

lehramt-informatik

Hermine Bschlangaul *

August 20, 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	39
2.10.10	Zustand	40
2.11	er.sty	42
2.12	formale-sprachen.sty	44
2.13	formatierung.sty	48
2.13.1	Schriftarten / Typographie	48
2.13.2	Farben	48
2.13.3	Überschriften	48
2.13.4	Listen	48
2.13.5	Kasten	48
2.13.6	Header	48

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

2.14	gantt.sty	49
2.15	grafik.sty	50
2.16	graph.sty	51
2.17	hanoi.sty	53
2.18	komplexitaetstheorie.sty	54
2.19	kontrollflussgraph.sty	56
2.20	kopf-fusszeilen.sty	58
2.21	literatur-dummy.sty	59
2.22	literatur.sty	60
2.23	makros.sty	61
2.24	master-theorem.sty	65
2.25	mathe.sty	69
2.26	minimierung.sty	70
2.27	normalformen.sty	73
2.28	petri.sty	75
2.29	potenzmengen-konstruktion.sty	77
2.30	pseudo.sty	79
2.31	pumping-lemma.sty	80
2.32	quicksort.sty	81
2.33	relationale-algebra.sty	84
2.34	rmodell.sty	85
2.35	sortieren.sty	86
2.36	spalten.sty	88
2.37	struktogramm.sty	89
2.38	syntax.sty	90
2.39	syntaxbaum.sty	92
2.40	synthese-algorithmus.sty	93
2.41	tabelle.sty	96
2.42	typographie.sty	97
2.43	uml.sty	98
2.44	vollstaendige-induktion.sty	100
2.45	wasserfall.sty	102
2.46	wpkalkuel.sty	103

3 Index

104

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[$\langle automaten-name \rangle$]{ $\langle zustaende=Z, alphabet=\Sigma, delta=\delta, ende=E, start=z_0 \rangle$ }`

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

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

```

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

```

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

```

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

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

```

2.3.2 Kellerautomat

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

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

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

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



```
\liKellerAutomat \liKellerAutomat[\langle automaten-name \rangle]
{\langle zustaeende=Z,alphabet=Sigma,kelleralphabet=Gamma,delta=delta,start=z0,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-qtree.]

401 \RequirePackage{tikz}

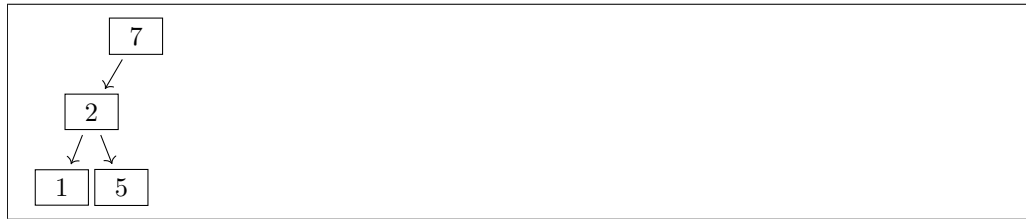
    für li binaer baum
402 \RequirePackage{tikz-qtree}

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

2.5.1 Binärbaum

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

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



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

2.5.2 AVL-Baum

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



2.5.3 B-Baum

```

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

```



```

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

```

2.6 checkbox.sty

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

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

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

465
```


2.7 chomsky-normalform.sty

```

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

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

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

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

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

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

```

523 }

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

524 \def\liChomskyUeberErklaerung#1{

525 \liChomskyUeberschrift{#1}\par

526 \liChomskyErklaerung{#1}

527 }

528 \ExplSyntaxOff

529

2.8 cpm.sty

```

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

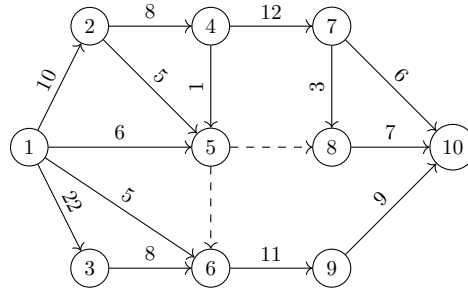
```

Let-Abkürzungen

```

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

```



```

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

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

\liCpmVorgang[schein]{5}{6}{}
\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
```

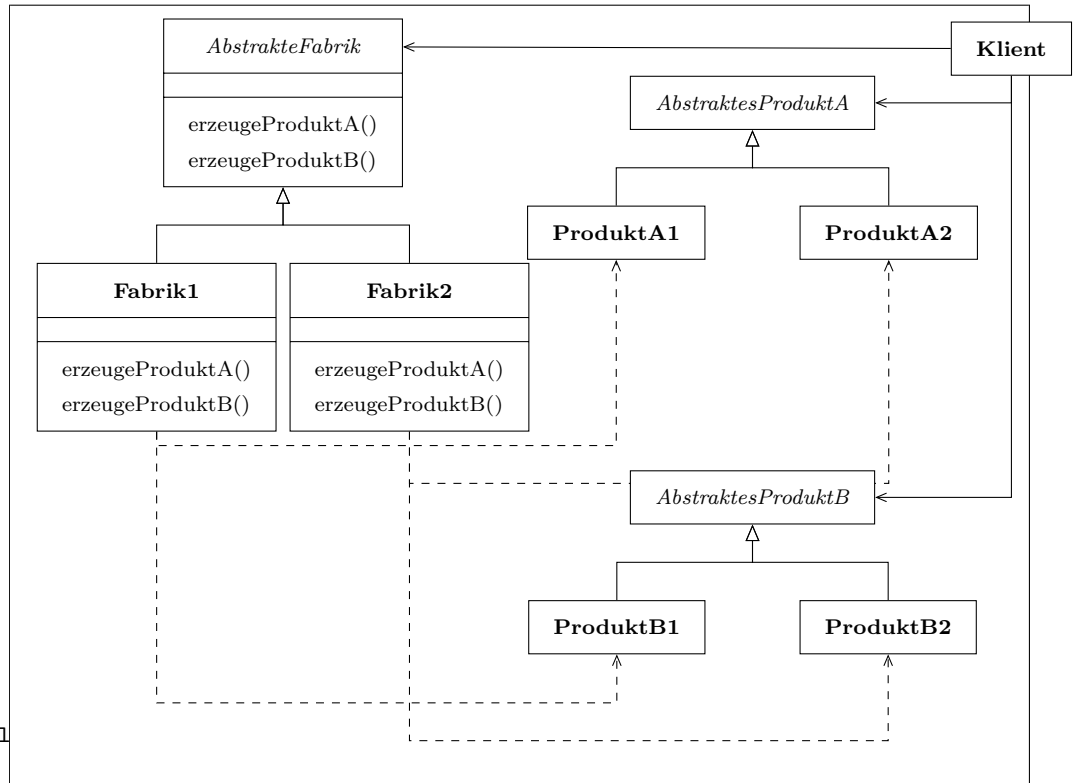
2.10 entwurfsmuster.sty

```

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()\
617       erzeugeProduktB()\
618     }
619     \umlclass[below left=1cm and -1.5cm of AbstrakteFabrik]{Fabrik1}{-}{
620       erzeugeProduktA()\
621       erzeugeProduktB()\
622     }
623     \umlclass[below right=1cm and -1.5cm of AbstrakteFabrik]{Fabrik2}{-}{
624       erzeugeProduktA()\
625       erzeugeProduktB()\
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}

```

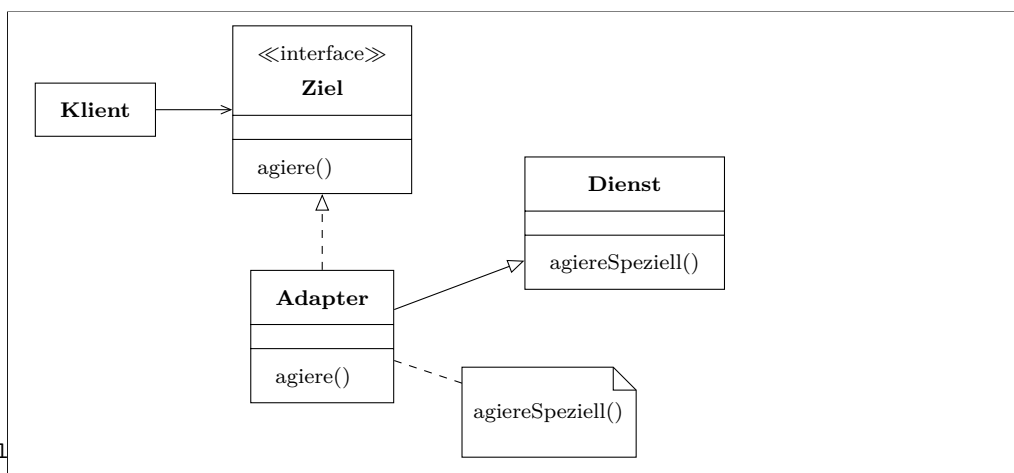
```

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)]

```

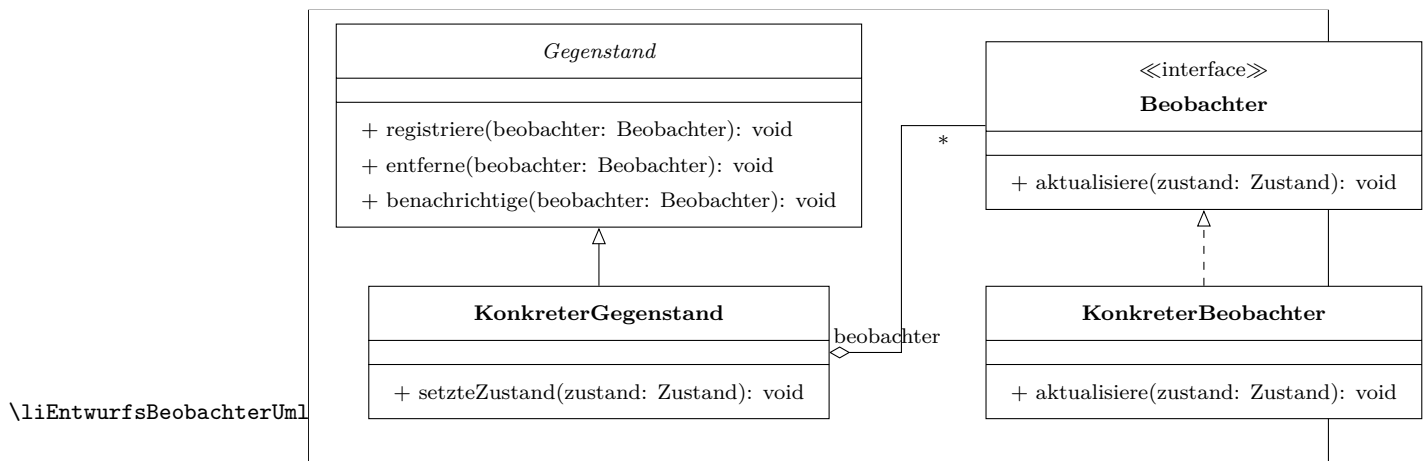


```

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)

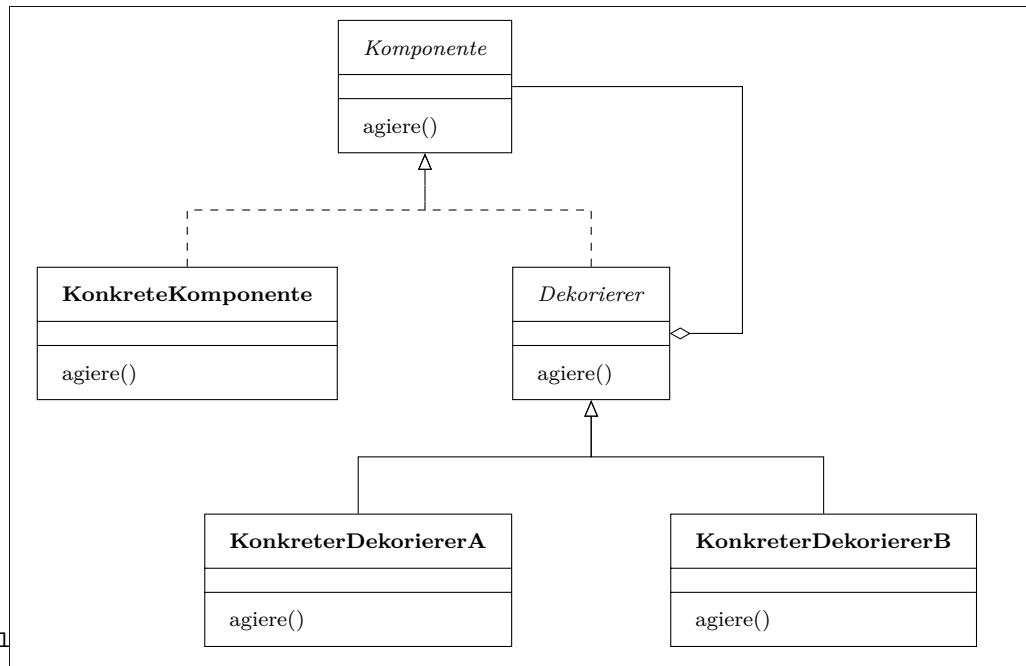
Ein konkretes Subjekt (konkretes, beobachtbares Objekt) speichert den relevanten Zustand und benachrichtigt alle Beobachter bei Zustandsänderungen über deren Aktualisierungsschnittstelle. Es verfügt über eine Schnittstelle zur Erfragung des aktuellen Zustands.

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.

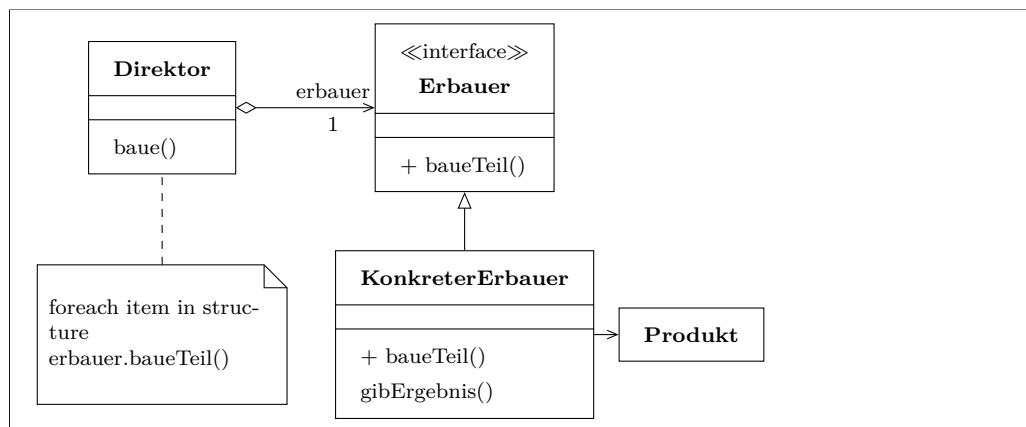
```

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.

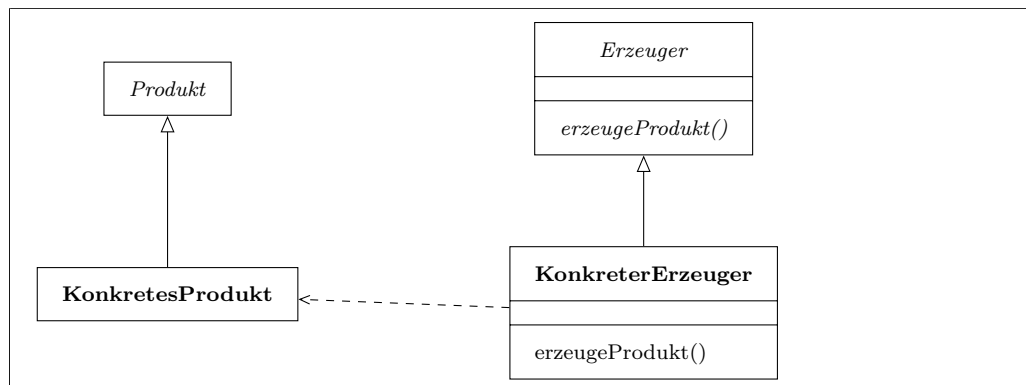
Produkt Das Produkt repräsentiert das zu konstruierende komplexe Objekt.

```

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{erzeugeProdukt()}\}
  
```

```

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 }

```

EntwurfsFabrikmethodeAkteure Quelle: deutsche Wikipedia

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

KonkretesProdukt KonkretesProdukt implementiert die Produkt-Schnittstelle.

