

lehramt-informatik

Hermine Bschlangaul *

September 1, 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	klassen-konfiguration-aufgabe.sty	56
2.19	komplexitaetstheorie.sty	57
2.20	kontrollflussgraph.sty	59
2.21	kopf-fusszeilen.sty	61
2.22	literatur-dummy.sty	62
2.23	literatur.sty	63
2.24	makros.sty	64
2.25	master-theorem.sty	68
2.26	mathe.sty	72
2.27	minimierung.sty	73
2.28	normalformen.sty	76
2.29	petri.sty	79
2.30	potenzmengen-konstruktion.sty	81
2.31	pseudo.sty	83
2.32	pumping-lemma.sty	84
2.33	quicksort.sty	85
2.34	relationale-algebra.sty	88
2.35	rmodell.sty	89
2.36	sortieren.sty	90
2.37	spalten.sty	92
2.38	struktogramm.sty	93
2.39	syntax.sty	94
2.40	syntaxbaum.sty	96
2.41	synthese-algorithmus.sty	97
2.41.1	TeX-Markup Grundgerüst	97
2.41.2	TeX-Markup Linksreduktion	97
2.41.3	TeX-Markup Rechtsreduktion	97
2.42	tabelle.sty	101
2.43	typographie.sty	102
2.44	uml.sty	103
2.45	vollstaendige-induktion.sty	105
2.46	wasserfall.sty	107
2.47	wpkalkuel.sty	108

3 Index

109

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 \_gib_github_url:
47
48 \section{\_gib_aufgaben_titel:}
49
50 \bigskip
51 }

52 \ExplSyntaxOff
53

```

2.3 automaten.sty

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

2.3.1 Endlicher Automat

```
56 \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}
```



```
57 \RequirePackage{tikz}
58 \usetikzlibrary{arrows,automata,positioning}
59 \liLadePakete{mathe}
60 \directlua{
61   automaten = require('lehramt-informatik-automaten')
62 }
```

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

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

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

```

77     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
78     dea .value_forbidden:n = true,
79     dea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{DEA}}}},
80     nea .value_forbidden:n = true,
81     nea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{NEA}}}},
82   }
83
84   \keys_set:nn { automat } { #2 }
85
86   $ #1 \l_typ_tl = (
87     \l_zustaende_tl,
88     \l_alphabet_tl,
89     \l_delta_tl,
90     \l_ende_tl,
91     \l_start_tl
92   )$
93 }
94 \ExplSyntaxOff

```

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

```

95 \def\liAutomatenKante#1#2#3#4{
96   \path (#1) edge[#4] node{#3} (#2);
97 }

98 \tikzset{
99   li automat/.style={
100     ->,
101     node distance=2cm
102   },
103 }

```

2.3.2 Kellerautomat

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

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

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

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



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

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

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

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

```

120     kellerboden .code:n = {\tl_set:Nn \l_kellerboden_tl {##1}},
121     ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
122   }
123
124   \keys_set:nn { kellerautomat } { #2 }
125
126   $#1 = (
127     \l_zustaende_tl,
128     \l_alphabet_tl,
129     \l_kelleralphabet_tl,
130     \l_delta_tl,
131     \l_start_tl,
132     \l_kellerboden_tl,
133     \l_ende_tl
134   )$
135 }
136 \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}`

```

137 \ExplSyntaxOn
138 \def\liKellerUebergang#1{
139   \directlua{automaten.drucke_keller_uebergaenge('#1')}
140 }
141 \ExplSyntaxOff

```

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

```

142 \NewDocumentCommand{\liKellerKante} { 0{above} m m m } {
143   \path (#2) edge[#1] node{\liKellerUebergang{#4}} (#3);
144 }

145 \tikzset{
146   li keller knoten/.style={
147     text width=2cm,
148     align=center,
149     font=\footnotesize,
150   },
151   li kellerautomat/.style={
152     li automat,
153     every edge/.append style={
154       every node/.style={
155         li keller knoten
156       }
157     }
158   }
159 }

```

2.3.3 Turingmaschine

```
160 \RequirePackage{amssymb}
```

\liTuringLeerzeichen

□

```
161 \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},
}

```

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

```

162 \ExplSyntaxOn
163 \NewDocumentCommand {\liTuringMaschine} { O{TM} m } {
164   \tl_set:Nn \l_zustaende_tl {Z}
165   \tl_set:Nn \l_alphabet_tl {\Sigma}
166   \tl_set:Nn \l_bandalphabet_tl {\Gamma}
167   \tl_set:Nn \l_delta_tl {\delta}
168   \tl_set:Nn \l_start_tl {z\sb{0}}
169   \tl_set:Nn \l_leerzeichen_tl {\liTuringLeerzeichen}
170   \tl_set:Nn \l_ende_tl {E}
171
172   \keys_define:nn { kellerautomat } {
173     zustaende .code:n = {\tl_set:Nn \l_zustaende_tl {\liMenge{##1}}},
174     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
175     bandalphabet .code:n = {\tl_set:Nn \l_bandalphabet_tl {\liMenge{##1}}},
176     delta .code:n = {\tl_set:Nn \l_delta_tl {##1}},
177     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
178     leerzeichen .code:n = {\tl_set:Nn \l_leerzeichen_tl {##1}},
179     ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
180   }
181
182   \keys_set:nn { kellerautomat } { #2 }
183
184   $\text{#1} = (
185     \l_zustaende_tl,
186     \l_alphabet_tl,
187     \l_bandalphabet_tl,
188     \l_delta_tl,
189     \l_start_tl,
190     \l_leerzeichen_tl,
191     \l_ende_tl
192   )$
193 }
194 \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)`

```

195 \ExplSyntaxOn
196 \def\liTuringUebergangZelle#1{
197   \directlua{tex.print(automaten.gib_einen_turing_uebergang('#1'))}
198 }
199 \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)`

```

200 \ExplSyntaxOn
201 \def\liTuringUebergaenge#1{
202   \directlua{automaten.drucke_turing_uebergaenge('#1')}
203 }
204 \ExplSyntaxOff

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

205 \NewDocumentCommand{\liTuringKante} { 0{above} m m m } {
206   \path (#2) edge[#1] node{\liTuringUebergaenge{#4}} (#3);
207 }

\liTuringUeberfuehrung

208 \def\liTuringUeberfuehrung{
209    $\delta : Z \times \Gamma \rightarrow Z \times \Gamma \times \{ L, R, N \}$ 
210 }

211 \tikzset{
212   li turingmaschine/.style={
213     li automat,
214     every edge/.append style={
215       every node/.style={
216         li keller knoten
217       }
218     }
219   }
220 }
221

```

2.4 basis.sty

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

224 \RequirePackage{xparse}

225 \ExplSyntaxOn

\liLadePakete

226 \NewDocumentCommand{\liLadePakete}{ m }
227 {
228   \clist_map_inline:nn { #1 } { \RequirePackage{lehramt-informatik-##1} }
229 }

\liLadeAllePakete

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

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.

```
272 \clist_new:N \g_auf_schluesssel_clist
273 \clist_set:Nn \g_auf_schluesssel_clist {
```



```

274  titel,
275  thematik,
276  stichwoerter,
277  zitat_schluessel,
278  zitat_beschreibung,
279  %
280  bearbeitungs_stand,
281  korrektheit,
282  %
283  relativer_pfad,
284  %
285  examen_nummer,
286  examen_jahr,
287  examen_monat,
288  examen_thema_nr,
289  examen_telaufgabe_nr,
290  examen_aufgabe_nr,
291 }

```

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

```

292 \clist_map_inline:Nn \g_auf_schluessel_clist {
293   \tl_new:c {g_auf_#1_tl}
294 }

```

Funktion über alle globalen Token-List-Variablen zurückzusetzen.

```

295 \cs_new:Npn \setze_variablen_zurueck: {
296   \clist_map_inline:Nn \g_auf_schluessel_clist {
297     \tl_clear:c {g_auf_##1_tl}
298   }
299 }

```

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

```

300 \keys_define:nn { aufgabenmetadaten }
301 {
302   Titel .tl_gset:N = \g_auf_titel_tl,
303   Thematik .tl_gset:N = \g_auf_thematik_tl,
304   Stichwoerter .tl_gset:N = \g_auf_stichwoerter_tl,
305   ZitatSchluessel .tl_gset:N = \g_auf_zitat_beschreibung_tl,
306   ZitatBeschreibung .tl_gset:N = \g_auf_zitat_schluessel_tl,
307   %
308   BearbeitungsStand .tl_gset:N = \g_auf_bearbeitungs_stand_tl,
309   Korrektheit .tl_gset:N = \g_auf_korrekttheit_tl,
310   %
311   RelativerPfad .tl_gset:N = \g_auf_relativer_pfad_tl,
312   %
313   ExamenNummer .tl_gset:N = \g_auf_examen_nummer_tl,
314   ExamenJahr .tl_gset:N = \g_auf_examen_jahr_tl,
315   ExamenMonat .tl_gset:N = \g_auf_examen_monat_tl,
316   ExamenThemaNr .tl_gset:N = \g_auf_examen_thema_nr_tl,
317   ExamenTeilaufgabeNr .tl_gset:N = \g_auf_examen_telaufgabe_nr_tl,
318   ExamenAufgabeNr .tl_gset:N = \g_auf_examen_aufgabe_nr_tl,
319 }
320 \cs_gset:Npn \setze_relativen_pfad: {
321   \tl_if_empty:NTF \g_auf_relativer_pfad_tl
322   {
323     \bool_if:nTF
324     {
325       ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
326       ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
327       ! \tl_if_empty_p:N \g_auf_examen_monat_tl
328     }
329     {

```

```

330 \tl_gset:Nn \g_auf_relativer_pfad_tl {
331   Staatsexamen /
332   \g_auf_examen_nummer_tl /
333   \g_auf_examen_jahr_tl /
334   \g_auf_examen_monat_tl /
335   \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} { Thema - \g_auf_examen_thema_nr_tl / }
336   \tl_if_empty:NTF \g_auf_examen_telaufgabe_nr_tl {} { Teilaufgabe - \g_auf_examen_te
337   \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} { Aufgabe - \g_auf_examen_aufgabe_nr
338 }
339 }
340 {}
341 }
342 {}
343 }

344 \cs_gset:Npn \_gib_examen_titel: {
345   \cs_set:Nn \_trenner: { \, / \, }
346   \bool_if:nTF
347   {
348     ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
349     ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
350     ! \tl_if_empty_p:N \g_auf_examen_monat_tl &&
351     ! \tl_if_empty_p:N \g_auf_examen_aufgabe_nr_tl
352   }
353   {
354     {
355       \footnotesize
356       \par
357       \noindent
358       Staatsexamen ~
359       \g_auf_examen_nummer_tl \_trenner:
360       \g_auf_examen_jahr_tl \_trenner:
361
362       \tl_case:Nn \g_auf_examen_monat_tl
363       {
364         { 03 } { Frühjahr }
365         { 09 } { Herbst }
366       } \_trenner:
367
368       \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} {
369         Thema ~ Nr. ~ \g_auf_examen_thema_nr_tl \_trenner:
370       }
371       \tl_if_empty:NTF \g_auf_examen_telaufgabe_nr_tl {} {
372         Teilaufgabe ~ Nr. ~ \g_auf_examen_telaufgabe_nr_tl \_trenner:
373       }
374       \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} {
375         Aufgabe ~ Nr. ~ \g_auf_examen_aufgabe_nr_tl
376       }
377       \par
378       \bigskip
379     }
380   }
381 }

382 \cs_new:Npn \_gib_github_url: {
383   \tl_if_empty:NTF \g_auf_relativer_pfad_tl {} {
384     Github :~\href{
385       \LehramtInformatikGithubRawDomain /
386       \LehramtInformatikGithubTexRepo /
387       \LehramtInformatikGitBranch /
388       \g_auf_relativer_pfad_tl
389     }{
390       \g_auf_relativer_pfad_tl
391     }
392   }

```

```

393 }
394 \cs_new:Npn \_gib_aufgaben_titel: {
395   \g_auf_titel_tl
396
397   \tl_if_empty:NTF \g_auf_thematik_tl
398   {}
399   {
400     \, ~ [
401       \g_auf_thematik_tl
402     ]
403   }
404 }
405 \msg_new:nnn { aufgabenmetadaten } { kein-titel }
406 { Der~Schlüssel~ist~zwingend~notwendig }
407 \def\LehramtInformatikRepository{/pfad/zum/repository}
408 % \RequirePackage{polyglossia}
409 % \setmainlanguage{german}
410

```

2.5 baum.sty

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

414 \RequirePackage{tikz}

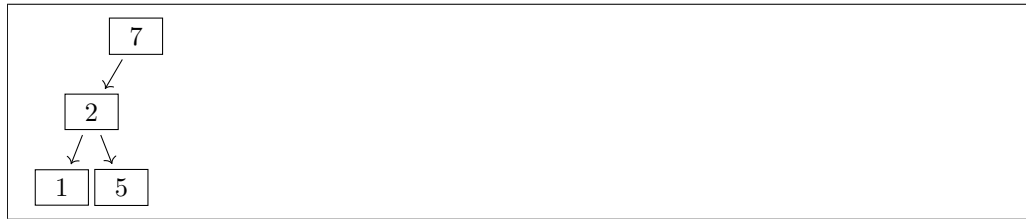
    für li binaer baum
415 \RequirePackage{tikz-qtree}

    Für b baum
416 \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}
```



```
417 \tikzset{
418   li binaer baum/.style={
419     shorten <=2pt,
420     shorten >=2pt,
421     ->,
422     every tree node/.style={
423       minimum width=2em,
424       draw,
425       rectangle
426     },
427     blank/.style={
428       draw=none
429     },
430     edge from parent/.style={
431       draw,
432       edge from parent path={(\tikzparentnode) -- (\tikzchildnode)}
433     },
434     level distance=1cm,
435     every label/.style={
436       gray,
437       font=\footnotesize,
438       label position=0,
439       label distance=0cm,
440     }
441   },
442 }
```

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}

```



```

443 \tikzset{
444   li bbaum knoten/.style={
445     rectangle split parts=10,
446     rectangle split,
447     rectangle split horizontal,
448     rectangle split ignore empty parts,
449     draw,
450     fill=white
451   },
452   li bbaum/.style={
453     every node/.style={
454       li bbaum knoten
455     },
456     level 1/.style={
457       level distance=12mm,
458       sibling distance=25mm,
459     },
460     every child/.style={
461       shorten <= 2pt,
462       shorten >= 6pt,
463       ->,
464     },
465     level 2/.style={
466       level distance=9mm,
467       sibling distance=15mm,
468     },
469   }
470 }
471

```

2.6 checkbox.sty

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

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

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

478
```


2.7 chomsky-normalform.sty

```

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

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

486 \def\liChomskyUeberschrift#1{
487   {
488     \bfseries
489     \sffamily
490     \str_case:nn {#1} {
491       {1} {Elimination~der~ $\varepsilon$ -Regeln}
492       {2} {Elimination~von~Kettenregeln}
493       {3} {Separation~von~Terminalzeichen}
494       {4} {Elimination~von~mehrelementigen~Nonterminalketten}
495     }
496   }
497 }

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

498 \def\li@chomsky@erklaerung@texte#1{
499   \str_case:nn {#1} {
500     %
501     {1} {
502       Alle~Regeln~der~Form~ $A \rightarrow \varepsilon$ ~werden~eliminiert.~
503       Die~Ersetzung~von~ $A$ ~wird~durch~ $\varepsilon$ ~in~allen~anderen~
504       Regeln~vorweggenommen.
505     }
506     {2} {
507       Jede~Produktion~der~Form~ $A \rightarrow B$ ~mit~ $A, B \in S$ ~wird~
508       als~Kettenregel~bezeichnet.~Diese~tragen~nicht~zur~Produktion~
509       von~Terminalzeichen~bei~und~lassen~sich~ebenfalls~eliminieren.
510     }
511     {3} {
512       Jedes~Terminalzeichen~ $\sigma$ ,~das~in~Kombination~mit~anderen~
513       Symbolen~auftaucht,~wird~durch~ein~neues~Nonterminal~
514        $S_{\sigma}$ ~ersetzt~und~die~Menge~der~Produktionen~durch~die~
515       Regel~ $S_{\sigma} \rightarrow \sigma$ ~ergänzt.
516     }
517     {4} {
518       Alle~Produktionen~der~Form~
519        $A \rightarrow B_{\{1\}} B_{\{2\}} \dots B_{\{n\}}$ ~
520       werden~in~die~Produktionen~
521        $A \rightarrow$ 
522        $A_{\{n-1\}} B_{\{n\}}, A_{\{n-1\}} \rightarrow$ 
523        $A_{\{n-2\}} B_{\{n-1\}}, \dots,$ 
524        $A_{\{2\}} \rightarrow B_{\{1\}} B_{\{2\}}$ ~zerteilt.~
525       Nach~der~Ersetzung~sind~alle~längeren~Nonterminalketten~
526       vollständig~heruntergebrochen~und~die~Chomsky-Normalform~erreicht.
527     }
528   }
529 }

530 \def\liChomskyErklaerung#1{
531   {
532     \itshape
533     \footnotesize
534     \liParagraphMitLinien{\li@chomsky@erklaerung@texte{#1}}
535   }

```

536 }

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

537 \def\liChomskyUeberErklaerung#1{

538 \liChomskyUeberschrift{#1}\par

539 \liChomskyErklaerung{#1}

540 }

541 \ExplSyntaxOff

542

2.8 cpm.sty

```

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

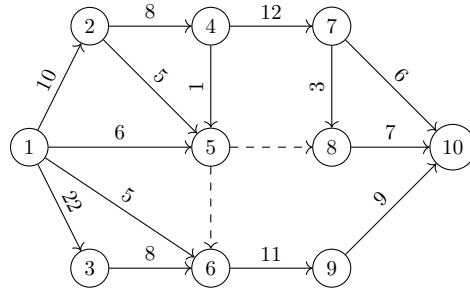
```

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}

```

```

547 \ExplSyntaxOn
548 \NewDocumentCommand { \liCpmEreignis } { 0{} m m m } {
549   \tl_set:Nn \l_name_tl {}
550
551   \keys_define:nn { cpmEreignis } {
552     name .code:n = {\tl_set:Nn \l_name_tl {##1}},
553   }
554
555   \keys_set:nn { cpmEreignis } { #1 }
556
557   \tl_if_empty:NT \l_name_tl {
558     \tl_set:Nn \l_name_tl {#2}
559   }

```

```

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

\liCpmVorgang liCpmVORGANG\((.*)>(.*))\{(.*)\}
564 \ExplSyntaxOn
565 \NewDocumentCommand { \liCpmVorgang } { 0{ } m m m } {
566   \tl_set:Nn \l_schein_tl {}
567   \tl_set:Nn \l_kritisch_tl {}
568
569   \keys_define:nn { cpmVorgang } {
570     schein .code:n = {\tl_set:Nn \l_schein_tl {dashed}},
571     kritisch .code:n = {\tl_set:Nn \l_kritisch_tl {thick}},
572   }
573
574   \keys_set:nn { cpmVorgang } { #1 }
575
576   \path[->,\l_schein_tl,\l_kritisch_tl] (#2) edge node[auto,sloped] {#4} (#3);
577 }
578 \ExplSyntaxOff

```

2.8.1 Hilfsmakros für Zeitpunkt-Tabelle

```

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

```

```

\liCpmVonZu Let-Abkürzung: \let\vz=\liCpmVonZu
\liCpmVonZu{1}(2-3):  $l_{(2 \rightarrow 3)}$ 
579 \def\liCpmVonZuOhneMathe#1(#2-#3){#1_{\scriptscriptstyle(#2\rightarrow#3)}}
580 \def\liCpmVonZu#1(#2-#3){%
581   \ifmmode%
582     \liCpmVonZuOhneMathe{#1}(#2-#3)%
583   \else%
584     $\liCpmVonZuOhneMathe{#1}(#2-#3)$%
585   \fi%
586 }

```

```

\liCpmVon Let-Abkürzung: \let\v=\liCpmVon
\liCpmVon{1}(2):  $l_{(\rightarrow 2)}$ 
587 \def\liCpmVonOhneMathe#1(#2){#1_{\scriptscriptstyle(\rightarrow#2)}}
588 \def\liCpmVon#1(#2){%
589   \ifmmode%
590     \liCpmVonOhneMathe{#1}(#2)%
591   \else%
592     $\liCpmVonOhneMathe{#1}(#2)$%
593   \fi%
594 }

```

```

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

```

```

601 \fi%
602 }

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

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

605

```

2.9 cyk-algorithmus.sty

```
606 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
607 \ProvidesPackage{lehramt-informatik-cyk-algorithmus}[2021/06/18 Hilfsmakros
608 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`

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

`\liWortInSprache` $\Rightarrow abc \in L(Y)$

```
610 \NewDocumentCommand{ \liWortInSprache } { m O{L(G)} } {
611   \bigskip
612   \noindent
613   $\Rrightarrow #1 \in #2$
614 }
```

`\liWortNichtInSprache` $\Rightarrow abc \notin L(G)$

```
615 \NewDocumentCommand{ \liWortNichtInSprache } { m O{L(G)} } {
616   \bigskip
617   \noindent
618   $\Rrightarrow #1 \notin #2$
619 }
```

```
620
```

2.10 entwurfsmuster.sty

```
621 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
622 \ProvidesPackage{lehramt-informatik-entwurfsmuster}[2021/05/06]
623 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

