

lehramt-informatik

Hermine Bschlangaul *

August 13, 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	Hilsmakros für Zeitpunkt-Tabelle	28
2.9	cyk-algorithmus.sty	30
2.10	entwurfsmuster.sty	31
2.10.1	Abstrakte Fabrik	31
2.10.2	Adapter	32
2.10.3	Beobachter	33
2.10.4	Dekorierer	34
2.10.5	Einzelstück	35
2.10.6	Erbauer	36
2.10.7	Fabrikmethode	37
2.10.8	Kompositum	38
2.10.9	Modell-Präsentation-Steuerung	38
2.10.10	Zustand	39
2.11	er.sty	41
2.12	formale-sprachen.sty	43
2.13	formatierung.sty	47
2.13.1	Schriftarten / Typographie	47
2.13.2	Farben	47
2.13.3	Überschriften	47
2.13.4	Listen	47
2.13.5	Kasten	47
2.13.6	Header	47

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

2.14	gantt.sty	48
2.15	grafik.sty	49
2.16	graph.sty	50
2.17	hanoi.sty	52
2.18	komplexitaetstheorie.sty	53
2.19	kontrollflussgraph.sty	55
2.20	literatur-dummy.sty	57
2.21	literatur.sty	58
2.22	makros.sty	59
2.23	master-theorem.sty	63
2.24	mathe.sty	67
2.25	minimierung.sty	68
2.26	normalformen.sty	71
2.27	petri.sty	73
2.28	potenzmengen-konstruktion.sty	75
2.29	pseudo.sty	77
2.30	pumping-lemma.sty	78
2.31	quicksort.sty	79
2.32	relationale-algebra.sty	82
2.33	rmodell.sty	83
2.34	sortieren.sty	84
2.35	spalten.sty	86
2.36	struktogramm.sty	87
2.37	syntax.sty	88
2.38	syntaxbaum.sty	90
2.39	synthese-algorithmus.sty	91
2.40	tabelle.sty	94
2.41	typographie.sty	95
2.42	uml.sty	96
2.43	vollstaendige-induktion.sty	98
2.44	wasserfall.sty	99
2.45	wpkalkuel.sty	100

3 Index 101

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}{1}{
  EPSILON, KELLERBODEN, EPSILON;
  1, KELLERBODEN, KELLERBODEN;
  0, KELLERBODEN, KELLERBODEN;
}
\end{tikzpicture}
```



```
\liKellerAutomat \liKellerAutomat[\langle automaten-name \rangle]
\{ \langle zustaeende=Z, alphabet=\Sigma, kelleralphabet=\Gamma, delta=\delta, start=z_0, kellerboden=\#, ende=E \rangle \}

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

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

```
102 \ExplSyntaxOn
103 \NewDocumentCommand {\liKellerAutomat} { O{K} m } {
104   \tl_set:Nn \l_zustaeende_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     zustaeende .code:n = {\tl_set:Nn \l_zustaeende_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]{\langle zustand-oder-lese \rangle}{\langle schreibe \rangle}{\langle richtung \rangle}
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-qtreet.]

401 \RequirePackage{tikz}

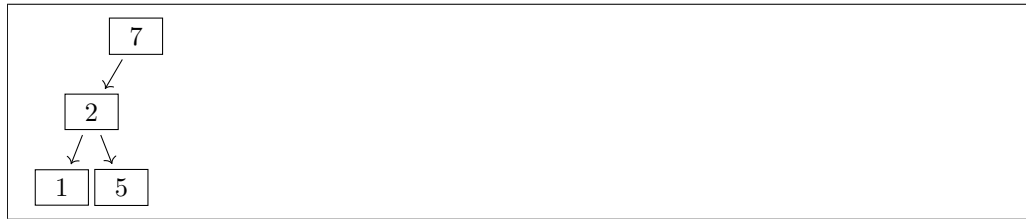
    für li binaer baum
402 \RequirePackage{tikz-qtreet}

    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}

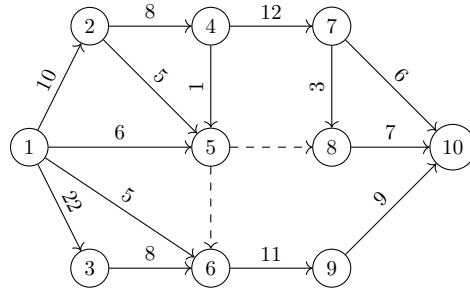
```

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}{}
\liCpmVorgang[schein]{5}{8}{}
\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):  $1_{(2 \rightarrow 3)}$ 
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):  $1_{(\rightarrow 2)}$ 
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):  $1_{(\leftarrow 2)}$ 
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
```

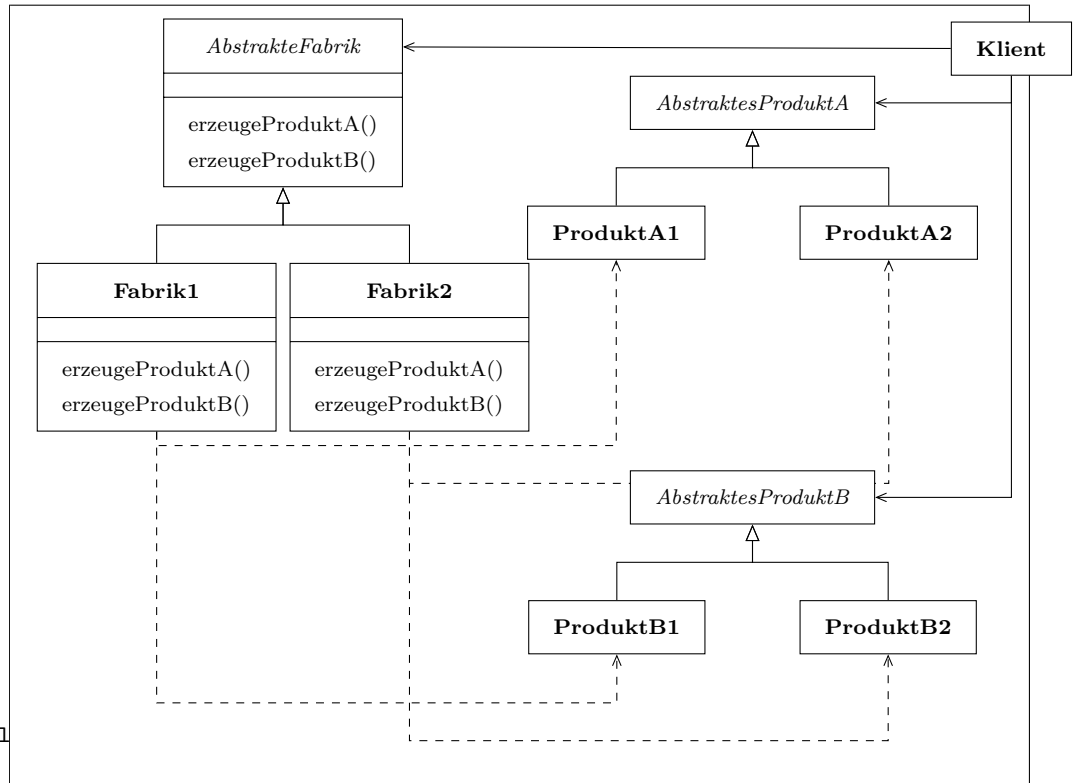
2.10 entwurfsmuster.sty

```

608 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
609 \ProvidesPackage{lehramt-informatik-entwurfsmuster}[2021/05/06 Hilfsmakros
610 zum Setzen von Entwurfsmuster/Design Patterns]
611
612 \RequirePackage{lehramt-informatik-uml}

```

2.10.1 Abstrakte Fabrik



```

613 \def\liEntwurfsAbstrakteFabrikUml{
614   \begin{tikzpicture}
615     \umlclass[type=abstract]{AbstrakteFabrik}{-}{-}{
616       erzeugeProduktA()\n
617       erzeugeProduktB()\n
618     }
619     \umlclass[below left=1cm and -1.5cm of AbstrakteFabrik]{Fabrik1}{-}{-}{
620       erzeugeProduktA()\n
621       erzeugeProduktB()\n
622     }
623     \umlclass[below right=1cm and -1.5cm of AbstrakteFabrik]{Fabrik2}{-}{-}{
624       erzeugeProduktA()\n
625       erzeugeProduktB()\n
626     }
627     \umlVHVinherit{Fabrik1}{AbstrakteFabrik}
628     \umlVHVinherit{Fabrik2}{AbstrakteFabrik}
629
630     \umlsimpleclass[right=3cm of AbstrakteFabrik,type=abstract]{AbstraktesProduktA}
631     \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktA]{ProduktA1}
632     \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktA]{ProduktA2}
633     \umlVHVinherit{ProduktA1}{AbstraktesProduktA}
634     \umlVHVinherit{ProduktA2}{AbstraktesProduktA}
635
636     \umlsimpleclass[above right=0cm and 1cm of AbstraktesProduktA]{Klient}
637
638     \umlsimpleclass[below=4.5cm of AbstraktesProduktA,type=abstract]{AbstraktesProduktB}
639     \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktB]{ProduktB1}
640     \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktB]{ProduktB2}

```

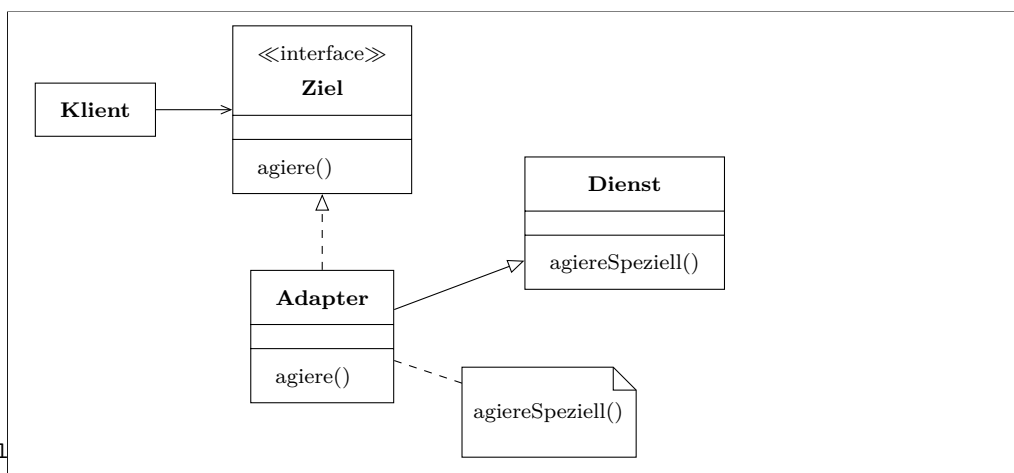
```

641 \umlVHVinherit{ProduktB1}{AbstraktesProduktB}
642 \umlVHVinherit{ProduktB2}{AbstraktesProduktB}
643
644 \umlVHVdep[arm1=-1.3cm,arm2=-1cm]{Fabrik1}{ProduktA1}
645 \umlVHVdep[arm2=-1cm]{Fabrik1}{ProduktB1}
646
647 \umlVHVdep[arm1=-1.8cm,arm2=-1.5cm]{Fabrik2}{ProduktA2}
648 \umlVHVdep[arm2=-1.5cm]{Fabrik2}{ProduktB2}
649
650 \umluniassoc[anchor2=25]{Klient}{AbstrakteFabrik}
651 \umlVHuniassoc[arm1=-1cm]{Klient}{AbstraktesProduktA}
652 \umlVHuniassoc{Klient}{AbstraktesProduktB}
653 \end{tikzpicture}
654 }

```

2.10.2 Adapter

\liEntwurfsAdapterUml



```

655 \def\liEntwurfsAdapterUml{
656 \begin{tikzpicture}
657 \umlsimpleclass[x=1,y=3]{Klient}{}{}
658 \umlclass[x=4,y=3,type=interface]{Ziel}{}{agiere()}
659 \umlclass[x=4,y=0]{Adapter}{}{agiere()}
660 \umlclass[x=8,y=1.5]{Dienst}{}{agiereSpeziell()}
661
662 \umlreal{Adapter}{Ziel}
663 \umluniassoc{Klient}{Ziel}
664 \umlinherit{Adapter}{Dienst}
665
666 \umlnote[x=7,y=-1,width=2cm]{Adapter}{agiereSpeziell()}
667 \end{tikzpicture}
668 \footcite[so ähnlich wie GoF]{wiki:adapter}
669 }

```

\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.

```

670 \def\liEntwurfsAdapterAkteure{
671 \begin{description}
672
673 \item[Ziel (Target)]

```

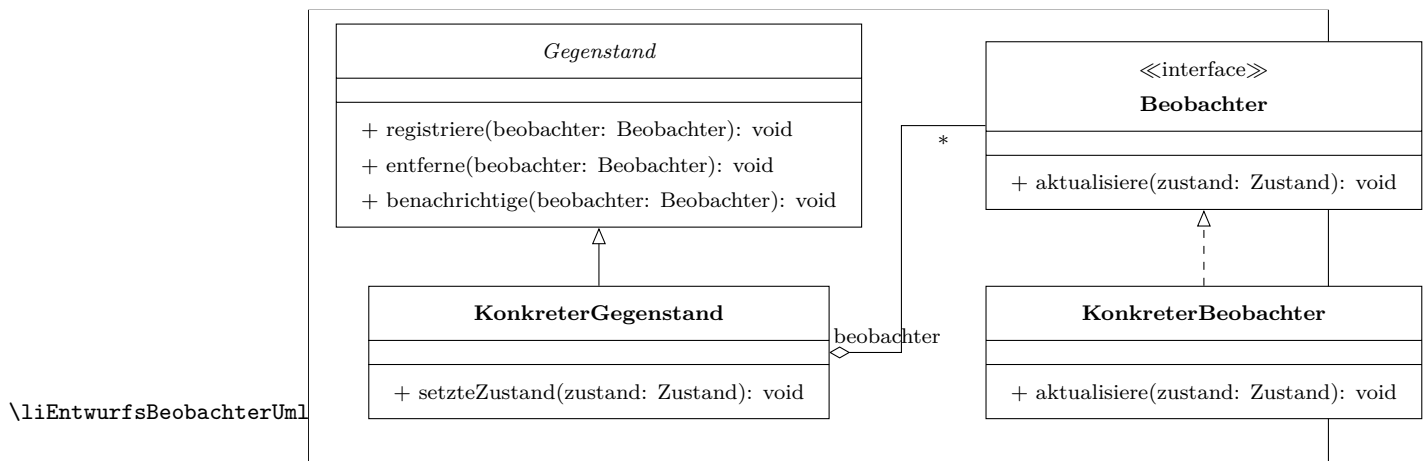


```

674
675 Das Ziel definiert die Schnittstelle, die der Klient nutzen kann.
676
677 \item[Klient (Client)]
678
679 Der Klient nutzt Dienste über inkompatible Schnittstellen und greift
680 dabei auf adaptierte Schnittstellen zurück.
681
682 \item[Dienst (Adaptee)]
683
684 Der Dienst bietet wiederzuverwendende Dienstleistungen mit fest
685 definierter Schnittstelle an.
686
687 \item[Adapter]
688
689 Der Adapter adaptiert die Schnittstelle des Dienstes auf die
690 Schnittstelle zum Klienten.\footcite{wiki:adapter}
691
692 \end{description}
693 }

```

2.10.3 Beobachter



```

694 \def\liEntwurfsBeobachterUml{
695   \begin{tikzpicture}
696     \umlclass[x=0,y=0,type=abstract]{Gegenstand}{-}{
697       + registriere(beobachter: Beobachter): void\\
698       + entferne(beobachter: Beobachter): void\\
699       + benachrichtige(beobachter: Beobachter): void\\
700     }
701     \umlclass[x=0,y=-3]{KonkreterGegenstand}{-}{
702       + setzteZustand(zustand: Zustand): void
703     }
704     \umlinherit{KonkreterGegenstand}{Gegenstand}
705
706     \umlclass[x=8,y=0,type=interface]{Beobachter}{-}{
707       + aktualisiere(zustand: Zustand): void
708     }
709     \umlclass[x=8,y=-3]{KonkreterBeobachter}{-}{
710       + aktualisiere(zustand: Zustand): void
711     }
712     \umlreal{KonkreterBeobachter}{Beobachter}
713
714     \umlHVHaggreg[arg1=beobachter,pos1=0.8,mult2=*,pos2=2.5]
715     {KonkreterGegenstand}{Beobachter}
716   \end{tikzpicture}
717 }

```

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)

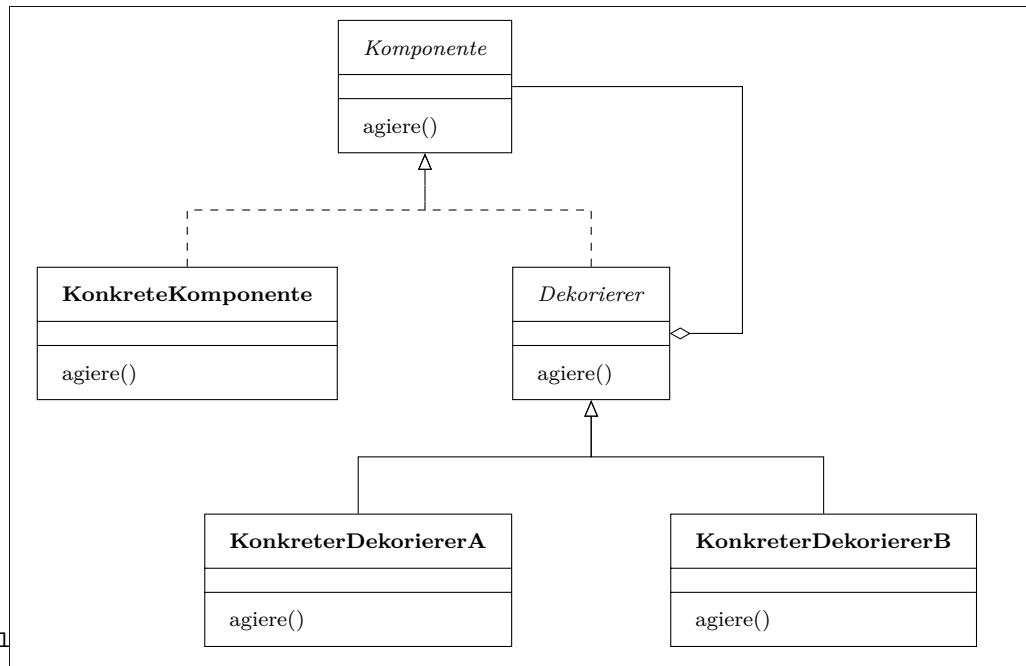
Ein konkretes Subjekt (konkretes, beobachtbares Objekt) speichert den relevanten Zustand und benachrichtigt alle Beobachter bei Zustandsänderungen über deren Aktualisierungsschnittstelle. Es verfügt über eine Schnittstelle zur Erfragung des aktuellen Zustands.

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.