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

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

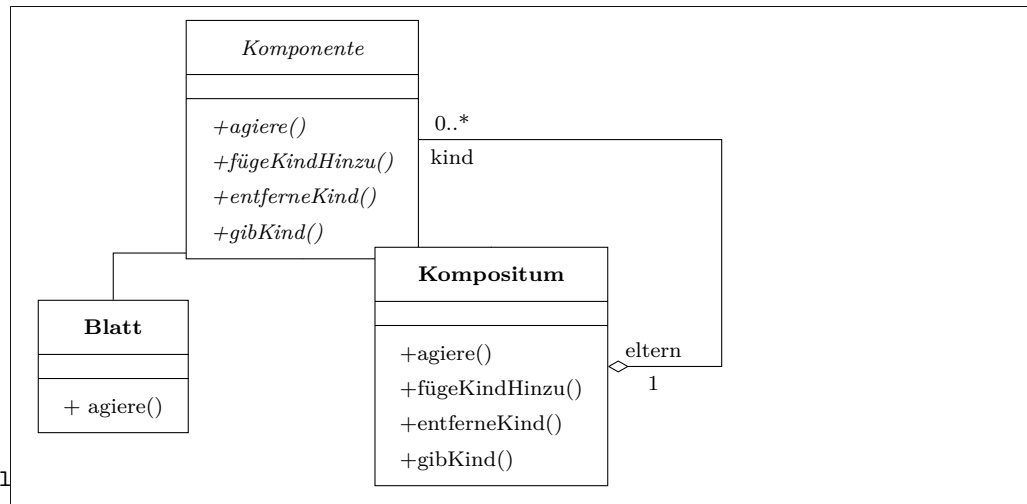
```

854 \def\liEntwurfsFabrikmethodeAkteure{
855   \begin{description}
856     \item[Produkt]
857
858     Das Produkt ist der Basistyp (Klasse oder Schnittstelle) für das
859     zu erzeugende Produkt.
860
861     \item[KonkretesProdukt]
862
863     KonkretesProdukt implementiert die Produkt-Schnittstelle.
864
865     \item[Erzeuger]
866
867     Der Erzeuger deklariert die Fabrikmethode, um ein solches Produkt
868     zu erzeugen und kann eine Default-Implementierung beinhalten.
869
870     \item[KonkreterErzeuger]
871
872     KonkreterErzeuger überschreibt die Fabrikmethode, um die ihm
873     entsprechenden konkreten Produkte zu erzeugen (z. B. indem er den
874     Konstruktor einer konkreten Produkt-Klasse aufruft).
875
876     \footcite{wiki:fabrikmethode}
877   \end{description}
878 }
879

```

2.10.8 Kompositum

\liEntwurfsKompositumUml



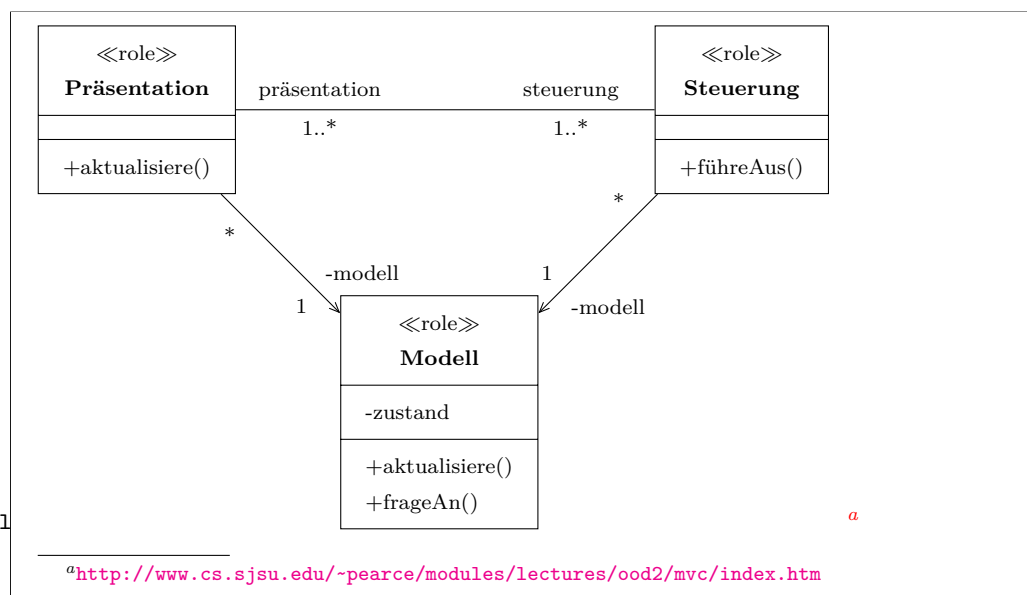
```

880 \def\liEntwurfsKompositumUml{
881   \begin{tikzpicture}
882     \umlclass[x=2.5,y=3,type=abstract]{Komponente}{-}{
883       \textit{+agiere()}\
884       \textit{+fügeKindHinzu()}\
885       \textit{+entferneKind()}\
886       \textit{+gibKind()}
887     }
888     \umlclass[x=0]{Blatt}{-}{+ agiere()}
889     \umlclass[x=5]{Kompositum}{-}{
890       +agiere()\
891       +fügeKindHinzu()\
892       +entferneKind()\
893       +gibKind()
894     }
895
896     \umlVHVinherit{Kompositum}{Komponente}
897     \umlVHVinherit{Blatt}{Komponente}
898     \umlHVVHaggreg[anchor1=east,arm1=1.5cm,arg1=eltern,mult1=1,arg2=kind,mult2=0..*,pos2=2.9,
899     \end{tikzpicture}
900 }

```

2.10.9 Modell-Präsentation-Steuerung

ModellPraesentationSteuerungUml



^a<http://www.cs.sjsu.edu/~pearce/modules/lectures/ood2/mvc/index.htm>

```

901 \def\liEntwurfsModellPraesentationSteuerungUml{
902   \begin{tikzpicture}

```

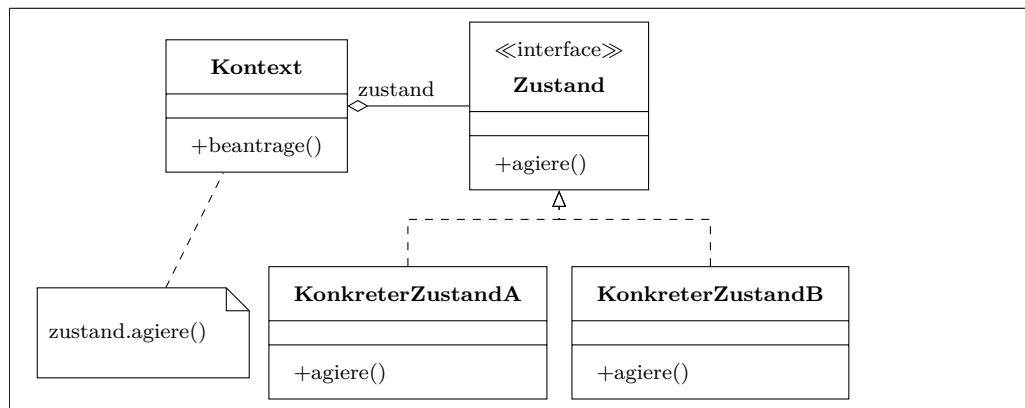
```

903 \umlclass[x=-4cm,type=role]{Präsentation}{+aktualisiere()}
904 \umlclass[x=4cm,type=role]{Steuerung}{+führeAus()}
905 \umlclass[y=-4cm,type=role]{Modell}{
906   -zustand
907 }{
908   +aktualisiere()\
909   +frageAn()
910 }
911
912 \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Präsentation}{Modell}
913 \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Steuerung}{Modell}
914 \umlassoc[arg1=präsentation,mult1=1..*,arg2=steuerung,mult2=1..*]{Präsentation}{Steuerung}
915 \end{tikzpicture}
916 \liFussnoteUrl{http://www.cs.sjsu.edu/~pearce/modules/lectures/ood2/mvc/index.htm}
917 }

```

2.10.10 Zustand

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



```

918 \def\liEntwurfsZustandUml{
919   \begin{tikzpicture}
920     \umlclass[x=-1,y=3]{Kontext}{+beantrage()}
921     \umlclass[x=3,y=3,type=interface]{Zustand}{+agiere()}
922     \umlclass[x=1,y=0]{KonkreterZustandA}{+agiere()}
923     \umlclass[x=5,y=0]{KonkreterZustandB}{+agiere()}
924
925     \umlVHVreal{KonkreterZustandA}{Zustand}
926     \umlVHVreal{KonkreterZustandB}{Zustand}
927
928     \umlaggreg[arg=zustand,pos=0.4]{Kontext}{Zustand}
929
930     \umlnote[x=-2.5,y=0,width=2.5cm]{Kontext}{zustand.agiere()}
931   \end{tikzpicture}
932 }

```

\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.

```

933 \def\liEntwurfsZustandAkteure{
934   \begin{description}
935     \item[Kontext (Context)]

```



```

936
937     definiert die clientseitige Schnittstelle und verwaltet die separaten
938     Zustandsklassen.
939
940     \item[State (Zustand)]
941
942     definiert eine einheitliche Schnittstelle aller Zustandsobjekte und
943     implementiert gegebenenfalls ein Standardverhalten.
944
945     \item[KontreterZustand (ConcreteState)]
946
947     implementiert das Verhalten, das mit dem Zustand des Kontextobjektes
948     verbunden ist.
949 \end{description}
950 }
951

```

2.11 er.sty

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

955 \RequirePackage{tikz-er2}
956 \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}

957 \RequirePackage{soul}
958 \RequirePackage{fontawesome}

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

959 \ExplSyntaxOn

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

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

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

\liErMpEntity mp = marginpar
Let-Abkürzung: \let\ e=\liErMpEntity
963 \def\liErMpEntity#1{
964   \liErEntity{#1}
965   \marginpar{
966     \liErEntity{\tiny\faSquareO{}}~E:~#1}
967   }
968 }
□

\liErMpRelationship Let-Abkürzung: \let\ r=\liErMpRelationship
969 \def\liErMpRelationship#1{
970   \liErRelationship{#1}
971   \marginpar{
972     \liErRelationship{\tiny\faGg{}}~R:~#1}
973   }
974 }

\liErMpAttribute Let-Abkürzung: \let\ a=\liErMpAttribute
975 \def\liErMpAttribute#1{
976   \liErAttribute{#1}
977   \marginpar{
978     \liErAttribute{\tiny\faCircleThin{}}~A:~#1}
979   }
980 }

\liErDatenbankName Let-Abkürzung: \let\ d=\liErDatenbankName
datenbank name
981 \def\liErDatenbankName#1{
982   {
983     \footnotesize\texttt{(#1)}
984   }
985 }

986 \ExplSyntaxOff
987

```

2.12 formale-sprachen.sty

```

988 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
989 \ProvidesPackage{lehramt-informatik-formale-sprachen}[2021/02/21 Hilfsmakros
990 zum Setzen von mathematischen Formeln bei dem Thema Formale Sprachen]
991
992 \directlua{
993   formale_sprachen = require('lehramt-informatik-formale-sprachen')
994 }
995
996 \RequirePackage{hyperref}
997 \RequirePackage{lehramt-informatik-mathe}

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

998 \def\liMengeOhneMathe#1{\{ #1 \}}
999 \def\liMenge#1{%
1000 \ifmmode%
1001 \liMengeOhneMathe{#1}%
1002 \else%
1003 $\liMengeOhneMathe{#1}$%
1004 \fi%
1005 }

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

1006 \def\liEpsilon{\varepsilon}

\liPotenzmenge Umgeben mit geschweiften Klammern in einer Mathematik-Umgebung
1007 \def\erzeuge@tiefgestellt#1{\directlua{formale_sprachen.erzeuge_tiefgestellt('#1')}}
1008 \def\liPotenzmengeOhneMathe#1{\{ \erzeuge@tiefgestellt{#1} \}}
1009 \def\liPotenzmenge#1{\liPotenzmengeOhneMathe{#1}}

\liZustandsmenge \liZustandsmenge{z1, z2}:  $\{z_1, z_2\}$ 
1010 \let\liZustandsmengeOhneMathe=\liPotenzmengeOhneMathe
1011 \let\liZustandsmenge=\liPotenzmenge

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

1012 \def\liUeberfuehrungsFunktionOhneMathe#1{\delta(\erzeuge@tiefgestellt{#1})}
1013 \def\liUeberfuehrungsFunktion#1{
1014 \ifmmode
1015 \liUeberfuehrungsFunktionOhneMathe{#1}
1016 \else
1017 $\liUeberfuehrungsFunktionOhneMathe{#1}$
1018 \fi
1019 }

\liAlphabet \liAlphabet{a, b} ( $\Sigma = \{a, b\}$ ):  $\Sigma = \{a, b\}$ 
1020 \def\liAlphabet#1{\Sigma = \{ #1 \}}

\liBandAlphabet \liBandAlphabet{\liTuringLeerzeichen}:  $\Gamma = \Sigma \cup \{\square\}$ 
1021 \def\liBandAlphabet#1{\Gamma = \Sigma \cup \{ #1 \}}

\liZustandsBuchstabe
1022 \def\liZustandsBuchstabe{z}

\liZustandsBuchstabeGross
1023 \def\liZustandsBuchstabeGross{Z}

```

```

\liZustandsmengeNr
1024 \def\zustandsnamens@liste#1#2{
1025   $
1026   \{
1027     \directlua{formale_sprachen.erzeuge_zustandsnamens_liste('#1', '#2')}
1028   \}
1029   $
1030 }
1031 \def\liZustandsmengeNr#1{\zustandsnamens@liste{\liZustandsBuchstabe}{#1}}

\liZustandsmengeNrGross
1032 \def\liZustandsmengeNrGross#1{\zustandsnamens@liste{\liZustandsBuchstabeGross}{#1}}

\liZustandsname \liZustandsname{1}: $z_1$
1033 \def\liZustandsname#1{\liZustandsBuchstabe_#1$}

\liZustandsnameGross \liZustandsnameGross{1}: $Z_1$
1034 \def\liZustandsnameGross#1{\liZustandsBuchstabeGross_#1$}

\liAbleitung \liAbleitung{S -> aB -> ab}:  $S \vdash aB \vdash ab$ 
1035 \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}

1036 \NewDocumentEnvironment { liProduktionsRegeln }
1037 { 0{P} +b }
1038 {
1039   \noindent
1040   $#1 = \{ $
1041   \vspace{-0.2cm}
1042   \begin{align*}
1043     \directlua{formale_sprachen.produktions_regeln('#2')}
1044   \end{align*}
1045   \vspace{-1.5cm}
1046   \begin{flushright}\}\end{flushright}
1047 } {}

\liProduktionen \liProduktionen{S -> A, A -> a}:  $\{S \rightarrow A, A \rightarrow a\}$ 
1048 \def\liProduktionen#1{
1049   \liMenge{\directlua{formale_sprachen.produktions_regeln('#1', true)}}
1050 }

\liZustandsnameTiefgestellt Automatisch tiefgestellte Nummerierung \z1
Let-Abkürzung: \let\z=\liZustandsnameTiefgestellt
1051 \def\liZustandsnameTiefgestellt#1{
1052   \ifmmode
1053     \liZustandsBuchstabe\sb{#1}
1054   \else
1055     $\liZustandsBuchstabe\sb{#1}$
1056   \fi
1057 }

1058 \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}

1059 \NewDocumentCommand{ \liAusdruck } { O{L} m m } {
1060   $
1061   \tl_if_empty:nTF {#1} {} {#1 =}
1062   \{
1063     \, #2 \,
1064     |
1065     \, #3 \,
1066   \}$
1067 }
1068 \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

1069 \def\liFlaci#1{%
1070   \par
1071   {%
1072     \scriptsize
1073     Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte
1074     Automaten, Compiler und Interpreter) Ein Projekt der Hochschule
1075     Zittau/Görlitz und der Pädagogischen Hochschule Schwyz:
1076     \href{https://flaci.com/#1}{flaci.com/#1}%
1077   }%
1078   \par
1079 }

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

1080 \ExplSyntaxOn
1081 \NewDocumentCommand {\liGrammatik} { O{G} m } {
1082   \tl_set:Nn \l_variablen_tl {V}
1083   \tl_set:Nn \l_alphabet_tl {\Sigma}
1084   \tl_set:Nn \l_produktionen_tl {P}
1085   \tl_set:Nn \l_start_tl {S}
1086
1087   \keys_define:nn { grammatik } {
1088     variablen .code:n = {\tl_set:Nn \l_variablen_tl {\liMenge{##1}}},
1089     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
1090     produktionen .code:n = {\tl_set:Nn \l_produktionen_tl {\liProduktionen{##1}}},
1091     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
1092   }
1093
1094   \keys_set:nn { grammatik } { #2 }
1095
1096   $#1 = (
1097     \l_variablen_tl,
1098     \l_alphabet_tl,
1099     \l_produktionen_tl,
1100     \l_start_tl
1101   )$

```

```
1102 }  
1103 \ExplSyntaxOff  
  
1104
```

2.13 formatierung.sty

```
1105 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1106 \ProvidesPackage{lehramt-informatik-formatierung}[2020/11/27]
```

2.13.1 Schriftarten / Typographie

```
1107 \RequirePackage{mathpazo}
1108 \RequirePackage[no-math]{fontspec}
1109 \setmainfont{texgyrepagella}
1110 \setsansfont{QTAncientOlive}
1111 \RequirePackage{sectsty}
1112 \allsectionsfont{\fontspec{QTAncientOlive}}
```

2.13.2 Farben

```
1113 \RequirePackage{xcolor}
1114 \definecolor{infogray}{rgb}{0.97,0.97,0.97}
```

2.13.3 Überschriften

```
1115 \RequirePackage{titlesec}
1116 \titleformat{\chapter}[display]{\sffamily\bfseries}{\Opt}{\LARGE}
1117 \titlespacing{\chapter}{0pt}{0pt}{*1}
1118 \titleformat{\paragraph}[hang]{\sffamily\normalsize\bfseries}{\theparagraph}{1em}{}
1119 \setcounter{secnumdepth}{0}
```

2.13.4 Listen

```
1120 \RequirePackage{paralist}
1121 \renewcommand\labelitemi{-}
1122 \renewcommand\labelitemii{-}
1123 \renewcommand\labelitemiii{-}
1124 \renewcommand\labelitemiv{-}
1125 % Counter: enumi enumii enumiii enumiv
1126 % Styles: \arabic{counter} \alph{counter} \Alph{counter} \roman{counter} \Roman{counter}
1127 \renewcommand{\labelenumi}{(\alph{enumi})}
1128 \renewcommand{\labelenumii}{(\roman{enumii})}
```

2.13.5 Kasten