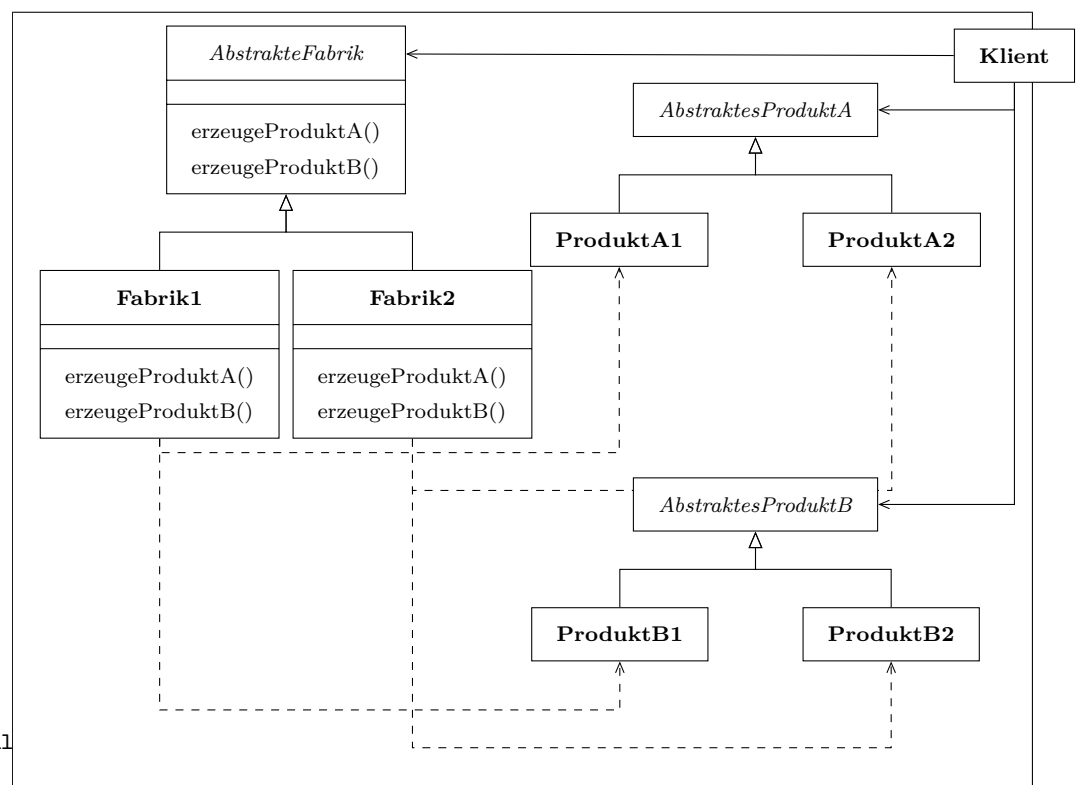
1. Uml: Uml-Klassendiagramm \liEntwurfsEinzelstueckUml
2. Akteure: Akteure, beteiligte Klassen \liEntwurfsEinzelstueckAkteure
3. Code: Allgemeines Code-Beispiel \liEntwurfsEinzelstueckCode
4. ohne: Ohne Suffix, Bündelung der einzelnen Makros eines Entwurfsmusters \liEntwurfsEinzelstueckAkteure

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

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

```
625 \RequirePackage{lehramt-informatik-syntax}
626 \def\liEntwurfsCode#1#2{
627   \liJavaDatei{entwurfsmuster/#1/allgemein/#2}
628 }
```

2.10.3 Abstrakte Fabrik (Abstract Factory)



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

```

635 \umlclass[below left=1cm and -1.5cm of AbstrakteFabrik]{Fabrik1}{-}{
636     erzeugeProduktA()\
637     erzeugeProduktB()\
638 }
639 \umlclass[below right=1cm and -1.5cm of AbstrakteFabrik]{Fabrik2}{-}{
640     erzeugeProduktA()\
641     erzeugeProduktB()\
642 }
643 \umlVHVinherit{Fabrik1}{AbstrakteFabrik}
644 \umlVHVinherit{Fabrik2}{AbstrakteFabrik}
645
646 \umlsimpleclass[right=3cm of AbstrakteFabrik,type=abstract]{AbstraktesProduktA}
647 \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktA]{ProduktA1}
648 \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktA]{ProduktA2}
649 \umlVHVinherit{ProduktA1}{AbstraktesProduktA}
650 \umlVHVinherit{ProduktA2}{AbstraktesProduktA}
651
652 \umlsimpleclass[above right=0cm and 1cm of AbstraktesProduktA]{Klient}
653
654 \umlsimpleclass[below=4.5cm of AbstraktesProduktA,type=abstract]{AbstraktesProduktB}
655 \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktB]{ProduktB1}
656 \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktB]{ProduktB2}
657 \umlVHVinherit{ProduktB1}{AbstraktesProduktB}
658 \umlVHVinherit{ProduktB2}{AbstraktesProduktB}
659
660 \umlVHVdep[arm1=-1.3cm,arm2=-1cm]{Fabrik1}{ProduktA1}
661 \umlVHVdep[arm2=-1cm]{Fabrik1}{ProduktB1}
662
663 \umlVHVdep[arm1=-1.8cm,arm2=-1.5cm]{Fabrik2}{ProduktA2}
664 \umlVHVdep[arm2=-1.5cm]{Fabrik2}{ProduktB2}
665
666 \umluniassoc[anchor2=25]{Klient}{AbstrakteFabrik}
667 \umlVHuniassoc[arm1=-1cm]{Klient}{AbstraktesProduktA}
668 \umlVHuniassoc{Klient}{AbstraktesProduktB}
669 \end{tikzpicture}
670 }

```

iEntwurfsAbstrakteFabrikCode

```

671 \def\liEntwurfsAbstrakteFabrikCode{
672     \li@EntwurfsCode{abstrakte_fabrik}{Produkte}
673     \li@EntwurfsCode{abstrakte_fabrik}{AbstrakteFabrik}
674     \li@EntwurfsCode{abstrakte_fabrik}{Klient}
675 }

```

\liEntwurfsAbstrakteFabrik

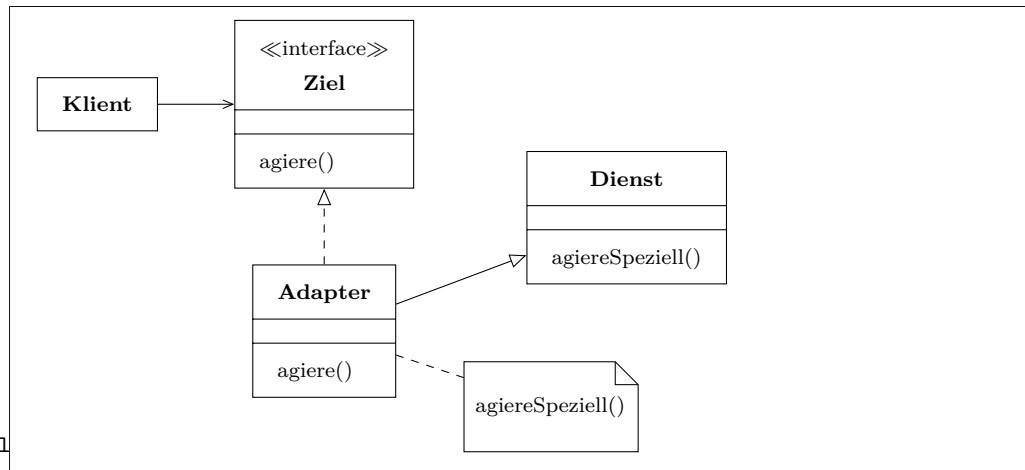
```

676 \def\liEntwurfsAbstrakteFabrik{
677     \liEntwurfsAbstrakteFabrikUml
678     \liEntwurfsAbstrakteFabrikCode
679 }

```

2.10.4 Adapter

\liEntwurfsAdapterUml



```

680 \def\liEntwurfsAdapterUml{
681   \begin{tikzpicture}
682     \umlsimpleclass[x=1,y=3]{Klient}{}{}
683     \umlclass[x=4,y=3,type=interface]{Ziel}{}{agiere()}
684     \umlclass[x=4,y=0]{Adapter}{}{agiere()}
685     \umlclass[x=8,y=1.5]{Dienst}{}{agiereSpeziell()}
686
687     \umlreal{Adapter}{Ziel}
688     \umluniassoc{Klient}{Ziel}
689     \umlinherit{Adapter}{Dienst}
690
691     \umlnote[x=7,y=-1,width=2cm]{Adapter}{agiereSpeziell()}
692   \end{tikzpicture}
693   \footcite[so ähnlich wie GoF]{wiki:adapter}
694 }

```

\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.

```

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

```

```

716
717 \end{description}
718 }

```

\liEntwurfsAdapterCode

```

719 \def\liEntwurfsAdapterCode{
720 \li@EntwurfsCode{adapter}{Dienst}
721 \li@EntwurfsCode{adapter}{Ziel}
722 \li@EntwurfsCode{adapter}{Adapter}
723 \li@EntwurfsCode{adapter}{Klient}
724 }

```

\liEntwurfsAdapter

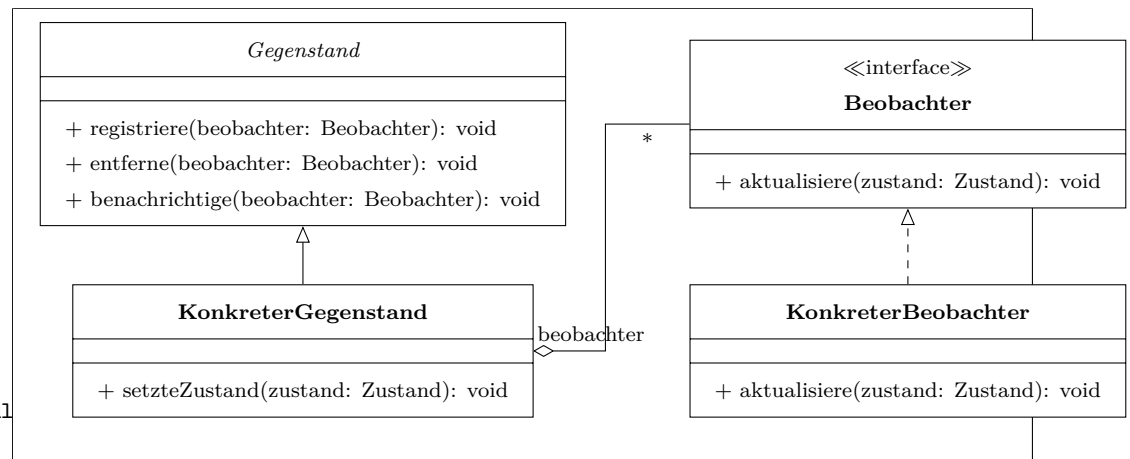
```

725 \def\liEntwurfsAdapter{
726 \liEntwurfsAdapterUml
727 \liEntwurfsAdapterAkteure
728 \liEntwurfsAdapterCode
729 }

```

2.10.5 Beobachter (Observer)

\liEntwurfsBeobachterUml



```

730 \def\liEntwurfsBeobachterUml{
731 \begin{tikzpicture}
732 \umlclass[x=0,y=0,type=abstract]{Gegenstand}{-}{
733 + registriere(beobachter: Beobachter): void\\
734 + entferne(beobachter: Beobachter): void\\
735 + benachrichtige(beobachter: Beobachter): void\\
736 }
737 \umlclass[x=0,y=-3]{KonkreterGegenstand}{-}{
738 + setzteZustand(zustand: Zustand): void
739 }
740 \umlinherit{KonkreterGegenstand}{Gegenstand}
741
742 \umlclass[x=8,y=0,type=interface]{Beobachter}{-}{
743 + aktualisiere(zustand: Zustand): void
744 }
745 \umlclass[x=8,y=-3]{KonkreterBeobachter}{-}{
746 + aktualisiere(zustand: Zustand): void
747 }
748 \umlreal{KonkreterBeobachter}{Beobachter}
749
750 \umlHVHaggreg[arg1=beobachter,pos1=0.8,mult2=*,pos2=2.5]
751 {KonkreterGegenstand}{Beobachter}
752 \end{tikzpicture}
753 }

```

\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.

```
754 \def\liEntwurfsBeobachterAkteure{
755   \begin{description}
756     \item[Gegenstand / Subjekt (Subject / Observable)]
757
758     Ein Subjekt (beobachtbares Objekt, auf Englisch publisher, also
759     „Veröffentlicher“, genannt) hat eine Liste von Beobachtern, ohne
760     deren konkrete Typen zu kennen. Es bietet eine Schnittstelle zur An-
761     und Abmeldung von Beobachtern und eine Schnittstelle zur
762     Benachrichtigung von Beobachtern über Änderungen an.\footcite[Seite
763     251]{gof}
764
765     \item[Beobachter (Observer)]
766
767     Die Beobachter (auf Englisch auch subscriber, also „Abonnent“,
768     genannt) definieren eine Aktualisierungsschnittstelle.
769
770     \item[konkreter/s Gegenstand / Subjekt (ConcreteSubject / ConcreteObservable)]
771
772     Ein konkretes Subjekt (konkretes, beobachtbares Objekt) speichert
773     den relevanten Zustand und benachrichtigt alle Beobachter bei
774     Zustandsänderungen über deren Aktualisierungsschnittstelle. Es
775     verfügt über eine Schnittstelle zur Erfragung des aktuellen
776     Zustands.
777
778     \item[Konkrete Beobachter (ConcreteObserver)]
779
780     Konkrete Beobachter verwalten die Referenz auf ein konkretes
781     Subjekt, dessen Zustand sie beobachten und speichern und dessen
782     Zustand konsistent ist. Sie implementieren eine
783     Aktualisierungsschnittstelle unter Verwendung der
784     Abfrageschnittstelle des konkreten Subjekts.
785     \footcite{wiki:beobachter}
786   \end{description}
787 }
```

\liEntwurfsBeobachterCode

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

```

794 \li@EntwurfsCode{beobachter}{Klient}
795 }

```

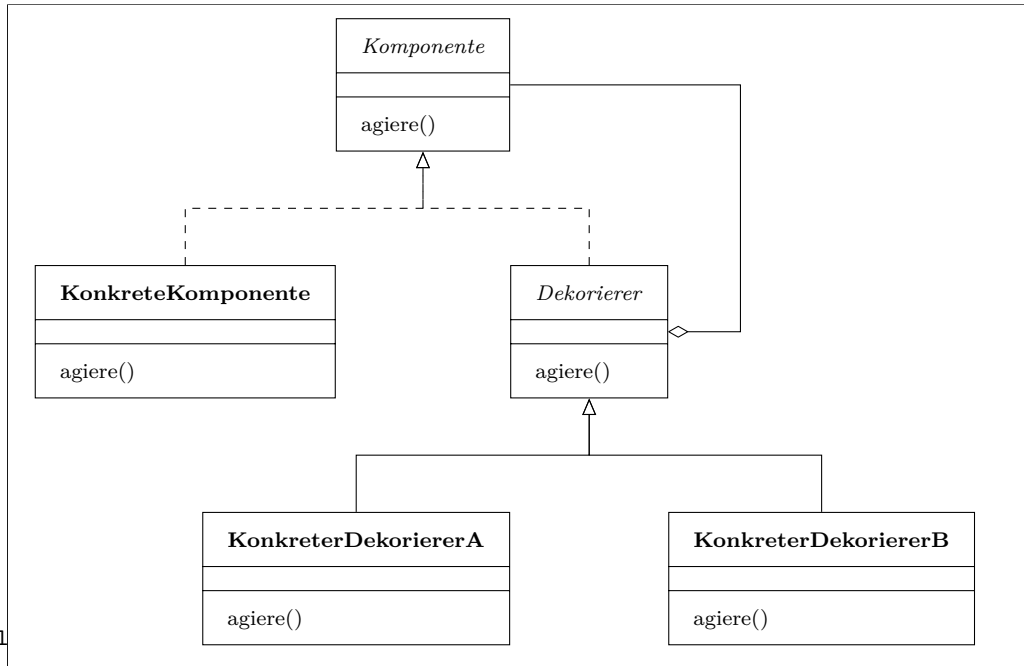
\liEntwurfsBeobachter

```

796 \def\liEntwurfsBeobachter{
797 \liEntwurfsBeobachterUml
798 \liEntwurfsBeobachterAkteure
799 \liEntwurfsBeobachterCode
800 }

```

2.10.6 Dekorierer (Decorator)



\liEntwurfsDekoriererUml

```

801 \def\liEntwurfsDekoriererUml{
802 \begin{tikzpicture}
803 \umlclass[type=abstract]{Komponente}{-}{agiere()}
804 \umlclass[below left=1.5cm and 0cm of Komponente]{KonkreteKomponente}{-}{agiere()}
805 \umlclass[below right=1.5cm and 0cm of Komponente,type=abstract]{Dekorierer}{-}{agiere()}
806
807 \umlVHVreal{KonkreteKomponente}{Komponente}
808 \umlVHVreal{Dekorierer}{Komponente}
809
810 \umlclass[below left=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererA}{-}{agiere()}
811 \umlclass[below right=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererB}{-}{agiere()}
812
813 \umlVHVinherit{KonkreterDekoriererA}{Dekorierer}
814 \umlVHVinherit{KonkreterDekoriererB}{Dekorierer}
815
816 \umlHVVHaggreg[arm1=2cm]{Dekorierer}{Komponente}
817 \footcite{wiki:dekorierer}
818 \end{tikzpicture}
819 }

```

\liEntwurfsDekoriererCode

```

820 \def\liEntwurfsDekoriererCode{
821 \li@EntwurfsCode{dekorierer}{Komponente}
822 \li@EntwurfsCode{dekorierer}{KonkreteKomponente}
823 \li@EntwurfsCode{dekorierer}{Dekorierer}
824 \li@EntwurfsCode{dekorierer}{KonkreterDekoriererA}
825 \li@EntwurfsCode{dekorierer}{KonkreterDekoriererB}
826 \li@EntwurfsCode{dekorierer}{Klient}
827 }

```

\liEntwurfsDekorierer

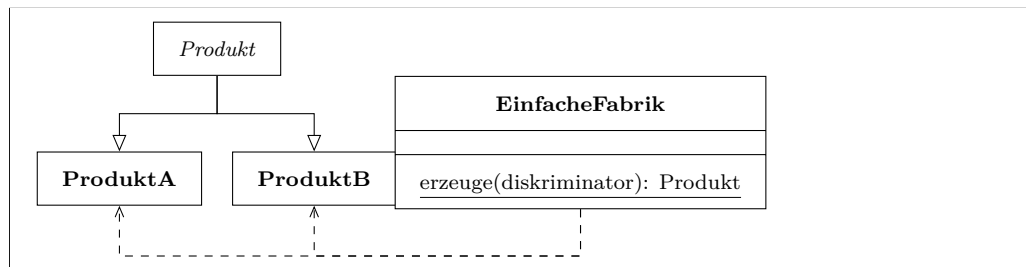
```

828 \def\liEntwurfsDekorierer{
829   \liEntwurfsDekoriererUml
830   \liEntwurfsDekoriererAkteure
831   \liEntwurfsDekoriererCode
832 }

```

2.10.7 Einfache Fabrik (Simple Factory)

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



```

833 \def\liEntwurfsEinfacheFabrikUml{
834   \begin{tikzpicture}
835     \umlsimpleclass[type=abstract]{Produkt}
836     \umlsimpleclass[below left=1cm and -0.65cm of Produkt]{ProduktA}
837     \umlsimpleclass[below right=1cm and -0.65cm of Produkt]{ProduktB}
838     \umlVHVinherit{Produkt}{ProduktA}
839     \umlVHVinherit{Produkt}{ProduktB}
840     \umlclass[below right=0cm and 1.5cm of Produkt]{EinfacheFabrik}{
841     }{
842       \umlstatic{erzeuge(diskriminator): Produkt}\\
843     }
844     \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktA}
845     \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktB}
846   \end{tikzpicture}
847 }

```

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.

```

848 \def\liEntwurfsEinfacheFabrikAkteure{
849   \begin{description}
850     \item[EinfacheFabrik]
851
852     Eine Klasse mit einer Erzeugungsmethode, die über eine größere
853     Bedingung verschiedene Objekt instanziiert.
854
855     \item[Produkt]
856
857     Eine abstrakte Klasse, die von den konkreten Produkten geerbt wird.
858
859     \item[KonkretesProdukt]
860
861     Ein konkretes Produkt, das von der einfachen Fabrik erzeugt wird.
862   \end{description}
863 }

```

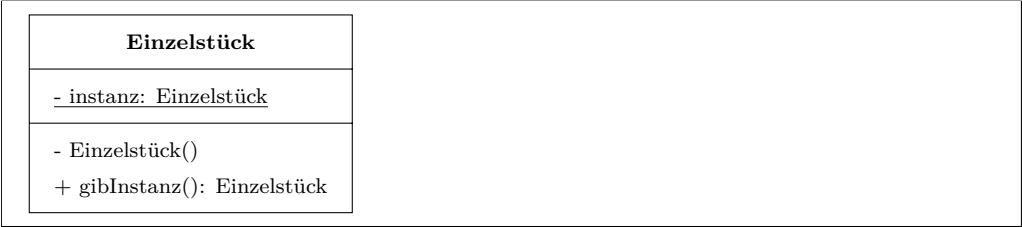
\liEntwurfsEinfacheFabrik

```
864 \def\liEntwurfsEinfacheFabrik{
865   \liEntwurfsEinfacheFabrikUml
866   \liEntwurfsEinfacheFabrikAkteure
867 }
```

2.10.8 Einzelstück (Singleton)

Quelle nach der deutschen Wikipedia

\liEntwurfsEinzelstueckUml



```
868 \def\liEntwurfsEinzelstueckUml{
869   \begin{tikzpicture}
870     \umlclass{Einzelstück}{
871       \umlstatic{- instanz: Einzelstück}\\
872     }{
873       - Einzelstück()\\
874       + gibInstanz(): Einzelstück
875     }
876   \end{tikzpicture}
877 }
```

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.

```
878 \def\liEntwurfsEinzelstueckAkteure{
879   \begin{description}
880     \item[Einzelstück (Singleton)]
881
882     stellt eine statische Methode bereit, mit deren Hilfe die Klienten
883     nur auf eine einzige Instanz der Klasse zugreifen können.
884   \end{description}
885 }
```

\liEntwurfsEinzelstueckCode

```
886 \def\liEntwurfsEinzelstueckCode{
887   \li@EntwurfsCode{einzelstueck}{Einzelstueck}
888 }
```

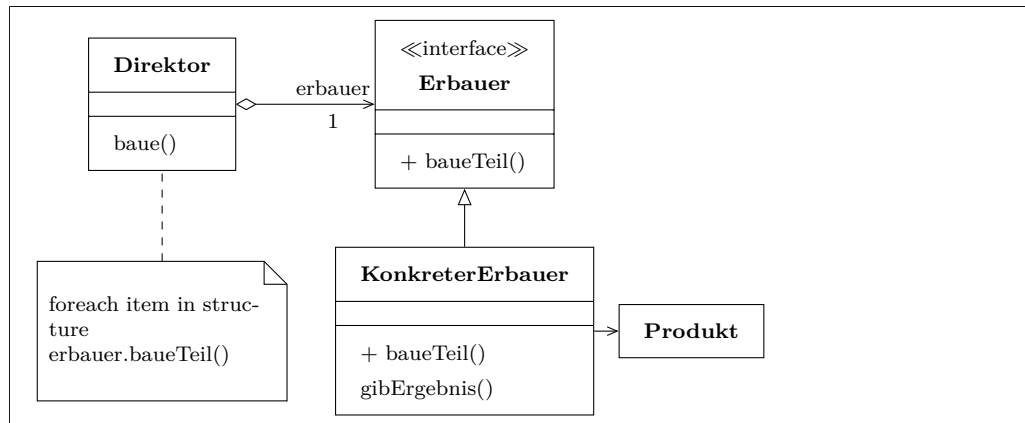
\liEntwurfsEinzelstueck

```
889 \def\liEntwurfsEinzelstueck{
890   \liEntwurfsEinzelstueckUml
891   \liEntwurfsEinzelstueckAkteure
892   \liEntwurfsEinzelstueckCode
893 }
```

