

lehramt-informatik

Hermine Bschlangaul *

September 1, 2021

Contents

1	Klassen	2
1.1	Vorlage Theorie-Teil	3
1.2	Vorlage Aufgabensammlung	4
1.3	Vorlage Aufgabe	5
2	Pakete	6
2.1	aufgaben-einbinden.sty	7
2.2	aufgaben-metadaten.sty	8
2.3	automaten.sty	9
2.3.1	Endlicher Automat	9
2.3.2	Kellerautomat	11
2.3.3	Turingmaschine	12
2.4	basis.sty	15
2.5	baum.sty	19
2.5.1	Binärbaum	20
2.5.2	AVL-Baum	21
2.5.3	B-Baum	22
2.6	checkbox.sty	23
2.7	chomsky-normalform.sty	24
2.8	cpm.sty	26
2.8.1	Hilfsmakros für Zeitpunkt-Tabelle	27
2.9	cyk-algorithmus.sty	29
2.10	entwurfsmuster.sty	30
2.10.1	Namensschema der Entwurfsmuster-Makros:	30
2.10.2	Reihenfolge	30
2.10.3	Abstrakte Fabrik (Abstract Factory)	30
2.10.4	Adapter	31
2.10.5	Beobachter (Observer)	33
2.10.6	Dekorierer (Decorator)	35
2.10.7	Einfache Fabrik (Simple Factory)	36
2.10.8	Einzelstück (Singleton)	37
2.10.9	Erbauer (Builder)	37
2.10.10	Fabrikmethode (Factory Method)	39
2.10.11	Kompositum (Composite)	40
2.10.12	Modell-Präsentation-Steuerung (Model-View-Controller)	41
2.10.13	Zustand (State)	42
2.11	er.sty	44
2.11.1	Tex-Markup-Beispiel: Komplettes Diagramm	44
2.11.2	Tex-Markup-Beispiel: EER Enhanced Entity-Relation-Modell nach Elmasri/Navante	45
2.12	formale-sprachen.sty	47
2.13	formatierung.sty	50

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

2.13.1	Schriftarten / Typographie	50
2.13.2	Farben	50
2.13.3	Überschriften	50
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
2.41.4	TeX-Markup Relationen formen	98
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 \section{\_gib_aufgaben_titel:}
47
48 \bigskip
49 }

50 \ExplSyntaxOff

51
```

2.3 automaten.sty

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

2.3.1 Endlicher Automat

```
54 \liLadePakete{formale-sprachen}

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

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



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

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

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

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

```

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

```

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

```

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

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

```

2.3.2 Kellerautomat

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

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

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

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



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

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

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

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

```

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

```

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

(a, #: A)
(b, #: ε)

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

```

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

```

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

```

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

```

2.3.3 Turingmaschine

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

\liTuringLeerzeichen

□

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

```

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

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

```

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

```

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

```

\liTuringUebergangZelle Formatiert einen Zustandsübergang für eine Übergangstabelle.

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

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

```

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

```

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

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

`(z1: \square , L)`

`(\square : \square , R)`

```

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

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

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

\liTuringUeberfuehrung

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

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

```

2.4 basis.sty

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

222 \RequirePackage{xparse}

223 \ExplSyntaxOn

\liLadePakete

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

\liLadeAllePakete

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

Definition einer Komma-getrennten-Liste mit deren Hilfe die vielen globalen Token-List-Variablen definiert werden können. Die einzelnen Schlüssel sind im Interface `AufgabenMetadaten` in der Typescript-Datei `.scripts/nodejs/src/aufgaben.ts` erklärt.

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



```

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

```

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

```

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

```

Funktion über alle globalen Token-List-Variablen zurückzusetzen.

```

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

```

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

```

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

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

```

```

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

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

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

```

```

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

```

2.5 baum.sty

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

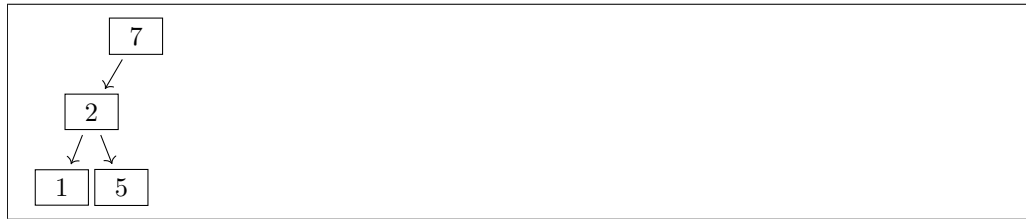
    für li binaer baum
413 \RequirePackage{tikz-qtree}

    Für b baum
414 \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}
```



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

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}

```



```

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

```

2.6 checkbox.sty

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

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

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

476
```


2.7 chomsky-normalform.sty

```

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

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

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

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

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

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

```

```
534 }
```

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

```
535 \def\liChomskyUeberErklaerung#1{
```

```
536   \liChomskyUeberschrift{#1}\par
```

```
537   \liChomskyErklaerung{#1}
```

```
538 }
```

```
539 \ExplSyntaxOff
```

```
540
```

2.8 cpm.sty

```

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

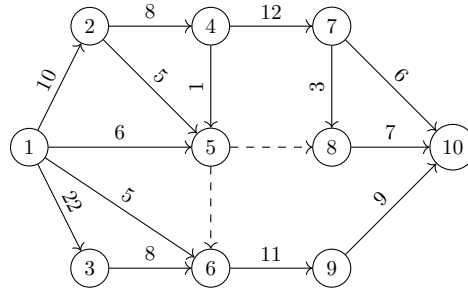
```

Let-Abkürzungen

```

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

```



```

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

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

\liCpmVorgang[schein]{5}{6}{5}
\liCpmVorgang[schein]{5}{8}{7}
\end{tikzpicture}

```

```

\begin{tabular}{|l|l|l|l|l|l|l|l|}
\hline
i & a & b & c & d & e & f & g \\
\hline
FZ & 0 & 2 & 5 & 6 & 10 & 3 & 12 \\
\hline
SZ & 0 & 2 & 5 & 10 & 10 & 6 & 12 \\
\hline
GP & 0 & 0 & 0 & 3 & 0 & 3 & 0 \\
\hline
\end{tabular}

```

```

\liCpmEreignis \liCpmEreignis\{(.*)\}\{((.*),(.*))\} -> \liCpmEreignis{$1}{$2}{$3}

```

```

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

```

```

558
559 \node[circle,draw] (\l_name_tl) at (#3,#4) {#2};
560 }
561 \ExplSyntaxOff

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

```

2.8.1 Hilfsmakros für Zeitpunkt-Tabelle

```

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

```

```

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

```

```

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

```

```

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

```

```

599 \fi%
600 }

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

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

603

```

2.9 cyk-algorithmus.sty

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

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

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

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

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

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

618
```

2.10 entwurfsmuster.sty

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

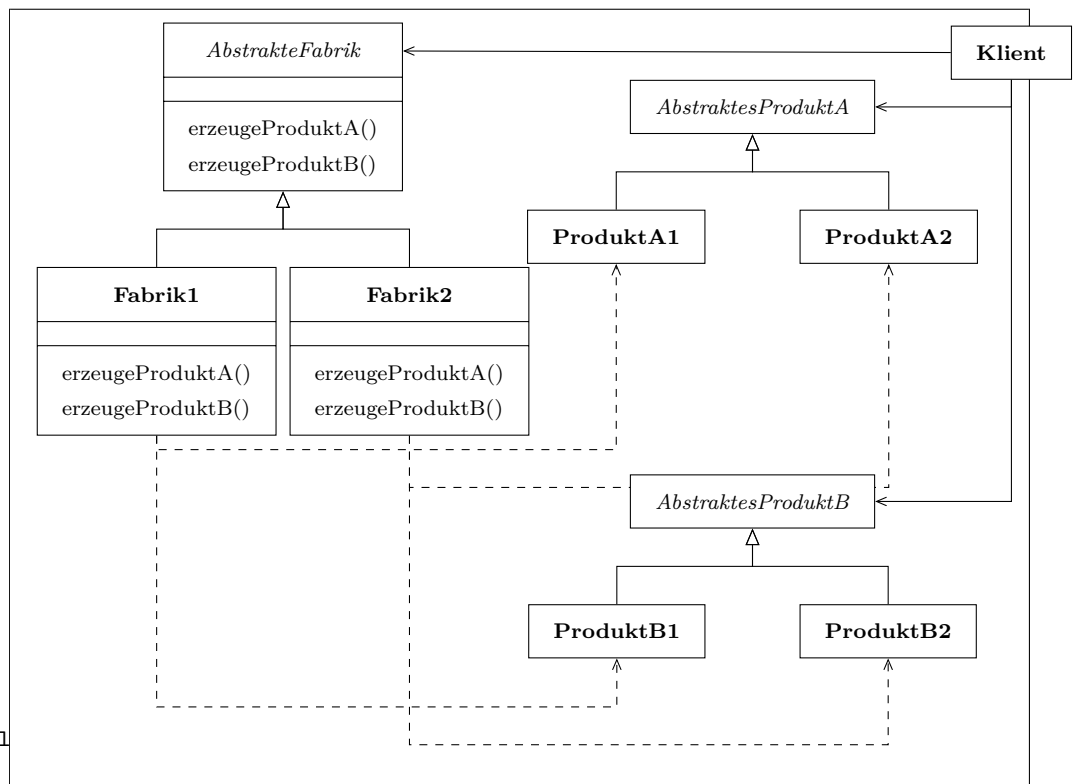
1. Uml: Uml-Klassendiagramm \liEntwurfsEinzelstueckUml
2. Akteure: Akteure, beteiligte Klassen \liEntwurfsEinzelstueckAkteure
3. Code: Allgemeines Code-Beispiel \liEntwurfsEinzelstueckCode
4. ohne: Ohne Suffix, Bündelung der einzelnen Makros eines Entwurfsmusters \liEntwurfsEinzelstueckAkteure

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

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

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

2.10.3 Abstrakte Fabrik (Abstract Factory)



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

```

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

```

iEntwurfsAbstrakteFabrikCode

```

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

```

\liEntwurfsAbstrakteFabrik

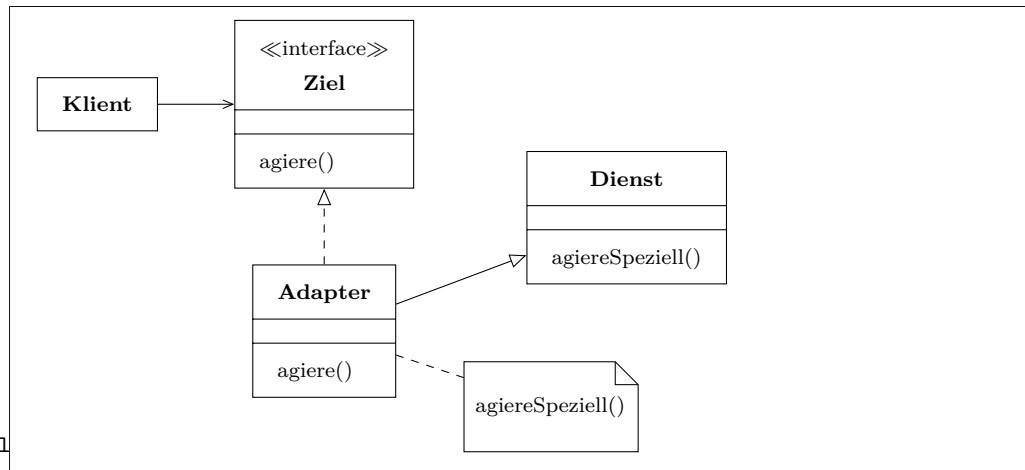
```

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

```

2.10.4 Adapter

\liEntwurfsAdapterUml



```

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

```

\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.

```

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

```

```

714
715 \end{description}
716 }

```

\liEntwurfsAdapterCode

```

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

```

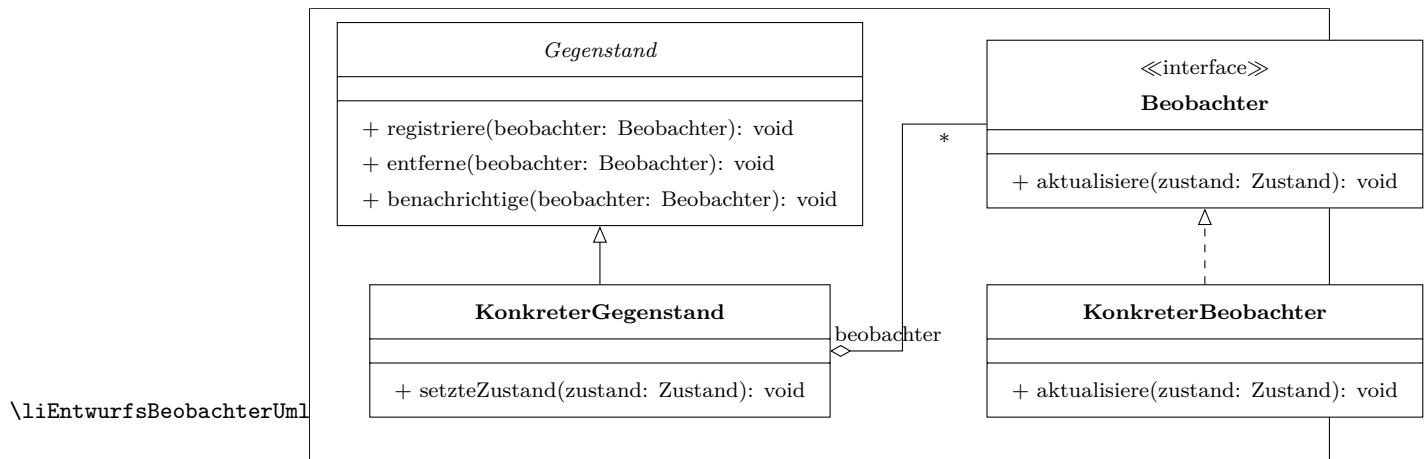
\liEntwurfsAdapter

```

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

```

2.10.5 Beobachter (Observer)



```

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

```

\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.

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

\liEntwurfsBeobachterCode

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

```

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

```

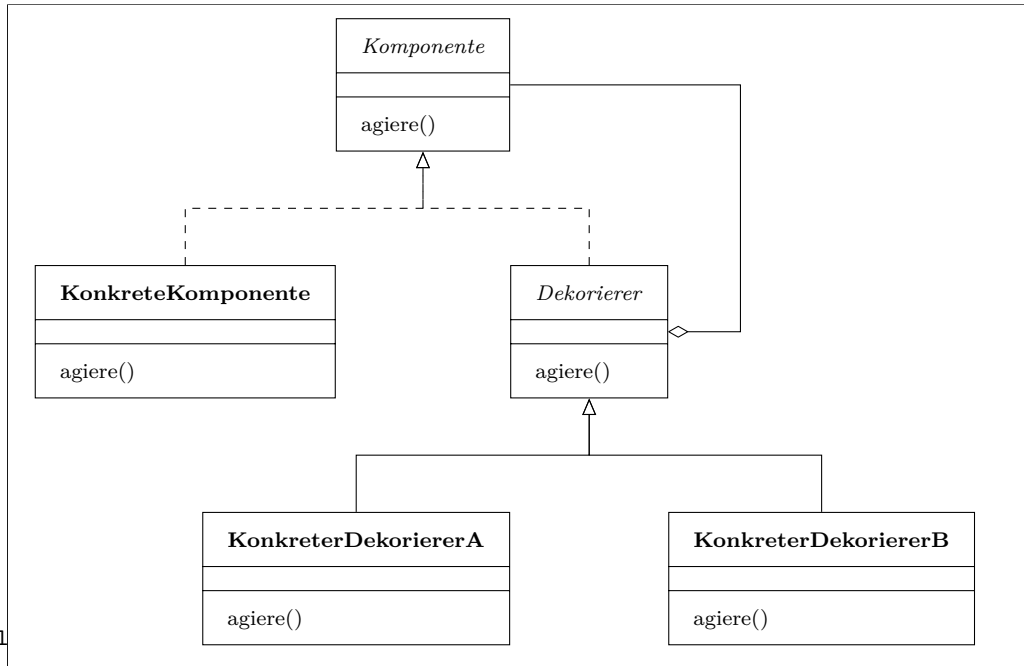
\liEntwurfsBeobachter

```

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

```

2.10.6 Dekorierer (Decorator)



\liEntwurfsDekoriererUml

```

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

```

\liEntwurfsDekoriererCode

```

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

```

\liEntwurfsDekorierer

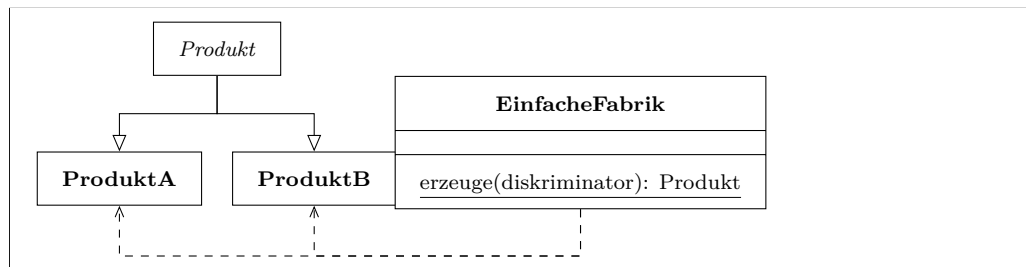
```

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

```

2.10.7 Einfache Fabrik (Simple Factory)

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



```

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

```

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.

```

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

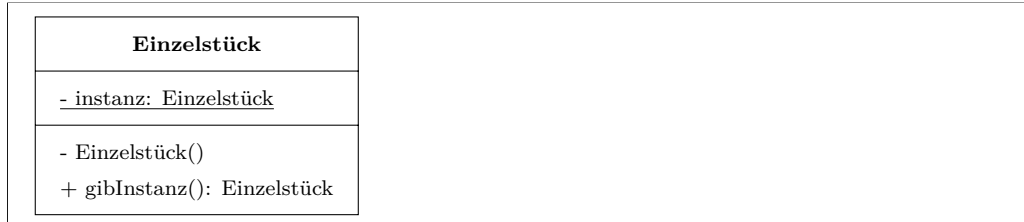
```

\liEntwurfsEinfacheFabrik

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

2.10.8 Einzelstück (Singleton)

\liEntwurfsEinzelstueckUml Quelle nach der deutschen Wikipedia



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

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.

