

lehramt-informatik

Hermine Bschlangaul *

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

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

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

3 Index

107

1 Klassen

1.1 Vorlage Theorie-Teil

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

\begin{document}

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

\chapter{Thema des Theorie-Teils}

\literatur

\end{document}
```

1.2 Vorlage Aufgabensammlung

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

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

1.3 Vorlage Aufgabe

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

\end{document}
```

2 Pakete

2.1 aufgaben-einbinden.sty

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

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

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

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

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

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

18
```


2.2 aufgaben-metadaten.sty

```
19 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
20 \ProvidesPackage{lehramt-informatik-aufgaben-metadaten}[2020/07/07 Makros zum
21 Setzen der Aufgaben-Metadaten.]

22 \ExplSyntaxOn

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

\liAufgabenMetadaten Setzen der Aufgaben-Metadaten über eine plist bzw. über key-values.
Die Schlüssel-Werte-Paare sind in der Datei basis.sty definiert. In der Typescript-
Datei .scripts/nodejs/src/aufgaben.ts gibt es ein entsprechendes Interface AufgabenMetadaten.

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

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

50 \ExplSyntaxOff

51
```

2.3 automaten.sty

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

2.3.1 Endlicher Automat

```
54 \liLadePakete{formale-sprachen}

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

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



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

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

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

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

```

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

```

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

```

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

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

```

2.3.2 Kellerautomat

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

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

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

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



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

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

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

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

```

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

```

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

(a, #: A)
(b, #: ε)

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

```

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

```

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

```

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

```

2.3.3 Turingmaschine

```
158 \RequirePackage{amssymb}
```

\liTuringLeerzeichen

□

```
159 \def\liTuringLeerzeichen{\Box}
```

```

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

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

```

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

```

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

```

\liTuringUebergangZelle Formatiert einen Zustandsübergang für eine Übergangstabelle.

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

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

```

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

```

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

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

`(z1: \square , L)`

`(\square : \square , R)`

```

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

\liTuringKante \liTuringKante[\tikz-optionen]{\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. Die einzelnen Schlüssel sind im Interface AufgabenMetadaten in der Typescript-Datei `.scripts/nodejs/src/aufgaben.ts` erklärt.

```
270 \clist_new:N \g_auf_schluesssel_clist
271 \clist_set:Nn \g_auf_schluesssel_clist {
```



```

272  titel,
273  thematik,
274  stichwoerter,
275  zitat_schluessel,
276  zitat_beschreibung,
277  %
278  bearbeitungs_stand,
279  korrektheit,
280  %
281  relativer_pfad,
282  %
283  examen_nummer,
284  examen_jahr,
285  examen_monat,
286  examen_thema_nr,
287  examen_telaufgabe_nr,
288  examen_aufgabe_nr,
289 }

```

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

```

290 \clist_map_inline:Nn \g_auf_schluessel_clist {
291   \tl_new:c {g_auf_#1_tl}
292 }

```

Funktion über alle globalen Token-List-Variablen zurückzusetzen.

```

293 \cs_new:Npn \setze_variablen_zurueck: {
294   \clist_map_inline:Nn \g_auf_schluessel_clist {
295     \tl_clear:c {g_auf_##1_tl}
296   }
297 }

```

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

```

298 \keys_define:nn { aufgabenmetadaten }
299 {
300   Titel .tl_gset:N = \g_auf_titel_tl,
301   Thematik .tl_gset:N = \g_auf_thematik_tl,
302   Stichwoerter .tl_gset:N = \g_auf_stichwoerter_tl,
303   ZitatSchluessel .tl_gset:N = \g_auf_zitat_beschreibung_tl,
304   ZitatBeschreibung .tl_gset:N = \g_auf_zitat_schluessel_tl,
305   %
306   BearbeitungsStand .tl_gset:N = \g_auf_bearbeitungs_stand_tl,
307   Korrektheit .tl_gset:N = \g_auf_korrekttheit_tl,
308   %
309   RelativerPfad .tl_gset:N = \g_auf_relativer_pfad_tl,
310   %
311   ExamenNummer .tl_gset:N = \g_auf_examen_nummer_tl,
312   ExamenJahr .tl_gset:N = \g_auf_examen_jahr_tl,
313   ExamenMonat .tl_gset:N = \g_auf_examen_monat_tl,
314   ExamenThemaNr .tl_gset:N = \g_auf_examen_thema_nr_tl,
315   ExamenTeilaufgabeNr .tl_gset:N = \g_auf_examen_telaufgabe_nr_tl,
316   ExamenAufgabeNr .tl_gset:N = \g_auf_examen_aufgabe_nr_tl,
317 }

318 \cs_gset:Npn \setze_relativen_pfad: {
319   \tl_if_empty:NTF \g_auf_relativer_pfad_tl
320   {
321     \bool_if:nTF
322     {
323       ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
324       ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
325       ! \tl_if_empty_p:N \g_auf_examen_monat_tl
326     }
327     {

```

```

328 \tl_gset:Nn \g_auf_relativer_pfad_tl {
329   Staatsexamen /
330   \g_auf_examen_nummer_tl /
331   \g_auf_examen_jahr_tl /
332   \g_auf_examen_monat_tl /
333   \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} { Thema - \g_auf_examen_thema_nr_tl / }
334   \tl_if_empty:NTF \g_auf_examen_telaufgabe_nr_tl {} { Teilaufgabe - \g_auf_examen_te
335   \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} { Aufgabe - \g_auf_examen_aufgabe_nr
336 }
337 }
338 {}
339 }
340 {}
341 }

342 \cs_gset:Npn \_gib_examen_titel: {
343   \cs_set:Nn \_trenner: { \, / \, }
344   \bool_if:nTF
345   {
346     ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
347     ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
348     ! \tl_if_empty_p:N \g_auf_examen_monat_tl &&
349     ! \tl_if_empty_p:N \g_auf_examen_aufgabe_nr_tl
350   }
351   {
352     {
353       \footnotesize
354       \par
355       \noindent
356       Staatsexamen ~
357       \g_auf_examen_nummer_tl \_trenner:
358       \g_auf_examen_jahr_tl \_trenner:
359
360       \tl_case:Nn \g_auf_examen_monat_tl
361       {
362         { 03 } { Frühjahr }
363         { 09 } { Herbst }
364       } \_trenner:
365
366       \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} {
367         Thema ~ Nr. ~ \g_auf_examen_thema_nr_tl \_trenner:
368       }
369       \tl_if_empty:NTF \g_auf_examen_telaufgabe_nr_tl {} {
370         Teilaufgabe ~ Nr. ~ \g_auf_examen_telaufgabe_nr_tl \_trenner:
371       }
372       \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} {
373         Aufgabe ~ Nr. ~ \g_auf_examen_aufgabe_nr_tl
374       }
375       \par
376       \bigskip
377     }
378   }
379 }

380 \cs_new:Npn \_gib_github_url: {
381   \tl_if_empty:NTF \g_auf_relativer_pfad_tl {} {
382     Github :~\href{
383       \LehramtInformatikGithubRawDomain /
384       \LehramtInformatikGithubTexRepo /
385       \LehramtInformatikGitBranch /
386       \g_auf_relativer_pfad_tl
387     }{
388       \g_auf_relativer_pfad_tl
389     }
390   }

```

```

391 }
392 \cs_new:Npn \_gib_aufgaben_titel: {
393   \g_auf_titel_tl
394
395   \tl_if_empty:NTF \g_auf_thematik_tl
396   {}
397   {
398     \, ~ [
399       \g_auf_thematik_tl
400     ]
401   }
402 }
403 \msg_new:nnn { aufgabenmetadaten } { kein-titel }
404   { Der~Schlüssel~ist~zwingend~notwendig }
405 \ExplSyntaxOff
406 \def\LehramtInformatikRepository{/pfad/zum/repository}
407 \input /etc/lehramt-informatik.config.tex
408 biblatex not working with lualatex and babel
409 % \RequirePackage{polyglossia}
410 % \setmainlanguage{german}

```

2.5 baum.sty

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

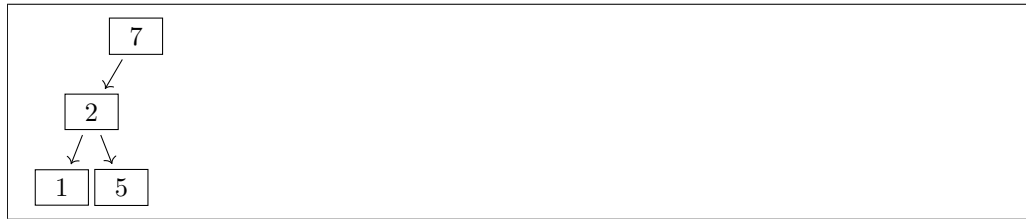
    für li binaer baum
414 \RequirePackage{tikz-qtrees}

    Für b baum
415 \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}
```



```
416 \tikzset{
417   li binaer baum/.style={
418     shorten <=2pt,
419     shorten >=2pt,
420     ->,
421     every tree node/.style={
422       minimum width=2em,
423       draw,
424       rectangle
425     },
426     blank/.style={
427       draw=none
428     },
429     edge from parent/.style={
430       draw,
431       edge from parent path={(\tikzparentnode) -- (\tikzchildnode)}
432     },
433     level distance=1cm,
434     every label/.style={
435       gray,
436       font=\footnotesize,
437       label position=0,
438       label distance=0cm,
439     }
440   },
441 }
```

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}

```



```

442 \tikzset{
443   li bbaum knoten/.style={
444     rectangle split parts=10,
445     rectangle split,
446     rectangle split horizontal,
447     rectangle split ignore empty parts,
448     draw,
449     fill=white
450   },
451   li bbaum/.style={
452     every node/.style={
453       li bbaum knoten
454     },
455     level 1/.style={
456       level distance=12mm,
457       sibling distance=25mm,
458     },
459     every child/.style={
460       shorten <= 2pt,
461       shorten >= 6pt,
462       ->,
463     },
464     level 2/.style={
465       level distance=9mm,
466       sibling distance=15mm,
467     },
468   }
469 }
470

```

2.6 checkbox.sty

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

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

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

477
```


2.7 chomsky-normalform.sty

```

478 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
479 \ProvidesPackage{lehramt-informatik-chomsky-normalform}[2021/03/26
480 Hilfsmakros zum Setzen des Algorithmus zum Erreichen der Chomsky-Normalform]
481
482 \ExplSyntaxOn
483
484 \liLadePakete{typographie}

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

485 \def\liChomskyUeberschrift#1{
486   {
487     \bfseries
488     \sffamily
489     \str_case:nn {#1} {
490       {1} {Elimination~der~ $\varepsilon$ -Regeln}
491       {2} {Elimination~von~Kettenregeln}
492       {3} {Separation~von~Terminalzeichen}
493       {4} {Elimination~von~mehrelementigen~Nonterminalketten}
494     }
495   }
496 }

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

497 \def\li@chomsky@erklaerung@texte#1{
498   \str_case:nn {#1} {
499     %
500     {1} {
501       Alle~Regeln~der~Form~ $A \rightarrow \varepsilon$ ~werden~eliminiert.~
502       Die~Ersetzung~von~ $A$ ~wird~durch~ $\varepsilon$ ~in~allen~anderen~
503       Regeln~vorweggenommen.
504     }
505     {2} {
506       Jede~Produktion~der~Form~ $A \rightarrow B$ ~mit~ $A, B \in S$ ~wird~
507       als~Kettenregel~bezeichnet.~Diese~tragen~nicht~zur~Produktion~
508       von~Terminalzeichen~bei~und~lassen~sich~ebenfalls~eliminieren.
509     }
510     {3} {
511       Jedes~Terminalzeichen~ $\sigma$ ,~das~in~Kombination~mit~anderen~
512       Symbolen~auftaucht,~wird~durch~ein~neues~Nonterminal~
513        $S_{\sigma}$ ~ersetzt~und~die~Menge~der~Produktionen~durch~die~
514       Regel~ $S_{\sigma} \rightarrow \sigma$ ~ergänzt.
515     }
516     {4} {
517       Alle~Produktionen~der~Form~
518        $A \rightarrow B_{\{1\}} B_{\{2\}} \dots B_{\{n\}}$ ~
519       werden~in~die~Produktionen~
520        $A \rightarrow$ 
521        $A_{\{n-1\}} B_{\{n\}}, A_{\{n-1\}} \rightarrow$ 
522        $A_{\{n-2\}} B_{\{n-1\}}, \dots,$ 
523        $A_{\{2\}} \rightarrow B_{\{1\}} B_{\{2\}}$ ~zerteilt.~
524       Nach~der~Ersetzung~sind~alle~längeren~Nonterminalketten~
525       vollständig~heruntergebrochen~und~die~Chomsky-Normalform~erreicht.
526     }
527   }
528 }

529 \def\liChomskyErklaerung#1{
530   {
531     \itshape
532     \footnotesize
533     \liParagraphMitLinien{\li@chomsky@erklaerung@texte{#1}}
534   }

```

535 }

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

536 \def\liChomskyUeberErklaerung#1{

537 \liChomskyUeberschrift{#1}\par

538 \liChomskyErklaerung{#1}

539 }

540 \ExplSyntaxOff

541

2.8 cpm.sty

```

542 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
543 \ProvidesPackage{lehramt-informatik-cpm}[2020/09/03]
544 \RequirePackage{tikz}
545 \liLadePakete{mathe}

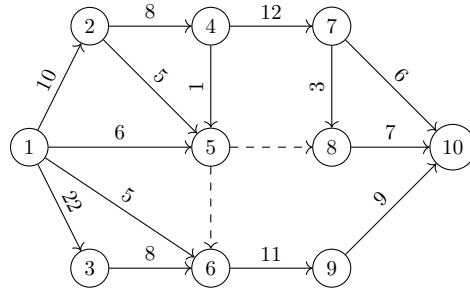
```

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}

```

```

546 \ExplSyntaxOn
547 \NewDocumentCommand { \liCpmEreignis } { 0{} m m m } {
548   \tl_set:Nn \l_name_tl {}
549
550   \keys_define:nn { cpmEreignis } {
551     name .code:n = {\tl_set:Nn \l_name_tl {##1}},
552   }
553
554   \keys_set:nn { cpmEreignis } { #1 }
555
556   \tl_if_empty:NT \l_name_tl {
557     \tl_set:Nn \l_name_tl {#2}
558   }

```

```

559
560 \node[circle,draw] (\l_name_tl) at (#3,#4) {#2};
561 }
562 \ExplSyntaxOff

\liCpmVorgang liCpmVORGANG\((.*)>(.*))\{(.*)\}
563 \ExplSyntaxOn
564 \NewDocumentCommand { \liCpmVorgang } { 0{ } m m m } {
565   \tl_set:Nn \l_schein_tl {}
566   \tl_set:Nn \l_kritisch_tl {}
567
568   \keys_define:nn { cpmVorgang } {
569     schein .code:n = {\tl_set:Nn \l_schein_tl {dashed}},
570     kritisch .code:n = {\tl_set:Nn \l_kritisch_tl {thick}},
571   }
572
573   \keys_set:nn { cpmVorgang } { #1 }
574
575   \path[->,\l_schein_tl,\l_kritisch_tl] (#2) edge node[auto,sloped] {#4} (#3);
576 }
577 \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):  $1_{(2 \rightarrow 3)}$ 
578 \def\liCpmVonZuOhneMathe#1(#2-#3){#1_{\scriptscriptstyle(#2\rightarrow#3)}}
579 \def\liCpmVonZu#1(#2-#3){%
580   \ifmmode%
581     \liCpmVonZuOhneMathe{#1}(#2-#3)%
582   \else%
583     $\liCpmVonZuOhneMathe{#1}(#2-#3)$%
584   \fi%
585 }

```

```

\liCpmVon Let-Abkürzung: \let\v=\liCpmVon
\liCpmVon{1}(2):  $1_{(\rightarrow 2)}$ 
586 \def\liCpmVonOhneMathe#1(#2){#1_{\scriptscriptstyle(\rightarrow#2)}}
587 \def\liCpmVon#1(#2){%
588   \ifmmode%
589     \liCpmVonOhneMathe{#1}(#2)%
590   \else%
591     $\liCpmVonOhneMathe{#1}(#2)$%
592   \fi%
593 }

```

```

\liCpmZu Let-Abkürzung: \let\z=\liCpmZu
\liCpmZu{1}(2):  $1_{(\leftarrow 2)}$ 
594 \def\liCpmZuOhneMathe#1(#2){#1_{\scriptscriptstyle(\leftarrow#2)}}
595 \def\liCpmZu#1(#2){%
596   \ifmmode%
597     \liCpmZuOhneMathe{#1}(#2)%
598   \else%
599     $\liCpmZuOhneMathe{#1}(#2)$%

```

```

600 \fi%
601 }

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

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

604

```

2.9 cyk-algorithmus.sty

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

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

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

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

```
\liWortInSprache  $\Rightarrow abc \in L(Y)$ 
609 \NewDocumentCommand{ \liWortInSprache } { m O{L(G)} } {
610   \bigskip
611   \noindent
612   $\Rrightarrow$ #1 \in #2$
613 }
```

```
\liWortNichtInSprache  $\Rightarrow abc \notin L(G)$ 
614 \NewDocumentCommand{ \liWortNichtInSprache } { m O{L(G)} } {
615   \bigskip
616   \noindent
617   $\Rrightarrow$ #1 \notin #2$
618 }

619
```

2.10 entwurfsmuster.sty

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

2.10.1 Namensschema der Entwurfsmuster-Makros:

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

2.10.2 Reihenfolge

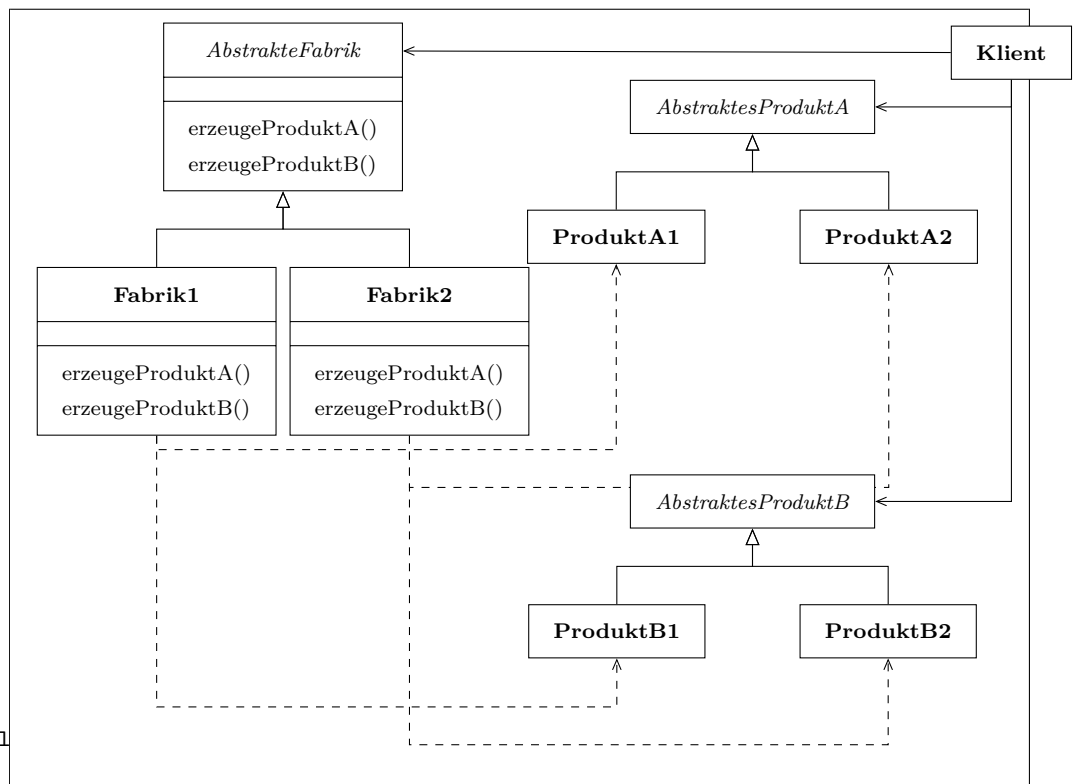
1. Uml: Uml-Klassendiagramm \liEntwurfsEinzelstueckUml
2. Akteure: Akteure, beteiligte Klassen \liEntwurfsEinzelstueckAkteure
3. Code: Allgemeines Code-Beispiel \liEntwurfsEinzelstueckCode
4. ohne: Ohne Suffix, Bündelung der einzelnen Makros eines Entwurfsmusters \liEntwurfsEinzelstueckAkteure

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

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

```
624 \RequirePackage{lehramt-informatik-syntax}
625 \def\liEntwurfsCode#1#2{
626   \liJavaDatei{entwurfsmuster/#1/allgemein/#2}
627 }
```

2.10.3 Abstrakte Fabrik (Abstract Factory)



```
628 \def\liEntwurfsAbstrakteFabrikUml{
629   \begin{tikzpicture}
630     \umlclass[type=abstract]{AbstrakteFabrik}{-}{-}{
631       erzeugeProduktA()\
632       erzeugeProduktB()\
633     }
634   }
```

```

634 \umlclass[below left=1cm and -1.5cm of AbstrakteFabrik]{Fabrik1}{-}{
635     erzeugeProduktA()\
636     erzeugeProduktB()\
637 }
638 \umlclass[below right=1cm and -1.5cm of AbstrakteFabrik]{Fabrik2}{-}{
639     erzeugeProduktA()\
640     erzeugeProduktB()\
641 }
642 \umlVHVinherit{Fabrik1}{AbstrakteFabrik}
643 \umlVHVinherit{Fabrik2}{AbstrakteFabrik}
644
645 \umlsimpleclass[right=3cm of AbstrakteFabrik,type=abstract]{AbstraktesProduktA}
646 \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktA]{ProduktA1}
647 \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktA]{ProduktA2}
648 \umlVHVinherit{ProduktA1}{AbstraktesProduktA}
649 \umlVHVinherit{ProduktA2}{AbstraktesProduktA}
650
651 \umlsimpleclass[above right=0cm and 1cm of AbstraktesProduktA]{Klient}
652
653 \umlsimpleclass[below=4.5cm of AbstraktesProduktA,type=abstract]{AbstraktesProduktB}
654 \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktB]{ProduktB1}
655 \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktB]{ProduktB2}
656 \umlVHVinherit{ProduktB1}{AbstraktesProduktB}
657 \umlVHVinherit{ProduktB2}{AbstraktesProduktB}
658
659 \umlVHVdep[arm1=-1.3cm,arm2=-1cm]{Fabrik1}{ProduktA1}
660 \umlVHVdep[arm2=-1cm]{Fabrik1}{ProduktB1}
661
662 \umlVHVdep[arm1=-1.8cm,arm2=-1.5cm]{Fabrik2}{ProduktA2}
663 \umlVHVdep[arm2=-1.5cm]{Fabrik2}{ProduktB2}
664
665 \umluniassoc[anchor2=25]{Klient}{AbstrakteFabrik}
666 \umlVHuniassoc[arm1=-1cm]{Klient}{AbstraktesProduktA}
667 \umlVHuniassoc{Klient}{AbstraktesProduktB}
668 \end{tikzpicture}
669 }

```

iEntwurfsAbstrakteFabrikCode

```

670 \def\liEntwurfsAbstrakteFabrikCode{
671     \li@EntwurfsCode{abstrakte_fabrik}{Produkte}
672     \li@EntwurfsCode{abstrakte_fabrik}{AbstrakteFabrik}
673     \li@EntwurfsCode{abstrakte_fabrik}{Klient}
674 }

```

\liEntwurfsAbstrakteFabrik

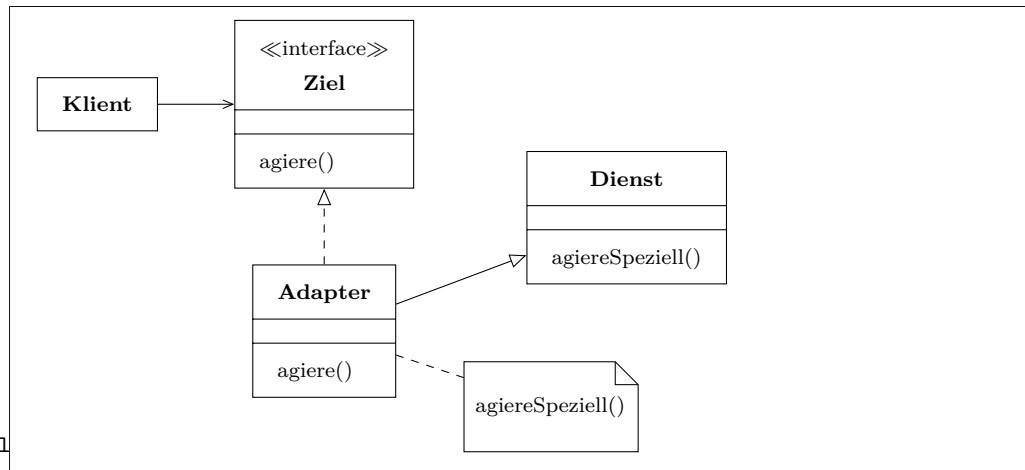
```

675 \def\liEntwurfsAbstrakteFabrik{
676     \liEntwurfsAbstrakteFabrikUml
677     \liEntwurfsAbstrakteFabrikCode
678 }

```

2.10.4 Adapter

\liEntwurfsAdapterUml



```

679 \def\liEntwurfsAdapterUml{
680   \begin{tikzpicture}
681     \umlsimpleclass[x=1,y=3]{Klient}{}{}
682     \umlclass[x=4,y=3,type=interface]{Ziel}{}{agiere()}
683     \umlclass[x=4,y=0]{Adapter}{}{agiere()}
684     \umlclass[x=8,y=1.5]{Dienst}{}{agiereSpeziell()}
685
686     \umlreal{Adapter}{Ziel}
687     \umluniassoc{Klient}{Ziel}
688     \umlinherit{Adapter}{Dienst}
689
690     \umlnote[x=7,y=-1,width=2cm]{Adapter}{agiereSpeziell()}
691   \end{tikzpicture}
692   \footcite[so ähnlich wie GoF]{wiki:adapter}
693 }

```

\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.