2.10.9 Erbauer (Builder)

Quelle nach der deutschen Wikipedia

\liEntwurfsErbauerUml



```

894 \def\liEntwurfsErbauerUml{
895   \begin{tikzpicture}
896     \umlsimpleclass[x=7,y=0]{Produkt}{}{}
897     \umlclass[x=4,y=3,type=interface]{Erbauer}{}{+ baueTeil()}
898     \umlclass[x=4,y=0]{KonkreterErbauer}{}{
899       + baueTeil()\
900       gibErgebnis()}
901     \umlclass[x=0,y=3]{Direktor}{}{baue()}
902
903     \umluniagg[arg2=erbauer,pos2=0.7,mult2=1]{Direktor}{Erbauer}
904     \umluniassoc{KonkreterErbauer}{Produkt}
905     \umlinherit{KonkreterErbauer}{Erbauer}
906
907     \umlnote[x=0,y=0,width=3cm]{Direktor}{
908       foreach item in structure\
909       erbauer.baueTeil()
910     }
911   \end{tikzpicture}
912   \footcite{wiki:erbauer}
913 }

```

\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.

```

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

```

```

925     die von ihm erzeugte Repräsentation des Produkts. Er bietet auch eine
926     Schnittstelle zum Auslesen des Produkts.
927
928     \item[Direktor]
929
930     Der Direktor konstruiert ein komplexes Objekt unter Verwendung der
931     Schnittstelle des Erbauers. Der Direktor arbeitet eng mit dem Erbauer
932     zusammen: Er weiß, welche Baureihenfolge der Erbauer verträgt oder
933     benötigt. Der Direktor entkoppelt somit den Konstruktionsablauf vom
934     Klienten.
935
936     \item[Produkt]
937
938     Das Produkt repräsentiert das zu konstruierende komplexe Objekt.
939     \footcite{wiki:erbauer}
940 \end{description}
941 }

```

\liEntwurfsErbauer

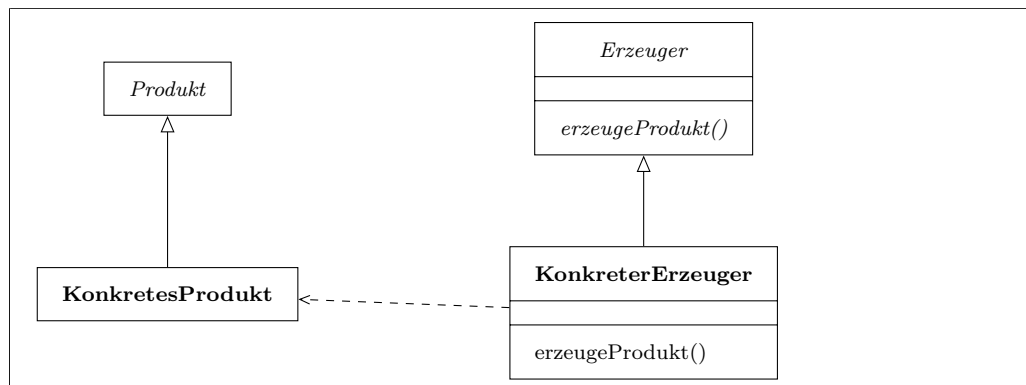
```

942 \def\liEntwurfsErbauer{
943   \liEntwurfsErbauerUml
944   \liEntwurfsErbauerAkteure
945 }

```

2.10.10 Fabrikmethode (Factory Method)

\liEntwurfsFabrikmethodeUml Quelle nach der deutschen Wikipedia



```

946 \def\liEntwurfsFabrikmethodeUml{
947   \begin{tikzpicture}
948     \umlsimpleclass[type=abstract]{Produkt}
949     \umlsimpleclass[below=2cm of Produkt]{KonkretesProdukt}
950     \umlinherit{KonkretesProdukt}{Produkt}
951
952     \umlclass[type=abstract,right=4cm of Produkt]{Erzeuger}{\{
953       \textit{erzeugeProdukt()}\}
954     }
955     \umlclass[below=1.2cm of Erzeuger]{KonkreterErzeuger}{\{
956       erzeugeProdukt()
957     }
958     \umlinherit{KonkreterErzeuger}{Erzeuger}
959
960     \umldep{KonkreterErzeuger}{KonkretesProdukt}
961   \end{tikzpicture}
962 }

```

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).

```

963 \def\liEntwurfsFabrikmethodeAkteure{
964   \begin{description}
965     \item[Produkt]
966
967     Das Produkt ist der Basistyp (Klasse oder Schnittstelle) für das
968     zu erzeugende Produkt.
969
970     \item[KonkretesProdukt]
971
972     KonkretesProdukt implementiert die Produkt-Schnittstelle.
973
974     \item[Erzeuger]
975
976     Der Erzeuger deklariert die Fabrikmethode, um ein solches Produkt
977     zu erzeugen und kann eine Default-Implementierung beinhalten.
978
979     \item[KonkreterErzeuger]
980
981     KonkreterErzeuger überschreibt die Fabrikmethode, um die ihm
982     entsprechenden konkreten Produkte zu erzeugen (z. B. indem er den
983     Konstruktor einer konkreten Produkt-Klasse aufruft).
984
985     \footcite{wiki:fabrikmethode}
986   \end{description}
987 }

```

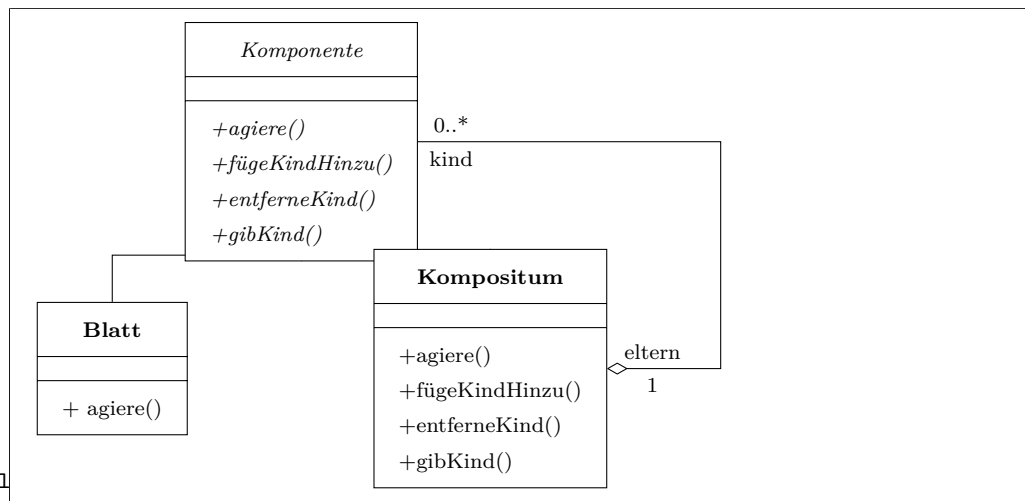
\liEntwurfsFabrikmethode

```

988 \def\liEntwurfsFabrikmethode{
989   \liEntwurfsFabrikmethodeUml
990   \liEntwurfsFabrikmethodeAkteure
991 }

```

2.10.11 Kompositum (Composite)



\liEntwurfsKompositumUml

```

992 \def\liEntwurfsKompositumUml{
993   \begin{tikzpicture}
994     \umlclass[x=2.5,y=3,type=abstract]{Komponente}{-}{
995       \textit{+agiere()}\
996       \textit{+fügeKindHinzu()}\
997       \textit{+entferneKind()}\
998       \textit{+gibKind()}
999     }
1000     \umlclass[x=0]{Blatt}{-}{+ agiere()}
1001     \umlclass[x=5]{Kompositum}{-}{
1002       +agiere()\
1003       +fügeKindHinzu()\
1004       +entferneKind()\
1005       +gibKind()
1006     }
1007
1008     \umlVHVinherit{Kompositum}{Komponente}
1009     \umlVHVinherit{Blatt}{Komponente}
1010     \umlHVVhaggreg[anchor1=east,arm1=1.5cm,arg1=eltern,mult1=1,arg2=kind,mult2=0..*,pos2=2.9,
1011   \end{tikzpicture}
1012 }

```

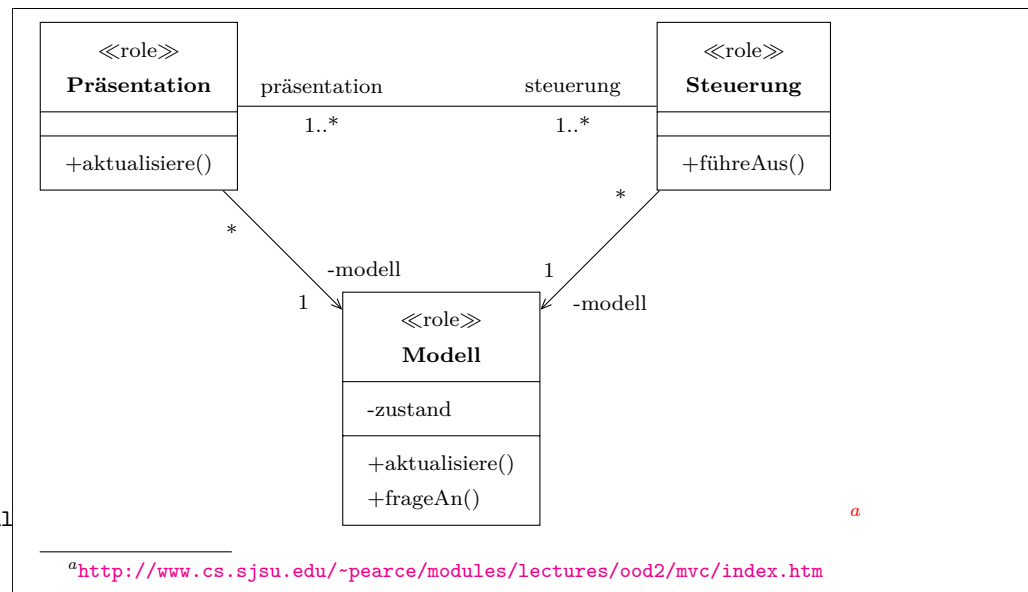
\liEntwurfsFabrikmethode

```

1013 \def\liEntwurfsKompositum{
1014   \liEntwurfsKompositumUml
1015   \liEntwurfsKompositumAkteure
1016 }

```

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



```

1017 \def\liEntwurfsModellPraesentationSteuerungUml{
1018   \begin{tikzpicture}
1019     \umlclass[x=-4cm,type=role]{Präsentation}{-}{+aktualisiere()}
1020     \umlclass[x=4cm,type=role]{Steuerung}{-}{+führeAus()}
1021     \umlclass[y=-4cm,type=role]{Modell}{-}{
1022       -zustand
1023     }{
1024       +aktualisiere()\
1025       +frageAn()
1026     }
1027
1028     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Präsentation}{Modell}
1029     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Steuerung}{Modell}

```

```

1030 \umlassoc[arg1=präsentation,mult1=1..*,arg2=steuerung,mult2=1..*]{Präsentation}{Steuerung}
1031 \end{tikzpicture}
1032 \liFussnoteUrl{http://www.cs.sjsu.edu/~pearce/modules/lectures/ood2/mvc/index.htm}
1033 }

```

ModellPraesentationSteuerung

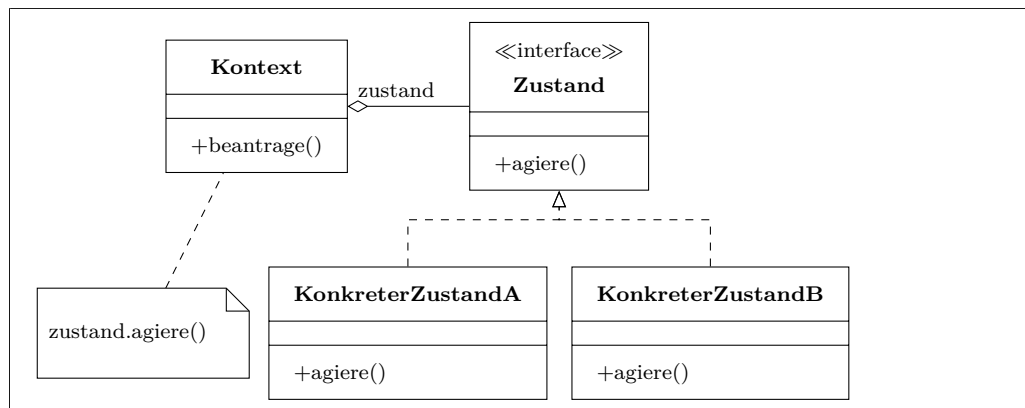
```

1034 \def\liEntwurfs{
1035 \liEntwurfsModellPraesentationSteuerungUml
1036 \liEntwurfsModellPraesentationSteuerungAkteure
1037 }

```

2.10.13 Zustand (State)

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

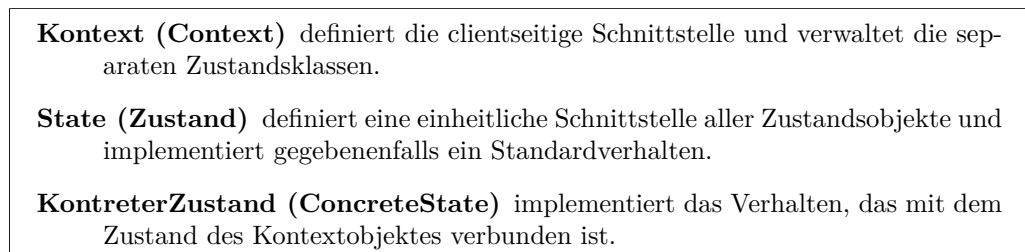


```

1038 \def\liEntwurfsZustandUml{
1039 \begin{tikzpicture}
1040 \umlcclass[x=-1,y=3]{Kontext}{+beantrage()}
1041 \umlcclass[x=3,y=3,type=interface]{Zustand}{+agiere()}
1042 \umlcclass[x=1,y=0]{KonkreterZustandA}{+agiere()}
1043 \umlcclass[x=5,y=0]{KonkreterZustandB}{+agiere()}
1044
1045 \umlVHVreal{KonkreterZustandA}{Zustand}
1046 \umlVHVreal{KonkreterZustandB}{Zustand}
1047
1048 \umlagg[reg,arg=zustand,pos=0.4]{Kontext}{Zustand}
1049
1050 \umlnote[x=-2.5,y=0,width=2.5cm]{Kontext}{zustand.agiere()}
1051 \end{tikzpicture}
1052 }

```

\liEntwurfsZustandAkteure Quelle: Deutsche Wikipedia



```

1053 \def\liEntwurfsZustandAkteure{
1054 \begin{description}
1055 \item[Kontext (Context)]
1056
1057 definiert die clientseitige Schnittstelle und verwaltet die separaten
1058 Zustandsklassen.
1059

```

```

1060     \item[State (Zustand)]
1061
1062     definiert eine einheitliche Schnittstelle aller Zustandsobjekte und
1063     implementiert gegebenenfalls ein Standardverhalten.
1064
1065     \item[KontreterZustand (ConcreteState)]
1066
1067     implementiert das Verhalten, das mit dem Zustand des Kontextobjektes
1068     verbunden ist.
1069 \end{description}
1070 }

\liEntwurfsZustand
1071 \def\liEntwurfsZustand{
1072     \liEntwurfsZustandUml
1073     \liEntwurfsZustandAkteure
1074 }

1075

```

2.11 er.sty

```
1076 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1077 \ProvidesPackage{lehramt-informatik-er}[2020/06/13 Zum Setzen von
1078 ER-Diagrammen]

1079 \RequirePackage{tikz-er2}
1080 \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}

1081 \RequirePackage{soul}
1082 \RequirePackage{fontawesome}

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

1083 \ExplSyntaxOn

\liErEntity
1084 \def\liErEntity#1{\textbf{#1}}

\liErRelationship
1085 \def\liErRelationship#1{\ul{#1}}

\liErAttribute
1086 \def\liErAttribute#1{\emph{#1}}

\liErMpEntity mp = marginpar
Let-Abkürzung: \let\e=\liErMpEntity
1087 \def\liErMpEntity#1{
1088   \liErEntity{#1}
1089   \marginpar{
1090     \liErEntity{\tiny\faSquareO{}}~E:~#1}
1091   }
1092 }
□

\liErMpRelationship Let-Abkürzung: \let\r=\liErMpRelationship
1093 \def\liErMpRelationship#1{
1094   \liErRelationship{#1}
1095   \marginpar{
1096     \liErRelationship{\tiny\faGg{}}~R:~#1}
1097   }
1098 }

\liErMpAttribute Let-Abkürzung: \let\a=\liErMpAttribute
1099 \def\liErMpAttribute#1{
1100   \liErAttribute{#1}
1101   \marginpar{
1102     \liErAttribute{\tiny\faCircleThin{}}~A:~#1}
1103   }
1104 }

\liErDatenbankName Let-Abkürzung: \let\d=\liErDatenbankName
datenbank name
1105 \def\liErDatenbankName#1{
1106   {
1107     \footnotesize\texttt{(#1)}
1108   }
1109 }

1110 \ExplSyntaxOff
1111

```

2.12 formale-sprachen.sty

```

1112 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1113 \ProvidesPackage{lehramt-informatik-formale-sprachen}[2021/02/21 Hilfsmakros
1114 zum Setzen von mathematischen Formeln bei dem Thema Formale Sprachen]

1115 \directlua{
1116   formale_sprachen = require('lehramt-informatik-formale-sprachen')
1117 }

1118 \RequirePackage{hyperref}

1119 \liLadePakete{mathe,typographie}

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

1120 \def\liMengeOhneMathe#1{\{ #1 \}}
1121 \def\liMenge#1{%
1122   \ifmode%
1123     \liMengeOhneMathe{#1}%
1124   \else%
1125     $\liMengeOhneMathe{#1}$%
1126   \fi%
1127 }

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

1128 \def\liEpsilon{$\varepsilon$}

\liPotenzmenge Umgeben mit geschweiften Klammern in einer Mathematik-Umgebung

1129 \def\erzeuge@tiefgestellt#1{\directlua{formale_sprachen.erzeuge_tiefgestellt('#1')}}
1130 \def\liPotenzmengeOhneMathe#1{\{ \erzeuge@tiefgestellt{#1} \}}
1131 \def\liPotenzmenge#1{$\liPotenzmengeOhneMathe{#1}$}

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

1132 \let\liZustandsmengeOhneMathe=\liPotenzmengeOhneMathe
1133 \let\liZustandsmenge=\liPotenzmenge

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

1134 \def\liUeberfuehrungsFunktionOhneMathe#1{\delta(\erzeuge@tiefgestellt{#1})}
1135 \def\liUeberfuehrungsFunktion#1{
1136   \ifmode
1137     \liUeberfuehrungsFunktionOhneMathe{#1}
1138   \else
1139     $\liUeberfuehrungsFunktionOhneMathe{#1}$
1140   \fi
1141 }

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

1142 \def\liAlphabet#1{$\Sigma = \{ #1 \}$}

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

1143 \def\liBandAlphabet#1{$\Gamma = \Sigma \cup \{ #1 \}$}

\liZustandsBuchstabe

1144 \def\liZustandsBuchstabe{z}

\liZustandsBuchstabeGross

1145 \def\liZustandsBuchstabeGross{Z}

```

```

\liZustandsmengeNr
1146 \def\zustandsnamens@liste#1#2{
1147   $
1148   \{
1149     \directlua{formale_sprachen.erzeuge_zustandsnamens_liste('#1', '#2')}
1150   \}
1151   $
1152 }
1153 \def\liZustandsmengeNr#1{\zustandsnamens@liste{\liZustandsBuchstabe}{#1}}

\liZustandsmengeNrGross
1154 \def\liZustandsmengeNrGross#1{\zustandsnamens@liste{\liZustandsBuchstabeGross}{#1}}

\liZustandsname \liZustandsname{1}: $z_1$
1155 \def\liZustandsname#1{\liZustandsBuchstabe_#1$}

\liZustandsnameGross \liZustandsnameGross{1}: $Z_1$
1156 \def\liZustandsnameGross#1{\liZustandsBuchstabeGross_#1$}

\liAbleitung \liAbleitung{S -> aB -> ab}:  $S \vdash aB \vdash ab$ 
1157 \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}

1158 \NewDocumentEnvironment { liProduktionsRegeln }
1159 { 0{P} +b }
1160 {
1161   \liGeschweifteKlammern{#1}
1162   {
1163     \begin{align*}
1164       \directlua{formale_sprachen.produktions_regeln('#2')}
1165     \end{align*}
1166   }{-0.2cm}{-1.5cm}
1167 } {}

\liProduktionen \liProduktionen{S -> A, A -> a}:  $\{S \rightarrow A, A \rightarrow a\}$ 
1168 \def\liProduktionen#1{
1169   \liMenge{\directlua{formale_sprachen.produktions_regeln('#1', true)}}
1170 }

\liZustandsnameTiefgestelltt Automatisch tiefgestellte Nummerierung \z1
Let-Abkürzung: \let\z=\liZustandsnameTiefgestelltt
1171 \def\liZustandsnameTiefgestelltt#1{
1172   \ifmmode
1173     \liZustandsBuchstabe\sb{#1}
1174   \else
1175     $\liZustandsBuchstabe\sb{#1}$
1176   \fi
1177 }

1178 \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}

```



```

1179 \NewDocumentCommand{ \liAusdruck } { O{L} m m } {
1180   $
1181   \tl_if_empty:nTF {#1} {} {#1 =}
1182   \{
1183     \, #2 \,
1184     |
1185     \, #3 \,
1186   \}$
1187 }
1188 \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

1189 \def\liFlaci#1{%
1190   \par
1191   {%
1192     \scriptsize
1193     Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte
1194     Automaten, Compiler und Interpreter) Ein Projekt der Hochschule
1195     Zittau/Görlitz und der Pädagogischen Hochschule Schwyz:
1196     \href{https://flaci.com/#1}{flaci.com/#1}%
1197   }%
1198   \par
1199 }

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

1200 \ExplSyntaxOn
1201 \NewDocumentCommand {\liGrammatik} { O{G} m } {
1202   \tl_set:Nn \l_variablen_tl {V}
1203   \tl_set:Nn \l_alphabet_tl {\Sigma}
1204   \tl_set:Nn \l_produktionen_tl {P}
1205   \tl_set:Nn \l_start_tl {S}
1206
1207   \keys_define:nn { grammatik } {
1208     variablen .code:n = {\tl_set:Nn \l_variablen_tl {\liMenge{##1}}},
1209     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
1210     produktionen .code:n = {\tl_set:Nn \l_produktionen_tl {\liProduktionen{##1}}},
1211     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
1212   }
1213
1214   \keys_set:nn { grammatik } { #2 }
1215
1216   $#1 = (
1217     \l_variablen_tl,
1218     \l_alphabet_tl,
1219     \l_produktionen_tl,
1220     \l_start_tl
1221   )$
1222 }
1223 \ExplSyntaxOff
1224

