

lehramt-informatik

Hermine Bschlangaul *

August 29, 2021

Contents

1	Klassen	3
1.1	Vorlage Theorie-Teil	4
1.2	Vorlage Aufgabensammlung	5
1.3	Vorlage Aufgabe	6
2	Pakete	7
2.1	aufgaben-einbinden.sty	8
2.2	aufgaben-titel.sty	9
2.3	automaten.sty	10
2.3.1	Endlicher Automat	10
2.3.2	Kellerautomat	12
2.3.3	Turingmaschine	13
2.4	basis.sty	16
2.5	baum.sty	20
2.5.1	Binärbaum	21
2.5.2	AVL-Baum	22
2.5.3	B-Baum	23
2.6	checkbox.sty	24
2.7	chomsky-normalform.sty	25
2.8	cpm.sty	27
2.8.1	Hilfsmakros für Zeitpunkt-Tabelle	28
2.9	cyk-algorithmus.sty	30
2.10	entwurfsmuster.sty	31
2.10.1	Namensschema der Entwurfsmuster-Makros:	31
2.10.2	Reihenfolge	31
2.10.3	Abstrakte Fabrik (Abstract Factory)	31
2.10.4	Adapter	32
2.10.5	Beobachter (Observer)	34
2.10.6	Dekorierer (Decorator)	36
2.10.7	Einfache Fabrik (Simple Factory)	37
2.10.8	Einzelstück (Singleton)	38
2.10.9	Erbauer (Builder)	38
2.10.10	Fabrikmethode (Factory Method)	40
2.10.11	Kompositum (Composite)	41
2.10.12	Modell-Präsentation-Steuerung (Model-View-Controller)	42
2.10.13	Zustand (State)	43
2.11	er.sty	45
2.12	formale-sprachen.sty	47
2.13	formatierung.sty	50
2.13.1	Schriftarten / Typographie	50
2.13.2	Farben	50
2.13.3	Überschriften	50

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

2.13.4	Listen	50
2.13.5	Kasten	50
2.13.6	Header	50
2.14	gantt.sty	51
2.15	grafik.sty	52
2.16	graph.sty	53
2.17	hanoi.sty	55
2.18	komplexitaetstheorie.sty	56
2.19	kontrollflussgraph.sty	58
2.20	kopf-fusszeilen.sty	60
2.21	literatur-dummy.sty	61
2.22	literatur.sty	62
2.23	makros.sty	63
2.24	master-theorem.sty	67
2.25	mathe.sty	71
2.26	minimierung.sty	72
2.27	normalformen.sty	75
2.28	petri.sty	78
2.29	potenzmengen-konstruktion.sty	80
2.30	pseudo.sty	82
2.31	pumping-lemma.sty	83
2.32	quicksort.sty	84
2.33	relationale-algebra.sty	87
2.34	rmodell.sty	88
2.35	sortieren.sty	89
2.36	spalten.sty	91
2.37	struktogramm.sty	92
2.38	syntax.sty	93
2.39	syntaxbaum.sty	95
2.40	synthese-algorithmus.sty	96
2.41	tabelle.sty	99
2.42	typographie.sty	100
2.43	uml.sty	101
2.44	vollstaendige-induktion.sty	103
2.45	wasserfall.sty	105
2.46	wpkalkuel.sty	106

3 Index

107

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 aufgaben-einbinden.sty

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

\liAufgabe Eine Aufgaben mit den Pfad relativ zu Wurzelverzeichnis dieses Repository einbinden.
3 \def\liAufgabe#1{
4   \input{\LehramtInformatikRepository/#1.tex}
5 }

\liExamensAufgabe Eine Examensaufgaben mit den Pfad relativ zu Staatsexamen einbinden, z. B.
\liExamensAufgabe{46116/2015/09/Thema-2/Teilaufgabe-1/Aufgabe-3}
6 \def\liExamensAufgabe#1{
7   \input{\LehramtInformatikRepository/Staatsexamen/#1.tex}
8 }

\liExamensAufgabeTTA
9 \def\liExamensAufgabeTTA #1 / #2 / #3 : Thema #4 Teilaufgabe #5 Aufgabe #6 {
10  \input{\LehramtInformatikRepository/Staatsexamen/#1/#2/#3/Thema-#4/Teilaufgabe-
    #5/Aufgabe-#6.tex}
11 }

\liExamensAufgabeTA
12 \def\liExamensAufgabeTA #1 / #2 / #3 : Thema #4 Aufgabe #5 {
13   \input{\LehramtInformatikRepository/Staatsexamen/#1/#2/#3/Thema-#4/Aufgabe-#5.tex}
14 }

\liExamensAufgabeA
15 \def\liExamensAufgabeA #1 / #2 / #3 : Aufgabe #4 {
16   \input{\LehramtInformatikRepository/Staatsexamen/#1/#2/#3/Aufgabe-#4.tex}
17 }

18
```


2.2 aufgaben-titel.sty

```
19 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
20 \ProvidesPackage{lehramt-informatik-aufgaben-titel}[2020/07/07 Zum
21 Setzen von Aufgaben-Titel.]

22 \ExplSyntaxOn

\liAufgabenTitel Momentan eine dummy Makro das die Thematik enthält.
23 \def\liAufgabenTitel#1{
24 }

\liSetzeAufgabenTitel Setzen des Titels über eine plist bzw über key-values. Das Makro sollte irgendwann
einmal in \liAufgabenTitel umbenannt werden.
Die Schlüssel-Werte-Paare sind in der Datei basis.sty definiert.

\liSetzeAufgabenTitel{
Titel = Aufgabe 2,
Thematik = Petri-Netz,
Fussnote = sosy:pu:4,
FussnoteSeite = Seite 11,
RelativerPfad = Staatsexamen/46116/2016/03/Thema-2/Teilaufgabe-1/Aufgabe-2.tex,
ExamenNummer = 46116,
ExamenJahr = 2016,
ExamenMonat = 03,
ExamenThemaNr = 2,
ExamenTeilaufgabeNr = 1,
ExamenAufgabeNr = 2,
}

25 \def\liSetzeAufgabenTitel#1{
26
27   \_setze_variablen_zurueck:
28
29   \tl_clear:N \g_auf_examen_telaufgabe_nr_tl
30
31   \keys_set:nn { aufgabentitel } {
32     #1
33   }
34
35   \_setze_relativen_pfad:
36
37   \tl_if_empty:NTF \g_auf_titel_tl
38   {
39     \msg_fatal:nn { aufgabentitel } { kein-titel }
40   }
41   {
42   }
43
44   \_gib_examen_titel: {}
45
46   \section{\_gib_aufgaben_titel:}
47
48   \bigskip
49 }

50 \ExplSyntaxOff
51
```

2.3 automaten.sty

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

2.3.1 Endlicher Automat

```
54 \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}
```



```
55 \RequirePackage{tikz}
56 \usetikzlibrary{arrows,automata,positioning}
57 \liLadePakete{mathe}
58 \directlua{
59   automaten = require('lehramt-informatik-automaten')
60 }
```

`\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}`: $A_{NEA} = (Z, \Sigma, \delta, E, z_0)$

```
61 \ExplSyntaxOn
62 \NewDocumentCommand {\liAutomat} { 0{A} m } {
63   \tl_set:Nn \l_zustaende_tl {Z}
64   \tl_set:Nn \l_alphabet_tl {\Sigma}
65   \tl_set:Nn \l_delta_tl {\delta}
66   \tl_set:Nn \l_ende_tl {E}
67   \tl_set:Nn \l_start_tl {z\sb{0}}
68   \tl_set:Nn \l_typ_tl {}
69
70   \keys_define:nn { automat } {
71     zustaende .code:n = {\tl_set:Nn \l_zustaende_tl {\liMenge{##1}}},
72     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
73     delta .code:n = {\tl_set:Nn \l_delta_tl {##1}},
74     ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
```

```

75     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
76     dea .value_forbidden:n = true,
77     dea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{DEA}}}},
78     nea .value_forbidden:n = true,
79     nea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{NEA}}}},
80   }
81
82   \keys_set:nn { automat } { #2 }
83
84   $#1 \l_typ_tl = (
85     \l_zustaende_tl,
86     \l_alphabet_tl,
87     \l_delta_tl,
88     \l_ende_tl,
89     \l_start_tl
90   )$
91 }
92 \ExplSyntaxOff

```

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

```

93 \def\liAutomatenKante#1#2#3#4{
94   \path (#1) edge[#4] node{#3} (#2);
95 }

96 \tikzset{
97   li automat/.style={
98     ->,
99     node distance=2cm
100   },
101 }

```

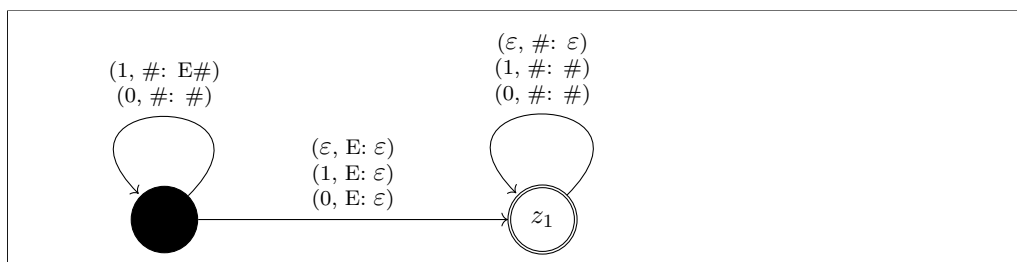
2.3.2 Kellerautomat

```
\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[\langle automaten-name \rangle]
{\zustaende=Z,alphabet=Sigma,kelleralphabet=Gamma,delta=delta,start=z0,kellerboden=#,ende=E}

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

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

```
102 \ExplSyntaxOn
103 \NewDocumentCommand {\liKellerAutomat} { O{K} m } {
104   \tl_set:Nn \l_zustaende_tl {Z}
105   \tl_set:Nn \l_alphabet_tl {\Sigma}
106   \tl_set:Nn \l_kelleralphabet_tl {\Gamma}
107   \tl_set:Nn \l_delta_tl {\delta}
108   \tl_set:Nn \l_start_tl {z\sb{0}}
109   \tl_set:Nn \l_kellerboden_tl {\#}
110   \tl_set:Nn \l_ende_tl {E}
111
112   \keys_define:nn { kellerautomat } {
113     zustaende .code:n = {\tl_set:Nn \l_zustaende_tl {\liMenge{##1}}},
114     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
115     kelleralphabet .code:n = {\tl_set:Nn \l_kelleralphabet_tl {\liMenge{##1}}},
116     delta .code:n = {\tl_set:Nn \l_delta_tl {##1}},
117     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
```

```

118     kellerboden .code:n = {\tl_set:Nn \l_kellerboden_tl {##1}},
119     ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
120 }
121
122 \keys_set:nn { kellerautomat } { #2 }
123
124 $#1 = (
125     \l_zustaende_tl,
126     \l_alphabet_tl,
127     \l_kelleralphabet_tl,
128     \l_delta_tl,
129     \l_start_tl,
130     \l_kellerboden_tl,
131     \l_ende_tl
132 )$
133 }
134 \ExplSyntaxOff

```

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

(a, #: A)
(b, #: ε)

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

```

135 \ExplSyntaxOn
136 \def\liKellerUebergang#1{
137     \directlua{automaten.drucke_keller_uebergaenge('#1')}
138 }
139 \ExplSyntaxOff

```

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

```

140 \NewDocumentCommand{\liKellerKante} { 0{above} m m m } {
141     \path (#2) edge[#1] node{\liKellerUebergang{#4}} (#3);
142 }
143 \tikzset{
144     li keller knoten/.style={
145         text width=2cm,
146         align=center,
147         font=\footnotesize,
148     },
149     li kellerautomat/.style={
150         li automat,
151         every edge/.append style={
152             every node/.style={
153                 li keller knoten
154             }
155         }
156     }
157 }

```

2.3.3 Turingmaschine

```

158 \RequirePackage{amssymb}

```

\liTuringLeerzeichen

□

```

159 \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},
}

```

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

```

160 \ExplSyntaxOn
161 \NewDocumentCommand {\liTuringMaschine} { O{TM} m } {
162   \tl_set:Nn \l_zustaende_tl {Z}
163   \tl_set:Nn \l_alphabet_tl {\Sigma}
164   \tl_set:Nn \l_bandalphabet_tl {\Gamma}
165   \tl_set:Nn \l_delta_tl {\delta}
166   \tl_set:Nn \l_start_tl {z\sb{0}}
167   \tl_set:Nn \l_leerzeichen_tl {\liTuringLeerzeichen}
168   \tl_set:Nn \l_ende_tl {E}
169
170   \keys_define:nn { kellerautomat } {
171     zustaende .code:n = {\tl_set:Nn \l_zustaende_tl {\liMenge{##1}}},
172     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
173     bandalphabet .code:n = {\tl_set:Nn \l_bandalphabet_tl {\liMenge{##1}}},
174     delta .code:n = {\tl_set:Nn \l_delta_tl {##1}},
175     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
176     leerzeichen .code:n = {\tl_set:Nn \l_leerzeichen_tl {##1}},
177     ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
178   }
179
180   \keys_set:nn { kellerautomat } { #2 }
181
182   $\text{#1} = (
183     \l_zustaende_tl,
184     \l_alphabet_tl,
185     \l_bandalphabet_tl,
186     \l_delta_tl,
187     \l_start_tl,
188     \l_leerzeichen_tl,
189     \l_ende_tl
190   )$
191 }
192 \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)`

```

193 \ExplSyntaxOn
194 \def\liTuringUebergangZelle#1{
195   \directlua{tex.print(automaten.gib_einen_turing_uebergang('#1'))}
196 }
197 \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)`

```

198 \ExplSyntaxOn
199 \def\liTuringUebergaenge#1{
200   \directlua{automaten.drucke_turing_uebergaenge('#1')}
201 }
202 \ExplSyntaxOff

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

203 \NewDocumentCommand{\liTuringKante} { 0{above} m m m } {
204   \path (#2) edge[#1] node{\liTuringUebergaenge{#4}} (#3);
205 }

\liTuringUeberfuehrung

206 \def\liTuringUeberfuehrung{
207    $\Delta : Z \times \Gamma \rightarrow Z \times \Gamma \times \{ L, R, N \}$ 
208 }

209 \tikzset{
210   li turingmaschine/.style={
211     li automat,
212     every edge/.append style={
213       every node/.style={
214         li keller knoten
215       }
216     }
217   }
218 }
219

```

2.4 basis.sty

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

222 \RequirePackage{xparse}
223 \ExplSyntaxOn

\liLadePakete

224 \NewDocumentCommand{\liLadePakete}{ m }
225 {
226   \clist_map_inline:nn { #1 } { \RequirePackage{lehramt-informatik-##1} }
227 }

\liLadeAllePakete

228 \def\liLadeAllePakete{
229   \liLadePakete{
230     aufgaben-einbinden,
231     automaten,
232     baum,
233     checkbox,
234     chomsky-normalform,
235     cpm,
236     cyk-algorithmus,
237     entwurfsmuster,
238     er,
239     formale-sprachen,
240     gantt,
241     grafik,
242     graph,
243     hanoi,
244     kontrollflussgraph,
245     makros,
246     master-theorem,
247     mathe,
248     minimierung,
249     normalformen,
250     petri,
251     potenzmengen-konstruktion,
252     pseudo,
253     quicksort,
254     relationale-algebra,
255     rmodell,
256     sortieren,
257     spalten,
258     struktogramm,
259     syntax,
260     syntaxbaum,
261     synthese-algorithmus,
262     tabelle,
263     typographie,
264     uml,
265     vollstaendige-induktion,
266     wasserfall,
267     wpkalkuel,
268   }
269 }
```

Definition einer Komma-getrennten-Liste mit deren Hilfe die vielen globalen Token-List-Variablen definiert werden können.

```
270 \clist_new:N \g_auf_schluesel_clist
271 \clist_set:Nn \g_auf_schluesel_clist {
272   titel,
273   thematik,
```



```

274 fussnote,
275 fussnote_seite,
276 relativer_pfad,
277 examen_nummer,
278 examen_jahr,
279 examen_monat,
280 examen_thema_nr,
281 examen_teilaufgabe_nr,
282 examen_aufgabe_nr,
283 }

```

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

```

284 \clist_map_inline:Nn \g_auf_schluesssel_clist {
285   \tl_new:c {g_auf_#1_tl}
286 }

```

Funktion über alle globalen Token-List-Variablen zurückzusetzen.

```

287 \cs_new:Npn \setze_variablen_zurueck: {
288   \clist_map_inline:Nn \g_auf_schluesssel_clist {
289     \tl_clear:c {g_auf_##1_tl}
290   }
291 }

```

```

292 \keys_define:nn { aufgabentitel }
293 {
294   Titel .tl_gset:N = \g_auf_titel_tl,
295   Thematik .tl_gset:N = \g_auf_thematik_tl,
296   Fussnote .tl_gset:N = \g_auf_fussnote_tl,
297   FussnoteSeite .tl_gset:N = \g_auf_fussnote_seite_tl,
298   RelativerPfad .tl_gset:N = \g_auf_relativer_pfad_tl,
299   ExamenNummer .tl_gset:N = \g_auf_examen_nummer_tl,
300   ExamenJahr .tl_gset:N = \g_auf_examen_jahr_tl,
301   ExamenMonat .tl_gset:N = \g_auf_examen_monat_tl,
302   ExamenThemaNr .tl_gset:N = \g_auf_examen_thema_nr_tl,
303   ExamenTeilaufgabeNr .tl_gset:N = \g_auf_examen_teilaufgabe_nr_tl,
304   ExamenAufgabeNr .tl_gset:N = \g_auf_examen_aufgabe_nr_tl,
305 }

```

```

306 \cs_gset:Npn \setze_relativen_pfad: {
307   \tl_if_empty:NTF \g_auf_relativer_pfad_tl
308   {
309     \bool_if:nTF
310     {
311       ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
312       ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
313       ! \tl_if_empty_p:N \g_auf_examen_monat_tl
314     }
315     {
316       \tl_gset:Nn \g_auf_relativer_pfad_tl {
317         Staatsexamen /
318         \g_auf_examen_nummer_tl /
319         \g_auf_examen_jahr_tl /
320         \g_auf_examen_monat_tl /
321         \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} { Thema - \g_auf_examen_thema_nr_tl / }
322         \tl_if_empty:NTF \g_auf_examen_teilaufgabe_nr_tl {} { Teilaufgabe - \g_auf_examen_teilaufgabe_nr_tl / }
323         \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} { Aufgabe - \g_auf_examen_aufgabe_nr_tl / }
324       }
325     }
326   } {}
327 }
328 {}
329 }

```

```

330 \cs_gset:Npn \gib_examen_titel: {
331   \cs_set:Nn \trenner: { \, / \, }

```

```

332 \bool_if:nTF
333 {
334   ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
335   ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
336   ! \tl_if_empty_p:N \g_auf_examen_monat_tl &&
337   ! \tl_if_empty_p:N \g_auf_examen_aufgabe_nr_tl
338 }
339 {
340   {
341     \footnotesize
342     \par
343     \noindent
344     Staatsexamen ~
345     \g_auf_examen_nummer_tl \_trenner:
346     \g_auf_examen_jahr_tl \_trenner:
347
348     \tl_case:Nn \g_auf_examen_monat_tl
349     {
350       { 03 } { Frühjahr }
351       { 09 } { Herbst }
352     } \_trenner:
353
354     \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} {
355       Thema ~ Nr. ~ \g_auf_examen_thema_nr_tl \_trenner:
356     }
357     \tl_if_empty:NTF \g_auf_examen_teilaufgabe_nr_tl {} {
358       Teilaufgabe ~ Nr. ~ \g_auf_examen_teilaufgabe_nr_tl \_trenner:
359     }
360     \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} {
361       Aufgabe ~ Nr. ~ \g_auf_examen_aufgabe_nr_tl
362     }
363     \par
364     \bigskip
365   }
366 }
367 }

368 \cs_new:Npn \_gib_github_url: {
369   \tl_if_empty:NTF \g_auf_relativer_pfad_tl {} {
370     Github :~\href{
371       \LehramtInformatikGithubRawDomain /
372       \LehramtInformatikGithubTexRepo /
373       \LehramtInformatikGitBranch /
374       \g_auf_relativer_pfad_tl
375     }{
376       \g_auf_relativer_pfad_tl
377     }
378   }
379 }

380 \cs_new:Npn \_gib_aufgaben_titel: {
381   \g_auf_titel_tl
382
383   \tl_if_empty:NTF \g_auf_thematik_tl
384   {}
385   {
386     \, ~ [
387       \g_auf_thematik_tl
388     ]
389   }
390 }

391 \msg_new:nnn { aufgabentitel } { kein-titel }
392 { Der~Schlüssel~ist~zwingend~notwendig }
393 \ExplSyntaxOff

```

```

\def\LehramtInformatikRepository{/pfad/zum/repository}
394 \input /etc/lehramt-informatik.config.tex
    biblatex not working with lualatex and babel
395 % \RequirePackage{polyglossia}
396 % \setmainlanguage{german}
397
```

2.5 baum.sty

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

401 \RequirePackage{tikz}

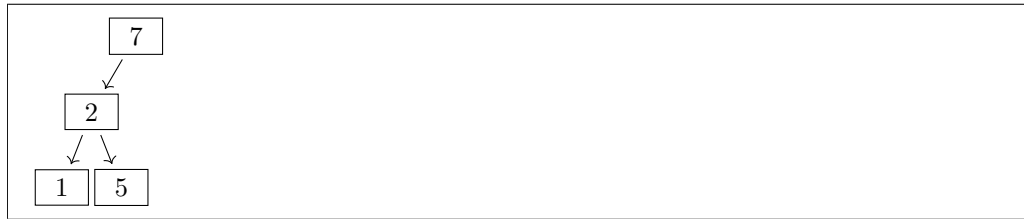
    für li binaer baum
402 \RequirePackage{tikz-qtree}

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

2.5.1 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}
```



```
404 \tikzset{
405   li binaer baum/.style={
406     shorten <=2pt,
407     shorten >=2pt,
408     ->,
409     every tree node/.style={
410       minimum width=2em,
411       draw,
412       rectangle
413     },
414     blank/.style={
415       draw=none
416     },
417     edge from parent/.style={
418       draw,
419       edge from parent path={(\tikzparentnode) -- (\tikzchildnode)}
420     },
421     level distance=1cm,
422     every label/.style={
423       gray,
424       font=\footnotesize,
425       label position=0,
426       label distance=0cm,
427     }
428   },
429 }
```

2.5.2 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}
```



2.5.3 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}

```



```

430 \tikzset{
431   li bbaum knoten/.style={
432     rectangle split parts=10,
433     rectangle split,
434     rectangle split horizontal,
435     rectangle split ignore empty parts,
436     draw,
437     fill=white
438   },
439   li bbaum/.style={
440     every node/.style={
441       li bbaum knoten
442     },
443     level 1/.style={
444       level distance=12mm,
445       sibling distance=25mm,
446     },
447     every child/.style={
448       shorten <= 2pt,
449       shorten >= 6pt,
450       ->,
451     },
452     level 2/.style={
453       level distance=9mm,
454       sibling distance=15mm,
455     },
456   }
457 }
458

```

2.6 checkbox.sty

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

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

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

465
```


2.7 chomsky-normalform.sty

```

466 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
467 \ProvidesPackage{lehramt-informatik-chomsky-normalform}[2021/03/26
468 Hilfsmakros zum Setzen des Algorithmus zum Erreichen der Chomsky-Normalform]
469
470 \ExplSyntaxOn
471
472 \liLadePakete{typographie}

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

473 \def\liChomskyUeberschrift#1{
474   {
475     \bfseries
476     \sffamily
477     \str_case:nn {#1} {
478       {1} {Elimination~der~ $\varepsilon$ -Regeln}
479       {2} {Elimination~von~Kettenregeln}
480       {3} {Separation~von~Terminalzeichen}
481       {4} {Elimination~von~mehrelementigen~Nonterminalketten}
482     }
483   }
484 }