```

694 \def\liEntwurfsAdapterAkteure{
695   \begin{description}
696
697     \item[Ziel (Target)]
698
699     Das Ziel definiert die Schnittstelle, die der Klient nutzen kann.
700
701     \item[Klient (Client)]
702
703     Der Klient nutzt Dienste über inkompatible Schnittstellen und greift
704     dabei auf adaptierte Schnittstellen zurück.
705
706     \item[Dienst (Adaptee)]
707
708     Der Dienst bietet wiederzuverwendende Dienstleistungen mit fest
709     definierter Schnittstelle an.
710
711     \item[Adapter]
712
713     Der Adapter adaptiert die Schnittstelle des Dienstes auf die
714     Schnittstelle zum Klienten.\footcite{wiki:adapter}

```

```

715
716 \end{description}
717 }

```

\liEntwurfsAdapterCode

```

718 \def\liEntwurfsAdapterCode{
719 \li@EntwurfsCode{adapter}{Dienst}
720 \li@EntwurfsCode{adapter}{Ziel}
721 \li@EntwurfsCode{adapter}{Adapter}
722 \li@EntwurfsCode{adapter}{Klient}
723 }

```

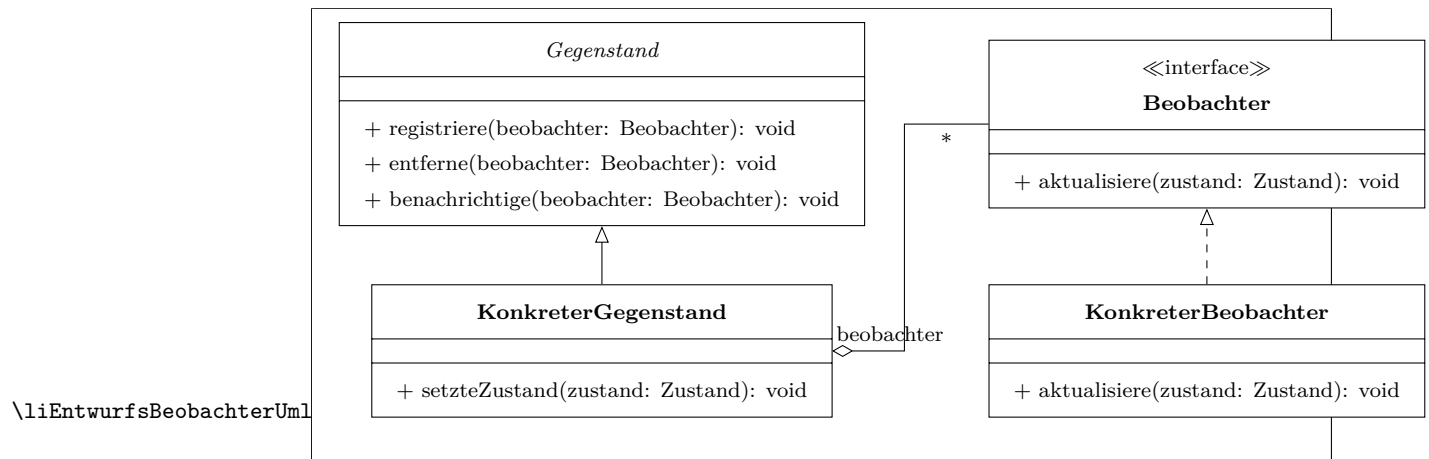
\liEntwurfsAdapter

```

724 \def\liEntwurfsAdapter{
725 \liEntwurfsAdapterUml
726 \liEntwurfsAdapterAkteure
727 \liEntwurfsAdapterCode
728 }

```

2.10.5 Beobachter (Observer)



```

729 \def\liEntwurfsBeobachterUml{
730 \begin{tikzpicture}
731 \umlclass[x=0,y=0,type=abstract]{Gegenstand}{-}{
732 + registriere(beobachter: Beobachter): void\\
733 + entferne(beobachter: Beobachter): void\\
734 + benachrichtige(beobachter: Beobachter): void\\
735 }
736 \umlclass[x=0,y=-3]{KonkreterGegenstand}{-}{
737 + setzteZustand(zustand: Zustand): void
738 }
739 \umlinherit{KonkreterGegenstand}{Gegenstand}
740
741 \umlclass[x=8,y=0,type=interface]{Beobachter}{-}{
742 + aktualisiere(zustand: Zustand): void
743 }
744 \umlclass[x=8,y=-3]{KonkreterBeobachter}{-}{
745 + aktualisiere(zustand: Zustand): void
746 }
747 \umlreal{KonkreterBeobachter}{Beobachter}
748
749 \umlHVHaggreg[arg1=beobachter,pos1=0.8,mult2=*,pos2=2.5]
750 {KonkreterGegenstand}{Beobachter}
751 \end{tikzpicture}
752 }

```

\liEntwurfsBeobachterAkteure

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

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

konkreter/s Gegenstand / Subjekt (ConcreteSubject / ConcreteObservable)

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

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

```

753 \def\liEntwurfsBeobachterAkteure{
754   \begin{description}
755     \item[Gegenstand / Subjekt (Subject / Observable)]
756
757     Ein Subjekt (beobachtbares Objekt, auf Englisch publisher, also
758     „Veröffentlicher“, genannt) hat eine Liste von Beobachtern, ohne
759     deren konkrete Typen zu kennen. Es bietet eine Schnittstelle zur An-
760     und Abmeldung von Beobachtern und eine Schnittstelle zur
761     Benachrichtigung von Beobachtern über Änderungen an.\footcite[Seite
762     251]{gof}
763
764     \item[Beobachter (Observer)]
765
766     Die Beobachter (auf Englisch auch subscriber, also „Abonnent“,
767     genannt) definieren eine Aktualisierungsschnittstelle.
768
769     \item[konkreter/s Gegenstand / Subjekt (ConcreteSubject / ConcreteObservable)]
770
771     Ein konkretes Subjekt (konkretes, beobachtbares Objekt) speichert
772     den relevanten Zustand und benachrichtigt alle Beobachter bei
773     Zustandsänderungen über deren Aktualisierungsschnittstelle. Es
774     verfügt über eine Schnittstelle zur Erfragung des aktuellen
775     Zustands.
776
777     \item[Konkrete Beobachter (ConcreteObserver)]
778
779     Konkrete Beobachter verwalten die Referenz auf ein konkretes
780     Subjekt, dessen Zustand sie beobachten und speichern und dessen
781     Zustand konsistent ist. Sie implementieren eine
782     Aktualisierungsschnittstelle unter Verwendung der
783     Abfrageschnittstelle des konkreten Subjekts.
784     \footcite{wiki:beobachter}
785   \end{description}
786 }
```

\liEntwurfsBeobachterCode

```

787 \def\liEntwurfsBeobachterCode{
788   \li@EntwurfsCode{beobachter}{Gegenstand}
789   \li@EntwurfsCode{beobachter}{KonkreterGegenstand}
790   \li@EntwurfsCode{beobachter}{Beobachter}
791   \li@EntwurfsCode{beobachter}{KonkreterBeobachterA}
792   \li@EntwurfsCode{beobachter}{KonkreterBeobachterB}
```

```

793 \li@EntwurfsCode{beobachter}{Klient}
794 }

```

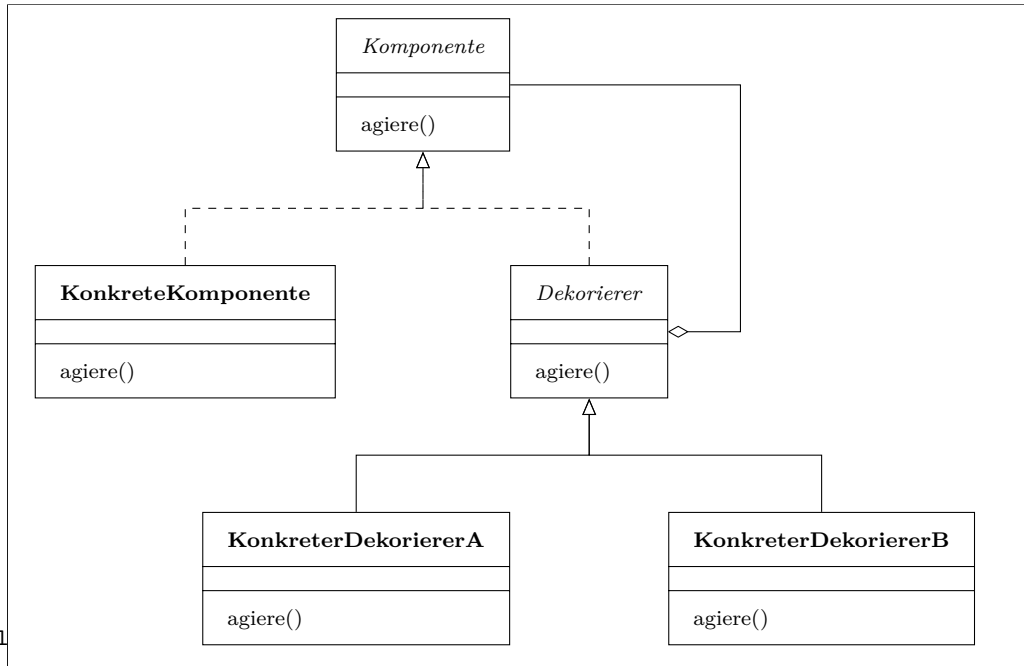
\liEntwurfsBeobachter

```

795 \def\liEntwurfsBeobachter{
796 \liEntwurfsBeobachterUml
797 \liEntwurfsBeobachterAkteure
798 \liEntwurfsBeobachterCode
799 }

```

2.10.6 Dekorierer (Decorator)



\liEntwurfsDekoriererUml

```

800 \def\liEntwurfsDekoriererUml{
801 \begin{tikzpicture}
802 \umlclass[type=abstract]{Komponente}{-}{agiere()}
803 \umlclass[below left=1.5cm and 0cm of Komponente]{KonkreteKomponente}{-}{agiere()}
804 \umlclass[below right=1.5cm and 0cm of Komponente,type=abstract]{Dekorierer}{-}{agiere()}
805
806 \umlVHVreal{KonkreteKomponente}{Komponente}
807 \umlVHVreal{Dekorierer}{Komponente}
808
809 \umlclass[below left=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererA}{-}{agiere()}
810 \umlclass[below right=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererB}{-}{agiere()}
811
812 \umlVHVinherit{KonkreterDekoriererA}{Dekorierer}
813 \umlVHVinherit{KonkreterDekoriererB}{Dekorierer}
814
815 \umlHVVHaggreg[arm1=2cm]{Dekorierer}{Komponente}
816 \footcite{wiki:dekorierer}
817 \end{tikzpicture}
818 }

```

\liEntwurfsDekoriererCode

```

819 \def\liEntwurfsDekoriererCode{
820 \li@EntwurfsCode{dekorierer}{Komponente}
821 \li@EntwurfsCode{dekorierer}{KonkreteKomponente}
822 \li@EntwurfsCode{dekorierer}{Dekorierer}
823 \li@EntwurfsCode{dekorierer}{KonkreterDekoriererA}
824 \li@EntwurfsCode{dekorierer}{KonkreterDekoriererB}
825 \li@EntwurfsCode{dekorierer}{Klient}
826 }

```

\liEntwurfsDekorierer

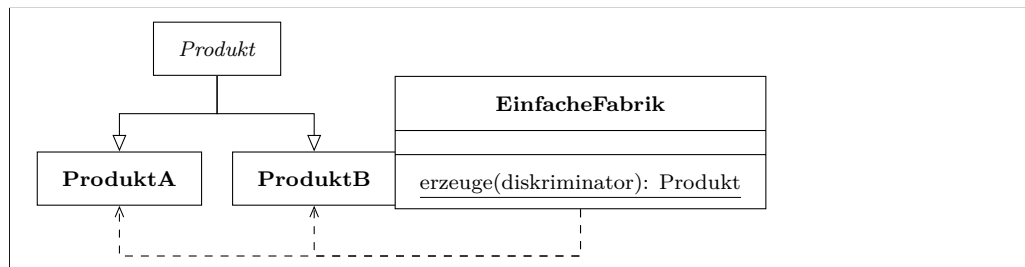
```

827 \def\liEntwurfsDekorierer{
828   \liEntwurfsDekoriererUml
829   \liEntwurfsDekoriererAkteure
830   \liEntwurfsDekoriererCode
831 }

```

2.10.7 Einfache Fabrik (Simple Factory)

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



```

832 \def\liEntwurfsEinfacheFabrikUml{
833   \begin{tikzpicture}
834     \umlsimpleclass[type=abstract]{Produkt}
835     \umlsimpleclass[below left=1cm and -0.65cm of Produkt]{ProduktA}
836     \umlsimpleclass[below right=1cm and -0.65cm of Produkt]{ProduktB}
837     \umlVHVinherit{Produkt}{ProduktA}
838     \umlVHVinherit{Produkt}{ProduktB}
839     \umlclass[below right=0cm and 1.5cm of Produkt]{EinfacheFabrik}{
840     }{
841       \umlstatic{erzeuge(diskriminator): Produkt}\\
842     }
843     \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktA}
844     \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktB}
845   \end{tikzpicture}
846 }

```

ntwurfsEinfacheFabrikAkteure Quelle: frei überstetzt aus GoF

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

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

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

```

847 \def\liEntwurfsEinfacheFabrikAkteure{
848   \begin{description}
849     \item[EinfacheFabrik]
850
851     Eine Klasse mit einer Erzeugungsmethode, die über eine größere
852     Bedingung verschiedene Objekt instanziiert.
853
854     \item[Produkt]
855
856     Eine abstrakte Klasse, die von den konkreten Produkten geerbt wird.
857
858     \item[KonkretesProdukt]
859
860     Ein konkretes Produkt, das von der einfachen Fabrik erzeugt wird.
861   \end{description}
862 }

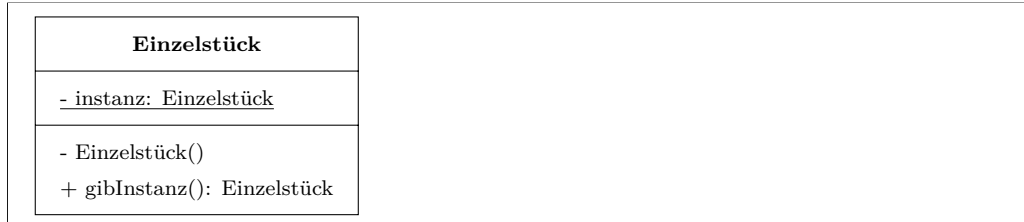
```

\liEntwurfsEinfacheFabrik

```
863 \def\liEntwurfsEinfacheFabrik{
864   \liEntwurfsEinfacheFabrikUml
865   \liEntwurfsEinfacheFabrikAkteure
866 }
```

2.10.8 Einzelstück (Singleton)

\liEntwurfsEinzelstueckUml Quelle nach der deutschen Wikipedia



```
867 \def\liEntwurfsEinzelstueckUml{
868   \begin{tikzpicture}
869     \umlclass{Einzelstück}{
870       \umlstatic{- instanz: Einzelstück}\\
871     }{
872       - Einzelstück()\\
873       + gibInstanz(): Einzelstück
874     }
875   \end{tikzpicture}
876 }
```

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.

```
877 \def\liEntwurfsEinzelstueckAkteure{
878   \begin{description}
879     \item[Einzelstück (Singleton)]
880
881     stellt eine statische Methode bereit, mit deren Hilfe die Klienten
882     nur auf eine einzige Instanz der Klasse zugreifen können.
883   \end{description}
884 }
```

\liEntwurfsEinzelstueckCode

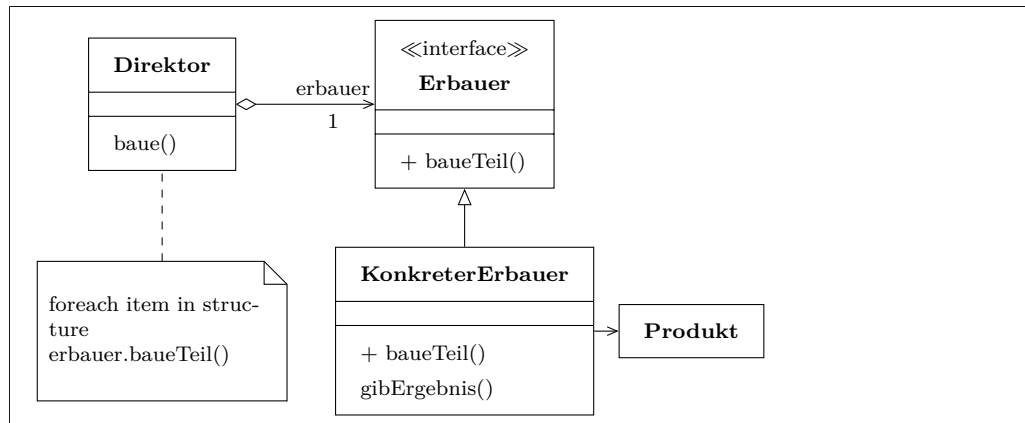
```
885 \def\liEntwurfsEinzelstueckCode{
886   \li@EntwurfsCode{einzelstueck}{Einzelstueck}
887 }
```

\liEntwurfsEinzelstueck

```
888 \def\liEntwurfsEinzelstueck{
889   \liEntwurfsEinzelstueckUml
890   \liEntwurfsEinzelstueckAkteure
891   \liEntwurfsEinzelstueckCode
892 }
```

2.10.9 Erbauer (Builder)

\liEntwurfsErbauerUml Quelle nach der deutschen Wikipedia



```

893 \def\liEntwurfsErbauerUml{
894   \begin{tikzpicture}
895     \umlsimpleclass[x=7,y=0]{Produkt}{}{}
896     \umlclass[x=4,y=3,type=interface]{Erbauer}{}{+ baueTeil()}
897     \umlclass[x=4,y=0]{KonkreterErbauer}{}{
898       + baueTeil()\
899       gibErgebnis()}
900     \umlclass[x=0,y=3]{Direktor}{}{baue()}
901
902     \umluniagg[arg2=erbauer,pos2=0.7,mult2=1]{Direktor}{Erbauer}
903     \umluniassoc{KonkreterErbauer}{Produkt}
904     \umlinherit{KonkreterErbauer}{Erbauer}
905
906     \umlnote[x=0,y=0,width=3cm]{Direktor}{
907       foreach item in structure\
908       erbauer.baueTeil()
909     }
910   \end{tikzpicture}
911   \footcite{wiki:erbauer}
912 }

```

\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.

```

913 \def\liEntwurfsErbauerAkteure{
914   \begin{description}
915     \item[Erbauer]
916
917     Der Erbauer spezifiziert eine abstrakte Schnittstelle zur Erzeugung der
918     Teile eines komplexen Objektes.
919
920     \item[KonkreterErbauer]
921
922     Der konkrete Erbauer erzeugt die Teile des komplexen Objekts durch
923     Implementierung der Schnittstelle. Außerdem definiert und verwaltet er

```

```

924     die von ihm erzeugte Repräsentation des Produkts. Er bietet auch eine
925     Schnittstelle zum Auslesen des Produkts.
926
927     \item[Direktor]
928
929     Der Direktor konstruiert ein komplexes Objekt unter Verwendung der
930     Schnittstelle des Erbauers. Der Direktor arbeitet eng mit dem Erbauer
931     zusammen: Er weiß, welche Baureihenfolge der Erbauer verträgt oder
932     benötigt. Der Direktor entkoppelt somit den Konstruktionsablauf vom
933     Klienten.
934
935     \item[Produkt]
936
937     Das Produkt repräsentiert das zu konstruierende komplexe Objekt.
938     \footcite{wiki:erbauer}
939 \end{description}
940 }

```

\liEntwurfsErbauer

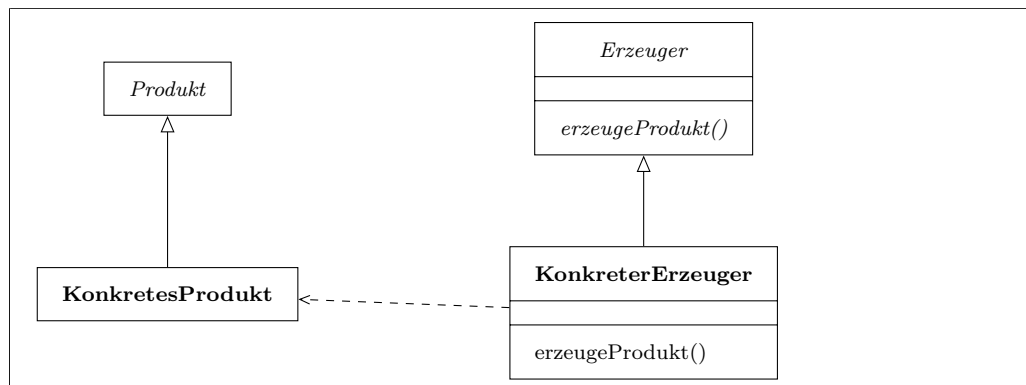
```

941 \def\liEntwurfsErbauer{
942   \liEntwurfsErbauerUml
943   \liEntwurfsErbauerAkteure
944 }

```

2.10.10 Fabrikmethode (Factory Method)

\liEntwurfsFabrikmethodeUml Quelle nach der deutschen Wikipedia



```

945 \def\liEntwurfsFabrikmethodeUml{
946   \begin{tikzpicture}
947     \umlsimpleclass[type=abstract]{Produkt}
948     \umlsimpleclass[below=2cm of Produkt]{KonkretesProdukt}
949     \umlinherit{KonkretesProdukt}{Produkt}
950
951     \umlclass[type=abstract,right=4cm of Produkt]{Erzeuger}{\{
952       \textit{erzeugeProdukt()}\}
953   }
954     \umlclass[below=1.2cm of Erzeuger]{KonkreterErzeuger}{\{
955       erzeugeProdukt()
956   }
957     \umlinherit{KonkreterErzeuger}{Erzeuger}
958
959     \umldep{KonkreterErzeuger}{KonkretesProdukt}
960   \end{tikzpicture}
961 }

```

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).