```

2.13 formatierung.sty

```
1225 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1226 \ProvidesPackage{lehramt-informatik-formatierung}[2020/11/27]
```

2.13.1 Schriftarten / Typographie

```
1227 \RequirePackage{mathpazo}
1228 \RequirePackage[no-math]{fontspec}
1229 \setmainfont{texgyrepagella}
1230 \setsansfont{QTAncientOlive}
1231 \RequirePackage{sectsty}
1232 \allsectionsfont{\fontspec{QTAncientOlive}}
```

2.13.2 Farben

```
1233 \RequirePackage{xcolor}
1234 \definecolor{infogray}{rgb}{0.97,0.97,0.97}
```

2.13.3 Überschriften

```
1235 \RequirePackage{titlesec}
1236 \titleformat{\chapter}[display]{\sffamily\bfseries}{\Opt}{\LARGE}
1237 \titlespacing{\chapter}{0pt}{0pt}{*1}
1238 \titleformat{\paragraph}[hang]{\sffamily\normalsize\bfseries}{\theparagraph}{1em}{}
1239 \setcounter{secnumdepth}{0}
```

2.13.4 Listen

```
1240 \RequirePackage{paralist}
1241 \renewcommand\labelitemi{-}
1242 \renewcommand\labelitemii{-}
1243 \renewcommand\labelitemiii{-}
1244 \renewcommand\labelitemiv{-}
1245 % Counter: enumi enumii enumiii enumiv
1246 % Styles: \arabic{counter} \alph{counter} \Alph{counter} \roman{counter} \Roman{counter}
1247 \renewcommand{\labelenumi}{(\alph{enumi})}
1248 \renewcommand{\labelenumii}{(\roman{enumii})}
```

2.13.5 Kasten

```
1249 \RequirePackage{mdframed}
1250 \mdfsetup{backgroundcolor=infogray}
```

liKasten

```
1251 \NewDocumentEnvironment { liKasten }{ } {
1252   \begin{mdframed}[backgroundcolor=white!0]
1253 } {
1254   \end{mdframed}
1255 }
```

2.13.6 Header

```
1256 \RequirePackage{fancyhdr}
1257 \fancyhead[L,C,R]{}
1258 \fancyfoot[L]{}
1259 \fancyfoot[C]{}
1260 \fancyfoot[R]{\thepage}
1261 \pagestyle{fancy}
1262 \renewcommand{\headrulewidth}{0pt}
1263 \renewcommand{\footrulewidth}{0pt}
1264
```

2.14 gantt.sty

```

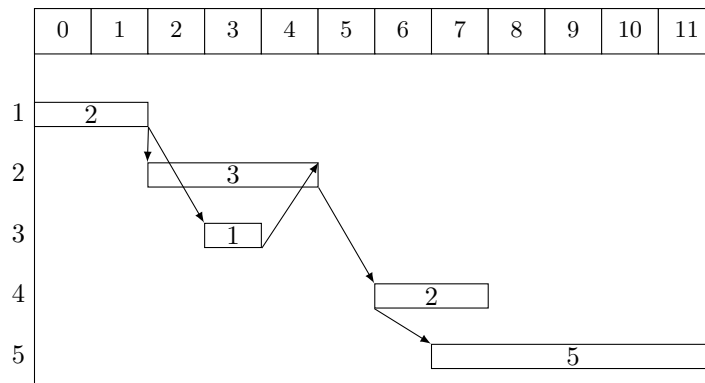
1265 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1266 \ProvidesPackage{lehramt-informatik-gantt}[2020/09/05]

\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}

```



```

1267 \RequirePackage{tikz-uml}
1268 \RequirePackage{pgfgantt}
1269 \setganttlinklabel{f-s}{}
1270 \setganttlinklabel{s-s}{}
1271 \setganttlinklabel{f-f}{}
1272 \setganttlinklabel{s-f}{}
1273

```

2.15 grafik.sty

```
1274 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1275 \ProvidesPackage{lehramt-informatik-grafik}[2020/12/27 Lädt das Paket
1276 TikZ, um Grafiken zeichnen zu können.]
1277 \RequirePackage{tikz}
1278
```

2.16 graph.sty

```

1279 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1280 \ProvidesPackage{lehramt-informatik-graph}[2020/06/09]
1281 \RequirePackage{tikz}

```

Für die die Adjazenzliste (`xrightarrow`)

```

1282 \RequirePackage{amsmath}

```

Für Adjazenz-Matrix

```

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

```

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

```

1283 \RequirePackage{blkarray}
1284 \usetikzlibrary{arrows.meta}

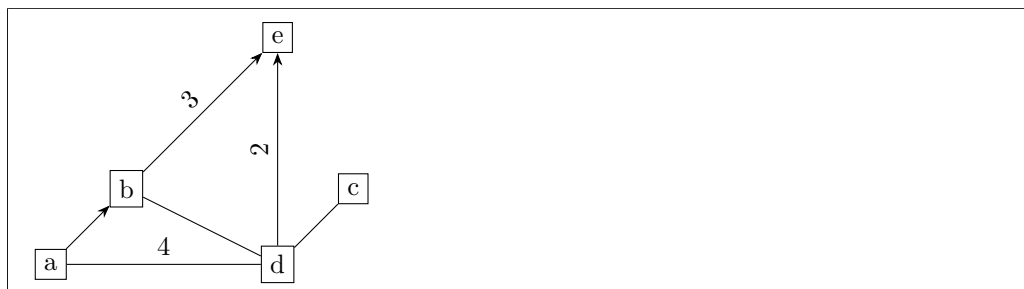
```

```

\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}

```



```

1285 \tikzset{
1286   li graph/.style={
1287     every node/.style={
1288       rectangle,
1289       draw,
1290     },
1291     every edge/.style={
1292       >={Stealth[black]},
1293       draw,
1294     },
1295     every edge/.append style={
1296       every node/.style={
1297         sloped,
1298         auto,
1299       }
1300     }
1301   },
1302   li markierung/.style={
1303     ultra thick,
1304   }
1305 }

```

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}

```

```

1306 \NewDocumentEnvironment { liGraphenFormat }{ +b } {} {}

```

```

1307

```

2.17 hanoi.sty

1308 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 1309 \ProvidesPackage{lehramt-informatik-hanoi}[2020/12/19 Paket zum Setzen
 1310 von Hanoi-Grafiken]

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

```
1311 \RequirePackage{tikz}
1312 \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}

1313 \def\li@mset #1[#2]=#3{%
1314 \expandafter\xdef\csname #1#2\endcsname{#3}
1315 }
1316 \def\li@mget #1[#2]{%
1317 \csname #1#2\endcsname
1318 }
1319 \def\li@minc #1[#2] += #3{%
1320 \pgfmthparse{\li@mget #1[#2] + #3}%
1321 \li@mset #1[#2] = \pgfmthresult
1322 }
1323
1324 \def\liHanoi#1#2{
1325   \edef\li@numdiscs{#1}
1326   \def\li@sequence{#2}
1327   \begin{tikzpicture}[line width=4mm,brown!40,line cap=round,xscale=3]
1328     % init colors
1329     \foreach[count=\j] \c in {red,green!80!black,blue,orange,violet,gray,yellow!80!black,purple!80!black}{
1330       \li@mset col[\j]={\c};
1331     }
1332     % draw poles and init pole counters
1333     \foreach \j in {1,2,3}{
1334       \li@mset pos[\j]=0
1335       \draw (\j,-.5) -- +(0, .5 + .5*\li@numdiscs);
1336     }
1337     % draw base
1338     \draw (.5,-.5) -- +(3,0);
1339     % draw discs
1340     \foreach[count=\k] \i/\j in \li@sequence{
1341       \draw[draw={\li@mget col[\i]}] (\j,\li@mget pos[\j]) +(-.4*\i/\li@numdiscs,0) -- +(.4*\i/\li@numdiscs,0);
1342       \li@minc pos[\j] += {.5}
1343     }
1344   \end{tikzpicture}
1345 }
```

2.18 klassen-konfiguration-aufgabe.sty

```
1346 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1347 \ProvidesPackage{lehramt-informatik-klassen-konfiguration-aufgabe}[2021/09/01 Die
1348 Klasse Aufgabe konfigurieren, d. h. Laden von einigen Paketen, Registrierung von Hooks]
1349 \liLadePakete{mathe}
1350 \ExplSyntaxOn
1351 \AddToHook{enddocument}{
1352 Github \_gib_github_url:
1353 }
1354 \ExplSyntaxOff
1355
```


2.19 komplexitaetstheorie.sty

```

1356 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1357 \ProvidesPackage{lehramt-informatik-komplexitaetstheorie}[2021/07/08 Zum
1358 Setzen von Karps NP-vollständigen Problemen, Hilfsmakros für die
1359 Polynomialzeitreduktion.]

```

Let-Abkürzungen

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

1360 \liLadePakete{mathe}

Für das Makro `\liProblemBeschreibung` benötigt.

```
1361 \RequirePackage{mdframed}
```

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

```
1362 \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
```

1363 \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

```

1364 \def\liProblemBeschreibung#1#2#3{
1365   \begin{mdframed}[
1366     userdefinedwidth=9cm,
1367     align=center,
1368     backgroundcolor=white!0,
1369   ]
1370     \centerline{\large\liProblemName{#1}}
1371
1372     \medskip
1373
1374     \begin{description}
1375       \item[Gegeben:] #2
1376       \item[Frage:] #3
1377     \end{description}
1378   \end{mdframed}
1379 }
```

```

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

1380 \NewDocumentCommand{ \liPolynomiellReduzierbar } { m O{p} m } {
1381 \begin{displaymath}
1382 \liProblemName{#1}
1383 \preceq_{#2}
1384 \liProblemName{#3}
1385 \end{displaymath}
1386 }

\liProblemVertexCover

1387 \def\liProblemClique{%
1388 Das \textbf{Cliquesproblem} fragt nach der Existenz einer Clique der
1389 Mindestgröße  $n$  in einem gegebenen Graphen.
1390 \footcite[Seite 76]{theo:fs:4}
1391 Eine Clique ist eine Teilmenge von Knoten in einem ungerichteten
1392 Graphen, bei der \emph{jedes Knotenpaar durch eine Kante} verbunden ist.
1393 \footcite{wiki:cliquenproblem}
1394 }

\liProblemVertexCover

1395 \def\liProblemVertexCover{%
1396 %
1397 Das \textbf{Knotenüberdeckungsproblem} (\liProblemName{Vertex Cover})
1398 fragt, ob zu einem gegebenen einfachen Graphen und einer natürlichen
1399 Zahl  $k$  eine Knotenüberdeckung der Größe von höchstens  $k$  existiert.
1400 \footcite{wiki:knotenueberdeckung}
1401
1402 Das heißt, ob es eine aus maximal  $k$  Knoten bestehende Teilmenge  $U$ 
1403 der Knotenmenge gibt, so dass jede Kante des Graphen mit
1404 mindestens einem Knoten aus  $U$  verbunden ist.
1405 \footcite[Seite 78]{theo:fs:4}%
1406 }

\liProblemSubsetSum Kein Karp-21-Problem

1407 \def\liProblemSubsetSum{%
1408 Das \textbf{Teilsommenproblem} (\liProblemName{Subset Sum} oder
1409 \liProblemName{SSP}) ist ein spezielles Rucksackproblem.
1410 \footcite{wiki:teilsommenproblem}
1411 Gegeben sei eine Menge von ganzen Zahlen  $I = \{w_1, w_2, \dots, w_n\}$ .
1412 Gesucht ist eine Untermenge, deren Elementsumme maximal, aber nicht
1413 größer als eine gegebene obere Schranke  $c$  ist.
1414 \footcite[Seite 74]{theo:fs:4}
1415 }

\liProblemSubsetSum Kein Karp-21-Problem

1416 \def\liProblemSat{%
1417 Das \textbf{Erfüllbarkeitsproblem der Aussagenlogik} \liProblemName{Sat}
1418 und \liProblemName{k-SAT} mit  $k \geq 3$ ,  $k \in \mathbb{N}$  (Satz von
1419 Cook) fragt, ob eine aussagenlogische Formel erfüllbar
1420 ist.\footcite{wiki:sat} Das Erfüllbarkeitsproblem der
1421 \emph{Aussagenlogik} ist in exponentieller Zeit in Abhängigkeit der
1422 Anzahl der Variablen mit Hilfe einer Wahrheitstabelle entscheidbar.
1423 Diese \emph{Wahrheitstabelle} kann nicht in polynomieller Zeit
1424 aufgestellt werden.
1425 \footcite[Seite 71]{theo:fs:4}
1426 }

1427

```

2.20 kontrollflussgraph.sty

```
1428 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1429 \ProvidesPackage{lehramt-informatik-kontrollflussgraph}[2020/11/07]

1430 \RequirePackage{tikz}
1431 \usetikzlibrary{positioning}
1432 \tikzset{
1433   li kontrollfluss/.style={
1434     knoten/.style={
1435       circle,
1436       draw
1437     },
1438     usebox/.style={
1439       draw,
1440       rectangle,
1441       font=\scriptsize,
1442       anchor=west,
1443       align=left,
1444     },
1445     bedingung/.style={
1446       midway,
1447       draw=none,
1448       font=\scriptsize
1449     },
1450     knotenbeschriftung/.style={
1451       draw,
1452       rectangle,
1453       midway,
1454       font=\scriptsize
1455     },
1456     wahr/.style={
1457       thick
1458     },
1459     falsch/.style={
1460       dashed
1461     },
1462     every node/.style={
1463       circle,
1464       draw,
1465     },
1466     every edge/.append style={
1467       every node/.style={
1468         draw=none,
1469         bedingung,
1470       }
1471     },
1472     every path/.style={
1473       draw,
1474       ->,
1475     },
1476     every pin/.style={
1477       draw,
1478       dotted,
1479       rectangle,
1480       pin position=right
1481     },
1482     every pin edge/.style={
1483       dotted,
1484       arrows=-,
1485     }
1486   }
1487 }
```

liKontrollflussgraph

```

1488 \NewDocumentEnvironment { liKontrollflussgraph } { 0{ } } {
1489   \begin{tikzpicture}[
1490     li kontrollfluss,
1491     #1
1492   ]
1493 } {
1494   \end{tikzpicture}
1495 }

\liAnweisung
1496 \def\liAnweisung#1(#2,#3){\node[knoten] at (#2,#3) (#1) {#1};}

\liBedingung Let-Abkürzung: \let\b=\liBedingung
1497 \def\liBedingung#1#2{node[bedingung,#1]{\texttt{#2}}}}

\liBedingungWahr Let-Abkürzung: \let\bWahr=\liBedingungWahr
1498 \def\liBedingungWahr#1{node[bedingung,#1]{\texttt{[true]}}}}

\liBedingungFalsch Let-Abkürzung: \let\bFalsch=\liBedingungFalsch
1499 \def\liBedingungFalsch#1{node[bedingung,#1]{\texttt{[false]}}}}

\liKontrollCode Let-Abkürzung: \let\c=\liKontrollCode
1500 \def\liKontrollCode#1{{\tiny\texttt{\textcolor{gray}{#1}}}}

\liKontrollTextzeileKnoten Let-Abkürzung: \let\k=\liKontrollTextzeileKnoten
1501 \def\liKontrollTextzeileKnoten#1{\raisebox{-2pt}{\tikz[scale=0.5,transform shape] \node[draw

\liKontrollKnotenPfad Let-Abkürzung: \let\p=\liKontrollKnotenPfad
1502 \ExplSyntaxOn
1503 \NewDocumentCommand { \liKontrollKnotenPfad }{ m }
1504 {
1505   \seq_set_split:Nnn \l_tmpa_seq { - } { #1 }
1506   \seq_set_map:Nnn \l_tmpa_seq \l_tmpa_seq {\liKontrollTextzeileKnoten{##1}}
1507   \seq_use:Nn \l_tmpa_seq { ~~~~ }
1508 }
1509 \ExplSyntaxOff

1510

```

2.21 kopf-fusszeilen.sty

```
1511 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1512 \ProvidesPackage{lehramt-informatik-kopf-fusszeilen}[2021/08/20 Kopf-
1513 und Fußzeilen mit Hilfe des Pakets fancyhdr.]

1514 \ExplSyntaxOn

1515 \fancyhead{}
1516 \fancyhead[R0,LE]{\scriptsize\LehramtInformatikTitel}
1517 \fancyhead[L0,RE]{\scriptsize\today}
1518 \fancyfoot{}
1519 \fancyfoot[LE,R0]{\thepage}
1520 \fancyfoot[L0,CE]{\scriptsize\LehramtInformatikAutorName}
1521 \fancyfoot[CO,RE]{\scriptsize\LehramtInformatikAutorEmail}
1522 \renewcommand{\headrulewidth}{0.4pt}
1523 \renewcommand{\footrulewidth}{0.4pt}

1524 \ExplSyntaxOff

1525
```

2.22 literatur-dummy.sty

```
1526 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1527 \ProvidesPackage{lehramt-informatik-literatur-dummy}[2020/11/27]

\literatur

1528 \def\literatur{}

\footcite

1529 % \RequirePackage[stable,multiple]{footmisc}
1530 \NewDocumentCommand{ \footcite } { o m }{}

1531
```

2.23 literatur.sty

```
1532 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1533 \ProvidesPackage{lehramt-informatik-literatur}[2020/11/27]

1534 \RequirePackage{csquotes}
1535 \RequirePackage[
1536   bibencoding=utf8,
1537   citestyle=authortitle,
1538   backend=biber,
1539 ]{biblatex}
1540 \addbibresource{\LehramtInformatikRepository/Quellen/10_DB.bib}
1541 \addbibresource{\LehramtInformatikRepository/Quellen/20_OOMUP.bib}
1542 \addbibresource{\LehramtInformatikRepository/Quellen/30_AUD.bib}
1543 \addbibresource{\LehramtInformatikRepository/Quellen/40_SOSY.bib}
1544 \addbibresource{\LehramtInformatikRepository/Quellen/60_FUMUP.bib}
1545 \addbibresource{\LehramtInformatikRepository/Quellen/70_THEO.bib}
1546 \addbibresource{\LehramtInformatikRepository/Quellen/80_DDI.bib}
1547 \addbibresource{\LehramtInformatikRepository/Quellen/Allgemein.bib}
1548 \addbibresource{\LehramtInformatikRepository/Quellen/Examen.bib}
1549 \addbibresource{\LehramtInformatikRepository/Quellen/Wikipedia.bib}
1550 % To allow footnotes in the heading
1551 \RequirePackage[stable,multiple]{footmisc}

\literatur

1552 \def\literatur{\printbibliography[heading=subbibliography]}

1553
```

2.24 makros.sty

```
1554 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1555 \ProvidesPackage{lehramt-informatik-makros}[2020/11/27 Eine Sammlung von
1556 häufig verwendeten Makros und Umgebungen, die thematisch zu keinem
1557 anderen Paket passen]
1558 \RequirePackage{hyperref}
1559 \RequirePackage{graphicx}
    Für die Umgebung liQuellen benötigt.
1560 \RequirePackage{paralist}

\inhaltsverzeichnis
1561 \def\inhaltsverzeichnis {
1562   \begin{mdframed}
1563     \begin{group}
1564       \let\clearpage\relax
1565       \tableofcontents
1566     \end{group}
1567   \end{mdframed}
1568 }

\memph \mephm (\marginpar and \emph)
1569 \newcommand{\memph}[1]{\emph{#1}\marginpar{\tiny#1}}

\SLASH
1570 \newcommand\SLASH{\char`\\}

\liPseudoUeberschrift Text, der sich wie eine Überschrift verhält.
1571 \newcommand{\liPseudoUeberschrift}[1]{
1572   \bigskip
1573   \noindent
1574   \textsf{\textbf{#1}}
1575   \noindent
1576 }

\liBeschriftung Ähnlich dem Makro \liPseudoUeberschrift{}. Am Ende des Textes wird ein Doppelpunktzeichen angehängt.
1577 \newcommand{\liBeschriftung}[1]{
1578   \par
1579   \noindent
1580   \medskip
1581   \textbf{#1}:
1582   \medskip
1583   \noindent
1584 }

\hinweis
1585 \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.
1586 \NewDocumentEnvironment { liProjektSprache }{ o +b } {} {}

liEinbettung
1587 \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.
1588 \RequirePackage{xparse}
1589 \ExplSyntaxOn
```



```

1590 \NewDocumentEnvironment{ liAntwort } { 0{standard} }
1591 {
1592   \str_case:nn {#1} {
1593     {standard} {
1594       \def\beschriftung{}
1595       \mdfsetup{backgroundcolor=gray!1,linecolor=gray}
1596     }
1597     {richtig} {
1598       \def\beschriftung{richtig}
1599       \mdfsetup{backgroundcolor=green!5,linecolor=green}
1600     }
1601     {falsch} {
1602       \def\beschriftung{falsch}
1603       \mdfsetup{backgroundcolor=red!3,linecolor=red}
1604     }
1605     {muster} {
1606       \def\beschriftung{Musterlösung}
1607       \mdfsetup{backgroundcolor=green!12,linecolor=green}
1608     }
1609   }
1610   \ifx\beschriftung\empty\else
1611     \noindent
1612     \textbf{\beschriftung{:}}
1613   \fi
1614   \begin{mdframed}
1615 }
1616 {\end{mdframed}}

```

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

```

1617 \NewDocumentEnvironment{ liAdditum } { o }
1618 {
1619   \begin{mdframed}[backgroundcolor=yellow!5]
1620     \IfNoValueTF {#1}
1621     { \liPseudoUeberschrift{Additum} }
1622     { \liPseudoUeberschrift{Additum:~#1} }
1623   }
1624 {\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.

```

1625 \NewDocumentEnvironment{ liExkurs }{o +b}{
1626   \vspace{0.2cm}%
1627   \begin{mdframed}[
1628     backgroundcolor=white,
1629     bottomline=false,
1630     innermargin=1cm,
1631     leftline=true,
1632     linecolor=black,
1633     linewidth=0.1cm,
1634     outermargin=1cm,
1635     rightline=false,
1636     topline=false,
1637   ]