```

718 \def\liEntwurfsBeobachterAkteure{
719   \begin{description}
720     \item[Gegenstand / Subjekt (Subject / Observable)]
721
722     Ein Subjekt (beobachtbares Objekt, auf Englisch publisher, also
723     „Veröffentlicher“, genannt) hat eine Liste von Beobachtern, ohne
724     deren konkrete Typen zu kennen. Es bietet eine Schnittstelle zur An-
725     und Abmeldung von Beobachtern und eine Schnittstelle zur
726     Benachrichtigung von Beobachtern über Änderungen an.\footcite[Seite
727     251]{gof}
728
729     \item[Beobachter (Observer)]
730
731     Die Beobachter (auf Englisch auch subscriber, also „Abonnent“,
732     genannt) definieren eine Aktualisierungsschnittstelle.
733
734     \item[konkreter/s Gegenstand / Subjekt (ConcreteSubject / ConcreteObservable)]
735
736     Ein konkretes Subjekt (konkretes, beobachtbares Objekt) speichert
737     den relevanten Zustand und benachrichtigt alle Beobachter bei
738     Zustandsänderungen über deren Aktualisierungsschnittstelle. Es
739     verfügt über eine Schnittstelle zur Erfragung des aktuellen
740     Zustands.
741
742     \item[Konkrete Beobachter (ConcreteObserver)]
743
744     Konkrete Beobachter verwalten die Referenz auf ein konkretes
745     Subjekt, dessen Zustand sie beobachten und speichern und dessen
746     Zustand konsistent ist. Sie implementieren eine
747     Aktualisierungsschnittstelle unter Verwendung der
748     Abfrageschnittstelle des konkreten Subjekts.
749     \footcite{wiki:beobachter}
750   \end{description}
751 }
```

2.10.4 Dekorierer



\liEntwurfsDekoriererUml

```

752 \def\liEntwurfsDekoriererUml{
753   \begin{tikzpicture}
754     \umlclass[type=abstract]{Komponente}{-}{agiere()}
755     \umlclass[below left=1.5cm and 0cm of Komponente]{KonkreteKomponente}{-}{agiere()}
756     \umlclass[below right=1.5cm and 0cm of Komponente,type=abstract]{Dekorierer}{-}{agiere()}
757
758     \umlVHVreal{KonkreteKomponente}{Komponente}
759     \umlVHVreal{Dekorierer}{Komponente}
760
761     \umlclass[below left=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererA}{-}{agiere()}
762     \umlclass[below right=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererB}{-}{agiere()}
763
764     \umlVHVinherit{KonkreterDekoriererA}{Dekorierer}
765     \umlVHVinherit{KonkreterDekoriererB}{Dekorierer}
766
767     \umlHVVHaggreg[arm1=2cm]{Dekorierer}{Komponente}
768     \footcite{wiki:dekorierer}
769   \end{tikzpicture}
770 }
  
```

2.10.5 Einzelstück

\liEntwurfsEinzelstueckUml Quelle nach der deutschen Wikipedia



```

771 \def\liEntwurfsEinzelstueckUml{
772   \begin{tikzpicture}
773     \umlclass{Einzelstück}{
774       \umlstatic{- instanz: Einzelstück}\\
775     }{
776       - Einzelstück()\\
777       + gibInstanz(): Einzelstück
778     }
779   \end{tikzpicture}
  
```

780 }

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.

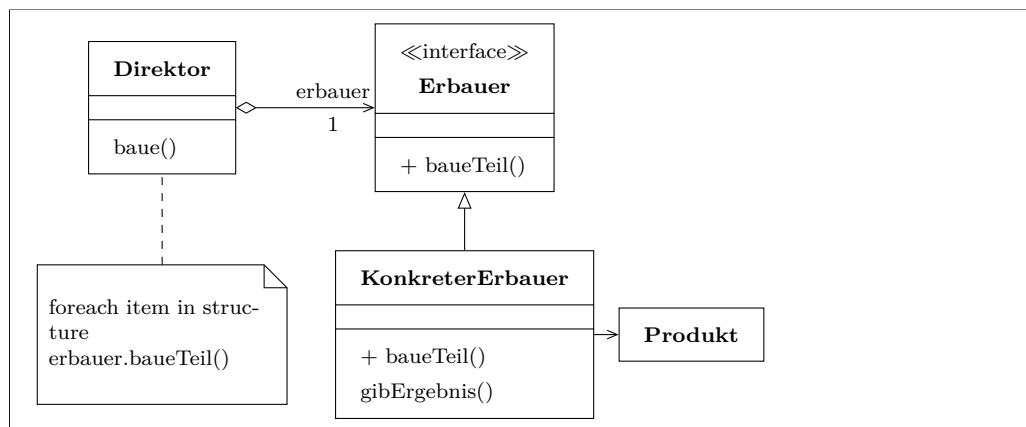
```

781 \def\liEntwurfsEinzelstueckAkteure{
782   \begin{description}
783     \item[Einzelstück (Singleton)]
784
785     stellt eine statische Methode bereit, mit deren Hilfe die Klienten
786     nur auf eine einzige Instanz der Klasse zugreifen können.
787   \end{description}
788 }

```

2.10.6 Erbauer

\liEntwurfsErbauerUml Quelle nach der deutschen Wikipedia



```

789 \def\liEntwurfsErbauerUml{
790   \begin{tikzpicture}
791     \umlsimpleclass[x=7,y=0]{Produkt}{}{}
792     \umlclass[x=4,y=3,type=interface]{Erbauer}{}{+ baueTeil()}
793     \umlclass[x=4,y=0]{KonkreterErbauer}{}{
794       + baueTeil()\
795       gibErgebnis()
796     }
797     \umlclass[x=0,y=3]{Direktor}{}{baue()}
798     \umluniagg[arg2=erbauer,pos2=0.7,mult2=1]{Direktor}{Erbauer}
799     \umluniassoc{KonkreterErbauer}{Produkt}
800     \umlinherit{KonkreterErbauer}{Erbauer}
801
802     \umlnote[x=0,y=0,width=3cm]{Direktor}{
803       foreach item in structure\
804       erbauer.baueTeil()
805     }
806   \end{tikzpicture}
807   \footcite{wiki:erbauer}
808 }

```

\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.

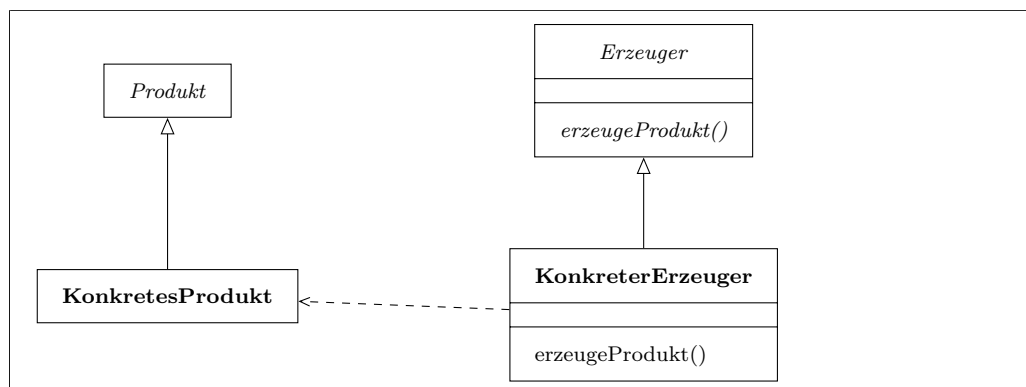
Produkt Das Produkt repräsentiert das zu konstruierende komplexe Objekt.

```

809 \def\liEntwurfsErbauerAkteure{
810   \begin{description}
811     \item[Erbauer]
812
813     Der Erbauer spezifiziert eine abstrakte Schnittstelle zur Erzeugung der
814     Teile eines komplexen Objektes.
815
816     \item[KonkreterErbauer]
817
818     Der konkrete Erbauer erzeugt die Teile des komplexen Objekts durch
819     Implementierung der Schnittstelle. Außerdem definiert und verwaltet er
820     die von ihm erzeugte Repräsentation des Produkts. Er bietet auch eine
821     Schnittstelle zum Auslesen des Produkts.
822
823     \item[Direktor]
824
825     Der Direktor konstruiert ein komplexes Objekt unter Verwendung der
826     Schnittstelle des Erbauers. Der Direktor arbeitet eng mit dem Erbauer
827     zusammen: Er weiß, welche Baureihenfolge der Erbauer verträgt oder
828     benötigt. Der Direktor entkoppelt somit den Konstruktionsablauf vom
829     Klienten.
830
831     \item[Produkt]
832
833     Das Produkt repräsentiert das zu konstruierende komplexe Objekt.
834   \footcite{wiki:erbauer}
835 \end{description}
836 }
```

2.10.7 Fabrikmethode

\liEntwurfsFabrikmethodeUml Quelle nach der deutschen Wikipedia



```

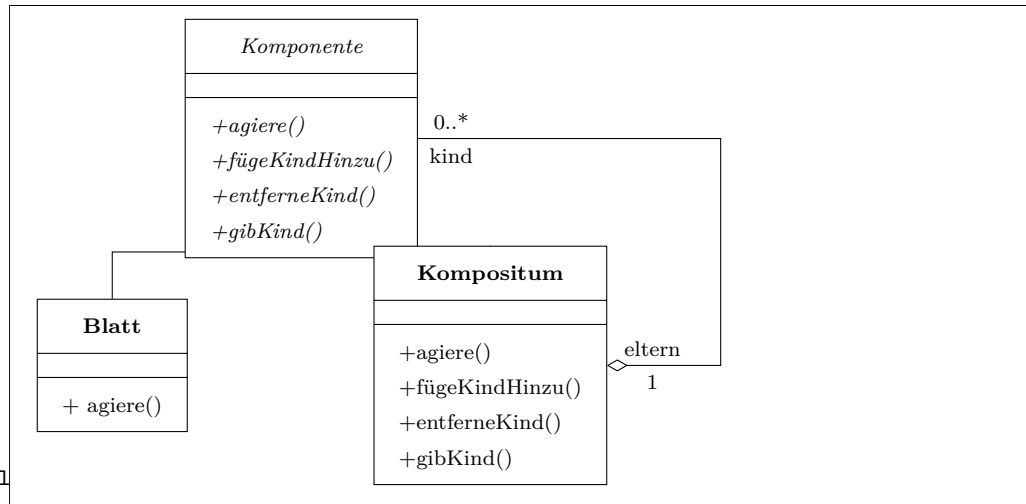
837 \def\liEntwurfsFabrikmethodeUml{
838   \begin{tikzpicture}
839     \umlsimpleclass[type=abstract]{Produkt}
840     \umlsimpleclass[below=2cm of Produkt]{KonkretesProdukt}
841     \umlinherit{KonkretesProdukt}{Produkt}
842
843     \umlclass[type=abstract,right=4cm of Produkt]{Erzeuger}{\{
844       \textit{erzeugenProdukt()}\}
  
```

```

845     }
846     \umlclass[below=1.2cm of Erzeuger]{KonkreterErzeuger}{-}{
847     erzeugeProdukt()
848     }
849     \umlinherit{KonkreterErzeuger}{Erzeuger}
850
851     \umldep{KonkreterErzeuger}{KonkretesProdukt}
852 \end{tikzpicture}
853 }

```

2.10.8 Kompositum



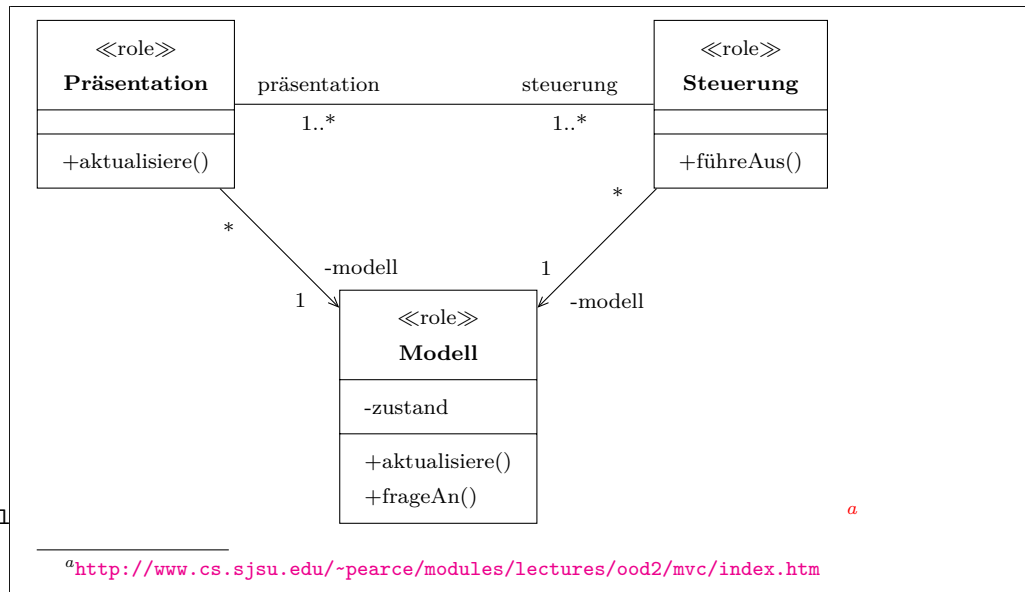
\liEntwurfsKompositumUml

```

854 \def\liEntwurfsKompositumUml{
855     \begin{tikzpicture}
856         \umlclass[x=2.5,y=3,type=abstract]{Komponente}{-}{
857             \textit{+agiere()}\
858             \textit{+fügeKindHinzu()}\
859             \textit{+entferneKind()}\
860             \textit{+gibKind()}
861         }
862         \umlclass[x=0]{Blatt}{-}{+ agiere()}
863         \umlclass[x=5]{Kompositum}{-}{
864             +agiere()\
865             +fügeKindHinzu()\
866             +entferneKind()\
867             +gibKind()
868         }
869
870         \umlVHVinherit{Kompositum}{Komponente}
871         \umlVHVinherit{Blatt}{Komponente}
872         \umlHVVaggreg[anchor1=east,arm1=1.5cm,arg1=eltern,mult1=1,arg2=kind,mult2=0..*,pos2=2.9,
873 \end{tikzpicture}
874 }

```

2.10.9 Modell-Präsentation-Steuerung



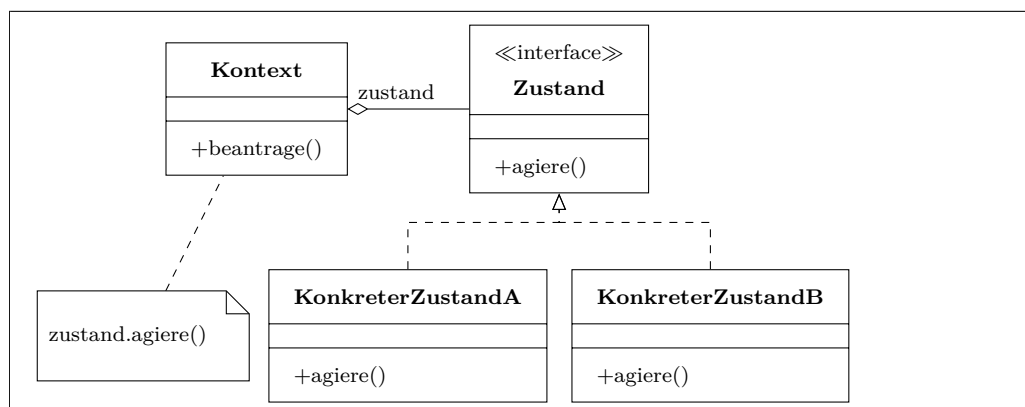
```

875 \def\liEntwurfsModellPraesentationSteuerungUml{
876   \begin{tikzpicture}
877     \umlclass[x=-4cm,type=role]{Präsentation}{+aktualisiere()}
878     \umlclass[x=4cm,type=role]{Steuerung}{+führeAus()}
879     \umlclass[y=-4cm,type=role]{Modell}{
880       -zustand
881     }{
882       +aktualisiere()\
883       +frageAn()
884     }
885
886     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Präsentation}{Modell}
887     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Steuerung}{Modell}
888     \umlassoc[arg1=präsentation,mult1=1..*,arg2=steuerung,mult2=1..*]{Präsentation}{Steuerung}
889   \end{tikzpicture}
890   \liFussnoteUrl{http://www.cs.sjsu.edu/~pearce/modules/lectures/ood2/mvc/index.htm}
891 }

```

2.10.10 Zustand

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



```

892 \def\liEntwurfsZustandUml{
893   \begin{tikzpicture}
894     \umlclass[x=-1,y=3]{Kontext}{+beantrage()}
895     \umlclass[x=3,y=3,type=interface]{Zustand}{+agiere()}
896     \umlclass[x=1,y=0]{KonkreterZustandA}{+agiere()}
897     \umlclass[x=5,y=0]{KonkreterZustandB}{+agiere()}
898

```

```

899 \umlVHVreal{KonkreterZustandA}{Zustand}
900 \umlVHVreal{KonkreterZustandB}{Zustand}
901
902 \umlaggreg[arg=zustand,pos=0.4]{Kontext}{Zustand}
903
904 \umlnote[x=-2.5,y=0,width=2.5cm]{Kontext}{zustand.agiere()}
905 \end{tikzpicture}
906 }

```

\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.

```

907 \def\liEntwurfsZustandAkteure{
908 \begin{description}
909 \item[Kontext (Context)]
910
911 definiert die clientseitige Schnittstelle und verwaltet die separaten
912 Zustandsklassen.
913
914 \item[State (Zustand)]
915
916 definiert eine einheitliche Schnittstelle aller Zustandsobjekte und
917 implementiert gegebenenfalls ein Standardverhalten.
918
919 \item[KontreterZustand (ConcreteState)]
920
921 implementiert das Verhalten, das mit dem Zustand des Kontextobjektes
922 verbunden ist.
923 \end{description}
924 }

```

925

2.11 er.sty

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

299 \RequirePackage{tikz-er2}
300 \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}

931 \RequirePackage{soul}
932 \RequirePackage{fontawesome}

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

933 \ExplSyntaxOn

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

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

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

\liErMpEntity mp = marginpar
Let-Abkürzung: \let\ e=\liErMpEntity
937 \def\liErMpEntity#1{
938   \liErEntity{#1}
939   \marginpar{
940     \liErEntity{\tiny\faSquareO{}}~E:~#1}
941   }
942 }
□

\liErMpRelationship Let-Abkürzung: \let\ r=\liErMpRelationship
943 \def\liErMpRelationship#1{
944   \liErRelationship{#1}
945   \marginpar{
946     \liErRelationship{\tiny\faGg{}}~R:~#1}
947   }
948 }

\liErMpAttribute Let-Abkürzung: \let\ a=\liErMpAttribute
949 \def\liErMpAttribute#1{
950   \liErAttribute{#1}
951   \marginpar{
952     \liErAttribute{\tiny\faCircleThin{}}~A:~#1}
953   }
954 }

\liErDatenbankName Let-Abkürzung: \let\ d=\liErDatenbankName
datenbank name
955 \def\liErDatenbankName#1{
956   {
957     \footnotesize\texttt{(#1)}
958   }
959 }

960 \ExplSyntaxOff
961

```

2.12 formale-sprachen.sty

```

962 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
963 \ProvidesPackage{lehramt-informatik-formale-sprachen}[2021/02/21 Hilfsmakros
964 zum Setzen von mathematischen Formeln bei dem Thema Formale Sprachen]
965
966 \directlua{
967   formale_sprachen = require('lehramt-informatik-formale-sprachen')
968 }
969
970 \RequirePackage{hyperref}
971 \RequirePackage{lehramt-informatik-mathe}

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

972 \def\liMengeOhneMathe#1{\{ #1 \}}
973 \def\liMenge#1{%
974   \ifmmode%
975     \liMengeOhneMathe{#1}%
976   \else%
977     $\liMengeOhneMathe{#1}$%
978   \fi%
979 }

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

980 \def\liEpsilon{$\varepsilon$}

\liPotenzmenge Umgeben mit geschweiften Klammern in einer Mathematik-Umgebung

981 \def\erzeuge@tiefgestellt#1{\directlua{formale_sprachen.erzeuge_tiefgestellt('#1')}}
982 \def\liPotenzmengeOhneMathe#1{\{ \erzeuge@tiefgestellt{#1} \}}
983 \def\liPotenzmenge#1{$\liPotenzmengeOhneMathe{#1}$}

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

984 \let\liZustandsmengeOhneMathe=\liPotenzmengeOhneMathe
985 \let\liZustandsmenge=\liPotenzmenge

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

986 \def\liUeberfuehrungsFunktionOhneMathe#1{\delta(\erzeuge@tiefgestellt{#1})}
987 \def\liUeberfuehrungsFunktion#1{
988   \ifmmode
989     \liUeberfuehrungsFunktionOhneMathe{#1}
990   \else
991     $\liUeberfuehrungsFunktionOhneMathe{#1}$
992   \fi
993 }

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

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

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

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

\liZustandsBuchstabe

996 \def\liZustandsBuchstabe{z}

\liZustandsBuchstabeGross

997 \def\liZustandsBuchstabeGross{Z}