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

\liEntwurfsEinzelstueckCode

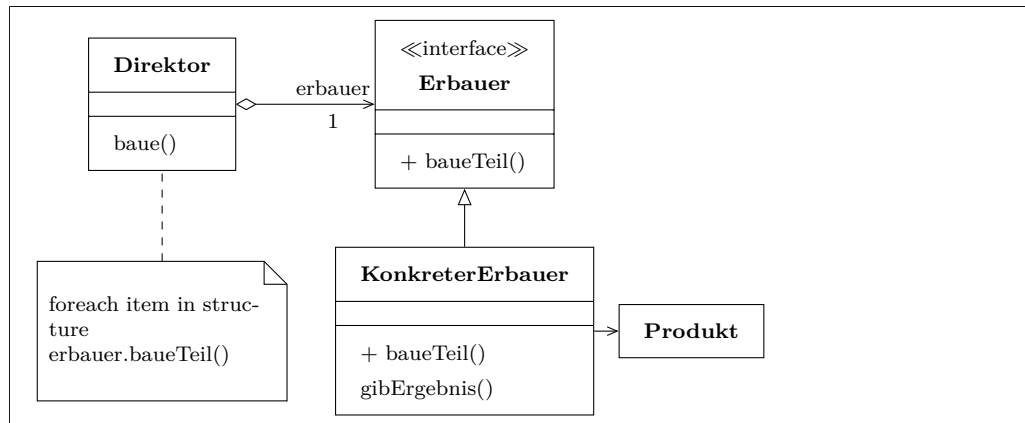
```
884 \def\liEntwurfsEinzelstueckCode{
885   \li@EntwurfsCode{einzelstueck}{Einzelstueck}
886 }
```

\liEntwurfsEinzelstueck

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

2.10.9 Erbauer (Builder)

\liEntwurfsErbauerUml Quelle nach der deutschen Wikipedia



```

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

```

\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.

```

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

```

```

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

```

\liEntwurfsErbauer

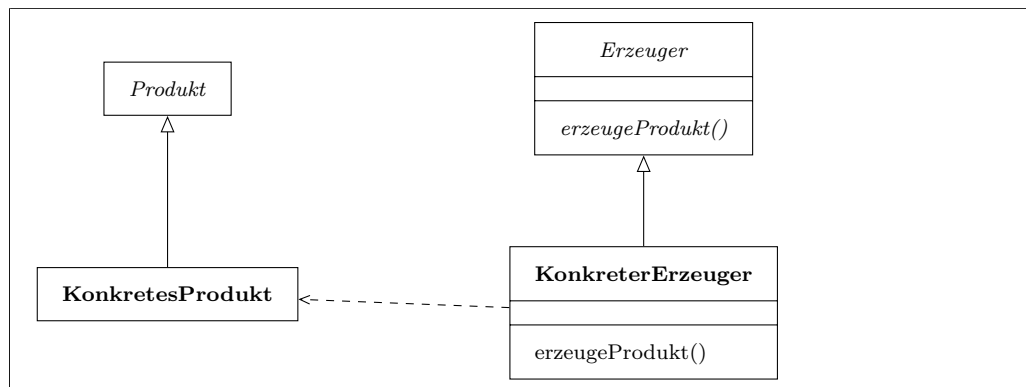
```

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

```

2.10.10 Fabrikmethode (Factory Method)

\liEntwurfsFabrikmethodeUml Quelle nach der deutschen Wikipedia



```

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

```

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).

```

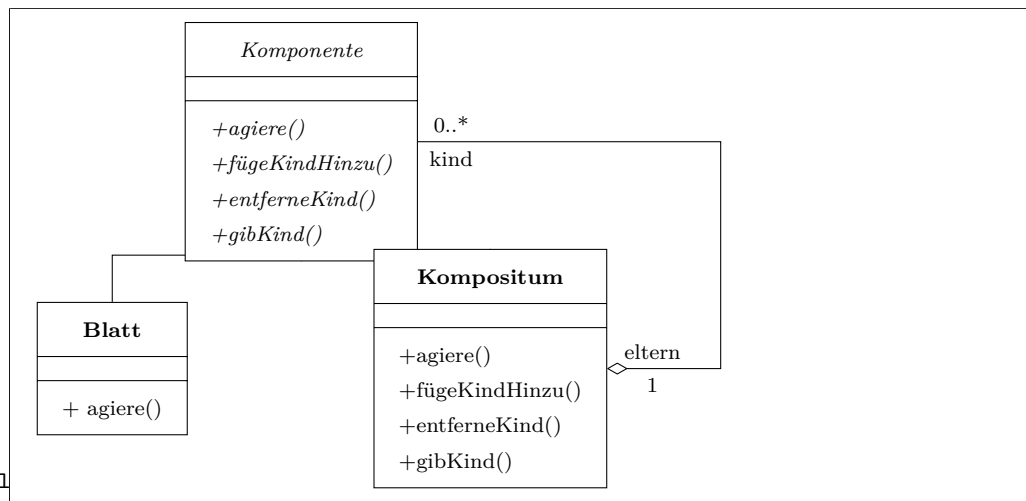
961 \def\liEntwurfsFabrikmethodeAkteure{
962   \begin{description}
963     \item[Produkt]
964
965     Das Produkt ist der Basistyp (Klasse oder Schnittstelle) für das
966     zu erzeugende Produkt.
967
968     \item[KonkretesProdukt]
969
970     KonkretesProdukt implementiert die Produkt-Schnittstelle.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

\liEntwurfsFabrikmethode

```

986 \def\liEntwurfsFabrikmethode{
987   \liEntwurfsFabrikmethodeUml
988   \liEntwurfsFabrikmethodeAkteure
989 }
```

2.10.11 Kompositum (Composite)



\liEntwurfsKompositumUml

```

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

```

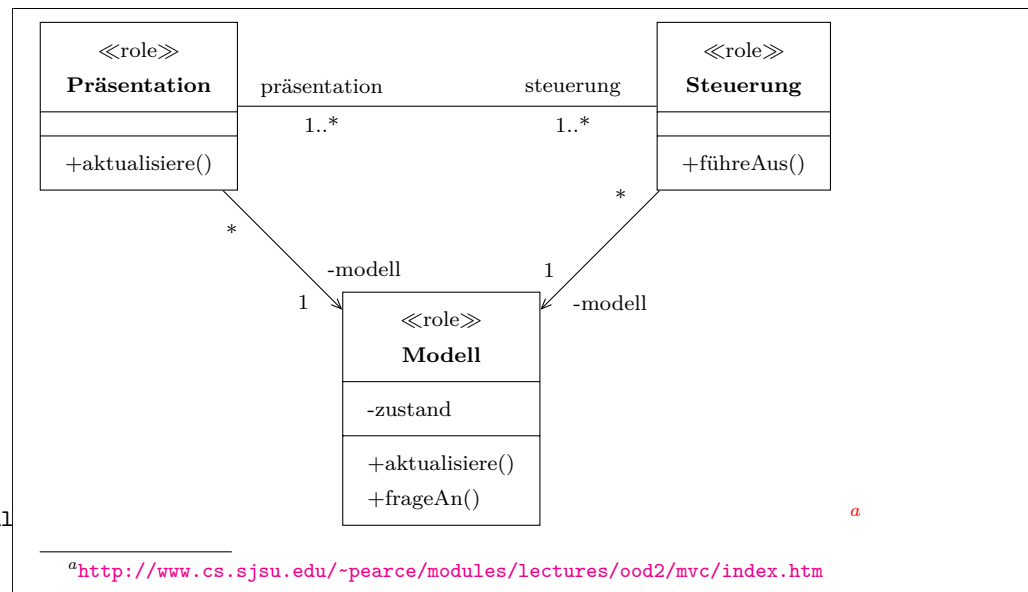
\liEntwurfsFabrikmethode

```

1011 \def\liEntwurfsKompositum{
1012   \liEntwurfsKompositumUml
1013   \liEntwurfsKompositumAkteure
1014 }

```

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



```

1015 \def\liEntwurfsModellPraesentationSteuerungUml{
1016   \begin{tikzpicture}
1017     \umlclass[x=-4cm,type=role]{Präsentation}{-}{+aktualisiere()}
1018     \umlclass[x=4cm,type=role]{Steuerung}{-}{+führeAus()}
1019     \umlclass[y=-4cm,type=role]{Modell}{-}{
1020       -zustand
1021     }{
1022       +aktualisiere()\
1023       +frageAn()
1024     }
1025
1026     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Präsentation}{Modell}
1027     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Steuerung}{Modell}

```

```

1028 \umlassoc[arg1=präsentation,mult1=1..*,arg2=steuerung,mult2=1..*]{Präsentation}{Steuerung}
1029 \end{tikzpicture}
1030 \liFussnoteUrl{http://www.cs.sjsu.edu/~pearce/modules/lectures/ood2/mvc/index.htm}
1031 }

```

ModellPraesentationSteuerung

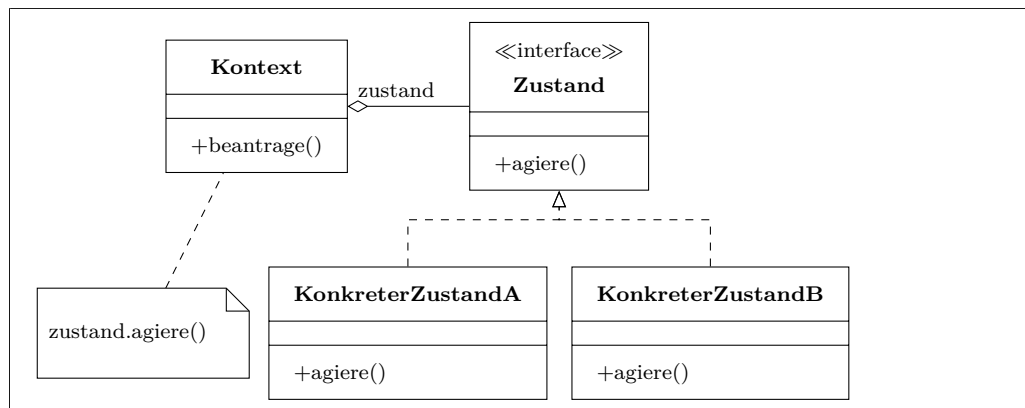
```

1032 \def\liEntwurfs{
1033 \liEntwurfsModellPraesentationSteuerungUml
1034 \liEntwurfsModellPraesentationSteuerungAkteure
1035 }

```

2.10.13 Zustand (State)

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



```

1036 \def\liEntwurfsZustandUml{
1037 \begin{tikzpicture}
1038 \umlcclass[x=-1,y=3]{Kontext}{+beantrage()}
1039 \umlcclass[x=3,y=3,type=interface]{Zustand}{+agiere()}
1040 \umlcclass[x=1,y=0]{KonkreterZustandA}{+agiere()}
1041 \umlcclass[x=5,y=0]{KonkreterZustandB}{+agiere()}
1042
1043 \umlVHVreal{KonkreterZustandA}{Zustand}
1044 \umlVHVreal{KonkreterZustandB}{Zustand}
1045
1046 \umlagg[reg[arg=zustand,pos=0.4]{Kontext}{Zustand}
1047
1048 \umlnote[x=-2.5,y=0,width=2.5cm]{Kontext}{zustand.agiere()}
1049 \end{tikzpicture}
1050 }

```

\liEntwurfsZustandAkteure Quelle: Deutsche Wikipedia

Kontext (Context) definiert die clientseitige Schnittstelle und verwaltet die separaten Zustandsklassen.

State (Zustand) definiert eine einheitliche Schnittstelle aller Zustandsobjekte und implementiert gegebenenfalls ein Standardverhalten.

KontreterZustand (ConcreteState) implementiert das Verhalten, das mit dem Zustand des Kontextobjektes verbunden ist.

```

1051 \def\liEntwurfsZustandAkteure{
1052 \begin{description}
1053 \item[Kontext (Context)]
1054
1055 definiert die clientseitige Schnittstelle und verwaltet die separaten
1056 Zustandsklassen.
1057

```

```

1058 \item[State (Zustand)]
1059
1060 definiert eine einheitliche Schnittstelle aller Zustandsobjekte und
1061 implementiert gegebenenfalls ein Standardverhalten.
1062
1063 \item[KontreterZustand (ConcreteState)]
1064
1065 implementiert das Verhalten, das mit dem Zustand des Kontextobjektes
1066 verbunden ist.
1067 \end{description}
1068 }

\liEntwurfsZustand
1069 \def\liEntwurfsZustand{
1070 \liEntwurfsZustandUml
1071 \liEntwurfsZustandAkteure
1072 }

1073

```

2.11 er.sty

```
1074 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1075 \ProvidesPackage{lehramt-informatik-er}[2020/06/13 Zum Setzen von
1076 ER-Diagrammen]
```

```
1077 \RequirePackage{tikz-er2}
1078 \usetikzlibrary{positioning}
```

2.11.1 Tex-Markup-Beispiel: Komplettes Diagramm

```
\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}

```

2.11.2 Tex-Markup-Beispiel: EER Enhanced Entity-Relation-Modell nach Elmasri/Navante

```

\node[below=1cm of Forscher,circle,draw] (union) {u}
  edge (Sekretär) edge (Forscher) edge (Techniker);

\node[below=1cm of union,entity] (AktiveMitarbeiter) {Aktive Mitarbeiter}
  edge node {$\bigcup$} (union);

```

```
1079 \RequirePackage{soul}
```

```
1080 \RequirePackage{fontawesome}
```

Let-Abkürzungen

```

\let\a=\liErMpAttribute
\let\d=\liErDatenbankName
\let\e=\liErMpEntity
\let\r=\liErMpRelationship

```

```
1081 \ExplSyntaxOn
```

`\liErEntity`

```
1082 \def\liErEntity#1{\textbf{#1}}
```

`\liErRelationship`

```
1083 \def\liErRelationship#1{\ul{#1}}
```

`\liErAttribute`

```
1084 \def\liErAttribute#1{\emph{#1}}
```

`\liErMpEntity` mp = marginpar

Let-Abkürzung: `\let\e=\liErMpEntity`

```

1085 \def\liErMpEntity#1{
1086   \liErEntity{#1}
1087   \marginpar{
1088     \liErEntity{\tiny\faSquareO{}}~E:~#1}
1089   }
1090 }

```

□

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

```

1091 \def\liErMpRelationship#1{
1092   \liErRelationship{#1}
1093   \marginpar{
1094     \liErRelationship{\tiny\faGg{}}~R:~#1}
1095   }
1096 }

```

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

```

1097 \def\liErMpAttribute#1{
1098   \liErAttribute{#1}
1099   \marginpar{
1100     \liErAttribute{\tiny\faCircleThin{}}~A:~#1}
1101   }
1102 }

```

```

\liErDatenbankName Let-Abkürzung: \let\d=\liErDatenbankName
                    datenbank name
1103 \def\liErDatenbankName#1{
1104   {
1105     \footnotesize\texttt{(#1)}
1106   }
1107 }

1108 \ExplSyntaxOff
1109

```

2.12 formale-sprachen.sty

```

1110 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1111 \ProvidesPackage{lehramt-informatik-formale-sprachen}[2021/02/21 Hilfsmakros
1112 zum Setzen von mathematischen Formeln bei dem Thema Formale Sprachen]

1113 \directlua{
1114   formale_sprachen = require('lehramt-informatik-formale-sprachen')
1115 }

1116 \RequirePackage{hyperref}

1117 \liLadePakete{mathe,typographie}

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

1118 \def\liMengeOhneMathe#1{\{ #1 \}}
1119 \def\liMenge#1{%
1120   \ifmode%
1121     \liMengeOhneMathe{#1}%
1122   \else%
1123     $\liMengeOhneMathe{#1}$%
1124   \fi%
1125 }

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

1126 \def\liEpsilon{$\varepsilon$}

\liPotenzmenge Umgeben mit geschweiften Klammern in einer Mathematik-Umgebung

1127 \def\erzeuge@tiefgestellt#1{\directlua{formale_sprachen.erzeuge_tiefgestellt('#1')}}
1128 \def\liPotenzmengeOhneMathe#1{\{ \erzeuge@tiefgestellt{#1} \}}
1129 \def\liPotenzmenge#1{$\liPotenzmengeOhneMathe{#1}$}

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

1130 \let\liZustandsmengeOhneMathe=\liPotenzmengeOhneMathe
1131 \let\liZustandsmenge=\liPotenzmenge