```

```

1638 \footnotesize
1639 \noindent%
1640 \textbf{Exkurs:~#1}\par%
1641 \noindent%
1642 #2
1643 \end{mdframed}
1644 \vspace{0.2cm}
1645 }{}

```

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

```

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

```

Weiterführende Literatur:

- Quelle 1
- Quelle 2

```

1646 \cs_new:Npn \listen@punkt #1 {\item #1}
1647 \NewDocumentEnvironment { liQuellen }{ +b }
1648 {
1649   \seq_clear_new:N \l_quellen
1650   \seq_set_split:Nnn \l_quellen {\item} {#1}
1651   \seq_remove_all:Nn \l_quellen {}
1652   \begin{mdframed}[backgroundcolor=white!0]
1653     \footnotesize
1654     \noindent
1655     \textsf{\textbf{Weiterführende~Literatur:}}
1656     \medskip
1657     \begin{compactitem}
1658       \seq_map_function:NN \l_quellen {\listen@punkt}
1659     \end{compactitem}
1660   \end{mdframed}
1661   %
1662   \makeatletter
1663   \par\@afterindentfalse\@afterheading
1664   \makeatother
1665 } {}

```

liLernkartei

```

1666 \NewDocumentEnvironment { liLernkartei }{ m +b }
1667 {
1668   \begin{mdframed}
1669     \footnotesize
1670     \noindent%
1671     \textbf{Lernkarteikarte:~#1}\par%
1672     \noindent%
1673     #2
1674   \end{mdframed}
1675 } {}

```

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

```

1676 \NewDocumentEnvironment { liDiagramm }{ m +b }
1677 {
1678   \begin{mdframed}[backgroundcolor=white!0]
1679     \small
1680     \noindent%
1681     \textit{#1}:
1682     \begin{center}

```

```

1683   #2
1684   \medskip
1685   \end{center}
1686   \end{mdframed}
1687 } {}
1688 \ExplSyntaxOff

\liFussnoteUrl \liFussnoteUrl[\langle zusätzlicher-text \rangle]{\langle url \rangle} \liFussnoteUrl[zusätzlicher Text]{url}:
Eine HTTP-URL als Fußnote setzen.
1689 \NewDocumentCommand{\liFussnoteUrl} { o m } {
1690   \footnote{\url{#2}\IfNoValueTF{#1}{}{ ( #1 ) }}
1691 }
1692

\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.
1693 \NewDocumentCommand{\liFussnoteLink} { o m m } {
1694   \footnote{\href{#3}{#2}\IfNoValueTF{#1}{}{ ( #1 ) }}
1695 }

\zB
1696 \def\zB{z.\.,B. }

\ZB
1697 \def\ZB{Z.\.,B. }

\dh
1698 \def\dh{d.\.,h. }

1699

```

2.25 master-theorem.sty

1700 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

1701 \ProvidesPackage{lehramt-informatik-master-theorem}[2021/04/13]

Let-Abkürzungen

\let\0=\li0

\let\o=\li0omega

\let\T=\liT

\let\t=\liTheta

\liMasterVariablenDeklaration

{3} % a

{3} % b

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

\liMasterFallRechnung

% 1. Fall

{für $\varepsilon = 4$: \}

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

% 2. Fall

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

% 3. Fall

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

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

\liMasterVariablenDeklaration

{ } % a

{ } % b

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

\liMasterFallRechnung

% 1. Fall

{ }

% 2. Fall

{ }

% 3. Fall

{ }

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

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

1702 \ExplSyntaxOn

1703 \RequirePackage{amsmath}

\liRundeKlammer

1704 \def\liRundeKlammer#1{

1705 \negthinspace \left(#1 \right)

1706 }

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

1707 \def\liThetaOhneMathe#1{

1708 \Theta \liRundeKlammer{#1}

1709 }

1710 \def\liTheta#1{

1711 \ifmmode

1712 \liThetaOhneMathe{#1}

1713 \else

1714 \$\liThetaOhneMathe{#1}\$

1715 \fi

1716 }

```

\liOmega \liOmega{n^2}:  $\Omega(n^2)$ 
1717 \def\liOmegaOhneMathe#1{
1718 \Omega \liRundeKlammer{#1}
1719 }
1720 \def\liOmega#1{
1721 \ifmmode
1722 \liOmegaOhneMathe{#1}
1723 \else
1724 $\liOmegaOhneMathe{#1}$
1725 \fi
1726 }

\liO \liO{n^2}:  $\mathcal{O}(n^2)$ 
1727 \def\liOOhneMathe#1{
1728 \mathcal{O} \liRundeKlammer{#1}
1729 }
1730 \def\liO#1{
1731 \ifmmode
1732 \liOOhneMathe{#1}
1733 \else
1734 $\liOOhneMathe{#1}$
1735 \fi
1736 }

\liT Let-Abkürzung: \let\T=\liT
\liT{16}{2}:  $16 \cdot T\left(\frac{n}{2}\right)$  \liT{}{2}:  $T\left(\frac{n}{2}\right)$ 
1737 \def\liTOhneMathe#1#2{
1738 \tl_if_blank:nTF {#1}
1739 {}
1740 {#1 \cdot }
1741 T
1742 \liRundeKlammer{\frac{n}{#2}}
1743 }
1744 \def\liT#1#2{
1745 \ifmmode
1746 \liTOhneMathe{#1}{#2}
1747 \else
1748 $\liTOhneMathe{#1}{#2}$
1749 \fi
1750 }

\liRekursionsGleichung \liRekursionsGleichung:  $T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$ 
1751 \def\liRekursionsGleichung{
1752 $T(n) = \liT{a}{b} + f(n)$
1753 }

\liBedingungEins \liBedingungEins:  $f(n) \in \mathcal{O}(n^{\log_b a - \varepsilon})$ 
1754 \def\liBedingungEins{
1755 $f(n) \in \liO{n^{\log\sb{b}a - \varepsilon}}$
1756 }

\liBedingungZwei \liBedingungZwei:  $f(n) \in \Theta(n^{\log_b a})$ 
1757 \def\liBedingungZwei{
1758 $f(n) \in \liTheta{n^{\log\sb{b}a}}$
1759 }

\liBedingungDrei \liBedingungDrei:  $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ 
1760 \def\liBedingungDrei{
1761 $f(n) \in \liOmega{n^{\log\sb{b}a + \varepsilon}}$
1762 }

1763 \ExplSyntaxOff

```

\liMasterVariablen

```

1764 \def\liMasterVariablen{
1765   \begin{displaymath}
1766     T(n) = \liT{a}{b} + f(n)
1767   \end{displaymath}
1768
1769   \begin{itemize}
1770     \item[$a = $]
1771       Anzahl der Unterprobleme in der Rekursion
1772
1773     \item[$\textstyle{\frac{1}{b}} = $]
1774       Teil des Originalproblems, welches wiederum durch alle Unterprobleme
1775       repräsentiert wird
1776
1777     \item[$f(n) = $]
1778       Kosten (Aufwand, Nebenkosten), die durch die Division des Problems und
1779       die Kombination der Teillösungen entstehen
1780   \end{itemize}
1781   \footcite{wiki:master-theorem}
1782   \footcite[Seite 19-35 (PDF 11-24)]{aud:fs:2}
1783 }

```

\liMasterFaelle

```

1784 \def\liMasterFaelle{
1785   \begin{description}
1786     \item[1. Fall:]
1787        $T(n) \in \liTheta{n^{\log_{sb{b}}a}}$ 
1788
1789       \hfill falls \liBedingungEins
1790       für  $\varepsilon > 0$ 
1791
1792     \item[2. Fall:]
1793        $T(n) \in \liTheta{n^{\log_{sb{b}}a} \cdot \log n}$ 
1794
1795       \hfill falls \liBedingungZwei
1796
1797     \item[3. Fall:]
1798        $T(n) \in \liTheta{f(n)}$ 
1799
1800       \hfill falls \liBedingungDrei
1801       für  $\varepsilon > 0$ 
1802       und ebenfalls für ein  $c$  mit  $0 < c < 1$  und alle hinreichend großen  $n$ 
1803       gilt:
1804        $a \cdot f(\textstyle{\frac{n}{b}}) \leq c \cdot f(n)$ 
1805     \end{description}
1806 }

```

\liMasterVariablenDeklaration

```

1807 \def\liMasterVariablenDeklaration#1#2#3{
1808   \begin{description}
1809     \item[Allgemeine Rekursionsgleichung:] \strut
1810
1811     \liRekursionsGleichung
1812
1813     \item[Anzahl der rekursiven Aufrufe ( $a$ ):] \strut
1814
1815     #1
1816
1817     \item[Anteil Verkleinerung des Problems ( $b$ ):] \strut
1818
1819     um  $\frac{1}{#2}$  also  $b = #2$ 
1820
1821     \item[Laufzeit der rekursiven Funktion ( $f(n)$ ):] \strut

```

```

1822
1823     $#3$
1824
1825     \item[Ergibt folgende Rekursionsgleichung:] \strut
1826
1827      $T(n) = T_{\#1}^{\#2} + \#3$ 
1828     \end{description}
1829 }

\liMasterFallRechnung
1830 \def\liMasterFallRechnung#1#2#3{
1831     \begin{description}
1832     \item[1. Fall:] \liBedingungEins:
1833
1834     #1
1835
1836     \item[2. Fall:] \liBedingungZwei:
1837
1838     #2
1839
1840     \item[3. Fall:] \liBedingungDrei:
1841
1842     #3
1843     \end{description}
1844 }

\liMasterExkurs
1845 \def\liMasterExkurs{
1846     \begin{liExkurs}[Master-Theorem]
1847     \liMasterVariablen
1848
1849     \noindent
1850     Dann gilt:
1851
1852     \liMasterFaelle
1853     \end{liExkurs}
1854 }

\liMasterWolframLink Link zu Wolframalpha (+ durch Prozent 2B ersetzen)
1855 \def\liMasterWolframLink#1{
1856     Berechne die Rekursionsgleichung auf WolframAlpha:
1857     \href{https://www.wolframalpha.com/input/?i=\#1}{WolframAlpha}
1858 }

1859

```

2.26 mathe.sty

```
1860 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1861 \ProvidesPackage{lehramt-informatik-mathe}[2020/06/10]
1862
1863 % for example \ltimes \rtimes
1864 %\RequirePackage{amssymb}
1865 \RequirePackage{amsmath}
1866
1867 %%
1868 % \mlq \mrq
1869 %%
1870 \DeclareMathSymbol{\mlq}{\mathord}{operators}{``}
1871 \DeclareMathSymbol{\mrq}{\mathord}{operators}{`}
1872
```


2.27 minimierung.sty

```

1873 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1874 \ProvidesPackage{lehramt-informatik-mathe}[2021/03/13 Für den
1875 Minimierungsalgorithmus von einem NEA zu einem DEA]

1876 \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

1877 \def\liFussnote#1{\$x_{#1}\$}

1878 \def\li@fussnote@text#1#2{
1879 \liFussnote{#1}
1880 \quad
1881 {\footnotesize #2}
1882 }

\liFussnoteEinsText

1883 \def\liFussnoteEinsText{
1884 \li@fussnote@text{1}
1885 {Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.}
1886 }

\liFussnoteZweiText

1887 \def\liFussnoteZweiText{
1888 \li@fussnote@text{2}
1889 {Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt.}
1890 }

\liFussnoteDreiText

1891 \def\liFussnoteDreiText{
1892 \li@fussnote@text{3}

```

```

1893 {In weiteren Iterationen markierte Zustände.}
1894 }

\liFussnoteVierText
1895 \def\liFussnoteVierText{
1896   \li@fussnote@text{4}
1897   {...}
1898 }

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



1899 \def\liFussnoten{
1900   \bigskip
1901
1902   \noindent
1903   \liFussnoteEinsText
1904
1905   \noindent
1906   \liFussnoteZweiText
1907
1908   \noindent
1909   \liFussnoteDreiText
1910
1911   \noindent
1912   \liFussnoteVierText
1913 }

\liLeereZelle \liLeereZelle:  $\emptyset$ 
Let-Abkürzung: \let\l=\liLeereZelle
1914 \def\liLeereZelle{$\emptyset$}

\liZustandsPaarVariablenName
1915 \def\liZustandsPaarVariablenName{z}

\liZustandsPaar
1916 \def\liZustandsPaar#1#2{
1917   $(
1918     \liZustandsPaarVariablenName_#1,
1919     \liZustandsPaarVariablenName_#2
1920   )$
1921 }

liUebergangsTabelle
1922 \renewcommand{\arraystretch}{1.4}
1923 \NewDocumentEnvironment{ liUebergangsTabelle } { m m } {
1924   \liPseudoUeberschrift{Übergangstabelle}
1925   \begin{center}
1926     \begin{tabular}{r|l|l}
1927       \textbf{Zustandspaar} & \textbf{#1} & \textbf{#2} \\ \hline
1928     \end{tabular}
1929   \end{center}
1930 }

liUeberschriftDreiecksTabelle \liUeberschriftDreiecksTabelle: Minimierungstabelle (Table filling)

1932 \ExplSyntaxOn
1933 \def\liUeberschriftDreiecksTabelle{
1934   \liPseudoUeberschrift{Minimierungstabelle~(Table~filling)}
1935 }

```

\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. —————

```

1936 \def\liMinimierungErklaerung{
1937   %\footcite[Seite~19]{koenig}
1938   \liParagraphMitLinien{
1939     Der~Minimierungs-Algorithmus~(auch~Table-Filling-Algorithmus-genannt)~
1940     trägt~in~seinem~Verlauf~eine~Markierung~in~alle~
1941     diejenigen~Zellen~der~Tabelle~ein,~die~zueinander~nicht~äquivalente~
1942     Zustände~bezeichnen.~Die~Markierung~„ $x_{\sb{n}}$ “~in~einer~Tabellenzelle~( $i$ ,~
1943      $j$ )~bedeutet~dabei,~dass~das~Zustandspaar~( $i$ ,~ $j$ )~in~der~ $k$ -ten~
1944     Iteration~des~Algorithmus~markiert~wurde~und~die~Zustände~ $i$ ~und~ $j$ ~
1945     somit~zueinander~( $k - 1$ )~äquivalent,~aber~nicht~ $k$ -äquivalent~und~
1946     somit~insbesondere~nicht~äquivalent~sind.~Bleibt~eine~Zelle~bis~zum~Ende~
1947     unmarkiert,~sind~die~entsprechenden~Zustände~zueinander~äquivalent.
1948   }
1949 }
1950 \ExplSyntaxOff
1951

```

2.28 normalformen.sty

```
1952 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1953 \ProvidesPackage{lehramt-informatik-normalformen}[2020/12/10
1954 Hilfsmakros zum Setzen von Normalformen, Funktionalen Abhängigkeiten,
1955 Attributhülle]
```

Mathe für textit tex etc, typographie für geschweifte Klammer

```
1956 \liLadePakete{mathe,typographie}
1957 \directlua{
1958   helper = require('lehramt-informatik-helfer')
1959   normalformen = require('lehramt-informatik-normalformen')
1960 }
```

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
```

```
1961 \def\liTeilen#1{
1962   \directlua{tex.print(normalformen.teilen('#1'))}
1963 }
```

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

```
1964 \def\liAttributHuelleOhneMathe#1{\text{AttrHülle}\{#1\}}
1965 \def\liAttributHuelle#1{
1966   \ifmmode
1967     \liAttributHuelleOhneMathe{#1}
1968   \else
1969     $\liAttributHuelleOhneMathe{#1}$
1970   \fi
1971 }
```

\liAttributMenge Let-Abkürzung: $\text{let}\ \text{m}=\text{liAttributMenge}$

```
1972 \def\liAttributMenge#1{\{ \textit{#1} \}}
```

liAHuelle

```
1973 \NewDocumentEnvironment{ liAHuelle } { +b } {
1974   \begin{group}
1975   \footnotesize
1976   \begin{multline*}
1977     #1
1978   \end{multline*}
1979   \end{group}
1980 } { }
```

\liLinksReduktion Nur innerhalb von liAHuelle zu verwenden bzw. multline

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

```
1981 \def\liLinksReduktion#1#2#3{
1982   \shoveleft{
1983     \liAttributHuelleOhneMathe{FA,
1984       \liAttributMenge{#1 \string\ #2}} =
1985     } \}
```

```

1986 \shoveright{
1987 \liAttributMenge{#3}
1988 } \\\
1989 }

```

```

\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( $F, \{A, E \setminus E\}$ ) =  $\{A, E, F, B, D\}$ 
1990 \def\liLinksReduktionInline#1#2#3{%
1991 {%
1992 \footnotesize%
1993 $\liAttributHuelleOhneMathe{F,
1994 \liAttributMenge{#1 \string\ #2}} =
1995 \liAttributMenge{#3}$
1996 }
1997 }

```

```

\liLinksReduktionInline Let-Abkürzung: \let\ahr=\liLinksReduktionInline
\ahr{gelöschte FA}{neue FA ohne rechts Attribut}{gegebene Attribute}{Ergebnis}
1998 \def\liRechtsReduktionInline#1#2#3#4{%
1999 {%
2000 \footnotesize%
2001 $\liAttributHuelleOhneMathe{
2002 F \setminus
2003 \liFunktionaleAbhaengigkeit{#1}
2004 \def\tmp{#2}\ifx\tmp\empty
2005 \else
2006 \cup \liFunktionaleAbhaengigkeit{#2}
2007 \fi
2008 ,
2009 \liAttributMenge{#3}
2010 } =
2011 \liAttributMenge{#4}$
2012 }
2013 }

```

```

\liFunktionaleAbhaengigkeit Let-Abkürzung: \let\fa=\liFunktionaleAbhaengigkeit
\fa{A, B -> C, D}:  $\{A, B\} \rightarrow \{C, D\}$  Regulärer Ausdruck zum Konvertieren:
\$(.*?) \\\rightarrow (.*)\$ \fa{$1 -> $2}
2014 \def\liFunktionaleAbhaengigkeit#1{%
2015 \directlua{normalformen.drucke_funk_abhaengigkeit('#1', false)}%
2016 }

```

```

\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;

```

```

2017 \NewDocumentCommand {\liFunktionaleAbhaengigkeiten} { O{FA} m } {
2018   \liGeschweifteKlammern
2019   {#1}
2020   {
2021     \begin{align*}
2022       \directlua{normalformen.drucke_funk_abhaengigkeiten('#2')}
2023     \end{align*}
2024   }
2025   {-0.5cm}
2026   {-1.7cm}
2027 }

\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}

2028 \NewDocumentCommand {\liRelation} { O{R} m } {
2029   $\directlua{
2030     local name = helper.konvertiere_tiefgestellt('#1')
2031     tex.print(name)
2032   }$(\textit{\, #2\,})
2033 }

2034

```

2.29 petri.sty

2035 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2036 \ProvidesPackage{lehramt-informatik-petri}[2020/12/03]

Let-Abkürzungen

\let\t=\liPetriTransitionsName

\let\tp=\liPetriTransPfeile

\let\k=\liPetriErreichKnotenDrei

2037 \RequirePackage{tikz}

2038 \usetikzlibrary{petri,arrows.meta}

Für die Darstellungsmatrix

2039 \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

```
2040 \def\liPetriSetzeSchluessel{%
2041   \def\TmpTransitionOne{}%
2042   \def\TmpTransitionTwo{}%
2043   \def\TmpTransitionThree{}%
2044   \def\TmpTransitionFour{}%
2045   \def\TmpTransitionFive{}%
2046   \def\TmpTransitionSix{}%
2047   \def\TmpTransitionSeven{}%
2048   \def\TmpTransitionEight{}%
2049   \def\TmpTransitionNine{}%
2050   \def\TmpTransitionTen{}%
2051   \pgfkeys{/petri/.cd,
2052     p1/.store in=\TmpPlaceOne,p1/.default=0,p1,
2053     p2/.store in=\TmpPlaceTwo,p2/.default=0,p2,
2054     p3/.store in=\TmpPlaceThree,p3/.default=0,p3,
2055     p4/.store in=\TmpPlaceFour,p4/.default=0,p4,
2056     p5/.store in=\TmpPlaceFive,p5/.default=0,p5,
```

```

2057     p6/.store in=\TmpPlaceSix,p6/.default=0,p6,
2058     p7/.store in=\TmpPlaceSeven,p7/.default=0,p7,
2059     p8/.store in=\TmpPlaceEight,p8/.default=0,p8,
2060     p9/.store in=\TmpPlaceNine,p9/.default=0,p9,
2061     p10/.store in=\TmpPlaceTen,p10/.default=0,p10,
2062     t1/.store in=\TmpTransitionOne,t1/.default=activated,
2063     t2/.store in=\TmpTransitionTwo,t2/.default=activated,
2064     t3/.store in=\TmpTransitionThree,t3/.default=activated,
2065     t4/.store in=\TmpTransitionFour,t4/.default=activated,
2066     t5/.store in=\TmpTransitionFive,t5/.default=activated,
2067     t6/.store in=\TmpTransitionSix,t6/.default=activated,
2068     t7/.store in=\TmpTransitionSeven,t7/.default=activated,
2069     t8/.store in=\TmpTransitionEight,t8/.default=activated,
2070     t9/.store in=\TmpTransitionNine,t9/.default=activated,
2071     t10/.store in=\TmpTransitionTen,t10/.default=activated,
2072     scale/.store in=\TmpScale,scale/.default=0.5,
2073     x/.store in=\TmpX,x/.default=5,
2074     y/.store in=\TmpY,y/.default=5,
2075   }%
2076 }

2077 \tikzset{
2078   li petri/.style={
2079     activated/.style={
2080       very thick
2081     },
2082     inhibitor/.style={
2083       {Circle[open,length=2mm,fill=white]}-
2084     }
2085   }
2086 }

```

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

```

2087 \def\liPetriTransitionsNameOhneMathe#1{t\sb{#1}}
2088 \def\liPetriTransitionsName#1{
2089   \ifmmode
2090     \liPetriTransitionsNameOhneMathe{#1}
2091   \else
2092     $\liPetriTransitionsNameOhneMathe{#1}$
2093   \fi
2094 }

```

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

```

2095 \NewDocumentCommand{ \liPetriErreichTransition } { m m m O{} O{} } {
2096   \draw[->] (#1) edge[#4] node[pos=0.5,auto,sloped,#5]{t\sb{#3}$} (#2);
2097 }

```

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

```

2098 \def\liPetriErreichKnotenDrei#1#2#3{(#1,#2,#3)}

```

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

```

2099 \def\liPetriTransPfeile#1{ $\rightarrow \hspace{0.4cm} \liPetriTransitionsName{#1} \hspace{0.3cm}
2100 }

```


2.30 potenzmengen-konstruktion.sty