```

```

\liZustandsmengeNr
    998 \def\zustandsnamens@liste#1#2{
    999   $
    1000   \{
    1001     \directlua{formale_sprachen.erzeuge_zustandsnamens_liste('#1', '#2')}
    1002   \}
    1003   $
    1004 }
    1005 \def\liZustandsmengeNr#1{\zustandsnamens@liste{\liZustandsBuchstabe}{#1}}

\liZustandsmengeNrGross
    1006 \def\liZustandsmengeNrGross#1{\zustandsnamens@liste{\liZustandsBuchstabeGross}{#1}}

\liZustandsname \liZustandsname{1}: $z_1$
    1007 \def\liZustandsname#1{\liZustandsBuchstabe_#1$}

\liZustandsnameGross \liZustandsnameGross{1}: $Z_1$
    1008 \def\liZustandsnameGross#1{\liZustandsBuchstabeGross_#1$}

\liAbleitung \liAbleitung{S -> aB -> ab}:  $S \vdash aB \vdash ab$ 
    1009 \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}

    1010 \NewDocumentEnvironment { liProduktionsRegeln }
    1011 { 0{P} +b }
    1012 {
    1013   \noindent
    1014   $#1 = \{ $
    1015   \vspace{-0.2cm}
    1016   \begin{align*}
    1017     \directlua{formale_sprachen.produktions_regeln('#2')}
    1018   \end{align*}
    1019   \vspace{-1.5cm}
    1020   \begin{flushright}\}\end{flushright}
    1021 } {}

\liProduktionen \liProduktionen{S -> A, A -> a}:  $\{S \rightarrow A, A \rightarrow a\}$ 
    1022 \def\liProduktionen#1{
    1023   \liMenge{\directlua{formale_sprachen.produktions_regeln('#1', true)}}
    1024 }

\liZustandsnameTiefgestellt Automatisch tiefgestellte Nummerierung \z1
Let-Abkürzung: \let\z=\liZustandsnameTiefgestellt
    1025 \def\liZustandsnameTiefgestellt#1{
    1026   \ifmmode
    1027     \liZustandsBuchstabe\sb{#1}
    1028   \else
    1029     $\liZustandsBuchstabe\sb{#1}$
    1030   \fi
    1031 }

    1032 \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}

1033 \NewDocumentCommand{ \liAusdruck } { O{L} m m } {
1034 $
1035 \tl_if_empty:nTF {#1} {} {#1 =}
1036 \{
1037 \, #2 \,
1038 |
1039 \, #3 \,
1040 \}$
1041 }
1042 \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

1043 \def\liFlaci#1{%
1044 \par
1045 {%
1046 \scriptsize
1047 Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte
1048 Automaten, Compiler und Interpreter) Ein Projekt der Hochschule
1049 Zittau/Görlitz und der Pädagogischen Hochschule Schwyz:
1050 \href{https://flaci.com/#1}{flaci.com/#1}%
1051 }%
1052 \par
1053 }

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

1054 \ExplSyntaxOn
1055 \NewDocumentCommand {\liGrammatik} { O{G} m } {
1056 \tl_set:Nn \l_variablen_tl {V}
1057 \tl_set:Nn \l_alphabet_tl {\Sigma}
1058 \tl_set:Nn \l_produktionen_tl {P}
1059 \tl_set:Nn \l_start_tl {S}
1060
1061 \keys_define:nn { grammatik } {
1062 variablen .code:n = {\tl_set:Nn \l_variablen_tl {\liMenge{##1}}},
1063 alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
1064 produktionen .code:n = {\tl_set:Nn \l_produktionen_tl {\liProduktionen{##1}}},
1065 start .code:n = {\tl_set:Nn \l_start_tl {##1}},
1066 }
1067
1068 \keys_set:nn { grammatik } { #2 }
1069
1070 $#1 = (
1071 \l_variablen_tl,
1072 \l_alphabet_tl,
1073 \l_produktionen_tl,
1074 \l_start_tl
1075 )$

```

```
1076 }  
1077 \ExplSyntaxOff  
  
1078
```

2.13 formatierung.sty

```
1079 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1080 \ProvidesPackage{lehramt-informatik-formatierung}[2020/11/27]
```

2.13.1 Schriftarten / Typographie

```
1081 \RequirePackage{mathpazo}
1082 \RequirePackage[no-math]{fontspec}
1083 \setmainfont{texgyrepagella}
1084 \setsansfont{QTAncientOlive}
1085 \RequirePackage{sectsty}
1086 \allsectionsfont{\fontspec{QTAncientOlive}}
```

2.13.2 Farben

```
1087 \RequirePackage{xcolor}
1088 \definecolor{infogray}{rgb}{0.97,0.97,0.97}
```

2.13.3 Überschriften

```
1089 \RequirePackage{titlesec}
1090 \titleformat{\chapter}[display]{\sffamily\bfseries}{\Opt}{\LARGE}
1091 \titlespacing{\chapter}{0pt}{0pt}{*1}
1092 \titleformat{\paragraph}[hang]{\sffamily\normalsize\bfseries}{\theparagraph}{1em}{\}
1093 \setcounter{secnumdepth}{0}
```

2.13.4 Listen

```
1094 \RequirePackage{paralist}
1095 \renewcommand\labelitemi{-}
1096 \renewcommand\labelitemii{-}
1097 \renewcommand\labelitemiii{-}
1098 \renewcommand\labelitemiv{-}
1099 % Counter: enumi enumii enumiii enumiv
1100 % Styles: \arabic{counter} \alph{counter} \Alph{counter} \roman{counter} \Roman{counter}
1101 \renewcommand{\labelenumi}{(\alph{enumi})}
1102 \renewcommand{\labelenumii}{(\roman{enumii})}
```

2.13.5 Kasten

```
1103 \RequirePackage{mdframed}
1104 \mdfsetup{backgroundcolor=infogray}
```

liKasten

```
1105 \NewDocumentEnvironment { liKasten }{ } {
1106   \begin{mdframed}[backgroundcolor=white!0]
1107 } {
1108   \end{mdframed}
1109 }
```

2.13.6 Header

```
1110 \RequirePackage{fancyhdr}
1111 \fancyhead[L,C,R]{\}
1112 \fancyfoot[L]{\}
1113 \fancyfoot[C]{\}
1114 \fancyfoot[R]{\thepage}
1115 \pagestyle{fancy}
1116 \renewcommand{\headrulewidth}{0pt}
1117 \renewcommand{\footrulewidth}{0pt}
1118
```

2.14 gantt.sty

```

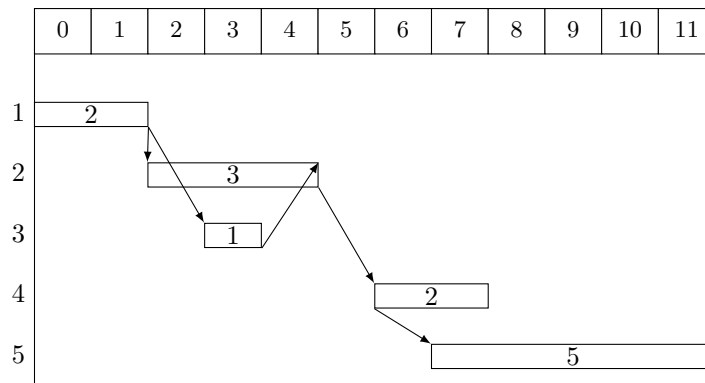
1119 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1120 \ProvidesPackage{lehramt-informatik-gantt}[2020/09/05]

\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}

```



```

1121 \RequirePackage{tikz-uml}
1122 \RequirePackage{pgfgantt}
1123 \setganttlinklabel{f-s}{}
1124 \setganttlinklabel{s-s}{}
1125 \setganttlinklabel{f-f}{}
1126 \setganttlinklabel{s-f}{}
1127

```


2.15 grafik.sty

```
1128 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1129 \ProvidesPackage{lehramt-informatik-grafik}[2020/12/27 Lädt das Paket
1130 TikZ, um Grafiken zeichnen zu können.]
1131 \RequirePackage{tikz}
1132
```

2.16 graph.sty

```

1133 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1134 \ProvidesPackage{lehramt-informatik-graph}[2020/06/09]
1135 \RequirePackage{tikz}

```

Für die die Adjazenzliste (`xrightharpoon`)

```

1136 \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}

```

$$\begin{array}{c}
 a \\
 b \\
 c \\
 d \\
 e
 \end{array}
 \begin{array}{ccccc}
 a & b & c & d & e \\
 \left(\begin{array}{ccccc}
 0 & 1 & 0 & 4 & 0 \\
 0 & 0 & 0 & 1 & 3 \\
 0 & 0 & 0 & 1 & 0 \\
 4 & 1 & 1 & 0 & 2 \\
 0 & 0 & 0 & 0 & 0
 \end{array} \right)
 \end{array}$$

```

1137 \RequirePackage{blkarray}
1138 \usetikzlibrary{arrows.meta}

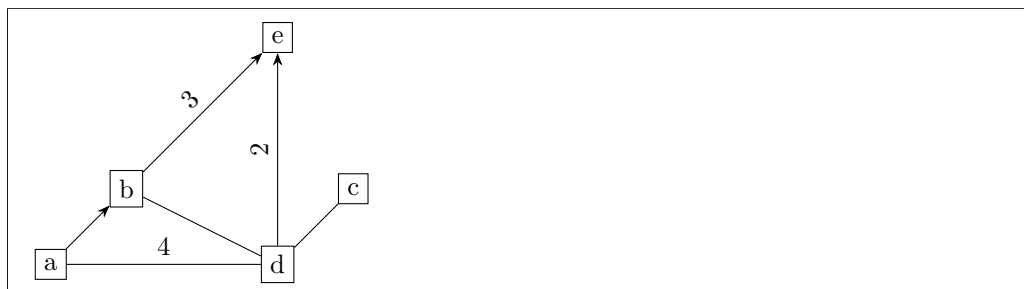
```

```

\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}

```



```

1139 \tikzset{
1140   li graph/.style={
1141     every node/.style={
1142       rectangle,
1143       draw,
1144     },
1145     every edge/.style={
1146       >={Stealth[black]},
1147       draw,
1148     },
1149     every edge/.append style={
1150       every node/.style={
1151         sloped,
1152         auto,
1153       }
1154     }
1155   },
1156   li markierung/.style={
1157     ultra thick,
1158   }
1159 }

```

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}

```

```

1160 \NewDocumentEnvironment { liGraphenFormat }{ +b } {} {}

```

```

1161

```

2.17 hanoi.sty

```
1162 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1163 \ProvidesPackage{lehramt-informatik-hanoi}[2020/12/19 Paket zum Setzen
1164 von Hanoi-Grafiken]
```

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

```
1165 \RequirePackage{tikz}
1166 \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}

1167 \def\li@mset #1[#2]=#3{%
1168 \expandafter\xdef\csname #1#2\endcsname{#3}
1169 }
1170 \def\li@mget #1[#2]{%
1171 \csname #1#2\endcsname
1172 }
1173 \def\li@minc #1[#2]+=#3{%
1174 \pgfmathparse{\li@mget #1[#2]+#3}%
1175 \li@mset #1[#2]=\pgfmathresult
1176 }
1177
1178 \def\liHanoi#1#2{
1179   \edef\li@numdiscs{#1}
1180   \def\li@sequence{#2}
1181   \begin{tikzpicture}[line width=4mm,brown!40,line cap=round,xscale=3]
1182     % init colors
1183     \foreach[count=\j] \c in {red,green!80!black,blue,orange,violet,gray,yellow!80!black,purp
1184     \li@mset col[\j]={\c};
1185     % draw poles and init pole counters
1186     \foreach \j in {1,2,3}{
1187       \li@mset pos[\j]=0
1188       \draw (\j,-.5) -- +(0, .5 + .5*\li@numdiscs);
1189     }
1190     % draw base
1191     \draw (.5,-.5) -- +(3,0);
1192     % draw discs
1193     \foreach[count=\k] \i/\j in \li@sequence{
1194       \draw[draw={\li@mget col[\i]}] (\j,\li@mget pos[\j]) +(-.4*\i/\li@numdiscs,0) -- +(.4*
1195       \li@minc pos[\j]+=.5}
1196     }
1197   \end{tikzpicture}
1198 }

1199
```

2.18 komplexitaetstheorie.sty

```

1200 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1201 \ProvidesPackage{lehramt-informatik-komplexitaetstheorie}[2021/07/08 Zum
1202 Setzen von Karps NP-vollständigen Problemen, Hilfsmakros für die
1203 Polynomialzeitreduktion.]

```

Let-Abkürzungen

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

1204 \liLadePakete{mathe}

Für das Makro `\liProblemBeschreibung` benötigt.

```
1205 \RequirePackage{mdframed}
```

$$\backslash\mathrm{liStrich} \quad \$L, \backslash\mathrm{liStrich}\{L\}$: $L, L'$$$

```
1206 \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
```

1207 \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

```

1208 \def\liProblemBeschreibung#1#2#3{
1209   \begin{mdframed}[
1210     userdefinedwidth=9cm,
1211     align=center,
1212     backgroundcolor=white!0,
1213   ]
1214     \centerline{\large\liProblemName{#1}}
1215
1216     \medskip
1217
1218     \begin{description}
1219       \item[Gegeben:] #2
1220       \item[Frage:] #3
1221     \end{description}
1222   \end{mdframed}
1223 }
```

```

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

1224 \NewDocumentCommand{ \liPolynomiellReduzierbar } { m O{p} m } {
1225 \begin{displaymath}
1226 \liProblemName{#1}
1227 \preceq_{#2}
1228 \liProblemName{#3}
1229 \end{displaymath}
1230 }

\liProblemVertexCover

1231 \def\liProblemClique{%
1232 Das \textbf{Cliquesproblem} fragt nach der Existenz einer Clique der
1233 Mindestgröße  $n$  in einem gegebenen Graphen.
1234 \footcite[Seite 76]{theo:fs:4}
1235 Eine Clique ist eine Teilmenge von Knoten in einem ungerichteten
1236 Graphen, bei der \emph{jedes Knotenpaar durch eine Kante} verbunden ist.
1237 \footcite{wiki:cliquenproblem}
1238 }

\liProblemVertexCover

1239 \def\liProblemVertexCover{%
1240 %
1241 Das \textbf{Knotenüberdeckungsproblem} (\liProblemName{Vertex Cover})
1242 fragt, ob zu einem gegebenen einfachen Graphen und einer natürlichen
1243 Zahl  $k$  eine Knotenüberdeckung der Größe von höchstens  $k$  existiert.
1244 \footcite{wiki:knotenueberdeckung}
1245
1246 Das heißt, ob es eine aus maximal  $k$  Knoten bestehende Teilmenge  $U$ 
1247 der Knotenmenge gibt, so dass jede Kante des Graphen mit
1248 mindestens einem Knoten aus  $U$  verbunden ist.
1249 \footcite[Seite 78]{theo:fs:4}%
1250 }

\liProblemSubsetSum Kein Karp-21-Problem

1251 \def\liProblemSubsetSum{%
1252 Das \textbf{Teilsommenproblem} (\liProblemName{Subset Sum} oder
1253 \liProblemName{SSP}) ist ein spezielles Rucksackproblem.
1254 \footcite{wiki:teilsommenproblem}
1255 Gegeben sei eine Menge von ganzen Zahlen  $I = \{w_1, w_2, \dots, w_n\}$ .
1256 Gesucht ist eine Untermenge, deren Elementsumme maximal, aber nicht
1257 größer als eine gegebene obere Schranke  $c$  ist.
1258 \footcite[Seite 74]{theo:fs:4}
1259 }

\liProblemSubsetSum Kein Karp-21-Problem

1260 \def\liProblemSat{%
1261 Das \textbf{Erfüllbarkeitsproblem der Aussagenlogik} \liProblemName{Sat}
1262 und \liProblemName{k-SAT} mit  $k \geq 3$ ,  $k \in \mathbb{N}$  (Satz von
1263 Cook) fragt, ob eine aussagenlogische Formel erfüllbar
1264 ist.\footcite{wiki:sat} Das Erfüllbarkeitsproblem der
1265 \emph{Aussagenlogik} ist in exponentieller Zeit in Abhängigkeit der
1266 Anzahl der Variablen mit Hilfe einer Wahrheitstabelle entscheidbar.
1267 Diese \emph{Wahrheitstabelle} kann nicht in polynomieller Zeit
1268 aufgestellt werden.
1269 \footcite[Seite 71]{theo:fs:4}
1270 }

1271

```

2.19 kontrollflussgraph.sty

```
1272 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1273 \ProvidesPackage{lehramt-informatik-kontrollflussgraph}[2020/11/07]

1274 \RequirePackage{tikz}
1275 \usetikzlibrary{positioning}
1276 \tikzset{
1277   li kontrollfluss/.style={
1278     knoten/.style={
1279       circle,
1280       draw
1281     },
1282     usebox/.style={
1283       draw,
1284       rectangle,
1285       font=\scriptsize,
1286       anchor=west,
1287       align=left,
1288     },
1289     bedingung/.style={
1290       midway,
1291       draw=none,
1292       font=\scriptsize
1293     },
1294     knotenbeschriftung/.style={
1295       draw,
1296       rectangle,
1297       midway,
1298       font=\scriptsize
1299     },
1300     wahr/.style={
1301       thick
1302     },
1303     falsch/.style={
1304       dashed
1305     },
1306     every node/.style={
1307       circle,
1308       draw,
1309     },
1310     every edge/.append style={
1311       every node/.style={
1312         draw=none,
1313         bedingung,
1314       }
1315     },
1316     every path/.style={
1317       draw,
1318       ->,
1319     },
1320     every pin/.style={
1321       draw,
1322       dotted,
1323       rectangle,
1324       pin position=right
1325     },
1326     every pin edge/.style={
1327       dotted,
1328       arrows=-,
1329     }
1330   }
1331 }
```

liKontrollflussgraph

```

1332 \NewDocumentEnvironment { liKontrollflussgraph } { 0{ } } {
1333   \begin{tikzpicture}[
1334     li kontrollfluss,
1335     #1
1336   ]
1337 } {
1338   \end{tikzpicture}
1339 }

\liAnweisung
1340 \def\liAnweisung#1(#2,#3){\node[knoten] at (#2,#3) (#1) {#1};}

\liBedingung Let-Abkürzung: \let\b=\liBedingung
1341 \def\liBedingung#1#2{node[bedingung,#1]{\texttt{#2}}}}

\liBedingungWahr Let-Abkürzung: \let\bWahr=\liBedingungWahr
1342 \def\liBedingungWahr#1{node[bedingung,#1]{\texttt{[true]}}}}

\liBedingungFalsch Let-Abkürzung: \let\bFalsch=\liBedingungFalsch
1343 \def\liBedingungFalsch#1{node[bedingung,#1]{\texttt{[false]}}}}

\liKontrollCode Let-Abkürzung: \let\c=\liKontrollCode
1344 \def\liKontrollCode#1{{\tiny\texttt{\textcolor{gray}{#1}}}}

\liKontrollTextzeileKnoten Let-Abkürzung: \let\k=\liKontrollTextzeileKnoten
1345 \def\liKontrollTextzeileKnoten#1{\raisebox{-2pt}{\tikz[scale=0.5,transform shape] \node[draw

\liKontrollKnotenPfad Let-Abkürzung: \let\p=\liKontrollKnotenPfad
1346 \ExplSyntaxOn
1347 \NewDocumentCommand { \liKontrollKnotenPfad }{ m }
1348 {
1349   \seq_set_split:Nnn \l_tmpa_seq { - } { #1 }
1350   \seq_set_map:Nnn \l_tmpa_seq \l_tmpa_seq {\liKontrollTextzeileKnoten{##1}}
1351   \seq_use:Nn \l_tmpa_seq { ~~~~ }
1352 }
1353 \ExplSyntaxOff

1354

```


2.20 literatur-dummy.sty

```
1355 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1356 \ProvidesPackage{lehramt-informatik-literatur-dummy}[2020/11/27]

\literatur

1357 \def\literatur{}

\footcite

1358 % \RequirePackage[stable,multiple]{footmisc}
1359 \NewDocumentCommand{ \footcite } { o m }{}

1360
```

2.21 literatur.sty

```
1361 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1362 \ProvidesPackage{lehramt-informatik-literatur}[2020/11/27]

1363 \RequirePackage{csquotes}
1364 \RequirePackage[
1365   bibencoding=utf8,
1366   citestyle=authortitle,
1367   backend=biber,
1368 ]{biblatex}
1369 \addbibresource{\LehramtInformatikRepository/Quellen/10_DB.bib}
1370 \addbibresource{\LehramtInformatikRepository/Quellen/20_OOMUP.bib}
1371 \addbibresource{\LehramtInformatikRepository/Quellen/30_AUD.bib}
1372 \addbibresource{\LehramtInformatikRepository/Quellen/40_SOSY.bib}
1373 \addbibresource{\LehramtInformatikRepository/Quellen/60_FUMUP.bib}
1374 \addbibresource{\LehramtInformatikRepository/Quellen/70_THEO.bib}
1375 \addbibresource{\LehramtInformatikRepository/Quellen/80_DDI.bib}
1376 \addbibresource{\LehramtInformatikRepository/Quellen/Allgemein.bib}
1377 \addbibresource{\LehramtInformatikRepository/Quellen/Examen.bib}
1378 \addbibresource{\LehramtInformatikRepository/Quellen/Wikipedia.bib}
1379 % To allow footnotes in the heading
1380 \RequirePackage[stable,multiple]{footmisc}