```

962 \def\liEntwurfsFabrikmethodeAkteure{
963   \begin{description}
964     \item[Produkt]
965
966     Das Produkt ist der Basistyp (Klasse oder Schnittstelle) für das
967     zu erzeugende Produkt.
968
969     \item[KonkretesProdukt]
970
971     KonkretesProdukt implementiert die Produkt-Schnittstelle.
972
973     \item[Erzeuger]
974
975     Der Erzeuger deklariert die Fabrikmethode, um ein solches Produkt
976     zu erzeugen und kann eine Default-Implementierung beinhalten.
977
978     \item[KonkreterErzeuger]
979
980     KonkreterErzeuger überschreibt die Fabrikmethode, um die ihm
981     entsprechenden konkreten Produkte zu erzeugen (z. B. indem er den
982     Konstruktor einer konkreten Produkt-Klasse aufruft).
983
984     \footcite{wiki:fabrikmethode}
985   \end{description}
986 }

```

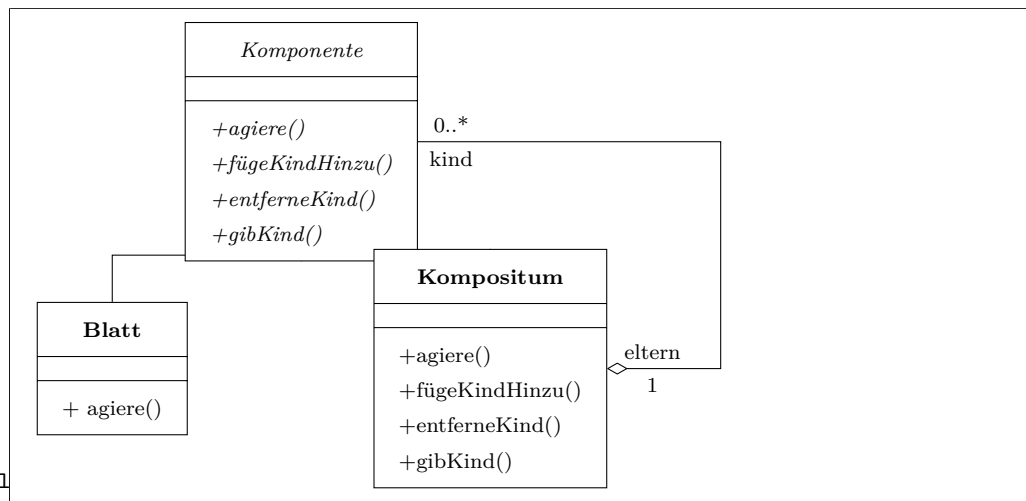
\liEntwurfsFabrikmethode

```

987 \def\liEntwurfsFabrikmethode{
988   \liEntwurfsFabrikmethodeUml
989   \liEntwurfsFabrikmethodeAkteure
990 }

```

2.10.11 Kompositum (Composite)



\liEntwurfsKompositumUml

```

991 \def\liEntwurfsKompositumUml{
992   \begin{tikzpicture}
993     \umlclass[x=2.5,y=3,type=abstract]{Komponente}{+agiere()}\
994     \textit{+fügeKindHinzu()}\
995     \textit{+entferneKind()}\
996     \textit{+gibKind()}
997   }
998 }
999 \umlclass[x=0]{Blatt}{+ agiere()}
1000 \umlclass[x=5]{Kompositum}{+
1001   +agiere()\\
1002   +fügeKindHinzu()\\
1003   +entferneKind()\\
1004   +gibKind()
1005 }
1006
1007 \umlVHVinherit{Kompositum}{Komponente}
1008 \umlVHVinherit{Blatt}{Komponente}
1009 \umlHVHaggreg[anchor1=east,arm1=1.5cm,arg1=eltern,mult1=1,arg2=kind,mult2=0..*,pos2=2.9,
1010 \end{tikzpicture}
1011 }

```

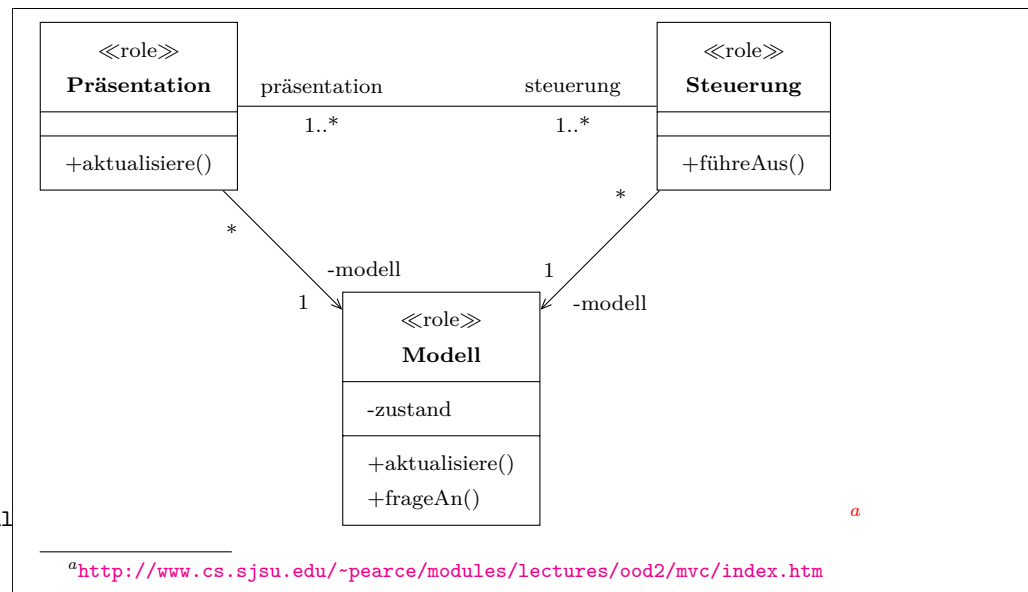
\liEntwurfsFabrikmethode

```

1012 \def\liEntwurfsKompositum{
1013   \liEntwurfsKompositumUml
1014   \liEntwurfsKompositumAkteure
1015 }

```

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



```

1016 \def\liEntwurfsModellPraesentationSteuerungUml{
1017   \begin{tikzpicture}
1018     \umlclass[x=-4cm,type=role]{Präsentation}{+aktualisiere()}
1019     \umlclass[x=4cm,type=role]{Steuerung}{+führeAus()}
1020     \umlclass[y=-4cm,type=role]{Modell}{
1021       -zustand
1022     }{
1023       +aktualisiere()\\
1024       +frageAn()
1025     }
1026
1027     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Präsentation}{Modell}
1028     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Steuerung}{Modell}

```

```

1029 \umlassoc[arg1=präsentation,mult1=1..*,arg2=steuerung,mult2=1..*]{Präsentation}{Steuerung}
1030 \end{tikzpicture}
1031 \liFussnoteUrl{http://www.cs.sjsu.edu/~pearce/modules/lectures/ood2/mvc/index.htm}
1032 }

```

ModellPraesentationSteuerung

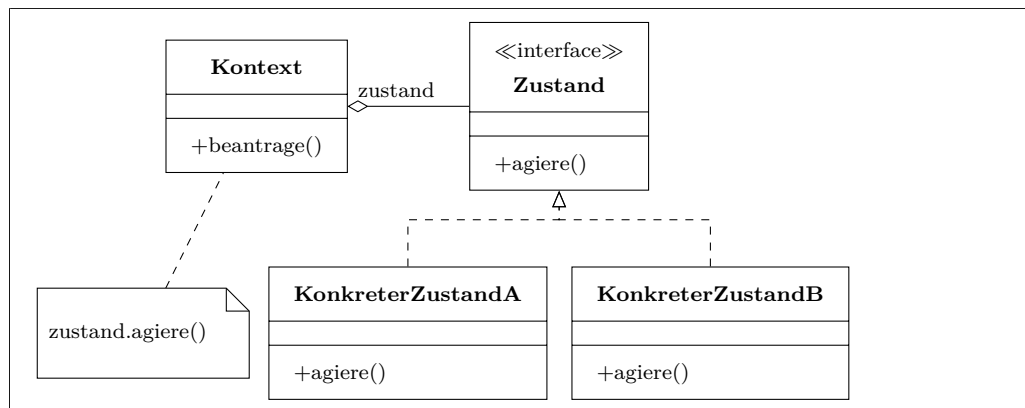
```

1033 \def\liEntwurfs{
1034 \liEntwurfsModellPraesentationSteuerungUml
1035 \liEntwurfsModellPraesentationSteuerungAkteure
1036 }

```

2.10.13 Zustand (State)

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

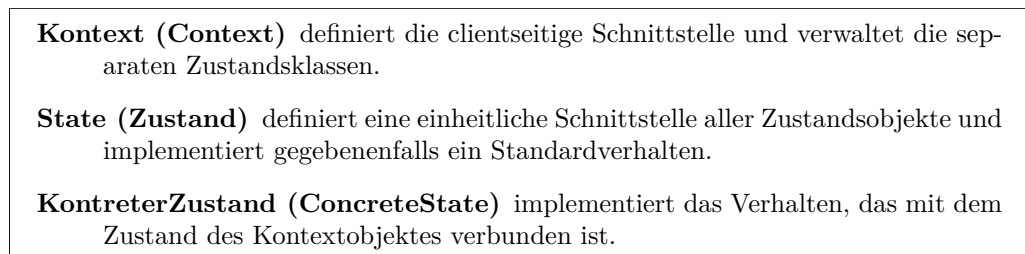


```

1037 \def\liEntwurfsZustandUml{
1038 \begin{tikzpicture}
1039 \umlcclass[x=-1,y=3]{Kontext}{+beantrage()}
1040 \umlcclass[x=3,y=3,type=interface]{Zustand}{+agiere()}
1041 \umlcclass[x=1,y=0]{KonkreterZustandA}{+agiere()}
1042 \umlcclass[x=5,y=0]{KonkreterZustandB}{+agiere()}
1043
1044 \umlVHVreal{KonkreterZustandA}{Zustand}
1045 \umlVHVreal{KonkreterZustandB}{Zustand}
1046
1047 \umlagg[reg[arg=zustand,pos=0.4]{Kontext}{Zustand}
1048
1049 \umlnote[x=-2.5,y=0,width=2.5cm]{Kontext}{zustand.agiere()}
1050 \end{tikzpicture}
1051 }

```

\liEntwurfsZustandAkteure Quelle: Deutsche Wikipedia



```

1052 \def\liEntwurfsZustandAkteure{
1053 \begin{description}
1054 \item[Kontext (Context)]
1055
1056 definiert die clientseitige Schnittstelle und verwaltet die separaten
1057 Zustandsklassen.
1058

```

```

1059 \item[State (Zustand)]
1060
1061 definiert eine einheitliche Schnittstelle aller Zustandsobjekte und
1062 implementiert gegebenenfalls ein Standardverhalten.
1063
1064 \item[KontreterZustand (ConcreteState)]
1065
1066 implementiert das Verhalten, das mit dem Zustand des Kontextobjektes
1067 verbunden ist.
1068 \end{description}
1069 }

\liEntwurfsZustand
1070 \def\liEntwurfsZustand{
1071 \liEntwurfsZustandUml
1072 \liEntwurfsZustandAkteure
1073 }

1074

```

2.11 er.sty

```
1075 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1076 \ProvidesPackage{lehramt-informatik-er}[2020/06/13 Zum Setzen von
1077 ER-Diagrammen]

1078 \RequirePackage{tikz-er2}
1079 \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}

1080 \RequirePackage{soul}
1081 \RequirePackage{fontawesome}

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

1082 \ExplSyntaxOn

\liErEntity
1083 \def\liErEntity#1{\textbf{#1}}

\liErRelationship
1084 \def\liErRelationship#1{\ul{#1}}

\liErAttribute
1085 \def\liErAttribute#1{\emph{#1}}

\liErMpEntity mp = marginpar
Let-Abkürzung: \let\e=\liErMpEntity
1086 \def\liErMpEntity#1{
1087   \liErEntity{#1}
1088   \marginpar{
1089     \liErEntity{\tiny\faSquareO{}}~E:~#1}
1090   }
1091 }
□

\liErMpRelationship Let-Abkürzung: \let\r=\liErMpRelationship
1092 \def\liErMpRelationship#1{
1093   \liErRelationship{#1}
1094   \marginpar{
1095     \liErRelationship{\tiny\faGg{}}~R:~#1}
1096   }
1097 }

\liErMpAttribute Let-Abkürzung: \let\a=\liErMpAttribute
1098 \def\liErMpAttribute#1{
1099   \liErAttribute{#1}
1100   \marginpar{
1101     \liErAttribute{\tiny\faCircleThin{}}~A:~#1}
1102   }
1103 }

\liErDatenbankName Let-Abkürzung: \let\d=\liErDatenbankName
datenbank name
1104 \def\liErDatenbankName#1{
1105   {
1106     \footnotesize\texttt{(#1)}
1107   }
1108 }

1109 \ExplSyntaxOff
1110

```

2.12 formale-sprachen.sty

```

1111 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1112 \ProvidesPackage{lehramt-informatik-formale-sprachen}[2021/02/21 Hilfsmakros
1113 zum Setzen von mathematischen Formeln bei dem Thema Formale Sprachen]

1114 \directlua{
1115   formale_sprachen = require('lehramt-informatik-formale-sprachen')
1116 }

1117 \RequirePackage{hyperref}

1118 \liLadePakete{mathe,typographie}

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

1119 \def\liMengeOhneMathe#1{\{ #1 \}}
1120 \def\liMenge#1{%
1121   \ifmode%
1122     \liMengeOhneMathe{#1}%
1123   \else%
1124     $\liMengeOhneMathe{#1}$%
1125   \fi%
1126 }

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

1127 \def\liEpsilon{$\varepsilon$}

\liPotenzmenge Umgeben mit geschweiften Klammern in einer Mathematik-Umgebung

1128 \def\erzeuge@tiefgestellt#1{\directlua{formale_sprachen.erzeuge_tiefgestellt('#1')}}
1129 \def\liPotenzmengeOhneMathe#1{\{ \erzeuge@tiefgestellt{#1} \}}
1130 \def\liPotenzmenge#1{$\liPotenzmengeOhneMathe{#1}$}

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

1131 \let\liZustandsmengeOhneMathe=\liPotenzmengeOhneMathe
1132 \let\liZustandsmenge=\liPotenzmenge

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

1133 \def\liUeberfuehrungsFunktionOhneMathe#1{\delta(\erzeuge@tiefgestellt{#1})}
1134 \def\liUeberfuehrungsFunktion#1{
1135   \ifmode
1136     \liUeberfuehrungsFunktionOhneMathe{#1}
1137   \else
1138     $\liUeberfuehrungsFunktionOhneMathe{#1}$
1139   \fi
1140 }

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

1141 \def\liAlphabet#1{$\Sigma = \{ #1 \}$}

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

1142 \def\liBandAlphabet#1{$\Gamma = \Sigma \cup \{ #1 \}$}

\liZustandsBuchstabe

1143 \def\liZustandsBuchstabe{z}

\liZustandsBuchstabeGross

1144 \def\liZustandsBuchstabeGross{Z}

```

\liZustandsmengeNr

```

1145 \def\zustandsnamens@liste#1#2{
1146   $
1147   \{
1148     \directlua{formale_sprachen.erzeuge_zustandsnamens_liste('#1', '#2')}
1149   \}
1150   $
1151 }
1152 \def\liZustandsmengeNr#1{\zustandsnamens@liste{\liZustandsBuchstabe}{#1}}

```

\liZustandsmengeNrGross

```

1153 \def\liZustandsmengeNrGross#1{\zustandsnamens@liste{\liZustandsBuchstabeGross}{#1}}

```

\liZustandsname

```

\liZustandsname{1}: $z_1$
1154 \def\liZustandsname#1{\liZustandsBuchstabe_#1$}

```

\liZustandsnameGross

```

\liZustandsnameGross{1}: $Z_1$
1155 \def\liZustandsnameGross#1{\liZustandsBuchstabeGross_#1$}

```

\liAbleitung

```

\liAbleitung{S -> aB -> ab}:  $S \vdash aB \vdash ab$ 
1156 \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}

1157 \NewDocumentEnvironment { liProduktionsRegeln }
1158 { 0{P} +b }
1159 {
1160   \liGeschweifteKlammern{#1}
1161   {
1162     \begin{align*}
1163       \directlua{formale_sprachen.produktions_regeln('#2')}
1164       \end{align*}
1165     }{-0.2cm}{-1.5cm}
1166   } {}

```

\liProduktionen

```

\liProduktionen{S -> A, A -> a}:  $\{S \rightarrow A, A \rightarrow a\}$ 
1167 \def\liProduktionen#1{
1168   \liMenge{\directlua{formale_sprachen.produktions_regeln('#1', true)}}
1169 }

```

\liZustandsnameTiefgestellt

Automatisch tiefgestellte Nummerierung \z1
Let-Abkürzung: \let\z=\liZustandsnameTiefgestellt

```

1170 \def\liZustandsnameTiefgestellt#1{
1171   \ifmmode
1172     \liZustandsBuchstabe\sb{#1}
1173   \else
1174     $\liZustandsBuchstabe\sb{#1}$
1175   \fi
1176 }

```

```

1177 \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}

```



```

1178 \NewDocumentCommand{ \liAusdruck } { O{L} m m } {
1179   $
1180   \tl_if_empty:nTF {#1} {} {#1 =}
1181   \{
1182     \, #2 \,
1183     |
1184     \, #3 \,
1185   \}$
1186 }
1187 \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

1188 \def\liFlaci#1{%
1189   \par
1190   {%
1191     \scriptsize
1192     Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte
1193     Automaten, Compiler und Interpreter) Ein Projekt der Hochschule
1194     Zittau/Görlitz und der Pädagogischen Hochschule Schwyz:
1195     \href{https://flaci.com/#1}{flaci.com/#1}%
1196   }%
1197   \par
1198 }

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

1199 \ExplSyntaxOn
1200 \NewDocumentCommand {\liGrammatik} { O{G} m } {
1201   \tl_set:Nn \l_variablen_tl {V}
1202   \tl_set:Nn \l_alphabet_tl {\Sigma}
1203   \tl_set:Nn \l_produktionen_tl {P}
1204   \tl_set:Nn \l_start_tl {S}
1205
1206   \keys_define:nn { grammatik } {
1207     variablen .code:n = {\tl_set:Nn \l_variablen_tl {\liMenge{##1}}},
1208     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
1209     produktionen .code:n = {\tl_set:Nn \l_produktionen_tl {\liProduktionen{##1}}},
1210     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
1211   }
1212
1213   \keys_set:nn { grammatik } { #2 }
1214
1215   $#1 = (
1216     \l_variablen_tl,
1217     \l_alphabet_tl,
1218     \l_produktionen_tl,
1219     \l_start_tl
1220   )$
1221 }
1222 \ExplSyntaxOff

1223

```

2.13 formatierung.sty

```
1224 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1225 \ProvidesPackage{lehramt-informatik-formatierung}[2020/11/27]
```

2.13.1 Schriftarten / Typographie

```
1226 \RequirePackage{mathpazo}
1227 \RequirePackage[no-math]{fontspec}
1228 \setmainfont{texgyrepagella}
1229 \setsansfont{QTAncientOlive}
1230 \RequirePackage{sectsty}
1231 \allsectionsfont{\fontspec{QTAncientOlive}}
```

2.13.2 Farben

```
1232 \RequirePackage{xcolor}
1233 \definecolor{infogray}{rgb}{0.97,0.97,0.97}
```

2.13.3 Überschriften

```
1234 \RequirePackage{titlesec}
1235 \titleformat{\chapter}[display]{\sffamily\bfseries}{\Opt}{\LARGE}
1236 \titlespacing{\chapter}{0pt}{0pt}{*1}
1237 \titleformat{\paragraph}[hang]{\sffamily\normalsize\bfseries}{\theparagraph}{1em}{}
1238 \setcounter{secnumdepth}{0}
```

2.13.4 Listen

```
1239 \RequirePackage{paralist}
1240 \renewcommand\labelitemi{-}
1241 \renewcommand\labelitemii{-}
1242 \renewcommand\labelitemiii{-}
1243 \renewcommand\labelitemiv{-}
1244 % Counter: enumi enumii enumiii enumiv
1245 % Styles: \arabic{counter} \alph{counter} \Alph{counter} \roman{counter} \Roman{counter}
1246 \renewcommand{\labelenumi}{(\alph{enumi})}
1247 \renewcommand{\labelenumii}{(\roman{enumii})}
```

2.13.5 Kasten

```
1248 \RequirePackage{mdframed}
1249 \mdfsetup{backgroundcolor=infogray}
```

liKasten

```
1250 \NewDocumentEnvironment { liKasten }{ } {
1251   \begin{mdframed}[backgroundcolor=white!0]
1252 } {
1253   \end{mdframed}
1254 }
```

2.13.6 Header

```
1255 \RequirePackage{fancyhdr}
1256 \fancyhead[L,C,R]{}
1257 \fancyfoot[L]{}
1258 \fancyfoot[C]{}
1259 \fancyfoot[R]{\thepage}
1260 \pagestyle{fancy}
1261 \renewcommand{\headrulewidth}{0pt}
1262 \renewcommand{\footrulewidth}{0pt}
1263
```

2.14 gantt.sty