```
1129 \RequirePackage{mdframed}
1130 \mdfsetup{backgroundcolor=infogray}
```

liKasten

```
1131 \NewDocumentEnvironment { liKasten }{ } {
1132   \begin{mdframed}[backgroundcolor=white!0]
1133 } {
1134   \end{mdframed}
1135 }
```

2.13.6 Header

```
1136 \RequirePackage{fancyhdr}
1137 \fancyhead[L,C,R]{}
1138 \fancyfoot[L]{}
1139 \fancyfoot[C]{}
1140 \fancyfoot[R]{\thepage}
1141 \pagestyle{fancy}
1142 \renewcommand{\headrulewidth}{0pt}
1143 \renewcommand{\footrulewidth}{0pt}
1144
```


2.14 gantt.sty

```

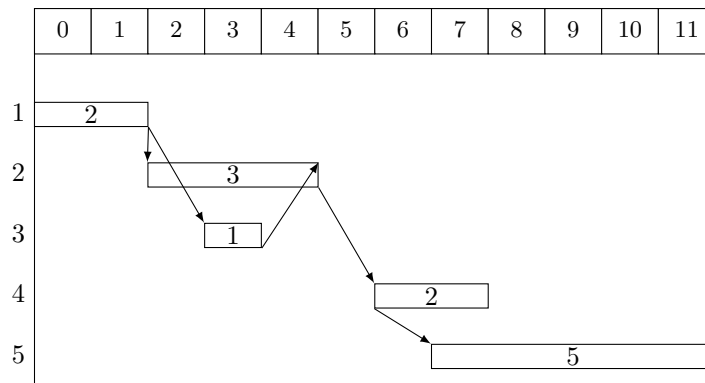
1145 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1146 \ProvidesPackage{lehramt-informatik-gantt}[2020/09/05]

\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}

```



```

1147 \RequirePackage{tikz-uml}
1148 \RequirePackage{pgfgantt}
1149 \setganttlinklabel{f-s}{}
1150 \setganttlinklabel{s-s}{}
1151 \setganttlinklabel{f-f}{}
1152 \setganttlinklabel{s-f}{}
1153

```

2.15 grafik.sty

```
1154 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1155 \ProvidesPackage{lehramt-informatik-grafik}[2020/12/27 Lädt das Paket
1156 TikZ, um Grafiken zeichnen zu können.]
1157 \RequirePackage{tikz}
1158
```

2.16 graph.sty

```

1159 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1160 \ProvidesPackage{lehramt-informatik-graph}[2020/06/09]
1161 \RequirePackage{tikz}

```

Für die die Adjazenzliste (`xrightrightarrow`)

```

1162 \RequirePackage{amsmath}

```

Für Adjazenz-Matrix

```

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

```

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

```

1163 \RequirePackage{blkarray}
1164 \usetikzlibrary{arrows.meta}

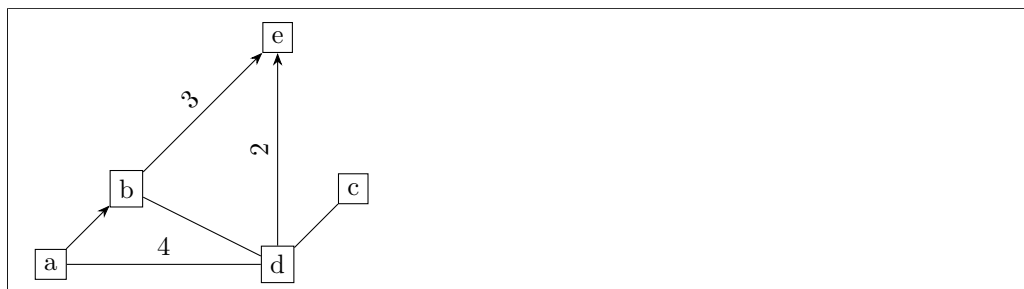
```

```

\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}

```



```

1165 \tikzset{
1166   li graph/.style={
1167     every node/.style={
1168       rectangle,
1169       draw,
1170     },
1171     every edge/.style={
1172       >={Stealth[black]},
1173       draw,
1174     },
1175     every edge/.append style={
1176       every node/.style={
1177         sloped,
1178         auto,
1179       }
1180     }
1181   },
1182   li markierung/.style={
1183     ultra thick,
1184   }
1185 }

```

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}

```

```

1186 \NewDocumentEnvironment { liGraphenFormat }{ +b } {} {}

```

```

1187

```

2.17 hanoi.sty

1188 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 1189 \ProvidesPackage{lehramt-informatik-hanoi}[2020/12/19 Paket zum Setzen
 1190 von Hanoi-Grafiken]

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

```
1191 \RequirePackage{tikz}
1192 \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}

1193 \def\li@mset #1[#2]=#3{%
1194 \expandafter\xdef\csname #1#2\endcsname{#3}
1195 }
1196 \def\li@mget #1[#2]{%
1197 \csname #1#2\endcsname
1198 }
1199 \def\li@minc #1[#2] += #3{%
1200 \pgfmathparse{\li@mget #1[#2] + #3}%
1201 \li@mset #1[#2] = \pgfmathresult
1202 }
1203
1204 \def\liHanoi#1#2{
1205   \edef\li@numdiscs{#1}
1206   \def\li@sequence{#2}
1207   \begin{tikzpicture}[line width=4mm,brown!40,line cap=round,xscale=3]
1208     % init colors
1209     \foreach[count=\j] \c in {red,green!80!black,blue,orange,violet,gray,yellow!80!black,purple!80!black}{
1210       \li@mset col[\j]={\c};
1211     }
1212     % draw poles and init pole counters
1213     \foreach \j in {1,2,3}{
1214       \li@mset pos[\j]=0
1215       \draw (\j,-.5) -- +(0, .5 + .5*\li@numdiscs);
1216     }
1217     % draw base
1218     \draw (.5,-.5) -- +(3,0);
1219     % draw discs
1220     \foreach[count=\k] \i/\j in \li@sequence{
1221       \draw[draw={\li@mget col[\i]}] (\j,\li@mget pos[\j]) +(-.4*\i/\li@numdiscs,0) -- +(.4*\i/\li@numdiscs,0);
1222       \li@minc pos[\j] += {.5}
1223     }
1224   \end{tikzpicture}
1225 }
```

2.18 komplexitaetstheorie.sty

```

1226 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1227 \ProvidesPackage{lehramt-informatik-komplexitaetstheorie}[2021/07/08 Zum
1228 Setzen von Karp's NP-vollständigen Problemen, Hilfsmakros für die
1229 Polynomialzeitreduktion.]

```

Let-Abkürzungen

```

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

```

```

1230 \liLadePakete{mathe}

```

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

```

1231 \RequirePackage{mdframed}

```

```

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

```

```

1232 \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

```

```

1233 \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`

```

1234 \def\liProblemBeschreibung#1#2#3{
1235   \begin{mdframed}[
1236     userdefinedwidth=9cm,
1237     align=center,
1238     backgroundcolor=white!0,
1239   ]
1240     \centerline{\large\liProblemName{#1}}
1241
1242     \medskip
1243
1244     \begin{description}
1245       \item[Gegeben:] #2
1246       \item[Frage:] #3
1247     \end{description}
1248   \end{mdframed}
1249 }

```

```

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

1250 \NewDocumentCommand{ \liPolynomiellReduzierbar } { m O{p} m } {
1251 \begin{displaymath}
1252 \liProblemName{#1}
1253 \preceq_{#2}
1254 \liProblemName{#3}
1255 \end{displaymath}
1256 }

\liProblemVertexCover

1257 \def\liProblemClique{%
1258 Das \textbf{Cliquesproblem} fragt nach der Existenz einer Clique der
1259 Mindestgröße  $n$  in einem gegebenen Graphen.
1260 \footcite[Seite 76]{theo:fs:4}
1261 Eine Clique ist eine Teilmenge von Knoten in einem ungerichteten
1262 Graphen, bei der \emph{jedes Knotenpaar durch eine Kante} verbunden ist.
1263 \footcite{wiki:cliquenproblem}
1264 }

\liProblemVertexCover

1265 \def\liProblemVertexCover{%
1266 %
1267 Das \textbf{Knotenüberdeckungsproblem} (\liProblemName{Vertex Cover})
1268 fragt, ob zu einem gegebenen einfachen Graphen und einer natürlichen
1269 Zahl  $k$  eine Knotenüberdeckung der Größe von höchstens  $k$  existiert.
1270 \footcite{wiki:knotenueberdeckung}
1271
1272 Das heißt, ob es eine aus maximal  $k$  Knoten bestehende Teilmenge  $U$ 
1273 der Knotenmenge gibt, so dass jede Kante des Graphen mit
1274 mindestens einem Knoten aus  $U$  verbunden ist.
1275 \footcite[Seite 78]{theo:fs:4}%
1276 }

\liProblemSubsetSum Kein Karp-21-Problem

1277 \def\liProblemSubsetSum{%
1278 Das \textbf{Teilsommenproblem} (\liProblemName{Subset Sum} oder
1279 \liProblemName{SSP}) ist ein spezielles Rucksackproblem.
1280 \footcite{wiki:teilsommenproblem}
1281 Gegeben sei eine Menge von ganzen Zahlen  $I = \{w_1, w_2, \dots, w_n\}$ .
1282 Gesucht ist eine Untermenge, deren Elementsumme maximal, aber nicht
1283 größer als eine gegebene obere Schranke  $c$  ist.
1284 \footcite[Seite 74]{theo:fs:4}
1285 }

\liProblemSubsetSum Kein Karp-21-Problem

1286 \def\liProblemSat{%
1287 Das \textbf{Erfüllbarkeitsproblem der Aussagenlogik} \liProblemName{Sat}
1288 und \liProblemName{k-SAT} mit  $k \geq 3$ ,  $k \in \mathbb{N}$  (Satz von
1289 Cook) fragt, ob eine aussagenlogische Formel erfüllbar
1290 ist.\footcite{wiki:sat} Das Erfüllbarkeitsproblem der
1291 \emph{Aussagenlogik} ist in exponentieller Zeit in Abhängigkeit der
1292 Anzahl der Variablen mit Hilfe einer Wahrheitstabelle entscheidbar.
1293 Diese \emph{Wahrheitstabelle} kann nicht in polynomieller Zeit
1294 aufgestellt werden.
1295 \footcite[Seite 71]{theo:fs:4}
1296 }

1297

```

2.19 kontrollflussgraph.sty

```
1298 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1299 \ProvidesPackage{lehramt-informatik-kontrollflussgraph}[2020/11/07]

1300 \RequirePackage{tikz}
1301 \usetikzlibrary{positioning}
1302 \tikzset{
1303   li kontrollfluss/.style={
1304     knoten/.style={
1305       circle,
1306       draw
1307     },
1308     usebox/.style={
1309       draw,
1310       rectangle,
1311       font=\scriptsize,
1312       anchor=west,
1313       align=left,
1314     },
1315     bedingung/.style={
1316       midway,
1317       draw=none,
1318       font=\scriptsize
1319     },
1320     knotenbeschriftung/.style={
1321       draw,
1322       rectangle,
1323       midway,
1324       font=\scriptsize
1325     },
1326     wahr/.style={
1327       thick
1328     },
1329     falsch/.style={
1330       dashed
1331     },
1332     every node/.style={
1333       circle,
1334       draw,
1335     },
1336     every edge/.append style={
1337       every node/.style={
1338         draw=none,
1339         bedingung,
1340       }
1341     },
1342     every path/.style={
1343       draw,
1344       ->,
1345     },
1346     every pin/.style={
1347       draw,
1348       dotted,
1349       rectangle,
1350       pin position=right
1351     },
1352     every pin edge/.style={
1353       dotted,
1354       arrows=-,
1355     }
1356   }
1357 }
```

liKontrollflussgraph


```

1358 \NewDocumentEnvironment { liKontrollflussgraph } { 0{ } } {
1359   \begin{tikzpicture}[
1360     li kontrollfluss,
1361     #1
1362   ]
1363 } {
1364   \end{tikzpicture}
1365 }

\liAnweisung
1366 \def\liAnweisung#1(#2,#3){\node[knoten] at (#2,#3) (#1) {#1};}

\liBedingung Let-Abkürzung: \let\b=\liBedingung
1367 \def\liBedingung#1#2{node[bedingung,#1]{\texttt{#2}}}}

\liBedingungWahr Let-Abkürzung: \let\bWahr=\liBedingungWahr
1368 \def\liBedingungWahr#1{node[bedingung,#1]{\texttt{[true]}}}}

\liBedingungFalsch Let-Abkürzung: \let\bFalsch=\liBedingungFalsch
1369 \def\liBedingungFalsch#1{node[bedingung,#1]{\texttt{[false]}}}}

\liKontrollCode Let-Abkürzung: \let\c=\liKontrollCode
1370 \def\liKontrollCode#1{{\tiny\texttt{\textcolor{gray}{#1}}}}

\liKontrollTextzeileKnoten Let-Abkürzung: \let\k=\liKontrollTextzeileKnoten
1371 \def\liKontrollTextzeileKnoten#1{\raisebox{-2pt}{\tikz[scale=0.5,transform shape] \node[draw

\liKontrollKnotenPfad Let-Abkürzung: \let\p=\liKontrollKnotenPfad
1372 \ExplSyntaxOn
1373 \NewDocumentCommand { \liKontrollKnotenPfad }{ m }
1374 {
1375   \seq_set_split:Nnn \l_tmpa_seq { - } { #1 }
1376   \seq_set_map:Nnn \l_tmpa_seq \l_tmpa_seq {\liKontrollTextzeileKnoten{##1}}
1377   \seq_use:Nn \l_tmpa_seq { ~~~~ }
1378 }
1379 \ExplSyntaxOff

1380

```

2.20 kopf-fusszeilen.sty

```
1381 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1382 \ProvidesPackage{lehramt-informatik-kopf-fusszeilen}[2021/08/20 Kopf-
1383 und Fußzeilen mit Hilfe des Pakets fancyhdr.]

1384 \ExplSyntaxOn

1385 \fancyhead{}
1386 \fancyhead[R0,LE]{\scriptsize\LehramtInformatikTitel}
1387 \fancyhead[L0,RE]{\scriptsize\today}
1388 \fancyfoot{}
1389 \fancyfoot[LE,R0]{\thepage}
1390 \fancyfoot[L0,CE]{\scriptsize\LehramtInformatikAutorName}
1391 \fancyfoot[CO,RE]{\scriptsize\LehramtInformatikAutorEmail}
1392 \renewcommand{\headrulewidth}{0.4pt}
1393 \renewcommand{\footrulewidth}{0.4pt}

1394 \ExplSyntaxOff

1395
```

2.21 literatur-dummy.sty

```
1396 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1397 \ProvidesPackage{lehramt-informatik-literatur-dummy}[2020/11/27]

\literatur

1398 \def\literatur{}

\footcite

1399 % \RequirePackage[stable,multiple]{footmisc}
1400 \NewDocumentCommand{ \footcite } { o m }{}

1401
```

2.22 literatur.sty

```
1402 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1403 \ProvidesPackage{lehramt-informatik-literatur}[2020/11/27]

1404 \RequirePackage{csquotes}
1405 \RequirePackage[
1406   bibencoding=utf8,
1407   citestyle=authortitle,
1408   backend=biber,
1409 ]{biblatex}
1410 \addbibresource{\LehramtInformatikRepository/Quellen/10_DB.bib}
1411 \addbibresource{\LehramtInformatikRepository/Quellen/20_OOMUP.bib}
1412 \addbibresource{\LehramtInformatikRepository/Quellen/30_AUD.bib}
1413 \addbibresource{\LehramtInformatikRepository/Quellen/40_SOSY.bib}
1414 \addbibresource{\LehramtInformatikRepository/Quellen/60_FUMUP.bib}
1415 \addbibresource{\LehramtInformatikRepository/Quellen/70_THEO.bib}
1416 \addbibresource{\LehramtInformatikRepository/Quellen/80_DDI.bib}
1417 \addbibresource{\LehramtInformatikRepository/Quellen/Allgemein.bib}
1418 \addbibresource{\LehramtInformatikRepository/Quellen/Examen.bib}
1419 \addbibresource{\LehramtInformatikRepository/Quellen/Wikipedia.bib}
1420 % To allow footnotes in the heading
1421 \RequirePackage[stable,multiple]{footmisc}

\literatur

1422 \def\literatur{\printbibliography[heading=subbibliography]}

1423
```

2.23 makros.sty

```

1424 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1425 \ProvidesPackage{lehramt-informatik-makros}[2020/11/27 Eine Sammlung von
1426 häufig verwendeten Makros und Umgebungen, die thematisch zu keinem
1427 anderen Paket passen]

1428 \RequirePackage{hyperref}
1429 \RequirePackage{graphicx}

    Für die Umgebung liQuellen benötigt.
1430 \RequirePackage{paralist}

\inhaltsverzeichnis
1431 \def\inhaltsverzeichnis {
1432   \begin{mdframed}
1433     \begin{group}
1434       \let\clearpage\relax
1435       \tableofcontents
1436     \end{group}
1437   \end{mdframed}
1438 }

\memph \mephm (\marginpar and \emph)
1439 \newcommand{\memph}[1]{\emph{#1}\marginpar{\tiny#1}}

\SLASH
1440 \newcommand\SLASH{\char`\\}

\liPseudoUeberschrift Text, der sich wie eine Überschrift verhält.
1441 \newcommand{\liPseudoUeberschrift}[1]{
1442   \bigskip
1443   \noindent
1444   \textsf{\textbf{#1}}
1445   \noindent
1446 }