```
2101 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2102 \ProvidesPackage{lehramt-informatik-potenzmengen-konstruktion}[2021/02/21 Hilfsmakros
2103 einzusetzen bei der Potenzmengen-Konstruktion (Potenzmengen-Algorithmus)]
```

```
2104 \liLadePakete{formale-sprachen}
2105 \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}
    }
  }
}
```

```
2106 \def\liZustandsMengenSammlung#1#2{
2107   \liZustandsnameGross{#1}
2108   {
2109     \footnotesize
2110     \liPotenzmenge{
2111       \str_case:nn {#1} {#2
2112       }
2113     }
2114 }
```

```
\liZustandsMengenSammlungNr
```

```
2115 \def\liZustandsMengenSammlungNr#1#2{
2116   \liZustandsnameGross{#1}
2117   {
```

```
2118     \footnotesize
2119     \liZustandsmengeNr{
2120         \str_case:nn {#1} #2
2121     }
2122 }
2123 }

2124 \ExplSyntaxOff
2125
```

2.31 pseudo.sty

```

2126 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2127 \ProvidesPackage{lehramt-informatik-pseudo}[2020/12/30 Paket zum Setzen
2128 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>

```

2129 \RequirePackage[german,boxruled]{algorithm2e}

```

```

2130

```

2.32 pumping-lemma.sty

2131 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2132 \ProvidesPackage{lehramt-informatik-pumping-lemma}[2021/08/11 Enthält die
 2133 Definitionen für das Pumping-Lemma in der Regulären Sprache und
 2134 in der Kontextfreien Sprache]

\liPumpingRegulaer

```
2135 \def\liPumpingRegulaer{%
2136   Es sei  $L$  eine reguläre Sprache. Dann gibt es eine Zahl  $j$ , sodass für
2137   alle Wörter  $\omega$  in  $L$  mit  $|\omega| \geq j$  (jedes Wort  $\omega$  in
2138    $L$  mit Mindestlänge  $j$ ) jeweils eine Zerlegung  $\omega = uvw$ 
2139   existiert, sodass die folgenden Eigenschaften erfüllt sind:
2140
2141   \begin{enumerate}
2142     \item  $|v| \geq 1$ 
2143     (Das Wort  $v$  ist nicht leer.)
2144
2145     \item  $|uv| \leq j$ 
2146     (Die beiden Wörter  $u$  und  $v$  haben zusammen höchstens die Länge  $j$ .)
2147
2148     \item Für alle  $i = 0, 1, 2, \dots$  gilt  $uv^i w$  in  $L$ 
2149     (Für jede natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^i w$  in der
2150     Sprache  $L$ )
2151   \end{enumerate}
2152
2153   Die kleinste Zahl  $j$ , die diese Eigenschaften erfüllt, wird
2154   Pumping-Zahl der Sprache  $L$  genannt.\footcite{wiki:pumping-lemma}
2155 }
```

\liPumpingKontextfrei

```
2156 \def\liPumpingKontextfrei{%
2157   Es sei  $L$  eine kontextfreie Sprache. Dann gibt es eine Zahl  $j$ , sodass
2158   sich alle Wörter  $\omega$  in  $L$  mit  $|\omega| \geq j$  zerlegen lassen in
2159    $\omega = uvwxy$ , sodass die folgenden Eigenschaften erfüllt sind:
2160
2161   \begin{enumerate}
2162     \item  $|vx| \geq 1$ 
2163     (Die Wörter  $v$  und  $x$  sind nicht leer.)
2164
2165     \item  $|vwx| \leq j$ 
2166     (Die Wörter  $v$ ,  $w$  und  $x$  haben zusammen höchstens die Länge  $j$ .)
2167
2168     \item Für alle  $i$  in  $\mathbb{N}_0$  gilt  $uv^iwx^iy$  in  $L$  (Für jede
2169     natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^iwx^iy$  in der
2170     Sprache  $L$ )
2171   \end{enumerate}
2172 }
```

2173

2.33 quicksort.sty

```

2174 % https://tex.stackexchange.com/a/142634
2175 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2176 \ProvidesPackage{lehramt-informatik-quicksort}[2020/06/12]
2177
2178 %-----
2179 % USAGE:
2180 % \QSinitialize{comma, separated, numerical, values}
2181 % \loop
2182 % \QSpivotStep
2183 % \ifnum\value{pivotcount}>0
2184 %   \QSSortStep
2185 % \repeat
2186 %-----
2187
2188 % xintfrac does not load xinttools, this must be done explicitly if needed as here.
2189 \RequirePackage{xintfrac, xinttools}
2190
2191 \RequirePackage{tikz}
2192
2193 %-----
2194 % FIRST PART: TikZ styles and macros for the actual drawing
2195 \newcounter{cellcount}% used for coordinates of the node
2196 \newcounter{pivotcount}% when it will remain at zero, will signal the sort is finished.
2197
2198 % Styles defined by Tom Bombaldi. (modified: all share the same size)
2199 % (re-modified \bf -> \bfseries due to extremely annoying warnings from
2200 % KOMA-script which are truly a pain and do not make any sense regarding \bf:
2201 % if I want to use \bf, and know what I am doing, why should I get HARASSED
2202 % by police of LaTeX good conduct ? )
2203 \tikzset{!/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=lime!70!gray},
2204          o/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=olive!50},
2205          r/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2206 % this is the "b" style as used in the image below
2207 %   b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2208 % nicer:
2209 %   b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=white, text=magenta},
2210 %   g/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=gray, text=white, text=white},
2211
2212 % NOTE the b style was originally the same as the r(aised) style apart from
2213 % not being raised, but I find it nicer with a somewhat different
2214 % specification. I have not updated the images though.
2215
2216 % How the nodes are drawn depending on whether on the left of the pivot value
2217 % or on the right, or is a pivot value, or a raised pivot during selection phase.
2218
2219 \def\DecoLEFT #1{%
2220   \xintFor* ##1 in {#1} \do
2221     {\stepcounter{cellcount}\node[o] at (\arabic{cellcount},0) {##1};}%
2222 }
2223
2224 \def\DecoINERT #1{%
2225   \xintFor* ##1 in {#1} \do
2226     {\stepcounter{cellcount}\node[g] at (\arabic{cellcount},0) {##1};}%
2227 }
2228
2229 \def\DecoRIGHT #1{%
2230   \xintFor* ##1 in {#1} \do
2231     {\stepcounter{cellcount}\node[l] at (\arabic{cellcount},0) {##1};}%
2232 }
2233
2234 \def\DecoLEFTwithPivot #1{\stepcounter{pivotcount}%
2235   \xintFor* ##1 in {#1} \do

```

```

2236     {\stepcounter{cellcount}}%
2237     \xintifForLast {\node[r]}\node[o]} at (\arabic{cellcount},0) {##1};}%
2238 }
2239
2240 \def\DecoINERTwithPivot #1{\stepcounter{pivotcount}%
2241     \xintFor* ##1 in {#1} \do
2242     {\stepcounter{cellcount}}%
2243     \xintifForLast {\node[b]}\node[g]} at (\arabic{cellcount},0) {##1};}%
2244 }
2245
2246 \def\DecoRIGHTwithPivot #1{\stepcounter{pivotcount}%
2247     \xintFor* ##1 in {#1} \do
2248     {\stepcounter{cellcount}}%
2249     \xintifForLast {\node[r]}\node[l]} at (\arabic{cellcount},0) {##1};}%
2250 }
2251
2252 %-----
2253 % SECOND PART: the actual sorting routines.
2254
2255 \def\QS@sort@a #1{\expandafter \QS@sort@b \expandafter {\xintLength {#1}}{#1}}
2256 \def\QS@sort@b #1{\ifcase #1
2257     \expandafter\QS@sort@empty
2258     \or\expandafter\QS@sort@single
2259     \else\expandafter\QS@sort@c
2260     \fi
2261 }%
2262 \def\QS@sort@empty #1{}
2263 \def\QS@sort@single #1{\QSIr {#1}}
2264
2265 % This step is to pick the last as pivot.
2266 \def\QS@sort@c #1%
2267     {\expandafter\QS@sort@d\expandafter {\romannumeral0\xintnthelt {-1}{#1}}{#1}}%
2268
2269 % Here \QSLr, \QSIr, \QSR have been let to \relax.
2270 % The trick with \xintApplyUnbraced is that for example when selecting
2271 % the elements smaller than pivot, if we had been using \xintApply we
2272 % would have had at the minimum an empty brace pair. Thus we use the
2273 % "unbraced" variant, but then the \QS@select@smaller has added in
2274 % anticipation a level of braces.
2275 \def\QS@sort@d #1#2{%
2276     \QSLr {\xintApplyUnbraced {\QS@select@smaller {#1}}{#2}}}%
2277     \QSIr {\xintApplyUnbraced {\QS@select@equal {#1}}{#2}}}%
2278     \QSRr {\xintApplyUnbraced {\QS@select@greater {#1}}{#2}}}%
2279 }%
2280 \def\QS@select@smaller #1#2{\xintifLt {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2281 \def\QS@select@equal #1#2{\xintifEq {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2282 \def\QS@select@greater #1#2{\xintifGt {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2283
2284 %
2285 % NOTE 1: thus, each comparison with the pivot is done three (!) times.
2286 %
2287 % NOTE 2: we may well end up with \QSLr {<empty>} situations. This is handled
2288 % silently by the \xintFor loops, and also when \QSLr becomes \QS@sort@a, the
2289 % latter must handle correctly an empty argument.
2290
2291 %-----
2292 % THIRD PART: the main macros \QSpivotStep, \QSsortStep and \QSinitialize.
2293
2294 % This draws all with suitable highlighting for the newly chosen pivots
2295 % (which will be shown raised)

```

```

2296 \def\QSpivotStep {\let\QSLr\DecoLEFTwithPivot
2297             \let\QSIr\DecoINERT
2298             \let\QSIrr\DecoINERT
2299             \let\QSRr\DecoRIGHTwithPivot
2300 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2301             \setcounter{cellcount}{0}\setcounter{pivotcount}{0}%
2302             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2303 }
2304
2305 % This sorts and then draws, showing where the pivot chosen in the previous
2306 % step go. Next time they will have become "inert". If pivotcount is still at
2307 % zero on exit from \QSpivotStep, then this is the signal to stop before
2308 % executing \QSsortStep.
2309 \def\QSsortStep {\def\QSLr {\noexpand\QS@sort@a}%
2310             \def\QSRr {\noexpand\QS@sort@a}%
2311             \def\QSIr {\noexpand\QSIrr}%
2312             \let\QSIrr\relax
2313             \edef\QS@list{\QS@list}%
2314             \let\QSLr\relax
2315             \let\QSRr\relax
2316             \let\QSIr\relax
2317             \edef\QS@list{\QS@list}%
2318             \let\QSLr\DecoLEFT
2319             \let\QSIr\DecoINERTwithPivot
2320             \let\QSIrr\DecoINERT
2321             \let\QSRr\DecoRIGHT
2322 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2323             \setcounter{cellcount}{0}%
2324             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2325 }
2326
2327 \def\QSinitialize #1{%
2328     % first, we convert the comma separated values into a list of braced items
2329     % we use an \edef, and anyhow many \edef's will be used later
2330     \edef\QS@list {\noexpand\QSRr {\xintCSVtoList {#1}}}%
2331     \let\QSRr\DecoRIGHT
2332     % The \QSRr marker mutated to draw the last element as
2333     % pivot and the earlier ones with the suitable style.
2334     %
2335     % The list of marked braced items \QS@list is used both for drawing
2336     % (as here) and for doing the exchange of elements during sort.
2337     \par\centerline{\rule[1.5mm]{0pt}{8mm}\setcounter{cellcount}{0}%
2338             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2339 }
2340

```

2.34 relationale-algebra.sty

```

2341 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2342 \ProvidesPackage{lehramt-informatik-relationale-algebra}[2020/12/11]
2343 \RequirePackage{amsmath}
2344 \RequirePackage{amssymb}

```

```

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

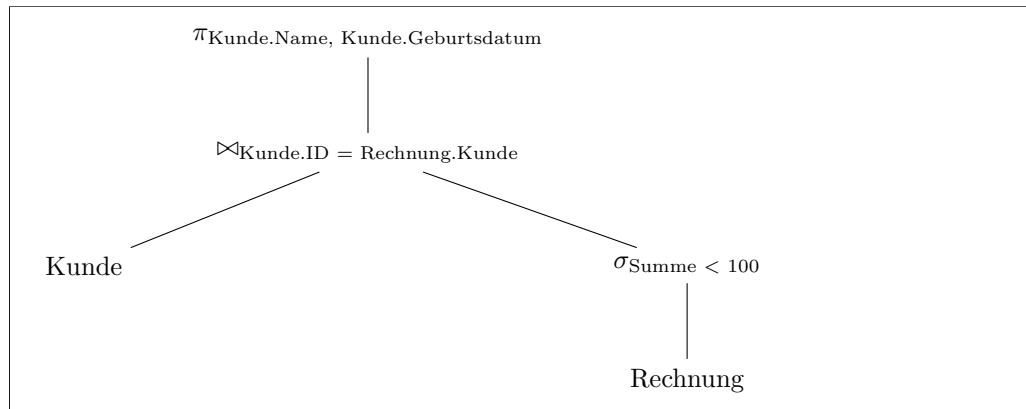
  \node[below=of pi]
    (theta join) {\bowtie_{\text{Kunde.ID = Rechnung.Kunde}}};
    edge (pi);

  \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}

```



```

2345 \RequirePackage{tikz}
2346 \usetikzlibrary{positioning}

    Privates Makros, das zwei Querstriche erzeugt.
2347 \def\o@join{\setbox0=\hbox{\bowtie}}%
2348   \rule[-.02ex]{.25em}{.4pt}\llap{\rule[\ht0]{.25em}{.4pt}}%
2349 }

```

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

```
2350 \def\leftouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie}}
```

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

```
2351 \def\rightouterjoin{\mathbin{\bowtie\mkern-5.8mu\o@join}}
```

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

```
2352 \def\fullouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie\mkern-5.8mu\o@join}}
```

```
2353
```


2.35 rmodell.sty

```

2354 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2355 \ProvidesPackage{lehramt-informatik-rmodell}[2020/09/01
2356 Makros und Umgebungen zum Setzen des Relationenmodells beim Thema
2357 Datenbanken.]
2358 \RequirePackage{soul}

```

Let-Abkürzungen

```

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

```

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

```

2359 \def\liPrimaer#1{\ul{#1}}

```

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

```

2360 \def\liFremd#1{{\setul{-0.9em}{}\ul{#1}}}

```

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

```

2361 \def\li@Rmodell@Schrift{\footnotesize\ttfamily}
2362 \ExplSyntaxOn
2363 \NewDocumentEnvironment { liRmodell }
2364 { +b }
2365 {
2366   \medskip
2367   {
2368     \linespread{2}
2369     \setlength{\parindent}{0pt}
2370     \li@Rmodell@Schrift#1
2371   }
2372   \medskip
2373 } {}
2374 \ExplSyntaxOff

```

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

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

```

2375 \def\liRelationMenge#1#2{
2376 \noindent
2377 #1 : \[ #2 ]\}
2378 \par
2379 }

```

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

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

```

2380 \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}

```

```

2381 \NewDocumentEnvironment { liRelationenSchemaFormat }{ +b } {} {}

```

```

2382

```

2.36 sortieren.sty

```
2383 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2384 \ProvidesPackage{lehramt-informatik-sortieren}[2020/06/10 Schaubilder
2385 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}
}
```



```
2386 \RequirePackage{tikz}
2387 \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.

```
2388 \def\liVertauschen#1{
2389   \directlua{
2390     local sortieren = require('lehramt-informatik-sortieren')
2391     sortieren('#1')
2392   }
2393 }
```

`\liSortierPfeil`

```
2394 \def\liSortierPfeil#1#2{
2395   \draw[-latex] ([xshift=1mm]reihe.#1 north) -- ++(0,0.25) -| ([xshift=-1mm]reihe.#2 north);
2396 }
```

`\liSortierPfeilUnten`

```
2397 \def\liSortierPfeilUnten#1#2{
2398   \draw[-latex] ([xshift=1mm]reihe.#1 south) -- ++(0,-0.25) -| ([xshift=-1mm]reihe.#2 south);
2399 }
```

`\liSortierMarkierung`

```
2400 \def\liSortierMarkierung#1#2{\node[
2401   draw,
2402   very thick,
2403   fit=(reihe.#1) (reihe.#2),
2404   inner sep=0pt
2405 ] {}
2406 }

2407 \tikzset{
2408   li sortierung zahlenreihe/.style={
2409     draw,
2410     thin,
2411     font=\large,
2412     rectangle split horizontal,
2413     rectangle split,
2414   }
2415 }
```

```

2416 % https://tex.stackexchange.com/a/140895
2417 \RequirePackage{forest,xstring}
2418 \usetikzlibrary{calc}
2419
2420 \makeatletter
2421 \pgfmathdeclarefunction{strrepeat}{2}{%
2422   \begingroup\pgfmathint{#2}\pgfmath@count\pgfmathresult
2423   \let\pgfmathresult\pgfutil@empty
2424   \pgfutil@loop\ifnum\pgfmath@count>0\relax
2425     \expandafter\def\expandafter\pgfmathresult\expandafter{\pgfmathresult#1}%
2426     \advance\pgfmath@count-1\relax
2427   \pgfutil@repeat\pgfmath@smuggleone\pgfmathresult\endgroup}
2428 \makeatother
2429
2430 \def\myNodes{}
2431
2432 \ExplSyntaxOn
2433 \newcommand*\sortList[1]{%
2434   \clist_sort:Nn#1{\int_compare:nNnTF{##1}>{##2}\sort_return_swapped:\sort_return_same:}}
2435 \ExplSyntaxOff
2436
2437 \forestset{
2438   sort/.code={%
2439     \pgfmathparse{level()}>\forestSortLevel}%
2440     \ifnum\pgfmathresult=0
2441       \StrSubstitute{\forestov{content}}{ }{,}{\myList}%
2442       \sortList\myList
2443       \StrSubstitute{\myList}{,}{ }{\myList}%
2444       \pgfmathparse{strrepeat("1",level())}%
2445       \xappto\myNodes{\noexpand\node at ({\forestov{name}}|-m)!-1!({\forestov{name}}$)
2446         (m\forestov{name}) {\myList}}%
2447       \pgfmathparse{level()}==\forestSortLevel}%
2448       \ifnum\pgfmathresult=1
2449         \forestOget{\forestov{@first}}{name}\forestFirst
2450         \forestOget{\forestov{@last}}{name}\forestLast
2451         \xappto\myNodes{{[<-]edge (\forestOv{\forestov{@first}}{name})
2452           \ifx\forestFirst\forestLast\else edge (\forestOv{\forestov{@last}}{name})\fi}}%
2453       \fi
2454       \ifnum\forestov{@parent}=0\else
2455         \xappto\myNodes{edge (m\forestOv{\forestov{@parent}}{name})}%
2456       \fi
2457       \gappto\myNodes{;}%
2458     \fi}}
2459
2460 \forestset{sort level/.code=%
2461   \pgfmathparse{#1}\let\forestSortLevel\pgfmathresult
2462   \pgfmathparse{strrepeat("1",\forestSortLevel+1)}\let\forestOnes\pgfmathresult}
2463

```

2.37 spalten.sty

```
2464 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2465 \ProvidesPackage{lehramt-informatik-spalten}[2020/12/07 Lädt das Paket
2466 „multicol“, damit mehrspaltiger Satz mit Hilfe der Umgebung „multicols“
2467 realisiert werden kann.]
2468 \RequirePackage{multicol}
```

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

```
2469 \def\liSpaltenUmbruch{\vfill\strut\columnbreak}
```

```
2470
```

2.38 struktogramm.sty

```
2471 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2472 \ProvidesPackage{lehramt-informatik-struktogramm}[2021/01/31 Lädt das
2473 Paket struktex zum Setzen von Struktogrammen]
2474 \RequirePackage{struktex}
2475
```

2.39 syntax.sty

```
2476 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2477 \ProvidesPackage{lehramt-informatik-syntax}[2020/06/10 Ein Hüll-Paket um
2478 `minted`, das sich um die Syntax-Hervorhebung des Java-Codes kümmert.]
2479 \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
```

```
2480 \ExplSyntaxOn
2481 \directlua{
2482   syntax = require('lehramt-informatik-syntax')
2483   syntax.importiere_konfiguration('tex_repo_lokaler_pfad', '\LehramtInformatikRepository')
2484   syntax.importiere_konfiguration('github_domain', '\LehramtInformatikGithubDomain')
2485   syntax.importiere_konfiguration('github_raw_domain', '\LehramtInformatikGithubRawDomain')
2486   syntax.importiere_konfiguration('github_tex_repo', '\LehramtInformatikGithubTexRepo')
2487   syntax.importiere_konfiguration('github_code_repo', '\LehramtInformatikGithubCodeRepo')
2488   syntax.importiere_konfiguration('git_branch', '\LehramtInformatikGitBranch')
2489 }
2490 \RequirePackage{hyperref}
2491 \RequirePackage{minted}
2492 % pygmentize -L styles
2493 \usemintedstyle{colorful}
2494 %\BeforeBeginEnvironment{minted}{\begin{mdframed}}
2495 %\AfterEndEnvironment{minted}{\end{mdframed}}
2496 %\setminted{breaklines=true,linenos}
2497 \setminted{
2498   breaklines=true,
2499   linenos,
2500   fontsize=\footnotesize,
2501 }
```

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

Let-Abkürzung: `\let\j=\liJavaCode`

```
2502 \def\liJavaCode#1{\,\mintinline{java}|#1|\,}
```

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

```
2503 \def\liLatexCode#1{\mintinline{latex}|#1|}
```

```
2504 \def\li@GithubLink#1#2{
2505   \begin{flushright}
2506     \tiny
2507     Code-Beispiel~auf~Github~ansehen:~
2508     \href{#1}{\nolinkurl{#2}}
2509   \end{flushright}
2510 }
```

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

```
2511 \NewDocumentCommand{\liJavaDatei}{ O{firstline=3} m }{
2512   \inputminted[#1]{java}{
2513     \directlua{
2514       syntax.drucke_absoluten_pfad('#2', false)
2515     }
2516   }
2517   \li@GithubLink
```

```

2518     {\directlua{syntax.drucke_github_url('#2', false)}}
2519     {\directlua{syntax.drucke_relativen_repo_pfad('#2', false)}}
2520 }