\literatur

1381 \def\literatur{\printbibliography[heading=subbibliography]}

1382
```

2.22 makros.sty

```
1383 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1384 \ProvidesPackage{lehramt-informatik-makros}[2020/11/27 Eine Sammlung von
1385 häufig verwendeten Makros und Umgebungen, die thematisch zu keinem
1386 anderen Paket passen]
1387 \RequirePackage{hyperref}
1388 \RequirePackage{graphicx}
    Für die Umgebung liQuellen benötigt.
1389 \RequirePackage{paralist}

\inhaltsverzeichnis
1390 \def\inhaltsverzeichnis {
1391   \begin{mdframed}
1392     \begin{group}
1393       \let\clearpage\relax
1394       \tableofcontents
1395     \end{group}
1396   \end{mdframed}
1397 }

\memph \mephm (\marginpar and \emph)
1398 \newcommand\memph{1}{\emph{#1}\marginpar{\tiny#1}}

\SLASH
1399 \newcommand\SLASH{\char`\\}

\liPseudoUeberschrift Text, der sich wie eine Überschrift verhält.
1400 \newcommand\liPseudoUeberschrift{1}{
1401   \bigskip
1402   \noindent
1403   \textsf{\textbf{#1}}
1404   \noindent
1405 }

\liBeschriftung Ähnlich dem Makro \liPseudoUeberschrift{}. Am Ende des Textes wird ein Doppelpunktzeichen angehängt.
1406 \newcommand\liBeschriftung{1}{
1407   \par
1408   \noindent
1409   \medskip
1410   \textbf{#1}:
1411   \medskip
1412   \noindent
1413 }

\hinweis
1414 \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.
1415 \NewDocumentEnvironment { liProjektSprache }{ o +b } {} {}

liEinbettung
1416 \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.
1417 \RequirePackage{xparse}
1418 \ExplSyntaxOn
```

```

1419 \NewDocumentEnvironment{ liAntwort } { 0{standard} }
1420 {
1421   \str_case:nn {#1} {
1422     {standard} {
1423       \def\beschriftung{}
1424       \mdfsetup{backgroundcolor=gray!5,linecolor=gray}
1425     }
1426     {richtig} {
1427       \def\beschriftung{richtig}
1428       \mdfsetup{backgroundcolor=green!5,linecolor=green}
1429     }
1430     {falsch} {
1431       \def\beschriftung{falsch}
1432       \mdfsetup{backgroundcolor=red!3,linecolor=red}
1433     }
1434     {muster} {
1435       \def\beschriftung{Musterlösung}
1436       \mdfsetup{backgroundcolor=green!12,linecolor=green}
1437     }
1438   }
1439   \ifx\beschriftung\empty\else
1440     \noindent
1441     \textbf{\beschriftung{:}}
1442     \fi
1443     \begin{mdframed}
1444   }
1445   {\end{mdframed}}

```

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

```

1446 \NewDocumentEnvironment{ liAdditum } { o }
1447 {
1448   \begin{mdframed}[backgroundcolor=yellow!5]
1449     \IfNoValueTF {#1}
1450     { \liPseudoUeberschrift{Additum} }
1451     { \liPseudoUeberschrift{Additum:~#1} }
1452   }
1453   {\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.

```

1454 \NewDocumentEnvironment{ liExkurs }{o +b}{
1455   \vspace{0.2cm}%
1456   \begin{mdframed}[
1457     backgroundcolor=white,
1458     bottomline=false,
1459     innermargin=1cm,
1460     leftline=true,
1461     linecolor=black,
1462     linewidth=0.1cm,
1463     outermargin=1cm,
1464     rightline=false,
1465     topline=false,
1466   ]

```

```

1467 \footnotesize
1468 \noindent%
1469 \textbf{Exkurs:~#1}\par%
1470 \noindent%
1471 #2
1472 \end{mdframed}
1473 \vspace{0.2cm}
1474 }{}

```

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

```

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

```

Weiterführende Literatur:

- Quelle 1
- Quelle 2

```

1475 \cs_new:Npn \listen@punkt #1 {\item #1}
1476 \NewDocumentEnvironment { liQuellen }{ +b }
1477 {
1478   \seq_clear_new:N \l_quellen
1479   \seq_set_split:Nnn \l_quellen {\item} {#1}
1480   \seq_remove_all:Nn \l_quellen {}
1481   \begin{mdframed}[backgroundcolor=white!0]
1482     \footnotesize
1483     \noindent
1484     \textsf{\textbf{Weiterführende~Literatur:}}
1485     \medskip
1486     \begin{compactitem}
1487       \seq_map_function:NN \l_quellen {\listen@punkt}
1488     \end{compactitem}
1489   \end{mdframed}
1490   %
1491   \makeatletter
1492   \par\@afterindentfalse\@afterheading
1493   \makeatother
1494 } {}

```

liLernkartei

```

1495 \NewDocumentEnvironment { liLernkartei }{ m +b }
1496 {
1497   \begin{mdframed}
1498     \footnotesize
1499     \noindent%
1500     \textbf{Lernkarteikarte:~#1}\par%
1501     \noindent%
1502     #2
1503   \end{mdframed}
1504 } {}

```

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

```

1505 \NewDocumentEnvironment { liDiagramm }{ m +b }
1506 {
1507   \begin{mdframed}[backgroundcolor=white!0]
1508     \small
1509     \noindent%
1510     \textit{#1}:
1511     \begin{center}

```

```

1512 #2
1513 \medskip
1514 \end{center}
1515 \end{mdframed}
1516 } {}
1517 \ExplSyntaxOff

\liFussnoteUrl \liFussnoteUrl[\langlezusätzlicher-text\rangle]{\langleurl\rangle} \liFussnoteUrl[zusätzlicher Text]{url}:
Eine HTTP-URL als Fußnote setzen.
1518 \NewDocumentCommand{\liFussnoteUrl} { o m } {
1519 \footnote{\url{#2}\IfNoValueTF{#1}{\{ ( #1 ) \}}
1520 }
1521

\liFussnoteLink \liFussnoteLink[\langlezusätzlicher-text\rangle]{\langlelink-text\rangle}{\langleurl\rangle} \liFussnoteLink[zusätzlicher Text]{tex
Einen Link, d. h. einen Link-Text und eine URL als Fußnote setzen.
1522 \NewDocumentCommand{\liFussnoteLink} { o m m } {
1523 \footnote{\href{#3}{#2}\IfNoValueTF{#1}{\{ ( #1 ) \}}
1524 }

\zB
1525 \def\zB{z.\,B. }

\ZB
1526 \def\ZB{Z.\,B. }

\dh
1527 \def\dh{d.\,h. }

1528

```

2.23 master-theorem.sty

1529 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

1530 \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{0}(1)} % f(n)

\liMasterFallRechnung

% 1. Fall

{für $\varepsilon = 4$: \}

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

% 2. Fall

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

% 3. Fall

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

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

\liMasterVariablenDeklaration

{ } % a

{ } % b

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

\liMasterFallRechnung

% 1. Fall

{ }

% 2. Fall

{ }

% 3. Fall

{ }

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

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

1531 \ExplSyntaxOn

1532 \RequirePackage{amsmath}

\liRundeKlammer

1533 \def\liRundeKlammer#1{

1534 \negthinspace \left(#1 \right)

1535 }

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

1536 \def\liThetaOhneMathe#1{

1537 \Theta \liRundeKlammer{#1}

1538 }

1539 \def\liTheta#1{

1540 \ifmmode

1541 \liThetaOhneMathe{#1}

1542 \else

1543 \$\liThetaOhneMathe{#1}\$

1544 \fi

1545 }

```

\liOmega \liOmega{n^2}:  $\Omega(n^2)$ 
1546 \def\liOmegaOhneMathe#1{
1547 \Omega \liRundeKlammer{#1}
1548 }
1549 \def\liOmega#1{
1550 \ifmmode
1551 \liOmegaOhneMathe{#1}
1552 \else
1553 $\liOmegaOhneMathe{#1}$
1554 \fi
1555 }

\liO \liO{n^2}:  $\mathcal{O}(n^2)$ 
1556 \def\liOOhneMathe#1{
1557 \mathcal{O} \liRundeKlammer{#1}
1558 }
1559 \def\liO#1{
1560 \ifmmode
1561 \liOOhneMathe{#1}
1562 \else
1563 $\liOOhneMathe{#1}$
1564 \fi
1565 }

\liT Let-Abkürzung: \let\T=\liT
\liT{16}{2}:  $16 \cdot T\left(\frac{n}{2}\right)$  \liT{}{2}:  $T\left(\frac{n}{2}\right)$ 
1566 \def\liTOhneMathe#1#2{
1567 \tl_if_blank:nTF {#1}
1568 {}
1569 {#1 \cdot }
1570 T
1571 \liRundeKlammer{\frac{n}{#2}}
1572 }
1573 \def\liT#1#2{
1574 \ifmmode
1575 \liTOhneMathe{#1}{#2}
1576 \else
1577 $\liTOhneMathe{#1}{#2}$
1578 \fi
1579 }

\liRekursionsGleichung \liRekursionsGleichung:  $T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$ 
1580 \def\liRekursionsGleichung{
1581 $T(n) = \liT{a}{b} + f(n)$
1582 }

\liBedingungEins \liBedingungEins:  $f(n) \in \mathcal{O}(n^{\log_b a - \varepsilon})$ 
1583 \def\liBedingungEins{
1584 $f(n) \in \liO{n^{\log\sb{b}a - \varepsilon}}$
1585 }

\liBedingungZwei \liBedingungZwei:  $f(n) \in \Theta(n^{\log_b a})$ 
1586 \def\liBedingungZwei{
1587 $f(n) \in \liTheta{n^{\log\sb{b}a}}$
1588 }

\liBedingungDrei \liBedingungDrei:  $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ 
1589 \def\liBedingungDrei{
1590 $f(n) \in \liOmega{n^{\log\sb{b}a + \varepsilon}}$
1591 }

1592 \ExplSyntaxOff

```


\liMasterVariablen

```

1593 \def\liMasterVariablen{
1594   \begin{displaymath}
1595     T(n) = \liT{a}{b} + f(n)
1596   \end{displaymath}
1597
1598   \begin{itemize}
1599     \item[$a = $]
1600       Anzahl der Unterprobleme in der Rekursion
1601
1602     \item[$\textstyle{\frac{1}{b}} = $]
1603       Teil des Originalproblems, welches wiederum durch alle Unterprobleme
1604       repräsentiert wird
1605
1606     \item[$f(n) = $]
1607       Kosten (Aufwand, Nebenkosten), die durch die Division des Problems und
1608       die Kombination der Teillösungen entstehen
1609   \end{itemize}
1610   \footcite{wiki:master-theorem}
1611   \footcite[Seite 19-35 (PDF 11-24)]{aud:fs:2}
1612 }

```

\liMasterFaelle

```

1613 \def\liMasterFaelle{
1614   \begin{description}
1615     \item[1. Fall:]
1616        $T(n) \in \liTheta{n^{\log_{sb{b}}a}}$ 
1617
1618       \hfill falls \liBedingungEins
1619       für  $\varepsilon > 0$ 
1620
1621     \item[2. Fall:]
1622        $T(n) \in \liTheta{n^{\log_{sb{b}}a} \cdot \log n}$ 
1623
1624       \hfill falls \liBedingungZwei
1625
1626     \item[3. Fall:]
1627        $T(n) \in \liTheta{f(n)}$ 
1628
1629       \hfill falls \liBedingungDrei
1630       für  $\varepsilon > 0$ 
1631       und ebenfalls für ein  $c$  mit  $0 < c < 1$  und alle hinreichend großen  $n$ 
1632       gilt:
1633        $a \cdot f(\textstyle{\frac{n}{b}}) \leq c \cdot f(n)$ 
1634     \end{description}
1635 }

```

\liMasterVariablenDeklaration

```

1636 \def\liMasterVariablenDeklaration#1#2#3{
1637   \begin{description}
1638     \item[Allgemeine Rekursionsgleichung:] \strut
1639
1640     \liRekursionsGleichung
1641
1642     \item[Anzahl der rekursiven Aufrufe ( $a$ ):] \strut
1643
1644     #1
1645
1646     \item[Anteil Verkleinerung des Problems ( $b$ ):] \strut
1647
1648     um  $\frac{1}{\#2}$  also  $b = \#2$ 
1649
1650     \item[Laufzeit der rekursiven Funktion ( $f(n)$ ):] \strut

```

```

1651
1652     $#3$
1653
1654     \item[Ergibt folgende Rekursionsgleichung:] \strut
1655
1656      $T(n) = T_{\#1}^{\#2} + \#3$ 
1657 \end{description}
1658 }

\liMasterFallRechnung
1659 \def\liMasterFallRechnung#1#2#3{
1660 \begin{description}
1661 \item[1. Fall:] \liBedingungEins:
1662
1663     #1
1664
1665 \item[2. Fall:] \liBedingungZwei:
1666
1667     #2
1668
1669 \item[3. Fall:] \liBedingungDrei:
1670
1671     #3
1672 \end{description}
1673 }

\liMasterExkurs
1674 \def\liMasterExkurs{
1675 \begin{liExkurs}[Master-Theorem]
1676 \liMasterVariablen
1677
1678 \noindent
1679 Dann gilt:
1680
1681 \liMasterFaelle
1682 \end{liExkurs}
1683 }

\liMasterWolframLink Link zu Wolframalpha (+ durch Prozent 2B ersetzen)
1684 \def\liMasterWolframLink#1{
1685 Berechne die Rekursionsgleichung auf WolframAlpha:
1686 \href{https://www.wolframalpha.com/input/?i=\#1}{WolframAlpha}
1687 }

1688

```

2.24 mathe.sty

```
1689 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1690 \ProvidesPackage{lehramt-informatik-mathe}[2020/06/10]
1691
1692 % for example \ltimes \rtimes
1693 %\RequirePackage{amssymb}
1694 \RequirePackage{amsmath}
1695
1696 %%
1697 % \mlq \mrq
1698 %%
1699 \DeclareMathSymbol{\mlq}{\mathord}{operators}{``}
1700 \DeclareMathSymbol{\mrq}{\mathord}{operators}{`}
1701
```

2.25 minimierung.sty

```

1702 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1703 \ProvidesPackage{lehramt-informatik-mathe}[2021/03/13 Für den
1704 Minimierungsalgorithmus von einem NEA zu einem DEA]

1705 \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 & \l \\
\Z05 & \Z15 & \Z25 & \f2 \l \\
\Z15 & \Z05 & \Z35 & \f2 \l \\
\Z23 & \Z44 & \Z55 & \l \\
\Z24 & \Z44 & \Z55 & \l \\
\Z34 & \Z44 & \Z55 & \l \\
\end{liUebergangsTabelle}

\liFussnote

1706 \def\liFussnote#1{\$x_{#1}\$}

1707 \def\li@fussnote@text#1#2{
1708 \liFussnote{#1}
1709 \quad
1710 {\footnotesize #2}
1711 }

\liFussnoteEinsText
1712 \def\liFussnoteEinsText{
1713 \li@fussnote@text{1}
1714 {Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.}
1715 }

\liFussnoteZweiText
1716 \def\liFussnoteZweiText{
1717 \li@fussnote@text{2}
1718 {Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt.}
1719 }

\liFussnoteDreiText
1720 \def\liFussnoteDreiText{
1721 \li@fussnote@text{3}

```

```

1722 {In weiteren Iterationen markierte Zustände.}
1723 }

\liFussnoteVierText
1724 \def\liFussnoteVierText{
1725   \li@fussnote@text{4}
1726   {...}
1727 }

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



1728 \def\liFussnoten{
1729   \bigskip
1730
1731   \noindent
1732   \liFussnoteEinsText
1733
1734   \noindent
1735   \liFussnoteZweiText
1736
1737   \noindent
1738   \liFussnoteDreiText
1739
1740   \noindent
1741   \liFussnoteVierText
1742 }

\liLeereZelle \liLeereZelle:  $\emptyset$ 
Let-Abkürzung: \let\l=\liLeereZelle
1743 \def\liLeereZelle{\$\emptyset\$}

\liZustandsPaarVariablenName
1744 \def\liZustandsPaarVariablenName{z}

\liZustandsPaar
1745 \def\liZustandsPaar#1#2{
1746   $(
1747     \liZustandsPaarVariablenName_#1,
1748     \liZustandsPaarVariablenName_#2
1749   )$
1750 }

liUebergangsTabelle
1751 \renewcommand{\arraystretch}{1.4}
1752 \NewDocumentEnvironment{ liUebergangsTabelle } { m m } {
1753   \liPseudoUeberschrift{Übergangstabelle}
1754   \begin{center}
1755     \begin{tabular}{r|l|l}
1756       \textbf{Zustandspaar} & \textbf{\#1} & \textbf{\#2} \\ \hline
1757     \end{tabular}
1758   \end{center}
1759 }
1760 }

liUeberschriftDreiecksTabelle \liUeberschriftDreiecksTabelle: Minimierungstabelle (Table filling)

1761 \ExplSyntaxOn
1762 \def\liUeberschriftDreiecksTabelle{
1763   \liPseudoUeberschrift{Minimierungstabelle~(Table~filling)}
1764 }

```

\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. —

```

1765 \def\liMinimierungErklaerung{
1766   %\footcite[Seite~19]{koenig}
1767   \liParagraphMitLinien{
1768     Der~Minimierungs-Algorithmus~(auch~Table-Filling-Algorithmus-genannt)~
1769     trägt~in~seinem~Verlauf~eine~Markierung~in~alle~
1770     diejenigen~Zellen~der~Tabelle~ein,~die~zueinander~nicht~äquivalente~
1771     Zustände~bezeichnen.~Die~Markierung~„ $x_{\sb{n}}$ “~in~einer~Tabellenzelle~( $i$ ,~
1772      $j$ )~bedeutet~dabei,~dass~das~Zustandspaar~( $i$ ,~ $j$ )~in~der~ $k$ -ten~
1773     Iteration~des~Algorithmus~markiert~wurde~und~die~Zustände~ $i$ ~und~ $j$ ~
1774     somit~zueinander~( $k - 1$ )~äquivalent,~aber~nicht~ $k$ -äquivalent~und~
1775     somit~insbesondere~nicht~äquivalent~sind.~Bleibt~eine~Zelle~bis~zum~Ende~
1776     unmarkiert,~sind~die~entsprechenden~Zustände~zueinander~äquivalent.
1777   }
1778 }
1779 \ExplSyntaxOff
1780

```

2.26 normalformen.sty

```

1781 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1782 \ProvidesPackage{lehramt-informatik-normalformen}[2020/12/10]
1783 \Hilfsmakros zum Setzen von Normalformen, Funktionalen Abhängigkeiten,
1784 \Attributhülle]
1785 \liLadePakete{mathe}
1786 \directlua{
1787   helfer = require('lehramt-informatik-helfer')
1788   normalformen = require('lehramt-informatik-normalformen')
1789 }
```