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

485 \def\li@chomsky@erklaerung@texte#1{
486   \str_case:nn {#1} {
487     %
488     {1} {
489       Alle~Regeln~der~Form~ $A \rightarrow \varepsilon$ ~werden~eliminiert.~
490       Die~Ersetzung~von~ $A$ ~wird~durch~ $\varepsilon$ ~in~allen~anderen~
491       Regeln~vorweggenommen.
492     }
493     {2} {
494       Jede~Produktion~der~Form~ $A \rightarrow B$ ~mit~ $A, B \in S$ ~wird~
495       als~Kettenregel~bezeichnet.~Diese~tragen~nicht~zur~Produktion~
496       von~Terminalzeichen~bei~und~lassen~sich~ebenfalls~eliminieren.
497     }
498     {3} {
499       Jedes~Terminalzeichen~ $\sigma$ ,~das~in~Kombination~mit~anderen~
500       Symbolen~auftaucht,~wird~durch~ein~neues~Nonterminal~
501        $S_{\sigma}$ ~ersetzt~und~die~Menge~der~Produktionen~durch~die~
502       Regel~ $S_{\sigma} \rightarrow \sigma$ ~ergänzt.
503     }
504     {4} {
505       Alle~Produktionen~der~Form~
506        $A \rightarrow B_{\{1\}} B_{\{2\}} \dots B_{\{n\}}$ ~
507       werden~in~die~Produktionen~
508        $A \rightarrow$ 
509        $A_{\{n-1\}} B_{\{n\}}, A_{\{n-1\}} \rightarrow$ 
510        $A_{\{n-2\}} B_{\{n-1\}}, \dots,$ 
511        $A_{\{2\}} \rightarrow B_{\{1\}} B_{\{2\}}$ ~zerteilt.~
512       Nach~der~Ersetzung~sind~alle~längeren~Nonterminalketten~
513       vollständig~heruntergebrochen~und~die~Chomsky-Normalform~erreicht.
514     }
515   }
516 }

517 \def\liChomskyErklaerung#1{
518   {
519     \itshape
520     \footnotesize
521     \liParagraphMitLinien{\li@chomsky@erklaerung@texte{#1}}
522   }

```

523 }

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

524 \def\liChomskyUeberErklaerung#1{

525 \liChomskyUeberschrift{#1}\par

526 \liChomskyErklaerung{#1}

527 }

528 \ExplSyntaxOff

529

2.8 cpm.sty

```

530 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
531 \ProvidesPackage{lehramt-informatik-cpm}[2020/09/03]
532 \RequirePackage{tikz}
533 \liLadePakete{mathe}

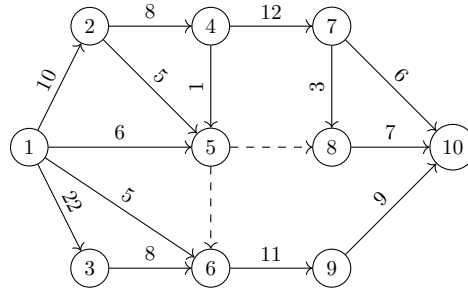
```

Let-Abkürzungen

```

\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}{5}
\liCpmVorgang[schein]{5}{8}{7}
\end{tikzpicture}

```

```

\begin{tabular}{|l|l|l|l|l|l|l|l|}
\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}

```

```

534 \ExplSyntaxOn
535 \NewDocumentCommand { \liCpmEreignis } { 0{} m m m } {
536   \tl_set:Nn \l_name_tl {}
537
538   \keys_define:nn { cpmEreignis } {
539     name .code:n = {\tl_set:Nn \l_name_tl {##1}},
540   }
541
542   \keys_set:nn { cpmEreignis } { #1 }
543
544   \tl_if_empty:NT \l_name_tl {
545     \tl_set:Nn \l_name_tl {#2}
546   }

```

```

547
548 \node[circle,draw] (\l_name_tl) at (#3,#4) {#2};
549 }
550 \ExplSyntaxOff

\liCpmVorgang liCpmVORGANG\((.*)>(.*))\{(.*)\}
551 \ExplSyntaxOn
552 \NewDocumentCommand { \liCpmVorgang } { 0{ } m m m } {
553   \tl_set:Nn \l_schein_tl {}
554   \tl_set:Nn \l_kritisch_tl {}
555
556   \keys_define:nn { cpmVorgang } {
557     schein .code:n = {\tl_set:Nn \l_schein_tl {dashed}},
558     kritisch .code:n = {\tl_set:Nn \l_kritisch_tl {thick}},
559   }
560
561   \keys_set:nn { cpmVorgang } { #1 }
562
563   \path[->,\l_schein_tl,\l_kritisch_tl] (#2) edge node[auto,sloped] {#4} (#3);
564 }
565 \ExplSyntaxOff

```

2.8.1 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):  $l_{(2 \rightarrow 3)}$ 
566 \def\liCpmVonZuOhneMathe#1(#2-#3){#1_{\scriptscriptstyle(#2\rightarrow#3)}}
567 \def\liCpmVonZu#1(#2-#3){%
568   \ifmmode%
569     \liCpmVonZuOhneMathe{#1}(#2-#3)%
570   \else%
571     $\liCpmVonZuOhneMathe{#1}(#2-#3)$%
572   \fi%
573 }

```

```

\liCpmVon Let-Abkürzung: \let\v=\liCpmVon
\liCpmVon{1}(2):  $l_{(\rightarrow 2)}$ 
574 \def\liCpmVonOhneMathe#1(#2){#1_{\scriptscriptstyle(\rightarrow#2)}}
575 \def\liCpmVon#1(#2){%
576   \ifmmode%
577     \liCpmVonOhneMathe{#1}(#2)%
578   \else%
579     $\liCpmVonOhneMathe{#1}(#2)$%
580   \fi%
581 }

```

```

\liCpmZu Let-Abkürzung: \let\z=\liCpmZu
\liCpmZu{1}(2):  $l_{(\leftarrow 2)}$ 
582 \def\liCpmZuOhneMathe#1(#2){#1_{\scriptscriptstyle(\leftarrow#2)}}
583 \def\liCpmZu#1(#2){%
584   \ifmmode%
585     \liCpmZuOhneMathe{#1}(#2)%
586   \else%
587     $\liCpmZuOhneMathe{#1}(#2)$%

```

```

588 \fi%
589 }

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

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

592

```

2.9 cyk-algorithmus.sty

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

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

$R_a$ & $R_c$ & $R_b$ & $R_c$ & $R_a$ & $R_b$ \16
B      & A      & A      & B      & C \15
S      & -      & S      & S \14
-      & -      & - \13
-      & - \12
S \11
\end{tabular}
```

```
\liKurzeTabellenLinie Let-Abkürzung: \let\l=\liKurzeTabellenLinie
596 \def\liKurzeTabellenLinie#1{\ccline{1-#1}}
```

```
\liWortInSprache  $\Rightarrow abc \in L(Y)$ 
597 \NewDocumentCommand{ \liWortInSprache } { m O{L(G)} } {
598   \bigskip
599   \noindent
600   $\Rrightarrow$ #1 \in #2$
601 }
```

```
\liWortNichtInSprache  $\Rightarrow abc \notin L(G)$ 
602 \NewDocumentCommand{ \liWortNichtInSprache } { m O{L(G)} } {
603   \bigskip
604   \noindent
605   $\Rrightarrow$ #1 \notin #2$
606 }

607
```

2.10 entwurfsmuster.sty

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

2.10.1 Namensschema der Entwurfsmuster-Makros:

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

2.10.2 Reihenfolge

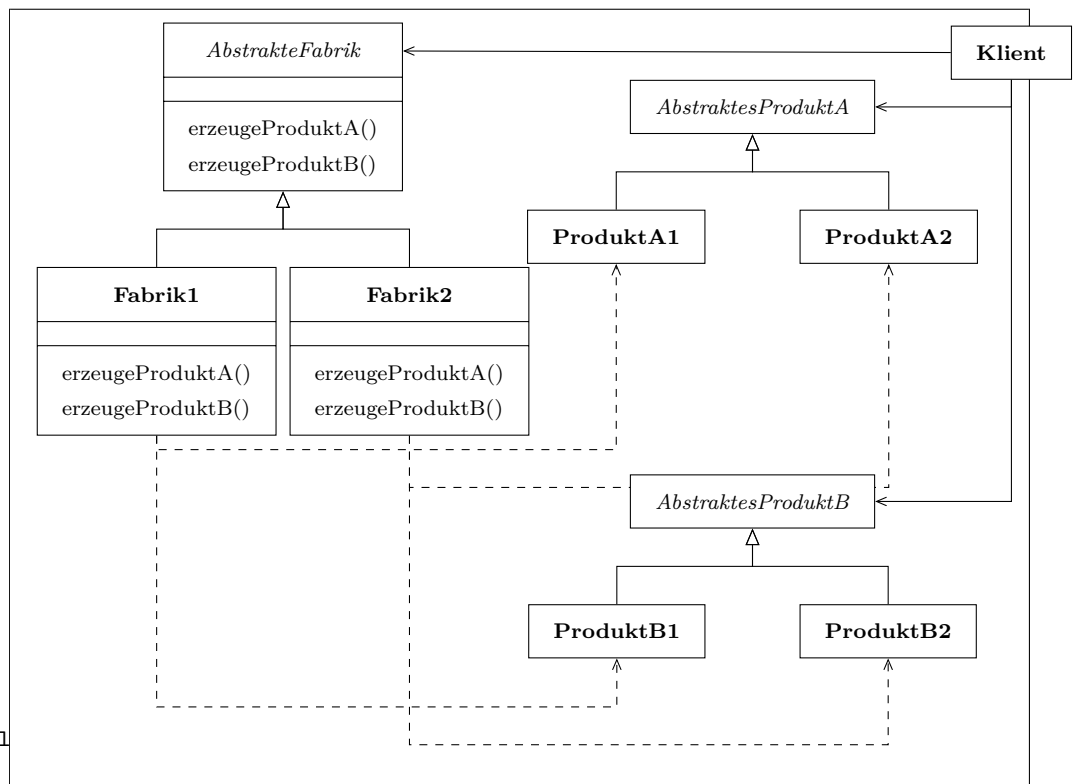
1. Uml: Uml-Klassendiagramm \liEntwurfsEinzelstueckUml
2. Akteure: Akteure, beteiligte Klassen \liEntwurfsEinzelstueckAkteure
3. Code: Allgemeines Code-Beispiel \liEntwurfsEinzelstueckCode
4. ohne: Ohne Suffix, Bündelung der einzelnen Makros eines Entwurfsmusters \liEntwurfsEinzelstueckAkteure

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

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

```
612 \RequirePackage{lehramt-informatik-syntax}
613 \def\liEntwurfsCode#1#2{
614   \liJavaDatei{entwurfsmuster/#1/allgemein/#2}
615 }
```

2.10.3 Abstrakte Fabrik (Abstract Factory)



```
616 \def\liEntwurfsAbstrakteFabrikUml{
617   \begin{tikzpicture}
618     \umlclass[type=abstract]{AbstrakteFabrik}{-}{-}{
619       erzeugeProduktA()\
620       erzeugeProduktB()\
621     }
622   }
```

```

622 \umlclass[below left=1cm and -1.5cm of AbstrakteFabrik]{Fabrik1}{-}{
623     erzeugeProduktA()\
624     erzeugeProduktB()\
625 }
626 \umlclass[below right=1cm and -1.5cm of AbstrakteFabrik]{Fabrik2}{-}{
627     erzeugeProduktA()\
628     erzeugeProduktB()\
629 }
630 \umlVHVinherit{Fabrik1}{AbstrakteFabrik}
631 \umlVHVinherit{Fabrik2}{AbstrakteFabrik}
632
633 \umlsimpleclass[right=3cm of AbstrakteFabrik,type=abstract]{AbstraktesProduktA}
634 \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktA]{ProduktA1}
635 \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktA]{ProduktA2}
636 \umlVHVinherit{ProduktA1}{AbstraktesProduktA}
637 \umlVHVinherit{ProduktA2}{AbstraktesProduktA}
638
639 \umlsimpleclass[above right=0cm and 1cm of AbstraktesProduktA]{Klient}
640
641 \umlsimpleclass[below=4.5cm of AbstraktesProduktA,type=abstract]{AbstraktesProduktB}
642 \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktB]{ProduktB1}
643 \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktB]{ProduktB2}
644 \umlVHVinherit{ProduktB1}{AbstraktesProduktB}
645 \umlVHVinherit{ProduktB2}{AbstraktesProduktB}
646
647 \umlVHVdep[arm1=-1.3cm,arm2=-1cm]{Fabrik1}{ProduktA1}
648 \umlVHVdep[arm2=-1cm]{Fabrik1}{ProduktB1}
649
650 \umlVHVdep[arm1=-1.8cm,arm2=-1.5cm]{Fabrik2}{ProduktA2}
651 \umlVHVdep[arm2=-1.5cm]{Fabrik2}{ProduktB2}
652
653 \umluniassoc[anchor2=25]{Klient}{AbstrakteFabrik}
654 \umlVHuniassoc[arm1=-1cm]{Klient}{AbstraktesProduktA}
655 \umlVHuniassoc{Klient}{AbstraktesProduktB}
656 \end{tikzpicture}
657 }

```

iEntwurfsAbstrakteFabrikCode

```

658 \def\liEntwurfsAbstrakteFabrikCode{
659     \li@EntwurfsCode{abstrakte_fabrik}{Produkte}
660     \li@EntwurfsCode{abstrakte_fabrik}{AbstrakteFabrik}
661     \li@EntwurfsCode{abstrakte_fabrik}{Klient}
662 }

```

\liEntwurfsAbstrakteFabrik

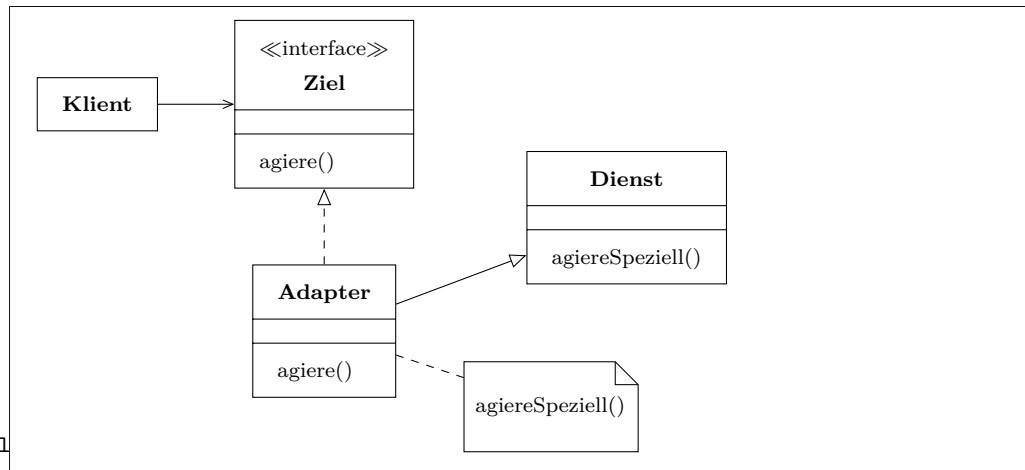
```

663 \def\liEntwurfsAbstrakteFabrik{
664     \liEntwurfsAbstrakteFabrikUml
665     \liEntwurfsAbstrakteFabrikCode
666 }

```

2.10.4 Adapter

\liEntwurfsAdapterUml



```

667 \def\liEntwurfsAdapterUml{
668   \begin{tikzpicture}
669     \umlsimpleclass[x=1,y=3]{Klient}{}{}
670     \umlclass[x=4,y=3,type=interface]{Ziel}{}{agiere()}
671     \umlclass[x=4,y=0]{Adapter}{}{agiere()}
672     \umlclass[x=8,y=1.5]{Dienst}{}{agiereSpeziell()}
673
674     \umlreal{Adapter}{Ziel}
675     \umluniassoc{Klient}{Ziel}
676     \umlinherit{Adapter}{Dienst}
677
678     \umlnote[x=7,y=-1,width=2cm]{Adapter}{agiereSpeziell()}
679   \end{tikzpicture}
680   \footcite[so ähnlich wie GoF]{wiki:adapter}
681 }

```

\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.

```

682 \def\liEntwurfsAdapterAkteure{
683   \begin{description}
684     \item[Ziel (Target)]
685
686     Das Ziel definiert die Schnittstelle, die der Klient nutzen kann.
687
688     \item[Klient (Client)]
689
690     Der Klient nutzt Dienste über inkompatible Schnittstellen und greift
691     dabei auf adaptierte Schnittstellen zurück.
692
693     \item[Dienst (Adaptee)]
694
695     Der Dienst bietet wiederzuverwendende Dienstleistungen mit fest
696     definierter Schnittstelle an.
697
698     \item[Adapter]
699
700     Der Adapter adaptiert die Schnittstelle des Dienstes auf die
701     Schnittstelle zum Klienten.\footcite{wiki:adapter}
702   \end{description}

```

```

703
704 \end{description}
705 }

```

\liEntwurfsAdapterCode

```

706 \def\liEntwurfsAdapterCode{
707 \li@EntwurfsCode{adapter}{Dienst}
708 \li@EntwurfsCode{adapter}{Ziel}
709 \li@EntwurfsCode{adapter}{Adapter}
710 \li@EntwurfsCode{adapter}{Klient}
711 }

```

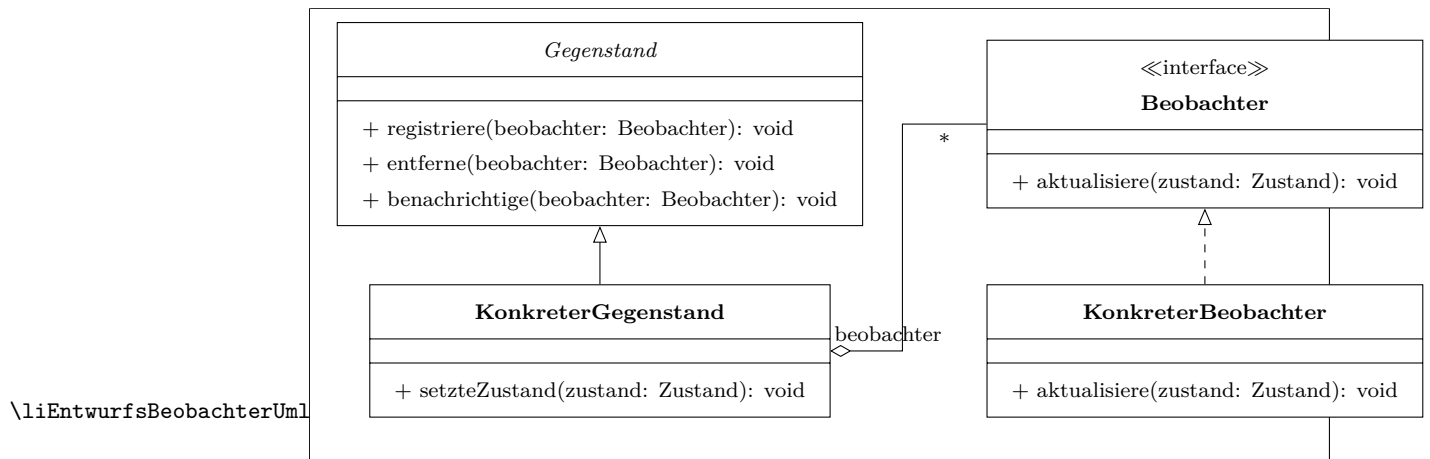
\liEntwurfsAdapter

```

712 \def\liEntwurfsAdapter{
713 \liEntwurfsAdapterUml
714 \liEntwurfsAdapterAkteure
715 \liEntwurfsAdapterCode
716 }

```

2.10.5 Beobachter (Observer)



```

717 \def\liEntwurfsBeobachterUml{
718 \begin{tikzpicture}
719 \umlclass[x=0,y=0,type=abstract]{Gegenstand}{-}{
720 + registriere(beobachter: Beobachter): void\\
721 + entferne(beobachter: Beobachter): void\\
722 + benachrichtige(beobachter: Beobachter): void\\
723 }
724 \umlclass[x=0,y=-3]{KonkreterGegenstand}{-}{
725 + setzteZustand(zustand: Zustand): void
726 }
727 \umlinherit{KonkreterGegenstand}{Gegenstand}
728
729 \umlclass[x=8,y=0,type=interface]{Beobachter}{-}{
730 + aktualisiere(zustand: Zustand): void
731 }
732 \umlclass[x=8,y=-3]{KonkreterBeobachter}{-}{
733 + aktualisiere(zustand: Zustand): void
734 }
735 \umlreal{KonkreterBeobachter}{Beobachter}
736
737 \umlHVHaggreg[arg1=beobachter,pos1=0.8,mult2=*,pos2=2.5]
738 {KonkreterGegenstand}{Beobachter}
739 \end{tikzpicture}
740 }

```

\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.

```
741 \def\liEntwurfsBeobachterAkteure{
742   \begin{description}
743     \item[Gegenstand / Subjekt (Subject / Observable)]
744
745     Ein Subjekt (beobachtbares Objekt, auf Englisch publisher, also
746     „Veröffentlicher“, genannt) hat eine Liste von Beobachtern, ohne
747     deren konkrete Typen zu kennen. Es bietet eine Schnittstelle zur An-
748     und Abmeldung von Beobachtern und eine Schnittstelle zur
749     Benachrichtigung von Beobachtern über Änderungen an.\footcite[Seite
750     251]{gof}
751
752     \item[Beobachter (Observer)]
753
754     Die Beobachter (auf Englisch auch subscriber, also „Abonnent“,
755     genannt) definieren eine Aktualisierungsschnittstelle.
756
757     \item[konkreter/s Gegenstand / Subjekt (ConcreteSubject / ConcreteObservable)]
758
759     Ein konkretes Subjekt (konkretes, beobachtbares Objekt) speichert
760     den relevanten Zustand und benachrichtigt alle Beobachter bei
761     Zustandsänderungen über deren Aktualisierungsschnittstelle. Es
762     verfügt über eine Schnittstelle zur Erfragung des aktuellen
763     Zustands.
764
765     \item[Konkrete Beobachter (ConcreteObserver)]
766
767     Konkrete Beobachter verwalten die Referenz auf ein konkretes
768     Subjekt, dessen Zustand sie beobachten und speichern und dessen
769     Zustand konsistent ist. Sie implementieren eine
770     Aktualisierungsschnittstelle unter Verwendung der
771     Abfrageschnittstelle des konkreten Subjekts.
772     \footcite{wiki:beobachter}
773   \end{description}
774 }
```

\liEntwurfsBeobachterCode

```
775 \def\liEntwurfsBeobachterCode{
776   \li@EntwurfsCode{beobachter}{Gegenstand}
777   \li@EntwurfsCode{beobachter}{KonkreterGegenstand}
778   \li@EntwurfsCode{beobachter}{Beobachter}
779   \li@EntwurfsCode{beobachter}{KonkreterBeobachterA}
780   \li@EntwurfsCode{beobachter}{KonkreterBeobachterB}
```

```

781 \li@EntwurfsCode{beobachter}{Klient}
782 }

```

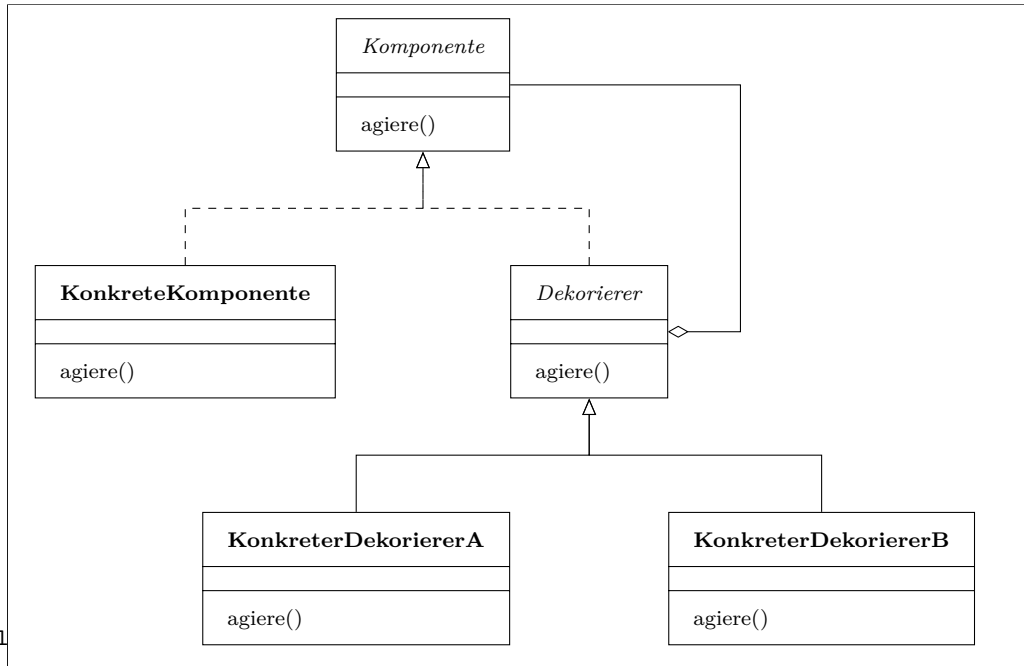
\liEntwurfsBeobachter