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

1132 \def\liUeberfuehrungsFunktionOhneMathe#1{\delta(\erzeuge@tiefgestellt{#1})}
1133 \def\liUeberfuehrungsFunktion#1{
1134   \ifmode
1135     \liUeberfuehrungsFunktionOhneMathe{#1}
1136   \else
1137     $\liUeberfuehrungsFunktionOhneMathe{#1}$
1138   \fi
1139 }

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

1140 \def\liAlphabet#1{$\Sigma = \{ #1 \}$}

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

1141 \def\liBandAlphabet#1{$\Gamma = \Sigma \cup \{ #1 \}$}

\liZustandsBuchstabe

1142 \def\liZustandsBuchstabe{z}

\liZustandsBuchstabeGross

1143 \def\liZustandsBuchstabeGross{Z}

```



```

\liZustandsmengeNr
1144 \def\zustandsnamens@liste#1#2{
1145   $
1146   \{
1147     \directlua{formale_sprachen.erzeuge_zustandsnamens_liste('#1', '#2')}
1148   \}
1149   $
1150 }
1151 \def\liZustandsmengeNr#1{\zustandsnamens@liste{\liZustandsBuchstabe}{#1}}

\liZustandsmengeNrGross
1152 \def\liZustandsmengeNrGross#1{\zustandsnamens@liste{\liZustandsBuchstabeGross}{#1}}

\liZustandsname \liZustandsname{1}: $z_1$
1153 \def\liZustandsname#1{\liZustandsBuchstabe_#1$}

\liZustandsnameGross \liZustandsnameGross{1}: $Z_1$
1154 \def\liZustandsnameGross#1{\liZustandsBuchstabeGross_#1$}

\liAbleitung \liAbleitung{S -> aB -> ab}:  $S \vdash aB \vdash ab$ 
1155 \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}

1156 \NewDocumentEnvironment { liProduktionsRegeln }
1157 { 0{P} +b }
1158 {
1159   \liGeschweifteKlammern{#1}
1160   {
1161     \begin{align*}
1162       \directlua{formale_sprachen.produktions_regeln('#2')}
1163     \end{align*}
1164   }{-0.2cm}{-1.5cm}
1165 } {}

\liProduktionen \liProduktionen{S -> A, A -> a}:  $\{S \rightarrow A, A \rightarrow a\}$ 
1166 \def\liProduktionen#1{
1167   \liMenge{\directlua{formale_sprachen.produktions_regeln('#1', true)}}
1168 }

\liZustandsnameTiefgestellt Automatisch tiefgestellte Nummerierung \z1
Let-Abkürzung: \let\z=\liZustandsnameTiefgestellt
1169 \def\liZustandsnameTiefgestellt#1{
1170   \ifmmode
1171     \liZustandsBuchstabe\sb{#1}
1172   \else
1173     $\liZustandsBuchstabe\sb{#1}$
1174   \fi
1175 }

1176 \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}

```

```

1177 \NewDocumentCommand{ \liAusdruck } { O{L} m m } {
1178   $
1179   \tl_if_empty:nTF {#1} {} {#1 =}
1180   \{
1181     \, #2 \,
1182     |
1183     \, #3 \,
1184   \}$
1185 }
1186 \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

1187 \def\liFlaci#1{%
1188   \par
1189   {%
1190     \scriptsize
1191     Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte
1192     Automaten, Compiler und Interpreter) Ein Projekt der Hochschule
1193     Zittau/Görlitz und der Pädagogischen Hochschule Schwyz:
1194     \href{https://flaci.com/#1}{flaci.com/#1}%
1195   }%
1196   \par
1197 }

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

1198 \ExplSyntaxOn
1199 \NewDocumentCommand {\liGrammatik} { O{G} m } {
1200   \tl_set:Nn \l_variablen_tl {V}
1201   \tl_set:Nn \l_alphabet_tl {\Sigma}
1202   \tl_set:Nn \l_produktionen_tl {P}
1203   \tl_set:Nn \l_start_tl {S}
1204
1205   \keys_define:nn { grammatik } {
1206     variablen .code:n = {\tl_set:Nn \l_variablen_tl {\liMenge{##1}}},
1207     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
1208     produktionen .code:n = {\tl_set:Nn \l_produktionen_tl {\liProduktionen{##1}}},
1209     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
1210   }
1211
1212   \keys_set:nn { grammatik } { #2 }
1213
1214   $#1 = (
1215     \l_variablen_tl,
1216     \l_alphabet_tl,
1217     \l_produktionen_tl,
1218     \l_start_tl
1219   )$
1220 }
1221 \ExplSyntaxOff
1222

```

2.13 formatierung.sty

```
1223 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1224 \ProvidesPackage{lehramt-informatik-formatierung}[2020/11/27]
```

2.13.1 Schriftarten / Typographie

```
1225 \RequirePackage{mathpazo}
1226 \RequirePackage[no-math]{fontspec}
1227 \setmainfont{texgyrepagella}
1228 \setsansfont{QTAncientOlive}
1229 \RequirePackage{sectsty}
1230 \allsectionsfont{\fontspec{QTAncientOlive}}
```

2.13.2 Farben

```
1231 \RequirePackage{xcolor}
1232 \definecolor{infogray}{rgb}{0.97,0.97,0.97}
```

2.13.3 Überschriften

```
1233 \RequirePackage{titlesec}
1234 \titleformat{\chapter}[display]{\sffamily\bfseries}{\Opt}{\LARGE}
1235 \titlespacing{\chapter}{0pt}{0pt}{*1}
1236 \titleformat{\paragraph}[hang]{\sffamily\normalsize\bfseries}{\theparagraph}{1em}{}
1237 \setcounter{secnumdepth}{0}
```

2.13.4 Listen

```
1238 \RequirePackage{paralist}
1239 \renewcommand\labelitemi{-}
1240 \renewcommand\labelitemii{-}
1241 \renewcommand\labelitemiii{-}
1242 \renewcommand\labelitemiv{-}
1243 % Counter: enumi enumii enumiii enumiv
1244 % Styles: \arabic{counter} \alph{counter} \Alph{counter} \roman{counter} \Roman{counter}
1245 \renewcommand{\labelenumi}{(\alph{enumi})}
1246 \renewcommand{\labelenumii}{(\roman{enumii})}
```

2.13.5 Kasten

```
1247 \RequirePackage{mdframed}
1248 \mdfsetup{backgroundcolor=infogray}
```

liKasten

```
1249 \NewDocumentEnvironment { liKasten }{ } {
1250   \begin{mdframed}[backgroundcolor=white!0]
1251 } {
1252   \end{mdframed}
1253 }
```

2.13.6 Header

```
1254 \RequirePackage{fancyhdr}
1255 \fancyhead[L,C,R]{}
1256 \fancyfoot[L]{}
1257 \fancyfoot[C]{}
1258 \fancyfoot[R]{\thepage}
1259 \pagestyle{fancy}
1260 \renewcommand{\headrulewidth}{0pt}
1261 \renewcommand{\footrulewidth}{0pt}
1262
```

2.14 gantt.sty

```

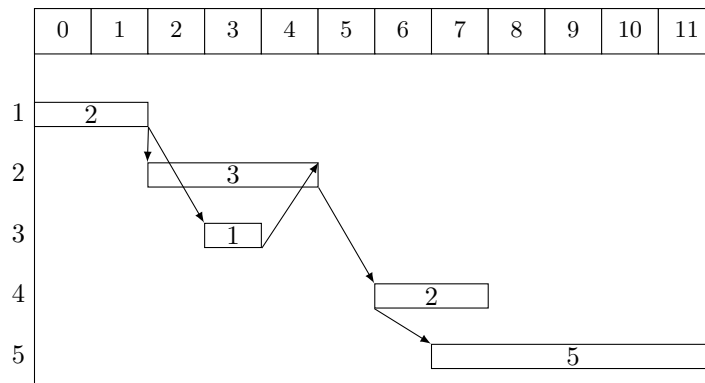
1263 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1264 \ProvidesPackage{lehramt-informatik-gantt}[2020/09/05]

\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}

```



```

1265 \RequirePackage{tikz-uml}
1266 \RequirePackage{pgfgantt}
1267 \setganttlinklabel{f-s}{}
1268 \setganttlinklabel{s-s}{}
1269 \setganttlinklabel{f-f}{}
1270 \setganttlinklabel{s-f}{}

1271

```

2.15 grafik.sty

```
1272 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1273 \ProvidesPackage{lehramt-informatik-grafik}[2020/12/27 Lädt das Paket
1274 TikZ, um Grafiken zeichnen zu können.]
1275 \RequirePackage{tikz}
1276
```

2.16 graph.sty

```

1277 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1278 \ProvidesPackage{lehramt-informatik-graph}[2020/06/09]
1279 \RequirePackage{tikz}

```

Für die die Adjazenzliste (`xrightharrow`)

```

1280 \RequirePackage{amsmath}

```

Für Adjazenz-Matrix

```

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

```

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

```

1281 \RequirePackage{blkarray}
1282 \usetikzlibrary{arrows.meta}

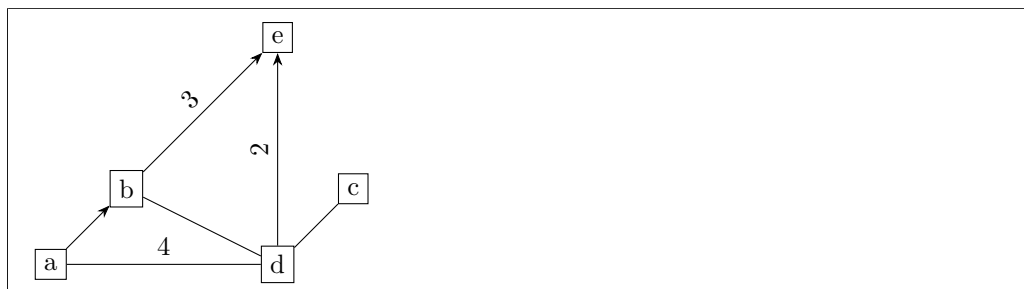
```

```

\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}

```



```

1283 \tikzset{
1284   li graph/.style={
1285     every node/.style={
1286       rectangle,
1287       draw,
1288     },
1289     every edge/.style={
1290       >={Stealth[black]},
1291       draw,
1292     },
1293     every edge/.append style={
1294       every node/.style={
1295         sloped,
1296         auto,
1297       }
1298     }
1299   },
1300   li markierung/.style={
1301     ultra thick,
1302   }
1303 }

```

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}

```

```

1304 \NewDocumentEnvironment { liGraphenFormat }{ +b } {} {}

```

```

1305

```

2.17 hanoi.sty

```
1306 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1307 \ProvidesPackage{lehramt-informatik-hanoi}[2020/12/19 Paket zum Setzen
1308 von Hanoi-Grafiken]
```

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

```
1309 \RequirePackage{tikz}
1310 \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}

1311 \def\li@mset #1[#2]=#3{%
1312 \expandafter\xdef\csname #1#2\endcsname{#3}
1313 }
1314 \def\li@mget #1[#2]{%
1315 \csname #1#2\endcsname
1316 }
1317 \def\li@minc #1[#2]+=#3{%
1318 \pgfmthparse{\li@mget #1[#2]+#3}%
1319 \li@mset #1[#2]=\pgfmthresult
1320 }
1321
1322 \def\liHanoi#1#2{
1323   \edef\li@numdiscs{#1}
1324   \def\li@sequence{#2}
1325   \begin{tikzpicture}[line width=4mm,brown!40,line cap=round,xscale=3]
1326     % init colors
1327     \foreach[count=\j] \c in {red,green!80!black,blue,orange,violet,gray,yellow!80!black,purp
1328     \li@mset col[\j]={\c};
1329     % draw poles and init pole counters
1330     \foreach \j in {1,2,3}{
1331       \li@mset pos[\j]=0
1332       \draw (\j,-.5) -- +(0, .5 + .5*\li@numdiscs);
1333     }
1334     % draw base
1335     \draw (.5,-.5) -- +(3,0);
1336     % draw discs
1337     \foreach[count=\k] \i/\j in \li@sequence{
1338       \draw[draw={\li@mget col[\i]}] (\j,\li@mget pos[\j]) +(-.4*\i/\li@numdiscs,0) -- +(.4*
1339       \li@minc pos[\j]+=.5}
1340     }
1341   \end{tikzpicture}
1342 }

1343
```


2.18 klassen-konfiguration-aufgabe.sty

```
1344 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1345 \ProvidesPackage{lehramt-informatik-klassen-konfiguration-aufgabe}[2021/09/01 Die
1346 Klasse Aufgabe konfigurieren, d. h. Laden von einigen Paketen, Registrierung von Hooks]
1347 \liLadePakete{mathe}
1348 \ExplSyntaxOn
1349 \AddToHook{enddocument}{
1350 \_gib_github_url:
1351 }
1352 \ExplSyntaxOff
1353
```

2.19 komplexitaetstheorie.sty

```

1354 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1355 \ProvidesPackage{lehramt-informatik-komplexitaetstheorie}[2021/07/08 Zum
1356 Setzen von Karp's NP-vollständigen Problemen, Hilfsmakros für die
1357 Polynomialzeitreduktion.]

```

Let-Abkürzungen

```

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

```

```

1358 \liLadePakete{mathe}

```

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

```

1359 \RequirePackage{mdframed}

```

```

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

```

```

1360 \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`

```

1361 \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`

```

1362 \def\liProblemBeschreibung#1#2#3{
1363   \begin{mdframed}[
1364     userdefinedwidth=9cm,
1365     align=center,
1366     backgroundcolor=white!0,
1367   ]
1368     \centerline{\large\liProblemName{#1}}
1369
1370     \medskip
1371
1372     \begin{description}
1373       \item[Gegeben:] #2
1374       \item[Frage:] #3
1375     \end{description}
1376   \end{mdframed}
1377 }

```

```

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

1378 \NewDocumentCommand{ \liPolynomiellReduzierbar } { m O{p} m } {
1379 \begin{displaymath}
1380 \liProblemName{#1}
1381 \preceq_{#2}
1382 \liProblemName{#3}
1383 \end{displaymath}
1384 }

\liProblemVertexCover

1385 \def\liProblemClique{%
1386 Das \textbf{Cliquesproblem} fragt nach der Existenz einer Clique der
1387 Mindestgröße  $n$  in einem gegebenen Graphen.
1388 \footcite[Seite 76]{theo:fs:4}
1389 Eine Clique ist eine Teilmenge von Knoten in einem ungerichteten
1390 Graphen, bei der \emph{jedes Knotenpaar durch eine Kante} verbunden ist.
1391 \footcite{wiki:cliquenproblem}
1392 }

\liProblemVertexCover

1393 \def\liProblemVertexCover{%
1394 %
1395 Das \textbf{Knotenüberdeckungsproblem} (\liProblemName{Vertex Cover})
1396 fragt, ob zu einem gegebenen einfachen Graphen und einer natürlichen
1397 Zahl  $k$  eine Knotenüberdeckung der Größe von höchstens  $k$  existiert.
1398 \footcite{wiki:knotenueberdeckung}
1399
1400 Das heißt, ob es eine aus maximal  $k$  Knoten bestehende Teilmenge  $U$ 
1401 der Knotenmenge gibt, so dass jede Kante des Graphen mit
1402 mindestens einem Knoten aus  $U$  verbunden ist.
1403 \footcite[Seite 78]{theo:fs:4}%
1404 }

\liProblemSubsetSum Kein Karp-21-Problem

1405 \def\liProblemSubsetSum{%
1406 Das \textbf{Teilsommenproblem} (\liProblemName{Subset Sum} oder
1407 \liProblemName{SSP}) ist ein spezielles Rucksackproblem.
1408 \footcite{wiki:teilsommenproblem}
1409 Gegeben sei eine Menge von ganzen Zahlen  $I = \{w_1, w_2, \dots, w_n\}$ .
1410 Gesucht ist eine Untermenge, deren Elementsumme maximal, aber nicht
1411 größer als eine gegebene obere Schranke  $c$  ist.
1412 \footcite[Seite 74]{theo:fs:4}
1413 }

\liProblemSubsetSum Kein Karp-21-Problem

1414 \def\liProblemSat{%
1415 Das \textbf{Erfüllbarkeitsproblem der Aussagenlogik} \liProblemName{Sat}
1416 und \liProblemName{k-SAT} mit  $k \geq 3$ ,  $k \in \mathbb{N}$  (Satz von
1417 Cook) fragt, ob eine aussagenlogische Formel erfüllbar
1418 ist.\footcite{wiki:sat} Das Erfüllbarkeitsproblem der
1419 \emph{Aussagenlogik} ist in exponentieller Zeit in Abhängigkeit der
1420 Anzahl der Variablen mit Hilfe einer Wahrheitstabelle entscheidbar.
1421 Diese \emph{Wahrheitstabelle} kann nicht in polynomieller Zeit
1422 aufgestellt werden.
1423 \footcite[Seite 71]{theo:fs:4}
1424 }

1425