Let-Abkürzungen

```

\let\ah=\liAttributHuelle
\let\ahL=\liAttributHuelleLinksReduktion
\let\fa=\liFunktionaleAbhaengigkeit
\let\FA=\liFunktionaleAbhaengigkeiten
\let\m=\liAttributMenge
\let\r=\liRelation
```

```

1790 \def\liTeilen#1{
1791   \directlua{tex.print(normalformen.teilen('#1'))}
1792 }
```

```

\liAttributHuelle Let-Abkürzung: \let\ah=\liAttributHuelle
                  Regulärer Ausdruck zum Konvertieren AttrHülle\((.*)\) \ah{$1}
1793 \def\liAttributHuelleOhneMathe#1{\text{AttrHülle}(\#1)}
1794 \def\liAttributHuelle#1{
1795   \ifmmode
1796     \liAttributHuelleOhneMathe{\#1}
1797   \else
1798     $\liAttributHuelleOhneMathe{\#1}$
1799   \fi
1800 }
```

```

\liAttributMenge Let-Abkürzung: \let\m=\liAttributMenge
1801 \def\liAttributMenge#1{\{ \textit{\#1} \}}
```

liAHuelle

```

1802 \NewDocumentEnvironment{ liAHuelle } { +b } {
1803   \begin{group}
1804   \footnotesize
1805   \begin{multline*}
1806     \#1
1807   \end{multline*}
1808   \end{group}
1809 } { }
```

```

AttributHuelleLinksReduktion Nur innerhalb von liAHuelle zu verwenden bzw. multiline
Let-Abkürzung: \let\ahL=\liAttributHuelleLinksReduktion
                  \ahL{ursprüngliche linke Attributmenge}{ohne dieses Attribut}{Ergebnis}
1810 \def\liAttributHuelleLinksReduktion#1#2#3{
1811   \shoveleft{
1812     \liAttributHuelleOhneMathe{FA, \liAttributMenge{\liAttributMenge{\#1} - \liAttributMenge{\#2}}
1813   } \\\
1814   \shoveright{
1815     \liAttributMenge{\#3}
1816   } \\\
1817 }
```

```

\liFunktionaleAbhaengigkeit Let-Abkürzung: \let\fa=\liFunktionaleAbhaengigkeit
    \liFunktionaleAbhaengigkeit{A, B -> C, D} Regulärer Ausdruck zum Konvertieren:
    \$(.*) \rightarrow (.*)\$ \fa{$1 -> $2}
1818 \def\liFunktionaleAbhaengigkeit#1{%
1819     \directlua{normalformen.drucke_funk_abhaengigkeit('#1')}%
1820 }

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

Let-Abkürzung: \let\FA=\liFunktionaleAbhaengigkeiten
    Regulärer Ausdruck zum Konvertieren: \item \$(.*) \rightarrow (.*)\$ \$(.*) \rightarrow
    $1 -> $2;
1821 \NewDocumentCommand {\liFunktionaleAbhaengigkeiten} { O{FA} m } {
1822     \par
1823     \noindent
1824     #1 $= \{$
1825     \par
1826     \directlua{normalformen.drucke_funk_abhaengigkeiten('#2')}
1827     \par
1828     \noindent$}\$
1829 }

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

```

$$R_3(A, B, C)$$

```

    Regulärer Ausdruck zum Konvertieren: \$(R.*)\((.*)\)\$ \liRelation[$1]{$2}
1830 \NewDocumentCommand {\liRelation} { O{R} m } {
1831     $\directlua{
1832         local name = helper.konvertiere_tiefgestellt('#1')
1833         tex.print(name)
1834     }\textit{\,#2\,}
1835 }
1836

```


2.27 petri.sty

1837 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 1838 \ProvidesPackage{lehramt-informatik-petri}[2020/12/03]

Let-Abkürzungen

\let\t=\liPetriTransitionsName
 \let\tp=\liPetriTransPfeile
 \let\k=\liPetriErreichKnotenDrei

1839 \RequirePackage{tikz}
 1840 \usetikzlibrary{petri,arrows.meta}

Für die Darstellungsmatrix

1841 \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

```

1842 \def\liPetriSetzeSchluessel{%
1843   \def\TmpTransitionOne{}%
1844   \def\TmpTransitionTwo{}%
1845   \def\TmpTransitionThree{}%
1846   \def\TmpTransitionFour{}%
1847   \def\TmpTransitionFive{}%
1848   \def\TmpTransitionSix{}%
1849   \def\TmpTransitionSeven{}%
1850   \def\TmpTransitionEight{}%
1851   \def\TmpTransitionNine{}%
1852   \def\TmpTransitionTen{}%
1853   \pgfkeys{/petri/.cd,
1854     p1/.store in=\TmpPlaceOne,p1/.default=0,p1,
1855     p2/.store in=\TmpPlaceTwo,p2/.default=0,p2,
1856     p3/.store in=\TmpPlaceThree,p3/.default=0,p3,
1857     p4/.store in=\TmpPlaceFour,p4/.default=0,p4,
1858     p5/.store in=\TmpPlaceFive,p5/.default=0,p5,

```

```

1859 p6/.store in=\TmpPlaceSix,p6/.default=0,p6,
1860 p7/.store in=\TmpPlaceSeven,p7/.default=0,p7,
1861 p8/.store in=\TmpPlaceEight,p8/.default=0,p8,
1862 p9/.store in=\TmpPlaceNine,p9/.default=0,p9,
1863 p10/.store in=\TmpPlaceTen,p10/.default=0,p10,
1864 t1/.store in=\TmpTransitionOne,t1/.default=activated,
1865 t2/.store in=\TmpTransitionTwo,t2/.default=activated,
1866 t3/.store in=\TmpTransitionThree,t3/.default=activated,
1867 t4/.store in=\TmpTransitionFour,t4/.default=activated,
1868 t5/.store in=\TmpTransitionFive,t5/.default=activated,
1869 t6/.store in=\TmpTransitionSix,t6/.default=activated,
1870 t7/.store in=\TmpTransitionSeven,t7/.default=activated,
1871 t8/.store in=\TmpTransitionEight,t8/.default=activated,
1872 t9/.store in=\TmpTransitionNine,t9/.default=activated,
1873 t10/.store in=\TmpTransitionTen,t10/.default=activated,
1874 scale/.store in=\TmpScale,scale/.default=0.5,
1875 x/.store in=\TmpX,x/.default=5,
1876 y/.store in=\TmpY,y/.default=5,
1877 }%
1878 }

1879 \tikzset{
1880   li petri/.style={
1881     activated/.style={
1882       very thick
1883     },
1884     inhibitor/.style={
1885       {Circle[open,length=2mm,fill=white]}-
1886     }
1887   }
1888 }

```

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

```

1889 \def\liPetriTransitionsNameOhneMathe#1{t\sb{#1}}
1890 \def\liPetriTransitionsName#1{
1891   \ifmmode
1892     \liPetriTransitionsNameOhneMathe{#1}
1893   \else
1894     $\liPetriTransitionsNameOhneMathe{#1}$
1895   \fi
1896 }

```

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

```

1897 \NewDocumentCommand{ \liPetriErreichTransition } { m m m O{} O{} } {
1898   \draw[->] (#1) edge[#4] node[pos=0.5,auto,sloped,#5]{t\sb{#3}$} (#2);
1899 }

```

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

```

1900 \def\liPetriErreichKnotenDrei#1#2#3{(#1,#2,#3)}

```

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

```

1901 \def\liPetriTransPfeile#1{ $\rightarrow \hspace{0.4cm} \liPetriTransitionsName{#1} \hspace{0.3cm} }
1902

```

2.28 potenzmengen-konstruktion.sty

```
1903 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1904 \ProvidesPackage{lehramt-informatik-potenzmengen-konstruktion}[2021/02/21 Hilfsmakros
1905 einzusetzen bei der Potenzmengen-Konstruktion (Potenzmengen-Algorithmus)]
```

```
1906 \liLadePakete{formale-sprachen}
1907 \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}
    }
  }
}
```

```
1908 \def\liZustandsMengenSammlung#1#2{
1909   \liZustandsnameGross{#1}
1910   {
1911     \footnotesize
1912     \liPotenzmenge{
1913       \str_case:nn {#1} {#2
1914       }
1915     }
1916 }
```

```
\liZustandsMengenSammlungNr
```

```
1917 \def\liZustandsMengenSammlungNr#1#2{
1918   \liZustandsnameGross{#1}
1919   {
```

```
1920     \footnotesize
1921     \liZustandsmengeNr{
1922         \str_case:nn {#1} #2
1923     }
1924 }
1925 }

1926 \ExplSyntaxOff
1927
```

2.29 pseudo.sty

1928 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 1929 \ProvidesPackage{lehramt-informatik-pseudo}[2020/12/30 Paket zum Setzen
 1930 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>

1931 \RequirePackage[german,boxruled]{algorithm2e}

1932

2.30 pumping-lemma.sty

1933 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 1934 \ProvidesPackage{lehramt-informatik-pumping-lemma}[2021/08/11 Enthält die
 1935 Definitionen für das Pumping-Lemma in der Regulären Sprache und
 1936 in der Kontextfreien Sprache]

\liPumpingRegulaer

```
1937 \def\liPumpingRegulaer{%
1938   Es sei  $L$  eine reguläre Sprache. Dann gibt es eine Zahl  $j$ , sodass für
1939   alle Wörter  $\omega$  in  $L$  mit  $|\omega| \geq j$  (jedes Wort  $\omega$  in
1940    $L$  mit Mindestlänge  $j$ ) jeweils eine Zerlegung  $\omega = uvw$ 
1941   existiert, sodass die folgenden Eigenschaften erfüllt sind:
1942
1943   \begin{enumerate}
1944     \item  $|v| \geq 1$ 
1945     (Das Wort  $v$  ist nicht leer.)
1946
1947     \item  $|uv| \leq j$ 
1948     (Die beiden Wörter  $u$  und  $v$  haben zusammen höchstens die Länge  $j$ .)
1949
1950     \item Für alle  $i = 0, 1, 2, \dots$  gilt  $uv^i w$  in  $L$ 
1951     (Für jede natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^i w$  in der
1952     Sprache  $L$ )
1953   \end{enumerate}
1954
1955   Die kleinste Zahl  $j$ , die diese Eigenschaften erfüllt, wird
1956   Pumping-Zahl der Sprache  $L$  genannt.\footcite{wiki:pumping-lemma}
1957 }
```

\liPumpingKontextfrei

```
1958 \def\liPumpingKontextfrei{%
1959   Es sei  $L$  eine kontextfreie Sprache. Dann gibt es eine Zahl  $j$ , sodass
1960   sich alle Wörter  $\omega$  in  $L$  mit  $|\omega| \geq j$  zerlegen lassen in
1961    $\omega = uvwxy$ , sodass die folgenden Eigenschaften erfüllt sind:
1962
1963   \begin{enumerate}
1964     \item  $|vx| \geq 1$ 
1965     (Die Wörter  $v$  und  $x$  sind nicht leer.)
1966
1967     \item  $|vwx| \leq j$ 
1968     (Die Wörter  $v$ ,  $w$  und  $x$  haben zusammen höchstens die Länge  $j$ .)
1969
1970     \item Für alle  $i$  in  $\mathbb{N}_0$  gilt  $uv^iwx^iy$  in  $L$  (Für jede
1971     natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^iwx^iy$  in der
1972     Sprache  $L$ )
1973   \end{enumerate}
1974 }

1975
```

2.31 quicksort.sty

```

1976 % https://tex.stackexchange.com/a/142634
1977 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1978 \ProvidesPackage{lehramt-informatik-quicksort}[2020/06/12]
1979
1980 %-----
1981 % USAGE:
1982 % \QSinitialize{comma, separated, numerical, values}
1983 % \loop
1984 % \QSpivotStep
1985 % \ifnum\value{pivotcount}>0
1986 %   \QSSortStep
1987 % \repeat
1988 %-----
1989
1990 % xintfrac does not load xinttools, this must be done explicitly if needed as here.
1991 \RequirePackage{xintfrac, xinttools}
1992
1993 \RequirePackage{tikz}
1994
1995 %-----
1996 % FIRST PART: TikZ styles and macros for the actual drawing
1997 \newcounter{cellcount}% used for coordinates of the node
1998 \newcounter{pivotcount}% when it will remain at zero, will signal the sort is finished.
1999
2000 % Styles defined by Tom Bombaldi. (modified: all share the same size)
2001 % (re-modified \bf -> \bfseries due to extremely annoying warnings from
2002 % KOMA-script which are truly a pain and do not make any sense regarding \bf:
2003 % if I want to use \bf, and know what I am doing, why should I get HARASSED
2004 % by police of LaTeX good conduct ? )
2005 \tikzset{l/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=lime!70!gray},
2006          o/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=olive!50},
2007          r/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2008 % this is the "b" style as used in the image below
2009 %   b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2010 % nicer:
2011 %   b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=white, text=magenta},
2012 %   g/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=gray, text=white, text=white},
2013
2014 % NOTE the b style was originally the same as the r(aised) style apart from
2015 % not being raised, but I find it nicer with a somewhat different
2016 % specification. I have not updated the images though.
2017
2018 % How the nodes are drawn depending on whether on the left of the pivot value
2019 % or on the right, or is a pivot value, or a raised pivot during selection phase.
2020
2021 \def\DecoLEFT #1{%
2022   \xintFor* ##1 in {#1} \do
2023     {\stepcounter{cellcount}\node[o] at (\arabic{cellcount},0) {##1};}%
2024 }
2025
2026 \def\DecoINERT #1{%
2027   \xintFor* ##1 in {#1} \do
2028     {\stepcounter{cellcount}\node[g] at (\arabic{cellcount},0) {##1};}%
2029 }
2030
2031 \def\DecoRIGHT #1{%
2032   \xintFor* ##1 in {#1} \do
2033     {\stepcounter{cellcount}\node[l] at (\arabic{cellcount},0) {##1};}%
2034 }
2035
2036 \def\DecoLEFTwithPivot #1{\stepcounter{pivotcount}%
2037   \xintFor* ##1 in {#1} \do

```

```

2038     {\stepcounter{cellcount}}%
2039     \xintifForLast {\node[r]}\node[o]} at (\arabic{cellcount},0) {##1};}%
2040 }
2041
2042 \def\DecoINERTwithPivot #1{\stepcounter{pivotcount}%
2043     \xintFor* ##1 in {#1} \do
2044     {\stepcounter{cellcount}}%
2045     \xintifForLast {\node[b]}\node[g]} at (\arabic{cellcount},0) {##1};}%
2046 }
2047
2048 \def\DecoRIGHTwithPivot #1{\stepcounter{pivotcount}%
2049     \xintFor* ##1 in {#1} \do
2050     {\stepcounter{cellcount}}%
2051     \xintifForLast {\node[r]}\node[l]} at (\arabic{cellcount},0) {##1};}%
2052 }
2053
2054 %-----
2055 % SECOND PART: the actual sorting routines.
2056
2057 \def\QS@sort@a #1{\expandafter \QS@sort@b \expandafter {\xintLength {#1}}{#1}}
2058 \def\QS@sort@b #1{\ifcase #1
2059     \expandafter\QS@sort@empty
2060     \or\expandafter\QS@sort@single
2061     \else\expandafter\QS@sort@c
2062     \fi
2063 }%
2064 \def\QS@sort@empty #1{}
2065 \def\QS@sort@single #1{\QS@Ir {#1}}
2066
2067 % This step is to pick the last as pivot.
2068 \def\QS@sort@c #1%
2069     {\expandafter\QS@sort@d\expandafter {\romannumeral0\xintnthelt {-1}{#1}}{#1}}%
2070
2071 % Here \QSLr, \QSIr, \QSR have been let to \relax.
2072 % The trick with \xintApplyUnbraced is that for example when selecting
2073 % the elements smaller than pivot, if we had been using \xintApply we
2074 % would have had at the minimum an empty brace pair. Thus we use the
2075 % "unbraced" variant, but then the \QS@select@smaller has added in
2076 % anticipation a level of braces.
2077 \def\QS@sort@d #1#2{%
2078     \QSLr {\xintApplyUnbraced {\QS@select@smaller {#1}}{#2}}}%
2079     \QSIr {\xintApplyUnbraced {\QS@select@equal {#1}}{#2}}}%
2080     \QSRr {\xintApplyUnbraced {\QS@select@greater {#1}}{#2}}}%
2081 }%
2082 \def\QS@select@smaller #1#2{\xintifLt {#2}{#1}{#{#2}}{ }}% space will stop a f-
    expansion
2083 \def\QS@select@equal #1#2{\xintifEq {#2}{#1}{#{#2}}{ }}% space will stop a f-
    expansion
2084 \def\QS@select@greater #1#2{\xintifGt {#2}{#1}{#{#2}}{ }}% space will stop a f-
    expansion
2085
2086 %
2087 % NOTE 1: thus, each comparison with the pivot is done three (!) times.
2088 %
2089 % NOTE 2: we may well end up with \QSLr {<empty>} situations. This is handled
2090 % silently by the \xintFor loops, and also when \QSLr becomes \QS@sort@a, the
2091 % latter must handle correctly an empty argument.
2092
2093 %-----
2094 % THIRD PART: the main macros \QSpivotStep, \QSsortStep and \QS@initialize.
2095
2096 % This draws all with suitable highlighting for the newly chosen pivots
2097 % (which will be shown raised)

```



```

2098 \def\QSpivotStep {\let\QSLr\DecoLEFTwithPivot
2099             \let\QSIr\DecoINERT
2100             \let\QSIrr\DecoINERT
2101             \let\QSRr\DecoRIGHTwithPivot
2102 \par\centerline{\rule[1.5mm]{0pt}{8mm}%
2103             \setcounter{cellcount}{0}\setcounter{pivotcount}{0}%
2104             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2105 }
2106
2107 % This sorts and then draws, showing where the pivot chosen in the previous
2108 % step go. Next time they will have become "inert". If pivotcount is still at
2109 % zero on exit from \QSpivotStep, then this is the signal to stop before
2110 % executing \QSsortStep.
2111 \def\QSsortStep {\def\QSLr {\noexpand\QS@sort@a}%
2112             \def\QSRr {\noexpand\QS@sort@a}%
2113             \def\QSIr {\noexpand\QSIrr}%
2114             \let\QSIrr\relax
2115             \edef\QS@list{\QS@list}%
2116             \let\QSLr\relax
2117             \let\QSRr\relax
2118             \let\QSIr\relax
2119             \edef\QS@list{\QS@list}%
2120             \let\QSLr\DecoLEFT
2121             \let\QSIr\DecoINERTwithPivot
2122             \let\QSIrr\DecoINERT
2123             \let\QSRr\DecoRIGHT
2124 \par\centerline{\rule[1.5mm]{0pt}{8mm}%
2125             \setcounter{cellcount}{0}%
2126             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2127 }
2128
2129 \def\QSinitialize #1{%
2130     % first, we convert the comma separated values into a list of braced items
2131     % we use an \edef, and anyhow many \edef's will be used later
2132     \edef\QS@list {\noexpand\QSRr {\xintCSVtoList {#1}}}%
2133     \let\QSRr\DecoRIGHT
2134     % The \QSRr marker mutated to draw the last element as
2135     % pivot and the earlier ones with the suitable style.
2136     %
2137     % The list of marked braced items \QS@list is used both for drawing
2138     % (as here) and for doing the exchange of elements during sort.
2139     \par\centerline{\rule[1.5mm]{0pt}{8mm}\setcounter{cellcount}{0}%
2140             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2141 }
2142

```

2.32 relationale-algebra.sty

```

2143 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2144 \ProvidesPackage{lehramt-informatik-relationale-algebra}[2020/12/11]
2145 \RequirePackage{amsmath}
2146 \RequirePackage{amssymb}

```

```

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

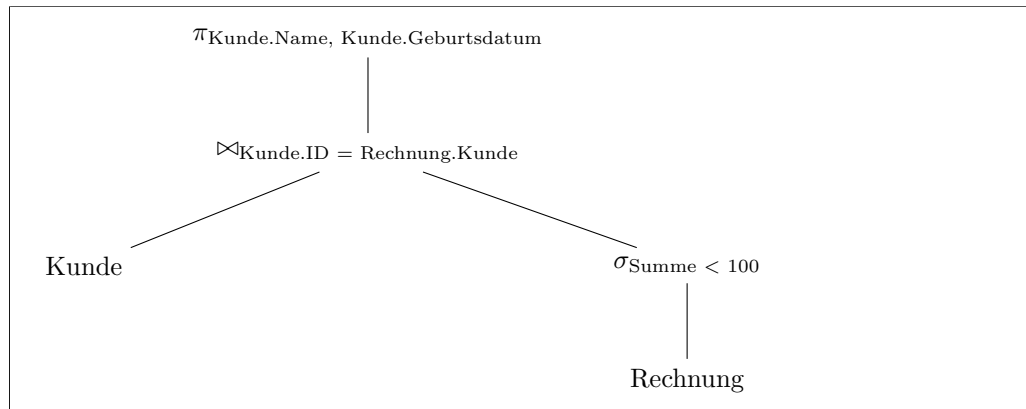
  \node[below=of pi]
    (theta join) {\bowtie_{\text{Kunde.ID = Rechnung.Kunde}}};
    edge (pi);

  \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}

```



```

2147 \RequirePackage{tikz}
2148 \usetikzlibrary{positioning}

    Privates Makros, das zwei Querstriche erzeugt.
2149 \def\o@join{\setbox0=\hbox{\bowtie}}%
2150 \rule[-.02ex]{.25em}{.4pt}\llap{\rule[\ht0]{.25em}{.4pt}}%
2151 }

```

```
\leftouterjoin A \leftouterjoin B: A ⋈ B
```

```
2152 \def\leftouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie}}
```

```
\rightouterjoin A \rightouterjoin B: A ⋉ B
```

```
2153 \def\rightouterjoin{\mathbin{\bowtie\mkern-5.8mu\o@join}}
```

```
\fullouterjoin A \fullouterjoin B: A ⋈⋉ B
```

```
2154 \def\fullouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie\mkern-5.8mu\o@join}}
```

```
2155
```

2.33 rmodell.sty

```

2156 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2157 \ProvidesPackage{lehramt-informatik-rmodell}[2020/09/01
2158 Makros und Umgebungen zum Setzen des Relationenmodells beim Thema
2159 Datenbanken.]
2160 \RequirePackage{soul}

```

Let-Abkürzungen

```

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

```

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

```
2161 \def\liPrimaer#1{\ul{#1}}
```

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

```
2162 \def\liFremd#1{{\setul{-0.9em}{}\ul{#1}}}
```

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

```

2163 \def\li@Rmodell@Schrift{\footnotesize\ttfamily}
2164 \ExplSyntaxOn
2165 \NewDocumentEnvironment { liRmodell }
2166 { +b }
2167 {
2168   \medskip
2169   {
2170     \linespread{2}
2171     \setlength{\parindent}{0pt}
2172     \li@Rmodell@Schrift#1
2173   }
2174   \medskip
2175 } {}
2176 \ExplSyntaxOff

```

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

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

```

2177 \def\liRelationMenge#1#2{
2178 \noindent
2179 #1 : \[ #2 ]\}
2180 \par
2181 }

```

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

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

```
2182 \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}

```

```
2183 \NewDocumentEnvironment { liRelationenSchemaFormat }{ +b } {} {}
```

```
2184
```

2.34 sortieren.sty

```
2185 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2186 \ProvidesPackage{lehramt-informatik-sortieren}[2020/06/10 Schaubilder
2187 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}
}
```



```
2188 \RequirePackage{tikz}
2189 \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.

