2305, 2307, 2311, 2313, 2317, 2319	\umlnote ... 703, 919, 1062
\str .. 502, 511, 1662, 2182, 2191, 2659, 2672	\TmpPlaceSix ..... 2128	\str .. 502, 511, 1662, 2182, 2191, 2659, 2672	\umlreal ..... 699, 760
\string ..... 2055, 2065	\TmpPlaceTen ..... 2132	\string ..... 2055, 2065	\umlsimpleclass .. 658, 659, 660, 664, 666, 667, 668, 694, 847, 848, 849, 908, 960, 961
\StrSubstitute . 2512, 2514	\TmpPlaceThree ..... 2125	\StrSubstitute . 2512, 2514	\umlstatic ..... 854, 883
\strut .... 1880, 1884, 1888, 1892, 1896, 2540	\TmpPlaceTwo ..... 2124	\strut .... 1880, 1884, 1888, 1892, 1896, 2540	\umluniaggreg ..... 915
\subsection ..... 1404	\TmpScale ..... 2143	\subsection ..... 1404	\umluniassoc .... 678, 700, 916, 1040, 1041
\subseq 2683, 2716, 2723		\subseq 2683, 2716, 2723	\umlVHuniassoc .. 679, 680
			\umlVHVdep ..... 672, 673, 675, 676, 856, 857
T			\umlVHVinherit ..... 655, 656, 661, 662, 669, 670, 825, 826, 850, 851, 1020, 1021
\tableofcontents ... 1635			\umlVHVreal ..... 819, 820, 1057, 1058
\text ..... 89, 91, 194, 2035, 2826, 2879			\UParrow ..... 2799
\textbf ... 1096, 1457, 1466, 1477, 1486, 1644, 1651, 1682, 1710, 1725, 1741, 1998			\url ..... 1760
\textcolor .... 1569, 2825			

<code>\usemintedstyle</code> .... 2571	<code>\vspace</code> ..... 1696, 1714, 2767, 2769	<code>\xintifEq</code> ..... 2352
<code>\usetikzlibrary</code> .. 68, 428, 1092, 1296, 1500, 2109, 2417, 2458, 2489, 2784, 2872		<code>\xintifForLast</code> ..... ... 2308, 2314, 2320
	<b>X</b>	<code>\xintifGt</code> ..... 2353
	<code>\xappto</code> .. 2516, 2522, 2526	<code>\xintifLt</code> ..... 2351
	<code>\xdef</code> ..... 1326	<code>\xintLength</code> ..... 2326
<b>V</b>	<code>\xintApply</code> ..... 2342	<code>\xintnthelt</code> ..... 2338
<code>\value</code> ..... 2254	<code>\xintApplyUnbraced</code> .. 2341, 2347, 2348, 2349	<b>Z</b>
<code>\varepsilon</code> ..... 503, 514, 515, 1140, 1825, 1831, 1861, 1872	<code>\xintCSVtoList</code> ..... 2401	<code>\ZB</code> ..... 1767
	<code>\xintFor</code> ..... 2291, 2296, 2301, 2306, 2312, 2318, 2359	<code>\zB</code> ..... 1766
<code>\vfill</code> ..... 2540		<code>\zustandsnamens@liste</code> ... 1158, 1165, 1166
<code>\vrule</code> ..... 2754, 2758		