```

2.20 kontrollflussgraph.sty

```
1426 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1427 \ProvidesPackage{lehramt-informatik-kontrollflussgraph}[2020/11/07]

1428 \RequirePackage{tikz}
1429 \usetikzlibrary{positioning}
1430 \tikzset{
1431   li kontrollfluss/.style={
1432     knoten/.style={
1433       circle,
1434       draw
1435     },
1436     usebox/.style={
1437       draw,
1438       rectangle,
1439       font=\scriptsize,
1440       anchor=west,
1441       align=left,
1442     },
1443     bedingung/.style={
1444       midway,
1445       draw=none,
1446       font=\scriptsize
1447     },
1448     knotenbeschriftung/.style={
1449       draw,
1450       rectangle,
1451       midway,
1452       font=\scriptsize
1453     },
1454     wahr/.style={
1455       thick
1456     },
1457     falsch/.style={
1458       dashed
1459     },
1460     every node/.style={
1461       circle,
1462       draw,
1463     },
1464     every edge/.append style={
1465       every node/.style={
1466         draw=none,
1467         bedingung,
1468       }
1469     },
1470     every path/.style={
1471       draw,
1472       ->,
1473     },
1474     every pin/.style={
1475       draw,
1476       dotted,
1477       rectangle,
1478       pin position=right
1479     },
1480     every pin edge/.style={
1481       dotted,
1482       arrows=-,
1483     }
1484   }
1485 }
```

liKontrollflussgraph

```

1486 \NewDocumentEnvironment { liKontrollflussgraph } { 0{ } } {
1487   \begin{tikzpicture}[
1488     li kontrollfluss,
1489     #1
1490   ]
1491 } {
1492   \end{tikzpicture}
1493 }

\liAnweisung
1494 \def\liAnweisung#1(#2,#3){\node[knoten] at (#2,#3) (#1) {#1};}

\liBedingung Let-Abkürzung: \let\b=\liBedingung
1495 \def\liBedingung#1#2{node[bedingung,#1]{\texttt{#2}}}}

\liBedingungWahr Let-Abkürzung: \let\bWahr=\liBedingungWahr
1496 \def\liBedingungWahr#1{node[bedingung,#1]{\texttt{[true]}}}}

\liBedingungFalsch Let-Abkürzung: \let\bFalsch=\liBedingungFalsch
1497 \def\liBedingungFalsch#1{node[bedingung,#1]{\texttt{[false]}}}}

\liKontrollCode Let-Abkürzung: \let\c=\liKontrollCode
1498 \def\liKontrollCode#1{{\tiny\texttt{\textcolor{gray}{#1}}}}

\liKontrollTextzeileKnoten Let-Abkürzung: \let\k=\liKontrollTextzeileKnoten
1499 \def\liKontrollTextzeileKnoten#1{\raisebox{-2pt}{\tikz[scale=0.5,transform shape] \node[draw

\liKontrollKnotenPfad Let-Abkürzung: \let\p=\liKontrollKnotenPfad
1500 \ExplSyntaxOn
1501 \NewDocumentCommand { \liKontrollKnotenPfad }{ m }
1502 {
1503   \seq_set_split:Nnn \l_tmpa_seq { - } { #1 }
1504   \seq_set_map:Nnn \l_tmpa_seq \l_tmpa_seq {\liKontrollTextzeileKnoten{##1}}
1505   \seq_use:Nn \l_tmpa_seq { ~~~~ }
1506 }
1507 \ExplSyntaxOff

1508

```

2.21 kopf-fusszeilen.sty

```
1509 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1510 \ProvidesPackage{lehramt-informatik-kopf-fusszeilen}[2021/08/20 Kopf-
1511 und Fußzeilen mit Hilfe des Pakets fancyhdr.]

1512 \ExplSyntaxOn

1513 \fancyhead{}
1514 \fancyhead[R0,LE]{\scriptsize\LehramtInformatikTitel}
1515 \fancyhead[L0,RE]{\scriptsize\today}
1516 \fancyfoot{}
1517 \fancyfoot[LE,R0]{\thepage}
1518 \fancyfoot[L0,CE]{\scriptsize\LehramtInformatikAutorName}
1519 \fancyfoot[CO,RE]{\scriptsize\LehramtInformatikAutorEmail}
1520 \renewcommand{\headrulewidth}{0.4pt}
1521 \renewcommand{\footrulewidth}{0.4pt}

1522 \ExplSyntaxOff

1523
```

2.22 literatur-dummy.sty

```
1524 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1525 \ProvidesPackage{lehramt-informatik-literatur-dummy}[2020/11/27]

\literatur

1526 \def\literatur{}

\footcite

1527 % \RequirePackage[stable,multiple]{footmisc}
1528 \NewDocumentCommand{ \footcite } { o m }{}

1529
```

2.23 literatur.sty

```
1530 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1531 \ProvidesPackage{lehramt-informatik-literatur}[2020/11/27]

1532 \RequirePackage{csquotes}
1533 \RequirePackage[
1534   bibencoding=utf8,
1535   citestyle=authortitle,
1536   backend=biber,
1537 ]{biblatex}
1538 \addbibresource{\LehramtInformatikRepository/Quellen/10_DB.bib}
1539 \addbibresource{\LehramtInformatikRepository/Quellen/20_OOMUP.bib}
1540 \addbibresource{\LehramtInformatikRepository/Quellen/30_AUD.bib}
1541 \addbibresource{\LehramtInformatikRepository/Quellen/40_SOSY.bib}
1542 \addbibresource{\LehramtInformatikRepository/Quellen/60_FUMUP.bib}
1543 \addbibresource{\LehramtInformatikRepository/Quellen/70_THEO.bib}
1544 \addbibresource{\LehramtInformatikRepository/Quellen/80_DDI.bib}
1545 \addbibresource{\LehramtInformatikRepository/Quellen/Allgemein.bib}
1546 \addbibresource{\LehramtInformatikRepository/Quellen/Examen.bib}
1547 \addbibresource{\LehramtInformatikRepository/Quellen/Wikipedia.bib}
1548 % To allow footnotes in the heading
1549 \RequirePackage[stable,multiple]{footmisc}

\literatur

1550 \def\literatur{\printbibliography[heading=subbibliography]}

1551
```


2.24 makros.sty

```

1552 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1553 \ProvidesPackage{lehramt-informatik-makros}[2020/11/27 Eine Sammlung von
1554 häufig verwendeten Makros und Umgebungen, die thematisch zu keinem
1555 anderen Paket passen]
1556 \RequirePackage{hyperref}
1557 \RequirePackage{graphicx}

Für die Umgebung liQuellen benötigt.
1558 \RequirePackage{paralist}

\inhaltsverzeichnis
1559 \def\inhaltsverzeichnis {
1560   \begin{mdframed}
1561     \begin{group}
1562       \let\clearpage\relax
1563       \tableofcontents
1564     \end{group}
1565   \end{mdframed}
1566 }

\memph \mephm (\marginpar and \emph)
1567 \newcommand{\memph}[1]{\emph{#1}\marginpar{\tiny#1}}

\SLASH
1568 \newcommand\SLASH{\char`\\}

\liPseudoUeberschrift Text, der sich wie eine Überschrift verhält.
1569 \newcommand{\liPseudoUeberschrift}[1]{
1570   \bigskip
1571   \noindent
1572   \textsf{\textbf{#1}}
1573   \noindent
1574 }

\liBeschriftung Ähnlich dem Makro \liPseudoUeberschrift{}. Am Ende des Textes wird ein Doppelpunktzeichen angehängt.
1575 \newcommand{\liBeschriftung}[1]{
1576   \par
1577   \noindent
1578   \medskip
1579   \textbf{#1}:
1580   \medskip
1581   \noindent
1582 }

\hinweis
1583 \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.
1584 \NewDocumentEnvironment { liProjektSprache }{ o +b } {} {}

liEinbettung
1585 \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.
1586 \RequirePackage{xparse}
1587 \ExplSyntaxOn

```

```

1588 \NewDocumentEnvironment{ liAntwort } { 0{standard} }
1589 {
1590   \str_case:nn {#1} {
1591     {standard} {
1592       \def\beschriftung{}
1593       \mdfsetup{backgroundcolor=gray!1,linecolor=gray}
1594     }
1595     {richtig} {
1596       \def\beschriftung{richtig}
1597       \mdfsetup{backgroundcolor=green!5,linecolor=green}
1598     }
1599     {falsch} {
1600       \def\beschriftung{falsch}
1601       \mdfsetup{backgroundcolor=red!3,linecolor=red}
1602     }
1603     {muster} {
1604       \def\beschriftung{Musterlösung}
1605       \mdfsetup{backgroundcolor=green!12,linecolor=green}
1606     }
1607   }
1608   \ifx\beschriftung\empty\else
1609     \noindent
1610     \textbf{\beschriftung{:}}
1611   \fi
1612   \begin{mdframed}
1613 }
1614 {\end{mdframed}}

```

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

```

1615 \NewDocumentEnvironment{ liAdditum } { o }
1616 {
1617   \begin{mdframed}[backgroundcolor=yellow!5]
1618     \IfNoValueTF {#1}
1619     { \liPseudoUeberschrift{Additum} }
1620     { \liPseudoUeberschrift{Additum:~#1} }
1621   }
1622 {\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.

```

1623 \NewDocumentEnvironment{ liExkurs }{o +b}{
1624   \vspace{0.2cm}%
1625   \begin{mdframed}[
1626     backgroundcolor=white,
1627     bottomline=false,
1628     innermargin=1cm,
1629     leftline=true,
1630     linecolor=black,
1631     linewidth=0.1cm,
1632     outermargin=1cm,
1633     rightline=false,
1634     topline=false,
1635   ]

```

```

1636 \footnotesize
1637 \noindent%
1638 \textbf{Exkurs:~#1}\par%
1639 \noindent%
1640 #2
1641 \end{mdframed}
1642 \vspace{0.2cm}
1643 }{}

```

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

```

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

```

Weiterführende Literatur:

- Quelle 1
- Quelle 2

```

1644 \cs_new:Npn \listen@punkt #1 {\item #1}
1645 \NewDocumentEnvironment { liQuellen }{ +b }
1646 {
1647   \seq_clear_new:N \l_quellen
1648   \seq_set_split:Nnn \l_quellen {\item} {#1}
1649   \seq_remove_all:Nn \l_quellen {}
1650   \begin{mdframed}[backgroundcolor=white!0]
1651     \footnotesize
1652     \noindent
1653     \textsf{\textbf{Weiterführende~Literatur:}}
1654     \medskip
1655     \begin{compactitem}
1656       \seq_map_function:NN \l_quellen {\listen@punkt}
1657     \end{compactitem}
1658   \end{mdframed}
1659   %
1660   \makeatletter
1661   \par\@afterindentfalse\@afterheading
1662   \makeatother
1663 } {}

```

liLernkartei

```

1664 \NewDocumentEnvironment { liLernkartei }{ m +b }
1665 {
1666   \begin{mdframed}
1667     \footnotesize
1668     \noindent%
1669     \textbf{Lernkarteikarte:~#1}\par%
1670     \noindent%
1671     #2
1672   \end{mdframed}
1673 } {}

```

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

```

1674 \NewDocumentEnvironment { liDiagramm }{ m +b }
1675 {
1676   \begin{mdframed}[backgroundcolor=white!0]
1677     \small
1678     \noindent%
1679     \textit{#1}:
1680     \begin{center}

```

```

1681 #2
1682 \medskip
1683 \end{center}
1684 \end{mdframed}
1685 } {}
1686 \ExplSyntaxOff

\liFussnoteUrl \liFussnoteUrl[\langle zusätzlicher-text \rangle]{\langle url \rangle} \liFussnoteUrl[zusätzlicher Text]{url}:
Eine HTTP-URL als Fußnote setzen.
1687 \NewDocumentCommand{\liFussnoteUrl} { o m } {
1688 \footnote{\url{#2}\IfNoValueTF{#1}{}{ ( #1 )}}
1689 }
1690

\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.
1691 \NewDocumentCommand{\liFussnoteLink} { o m m } {
1692 \footnote{\href{#3}{#2}\IfNoValueTF{#1}{}{ ( #1 )}}
1693 }

\zB
1694 \def\zB{z.\,B. }

\ZB
1695 \def\ZB{Z.\,B. }

\dh
1696 \def\dh{d.\,h. }

1697

```

2.25 master-theorem.sty

1698 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

1699 \ProvidesPackage{lehramt-informatik-master-theorem}[2021/04/13]

Let-Abkürzungen

\let\0=\li0

\let\o=\li0mega

\let\T=\liT

\let\t=\liTheta

\liMasterVariablenDeklaration

{3} % a

{3} % b

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

\liMasterFallRechnung

% 1. Fall

{für $\varepsilon = 4$: \}

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

% 2. Fall

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

% 3. Fall

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

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

\liMasterVariablenDeklaration

{ } % a

{ } % b

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

\liMasterFallRechnung

% 1. Fall

{ }

% 2. Fall

{ }

% 3. Fall

{ }

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

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

1700 \ExplSyntaxOn

1701 \RequirePackage{amsmath}

\liRundeKlammer

1702 \def\liRundeKlammer#1{

1703 \negthinspace \left(#1 \right)

1704 }

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

1705 \def\liThetaOhneMathe#1{

1706 \Theta \liRundeKlammer{#1}

1707 }

1708 \def\liTheta#1{

1709 \ifmmode

1710 \liThetaOhneMathe{#1}

1711 \else

1712 \$\liThetaOhneMathe{#1}\$

1713 \fi

1714 }

```

\liOmega \liOmega{n^2}:  $\Omega(n^2)$ 
1715 \def\liOmegaOhneMathe#1{
1716 \Omega \liRundeKlammer{#1}
1717 }
1718 \def\liOmega#1{
1719 \ifmmode
1720 \liOmegaOhneMathe{#1}
1721 \else
1722 $\liOmegaOhneMathe{#1}$
1723 \fi
1724 }

\liO \liO{n^2}:  $\mathcal{O}(n^2)$ 
1725 \def\liOOhneMathe#1{
1726 \mathcal{O} \liRundeKlammer{#1}
1727 }
1728 \def\liO#1{
1729 \ifmmode
1730 \liOOhneMathe{#1}
1731 \else
1732 $\liOOhneMathe{#1}$
1733 \fi
1734 }

\liT Let-Abkürzung: \let\T=\liT
\liT{16}{2}:  $16 \cdot T\left(\frac{n}{2}\right)$  \liT{}{2}:  $T\left(\frac{n}{2}\right)$ 
1735 \def\liTOhneMathe#1#2{
1736 \tl_if_blank:nTF {#1}
1737 {}
1738 {#1 \cdot }
1739 T
1740 \liRundeKlammer{\frac{n}{#2}}
1741 }
1742 \def\liT#1#2{
1743 \ifmmode
1744 \liTOhneMathe{#1}{#2}
1745 \else
1746 $\liTOhneMathe{#1}{#2}$
1747 \fi
1748 }

\liRekursionsGleichung \liRekursionsGleichung:  $T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$ 
1749 \def\liRekursionsGleichung{
1750 $T(n) = \liT{a}{b} + f(n)$
1751 }

\liBedingungEins \liBedingungEins:  $f(n) \in \mathcal{O}(n^{\log_b a - \varepsilon})$ 
1752 \def\liBedingungEins{
1753 $f(n) \in \liO{n^{\log\sb{b}a - \varepsilon}}$
1754 }

\liBedingungZwei \liBedingungZwei:  $f(n) \in \Theta(n^{\log_b a})$ 
1755 \def\liBedingungZwei{
1756 $f(n) \in \liTheta{n^{\log\sb{b}a}}$
1757 }

\liBedingungDrei \liBedingungDrei:  $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ 
1758 \def\liBedingungDrei{
1759 $f(n) \in \liOmega{n^{\log\sb{b}a + \varepsilon}}$
1760 }

1761 \ExplSyntaxOff

```

\liMasterVariablen

```

1762 \def\liMasterVariablen{
1763   \begin{displaymath}
1764     T(n) = \liT{a}{b} + f(n)
1765   \end{displaymath}
1766
1767   \begin{itemize}
1768     \item[$a = $]
1769     Anzahl der Unterprobleme in der Rekursion
1770
1771     \item[$\textstyle{\frac{1}{b}} = $]
1772     Teil des Originalproblems, welches wiederum durch alle Unterprobleme
1773     repräsentiert wird
1774
1775     \item[$f(n) = $]
1776     Kosten (Aufwand, Nebenkosten), die durch die Division des Problems und
1777     die Kombination der Teillösungen entstehen
1778   \end{itemize}
1779   \footcite{wiki:master-theorem}
1780   \footcite[Seite 19-35 (PDF 11-24)]{aud:fs:2}
1781 }

```

\liMasterFaelle

```

1782 \def\liMasterFaelle{
1783   \begin{description}
1784     \item[1. Fall:]
1785      $T(n) \in \liTheta{n^{\log_{sb{b}}a}}$ 
1786
1787     \hfill falls \liBedingungEins
1788     für  $\varepsilon > 0$ 
1789
1790     \item[2. Fall:]
191      $T(n) \in \liTheta{n^{\log_{sb{b}}a} \cdot \log n}$ 
1792
1793     \hfill falls \liBedingungZwei
1794
1795     \item[3. Fall:]
1796      $T(n) \in \liTheta{f(n)}$ 
1797
1798     \hfill falls \liBedingungDrei
1799     für  $\varepsilon > 0$ 
1800     und ebenfalls für ein  $c$  mit  $0 < c < 1$  und alle hinreichend großen  $n$ 
1801     gilt:
1802      $a \cdot f(\textstyle{\frac{n}{b}}) \leq c \cdot f(n)$ 
1803   \end{description}
1804 }

```

\liMasterVariablenDeklaration

```

1805 \def\liMasterVariablenDeklaration#1#2#3{
1806   \begin{description}
1807     \item[Allgemeine Rekursionsgleichung:] \strut
1808
1809     \liRekursionsGleichung
1810
1811     \item[Anzahl der rekursiven Aufrufe ( $a$ ):] \strut
1812
1813     #1
1814
1815     \item[Anteil Verkleinerung des Problems ( $b$ ):] \strut
1816
1817     um  $\frac{1}{#2}$  also  $b = #2$ 
1818
1819     \item[Laufzeit der rekursiven Funktion ( $f(n)$ ):] \strut

```