```
2190 \def\liVertauschen#1{
2191   \directlua{
2192     local sortieren = require('lehramt-informatik-sortieren')
2193     sortieren('#1')
2194   }
2195 }
```

`\liSortierPfeil`

```
2196 \def\liSortierPfeil#1#2{
2197   \draw[-latex] ([xshift=1mm]reihe.#1 north) -- ++(0,0.25) -| ([xshift=-1mm]reihe.#2 north);
2198 }
```

`\liSortierPfeilUnten`

```
2199 \def\liSortierPfeilUnten#1#2{
2200   \draw[-latex] ([xshift=1mm]reihe.#1 south) -- ++(0,-0.25) -| ([xshift=-1mm]reihe.#2 south);
2201 }
```

`\liSortierMarkierung`

```
2202 \def\liSortierMarkierung#1#2{\node[
2203   draw,
2204   very thick,
2205   fit=(reihe.#1) (reihe.#2),
2206   inner sep=0pt
2207 ] {}
2208 }

2209 \tikzset{
2210   li sortierung zahlenreihe/.style={
2211     draw,
2212     thin,
2213     font=\large,
2214     rectangle split horizontal,
2215     rectangle split,
2216   }
2217 }
```

```

2218 % https://tex.stackexchange.com/a/140895
2219 \RequirePackage{forest,xstring}
2220 \usetikzlibrary{calc}
2221
2222 \makeatletter
2223 \pgfmathdeclarefunction{strrepeat}{2}{%
2224   \begingroup\pgfmathint{#2}\pgfmath@count\pgfmathresult
2225   \let\pgfmathresult\pgfutil@empty
2226   \pgfutil@loop\ifnum\pgfmath@count>0\relax
2227     \expandafter\def\expandafter\pgfmathresult\expandafter{\pgfmathresult#1}%
2228     \advance\pgfmath@count-1\relax
2229   \pgfutil@repeat\pgfmath@smuggleone\pgfmathresult\endgroup}
2230 \makeatother
2231
2232 \def\myNodes{}
2233
2234 \ExplSyntaxOn
2235 \newcommand*\sortList[1]{%
2236   \clist_sort:Nn#1{\int_compare:nNnTF{##1}>{##2}\sort_return_swapped:\sort_return_same:}}
2237 \ExplSyntaxOff
2238
2239 \forestset{
2240   sort/.code={%
2241     \pgfmathparse{level()>\forestSortLevel}%
2242     \ifnum\pgfmathresult=0
2243       \StrSubstitute{\forestov{content}}{ }{,}{\myList}%
2244       \sortList\myList
2245       \StrSubstitute{\myList}{,}{ }{\myList}%
2246       \pgfmathparse{strrepeat("1",level())}%
2247       \xappto\myNodes{\noexpand\node at ({\forestov{name}}|-m)!-1!({\forestov{name}}$)
2248         (m\forestov{name}) {\myList}}%
2249       \pgfmathparse{level()=\forestSortLevel}%
2250       \ifnum\pgfmathresult=1
2251         \forestOget{\forestov{@first}}{name}\forestFirst
2252         \forestOget{\forestov{@last}}{name}\forestLast
2253         \xappto\myNodes{{[<-]edge (\forestOv{\forestov{@first}}{name})
2254           \ifx\forestFirst\forestLast\else edge (\forestOv{\forestov{@last}}{name})\fi}}%
2255       \fi
2256       \ifnum\forestov{@parent}=0\else
2257         \xappto\myNodes{edge (m\forestOv{\forestov{@parent}}{name})}%
2258       \fi
2259       \gappto\myNodes{;}%
2260     \fi}}
2261
2262 \forestset{sort level/.code=%
2263   \pgfmathparse{#1}\let\forestSortLevel\pgfmathresult
2264   \pgfmathparse{strrepeat("1",\forestSortLevel+1)}\let\forestOnes\pgfmathresult}
2265

```

2.35 spalten.sty

```
2266 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2267 \ProvidesPackage{lehramt-informatik-spalten}[2020/12/07 Lädt das Paket
2268 „multicol“, damit mehrspaltiger Satz mit Hilfe der Umgebung „multicols“
2269 realisiert werden kann.]
2270 \RequirePackage{multicol}
```

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

```
2271 \def\liSpaltenUmbruch{\vfill\strut\columnbreak}
```

```
2272
```

2.36 struktogramm.sty

```
2273 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2274 \ProvidesPackage{lehramt-informatik-struktogramm}[2021/01/31 Lädt das
2275 Paket struktex zum Setzen von Struktogrammen]
2276 \RequirePackage{struktex}
2277
```

2.37 syntax.sty

```
2278 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2279 \ProvidesPackage{lehramt-informatik-syntax}[2020/06/10 Ein Hüll-Paket um
2280 `minted`, das sich um die Syntax-Hervorhebung des Java-Codes kümmert.]
2281 \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
```

```
2282 \ExplSyntaxOn
2283 \directlua{
2284   syntax = require('lehramt-informatik-syntax')
2285   syntax.importiere_konfiguration('tex_repo_lokaler_pfad', '\LehramtInformatikRepository')
2286   syntax.importiere_konfiguration('github_domain', '\LehramtInformatikGithubDomain')
2287   syntax.importiere_konfiguration('github_raw_domain', '\LehramtInformatikGithubRawDomain')
2288   syntax.importiere_konfiguration('github_tex_repo', '\LehramtInformatikGithubTexRepo')
2289   syntax.importiere_konfiguration('github_code_repo', '\LehramtInformatikGithubCodeRepo')
2290   syntax.importiere_konfiguration('git_branch', '\LehramtInformatikGitBranch')
2291 }
2292 \RequirePackage{hyperref}
2293 \RequirePackage{minted}
2294 % pygmentize -L styles
2295 \usemintedstyle{colorful}
2296 %\BeforeBeginEnvironment{minted}{\begin{mdframed}}
2297 %\AfterEndEnvironment{minted}{\end{mdframed}}
2298 %\setminted{breaklines=true,linenos}
2299 \setminted{
2300   breaklines=true,
2301   linenos,
2302   fontsize=\footnotesize,
2303 }
```

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

Let-Abkürzung: `\let\j=\liJavaCode`

```
2304 \def\liJavaCode#1{\mintinline{java}|#1|}
```

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

```
2305 \def\liLatexCode#1{\mintinline{latex}|#1|}
```

```
2306 \def\li@GithubLink#1#2{
2307   \begin{flushright}
2308     \tiny
2309     Code-Beispiel~auf~Github~ansehen:~
2310     \href{#1}{\nolinkurl{#2}}
2311   \end{flushright}
2312 }
```

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

```
2313 \NewDocumentCommand{\liJavaDatei}{ O{firstline=3} m }{
2314   \inputminted[#1]{java}{
2315     \directlua{
2316       syntax.drucke_absoluten_pfad('#2', false)
2317     }
2318   }
2319   \li@GithubLink
```



```

2320     {\directlua{syntax.drucke_github_url('#2', false)}}
2321     {\directlua{syntax.drucke_relativen_repo_pfad('#2', false)}}
2322 }

\liJavaTestDatei Eine komplette Java-Test-Datei einbinden, die Verzeichnis ./Code/src/test/java/org/bschlangaul
liegt.
2323 \NewDocumentCommand{\liJavaTestDatei}{ O{firstline=3} m }{
2324   \inputminted[#1]{java}{
2325     \directlua{
2326       syntax.drucke_absoluten_pfad('#2', true)
2327     }
2328   }
2329   \li@GithubLink
2330   {\directlua{syntax.drucke_github_url('#2', true)}}
2331   {\directlua{syntax.drucke_relativen_repo_pfad('#2', true)}}
2332 }

\liJavaExamen \liJavaExamen{66116}{2015}{03}{Kunde} Regulärer Ausdruck zum Konvertieren: \liJavaDatei([.
\liJavaExamen$1{$2}{$3}{$4}{$5}
2333 \NewDocumentCommand{\liJavaExamen}{ O{firstline=3} m m m m }{
2334   \inputminted[#1]{java}{
2335     \directlua{
2336       syntax.drucke_absoluten_examens_pfad('#2', '#3', '#4', '#5')
2337     }
2338   }
2339
2340   \li@GithubLink
2341   {\directlua{syntax.drucke_github_examens_url('#2', '#3', '#4', '#5')}}
2342   {\directlua{syntax.drucke_relativen_examens_repo_pfad('#2', '#3', '#4', '#5')}}
2343 }

\liAssemblerCode
2344 \def\liAssemblerCode#1{\mintinline{asm}|#1|}

\liAssemblerDatei \liAssemblerDatei{relativer-pfad}: Relativer Pfad zu einer Assembler-Datei.
2345 \NewDocumentCommand{\liAssemblerDatei}{ m }{
2346   \inputminted{asm}{#1}
2347 }

\liMinispracheDatei \liMinispracheDatei{relativer-pfad}: Relativer Pfad zu einer Minisprachen-Datei
(Hochsprache für die Minimaschine von Albert Wiedemann).
2348 \NewDocumentCommand{\liMinispracheDatei}{ m }{
2349   \inputminted{componentpascal}{#1}
2350 }

\liHaskellCode \liHaskellCode{haskell}: Zum Setzen von Haskell-Code.
2351 \def\liHaskellCode#1{\mintinline{haskell}|#1|}

\liHaskellDatei \liHaskellDatei{relativer-pfad}: Relativer Pfad zu einer Haskell-Datei.
2352 \NewDocumentCommand{\liHaskellDatei}{ m }{
2353   \inputminted{haskell}{#1}
2354 }

2355 \ExplSyntaxOff

\liSqlCode \liHaskellCode{sql}: Zum Setzen von SQL-Code.
Let-Abkürzung: \let\s=\liSqlCode
2356 \def\liSqlCode#1{\mintinline{sql}|#1|}
2357

```

2.38 syntaxbaum.sty

```
2358 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2359 \ProvidesPackage{lehramt-informatik-syntaxbaum}[2021/02/14 Zum Setzen von
2360 Syntaxbäumen mit Hilfe des Pakets tikz-qtrees]
2361 \RequirePackage{tikz-qtrees}
2362
2363 \tikzset{li parsetree/.style={
2364     every internal node/.style={
2365         draw,circle
2366     },
2367     every leaf node/.style={
2368         draw,rectangle
2369     },
2370 }
2371 }
2372
```

2.39 synthese-algorithmus.sty

```
2373 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2374 \ProvidesPackage{lehramt-informatik-synthese-algorithmus}[2021/03/19
2375 Hilfsmakros zum Setzen des Synthese-Algorithmuses zur Umformung einer
2376 Relation in die 3. Normalform]

2377 \liLadePakete{normalformen,mathe,typographie}
2378 \ExplSyntaxOn

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

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

```

2379 \def\liSyntheseUeberschrift#1{
2380   {
2381     \bfseries
2382     \sffamily
2383     \str_case:nn {#1} {
2384       {1} {Kanonische-Überdeckung}
2385       {1-1} {Linksreduktion}
2386       {1-2} {Rechtsreduktion}
2387       {1-3} {Löschen-leerer-Klauseln}
2388       {1-4} {Vereinigung}
2389       {2} {Relationsschemata-formen}
2390       {3} {Schlüssel-hinzufügen}
2391       {4} {Entfernung-überflüssiger-Teilschemata}
2392     }
2393   }
2394 }

```

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

```

2395 \def\li@synthese@erklaerung@texte#1{
2396   \str_case:nn {#1} {
2397     {1} {
2398       Die~kanonische-Überdeckung---also~die~kleinst~mögliche~noch~
2399       äquivalente~Menge~von~funktionalen~Abhängigkeiten~kann~in~vier~
2400       Schritten~erreicht~werden.
2401     }
2402     {1-1} {
2403       Führe~für~jede~funktionale~Anhängigkeit~
2404       $\alpha\rightarrow\beta$~in~F$~die~Linksreduktion~durch,~
2405       überprüfe~also~für~alle~
2406       $A\in\alpha$,~ob~$A$~überflüssig~ist,~d.~h.~ob~
2407       $\beta\subseteqq\liAttributHuelle{F,\alpha\cup A}$.
2408     }
2409     {1-2} {
2410       Führe~für~jede~(verbliebene)~funktionale~Abhängigkeit~$\alpha\rightarrow\beta$~die~Rechtsreduktion~durch,~überprüfe~also~für~
2411       alle~$B\in\beta$,~ob~$B\in\liAttributHuelle{F\cup(\alpha\rightarrow\beta)}$,~
2412       $\alpha\rightarrow\beta\cup(\alpha\rightarrow B)$,~
2413       $\alpha$~gilt.~In~diesem~Fall~ist~B~auf~der~rechten~Seite~
2414       überflüssig~und~kann~eliminiert~werden,~\dh~$\alpha\rightarrow\beta$~wird~durch~$\alpha\rightarrow(\beta\cup B)$~
2415       ersetzt.
2416     }
2417     {1-3} {
2418       Entferne~die~funktionalen~Abhängigkeiten~der~Form~$\alpha\rightarrow\emptyset$,~die~im~2.~Schritt~möglicherweise~
2419       entstanden~sind.
2420     }
2421     {1-4} {
2422       Fasse~mittels~der~Vereinigungsregel~funktionale~Abhängigkeiten~
2423       der~Form~$\alpha\rightarrow\beta\sb{1},\dots,\alpha\rightarrow\beta\sb{n}$,~so~dass~$\alpha\rightarrow\beta\sb{1}\cup\dots\cup\beta\sb{n}$~verbleibt.
2424     }
2425     {2} {
2426       % Kemper Seite 197
2427       Erzeuge~für~jede~funktionale~Abhängigkeit~$\alpha\rightarrow\beta\in F\sb{c}$~ein~Relationenschema~$\mathcal{R}\sb{\alpha}$~
2428       :=~$\alpha\cup\beta$.
2429     }
2430     {3} {
2431       Falls~eines~der~in~Schritt~2.~erzeugten~Schemata~$\mathcal{R}\sb{\alpha}$~
2432       einen~Schlüsselkandidaten~von~$\mathcal{R}$~bezüglich~$F\sb{c}$~
2433     }
2434   }
2435 }

```

```

2439 enthält,~sind~wir~fertig,~sonst~wähle~einen~Schlüsselkandidaten~
2440  $\mathcal{K} \sim \text{subseq} \mathcal{R}$ ~aus~und~definiere~folgendes~
2441 zusätzliche~Schema:~ $\mathcal{R} \setminus \mathcal{K} \sim \mathcal{K}$ ~
2442 und~ $\mathcal{F} \setminus \mathcal{K} \sim \emptyset$ 
2443 }
2444 {4} {
2445 Eliminiere~diejenigen~Schemata~ $\mathcal{R} \setminus \alpha$ ~,~die~in~einem~
2446 anderen~Relationenschema~ $\mathcal{R} \setminus \alpha'$ ~enthalten~sind,~d.h.~
2447  $\mathcal{R} \setminus \alpha \sim \text{subseq} \mathcal{R} \setminus \alpha'$ .
2448 }
2449 }
2450 }
2451 \def\liSyntheseErklaerung#1{
2452 {
2453 \itshape
2454 \footnotesize
2455 \liParagraphMitLinien{\li@synthese@erklaerung@texte{#1}}
2456 }
2457 }

```

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

```

2458 \def\liSyntheseUeberErklaerung#1{
2459 \liSyntheseUeberschrift{#1}\par
2460 \liSyntheseErklaerung{#1}
2461 }

2462 \ExplSyntaxOff
2463

```

2.40 tabelle.sty

2464 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2465 \ProvidesPackage{lehramt-informatik-tabelle}[2020/12/05 Lädt das Paket tabluarx]

2466 \RequirePackage{tabularx}

2467

2.41 typographie.sty

```
2468 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2469 \ProvidesPackage{lehramt-informatik-typographie}[2021/03/20 Typographische Makros,
2470 die das Erscheinungsbild verändern. Die Schriftdefinition sind in
2471 formatierung.sty definiert.]
```

```
2472 \ExplSyntaxOn
```

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

```
2473 \RequirePackage{fontawesome}
```

```
\liErledigt \liErledigt: ☑
```

```
2474 \let\liErledigt=\faCheckSquareO
```

```
\liNichtsZuTun \liNichtsZuTun: ☑ Nichts zu tun
```

```
2475 \def\liNichtsZuTun{\faCheckSquareO{}}~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 —

```
2476 \def\liParagraphMitLinien#1{
2477   \noindent
2478   \vrule height 2pt depth -1.6pt width 0.4cm
2479   \enspace
2480   #1
2481   \enspace
2482   \leaders\vrule height 2pt depth -1.6pt \hfill \null
2483   \par
2484   \medskip
2485 }
```