\liBeschriftung Ähnlich dem Makro \liPseudoUeberschrift{}. Am Ende des Textes wird ein Doppelpunktzeichen angehängt.
1447 \newcommand{\liBeschriftung}[1]{
1448   \par
1449   \noindent
1450   \medskip
1451   \textbf{#1}:
1452   \medskip
1453   \noindent
1454 }

\hinweis
1455 \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.
1456 \NewDocumentEnvironment { liProjektSprache }{ o +b } {} {}

liEinbettung
1457 \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.
1458 \RequirePackage{xparse}
1459 \ExplSyntaxOn

```

```

1460 \NewDocumentEnvironment{ liAntwort } { 0{standard} }
1461 {
1462   \str_case:nn {#1} {
1463     {standard} {
1464       \def\beschriftung{}
1465       \mdfsetup{backgroundcolor=gray!5,linecolor=gray}
1466     }
1467     {richtig} {
1468       \def\beschriftung{richtig}
1469       \mdfsetup{backgroundcolor=green!5,linecolor=green}
1470     }
1471     {falsch} {
1472       \def\beschriftung{falsch}
1473       \mdfsetup{backgroundcolor=red!3,linecolor=red}
1474     }
1475     {muster} {
1476       \def\beschriftung{Musterlösung}
1477       \mdfsetup{backgroundcolor=green!12,linecolor=green}
1478     }
1479   }
1480   \ifx\beschriftung\empty\else
1481     \noindent
1482     \textbf{\beschriftung{:}}
1483     \fi
1484     \begin{mdframed}
1485   }
1486 {\end{mdframed}}

```

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

```

1487 \NewDocumentEnvironment{ liAdditum } { o }
1488 {
1489   \begin{mdframed}[backgroundcolor=yellow!5]
1490     \IfNoValueTF {#1}
1491     { \liPseudoUeberschrift{Additum} }
1492     { \liPseudoUeberschrift{Additum:~#1} }
1493   }
1494 {\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.

```

1495 \NewDocumentEnvironment{ liExkurs }{o +b}{
1496   \vspace{0.2cm}%
1497   \begin{mdframed}[
1498     backgroundcolor=white,
1499     bottomline=false,
1500     innermargin=1cm,
1501     leftline=true,
1502     linecolor=black,
1503     linewidth=0.1cm,
1504     outermargin=1cm,
1505     rightline=false,
1506     topline=false,
1507   ]

```

```

1508 \footnotesize
1509 \noindent%
1510 \textbf{Exkurs:~#1}\par%
1511 \noindent%
1512 #2
1513 \end{mdframed}
1514 \vspace{0.2cm}
1515 }{}

```

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

```

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

```

Weiterführende Literatur:

- Quelle 1
- Quelle 2

```

1516 \cs_new:Npn \listen@punkt #1 {\item #1}
1517 \NewDocumentEnvironment { liQuellen }{ +b }
1518 {
1519   \seq_clear_new:N \l_quellen
1520   \seq_set_split:Nnn \l_quellen {\item} {#1}
1521   \seq_remove_all:Nn \l_quellen {}
1522   \begin{mdframed}[backgroundcolor=white!0]
1523     \footnotesize
1524     \noindent
1525     \textsf{\textbf{Weiterführende~Literatur:}}
1526     \medskip
1527     \begin{compactitem}
1528       \seq_map_function:NN \l_quellen {\listen@punkt}
1529     \end{compactitem}
1530   \end{mdframed}
1531   %
1532   \makeatletter
1533   \par\@afterindentfalse\@afterheading
1534   \makeatother
1535 } {}

```

liLernkartei

```

1536 \NewDocumentEnvironment { liLernkartei }{ m +b }
1537 {
1538   \begin{mdframed}
1539     \footnotesize
1540     \noindent%
1541     \textbf{Lernkarteikarte:~#1}\par%
1542     \noindent%
1543     #2
1544   \end{mdframed}
1545 } {}

```

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

```

1546 \NewDocumentEnvironment { liDiagramm }{ m +b }
1547 {
1548   \begin{mdframed}[backgroundcolor=white!0]
1549     \small
1550     \noindent%
1551     \textit{#1}:
1552     \begin{center}

```

```

1553 #2
1554 \medskip
1555 \end{center}
1556 \end{mdframed}
1557 } {}
1558 \ExplSyntaxOff

\liFussnoteUrl \liFussnoteUrl[\langle zusätzlicher-text \rangle]{\langle url \rangle} \liFussnoteUrl[zusätzlicher Text]{url}:
Eine HTTP-URL als Fußnote setzen.
1559 \NewDocumentCommand{\liFussnoteUrl} { o m } {
1560 \footnote{\url{#2}\IfNoValueTF{#1}{}{ ( #1 )}}
1561 }
1562

\liFussnoteLink \liFussnoteLink[\langle zusätzlicher-text \rangle]{\langle link-text \rangle}{\langle url \rangle} \liFussnoteLink[zusätzlicher Text]{tex
Einen Link, d. h. einen Link-Text und eine URL als Fußnote setzen.
1563 \NewDocumentCommand{\liFussnoteLink} { o m m } {
1564 \footnote{\href{#3}{#2}\IfNoValueTF{#1}{}{ ( #1 )}}
1565 }

\zB
1566 \def\zB{z.\,B. }

\ZB
1567 \def\ZB{Z.\,B. }

\dh
1568 \def\dh{d.\,h. }

1569

```


2.24 master-theorem.sty

1570 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

1571 \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}

1572 \ExplSyntaxOn

1573 \RequirePackage{amsmath}

\liRundeKlammer

1574 \def\liRundeKlammer#1{

1575 \negthinspace \left(#1 \right)

1576 }

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

1577 \def\liThetaOhneMathe#1{

1578 \Theta \liRundeKlammer{#1}

1579 }

1580 \def\liTheta#1{

1581 \ifmmode

1582 \liThetaOhneMathe{#1}

1583 \else

1584 \$\liThetaOhneMathe{#1}\$

1585 \fi

1586 }

```

\liOmega \liOmega{n^2}:  $\Omega(n^2)$ 
1587 \def\liOmegaOhneMathe#1{
1588 \Omega \liRundeKlammer{#1}
1589 }
1590 \def\liOmega#1{
1591 \ifmmode
1592 \liOmegaOhneMathe{#1}
1593 \else
1594 $\liOmegaOhneMathe{#1}$
1595 \fi
1596 }

\liO \liO{n^2}:  $\mathcal{O}(n^2)$ 
1597 \def\liOOhneMathe#1{
1598 \mathcal{O} \liRundeKlammer{#1}
1599 }
1600 \def\liO#1{
1601 \ifmmode
1602 \liOOhneMathe{#1}
1603 \else
1604 $\liOOhneMathe{#1}$
1605 \fi
1606 }

\liT Let-Abkürzung: \let\T=\liT
\liT{16}{2}:  $16 \cdot T\left(\frac{n}{2}\right)$  \liT{}{2}:  $T\left(\frac{n}{2}\right)$ 
1607 \def\liTOhneMathe#1#2{
1608 \tl_if_blank:nTF {#1}
1609 {}
1610 {#1 \cdot }
1611 T
1612 \liRundeKlammer{\frac{n}{#2}}
1613 }
1614 \def\liT#1#2{
1615 \ifmmode
1616 \liTOhneMathe{#1}{#2}
1617 \else
1618 $\liTOhneMathe{#1}{#2}$
1619 \fi
1620 }

\liRekursionsGleichung \liRekursionsGleichung:  $T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$ 
1621 \def\liRekursionsGleichung{
1622 $T(n) = \liT{a}{b} + f(n)$
1623 }

\liBedingungEins \liBedingungEins:  $f(n) \in \mathcal{O}(n^{\log_b a - \varepsilon})$ 
1624 \def\liBedingungEins{
1625 $f(n) \in \liO{n^{\log\sb{b}a - \varepsilon}}$
1626 }

\liBedingungZwei \liBedingungZwei:  $f(n) \in \Theta(n^{\log_b a})$ 
1627 \def\liBedingungZwei{
1628 $f(n) \in \liTheta{n^{\log\sb{b}a}}$
1629 }

\liBedingungDrei \liBedingungDrei:  $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ 
1630 \def\liBedingungDrei{
1631 $f(n) \in \liOmega{n^{\log\sb{b}a + \varepsilon}}$
1632 }

1633 \ExplSyntaxOff

```

\liMasterVariablen

```

1634 \def\liMasterVariablen{
1635   \begin{displaymath}
1636     T(n) = \liT{a}{b} + f(n)
1637   \end{displaymath}
1638
1639   \begin{itemize}
1640     \item[$a = $]
1641     Anzahl der Unterprobleme in der Rekursion
1642
1643     \item[$\textstyle{\frac{1}{b}} = $]
1644     Teil des Originalproblems, welches wiederum durch alle Unterprobleme
1645     repräsentiert wird
1646
1647     \item[$f(n) = $]
1648     Kosten (Aufwand, Nebenkosten), die durch die Division des Problems und
1649     die Kombination der Teillösungen entstehen
1650   \end{itemize}
1651   \footcite{wiki:master-theorem}
1652   \footcite[Seite 19-35 (PDF 11-24)]{aud:fs:2}
1653 }

```

\liMasterFaelle

```

1654 \def\liMasterFaelle{
1655   \begin{description}
1656     \item[1. Fall:]
1657      $T(n) \in \liTheta{n^{\log_{sb{b}}a}}$ 
1658
1659     \hfill falls \liBedingungEins
1660     für  $\varepsilon > 0$ 
1661
1662     \item[2. Fall:]
1663      $T(n) \in \liTheta{n^{\log_{sb{b}}a} \cdot \log n}$ 
1664
1665     \hfill falls \liBedingungZwei
1666
1667     \item[3. Fall:]
1668      $T(n) \in \liTheta{f(n)}$ 
1669
1670     \hfill falls \liBedingungDrei
1671     für  $\varepsilon > 0$ 
1672     und ebenfalls für ein  $c$  mit  $0 < c < 1$  und alle hinreichend großen  $n$ 
1673     gilt:
1674      $a \cdot f(\textstyle{\frac{n}{b}}) \leq c \cdot f(n)$ 
1675   \end{description}
1676 }

```

\liMasterVariablenDeklaration

```

1677 \def\liMasterVariablenDeklaration#1#2#3{
1678   \begin{description}
1679     \item[Allgemeine Rekursionsgleichung:] \strut
1680
1681     \liRekursionsGleichung
1682
1683     \item[Anzahl der rekursiven Aufrufe ( $a$ ):] \strut
1684
1685     #1
1686
1687     \item[Anteil Verkleinerung des Problems ( $b$ ):] \strut
1688
1689     um  $\frac{1}{\#2}$  also  $b = \#2$ 
1690
1691     \item[Laufzeit der rekursiven Funktion ( $f(n)$ ):] \strut

```

```

1692
1693     $#3$
1694
1695     \item[Ergibt folgende Rekursionsgleichung:] \strut
1696
1697      $T(n) = T_{\#1\#2} + \#3$ 
1698     \end{description}
1699 }

\liMasterFallRechnung
1700 \def\liMasterFallRechnung#1#2#3{
1701     \begin{description}
1702     \item[1. Fall:] \liBedingungEins:
1703
1704     #1
1705
1706     \item[2. Fall:] \liBedingungZwei:
1707
1708     #2
1709
1710     \item[3. Fall:] \liBedingungDrei:
1711
1712     #3
1713     \end{description}
1714 }

\liMasterExkurs
1715 \def\liMasterExkurs{
1716     \begin{liExkurs}[Master-Theorem]
1717     \liMasterVariablen
1718
1719     \noindent
1720     Dann gilt:
1721
1722     \liMasterFaelle
1723     \end{liExkurs}
1724 }

\liMasterWolframLink Link zu Wolframalpha (+ durch Prozent 2B ersetzen)
1725 \def\liMasterWolframLink#1{
1726     Berechne die Rekursionsgleichung auf WolframAlpha:
1727     \href{https://www.wolframalpha.com/input/?i=\#1\{WolframAlpha\}}
1728 }

1729

```

2.25 mathe.sty

```
1730 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1731 \ProvidesPackage{lehramt-informatik-mathe}[2020/06/10]
1732
1733 % for example \ltimes \rtimes
1734 %\RequirePackage{amssymb}
1735 \RequirePackage{amsmath}
1736
1737 %%
1738 % \mlq \mrq
1739 %%
1740 \DeclareMathSymbol{\mlq}{\mathord}{operators}{``}
1741 \DeclareMathSymbol{\mrq}{\mathord}{operators}{`}
1742
```

2.26 minimierung.sty

```

1743 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1744 \ProvidesPackage{lehramt-informatik-mathe}[2021/03/13 Für den
1745 Minimierungsalgorithmus von einem NEA zu einem DEA]

1746 \liLadePakete{typographie}

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

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

\liFussnoten

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

\liFussnote

1747 \def\liFussnote#1{\$x_{#1}\$}

1748 \def\li@fussnote@text#1#2{
1749 \liFussnote{#1}
1750 \quad
1751 {\footnotesize #2}
1752 }

\liFussnoteEinsText

1753 \def\liFussnoteEinsText{
1754 \li@fussnote@text{1}
1755 {Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.}
1756 }

\liFussnoteZweiText

1757 \def\liFussnoteZweiText{
1758 \li@fussnote@text{2}
1759 {Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt.}
1760 }

\liFussnoteDreiText

1761 \def\liFussnoteDreiText{
1762 \li@fussnote@text{3}

```

```

1763 {In weiteren Iterationen markierte Zustände.}
1764 }

\liFussnoteVierText
1765 \def\liFussnoteVierText{
1766   \li@fussnote@text{4}
1767   {...}
1768 }

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



1769 \def\liFussnoten{
1770   \bigskip
1771
1772   \noindent
1773   \liFussnoteEinsText
1774
1775   \noindent
1776   \liFussnoteZweiText
1777
1778   \noindent
1779   \liFussnoteDreiText
1780
1781   \noindent
1782   \liFussnoteVierText
1783 }

\liLeereZelle \liLeereZelle:  $\emptyset$ 
Let-Abkürzung: \let\l=\liLeereZelle
1784 \def\liLeereZelle{\$\emptyset\$}

\liZustandsPaarVariablenName
1785 \def\liZustandsPaarVariablenName{z}

\liZustandsPaar
1786 \def\liZustandsPaar#1#2{
1787   $(
1788     \liZustandsPaarVariablenName_#1,
1789     \liZustandsPaarVariablenName_#2
1790   )$
1791 }

liUebergangsTabelle
1792 \renewcommand{\arraystretch}{1.4}
1793 \NewDocumentEnvironment{ liUebergangsTabelle } { m m } {
1794   \liPseudoUeberschrift{Übergangstabelle}
1795   \begin{center}
1796     \begin{tabular}{r|l|l}
1797       \textbf{Zustandspaar} & \textbf{#1} & \textbf{#2} \\ \hline
1798     \end{tabular}
1799   \end{center}
1800 }

liUeberschriftDreiecksTabelle \liUeberschriftDreiecksTabelle: Minimierungstabelle (Table filling)

1802 \ExplSyntaxOn
1803 \def\liUeberschriftDreiecksTabelle{
1804   \liPseudoUeberschrift{Minimierungstabelle~(Table~filling)}
1805 }

```

\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. —————

```

1806 \def\liMinimierungErklaerung{
1807   %\footcite[Seite~19]{koenig}
1808   \liParagraphMitLinien{
1809     Der~Minimierungs-Algorithmus~(auch~Table-Filling-Algorithmus-genannt)~
1810     trägt~in~seinem~Verlauf~eine~Markierung~in~alle~
1811     diejenigen~Zellen~der~Tabelle~ein,~die~zueinander~nicht~äquivalente~
1812     Zustände~bezeichnen.~Die~Markierung~„ $x_{\sb{n}}$ “~in~einer~Tabellenzelle~( $i$ ,~
1813      $j$ )~bedeutet~dabei,~dass~das~Zustandspaar~( $i$ ,~ $j$ )~in~der~ $k$ -ten~
1814     Iteration~des~Algorithmus~markiert~wurde~und~die~Zustände~ $i$ ~und~ $j$ ~
1815     somit~zueinander~( $k - 1$ )~äquivalent,~aber~nicht~ $k$ -äquivalent~und~
1816     somit~insbesondere~nicht~äquivalent~sind.~Bleibt~eine~Zelle~bis~zum~Ende~
1817     unmarkiert,~sind~die~entsprechenden~Zustände~zueinander~äquivalent.
1818   }
1819 }
1820 \ExplSyntaxOff
1821

```


2.27 normalformen.sty

```

1822 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1823 \ProvidesPackage{lehramt-informatik-normalformen}[2020/12/10
1824 Hilfsmakros zum Setzen von Normalformen, Funktionalen Abhängigkeiten,
1825 Attributhülle]
1826 \liLadePakete{mathe}
1827 \directlua{
1828   helfer = require('lehramt-informatik-helfer')
1829   normalformen = require('lehramt-informatik-normalformen')
1830 }

```

Let-Abkürzungen

```

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

```

```

1831 \def\liTeilen#1{
1832   \directlua{tex.print(normalformen.teilen('#1'))}
1833 }

```

\liAttributHuelle Let-Abkürzung: `\let\ah=\liAttributHuelle`
 Regulärer Ausdruck zum Konvertieren `AttrHülle\((.*)\)` in `\ah{$1}`

```

1834 \def\liAttributHuelleOhneMathe#1{\text{AttrHülle}(#1)}
1835 \def\liAttributHuelle#1{
1836   \ifmmode
1837     \liAttributHuelleOhneMathe{#1}
1838   \else
1839     $\liAttributHuelleOhneMathe{#1}$
1840   \fi
1841 }