```

783 \def\liEntwurfsBeobachter{
784 \liEntwurfsBeobachterUml
785 \liEntwurfsBeobachterAkteure
786 \liEntwurfsBeobachterCode
787 }

```

2.10.6 Dekorierer (Decorator)



\liEntwurfsDekoriererUml

```

788 \def\liEntwurfsDekoriererUml{
789 \begin{tikzpicture}
790 \umlclass[type=abstract]{Komponente}{-}{agiere()}
791 \umlclass[below left=1.5cm and 0cm of Komponente]{KonkreteKomponente}{-}{agiere()}
792 \umlclass[below right=1.5cm and 0cm of Komponente,type=abstract]{Dekorierer}{-}{agiere()}
793
794 \umlVHVreal{KonkreteKomponente}{Komponente}
795 \umlVHVreal{Dekorierer}{Komponente}
796
797 \umlclass[below left=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererA}{-}{agiere()}
798 \umlclass[below right=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererB}{-}{agiere()}
799
800 \umlVHVinherit{KonkreterDekoriererA}{Dekorierer}
801 \umlVHVinherit{KonkreterDekoriererB}{Dekorierer}
802
803 \umlHVVHaggreg[arm1=2cm]{Dekorierer}{Komponente}
804 \footcite{wiki:dekorierer}
805 \end{tikzpicture}
806 }

```

\liEntwurfsDekoriererCode

```

807 \def\liEntwurfsDekoriererCode{
808 \li@EntwurfsCode{dekorierer}{Komponente}
809 \li@EntwurfsCode{dekorierer}{KonkreteKomponente}
810 \li@EntwurfsCode{dekorierer}{Dekorierer}
811 \li@EntwurfsCode{dekorierer}{KonkreterDekoriererA}
812 \li@EntwurfsCode{dekorierer}{KonkreterDekoriererB}
813 \li@EntwurfsCode{dekorierer}{Klient}
814 }

```

\liEntwurfsDekorierer

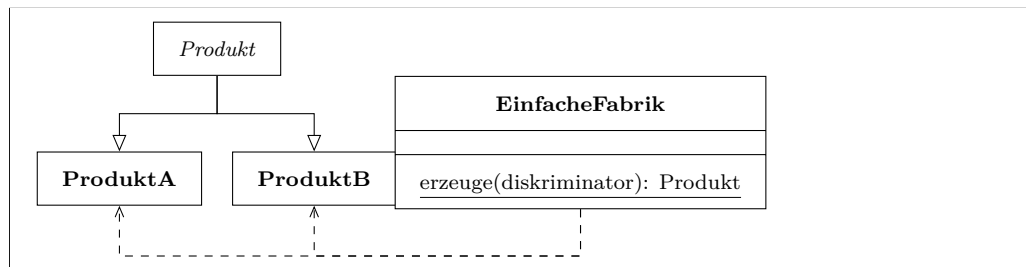
```

815 \def\liEntwurfsDekorierer{
816   \liEntwurfsDekoriererUml
817   \liEntwurfsDekoriererAkteure
818   \liEntwurfsDekoriererCode
819 }

```

2.10.7 Einfache Fabrik (Simple Factory)

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



```

820 \def\liEntwurfsEinfacheFabrikUml{
821   \begin{tikzpicture}
822     \umlsimpleclass[type=abstract]{Produkt}
823     \umlsimpleclass[below left=1cm and -0.65cm of Produkt]{ProduktA}
824     \umlsimpleclass[below right=1cm and -0.65cm of Produkt]{ProduktB}
825     \umlVHVinherit{Produkt}{ProduktA}
826     \umlVHVinherit{Produkt}{ProduktB}
827     \umlclass[below right=0cm and 1.5cm of Produkt]{EinfacheFabrik}{
828     }{
829       \umlstatic{erzeuge(diskriminator): Produkt}\\
830     }
831     \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktA}
832     \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktB}
833   \end{tikzpicture}
834 }

```

\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.

```

835 \def\liEntwurfsEinfacheFabrikAkteure{
836   \begin{description}
837     \item[EinfacheFabrik]
838
839     Eine Klasse mit einer Erzeugungsmethode, die über eine größere
840     Bedingung verschiedene Objekt instanziiert.
841
842     \item[Produkt]
843
844     Eine abstrakte Klasse, die von den konkreten Produkten geerbt wird.
845
846     \item[KonkretesProdukt]
847
848     Ein konkretes Produkt, das von der einfachen Fabrik erzeugt wird.
849   \end{description}
850 }

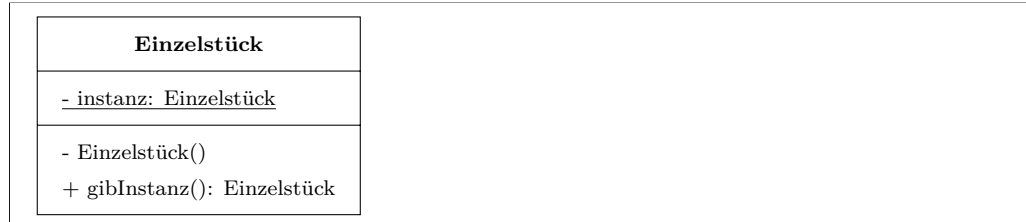
```

\liEntwurfsEinfacheFabrik

```
851 \def\liEntwurfsEinfacheFabrik{
852   \liEntwurfsEinfacheFabrikUml
853   \liEntwurfsEinfacheFabrikAkteure
854 }
```

2.10.8 Einzelstück (Singleton)

\liEntwurfsEinzelstueckUml Quelle nach der deutschen Wikipedia



```
855 \def\liEntwurfsEinzelstueckUml{
856   \begin{tikzpicture}
857     \umlclass{Einzelstück}{
858       \umlstatic{- instanz: Einzelstück}\\
859     }{
860       - Einzelstück()\\
861       + gibInstanz(): Einzelstück
862     }
863   \end{tikzpicture}
864 }
```

iEntwurfsEinzelstueckAkteure Quelle: frei überstetzt aus GoF

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

```
865 \def\liEntwurfsEinzelstueckAkteure{
866   \begin{description}
867     \item[Einzelstück (Singleton)]
868
869     stellt eine statische Methode bereit, mit deren Hilfe die Klienten
870     nur auf eine einzige Instanz der Klasse zugreifen können.
871   \end{description}
872 }
```

\liEntwurfsEinzelstueckCode

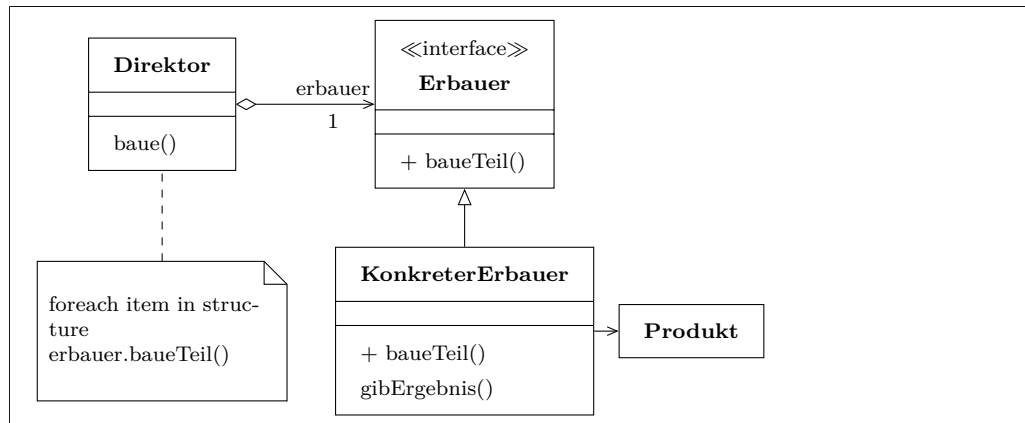
```
873 \def\liEntwurfsEinzelstueckCode{
874   \li@EntwurfsCode{einzelstueck}{Einzelstueck}
875 }
```

\liEntwurfsEinzelstueck

```
876 \def\liEntwurfsEinzelstueck{
877   \liEntwurfsEinzelstueckUml
878   \liEntwurfsEinzelstueckAkteure
879   \liEntwurfsEinzelstueckCode
880 }
```

2.10.9 Erbauer (Builder)

\liEntwurfsErbauerUml Quelle nach der deutschen Wikipedia



```

881 \def\liEntwurfsErbauerUml{
882   \begin{tikzpicture}
883     \umlsimpleclass[x=7,y=0]{Produkt}{}{}
884     \umlclass[x=4,y=3,type=interface]{Erbauer}{}{+ baueTeil()}
885     \umlclass[x=4,y=0]{KonkreterErbauer}{}{
886       + baueTeil()\
887       gibErgebnis()}
888     \umlclass[x=0,y=3]{Direktor}{}{baue()}
889
890     \umluniagg[arg2=erbauer,pos2=0.7,mult2=1]{Direktor}{Erbauer}
891     \umluniassoc{KonkreterErbauer}{Produkt}
892     \umlinherit{KonkreterErbauer}{Erbauer}
893
894     \umlnote[x=0,y=0,width=3cm]{Direktor}{
895       foreach item in structure\
896       erbauer.baueTeil()
897     }
898   \end{tikzpicture}
899   \footcite{wiki:erbauer}
900 }

```

\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.

```

901 \def\liEntwurfsErbauerAkteure{
902   \begin{description}
903     \item[Erbauer]
904
905     Der Erbauer spezifiziert eine abstrakte Schnittstelle zur Erzeugung der
906     Teile eines komplexen Objektes.
907
908     \item[KonkreterErbauer]
909
910     Der konkrete Erbauer erzeugt die Teile des komplexen Objekts durch
911     Implementierung der Schnittstelle. Außerdem definiert und verwaltet er

```

```

912     die von ihm erzeugte Repräsentation des Produkts. Er bietet auch eine
913     Schnittstelle zum Auslesen des Produkts.
914
915     \item[Direktor]
916
917     Der Direktor konstruiert ein komplexes Objekt unter Verwendung der
918     Schnittstelle des Erbauers. Der Direktor arbeitet eng mit dem Erbauer
919     zusammen: Er weiß, welche Baureihenfolge der Erbauer verträgt oder
920     benötigt. Der Direktor entkoppelt somit den Konstruktionsablauf vom
921     Klienten.
922
923     \item[Produkt]
924
925     Das Produkt repräsentiert das zu konstruierende komplexe Objekt.
926     \footcite{wiki:erbauer}
927 \end{description}
928 }

```

\liEntwurfsErbauer

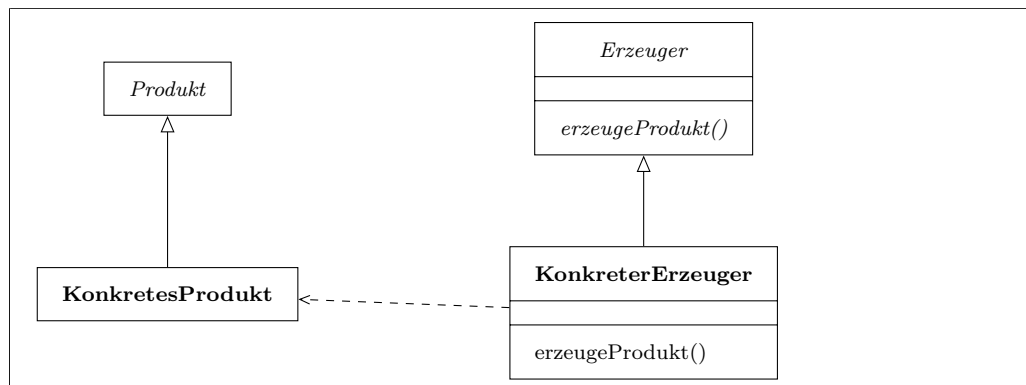
```

929 \def\liEntwurfsErbauer{
930   \liEntwurfsErbauerUml
931   \liEntwurfsErbauerAkteure
932 }

```

2.10.10 Fabrikmethode (Factory Method)

\liEntwurfsFabrikmethodeUml Quelle nach der deutschen Wikipedia



```

933 \def\liEntwurfsFabrikmethodeUml{
934   \begin{tikzpicture}
935     \umlsimpleclass[type=abstract]{Produkt}
936     \umlsimpleclass[below=2cm of Produkt]{KonkretesProdukt}
937     \umlinherit{KonkretesProdukt}{Produkt}
938
939     \umlclass[type=abstract,right=4cm of Produkt]{Erzeuger}{\{
940       \textit{erzeugeProdukt()}\}
941     }
942     \umlclass[below=1.2cm of Erzeuger]{KonkreterErzeuger}{\{
943       erzeugeProdukt()
944     }
945     \umlinherit{KonkreterErzeuger}{Erzeuger}
946
947     \umldep{KonkreterErzeuger}{KonkretesProdukt}
948   \end{tikzpicture}
949 }

```

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).

```

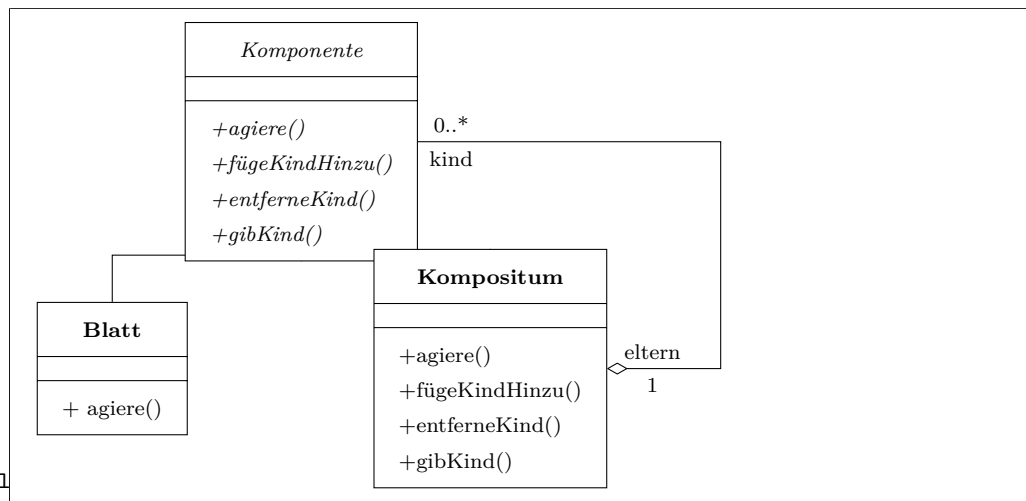
950 \def\liEntwurfsFabrikmethodeAkteure{
951   \begin{description}
952     \item[Produkt]
953
954     Das Produkt ist der Basistyp (Klasse oder Schnittstelle) für das
955     zu erzeugende Produkt.
956
957     \item[KonkretesProdukt]
958
959     KonkretesProdukt implementiert die Produkt-Schnittstelle.
960
961     \item[Erzeuger]
962
963     Der Erzeuger deklariert die Fabrikmethode, um ein solches Produkt
964     zu erzeugen und kann eine Default-Implementierung beinhalten.
965
966     \item[KonkreterErzeuger]
967
968     KonkreterErzeuger überschreibt die Fabrikmethode, um die ihm
969     entsprechenden konkreten Produkte zu erzeugen (z. B. indem er den
970     Konstruktor einer konkreten Produkt-Klasse aufruft).
971
972     \footcite{wiki:fabrikmethode}
973   \end{description}
974 }
```

\liEntwurfsFabrikmethode

```

975 \def\liEntwurfsFabrikmethode{
976   \liEntwurfsFabrikmethodeUml
977   \liEntwurfsFabrikmethodeAkteure
978 }
```

2.10.11 Kompositum (Composite)



\liEntwurfsKompositumUml

```

979 \def\liEntwurfsKompositumUml{
980   \begin{tikzpicture}
981     \umlclass[x=2.5,y=3,type=abstract]{Komponente}{-}{
982       \textit{+agiere()}\
983       \textit{+fügeKindHinzu()}\
984       \textit{+entferneKind()}\
985       \textit{+gibKind()}
986     }
987     \umlclass[x=0]{Blatt}{-}{+ agiere()}
988     \umlclass[x=5]{Kompositum}{-}{
989       +agiere()\
990       +fügeKindHinzu()\
991       +entferneKind()\
992       +gibKind()
993     }
994
995     \umlVHVinherit{Kompositum}{Komponente}
996     \umlVHVinherit{Blatt}{Komponente}
997     \umlHVVhaggreg[anchor1=east,arm1=1.5cm,arg1=eltern,mult1=1,arg2=kind,mult2=0..*,pos2=2.9,
998   \end{tikzpicture}
999 }

```

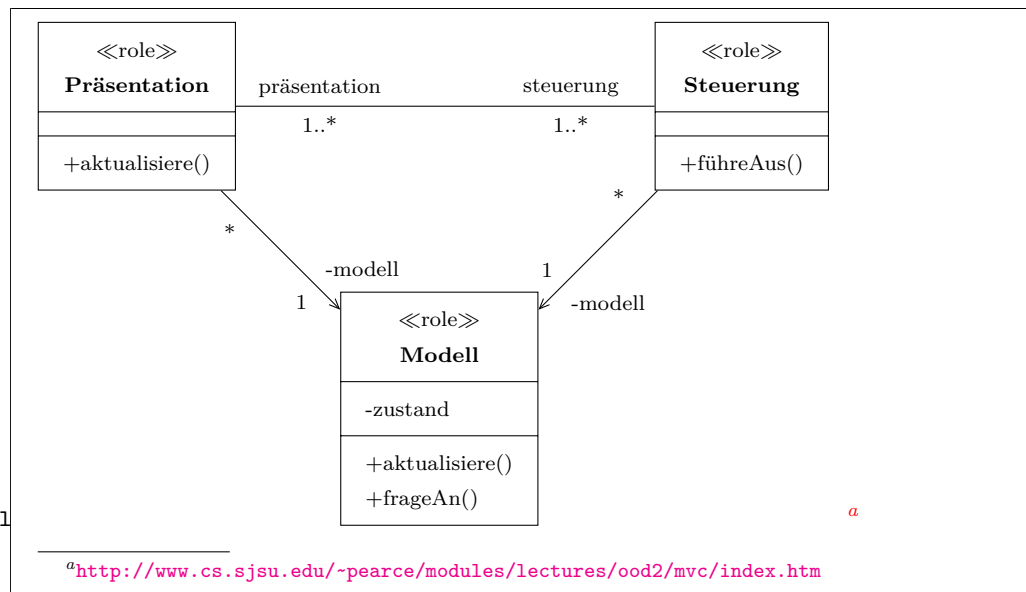
\liEntwurfsFabrikmethode

```

1000 \def\liEntwurfsKompositum{
1001   \liEntwurfsKompositumUml
1002   \liEntwurfsKompositumAkteure
1003 }

```

2.10.12 Modell-Präsentation-Steuerung (Model-View-Controller)



```

1004 \def\liEntwurfsModellPraesentationSteuerungUml{
1005   \begin{tikzpicture}
1006     \umlclass[x=-4cm,type=role]{Präsentation}{-}{+aktualisiere()}
1007     \umlclass[x=4cm,type=role]{Steuerung}{-}{+führeAus()}
1008     \umlclass[y=-4cm,type=role]{Modell}{-}{
1009       -zustand
1010     }{
1011       +aktualisiere()\
1012       +frageAn()
1013     }
1014
1015     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Präsentation}{Modell}
1016     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Steuerung}{Modell}

```

```

1017 \umlassoc[arg1=präsentation,mult1=1..*,arg2=steuerung,mult2=1..*]{Präsentation}{Steuerung}
1018 \end{tikzpicture}
1019 \liFussnoteUrl{http://www.cs.sjsu.edu/~pearce/modules/lectures/ood2/mvc/index.htm}
1020 }

```

ModellPraesentationSteuerung

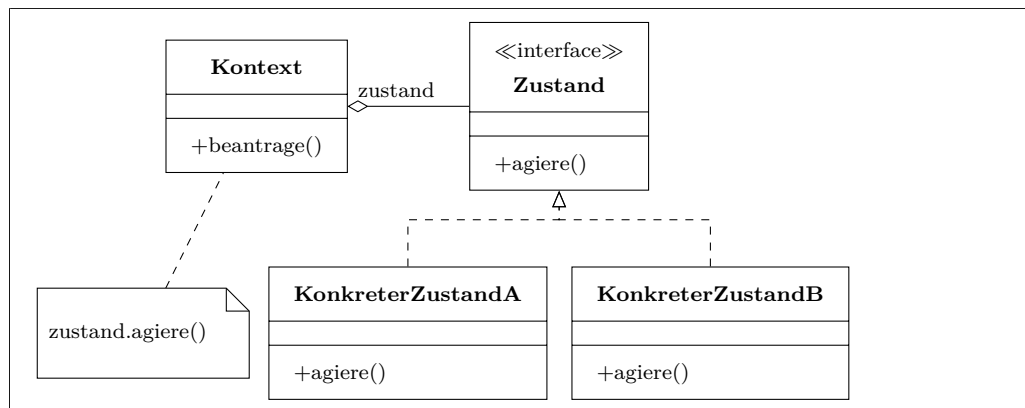
```

1021 \def\liEntwurfs{
1022 \liEntwurfsModellPraesentationSteuerungUml
1023 \liEntwurfsModellPraesentationSteuerungAkteure
1024 }

```

2.10.13 Zustand (State)

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



```

1025 \def\liEntwurfsZustandUml{
1026 \begin{tikzpicture}
1027 \umlcclass[x=-1,y=3]{Kontext}{+beantrage()}
1028 \umlcclass[x=3,y=3,type=interface]{Zustand}{+agiere()}
1029 \umlcclass[x=1,y=0]{KonkreterZustandA}{+agiere()}
1030 \umlcclass[x=5,y=0]{KonkreterZustandB}{+agiere()}
1031
1032 \umlVHVreal{KonkreterZustandA}{Zustand}
1033 \umlVHVreal{KonkreterZustandB}{Zustand}
1034
1035 \umlagg[reg[arg=zustand,pos=0.4]{Kontext}{Zustand}
1036
1037 \umlnote[x=-2.5,y=0,width=2.5cm]{Kontext}{zustand.agiere()}
1038 \end{tikzpicture}
1039 }

```

\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.