```

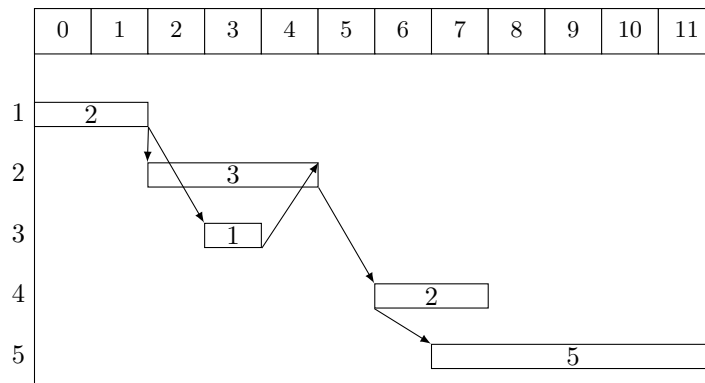
1264 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1265 \ProvidesPackage{lehramt-informatik-gantt}[2020/09/05]

\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}

```



```

1266 \RequirePackage{tikz-uml}
1267 \RequirePackage{pgfgantt}
1268 \setganttlinklabel{f-s}{}
1269 \setganttlinklabel{s-s}{}
1270 \setganttlinklabel{f-f}{}
1271 \setganttlinklabel{s-f}{}

1272

```

2.15 grafik.sty

```
1273 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1274 \ProvidesPackage{lehramt-informatik-grafik}[2020/12/27 Lädt das Paket
1275 TikZ, um Grafiken zeichnen zu können.]
1276 \RequirePackage{tikz}
1277
```

2.16 graph.sty

```

1278 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1279 \ProvidesPackage{lehramt-informatik-graph}[2020/06/09]
1280 \RequirePackage{tikz}

```

Für die die Adjazenzliste (`xrightharpoon`)

```

1281 \RequirePackage{amsmath}

```

Für Adjazenz-Matrix

```

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

```

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

```

1282 \RequirePackage{blkarray}

```

```

1283 \usetikzlibrary{arrows.meta}

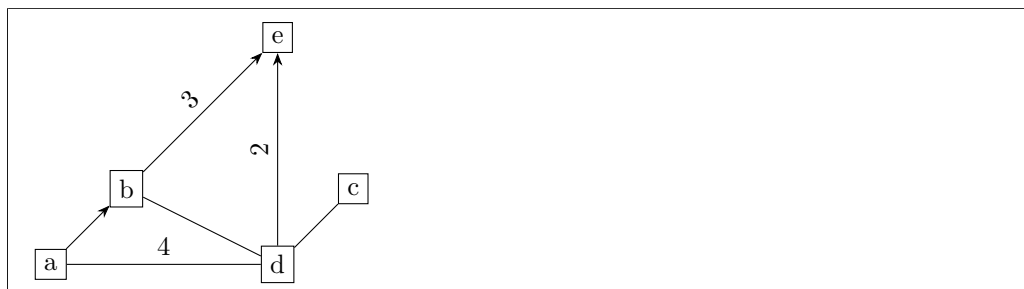
```

```

\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}

```



```

1284 \tikzset{
1285   li graph/.style={
1286     every node/.style={
1287       rectangle,
1288       draw,
1289     },
1290     every edge/.style={
1291       >={Stealth[black]},
1292       draw,
1293     },
1294     every edge/.append style={
1295       every node/.style={
1296         sloped,
1297         auto,
1298       }
1299     }
1300   },
1301   li markierung/.style={
1302     ultra thick,
1303   }
1304 }

```

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}

```

```

1305 \NewDocumentEnvironment { liGraphenFormat }{ +b } {} {}

```

```

1306

```

2.17 hanoi.sty

```
1307 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1308 \ProvidesPackage{lehramt-informatik-hanoi}[2020/12/19 Paket zum Setzen
1309 von Hanoi-Grafiken]
```

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

```
1310 \RequirePackage{tikz}
1311 \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}

1312 \def\li@mset #1[#2]=#3{%
1313 \expandafter\xdef\csname #1#2\endcsname{#3}
1314 }
1315 \def\li@mget #1[#2]{%
1316 \csname #1#2\endcsname
1317 }
1318 \def\li@minc #1[#2]+=#3{%
1319 \pgfmathparse{\li@mget #1[#2]+#3}%
1320 \li@mset #1[#2]=\pgfmathresult
1321 }
1322
1323 \def\liHanoi#1#2{
1324   \edef\li@numdiscs{#1}
1325   \def\li@sequence{#2}
1326   \begin{tikzpicture}[line width=4mm,brown!40,line cap=round,xscale=3]
1327     % init colors
1328     \foreach[count=\j] \c in {red,green!80!black,blue,orange,violet,gray,yellow!80!black,purp
1329     \li@mset col[\j]={\c};
1330     % draw poles and init pole counters
1331     \foreach \j in {1,2,3}{
1332       \li@mset pos[\j]=0
1333       \draw (\j,-.5) -- +(0, .5 + .5*\li@numdiscs);
1334     }
1335     % draw base
1336     \draw (.5,-.5) -- +(3,0);
1337     % draw discs
1338     \foreach[count=\k] \i/\j in \li@sequence{
1339       \draw[draw={\li@mget col[\i]}] (\j,\li@mget pos[\j]) +(-.4*\i/\li@numdiscs,0) -- +(.4*
1340       \li@minc pos[\j]+=.5}
1341     }
1342   \end{tikzpicture}
1343 }

1344
```

2.18 komplexitaetstheorie.sty

```

1345 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1346 \ProvidesPackage{lehramt-informatik-komplexitaetstheorie}[2021/07/08 Zum
1347 Setzen von Karp's NP-vollständigen Problemen, Hilfsmakros für die
1348 Polynomialzeitreduktion.]

```

Let-Abkürzungen

```

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

```

```

1349 \liLadePakete{mathe}

```

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

```

1350 \RequirePackage{mdframed}

```

```

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

```

```

1351 \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

```

```

1352 \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`

```

1353 \def\liProblemBeschreibung#1#2#3{
1354   \begin{mdframed}[
1355     userdefinedwidth=9cm,
1356     align=center,
1357     backgroundcolor=white!0,
1358   ]
1359     \centerline{\large\liProblemName{#1}}
1360
1361     \medskip
1362
1363     \begin{description}
1364       \item[Gegeben:] #2
1365       \item[Frage:] #3
1366     \end{description}
1367   \end{mdframed}
1368 }

```



```

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

1369 \NewDocumentCommand{ \liPolynomiellReduzierbar } { m O{p} m } {
1370 \begin{displaymath}
1371 \liProblemName{#1}
1372 \preceq_{#2}
1373 \liProblemName{#3}
1374 \end{displaymath}
1375 }

\liProblemVertexCover

1376 \def\liProblemClique{%
1377 Das \textbf{Cliquesproblem} fragt nach der Existenz einer Clique der
1378 Mindestgröße  $n$  in einem gegebenen Graphen.
1379 \footcite[Seite 76]{theo:fs:4}
1380 Eine Clique ist eine Teilmenge von Knoten in einem ungerichteten
1381 Graphen, bei der \emph{jedes Knotenpaar durch eine Kante} verbunden ist.
1382 \footcite{wiki:cliquenproblem}
1383 }

\liProblemVertexCover

1384 \def\liProblemVertexCover{%
1385 %
1386 Das \textbf{Knotenüberdeckungsproblem} (\liProblemName{Vertex Cover})
1387 fragt, ob zu einem gegebenen einfachen Graphen und einer natürlichen
1388 Zahl  $k$  eine Knotenüberdeckung der Größe von höchstens  $k$  existiert.
1389 \footcite{wiki:knotenueberdeckung}
1390
1391 Das heißt, ob es eine aus maximal  $k$  Knoten bestehende Teilmenge  $U$ 
1392 der Knotenmenge gibt, so dass jede Kante des Graphen mit
1393 mindestens einem Knoten aus  $U$  verbunden ist.
1394 \footcite[Seite 78]{theo:fs:4}%
1395 }

\liProblemSubsetSum Kein Karp-21-Problem

1396 \def\liProblemSubsetSum{%
1397 Das \textbf{Teilsommenproblem} (\liProblemName{Subset Sum} oder
1398 \liProblemName{SSP}) ist ein spezielles Rucksackproblem.
1399 \footcite{wiki:teilsommenproblem}
1400 Gegeben sei eine Menge von ganzen Zahlen  $I = \{w_1, w_2, \dots, w_n\}$ .
1401 Gesucht ist eine Untermenge, deren Elementsumme maximal, aber nicht
1402 größer als eine gegebene obere Schranke  $c$  ist.
1403 \footcite[Seite 74]{theo:fs:4}
1404 }

\liProblemSubsetSum Kein Karp-21-Problem

1405 \def\liProblemSat{%
1406 Das \textbf{Erfüllbarkeitsproblem der Aussagenlogik} \liProblemName{Sat}
1407 und \liProblemName{k-SAT} mit  $k \geq 3$ ,  $k \in \mathbb{N}$  (Satz von
1408 Cook) fragt, ob eine aussagenlogische Formel erfüllbar
1409 ist.\footcite{wiki:sat} Das Erfüllbarkeitsproblem der
1410 \emph{Aussagenlogik} ist in exponentieller Zeit in Abhängigkeit der
1411 Anzahl der Variablen mit Hilfe einer Wahrheitstabelle entscheidbar.
1412 Diese \emph{Wahrheitstabelle} kann nicht in polynomieller Zeit
1413 aufgestellt werden.
1414 \footcite[Seite 71]{theo:fs:4}
1415 }

1416

```

2.19 kontrollflussgraph.sty

```
1417 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1418 \ProvidesPackage{lehramt-informatik-kontrollflussgraph}[2020/11/07]

1419 \RequirePackage{tikz}
1420 \usetikzlibrary{positioning}
1421 \tikzset{
1422   li kontrollfluss/.style={
1423     knoten/.style={
1424       circle,
1425       draw
1426     },
1427     usebox/.style={
1428       draw,
1429       rectangle,
1430       font=\scriptsize,
1431       anchor=west,
1432       align=left,
1433     },
1434     bedingung/.style={
1435       midway,
1436       draw=none,
1437       font=\scriptsize
1438     },
1439     knotenbeschriftung/.style={
1440       draw,
1441       rectangle,
1442       midway,
1443       font=\scriptsize
1444     },
1445     wahr/.style={
1446       thick
1447     },
1448     falsch/.style={
1449       dashed
1450     },
1451     every node/.style={
1452       circle,
1453       draw,
1454     },
1455     every edge/.append style={
1456       every node/.style={
1457         draw=none,
1458         bedingung,
1459       }
1460     },
1461     every path/.style={
1462       draw,
1463       ->,
1464     },
1465     every pin/.style={
1466       draw,
1467       dotted,
1468       rectangle,
1469       pin position=right
1470     },
1471     every pin edge/.style={
1472       dotted,
1473       arrows=-,
1474     }
1475   }
1476 }
```

liKontrollflussgraph

```

1477 \NewDocumentEnvironment { liKontrollflussgraph } { 0{ } } {
1478   \begin{tikzpicture}[
1479     li kontrollfluss,
1480     #1
1481   ]
1482 } {
1483   \end{tikzpicture}
1484 }

\liAnweisung
1485 \def\liAnweisung#1(#2,#3){\node[knoten] at (#2,#3) (#1) {#1};}

\liBedingung Let-Abkürzung: \let\b=\liBedingung
1486 \def\liBedingung#1#2{node[bedingung,#1]{\texttt{#2}}}}

\liBedingungWahr Let-Abkürzung: \let\bWahr=\liBedingungWahr
1487 \def\liBedingungWahr#1{node[bedingung,#1]{\texttt{[true]}}}}

\liBedingungFalsch Let-Abkürzung: \let\bFalsch=\liBedingungFalsch
1488 \def\liBedingungFalsch#1{node[bedingung,#1]{\texttt{[false]}}}}

\liKontrollCode Let-Abkürzung: \let\c=\liKontrollCode
1489 \def\liKontrollCode#1{{\tiny\texttt{\textcolor{gray}{#1}}}}

\liKontrollTextzeileKnoten Let-Abkürzung: \let\k=\liKontrollTextzeileKnoten
1490 \def\liKontrollTextzeileKnoten#1{\raisebox{-2pt}{\tikz[scale=0.5,transform shape] \node[draw

\liKontrollKnotenPfad Let-Abkürzung: \let\p=\liKontrollKnotenPfad
1491 \ExplSyntaxOn
1492 \NewDocumentCommand { \liKontrollKnotenPfad }{ m }
1493 {
1494   \seq_set_split:Nnn \l_tmpa_seq { - } { #1 }
1495   \seq_set_map:Nnn \l_tmpa_seq \l_tmpa_seq {\liKontrollTextzeileKnoten{##1}}
1496   \seq_use:Nn \l_tmpa_seq { ~~~~ }
1497 }
1498 \ExplSyntaxOff

1499

```

2.20 kopf-fusszeilen.sty

```
1500 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1501 \ProvidesPackage{lehramt-informatik-kopf-fusszeilen}[2021/08/20 Kopf-
1502 und Fußzeilen mit Hilfe des Pakets fancyhdr.]

1503 \ExplSyntaxOn

1504 \fancyhead{}
1505 \fancyhead[R0,LE]{\scriptsize\LehramtInformatikTitel}
1506 \fancyhead[L0,RE]{\scriptsize\today}
1507 \fancyfoot{}
1508 \fancyfoot[LE,R0]{\thepage}
1509 \fancyfoot[L0,CE]{\scriptsize\LehramtInformatikAutorName}
1510 \fancyfoot[CO,RE]{\scriptsize\LehramtInformatikAutorEmail}
1511 \renewcommand{\headrulewidth}{0.4pt}
1512 \renewcommand{\footrulewidth}{0.4pt}

1513 \ExplSyntaxOff

1514
```

2.21 literatur-dummy.sty

```
1515 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1516 \ProvidesPackage{lehramt-informatik-literatur-dummy}[2020/11/27]

\literatur

1517 \def\literatur{}

\footcite

1518 % \RequirePackage[stable,multiple]{footmisc}
1519 \NewDocumentCommand{ \footcite } { o m }{}

1520
```

2.22 literatur.sty

```
1521 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1522 \ProvidesPackage{lehramt-informatik-literatur}[2020/11/27]

1523 \RequirePackage{csquotes}
1524 \RequirePackage[
1525   bibencoding=utf8,
1526   citestyle=authortitle,
1527   backend=biber,
1528 ]{biblatex}
1529 \addbibresource{\LehramtInformatikRepository/Quellen/10_DB.bib}
1530 \addbibresource{\LehramtInformatikRepository/Quellen/20_OOMUP.bib}
1531 \addbibresource{\LehramtInformatikRepository/Quellen/30_AUD.bib}
1532 \addbibresource{\LehramtInformatikRepository/Quellen/40_SOSY.bib}
1533 \addbibresource{\LehramtInformatikRepository/Quellen/60_FUMUP.bib}
1534 \addbibresource{\LehramtInformatikRepository/Quellen/70_THEO.bib}
1535 \addbibresource{\LehramtInformatikRepository/Quellen/80_DDI.bib}
1536 \addbibresource{\LehramtInformatikRepository/Quellen/Allgemein.bib}
1537 \addbibresource{\LehramtInformatikRepository/Quellen/Examen.bib}
1538 \addbibresource{\LehramtInformatikRepository/Quellen/Wikipedia.bib}
1539 % To allow footnotes in the heading
1540 \RequirePackage[stable,multiple]{footmisc}

\literatur

1541 \def\literatur{\printbibliography[heading=subbibliography]}

1542
```

2.23 makros.sty

```

1543 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1544 \ProvidesPackage{lehramt-informatik-makros}[2020/11/27 Eine Sammlung von
1545 häufig verwendeten Makros und Umgebungen, die thematisch zu keinem
1546 anderen Paket passen]
1547 \RequirePackage{hyperref}
1548 \RequirePackage{graphicx}

Für die Umgebung liQuellen benötigt.
1549 \RequirePackage{paralist}

\inhaltsverzeichnis
1550 \def\inhaltsverzeichnis {
1551   \begin{mdframed}
1552     \begin{group}
1553       \let\clearpage\relax
1554       \tableofcontents
1555     \end{group}
1556   \end{mdframed}
1557 }

\memph \mephm (\marginpar and \emph)
1558 \newcommand\memph{1}{\emph{#1}\marginpar{\tiny#1}}

\SLASH
1559 \newcommand\SLASH{\char`\\}

\liPseudoUeberschrift Text, der sich wie eine Überschrift verhält.
1560 \newcommand\liPseudoUeberschrift{1}{
1561   \bigskip
1562   \noindent
1563   \textsf{\textbf{#1}}
1564   \noindent
1565 }

\liBeschriftung Ähnlich dem Makro \liPseudoUeberschrift{}. Am Ende des Textes wird ein Doppelpunktzeichen angehängt.
1566 \newcommand\liBeschriftung{1}{
1567   \par
1568   \noindent
1569   \medskip
1570   \textbf{#1}:
1571   \medskip
1572   \noindent
1573 }

\hinweis
1574 \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.
1575 \NewDocumentEnvironment { liProjektSprache }{ o +b } {} {}

liEinbettung
1576 \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.
1577 \RequirePackage{xparse}
1578 \ExplSyntaxOn

```

```

1579 \NewDocumentEnvironment{ liAntwort } { 0{standard} }
1580 {
1581   \str_case:nn {#1} {
1582     {standard} {
1583       \def\beschriftung{}
1584       \mdfsetup{backgroundcolor=gray!1,linecolor=gray}
1585     }
1586     {richtig} {
1587       \def\beschriftung{richtig}
1588       \mdfsetup{backgroundcolor=green!5,linecolor=green}
1589     }
1590     {falsch} {
1591       \def\beschriftung{falsch}
1592       \mdfsetup{backgroundcolor=red!3,linecolor=red}
1593     }
1594     {muster} {
1595       \def\beschriftung{Musterlösung}
1596       \mdfsetup{backgroundcolor=green!12,linecolor=green}
1597     }
1598   }
1599   \ifx\beschriftung\empty\else
1600   \noindent
1601   \textbf{\beschriftung{:}}
1602   \fi
1603   \begin{mdframed}
1604 }
1605 {\end{mdframed}}

```

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

```

1606 \NewDocumentEnvironment{ liAdditum } { o }
1607 {
1608   \begin{mdframed}[backgroundcolor=yellow!5]
1609   \IfNoValueTF {#1}
1610   { \liPseudoUeberschrift{Additum} }
1611   { \liPseudoUeberschrift{Additum:~#1} }
1612 }
1613 {\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.

```

1614 \NewDocumentEnvironment{ liExkurs }{o +b}{
1615   \vspace{0.2cm}%
1616   \begin{mdframed}[
1617     backgroundcolor=white,
1618     bottomline=false,
1619     innermargin=1cm,
1620     leftline=true,
1621     linecolor=black,
1622     linewidth=0.1cm,
1623     outermargin=1cm,
1624     rightline=false,
1625     topline=false,
1626   ]

```



```

1627 \footnotesize
1628 \noindent%
1629 \textbf{Exkurs:~#1}\par%
1630 \noindent%
1631 #2
1632 \end{mdframed}
1633 \vspace{0.2cm}
1634 }{}

```

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

```

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

```

Weiterführende Literatur:

- Quelle 1
- Quelle 2

```

1635 \cs_new:Npn \listen@punkt #1 {\item #1}
1636 \NewDocumentEnvironment { liQuellen }{ +b }
1637 {
1638   \seq_clear_new:N \l_quellen
1639   \seq_set_split:Nnn \l_quellen {\item} {#1}
1640   \seq_remove_all:Nn \l_quellen {}
1641   \begin{mdframed}[backgroundcolor=white!0]
1642     \footnotesize
1643     \noindent
1644     \textsf{\textbf{Weiterführende~Literatur:}}
1645     \medskip
1646     \begin{compactitem}
1647       \seq_map_function:NN \l_quellen {\listen@punkt}
1648     \end{compactitem}
1649   \end{mdframed}
1650   %
1651   \makeatletter
1652   \par\@afterindentfalse\@afterheading
1653   \makeatother
1654 } {}

```

liLernkartei

```

1655 \NewDocumentEnvironment { liLernkartei }{ m +b }
1656 {
1657   \begin{mdframed}
1658     \footnotesize
1659     \noindent%
1660     \textbf{Lernkarteikarte:~#1}\par%
1661     \noindent%
1662     #2
1663   \end{mdframed}
1664 } {}

```

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

```

1665 \NewDocumentEnvironment { liDiagramm }{ m +b }
1666 {
1667   \begin{mdframed}[backgroundcolor=white!0]
1668     \small
1669     \noindent%
1670     \textit{#1}:
1671     \begin{center}

```

```

1672 #2
1673 \medskip
1674 \end{center}
1675 \end{mdframed}
1676 } {}
1677 \ExplSyntaxOff

\liFussnoteUrl \liFussnoteUrl[\langle zusätzlicher-text \rangle]{\langle url \rangle} \liFussnoteUrl[zusätzlicher Text]{url}:
Eine HTTP-URL als Fußnote setzen.
1678 \NewDocumentCommand{\liFussnoteUrl} { o m } {
1679 \footnote{\url{#2}\IfNoValueTF{#1}{}{ ( #1 )}}
1680 }
1681

\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.
1682 \NewDocumentCommand{\liFussnoteLink} { o m m } {
1683 \footnote{\href{#3}{#2}\IfNoValueTF{#1}{}{ ( #1 )}}
1684 }

\zB
1685 \def\zB{z.\,B. }

\ZB
1686 \def\ZB{Z.\,B. }

\dh
1687 \def\dh{d.\,h. }

1688

```

2.24 master-theorem.sty

1689 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

1690 \ProvidesPackage{lehramt-informatik-master-theorem}[2021/04/13]

Let-Abkürzungen

\let\0=\li0

\let\o=\li0mega

\let\T=\liT

\let\t=\liTheta

\liMasterVariablenDeklaration

{3} % a

{3} % b

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

\liMasterFallRechnung

% 1. Fall

{für $\varepsilon = 4$: \}

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

% 2. Fall

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

% 3. Fall

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

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

\liMasterVariablenDeklaration

{ } % a

{ } % b

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

\liMasterFallRechnung

% 1. Fall

{ }

% 2. Fall

{ }

% 3. Fall

{ }

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

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

1691 \ExplSyntaxOn

1692 \RequirePackage{amsmath}

\liRundeKlammer

1693 \def\liRundeKlammer#1{

1694 \negthinspace \left(#1 \right)