```

\liAttributMenge Let-Abkürzung: `\let\m=\liAttributMenge`
 1842 \def\liAttributMenge#1{\{ \textit{#1} \}}

liAHuelle

```

1843 \NewDocumentEnvironment{ liAHuelle } { +b } {
1844   \begin{group}
1845   \footnotesize
1846   \begin{multline*}
1847     #1
1848   \end{multline*}
1849   \end{group}
1850 } { }

```

AttributHuelleLinksReduktion Nur innerhalb von `liAHuelle` zu verwenden bzw. `multline`

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

```

1851 \def\liAttributHuelleLinksReduktion#1#2#3{
1852   \shoveleft{
1853     \liAttributHuelleOhneMathe{FA, \liAttributMenge{\liAttributMenge{#1} - \liAttributMenge{#2}}
1854   } \\\
1855   \shoveright{
1856     \liAttributMenge{#3}
1857   } \\\
1858 }

```

```

\liFunktionaleAbhaengigkeit Let-Abkürzung: \let\fa=\liFunktionaleAbhaengigkeit
    \liFunktionaleAbhaengigkeit{A, B -> C, D} Regulärer Ausdruck zum Konvertieren:
    \$(.*) \rightarrow (.*)\$ \fa{$1 -> $2}
1859 \def\liFunktionaleAbhaengigkeit#1{%
1860   \directlua{normalformen.drucke_funk_abhaengigkeit('#1')}%
1861 }

\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;
1862 \NewDocumentCommand {\liFunktionaleAbhaengigkeiten} { O{FA} m } {
1863   \par
1864   \noindent
1865   #1 $= \{$
1866   \par
1867   \directlua{normalformen.drucke_funk_abhaengigkeiten('#2')}
1868   \par
1869   \noindent$}\$
1870 }

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

```

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

```

    Regulärer Ausdruck zum Konvertieren: \$(R.*)\((.*)\)\$ \liRelation[$1]{$2}
1871 \NewDocumentCommand {\liRelation} { O{R} m } {
1872   $\directlua{
1873     local name = helper.konvertiere_tiefgestellt('#1')
1874     tex.print(name)
1875   }$(\textit{\,#2\,})
1876 }
1877

```

2.28 petri.sty

```
1878 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1879 \ProvidesPackage{lehramt-informatik-petri}[2020/12/03]
```

Let-Abkürzungen

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

```
1880 \RequirePackage{tikz}
1881 \usetikzlibrary{petri,arrows.meta}
```

Für die Darstellungsmatrix

```
1882 \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

```
1883 \def\liPetriSetzeSchluessel{%
1884   \def\TmpTransitionOne{}%
1885   \def\TmpTransitionTwo{}%
1886   \def\TmpTransitionThree{}%
1887   \def\TmpTransitionFour{}%
1888   \def\TmpTransitionFive{}%
1889   \def\TmpTransitionSix{}%
1890   \def\TmpTransitionSeven{}%
1891   \def\TmpTransitionEight{}%
1892   \def\TmpTransitionNine{}%
1893   \def\TmpTransitionTen{}%
1894   \pgfkeys{/petri/.cd,
1895     p1/.store in=\TmpPlaceOne,p1/.default=0,p1,
1896     p2/.store in=\TmpPlaceTwo,p2/.default=0,p2,
1897     p3/.store in=\TmpPlaceThree,p3/.default=0,p3,
1898     p4/.store in=\TmpPlaceFour,p4/.default=0,p4,
1899     p5/.store in=\TmpPlaceFive,p5/.default=0,p5,
```

```

1900     p6/.store in=\TmpPlaceSix,p6/.default=0,p6,
1901     p7/.store in=\TmpPlaceSeven,p7/.default=0,p7,
1902     p8/.store in=\TmpPlaceEight,p8/.default=0,p8,
1903     p9/.store in=\TmpPlaceNine,p9/.default=0,p9,
1904     p10/.store in=\TmpPlaceTen,p10/.default=0,p10,
1905     t1/.store in=\TmpTransitionOne,t1/.default=activated,
1906     t2/.store in=\TmpTransitionTwo,t2/.default=activated,
1907     t3/.store in=\TmpTransitionThree,t3/.default=activated,
1908     t4/.store in=\TmpTransitionFour,t4/.default=activated,
1909     t5/.store in=\TmpTransitionFive,t5/.default=activated,
1910     t6/.store in=\TmpTransitionSix,t6/.default=activated,
1911     t7/.store in=\TmpTransitionSeven,t7/.default=activated,
1912     t8/.store in=\TmpTransitionEight,t8/.default=activated,
1913     t9/.store in=\TmpTransitionNine,t9/.default=activated,
1914     t10/.store in=\TmpTransitionTen,t10/.default=activated,
1915     scale/.store in=\TmpScale,scale/.default=0.5,
1916     x/.store in=\TmpX,x/.default=5,
1917     y/.store in=\TmpY,y/.default=5,
1918   }%
1919 }

1920 \tikzset{
1921   li petri/.style={
1922     activated/.style={
1923       very thick
1924     },
1925     inhibitor/.style={
1926       {Circle[open,length=2mm,fill=white]}-
1927     }
1928   }
1929 }

```

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

```

1930 \def\liPetriTransitionsNameOhneMathe#1{t\sb{#1}}
1931 \def\liPetriTransitionsName#1{
1932   \ifmmode
1933     \liPetriTransitionsNameOhneMathe{#1}
1934   \else
1935     $\liPetriTransitionsNameOhneMathe{#1}$
1936   \fi
1937 }

```

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

```

1938 \NewDocumentCommand{ \liPetriErreichTransition } { m m m O{} O{} } {
1939   \draw[->] (#1) edge[#4] node[pos=0.5,auto,sloped,#5]{t\sb{#3}$} (#2);
1940 }

```

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

```

1941 \def\liPetriErreichKnotenDrei#1#2#3{(#1,#2,#3)}

```

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

```

1942 \def\liPetriTransPfeile#1{ $\rightarrow \hspace{0.4cm} \liPetriTransitionsName{#1} \hspace{0.3cm} }
1943

```

2.29 potenzmengen-konstruktion.sty

```
1944 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1945 \ProvidesPackage{lehramt-informatik-potenzmengen-konstruktion}[2021/02/21 Hilfsmakros
1946 einzusetzen bei der Potenzmengen-Konstruktion (Potenzmengen-Algorithmus)]
```

```
1947 \liLadePakete{formale-sprachen}
1948 \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}
    }
  }
}
```

```
1949 \def\liZustandsMengenSammlung#1#2{
1950   \liZustandsnameGross{#1}
1951   {
1952     \footnotesize
1953     \liPotenzmenge{
1954       \str_case:nn {#1} {#2
1955       }
1956     }
1957 }
```

```
\liZustandsMengenSammlungNr
```

```
1958 \def\liZustandsMengenSammlungNr#1#2{
1959   \liZustandsnameGross{#1}
1960   {
```

```
1961    \footnotesize
1962    \liZustandsmengeNr{
1963        \str_case:nn {#1} #2
1964    }
1965 }
1966 }

1967 \ExplSyntaxOff
1968
```

2.30 pseudo.sty

1969 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 1970 \ProvidesPackage{lehramt-informatik-pseudo}[2020/12/30 Paket zum Setzen
 1971 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>

1972 \RequirePackage[german,boxruled]{algorithm2e}

1973

2.31 pumping-lemma.sty

1974 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 1975 \ProvidesPackage{lehramt-informatik-pumping-lemma}[2021/08/11 Enthält die
 1976 Definitionen für das Pumping-Lemma in der Regulären Sprache und
 1977 in der Kontextfreien Sprache]

\liPumpingRegulaer

```

1978 \def\liPumpingRegulaer{%
1979   Es sei  $L$  eine reguläre Sprache. Dann gibt es eine Zahl  $j$ , sodass für
1980   alle Wörter  $\omega \in L$  mit  $|\omega| \geq j$  (jedes Wort  $\omega$  in
1981    $L$  mit Mindestlänge  $j$ ) jeweils eine Zerlegung  $\omega = uvw$ 
1982   existiert, sodass die folgenden Eigenschaften erfüllt sind:
1983
1984   \begin{enumerate}
1985     \item  $|v| \geq 1$ 
1986     (Das Wort  $v$  ist nicht leer.)
1987
1988     \item  $|uv| \leq j$ 
1989     (Die beiden Wörter  $u$  und  $v$  haben zusammen höchstens die Länge  $j$ .)
1990
1991     \item Für alle  $i = 0, 1, 2, \dots$  gilt  $uv^i w \in L$ 
1992     (Für jede natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^i w$  in der
1993     Sprache  $L$ )
1994   \end{enumerate}
1995
1996   Die kleinste Zahl  $j$ , die diese Eigenschaften erfüllt, wird
1997   Pumping-Zahl der Sprache  $L$  genannt.\footcite{wiki:pumping-lemma}
1998 }
```

\liPumpingKontextfrei

```

1999 \def\liPumpingKontextfrei{%
2000   Es sei  $L$  eine kontextfreie Sprache. Dann gibt es eine Zahl  $j$ , sodass
2001   sich alle Wörter  $\omega \in L$  mit  $|\omega| \geq j$  zerlegen lassen in
2002    $\omega = uvwxy$ , sodass die folgenden Eigenschaften erfüllt sind:
2003
2004   \begin{enumerate}
2005     \item  $|vx| \geq 1$ 
2006     (Die Wörter  $v$  und  $x$  sind nicht leer.)
2007
2008     \item  $|vwx| \leq j$ 
2009     (Die Wörter  $v$ ,  $w$  und  $x$  haben zusammen höchstens die Länge  $j$ .)
2010
2011     \item Für alle  $i \in \mathbb{N}_0$  gilt  $uv^iwx^iy \in L$  (Für jede
2012     natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^iwx^iy$  in der
2013     Sprache  $L$ )
2014   \end{enumerate}
2015 }

2016
```


2.32 quicksort.sty

```

2017 % https://tex.stackexchange.com/a/142634
2018 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2019 \ProvidesPackage{lehramt-informatik-quicksort}[2020/06/12]
2020
2021 %-----
2022 % USAGE:
2023 % \QSinitialize{comma, separated, numerical, values}
2024 % \loop
2025 % \QSpivotStep
2026 % \ifnum\value{pivotcount}>0
2027 %   \QSSortStep
2028 % \repeat
2029 %-----
2030
2031 % xintfrac does not load xinttools, this must be done explicitly if needed as here.
2032 \RequirePackage{xintfrac, xinttools}
2033
2034 \RequirePackage{tikz}
2035
2036 %-----
2037 % FIRST PART: TikZ styles and macros for the actual drawing
2038 \newcounter{cellcount}% used for coordinates of the node
2039 \newcounter{pivotcount}% when it will remain at zero, will signal the sort is finished.
2040
2041 % Styles defined by Tom Bombaldi. (modified: all share the same size)
2042 % (re-modified \bf -> \bfseries due to extremely annoying warnings from
2043 % KOMA-script which are truly a pain and do not make any sense regarding \bf:
2044 % if I want to use \bf, and know what I am doing, why should I get HARASSED
2045 % by police of LaTeX good conduct ? )
2046 \tikzset{ll/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=lime!70!gray},
2047          oo/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=olive!50},
2048          rr/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2049 % this is the "b" style as used in the image below
2050          bb/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2051 % nicer:
2052          bb/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=white, text=magenta},
2053          gg/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=gray, text=white, text=white},
2054
2055 % NOTE the b style was originally the same as the r(aised) style apart from
2056 % not being raised, but I find it nicer with a somewhat different
2057 % specification. I have not updated the images though.
2058
2059 % How the nodes are drawn depending on whether on the left of the pivot value
2060 % or on the right, or is a pivot value, or a raised pivot during selection phase.
2061
2062 \def\DecoLEFT #1{%
2063   \xintFor* ##1 in {#1} \do
2064     {\stepcounter{cellcount}\node[o] at (\arabic{cellcount},0) {##1};}%
2065 }
2066
2067 \def\DecoINERT #1{%
2068   \xintFor* ##1 in {#1} \do
2069     {\stepcounter{cellcount}\node[g] at (\arabic{cellcount},0) {##1};}%
2070 }
2071
2072 \def\DecoRIGHT #1{%
2073   \xintFor* ##1 in {#1} \do
2074     {\stepcounter{cellcount}\node[l] at (\arabic{cellcount},0) {##1};}%
2075 }
2076
2077 \def\DecoLEFTwithPivot #1{\stepcounter{pivotcount}%
2078   \xintFor* ##1 in {#1} \do

```

```

2079     {\stepcounter{cellcount}}%
2080     \xintifForLast {\node[r]}\node[o]} at (\arabic{cellcount},0) {##1};}%
2081 }
2082
2083 \def\DecoINERTwithPivot #1{\stepcounter{pivotcount}%
2084     \xintFor* ##1 in {#1} \do
2085     {\stepcounter{cellcount}}%
2086     \xintifForLast {\node[b]}\node[g]} at (\arabic{cellcount},0) {##1};}%
2087 }
2088
2089 \def\DecoRIGHTwithPivot #1{\stepcounter{pivotcount}%
2090     \xintFor* ##1 in {#1} \do
2091     {\stepcounter{cellcount}}%
2092     \xintifForLast {\node[r]}\node[l]} at (\arabic{cellcount},0) {##1};}%
2093 }
2094
2095 %-----
2096 % SECOND PART: the actual sorting routines.
2097
2098 \def\QS@sort@a #1{\expandafter \QS@sort@b \expandafter {\xintLength {#1}}{#1}}
2099 \def\QS@sort@b #1{\ifcase #1
2100     \expandafter\QS@sort@empty
2101     \or\expandafter\QS@sort@single
2102     \else\expandafter\QS@sort@c
2103     \fi
2104 }%
2105 \def\QS@sort@empty #1{}
2106 \def\QS@sort@single #1{\QSIr {#1}}
2107
2108 % This step is to pick the last as pivot.
2109 \def\QS@sort@c #1%
2110     {\expandafter\QS@sort@d\expandafter {\romannumeral0\xintnthelt {-1}{#1}}{#1}}%
2111
2112 % Here \QSLr, \QSIr, \QSR have been let to \relax.
2113 % The trick with \xintApplyUnbraced is that for example when selecting
2114 % the elements smaller than pivot, if we had been using \xintApply we
2115 % would have had at the minimum an empty brace pair. Thus we use the
2116 % "unbraced" variant, but then the \QS@select@smaller has added in
2117 % anticipation a level of braces.
2118 \def\QS@sort@d #1#2{%
2119     \QSLr {\xintApplyUnbraced {\QS@select@smaller {#1}}{#2}}}%
2120     \QSIr {\xintApplyUnbraced {\QS@select@equal {#1}}{#2}}}%
2121     \QSRr {\xintApplyUnbraced {\QS@select@greater {#1}}{#2}}}%
2122 }%
2123 \def\QS@select@smaller #1#2{\xintifLt {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2124 \def\QS@select@equal #1#2{\xintifEq {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2125 \def\QS@select@greater #1#2{\xintifGt {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2126
2127 %
2128 % NOTE 1: thus, each comparison with the pivot is done three (!) times.
2129 %
2130 % NOTE 2: we may well end up with \QSLr {<empty>} situations. This is handled
2131 % silently by the \xintFor loops, and also when \QSLr becomes \QS@sort@a, the
2132 % latter must handle correctly an empty argument.
2133
2134 %-----
2135 % THIRD PART: the main macros \QSpivotStep, \QSsortStep and \QSinitialize.
2136
2137 % This draws all with suitable highlighting for the newly chosen pivots
2138 % (which will be shown raised)

```

```

2139 \def\QSpivotStep {\let\QSLr\DecoLEFTwithPivot
2140             \let\QSIr\DecoINERT
2141             \let\QSIrr\DecoINERT
2142             \let\QSRr\DecoRIGHTwithPivot
2143 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2144             \setcounter{cellcount}{0}\setcounter{pivotcount}{0}%
2145             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2146 }
2147
2148 % This sorts and then draws, showing where the pivot chosen in the previous
2149 % step go. Next time they will have become "inert". If pivotcount is still at
2150 % zero on exit from \QSpivotStep, then this is the signal to stop before
2151 % executing \QSsortStep.
2152 \def\QSsortStep {\def\QSLr {\noexpand\QS@sort@a}%
2153             \def\QSRr {\noexpand\QS@sort@a}%
2154             \def\QSIr {\noexpand\QSIrr}%
2155             \let\QSIrr\relax
2156             \edef\QS@list{\QS@list}%
2157             \let\QSLr\relax
2158             \let\QSRr\relax
2159             \let\QSIr\relax
2160             \edef\QS@list{\QS@list}%
2161             \let\QSLr\DecoLEFT
2162             \let\QSIr\DecoINERTwithPivot
2163             \let\QSIrr\DecoINERT
2164             \let\QSRr\DecoRIGHT
2165 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2166             \setcounter{cellcount}{0}%
2167             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2168 }
2169
2170 \def\QSinitialize #1{%
2171     % first, we convert the comma separated values into a list of braced items
2172     % we use an \edef, and anyhow many \edef's will be used later
2173     \edef\QS@list {\noexpand\QSRr {\xintCSVtoList {#1}}}%
2174     \let\QSRr\DecoRIGHT
2175     % The \QSRr marker mutated to draw the last element as
2176     % pivot and the earlier ones with the suitable style.
2177     %
2178     % The list of marked braced items \QS@list is used both for drawing
2179     % (as here) and for doing the exchange of elements during sort.
2180     \par\centerline{\rule[1.5mm]{0pt}{8mm}\setcounter{cellcount}{0}%
2181             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2182 }
2183

```

2.33 relationale-algebra.sty

```

2184 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2185 \ProvidesPackage{lehramt-informatik-relationale-algebra}[2020/12/11]
2186 \RequirePackage{amsmath}
2187 \RequirePackage{amssymb}

```