```

1040 \def\liEntwurfsZustandAkteure{
1041 \begin{description}
1042 \item[Kontext (Context)]
1043
1044 definiert die clientseitige Schnittstelle und verwaltet die separaten
1045 Zustandsklassen.
1046

```

```

1047 \item[State (Zustand)]
1048
1049 definiert eine einheitliche Schnittstelle aller Zustandsobjekte und
1050 implementiert gegebenenfalls ein Standardverhalten.
1051
1052 \item[KontreterZustand (ConcreteState)]
1053
1054 implementiert das Verhalten, das mit dem Zustand des Kontextobjektes
1055 verbunden ist.
1056 \end{description}
1057 }

\liEntwurfsZustand
1058 \def\liEntwurfsZustand{
1059 \liEntwurfsZustandUml
1060 \liEntwurfsZustandAkteure
1061 }

1062

```

2.11 er.sty

```
1063 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1064 \ProvidesPackage{lehramt-informatik-er}[2020/06/13 Zum Setzen von
1065 ER-Diagrammen]

1066 \RequirePackage{tikz-er2}
1067 \usetikzlibrary{positioning}

\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}

1068 \RequirePackage{soul}
1069 \RequirePackage{fontawesome}

Let-Abkürzungen
\let\ a=\liErMpAttribute
\let\ d=\liErDatenbankName
\let\ e=\liErMpEntity
\let\ r=\liErMpRelationship

1070 \ExplSyntaxOn

\liErEntity
1071 \def\liErEntity#1{\textbf{#1}}

\liErRelationship
1072 \def\liErRelationship#1{\ul{#1}}

\liErAttribute
1073 \def\liErAttribute#1{\emph{#1}}

\liErMpEntity mp = marginpar
Let-Abkürzung: \let\ e=\liErMpEntity
1074 \def\liErMpEntity#1{
1075   \liErEntity{#1}
1076   \marginpar{
1077     \liErEntity{\tiny\faSquareO{}}~E:~#1}
1078   }
1079 }
□

\liErMpRelationship Let-Abkürzung: \let\ r=\liErMpRelationship
1080 \def\liErMpRelationship#1{
1081   \liErRelationship{#1}
1082   \marginpar{
1083     \liErRelationship{\tiny\faGg{}}~R:~#1}
1084   }
1085 }

\liErMpAttribute Let-Abkürzung: \let\ a=\liErMpAttribute
1086 \def\liErMpAttribute#1{
1087   \liErAttribute{#1}
1088   \marginpar{
1089     \liErAttribute{\tiny\faCircleThin{}}~A:~#1}
1090   }
1091 }

\liErDatenbankName Let-Abkürzung: \let\ d=\liErDatenbankName
datenbank name
1092 \def\liErDatenbankName#1{
1093   {
1094     \footnotesize\texttt{(#1)}
1095   }
1096 }

1097 \ExplSyntaxOff
1098

```

2.12 formale-sprachen.sty

```

1099 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1100 \ProvidesPackage{lehramt-informatik-formale-sprachen}[2021/02/21 Hilfsmakros
1101 zum Setzen von mathematischen Formeln bei dem Thema Formale Sprachen]

1102 \directlua{
1103   formale_sprachen = require('lehramt-informatik-formale-sprachen')
1104 }

1105 \RequirePackage{hyperref}

1106 \liLadePakete{mathe,typographie}

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

1107 \def\liMengeOhneMathe#1{\{ #1 \}}
1108 \def\liMenge#1{%
1109   \ifmmode%
1110     \liMengeOhneMathe{#1}%
1111   \else%
1112     $\liMengeOhneMathe{#1}$%
1113   \fi%
1114 }

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

1115 \def\liEpsilon{$\varepsilon$}

\liPotenzmenge Umgeben mit geschweiften Klammern in einer Mathematik-Umgebung

1116 \def\erzeuge@tiefgestellt#1{\directlua{formale_sprachen.erzeuge_tiefgestellt('#1')}}
1117 \def\liPotenzmengeOhneMathe#1{\{ \erzeuge@tiefgestellt{#1} \}}
1118 \def\liPotenzmenge#1{$\liPotenzmengeOhneMathe{#1}$}

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

1119 \let\liZustandsmengeOhneMathe=\liPotenzmengeOhneMathe
1120 \let\liZustandsmenge=\liPotenzmenge

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

1121 \def\liUeberfuehrungsFunktionOhneMathe#1{\delta(\erzeuge@tiefgestellt{#1})}
1122 \def\liUeberfuehrungsFunktion#1{
1123   \ifmmode
1124     \liUeberfuehrungsFunktionOhneMathe{#1}
1125   \else
1126     $\liUeberfuehrungsFunktionOhneMathe{#1}$
1127   \fi
1128 }

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

1129 \def\liAlphabet#1{$\Sigma = \{ #1 \}$}

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

1130 \def\liBandAlphabet#1{$\Gamma = \Sigma \cup \{ #1 \}$}

\liZustandsBuchstabe

1131 \def\liZustandsBuchstabe{z}

\liZustandsBuchstabeGross

1132 \def\liZustandsBuchstabeGross{Z}

```

```

\liZustandsmengeNr
1133 \def\zustandsnamens@liste#1#2{
1134 $
1135 \{
1136 \directlua{formale_sprachen.erzeuge_zustandsnamens_liste('#1', '#2')}
1137 \}
1138 $
1139 }
1140 \def\liZustandsmengeNr#1{\zustandsnamens@liste{\liZustandsBuchstabe}{#1}}

\liZustandsmengeNrGross
1141 \def\liZustandsmengeNrGross#1{\zustandsnamens@liste{\liZustandsBuchstabeGross}{#1}}

\liZustandsname \liZustandsname{1}: $z_1$
1142 \def\liZustandsname#1{\liZustandsBuchstabe_#1$}

\liZustandsnameGross \liZustandsnameGross{1}: $Z_1$
1143 \def\liZustandsnameGross#1{\liZustandsBuchstabeGross_#1$}

\liAbleitung \liAbleitung{S -> aB -> ab}:  $S \vdash aB \vdash ab$ 
1144 \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}

1145 \NewDocumentEnvironment { liProduktionsRegeln }
1146 { 0{P} +b }
1147 {
1148 \liGeschweifteKlammern{#1}
1149 {
1150 \begin{align*}
1151 \directlua{formale_sprachen.produktions_regeln('#2')}
1152 \end{align*}
1153 }{-0.2cm}{-1.5cm}
1154 } {}

\liProduktionen \liProduktionen{S -> A, A -> a}:  $\{S \rightarrow A, A \rightarrow a\}$ 
1155 \def\liProduktionen#1{
1156 \liMenge{\directlua{formale_sprachen.produktions_regeln('#1', true)}}
1157 }

\liZustandsnameTiefgestelltt Automatisch tiefgestellte Nummerierung \z1
Let-Abkürzung: \let\z=\liZustandsnameTiefgestelltt
1158 \def\liZustandsnameTiefgestelltt#1{
1159 \ifmmode
1160 \liZustandsBuchstabe\sb{#1}
1161 \else
1162 $\liZustandsBuchstabe\sb{#1}$
1163 \fi
1164 }

1165 \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}

```



```

1166 \NewDocumentCommand{ \liAusdruck } { O{L} m m } {
1167   $
1168   \tl_if_empty:nTF {#1} {} {#1 =}
1169   \{
1170     \, #2 \,
1171     |
1172     \, #3 \,
1173   \}$
1174 }
1175 \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

1176 \def\liFlaci#1{%
1177   \par
1178   {%
1179     \scriptsize
1180     Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte
1181     Automaten, Compiler und Interpreter) Ein Projekt der Hochschule
1182     Zittau/Görlitz und der Pädagogischen Hochschule Schwyz:
1183     \href{https://flaci.com/#1}{flaci.com/#1}%
1184   }%
1185   \par
1186 }

\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}:  $G = (V, \Sigma, P, X)$ 

1187 \ExplSyntaxOn
1188 \NewDocumentCommand {\liGrammatik} { O{G} m } {
1189   \tl_set:Nn \l_variablen_tl {V}
1190   \tl_set:Nn \l_alphabet_tl {\Sigma}
1191   \tl_set:Nn \l_produktionen_tl {P}
1192   \tl_set:Nn \l_start_tl {S}
1193
1194   \keys_define:nn { grammatik } {
1195     variablen .code:n = {\tl_set:Nn \l_variablen_tl {\liMenge{##1}}},
1196     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
1197     produktionen .code:n = {\tl_set:Nn \l_produktionen_tl {\liProduktionen{##1}}},
1198     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
1199   }
1200
1201   \keys_set:nn { grammatik } { #2 }
1202
1203   $#1 = (
1204     \l_variablen_tl,
1205     \l_alphabet_tl,
1206     \l_produktionen_tl,
1207     \l_start_tl
1208   )$
1209 }
1210 \ExplSyntaxOff
1211

```

2.13 formatierung.sty

```
1212 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1213 \ProvidesPackage{lehramt-informatik-formatierung}[2020/11/27]
```

2.13.1 Schriftarten / Typographie

```
1214 \RequirePackage{mathpazo}
1215 \RequirePackage[no-math]{fontspec}
1216 \setmainfont{texgyrepagella}
1217 \setsansfont{QTAncientOlive}
1218 \RequirePackage{sectsty}
1219 \allsectionsfont{\fontspec{QTAncientOlive}}
```

2.13.2 Farben

```
1220 \RequirePackage{xcolor}
1221 \definecolor{infogray}{rgb}{0.97,0.97,0.97}
```

2.13.3 Überschriften

```
1222 \RequirePackage{titlesec}
1223 \titleformat{\chapter}[display]{\sffamily\bfseries}{\Opt}{\LARGE}
1224 \titlespacing{\chapter}{0pt}{0pt}{*1}
1225 \titleformat{\paragraph}[hang]{\sffamily\normalsize\bfseries}{\theparagraph}{1em}{}
1226 \setcounter{secnumdepth}{0}
```

2.13.4 Listen

```
1227 \RequirePackage{paralist}
1228 \renewcommand\labelitemi{-}
1229 \renewcommand\labelitemii{-}
1230 \renewcommand\labelitemiii{-}
1231 \renewcommand\labelitemiv{-}
1232 % Counter: enumi enumii enumiii enumiv
1233 % Styles: \arabic{counter} \alph{counter} \Alph{counter} \roman{counter} \Roman{counter}
1234 \renewcommand{\labelenumi}{(\alph{enumi})}
1235 \renewcommand{\labelenumii}{(\roman{enumii})}
```

2.13.5 Kasten

```
1236 \RequirePackage{mdframed}
1237 \mdfsetup{backgroundcolor=infogray}
```

liKasten

```
1238 \NewDocumentEnvironment { liKasten }{ } {
1239   \begin{mdframed}[backgroundcolor=white!0]
1240 } {
1241   \end{mdframed}
1242 }
```

2.13.6 Header

```
1243 \RequirePackage{fancyhdr}
1244 \fancyhead[L,C,R]{}
1245 \fancyfoot[L]{}
1246 \fancyfoot[C]{}
1247 \fancyfoot[R]{\thepage}
1248 \pagestyle{fancy}
1249 \renewcommand{\headrulewidth}{0pt}
1250 \renewcommand{\footrulewidth}{0pt}
1251
```

2.14 gantt.sty

```

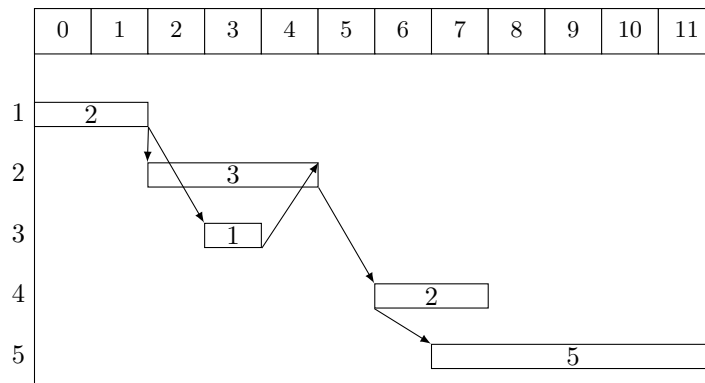
1252 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1253 \ProvidesPackage{lehramt-informatik-gantt}[2020/09/05]

\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}

```



```

1254 \RequirePackage{tikz-uml}
1255 \RequirePackage{pgfgantt}
1256 \setganttlinklabel{f-s}{}
1257 \setganttlinklabel{s-s}{}
1258 \setganttlinklabel{f-f}{}
1259 \setganttlinklabel{s-f}{}

1260

```

2.15 grafik.sty

```
1261 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1262 \ProvidesPackage{lehramt-informatik-grafik}[2020/12/27 Lädt das Paket
1263 TikZ, um Grafiken zeichnen zu können.]
1264 \RequirePackage{tikz}
1265
```

2.16 graph.sty

```

1266 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1267 \ProvidesPackage{lehramt-informatik-graph}[2020/06/09]
1268 \RequirePackage{tikz}

```

Für die die Adjazenzliste (`xrightharrow`)

```

1269 \RequirePackage{amsmath}

```

Für Adjazenz-Matrix

```

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

```

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

```

1270 \RequirePackage{blkarray}

```

```

1271 \usetikzlibrary{arrows.meta}

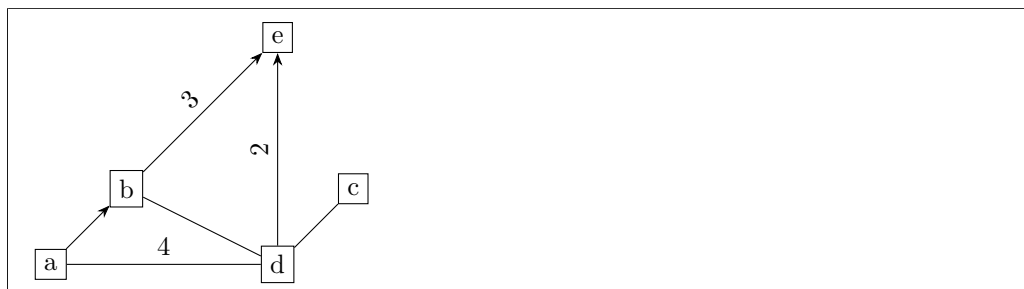
```

```

\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}

```



```

1272 \tikzset{
1273   li graph/.style={
1274     every node/.style={
1275       rectangle,
1276       draw,
1277     },
1278     every edge/.style={
1279       >={Stealth[black]},
1280       draw,
1281     },
1282     every edge/.append style={
1283       every node/.style={
1284         sloped,
1285         auto,
1286       }
1287     }
1288   },
1289   li markierung/.style={
1290     ultra thick,
1291   }
1292 }

```

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}

```

```

1293 \NewDocumentEnvironment { liGraphenFormat }{ +b } {} {}

```

```

1294

```

2.17 hanoi.sty

1295 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 1296 \ProvidesPackage{lehramt-informatik-hanoi}[2020/12/19 Paket zum Setzen
 1297 von Hanoi-Grafiken]

Quelle: <https://kleinco.de/latex-snippets/7/tikz-towers-of-hanoi-illustration-for-lat>

1298 \RequirePackage{tikz}
 1299 \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}

1300 \def\li@mset #1[#2]=#3{%
1301 \expandafter\xdef\csname #1#2\endcsname{#3}
1302 }
1303 \def\li@mget #1[#2]{%
1304 \csname #1#2\endcsname
1305 }
1306 \def\li@minc #1[#2] += #3{%
1307 \pgfmathparse{\li@mget #1[#2] + #3}%
1308 \li@mset #1[#2] = \pgfmathresult
1309 }
1310
1311 \def\liHanoi#1#2{
1312   \edef\li@numdiscs{#1}
1313   \def\li@sequence{#2}
1314   \begin{tikzpicture}[line width=4mm,brown!40,line cap=round,xscale=3]
1315     % init colors
1316     \foreach[count=\j] \c in {red,green!80!black,blue,orange,violet,gray,yellow!80!black,purp
1317     \li@mset col[\j]={\c};
1318     % draw poles and init pole counters
1319     \foreach \j in {1,2,3}{
1320       \li@mset pos[\j]=0
1321       \draw (\j,-.5) -- +(0, .5 + .5*\li@numdiscs);
1322     }
1323     % draw base
1324     \draw (.5,-.5) -- +(3,0);
1325     % draw discs
1326     \foreach[count=\k] \i/\j in \li@sequence{
1327       \draw[draw={\li@mget col[\i]}] (\j,\li@mget pos[\j]) +(-.4*\i/\li@numdiscs,0) -- +(.4*
1328       \li@minc pos[\j] += {.5}
1329     }
1330   \end{tikzpicture}
1331 }

1332
```

2.18 komplexitaetstheorie.sty

```

1333 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1334 \ProvidesPackage{lehramt-informatik-komplexitaetstheorie}[2021/07/08 Zum
1335 Setzen von Karp's NP-vollständigen Problemen, Hilfsmakros für die
1336 Polynomialzeitreduktion.]

```

Let-Abkürzungen

```

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

```

```

1337 \liLadePakete{mathe}

```

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

```

1338 \RequirePackage{mdframed}

```

```

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

```

```

1339 \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

```

```

1340 \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`

```

1341 \def\liProblemBeschreibung#1#2#3{
1342   \begin{mdframed}[
1343     userdefinedwidth=9cm,
1344     align=center,
1345     backgroundcolor=white!0,
1346   ]
1347   \centerline{\large\liProblemName{#1}}
1348
1349   \medskip
1350
1351   \begin{description}
1352     \item[Gegeben:] #2
1353     \item[Frage:] #3
1354   \end{description}
1355   \end{mdframed}
1356 }

```



```

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

1357 \NewDocumentCommand{ \liPolynomiellReduzierbar } { m O{p} m } {
1358 \begin{displaymath}
1359 \liProblemName{#1}
1360 \preceq_{#2}
1361 \liProblemName{#3}
1362 \end{displaymath}
1363 }

\liProblemVertexCover

1364 \def\liProblemClique{%
1365 Das \textbf{Cliquesproblem} fragt nach der Existenz einer Clique der
1366 Mindestgröße  $n$  in einem gegebenen Graphen.
1367 \footcite[Seite 76]{theo:fs:4}
1368 Eine Clique ist eine Teilmenge von Knoten in einem ungerichteten
1369 Graphen, bei der \emph{jedes Knotenpaar durch eine Kante} verbunden ist.
1370 \footcite{wiki:cliquenproblem}
1371 }

\liProblemVertexCover

1372 \def\liProblemVertexCover{%
1373 %
1374 Das \textbf{Knotenüberdeckungsproblem} (\liProblemName{Vertex Cover})
1375 fragt, ob zu einem gegebenen einfachen Graphen und einer natürlichen
1376 Zahl  $k$  eine Knotenüberdeckung der Größe von höchstens  $k$  existiert.
1377 \footcite{wiki:knotenueberdeckung}
1378
1379 Das heißt, ob es eine aus maximal  $k$  Knoten bestehende Teilmenge  $U$ 
1380 der Knotenmenge gibt, so dass jede Kante des Graphen mit
1381 mindestens einem Knoten aus  $U$  verbunden ist.
1382 \footcite[Seite 78]{theo:fs:4}%
1383 }

\liProblemSubsetSum Kein Karp-21-Problem

1384 \def\liProblemSubsetSum{%
1385 Das \textbf{Teilsommenproblem} (\liProblemName{Subset Sum} oder
1386 \liProblemName{SSP}) ist ein spezielles Rucksackproblem.
1387 \footcite{wiki:teilsommenproblem}
1388 Gegeben sei eine Menge von ganzen Zahlen  $I = \{w_1, w_2, \dots, w_n\}$ .
1389 Gesucht ist eine Untermenge, deren Elementsumme maximal, aber nicht
1390 größer als eine gegebene obere Schranke  $c$  ist.
1391 \footcite[Seite 74]{theo:fs:4}
1392 }

\liProblemSubsetSum Kein Karp-21-Problem

1393 \def\liProblemSat{%
1394 Das \textbf{Erfüllbarkeitsproblem der Aussagenlogik} \liProblemName{Sat}
1395 und \liProblemName{k-SAT} mit  $k \geq 3$ ,  $k \in \mathbb{N}$  (Satz von
1396 Cook) fragt, ob eine aussagenlogische Formel erfüllbar
1397 ist.\footcite{wiki:sat} Das Erfüllbarkeitsproblem der
1398 \emph{Aussagenlogik} ist in exponentieller Zeit in Abhängigkeit der
1399 Anzahl der Variablen mit Hilfe einer Wahrheitstabelle entscheidbar.
1400 Diese \emph{Wahrheitstabelle} kann nicht in polynomieller Zeit
1401 aufgestellt werden.
1402 \footcite[Seite 71]{theo:fs:4}
1403 }

1404

```

2.19 kontrollflussgraph.sty

```
1405 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1406 \ProvidesPackage{lehramt-informatik-kontrollflussgraph}[2020/11/07]

1407 \RequirePackage{tikz}
1408 \usetikzlibrary{positioning}
1409 \tikzset{
1410   li kontrollfluss/.style={
1411     knoten/.style={
1412       circle,
1413       draw
1414     },
1415     usebox/.style={
1416       draw,
1417       rectangle,
1418       font=\scriptsize,
1419       anchor=west,
1420       align=left,
1421     },
1422     bedingung/.style={
1423       midway,
1424       draw=none,
1425       font=\scriptsize
1426     },
1427     knotenbeschriftung/.style={
1428       draw,
1429       rectangle,
1430       midway,
1431       font=\scriptsize
1432     },
1433     wahr/.style={
1434       thick
1435     },
1436     falsch/.style={
1437       dashed
1438     },
1439     every node/.style={
1440       circle,
1441       draw,
1442     },
1443     every edge/.append style={
1444       every node/.style={
1445         draw=none,
1446         bedingung,
1447       }
1448     },
1449     every path/.style={
1450       draw,
1451       ->,
1452     },
1453     every pin/.style={
1454       draw,
1455       dotted,
1456       rectangle,
1457       pin position=right
1458     },
1459     every pin edge/.style={
1460       dotted,
1461       arrows=-,
1462     }
1463   }
1464 }
```

liKontrollflussgraph

```

1465 \NewDocumentEnvironment { liKontrollflussgraph } { 0{ } } {
1466   \begin{tikzpicture}[
1467     li kontrollfluss,
1468     #1
1469   ]
1470 } {
1471   \end{tikzpicture}
1472 }

\liAnweisung
1473 \def\liAnweisung#1(#2,#3){\node[knoten] at (#2,#3) (#1) {#1};}

\liBedingung Let-Abkürzung: \let\b=\liBedingung
1474 \def\liBedingung#1#2{node[bedingung,#1]{\texttt{#2}}}}

\liBedingungWahr Let-Abkürzung: \let\bWahr=\liBedingungWahr
1475 \def\liBedingungWahr#1{node[bedingung,#1]{\texttt{[true]}}}}

\liBedingungFalsch Let-Abkürzung: \let\bFalsch=\liBedingungFalsch
1476 \def\liBedingungFalsch#1{node[bedingung,#1]{\texttt{[false]}}}}

\liKontrollCode Let-Abkürzung: \let\c=\liKontrollCode
1477 \def\liKontrollCode#1{{\tiny\texttt{\textcolor{gray}{#1}}}}

\liKontrollTextzeileKnoten Let-Abkürzung: \let\k=\liKontrollTextzeileKnoten
1478 \def\liKontrollTextzeileKnoten#1{\raisebox{-2pt}{\tikz[scale=0.5,transform shape] \node[draw

\liKontrollKnotenPfad Let-Abkürzung: \let\p=\liKontrollKnotenPfad
1479 \ExplSyntaxOn
1480 \NewDocumentCommand { \liKontrollKnotenPfad }{ m }
1481 {
1482   \seq_set_split:Nnn \l_tmpa_seq { - } { #1 }
1483   \seq_set_map:Nnn \l_tmpa_seq \l_tmpa_seq {\liKontrollTextzeileKnoten{##1}}
1484   \seq_use:Nn \l_tmpa_seq { ~~~~ }
1485 }
1486 \ExplSyntaxOff

1487

```

2.20 kopf-fusszeilen.sty

```
1488 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1489 \ProvidesPackage{lehramt-informatik-kopf-fusszeilen}[2021/08/20 Kopf-
1490 und Fußzeilen mit Hilfe des Pakets fancyhdr.]

1491 \ExplSyntaxOn

1492 \fancyhead{}
1493 \fancyhead[R0,LE]{\scriptsize\LehramtInformatikTitel}
1494 \fancyhead[L0,RE]{\scriptsize\today}
1495 \fancyfoot{}
1496 \fancyfoot[LE,R0]{\thepage}
1497 \fancyfoot[L0,CE]{\scriptsize\LehramtInformatikAutorName}
1498 \fancyfoot[CO,RE]{\scriptsize\LehramtInformatikAutorEmail}
1499 \renewcommand{\headrulewidth}{0.4pt}
1500 \renewcommand{\footrulewidth}{0.4pt}

1501 \ExplSyntaxOff

1502
```

2.21 literatur-dummy.sty

```
1503 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1504 \ProvidesPackage{lehramt-informatik-literatur-dummy}[2020/11/27]

\literatur

1505 \def\literatur{}

\footcite

1506 % \RequirePackage[stable,multiple]{footmisc}
1507 \NewDocumentCommand{ \footcite } { o m }{}

1508
```

2.22 literatur.sty

```
1509 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1510 \ProvidesPackage{lehramt-informatik-literatur}[2020/11/27]

1511 \RequirePackage{csquotes}
1512 \RequirePackage[
1513   bibencoding=utf8,
1514   citestyle=authortitle,
1515   backend=biber,
1516 ]{biblatex}
1517 \addbibresource{\LehramtInformatikRepository/Quellen/10_DB.bib}
1518 \addbibresource{\LehramtInformatikRepository/Quellen/20_OOMUP.bib}
1519 \addbibresource{\LehramtInformatikRepository/Quellen/30_AUD.bib}
1520 \addbibresource{\LehramtInformatikRepository/Quellen/40_SOSY.bib}
1521 \addbibresource{\LehramtInformatikRepository/Quellen/60_FUMUP.bib}
1522 \addbibresource{\LehramtInformatikRepository/Quellen/70_THEO.bib}
1523 \addbibresource{\LehramtInformatikRepository/Quellen/80_DDI.bib}
1524 \addbibresource{\LehramtInformatikRepository/Quellen/Allgemein.bib}
1525 \addbibresource{\LehramtInformatikRepository/Quellen/Examen.bib}
1526 \addbibresource{\LehramtInformatikRepository/Quellen/Wikipedia.bib}
1527 % To allow footnotes in the heading
1528 \RequirePackage[stable,multiple]{footmisc}

\literatur

1529 \def\literatur{\printbibliography[heading=subbibliography]}

1530
```

2.23 makros.sty

```
1531 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1532 \ProvidesPackage{lehramt-informatik-makros}[2020/11/27 Eine Sammlung von
1533 häufig verwendeten Makros und Umgebungen, die thematisch zu keinem
1534 anderen Paket passen]
1535 \RequirePackage{hyperref}
1536 \RequirePackage{graphicx}
    Für die Umgebung liQuellen benötigt.
1537 \RequirePackage{paralist}

\inhaltsverzeichnis
1538 \def\inhaltsverzeichnis {
1539   \begin{mdframed}
1540     \begin{group}
1541       \let\clearpage\relax
1542       \tableofcontents
1543     \end{group}
1544   \end{mdframed}
1545 }

\memph \mephm (\marginpar and \emph)
1546 \newcommand{\memph}[1]{\emph{#1}\marginpar{\tiny#1}}

\SLASH
1547 \newcommand\SLASH{\char`\\}

\liPseudoUeberschrift Text, der sich wie eine Überschrift verhält.
1548 \newcommand{\liPseudoUeberschrift}[1]{
1549   \bigskip
1550   \noindent
1551   \textsf{\textbf{#1}}
1552   \noindent
1553 }

\liBeschriftung Ähnlich dem Makro \liPseudoUeberschrift{}. Am Ende des Textes wird ein Doppelpunktzeichen angehängt.
1554 \newcommand{\liBeschriftung}[1]{
1555   \par
1556   \noindent
1557   \medskip
1558   \textbf{#1}:
1559   \medskip
1560   \noindent
1561 }

\hinweis
1562 \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.
1563 \NewDocumentEnvironment { liProjektSprache }{ o +b } {} {}

liEinbettung
1564 \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.
1565 \RequirePackage{xparse}
1566 \ExplSyntaxOn
```

```

1567 \NewDocumentEnvironment{ liAntwort } { 0{standard} }
1568 {
1569   \str_case:nn {#1} {
1570     {standard} {
1571       \def\beschriftung{}
1572       \mdfsetup{backgroundcolor=gray!1,linecolor=gray}
1573     }
1574     {richtig} {
1575       \def\beschriftung{richtig}
1576       \mdfsetup{backgroundcolor=green!5,linecolor=green}
1577     }
1578     {falsch} {
1579       \def\beschriftung{falsch}
1580       \mdfsetup{backgroundcolor=red!3,linecolor=red}
1581     }
1582     {muster} {
1583       \def\beschriftung{Musterlösung}
1584       \mdfsetup{backgroundcolor=green!12,linecolor=green}
1585     }
1586   }
1587   \ifx\beschriftung\empty\else
1588     \noindent
1589     \textbf{\beschriftung{:}}
1590   \fi
1591   \begin{mdframed}
1592 }
1593 {\end{mdframed}}

```

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

```

1594 \NewDocumentEnvironment{ liAdditum } { o }
1595 {
1596   \begin{mdframed}[backgroundcolor=yellow!5]
1597     \IfNoValueTF {#1}
1598     { \liPseudoUeberschrift{Additum} }
1599     { \liPseudoUeberschrift{Additum:~#1} }
1600 }
1601 {\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.