```

1820
1821     $#3$
1822
1823     \item[Ergibt folgende Rekursionsgleichung:] \strut
1824
1825     $T(n) = \liT{\#1}{\#2} + \#3$
1826 \end{description}
1827 }

\liMasterFallRechnung
1828 \def\liMasterFallRechnung#1#2#3{
1829 \begin{description}
1830 \item[1. Fall:] \liBedingungEins:
1831
1832     #1
1833
1834 \item[2. Fall:] \liBedingungZwei:
1835
1836     #2
1837
1838 \item[3. Fall:] \liBedingungDrei:
1839
1840     #3
1841 \end{description}
1842 }

\liMasterExkurs
1843 \def\liMasterExkurs{
1844 \begin{liExkurs}[Master-Theorem]
1845 \liMasterVariablen
1846
1847 \noindent
1848 Dann gilt:
1849
1850 \liMasterFaelle
1851 \end{liExkurs}
1852 }

\liMasterWolframLink Link zu Wolframalpha (+ durch Prozent 2B ersetzen)
1853 \def\liMasterWolframLink#1{
1854 Berechne die Rekursionsgleichung auf WolframAlpha:
1855 \href{https://www.wolframalpha.com/input/?i=#1}{WolframAlpha}
1856 }

1857

```


2.26 mathe.sty

```
1858 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1859 \ProvidesPackage{lehramt-informatik-mathe}[2020/06/10]
1860
1861 % for example \ltimes \rtimes
1862 %\RequirePackage{amssymb}
1863 \RequirePackage{amsmath}
1864
1865 %%
1866 % \mlq \mrq
1867 %%
1868 \DeclareMathSymbol{\mlq}{\mathord}{operators}{``}
1869 \DeclareMathSymbol{\mrq}{\mathord}{operators}{`}
1870
```

2.27 minimierung.sty

```

1871 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1872 \ProvidesPackage{lehramt-informatik-mathe}[2021/03/13 Für den
1873 Minimierungsalgorithmus von einem NEA zu einem DEA]

1874 \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 \\ \hline
\z1 & & & \l & & \l & & \l & & \l & & \l & \l \\ \hline
\z2 & & & & & \l & & \l & & \l & & \l & \l \\ \hline
\z3 & & & & & & & \l & & \l & & \l & \l \\ \hline
\z4 & & & & & & & & & \l & & \l & \l \\ \hline
\z5 & & & & & & & & & & & \l & \l \\ \hline
\z6 & & & & & & & & & & & & \l & \l \\ \hline
\z7 & & & & & & & & & & & & & \l & \l \\ \hline
\z8 & & & & & & & & & & & & & & \l & \l \\ \hline \hline
& \z0 & \z1 & \z2 & \z3 & \z4 & \z5 & \z6 & \z7 & \z8 & \l & \l & \l & \l \\ \hline
\end{tabular}

\liFussnoten

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

\liFussnote

1875 \def\liFussnote#1{\$x_{#1}\$}

1876 \def\li@fussnote@text#1#2{
1877 \liFussnote{#1}
1878 \quad
1879 {\footnotesize #2}
1880 }

\liFussnoteEinsText
1881 \def\liFussnoteEinsText{
1882 \li@fussnote@text{1}
1883 {Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.}
1884 }

\liFussnoteZweiText
1885 \def\liFussnoteZweiText{
1886 \li@fussnote@text{2}
1887 {Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt.}
1888 }

\liFussnoteDreiText
1889 \def\liFussnoteDreiText{
1890 \li@fussnote@text{3}

```

```

1891 {In weiteren Iterationen markierte Zustände.}
1892 }

\liFussnoteVierText
1893 \def\liFussnoteVierText{
1894   \li@fussnote@text{4}
1895   {...}
1896 }

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



1897 \def\liFussnoten{
1898   \bigskip
1899
1900   \noindent
1901   \liFussnoteEinsText
1902
1903   \noindent
1904   \liFussnoteZweiText
1905
1906   \noindent
1907   \liFussnoteDreiText
1908
1909   \noindent
1910   \liFussnoteVierText
1911 }

\liLeereZelle \liLeereZelle:  $\emptyset$ 
Let-Abkürzung: \let\l=\liLeereZelle
1912 \def\liLeereZelle{\$\emptyset\$}

\liZustandsPaarVariablenName
1913 \def\liZustandsPaarVariablenName{z}

\liZustandsPaar
1914 \def\liZustandsPaar#1#2{
1915   $(
1916     \liZustandsPaarVariablenName_#1,
1917     \liZustandsPaarVariablenName_#2
1918   )$
1919 }

liUebergangsTabelle
1920 \renewcommand{\arraystretch}{1.4}
1921 \NewDocumentEnvironment{ liUebergangsTabelle } { m m } {
1922   \liPseudoUeberschrift{Übergangstabelle}
1923   \begin{center}
1924     \begin{tabular}{r|l|l}
1925       \textbf{Zustandspaar} & \textbf{\#1} & \textbf{\#2} \\ \hline
1926     \end{tabular}
1927   \end{center}
1928 }
1929 }

liUeberschriftDreiecksTabelle \liUeberschriftDreiecksTabelle: Minimierungstabelle (Table filling)
1930 \ExplSyntaxOn
1931 \def\liUeberschriftDreiecksTabelle{
1932   \liPseudoUeberschrift{Minimierungstabelle~(Table~filling)}
1933 }

```

\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. —————

```

1934 \def\liMinimierungErklaerung{
1935   %\footcite[Seite~19]{koenig}
1936   \liParagraphMitLinien{
1937     Der~Minimierungs-Algorithmus~(auch~Table-Filling-Algorithmus-genannt)~
1938     trägt~in~seinem~Verlauf~eine~Markierung~in~alle~
1939     diejenigen~Zellen~der~Tabelle~ein,~die~zueinander~nicht~äquivalente~
1940     Zustände~bezeichnen.~Die~Markierung~„ $x_{\sb{n}}$ “~in~einer~Tabellenzelle~( $i$ ,~
1941      $j$ )~bedeutet~dabei,~dass~das~Zustandspaar~( $i$ ,~ $j$ )~in~der~ $k$ -ten~
1942     Iteration~des~Algorithmus~markiert~wurde~und~die~Zustände~ $i$ ~und~ $j$ ~
1943     somit~zueinander~( $k - 1$ )~äquivalent,~aber~nicht~ $k$ -äquivalent~und~
1944     somit~insbesondere~nicht~äquivalent~sind.~Bleibt~eine~Zelle~bis~zum~Ende~
1945     unmarkiert,~sind~die~entsprechenden~Zustände~zueinander~äquivalent.
1946   }
1947 }
1948 \ExplSyntaxOff

```

1949

2.28 normalformen.sty

```
1950 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1951 \ProvidesPackage{lehramt-informatik-normalformen}[2020/12/10
1952 Hilfsmakros zum Setzen von Normalformen, Funktionalen Abhängigkeiten,
1953 Attributhülle]
```

Mathe für textit tex etc, typographie für geschweifte Klammer

```
1954 \liLadePakete{mathe,typographie}
1955 \directlua{
1956   helper = require('lehramt-informatik-helfer')
1957   normalformen = require('lehramt-informatik-normalformen')
1958 }
```

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
```

```
1959 \def\liTeilen#1{
1960   \directlua{tex.print(normalformen.teilen('#1'))}
1961 }
```

```
\liAttributHuelle Let-Abkürzung: \let\ah=\liAttributHuelle
                  \ah{F, \m{A, B}} AttrHülle( $F, \{A, B\}$ ) Regulärer Ausdruck zum Konvertieren
                  AttrHülle((.*)\)\ \ah{$1}
1962 \def\liAttributHuelleOhneMathe#1{\text{AttrHülle}(\#1)}
1963 \def\liAttributHuelle#1{
1964   \ifmmode
1965     \liAttributHuelleOhneMathe{\#1}
1966   \else
1967     $\liAttributHuelleOhneMathe{\#1}$
1968   \fi
1969 }
```

```
\liAttributMenge Let-Abkürzung: \let\m=\liAttributMenge
1970 \def\liAttributMenge#1{\{ \textit{\#1} \}}
```

liAHuelle

```
1971 \NewDocumentEnvironment{ liAHuelle } { +b } {
1972   \begin{group}
1973   \footnotesize
1974   \begin{multline*}
1975     \#1
1976   \end{multline*}
1977   \end{group}
1978 } { }
```

```
\liLinksReduktion Nur innerhalb von liAHuelle zu verwenden bzw. multline
Let-Abkürzung: \let\ahL=\liLinksReduktion
                  \ahL{ursprüngliche linke Attributmenge}{ohne dieses Attribut}{Ergebnis}
1979 \def\liLinksReduktion#1#2#3{
1980   \shoveleft{
1981     \liAttributHuelleOhneMathe{FA,
1982       \liAttributMenge{\#1 \string\ #2}} =
1983     } \}
```

```

1984 \shoveright{
1985   \liAttributMenge{#3}
1986 } \\\
1987 }

```

```

\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( $F, \{A, E \setminus E\}$ ) =  $\{A, E, F, B, D\}$ 
1988 \def\liLinksReduktionInline#1#2#3{%
1989   {%
1990     \footnotesize%
1991     $\liAttributHuelleOhneMathe{F,
1992       \liAttributMenge{#1} \string\ #2}} =
1993     \liAttributMenge{#3}$
1994   }
1995 }

```

```

\liLinksReduktionInline Let-Abkürzung: \let\ahr=\liLinksReduktionInline
    \ahr{gelöschte FA}{neue FA ohne rechts Attribut}{gegebene Attribute}{Ergebnis}
1996 \def\liRechtsReduktionInline#1#2#3#4{%
1997   {%
1998     \footnotesize%
1999     $\liAttributHuelleOhneMathe{
2000       F \setminus
2001       \liFunktionaleAbhaengigkeit{#1}
2002       \def\tmp{#2}\ifx\tmp\empty
2003       \else
2004         \cup \liFunktionaleAbhaengigkeit{#2}
2005       \fi
2006       ,
2007       \liAttributMenge{#3}
2008     } =
2009     \liAttributMenge{#4}$
2010   }
2011 }

```

```

\liFunktionaleAbhaengigkeit Let-Abkürzung: \let\fa=\liFunktionaleAbhaengigkeit
    \fa{A, B -> C, D}:  $\{A, B\} \rightarrow \{C, D\}$  Regulärer Ausdruck zum Konvertieren:
    \$(.*?) \\\rightarrow (.*)\$ \fa{$1 -> $2}
2012 \def\liFunktionaleAbhaengigkeit#1{%
2013   \directlua{normalformen.drucke_funk_abhaengigkeit('#1', false)}%
2014 }

```

```

\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;

```

```

2015 \NewDocumentCommand {\liFunktionaleAbhaengigkeiten} { O{FA} m } {
2016   \liGeschweifteKlammern
2017   {#1}
2018   {
2019     \begin{align*}
2020       \directlua{normalformen.drucke_funk_abhaengigkeiten('#2')}
2021     \end{align*}
2022   }
2023   {-0.5cm}
2024   {-1.7cm}
2025 }

\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}

2026 \NewDocumentCommand {\liRelation} { O{R} m } {
2027   $\directlua{
2028     local name = helper.konvertiere_tiefgestellt('#1')
2029     tex.print(name)
2030   }$(\textit{\, #2\,})
2031 }

2032

```

2.29 petri.sty

2033 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2034 \ProvidesPackage{lehramt-informatik-petri}[2020/12/03]

Let-Abkürzungen

\let\t=\liPetriTransitionsName

\let\tp=\liPetriTransPfeile

\let\k=\liPetriErreichKnotenDrei

2035 \RequirePackage{tikz}

2036 \usetikzlibrary{petri,arrows.meta}

Für die Darstellungsmatrix

2037 \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

```
2038 \def\liPetriSetzeSchluessel{%
2039   \def\TmpTransitionOne{}%
2040   \def\TmpTransitionTwo{}%
2041   \def\TmpTransitionThree{}%
2042   \def\TmpTransitionFour{}%
2043   \def\TmpTransitionFive{}%
2044   \def\TmpTransitionSix{}%
2045   \def\TmpTransitionSeven{}%
2046   \def\TmpTransitionEight{}%
2047   \def\TmpTransitionNine{}%
2048   \def\TmpTransitionTen{}%
2049   \pgfkeys{/petri/.cd,
2050     p1/.store in=\TmpPlaceOne,p1/.default=0,p1,
2051     p2/.store in=\TmpPlaceTwo,p2/.default=0,p2,
2052     p3/.store in=\TmpPlaceThree,p3/.default=0,p3,
2053     p4/.store in=\TmpPlaceFour,p4/.default=0,p4,
2054     p5/.store in=\TmpPlaceFive,p5/.default=0,p5,
```



```

2055     p6/.store in=\TmpPlaceSix,p6/.default=0,p6,
2056     p7/.store in=\TmpPlaceSeven,p7/.default=0,p7,
2057     p8/.store in=\TmpPlaceEight,p8/.default=0,p8,
2058     p9/.store in=\TmpPlaceNine,p9/.default=0,p9,
2059     p10/.store in=\TmpPlaceTen,p10/.default=0,p10,
2060     t1/.store in=\TmpTransitionOne,t1/.default=activated,
2061     t2/.store in=\TmpTransitionTwo,t2/.default=activated,
2062     t3/.store in=\TmpTransitionThree,t3/.default=activated,
2063     t4/.store in=\TmpTransitionFour,t4/.default=activated,
2064     t5/.store in=\TmpTransitionFive,t5/.default=activated,
2065     t6/.store in=\TmpTransitionSix,t6/.default=activated,
2066     t7/.store in=\TmpTransitionSeven,t7/.default=activated,
2067     t8/.store in=\TmpTransitionEight,t8/.default=activated,
2068     t9/.store in=\TmpTransitionNine,t9/.default=activated,
2069     t10/.store in=\TmpTransitionTen,t10/.default=activated,
2070     scale/.store in=\TmpScale,scale/.default=0.5,
2071     x/.store in=\TmpX,x/.default=5,
2072     y/.store in=\TmpY,y/.default=5,
2073   }%
2074 }

2075 \tikzset{
2076   li petri/.style={
2077     activated/.style={
2078       very thick
2079     },
2080     inhibitor/.style={
2081       {Circle[open,length=2mm,fill=white]}-
2082     }
2083   }
2084 }

```

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

```

2085 \def\liPetriTransitionsNameOhneMathe#1{t\sb{#1}}
2086 \def\liPetriTransitionsName#1{
2087   \ifmmode
2088     \liPetriTransitionsNameOhneMathe{#1}
2089   \else
2090     $\liPetriTransitionsNameOhneMathe{#1}$
2091   \fi
2092 }

```

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

```

2093 \NewDocumentCommand{ \liPetriErreichTransition } { m m m O{} O{} } {
2094   \draw[->] (#1) edge[#4] node[pos=0.5,auto,sloped,#5]{t\sb{#3}$} (#2);
2095 }

```

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

```

2096 \def\liPetriErreichKnotenDrei#1#2#3{(#1,#2,#3)}

```

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

```

2097 \def\liPetriTransPfeile#1{ $\rightarrow \hspace{0.4cm} \liPetriTransitionsName{#1} \hspace{0.3cm}
2098 }

```

2.30 potenzmengen-konstruktion.sty

```
2099 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2100 \ProvidesPackage{lehramt-informatik-potenzmengen-konstruktion}[2021/02/21 Hilfsmakros
2101 einzusetzen bei der Potenzmengen-Konstruktion (Potenzmengen-Algorithmus)]
```

```
2102 \liLadePakete{formale-sprachen}
2103 \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}
    }
  }
}
```

```
2104 \def\liZustandsMengenSammlung#1#2{
2105   \liZustandsnameGross{#1}
2106   {
2107     \footnotesize
2108     \liPotenzmenge{
2109       \str_case:nn {#1} {#2
2110       }
2111     }
2112 }
```

```
\liZustandsMengenSammlungNr
```

```
2113 \def\liZustandsMengenSammlungNr#1#2{
2114   \liZustandsnameGross{#1}
2115   {
```

```
2116     \footnotesize
2117     \liZustandsmengeNr{
2118         \str_case:nn {#1} #2
2119     }
2120 }
2121 }

2122 \ExplSyntaxOff
2123
```

2.31 pseudo.sty

```

2124 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2125 \ProvidesPackage{lehramt-informatik-pseudo}[2020/12/30 Paket zum Setzen
2126 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>

```

2127 \RequirePackage[german,boxruled]{algorithm2e}

```

```

2128

```

2.32 pumping-lemma.sty

2129 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2130 \ProvidesPackage{lehramt-informatik-pumping-lemma}[2021/08/11 Enthält die
 2131 Definitionen für das Pumping-Lemma in der Regulären Sprache und
 2132 in der Kontextfreien Sprache]

\liPumpingRegulaer