```

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

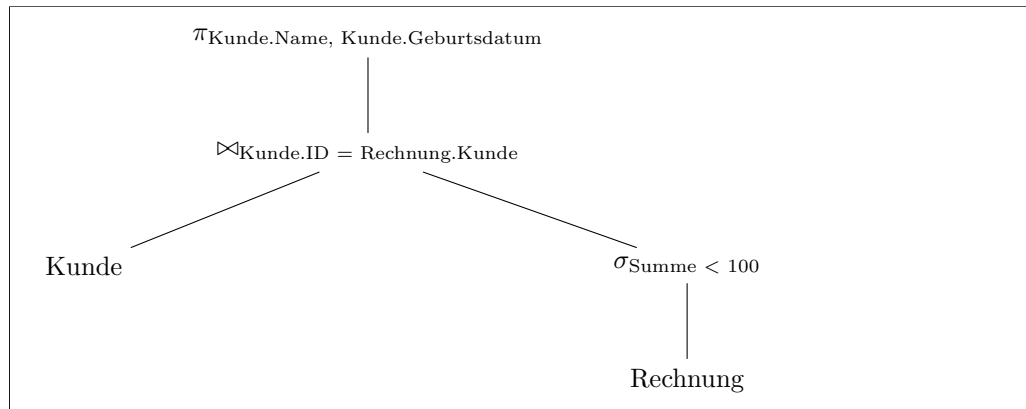
  \node[below=of pi]
    (theta join) {\bowtie_{\text{Kunde.ID = Rechnung.Kunde}}};
    edge (pi);

  \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}

```



```

2188 \RequirePackage{tikz}
2189 \usetikzlibrary{positioning}

    Privates Makros, das zwei Querstriche erzeugt.
2190 \def\o@join{\setbox0=\hbox{\bowtie}}%
2191   \rule[-.02ex]{.25em}{.4pt}\llap{\rule[\ht0]{.25em}{.4pt}}%
2192 }

\leftouterjoin A \leftouterjoin B: A ⋈ B
2193 \def\leftouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie}}

\rightouterjoin A \rightouterjoin B: A ⋉ B
2194 \def\rightouterjoin{\mathbin{\bowtie\mkern-5.8mu\o@join}}

\fullouterjoin A \fullouterjoin B: A ⋈⋉ B
2195 \def\fullouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie\mkern-5.8mu\o@join}}
2196

```

2.34 rmodell.sty

```

2197 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2198 \ProvidesPackage{lehramt-informatik-rmodell}[2020/09/01
2199 Makros und Umgebungen zum Setzen des Relationenmodells beim Thema
2200 Datenbanken.]
2201 \RequirePackage{soul}

```

Let-Abkürzungen

```

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

```

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

```
2202 \def\liPrimaer#1{\ul{#1}}
```

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

```
2203 \def\liFremd#1{{\setul{-0.9em}{}}\ul{#1}}
```

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

```

2204 \def\li@Rmodell@Schrift{\footnotesize\ttfamily}
2205 \ExplSyntaxOn
2206 \NewDocumentEnvironment { liRmodell }
2207 { +b }
2208 {
2209   \medskip
2210   {
2211     \linespread{2}
2212     \setlength{\parindent}{0pt}
2213     \li@Rmodell@Schrift#1
2214   }
2215   \medskip
2216 } {}
2217 \ExplSyntaxOff

```

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

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

```

2218 \def\liRelationMenge#1#2{
2219 \noindent
2220 #1 : \{ [ #2 ] \}
2221 \par
2222 }

```

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

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

```
2223 \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}

```

```
2224 \NewDocumentEnvironment { liRelationenSchemaFormat }{ +b } {} {}
```

```
2225
```

2.35 sortieren.sty

```
2226 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2227 \ProvidesPackage{lehramt-informatik-sortieren}[2020/06/10 Schaubilder
2228 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}
}
```



```
2229 \RequirePackage{tikz}
2230 \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.

```
2231 \def\liVertauschen#1{
2232   \directlua{
2233     local sortieren = require('lehramt-informatik-sortieren')
2234     sortieren('#1')
2235   }
2236 }
```

`\liSortierPfeil`

```
2237 \def\liSortierPfeil#1#2{
2238   \draw[-latex] ([xshift=1mm]reihe.#1 north) -- ++(0,0.25) -| ([xshift=-1mm]reihe.#2 north);
2239 }
```

`\liSortierPfeilUnten`

```
2240 \def\liSortierPfeilUnten#1#2{
2241   \draw[-latex] ([xshift=1mm]reihe.#1 south) -- ++(0,-0.25) -| ([xshift=-1mm]reihe.#2 south);
2242 }
```

`\liSortierMarkierung`

```
2243 \def\liSortierMarkierung#1#2{\node[
2244   draw,
2245   very thick,
2246   fit=(reihe.#1) (reihe.#2),
2247   inner sep=0pt
2248 ] {}
2249 }

2250 \tikzset{
2251   li sortierung zahlenreihe/.style={
2252     draw,
2253     thin,
2254     font=\large,
2255     rectangle split horizontal,
2256     rectangle split,
2257   }
2258 }
```

```

2259 % https://tex.stackexchange.com/a/140895
2260 \RequirePackage{forest,xstring}
2261 \usetikzlibrary{calc}
2262
2263 \makeatletter
2264 \pgfmathdeclarefunction{strrepeat}{2}{%
2265   \begingroup\pgfmathint{#2}\pgfmath@count\pgfmathresult
2266   \let\pgfmathresult\pgfutil@empty
2267   \pgfutil@loop\ifnum\pgfmath@count>0\relax
2268     \expandafter\def\expandafter\pgfmathresult\expandafter{\pgfmathresult#1}%
2269     \advance\pgfmath@count-1\relax
2270   \pgfutil@repeat\pgfmath@smuggleone\pgfmathresult\endgroup}
2271 \makeatother
2272
2273 \def\myNodes{}
2274
2275 \ExplSyntaxOn
2276 \newcommand*\sortList[1]{%
2277   \clist_sort:Nn#1{\int_compare:nNnTF{##1}>{##2}\sort_return_swapped:\sort_return_same:}}
2278 \ExplSyntaxOff
2279
2280 \forestset{
2281   sort/.code={%
2282     \pgfmathparse{level()>\forestSortLevel}%
2283     \ifnum\pgfmathresult=0
2284       \StrSubstitute{\forestov{content}}{ }{,}{\myList}%
2285       \sortList\myList
2286       \StrSubstitute{\myList}{,}{ }{\myList}%
2287       \pgfmathparse{strrepeat("1",level())}%
2288       \xappto\myNodes{\noexpand\node at ({\forestov{name}}|-m)!-1!({\forestov{name}}$)
2289         (m\forestov{name}) {\myList}}%
2290       \pgfmathparse{level()=\forestSortLevel}%
2291       \ifnum\pgfmathresult=1
2292         \forestOget{\forestov{@first}}{name}\forestFirst
2293         \forestOget{\forestov{@last}}{name}\forestLast
2294         \xappto\myNodes{{[<-]edge (\forestOv{\forestov{@first}}{name})
2295           \ifx\forestFirst\forestLast\else edge (\forestOv{\forestov{@last}}{name})\fi}}%
2296       \fi
2297       \ifnum\forestov{@parent}=0\else
2298         \xappto\myNodes{edge (m\forestOv{\forestov{@parent}}{name})}%
2299       \fi
2300       \gappto\myNodes{;}%
2301     \fi}}
2302
2303 \forestset{sort level/.code=%
2304   \pgfmathparse{#1}\let\forestSortLevel\pgfmathresult
2305   \pgfmathparse{strrepeat("1",\forestSortLevel+1)}\let\forestOnes\pgfmathresult}
2306

```

2.36 spalten.sty

```
2307 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2308 \ProvidesPackage{lehramt-informatik-spalten}[2020/12/07 Lädt das Paket
2309 „multicol“, damit mehrspaltiger Satz mit Hilfe der Umgebung „multicols“
2310 realisiert werden kann.]
2311 \RequirePackage{multicol}
```

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

```
2312 \def\liSpaltenUmbruch{\vfill\strut\columnbreak}
```

```
2313
```


2.37 struktogramm.sty

```
2314 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2315 \ProvidesPackage{lehramt-informatik-struktogramm}[2021/01/31 Lädt das
2316 Paket struktex zum Setzen von Struktogrammen]
2317 \RequirePackage{struktex}
2318
```

2.38 syntax.sty

```
2319 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2320 \ProvidesPackage{lehramt-informatik-syntax}[2020/06/10 Ein Hüll-Paket um
2321 `minted`, das sich um die Syntax-Hervorhebung des Java-Codes kümmert.]
2322 \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
```

```
2323 \ExplSyntaxOn
2324 \directlua{
2325   syntax = require('lehramt-informatik-syntax')
2326   syntax.importiere_konfiguration('tex_repo_lokaler_pfad', '\LehramtInformatikRepository')
2327   syntax.importiere_konfiguration('github_domain', '\LehramtInformatikGithubDomain')
2328   syntax.importiere_konfiguration('github_raw_domain', '\LehramtInformatikGithubRawDomain')
2329   syntax.importiere_konfiguration('github_tex_repo', '\LehramtInformatikGithubTexRepo')
2330   syntax.importiere_konfiguration('github_code_repo', '\LehramtInformatikGithubCodeRepo')
2331   syntax.importiere_konfiguration('git_branch', '\LehramtInformatikGitBranch')
2332 }
2333 \RequirePackage{hyperref}
2334 \RequirePackage{minted}
2335 % pygmentize -L styles
2336 \usemintedstyle{colorful}
2337 %\BeforeBeginEnvironment{minted}{\begin{mdframed}}
2338 %\AfterEndEnvironment{minted}{\end{mdframed}}
2339 %\setminted{breaklines=true,linenos}
2340 \setminted{
2341   breaklines=true,
2342   linenos,
2343   fontsize=\footnotesize,
2344 }
```

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

Let-Abkürzung: `\let\j=\liJavaCode`

```
2345 \def\liJavaCode#1{\,\mintinline{java}|#1|\,}
```

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

```
2346 \def\liLatexCode#1{\mintinline{latex}|#1|}
```

```
2347 \def\li@GithubLink#1#2{
2348   \begin{flushright}
2349     \tiny
2350     Code-Beispiel~auf~Github~ansehen:~
2351     \href{#1}{\nolinkurl{#2}}
2352   \end{flushright}
2353 }
```

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

```
2354 \NewDocumentCommand{\liJavaDatei}{ O{firstline=3} m }{
2355   \inputminted[#1]{java}{
2356     \directlua{
2357       syntax.drucke_absoluten_pfad('#2', false)
2358     }
2359   }
2360   \li@GithubLink
```

```

2361     {\directlua{syntax.drucke_github_url('#2', false)}}
2362     {\directlua{syntax.drucke_relativen_repo_pfad('#2', false)}}
2363 }

\liJavaTestDatei Eine komplette Java-Test-Datei einbinden, die Verzeichnis ./Code/src/test/java/org/bschlangaul
liegt.
2364 \NewDocumentCommand{\liJavaTestDatei}{ 0{firstline=3} m }{
2365   \inputminted[#1]{java}{
2366     \directlua{
2367       syntax.drucke_absoluten_pfad('#2', true)
2368     }
2369   }
2370   \li@GithubLink
2371   {\directlua{syntax.drucke_github_url('#2', true)}}
2372   {\directlua{syntax.drucke_relativen_repo_pfad('#2', true)}}
2373 }

\liJavaExamen \liJavaExamen{66116}{2015}{03}{Kunde} Regulärer Ausdruck zum Konvertieren: \liJavaDatei([.
\liJavaExamen$1{$2}{$3}{$4}{$5}
2374 \NewDocumentCommand{\liJavaExamen}{ 0{firstline=3} m m m m }{
2375   \inputminted[#1]{java}{
2376     \directlua{
2377       syntax.drucke_absoluten_examens_pfad('#2', '#3', '#4', '#5')
2378     }
2379   }
2380
2381   \li@GithubLink
2382   {\directlua{syntax.drucke_github_examens_url('#2', '#3', '#4', '#5')}}
2383   {\directlua{syntax.drucke_relativen_examens_repo_pfad('#2', '#3', '#4', '#5')}}
2384 }

\liAssemblerCode
2385 \def\liAssemblerCode#1{\mintinline{asm}|#1|}

\liAssemblerDatei \liAssemblerDatei{relativer-pfad}: Relativer Pfad zu einer Assembler-Datei.
2386 \NewDocumentCommand{\liAssemblerDatei}{ m }{
2387   \inputminted{asm}{#1}
2388 }

\liMinispracheDatei \liMinispracheDatei{relativer-pfad}: Relativer Pfad zu einer Minisprachen-Datei
(Hochsprache für die Minimaschine von Albert Wiedemann).
2389 \NewDocumentCommand{\liMinispracheDatei}{ m }{
2390   \inputminted{componentpascal}{#1}
2391 }

\liHaskellCode \liHaskellCode{haskell}: Zum Setzen von Haskell-Code.
2392 \def\liHaskellCode#1{\mintinline{haskell}|#1|}

\liHaskellDatei \liHaskellDatei{relativer-pfad}: Relativer Pfad zu einer Haskell-Datei.
2393 \NewDocumentCommand{\liHaskellDatei}{ m }{
2394   \inputminted{haskell}{#1}
2395 }

2396 \ExplSyntaxOff

\liSqlCode \liHaskellCode{sql}: Zum Setzen von SQL-Code.
Let-Abkürzung: \let\s=\liSqlCode
2397 \def\liSqlCode#1{\mintinline{sql}|#1|}

2398

```

2.39 syntaxbaum.sty

```
2399 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2400 \ProvidesPackage{lehramt-informatik-syntaxbaum}[2021/02/14 Zum Setzen von
2401 Syntaxbäumen mit Hilfe des Pakets tikz-qtrees]
2402 \RequirePackage{tikz-qtrees}
2403
2404 \tikzset{li parsetree/.style={
2405     every internal node/.style={
2406         draw,circle
2407     },
2408     every leaf node/.style={
2409         draw,rectangle
2410     },
2411 }
2412 }
2413
```

2.40 synthese-algorithmus.sty

```

2414 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2415 \ProvidesPackage{lehramt-informatik-synthese-algorithmus}[2021/03/19
2416 Hilfsmakros zum Setzen des Synthese-Algorithmuses zur Umformung einer
2417 Relation in die 3. Normalform]

2418 \liLadePakete{normalformen,mathe,typographie}
2419 \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

```

2420 \def\liSyntheseUeberschrift#1{
2421   {
2422     \bfseries
2423     \sffamily
2424     \str_case:nn {#1} {
2425       {1} {Kanonische-Überdeckung}
2426       {1-1} {Linksreduktion}
2427       {1-2} {Rechtsreduktion}
2428       {1-3} {Löschen-leerer-Klauseln}
2429       {1-4} {Vereinigung}
2430       {2} {Relationsschemata-formen}
2431       {3} {Schlüssel-hinzufügen}
2432       {4} {Entfernung-überflüssiger-Teilschemata}
2433     }
2434   }
2435 }

```

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

```

2436 \def\li@synthese@erklaerung@texte#1{
2437   \str_case:nn {#1} {
2438     {1} {
2439       Die-kanonische-Überdeckung---also-die-kleinst-mögliche-noch-
2440       äquivalente-Menge-von-funktionalen-Abhängigkeiten-kann-in-vier-
2441       Schritten-erreicht-werden.
2442     }
2443     {1-1} {
2444       Führe-für-jede-funktionale-Anhängigkeit-
2445       $\alpha\rightarrow\beta$-in-$F$-die-Linksreduktion-durch,-
2446       überprüfe-also-für-alle-
2447       $A\in\alpha$,~ob-$A$-überflüssig-ist,-d.h.-ob-
2448       $\beta\subseteqq\liAttributHuelle{F,\alpha\cup A}$.
2449     }
2450     {1-2} {
2451       Führe-für-jede-(verbliebene)-funktionale-Abhängigkeit-$\alpha\rightarrow\beta$-die-Rechtsreduktion-durch,-überprüfe-also-für-
2452       alle-$B\in\beta$,~ob-$B\in\liAttributHuelle{F,\alpha\rightarrow\beta\cup B}$,-
2453       $\alpha\rightarrow\beta\cup B$-gilt.-In-diesem-Fall-ist-$B$-auf-der-rechten-Seite-
2454       überflüssig-und-kann-eliminiert-werden,-dh-$\alpha\rightarrow\beta$-wird-durch-$\alpha\rightarrow\beta\cup B$-
2455       ersetzt.
2456     }
2457     {1-3} {
2458       Entferne-die-funktionalen-Abhängigkeiten-der-Form-$\alpha\rightarrow\emptyset$,~die-im-2.-Schritt-möglicherweise-
2459       entstanden-sind.
2460     }
2461     {1-4} {
2462       Fasse-mittels-der-Vereinigungsregel-funktionale-Abhängigkeiten-
2463       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.
2464     }
2465     % Kemper Seite 197
2466     {2} {
2467       Erzeuge-für-jede-funktionale-Abhängigkeit-$\alpha\rightarrow\beta\in F\sb{c}$-ein-Relationenschema-$\mathcal{R}\sb{\alpha}$-
2468       :=$\alpha\cup\beta$.
2469     }
2470     {3} {
2471       Falls-eines-der-in-Schritt-2.-erzeugten-Schemata-$R\sb{\alpha}$-einen-Schlüsselkandidaten-von-$\mathcal{R}$-bezüglich-$F\sb{c}$-
2472     }
2473   }
2474 }
2475 }
2476 }
2477 }
2478 }
2479 }

```