```

1602 \NewDocumentEnvironment{ liExkurs }{o +b}{
1603   \vspace{0.2cm}%
1604   \begin{mdframed}[
1605     backgroundcolor=white,
1606     bottomline=false,
1607     innermargin=1cm,
1608     leftline=true,
1609     linecolor=black,
1610     linewidth=0.1cm,
1611     outermargin=1cm,
1612     rightline=false,
1613     topline=false,
1614   ]

```



```

1615 \footnotesize
1616 \noindent%
1617 \textbf{Exkurs:~#1}\par%
1618 \noindent%
1619 #2
1620 \end{mdframed}
1621 \vspace{0.2cm}
1622 }{}

```

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

```

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

```

Weiterführende Literatur:

- Quelle 1
- Quelle 2

```

1623 \cs_new:Npn \listen@punkt #1 {\item #1}
1624 \NewDocumentEnvironment { liQuellen }{ +b }
1625 {
1626   \seq_clear_new:N \l_quellen
1627   \seq_set_split:Nnn \l_quellen {\item} {#1}
1628   \seq_remove_all:Nn \l_quellen {}
1629   \begin{mdframed}[backgroundcolor=white!0]
1630     \footnotesize
1631     \noindent
1632     \textsf{\textbf{Weiterführende~Literatur:}}
1633     \medskip
1634     \begin{compactitem}
1635       \seq_map_function:NN \l_quellen {\listen@punkt}
1636     \end{compactitem}
1637   \end{mdframed}
1638   %
1639   \makeatletter
1640   \par\@afterindentfalse\@afterheading
1641   \makeatother
1642 } {}

```

liLernkartei

```

1643 \NewDocumentEnvironment { liLernkartei }{ m +b }
1644 {
1645   \begin{mdframed}
1646     \footnotesize
1647     \noindent%
1648     \textbf{Lernkarteikarte:~#1}\par%
1649     \noindent%
1650     #2
1651   \end{mdframed}
1652 } {}

```

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

```

1653 \NewDocumentEnvironment { liDiagramm }{ m +b }
1654 {
1655   \begin{mdframed}[backgroundcolor=white!0]
1656     \small
1657     \noindent%
1658     \textit{#1}:
1659     \begin{center}

```

```

1660 #2
1661 \medskip
1662 \end{center}
1663 \end{mdframed}
1664 } {}
1665 \ExplSyntaxOff

\liFussnoteUrl \liFussnoteUrl[\langle zusätzlicher-text \rangle]{\langle url \rangle} \liFussnoteUrl[zusätzlicher Text]{url}:
Eine HTTP-URL als Fußnote setzen.
1666 \NewDocumentCommand{\liFussnoteUrl} { o m } {
1667 \footnote{\url{#2}\IfNoValueTF{#1}{}{ ( #1 )}}
1668 }
1669

\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.
1670 \NewDocumentCommand{\liFussnoteLink} { o m m } {
1671 \footnote{\href{#3}{#2}\IfNoValueTF{#1}{}{ ( #1 )}}
1672 }

\zB
1673 \def\zB{z.\,B. }

\ZB
1674 \def\ZB{Z.\,B. }

\dh
1675 \def\dh{d.\,h. }

1676

```

2.24 master-theorem.sty

1677 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

1678 \ProvidesPackage{lehramt-informatik-master-theorem}[2021/04/13]

Let-Abkürzungen

\let\0=\li0

\let\o=\li0mega

\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}

1679 \ExplSyntaxOn

1680 \RequirePackage{amsmath}

\liRundeKlammer

1681 \def\liRundeKlammer#1{

1682 \negthinspace \left(#1 \right)

1683 }

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

1684 \def\liThetaOhneMathe#1{

1685 \Theta \liRundeKlammer{#1}

1686 }

1687 \def\liTheta#1{

1688 \ifmmode

1689 \liThetaOhneMathe{#1}

1690 \else

1691 \$\liThetaOhneMathe{#1}\$

1692 \fi

1693 }

```

\liOmega \liOmega{n^2}:  $\Omega(n^2)$ 
1694 \def\liOmegaOhneMathe#1{
1695 \Omega \liRundeKlammer{#1}
1696 }
1697 \def\liOmega#1{
1698 \ifmmode
1699 \liOmegaOhneMathe{#1}
1700 \else
1701 $\liOmegaOhneMathe{#1}$
1702 \fi
1703 }

\liO \liO{n^2}:  $\mathcal{O}(n^2)$ 
1704 \def\liOOhneMathe#1{
1705 \mathcal{O} \liRundeKlammer{#1}
1706 }
1707 \def\liO#1{
1708 \ifmmode
1709 \liOOhneMathe{#1}
1710 \else
1711 $\liOOhneMathe{#1}$
1712 \fi
1713 }

\liT Let-Abkürzung: \let\T=\liT
\liT{16}{2}:  $16 \cdot T\left(\frac{n}{2}\right)$  \liT{}{2}:  $T\left(\frac{n}{2}\right)$ 
1714 \def\liTOhneMathe#1#2{
1715 \tl_if_blank:nTF {#1}
1716 {}
1717 {#1 \cdot }
1718 T
1719 \liRundeKlammer{\frac{n}{#2}}
1720 }
1721 \def\liT#1#2{
1722 \ifmmode
1723 \liTOhneMathe{#1}{#2}
1724 \else
1725 $\liTOhneMathe{#1}{#2}$
1726 \fi
1727 }

\liRekursionsGleichung \liRekursionsGleichung:  $T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$ 
1728 \def\liRekursionsGleichung{
1729 $T(n) = \liT{a}{b} + f(n)$
1730 }

\liBedingungEins \liBedingungEins:  $f(n) \in \mathcal{O}(n^{\log_b a - \varepsilon})$ 
1731 \def\liBedingungEins{
1732 $f(n) \in \liO{n^{\log\sb{b}a - \varepsilon}}$
1733 }

\liBedingungZwei \liBedingungZwei:  $f(n) \in \Theta(n^{\log_b a})$ 
1734 \def\liBedingungZwei{
1735 $f(n) \in \liTheta{n^{\log\sb{b}a}}$
1736 }

\liBedingungDrei \liBedingungDrei:  $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ 
1737 \def\liBedingungDrei{
1738 $f(n) \in \liOmega{n^{\log\sb{b}a + \varepsilon}}$
1739 }

1740 \ExplSyntaxOff

```

\liMasterVariablen

```

1741 \def\liMasterVariablen{
1742   \begin{displaymath}
1743     T(n) = \liT{a}{b} + f(n)
1744   \end{displaymath}
1745
1746   \begin{itemize}
1747     \item[$a = $]
1748       Anzahl der Unterprobleme in der Rekursion
1749
1750     \item[$\textstyle{\frac{1}{b}} = $]
1751       Teil des Originalproblems, welches wiederum durch alle Unterprobleme
1752       repräsentiert wird
1753
1754     \item[$f(n) = $]
1755       Kosten (Aufwand, Nebenkosten), die durch die Division des Problems und
1756       die Kombination der Teillösungen entstehen
1757   \end{itemize}
1758   \footcite{wiki:master-theorem}
1759   \footcite[Seite 19-35 (PDF 11-24)]{aud:fs:2}
1760 }

```

\liMasterFaelle

```

1761 \def\liMasterFaelle{
1762   \begin{description}
1763     \item[1. Fall:]
1764        $T(n) \in \liTheta{n^{\log_{sb{b}}a}}$ 
1765
1766       \hfill falls \liBedingungEins
1767       für  $\varepsilon > 0$ 
1768
1769     \item[2. Fall:]
1770        $T(n) \in \liTheta{n^{\log_{sb{b}}a} \cdot \log n}$ 
1771
1772       \hfill falls \liBedingungZwei
1773
1774     \item[3. Fall:]
1775        $T(n) \in \liTheta{f(n)}$ 
1776
1777       \hfill falls \liBedingungDrei
1778       für  $\varepsilon > 0$ 
1779       und ebenfalls für ein  $c$  mit  $0 < c < 1$  und alle hinreichend großen  $n$ 
1780       gilt:
1781        $a \cdot f(\textstyle{\frac{n}{b}}) \leq c \cdot f(n)$ 
1782     \end{description}
1783 }

```

\liMasterVariablenDeklaration

```

1784 \def\liMasterVariablenDeklaration#1#2#3{
1785   \begin{description}
1786     \item[Allgemeine Rekursionsgleichung:] \strut
1787
1788     \liRekursionsGleichung
1789
1790     \item[Anzahl der rekursiven Aufrufe ( $a$ ):] \strut
1791
1792     #1
1793
1794     \item[Anteil Verkleinerung des Problems ( $b$ ):] \strut
1795
1796     um  $\frac{1}{#2}$  also  $b = #2$ 
1797
1798     \item[Laufzeit der rekursiven Funktion ( $f(n)$ ):] \strut

```

```

1799
1800     $#3$
1801
1802     \item[Ergibt folgende Rekursionsgleichung:] \strut
1803
1804     $T(n) = \liT{\#1}{\#2} + \#3$
1805     \end{description}
1806 }

\liMasterFallRechnung
1807 \def\liMasterFallRechnung#1#2#3{
1808     \begin{description}
1809     \item[1. Fall:] \liBedingungEins:
1810
1811     #1
1812
1813     \item[2. Fall:] \liBedingungZwei:
1814
1815     #2
1816
1817     \item[3. Fall:] \liBedingungDrei:
1818
1819     #3
1820     \end{description}
1821 }

\liMasterExkurs
1822 \def\liMasterExkurs{
1823     \begin{liExkurs}[Master-Theorem]
1824     \liMasterVariablen
1825
1826     \noindent
1827     Dann gilt:
1828
1829     \liMasterFaelle
1830     \end{liExkurs}
1831 }

\liMasterWolframLink Link zu Wolframalpha (+ durch Prozent 2B ersetzen)
1832 \def\liMasterWolframLink#1{
1833     Berechne die Rekursionsgleichung auf WolframAlpha:
1834     \href{https://www.wolframalpha.com/input/?i=#1}{WolframAlpha}
1835 }

1836

```

2.25 mathe.sty

```
1837 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1838 \ProvidesPackage{lehramt-informatik-mathe}[2020/06/10]
1839
1840 % for example \ltimes \rtimes
1841 %\RequirePackage{amssymb}
1842 \RequirePackage{amsmath}
1843
1844 %%
1845 % \mlq \mrq
1846 %%
1847 \DeclareMathSymbol{\mlq}{\mathord}{operators}{``}
1848 \DeclareMathSymbol{\mrq}{\mathord}{operators}{`}
1849
```

2.26 minimierung.sty

```

1850 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1851 \ProvidesPackage{lehramt-informatik-mathe}[2021/03/13 Für den
1852 Minimierungsalgorithmus von einem NEA zu einem DEA]

1853 \liLadePakete{typographie}

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

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

\liFussnote

1854 \def\liFussnote#1{\$x_{#1}\$}

1855 \def\li@fussnote@text#1#2{
1856 \liFussnote{#1}
1857 \quad
1858 {\footnotesize #2}
1859 }

\liFussnoteEinsText

1860 \def\liFussnoteEinsText{
1861 \li@fussnote@text{1}
1862 {Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.}
1863 }

\liFussnoteZweiText

1864 \def\liFussnoteZweiText{
1865 \li@fussnote@text{2}
1866 {Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt.}
1867 }

\liFussnoteDreiText

1868 \def\liFussnoteDreiText{
1869 \li@fussnote@text{3}

```



```

1870 {In weiteren Iterationen markierte Zustände.}
1871 }

\liFussnoteVierText
1872 \def\liFussnoteVierText{
1873   \li@fussnote@text{4}
1874   {...}
1875 }

\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.                          |
| $x_4$ | ...                                                                  |



1876 \def\liFussnoten{
1877   \bigskip
1878
1879   \noindent
1880   \liFussnoteEinsText
1881
1882   \noindent
1883   \liFussnoteZweiText
1884
1885   \noindent
1886   \liFussnoteDreiText
1887
1888   \noindent
1889   \liFussnoteVierText
1890 }

\liLeereZelle \liLeereZelle:  $\emptyset$ 
Let-Abkürzung: \let\l=\liLeereZelle
1891 \def\liLeereZelle{\$\emptyset\$}

\liZustandsPaarVariablenName
1892 \def\liZustandsPaarVariablenName{z}

\liZustandsPaar
1893 \def\liZustandsPaar#1#2{
1894   $(
1895     \liZustandsPaarVariablenName_#1,
1896     \liZustandsPaarVariablenName_#2
1897   )$
1898 }

liUebergangsTabelle
1899 \renewcommand{\arraystretch}{1.4}
1900 \NewDocumentEnvironment{ liUebergangsTabelle } { m m } {
1901   \liPseudoUeberschrift{Übergangstabelle}
1902   \begin{center}
1903     \begin{tabular}{r|l|l}
1904       \textbf{Zustandspaar} & \textbf{#1} & \textbf{#2} \\ \hline
1905     \end{tabular}
1906   \end{center}
1907 }

liUeberschriftDreiecksTabelle \liUeberschriftDreiecksTabelle: Minimierungstabelle (Table filling)

1909 \ExplSyntaxOn
1910 \def\liUeberschriftDreiecksTabelle{
1911   \liPseudoUeberschrift{Minimierungstabelle~(Table~filling)}
1912 }

```

\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. —————

```

1913 \def\liMinimierungErklaerung{
1914   %\footcite[Seite~19]{koenig}
1915   \liParagraphMitLinien{
1916     Der~Minimierungs-Algorithmus~(auch~Table-Filling-Algorithmus-genannt)~
1917     trägt~in~seinem~Verlauf~eine~Markierung~in~alle~
1918     diejenigen~Zellen~der~Tabelle~ein,~die~zueinander~nicht~äquivalente~
1919     Zustände~bezeichnen.~Die~Markierung~„ $x_{\sb{n}}$ “~in~einer~Tabellenzelle~( $i$ ,~
1920      $j$ )~bedeutet~dabei,~dass~das~Zustandspaar~( $i$ ,~ $j$ )~in~der~ $k$ -ten~
1921     Iteration~des~Algorithmus~markiert~wurde~und~die~Zustände~ $i$ ~und~ $j$ ~
1922     somit~zueinander~( $k - 1$ )~äquivalent,~aber~nicht~ $k$ -äquivalent~und~
1923     somit~insbesondere~nicht~äquivalent~sind.~Bleibt~eine~Zelle~bis~zum~Ende~
1924     unmarkiert,~sind~die~entsprechenden~Zustände~zueinander~äquivalent.
1925   }
1926 }
1927 \ExplSyntaxOff
1928

```

2.27 normalformen.sty

```
1929 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1930 \ProvidesPackage{lehramt-informatik-normalformen}[2020/12/10]
1931 Hilfsmakros zum Setzen von Normalformen, Funktionalen Abhängigkeiten,
1932 Attributhülle]
```

Mathe für textit tex etc, typographie für geschweifte Klammer

```
1933 \liLadePakete{mathe,typographie}
1934 \directlua{
1935   helper = require('lehramt-informatik-helfer')
1936   normalformen = require('lehramt-informatik-normalformen')
1937 }
```

Let-Abkürzungen

```
\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
```

```
1938 \def\liTeilen#1{
1939   \directlua{tex.print(normalformen.teilen('#1'))}
1940 }
```

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

```
1941 \def\liAttributHuelleOhneMathe#1{\text{AttrHülle}(\#1)}
1942 \def\liAttributHuelle#1{
1943   \ifmmode
1944     \liAttributHuelleOhneMathe{\#1}
1945   \else
1946     $\liAttributHuelleOhneMathe{\#1}$
1947   \fi
1948 }
```

\liAttributMenge Let-Abkürzung: `\let\m=\liAttributMenge`

```
1949 \def\liAttributMenge#1{\{ \textit{\#1} \}}
```

liAHuelle

```
1950 \NewDocumentEnvironment{ liAHuelle } { +b } {
1951   \begin{group}
1952     \footnotesize
1953     \begin{multline*}
1954       \#1
1955     \end{multline*}
1956   \end{group}
1957 } { }
```

\liLinksReduktion Nur innerhalb von liAHuelle zu verwenden bzw. multline

Let-Abkürzung: `\let\ahL=\liLinksReduktion`
`\ahL{ursprüngliche linke Attributmenge}{ohne dieses Attribut}{Ergebnis}`

```
1958 \def\liLinksReduktion#1#2#3{
1959   \shoveleft{
1960     \liAttributHuelleOhneMathe{FA,
1961       \liAttributMenge{\#1 \string\ #2}} =
1962   } \}
```

```

1963 \shoveright{
1964   \liAttributMenge{#3}
1965 } \\\
1966 }

```

```

\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( $F, \{A, E \setminus E\}$ ) =  $\{A, E, F, B, D\}$ 
1967 \def\liLinksReduktionInline#1#2#3{
1968   $\liAttributHuelleOhneMathe{F,
1969   \liAttributMenge{#1 \string\ #2}} =
1970   \liAttributMenge{#3}$
1971 }

```

```

\liLinksReduktionInline Let-Abkürzung: \let\ahr=\liLinksReduktionInline
    \ahr{gelöschte FA}{neue FA ohne rechts Attribut}{gegebene Attribute}{Ergebnis}
1972 \def\liRechtsReduktionInline#1#2#3#4{
1973   $\liAttributHuelleOhneMathe{
1974     F \setminus
1975     \liFunktionaleAbhaengigkeit{#1}
1976     \def\tmp{#2}\ifx\tmp\empty
1977     \else
1978       \cup \liFunktionaleAbhaengigkeit{#2}
1979     \fi
1980     ,
1981     \liAttributMenge{#3}
1982   } =
1983   \liAttributMenge{#4}$
1984 }

```

```

\liFunktionaleAbhaengigkeit Let-Abkürzung: \let\fa=\liFunktionaleAbhaengigkeit
    \fa{A, B -> C, D}:  $\{A, B\} \rightarrow \{C, D\}$  Regulärer Ausdruck zum Konvertieren:
    \$(.*?) \\\rightarrow (.*)\$ \\\fa{$1 -> $2}
1985 \def\liFunktionaleAbhaengigkeit#1{%
1986   \directlua{normalformen.drucke_funk_abhaengigkeit('#1', false)}%
1987 }

```

```

\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{$1 -> $2};
1988 \NewDocumentCommand {\liFunktionaleAbhaengigkeiten} { O{FA} m } {
1989   \liGeschweifteKlammern
1990   {#1}
1991   {
1992     \begin{align*}
1993     \directlua{normalformen.drucke_funk_abhaengigkeiten('#2')}

```

```

1994     \end{align*}
1995   }
1996   {-0.5cm}
1997   {-1.7cm}
1998 }

\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}

1999 \NewDocumentCommand {\liRelation} { O{R} m } {
2000   $\directlua{
2001     local name = helper.konvertiere_tiefgestellt('#1')
2002     tex.print(name)
2003   }$(\textit{\,#2\,})
2004 }

2005

```

2.28 petri.sty

2006 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2007 \ProvidesPackage{lehramt-informatik-petri}[2020/12/03]

Let-Abkürzungen

\let\t=\liPetriTransitionsName

\let\tp=\liPetriTransPfeile

\let\k=\liPetriErreichKnotenDrei

2008 \RequirePackage{tikz}

2009 \usetikzlibrary{petri,arrows.meta}

Für die Darstellungsmatrix

2010 \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

```
2011 \def\liPetriSetzeSchluessel{%
2012   \def\TmpTransitionOne{}%
2013   \def\TmpTransitionTwo{}%
2014   \def\TmpTransitionThree{}%
2015   \def\TmpTransitionFour{}%
2016   \def\TmpTransitionFive{}%
2017   \def\TmpTransitionSix{}%
2018   \def\TmpTransitionSeven{}%
2019   \def\TmpTransitionEight{}%
2020   \def\TmpTransitionNine{}%
2021   \def\TmpTransitionTen{}%
2022   \pgfkeys{/petri/.cd,
2023     p1/.store in=\TmpPlaceOne,p1/.default=0,p1,
2024     p2/.store in=\TmpPlaceTwo,p2/.default=0,p2,
2025     p3/.store in=\TmpPlaceThree,p3/.default=0,p3,
2026     p4/.store in=\TmpPlaceFour,p4/.default=0,p4,
2027     p5/.store in=\TmpPlaceFive,p5/.default=0,p5,
```

```

2028   p6/.store in=\TmpPlaceSix,p6/.default=0,p6,
2029   p7/.store in=\TmpPlaceSeven,p7/.default=0,p7,
2030   p8/.store in=\TmpPlaceEight,p8/.default=0,p8,
2031   p9/.store in=\TmpPlaceNine,p9/.default=0,p9,
2032   p10/.store in=\TmpPlaceTen,p10/.default=0,p10,
2033   t1/.store in=\TmpTransitionOne,t1/.default=activated,
2034   t2/.store in=\TmpTransitionTwo,t2/.default=activated,
2035   t3/.store in=\TmpTransitionThree,t3/.default=activated,
2036   t4/.store in=\TmpTransitionFour,t4/.default=activated,
2037   t5/.store in=\TmpTransitionFive,t5/.default=activated,
2038   t6/.store in=\TmpTransitionSix,t6/.default=activated,
2039   t7/.store in=\TmpTransitionSeven,t7/.default=activated,
2040   t8/.store in=\TmpTransitionEight,t8/.default=activated,
2041   t9/.store in=\TmpTransitionNine,t9/.default=activated,
2042   t10/.store in=\TmpTransitionTen,t10/.default=activated,
2043   scale/.store in=\TmpScale,scale/.default=0.5,
2044   x/.store in=\TmpX,x/.default=5,
2045   y/.store in=\TmpY,y/.default=5,
2046   }%
2047 }