\liJavaTestDatei Eine komplette Java-Test-Datei einbinden, die Verzeichnis ./Code/src/test/java/org/bschlangaul
liegt.
2521 \NewDocumentCommand{\liJavaTestDatei}{ 0{firstline=3} m }{
2522   \inputminted[#1]{java}{
2523     \directlua{
2524       syntax.drucke_absoluten_pfad('#2', true)
2525     }
2526   }
2527   \li@GithubLink
2528   {\directlua{syntax.drucke_github_url('#2', true)}}
2529   {\directlua{syntax.drucke_relativen_repo_pfad('#2', true)}}
2530 }

\liJavaExamen \liJavaExamen{66116}{2015}{03}{Kunde} Regulärer Ausdruck zum Konvertieren: \liJavaDatei([.
\liJavaExamen$1{$2}{$3}{$4}{$5}
2531 \NewDocumentCommand{\liJavaExamen}{ 0{firstline=3} m m m m }{
2532   \inputminted[#1]{java}{
2533     \directlua{
2534       syntax.drucke_absoluten_examens_pfad('#2', '#3', '#4', '#5')
2535     }
2536   }
2537
2538   \li@GithubLink
2539   {\directlua{syntax.drucke_github_examens_url('#2', '#3', '#4', '#5')}}
2540   {\directlua{syntax.drucke_relativen_examens_repo_pfad('#2', '#3', '#4', '#5')}}
2541 }

\liAssemblerCode
2542 \def\liAssemblerCode#1{\mintinline{asm}|#1|}

\liAssemblerDatei \liAssemblerDatei{relativer-pfad}: Relativer Pfad zu einer Assembler-Datei.
2543 \NewDocumentCommand{\liAssemblerDatei}{ m }{
2544   \inputminted{asm}{#1}
2545 }

\liMinispracheDatei \liMinispracheDatei{relativer-pfad}: Relativer Pfad zu einer Minisprachen-Datei
(Hochsprache für die Minimaschine von Albert Wiedemann).
2546 \NewDocumentCommand{\liMinispracheDatei}{ m }{
2547   \inputminted{componentpascal}{#1}
2548 }

\liHaskellCode \liHaskellCode{haskell}: Zum Setzen von Haskell-Code.
2549 \def\liHaskellCode#1{\mintinline{haskell}|#1|}

\liHaskellDatei \liHaskellDatei{relativer-pfad}: Relativer Pfad zu einer Haskell-Datei.
2550 \NewDocumentCommand{\liHaskellDatei}{ m }{
2551   \inputminted{haskell}{#1}
2552 }

2553 \ExplSyntaxOff

\liSqlCode \liHaskellCode{sql}: Zum Setzen von SQL-Code.
Let-Abkürzung: \let\s=\liSqlCode
2554 \def\liSqlCode#1{\mintinline{sql}|#1|}

2555

```

2.40 syntaxbaum.sty

```
2556 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2557 \ProvidesPackage{lehramt-informatik-syntaxbaum}[2021/02/14 Zum Setzen von
2558 Syntaxbäumen mit Hilfe des Pakets tikz-qtrees]
2559 \RequirePackage{tikz-qtrees}
2560
2561 \tikzset{li parsetree/.style={
2562     every internal node/.style={
2563         draw,circle
2564     },
2565     every leaf node/.style={
2566         draw,rectangle
2567     },
2568 }
2569 }
2570
```


2.41 synthese-algorithmus.sty

```
2571 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2572 \ProvidesPackage{lehramt-informatik-synthese-algorithmus}[2021/03/19
2573 Hilfsmakros zum Setzen des Synthese-Algorithmuses zur Umformung einer
2574 Relation in die 3. Normalform]

2575 \liLadePakete{normalformen,mathe,typographie}
2576 \ExplSyntaxOn
```

Let-Abkürzungen

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

2.41.1 TeX-Markup Grundgerüst

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

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

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

2.41.2 TeX-Markup Linksreduktion

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

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

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

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

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

2.41.3 TeX-Markup Rechtreduktion

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

\liPseudoUeberschrift{F}

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

\liPseudoUeberschrift{A}

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

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 Anhängigkeit $\alpha \rightarrow \beta \in F$ 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 $R_\alpha := \alpha \cup \beta$. —

3. Schlüssel hinzufügen

— Falls eines der in Schritt 2. erzeugten Schemata 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

```
2577 \def\liSyntheseUeberschrift#1{
2578   {
2579     \bfseries
2580     \sffamily
2581     \str_case:nn {#1} {
2582       {1} {Kanonische-Überdeckung}
2583       {1-1} {Linksreduktion}
2584       {1-2} {Rechtsreduktion}
2585       {1-3} {Löschen-leerer-Klauseln}
2586       {1-4} {Vereinigung}
2587       {2} {Relationsschemata~formen}
2588       {3} {Schlüssel-hinzufügen}
2589       {4} {Entfernung-überflüssiger-Teilschemata}
2590     }
2591   }
2592 }
```

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

```
2593 \def\li@synthese@erklaerung@texte#1{
2594   \str_case:nn {#1} {
2595     {1} {
2596       Die-kanonische-Überdeckung---also-die-kleinst-mögliche-noch-
2597       äquivalente-Menge-von-funktionalen-Abhängigkeiten-kann-in-vier-
2598       Schritten-erreicht-werden.

```

```

2599 }
2600 {1-1} {
2601   Führe-für-jede-funktionale-Anhängigkeit~
2602    $\alpha \rightarrow \beta$ -in-F$-die-Linksreduktion-durch,~
2603   überprüfe-also-für-alle~
2604    $A$ -in- $\alpha$ ~,~ob- $A$ -überflüssig-ist,~d.h.~ob~
2605    $\beta \subseteqq \text{AttributHuelle}(F, \alpha \cup A)$ ~.~$
2606 }
2607 {1-2} {
2608   Führe-für-jede-(verbliebene)-funktionale-Abhängigkeit- $\alpha \rightarrow \beta$ -die-Rechtsreduktion-durch,~überprüfe-also-für-
2609   alle- $B$ -in- $\beta$ ~,~ob- $B$ -in- $\text{AttributHuelle}(F, \alpha \rightarrow \beta) \cup (\alpha \rightarrow \beta)$ ~,~
2610    $\alpha$ ~-gilt.~In-diesem-Fall-ist- $B$ -auf-der-rechten-Seite~
2611   überflüssig-und-kann-eliminiert-werden,~dh- $\alpha \rightarrow \beta$ -wird-durch- $\alpha \rightarrow (\beta \cup B)$ ~
2612   ersetzt.
2613 }
2614 {1-3} {
2615   Entferne-die-funktionalen-Abhängigkeiten-der-Form- $\alpha \rightarrow \emptyset$ ~,~die-im-2.-Schritt-möglicherweise~
2616   entstanden-sind.
2617 }
2618 {1-4} {
2619   Fasse-mittels-der-Vereinigungsregel-funktionale-Abhängigkeiten-
2620   der-Form- $\alpha \rightarrow \beta_{\{1\}}$ ~,~dots,~ $\alpha \rightarrow \beta_{\{n\}}$ ~,~so-dass- $\alpha \rightarrow \beta_{\{1\}} \cup \dots \cup \beta_{\{n\}}$ ~-verbleibt.
2621 }
2622 % Kemper Seite 197
2623 {2} {
2624   Erzeuge-für-jede-funktionale-Abhängigkeit- $\alpha \rightarrow \beta$ -in-F $\{c\}$ ~-ein-Relationenschema- $\mathcal{R}_{\alpha \rightarrow \beta}$ ~
2625   := $\alpha \cup \beta$ ~.
2626 }
2627 {3} {
2628   Falls-eines-der-in-Schritt-2.-erzeugten-Schemata- $\mathcal{R}_{\alpha \rightarrow \beta}$ ~-einen-Schlüsselkandidaten-von- $\mathcal{R}$ ~-bezüglich-F $\{c\}$ ~-
2629   enthält,~sind-wir-fertig,~sonst-wähle-einen-Schlüsselkandidaten-
2630    $K \subseteqq \mathcal{R}$ ~-aus-und-definiere-folgendes-
2631   zusätzliche-Schema:~ $\mathcal{R}_{\alpha \rightarrow \beta} \setminus K := \mathcal{R}_{\alpha \rightarrow \beta} \setminus K$ ~
2632   und- $\mathcal{F} \setminus K := \mathcal{F} \setminus K$ ~.
2633 }
2634 {4} {
2635   Eliminiere-diejenigen-Schemata- $\mathcal{R}_{\alpha \rightarrow \beta}$ ~,~die-in-einem-
2636   anderen-Relationenschema- $\mathcal{R}_{\alpha' \rightarrow \beta'}$ ~-enthalten-sind,~d.h.~
2637    $\mathcal{R}_{\alpha \rightarrow \beta} \subseteqq \mathcal{R}_{\alpha' \rightarrow \beta'}$ ~.
2638 }
2639 }
2640 }
2641 \def\liSyntheseErklaerung#1{
2642 {
2643   \itshape
2644   \footnotesize
2645   \liParagraphMitLinien{\li@synthese@erklaerung@texte{#1}}
2646 }
2647 }
2648 }
2649 \liSyntheseUeberErklaerung
2650 Let-Abkürzung: \let\schrittE=\liSyntheseUeberErklaerung
2651 \def\liSyntheseUeberErklaerung#1{
2652   \liSyntheseUeberschrift{#1}\par
2653   \liSyntheseErklaerung{#1}
2654 }

```

2660 \ExplSyntaxOff
2661

2.42 tabelle.sty

2662 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2663 \ProvidesPackage{lehramt-informatik-tabelle}[2020/12/05 Lädt das Paket tabluarx]

2664 \RequirePackage{tabularx}

2665

2.43 typographie.sty

```
2666 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2667 \ProvidesPackage{lehramt-informatik-typographie}[2021/03/20 Typographische Makros,
2668 die das Erscheinungsbild verändern. Die Schriftdefinition sind in
2669 formatierung.sty definiert.]
```

```
2670 \ExplSyntaxOn
```

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

```
2671 \RequirePackage{fontawesome}
```

```
\liErledigt \liErledigt: ☑
```

```
2672 \let\liErledigt=\faCheckSquareO
```

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

```
2673 \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 —

```
2674 \def\liParagraphMitLinien#1{
```

```
2675 \noindent
```

```
2676 \vrule height 2pt depth -1.6pt width 0.4cm
```

```
2677 \enspace
```

```
2678 #1
```

```
2679 \enspace
```

```
2680 \leaders\vrule height 2pt depth -1.6pt \hfill \null
```

```
2681 \par
```

```
2682 \medskip
```

```
2683 }
```

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

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

```
2684 \def\liGeschweifteKlammern#1#2#3#4{
```

```
2685 \par
```

```
2686 \medskip
```

```
2687 \noindent
```

```
2688 #1 \, $= \Bigl\{ \$
```

```
2689 \vspace{#3}
```

```
2690 #2
```

```
2691 \vspace{#4}
```

```
2692 \begin{flushright} \$\Bigr\} \$\end{flushright}
```

```
2693 \par
```

```
2694 }
```

```
2695 \ExplSyntaxOff
```

```
2696
```

2.44 uml.sty

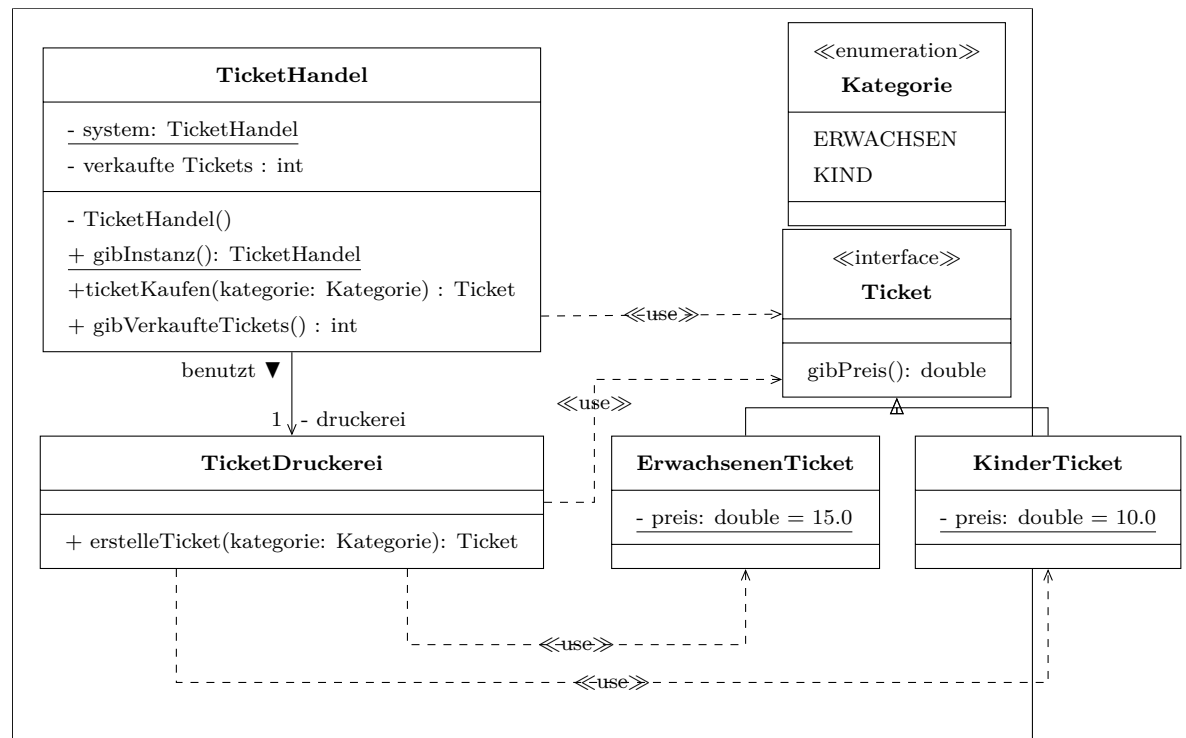
```

2697 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2698 \ProvidesPackage{lehramt-informatik-uml}[2020/06/13 Ein Hüll-Paket um
2699 `tikz-uml', das einige Design-Einstellungen vornimmt und manche
2700 Erweiterung bereitstellt]

2701 \RequirePackage{tikz-uml}
2702 \RequirePackage{tikz-uml-activity}
2703 % Not compatible with wasysym
2704 %\RequirePackage{mathabx}
2705 \RequirePackage{wasysym}
2706 \usetikzlibrary{positioning}

2707 \tikzumlset{
2708   fill class=white!0,
2709   font=\footnotesize,
2710   fill object=white!0,
2711   fill note=white!0,
2712   fill state=white!0,
2713   % Use case
2714   fill usecase=white!0,
2715   fill system=white!0,
2716 }

```



```

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

```

```

2717 \NewDocumentCommand{ \liUmlLeserichtung } { 0{dir=right} m } {
2718   \def\@liDirLeft{}
2719   \def\@liDirRight{ \RIGHTarrow}
2720   \pgfkeys{/lese/dir/.is choice}
2721   \pgfkeys{/lese/dir/up/.code={\def\@liDirRight{ \UParrow}}}
2722   \pgfkeys{/lese/dir/down/.code={\def\@liDirRight{ \DOWNarrow}}}
2723   \pgfkeys{/lese/dir/left/.code={\def\@liDirRight{}\def\@liDirLeft{\LEFTarrow }}}
2724   \pgfkeys{/lese/dir/right/.code={\def\@liDirRight{ \RIGHTarrow}}}
2725
2726   \def\@liPos{above}
2727   \pgfkeys{/lese/pos/.code={\def\@liPos{##1}}}
2728

```

```

2729 \def\@liDistance{0cm}
2730 \pgfkeys{/lese/distance/.code={\def\@liDistance{##1}}}}
2731
2732 \pgfkeys{/lese/.cd,#1}
2733
2734 \node[\@liPos = \@liDistance of #2-middle] {
2735   \@liDirLeft{\footnotesize#2}\@liDirRight
2736 };
2737 }
2738

```


2.45 vollstaendige-induktion.sty

2739 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2740 \ProvidesPackage{lehramt-informatik-vollstaendige-induktion}[2021/07/01
 2741 Hilfsmakros zum Setzen der Vollständigen Induktion, vor allem die
 2742 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

2743 \RequirePackage{lehramt-informatik-typographie}
 2744 \RequirePackage{lehramt-informatik-mathe}
 2745 \RequirePackage{lehramt-informatik-syntax}
 2746 \ExplSyntaxOn

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

Let-Abkürzung: \let\m=\liInduktionMarkierung

2747 \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

```
2748 \def\liInduktionErklaerung#1{\scriptsize\text{#1}}
```

\liInduktionAnfang

```
2749 \def\liInduktionAnfang{
```

```
2750   \liPseudoUeberschrift{Induktionsanfang}
```

```
2751
```

```
2752   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
2753   \liParagraphMitLinien{
```

```
2754     Beweise,~dass~ $A(1)$ ~eine~wahre~Aussage~ist.
```

```
2755   }
```

```
2756 }
```

\liInduktionVoraussetzung

```
2757 \def\liInduktionVoraussetzung{
```

```
2758   \liPseudoUeberschrift{Induktionsvoraussetzung}
```

```
2759
```

```
2760   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
2761   \liParagraphMitLinien{
```

```
2762     Die~Aussage~ $A(k)$ ~ist~wahr~für~ein~beliebiges~ $k \in \mathbb{N}$ .
```

```
2763   }
```

```
2764 }
```

\liInduktionSchritt

```
2765 \def\liInduktionSchritt{
```

```
2766   \liPseudoUeberschrift{Induktionsschritt}
```

```
2767
```

```
2768   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
2769   \liParagraphMitLinien{
```

```
2770     Beweise,~dass~wenn~ $A(n=k)$ ~wahr~ist,~
```

```
2771     auch~ $A(n=k+1)$ ~wahr~sein~muss.
```

```
2772   }
```

```
2773 }
```

```
2774 \ExplSyntaxOff
```

```
2775
```

2.46 wasserfall.sty

```
2776 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2777 \ProvidesPackage{lehramt-informatik-wasserfall}[2020/06/10]
2778 \RequirePackage{tikz}
2779 \tikzset{wasserfall/.style={
2780   >=stealth,
2781   node distance = 2mm and -8mm,
2782   start chain = A going below right,
2783   every node/.style = {
2784     draw,
2785     text width=24mm,
2786     minimum height=12mm,
2787     align=center,
2788     inner sep=1mm,
2789     fill=white,
2790     drop shadow={fill=black},
2791     on chain=A
2792   },
2793 }}
2794 \usetikzlibrary{chains,positioning,shadows}
2795
```

2.47 wpkalkuel.sty

2796 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2797 \ProvidesPackage{lehramt-informatik-wpkalkuel}[2020/06/13]

Let-Abkürzungen

\let\wp=\liWpKalkuel

\let\equivalent=\liWpEquivalent

\let\erklaerung=\liWpErklaerung

2798 \RequirePackage{amsmath}

2799 \ExplSyntaxOn

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

2800 \def\liWpKalkuelOhneMathe#1#2{

2801 \text{wp}(\texttt{\scriptsize"#1"},\thinspace #2)

2802 }

2803 \def\liWpKalkuel#1#2{

2804 \ifmmode

2805 \liWpKalkuelOhneMathe{#1}{#2}

2806 \else

2807 \$\liWpKalkuelOhneMathe{#1}{#2}\$

2808 \fi

2809 }

\MatheEnv

2810 \def\MatheEnv#1{

2811 \medskip

2812

2813 \hspace{1em}#1

2814

2815 \medskip

2816 }

\Mathe

2817 \def\Mathe#1{

2818 \MatheEnv{#1\$}

2819 }

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

2820 \def\liWpEquivalent#1{

2821 \MatheEnv{\$\equiv\$\hspace{1em}\$#1\$}

2822 }

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

2823 \newlength{@Skip@Erklaerung@Reset}

2824 \def\liWpErklaerung#1{

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

2826 \setlength{\leftskip}{0.5cm}

2827

2828 \par

2829 \noindent

2830 {

2831 \scriptsize

2832 #1

2833 }

2834 \par

2835

2836 \setlength{\leftskip}{@Skip@Erklaerung@Reset}

2837 }

\liWpErklaerungVerzweigung

```

2838 \def\liWpErklaerungVerzweigung{
2839   $\liWpKalkuelOhneMathe{if~\{-b~\}-then~\{-a1~\}-else~\{-a2~\}}{Q}
2840   \equiv
2841   (b \land \liWpKalkuelOhneMathe{a1}{Q})
2842   \lor
2843   (\neg b \land \liWpKalkuelOhneMathe{a2}{Q})$
2844 }

2845 \ExplSyntaxOff

2846

```

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		
\# 111	\AddToHook 1351	\bfseries .. 488, 1236,
\, 345, 400, 1183,	\advance 2426	1238, 2199, 2205,
1185, 1696, 1697,	\AfterEndEnvironment 2495	2207, 2209, 2210, 2579
1698, 2032, 2502, 2688	\allsectionsfont ... 1232	\Bigl 2688
\@Skip@Erklaerung@Reset	\Alph 1246	\Bigr 2692
... 2823, 2825, 2836	\alph 1246, 1247	\bigskip 50, 378,
\@afterheading 1663	\alpha 2602, 2604, 2605,	611, 616, 1572, 1900
\@afterindentfalse . 1663	2608, 2610, 2611,	\bool 323, 346
\@liDirLeft 2718, 2723, 2735	2612, 2613, 2614,	\bowtie
\@liDirRight 2719, 2721,	2618, 2624, 2625,	2347, 2350, 2351, 2352
2722, 2723, 2724, 2735	2630, 2631, 2632,	\Box 161
\@liDistance	2635, 2643, 2644, 2645	\boxtimes 476
... 2729, 2730, 2734	\arabic 1246, 2221, 2226,	
\@liPos .. 2726, 2727, 2734	2231, 2237, 2243, 2249	C
\\ 609,	\arraystretch 1922	\c 1329, 1330
632, 633, 636, 637,		\cdot 1740, 1793, 1804
640, 641, 733, 734,	B	\centerline
735, 842, 871, 873,	\BeforeBeginEnvironment	1370, 2300, 2322, 2337
899, 908, 953, 995, 2494	\chapter 1236, 1237
996, 997, 1002,	\begin .. 630, 681, 696,	\char 1570
1003, 1004, 1024,	731, 755, 802, 834,	\clearpage 1564
1570, 1927, 1985, 1988	849, 869, 879, 895,	\cline 609
\{ 209, 1120,	915, 947, 964, 993,	\clist 228, 272,
1130, 1142, 1143,	1018, 1039, 1054,	273, 292, 296, 2434
1148, 1182, 1411,	1163, 1252, 1327,	\columnbreak 2469
1972, 2377, 2688, 2839	1365, 1374, 1381,	\cs 295, 320, 344,
\} 209, 1120,	1489, 1562, 1614,	345, 382, 394, 1646
1130, 1142, 1143,	1619, 1627, 1652,	\csname 1314, 1317
1150, 1186, 1412,	1657, 1668, 1678,	\cup 1143,
1972, 2377, 2692, 2839	1682, 1765, 1769,	2006, 2611, 2625, 2632
_ .. 27, 35, 44, 46, 48,	1785, 1808, 1831,	
295, 320, 344, 345,	1846, 1925, 1926,	D
359, 360, 366, 369,	1976, 2021, 2141,	\DeclareMathSymbol ..
372, 382, 394, 1352	2161, 2302, 2324, 1870, 1871
	2338, 2494, 2505, 2692	\DecoINERT
	\begingroup 1563, 1974, 2422	2224, 2297, 2298, 2320
	\beschriftung	\DecoINERTwithPivot .
 1594, 1598, 2240, 2319
	1602, 1606, 1610, 1612	\DecoLEFT 2219, 2318
A	\beta 2602,	\DecoLEFTwithPivot ..
\addbibresource	2605, 2609, 2610, 2234, 2296
1540, 1541, 1542,	2611, 2614, 2624,	\DecoRIGHT 2229, 2321, 2331
1543, 1544, 1545,	2625, 2626, 2631, 2632	\DecoRIGHTwithPivot .
1546, 1547, 1548, 1549	\bf 2199, 2200, 2201 2246, 2299