```

2480 enthält,~sind~wir~fertig,~sonst~wähle~einen~Schlüsselkandidaten~
2481  $\mathcal{K} \sim \text{subseq} \mathcal{R}$ ~aus~und~definiere~folgendes~
2482 zusätzliche~Schema:~ $\mathcal{R} \setminus \mathcal{K} \sim \mathcal{K}$ ~
2483 und~ $\mathcal{F} \setminus \mathcal{K} \sim \emptyset$ 
2484 }
2485 {4} {
2486 Eliminiere~diejenigen~Schemata~ $\mathcal{R} \setminus \alpha$ ~,~die~in~einem~
2487 anderen~Relationenschema~ $\mathcal{R} \setminus \alpha'$ ~enthalten~sind,~d.h.~
2488  $\mathcal{R} \setminus \alpha \sim \text{subseq} \mathcal{R} \setminus \alpha'$ ~.
2489 }
2490 }
2491 }
2492 \def\liSyntheseErklaerung#1{
2493 {
2494 \itshape
2495 \footnotesize
2496 \liParagraphMitLinien{\li@synthese@erklaerung@texte{#1}}
2497 }
2498 }

```

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

```

2499 \def\liSyntheseUeberErklaerung#1{
2500 \liSyntheseUeberschrift{#1}\par
2501 \liSyntheseErklaerung{#1}
2502 }

2503 \ExplSyntaxOff
2504

```

2.41 tabelle.sty

```
2505 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2506 \ProvidesPackage{lehramt-informatik-tabelle}[2020/12/05 Lädt das Paket tabluarx]
2507 \RequirePackage{tabularx}
2508
```


2.42 typographie.sty

```
2509 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2510 \ProvidesPackage{lehramt-informatik-typographie}[2021/03/20 Typographische Makros,
2511 die das Erscheinungsbild verändern. Die Schriftdefinition sind in
2512 formatierung.sty definiert.]
```

```
2513 \ExplSyntaxOn
```

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

```
2514 \RequirePackage{fontawesome}
```

```
\liErledigt \liErledigt: ☑
```

```
2515 \let\liErledigt=\faCheckSquareO
```

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

```
2516 \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 —

```
2517 \def\liParagraphMitLinien#1{
```

```
2518 \noindent
```

```
2519 \vrule height 2pt depth -1.6pt width 0.4cm
```

```
2520 \enspace
```

```
2521 #1
```

```
2522 \enspace
```

```
2523 \leaders\vrule height 2pt depth -1.6pt \hfill \null
```

```
2524 \par
```

```
2525 \medskip
```

```
2526 }
```

```
2527 \ExplSyntaxOff
```

```
2528
```

2.43 uml.sty

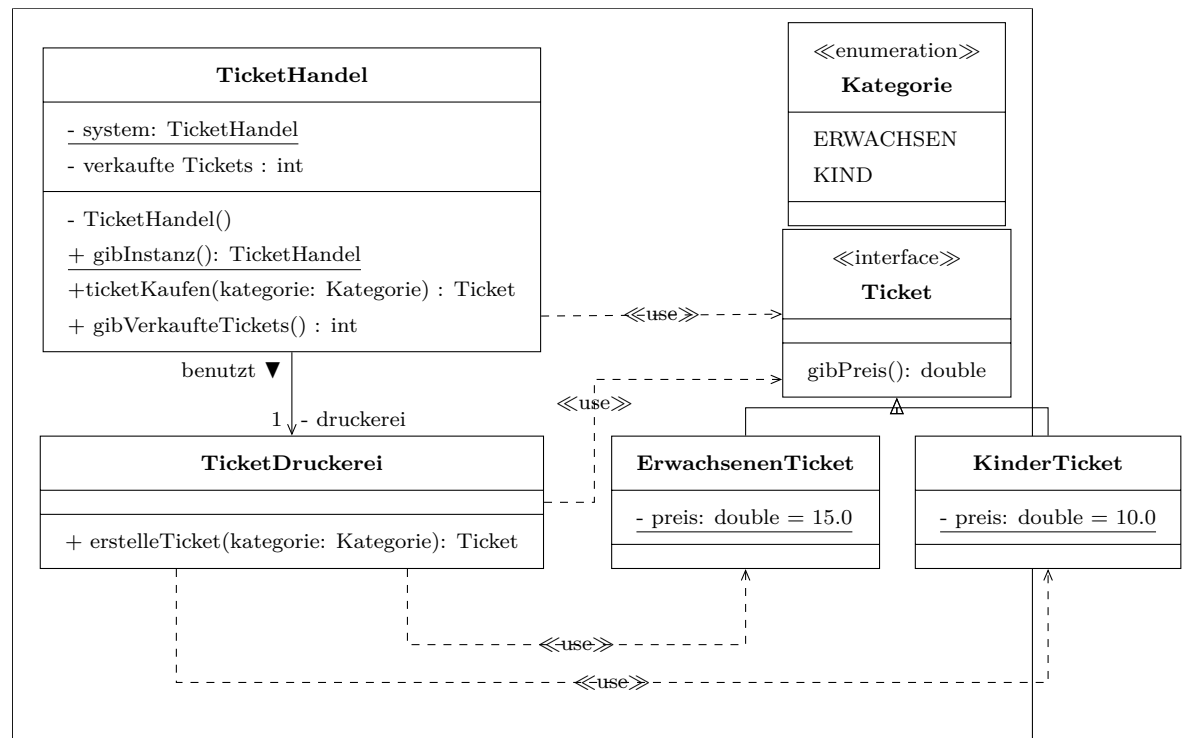
```

2529 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2530 \ProvidesPackage{lehramt-informatik-uml}[2020/06/13 Ein Hüll-Paket um
2531 `tikz-uml', das einige Design-Einstellungen vornimmt und manche
2532 Erweiterung bereitstellt]

2533 \RequirePackage{tikz-uml}
2534 \RequirePackage{tikz-uml-activity}
2535 % Not compatible with wasysym
2536 %\RequirePackage{mathabx}
2537 \RequirePackage{wasysym}
2538 \usetikzlibrary{positioning}

2539 \tikzumlset{
2540   fill class=white!0,
2541   font=\footnotesize,
2542   fill object=white!0,
2543   fill note=white!0,
2544   fill state=white!0,
2545   % Use case
2546   fill usecase=white!0,
2547   fill system=white!0,
2548 }

```



```

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

```

```

2549 \NewDocumentCommand{ \liUmlLeserichtung } { 0{dir=right} m } {
2550   \def\@liDirLeft{}
2551   \def\@liDirRight{ \RIGHTarrow}
2552   \pgfkeys{/lese/dir/.is choice}
2553   \pgfkeys{/lese/dir/up/.code={\def\@liDirRight{ \UParrow}}}
2554   \pgfkeys{/lese/dir/down/.code={\def\@liDirRight{ \DOWNarrow}}}
2555   \pgfkeys{/lese/dir/left/.code={\def\@liDirRight{}\def\@liDirLeft{\LEFTarrow }}}
2556   \pgfkeys{/lese/dir/right/.code={\def\@liDirRight{ \RIGHTarrow}}}
2557
2558   \def\@liPos{above}
2559   \pgfkeys{/lese/pos/.code={\def\@liPos{##1}}}
2560

```

```

2561 \def\@liDistance{0cm}
2562 \pgfkeys{/lese/distance/.code={\def\@liDistance{##1}}}}
2563
2564 \pgfkeys{/lese/.cd,#1}
2565
2566 \node[\@liPos = \@liDistance of #2-middle] {
2567   \@liDirLeft{\footnotesize#2}\@liDirRight
2568 };
2569 }
2570

```

2.44 vollstaendige-induktion.sty

2571 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2572 \ProvidesPackage{lehramt-informatik-vollstaendige-induktion}[2021/07/01
 2573 Hilfsmakros zum Setzen der Vollständigen Induktion, vor allem die
 2574 Überschriften für die einzelnen Schritte]

Let-Abkürzungen

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

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

Lade häufig benötigte Pakete

2575 \RequirePackage{lehramt-informatik-typographie}
 2576 \RequirePackage{lehramt-informatik-mathe}
 2577 \RequirePackage{lehramt-informatik-syntax}
 2578 \ExplSyntaxOn

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

Let-Abkürzung: \let\m=\liInduktionMarkierung

2579 \def\liInduktionMarkierung#1{\textcolor{violet}{#1}}

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

Let-Abkürzung: \let\e=\liInduktionErklaerung

2580 \def\liInduktionErklaerung#1{\scriptsize\text{#1}}

\liInduktionAnfang

2581 \def\liInduktionAnfang{

2582 \liPseudoUeberschrift{Induktionsanfang}

2583

2584 % https://de.wikibooks.org/wiki/Mathe_für_Nicht-Freaks:_Vollständige_Induktion

2585 \liParagraphMitLinien{

2586 Beweise,~dass~ $A(1)$ ~eine~wahre~Aussage~ist.

2587 }

2588 }

\liInduktionVoraussetzung

2589 \def\liInduktionVoraussetzung{

2590 \liPseudoUeberschrift{Induktionsvoraussetzung}

2591

2592 % https://de.wikibooks.org/wiki/Mathe_für_Nicht-Freaks:_Vollständige_Induktion

2593 \liParagraphMitLinien{

2594 Die~Aussage~ $A(k)$ ~ist~wahr~für~ein~beliebiges~ $k \in \mathbb{N}$ ~.

2595 }

2596 }

\liInduktionSchritt

2597 \def\liInduktionSchritt{

2598 \liPseudoUeberschrift{Induktionsschritt}

2599

2600 % https://de.wikibooks.org/wiki/Mathe_für_Nicht-Freaks:_Vollständige_Induktion

2601 \liParagraphMitLinien{

2602 Beweise,~dass~wenn~ $A(n=k)$ ~wahr~ist,~

2603 auch~ $A(n=k+1)$ ~wahr~sein~muss.

2604 }

2605 }

2606 \ExplSyntaxOff

2607

2.45 wasserfall.sty

```
2608 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2609 \ProvidesPackage{lehramt-informatik-wasserfall}[2020/06/10]
2610 \RequirePackage{tikz}
2611 \tikzset{wasserfall/.style={
2612   >=stealth,
2613   node distance = 2mm and -8mm,
2614   start chain = A going below right,
2615   every node/.style = {
2616     draw,
2617     text width=24mm,
2618     minimum height=12mm,
2619     align=center,
2620     inner sep=1mm,
2621     fill=white,
2622     drop shadow={fill=black},
2623     on chain=A
2624   },
2625 }}
2626 \usetikzlibrary{chains,positioning,shadows}
2627
```

2.46 wpkalkuel.sty

```
2628 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2629 \ProvidesPackage{lehramt-informatik-wpkalkuel}[2020/06/13]
```

Let-Abkürzungen

```
\let\wp=\liWpKalkuel
\let\equivalent=\liWpEquivalent
\let\erklaerung=\liWpErklaerung
```

```
2630 \RequirePackage{amsmath}
```

```
2631 \ExplSyntaxOn
```

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

```
2632 \def\liWpKalkuelOhneMathe#1#2{
2633   \text{wp}(\texttt{\scriptsize"#1"},\thinspace #2)
2634 }
2635 \def\liWpKalkuel#1#2{
2636   \ifmmode
2637     \liWpKalkuelOhneMathe{#1}{#2}
2638   \else
2639     $\liWpKalkuelOhneMathe{#1}{#2}$
2640   \fi
2641 }
```

```
\MatheEnv
```

```
2642 \def\MatheEnv#1{
2643   \medskip
2644   \hspace{1em}#1
2645   \medskip
2646   \hspace{1em}#1
2647   \medskip
2648 }
```

```
\Mathe
```

```
2649 \def\Mathe#1{
2650   \MatheEnv{#1$}
2651 }
```

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

```
2652 \def\liWpEquivalent#1{
2653   \MatheEnv{$\equiv$\hspace{1em}$#1$}
2654 }
```

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

```
2655 \newlength{\@Skip@Erklaerung@Reset}
2656 \def\liWpErklaerung#1{
2657   \setlength{\@Skip@Erklaerung@Reset}{\leftskip}
2658   \setlength{\leftskip}{0.5cm}
2659   \par
2660   \noindent
2661   {
2662     \scriptsize
2663     #1
2664   }
2665   \par
2666   \setlength{\leftskip}{\@Skip@Erklaerung@Reset}
2667   \par
2668   \setlength{\leftskip}{\@Skip@Erklaerung@Reset}
2669 }
```

```

2670 \def\liWpErklaerungVerzweigung{
2671   $\liWpKalkuel0hneMathe{if~\{{b~}\}~then~\{{a1~}\}~else~\{{a2~}\}}{Q}
2672   \equiv
2673   (b \land \liWpKalkuel0hneMathe{a1}{Q})
2674   \lor
2675   (\neg b \land \liWpKalkuel0hneMathe{a2}{Q})$
2676 }

2677 \ExplSyntaxOff

2678

```

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.

[illegible]

2324, 2356, 2361, 2362, 2366, 2371, 2372, 2376, 2382, 2383	\do 2063, 2068, 2073, 2078, 2084, 2090	\erzeuge@tiefgestellt . . . 1007, 1008, 1012	\forestov . 2284, 2288, 2289, 2292, 2293, 2294, 2295, 2297, 2298
\dots 506, 510, 1281, 1991, 2467, 2468	\DOWNarrow 2554	\expandafter 1194, 2098, 2100, 2101, 2102, 2110, 2268	\forestset 2280, 2303
\draw 1214, 1217, 1220, 1939, 2238, 2241		\ExplSyntaxOff 50, 92, 134, 139, 192, 197, 202, 393, 528, 550, 565, 986, 1068, 1103, 1379, 1394, 1558, 1633, 1820, 1967, 2217, 2278, 2396, 2503, 2527, 2606, 2677	\forestSortLevel 2282, 2290, 2304, 2305
E		\ExplSyntaxOn 22, 61, 102, 135, 160, 193, 198, 223, 470, 534, 551, 959, 1058, 1080, 1372, 1384, 1459, 1572, 1802, 1948, 2205, 2275, 2323, 2419, 2513, 2578, 2631	\frac 1612, 1643, 1674, 1689
\edef 1205, 2156, 2160, 2172, 2173			\fullouterjoin 2195
\else 570, 578, 586, 1002, 1016, 1054, 1480, 1583, 1593, 1603, 1617, 1838, 1934, 2102, 2295, 2297, 2638			G
\emph 962, 1262, 1291, 1293, 1439			\g 29, 37, 270, 271, 284, 288, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 307, 311, 312, 313, 316, 318, 319, 320, 321, 322, 323, 334, 335, 336, 337, 345, 346, 348, 354, 355, 357, 358, 360, 361, 369, 374, 376, 381, 383, 387
\empty 1480			\Gamma . 106, 164, 207, 1021
\emptyset 1784, 2462, 2483			\gappto 2300
\end . . . 653, 667, 692, 716, 750, 769, 779, 787, 806, 835, 852, 877, 899, 915, 931, 949, 1044, 1046, 1134, 1223, 1247, 1248, 1255, 1364, 1437, 1486, 1494, 1513, 1529, 1530, 1544, 1555, 1556, 1637, 1650, 1675, 1698, 1713, 1723, 1799, 1800, 1848, 1994, 2014, 2145, 2167, 2181, 2338, 2352		F	\geq 1288, 1980, 1985, 2001, 2005
\endcsname 1194, 1197		\faCheckSquare0 2515, 2516	H
\endgroup 1436, 1849, 2270		\faCircleThin 978	\hbox 2190
\enspace 2520, 2522		\faGg 972	\headrulewidth . 1142, 1392
environments:		\fancyfoot 1138, 1139, 1140, 1388, 1389, 1390, 1391	\hfill 1659, 1665, 1670, 2523
liAdditum 1487		\fancyhead 1137, 1385, 1386, 1387	\hinweis 1455
liAHuelle 1843		\faSquare0 966	\hline 1797
liAntwort 1458		\fi 572, 580, 588, 1004, 1018, 1056, 1483, 1585, 1595, 1605, 1619, 1840, 1936, 2103, 2295, 2296, 2299, 2301, 2640	\href 370, 1076, 1564, 1727, 2351
liDiagramm 1546		\fontspec 1112	\hspace . . 1942, 2645, 2653
liEinbettung 1457		\footcite 668, 690, 726, 749, 768, 807, 834, 876, 1260, 1263, 1270, 1275, 1280, 1284, 1290, 1295, 1399, 1651, 1652, 1807, 1997	\ht 2191
liExkurs 1495		\footnote 1560, 1564	I
liGraphenFormat . 1186		\footnotesize 147, 341, 424, 520, 983, 1455, 1508, 1523, 1539, 1751, 1845, 1952, 1961, 2204, 2343, 2495, 2541, 2567	\i 1219, 1220
liKasten 1131		\footrulewidth . 1143, 1393	\ifcase 2099
liKontrollflussgraph 1358		\foreach . 1209, 1212, 1219	\ifmode 568, 576, 584, 1000, 1014, 1052, 1581, 1591, 1601, 1615, 1836, 1932, 2636
liLernkartei 1536		\forestFirst . . 2292, 2295	\IfNoValueTF 1490, 1560, 1564
liProduktionsRegeln 1036		\forestLast . . . 2293, 2295	\ifnum 2026, 2267, 2283, 2291, 2297
liProjektSprache 1456		\forestOget . . . 2292, 2293	\ifx 1480, 2295
liQuellen 1516		\forestOnes 2305	\in 494, 600, 1288, 1625, 1628, 1631, 1657, 1663, 1668, 1980, 1991, 2001, 2011, 2445, 2447, 2453, 2474, 2594
liRelationenSchemaFormat 2224		\forestOv 2294, 2295, 2298	\inhaltsverzeichnis 1431
liRmodell 2204			\input . 4, 7, 10, 13, 16, 394
liUebergangsTabelle 1792			\inputminted 2355, 2365, 2375, 2387, 2390, 2394
\equiv 2653, 2672			\int 2277
			\item 463, 464, 673, 677, 682, 687, 720, 729, 734,