2048 \tikzset{
2049   li petri/.style={
2050     activated/.style={
2051       very thick
2052     },
2053     inhibitor/.style={
2054       {Circle[open,length=2mm,fill=white]}-
2055     }
2056   }
2057 }

```

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

```

2058 \def\liPetriTransitionsNameOhneMathe#1{t\sb{#1}}
2059 \def\liPetriTransitionsName#1{
2060   \ifmmode
2061     \liPetriTransitionsNameOhneMathe{#1}
2062   \else
2063     $\liPetriTransitionsNameOhneMathe{#1}$
2064   \fi
2065 }

```

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

```

2066 \NewDocumentCommand{ \liPetriErreichTransition } { m m m O{} O{} } {
2067   \draw[->] (#1) edge[#4] node[pos=0.5,auto,sloped,#5]{t\sb{#3}$} (#2);
2068 }

```

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

```

2069 \def\liPetriErreichKnotenDrei#1#2#3{(#1,#2,#3)}

```

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

```

2070 \def\liPetriTransPfeile#1{ $\rightarrow \hspace{0.4cm} \liPetriTransitionsName{#1} \hspace{0.3cm}
2071 }

```

2.29 potenzmengen-konstruktion.sty

```
2072 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2073 \ProvidesPackage{lehramt-informatik-potenzmengen-konstruktion}[2021/02/21 Hilfsmakros
2074 einzusetzen bei der Potenzmengen-Konstruktion (Potenzmengen-Algorithmus)]
```

```
2075 \liLadePakete{formale-sprachen}
2076 \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}
    }
  }
}
```

```
2077 \def\liZustandsMengenSammlung#1#2{
2078   \liZustandsnameGross{#1}
2079   {
2080     \footnotesize
2081     \liPotenzmenge{
2082       \str_case:nn {#1} {#2
2083       }
2084     }
2085 }
```

```
\liZustandsMengenSammlungNr
```

```
2086 \def\liZustandsMengenSammlungNr#1#2{
2087   \liZustandsnameGross{#1}
2088   {
```



```
2089     \footnotesize
2090     \liZustandsmengeNr{
2091         \str_case:nn {#1} #2
2092     }
2093 }
2094 }

2095 \ExplSyntaxOff
2096
```

2.30 pseudo.sty

2097 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2098 \ProvidesPackage{lehramt-informatik-pseudo}[2020/12/30 Paket zum Setzen
 2099 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>Data: $G = (V, E, w)$: ein zusammenhängender, ungerichteter, kantengewichteter Graph kruskal(G)</p> <p>$E' \leftarrow \emptyset$; $L \leftarrow E$; Sortiere die Kanten in L aufsteigend nach ihrem Kantengewicht.;</p> <p>while $L \neq \emptyset$ do</p> <div style="margin-left: 20px;"> <p>wähle eine Kante $e \in L$ mit kleinstem Kantengewicht;</p> <p>entferne die Kante e aus L;</p> <p>if der Graph $(V, E' \cup \{e\})$ keinen Kreis enthält then</p> <div style="margin-left: 20px;"> <p>$E' \leftarrow E' \cup \{e\}$;</p> </div> <p>end</p> </div> <p>end</p> <p>Result: $M = (V, E')$ ist ein minimaler Spannbaum von G.</p>

2100 \RequirePackage[german,boxruled]{algorithm2e}

2101

2.31 pumping-lemma.sty

2102 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2103 \ProvidesPackage{lehramt-informatik-pumping-lemma}[2021/08/11 Enthält die
 2104 Definitionen für das Pumping-Lemma in der Regulären Sprache und
 2105 in der Kontextfreien Sprache]

\liPumpingRegulaer

```
2106 \def\liPumpingRegulaer{%
2107   Es sei  $L$  eine reguläre Sprache. Dann gibt es eine Zahl  $j$ , sodass für
2108   alle Wörter  $\omega \in L$  mit  $|\omega| \geq j$  (jedes Wort  $\omega$  in
2109    $L$  mit Mindestlänge  $j$ ) jeweils eine Zerlegung  $\omega = uvw$ 
2110   existiert, sodass die folgenden Eigenschaften erfüllt sind:
2111
2112   \begin{enumerate}
2113     \item  $|v| \geq 1$ 
2114     (Das Wort  $v$  ist nicht leer.)
2115
2116     \item  $|uv| \leq j$ 
2117     (Die beiden Wörter  $u$  und  $v$  haben zusammen höchstens die Länge  $j$ .)
2118
2119     \item Für alle  $i = 0, 1, 2, \dots$  gilt  $uv^i w \in L$ 
2120     (Für jede natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^i w$  in der
2121     Sprache  $L$ )
2122   \end{enumerate}
2123
2124   Die kleinste Zahl  $j$ , die diese Eigenschaften erfüllt, wird
2125   Pumping-Zahl der Sprache  $L$  genannt.\footcite{wiki:pumping-lemma}
2126 }
```

\liPumpingKontextfrei

```
2127 \def\liPumpingKontextfrei{%
2128   Es sei  $L$  eine kontextfreie Sprache. Dann gibt es eine Zahl  $j$ , sodass
2129   sich alle Wörter  $\omega \in L$  mit  $|\omega| \geq j$  zerlegen lassen in
2130    $\omega = uvwxy$ , sodass die folgenden Eigenschaften erfüllt sind:
2131
2132   \begin{enumerate}
2133     \item  $|vx| \geq 1$ 
2134     (Die Wörter  $v$  und  $x$  sind nicht leer.)
2135
2136     \item  $|vwx| \leq j$ 
2137     (Die Wörter  $v$ ,  $w$  und  $x$  haben zusammen höchstens die Länge  $j$ .)
2138
2139     \item Für alle  $i \in \mathbb{N}_0$  gilt  $uv^iwx^iy \in L$  (Für jede
2140     natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^iwx^iy$  in der
2141     Sprache  $L$ )
2142   \end{enumerate}
2143 }
```

2144

2.32 quicksort.sty

```

2145 % https://tex.stackexchange.com/a/142634
2146 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2147 \ProvidesPackage{lehramt-informatik-quicksort}[2020/06/12]
2148
2149 %-----
2150 % USAGE:
2151 % \QSinitialize{comma, separated, numerical, values}
2152 % \loop
2153 % \QSpivotStep
2154 % \ifnum\value{pivotcount}>0
2155 %   \QSSortStep
2156 % \repeat
2157 %-----
2158
2159 % xintfrac does not load xinttools, this must be done explicitly if needed as here.
2160 \RequirePackage{xintfrac, xinttools}
2161
2162 \RequirePackage{tikz}
2163
2164 %-----
2165 % FIRST PART: TikZ styles and macros for the actual drawing
2166 \newcounter{cellcount}% used for coordinates of the node
2167 \newcounter{pivotcount}% when it will remain at zero, will signal the sort is finished.
2168
2169 % Styles defined by Tom Bombaldi. (modified: all share the same size)
2170 % (re-modified \bf -> \bfseries due to extremely annoying warnings from
2171 % KOMA-script which are truly a pain and do not make any sense regarding \bf:
2172 % if I want to use \bf, and know what I am doing, why should I get HARASSED
2173 % by police of LaTeX good conduct ? )
2174 \tikzset{ll/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=lime!70!gray},
2175          oo/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=olive!50},
2176          rr/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2177 % this is the "b" style as used in the image below
2178          bb/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2179 % nicer:
2180          bb/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=white, text=magenta},
2181          gg/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=gray, text=white, text=white},
2182
2183 % NOTE the b style was originally the same as the r(aised) style apart from
2184 % not being raised, but I find it nicer with a somewhat different
2185 % specification. I have not updated the images though.
2186
2187 % How the nodes are drawn depending on whether on the left of the pivot value
2188 % or on the right, or is a pivot value, or a raised pivot during selection phase.
2189
2190 \def\DecoLEFT #1{%
2191   \xintFor* ##1 in {#1} \do
2192     {\stepcounter{cellcount}\node[o] at (\arabic{cellcount},0) {##1};}%
2193 }
2194
2195 \def\DecoINERT #1{%
2196   \xintFor* ##1 in {#1} \do
2197     {\stepcounter{cellcount}\node[g] at (\arabic{cellcount},0) {##1};}%
2198 }
2199
2200 \def\DecoRIGHT #1{%
2201   \xintFor* ##1 in {#1} \do
2202     {\stepcounter{cellcount}\node[l] at (\arabic{cellcount},0) {##1};}%
2203 }
2204
2205 \def\DecoLEFTwithPivot #1{\stepcounter{pivotcount}%
2206   \xintFor* ##1 in {#1} \do

```

```

2207     {\stepcounter{cellcount}}%
2208     \xintifForLast {\node[r]}\node[o]} at (\arabic{cellcount},0) {##1};}%
2209 }
2210
2211 \def\DecoINERTwithPivot #1{\stepcounter{pivotcount}%
2212     \xintFor* ##1 in {#1} \do
2213     {\stepcounter{cellcount}}%
2214     \xintifForLast {\node[b]}\node[g]} at (\arabic{cellcount},0) {##1};}%
2215 }
2216
2217 \def\DecoRIGHTwithPivot #1{\stepcounter{pivotcount}%
2218     \xintFor* ##1 in {#1} \do
2219     {\stepcounter{cellcount}}%
2220     \xintifForLast {\node[r]}\node[l]} at (\arabic{cellcount},0) {##1};}%
2221 }
2222
2223 %-----
2224 % SECOND PART: the actual sorting routines.
2225
2226 \def\QS@sort@a #1{\expandafter \QS@sort@b \expandafter {\xintLength {#1}}{#1}}
2227 \def\QS@sort@b #1{\ifcase #1
2228     \expandafter\QS@sort@empty
2229     \or\expandafter\QS@sort@single
2230     \else\expandafter\QS@sort@c
2231     \fi
2232 }%
2233 \def\QS@sort@empty #1{}
2234 \def\QS@sort@single #1{\QSIr {#1}}
2235
2236 % This step is to pick the last as pivot.
2237 \def\QS@sort@c #1%
2238     {\expandafter\QS@sort@d\expandafter {\romannumeral0\xintnthelt {-1}{#1}}{#1}}%
2239
2240 % Here \QSLr, \QSIr, \QSR have been let to \relax.
2241 % The trick with \xintApplyUnbraced is that for example when selecting
2242 % the elements smaller than pivot, if we had been using \xintApply we
2243 % would have had at the minimum an empty brace pair. Thus we use the
2244 % "unbraced" variant, but then the \QS@select@smaller has added in
2245 % anticipation a level of braces.
2246 \def\QS@sort@d #1#2{%
2247     \QSLr {\xintApplyUnbraced {\QS@select@smaller {#1}}{#2}}}%
2248     \QSIr {\xintApplyUnbraced {\QS@select@equal {#1}}{#2}}}%
2249     \QSRr {\xintApplyUnbraced {\QS@select@greater {#1}}{#2}}}%
2250 }%
2251 \def\QS@select@smaller #1#2{\xintifLt {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2252 \def\QS@select@equal #1#2{\xintifEq {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2253 \def\QS@select@greater #1#2{\xintifGt {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2254
2255 %
2256 % NOTE 1: thus, each comparison with the pivot is done three (!) times.
2257 %
2258 % NOTE 2: we may well end up with \QSLr {<empty>} situations. This is handled
2259 % silently by the \xintFor loops, and also when \QSLr becomes \QS@sort@a, the
2260 % latter must handle correctly an empty argument.
2261
2262 %-----
2263 % THIRD PART: the main macros \QSpivotStep, \QSsortStep and \QSinitialize.
2264
2265 % This draws all with suitable highlighting for the newly chosen pivots
2266 % (which will be shown raised)

```

```

2267 \def\QSpivotStep {\let\QSLr\DecoLEFTwithPivot
2268             \let\QSIr\DecoINERT
2269             \let\QSIrr\DecoINERT
2270             \let\QSRr\DecoRIGHTwithPivot
2271 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2272             \setcounter{cellcount}{0}\setcounter{pivotcount}{0}%
2273             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2274 }
2275
2276 % This sorts and then draws, showing where the pivot chosen in the previous
2277 % step go. Next time they will have become "inert". If pivotcount is still at
2278 % zero on exit from \QSpivotStep, then this is the signal to stop before
2279 % executing \QSSortStep.
2280 \def\QSSortStep {\def\QSLr {\noexpand\QS@sort@a}%
2281             \def\QSRr {\noexpand\QS@sort@a}%
2282             \def\QSIr {\noexpand\QSIrr}%
2283             \let\QSIrr\relax
2284             \edef\QS@list{\QS@list}%
2285             \let\QSLr\relax
2286             \let\QSRr\relax
2287             \let\QSIr\relax
2288             \edef\QS@list{\QS@list}%
2289             \let\QSLr\DecoLEFT
2290             \let\QSIr\DecoINERTwithPivot
2291             \let\QSIrr\DecoINERT
2292             \let\QSRr\DecoRIGHT
2293 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2294             \setcounter{cellcount}{0}%
2295             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2296 }
2297
2298 \def\QSinitialize #1{%
2299     % first, we convert the comma separated values into a list of braced items
2300     % we use an \edef, and anyhow many \edef's will be used later
2301     \edef\QS@list {\noexpand\QSRr {\xintCSVtoList {#1}}}%
2302     \let\QSRr\DecoRIGHT
2303     % The \QSRr marker mutated to draw the last element as
2304     % pivot and the earlier ones with the suitable style.
2305     %
2306     % The list of marked braced items \QS@list is used both for drawing
2307     % (as here) and for doing the exchange of elements during sort.
2308     \par\centerline{\rule[1.5mm]{0pt}{8mm}\setcounter{cellcount}{0}%
2309             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2310 }
2311

```

2.33 relationale-algebra.sty

2312 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2313 \ProvidesPackage{lehramt-informatik-relationale-algebra}[2020/12/11]

2314 \RequirePackage{amsmath}

2315 \RequirePackage{amssymb}

Zum Zeichnen von Operatorenbäumen verwenden wir TikZ

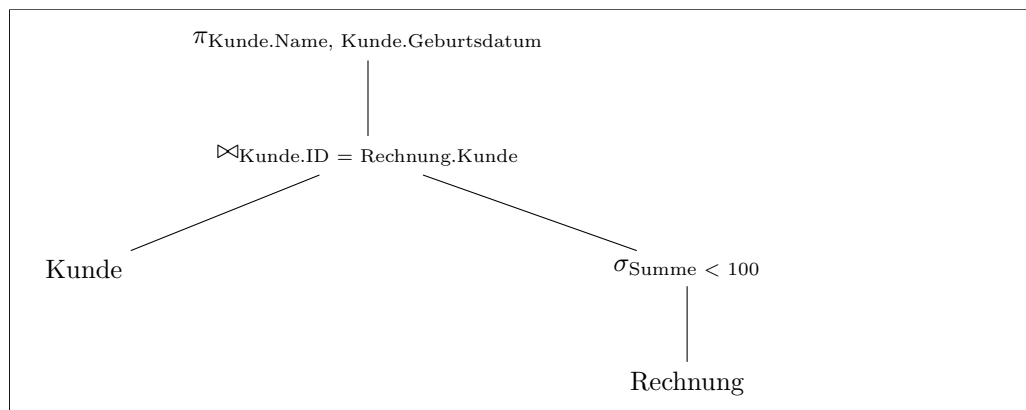
```
\begin{tikzpicture}
  \node
    (pi) {\pi_{\text{Kunde.Name, Kunde.Geburtsdatum}}};

  \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}
```



2316 \RequirePackage{tikz}

2317 \usetikzlibrary{positioning}

Privates Makros, das zwei Querstriche erzeugt.

```
2318 \def\o@join{\setbox0=\hbox{\bowtie}}%
2319 \rule[-.02ex]{.25em}{.4pt}\llap{\rule[\ht0]{.25em}{.4pt}}%
2320 }
```

\leftouterjoin A \leftouterjoin B: $A \bowtie B$

```
2321 \def\leftouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie}}
```

\rightouterjoin A \rightouterjoin B: $A \bowtie B$

```
2322 \def\rightouterjoin{\mathbin{\bowtie\mkern-5.8mu\o@join}}
```

\fullouterjoin A \fullouterjoin B: $A \bowtie B$

```
2323 \def\fullouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie\mkern-5.8mu\o@join}}
```

2324

2.34 rmodell.sty

```

2325 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2326 \ProvidesPackage{lehramt-informatik-rmodell}[2020/09/01]
2327 Makros und Umgebungen zum Setzen des Relationenmodells beim Thema
2328 Datenbanken.]
2329 \RequirePackage{soul}

```

Let-Abkürzungen

```

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

```

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

```

2330 \def\liPrimaer#1{\ul{#1}}

```

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

```

2331 \def\liFremd#1{{\setul{-0.9em}{}\ul{#1}}}

```

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

```

2332 \def\li@Rmodell@Schrift{\footnotesize\ttfamily}
2333 \ExplSyntaxOn
2334 \NewDocumentEnvironment { liRmodell }
2335 { +b }
2336 {
2337   \medskip
2338   {
2339     \linespread{2}
2340     \setlength{\parindent}{0pt}
2341     \li@Rmodell@Schrift#1
2342   }
2343   \medskip
2344 } {}
2345 \ExplSyntaxOff

```

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

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

```

2346 \def\liRelationMenge#1#2{
2347 \noindent
2348 #1 : \[ #2 ]\}
2349 \par
2350 }

```

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

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

```

2351 \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}

```

```

2352 \NewDocumentEnvironment { liRelationenSchemaFormat }{ +b } {} {}

```

```

2353

```


2.35 sortieren.sty

```
2354 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2355 \ProvidesPackage{lehramt-informatik-sortieren}[2020/06/10 Schaubilder
2356 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}
}
```



```
2357 \RequirePackage{tikz}
2358 \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.

```
2359 \def\liVertauschen#1{
2360   \directlua{
2361     local sortieren = require('lehramt-informatik-sortieren')
2362     sortieren('#1')
2363   }
2364 }
```

`\liSortierPfeil`

```
2365 \def\liSortierPfeil#1#2{
2366   \draw[-latex] ([xshift=1mm]reihe.#1 north) -- ++(0,0.25) -| ([xshift=-1mm]reihe.#2 north);
2367 }
```

`\liSortierPfeilUnten`

```
2368 \def\liSortierPfeilUnten#1#2{
2369   \draw[-latex] ([xshift=1mm]reihe.#1 south) -- ++(0,-0.25) -| ([xshift=-1mm]reihe.#2 south);
2370 }
```

`\liSortierMarkierung`

```
2371 \def\liSortierMarkierung#1#2{\node[
2372   draw,
2373   very thick,
2374   fit=(reihe.#1) (reihe.#2),
2375   inner sep=0pt
2376 ] {}
2377 }

2378 \tikzset{
2379   li sortierung zahlenreihe/.style={
2380     draw,
2381     thin,
2382     font=\large,
2383     rectangle split horizontal,
2384     rectangle split,
2385   }
2386 }
```

```

2387 % https://tex.stackexchange.com/a/140895
2388 \RequirePackage{forest,xstring}
2389 \usetikzlibrary{calc}
2390
2391 \makeatletter
2392 \pgfmathdeclarefunction{strrepeat}{2}{%
2393   \begingroup\pgfmathint{#2}\pgfmath@count\pgfmathresult
2394   \let\pgfmathresult\pgfutil@empty
2395   \pgfutil@loop\ifnum\pgfmath@count>0\relax
2396     \expandafter\def\expandafter\pgfmathresult\expandafter{\pgfmathresult#1}%
2397     \advance\pgfmath@count-1\relax
2398   \pgfutil@repeat\pgfmath@smuggleone\pgfmathresult\endgroup}
2399 \makeatother
2400
2401 \def\myNodes{}
2402
2403 \ExplSyntaxOn
2404 \newcommand*\sortList[1]{%
2405   \clist_sort:Nn#1{\int_compare:nNnTF{##1}>{##2}\sort_return_swapped:\sort_return_same:}}
2406 \ExplSyntaxOff
2407
2408 \forestset{
2409   sort/.code={%
2410     \pgfmathparse{level()>\forestSortLevel}%
2411     \ifnum\pgfmathresult=0
2412       \StrSubstitute{\forestov{content}}{ }{,}{\myList}%
2413       \sortList\myList
2414       \StrSubstitute{\myList}{,}{ }{\myList}%
2415       \pgfmathparse{strrepeat("1",level())}%
2416       \xappto\myNodes{\noexpand\node at ({\forestov{name}}|-m)!-1!({\forestov{name}}$)
2417         (m\forestov{name}) {\myList}}%
2418       \pgfmathparse{level()=\forestSortLevel}%
2419       \ifnum\pgfmathresult=1
2420         \forestOget{\forestov{@first}}{name}\forestFirst
2421         \forestOget{\forestov{@last}}{name}\forestLast
2422         \xappto\myNodes{{[<-]edge (\forestOv{\forestov{@first}}{name})
2423           \ifx\forestFirst\forestLast\else edge (\forestOv{\forestov{@last}}{name})\fi}}%
2424       \fi
2425       \ifnum\forestov{@parent}=0\else
2426         \xappto\myNodes{edge (m\forestOv{\forestov{@parent}}{name})}%
2427       \fi
2428       \gappto\myNodes{;}%
2429     \fi}}
2430
2431 \forestset{sort level/.code=%
2432   \pgfmathparse{#1}\let\forestSortLevel\pgfmathresult
2433   \pgfmathparse{strrepeat("1",\forestSortLevel+1)}\let\forestOnes\pgfmathresult}
2434

```

2.36 spalten.sty

```
2435 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2436 \ProvidesPackage{lehramt-informatik-spalten}[2020/12/07 Lädt das Paket
2437 „multicol“, damit mehrspaltiger Satz mit Hilfe der Umgebung „multicols“
2438 realisiert werden kann.]
2439 \RequirePackage{multicol}
```

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

```
2440 \def\liSpaltenUmbruch{\vfill\strut\columnbreak}
```

```
2441
```

2.37 struktogramm.sty

```
2442 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2443 \ProvidesPackage{lehramt-informatik-struktogramm}[2021/01/31 Lädt das
2444 Paket struktex zum Setzen von Struktogrammen]
2445 \RequirePackage{struktex}
2446
```

2.38 syntax.sty

```

2447 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2448 \ProvidesPackage{lehramt-informatik-syntax}[2020/06/10 Ein Hüll-Paket um
2449 `minted`, das sich um die Syntax-Hervorhebung des Java-Codes kümmert.]
2450 \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.

Let-Abkürzungen

```

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

```

```

2451 \ExplSyntaxOn
2452 \directlua{
2453   syntax = require('lehramt-informatik-syntax')
2454   syntax.importiere_konfiguration('tex_repo_lokaler_pfad', '\LehramtInformatikRepository')
2455   syntax.importiere_konfiguration('github_domain', '\LehramtInformatikGithubDomain')
2456   syntax.importiere_konfiguration('github_raw_domain', '\LehramtInformatikGithubRawDomain')
2457   syntax.importiere_konfiguration('github_tex_repo', '\LehramtInformatikGithubTexRepo')
2458   syntax.importiere_konfiguration('github_code_repo', '\LehramtInformatikGithubCodeRepo')
2459   syntax.importiere_konfiguration('git_branch', '\LehramtInformatikGitBranch')
2460 }
2461 \RequirePackage{hyperref}
2462 \RequirePackage{minted}
2463 % pygmentize -L styles
2464 \usemintedstyle{colorful}
2465 %\BeforeBeginEnvironment{minted}{\begin{mdframed}}
2466 %\AfterEndEnvironment{minted}{\end{mdframed}}
2467 %\setminted{breaklines=true,linenos}
2468 \setminted{
2469   breaklines=true,
2470   linenos,
2471   fontsize=\footnotesize,
2472 }

```

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

Let-Abkürzung: `\let\j=\liJavaCode`

```

2473 \def\liJavaCode#1{\,\mintinline{java}|#1|\,}

```

`\liLatexCode` Im Zeilenfluss einen kurzen L^AT_EX-Code-Ausschnitt setzen.