1695 }

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

1696 \def\liThetaOhneMathe#1{

1697 \Theta \liRundeKlammer{#1}

1698 }

1699 \def\liTheta#1{

1700 \ifmmode

1701 \liThetaOhneMathe{#1}

1702 \else

1703 \$\liThetaOhneMathe{#1}\$

1704 \fi

1705 }

```

\liOmega \liOmega{n^2}:  $\Omega(n^2)$ 
1706 \def\liOmegaOhneMathe#1{
1707 \Omega \liRundeKlammer{#1}
1708 }
1709 \def\liOmega#1{
1710 \ifmmode
1711 \liOmegaOhneMathe{#1}
1712 \else
1713 $\liOmegaOhneMathe{#1}$
1714 \fi
1715 }

\liO \liO{n^2}:  $\mathcal{O}(n^2)$ 
1716 \def\liOOhneMathe#1{
1717 \mathcal{O} \liRundeKlammer{#1}
1718 }
1719 \def\liO#1{
1720 \ifmmode
1721 \liOOhneMathe{#1}
1722 \else
1723 $\liOOhneMathe{#1}$
1724 \fi
1725 }

\liT Let-Abkürzung: \let\T=\liT
\liT{16}{2}:  $16 \cdot T\left(\frac{n}{2}\right)$  \liT{}{2}:  $T\left(\frac{n}{2}\right)$ 
1726 \def\liTOhneMathe#1#2{
1727 \tl_if_blank:nTF {#1}
1728 {}
1729 {#1 \cdot }
1730 T
1731 \liRundeKlammer{\frac{n}{#2}}
1732 }
1733 \def\liT#1#2{
1734 \ifmmode
1735 \liTOhneMathe{#1}{#2}
1736 \else
1737 $\liTOhneMathe{#1}{#2}$
1738 \fi
1739 }

\liRekursionsGleichung \liRekursionsGleichung:  $T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$ 
1740 \def\liRekursionsGleichung{
1741 $T(n) = \liT{a}{b} + f(n)$
1742 }

\liBedingungEins \liBedingungEins:  $f(n) \in \mathcal{O}(n^{\log_b a - \varepsilon})$ 
1743 \def\liBedingungEins{
1744 $f(n) \in \liO{n^{\log\sb{b}a - \varepsilon}}$
1745 }

\liBedingungZwei \liBedingungZwei:  $f(n) \in \Theta(n^{\log_b a})$ 
1746 \def\liBedingungZwei{
1747 $f(n) \in \liTheta{n^{\log\sb{b}a}}$
1748 }

\liBedingungDrei \liBedingungDrei:  $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ 
1749 \def\liBedingungDrei{
1750 $f(n) \in \liOmega{n^{\log\sb{b}a + \varepsilon}}$
1751 }

1752 \ExplSyntaxOff

```

\liMasterVariablen

```

1753 \def\liMasterVariablen{
1754   \begin{displaymath}
1755     T(n) = \liT{a}{b} + f(n)
1756   \end{displaymath}
1757
1758   \begin{itemize}
1759     \item[$a = $]
1760     Anzahl der Unterprobleme in der Rekursion
1761
1762     \item[$\textstyle{\frac{1}{b}} = $]
1763     Teil des Originalproblems, welches wiederum durch alle Unterprobleme
1764     repräsentiert wird
1765
1766     \item[$f(n) = $]
1767     Kosten (Aufwand, Nebenkosten), die durch die Division des Problems und
1768     die Kombination der Teillösungen entstehen
1769   \end{itemize}
1770   \footcite{wiki:master-theorem}
1771   \footcite[Seite 19-35 (PDF 11-24)]{aud:fs:2}
1772 }

```

\liMasterFaelle

```

1773 \def\liMasterFaelle{
1774   \begin{description}
1775     \item[1. Fall:]
1776      $T(n) \in \liTheta{n^{\log_{sb{b}}a}}$ 
1777
1778     \hfill falls \liBedingungEins
1779     für  $\varepsilon > 0$ 
1780
1781     \item[2. Fall:]
1782      $T(n) \in \liTheta{n^{\log_{sb{b}}a} \cdot \log n}$ 
1783
1784     \hfill falls \liBedingungZwei
1785
1786     \item[3. Fall:]
1787      $T(n) \in \liTheta{f(n)}$ 
1788
1789     \hfill falls \liBedingungDrei
1790     für  $\varepsilon > 0$ 
1791     und ebenfalls für ein  $c$  mit  $0 < c < 1$  und alle hinreichend großen  $n$ 
1792     gilt:
1793      $a \cdot f(\textstyle{\frac{n}{b}}) \leq c \cdot f(n)$ 
1794   \end{description}
1795 }

```

\liMasterVariablenDeklaration

```

1796 \def\liMasterVariablenDeklaration#1#2#3{
1797   \begin{description}
1798     \item[Allgemeine Rekursionsgleichung:] \strut
1799
1800     \liRekursionsGleichung
1801
1802     \item[Anzahl der rekursiven Aufrufe ( $a$ ):] \strut
1803
1804     #1
1805
1806     \item[Anteil Verkleinerung des Problems ( $b$ ):] \strut
1807
1808     um  $\frac{1}{#2}$  also  $b = #2$ 
1809
1810     \item[Laufzeit der rekursiven Funktion ( $f(n)$ ):] \strut

```

```

1811
1812     $#3$
1813
1814     \item[Ergibt folgende Rekursionsgleichung:] \strut
1815
1816     $T(n) = \liT{\#1}{\#2} + \#3$
1817 \end{description}
1818 }

\liMasterFallRechnung
1819 \def\liMasterFallRechnung#1#2#3{
1820 \begin{description}
1821 \item[1. Fall:] \liBedingungEins:
1822
1823     #1
1824
1825 \item[2. Fall:] \liBedingungZwei:
1826
1827     #2
1828
1829 \item[3. Fall:] \liBedingungDrei:
1830
1831     #3
1832 \end{description}
1833 }

\liMasterExkurs
1834 \def\liMasterExkurs{
1835 \begin{liExkurs}[Master-Theorem]
1836 \liMasterVariablen
1837
1838 \noindent
1839 Dann gilt:
1840
1841 \liMasterFaelle
1842 \end{liExkurs}
1843 }

\liMasterWolframLink Link zu Wolframalpha (+ durch Prozent 2B ersetzen)
1844 \def\liMasterWolframLink#1{
1845 Berechne die Rekursionsgleichung auf WolframAlpha:
1846 \href{https://www.wolframalpha.com/input/?i=#1}{WolframAlpha}
1847 }

1848

```

2.25 mathe.sty

```
1849 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1850 \ProvidesPackage{lehramt-informatik-mathe}[2020/06/10]
1851
1852 % for example \ltimes \rtimes
1853 %\RequirePackage{amssymb}
1854 \RequirePackage{amsmath}
1855
1856 %%
1857 % \mlq \mrq
1858 %%
1859 \DeclareMathSymbol{\mlq}{\mathord}{operators}{``}
1860 \DeclareMathSymbol{\mrq}{\mathord}{operators}{`}
1861
```

2.26 minimierung.sty

```

1862 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1863 \ProvidesPackage{lehramt-informatik-mathe}[2021/03/13 Für den
1864 Minimierungsalgorithmus von einem NEA zu einem DEA]

1865 \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

1866 \def\liFussnote#1{\$x_{#1}\$}

1867 \def\li@fussnote@text#1#2{
1868 \liFussnote{#1}
1869 \quad
1870 {\footnotesize #2}
1871 }

\liFussnoteEinsText

1872 \def\liFussnoteEinsText{
1873 \li@fussnote@text{1}
1874 {Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.}
1875 }

\liFussnoteZweiText

1876 \def\liFussnoteZweiText{
1877 \li@fussnote@text{2}
1878 {Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt.}
1879 }

\liFussnoteDreiText

1880 \def\liFussnoteDreiText{
1881 \li@fussnote@text{3}

```



```

1882 {In weiteren Iterationen markierte Zustände.}
1883 }

\liFussnoteVierText
1884 \def\liFussnoteVierText{
1885   \li@fussnote@text{4}
1886   {...}
1887 }

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



1888 \def\liFussnoten{
1889   \bigskip
1890
1891   \noindent
1892   \liFussnoteEinsText
1893
1894   \noindent
1895   \liFussnoteZweiText
1896
1897   \noindent
1898   \liFussnoteDreiText
1899
1900   \noindent
1901   \liFussnoteVierText
1902 }

\liLeereZelle \liLeereZelle:  $\emptyset$ 
Let-Abkürzung: \let\l=\liLeereZelle
1903 \def\liLeereZelle{\$\emptyset\$}

\liZustandsPaarVariablenName
1904 \def\liZustandsPaarVariablenName{z}

\liZustandsPaar
1905 \def\liZustandsPaar#1#2{
1906   $(
1907     \liZustandsPaarVariablenName_#1,
1908     \liZustandsPaarVariablenName_#2
1909   )$
1910 }

liUebergangsTabelle
1911 \renewcommand{\arraystretch}{1.4}
1912 \NewDocumentEnvironment{ liUebergangsTabelle } { m m } {
1913   \liPseudoUeberschrift{Übergangstabelle}
1914   \begin{center}
1915     \begin{tabular}{r|l|l}
1916       \textbf{Zustandspaar} & \textbf{\#1} & \textbf{\#2} \\ \hline
1917     \end{tabular}
1918   \end{center}
1919 }
1920 }

liUeberschriftDreiecksTabelle \liUeberschriftDreiecksTabelle: Minimierungstabelle (Table filling)

1921 \ExplSyntaxOn
1922 \def\liUeberschriftDreiecksTabelle{
1923   \liPseudoUeberschrift{Minimierungstabelle~(Table~filling)}
1924 }

```

\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. —————

```

1925 \def\liMinimierungErklaerung{
1926   %\footcite[Seite~19]{koenig}
1927   \liParagraphMitLinien{
1928     Der~Minimierungs-Algorithmus~(auch~Table-Filling-Algorithmus-genannt)~
1929     trägt~in~seinem~Verlauf~eine~Markierung~in~alle~
1930     diejenigen~Zellen~der~Tabelle~ein,~die~zueinander~nicht~äquivalente~
1931     Zustände~bezeichnen.~Die~Markierung~„ $x_{\sb{n}}$ “~in~einer~Tabellenzelle~( $i$ ,~
1932      $j$ )~bedeutet~dabei,~dass~das~Zustandspaar~( $i$ ,~ $j$ )~in~der~ $k$ -ten~
1933     Iteration~des~Algorithmus~markiert~wurde~und~die~Zustände~ $i$ ~und~ $j$ ~
1934     somit~zueinander~( $k - 1$ )~äquivalent,~aber~nicht~ $k$ -äquivalent~und~
1935     somit~insbesondere~nicht~äquivalent~sind.~Bleibt~eine~Zelle~bis~zum~Ende~
1936     unmarkiert,~sind~die~entsprechenden~Zustände~zueinander~äquivalent.
1937   }
1938 }
1939 \ExplSyntaxOff
1940

```

2.27 normalformen.sty

```
1941 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1942 \ProvidesPackage{lehramt-informatik-normalformen}[2020/12/10]
1943 Hilfsmakros zum Setzen von Normalformen, Funktionalen Abhängigkeiten,
1944 Attributhülle]
```

Mathe für textit tex etc, typographie für geschweifte Klammer

```
1945 \liLadePakete{mathe,typographie}
1946 \directlua{
1947   helper = require('lehramt-informatik-helfer')
1948   normalformen = require('lehramt-informatik-normalformen')
1949 }
```

Let-Abkürzungen

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

```
1950 \def\liTeilen#1{
1951   \directlua{tex.print(normalformen.teilen('#1'))}
1952 }
```

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

```
1953 \def\liAttributHuelleOhneMathe#1{\text{AttrHülle}(\#1)}
1954 \def\liAttributHuelle#1{
1955   \ifmmode
1956     \liAttributHuelleOhneMathe{\#1}
1957   \else
1958     $\liAttributHuelleOhneMathe{\#1}$
1959   \fi
1960 }
```

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

```
1961 \def\liAttributMenge#1{\{ \textit{\#1} \}}
```

liAHuelle

```
1962 \NewDocumentEnvironment{ liAHuelle } { +b } {
1963   \begin{group}
1964     \footnotesize
1965     \begin{multline*}
1966       \#1
1967     \end{multline*}
1968   \end{group}
1969 } { }
```

\liLinksReduktion Nur innerhalb von liAHuelle zu verwenden bzw. multline

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

```
1970 \def\liLinksReduktion#1#2#3{
1971   \shoveleft{
1972     \liAttributHuelleOhneMathe{FA,
1973       \liAttributMenge{\#1 \string\ #2}} =
1974   } \}
```

```

1975 \shoveright{
1976 \liAttributMenge{#3}
1977 } \\\
1978 }

```

```

\liLinksReduktionInline Let-Abkürzung: \let\ahL=\liLinksReduktionInline
\ahL{ursprüngliche linke Attributmenge}{ohne dieses Attribut}{Ergebnis}
\ahL{A, E}{E}{A, E, F, B, \textbf{D}}: AttrHülle( $F, \{A, E \setminus E\}$ ) =  $\{A, E, F, B, D\}$ 
1979 \def\liLinksReduktionInline#1#2#3{%
1980 {%
1981 \footnotesize%
1982 $\liAttributHuelleOhneMathe{F,
1983 \liAttributMenge{#1 \string\ #2}} =
1984 \liAttributMenge{#3}$
1985 }
1986 }

```

```

\liLinksReduktionInline Let-Abkürzung: \let\ahr=\liLinksReduktionInline
\ahr{gelöschte FA}{neue FA ohne rechts Attribut}{gegebene Attribute}{Ergebnis}
1987 \def\liRechtsReduktionInline#1#2#3#4{%
1988 {%
1989 \footnotesize%
1990 $\liAttributHuelleOhneMathe{
1991 F \setminus
1992 \liFunktionaleAbhaengigkeit{#1}
1993 \def\tmp{#2}\ifx\tmp\empty
1994 \else
1995 \cup \liFunktionaleAbhaengigkeit{#2}
1996 \fi
1997 ,
1998 \liAttributMenge{#3}
1999 } =
2000 \liAttributMenge{#4}$
2001 }
2002 }

```

```

\liFunktionaleAbhaengigkeit Let-Abkürzung: \let\fa=\liFunktionaleAbhaengigkeit
\fa{A, B -> C, D}:  $\{A, B\} \rightarrow \{C, D\}$  Regulärer Ausdruck zum Konvertieren:
\$(.*?) \\\rightarrow (.*)\$ \fa{$1 -> $2}
2003 \def\liFunktionaleAbhaengigkeit#1{%
2004 \directlua{normalformen.drucke_funk_abhaengigkeit('#1', false)}%
2005 }

```

```

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

```

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

```

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

```

```

2006 \NewDocumentCommand {\liFunktionaleAbhaengigkeiten} { O{FA} m } {
2007   \liGeschweifteKlammern
2008   {#1}
2009   {
2010     \begin{align*}
2011       \directlua{normalformen.drucke_funk_abhaengigkeiten('#2')}
2012     \end{align*}
2013   }
2014   {-0.5cm}
2015   {-1.7cm}
2016 }

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

2017 \NewDocumentCommand {\liRelation} { O{R} m } {
2018   $\directlua{
2019     local name = helper.konvertiere_tiefgestellt('#1')
2020     tex.print(name)
2021   }$(\textit{\, #2\,})
2022 }

2023

```

2.28 petri.sty

2024 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2025 \ProvidesPackage{lehramt-informatik-petri}[2020/12/03]

Let-Abkürzungen

\let\t=\liPetriTransitionsName

\let\tp=\liPetriTransPfeile

\let\k=\liPetriErreichKnotenDrei

2026 \RequirePackage{tikz}

2027 \usetikzlibrary{petri,arrows.meta}

Für die Darstellungsmatrix

2028 \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

```
2029 \def\liPetriSetzeSchluessel{%
2030   \def\TmpTransitionOne{}%
2031   \def\TmpTransitionTwo{}%
2032   \def\TmpTransitionThree{}%
2033   \def\TmpTransitionFour{}%
2034   \def\TmpTransitionFive{}%
2035   \def\TmpTransitionSix{}%
2036   \def\TmpTransitionSeven{}%
2037   \def\TmpTransitionEight{}%
2038   \def\TmpTransitionNine{}%
2039   \def\TmpTransitionTen{}%
2040   \pgfkeys{/petri/.cd,
2041     p1/.store in=\TmpPlaceOne,p1/.default=0,p1,
2042     p2/.store in=\TmpPlaceTwo,p2/.default=0,p2,
2043     p3/.store in=\TmpPlaceThree,p3/.default=0,p3,
2044     p4/.store in=\TmpPlaceFour,p4/.default=0,p4,
2045     p5/.store in=\TmpPlaceFive,p5/.default=0,p5,
```

```

2046 p6/.store in=\TmpPlaceSix,p6/.default=0,p6,
2047 p7/.store in=\TmpPlaceSeven,p7/.default=0,p7,
2048 p8/.store in=\TmpPlaceEight,p8/.default=0,p8,
2049 p9/.store in=\TmpPlaceNine,p9/.default=0,p9,
2050 p10/.store in=\TmpPlaceTen,p10/.default=0,p10,
2051 t1/.store in=\TmpTransitionOne,t1/.default=activated,
2052 t2/.store in=\TmpTransitionTwo,t2/.default=activated,
2053 t3/.store in=\TmpTransitionThree,t3/.default=activated,
2054 t4/.store in=\TmpTransitionFour,t4/.default=activated,
2055 t5/.store in=\TmpTransitionFive,t5/.default=activated,
2056 t6/.store in=\TmpTransitionSix,t6/.default=activated,
2057 t7/.store in=\TmpTransitionSeven,t7/.default=activated,
2058 t8/.store in=\TmpTransitionEight,t8/.default=activated,
2059 t9/.store in=\TmpTransitionNine,t9/.default=activated,
2060 t10/.store in=\TmpTransitionTen,t10/.default=activated,
2061 scale/.store in=\TmpScale,scale/.default=0.5,
2062 x/.store in=\TmpX,x/.default=5,
2063 y/.store in=\TmpY,y/.default=5,
2064 }%
2065 }

2066 \tikzset{
2067   li petri/.style={
2068     activated/.style={
2069       very thick
2070     },
2071     inhibitor/.style={
2072       {Circle[open,length=2mm,fill=white]}-
2073     }
2074   }
2075 }

```

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

```

2076 \def\liPetriTransitionsNameOhneMathe#1{t\sb{#1}}
2077 \def\liPetriTransitionsName#1{
2078   \ifmmode
2079     \liPetriTransitionsNameOhneMathe{#1}
2080   \else
2081     $\liPetriTransitionsNameOhneMathe{#1}$
2082   \fi
2083 }

```

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

```

2084 \NewDocumentCommand{ \liPetriErreichTransition } { m m m O{} O{} } {
2085   \draw[->] (#1) edge[#4] node[pos=0.5,auto,sloped,#5]{t\sb{#3}$} (#2);
2086 }

```

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

```

2087 \def\liPetriErreichKnotenDrei#1#2#3{(#1,#2,#3)}

```

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

```

2088 \def\liPetriTransPfeile#1{ $\rightarrow \hspace{0.4cm} \liPetriTransitionsName{#1} \hspace{0.3cm}
2089 }

```

2.29 potenzmengen-konstruktion.sty

```
2090 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2091 \ProvidesPackage{lehramt-informatik-potenzmengen-konstruktion}[2021/02/21 Hilfsmakros
2092 einzusetzen bei der Potenzmengen-Konstruktion (Potenzmengen-Algorithmus)]
```

```
2093 \liLadePakete{formale-sprachen}
2094 \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}
    }
  }
}
```

```
2095 \def\liZustandsMengenSammlung#1#2{
2096   \liZustandsnameGross{#1}
2097   {
2098     \footnotesize
2099     \liPotenzmenge{
2100       \str_case:nn {#1} {#2
2101       }
2102     }
2103 }
```

```
\liZustandsMengenSammlungNr
```

```
2104 \def\liZustandsMengenSammlungNr#1#2{
2105   \liZustandsnameGross{#1}
2106   {
```



```
2107     \footnotesize
2108     \liZustandsmengeNr{
2109         \str_case:nn {#1} #2
2110     }
2111 }
2112 }

2113 \ExplSyntaxOff
2114
```

2.30 pseudo.sty

```

2115 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2116 \ProvidesPackage{lehramt-informatik-pseudo}[2020/12/30 Paket zum Setzen
2117 von Pseudo-Code, Hüll-Paket um algorithm2e]

\begin{algorithm}[H]
\KwData{$G = (V, E, w)$: ein zusammenhängender, ungerichteter,
kantengewichteter Graph  $\text{kruskal}(G)$ }
$E' \leftarrow \emptyset$;
$L \leftarrow E$;
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 $\text{kruskal}(G)$</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;"> 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 then $E' \leftarrow E' \cup \{e\}$; end </div> <p>end</p> <p>Result: $M = (V, E')$ ist ein minimaler Spannbaum von G.</p>

```

2118 \RequirePackage[german,boxruled]{algorithm2e}

```

```

2119

```

2.31 pumping-lemma.sty

2120 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2121 \ProvidesPackage{lehramt-informatik-pumping-lemma}[2021/08/11 Enthält die
 2122 Definitionen für das Pumping-Lemma in der Regulären Sprache und
 2123 in der Kontextfreien Sprache]

\liPumpingRegulaer

```
2124 \def\liPumpingRegulaer{%
2125   Es sei  $L$  eine reguläre Sprache. Dann gibt es eine Zahl  $j$ , sodass für
2126   alle Wörter  $\omega \in L$  mit  $|\omega| \geq j$  (jedes Wort  $\omega$  in
2127    $L$  mit Mindestlänge  $j$ ) jeweils eine Zerlegung  $\omega = uvw$ 
2128   existiert, sodass die folgenden Eigenschaften erfüllt sind:
2129
2130   \begin{enumerate}
2131     \item  $|v| \geq 1$ 
2132     (Das Wort  $v$  ist nicht leer.)
2133
2134     \item  $|uv| \leq j$ 
2135     (Die beiden Wörter  $u$  und  $v$  haben zusammen höchstens die Länge  $j$ .)
2136
2137     \item Für alle  $i = 0, 1, 2, \dots$  gilt  $uv^i w \in L$ 
2138     (Für jede natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^i w$  in der
2139     Sprache  $L$ )
2140   \end{enumerate}
2141
2142   Die kleinste Zahl  $j$ , die diese Eigenschaften erfüllt, wird
2143   Pumping-Zahl der Sprache  $L$  genannt.\footcite{wiki:pumping-lemma}
2144 }
```

\liPumpingKontextfrei

```
2145 \def\liPumpingKontextfrei{%
2146   Es sei  $L$  eine kontextfreie Sprache. Dann gibt es eine Zahl  $j$ , sodass
2147   sich alle Wörter  $\omega \in L$  mit  $|\omega| \geq j$  zerlegen lassen in
2148    $\omega = uvwxy$ , sodass die folgenden Eigenschaften erfüllt sind:
2149
2150   \begin{enumerate}
2151     \item  $|vx| \geq 1$ 
2152     (Die Wörter  $v$  und  $x$  sind nicht leer.)
2153
2154     \item  $|vwx| \leq j$ 
2155     (Die Wörter  $v$ ,  $w$  und  $x$  haben zusammen höchstens die Länge  $j$ .)
2156
2157     \item Für alle  $i \in \mathbb{N}_0$  gilt  $uv^iwx^iy \in L$  (Für jede
2158     natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^iwx^iy$  in der
2159     Sprache  $L$ )
2160   \end{enumerate}
2161 }
```

2162

2.32 quicksort.sty