\liEntwurfsFabrikmethodeUml	\liKellerAutomat	102	\liPotenzmenge	1007, 1011, 1953
837	\liKellerKante	140	\liPotenzmengeOhneMathe	1008, 1009, 1010
\liEntwurfsKompositumUml	\liKellerUebergang	135, 141	\liPrimaer	2202
880	\liKontrollCode	1370	\liProblemBeschreibung	1234
\liEntwurfsModellPraesentation	liKontrollflussgraph	(environment) 1358	\liProblemClique	1257
\liEntwurfsZustandAkteure	\liKontrollKnotenPfad	1372	\liProblemName	1233, 1240,
933	\liKontrollTextzeileKnoten	1371, 1376	1252, 1254, 1267,	
\liEntwurfsZustandUml	\liKurzeTabellenLinie	596	1278, 1279, 1287, 1288	
\liEpsilon	\liLadeAllePakete	228	\liProblemSat	1286
\liErAttribute	\liLadePakete	54, 57, 224, 229,	\liProblemSubsetSum	1277, 1286
962, 976, 978	472, 533, 1230,		\liProblemVertexCover	1257, 1265
\liErDatenbankName	1746, 1826, 1947, 2418		\liProduktionen	1048, 1090
\liErEntity	\liLatexCode	2346	liProduktionsRegeln	(environment) 1036
\liErledigt	\liLeereZelle	1784	liProjektSprache	(environment) 1456
\liErMpAttribute	liLernkartei	(environment) 1536	\liPseudoUeberschrift	1441,
\liErMpEntity	\liMasterExkurs	1715	1491, 1492, 1794,	
\liErMpRelationship	\liMasterFaelle	1654, 1722	1804, 2582, 2590, 2598	
\liErRelationship	\liMasterFallRechnung	1700	\liPumpingKontextfrei	1999
961, 970, 972	\liMasterVariablen	1634, 1717	\liPumpingRegulaer	1978
\liExamensAufgabe	\liMasterVariablenDeklaration	1677	liQuellen	(environment) 1516
\liExamensAufgabeA	\liMasterWolframLink	1725	\liRekursionsGleichung	1621, 1681
\liExamensAufgabeTA	liMenge	71, 72, 74,	\liRelation	1871
\liExamensAufgabeTTA	113, 114, 115, 119,		liRelationenSchemaFormat	(environment) 2224
liExkurs	171, 172, 173, 177,		\liRelationMenge	2218
(environment) 1495	998, 1049, 1088, 1089		\liRichtig	463
\liFalsch	\liMengeOhneMathe	998, 1001, 1003	liRmodell	(environment) 2204
\liFlaci	\liMinimierungErklaerung	1806	\liRundeKlammer	1574,
\liFremd	\liMinispracheDatei	2389	1578, 1588, 1598, 1612	
\liFunktionaleAbhaengigkeit	\linespread	2211	\liSetzeAufgabenTitel	25
1859	\liNichtsZuTun	2516	\liSortierMarkierung	2243
\liFunktionaleAbhaengigkeiten	liO	1597, 1625	\liSortierPfeil	2237
1862	liOmega	1587, 1631	\liSortierPfeilUnten	2240
\liFussnote	liOmegaOhneMathe	1587, 1592, 1594	\liSpaltenUmbruch	2312
\liFussnoteDreiText	\liOOhneMathe	1597, 1602, 1604	\liSqlCode	2397
1761, 1779	\liParagraphMitLinien	521, 1808, 2496,	\listen@punkt	1516, 1528
\liFussnoteEinsText	\liPetriErreichKnotenDrei	2517, 2585, 2593, 2601	\liStrich	1232
1753, 1773	1941		\liSyntheseErklaerung	2436, 2501
\liFussnoteLink	\liPetriErreichTransition	1938	\liSyntheseUeberErklaerung	2499
\liFussnoten	\liPetriSetzeSchluessel	1883	\liSyntheseUeberschrift	2420, 2500
\liFussnoteUrl	\liPetriTransitionsName	1930, 1942	\liT	1607, 1622, 1636, 1697
\liFussnoteUrl	2517, 2585, 2593, 2601		\liTeilen	1831
\liFussnoteVierText	\liPetriTransitionsNameOhneMathe	1930, 1933, 1935	liLiteratur	1398, 1422
1765, 1782	\liPetriTransPfeile	1942	\liTheta	1577,
\liFussnoteZweiText	\liPolynomiellReduzierbar	1250	1628, 1657, 1663, 1668	
1757, 1776			\liThetaOhneMathe	1577, 1582, 1584
\liGrammatik				
liGraphenFormat				
(environment) 1186				
\liHanoi				
1193				
\liHaskellCode				
2392				
\liHaskellDatei				
2393				
\liInduktionAnfang				
2581				
\liInduktionErklaerung				
2580				
\liInduktionMarkierung				
2579				
\liInduktionSchritt				
2597				
\liInduktionVoraussetzung				
2589				
\liJavaCode				
2345				
\liJavaDatei				
2354				
\liJavaExamen				
2374				
\liJavaTestDatei				
2364				
liKasten				
(environment) 1131				

<code>\liTOhneMathe</code>	M	<code>\newlength</code>
. . . 1607, 1616, 1618	<code>\makeatletter</code> . . 1532, 2263 548, 1366,
<code>\liTuringKante</code> 203	<code>\makeatother</code> . . 1534, 2271	1371, 2064, 2069,
<code>\liTuringLeerzeichen</code>	<code>\marginpar</code>	2074, 2080, 2086,
. 159, 167	. . . 965, 971, 977, 1439	2092, 2243, 2288, 2566
<code>\liTuringMaschine</code> . . . 160	<code>\mathbb</code> . . 1288, 2011, 2594	<code>\noexpand</code> 2152,
<code>\liTuringUeberfuehrung</code>	<code>\mathbin</code> . . 2193, 2194, 2195	2153, 2154, 2173, 2288
. 206	<code>\mathcal</code> . . 1598, 2474,	<code>\noindent</code> 343, 599, 604,
<code>\liTuringUebergaenge</code>	2479, 2481, 2482, 2483	1039, 1443, 1445,
. 198, 204	<code>\Mathe</code> 2649	1449, 1453, 1481,
<code>\liTuringUebergangZelle</code>	<code>\MatheEnv</code> 2642, 2650, 2653	1509, 1511, 1524,
. 193	<code>\mathord</code> 1740, 1741	1540, 1542, 1550,
<code>\liUeberfuehrungsFunktion</code>	<code>\mdfsetup</code> 1130,	1719, 1772, 1775,
. 1012	1465, 1469, 1473, 1477	1778, 1781, 1864,
<code>\liUeberfuehrungsFunktionOhneMathe</code>	<code>\medskip</code>	1869, 2219, 2518, 2661
. . . 1012, 1015, 1017	1242, 1450, 1452,	<code>\nolinkurl</code> 2351
<code>liUebergangsTabelle</code>	1526, 1554, 2209,	<code>\normalsize</code> 1118
(environment) 1792	2215, 2525, 2643, 2647	<code>\notin</code> 605
<code>\liUeberschriftDreiecksTabelle</code>	<code>\memph</code> 1439	<code>\null</code> 2523
. 1802	<code>\mintinline</code> 2345,	
<code>\liUmlLeserichtung</code> . 2549	2346, 2385, 2392, 2397	O
<code>\liVertauschen</code> 2231	<code>\mkern</code> . . . 2193, 2194, 2195	<code>\o@join</code>
<code>\liWortInSprache</code> 597	<code>\mlq</code> 1738, 1740	2190, 2193, 2194, 2195
<code>\liWortNichtInSprache</code> 602	<code>\mrq</code> 1738, 1741	<code>\Omega</code> 1588
<code>\liWpEquivalent</code> 2652	<code>\msg</code> 39, 391	<code>\omega</code> 1980, 1981, 2001, 2002
<code>\liWpErklaerung</code> 2655	<code>\myList</code>	<code>\or</code> 2101
<code>\liWpErklaerungVerzweigung</code>	2284, 2285, 2286, 2289	
. 2670	<code>\myNodes</code> 2273,	P
<code>\liWpKalkuel</code> 2632	2288, 2294, 2298, 2300	<code>\pagestyle</code> 1141
<code>\liWpKalkuelOhneMathe</code>	N	<code>\par</code> 342, 363, 525, 1070,
. 2632, 2637,	<code>\NeedsTeXFormat</code>	1078, 1448, 1510,
2639, 2671, 2673, 2675	. . . 1, 19, 52, 220,	1533, 1541, 1863,
<code>\liZustandsBuchstabe</code>	398, 459, 466, 530,	1866, 1868, 2143,
. 1022,	593, 608, 952, 988,	2165, 2180, 2221,
1031, 1033, 1053, 1055	1105, 1145, 1154,	2500, 2524, 2660, 2666
<code>\liZustandsBuchstabeGross</code>	1159, 1188, 1226,	<code>\paragraph</code> 1118
. . . 1023, 1032, 1034	1298, 1381, 1396,	<code>\parindent</code> 2212
<code>\liZustandsmenge</code> . . . 1010	1402, 1424, 1570,	<code>\path</code> 94, 141, 204, 563
<code>\liZustandsmengeNr</code> . .	1730, 1743, 1822,	<code>\pgfkeys</code> . . 1894, 2552,
. 1024, 1962	1878, 1944, 1969,	2553, 2554, 2555,
<code>\liZustandsmengeNrGross</code>	1974, 2018, 2184,	2556, 2559, 2562, 2564
. 1032	2197, 2226, 2307,	<code>\pgfmath@count</code>
<code>\liZustandsMengenSammlung</code>	2314, 2319, 2399,	. . . 2265, 2267, 2269
. 1949	2414, 2505, 2509,	<code>\pgfmath@smuggleone</code> 2270
<code>\liZustandsMengenSammlungNr</code>	2529, 2571, 2608, 2628	<code>\pgfmathdeclarefunction</code>
. 1958	<code>\neg</code> 2675 2264
<code>\liZustandsmengeOhneMathe</code>	<code>\negthinspace</code> 1575	<code>\pgfmathhint</code> 2265
. 1010	<code>\newcounter</code> . . . 2038, 2039	<code>\pgfmathparse</code>
<code>\liZustandsname</code> 1033	<code>\NewDocumentCommand</code> 1200, 2282,
<code>\liZustandsnameGross</code>	62, 103, 140, 161,	2287, 2290, 2304, 2305
. . . 1034, 1950, 1959	203, 224, 535, 552,	<code>\pgfmathresult</code>
<code>\liZustandsnameTiefgestellt</code>	597, 602, 1059, 1201, 2265,
. 1051	1081, 1250, 1373,	2266, 2268, 2270,
<code>\liZustandsPaar</code> 1786	1400, 1559, 1563,	2283, 2291, 2304, 2305
<code>\liZustandsPaarVariablenName</code>	1862, 1871, 1938,	<code>\pgfutil@empty</code> 2266
. . . 1785, 1788, 1789	2354, 2364, 2374,	<code>\pgfutil@loop</code> 2267
<code>\llap</code> 2191	2386, 2389, 2393, 2549	<code>\pgfutil@repeat</code> 2270
<code>\log</code> 1625,	<code>\NewDocumentEnvironment</code>	<code>\preceq</code> 1253
1628, 1631, 1657, 1663	1036, 1131, 1186,	<code>\prime</code> 1232
<code>\loop</code> 2024	1358, 1456, 1457,	<code>\printbibliography</code> . 1422
<code>\lor</code> 2674	1460, 1487, 1495,	<code>\ProvidesPackage</code>
<code>\ltimes</code> 1733	1517, 1536, 1546,	. . . 2, 20, 53, 221,
	1793, 1843, 2206, 2224	399, 460, 467, 531,
		594, 609, 953, 989,

1106, 1146, 1155, 1160, 1189, 1227, 1299, 1382, 1397, 1403, 1425, 1571, 1731, 1744, 1823, 1879, 1945, 1970, 1975, 2019, 2185, 2198, 2227, 2308, 2315, 2320, 2400, 2415, 2506, 2510, 2530, 2572, 2609, 2629	1147, 1148, 1157, 1161, 1162, 1163, 1191, 1192, 1231, 1300, 1399, 1404, 1405, 1421, 1428, 1429, 1430, 1458, 1573, 1734, 1735, 1880, 1882, 1972, 2032, 2034, 2186, 2187, 2188, 2201, 2229, 2260, 2311, 2317, 2322, 2333, 2334, 2402, 2507, 2514, 2533, 2534, 2536, 2537, 2575, 2576, 2577, 2610, 2630	\sffamily 476, 1116, 1118, 1220, 2423 \shoveleft 1852 \shoveright 1855 \Sigma 64, 105, 163, 1020, 1021, 1083 \sigma 499, 501, 502 \SLASH 1440 \small 1549 \sort 2277 \sortList 2276, 2285 \square 464 \stepcounter 2064, 2069, 2074, 2077, 2079, 2083, 2085, 2089, 2091 \str .. 477, 486, 1462, 1954, 1963, 2424, 2437 \StrSubstitute . 2284, 2286 \strut 1679, 1683, 1687, 1691, 1695, 2312 \subseteq 2448, 2481, 2488
Q		T
\QS@list 2145, 2156, 2160, 2167, 2173, 2178, 2181	\right 1575 \RIGHTarrow ... 2551, 2556 \Rightarrow 600, 605 \rightarrow 207, 489, 494, 502, 506, 508, 509, 511, 566, 574, 1942, 2445, 2452, 2454, 2457, 2462, 2467, 2468, 2473	\tableofcontents ... 1435 \text 77, 79, 182, 1834, 2580, 2633 \textbf 960, 1258, 1267, 1278, 1287, 1444, 1451, 1482, 1510, 1525, 1541, 1797 \textcolor 1370, 2579 \textit 844, 883, 884, 885, 886, 1551, 1842, 1875 \textsc 1233 \textsf 1444, 1525 \textstyle 1643, 1674 \texttt 983, 1233, 1367, 1368, 1369, 1370, 2633 \thepage 1140, 1389 \theparagraph 1118 \Theta 1578 \thinspace 2633 \tikz 1371 tikz: bbaum 23 tikz: li binaer baum 21 \tikzchildnode 419 \tikzparentnode 419 \tikzset 96, 143, 209, 404, 430, 1165, 1302, 1920, 2046, 2250, 2404, 2611 \tikzumlset 2539 \times 207 \tiny 966, 972, 978, 1370, 1439, 2349 \titleformat .. 1116, 1118 \titlespacing 1117 \tl . 29, 37, 63, 64, 65, 66, 67, 68, 71, 72, 73, 74, 75, 77, 79, 104, 105, 106, 107, 108, 109, 110, 113, 114, 115, 116, 117,
\QS@select@equal 2120, 2124 \QS@select@greater 2121, 2125 \QS@select@smaller 2116, 2119, 2123 \QS@sort@a 2098, 2131, 2152, 2153 \QS@sort@b 2098, 2099 \QS@sort@c 2102, 2109 \QS@sort@d 2110, 2118 \QS@sort@empty . 2100, 2105 \QS@sort@single 2101, 2106 \QS@initialize 2023, 2135, 2170 \QSIr . 2106, 2112, 2120, 2140, 2154, 2159, 2162 \QSIrr 2141, 2154, 2155, 2163 \QSLr 2112, 2119, 2130, 2131, 2139, 2152, 2157, 2161 \QSPivotStep 2025, 2135, 2139, 2150 \QSR 2112 \QSRr 2121, 2142, 2153, 2158, 2164, 2173, 2174, 2175 \QSSortStep 2027, 2135, 2151, 2152 \quad 1750	\right 1575 \RIGHTarrow ... 2551, 2556 \Rightarrow 600, 605 \rightarrow 207, 489, 494, 502, 506, 508, 509, 511, 566, 574, 1942, 2445, 2452, 2454, 2457, 2462, 2467, 2468, 2473 \rightouterjoin 2194 \Roman 1126 \roman 1126, 1128 \romannumeral 2110 \rtimes 1733 \rule 2143, 2165, 2180, 2191	
R	S	
\raisebox 1371 \relax 1434, 2112, 2155, 2157, 2158, 2159, 2267, 2269 \renewcommand ... 1121, 1122, 1123, 1124, 1127, 1128, 1142, 1143, 1392, 1393, 1792 \repeat 2028 \RequirePackage 55, 158, 222, 226, 395, 401, 402, 462, 532, 612, 955, 957, 958, 996, 997, 1107, 1108, 1111, 1113, 1115, 1120, 1129, 1136,	\sb 67, 77, 79, 108, 166, 501, 502, 506, 509, 510, 511, 1053, 1055, 1625, 1628, 1631, 1657, 1663, 1812, 1930, 1939, 2467, 2468, 2469, 2474, 2478, 2479, 2482, 2483, 2486, 2487, 2488 \scriptscriptstyle 566, 574, 582 \scriptsize 1072, 1311, 1318, 1324, 1386, 1387, 1390, 1391, 2580, 2633, 2663 \section 46 \seq . 1375, 1376, 1377, 1519, 1520, 1521, 1528 \setbox 2190 \setcounter 1119, 2144, 2166, 2180 \setganttlinklabel .. 1149, 1150, 1151, 1152 \setlength 2212, 2657, 2658, 2668 \setmainfont 1109 \setmainlanguage 396 \setminted 2339, 2340 \setsansfont 1110 \setul 2203	