```

2474 \def\liLatexCode#1{\mintinline{latex}|#1|}

```

```

2475 \def\li@GithubLink#1#2{
2476   \begin{flushright}
2477     \tiny
2478     Code-Beispiel~auf~Github~ansehen:~
2479     \href{#1}{\nolinkurl{#2}}
2480   \end{flushright}
2481 }

```

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

```

2482 \NewDocumentCommand{\liJavaDatei}{O{firstline=3} m }{
2483   \inputminted[#1]{java}{
2484     \directlua{
2485       syntax.drucke_absoluten_pfad('#2', false)
2486     }
2487   }
2488   \li@GithubLink

```

```

2489     {\directlua{syntax.drucke_github_url('#2', false)}}
2490     {\directlua{syntax.drucke_relativen_repo_pfad('#2', false)}}
2491 }

\liJavaTestDatei Eine komplette Java-Test-Datei einbinden, die Verzeichnis ./Code/src/test/java/org/bschlangaul
liegt.
2492 \NewDocumentCommand{\liJavaTestDatei}{ O{firstline=3} m }{
2493   \inputminted[#1]{java}{
2494     \directlua{
2495       syntax.drucke_absoluten_pfad('#2', true)
2496     }
2497   }
2498   \li@GithubLink
2499   {\directlua{syntax.drucke_github_url('#2', true)}}
2500   {\directlua{syntax.drucke_relativen_repo_pfad('#2', true)}}
2501 }

\liJavaExamen \liJavaExamen{66116}{2015}{03}{Kunde} Regulärer Ausdruck zum Konvertieren: \liJavaDatei([.
\liJavaExamen$1{$2}{$3}{$4}{$5}
2502 \NewDocumentCommand{\liJavaExamen}{ O{firstline=3} m m m m }{
2503   \inputminted[#1]{java}{
2504     \directlua{
2505       syntax.drucke_absoluten_examens_pfad('#2', '#3', '#4', '#5')
2506     }
2507   }
2508
2509   \li@GithubLink
2510   {\directlua{syntax.drucke_github_examens_url('#2', '#3', '#4', '#5')}}
2511   {\directlua{syntax.drucke_relativen_examens_repo_pfad('#2', '#3', '#4', '#5')}}
2512 }

\liAssemblerCode
2513 \def\liAssemblerCode#1{\mintinline{asm}|#1|}

\liAssemblerDatei \liAssemblerDatei{relativer-pfad}: Relativer Pfad zu einer Assembler-Datei.
2514 \NewDocumentCommand{\liAssemblerDatei}{ m }{
2515   \inputminted{asm}{#1}
2516 }

\liMinispracheDatei \liMinispracheDatei{relativer-pfad}: Relativer Pfad zu einer Minisprachen-Datei
(Hochsprache für die Minimaschine von Albert Wiedemann).
2517 \NewDocumentCommand{\liMinispracheDatei}{ m }{
2518   \inputminted{componentpascal}{#1}
2519 }

\liHaskellCode \liHaskellCode{haskell}: Zum Setzen von Haskell-Code.
2520 \def\liHaskellCode#1{\mintinline{haskell}|#1|}

\liHaskellDatei \liHaskellDatei{relativer-pfad}: Relativer Pfad zu einer Haskell-Datei.
2521 \NewDocumentCommand{\liHaskellDatei}{ m }{
2522   \inputminted{haskell}{#1}
2523 }

2524 \ExplSyntaxOff

\liSqlCode \liHaskellCode{sql}: Zum Setzen von SQL-Code.
Let-Abkürzung: \let\s=\liSqlCode
2525 \def\liSqlCode#1{\mintinline{sql}|#1|}

2526

```

2.39 syntaxbaum.sty

```
2527 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2528 \ProvidesPackage{lehramt-informatik-syntaxbaum}[2021/02/14 Zum Setzen von
2529 Syntaxbäumen mit Hilfe des Pakets tikz-qtrees]
2530 \RequirePackage{tikz-qtrees}
2531
2532 \tikzset{li parsetree/.style={
2533     every internal node/.style={
2534         draw,circle
2535     },
2536     every leaf node/.style={
2537         draw,rectangle
2538     },
2539 }
2540 }
2541
```

2.40 synthese-algorithmus.sty

```
2542 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2543 \ProvidesPackage{lehramt-informatik-synthese-algorithmus}[2021/03/19
2544 Hilfsmakros zum Setzen des Synthese-Algorithmuses zur Umformung einer
2545 Relation in die 3. Normalform]

2546 \liLadePakete{normalformen,mathe,typographie}
2547 \ExplSyntaxOn
```

Let-Abkürzungen

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

```
\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}
```

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}_α 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 R_α , die in einem anderen Relationenschema $R_{\alpha'}$ enthalten sind, d. h. $R_\alpha \subseteq R_{\alpha'}$.*

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

```
2548 \def\liSyntheseUeberschrift#1{
2549   {
2550     \bfseries
2551     \sffamily
2552     \str_case:nn {#1} {
2553       {1} {Kanonische-Überdeckung}
2554       {1-1} {Linksreduktion}
2555       {1-2} {Rechtsreduktion}
2556       {1-3} {Löschen-leerer-Klauseln}
2557       {1-4} {Vereinigung}
2558       {2} {Relationsschemata-formen}
2559       {3} {Schlüssel-hinzufügen}
2560       {4} {Entfernung-überflüssiger-Teilschemata}
2561     }
2562   }
2563 }
```

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

```
2564 \def\li@synthese@erklaerung@texte#1{
2565   \str_case:nn {#1} {
2566     {1} {
2567       Die-kanonische-Überdeckung---also-die-kleinst-mögliche-noch-
2568       äquivalente-Menge-von-funktionalen-Abhängigkeiten-kann-in-vier-
2569       Schritten-erreicht-werden.
2570     }
2571     {1-1} {
2572       Führe-für-jede-funktionale-Anhängigkeit-
2573       $\alpha\rightarrow\beta$-in-F$-die-Linksreduktion-durch,-
2574       überprüfe-also-für-alle-
2575       $A$-in-$\alpha$,~ob-$A$-überflüssig-ist,~d.h.-ob-
2576       $\beta\subseteqq\liAttributHuelle{F,\alpha\---A}$.
2577     }
2578     {1-2} {
2579       Führe-für-jede-(verbliebene)-funktionale-Abhängigkeit-$\alpha\rightarrow\beta$-die-Rechtsreduktion-durch,-überprüfe-also-für-
2580       alle-$B$-in-$\beta$,~ob-$B$-in-$\liAttributHuelle{F---(\alpha\rightarrow\beta)\cup(\alpha\rightarrow\beta\---B)}$,~
2581       $\alpha$-gilt.-In-diesem-Fall-ist-B-auf-der-rechten-Seite-
2582       überflüssig-und-kann-eliminiert-werden,~dh-$\alpha\rightarrow\beta$-wird-durch-$\alpha\rightarrow(\beta\---B)$-
2583       ersetzt.
2584     }
2585     {1-3} {
2586       Entferne-die-funktionalen-Abhängigkeiten-der-Form-$\alpha\rightarrow\emptyset$,~die-im-2.-Schritt-möglicherweise-
2587       entstanden-sind.
2588     }
2589     {1-4} {
2590       Fasse-mittels-der-Vereinigungsregel-funktionale-Abhängigkeiten-
2591       der-Form-$\alpha\rightarrow\beta_{\{1\}},\dots,\alpha\rightarrow\beta_{\{n\}}$,~so-dass-$\alpha\rightarrow\beta_{\{1\}}\cup\dots\cup\beta_{\{n\}}$-verbleibt.
2592     }
2593     {2} {
2594       % Kemper Seite 197
2595       Erzeuge-für-jede-funktionale-Abhängigkeit-$\alpha\rightarrow\beta$-in-F$\sb{c}$-ein-Relationenschema-$\mathcal{R}\sb{\alpha}$-
2596     }
2597   }
```

```

2603      :=~\alpha~\cup~\beta$.
2604    }
2605    {3} {
2606      Falls-eines-der-in-Schritt-2.-erzeugten-Schemata- $R_{\alpha}$ -
2607      einen-Schlüsselkandidaten-von- $R$ -bezüglich- $F_c$ -
2608      enthält,~sind-wir-fertig,~sonst-wähle-einen-Schlüsselkandidaten-
2609       $K \subseteq R$ -aus-und-definiere-folgendes-
2610      zusätzliche-Schema:~ $R_{\alpha} \setminus K := \emptyset$ -
2611      und- $F_{\alpha} \setminus K := \emptyset$ 
2612    }
2613    {4} {
2614      Eliminiere-diejenigen-Schemata- $R_{\alpha}$ ,~die-in-einem-
2615      anderen-Relationenschema- $R_{\alpha'}$ -enthalten-sind,~d.h.~
2616       $R_{\alpha} \subseteq R_{\alpha'}$ .
2617    }
2618  }
2619 }
2620 \def\liSyntheseErklaerung#1{
2621   {
2622     \itshape
2623     \footnotesize
2624     \liParagraphMitLinien{\li@synthese@erklaerung@texte{#1}}
2625   }
2626 }

```

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

```

2627 \def\liSyntheseUeberErklaerung#1{
2628   \liSyntheseUeberschrift{#1}\par
2629   \liSyntheseErklaerung{#1}
2630 }

2631 \ExplSyntaxOff
2632

```

2.41 tabelle.sty

2633 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2634 \ProvidesPackage{lehramt-informatik-tabelle}[2020/12/05 Lädt das Paket tabluarx]

2635 \RequirePackage{tabularx}

2636

2.42 typographie.sty

```
2637 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2638 \ProvidesPackage{lehramt-informatik-typographie}[2021/03/20 Typographische Makros,
2639 die das Erscheinungsbild verändern. Die Schriftdefinition sind in
2640 formatierung.sty definiert.]
```

```
2641 \ExplSyntaxOn
```

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

```
2642 \RequirePackage{fontawesome}
```

```
\liErledigt \liErledigt: ☑
```

```
2643 \let\liErledigt=\faCheckSquareO
```

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

```
2644 \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 —

```
2645 \def\liParagraphMitLinien#1{
```

```
2646 \noindent
```

```
2647 \vrule height 2pt depth -1.6pt width 0.4cm
```

```
2648 \enspace
```

```
2649 #1
```

```
2650 \enspace
```

```
2651 \leaders\vrule height 2pt depth -1.6pt \hfill \null
```

```
2652 \par
```

```
2653 \medskip
```

```
2654 }
```

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

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

```
2655 \def\liGeschweifteKlammern#1#2#3#4{
```

```
2656 \par
```

```
2657 \noindent
```

```
2658 #1 \, $\= \Bigl\{ $\
```

```
2659 \vspace{#3}
```

```
2660 #2
```

```
2661 \vspace{#4}
```

```
2662 \begin{flushright} $\Bigr\} $\end{flushright}
```

```
2663 }
```

```
2664 \ExplSyntaxOff
```

```
2665
```

2.43 uml.sty

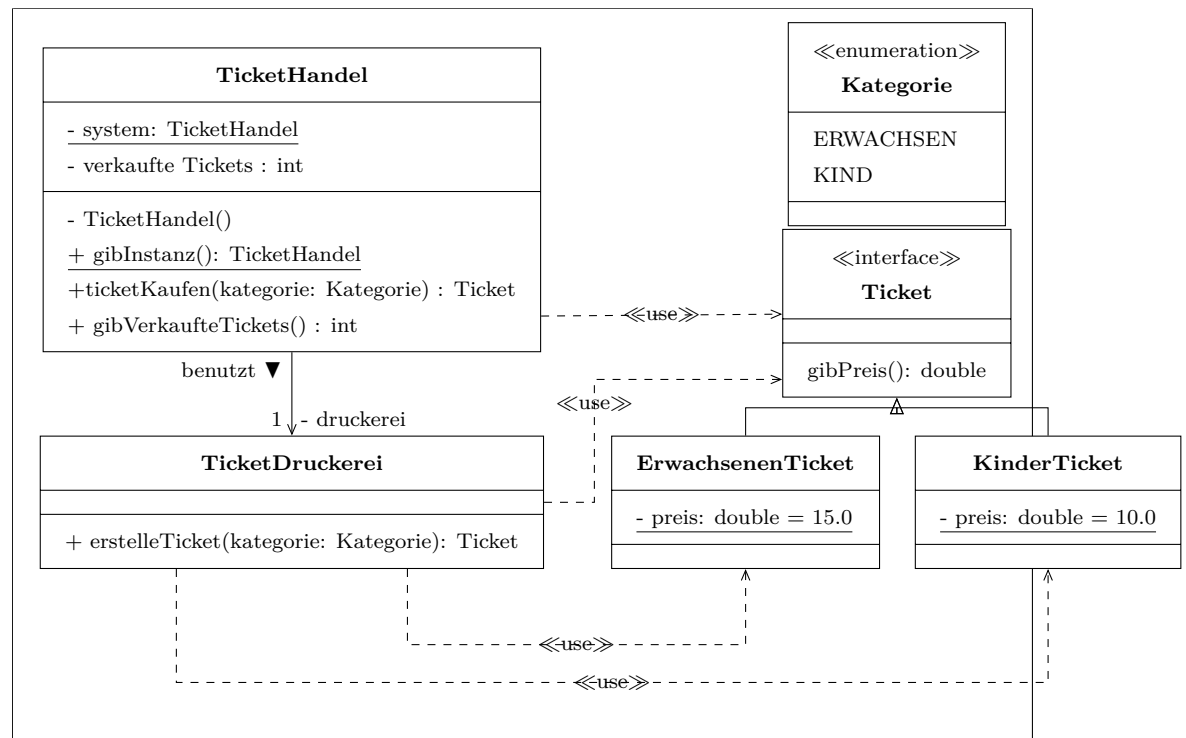
```

2666 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2667 \ProvidesPackage{lehramt-informatik-uml}[2020/06/13 Ein Hüll-Paket um
2668 `tikz-uml', das einige Design-Einstellungen vornimmt und manche
2669 Erweiterung bereitstellt]

2670 \RequirePackage{tikz-uml}
2671 \RequirePackage{tikz-uml-activity}
2672 % Not compatible with wasysym
2673 %\RequirePackage{mathabx}
2674 \RequirePackage{wasysym}
2675 \usetikzlibrary{positioning}

2676 \tikzumlset{
2677   fill class=white!0,
2678   font=\footnotesize,
2679   fill object=white!0,
2680   fill note=white!0,
2681   fill state=white!0,
2682   % Use case
2683   fill usecase=white!0,
2684   fill system=white!0,
2685 }

```



```

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

```

```

2686 \NewDocumentCommand{ \liUmlLeserichtung } { 0{dir=right} m } {
2687   \def\@liDirLeft{}
2688   \def\@liDirRight{ \RIGHTarrow}
2689   \pgfkeys{/lese/dir/.is choice}
2690   \pgfkeys{/lese/dir/up/.code={\def\@liDirRight{ \UParrow}}}
2691   \pgfkeys{/lese/dir/down/.code={\def\@liDirRight{ \DOWNarrow}}}
2692   \pgfkeys{/lese/dir/left/.code={\def\@liDirRight{}\def\@liDirLeft{ \LEFTarrow }}}
2693   \pgfkeys{/lese/dir/right/.code={\def\@liDirRight{ \RIGHTarrow}}}
2694
2695   \def\@liPos{above}
2696   \pgfkeys{/lese/pos/.code={\def\@liPos{##1}}}
2697

```

```

2698 \def\@liDistance{0cm}
2699 \pgfkeys{/lese/distance/.code={\def\@liDistance{##1}}}}
2700
2701 \pgfkeys{/lese/.cd,#1}
2702
2703 \node[\@liPos = \@liDistance of #2-middle] {
2704   \@liDirLeft{\footnotesize#2}\@liDirRight
2705 };
2706 }
2707

```

2.44 vollstaendige-induktion.sty

2708 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2709 \ProvidesPackage{lehramt-informatik-vollstaendige-induktion}[2021/07/01
 2710 Hilfsmakros zum Setzen der Vollständigen Induktion, vor allem die
 2711 Überschriften für die einzelnen Schritte]

Let-Abkürzungen

\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\}
}{
  \{(4\m{n} + 2) \cdot \text{cn}(\m{n})\}
  \{\m{n + 2}\}
}
&\text{\e{Java nach Mathe}}\\
\%
&= \frac{
  \{(4\m{n} + 2) \cdot \text{cn}(\m{n})\}
  \{\m{n + 2}\}
}{
  \{(4n + 2) \cdot \m{(2n)!}\}
  \{(n + 2) \cdot \m{(n + 1)!} \cdot n!\}
}
&\text{\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)}\}
}{
  \{(n + 2) \cdot (n + 1)! \cdot n! \cdot \m{\cdot (n + 1)}\}
}
&\text{\e{\$(n + 1)\$ multipliziert}}\\
\%
&= \frac{
  \{(4n + 2) \cdot \m{(n + 1) \cdot (2n)!}\}
  \{(n + 2) \cdot (n + 1)! \cdot n! \cdot \m{(n + 1) \cdot n!}\}
}{
  \{(n + 2) \cdot (n + 1)! \cdot n! \cdot \m{(n + 1) \cdot n!}\}
}
&\text{\e{umsortiert}}\\
\%
&= \frac{
  \{\m{(2(n + 1))!}\}
  \{\m{(n + 2)! \cdot (n + 1)!}\}
}{
  \{\m{(n + 2)! \cdot (n + 1)!}\}
}
&\text{\e{Hilfsgleichungen verwendet}}\\
\%
&= \frac{
  \{(2(\m{n + 1}))!\}
  \{((\m{n + 1}) + 1)! \cdot (\m{n + 1})!\}
}{
  \{((\m{n + 1}) + 1)! \cdot (\m{n + 1})!\}
}
&\text{\e{\$(n + 1)\$ verdeutlicht}}\\
\end{align*}
```

Lade häufig benötigte Pakete

2712 \RequirePackage{lehramt-informatik-typographie}
 2713 \RequirePackage{lehramt-informatik-mathe}
 2714 \RequirePackage{lehramt-informatik-syntax}
 2715 \ExplSyntaxOn

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

Let-Abkürzung: \let\m=\liInduktionMarkierung

2716 \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

```
2717 \def\liInduktionErklaerung#1{\scriptsize\text{#1}}
```

\liInduktionAnfang

```
2718 \def\liInduktionAnfang{
2719   \liPseudoUeberschrift{Induktionsanfang}
2720
2721   % https://de.wikibooks.org/wiki/Mathe_für_Nicht-Freaks:_Vollständige_Induktion
2722   \liParagraphMitLinien{
2723     Beweise,~dass~ $A(1)$ ~eine~wahre~Aussage~ist.
2724   }
2725 }
```

\liInduktionVoraussetzung

```
2726 \def\liInduktionVoraussetzung{
2727   \liPseudoUeberschrift{Induktionsvoraussetzung}
2728
2729   % https://de.wikibooks.org/wiki/Mathe_für_Nicht-Freaks:_Vollständige_Induktion
2730   \liParagraphMitLinien{
2731     Die~Aussage~ $A(k)$ ~ist~wahr~für~ein~beliebiges~ $k \in \mathbb{N}$ $.
2732   }
2733 }
```

\liInduktionSchritt

```
2734 \def\liInduktionSchritt{
2735   \liPseudoUeberschrift{Induktionsschritt}
2736
2737   % https://de.wikibooks.org/wiki/Mathe_für_Nicht-Freaks:_Vollständige_Induktion
2738   \liParagraphMitLinien{
2739     Beweise,~dass~wenn~ $A(n=k)$ ~wahr~ist,~
2740     auch~ $A(n=k+1)$ ~wahr~sein~muss.
2741   }
2742 }
```

```
2743 \ExplSyntaxOff
```

```
2744
```


2.45 wasserfall.sty

```
2745 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2746 \ProvidesPackage{lehramt-informatik-wasserfall}[2020/06/10]
2747 \RequirePackage{tikz}
2748 \tikzset{wasserfall/.style={
2749   >=stealth,
2750   node distance = 2mm and -8mm,
2751   start chain = A going below right,
2752   every node/.style = {
2753     draw,
2754     text width=24mm,
2755     minimum height=12mm,
2756     align=center,
2757     inner sep=1mm,
2758     fill=white,
2759     drop shadow={fill=black},
2760     on chain=A
2761   },
2762 }}
2763 \usetikzlibrary{chains,positioning,shadows}
2764
```

2.46 wpkalkuel.sty

2765 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2766 \ProvidesPackage{lehramt-informatik-wpkalkuel}[2020/06/13]

Let-Abkürzungen

\let\wp=\liWpKalkuel

\let\equivalent=\liWpEquivalent

\let\erklaerung=\liWpErklaerung

2767 \RequirePackage{amsmath}

2768 \ExplSyntaxOn

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

2769 \def\liWpKalkuelOhneMathe#1#2{

2770 \text{wp}(\texttt{\scriptsize"#1"},\thinspace #2)

2771 }

2772 \def\liWpKalkuel#1#2{

2773 \ifmmode

2774 \liWpKalkuelOhneMathe{#1}{#2}

2775 \else

2776 \$\liWpKalkuelOhneMathe{#1}{#2}\$

2777 \fi

2778 }

\MatheEnv

2779 \def\MatheEnv#1{

2780 \medskip

2781

2782 \hspace{1em}#1

2783

2784 \medskip

2785 }

\Mathe

2786 \def\Mathe#1{

2787 \MatheEnv{#1\$}

2788 }

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

2789 \def\liWpEquivalent#1{

2790 \MatheEnv{\$\equiv\$\hspace{1em}#1\$}

2791 }

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

2792 \newlength{\@Skip@Erklaerung@Reset}

2793 \def\liWpErklaerung#1{

2794 \setlength{\@Skip@Erklaerung@Reset}{\leftskip}

2795 \setlength{\leftskip}{0.5cm}

2796

2797 \par

2798 \noindent

2799 {

2800 \scriptsize

2801 #1

2802 }

2803 \par

2804

2805 \setlength{\leftskip}{\@Skip@Erklaerung@Reset}

2806 }

\liWpErklaerungVerzweigung

```

2807 \def\liWpErklaerungVerzweigung{
2808   $\liWpKalkuelOhneMathe{if~\{-b~\}-then~\{-a1~\}-else~\{-a2~\}}{Q}
2809   \equiv
2810   (b \land \liWpKalkuelOhneMathe{a1}{Q})
2811   \lor
2812   (\neg b \land \liWpKalkuelOhneMathe{a2}{Q})$
2813 }