```
2486 \ExplSyntaxOff
```

```
2487
```

2.42 uml.sty

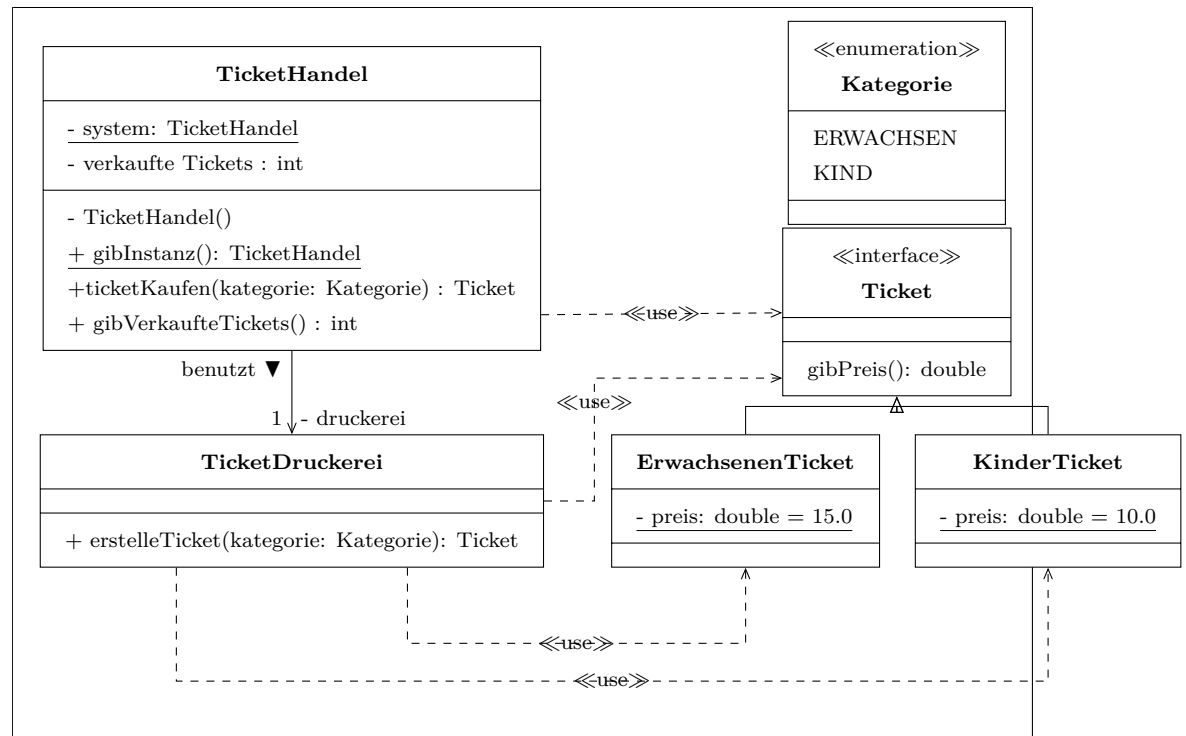
```

2488 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2489 \ProvidesPackage{lehramt-informatik-uml}[2020/06/13 Ein Hüll-Paket um
2490 `tikz-uml', das einige Design-Einstellungen vornimmt und manche
2491 Erweiterung bereitstellt]

2492 \RequirePackage{tikz-uml}
2493 \RequirePackage{tikz-uml-activity}
2494 % Not compatible with wasysym
2495 %\RequirePackage{mathabx}
2496 \RequirePackage{wasysym}
2497 \usetikzlibrary{positioning}

2498 \tikzumlset{
2499   fill class=white!0,
2500   font=\footnotesize,
2501   fill object=white!0,
2502   fill note=white!0,
2503   fill state=white!0,
2504   % Use case
2505   fill usecase=white!0,
2506   fill system=white!0,
2507 }

```



\liUmlLeserichtung

```

2508 \NewDocumentCommand{ \liUmlLeserichtung } { 0{dir=right} m } {
2509   \def\@liDirLeft{}
2510   \def\@liDirRight{ \RIGHTarrow}
2511   \pgfkeys{/lese/dir/.is choice}
2512   \pgfkeys{/lese/dir/up/.code={\def\@liDirRight{ \UParrow}}}
2513   \pgfkeys{/lese/dir/down/.code={\def\@liDirRight{ \DOWNarrow}}}
2514   \pgfkeys{/lese/dir/left/.code={\@liDirRight{}\def\@liDirLeft{ \LEFTarrow }}}
2515   \pgfkeys{/lese/dir/right/.code={\def\@liDirRight{ \RIGHTarrow}}}
2516
2517   \def\@liPos{above}
2518   \pgfkeys{/lese/pos/.code={\def\@liPos{##1}}}
2519
2520   \def\@liDistance{0cm}
2521   \pgfkeys{/lese/distance/.code={\def\@liDistance{##1}}}

```



```

2522
2523 \pgfkeys{/lese/.cd,#1}
2524
2525 \node[\@liPos = \@liDistance of #2-middle] {
2526   \@liDirLeft{\footnotesize#2}\@liDirRight
2527 };
2528 }
2529

```

2.43 vollstaendige-induktion.sty

```
2530 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2531 \ProvidesPackage{lehramt-informatik-vollstaendige-induktion}[2021/07/01
2532 Hilfsmakros zum Setzen der Vollständigen Induktion, vor allem die
2533 Überschriften für die einzelnen Schritte]
```

Lade häufig benötigte Pakete

```
2534 \RequirePackage{lehramt-informatik-typographie}
2535 \RequirePackage{lehramt-informatik-mathe}
2536 \RequirePackage{lehramt-informatik-syntax}
2537 \ExplSyntaxOn
```

`\liInduktionAnfang`

```
2538 \def\liInduktionAnfang{
2539   \liPseudoUeberschrift{Induktionsanfang}
2540
2541   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
2542   \liParagraphMitLinien{
2543     Beweise,~dass~ $A(1)$ ~eine~wahre~Aussage~ist.
2544   }
2545 }
```

`\liInduktionVoraussetzung`

```
2546 \def\liInduktionVoraussetzung{
2547   \liPseudoUeberschrift{Induktionsvoraussetzung}
2548
2549   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
2550   \liParagraphMitLinien{
2551     Die~Aussage~ $A(k)$ ~ist~wahr~für~ein~beliebiges~ $k \in \mathbb{N}$ $.
2552   }
2553 }
```

`\liInduktionSchritt`

```
2554 \def\liInduktionSchritt{
2555   \liPseudoUeberschrift{Induktionsschritt}
2556
2557   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
2558   \liParagraphMitLinien{
2559     Beweise,~dass~wenn~ $A(n=k)$ ~wahr~ist,~
2560     auch~ $A(n=k+1)$ ~wahr~sein~muss.
2561   }
2562 }

2563 \ExplSyntaxOff
2564
```

2.44 wasserfall.sty

```
2565 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2566 \ProvidesPackage{lehramt-informatik-wasserfall}[2020/06/10]
2567 \RequirePackage{tikz}
2568 \tikzset{wasserfall/.style={
2569   >=stealth,
2570   node distance = 2mm and -8mm,
2571   start chain = A going below right,
2572   every node/.style = {
2573     draw,
2574     text width=24mm,
2575     minimum height=12mm,
2576     align=center,
2577     inner sep=1mm,
2578     fill=white,
2579     drop shadow={fill=black},
2580     on chain=A
2581   },
2582 }}
2583 \usetikzlibrary{chains,positioning,shadows}
2584
```

2.45 wpkalkuel.sty

2585 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2586 \ProvidesPackage{lehramt-informatik-wpkalkuel}[2020/06/13]

Let-Abkürzungen

\let\wp=\liWpKalkuel

\let\equivalent=\liWpEquivalent

\let\erklaerung=\liWpErklaerung

2587 \RequirePackage{amsmath}

2588 \ExplSyntaxOn

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

2589 \def\liWpKalkuelOhneMathe#1#2{

2590 \text{wp}(\texttt{\scriptsize"#1"},\thinspace #2)

2591 }

2592 \def\liWpKalkuel#1#2{

2593 \ifmmode

2594 \liWpKalkuelOhneMathe{#1}{#2}

2595 \else

2596 \$\liWpKalkuelOhneMathe{#1}{#2}\$

2597 \fi

2598 }

\MatheEnv

2599 \def\MatheEnv#1{

2600 \medskip

2601

2602 \hspace{1em}#1

2603

2604 \medskip

2605 }

\Mathe

2606 \def\Mathe#1{

2607 \MatheEnv{#1\$}

2608 }

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

2609 \def\liWpEquivalent#1{

2610 \MatheEnv{\$\equiv\$\hspace{1em}\$#1\$}

2611 }

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

2612 \newlength{\@Skip@Erklaerung@Reset}

2613 \def\liWpErklaerung#1{

2614 \setlength{\@Skip@Erklaerung@Reset}{\leftskip}

2615 \setlength{\leftskip}{0.5cm}

2616

2617 \par

2618 \noindent

2619 {

2620 \scriptsize

2621 #1

2622 }

2623 \par

2624

2625 \setlength{\leftskip}{\@Skip@Erklaerung@Reset}

2626 }

\liWpErklaerungVerzweigung

```

2627 \def\liWpErklaerungVerzweigung{
2628   $\liWpKalkuelOhneMathe{if~\{-b~\}~then~\{-a1~\}~else~\{-a2~\}}{Q}
2629   \equiv
2630   (b \land \liWpKalkuelOhneMathe{a1}{Q})
2631   \lor
2632   (\neg b \land \liWpKalkuelOhneMathe{a2}{Q})$
2633 }