```
2133 \def\liPumpingRegulaer{%
2134   Es sei  $L$  eine reguläre Sprache. Dann gibt es eine Zahl  $j$ , sodass für
2135   alle Wörter  $\omega \in L$  mit  $|\omega| \geq j$  (jedes Wort  $\omega$  in
2136    $L$  mit Mindestlänge  $j$ ) jeweils eine Zerlegung  $\omega = uvw$ 
2137   existiert, sodass die folgenden Eigenschaften erfüllt sind:
2138
2139   \begin{enumerate}
2140     \item  $|v| \geq 1$ 
2141     (Das Wort  $v$  ist nicht leer.)
2142
2143     \item  $|uv| \leq j$ 
2144     (Die beiden Wörter  $u$  und  $v$  haben zusammen höchstens die Länge  $j$ .)
2145
2146     \item Für alle  $i = 0, 1, 2, \dots$  gilt  $uv^i w \in L$ 
2147     (Für jede natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^i w$  in der
2148     Sprache  $L$ )
2149   \end{enumerate}
2150
2151   Die kleinste Zahl  $j$ , die diese Eigenschaften erfüllt, wird
2152   Pumping-Zahl der Sprache  $L$  genannt.\footcite{wiki:pumping-lemma}
2153 }
```

\liPumpingKontextfrei

```
2154 \def\liPumpingKontextfrei{%
2155   Es sei  $L$  eine kontextfreie Sprache. Dann gibt es eine Zahl  $j$ , sodass
2156   sich alle Wörter  $\omega \in L$  mit  $|\omega| \geq j$  zerlegen lassen in
2157    $\omega = uvwxy$ , sodass die folgenden Eigenschaften erfüllt sind:
2158
2159   \begin{enumerate}
2160     \item  $|vx| \geq 1$ 
2161     (Die Wörter  $v$  und  $x$  sind nicht leer.)
2162
2163     \item  $|vwx| \leq j$ 
2164     (Die Wörter  $v$ ,  $w$  und  $x$  haben zusammen höchstens die Länge  $j$ .)
2165
2166     \item Für alle  $i \in \mathbb{N}_0$  gilt  $uv^iwx^iy \in L$  (Für jede
2167     natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^iwx^iy$  in der
2168     Sprache  $L$ )
2169   \end{enumerate}
2170 }
```

2171

2.33 quicksort.sty

```

2172 % https://tex.stackexchange.com/a/142634
2173 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2174 \ProvidesPackage{lehramt-informatik-quicksort}[2020/06/12]
2175
2176 %-----
2177 % USAGE:
2178 % \QSinitialize{comma, separated, numerical, values}
2179 % \loop
2180 % \QSpivotStep
2181 % \ifnum\value{pivotcount}>0
2182 %   \QSSortStep
2183 % \repeat
2184 %-----
2185
2186 % xintfrac does not load xinttools, this must be done explicitly if needed as here.
2187 \RequirePackage{xintfrac, xinttools}
2188
2189 \RequirePackage{tikz}
2190
2191 %-----
2192 % FIRST PART: TikZ styles and macros for the actual drawing
2193 \newcounter{cellcount}% used for coordinates of the node
2194 \newcounter{pivotcount}% when it will remain at zero, will signal the sort is finished.
2195
2196 % Styles defined by Tom Bombaldi. (modified: all share the same size)
2197 % (re-modified \bf -> \bfseries due to extremely annoying warnings from
2198 % KOMA-script which are truly a pain and do not make any sense regarding \bf:
2199 % if I want to use \bf, and know what I am doing, why should I get HARASSED
2200 % by police of LaTeX good conduct ? )
2201 \tikzset{l/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=lime!70!gray},
2202          o/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=olive!50},
2203          r/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, t
2204 % this is the "b" style as used in the image below
2205 %   b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black,
2206 % nicer:
2207 %   b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=white, text=magenta
2208 %   g/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=gray, text=white,
2209
2210 % NOTE the b style was originally the same as the r(aised) style apart from
2211 % not being raised, but I find it nicer with a somewhat different
2212 % specification. I have not updated the images though.
2213
2214 % How the nodes are drawn depending on whether on the left of the pivot value
2215 % or on the right, or is a pivot value, or a raised pivot during selection phase.
2216
2217 \def\DecoLEFT #1{%
2218   \xintFor* ##1 in {#1} \do
2219     {\stepcounter{cellcount}\node[o] at (\arabic{cellcount},0) {##1};}%
2220 }
2221
2222 \def\DecoINERT #1{%
2223   \xintFor* ##1 in {#1} \do
2224     {\stepcounter{cellcount}\node[g] at (\arabic{cellcount},0) {##1};}%
2225 }
2226
2227 \def\DecoRIGHT #1{%
2228   \xintFor* ##1 in {#1} \do
2229     {\stepcounter{cellcount}\node[l] at (\arabic{cellcount},0) {##1};}%
2230 }
2231
2232 \def\DecoLEFTwithPivot #1{\stepcounter{pivotcount}%
2233   \xintFor* ##1 in {#1} \do

```

```

2234     {\stepcounter{cellcount}}%
2235     \xintifForLast {\node[r]}\node[o]} at (\arabic{cellcount},0) {##1};}%
2236 }
2237
2238 \def\DecoINERTwithPivot #1{\stepcounter{pivotcount}%
2239     \xintFor* ##1 in {#1} \do
2240     {\stepcounter{cellcount}}%
2241     \xintifForLast {\node[b]}\node[g]} at (\arabic{cellcount},0) {##1};}%
2242 }
2243
2244 \def\DecoRIGHTwithPivot #1{\stepcounter{pivotcount}%
2245     \xintFor* ##1 in {#1} \do
2246     {\stepcounter{cellcount}}%
2247     \xintifForLast {\node[r]}\node[l]} at (\arabic{cellcount},0) {##1};}%
2248 }
2249
2250 %-----
2251 % SECOND PART: the actual sorting routines.
2252
2253 \def\QS@sort@a #1{\expandafter \QS@sort@b \expandafter {\xintLength {#1}}{#1}}
2254 \def\QS@sort@b #1{\ifcase #1
2255     \expandafter\QS@sort@empty
2256     \or\expandafter\QS@sort@single
2257     \else\expandafter\QS@sort@c
2258     \fi
2259 }%
2260 \def\QS@sort@empty #1{}
2261 \def\QS@sort@single #1{\QSIr {#1}}
2262
2263 % This step is to pick the last as pivot.
2264 \def\QS@sort@c #1%
2265     {\expandafter\QS@sort@d\expandafter {\romannumeral0\xintnthelt {-1}{#1}}{#1}}%
2266
2267 % Here \QSLr, \QSIr, \QSR have been let to \relax.
2268 % The trick with \xintApplyUnbraced is that for example when selecting
2269 % the elements smaller than pivot, if we had been using \xintApply we
2270 % would have had at the minimum an empty brace pair. Thus we use the
2271 % "unbraced" variant, but then the \QS@select@smaller has added in
2272 % anticipation a level of braces.
2273 \def\QS@sort@d #1#2{%
2274     \QSLr {\xintApplyUnbraced {\QS@select@smaller {#1}}{#2}}}%
2275     \QSIr {\xintApplyUnbraced {\QS@select@equal {#1}}{#2}}}%
2276     \QSRr {\xintApplyUnbraced {\QS@select@greater {#1}}{#2}}}%
2277 }%
2278 \def\QS@select@smaller #1#2{\xintifLt {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2279 \def\QS@select@equal #1#2{\xintifEq {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2280 \def\QS@select@greater #1#2{\xintifGt {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2281
2282 %
2283 % NOTE 1: thus, each comparison with the pivot is done three (!) times.
2284 %
2285 % NOTE 2: we may well end up with \QSLr {<empty>} situations. This is handled
2286 % silently by the \xintFor loops, and also when \QSLr becomes \QS@sort@a, the
2287 % latter must handle correctly an empty argument.
2288
2289 %-----
2290 % THIRD PART: the main macros \QSpivotStep, \QSsortStep and \QSinitialize.
2291
2292 % This draws all with suitable highlighting for the newly chosen pivots
2293 % (which will be shown raised)

```

```

2294 \def\QSpivotStep {\let\QSLr\DecoLEFTwithPivot
2295             \let\QSIr\DecoINERT
2296             \let\QSIrr\DecoINERT
2297             \let\QSRr\DecoRIGHTwithPivot
2298 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2299             \setcounter{cellcount}{0}\setcounter{pivotcount}{0}%
2300             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2301 }
2302
2303 % This sorts and then draws, showing where the pivot chosen in the previous
2304 % step go. Next time they will have become "inert". If pivotcount is still at
2305 % zero on exit from \QSpivotStep, then this is the signal to stop before
2306 % executing \QSSortStep.
2307 \def\QSSortStep {\def\QSLr {\noexpand\QS@sort@a}%
2308             \def\QSRr {\noexpand\QS@sort@a}%
2309             \def\QSIr {\noexpand\QSIrr}%
2310             \let\QSIrr\relax
2311             \edef\QS@list{\QS@list}%
2312             \let\QSLr\relax
2313             \let\QSRr\relax
2314             \let\QSIr\relax
2315             \edef\QS@list{\QS@list}%
2316             \let\QSLr\DecoLEFT
2317             \let\QSIr\DecoINERTwithPivot
2318             \let\QSIrr\DecoINERT
2319             \let\QSRr\DecoRIGHT
2320 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2321             \setcounter{cellcount}{0}%
2322             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2323 }
2324
2325 \def\QSinitialize #1{%
2326     % first, we convert the comma separated values into a list of braced items
2327     % we use an \edef, and anyhow many \edef's will be used later
2328     \edef\QS@list {\noexpand\QSRr {\xintCSVtoList {#1}}}%
2329     \let\QSRr\DecoRIGHT
2330     % The \QSRr marker mutated to draw the last element as
2331     % pivot and the earlier ones with the suitable style.
2332     %
2333     % The list of marked braced items \QS@list is used both for drawing
2334     % (as here) and for doing the exchange of elements during sort.
2335     \par\centerline{\rule[1.5mm]{0pt}{8mm}\setcounter{cellcount}{0}%
2336             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2337 }
2338

```


2.34 relationale-algebra.sty

```
2339 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2340 \ProvidesPackage{lehramt-informatik-relationale-algebra}[2020/12/11]
```

```
2341 \RequirePackage{amsmath}
```

```
2342 \RequirePackage{amssymb}
```

Zum Zeichnen von Operatorenbäumen verwenden wir TikZ

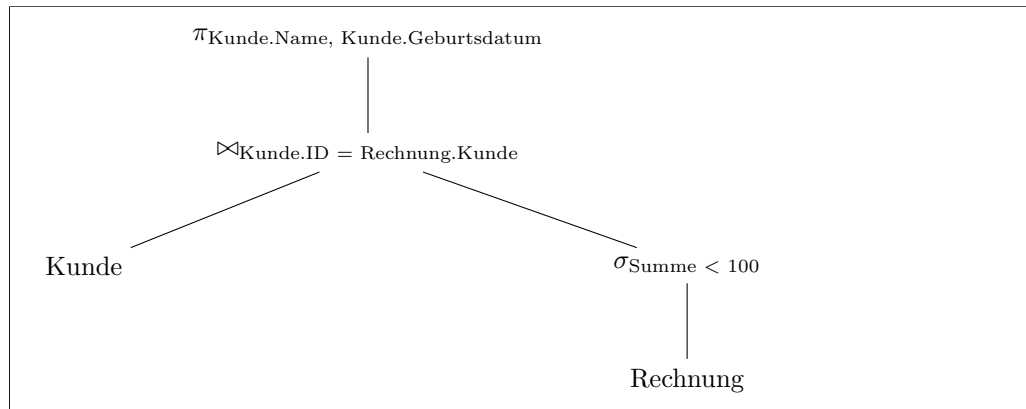
```
\begin{tikzpicture}
  \node
    (pi) {\pi_{\text{Kunde.Name, Kunde.Geburtsdatum}}};

  \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}
```



```
2343 \RequirePackage{tikz}
```

```
2344 \usetikzlibrary{positioning}
```

Privates Makros, das zwei Querstriche erzeugt.

```
2345 \def\o@join{\setbox0=\hbox{\bowtie}%
2346   \rule[-.02ex]{.25em}{.4pt}\llap{\rule[\ht0]{.25em}{.4pt}}%
2347 }
```

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

```
2348 \def\leftouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie}}
```

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

```
2349 \def\rightouterjoin{\mathbin{\bowtie\mkern-5.8mu\o@join}}
```

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

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

```
2351
```

2.35 rmodell.sty

```
2352 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2353 \ProvidesPackage{lehramt-informatik-rmodell}[2020/09/01
2354 Makros und Umgebungen zum Setzen des Relationenmodells beim Thema
2355 Datenbanken.]
2356 \RequirePackage{soul}
```

Let-Abkürzungen

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

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

```
2357 \def\liPrimaer#1{\ul{#1}}
```

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

```
2358 \def\liFremd#1{{\setul{-0.9em}{}\ul{#1}}}
```

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

```
2359 \def\li@Rmodell@Schrift{\footnotesize\ttfamily}
2360 \ExplSyntaxOn
2361 \NewDocumentEnvironment { liRmodell }
2362 { +b }
2363 {
2364   \medskip
2365   {
2366     \linespread{2}
2367     \setlength{\parindent}{0pt}
2368     \li@Rmodell@Schrift#1
2369   }
2370   \medskip
2371 } {}
2372 \ExplSyntaxOff
```

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

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

```
2373 \def\liRelationMenge#1#2{
2374 \noindent
2375 #1 : \[ #2 ]\}
2376 \par
2377 }
```

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

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

```
2378 \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}
```

```
2379 \NewDocumentEnvironment { liRelationenSchemaFormat }{ +b } {} {}
```

```
2380
```

2.36 sortieren.sty

```
2381 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2382 \ProvidesPackage{lehramt-informatik-sortieren}[2020/06/10 Schaubilder
2383 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} \nodepart{five} }
  \liSortierPfeil{one}{two}
  \liSortierPfeil{two}{three}
  \liSortierMarkierung{two split south}{three split north}
  \liSortierPfeilUnten{three}{one}
}
```



```
2384 \RequirePackage{tikz}
2385 \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.

```
2386 \def\liVertauschen#1{
2387   \directlua{
2388     local sortieren = require('lehramt-informatik-sortieren')
2389     sortieren('#1')
2390   }
2391 }
```

`\liSortierPfeil`

```
2392 \def\liSortierPfeil#1#2{
2393   \draw[-latex] ([xshift=1mm]reihe.#1 north) -- ++(0,0.25) -| ([xshift=-1mm]reihe.#2 north);
2394 }
```

`\liSortierPfeilUnten`

```
2395 \def\liSortierPfeilUnten#1#2{
2396   \draw[-latex] ([xshift=1mm]reihe.#1 south) -- ++(0,-0.25) -| ([xshift=-1mm]reihe.#2 south);
2397 }
```

`\liSortierMarkierung`

```
2398 \def\liSortierMarkierung#1#2{\node[
2399   draw,
2400   very thick,
2401   fit=(reihe.#1) (reihe.#2),
2402   inner sep=0pt
2403 ] {}
2404 }

2405 \tikzset{
2406   li sortierung zahlenreihe/.style={
2407     draw,
2408     thin,
2409     font=\large,
2410     rectangle split horizontal,
2411     rectangle split,
2412   }
2413 }
```

```

2414 % https://tex.stackexchange.com/a/140895
2415 \RequirePackage{forest,xstring}
2416 \usetikzlibrary{calc}
2417
2418 \makeatletter
2419 \pgfmathdeclarefunction{strrepeat}{2}{%
2420   \begingroup\pgfmathint{#2}\pgfmath@count\pgfmathresult
2421   \let\pgfmathresult\pgfutil@empty
2422   \pgfutil@loop\ifnum\pgfmath@count>0\relax
2423     \expandafter\def\expandafter\pgfmathresult\expandafter{\pgfmathresult#1}%
2424     \advance\pgfmath@count-1\relax
2425   \pgfutil@repeat\pgfmath@smuggleone\pgfmathresult\endgroup}
2426 \makeatother
2427
2428 \def\myNodes{}
2429
2430 \ExplSyntaxOn
2431 \newcommand*\sortList[1]{%
2432   \clist_sort:Nn#1{\int_compare:nNnTF{##1}>{##2}\sort_return_swapped:\sort_return_same:}}
2433 \ExplSyntaxOff
2434
2435 \forestset{
2436   sort/.code={%
2437     \pgfmathparse{level()>\forestSortLevel}%
2438     \ifnum\pgfmathresult=0
2439       \StrSubstitute{\forestov{content}}{ }{,}{\myList}%
2440       \sortList\myList
2441       \StrSubstitute{\myList}{,}{ }{\myList}%
2442       \pgfmathparse{strrepeat("1",level())}%
2443       \xappto\myNodes{\noexpand\node at ({\forestov{name}}|-m)!-1!({\forestov{name}}$)
2444         (m\forestov{name}) {\myList}}%
2445       \pgfmathparse{level()=\forestSortLevel}%
2446       \ifnum\pgfmathresult=1
2447         \forestOget{\forestov{@first}}{name}\forestFirst
2448         \forestOget{\forestov{@last}}{name}\forestLast
2449         \xappto\myNodes{{[<-]edge (\forestOv{\forestov{@first}}{name})
2450           \ifx\forestFirst\forestLast\else edge (\forestOv{\forestov{@last}}{name})\fi}}%
2451       \fi
2452       \ifnum\forestov{@parent}=0\else
2453         \xappto\myNodes{edge (m\forestOv{\forestov{@parent}}{name})}%
2454       \fi
2455       \gappto\myNodes{;}%
2456     \fi}}
2457
2458 \forestset{sort level/.code=%
2459   \pgfmathparse{#1}\let\forestSortLevel\pgfmathresult
2460   \pgfmathparse{strrepeat("1",\forestSortLevel+1)}\let\forestOnes\pgfmathresult}
2461

```

2.37 spalten.sty

```
2462 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2463 \ProvidesPackage{lehramt-informatik-spalten}[2020/12/07 Lädt das Paket
2464 „multicol“, damit mehrspaltiger Satz mit Hilfe der Umgebung „multicols“
2465 realisiert werden kann.]
2466 \RequirePackage{multicol}
```