```

2163 % https://tex.stackexchange.com/a/142634
2164 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2165 \ProvidesPackage{lehramt-informatik-quicksort}[2020/06/12]
2166
2167 %-----
2168 % USAGE:
2169 % \QSinitialize{comma, separated, numerical, values}
2170 % \loop
2171 % \QSpivotStep
2172 % \ifnum\value{pivotcount}>0
2173 %   \QSSortStep
2174 % \repeat
2175 %-----
2176
2177 % xintfrac does not load xinttools, this must be done explicitly if needed as here.
2178 \RequirePackage{xintfrac, xinttools}
2179
2180 \RequirePackage{tikz}
2181
2182 %-----
2183 % FIRST PART: TikZ styles and macros for the actual drawing
2184 \newcounter{cellcount}% used for coordinates of the node
2185 \newcounter{pivotcount}% when it will remain at zero, will signal the sort is finished.
2186
2187 % Styles defined by Tom Bombaldi. (modified: all share the same size)
2188 % (re-modified \bf -> \bfseries due to extremely annoying warnings from
2189 % KOMA-script which are truly a pain and do not make any sense regarding \bf:
2190 % if I want to use \bf, and know what I am doing, why should I get HARASSED
2191 % by police of LaTeX good conduct ? )
2192 \tikzset{
2193   l/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=lime!70!gray},
2194   o/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=olive!50},
2195   r/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2196   b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2197   n/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=white, text=magenta},
2198   g/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=gray, text=white, text=white},
2199 }
2200
2201 % NOTE the b style was originally the same as the r(aised) style apart from
2202 % not being raised, but I find it nicer with a somewhat different
2203 % specification. I have not updated the images though.
2204
2205 % How the nodes are drawn depending on whether on the left of the pivot value
2206 % or on the right, or is a pivot value, or a raised pivot during selection phase.
2207
2208 \def\DecoLEFT #1{%
2209   \xintFor* ##1 in {#1} \do
2210     {\stepcounter{cellcount}\node[o] at (\arabic{cellcount},0) {##1};}%
2211 }
2212
2213 \def\DecoINERT #1{%
2214   \xintFor* ##1 in {#1} \do
2215     {\stepcounter{cellcount}\node[g] at (\arabic{cellcount},0) {##1};}%
2216 }
2217
2218 \def\DecoRIGHT #1{%
2219   \xintFor* ##1 in {#1} \do
2220     {\stepcounter{cellcount}\node[l] at (\arabic{cellcount},0) {##1};}%
2221 }
2222
2223 \def\DecoLEFTwithPivot #1{\stepcounter{pivotcount}%
2224   \xintFor* ##1 in {#1} \do

```

```

2225     {\stepcounter{cellcount}}%
2226     \xintifForLast {\node[r]}\node[o]} at (\arabic{cellcount},0) {##1};}%
2227 }
2228
2229 \def\DecoINERTwithPivot #1{\stepcounter{pivotcount}%
2230     \xintFor* ##1 in {#1} \do
2231     {\stepcounter{cellcount}}%
2232     \xintifForLast {\node[b]}\node[g]} at (\arabic{cellcount},0) {##1};}%
2233 }
2234
2235 \def\DecoRIGHTwithPivot #1{\stepcounter{pivotcount}%
2236     \xintFor* ##1 in {#1} \do
2237     {\stepcounter{cellcount}}%
2238     \xintifForLast {\node[r]}\node[l]} at (\arabic{cellcount},0) {##1};}%
2239 }
2240
2241 %-----
2242 % SECOND PART: the actual sorting routines.
2243
2244 \def\QS@sort@a #1{\expandafter \QS@sort@b \expandafter {\xintLength {#1}}{#1}}
2245 \def\QS@sort@b #1{\ifcase #1
2246     \expandafter\QS@sort@empty
2247     \or\expandafter\QS@sort@single
2248     \else\expandafter\QS@sort@c
2249     \fi
2250 }%
2251 \def\QS@sort@empty #1{}
2252 \def\QS@sort@single #1{\QS@Ir {#1}}
2253
2254 % This step is to pick the last as pivot.
2255 \def\QS@sort@c #1%
2256     {\expandafter\QS@sort@d\expandafter {\romannumeral0\xintnthelt {-1}{#1}}{#1}}%
2257
2258 % Here \QSLr, \QSIr, \QSR have been let to \relax.
2259 % The trick with \xintApplyUnbraced is that for example when selecting
2260 % the elements smaller than pivot, if we had been using \xintApply we
2261 % would have had at the minimum an empty brace pair. Thus we use the
2262 % "unbraced" variant, but then the \QS@select@smaller has added in
2263 % anticipation a level of braces.
2264 \def\QS@sort@d #1#2{%
2265     \QSLr {\xintApplyUnbraced {\QS@select@smaller {#1}}{#2}}}%
2266     \QSIr {\xintApplyUnbraced {\QS@select@equal {#1}}{#2}}}%
2267     \QSRr {\xintApplyUnbraced {\QS@select@greater {#1}}{#2}}}%
2268 }%
2269 \def\QS@select@smaller #1#2{\xintifLt {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2270 \def\QS@select@equal #1#2{\xintifEq {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2271 \def\QS@select@greater #1#2{\xintifGt {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2272
2273 %
2274 % NOTE 1: thus, each comparison with the pivot is done three (!) times.
2275 %
2276 % NOTE 2: we may well end up with \QSLr {<empty>} situations. This is handled
2277 % silently by the \xintFor loops, and also when \QSLr becomes \QS@sort@a, the
2278 % latter must handle correctly an empty argument.
2279
2280 %-----
2281 % THIRD PART: the main macros \QSpivotStep, \QSsortStep and \QS@initialize.
2282
2283 % This draws all with suitable highlighting for the newly chosen pivots
2284 % (which will be shown raised)

```

```

2285 \def\QSpivotStep {\let\QSLr\DecoLEFTwithPivot
2286             \let\QSIr\DecoINERT
2287             \let\QSIrr\DecoINERT
2288             \let\QSRr\DecoRIGHTwithPivot
2289 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2290             \setcounter{cellcount}{0}\setcounter{pivotcount}{0}%
2291             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2292 }
2293
2294 % This sorts and then draws, showing where the pivot chosen in the previous
2295 % step go. Next time they will have become "inert". If pivotcount is still at
2296 % zero on exit from \QSpivotStep, then this is the signal to stop before
2297 % executing \QSSortStep.
2298 \def\QSSortStep {\def\QSLr {\noexpand\QS@sort@a}%
2299             \def\QSRr {\noexpand\QS@sort@a}%
2300             \def\QSIr {\noexpand\QSIrr}%
2301             \let\QSIrr\relax
2302             \edef\QS@list{\QS@list}%
2303             \let\QSLr\relax
2304             \let\QSRr\relax
2305             \let\QSIr\relax
2306             \edef\QS@list{\QS@list}%
2307             \let\QSLr\DecoLEFT
2308             \let\QSIr\DecoINERTwithPivot
2309             \let\QSIrr\DecoINERT
2310             \let\QSRr\DecoRIGHT
2311 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2312             \setcounter{cellcount}{0}%
2313             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2314 }
2315
2316 \def\QSinitialize #1{%
2317     % first, we convert the comma separated values into a list of braced items
2318     % we use an \edef, and anyhow many \edef's will be used later
2319     \edef\QS@list {\noexpand\QSRr {\xintCSVtoList {#1}}}%
2320     \let\QSRr\DecoRIGHT
2321     % The \QSRr marker mutated to draw the last element as
2322     % pivot and the earlier ones with the suitable style.
2323     %
2324     % The list of marked braced items \QS@list is used both for drawing
2325     % (as here) and for doing the exchange of elements during sort.
2326     \par\centerline{\rule[1.5mm]{0pt}{8mm}\setcounter{cellcount}{0}%
2327             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2328 }
2329

```

2.33 relationale-algebra.sty

2330 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2331 \ProvidesPackage{lehramt-informatik-relationale-algebra}[2020/12/11]

2332 \RequirePackage{amsmath}

2333 \RequirePackage{amssymb}

Zum Zeichnen von Operatorenbäumen verwenden wir TikZ

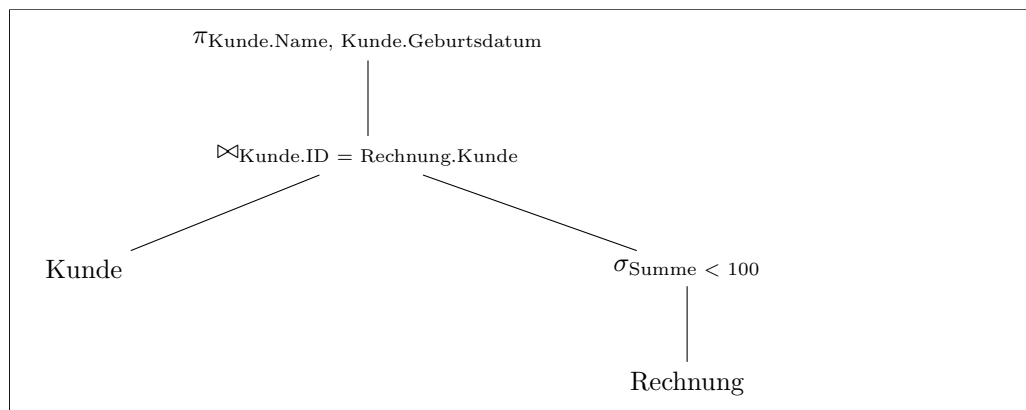
```
\begin{tikzpicture}
  \node
    (pi) {\pi_{\text{Kunde.Name, Kunde.Geburtsdatum}}};

  \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}
```



2334 \RequirePackage{tikz}

2335 \usetikzlibrary{positioning}

Privates Makros, das zwei Querstriche erzeugt.

```
2336 \def\o@join{\setbox0=\hbox{\bowtie}}%
2337 \rule[-.02ex]{.25em}{.4pt}\llap{\rule[\ht0]{.25em}{.4pt}}%
2338 }
```

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

```
2339 \def\leftouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie}}
```

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

```
2340 \def\rightouterjoin{\mathbin{\bowtie\mkern-5.8mu\o@join}}
```

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

```
2341 \def\fullouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie\mkern-5.8mu\o@join}}
```

2342

2.34 rmodell.sty

```

2343 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2344 \ProvidesPackage{lehramt-informatik-rmodell}[2020/09/01]
2345 Makros und Umgebungen zum Setzen des Relationenmodells beim Thema
2346 Datenbanken.]
2347 \RequirePackage{soul}

```

Let-Abkürzungen

```

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

```

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

```

2348 \def\liPrimaer#1{\ul{#1}}

```

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

```

2349 \def\liFremd#1{{\setul{-0.9em}{}\ul{#1}}}

```

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

```

2350 \def\li@Rmodell@Schrift{\footnotesize\ttfamily}
2351 \ExplSyntaxOn
2352 \NewDocumentEnvironment { liRmodell }
2353 { +b }
2354 {
2355   \medskip
2356   {
2357     \linespread{2}
2358     \setlength{\parindent}{0pt}
2359     \li@Rmodell@Schrift#1
2360   }
2361   \medskip
2362 } {}
2363 \ExplSyntaxOff

```

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

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

```

2364 \def\liRelationMenge#1#2{
2365 \noindent
2366 #1 : \[ #2 ]\}
2367 \par
2368 }

```

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

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

```

2369 \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}

```

```

2370 \NewDocumentEnvironment { liRelationenSchemaFormat }{ +b } {} {}

```

```

2371

```


2.35 sortieren.sty

2372 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2373 \ProvidesPackage{lehramt-informatik-sortieren}[2020/06/10 Schaubilder
 2374 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}
}
```



2375 \RequirePackage{tikz}
 2376 \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.

```
2377 \def\liVertauschen#1{
2378   \directlua{
2379     local sortieren = require('lehramt-informatik-sortieren')
2380     sortieren('#1')
2381   }
2382 }
```

\liSortierPfeil

```
2383 \def\liSortierPfeil#1#2{
2384   \draw[-latex] ([xshift=1mm]reihe.#1 north) -- ++(0,0.25) -| ([xshift=-1mm]reihe.#2 north);
2385 }
```

\liSortierPfeilUnten

```
2386 \def\liSortierPfeilUnten#1#2{
2387   \draw[-latex] ([xshift=1mm]reihe.#1 south) -- ++(0,-0.25) -| ([xshift=-1mm]reihe.#2 south);
2388 }
```

\liSortierMarkierung

```
2389 \def\liSortierMarkierung#1#2{\node[
2390   draw,
2391   very thick,
2392   fit=(reihe.#1) (reihe.#2),
2393   inner sep=0pt
2394 ] {}
2395 }

2396 \tikzset{
2397   li sortierung zahlenreihe/.style={
2398     draw,
2399     thin,
2400     font=\large,
2401     rectangle split horizontal,
2402     rectangle split,
2403   }
2404 }
```

```

2405 % https://tex.stackexchange.com/a/140895
2406 \RequirePackage{forest,xstring}
2407 \usetikzlibrary{calc}
2408
2409 \makeatletter
2410 \pgfmathdeclarefunction{strrepeat}{2}{%
2411   \begingroup\pgfmathint{#2}\pgfmath@count\pgfmathresult
2412   \let\pgfmathresult\pgfutil@empty
2413   \pgfutil@loop\ifnum\pgfmath@count>0\relax
2414     \expandafter\def\expandafter\pgfmathresult\expandafter{\pgfmathresult#1}%
2415     \advance\pgfmath@count-1\relax
2416   \pgfutil@repeat\pgfmath@smuggleone\pgfmathresult\endgroup}
2417 \makeatother
2418
2419 \def\myNodes{}
2420
2421 \ExplSyntaxOn
2422 \newcommand*\sortList[1]{%
2423   \clist_sort:Nn#1{\int_compare:nNnTF{##1}>{##2}\sort_return_swapped:\sort_return_same:}}
2424 \ExplSyntaxOff
2425
2426 \forestset{
2427   sort/.code={%
2428     \pgfmathparse{level()>\forestSortLevel}%
2429     \ifnum\pgfmathresult=0
2430       \StrSubstitute{\forestov{content}}{ }{,}{\myList}%
2431       \sortList\myList
2432       \StrSubstitute{\myList}{,}{ }{\myList}%
2433       \pgfmathparse{strrepeat("1",level())}%
2434       \xappto\myNodes{\noexpand\node at ({\forestov{name}}|-m)!-!({\forestov{name}}$)
2435         (m\forestov{name}) {\myList}}%
2436       \pgfmathparse{level()=\forestSortLevel}%
2437       \ifnum\pgfmathresult=1
2438         \forestOget{\forestov{@first}}{name}\forestFirst
2439         \forestOget{\forestov{@last}}{name}\forestLast
2440         \xappto\myNodes{{[<-]edge (\forestOv{\forestov{@first}}{name})
2441           \ifx\forestFirst\forestLast\else edge (\forestOv{\forestov{@last}}{name})\fi}}%
2442       \fi
2443       \ifnum\forestov{@parent}=0\else
2444         \xappto\myNodes{edge (m\forestOv{\forestov{@parent}}{name})}%
2445       \fi
2446       \gappto\myNodes{;}%
2447     \fi}}
2448
2449 \forestset{sort level/.code=%
2450   \pgfmathparse{#1}\let\forestSortLevel\pgfmathresult
2451   \pgfmathparse{strrepeat("1",\forestSortLevel+1)}\let\forestOnes\pgfmathresult}
2452

```

2.36 spalten.sty

```
2453 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2454 \ProvidesPackage{lehramt-informatik-spalten}[2020/12/07 Lädt das Paket
2455 „multicol“, damit mehrspaltiger Satz mit Hilfe der Umgebung „multicols“
2456 realisiert werden kann.]
2457 \RequirePackage{multicol}
```

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

```
2458 \def\liSpaltenUmbruch{\vfill\strut\columnbreak}
```

```
2459
```

2.37 struktogramm.sty

```
2460 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2461 \ProvidesPackage{lehramt-informatik-struktogramm}[2021/01/31 Lädt das
2462 Paket struktex zum Setzen von Struktogrammen]
2463 \RequirePackage{struktex}
2464
```

2.38 syntax.sty

```

2465 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2466 \ProvidesPackage{lehramt-informatik-syntax}[2020/06/10 Ein Hüll-Paket um
2467 `minted`, das sich um die Syntax-Hervorhebung des Java-Codes kümmert.]
2468 \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

```

```

2469 \ExplSyntaxOn
2470 \directlua{
2471   syntax = require('lehramt-informatik-syntax')
2472   syntax.importiere_konfiguration('tex_repo_lokaler_pfad', '\LehramtInformatikRepository')
2473   syntax.importiere_konfiguration('github_domain', '\LehramtInformatikGithubDomain')
2474   syntax.importiere_konfiguration('github_raw_domain', '\LehramtInformatikGithubRawDomain')
2475   syntax.importiere_konfiguration('github_tex_repo', '\LehramtInformatikGithubTexRepo')
2476   syntax.importiere_konfiguration('github_code_repo', '\LehramtInformatikGithubCodeRepo')
2477   syntax.importiere_konfiguration('git_branch', '\LehramtInformatikGitBranch')
2478 }
2479 \RequirePackage{hyperref}
2480 \RequirePackage{minted}
2481 % pygmentize -L styles
2482 \usemintedstyle{colorful}
2483 %\BeforeBeginEnvironment{minted}{\begin{mdframed}}
2484 %\AfterEndEnvironment{minted}{\end{mdframed}}
2485 %\setminted{breaklines=true,linenos}
2486 \setminted{
2487   breaklines=true,
2488   linenos,
2489   fontsize=\footnotesize,
2490 }

```

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

Let-Abkürzung: `\let\j=\liJavaCode`

```

2491 \def\liJavaCode#1{\,\mintinline{java}|#1|\,}

```

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

```

2492 \def\liLatexCode#1{\mintinline{latex}|#1|}

```

```

2493 \def\li@GithubLink#1#2{
2494   \begin{flushright}
2495     \tiny
2496     Code-Beispiel~auf~Github~ansehen:~
2497     \href{#1}{\nolinkurl{#2}}
2498   \end{flushright}
2499 }

```

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

```

2500 \NewDocumentCommand{\liJavaDatei}{ O{firstline=3} m }{
2501   \inputminted[#1]{java}{
2502     \directlua{
2503       syntax.drucke_absoluten_pfad('#2', false)
2504     }
2505   }
2506   \li@GithubLink

```

```

2507     {\directlua{syntax.drucke_github_url('#2', false)}}
2508     {\directlua{syntax.drucke_relativen_repo_pfad('#2', false)}}
2509 }

\liJavaTestDatei Eine komplette Java-Test-Datei einbinden, die Verzeichnis ./Code/src/test/java/org/bschlangaul
liegt.
2510 \NewDocumentCommand{\liJavaTestDatei}{ 0{firstline=3} m }{
2511     \inputminted[#1]{java}{
2512         \directlua{
2513             syntax.drucke_absoluten_pfad('#2', true)
2514         }
2515     }
2516     \li@GithubLink
2517     {\directlua{syntax.drucke_github_url('#2', true)}}
2518     {\directlua{syntax.drucke_relativen_repo_pfad('#2', true)}}
2519 }

\liJavaExamen \liJavaExamen{66116}{2015}{03}{Kunde} Regulärer Ausdruck zum Konvertieren: \liJavaDatei([.
\liJavaExamen$1{$2}{$3}{$4}{$5}
2520 \NewDocumentCommand{\liJavaExamen}{ 0{firstline=3} m m m m }{
2521     \inputminted[#1]{java}{
2522         \directlua{
2523             syntax.drucke_absoluten_examens_pfad('#2', '#3', '#4', '#5')
2524         }
2525     }
2526
2527     \li@GithubLink
2528     {\directlua{syntax.drucke_github_examens_url('#2', '#3', '#4', '#5')}}
2529     {\directlua{syntax.drucke_relativen_examens_repo_pfad('#2', '#3', '#4', '#5')}}
2530 }

\liAssemblerCode
2531 \def\liAssemblerCode#1{\mintinline{asm}|#1|}

\liAssemblerDatei \liAssemblerDatei{relativer-pfad}: Relativer Pfad zu einer Assembler-Datei.
2532 \NewDocumentCommand{\liAssemblerDatei}{ m }{
2533     \inputminted{asm}{#1}
2534 }

\liMinispracheDatei \liMinispracheDatei{relativer-pfad}: Relativer Pfad zu einer Minisprachen-Datei
(Hochsprache für die Minimaschine von Albert Wiedemann).
2535 \NewDocumentCommand{\liMinispracheDatei}{ m }{
2536     \inputminted{componentpascal}{#1}
2537 }

\liHaskellCode \liHaskellCode{haskell}: Zum Setzen von Haskell-Code.
2538 \def\liHaskellCode#1{\mintinline{haskell}|#1|}

\liHaskellDatei \liHaskellDatei{relativer-pfad}: Relativer Pfad zu einer Haskell-Datei.
2539 \NewDocumentCommand{\liHaskellDatei}{ m }{
2540     \inputminted{haskell}{#1}
2541 }

2542 \ExplSyntaxOff

\liSqlCode \liHaskellCode{sql}: Zum Setzen von SQL-Code.
Let-Abkürzung: \let\s=\liSqlCode
2543 \def\liSqlCode#1{\mintinline{sql}|#1|}

2544

```

2.39 syntaxbaum.sty

```
2545 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2546 \ProvidesPackage{lehramt-informatik-syntaxbaum}[2021/02/14 Zum Setzen von
2547 Syntaxbäumen mit Hilfe des Pakets tikz-qtrees]
2548 \RequirePackage{tikz-qtrees}
2549
2550 \tikzset{li parsetree/.style={
2551     every internal node/.style={
2552         draw,circle
2553     },
2554     every leaf node/.style={
2555         draw,rectangle
2556     },
2557 }
2558 }
2559
```

2.40 synthese-algorithmus.sty

```
2560 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2561 \ProvidesPackage{lehramt-informatik-synthese-algorithmus}[2021/03/19
2562 Hilfsmakros zum Setzen des Synthese-Algorithmuses zur Umformung einer
2563 Relation in die 3. Normalform]

2564 \liLadePakete{normalformen,mathe,typographie}
2565 \ExplSyntaxOn
```

Let-Abkürzungen

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

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

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

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

1. Kanonische Überdeckung

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

(a) Linksreduktion

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

(b) Rechtsreduktion

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

(c) Löschen leerer Klauseln

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

(d) Vereinigung

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

2. Relationsschemata formen

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

3. Schlüssel hinzufügen

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

4. Entfernung überflüssiger Teilschemata

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

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

```

2566 \def\liSyntheseUeberschrift#1{
2567   {
2568     \bfseries
2569     \sffamily
2570     \str_case:nn {#1} {
2571       {1} {Kanonische-Überdeckung}
2572       {1-1} {Linksreduktion}
2573       {1-2} {Rechtsreduktion}
2574       {1-3} {Löschen-leerer-Klauseln}
2575       {1-4} {Vereinigung}
2576       {2} {Relationsschemata-formen}
2577       {3} {Schlüssel-hinzufügen}
2578       {4} {Entfernung-überflüssiger-Teilschemata}
2579     }
2580   }
2581 }