2634 \ExplSyntaxOff

2635

```

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		
\# 109	\alpha 2404, 2406, 2407, 2410, 2412, 2413, 2414, 2415, 2416, 2420, 2426, 2427, 2432, 2433, 2434, 2437, 2445, 2446, 2447	\bowtie 2149, 2152, 2153, 2154
\, 331, 386, 1037, 1039, 1525, 1526, 1527, 1834	\arabic 1100, 2023, 2028, 2033, 2039, 2045, 2051	\Box 159
\@Skip@Erklaerung@Reset ... 2612, 2614, 2625	\arraystretch 1751	\boxtimes 463
\@afterheading 1492		C
\@afterindentfalse . 1492		\c 1183, 1184
\@liDirLeft 2509, 2514, 2526		\cdot 1569, 1622, 1633
\@liDirRight 2510, 2512, 2513, 2514, 2515, 2526		\centerline 1214, 2102, 2124, 2139
\@liDistance 2520, 2521, 2525	B	\chapter 1090, 1091
\@liPos .. 2517, 2518, 2525	\BeforeBeginEnvironment 2296	\char 1399
\\ 596, 616, 617, 620, 621, 624, 625, 697, 698, 699, 774, 776, 794, 803, 844, 857, 858, 859, 864, 865, 866, 882, 1399, 1756, 1813, 1816	\begin 614, 656, 671, 695, 719, 753, 772, 782, 790, 810, 838, 855, 876, 893, 908, 1016, 1020, 1106, 1181, 1209, 1218, 1225, 1333, 1391, 1443, 1448, 1456, 1481, 1486, 1497, 1507, 1511, 1594, 1598, 1614, 1637, 1660, 1675, 1754, 1755, 1805, 1943, 1963, 2104, 2126, 2140, 2296, 2307	\clist 226, 270, 271, 284, 288, 2236
\{ 207, 972, 982, 994, 995, 1000, 1014, 1036, 1255, 1801, 1824, 2179, 2628	\begingroup 1392, 1803, 2224	\columnbreak 2271
\} 207, 972, 982, 994, 995, 1002, 1020, 1040, 1256, 1801, 1828, 2179, 2628	\beschriftung 1423, 1427, 1431, 1435, 1439, 1441	\cs 287, 306, 330, 331, 368, 380, 1475
_ 27, 35, 44, 46, 287, 306, 330, 331, 345, 346, 352, 355, 358, 368, 380	\beta 2404, 2407, 2411, 2412, 2413, 2416, 2426, 2427, 2428, 2433, 2434	\csname 1168, 1171
A	\bf 2001, 2002, 2003	\cup 995, 2413, 2427, 2434
\addbibresource 1369, 1370, 1371, 1372, 1373, 1374, 1375, 1376, 1377, 1378	\bfseries .. 475, 1090, 1092, 2001, 2007, 2009, 2011, 2012, 2381	D
\advance 2228	\bigskip 48, 364, 598, 603, 1401, 1729	\DeclareMathSymbol 1699, 1700
\AfterEndEnvironment 2297	\bool 309, 332	\DecoINERT 2026, 2099, 2100, 2122
\allsectionsfont ... 1086		\DecoINERTwithPivot 2042, 2121
\Alpha 1100		\DecoLEFT 2021, 2120
\alph 1100, 1101		\DecoLEFTwithPivot 2036, 2098
		\DecoRIGHT 2031, 2123, 2133
		\DecoRIGHTwithPivot 2048, 2101
		\definecolor 1088
		\delta 65, 107, 165, 207, 986
		\dh 1527, 2415
		\directlua 58, 137, 195, 200, 966, 981, 1001, 1009, 1017, 1023, 1786, 1791, 1819, 1826, 1831, 2191,

1669, 1944, 1947, 1950, 1964, 1967, 1970	\itshape 519, 2453	\LehramtInformatikRepository\liBedingungZwei 4, 7, 10, 13, 16, 1369, 1370, 1371, 1372, 1373, 1374, 1375, 1376, 1377, 1378, 2285 \leq 1633, 1947, 1967 \let 984, 985, 1393, 2098, 2099, 2100, 2101, 2114, 2116, 2117, 2118, 2120, 2121, 2122, 2123, 2133, 2225, 2263, 2264, 2474 \li@chomsky@erklaerung@texte 485, 521 \li@fussnote@text 1707, 1713, 1717, 1721, 1725 \li@GithubLink 2306, 2319, 2329, 2340 \li@mget . 1170, 1174, 1194 \li@minc 1173, 1195 \li@mset 1167, 1175, 1184, 1187 \li@numdiscs 1179, 1188, 1194 \li@Rmodell@Schrift 2163, 2172, 2182 \li@sequence .. 1180, 1193 \li@synthese@erklaerung@texte 2395, 2455 \liAbleitung 1009 liAdditum (environment) 1446 liAHuelle (environment) 1802 \liAlphabet 994 liAntwort (environment) 1417 \liAnweisung 1340 \liAssemblerCode . . . 2344 \liAssemblerDatei . . 2345 \liAttribut 2182 \liAttributHuelle 1793, 2407, 2412 \liAttributHuelleLinksReduktion 1810 \liAttributHuelleOhneMathe 1793, 1796, 1798, 1812 \liAttributMenge 1801, 1812, 1815 \liAufgabe 3 \liAufgabenTitel 23 \liAusdruck 1033 \liAutomat 61 \liAutomatenKante . . . 93 \liBandAlphabet 995 \liBedingung 1341 \liBedingungDrei 1589, 1629, 1669 \liBedingungEins 1583, 1618, 1661 \liBedingungFalsch . 1343 \liBedingungWahr . . . 1342 1586, 1624, 1665 \liBeschriftung 1406 \liChomskyErklaerung 485, 526 \liChomskyUeberErklaerung 524 \liChomskyUeberschrift 473, 525 \liCpmEreignis 534 \liCpmFruehesterI . . . 591 \liCpmSpaetesterI . . . 590 \liCpmVon 574 \liCpmVonOhneMathe 574, 577, 579 \liCpmVonZu 566 \liCpmVonZuOhneMathe 566, 569, 571 \liCpmVorgang 551 \liCpmZu 582 \liCpmZuOhneMathe 582, 585, 587 liDiagramm (environ- ment) 1505 liEinbettung (environ- ment) 1416 \liEntwurfsAbstrakteFabrikUml 613 \liEntwurfsAdapterAkteure 670 \liEntwurfsAdapterUml 655 \liEntwurfsBeobachterAkteure 718 \liEntwurfsBeobachterUml 694 \liEntwurfsDekoriererUml 752 \liEntwurfsEinzelstueckAkteure 781 \liEntwurfsEinzelstueckUml 771 \liEntwurfsErbauerAkteure 809 \liEntwurfsErbauerUml 789 \liEntwurfsFabrikmethodeUml 837 \liEntwurfsKompositumUml 854 \liEntwurfsModellPraesentationSteuerungUml 875 \liEntwurfsZustandAkteure 907 \liEntwurfsZustandUml 892 \liEpsilon 980 \liErAttribute 936, 950, 952 \liErDatenbankName . . 955 \liErEntity . 934, 938, 940 \liErledigt 2474 \liErMpAttribute 949 \liErMpEntity 937 \liErMpRelationship . 943	
J	\j 1183, 1184, 1186, 1187, 1188, 1193, 1194, 1195			
K	\k 1193 \keys 31, 70, 82, 112, 122, 170, 180, 292, 538, 542, 556, 561, 1061, 1068			
L	\l 63, 64, 65, 66, 67, 68, 71, 72, 73, 74, 75, 77, 79, 84, 85, 86, 87, 88, 89, 104, 105, 106, 107, 108, 109, 110, 113, 114, 115, 116, 117, 118, 119, 125, 126, 127, 128, 129, 130, 131, 162, 163, 164, 165, 166, 167, 168, 171, 172, 173, 174, 175, 176, 177, 183, 184, 185, 186, 187, 188, 189, 536, 539, 544, 545, 548, 553, 554, 557, 558, 563, 1056, 1057, 1058, 1059, 1062, 1063, 1064, 1065, 1071, 1072, 1073, 1074, 1349, 1350, 1351, 1478, 1479, 1480, 1487 \labelenumi 1101 \labelenumii 1102 \labelitemi 1095 \labelitemii 1096 \labelitemiii 1097 \labelitemiv 1098 \land 2630, 2632 \LARGE 1090 \large 1214, 2213 \leaders 2482 \left 1534 \LEFTarrow 2514 \leftarrow 582 \leftouterjoin 2152 \leftskip 2614, 2615, 2625 \LehramtInformatikGitBranch 373, 2290 \LehramtInformatikGithubCodeRepositorium 2289 \LehramtInformatikGithubDomaine 2286 \LehramtInformatikGithubRawDomaine 371, 2287 \LehramtInformatikGithubTexRepository 372, 2288			

<code>\liErRelationship</code> ...	<code>\liMasterFaelle</code> 1613 , 1681	<code>\liPseudoUeberschrift</code>
..... 935 , 944 , 946	<code>\liMasterFallRechnung</code> 1400 ,
<code>\liExamensAufgabe</code> 6 1659	1450 , 1451 , 1753 ,
<code>\liExamensAufgabeA</code> ... 15	<code>\liMasterVariablen</code> ..	1763 , 2539 , 2547 , 2555
<code>\liExamensAufgabeTA</code> .. 12 1593 , 1676	<code>\liPumpingKontextfrei</code>
<code>\liExamensAufgabeTTA</code> .. 9	<code>\liMasterVariablenDeklaration</code>	1958
<code>liExkurs (environment)</code> 1454 1636	<code>\liPumpingRegulaer</code> . 1937
<code>\liFalsch</code> 464	<code>\liMasterWolframLink</code> 1684	<code>liQuellen (environment)</code>
<code>\liFlaci</code> 1043	<code>\liMenge</code> ... 71 , 72 , 74 , 1475
<code>\liFremd</code> 2162	113 , 114 , 115 , 119 ,	<code>\liRekursionsGleichung</code>
<code>\liFunktionaleAbhaengigkeit</code>	171 , 172 , 173 , 177 , 1580 , 1640
..... 1818	972 , 1023 , 1062 , 1063	<code>\liRelation</code> 1830
<code>\liFunktionaleAbhaengigkeiten</code>	<code>\liMengeOhneMathe</code> ...	<code>liRelationenSchemaFormat</code>
..... 1821 972 , 975 , 977	(environment) 2183
<code>\liFussnote</code> ... 1706 , 1708	<code>\liMinimierungErklaerung</code>	<code>\liRelationMenge</code> ... 2177
<code>\liFussnoteDreiText</code> 1765	<code>\liRichtig</code> 463
..... 1720 , 1738	<code>\liMinispracheDatei</code> 2348	<code>liRmodell (environment)</code>
<code>\liFussnoteEinsText</code> .	<code>\linespread</code> 2170 2163
..... 1712 , 1732	<code>\liNichtsZuTun</code> 2475	<code>\liRundeKlammer</code> . 1533 ,
<code>\liFussnoteLink</code> 1522	<code>\liO</code> 1556 , 1584	1537 , 1547 , 1557 , 1571
<code>\liFussnoten</code> 1728	<code>\liOmega</code> 1546 , 1590	<code>\liSetzeAufgabenTitel</code> . 25
<code>\liFussnoteUrl</code> . 890 , 1518	<code>\liOmegaOhneMathe</code> ...	<code>\liSortierMarkierung</code> 2202
<code>\liFussnoteVierText</code> 1546 , 1551 , 1553	<code>\liSortierPfeil</code> 2196
..... 1724 , 1741	<code>\liOOhneMathe</code>	<code>\liSortierPfeilUnten</code> 2199
<code>\liFussnoteZweiText</code> 1556 , 1561 , 1563	<code>\liSpaltenUmbruch</code> .. 2271
..... 1716 , 1735	<code>\liParagraphMitLinien</code>	<code>\liSqlCode</code> 2356
<code>\liGrammatik</code> 1054	. 521 , 1767 , 2455 ,	<code>\listen@punkt</code> .. 1475 , 1487
<code>liGraphenFormat (envi-</code>	2476 , 2542 , 2550 , 2558	<code>\liStrich</code> 1206
<code>ronment)</code> 1160	<code>\liPetriErreichKnotenDrei</code>	<code>\liSyntheseErklaerung</code>
<code>\liHanoi</code> 1167 1900 2395 , 2460
<code>\liHaskellCode</code> 2351	<code>\liPetriErreichTransition</code>	<code>\liSyntheseUeberErklaerung</code>
<code>\liHaskellDatei</code> 2352 1897 2458
<code>\liInduktionAnfang</code> . 2538	<code>\liPetriSetzeSchluessel</code>	<code>\liSyntheseUeberschrift</code>
<code>\liInduktionSchritt</code> 2554 1842 2379 , 2459
<code>\liInduktionVoraussetzung</code>	<code>\liPetriTransitionsName</code>	<code>\liT</code> 1566 , 1581 , 1595 , 1656
..... 2546 1889 , 1901	<code>\liTeilen</code> 1790
<code>\liJavaCode</code> 2304	<code>\liPetriTransitionsNameOhneMathe</code>	<code>\liLiteratur</code> 1357 , 1381
<code>\liJavaDatei</code> 2313	... 1889 , 1892 , 1894	<code>\liTheta</code> 1536 ,
<code>\liJavaExamen</code> 2333	<code>\liPetriTransPfeile</code> 1901	1587 , 1616 , 1622 , 1627
<code>\liJavaTestDatei</code> ... 2323	<code>\liPolynomiellReduzierbar</code>	<code>\liThetaOhneMathe</code> ...
<code>liKasten (environment)</code> 1105 1224	... 1536 , 1541 , 1543
<code>\liKellerAutomat</code> 102	<code>\liPotenzmenge</code>	<code>\liTOhneMathe</code>
<code>\liKellerKante</code> 140 981 , 985 , 1912	... 1566 , 1575 , 1577
<code>\liKellerUebergang</code> ..	<code>\liPotenzmengeOhneMathe</code>	<code>\liTuringKante</code> 203
..... 135 , 141 982 , 983 , 984	<code>\liTuringLeerzeichen</code>
<code>\liKontrollCode</code> 1344	<code>\liPrimaer</code> 2161 159 , 167
<code>liKontrollflussgraph</code>	<code>\liProblemBeschreibung</code>	<code>\liTuringMaschine</code> ... 160
(environment) 1332 1208	<code>\liTuringUeberfuehrung</code>
<code>\liKontrollKnotenPfad</code>	<code>\liProblemClique</code> ... 1231 206
..... 1346	<code>\liProblemName</code>	<code>\liTuringUebergaenge</code>
<code>\liKontrollTextzeileKnoten</code> 1207 , 1214 , 198 , 204
..... 1345 , 1350	1226 , 1228 , 1241 ,	<code>\liTuringUebergangZelle</code>
<code>\liKurzeTabellenLinie</code> 596	1252 , 1253 , 1261 , 1262 193
<code>\liLadeAllePakete</code> ... 228	<code>\liProblemSat</code> 1260	<code>\liUeberfuehrungsFunktion</code>
<code>\liLadePakete</code>	<code>\liProblemSubsetSum</code> 986
. 54 , 57 , 224 , 229 , 1251 , 1260	<code>\liUeberfuehrungsFunktionOhneMathe</code>
472 , 533 , 1204 ,	<code>\liProblemVertexCover</code> 986 , 989 , 991
1705 , 1785 , 1906 , 2377 1231 , 1239	<code>liUebergangsTabelle</code>
<code>\liLatexCode</code> 2305	<code>\liProduktionen</code> 1022 , 1064	(environment) 1751
<code>\liLeereZelle</code> 1743	<code>liProduktionsRegeln</code>	<code>\liUeberschriftDreiecksTabelle</code>
<code>liLernkartei (environ-</code>	(environment) 1010 1761
<code>ment)</code> 1495	<code>liProjektSprache (envi-</code>	<code>\liUmlLeserichtung</code> . 2508
<code>\liMasterExkurs</code> 1674	<code>ronment)</code> 1415	<code>\liVertauschen</code> 2190

<code>\liWortInSprache</code>	597	<code>\mrq</code>	1697, 1700	<code>\Omega</code>	1547	
<code>\liWortNichtInSprache</code> <u>602</u>		<code>\msg</code>	39, 391	<code>\omega</code> 1939, 1940, 1960, 1961		
<code>\liWpEquivalent</code>	<u>2609</u>	<code>\myList</code>		<code>\or</code>	2060	
<code>\liWpErklaerung</code>	<u>2612</u>		2243, 2244, 2245, 2248			
<code>\liWpErklaerungVerzweigung</code>	<u>2627</u>	<code>\myNodes</code>	2232, 2247, 2253, 2257, 2259	P		
<code>\liWpKalkuel</code>	<u>2589</u>			<code>\pagestyle</code>	1115	
<code>\liWpKalkuelOhneMathe</code>	2589, 2594, 2596, 2628, 2630, 2632	N		<code>\par</code> 342, 363, 525, 1044, 1052, 1407, 1469, 1492, 1500, 1822, 1825, 1827, 2102, 2124, 2139, 2180, 2459, 2483, 2617, 2623		
<code>\liZustandsBuchstabe</code>	996, 1005, 1007, 1027, 1029	<code>\NeedsTeXFormat</code>	1, 19, 52, 220, 398, 459, 466, 530, 593, 608, 926, 962, 1079, 1119, 1128, 1133, 1162, 1200, 1272, 1355, 1361, 1383, 1529, 1689, 1702, 1781, 1837, 1903, 1928, 1933, 1977, 2143, 2156, 2185, 2266, 2273, 2278, 2358, 2373, 2464, 2468, 2488, 2530, 2565, 2585		<code>\paragraph</code>	1092
<code>\liZustandsBuchstabeGross</code>	<u>997</u> , 1006, 1008	<code>\neg</code>	2632	<code>\parindent</code>	2171	
<code>\liZustandsmenge</code>	<u>984</u>	<code>\negthinspace</code>	1534	<code>\path</code>	94, 141, 204, 563	
<code>\liZustandsmengeNr</code>	<u>998</u> , 1921	<code>\newcounter</code>	1997, 1998	<code>\pgfkeys</code>	1853, 2511, 2512, 2513, 2514, 2515, 2518, 2521, 2523	
<code>\liZustandsmengeNrGross</code>	<u>1006</u>	<code>\NewDocumentCommand</code>	62, 103, 140, 161, 203, 224, 535, 552, 597, 602, 1033, 1055, 1224, 1347, 1359, 1518, 1522, 1821, 1830, 1897, 2313, 2323, 2333, 2345, 2348, 2352, 2508	<code>\pgfmath@count</code>	2224, 2226, 2228	
<code>\liZustandsMengenSammlung</code>	<u>1908</u>	<code>\NewDocumentEnvironment</code>	1010, 1105, 1160, 1332, 1415, 1416, 1419, 1446, 1454, 1476, 1495, 1505, 1752, 1802, 2165, 2183	<code>\pgfmath@smuggleone</code> 2229		
<code>\liZustandsMengenSammlungNr</code>	<u>1917</u>	<code>\newlength</code>	2612	<code>\pgfmathdeclarefunction</code>	2223	
<code>\liZustandsmengeOhneMathe</code>	984	<code>\node</code>	548, 1340, 1345, 2023, 2028, 2033, 2039, 2045, 2051, 2202, 2247, 2525	<code>\pgfmathhint</code>	2224	
<code>\liZustandsname</code>	<u>1007</u>	<code>\noexpand</code>	2111	<code>\pgfmathparse</code>	1174, 2241, 2246, 2249, 2263, 2264	
<code>\liZustandsnameGross</code>	1008, 1909, 1918	<code>\noindent</code>	343, 599, 604, 1013, 1402, 1404, 1408, 1412, 1440, 1468, 1470, 1483, 1499, 1501, 1509, 1678, 1731, 1734, 1737, 1740, 1823, 1828, 2178, 2477, 2618	<code>\pgfmathresult</code>	1175, 2224, 2225, 2227, 2229, 2242, 2250, 2263, 2264	
<code>\liZustandsnameTiefgestellt</code>	<u>1025</u>	<code>\nolinkurl</code>	2310	<code>\pgfutil@empty</code>	2225	
<code>\liZustandsPaar</code>	<u>1745</u>	<code>\normalsize</code>	1092	<code>\pgfutil@loop</code>	2226	
<code>\liZustandsPaarVariablenName</code>	1744, 1747, 1748	<code>\notin</code>	605	<code>\pgfutil@repeat</code>	2229	
<code>\llap</code>	2150	<code>\null</code>	2482	<code>\preceq</code>	1227	
<code>\log</code>	1584, 1587, 1590, 1616, 1622	O		<code>\prime</code>	1206	
<code>\loop</code>	1983			<code>\printbibliography</code>	1381	
<code>\lor</code>	2631			<code>\ProvidesPackage</code>	2, 20, 53, 221, 399, 460, 467, 531, 594, 609, 927, 963, 1080, 1120, 1129, 1134, 1163, 1201, 1273, 1356, 1362, 1384, 1530, 1690, 1703, 1782, 1838, 1904, 1929, 1934, 1978, 2144, 2157, 2186, 2267, 2274, 2279, 2359, 2374, 2465, 2469, 2489, 2531, 2566, 2586	
<code>\ltimes</code>	1692					
M				Q		
<code>\makeatletter</code>	1491, 2222			<code>\QS@list</code>	2104, 2115, 2119, 2126, 2132, 2137, 2140	
<code>\makeatother</code>	1493, 2230			<code>\QS@select@equal</code>	2079, 2083	
<code>\marginpar</code>	939, 945, 951, 1398			<code>\QS@select@greater</code>	2080, 2084	
<code>\mathbb</code>	1262, 1970, 2551			<code>\QS@select@smaller</code>	2075, 2078, 2082	
<code>\mathbin</code>	2152, 2153, 2154					
<code>\mathcal</code>	1557, 2433, 2438, 2440, 2441, 2442					
<code>\Mathe</code>	<u>2606</u>					
<code>\MatheEnv</code>	<u>2599</u> , 2607, 2610					
<code>\mathord</code>	1699, 1700					
<code>\mdfsetup</code>	1104, 1424, 1428, 1432, 1436					
<code>\medskip</code>	1216, 1409, 1411, 1485, 1513, 2168, 2174, 2484, 2600, 2604					
<code>\memp</code>	<u>1398</u>					
<code>\mintinline</code>	2304, 2305, 2344, 2351, 2356					
<code>\mkern</code>	2152, 2153, 2154					
<code>\mlq</code>	1697, 1699					

<code>\QS@sort@a</code>	574, 1901, 2404,	<code>\text</code> 77, 79, 182, 1793, 2590
2057, 2090, 2111, 2112	2411, 2413, 2416,	<code>\textbf</code> 934, 1232,
<code>\QS@sort@b</code> 2057, 2058	2421, 2426, 2427, 2432	1241, 1252, 1261,
<code>\QS@sort@c</code> 2061, 2068	<code>\rightouterjoin</code> 2153	1403, 1410, 1441,
<code>\QS@sort@d</code> 2069, 2077	<code>\Roman</code> 1100	1469, 1484, 1500, 1756
<code>\QS@sort@empty</code> . 2059, 2064	<code>\roman</code> 1100, 1102	<code>\textcolor</code> 1344
<code>\QS@sort@single</code> 2060, 2065	<code>\romannumeral</code> 2069	<code>\textit</code>
<code>\QSinitialize</code>	<code>\rtimes</code> 1692	844, 857, 858, 859,
. . . 1982, 2094, 2129	<code>\rule</code> 2102, 2124, 2139, 2150	860, 1510, 1801, 1834
<code>\QSIr</code> . 2065, 2071, 2079,		<code>\textsc</code> 1207
2099, 2113, 2118, 2121		<code>\textsf</code> 1403, 1484
<code>\QSIrr</code> 2100, 2113, 2114, 2122	S	<code>\textstyle</code> 1602, 1633
<code>\QSLr</code> 2071,	<code>\sb</code> 67, 77,	<code>\texttt</code> 957, 1207, 1341,
2078, 2089, 2090,	79, 108, 166, 501,	1342, 1343, 1344, 2590
2098, 2111, 2116, 2120	502, 506, 509, 510,	<code>\thepage</code> 1114
<code>\QSpivotStep</code>	511, 1027, 1029,	<code>\theparagraph</code> 1092
1984, 2094, 2098, 2109	1584, 1587, 1590,	<code>\Theta</code> 1537
<code>\QSR</code> 2071	1616, 1622, 1771,	<code>\thinspace</code> 2590
<code>\QSRr</code> 2080,	1889, 1898, 2426,	<code>\tikz</code> 1345
2101, 2112, 2117,	2427, 2428, 2433,	tikz: bbaum 23
2123, 2132, 2133, 2134	2437, 2438, 2441,	tikz: li binaer baum 21
<code>\QSortStep</code>	2442, 2445, 2446, 2447	<code>\tikzchildnode</code> 419
1986, 2094, 2110, 2111	<code>\scriptscriptstyle</code> . .	<code>\tikzparentnode</code> 419
<code>\quad</code> 1709 566, 574, 582	<code>\tikzset</code> 96,
	<code>\scriptsize</code> 1046, 1285,	143, 209, 404, 430,
R	1292, 1298, 2590, 2620	1139, 1276, 1879,
<code>\raisebox</code> 1345	<code>\section</code> 46	2005, 2209, 2363, 2568
<code>\relax</code> 1393,	<code>\seq</code> . 1349, 1350, 1351,	<code>\tikzumlset</code> 2498
2071, 2114, 2116,	1478, 1479, 1480, 1487	<code>\times</code> 207
2117, 2118, 2226, 2228	<code>\setbox</code> 2149	<code>\tiny</code> 940, 946,
<code>\renewcommand</code>	<code>\setcounter</code>	952, 1344, 1398, 2308
. 1095, 1096,	1093, 2103, 2125, 2139	<code>\titleformat</code> . . 1090, 1092
1097, 1098, 1101,	<code>\setganttlinklabel</code> . .	<code>\titlespacing</code> 1091
1102, 1116, 1117, 1751	1123, 1124, 1125, 1126	<code>\tl</code> . 29, 37, 63, 64, 65,
<code>\repeat</code> 1987	<code>\setlength</code>	66, 67, 68, 71, 72,
<code>\RequirePackage</code> 55, 158,	2171, 2614, 2615, 2625	73, 74, 75, 77, 79,
222, 226, 395, 401,	<code>\setmainfont</code> 1083	104, 105, 106, 107,
402, 462, 532, 612,	<code>\setmainlanguage</code> 396	108, 109, 110, 113,
929, 931, 932, 970,	<code>\setminted</code> 2298, 2299	114, 115, 116, 117,
971, 1081, 1082,	<code>\setsansfont</code> 1084	118, 119, 162, 163,
1085, 1087, 1089,	<code>\setul</code> 2162	164, 165, 166, 167,
1094, 1103, 1110,	<code>\sffamily</code> 476,	168, 171, 172, 173,
1121, 1122, 1131,	1090, 1092, 1194, 2382	174, 175, 176, 177,
1135, 1136, 1137,	<code>\shoveleft</code> 1811	285, 289, 307, 311,
1165, 1166, 1205,	<code>\shoveright</code> 1814	312, 313, 316, 321,
1274, 1358, 1363,	<code>\Sigma</code> 64, 105,	322, 323, 334, 335,
1364, 1380, 1387,	163, 994, 995, 1057	336, 337, 348, 354,
1388, 1389, 1417,	<code>\sigma</code> 499, 501, 502	357, 360, 369, 383,
1532, 1693, 1694,	<code>\SLASH</code> 1399	536, 539, 544, 545,
1839, 1841, 1931,	<code>\small</code> 1508	553, 554, 557, 558,
1991, 1993, 2145,	<code>\sort</code> 2236	1035, 1056, 1057,
2146, 2147, 2160,	<code>\sortList</code> 2235, 2244	1058, 1059, 1062,
2188, 2219, 2270,	<code>\square</code> 464	1063, 1064, 1065, 1567
2276, 2281, 2292,	<code>\stepcounter</code> 2023, 2028,	<code>\TmpPlaceEight</code> 1861
2293, 2361, 2466,	2033, 2036, 2038,	<code>\TmpPlaceFive</code> 1858
2473, 2492, 2493,	2042, 2044, 2048, 2050	<code>\TmpPlaceFour</code> 1857
2495, 2496, 2534,	<code>\str</code> . . 477, 486, 1421,	<code>\TmpPlaceNine</code> 1862
2535, 2536, 2567, 2587	1913, 1922, 2383, 2396	<code>\TmpPlaceOne</code> 1854
<code>\right</code> 1534	<code>\StrSubstitute</code> . 2243, 2245	<code>\TmpPlaceSeven</code> 1860
<code>\RIGHTarrow</code> . . . 2510, 2515	<code>\strut</code> 1638, 1642,	<code>\TmpPlaceSix</code> 1859
<code>\rightarrow</code> 600, 605	1646, 1650, 1654, 2271	<code>\TmpPlaceTen</code> 1863
<code>\rightarrowtail</code> 207,	<code>\subseq</code> 2407, 2440, 2447	<code>\TmpPlaceThree</code> 1856
489, 494, 502, 506,		<code>\TmpPlaceTwo</code> 1855
508, 509, 511, 566,	T	
	<code>\tableofcontents</code> . . . 1394	

<code>\TmpScale</code>	1874	762, 773, 792, 793,	V	
<code>\TmpTransitionEight</code> ..		796, 843, 846, 856,	<code>\value</code>	1985
.....	1850, 1871	862, 863, 877, 878,	<code>\varepsilon</code>	
<code>\TmpTransitionFive</code> ..		879, 894, 895, 896, 897	478, 489, 490, 980,	
.....	1847, 1868	<code>\umldep</code>	1584, 1590, 1619, 1630	
<code>\TmpTransitionFour</code> ..		851	<code>\vfill</code>	2271
.....	1846, 1867	<code>\umlHVHagg</code> 714, 767, 872	<code>\vrule</code>	2478, 2482
<code>\TmpTransitionNine</code> ..		<code>\umlinherit</code>	<code>\vspace</code>	
.....	1851, 1872	664, 704, 800, 841, 849	1015, 1019, 1455, 1473	
<code>\TmpTransitionOne</code> ...		<code>\umlnote</code> 666, 802, 904		
.....	1843, 1864	<code>\umlreal</code>		
<code>\TmpTransitionSeven</code> ..		662, 712	X	
.....	1849, 1870	<code>\umlsimpleclass</code>	<code>\xappto</code> ..	2247, 2253, 2257
<code>\TmpTransitionSix</code> 630, 631,	<code>\xdef</code>	1168
.....	1848, 1869	632, 636, 638, 639,	<code>\xintApply</code>	2073
<code>\TmpTransitionTen</code> ...		640, 657, 791, 839, 840	<code>\xintApplyUnbraced</code> ..	
.....	1852, 1873	<code>\umlstatic</code>	2072, 2078, 2079, 2080	
<code>\TmpTransitionThree</code> ..		774	<code>\xintCSVtoList</code>	2132
.....	1845, 1866	<code>\umluniagg</code>	<code>\xintFor</code>	
<code>\TmpTransitionTwo</code> ...		<code>\umluniassoc</code>	2022, 2027, 2032,	
.....	1844, 1865	650, 663, 799, 886, 887	2037, 2043, 2049, 2090	
<code>\TmpX</code>	1875	<code>\umlVHuniassoc</code> ..	<code>\xintifEq</code>	2083
<code>\TmpY</code>	1876	<code>\umlVHVdep</code>	2039, 2045, 2051
<code>\ttfamily</code>	2163	.. 644, 645, 647, 648	<code>\xintifGt</code>	2084
		<code>\umlVHVinherit</code> ... 627,	<code>\xintifLt</code>	2082
		628, 633, 634, 641,	<code>\xintLength</code>	2057
		642, 764, 765, 870, 871	<code>\xintntheft</code>	2069
		<code>\umlVHVreal</code>		
		.. 758, 759, 899, 900	Z	
U		<code>\UParrow</code>	<code>\ZB</code>	1526
<code>\ul</code>	935, 2161, 2162	2512	<code>\zB</code>	1525
<code>\umlagg</code>	902	<code>\url</code>	<code>\zustandsnamens@liste</code>	
<code>\umlassoc</code>	888	1519	998, 1005, 1006
<code>\umlclass</code> ...	615, 619,	<code>\usemintedstyle</code>		
623, 658, 659, 660,		2295		
696, 701, 706, 709,		<code>\usetikzlibrary</code>		
754, 755, 756, 761,		56, 403, 930, 1138,		
		1275, 1840, 2148,		
		2189, 2220, 2497, 2583		