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

```
2467 \def\liSpaltenUmbruch{\vfill\strut\columnbreak}
```

```
2468
```

2.38 struktogramm.sty

```
2469 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2470 \ProvidesPackage{lehramt-informatik-struktogramm}[2021/01/31 Lädt das
2471 Paket struktex zum Setzen von Struktogrammen]
2472 \RequirePackage{struktex}
2473
```

2.39 syntax.sty

```
2474 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2475 \ProvidesPackage{lehramt-informatik-syntax}[2020/06/10 Ein Hüll-Paket um
2476 `minted`, das sich um die Syntax-Hervorhebung des Java-Codes kümmert.]
2477 \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
```

```
2478 \ExplSyntaxOn
2479 \directlua{
2480   syntax = require('lehramt-informatik-syntax')
2481   syntax.importiere_konfiguration('tex_repo_lokaler_pfad', '\LehramtInformatikRepository')
2482   syntax.importiere_konfiguration('github_domain', '\LehramtInformatikGithubDomain')
2483   syntax.importiere_konfiguration('github_raw_domain', '\LehramtInformatikGithubRawDomain')
2484   syntax.importiere_konfiguration('github_tex_repo', '\LehramtInformatikGithubTexRepo')
2485   syntax.importiere_konfiguration('github_code_repo', '\LehramtInformatikGithubCodeRepo')
2486   syntax.importiere_konfiguration('git_branch', '\LehramtInformatikGitBranch')
2487 }
2488 \RequirePackage{hyperref}
2489 \RequirePackage{minted}
2490 % pygmentize -L styles
2491 \usemintedstyle{colorful}
2492 %\BeforeBeginEnvironment{minted}{\begin{mdframed}}
2493 %\AfterEndEnvironment{minted}{\end{mdframed}}
2494 %\setminted{breaklines=true,linenos}
2495 \setminted{
2496   breaklines=true,
2497   linenos,
2498   fontsize=\footnotesize,
2499 }
```

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

Let-Abkürzung: `\let\j=\liJavaCode`

```
2500 \def\liJavaCode#1{\,\mintinline{java}|#1|\,}
```

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

```
2501 \def\liLatexCode#1{\mintinline{latex}|#1|}
```

```
2502 \def\li@GithubLink#1#2{
2503   \begin{flushright}
2504     \tiny
2505     Code-Beispiel~auf~Github~ansehen:~
2506     \href{#1}{\nolinkurl{#2}}
2507   \end{flushright}
2508 }
```

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

```
2509 \NewDocumentCommand{\liJavaDatei}{ O{firstline=3} m }{
2510   \inputminted[#1]{java}{
2511     \directlua{
2512       syntax.drucke_absoluten_pfad('#2', false)
2513     }
2514   }
2515   \li@GithubLink
```

```

2516     {\directlua{syntax.drucke_github_url('#2', false)}}
2517     {\directlua{syntax.drucke_relativen_repo_pfad('#2', false)}}
2518 }

\liJavaTestDatei Eine komplette Java-Test-Datei einbinden, die Verzeichnis ./Code/src/test/java/org/bschlangaul
liegt.
2519 \NewDocumentCommand{\liJavaTestDatei}{ 0{firstline=3} m }{
2520   \inputminted[#1]{java}{
2521     \directlua{
2522       syntax.drucke_absoluten_pfad('#2', true)
2523     }
2524   }
2525   \li@GithubLink
2526   {\directlua{syntax.drucke_github_url('#2', true)}}
2527   {\directlua{syntax.drucke_relativen_repo_pfad('#2', true)}}
2528 }

\liJavaExamen \liJavaExamen{66116}{2015}{03}{Kunde} Regulärer Ausdruck zum Konvertieren: \liJavaDatei([.
\liJavaExamen$1{$2}{$3}{$4}{$5}
2529 \NewDocumentCommand{\liJavaExamen}{ 0{firstline=3} m m m m }{
2530   \inputminted[#1]{java}{
2531     \directlua{
2532       syntax.drucke_absoluten_examens_pfad('#2', '#3', '#4', '#5')
2533     }
2534   }
2535
2536   \li@GithubLink
2537   {\directlua{syntax.drucke_github_examens_url('#2', '#3', '#4', '#5')}}
2538   {\directlua{syntax.drucke_relativen_examens_repo_pfad('#2', '#3', '#4', '#5')}}
2539 }

\liAssemblerCode
2540 \def\liAssemblerCode#1{\mintinline{asm}|#1|}

\liAssemblerDatei \liAssemblerDatei{relativer-pfad}: Relativer Pfad zu einer Assembler-Datei.
2541 \NewDocumentCommand{\liAssemblerDatei}{ m }{
2542   \inputminted{asm}{#1}
2543 }

\liMinispracheDatei \liMinispracheDatei{relativer-pfad}: Relativer Pfad zu einer Minisprachen-Datei
(Hochsprache für die Minimaschine von Albert Wiedemann).
2544 \NewDocumentCommand{\liMinispracheDatei}{ m }{
2545   \inputminted{componentpascal}{#1}
2546 }

\liHaskellCode \liHaskellCode{haskell}: Zum Setzen von Haskell-Code.
2547 \def\liHaskellCode#1{\mintinline{haskell}|#1|}

\liHaskellDatei \liHaskellDatei{relativer-pfad}: Relativer Pfad zu einer Haskell-Datei.
2548 \NewDocumentCommand{\liHaskellDatei}{ m }{
2549   \inputminted{haskell}{#1}
2550 }

2551 \ExplSyntaxOff

\liSqlCode \liHaskellCode{sql}: Zum Setzen von SQL-Code.
Let-Abkürzung: \let\s=\liSqlCode
2552 \def\liSqlCode#1{\mintinline{sql}|#1|}

2553

```


2.40 syntaxbaum.sty

```
2554 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2555 \ProvidesPackage{lehramt-informatik-syntaxbaum}[2021/02/14 Zum Setzen von
2556 Syntaxbäumen mit Hilfe des Pakets tikz-qtrees]
2557 \RequirePackage{tikz-qtrees}
2558
2559 \tikzset{li parsetree/.style={
2560     every internal node/.style={
2561         draw,circle
2562     },
2563     every leaf node/.style={
2564         draw,rectangle
2565     },
2566 }
2567 }
2568
```

2.41 synthese-algorithmus.sty

```
2569 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2570 \ProvidesPackage{lehramt-informatik-synthese-algorithmus}[2021/03/19
2571 Hilfsmakros zum Setzen des Synthese-Algorithmuses zur Umformung einer
2572 Relation in die 3. Normalform]

2573 \liLadePakete{normalformen,mathe,typographie}
2574 \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}

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

2.41.4 TeX-Markup Relationen formen

```
\let\r=\liRelation
\let\u=\underline

\r[R1]{\u{A, D}, E}\
\r[R2]{\u{B, C}, A, E}\
\r[R3]{\u{D}, B}
```

1. Kanonische Überdeckung

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

(a) Linksreduktion

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

(b) Rechtsreduktion

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

(c) Löschen leerer Klauseln

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

(d) Vereinigung

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

2. Relationsschemata formen

— Erzeuge für jede funktionale Abhängigkeit $\alpha \rightarrow \beta \in F_c$ ein Relationenschema $R_\alpha := \alpha \cup \beta$. —

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

```
2575 \def\liSyntheseUeberschrift#1{
2576   {
2577     \bfseries
2578     \sffamily
2579     \str_case:nn {#1} {
2580       {1} {Kanonische-Überdeckung}
2581       {1-1} {Linksreduktion}
2582       {1-2} {Rechtsreduktion}
2583       {1-3} {Löschen-leerer-Klauseln}
2584       {1-4} {Vereinigung}
2585       {2} {Relationsschemata-formen}
2586       {3} {Schlüssel-hinzufügen}
2587       {4} {Entfernung-überflüssiger-Teilschemata}
2588     }
```

```

2589 }
2590 }

```

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

```

2591 \def\li@synthese@erklaerung@texte#1{
2592   \str_case:nn {#1} {
2593     {1} {
2594       Die~kanonische~Überdeckung~---also~die~kleinst~mögliche~noch~
2595       äquivalente~Menge~von~funktionalen~Abhängigkeiten~kann~in~vier~
2596       Schritten~erreicht~werden.
2597     }
2598     {1-1} {
2599       Führe~für~jede~funktionale~Anhängigkeit~
2600        $\alpha \rightarrow \beta$ ~die~Linksreduktion~durch,~
2601       überprüfe~also~für~alle~
2602        $A \in \alpha$ ,~ob~ $A$ ~überflüssig~ist,~d.h.~ob~
2603        $\beta \subseteqq \text{liAttributHuelle}\{F, \alpha \cup A\}$ .
2604     }
2605     {1-2} {
2606       Führe~für~jede~(verbliebene)~funktionale~Abhängigkeit~ $\alpha \rightarrow \beta$ ~die~Rechtsreduktion~durch,~überprüfe~also~für~
2607       alle~ $B \in \beta$ ,~ob~ $B \in \text{liAttributHuelle}\{F, (\alpha \rightarrow \beta) \cup (\alpha \rightarrow B)\}$ ,~
2608        $\alpha$ ~gilt.~In~diesem~Fall~ist~ $B$ ~auf~der~rechten~Seite~
2609       überflüssig~und~kann~eliminiert~werden,~d.h.~ $\alpha \rightarrow \beta$ ~wird~durch~ $\alpha \rightarrow (\beta \cup B)$ ~
2610       ersetzt.
2611     }
2612     {1-3} {
2613       Entferne~die~funktionalen~Abhängigkeiten~der~Form~ $\alpha \rightarrow \emptyset$ ,~die~im~2.~Schritt~möglicherweise~
2614       entstanden~sind.
2615     }
2616     {1-4} {
2617       Fasse~mittels~der~Vereinigungsregel~funktionale~Abhängigkeiten~
2618       der~Form~ $\alpha \rightarrow \beta_{\{1\}}, \dots, \alpha \rightarrow \beta_{\{n\}}$ ,~so~dass~ $\alpha \rightarrow \beta_{\{1\}} \cup \dots \cup \beta_{\{n\}}$ ~
2619       verbleibt.
2620     }
2621     % Kemper Seite 197
2622     {2} {
2623       Erzeuge~für~jede~funktionale~Abhängigkeit~ $\alpha \rightarrow \beta$ ~ein~Relationenschema~ $\mathcal{R}_{\alpha \rightarrow \beta}$ ~
2624        $:= \alpha \cup \beta$ .
2625     }
2626     {3} {
2627       Falls~eines~der~in~Schritt~2.~erzeugten~Schemata~ $\mathcal{R}_{\alpha \rightarrow \beta}$ ~
2628       einen~Schlüsselkandidaten~von~ $\mathcal{R}_{\alpha \rightarrow \beta}$ ~bezüglich~ $F_{\alpha \rightarrow \beta}$ ~
2629       enthält,~sind~wir~fertig,~sonst~wähle~einen~Schlüsselkandidaten~
2630        $\mathcal{K} \subseteqq \mathcal{R}_{\alpha \rightarrow \beta}$ ~aus~und~definiere~folgendes~
2631       zusätzliche~Schema:~ $\mathcal{R}_{\alpha \rightarrow \beta} \setminus \mathcal{K} := \mathcal{K}$ ~
2632       und~ $\mathcal{F}_{\alpha \rightarrow \beta} \setminus \mathcal{K} := \emptyset$ 
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

```

```

2651     \liParagraphMitLinien{\li@synthese@erklaerung@texte{#1}}
2652   }
2653 }

```

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

```

2654 \def\liSyntheseUeberErklaerung#1{
2655   \liSyntheseUeberschrift{#1}\par
2656   \liSyntheseErklaerung{#1}
2657 }

```

```

2658 \ExplSyntaxOff
2659

```

2.42 tabelle.sty

2660 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2661 \ProvidesPackage{lehramt-informatik-tabelle}[2020/12/05 Lädt das Paket tabluarx]

2662 \RequirePackage{tabularx}

2663

2.43 typographie.sty

```
2664 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2665 \ProvidesPackage{lehramt-informatik-typographie}[2021/03/20 Typographische Makros,
2666 die das Erscheinungsbild verändern. Die Schriftdefinition sind in
2667 formatierung.sty definiert.]
```

```
2668 \ExplSyntaxOn
```

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

```
2669 \RequirePackage{fontawesome}
```

```
\liErledigt \liErledigt: ☑
```

```
2670 \let\liErledigt=\faCheckSquareO
```

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

```
2671 \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 —

```
2672 \def\liParagraphMitLinien#1{
```

```
2673 \noindent
```

```
2674 \vrule height 2pt depth -1.6pt width 0.4cm
```

```
2675 \enspace
```

```
2676 #1
```

```
2677 \enspace
```

```
2678 \leaders\vrule height 2pt depth -1.6pt \hfill \null
```

```
2679 \par
```

```
2680 \medskip
```

```
2681 }
```

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

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

```
2682 \def\liGeschweifteKlammern#1#2#3#4{
```

```
2683 \par
```

```
2684 \medskip
```

```
2685 \noindent
```

```
2686 #1 \, \$= \Bigl\{ \$
```

```
2687 \vspace{#3}
```

```
2688 #2
```

```
2689 \vspace{#4}
```

```
2690 \begin{flushright} \$\Bigr\} \$\end{flushright}
```

```
2691 \par
```

```
2692 }
```

```
2693 \ExplSyntaxOff
```

```
2694
```

2.44 uml.sty

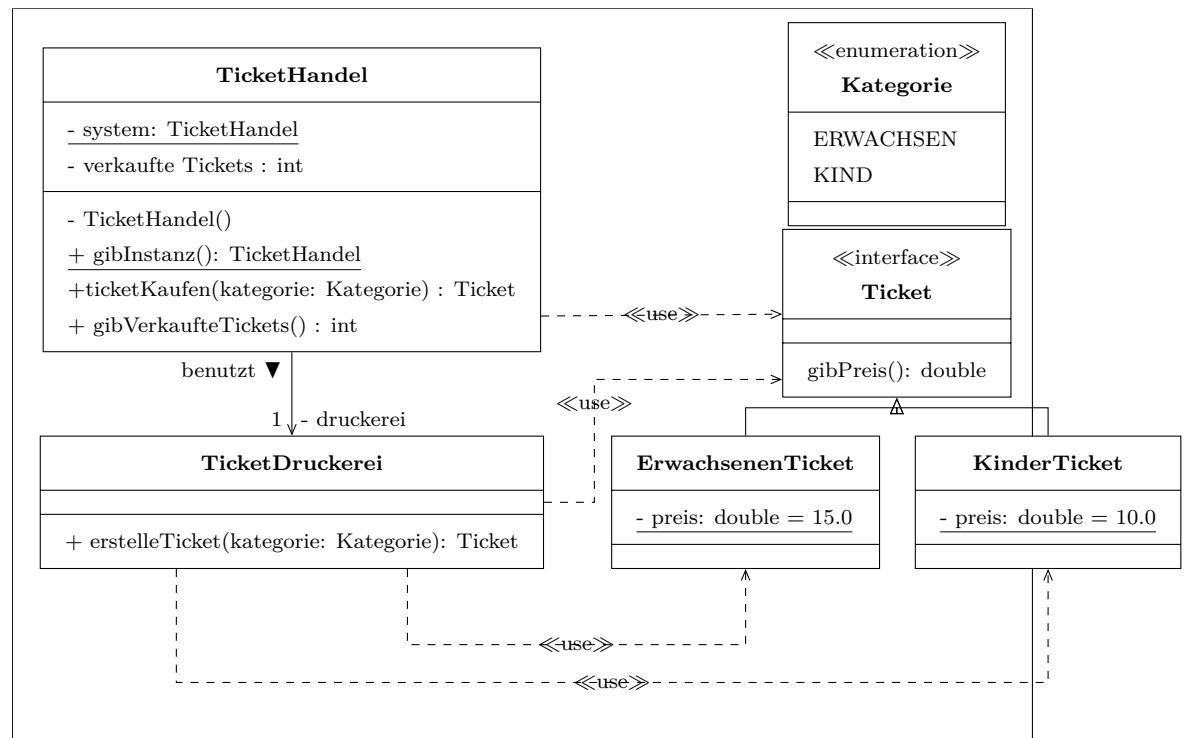
```

2695 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2696 \ProvidesPackage{lehramt-informatik-uml}[2020/06/13 Ein Hüll-Paket um
2697 `tikz-uml', das einige Design-Einstellungen vornimmt und manche
2698 Erweiterung bereitstellt]

2699 \RequirePackage{tikz-uml}
2700 \RequirePackage{tikz-uml-activity}
2701 % Not compatible with wasysym
2702 %\RequirePackage{mathabx}
2703 \RequirePackage{wasysym}
2704 \usetikzlibrary{positioning}

2705 \tikzumlset{
2706   fill class=white!0,
2707   font=\footnotesize,
2708   fill object=white!0,
2709   fill note=white!0,
2710   fill state=white!0,
2711   % Use case
2712   fill usecase=white!0,
2713   fill system=white!0,
2714 }