\definecolor 1234	liLernkartei 1666	1975, 1992, 2000,
\delta	67, 109, 167, 209, 1134	liProduktionsRegeln		2109, 2118, 2361,
\dh 1698, 2613	 1158	2500, 2652, 2709, 2735
\directlua	liProjektSprache	1586	\footrulewidth . 1263, 1523
	60, 139, 197, 202,	liQuellen 1646	\foreach . 1329, 1332, 1339
	1115, 1129, 1149,	liRelationenSchemaFormat		\forestFirst .. 2449, 2452
	1157, 1164, 1169,	2381	\forestLast ... 2450, 2452
	1957, 1962, 2015,	liRmodell 2361	\forestOget ... 2449, 2450
	2022, 2029, 2389,	liUebergangsTabelle		\forestOnes 2462
	2481, 2513, 2518,	1922	\forestOv 2451, 2452, 2455
	2519, 2523, 2528,	\equiv 2821, 2840	\forestov . 2441, 2445,
	2529, 2533, 2539, 2540	\erzeuge@tiefgestellt		2446, 2449, 2450,
\do 2220, 2225,	... 1129, 1130, 1134		2451, 2452, 2454, 2455
	2230, 2235, 2241, 2247	\expandafter	\forestset 2437, 2460
\dots 519, 523,		1314, 2255, 2257,	\forestSortLevel
	1411, 2148, 2624, 2625		2258, 2259, 2267, 2425	2439, 2447, 2461, 2462
\DOWNarrow 2722	\ExplSyntaxOff	\frac 1742, 1773, 1804, 1819
\draw 1334, 1337,	. 52, 94, 136, 141,		\fullouterjoin 2352
	1340, 2096, 2395, 2398	194, 199, 204, 541,		
		563, 578, 1110,		
		1188, 1223, 1354,		
		1509, 1524, 1688,		
		1763, 1950, 2124,		
		2374, 2435, 2553,		
		2660, 2695, 2774, 2845		
		\ExplSyntaxOn 22,	
		63, 104, 137, 162,		
		195, 200, 225, 483,		
		547, 564, 1083,		
		1178, 1200, 1350,		
		1502, 1514, 1589,		
		1702, 1932, 2105,		
		2362, 2432, 2480,		
		2576, 2670, 2746, 2799		
		F		
		\faCheckSquare0 2672	
		\faCircleThin 1102	
		\faGg 1096	
		\fancyfoot	
			1258, 1259, 1260,	
			1518, 1519, 1520, 1521	
		\fancyhead	
			1257, 1515, 1516, 1517	
		\faSquare0 1090	
		\fi	585, 593, 601, 1126,	
			1140, 1176, 1613,	
			1715, 1725, 1735,	
			1749, 1970, 2007,	
			2093, 2260, 2452,	
			2453, 2456, 2458, 2808	
		\fontspec 1232	
		\footcite	
			693, 715, 762, 785,	
			817, 912, 939, 985,	
			1390, 1393, 1400,	
			1405, 1410, 1414,	
			1420, 1425, 1529,	
			1781, 1782, 1937, 2154	
		\footnote 1690, 1694	
		\footnotesize	
			149, 355, 437, 533,	
			1107, 1585, 1638,	
			1653, 1669, 1881,	
		G		
		\g	29, 37, 272, 273, 292,	
			296, 302, 303, 304,	
			305, 306, 308, 309,	
			311, 313, 314, 315,	
			316, 317, 318, 321,	
			325, 326, 327, 330,	
			332, 333, 334, 335,	
			336, 337, 348, 349,	
			350, 351, 359, 360,	
			362, 368, 369, 371,	
			372, 374, 375, 383,	
			388, 390, 395, 397, 401	
		\Gamma	. 108, 166, 209, 1143	
		\gappto 2457	
		\geq 1418,	
			2137, 2142, 2158, 2162	
		H		
		\hbox 2347	
		\headrulewidth	. 1262, 1522	
		\hfill	1789, 1795, 1800, 2680	
		\hinweis 1585	
		\hline 1927	

\liEntwurfsAbstrakteFabrik	\liEntwurfsModellPraesentation	\liSteuerDiagramm	104
..... 676 1017, 1035	\liKellerKante	142
\liEntwurfsAbstrakteFabrikCode	\liEntwurfsZustand	\liKellerUebergang	..
..... 671, 678	\liEntwurfsZustandAkteure 137, 143	
\liEntwurfsAbstrakteFabrikUml 1053, 1073	\liKontrollCode	1500
..... 629, 677	\liEntwurfsZustandUml	liKontrollflussgraph	
\liEntwurfsAdapter 1038, 1072	(environment)	1488
\liEntwurfsAdapterAkteure	\liEpsilon	\liKontrollKnotenPfad 1502
..... 695, 727	\liErAttribute 1501, 1506	
\liEntwurfsAdapterCode	... 1086, 1100, 1102	\liKurzeTabellenLinie	609
..... 719, 728	\liErDatenbankName	\liLadeAllePakete	.. 230
\liEntwurfsAdapterUml	\liErEntity 1084, 1088, 1090	\liLadePakete	.. 56, 59,
..... 680, 726	\liErledigt	226, 231, 485, 546,	
\liEntwurfsBeobachter	\liErMpAttribute	1119, 1349, 1360,	
\liEntwurfsBeobachterAkteure	\liErMpEntity	1876, 1956, 2104, 2575	
..... 754, 798	\liErMpRelationship	\liLatexCode	2503
\liEntwurfsBeobachterCode	\liErRelationship	\liLeereZelle	1914
..... 788, 799	... 1085, 1094, 1096	liLernkartei (environ-	
\liEntwurfsBeobachterUml	\liExamensAufgabe	ment)	1666
..... 730, 797	\liExamensAufgabeA	\liLinksReduktion	.. 1981
\liEntwurfsDekorierer	\liExamensAufgabeTA	\liLinksReduktionInline 1990, 1998
\liEntwurfsDekoriererAkteure	\liExamensAufgabeTTA	\liMasterExkurs 1845
..... 830	liExkurs (environment)	\liMasterFaelle	1784, 1852
\liEntwurfsDekoriererCode	\liFalsch	\liMasterFallRechnung 1830
..... 820, 831	\liFlaci	liMasterVariablen	..
\liEntwurfsDekoriererUml	\liFremd 1764, 1847	
..... 801, 829	\liFunktionaleAbhaengigkeit	\liMasterVariablenDeklaration 1807
\liEntwurfsEinfacheFabrik	... 2003, 2006, 2014	\liMasterWolframLink	1855
..... 864	\liFunktionaleAbhaengigkeiten	\liMenge	... 73, 74, 76,
\liEntwurfsEinfacheFabrikAkteure 2017	115, 116, 117, 121,	
..... 848, 866	\liFussnote	173, 174, 175, 179,	
\liEntwurfsEinfacheFabrikUml	... 1877, 1879	1120, 1169, 1208, 1209	
..... 833, 865	\liFussnoteDreiText	\liMengeOhneMathe	...
\liEntwurfsEinzelstueck 1891, 1909	1120, 1123, 1125	
..... 889	\liFussnoteEinsText	\liMinimierungErklaerung 1936
\liEntwurfsEinzelstueckAkteure 1883, 1903	\liMinispracheDatei	2546
..... 878, 891	\liFussnoteLink	\linespread 2368
\liEntwurfsEinzelstueckCode 1693	\liNichtsZuTun 2673
..... 886, 892	\liFussnoten	\liO 1727, 1755
\liEntwurfsEinzelstueckUml	\liFussnoteUrl	\liOmega 1717, 1761
..... 868, 890	\liFussnoteVierText	\liOmegaOhneMathe	...
\liEntwurfsErbauer 1895, 1912	... 1717, 1722, 1724	
\liEntwurfsErbauerAkteure	\liFussnoteZweiText	\liOOhneMathe
..... 914, 944 1887, 1906	... 1727, 1732, 1734	
\liEntwurfsErbauerUml	\liGeschweifteKlammern	\liParagraphMitLinien	.. 534, 1938, 2653,
..... 894, 943	... 1161, 2018, 2684	2674, 2753, 2761, 2769	
\liEntwurfsFabrikmethode	\liGrammatik	\liPetriErreichKnotenDrei 2098
..... 988, 1013	liGraphenFormat (envi-	\liPetriErreichTransition 2095
\liEntwurfsFabrikmethodeAkteure	ronment) 1306	\liPetriSetzeSchluessel 2040
..... 963, 990	\liHanoi	\liPetriTransitionsName 2087, 2099
\liEntwurfsFabrikmethodeUml	\liHaskellCode	\liPetriTransitionsNameOhneMathe	... 2087, 2090, 2092
..... 946, 989	\liHaskellDatei 2087, 2090, 2092	
\liEntwurfsKompositum	\liInduktionAnfang		
..... 1013	\liInduktionErklaerung		
\liEntwurfsKompositumAkteure 2748		
..... 1015	\liInduktionMarkierung		
\liEntwurfsKompositumUml 2747		
..... 992, 1014	\liInduktionSchritt		
\liEntwurfsModellPraesentation	\liInduktionVoraussetzung		
..... 1034 2757		
\liEntwurfsModellPraesentation	\liJavaCode		
..... 1036 2502		
	\liJavaExamen		
 2531		
	\liJavaTestAkteure		
 2521		
	liKasten (environment)		
 1251		

<code>\liPetriTransPfeile</code> 2099	<code>\liTheta</code> 1707 ,	<code>\loop</code> 2181	
<code>\liPolynomiellReduzierbar</code> 1380	1758, 1787, 1793, 1798	<code>\lor</code> 2842	
<code>\liPotenzmenge</code> 1129, 1133, 2110	<code>\liThetaOhneMathe</code> 1707, 1712, 1714	<code>\ltimes</code> 1863	
<code>\liPotenzmengeOhneMathe</code> 1130, 1131, 1132	<code>\liTOhneMathe</code> 1737, 1746, 1748	M	
<code>\liPrimaer</code> 2359	<code>\liTuringKante</code> 205	<code>\makeatletter</code> 1662, 2420	<code>\makeatother</code> 1664, 2428
<code>\liProblemBeschreibung</code> 1364	<code>\liTuringLeerzeichen</code> 161, 169	<code>\marginpar</code> 1089, 1095, 1101, 1569	
<code>\liProblemClique</code> 1387	<code>\liTuringMaschine</code> 162	<code>\mathbb</code> 1418, 2168, 2762	<code>\mathbin</code> 2350, 2351, 2352
<code>\liProblemName</code> 1363, 1370, 1382, 1384, 1397, 1408, 1409, 1417, 1418	<code>\liTuringUeberfuehrung</code> 208	<code>\mathcal</code> 1728, 2631, 2636, 2638, 2639, 2640	<code>\Mathe</code> 2817
<code>\liProblemSat</code> 1416	<code>\liTuringUebergaenge</code> 200, 206	<code>\MatheEnv</code> 2810, 2818, 2821	<code>\mathord</code> 1870, 1871
<code>\liProblemSubsetSum</code> 1407, 1416	<code>\liTuringUebergangZelle</code> 195	<code>\mathord</code> 1870, 1871	<code>\mdfsetup</code> 1250, 1595, 1599, 1603, 1607
<code>\liProblemVertexCover</code> 1387, 1395	<code>\liUeberfuehrungsFunktion</code> 1134	<code>\nag</code> 1569	<code>\medskip</code> 1372, 1580, 1582, 1656, 1684, 2366, 2372, 2682, 2686, 2811, 2815
<code>\liProduktionen</code> 1168, 1210	<code>\liUeberfuehrungsFunktionOhneMathe</code> 1134, 1137, 1139	<code>\nag</code> 1569	<code>\mintinline</code> 2502, 2503, 2542, 2549, 2554
<code>liProduktionsRegeln</code> 1158	<code>liUebergangsTabelle</code> 1922	<code>\mkern</code> 2350, 2351, 2352	<code>\mlq</code> 1868, 1870
<code>(environment)</code> 1158	<code>(environment)</code> 1932	<code>\mrq</code> 1868, 1871	<code>\msg</code> 39, 405
<code>liProjektSprache</code> 1586	<code>\liUeberschriftDreiecksTabelle</code> 1932	<code>\myList</code> 2441, 2442, 2443, 2446	<code>\myNodes</code> 2430, 2445, 2451, 2455, 2457
<code>(environment)</code> 1586	<code>\liUmlLeserichtung</code> 2717		
<code>\liPseudoUeberschrift</code> 1571, 1621, 1622, 1924, 1934, 2750, 2758, 2766	<code>\liVertauschen</code> 2388		
<code>\liPumpingKontextfrei</code> 2156	<code>\liWortInSprache</code> 610		
<code>\liPumpingRegulaer</code> 2135	<code>\liWortNichtInSprache</code> 615		
<code>liQuellen</code> 1646	<code>\liWpEquivalent</code> 2820		
<code>(environment)</code> 1646	<code>\liWpErklaerung</code> 2823		
<code>\liRechtsReduktionInline</code> 1998	<code>\liWpErklaerungVerzweigung</code> 2838		
<code>\liRekursionsGleichung</code> 1751, 1811	<code>\liWpKalkuel</code> 2800		
<code>\liRelation</code> 2028	<code>\liWpKalkuelOhneMathe</code> 2800, 2805, 2807, 2839, 2841, 2843		
<code>liRelationenSchemaFormat</code> 2381	<code>\liZustandsBuchstabe</code> 1144, 1153, 1155, 1173, 1175		
<code>(environment)</code> 2381	<code>\liZustandsBuchstabeGross</code> 1145, 1154, 1156		
<code>\liRelationMenge</code> 2375	<code>\liZustandsmenge</code> 1132		
<code>\liRichtig</code> 476	<code>\liZustandsmengeNr</code> 1146, 2119		
<code>liRmodell</code> 2361	<code>\liZustandsmengeNrGross</code> 1154		
<code>(environment)</code> 2361	<code>\liZustandsMengenSammlung</code> 2106		
<code>\liRundeKlammer</code> 1704, 1708, 1718, 1728, 1742	<code>\liZustandsMengenSammlungNr</code> 2115		
<code>\liSortierMarkierung</code> 2400	<code>\liZustandsmengeOhneMathe</code> 1132		
<code>\liSortierPfeil</code> 2394	<code>\liZustandsname</code> 1155		
<code>\liSortierPfeilUnten</code> 2397	<code>\liZustandsnameGross</code> 1156, 2107, 2116		
<code>\liSpaltenUmbruch</code> 2469	<code>\liZustandsnameTiefgestellt</code> 1171		
<code>\liSqlCode</code> 2554	<code>\liZustandsPaar</code> 1916		
<code>\listen@punkt</code> 1646, 1658	<code>\liZustandsPaarVariablenName</code> 1915, 1918, 1919		
<code>\liStrich</code> 1362	<code>\llap</code> 2348		
<code>\liSyntheseErklaerung</code> 2593, 2658	<code>\log</code> 1755, 1758, 1761, 1787, 1793		
<code>\liSyntheseUeberErklaerung</code> 2656			
<code>\liSyntheseUeberschrift</code> 2577, 2657			
<code>\liT</code> 1737, 1752, 1766, 1827			
<code>\liTeilen</code> 1961			
<code>\literatur</code> 1528, 1552			

1488, 1586, 1587, 1590, 1617, 1625, 1647, 1666, 1676, 1923, 1973, 2363, 2381	\ProvidesPackage 2, 20, 55, 223, 412, 473, 480, 544, 607, 622, 1077, 1113, 1226, 1266, 1275, 1280, 1309, 1347, 1357, 1429, 1512, 1527, 1533, 1555, 1701, 1861, 1874, 1953, 2036, 2102, 2127, 2132, 2176, 2342, 2355, 2384, 2465, 2472, 2477, 2557, 2572, 2663, 2667, 2698, 2740, 2777, 2797	475, 545, 624, 625, 1079, 1081, 1082, 1118, 1227, 1228, 1231, 1233, 1235, 1240, 1249, 1256, 1267, 1268, 1277, 1281, 1282, 1283, 1311, 1312, 1361, 1430, 1529, 1534, 1535, 1551, 1558, 1559, 1560, 1588, 1703, 1864, 1865, 2037, 2039, 2129, 2189, 2191, 2343, 2344, 2345, 2358, 2386, 2417, 2468, 2474, 2479, 2490, 2491, 2559, 2664, 2671, 2701, 2702, 2704, 2705, 2743, 2744, 2745, 2778, 2798
\newlength 2823		
\node 561, 1496, 1501, 2221, 2226, 2231, 2237, 2243, 2249, 2400, 2445, 2734		
\noexpand 2309, 2310, 2311, 2330, 2445		
\noindent 357, 612, 617, 1573, 1575, 1579, 1583, 1611, 1639, 1641, 1654, 1670, 1672, 1680, 1849, 1902, 1905, 1908, 1911, 2376, 2675, 2687, 2829		
\nolinkurl 2508		
\normalsize 1238		
\notin 618		
\null 2680		
O		
\o@join 2347, 2350, 2351, 2352		
\Omega 1718		
\omega 2137, 2138, 2158, 2159		
\or 2258		
P		
\pagestyle 1261		
\par ... 356, 377, 538, 1190, 1198, 1578, 1640, 1663, 1671, 2300, 2322, 2337, 2378, 2657, 2681, 2685, 2693, 2828, 2834		
\paragraph 1238		
\parindent 2369		
\path 96, 143, 206, 576		
\pgfkeys .. 2051, 2720, 2721, 2722, 2723, 2724, 2727, 2730, 2732		
\pgfmath@count 2422, 2424, 2426		
\pgfmath@smuggleone 2427		
\pgfmathdeclarefunction 2421		
\pgfmathhint 2422		
\pgfmathparse 1320, 2439, 2444, 2447, 2461, 2462		
\pgfmathresult 1321, 2422, 2423, 2425, 2427, 2440, 2448, 2461, 2462		
\pgfutil@empty 2423		
\pgfutil@loop 2424		
\pgfutil@repeat 2427		
\preceq 1383		
\prime 1362		
\printbibliography . 1552		
	\Q	
	\QS@list 2302, 2313, 2317, 2324, 2330, 2335, 2338	
	\QS@select@equal 2277, 2281	
	\QS@select@greater 2278, 2282	
	\QS@select@smaller 2273, 2276, 2280	
	\QS@sort@a 2255, 2288, 2309, 2310	
	\QS@sort@b 2255, 2256	
	\QS@sort@c 2259, 2266	
	\QS@sort@d 2267, 2275	
	\QS@sort@empty . 2257, 2262	
	\QS@sort@single 2258, 2263	
	\QSinitialize 2180, 2292, 2327	
	\QSIr . 2263, 2269, 2277, 2297, 2311, 2316, 2319	
	\QSIrr 2298, 2311, 2312, 2320	
	\QSLr 2269, 2276, 2287, 2288, 2296, 2309, 2314, 2318	
	\QSpivotStep 2182, 2292, 2296, 2307	
	\QSR 2269	
	\QSRr 2278, 2299, 2310, 2315, 2321, 2330, 2331, 2332	
	\QSSortStep 2184, 2292, 2308, 2309	
	\quad 1880	
	R	
	\raisebox 1501	
	\relax 1564, 2269, 2312, 2314, 2315, 2316, 2424, 2426	
	\renewcommand ... 1241, 1242, 1243, 1244, 1247, 1248, 1262, 1263, 1522, 1523, 1922	
	\repeat 2185	
	\RequirePackage 57, 160, 224, 228, 408, 414, 415,	
		725, 745, 624, 625, 1079, 1081, 1082, 1118, 1227, 1228, 1231, 1233, 1235, 1240, 1249, 1256, 1267, 1268, 1277, 1281, 1282, 1283, 1311, 1312, 1361, 1430, 1529, 1534, 1535, 1551, 1558, 1559, 1560, 1588, 1703, 1864, 1865, 2037, 2039, 2129, 2189, 2191, 2343, 2344, 2345, 2358, 2386, 2417, 2468, 2474, 2479, 2490, 2491, 2559, 2664, 2671, 2701, 2702, 2704, 2705, 2743, 2744, 2745, 2778, 2798
		\right 1705
		\Rrightarrow ... 2719, 2724
		\Rrightarrow 613, 618
		\rrightarrow 209, 502, 507, 515, 519, 521, 522, 524, 579, 587, 2099, 2602, 2609, 2611, 2614, 2619, 2624, 2625, 2630
		\rightouterjoin 2351
		\Roman 1246
		\roman 1246, 1248
		\romannumeral 2267
		\rtimes 1863
		\rule 2300, 2322, 2337, 2348
	S	
	\sb 69, 79, 81, 110, 168, 514, 515, 519, 522, 523, 524, 1173, 1175, 1755, 1758, 1761, 1787, 1793, 1942, 2087, 2096, 2624, 2625, 2626, 2631, 2635, 2636, 2639, 2640, 2643, 2644, 2645	
		\scriptscriptstyle 579, 587, 595
		\scriptsize 1192, 1441, 1448, 1454, 1516, 1517, 1520, 1521, 2748, 2801, 2831
		\section 48
		\seq . 1505, 1506, 1507, 1649, 1650, 1651, 1658
		\setbox 2347
		\setcounter 1239, 2301, 2323, 2337
		\setganttlinklabel .. 1269, 1270, 1271, 1272
		\setlength 2369, 2825, 2826, 2836

\xintLength	2255		Z		\zustandsnamens@liste
			\ZB	<u>1697</u>	... 1146, 1153, 1154
\xintnthelt	2267	\zB	<u>1696</u>	