2814 \ExplSyntaxOff

2815

```

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	\allsectionsfont ... 1219	\Bigl 2658
\# 109	\Alph 1233	\Bigr 2662
\, 331, 386, 1170,	\alph 1233, 1234	\bigskip 48, 364,
1172, 1673, 1674,	\alpha 2573, 2575, 2576,	598, 603, 1549, 1877
1675, 2003, 2473, 2658	2579, 2581, 2582,	\bool 309, 332
\@Skip@Erklaerung@Reset	2583, 2584, 2585,	\bowtie 2318, 2321, 2322, 2323
... 2792, 2794, 2805	2589, 2595, 2596,	\Box 159
\@afterheading 1640	2601, 2602, 2603,	\boxtimes 463
\@afterindentfalse . 1640	2606, 2614, 2615, 2616	
\@liDirLeft 2687, 2692, 2704	\arabic 1233, 2192, 2197,	C
\@liDirRight 2688, 2690,	2202, 2208, 2214, 2220	\c 1316, 1317
2691, 2692, 2693, 2704	\arraystretch 1899	\cdot 1717, 1770, 1781
\@liDistance 2698, 2699, 2703	B	\centerline 1347, 2271, 2293, 2308
\@liPos .. 2695, 2696, 2703	\BeforeBeginEnvironment	\chapter 1223, 1224
\\ .. 596, 619, 620, 623, 2465	\char 1547
624, 627, 628, 720,	\begin .. 617, 668, 683,	\clearpage 1541
721, 722, 829, 858,	718, 742, 789, 821,	\cline 596
860, 886, 895, 940,	836, 856, 866, 882,	\clist 226, 270,
982, 983, 984, 989,	902, 934, 951, 980,	271, 284, 288, 2405
990, 991, 1011,	1005, 1026, 1041,	\columnbreak 2440
1547, 1904, 1962, 1965	1150, 1239, 1314,	\cs 287, 306, 330,
\{ 207, 1107,	1342, 1351, 1358,	331, 368, 380, 1623
1117, 1129, 1130,	1466, 1539, 1591,	\csname 1301, 1304
1135, 1169, 1388,	1596, 1604, 1629,	\cup 1130,
1949, 2348, 2658, 2808	1634, 1645, 1655,	1978, 2582, 2596, 2603
\} 207, 1107,	1659, 1742, 1746,	D
1117, 1129, 1130,	1762, 1785, 1808,	\DeclareMathSymbol ..
1137, 1173, 1389,	1823, 1902, 1903, 1847, 1848
1949, 2348, 2662, 2808	1953, 1992, 2112,	\DecoINERT 2195, 2268, 2269, 2291
_ 27, 35, 44, 46, 287, 306,	2132, 2273, 2295,	\DecoINERTwithPivot .
330, 331, 345, 346,	2309, 2465, 2476, 2662 2211, 2290
352, 355, 358, 368, 380	\begingroup 1540, 1951, 2393	\DecoLEFT 2190, 2289
	\beschriftung 1571, 1575,	\DecoLEFTwithPivot ..
	1579, 1583, 1587, 1589 2205, 2267
_ 1961, 1969	\beta 2573,	\DecoRIGHT 2200, 2292, 2302
A	2576, 2580, 2581,	\DecoRIGHTwithPivot .
\addbibresource 1517, 1518, 1519,	2582, 2585, 2595, 2217, 2270
1520, 1521, 1522,	2596, 2597, 2602, 2603	\definecolor 1221
1523, 1524, 1525, 1526	\bf 2170, 2171, 2172	\delta 65, 107, 165, 207, 1121
\advance 2397	\bfseries .. 475, 1223,	\dh 1675, 2584
\AfterEndEnvironment 2466	1225, 2170, 2176,	
	2178, 2180, 2181, 2550	

<code>\directlua</code>	<code>liProjektSprache</code> <u>1563</u>	<code>\foreach</code> . 1316, 1319, 1326
58, 137, 195, 200,	<code>liQuellen</code> <u>1623</u>	<code>\forestFirst</code> .. 2420, 2423
1102, 1116, 1136,	<code>liRelationenSchemaFormat</code>	<code>\forestLast</code> ... 2421, 2423
1144, 1151, 1156, <u>2352</u>	<code>\forestOget</code> ... 2420, 2421
1934, 1939, 1986,	<code>liRmodell</code> <u>2332</u>	<code>\forestOnes</code> 2433
1993, 2000, 2360,	<code>liUebergangsTabelle</code>	<code>\forestOv</code> 2422, 2423, 2426
2452, 2484, 2489, <u>1899</u>	<code>\forestov</code> . 2412, 2416,
2490, 2494, 2499,	<code>\equiv</code> 2790, 2809	2417, 2420, 2421,
2500, 2504, 2510, 2511	<code>\erzeuge@tiefgestellt</code>	2422, 2423, 2425, 2426
<code>\do</code> 2191, 2196,	. . . 1116, 1117, 1121	<code>\forestset</code> . . . 2408, 2431
2201, 2206, 2212, 2218	<code>\expandafter</code>	<code>\forestSortLevel</code>
<code>\dots</code> 506, 510,	1301, 2226, 2228,	2410, 2418, 2432, 2433
1388, 2119, 2595, 2596	2229, 2230, 2238, 2396	<code>\frac</code> 1719, 1750, 1781, 1796
<code>\DOWNarrow</code> 2691	<code>\ExplSyntaxOff</code>	<code>\fullouterjoin</code> <u>2323</u>
<code>\draw</code> 1321, 1324, 50, 92, 134,	
1327, 2067, 2366, 2369	139, 192, 197, 202,	
	393, 528, 550, 565,	G
E	1097, 1175, 1210,	<code>\g</code> 29, 37,
<code>\edef</code> 1312,	1486, 1501, 1665,	270, 271, 284, 288,
2284, 2288, 2300, 2301	1740, 1927, 2095,	294, 295, 296, 297,
<code>\else</code> 570, 578,	2345, 2406, 2524,	298, 299, 300, 301,
586, 1111, 1125,	2631, 2664, 2743, 2814	302, 303, 304, 307,
1161, 1587, 1690,	<code>\ExplSyntaxOn</code>	311, 312, 313, 316,
1700, 1710, 1724, 22, 61, 102,	318, 319, 320, 321,
1945, 1977, 2062,	135, 160, 193, 198,	322, 323, 334, 335,
2230, 2423, 2425, 2775	223, 470, 534, 551,	336, 337, 345, 346,
<code>\emph</code> 1073,	1070, 1165, 1187,	348, 354, 355, 357,
1369, 1398, 1400, 1546	1479, 1491, 1566,	358, 360, 361, 369,
<code>\empty</code> 1587, 1976	1679, 1909, 2076,	374, 376, 381, 383, 387
<code>\emptyset</code>	2333, 2403, 2451,	<code>\Gamma</code> . 106, 164, 207, 1130
1891, 2590, 2611, 2644	2547, 2641, 2715, 2768	<code>\gappto</code> 2428
<code>\end</code> . . . 656, 679, 704,	F	<code>\geq</code> 1395,
739, 773, 805, 833,	<code>\faCheckSquareO</code> 2643	2108, 2113, 2129, 2133
849, 863, 871, 898,	<code>\faCircleThin</code> 1089	
927, 948, 973, 998,	<code>\faG</code> 1083	H
1018, 1038, 1056,	<code>\fancyfoot</code>	<code>\hbox</code> 2318
1152, 1241, 1330,	1245, 1246, 1247,	<code>\headrulewidth</code> . 1249, 1499
1354, 1355, 1362,	1495, 1496, 1497, 1498	<code>\hfill</code> 1766, 1772, 1777, 2651
1471, 1544, 1593,	<code>\fancyhead</code>	<code>\hinweis</code> <u>1562</u>
1601, 1620, 1636,	1244, 1492, 1493, 1494	<code>\hline</code> 1904
1637, 1651, 1662,	<code>\faSquareO</code> 1077	<code>\href</code> 370,
1663, 1744, 1757,	<code>\fi</code> 572, 580, 588, 1113,	1183, 1671, 1834, 2479
1782, 1805, 1820,	1127, 1163, 1590,	<code>\hspace</code> .. 2070, 2782, 2790
1830, 1906, 1907,	1692, 1702, 1712,	<code>\ht</code> 2319
1955, 1994, 2122,	1726, 1947, 1979,	
2142, 2273, 2295,	2064, 2231, 2423,	I
2309, 2466, 2480, 2662	2424, 2427, 2429, 2777	<code>\i</code> 1326, 1327
<code>\endsname</code> 1301, 1304	<code>\fontspec</code> 1219	<code>\ifcase</code> 2227
<code>\endgroup</code> 1543, 1956, 2398	<code>\footcite</code>	<code>\ifmmode</code> 568, 576, 584,
<code>\enspace</code> 2648, 2650	680, 702, 749, 772,	1109, 1123, 1159,
environments:	804, 899, 926, 972,	1688, 1698, 1708,
<code>liAdditum</code> <u>1594</u>	1367, 1370, 1377,	1722, 1943, 2060, 2773
<code>liAHuelle</code> <u>1950</u>	1382, 1387, 1391,	<code>\IfNoValueTF</code>
<code>liAntwort</code> <u>1565</u>	1397, 1402, <u>1506</u> ,	... 1597, 1667, 1671
<code>liDiagramm</code> <u>1653</u>	1758, 1759, 1914, 2125	<code>\ifnum</code> 2154,
<code>liEinbettung</code> <u>1564</u>	<code>\footnote</code> 1667, 1671	2395, 2411, 24

\liEntwurfsAbstrakteFabrikUml	616, 664	\liEntwurfsZustandUml	1025, 1059	liKontrollflussgraph (environment)	1465
\liEntwurfsAdapter	712	\liEpsilon	1115	\liKontrollKnotenPfad	1479
\liEntwurfsAdapterAkteure	682, 714	\liErAttribute	1073, 1087, 1089	\liKontrollTextzeileKnoten	1478, 1483
\liEntwurfsAdapterCode	706, 715	\liErDatenbankName	1092	\liKurzeTabellenLinie	596
\liEntwurfsAdapterUml	667, 713	\liErEntity	1071, 1075, 1077	\liLadeAllePakete	228
\liEntwurfsBeobachter	783	\liErledigt	2643	\liLadePakete	54, 57, 224, 229, 472, 533, 1106, 1337, 1853, 1933, 2075, 2546
\liEntwurfsBeobachterAkteure	741, 785	\liErMpAttribute	1086	\liLatexCode	2474
\liEntwurfsBeobachterCode	775, 786	\liErMpEntity	1074	\liLeereZelle	1891
\liEntwurfsBeobachterUml	717, 784	\liErMpRelationship	1080	liLernkartei (environ- ment)	1643
\liEntwurfsDekorierer	815	\liErRelationship	1072, 1081, 1083	\liLinksReduktion	1958
\liEntwurfsDekoriererAkteure	817	\liExamensAufgabe	6	\liLinksReduktionInline	1967, 1972
\liEntwurfsDekoriererCode	807, 818	\liExamensAufgabeA	15	\liMasterExkurs	1822
\liEntwurfsDekoriererUml	788, 816	\liExamensAufgabeTA	12	\liMasterFaelle	1761, 1829
\liEntwurfsEinfacheFabrik	851	\liExamensAufgabeTTA	9	\liMasterFallRechnung	1807
\liEntwurfsEinfacheFabrikAkteure	835, 853	liExkurs (environment)	1602	\liMasterVariablen	1741, 1824
\liEntwurfsEinfacheFabrikUml	820, 852	\liFalsch	464	\liMasterVariablenDeklaration	1784
\liEntwurfsEinzelstueck	876	\liFlaci	1176	\liMasterWolframLink	1832
\liEntwurfsEinzelstueckAkteure	865, 878	\liFremd	2331	\liMenge	71, 72, 74, 113, 114, 115, 119, 171, 172, 173, 177, 1107, 1156, 1195, 1196
\liEntwurfsEinzelstueckCode	873, 879	\liFunktionaleAbhaengigkeit	1975, 1978, 1985	\liMengeOhneMathe	1107, 1110, 1112
\liEntwurfsEinzelstueckUml	855, 877	\liFunktionaleAbhaengigkeiten	1988	\liMinimierungErklaerung	1913
\liEntwurfsErbauer	929	\liFussnote	1854, 1856	\liMinispracheDatei	2517
\liEntwurfsErbauerAkteure	901, 931	\liFussnoteDreiText	1868, 1886	\linespread	2339
\liEntwurfsErbauerUml	881, 930	\liFussnoteEinsText	1860, 1880	\liNichtsZuTun	2644
\liEntwurfsFabrikmethode	975, 1000	\liFussnoteLink	1670	\liO	1704, 1732
\liEntwurfsFabrikmethodeAkteure	950, 977	\liFussnoteNoten	1876	\liOmega	1694, 1738
\liEntwurfsFabrikmethodeUml	933, 976	\liFussnoteUrl	1019, 1666	\liOmegaOhneMathe	1694, 1699, 1701
\liEntwurfsKompositum	1000	\liFussnoteVierText	1872, 1889	\liOOhneMathe	1704, 1709, 1711
\liEntwurfsKompositumAkteure	1002	\liFussnoteZweiText	1864, 1883	\liParagraphMitLinien	521, 1915, 2624, 2645, 2722, 2730, 2738
\liEntwurfsKompositumUml	979, 1001	\liGeschweifteKlammern	1148, 1989, 2655	\liPetriErreichKnotenDrei	2069
\liEntwurfsModellPraesentation	1021	\liGrammatik	1187	\liPetriErreichTransition	2066
\liEntwurfsModellPraesentationAkteure	1023	liGraphenFormat (envi- ronment)	1293	\liPetriSetzeSchluessel	2011
\liEntwurfsModellPraesentationUml	1004, 1022	\liHanoi	1300	\liPetriTransitionsName	2058, 2070
\liEntwurfsZustand	1058	\liHaskellCode	2520	\liPetriTransitionsNameOhneMathe	2058, 2061, 2063
\liEntwurfsZustandAkteure	1040, 1060	\liHaskellDatei	2521	\liPetriTransPfeile	2070
		\liInduktionAnfang	2718	\liPolynomiellReduzierbar	1357
		\liInduktionErklaerung	2717	\liPotenzmenge	1116, 1120, 2081
		\liInduktionMarkierung	2716		
		\liInduktionSchritt	2734		
		\liInduktionVoraussetzung	2726		
		\liJavaCode	2473		
		\liJavaDatei	614, 2482		
		\liJavaExamen	2502		
		\liJavaTestAkteure	2492		
		liKasten (environment)	1238		
		\liKellernAngUml	102		
		\liKellerKante	140		
		\liKellerUebergang	135, 141		
		\liKontrollCode	1477		

<code>\liPotenzmengeOhneMathe</code>	<code>\liTOhneMathe</code>	M
. . . 1117, 1118, 1119	. . . 1714, 1723, 1725	<code>\makeatletter</code> . . 1639, 2391
<code>\liPrimaer</code> 2330	<code>\liTuringKante</code> 203	<code>\makeatother</code> . . 1641, 2399
<code>\liProblemBeschreibung</code>	<code>\liTuringLeerzeichen</code>	<code>\marginpar</code>
. 1341 159, 167	1076, 1082, 1088, 1546
<code>\liProblemClique</code> . . . 1364	<code>\liTuringMaschine</code> . . . 160	<code>\mathbb</code> . . 1395, 2139, 2731
<code>\liProblemName</code>	<code>\liTuringUeberfuehrung</code>	<code>\mathbin</code> . 2321, 2322, 2323
. 1340, 1347, 206	<code>\mathcal</code> . . 1705, 2602,
1359, 1361, 1374,	<code>\liTuringUebergaenge</code>	2607, 2609, 2610, 2611
1385, 1386, 1394, 1395 198, 204	<code>\Mathe</code> 2786
<code>\liProblemSat</code> 1393	<code>\liTuringUebergangZelle</code>	<code>\MatheEnv</code> 2779, 2787, 2790
<code>\liProblemSubsetSum</code> 193	<code>\mathord</code> 1847, 1848
. 1384, 1393	<code>\liUeberfuehrungsFunktion</code>	<code>\mdfsetup</code> 1237,
<code>\liProblemVertexCover</code> 1121	1572, 1576, 1580, 1584
. 1364, 1372	<code>\liUeberfuehrungsFunktionOhneMathe</code>	<code>\medskip</code>
<code>\liProduktionen</code> 1155, 1197	. . . 1121, 1124, 1126	1349, 1557, 1559,
<code>liProduktionsRegeln</code>	<code>liUebergangsTabelle</code>	1633, 1661, 2337,
(environment) 1145	(environment) 1899	2343, 2653, 2780, 2784
<code>liProjektSprache</code> (envi-	<code>\liUeberschriftDreiecksTabelle</code>	<code>\memph</code> 1546
ronment) 1563 1909	<code>\mintinline</code> 2473,
<code>\liPseudoUeberschrift</code>	<code>\liUmlLeserichtung</code> . 2686	2474, 2513, 2520, 2525
. 1548,	<code>\liVertauschen</code> 2359	<code>\mkern</code> . . . 2321, 2322, 2323
1598, 1599, 1901,	<code>\liWortInSprache</code> 597	<code>\mlq</code> 1845, 1847
1911, 2719, 2727, 2735	<code>\liWortNichtInSprache</code> 602	<code>\mrq</code> 1845, 1848
<code>\liPumpingKontextfrei</code>	<code>\liWpEquivalent</code> 2789	<code>\msg</code> 39, 391
. 2127	<code>\liWpErklaerung</code> 2792	<code>\myList</code>
<code>\liPumpingRegulaer</code> . 2106	<code>\liWpErklaerungVerzweigung</code>	2412, 2413, 2414, 2417
<code>liQuellen</code> (environment) 2807	<code>\myNodes</code> 2401,
. 1623	<code>\liWpKalkuel</code> 2769	2416, 2422, 2426, 2428
<code>\liRechtsReduktionInline</code>	<code>\liWpKalkuelOhneMathe</code>	N
. 1972 2769, 2774,	<code>\NeedsTeXFormat</code>
<code>\liRekursionsGleichung</code>	2776, 2808, 2810, 2812	1, 19, 52, 220, 398,
. 1728, 1788	<code>\liZustandsBuchstabe</code>	459, 466, 530, 593,
<code>\liRelation</code> 1999 1131,	608, 1063, 1099,
<code>liRelationenSchemaFormat</code>	1140, 1142, 1160, 1162	1212, 1252, 1261,
(environment) 2352	<code>\liZustandsBuchstabeGross</code>	1266, 1295, 1333,
<code>\liRelationMenge</code> . . . 2346	. . . 1132, 1141, 1143	1405, 1488, 1503,
<code>\liRichtig</code> 463	<code>\liZustandsmenge</code> . . . 1119	1509, 1531, 1677,
<code>liRmodell</code> (environment)	<code>\liZustandsmengeNr</code> . .	1837, 1850, 1929,
. 2332 1133, 2090	2006, 2072, 2097,
<code>\liRundeKlammer</code> . 1681,	<code>\liZustandsmengeNrGross</code>	2102, 2146, 2312,
1685, 1695, 1705, 1719 1141	2325, 2354, 2435,
<code>\liSetzeAufgabenTitel</code> . 25	<code>\liZustandsMengenSammlung</code>	2442, 2447, 2527,
<code>\liSortierMarkierung</code> 2371 2077	2542, 2633, 2637,
<code>\liSortierPfeil</code> 2365	<code>\liZustandsMengenSammlungNr</code>	2666, 2708, 2745, 2765
<code>\liSortierPfeilUnten</code> 2368 2086	<code>\neg</code> 2812
<code>\liSpaltenUmbruch</code> . . 2440	<code>\liZustandsmengeOhneMathe</code>	<code>\negthinspace</code> 1682
<code>\liSqlCode</code> 2525 1119	<code>\newcounter</code> . . . 2166, 2167
<code>\listen@punkt</code> . . 1623, 1635	<code>\liZustandsname</code> 1142	<code>\NewDocumentCommand</code> .
<code>\liStrich</code> 1339	<code>\liZustandsnameGross</code>	62, 103, 140, 161,
<code>\liSyntheseErklaerung</code>	. . . 1143, 2078, 2087	203, 224, 535, 552,
. 2564, 2629	<code>\liZustandsnameTiefgestellt</code>	597, 602, 1166,
<code>\liSyntheseUeberErklaerung</code> 1158	1188, 1357, 1480,
. 2627	<code>\liZustandsPaar</code> 1893	1507, 1666, 1670,
<code>\liSyntheseUeberschrift</code>	<code>\liZustandsPaarVariablenName</code>	1988, 1999, 2066,
. 2548, 2628	. . . 1892, 1895, 1896	2482, 2492, 2502,
<code>\liT</code> 1714, 1729, 1743, 1804	<code>\llap</code> 2319	2514, 2517, 2521, 2686
<code>\liTeilen</code> 1938	<code>\log</code> 1732,	<code>\NewDocumentEnvironment</code>
<code>\literatur</code> 1505, 1529	1735, 1738, 1764, 1770	1145, 1238, 1293,
<code>\liTheta</code> 1684,	<code>\loop</code> 2152	1465, 1563, 1564,
1735, 1764, 1770, 1775	<code>\lor</code> 2811	1567, 1594, 1602,
<code>\liThetaOhneMathe</code> . . .	<code>\ltimes</code> 1840	1624, 1643, 1653,
. . . 1684, 1689, 1691		1900, 1950, 2334, 2352

<code>\setsansfont</code>	1217	108, 109, 110, 113,	1006, 1007, 1008,	
<code>\setul</code>	2331	114, 115, 116, 117,	1027, 1028, 1029, 1030	
<code>\sffamily</code>	476,	118, 119, 162, 163,	<code>\umlddep</code>	947
1223, 1225, 1327, 2551		164, 165, 166, 167,	<code>\umlHVHaggreg</code>	737, 803, 997
<code>\shoveleft</code>	1959	168, 171, 172, 173,	<code>\umlinherit</code>	
<code>\shoveright</code>	1963	174, 175, 176, 177,	676, 727, 892, 937, 945	
<code>\Sigma</code>	64, 105,	285, 289, 307, 311,	<code>\umlnote</code> ...	678, 894, 1037
163, 1129, 1130, 1190		312, 313, 316, 321,	<code>\umlreal</code>	674, 735
<code>\sigma</code>	499, 501, 502	322, 323, 334, 335,	<code>\umlsimpleclass</code> ..	633,
<code>\SLASH</code>	1547	336, 337, 348, 354,	634, 635, 639, 641,	
<code>\small</code>	1656	357, 360, 369, 383,	642, 643, 669, 822,	
<code>\sort</code>	2405	536, 539, 544, 545,	823, 824, 883, 935, 936	
<code>\sortList</code>	2404, 2413	553, 554, 557, 558,	<code>\umlstatic</code>	829, 858
<code>\square</code>	464	1168, 1189, 1190,	<code>\umluniaggreg</code>	890
<code>\stepcounter</code>	2192, 2197,	1191, 1192, 1195,	<code>\umluniasoc</code>	653,
2202, 2205, 2207,		1196, 1197, 1198, 1715	675, 891, 1015, 1016	
2211, 2213, 2217, 2219		<code>\tmp</code>	<code>\umlVHuniassoc</code> ..	654, 655
<code>\str</code> ..	477, 486, 1569,	<code>\TmpPlaceEight</code>	<code>\umlVHVdep</code>	647,
2082, 2091, 2552, 2565		<code>\TmpPlaceFive</code>	648, 650, 651, 831, 832	
<code>\string</code>	1961, 1969	<code>\TmpPlaceFour</code>	<code>\umlVHVinherit</code>	
<code>\StrSubstitute</code> ..	2412, 2414	<code>\TmpPlaceNine</code> 630, 631, 636,	
<code>\strut</code>	1786, 1790,	<code>\TmpPlaceOne</code>	637, 644, 645, 800,	
1794, 1798, 1802, 2440		<code>\TmpPlaceSeven</code>	801, 825, 826, 995, 996	
<code>\subsetq</code>	2576, 2609, 2616	<code>\TmpPlaceSix</code>	<code>\umlVHVreal</code>	
		<code>\TmpPlaceTen</code>	794, 795, 1032, 1033	
T		<code>\TmpPlaceThree</code>	<code>\UParrow</code>	2690
<code>\tableofcontents</code> ...	1542	<code>\TmpPlaceTwo</code>	<code>\url</code>	1667
<code>\text</code>	77, 79,	<code>\TmpScale</code>	<code>\usemintedstyle</code>	2464
182, 1941, 2717, 2770		<code>\TmpTransitionEight</code> .	<code>\usetikzlibrary</code> ..	56,
<code>\textbf</code> ...	1071, 1365,	2019, 2040	403, 1067, 1271,
1374, 1385, 1394,		<code>\TmpTransitionFive</code>	1408, 2009, 2317,
1551, 1558, 1589,		2016, 2037	2358, 2389, 2675, 2763
1617, 1632, 1648, 1904		<code>\TmpTransitionFour</code> ..		
<code>\textcolor</code>	1477, 2716	2015, 2036	
<code>\textit</code>		<code>\TmpTransitionNine</code> ..		
940, 982, 983, 984,		2020, 2041	
985, 1658, 1949, 2003		<code>\TmpTransitionOne</code>	2012, 2033
<code>\textsc</code>	1340	2018, 2039	
<code>\textsf</code>	1551, 1632	<code>\TmpTransitionSeven</code> .		
<code>\textstyle</code>	1750, 1781	2017, 2038	
<code>\texttt</code>	1094, 1340, 1474,	<code>\TmpTransitionSix</code> ...		
1475, 1476, 1477, 2770		2021, 2042	
<code>\thepage</code>	1247, 1496	<code>\TmpTransitionTen</code> ...		
<code>\theparagraph</code>	1225	2014, 2035	
<code>\Theta</code>	1685	<code>\TmpTransitionThree</code> .		
<code>\thinspace</code>	2770	2013, 2034	
<code>\tikz</code>	1478	<code>\TmpTransitionTwo</code> ...		
<code>tikz: bbaum</code>	23	2044	
<code>tikz: li binaer baum</code>	21	<code>\TmpX</code>	2045	
<code>\tikzchildnode</code>	419	<code>\TmpY</code>	1494	
<code>\tikzparentnode</code>	419	<code>\today</code>	2332	
<code>\tikzset</code>	96,	<code>\ttfamily</code>		
143, 209, 404, 430,				
1272, 1409, 2048,		U		
2174, 2378, 2532, 2748		<code>\ul</code>	1072, 2330, 2331	
<code>\tikzumlset</code>	2676	<code>\umlaggreg</code>	1035	
<code>\times</code>	207	<code>\umlassoc</code>	1017	
<code>\tiny</code>	1077, 1083,	<code>\umlclass</code>	618,	
1089, 1477, 1546, 2477		622, 626, 670, 671,		
<code>\titleformat</code> ..	1223, 1225	672, 719, 724, 729,		
<code>\titlespacing</code>	1224	732, 790, 791, 792,		
<code>\tl</code> .				