```

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

```

2582 \def\li@synthese@erklaerung@texte#1{
2583   \str_case:nn {#1} {
2584     {1} {
2585       Die-kanonische-Überdeckung---also-die-kleinst-mögliche-noch-
2586       äquivalente-Menge-von-funktionalen-Abhängigkeiten-kann-in-vier-
2587       Schritten-erreicht-werden.
2588     }
2589     {1-1} {
2590       Führe-für-jede-funktionale-Anhängigkeit-
2591       $\alpha\rightarrow\beta$-in-F$-die-Linksreduktion-durch,-
2592       überprüfe-also-für-alle-
2593       $A$-in-$\alpha$,~ob-$A$-überflüssig-ist,~d.h.-ob-
2594       $\beta\subseteqq\liAttributHuelle{F,\alpha\---A}$.
2595     }
2596     {1-2} {
2597       Führe-für-jede-(verbliebene)-funktionale-Abhängigkeit-$\alpha\rightarrow\beta$-die-Rechtsreduktion-durch,-überprüfe-also-für-
2598       alle-$B$-in-$\beta$,~ob-$B$-in-$\liAttributHuelle{F---(\alpha\rightarrow\beta)\cup(\alpha\rightarrow\beta)}$,~
2599       $\alpha$-gilt.-In-diesem-Fall-ist-B-auf-der-rechten-Seite-
2600       überflüssig-und-kann-eliminiert-werden,~dh-$\alpha\rightarrow\beta$-wird-durch-$\alpha\rightarrow(\beta\---B)$-
2601       ersetzt.
2602     }
2603     {1-3} {
2604       Entferne-die-funktionalen-Abhängigkeiten-der-Form-$\alpha\rightarrow\emptyset$,~die-im-2.-Schritt-möglicherweise-
2605       entstanden-sind.
2606     }
2607     {1-4} {
2608       Fasse-mittels-der-Vereinigungsregel-funktionale-Abhängigkeiten-
2609       der-Form-$\alpha\rightarrow\beta_{\{1\}},\dots,\alpha\rightarrow\beta_{\{n\}}$,~so-dass-$\alpha\rightarrow\beta_{\{1\}}\cup\dots\cup\beta_{\{n\}}$-verbleibt.
2610     }
2611   }
2612   % Kemper Seite 197
2613   {2} {
2614     Erzeuge-für-jede-funktionale-Abhängigkeit-$\alpha\rightarrow\beta$-in-F$\sb{c}$-ein-Relationenschema-$\mathcal{R}\sb{\alpha}$-
2615

```

```

2621      :=~\alpha~\cup~\beta$.
2622    }
2623    {3} {
2624      Falls-eines~der~in~Schritt~2.~erzeugten~Schemata~ $R\sb{\alpha}$ ~
2625      einen~Schlüsselkandidaten~von~ $\mathcal{R}$ ~bezüglich~ $F\sb{c}$ ~
2626      enthält,~sind~wir~fertig,~sonst~wähle~einen~Schlüsselkandidaten~
2627       $\mathcal{K}~\subseq \mathcal{R}$ ~aus~und~definiere~folgendes~
2628      zusätzliche~Schema:~ $\mathcal{R}\sb{\mathcal{K}}~:=~\mathcal{K}$ ~
2629      und~ $\mathcal{F}\sb{\mathcal{K}}~:=~\emptyset$ 
2630    }
2631    {4} {
2632      Eliminiere~diejenigen~Schemata~ $R\sb{\alpha}$ ,~die~in~einem~
2633      anderen~Relationenschema~ $R\sb{\alpha'}$ ~enthalten~sind,~d.h.~
2634       $R\sb{\alpha}~\subseq R\sb{\alpha'}$ .
2635    }
2636  }
2637 }
2638 \def\liSyntheseErklaerung#1{
2639   {
2640     \itshape
2641     \footnotesize
2642     \liParagraphMitLinien{\li@synthese@erklaerung@texte{#1}}
2643   }
2644 }

```

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

```

2645 \def\liSyntheseUeberErklaerung#1{
2646   \liSyntheseUeberschrift{#1}\par
2647   \liSyntheseErklaerung{#1}
2648 }

2649 \ExplSyntaxOff
2650

```

2.41 tabelle.sty

```
2651 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2652 \ProvidesPackage{lehramt-informatik-tabelle}[2020/12/05 Lädt das Paket tabluarx]
2653 \RequirePackage{tabularx}
2654
```

2.42 typographie.sty

```
2655 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2656 \ProvidesPackage{lehramt-informatik-typographie}[2021/03/20 Typographische Makros,
2657 die das Erscheinungsbild verändern. Die Schriftdefinition sind in
2658 formatierung.sty definiert.]
```

```
2659 \ExplSyntaxOn
```

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

```
2660 \RequirePackage{fontawesome}
```

```
\liErledigt \liErledigt: ☑
```

```
2661 \let\liErledigt=\faCheckSquareO
```

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

```
2662 \def\liNichtsZuTun{\emptyset$~Nichts~zu~tun}
```

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

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

```
2663 \def\liParagraphMitLinien#1{
```

```
2664 \noindent
```

```
2665 \vrule height 2pt depth -1.6pt width 0.4cm
```

```
2666 \enspace
```

```
2667 #1
```

```
2668 \enspace
```

```
2669 \leaders\vrule height 2pt depth -1.6pt \hfill \null
```

```
2670 \par
```

```
2671 \medskip
```

```
2672 }
```

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

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

```
2673 \def\liGeschweifteKlammern#1#2#3#4{
```

```
2674 \par
```

```
2675 \medskip
```

```
2676 \noindent
```

```
2677 #1 \, $= \Bigl\{
```

```
2678 \vspace{#3}
```

```
2679 #2
```

```
2680 \vspace{#4}
```

```
2681 \begin{flushright}$\Bigr\}$\end{flushright}
```

```
2682 \par
```

```
2683 }
```

```
2684 \ExplSyntaxOff
```

```
2685
```

2.43 uml.sty

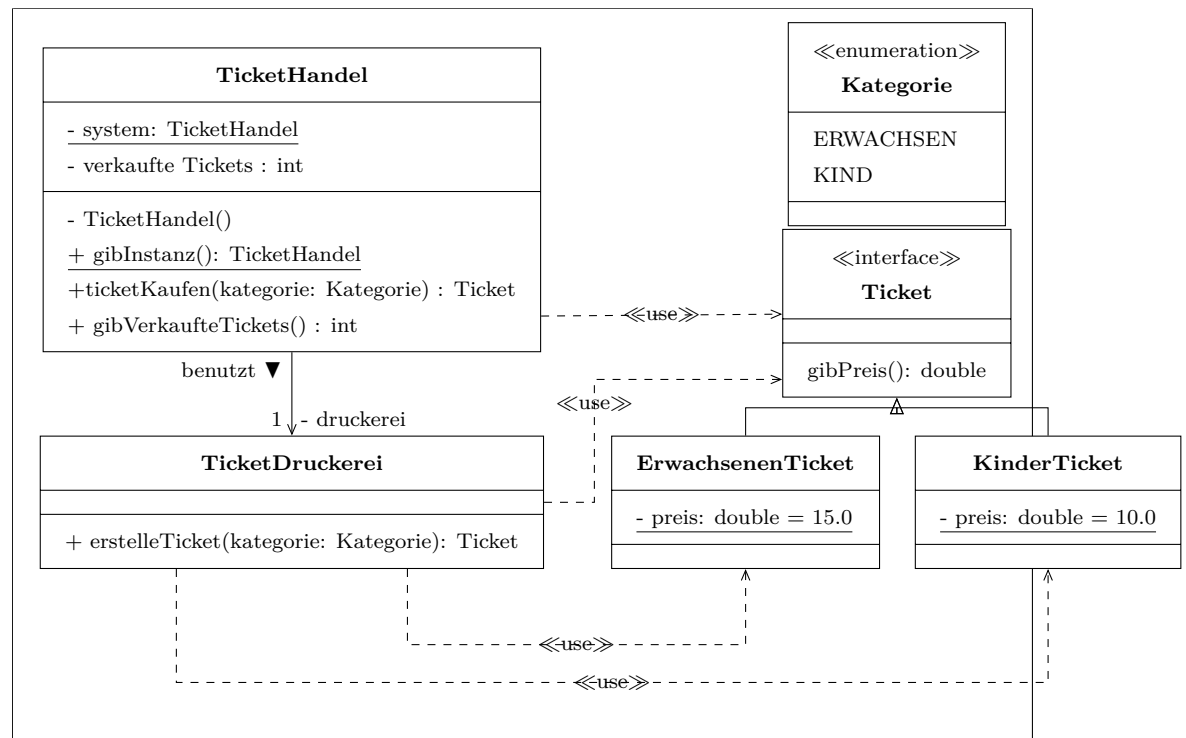
```

2686 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2687 \ProvidesPackage{lehramt-informatik-uml}[2020/06/13 Ein Hüll-Paket um
2688 `tikz-uml', das einige Design-Einstellungen vornimmt und manche
2689 Erweiterung bereitstellt]

2690 \RequirePackage{tikz-uml}
2691 \RequirePackage{tikz-uml-activity}
2692 % Not compatible with wasysym
2693 %\RequirePackage{mathabx}
2694 \RequirePackage{wasysym}
2695 \usetikzlibrary{positioning}

2696 \tikzumlset{
2697   fill class=white!0,
2698   font=\footnotesize,
2699   fill object=white!0,
2700   fill note=white!0,
2701   fill state=white!0,
2702   % Use case
2703   fill usecase=white!0,
2704   fill system=white!0,
2705 }

```



```

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

```

```

2706 \NewDocumentCommand{ \liUmlLeserichtung } { 0{dir=right} m } {
2707   \def\@liDirLeft{}
2708   \def\@liDirRight{ \RIGHTarrow}
2709   \pgfkeys{/lese/dir/.is choice}
2710   \pgfkeys{/lese/dir/up/.code={\def\@liDirRight{ \UParrow}}}
2711   \pgfkeys{/lese/dir/down/.code={\def\@liDirRight{ \DOWNarrow}}}
2712   \pgfkeys{/lese/dir/left/.code={\def\@liDirRight{}\def\@liDirLeft{\LEFTarrow }}}
2713   \pgfkeys{/lese/dir/right/.code={\def\@liDirRight{ \RIGHTarrow}}}
2714
2715   \def\@liPos{above}
2716   \pgfkeys{/lese/pos/.code={\def\@liPos{##1}}}
2717

```

```

2718 \def\@liDistance{0cm}
2719 \pgfkeys{/lese/distance/.code={\def\@liDistance{##1}}}}
2720
2721 \pgfkeys{/lese/.cd,#1}
2722
2723 \node[\@liPos = \@liDistance of #2-middle] {
2724   \@liDirLeft{\footnotesize#2}\@liDirRight
2725 };
2726 }
2727

```

2.44 vollstaendige-induktion.sty

2728 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2729 \ProvidesPackage{lehramt-informatik-vollstaendige-induktion}[2021/07/01
 2730 Hilfsmakros zum Setzen der Vollständigen Induktion, vor allem die
 2731 Überschriften für die einzelnen Schritte]

Let-Abkürzungen

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

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

Lade häufig benötigte Pakete

2732 \RequirePackage{lehramt-informatik-typographie}
 2733 \RequirePackage{lehramt-informatik-mathe}
 2734 \RequirePackage{lehramt-informatik-syntax}
 2735 \ExplSyntaxOn

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

Let-Abkürzung: \let\m=\liInduktionMarkierung

2736 \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

```
2737 \def\liInduktionErklaerung#1{\scriptsize\text{#1}}
```

\liInduktionAnfang

```
2738 \def\liInduktionAnfang{
```

```
2739   \liPseudoUeberschrift{Induktionsanfang}
```

```
2740
```

```
2741   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
2742   \liParagraphMitLinien{
```

```
2743     Beweise,~dass~ $A(1)$ ~eine~wahre~Aussage~ist.
```

```
2744   }
```

```
2745 }
```

\liInduktionVoraussetzung

```
2746 \def\liInduktionVoraussetzung{
```

```
2747   \liPseudoUeberschrift{Induktionsvoraussetzung}
```

```
2748
```

```
2749   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
2750   \liParagraphMitLinien{
```

```
2751     Die~Aussage~ $A(k)$ ~ist~wahr~für~ein~beliebiges~ $k \in \mathbb{N}$ ~.
```

```
2752   }
```

```
2753 }
```

\liInduktionSchritt

```
2754 \def\liInduktionSchritt{
```

```
2755   \liPseudoUeberschrift{Induktionsschritt}
```

```
2756
```

```
2757   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
2758   \liParagraphMitLinien{
```

```
2759     Beweise,~dass~wenn~ $A(n=k)$ ~wahr~ist,~
```

```
2760     auch~ $A(n=k+1)$ ~wahr~sein~muss.
```

```
2761   }
```

```
2762 }
```

```
2763 \ExplSyntaxOff
```

```
2764
```


2.45 wasserfall.sty

```
2765 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2766 \ProvidesPackage{lehramt-informatik-wasserfall}[2020/06/10]
2767 \RequirePackage{tikz}
2768 \tikzset{wasserfall/.style={
2769   >=stealth,
2770   node distance = 2mm and -8mm,
2771   start chain = A going below right,
2772   every node/.style = {
2773     draw,
2774     text width=24mm,
2775     minimum height=12mm,
2776     align=center,
2777     inner sep=1mm,
2778     fill=white,
2779     drop shadow={fill=black},
2780     on chain=A
2781   },
2782 }}
2783 \usetikzlibrary{chains,positioning,shadows}
2784
```

2.46 wpkalkuel.sty

2785 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2786 \ProvidesPackage{lehramt-informatik-wpkalkuel}[2020/06/13]

Let-Abkürzungen

\let\wp=\liWpKalkuel

\let\equivalent=\liWpEquivalent

\let\erklaerung=\liWpErklaerung

2787 \RequirePackage{amsmath}

2788 \ExplSyntaxOn

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

2789 \def\liWpKalkuelOhneMathe#1#2{

2790 \text{wp}(\texttt{\scriptsize"#1"},\thinspace #2)

2791 }

2792 \def\liWpKalkuel#1#2{

2793 \ifmmode

2794 \liWpKalkuelOhneMathe{#1}{#2}

2795 \else

2796 \$\liWpKalkuelOhneMathe{#1}{#2}\$

2797 \fi

2798 }

\MatheEnv

2799 \def\MatheEnv#1{

2800 \medskip

2801

2802 \hspace{1em}#1

2803

2804 \medskip

2805 }

\Mathe

2806 \def\Mathe#1{

2807 \MatheEnv{#1\$}

2808 }

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

2809 \def\liWpEquivalent#1{

2810 \MatheEnv{\$\equiv\$\hspace{1em}#1\$}

2811 }

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

2812 \newlength{\@Skip@Erklaerung@Reset}

2813 \def\liWpErklaerung#1{

2814 \setlength{\@Skip@Erklaerung@Reset}{\leftskip}

2815 \setlength{\leftskip}{0.5cm}

2816

2817 \par

2818 \noindent

2819 {

2820 \scriptsize

2821 #1

2822 }

2823 \par

2824

2825 \setlength{\leftskip}{\@Skip@Erklaerung@Reset}

2826 }

\liWpErklaerungVerzweigung

```

2827 \def\liWpErklaerungVerzweigung{
2828   $\liWpKalkuelOhneMathe{if~\{-b~\}~then~\{-a1~\}~else~\{-a2~\}}{Q}
2829   \equiv
2830   (b \land \liWpKalkuelOhneMathe{a1}{Q})
2831   \lor
2832   (\neg b \land \liWpKalkuelOhneMathe{a2}{Q})$
2833 }

2834 \ExplSyntaxOff

2835

```

3 Index

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

Symbols	\AfterEndEnvironment 2484	2196, 2198, 2199, 2568
\# 109	\allsectionsfont ... 1231	\Bigl 2677
\, 343, 398, 1182,	\Alph 1245	\Bigr 2681
1184, 1685, 1686,	\alph 1245, 1246	\bigskip 48, 376,
1687, 2021, 2491, 2677	\alpha 2591, 2593, 2594,	610, 615, 1561, 1889
\@Skip@Erklaerung@Reset	2597, 2599, 2600,	\bool 321, 344
... 2812, 2814, 2825	2601, 2602, 2603,	\bowtie 2336, 2339, 2340, 2341
\@afterheading 1652	2607, 2613, 2614,	\Box 159
\@afterindentfalse . 1652	2619, 2620, 2621,	\boxtimes 475
\@liDirLeft 2707, 2712, 2724	2624, 2632, 2633, 2634	
\@liDirRight 2708, 2710,	\arabic 1245, 2210, 2215,	C
2711, 2712, 2713, 2724	2220, 2226, 2232, 2238	\c 1328, 1329
\@liDistance 2718, 2719, 2723	\arraystretch 1911	\cdot 1729, 1782, 1793
\@liPos .. 2715, 2716, 2723	B	\centerline 1359, 2289, 2311, 2326
\ 608,	\BeforeBeginEnvironment	\chapter 1235, 1236
631, 632, 635, 636, 2483	\char 1559
639, 640, 732, 733,	\begin .. 629, 680, 695,	\clearpage 1553
734, 841, 870, 872,	730, 754, 801, 833,	\cline 608
898, 907, 952, 994,	848, 868, 878, 894,	\clist 226, 270,
995, 996, 1001,	914, 946, 963, 992,	271, 290, 296, 2423
1002, 1003, 1023,	1017, 1038, 1053,	\columnbreak 2458
1559, 1916, 1974, 1977	1162, 1251, 1326,	\cs 293, 318, 342,
\{ 207, 1119,	1354, 1363, 1370,	343, 380, 392, 1635
1129, 1141, 1142,	1478, 1551, 1603,	\csname 1313, 1316
1147, 1181, 1400,	1608, 1616, 1641,	\cup 1142,
1961, 2366, 2677, 2828	1646, 1657, 1667,	1995, 2600, 2614, 2621
\} 207, 1119,	1671, 1754, 1758,	
1129, 1141, 1142,	1774, 1797, 1820,	D
1149, 1185, 1401,	1835, 1914, 1915,	\DeclareMathSymbol ..
1961, 2366, 2681, 2828	1965, 2010, 2130, 1859, 1860
_ 27, 35, 44, 46, 293, 318,	2150, 2291, 2313,	\DecoINERT 2213, 2286, 2287, 2309
342, 343, 357, 358,	2327, 2483, 2494, 2681	\DecoINERTwithPivot .
364, 367, 370, 380, 392	\begingroup 1552, 1963, 2411 2229, 2308
	\beschriftung 1583, 1587,	\DecoLEFT 2208, 2307
	1591, 1595, 1599, 1601	\DecoLEFTwithPivot ..
\sqcup 1973, 1983	\beta 2591, 2223, 2285
A	2594, 2598, 2599,	\DecoRIGHT 2218, 2310, 2320
\addbibresource 1529, 1530, 1531,	2600, 2603, 2613,	\DecoRIGHTwithPivot .
1532, 1533, 1534,	2614, 2615, 2620, 2621 2235, 2288
1535, 1536, 1537, 1538	\bf 2188, 2189, 2190	\definecolor 1233
\advance 2415	\bfseries .. 487, 1235,	\delta 65, 107, 165, 207, 1133
	1237, 2188, 2194,	

2147, 2157, 2591, 2593, 2599, 2620, 2751	\inhaltsverzeichnis 1550	\labelitemiv 1243	\li@synthese@erklaerung@texte 2582, 2642
\input . 4, 7, 10, 13, 16, 406	\inputminted 2501, 2511, 2521, 2533, 2536, 2540	\land 2830, 2832	\liAbleitung 1156
\int 2423	\item 475, 476, 697, 701, 706, 711, 755, 764, 769, 777, 849, 854, 858, 879, 915, 920, 927, 935, 964, 969, 973, 978, 1054, 1059, 1064, 1364, 1365, 1635, 1639, 1759, 1762, 1766, 1775, 1781, 1786, 1798, 1802, 1806, 1810, 1814, 1821, 1825, 1829, 2131, 2134, 2137, 2151, 2154, 2157	\large 1359, 2400	liAdditum (environment) 1606
\itshape 531, 2640		\left 1694	liAHuelle (environment) 1962
J		\LEFTarrow 2712	\liAlphabet 1141
\j 1328, 1329, 1331, 1332, 1333, 1338, 1339, 1340		\leftarrow 594	liAntwort (environment) 1577
K		\leftouterjoin 2339	\liAnweisung 1485
\k 1338		\leftskip 2814, 2815, 2825	\liAssemblerCode ... 2531
\keys 31, 70, 82, 112, 122, 170, 180, 298, 550, 554, 568, 573, 1206, 1213		\LehramtInformatikAutorEmail 1510	\liAssemblerDatei .. 2532
L		\LehramtInformatikAutorName 1509	\liAttribut 2369
\l 63, 64, 65, 66, 67, 68, 71, 72, 73, 74, 75, 77, 79, 84, 85, 86, 87, 88, 89, 104, 105, 106, 107, 108, 109, 110, 113, 114, 115, 116, 117, 118, 119, 125, 126, 127, 128, 129, 130, 131, 162, 163, 164, 165, 166, 167, 168, 171, 172, 173, 174, 175, 176, 177, 183, 184, 185, 186, 187, 188, 189, 548, 551, 556, 557, 560, 565, 566, 569, 570, 575, 1201, 1202, 1203, 1204, 1207, 1208, 1209, 1210, 1216, 1217, 1218, 1219, 1494, 1495, 1496, 1638, 1639, 1640, 1647		\LehramtInformatikGitBranch 385, 2477	\liAttributHuelle ... 1953, 2594, 2599
		\LehramtInformatikGithubCodeRepo 2476	\liAttributHuelleOhneMathe 1953, 1956, 1958, 1972, 1982, 1990
		\LehramtInformatikGithubDomain 2473	\liAttributMenge
		\LehramtInformatikGithubRawDomain 383, 2474	\liAufgabe 3
		\LehramtInformatikGithubTexRepo 384, 2475	\liAufgabenMetadaten . 25
		\LehramtInformatikRepository 4, 7, 10, 13, 16, 1529, 1530, 1531, 1532, 1533, 1534, 1535, 1536, 1537, 1538, 2472	\liAufgabenTitel 23
		\LehramtInformatikTitel 1505	\liAusdruck 1178
		\leq 1793, 2134, 2154	\liAutomat 61
		\let 1131, 1132, 1553, 2285, 2286, 2287, 2288, 2301, 2303, 2304, 2305, 2307, 2308, 2309, 2310, 2320, 2412, 2450, 2451, 2661	\liAutomatenKante 93
		\li@chomsky@erklaerung@texte 497, 533	\liBandAlphabet 1142
		\li@EntwurfsCode	\liBedingung 1486
		625, 671, 672, 673, 719, 720, 721, 722, 788, 789, 790, 791, 792, 793, 820, 821, 822, 823, 824, 825, 886	\liBedingungDrei 1749, 1789, 1829
		\li@EntwurfsCodeAllgemein 624	\liBedingungEins
		\li@fussnote@text 1867, 1873, 1877, 1881, 1885	\liBedingungFalsch . 1488
		\li@GithubLink 2493, 2506, 2516, 2527	\liBedingungWahr ... 1487
		\li@mget . 1315, 1319, 1339	\liBedingungZwei
		\li@minc 1318, 1340	\liBeschriftung 1566
		\li@mset 1312, 1320, 1329, 1332	\liChomskyErklaerung
		\li@numdiscs 1324, 1333, 1339 497, 538
\labelenumi 1246		\li@Rmodell@Schrift .	\liChomskyUeberErklaerung
\labelenumii 1247	 2350, 2359, 2369 536
\labelitemi 1240		\li@sequence .. 1325, 1338	\liChomskyUeberschrift
\labelitemii 1241		 485, 537
\labelitemiii 1242			\liCpmEreignis 546
			\liCpmFruehsterI ... 603
			\liCpmSpaetesterI ... 602
			\liCpmVon 586
			\liCpmVonOhneMathe ..
		 586, 589, 591
			\liCpmVonZu 578
			\liCpmVonZuOhneMathe
		 578, 581, 583
			\liCpmVorgang 563
			\liCpmZu 594
			\liCpmZuOhneMathe ...
		 594, 597, 599
			liDiagramm (environ- ment) 1665
			liEinbettung (environ- ment) 1576
			\liEntwurfs 1033

\liEntwurfsAbstrakteFabrik	\liEntwurfsModellPraesentation	\liKellerKante	102
..... 675 1016, 1034	\liKellerKante	140
\liEntwurfsAbstrakteFabrikCode	\liEntwurfsZustand	\liKellerUebergang	135, 141
..... 670, 677	\liEntwurfsZustandAkteure	\liKontrollCode	1489
\liEntwurfsAbstrakteFabrikUml 1052, 1072	\liKontrollflussgraph	1477
..... 628, 676	\liEntwurfsZustandUml	(environment)	1491
\liEntwurfsAdapter 1037, 1071	\liKontrollKnotenPfad	1491
\liEntwurfsAdapterAkteure	\liEpsilon	\liKontrollTextzeileKnoten	1490, 1495
..... 694, 726	\liErAttribute	\liKurzeTabellenLinie	608
\liEntwurfsAdapterCode	... 1085, 1099, 1101	\liLadeAllePakete	228
..... 718, 727	\liErDatenbankName	\liLadePakete	54, 57, 224, 229, 484, 545, 1118, 1349, 1865, 1945, 2093, 2564
\liEntwurfsAdapterUml	\liErEntity	\liLatexCode	2492
..... 679, 725	\liErledigt	\liLeereZelle	1903
\liEntwurfsBeobachter 2661	\liLernkartei (environ-	1655
\liEntwurfsBeobachterAkteure	\liErMpAttribute	ment)	1970
..... 753, 797	... 1098	\liLinksReduktion	1970
\liEntwurfsBeobachterCode	\liErMpEntity	\liLinksReduktionInline	1979, 1987
..... 787, 798	\liErMpRelationship	\liMasterExkurs	1834
\liEntwurfsBeobachterUml	\liErRelationship	\liMasterFaelle	1773, 1841
..... 729, 796	... 1084, 1093, 1095	\liMasterFallRechnung	1819
\liEntwurfsDekorierer	\liExamensAufgabe	\liMasterVariablen	1753, 1836
\liEntwurfsDekoriererAkteure	\liExamensAufgabeA	\liMasterVariablenDeklaration	1796
..... 829	\liExamensAufgabeTA	\liMasterWolframLink	1844
\liEntwurfsDekoriererCode	\liExamensAufgabeTTA	\liMenge	71, 72, 74, 113, 114, 115, 119, 171, 172, 173, 177, 1119, 1168, 1207, 1208
..... 819, 830	\liExkurs (environment)	\liMinimierungErklaerung	1925
\liEntwurfsDekoriererUml	\liFalsch	\liMinispracheDatei	2535
..... 800, 828	\liFlaci	\liLinespread	2357
\liEntwurfsEinfacheFabrik	\liFremd	\liNichtsZuTun	2662
..... 863	\liFunktionaleAbhaengigkeit	\liO	1716, 1744
\liEntwurfsEinfacheFabrikAkteure	... 1992, 1995, 2003	\liOmega	1706, 1750
..... 847, 865	\liFunktionaleAbhaengigkeiten	\liOmegaOhneMathe	1706, 1711, 1713
\liEntwurfsEinfacheFabrikUml	\liFussnote	\liOOhneMathe	1716, 1721, 1723
..... 832, 864	\liFussnoteDreiText	\liParagraphMitLinien	533, 1927, 2642, 2663, 2742, 2750, 2758
\liEntwurfsEinzelstueck 1880, 1898	\liPetriErreichKnotenDrei	2087
..... 888	\liFussnoteEinsText	\liPetriErreichTransition	2084
\liEntwurfsEinzelstueckAkteure 1872, 1892	\liPetriSetzeSchluessel	2029
..... 877, 890	\liFussnoteLink	\liPetriTransitionsName	2076, 2088
\liEntwurfsEinzelstueckCode 1682	\liPetriTransitionsNameOhneMathe	2076, 2079, 2081
..... 885, 891	\liFussnoten		
\liEntwurfsEinzelstueckUml	\liFussnoteUrl		
..... 867, 889	... 1031, 1678		
\liEntwurfsErbauer	\liFussnoteVierText		
\liEntwurfsErbauerAkteure 1884, 1901		
..... 913, 943	\liFussnoteZweiText		
\liEntwurfsErbauerUml 1876, 1895		
..... 893, 942	\liGeschweifteKlammern		
\liEntwurfsFabrikmethode	... 1160, 2007, 2673		
..... 987, 1012	\liGrammatik		
\liEntwurfsFabrikmethodeAkteure 1199		
..... 962, 989	\liGraphenFormat (envi-		
\liEntwurfsFabrikmethodeUml	ronment)		
..... 945, 988	... 1305		
\liEntwurfsKompositum	\liHanoi		
..... 1012 1312		
\liEntwurfsKompositumAkteure	\liHaskellCode		
..... 1014 2538		
\liEntwurfsKompositumUml	\liHaskellDatei		
..... 991, 1013	... 2539		
\liEntwurfsModellPraesentation	\liInduktionAnfang		
..... 1033	... 2738		
\liEntwurfsModellPraesentation	\liInduktionErklaerung		
..... 1035 2737		
	\liInduktionMarkierung		
 2736		
	\liInduktionSchritt		
 2754		
	\liInduktionVoraussetzung		
 2746		
	\liJavaCode		
 2491		
	\liJavaExamen		
 2520		
	\liJavaTestAkteure		
 2510		
	\liKasten (environment)		
 1250		

<code>\liPetriTransPfeile</code>	2088	<code>\liTheta</code>	1696 , 1747 , 1776 , 1782 , 1787	<code>\loop</code>	2170
<code>\liPolynomiellReduzierbar</code>	1369	<code>\liThetaOhneMathe</code>	1696 , 1701 , 1703	<code>\lor</code>	2831
<code>\liPotenzmenge</code>	1128 , 1132 , 2099	<code>\liTOhneMathe</code>	1726 , 1735 , 1737	<code>\ltimes</code>	1852
<code>\liPotenzmengeOhneMathe</code>	1129 , 1130 , 1131	<code>\liTuringKante</code>	203	M	
<code>\liPrimaer</code>	2348	<code>\liTuringLeerzeichen</code>	159 , 167		
<code>\liProblemBeschreibung</code>	1353	<code>\liTuringMaschine</code>	160	<code>\makeatletter</code>	1651 , 2409
<code>\liProblemClique</code>	1376	<code>\liTuringUeberfuehrung</code>	206	<code>\makeatother</code>	1653 , 2417
<code>\liProblemName</code>	1352 , 1359 , 1371 , 1373 , 1386 , 1397 , 1398 , 1406 , 1407	<code>\liTuringUebergaenge</code>	198 , 204	<code>\marginpar</code>	1088 , 1094 , 1100 , 1558
<code>\liProblemSat</code>	1405	<code>\liTuringUebergangZelle</code>	193	<code>\mathbb</code>	1407 , 2157 , 2751
<code>\liProblemSubsetSum</code>	1396 , 1405	<code>\liUeberfuehrungsFunktion</code>	1133	<code>\mathbin</code>	2339 , 2340 , 2341
<code>\liProblemVertexCover</code>	1376 , 1384	<code>\liUeberfuehrungsFunktionOhneMathe</code>	1133 , 1136 , 1138	<code>\mathcal</code>	1717 , 2620 , 2625 , 2627 , 2628 , 2629
<code>\liProduktionen</code>	1167 , 1209	<code>liUebergangsTabelle</code>	1911	<code>\Mathe</code>	2806
<code>liProduktionsRegeln</code>	(environment) 1157	<code>\liUeberschriftDreiecksTabelle</code>	1921	<code>\MatheEnv</code>	2799 , 2807 , 2810
<code>liProjektSprache</code>	(environment) 1575	<code>\liUmlLeserichtung</code>	2706	<code>\mathord</code>	1859 , 1860
<code>\liPseudoUeberschrift</code>	1560 , 1610 , 1611 , 1913 , 1923 , 2739 , 2747 , 2755	<code>\liVertauschen</code>	2377	<code>\mdfsetup</code>	1249 , 1584 , 1588 , 1592 , 1596
<code>\liPumpingKontextfrei</code>	2145	<code>\liWortInSprache</code>	609	<code>\medskip</code>	1361 , 1569 , 1571 , 1645 , 1673 , 2355 , 2361 , 2671 , 2675 , 2800 , 2804
<code>\liPumpingRegulaer</code>	2124	<code>\liWortNichtInSprache</code>	614	<code>\emph</code>	1558
<code>liQuellen</code>	(environment) 1635	<code>\liWpEquivalent</code>	2809	<code>\mintinline</code>	2491 , 2492 , 2531 , 2538 , 2543
<code>\liRechtsReduktionInline</code>	1987	<code>\liWpErklaerung</code>	2812	<code>\mkern</code>	2339 , 2340 , 2341
<code>\liRekursionsGleichung</code>	1740 , 1800	<code>\liWpErklaerungVerzweigung</code>	2827	<code>\mlq</code>	1857 , 1859
<code>\liRelation</code>	2017	<code>\liWpKalkuel</code>	2789	<code>\mrq</code>	1857 , 1860
<code>liRelationenSchemaFormat</code>	(environment) 2370	<code>\liWpKalkuelOhneMathe</code>	2789 , 2794 , 2796 , 2828 , 2830 , 2832	<code>\msg</code>	39 , 403
<code>\liRelationMenge</code>	2364	<code>\liZustandsBuchstabe</code>	1143 , 1152 , 1154 , 1172 , 1174	<code>\myList</code>	2430 , 2431 , 2432 , 2435
<code>\liRichtig</code>	475	<code>\liZustandsBuchstabeGross</code>	1144 , 1153 , 1155	<code>\myNodes</code>	2419 , 2434 , 2440 , 2444 , 2446
<code>liRmodell</code>	(environment) 2350	<code>\liZustandsmenge</code>	1131	N	
<code>\liRundeKlammer</code>	1693 , 1697 , 1707 , 1717 , 1731	<code>\liZustandsmengeNr</code>	1145 , 2108		
<code>\liSortierMarkierung</code>	2389	<code>\liZustandsmengeNrGross</code>	1153	<code>\NeedsTeXFormat</code>	1 , 19 , 52 , 220 , 410 , 471 , 478 , 542 , 605 , 620 , 1075 , 1111 , 1224 , 1264 , 1273 , 1278 , 1307 , 1345 , 1417 , 1500 , 1515 , 1521 , 1543 , 1689 , 1849 , 1862 , 1941 , 2024 , 2090 , 2115 , 2120 , 2164 , 2330 , 2343 , 2372 , 2453 , 2460 , 2465 , 2545 , 2560 , 2651 , 2655 , 2686 , 2728 , 2765 , 2785
<code>\liSortierPfeil</code>	2383	<code>\liZustandsMengenSammlung</code>	2095	<code>\neg</code>	2832
<code>\liSortierPfeilUnten</code>	2386	<code>\liZustandsMengenSammlungNr</code>	2104	<code>\negthinspace</code>	1694
<code>\liSpaltenUmbruch</code>	2458	<code>\liZustandsmengeOhneMathe</code>	1131	<code>\newcounter</code>	2184 , 2185
<code>\liSqlCode</code>	2543	<code>\liZustandsname</code>	1154	<code>\NewDocumentCommand</code>	62 , 103 , 140 , 161 , 203 , 224 , 547 , 564 , 609 , 614 , 1178 , 1200 , 1369 , 1492 , 1519 , 1678 , 1682 , 2006 , 2017 , 2084 , 2500 , 2510 , 2520 , 2532 , 2535 , 2539 , 2706
<code>\listen@punkt</code>	1635 , 1647	<code>\liZustandsnameGross</code>	1155 , 2096 , 2105	<code>\NewDocumentEnvironment</code>	1157 , 1250 , 1305 , 1157 , 1250 , 1305
<code>\liStrich</code>	1351	<code>\liZustandsnameTiefgestellt</code>	1170		
<code>\liSyntheseErklaerung</code>	2582 , 2647	<code>\liZustandsPaar</code>	1905		
<code>\liSyntheseUeberErklaerung</code>	2645	<code>\liZustandsPaarVariablenName</code>	1904 , 1907 , 1908		
<code>\liSyntheseUeberschrift</code>	2566 , 2646	<code>\llap</code>	2337		
<code>\liT</code>	1726 , 1741 , 1755 , 1816	<code>\log</code>	1744 , 1747 , 1750 , 1776 , 1782		
<code>\liTeilen</code>	1950				
<code>\literatur</code>	1517 , 1541				

1477, 1575, 1576, 1579, 1606, 1614, 1636, 1655, 1665, 1912, 1962, 2352, 2370	\newlength 2812	\ProvidesPackage 2, 20, 53, 221, 411, 472, 479, 543, 606, 621, 1076, 1112, 1225, 1265, 1274, 1279, 1308, 1346, 1418, 1501, 1516, 1522, 1544, 1690, 1850, 1863, 1942, 2025, 2091, 2116, 2121, 2165, 2331, 2344, 2373, 2454, 2461, 2466, 2546, 2561, 2652, 2656, 2687, 2729, 2766, 2786	474, 544, 623, 624, 1078, 1080, 1081, 1117, 1226, 1227, 1230, 1232, 1234, 1239, 1248, 1255, 1266, 1267, 1276, 1280, 1281, 1282, 1310, 1311, 1350, 1419, 1518, 1523, 1524, 1540, 1547, 1548, 1549, 1577, 1692, 1853, 1854, 2026, 2028, 2118, 2178, 2180, 2332, 2333, 2334, 2347, 2375, 2406, 2457, 2463, 2468, 2479, 2480, 2548, 2653, 2660, 2690, 2691, 2693, 2694, 2732, 2733, 2734, 2767, 2787
\node 560, 1485, 1490, 2210, 2215, 2220, 2226, 2232, 2238, 2389, 2434, 2723	\noexpand 2298, 2299, 2300, 2319, 2434	Q	\right 1694
\noindent 355, 611, 616, 1562, 1564, 1568, 1572, 1600, 1628, 1630, 1643, 1659, 1661, 1669, 1838, 1891, 1894, 1897, 1900, 2365, 2664, 2676, 2818	\nolinkurl 2497	\QS@list 2291, 2302, 2306, 2313, 2319, 2324, 2327	\RIGHTarrow ... 2708, 2713
\normalsize 1237	\notin 617	\QS@select@equal 2266, 2270	\Rightarrow 612, 617
\null 2669	O	\QS@select@greater 2267, 2271	\rightarrow 207, 501, 506, 514, 518, 520, 521, 523, 578, 586, 2088, 2591, 2598, 2600, 2603, 2608, 2613, 2614, 2619
\o@join 2336, 2339, 2340, 2341	\Omega 1707	\QS@select@smaller 2262, 2265, 2269	\rightouterjoin 2340
\omega 2126, 2127, 2147, 2148	\or 2247	\QS@sort@a 2244, 2277, 2298, 2299	\Roman 1245
P	\pagestyle 1260	\QS@sort@b 2244, 2245	\roman 1245, 1247
\par ... 354, 375, 537, 1189, 1197, 1567, 1629, 1652, 1660, 2289, 2311, 2326, 2367, 2646, 2670, 2674, 2682, 2817, 2823	\paragraph 1237	\QS@sort@c 2248, 2255	\romannumeral 2256
\parindent 2358	\path 94, 141, 204, 575	\QS@sort@d 2256, 2264	\rtimes 1852
\pgfkeys .. 2040, 2709, 2710, 2711, 2712, 2713, 2716, 2719, 2721	\pgfmath@count 2411, 2413, 2415	\QS@sort@empty . 2246, 2251	\rule 2289, 2311, 2326, 2337
\pgfmath@smuggleone 2416	\pgfmathdeclarefunction 2410	\QS@sort@single 2247, 2252	S
\pgfmathhint 2411	\pgfmathparse 1319, 2428, 2433, 2436, 2450, 2451	\QSinitialize 2169, 2281, 2316	\sb 67, 77, 79, 108, 166, 513, 514, 518, 521, 522, 523, 1172, 1174, 1744, 1747, 1750, 1776, 1782, 1931, 2076, 2085, 2613, 2614, 2615, 2620, 2624, 2625, 2628, 2629, 2632, 2633, 2634
\pgfmathresult 1320, 2411, 2412, 2414, 2416, 2429, 2437, 2450, 2451	\pgfutil@empty 2412	\QSIr . 2252, 2258, 2266, 2286, 2300, 2305, 2308	\scriptscriptstyle 578, 586, 594
\pgfutil@loop 2413	\pgfutil@repeat 2416	\QSIrr 2287, 2300, 2301, 2309	\scriptsize 1191, 1430, 1437, 1443, 1505, 1506, 1509, 1510, 2737, 2790, 2820
\preceq 1372	\prime 1351	\QSLr 2258, 2265, 2276, 2277, 2285, 2298, 2303, 2307	\section 46
\printbibliography . 1541		\QSpivotStep 2171, 2281, 2285, 2296	\seq . 1494, 1495, 1496, 1638, 1639, 1640, 1647
		\QSR 2258	\setbox 2336
		\QSRr 2267, 2288, 2299, 2304, 2310, 2319, 2320, 2321	\setcounter 1238, 2290, 2312, 2326
		\QSsortStep 2173, 2281, 2297, 2298	\setganttlinklabel .. 1268, 1269, 1270, 1271
		\quad 1869	\setlength 2358, 2814, 2815, 2825
	R		
	\raisebox 1490		
	\relax 1553, 2258, 2301, 2303, 2304, 2305, 2413, 2415		
	\renewcommand ... 1240, 1241, 1242, 1243, 1246, 1247, 1261, 1262, 1511, 1512, 1911		
	\repeat 2174		
	\RequirePackage 55, 158, 222, 226, 407, 413, 414,		

<code>\setmainfont</code>	1228	<code>\titlespacing</code>	1236	731, 736, 741, 744,
<code>\setmainlanguage</code>	408	<code>\tl</code>	29, 37, 63, 64, 65,	802, 803, 804, 809,
<code>\setminted</code>	2485, 2486		66, 67, 68, 71, 72,	810, 839, 869, 896,
<code>\setminus</code>	1991		73, 74, 75, 77, 79,	897, 900, 951, 954,
<code>\setsansfont</code>	1229		104, 105, 106, 107,	993, 999, 1000,
<code>\setul</code>	2349		108, 109, 110, 113,	1018, 1019, 1020,
<code>\sffamily</code>	488,		114, 115, 116, 117,	1039, 1040, 1041, 1042
	1235, 1237, 1339, 2569		118, 119, 162, 163,	<code>\umlddep</code>
<code>\shoveleft</code>	1971		164, 165, 166, 167,	959
<code>\shoveright</code>	1975		168, 171, 172, 173,	<code>\umlHVHaggreg</code>
<code>\Sigma</code>	64, 105,		174, 175, 176, 177,	749, 815, 1009
	163, 1141, 1142, 1202		291, 295, 319, 323,	<code>\umlinherit</code>
<code>\sigma</code>	511, 513, 514		324, 325, 328, 333,	688, 739, 904, 949, 957
<code>\SLASH</code>	1559		334, 335, 346, 347,	<code>\umlnote</code>
<code>\small</code>	1668		348, 349, 360, 366,	690, 906, 1049
<code>\sort</code>	2423		369, 372, 381, 395,	<code>\umlreal</code>
<code>\sortList</code>	2422, 2431		548, 551, 556, 557,	686, 747
<code>\square</code>	476		565, 566, 569, 570,	<code>\umlsimpleclass</code>
<code>\stepcounter</code> 2210, 2215,			1180, 1201, 1202,	645,
	2220, 2223, 2225,		1203, 1204, 1207,	646, 647, 651, 653,
	2229, 2231, 2235, 2237		1208, 1209, 1210, 1727	654, 655, 681, 834,
<code>\str</code>	489, 498, 1581,	<code>\tmp</code>	1993	835, 836, 895, 947, 948
	2100, 2109, 2570, 2583	<code>\TmpPlaceEight</code>	2048	<code>\umlstatic</code>
<code>\string</code>	1973, 1983	<code>\TmpPlaceFive</code>	2045	841, 870
<code>\StrSubstitute</code>	2430, 2432	<code>\TmpPlaceFour</code>	2044	<code>\umluniaggreg</code>
<code>\strut</code>	1798, 1802,	<code>\TmpPlaceNine</code>	2049	902
	1806, 1810, 1814, 2458	<code>\TmpPlaceOne</code>	2041	<code>\umluniassoc</code>
<code>\subsetq</code> 2594, 2627, 2634		<code>\TmpPlaceSeven</code>	2047	665,
		<code>\TmpPlaceSix</code>	2046	687, 903, 1027, 1028
		<code>\TmpPlaceTen</code>	2050	<code>\umlVHuniassoc</code>
		<code>\TmpPlaceThree</code>	2043	666, 667
		<code>\TmpPlaceTwo</code>	2042	<code>\umlVHVdep</code>
		<code>\TmpScale</code>	2061	659,
		<code>\TmpTransitionEight</code>	2037, 2058	660, 662, 663, 843, 844
		<code>\TmpTransitionFive</code>	2034, 2055	<code>\umlVHVinherit</code>
		<code>\TmpTransitionFour</code>	2033, 2054	642, 643, 648, 649,
		<code>\TmpTransitionNine</code>	2038, 2059	656, 657, 812, 813,
		<code>\TmpTransitionOne</code>	2030, 2051	837, 838, 1007, 1008
		<code>\TmpTransitionSeven</code>	2036, 2057	<code>\umlVHVreal</code>
		<code>\TmpTransitionSix</code>	2035, 2056	806, 807, 1044, 1045
		<code>\TmpTransitionTen</code>	2039, 2060	<code>\UParrow</code>
		<code>\TmpTransitionThree</code>	2032, 2053	2710
		<code>\TmpTransitionTwo</code>	2031, 2052	<code>\url</code>
		<code>\TmpX</code>	2062	1679
		<code>\TmpY</code>	2063	<code>\usemintedstyle</code>
		<code>\today</code>	1506	2482
		<code>\ttfamily</code>	2350	<code>\usetikzlibrary</code>
				56,
				415, 1079, 1283,
				1420, 2027, 2335,
				2376, 2407, 2695, 2783
				V
				<code>\value</code>
				2172
				<code>\varepsilon</code>
				490,
				501, 502, 1127,
				1744, 1750, 1779, 1790
				<code>\vfill</code>
				2458
				<code>\vrule</code>
				2665, 2669
				<code>\vspace</code>
				1615, 1633, 2678, 2680
				X
				<code>\xappto</code>
				2434, 2440, 2444
				<code>\xdef</code>
				1313
				<code>\xintApply</code>
				2260
				<code>\xintApplyUnbraced</code>
				2259, 2265, 2266, 2267
				<code>\xintCSVtoList</code>
				2319
				<code>\xintFor</code>
				2209, 2214, 2219,
				2224, 2230, 2236, 2277
				<code>\xintifEq</code>
				2270
				<code>\xintifForLast</code>
				2226, 2232, 2238
				<code>\xintifGt</code>
				2271
				<code>\xintifLt</code>
				2269
				U
				<code>\ul</code>
				1084, 2348, 2349
				<code>\umlaggreg</code>
				1047
				<code>\umlclass</code>
				1029
				<code>\umlclass</code>
				630, 634,
				638, 682, 683, 684,

\xintLength	2244		Z		\zustandsnamens@liste
			\ZB	<u>1686</u>	... 1145, 1152, 1153
\xintnthelt	2256	\zB	<u>1685</u>	