```



```

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

```

```

2715 \NewDocumentCommand{ \liUmlLeserichtung } { 0{dir=right} m } {
2716   \def\@liDirLeft{
2717     \def\@liDirRight{ \RIGHTarrow}
2718     \pgfkeys{/lese/dir/.is choice}
2719     \pgfkeys{/lese/dir/up/.code={\def\@liDirRight{ \UParrow}}}
2720     \pgfkeys{/lese/dir/down/.code={\def\@liDirRight{ \DOWNarrow}}}
2721     \pgfkeys{/lese/dir/left/.code={\def\@liDirRight{\def\@liDirLeft{\LEFTarrow }}}
2722     \pgfkeys{/lese/dir/right/.code={\def\@liDirRight{ \RIGHTarrow}}}
2723
2724     \def\@liPos{above}
2725     \pgfkeys{/lese/pos/.code={\def\@liPos{##1}}}
2726

```



```

2727 \def\@liDistance{0cm}
2728 \pgfkeys{/lese/distance/.code={\def\@liDistance{##1}}}}
2729
2730 \pgfkeys{/lese/.cd,#1}
2731
2732 \node[\@liPos = \@liDistance of #2-middle] {
2733   \@liDirLeft{\footnotesize#2}\@liDirRight
2734 };
2735 }
2736

```

2.45 vollstaendige-induktion.sty

2737 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2738 \ProvidesPackage{lehramt-informatik-vollstaendige-induktion}[2021/07/01
 2739 Hilfsmakros zum Setzen der Vollständigen Induktion, vor allem die
 2740 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\}
&\text{\e{Java nach Mathe}}\\
\%
&= \frac{
  \{(4\m{n} + 2) \cdot \text{cn}(\m{n})\}
  \{\m{n} + 2\}
&\text{\e{addiert, subtrahiert}}\\
\%
&= \frac{
  \{(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)}\}
&\text{\e{\$(n + 1)\$ multipliziert}} \\
\%
&= \frac{
  \{(4n + 2) \cdot \m{(n + 1) \cdot (2n)!}\}
  \{(n + 2) \cdot (n + 1)! \cdot \m{(n + 1) \cdot n!}\}
&\text{\e{umsortiert}} \\
\%
&= \frac{
  \{\m{(2(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)!\}
&\text{\e{\$(n + 1)\$ verdeutlicht}}\\
\end{align*}
```

Lade häufig benötigte Pakete

2741 \RequirePackage{lehramt-informatik-typographie}
 2742 \RequirePackage{lehramt-informatik-mathe}
 2743 \RequirePackage{lehramt-informatik-syntax}
 2744 \ExplSyntaxOn

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

Let-Abkürzung: \let\m=\liInduktionMarkierung

2745 \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

```
2746 \def\liInduktionErklaerung#1{\scriptsize\text{#1}}
```

\liInduktionAnfang

```
2747 \def\liInduktionAnfang{
```

```
2748   \liPseudoUeberschrift{Induktionsanfang}
```

```
2749
```

```
2750   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
2751   \liParagraphMitLinien{
```

```
2752     Beweise,~dass~ $A(1)$ ~eine~wahre~Aussage~ist.
```

```
2753   }
```

```
2754 }
```

\liInduktionVoraussetzung

```
2755 \def\liInduktionVoraussetzung{
```

```
2756   \liPseudoUeberschrift{Induktionsvoraussetzung}
```

```
2757
```

```
2758   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
2759   \liParagraphMitLinien{
```

```
2760     Die~Aussage~ $A(k)$ ~ist~wahr~für~ein~beliebiges~ $k \in \mathbb{N}$ .
```

```
2761   }
```

```
2762 }
```

\liInduktionSchritt

```
2763 \def\liInduktionSchritt{
```

```
2764   \liPseudoUeberschrift{Induktionsschritt}
```

```
2765
```

```
2766   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
2767   \liParagraphMitLinien{
```

```
2768     Beweise,~dass~wenn~ $A(n=k)$ ~wahr~ist,~
```

```
2769     auch~ $A(n=k+1)$ ~wahr~sein~muss.
```

```
2770   }
```

```
2771 }
```

```
2772 \ExplSyntaxOff
```

```
2773
```

2.46 wasserfall.sty

```
2774 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2775 \ProvidesPackage{lehramt-informatik-wasserfall}[2020/06/10]
2776 \RequirePackage{tikz}
2777 \tikzset{wasserfall/.style={
2778   >=stealth,
2779   node distance = 2mm and -8mm,
2780   start chain = A going below right,
2781   every node/.style = {
2782     draw,
2783     text width=24mm,
2784     minimum height=12mm,
2785     align=center,
2786     inner sep=1mm,
2787     fill=white,
2788     drop shadow={fill=black},
2789     on chain=A
2790   },
2791 }}
2792 \usetikzlibrary{chains,positioning,shadows}
2793
```

2.47 wpkalkuel.sty

2794 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2795 \ProvidesPackage{lehramt-informatik-wpkalkuel}[2020/06/13]

Let-Abkürzungen

\let\wp=\liWpKalkuel

\let\equivalent=\liWpEquivalent

\let\erklaerung=\liWpErklaerung

2796 \RequirePackage{amsmath}

2797 \ExplSyntaxOn

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

2798 \def\liWpKalkuelOhneMathe#1#2{

2799 \text{wp}(\texttt{\scriptsize"#1"},\thinspace #2)

2800 }

2801 \def\liWpKalkuel#1#2{

2802 \ifmmode

2803 \liWpKalkuelOhneMathe{#1}{#2}

2804 \else

2805 \$\liWpKalkuelOhneMathe{#1}{#2}\$

2806 \fi

2807 }

\MatheEnv

2808 \def\MatheEnv#1{

2809 \medskip

2810

2811 \hspace{1em}#1

2812

2813 \medskip

2814 }

\Mathe

2815 \def\Mathe#1{

2816 \MatheEnv{#1\$}

2817 }

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

2818 \def\liWpEquivalent#1{

2819 \MatheEnv{\$\equiv\$\hspace{1em}#1\$}

2820 }

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

2821 \newlength{\@Skip@Erklaerung@Reset}

2822 \def\liWpErklaerung#1{

2823 \setlength{\@Skip@Erklaerung@Reset}{\leftskip}

2824 \setlength{\leftskip}{0.5cm}

2825

2826 \par

2827 \noindent

2828 {

2829 \scriptsize

2830 #1

2831 }

2832 \par

2833

2834 \setlength{\leftskip}{\@Skip@Erklaerung@Reset}

2835 }

\liWpErklaerungVerzweigung

```

2836 \def\liWpErklaerungVerzweigung{
2837   $\liWpKalkuelOhneMathe{if~\{-b~\}-then~\{-a1~\}-else~\{-a2~\}}{Q}
2838   \equiv
2839   (b \land \liWpKalkuelOhneMathe{a1}{Q})
2840   \lor
2841   (\neg b \land \liWpKalkuelOhneMathe{a2}{Q})$
2842 }

2843 \ExplSyntaxOff

2844

```

3 Index

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

Symbols		
\# 109	\AddToHook 1349	\bfseries .. 486, 1234,
\, 343, 398, 1181,	\advance 2424	1236, 2197, 2203,
1183, 1694, 1695,	\AfterEndEnvironment 2493	2205, 2207, 2208, 2577
1696, 2030, 2500, 2686	\allsectionsfont ... 1230	\Bigl 2686
\@Skip@Erklaerung@Reset	\Alph 1244	\Bigr 2690
... 2821, 2823, 2834	\alph 1244, 1245	\bigskip 48, 376,
\@afterheading 1661	\alpha 2600, 2602, 2603,	609, 614, 1570, 1898
\@afterindentfalse . 1661	2606, 2608, 2609,	\bool 321, 344
\@liDirLeft 2716, 2721, 2733	2610, 2611, 2612,	\bowtie
\@liDirRight 2717, 2719,	2616, 2622, 2623,	2345, 2348, 2349, 2350
2720, 2721, 2722, 2733	2628, 2629, 2630,	\Box 159
\@liDistance	2633, 2641, 2642, 2643	\boxtimes 474
... 2727, 2728, 2732	\arabic 1244, 2219, 2224,	
\@liPos .. 2724, 2725, 2732	2229, 2235, 2241, 2247	C
\\ 607,	\arraystretch 1920	\c 1327, 1328
630, 631, 634, 635,		\cdot 1738, 1791, 1802
638, 639, 731, 732,	B	\centerline
733, 840, 869, 871,	\BeforeBeginEnvironment	1368, 2298, 2320, 2335
897, 906, 951, 993, 2492	\chapter 1234, 1235
994, 995, 1000,	\begin .. 628, 679, 694,	\char 1568
1001, 1002, 1022,	729, 753, 800, 832,	\clearpage 1562
1568, 1925, 1983, 1986	847, 867, 877, 893,	\cline 607
\{ 207, 1118,	913, 945, 962, 991,	\clist 226, 270,
1128, 1140, 1141,	1016, 1037, 1052,	271, 290, 294, 2432
1146, 1180, 1409,	1161, 1250, 1325,	\columnbreak 2467
1970, 2375, 2686, 2837	1363, 1372, 1379,	\cs 293, 318, 342,
\} 207, 1118,	1487, 1560, 1612,	343, 380, 392, 1644
1128, 1140, 1141,	1617, 1625, 1650,	\csname 1312, 1315
1148, 1184, 1410,	1655, 1666, 1676,	\cup 1141,
1970, 2375, 2690, 2837	1680, 1763, 1767,	2004, 2609, 2623, 2630
_ 27, 35, 44, 46,	1783, 1806, 1829,	
293, 318, 342, 343,	1844, 1923, 1924,	D
357, 358, 364, 367,	1974, 2019, 2139,	\DeclareMathSymbol ..
370, 380, 392, 1350	2159, 2300, 2322, 1868, 1869
	2336, 2492, 2503, 2690	\DecoINERT
	\begingroup 1561, 1972, 2420	2222, 2295, 2296, 2318
	\beschriftung	\DecoINERTwithPivot .
 1592, 1596, 2238, 2317
	1600, 1604, 1608, 1610	\DecoLEFT 2217, 2316
A	\beta 2600,	\DecoLEFTwithPivot ..
\addbibresource	2603, 2607, 2608, 2232, 2294
1538, 1539, 1540,	2609, 2612, 2622,	\DecoRIGHT 2227, 2319, 2329
1541, 1542, 1543,	2623, 2624, 2629, 2630	\DecoRIGHTwithPivot .
1544, 1545, 1546, 1547	\bf 2197, 2198, 2199 2244, 2297

[illegible]

\liEntwurfsAbstrakteFabrik	674	\liEntwurfsModellPraesentation	1015, 1033	\liSteuerDiagramm	102
\liEntwurfsAbstrakteFabrikCode	669, 676	\liEntwurfsZustand	1069	\liKellerKante	140
\liEntwurfsAbstrakteFabrikUml	627, 675	\liEntwurfsZustandAkteure	135, 141	\liKellerUebergang	..
\liEntwurfsAdapter	723	\liEntwurfsZustandUml	1036, 1070	\liKontrollCode	1498
\liEntwurfsAdapterAkteure	693, 725	\liEpsilon	1126	\liKontrollflussgraph	(environment) 1486
\liEntwurfsAdapterCode	717, 726	\liErAttribute	1084, 1098, 1100	\liKontrollKnotenPfad	1500
\liEntwurfsAdapterUml	678, 724	\liErDatenbankName	1103	\liKontrollTextzeileKnoten	1499, 1504
\liEntwurfsBeobachter	794	\liErEntity	1082, 1086, 1088	\liKurzeTabellenLinie	607
\liEntwurfsBeobachterAkteure	752, 796	\liErledigt	2670	\liLadeAllePakete	228
\liEntwurfsBeobachterCode	786, 797	\liErMpAttribute	1097	\liLadePakete	54, 57, 224, 229, 483, 544, 1117, 1347, 1358, 1874, 1954, 2102, 2573
\liEntwurfsBeobachterUml	728, 795	\liErMpEntity	1085	\liLatexCode	2501
\liEntwurfsDekorierer	826	\liErMpRelationship	1091	\liLeereZelle	1912
\liEntwurfsDekoriererAkteure	828	\liErRelationship	1083, 1092, 1094	\liLernkartei	(environment) 1664
\liEntwurfsDekoriererCode	818, 829	\liExamensAufgabe	6	\liLinksReduktion	1979
\liEntwurfsDekoriererUml	799, 827	\liExamensAufgabeA	15	\liLinksReduktionInline	1988, 1996
\liEntwurfsEinfacheFabrik	862	\liExamensAufgabeTA	12	\liMasterExkurs	1843
\liEntwurfsEinfacheFabrikAkteure	846, 864	\liExamensAufgabeTTA	9	\liMasterFaelle	1782, 1850
\liEntwurfsEinfacheFabrikUml	831, 863	\liExkurs (environment)	1623	\liMasterFallRechnung	1828
\liEntwurfsEinzelstueck	887	\liFalsch	475	\liMasterVariablen	1762, 1845
\liEntwurfsEinzelstueckAkteure	876, 889	\liFlaci	1187	\liMasterVariablenDeklaration	1805
\liEntwurfsEinzelstueckCode	884, 890	\liFremd	2358	\liMasterWolframLink	1853
\liEntwurfsEinzelstueckUml	866, 888	\liFunktionaleAbhaengigkeit	2001, 2004, 2012	\liMenge	71, 72, 74, 113, 114, 115, 119, 171, 172, 173, 177, 1118, 1167, 1206, 1207
\liEntwurfsErbauer	940	\liFunktionaleAbhaengigkeiten	2015	\liMengeOhneMathe	1118, 1121, 1123
\liEntwurfsErbauerAkteure	912, 942	\liFussnote	1875, 1877	\liMinimierungErklaerung	1934
\liEntwurfsErbauerUml	892, 941	\liFussnoteDreiText	1889, 1907	\liMinispracheDatei	2544
\liEntwurfsFabrikmethode	986, 1011	\liFussnoteEinsText	1881, 1901	\liLinespread	2366
\liEntwurfsFabrikmethodeAkteure	961, 988	\liFussnoteLink	1691	\liNichtsZuTun	2671
\liEntwurfsFabrikmethodeUml	944, 987	\liFussnoten	1897	\liO	1725, 1753
\liEntwurfsKompositum	1011	\liFussnoteUrl	1030, 1687	\liOmega	1715, 1759
\liEntwurfsKompositumAkteure	1013	\liFussnoteVierText	1893, 1910	\liOmegaOhneMathe	1715, 1720, 1722
\liEntwurfsKompositumUml	990, 1012	\liFussnoteZweiText	1885, 1904	\liOOhneMathe	1725, 1730, 1732
\liEntwurfsModellPraesentation	1032	\liGeschweifteKlammern	1159, 2016, 2682	\liParagraphMitLinien	532, 1936, 2651, 2672, 2751, 2759, 2767
\liEntwurfsModellPraesentation	1034	\liGrammatik	1198	\liPetriErreichKnotenDrei	2096
\liEntwurfsModellPraesentation	1034	\liGraphenFormat (environment)	1304	\liPetriErreichTransition	2093
\liEntwurfsModellPraesentation	1034	\liHanoi	1311	\liPetriSetzeSchluessel	2038
\liEntwurfsModellPraesentation	1034	\liHaskellCode	2547	\liPetriTransitionsName	2085, 2097
\liEntwurfsModellPraesentation	1034	\liHaskellDatei	2548	\liPetriTransitionsNameOhneMathe	2085, 2088, 2090
\liEntwurfsModellPraesentation	1034	\liInduktionAnfang	2747		
\liEntwurfsModellPraesentation	1034	\liInduktionErklaerung	2746		
\liEntwurfsModellPraesentation	1034	\liInduktionMarkierung	2745		
\liEntwurfsModellPraesentation	1034	\liInduktionSchritt	2763		
\liEntwurfsModellPraesentation	1034	\liInduktionVoraussetzung	2755		
\liEntwurfsModellPraesentation	1034	\liJavaCode	2500		
\liEntwurfsModellPraesentation	1034	\liJavaExamen	2529		
\liEntwurfsModellPraesentation	1034	\liJavaTestAkteure	2519		
\liEntwurfsModellPraesentation	1034	\liKasten (environment)	1249		

<code>\liPetriTransPfeile</code> 2097	<code>\liTheta</code> 1705 , 1756 , 1785 , 1791 , 1796	<code>\loop</code> 2179	
<code>\liPolynomiellReduzierbar</code> 1378	<code>\liThetaOhneMathe</code> 1705 , 1710 , 1712	<code>\lor</code> 2840	
<code>\liPotenzmenge</code> 1127 , 1131 , 2108	<code>\liTOhneMathe</code> 1735 , 1744 , 1746	<code>\ltimes</code> 1861	
<code>\liPotenzmengeOhneMathe</code> ... 1128 , 1129 , 1130	<code>\liTuringKante</code> 203	M	
<code>\liPrimaer</code> 2357	<code>\liTuringLeerzeichen</code> 159 , 167	<code>\makeatletter</code> .. 1660 , 2418	
<code>\liProblemBeschreibung</code> 1362	<code>\liTuringMaschine</code> ... 160	<code>\makeatother</code> .. 1662 , 2426	
<code>\liProblemClique</code> ... 1385	<code>\liTuringUeberfuehrung</code> 206	<code>\marginpar</code> 1087 , 1093 , 1099 , 1567	
<code>\liProblemName</code> 1361 , 1368 , 1380 , 1382 , 1395 , 1406 , 1407 , 1415 , 1416	<code>\liTuringUebergaenge</code> 198 , 204	<code>\mathbb</code> .. 1416 , 2166 , 2760	
<code>\liProblemSat</code> 1414	<code>\liTuringUebergangZelle</code> 193	<code>\mathbin</code> . 2348 , 2349 , 2350	
<code>\liProblemSubsetSum</code> 1405 , 1414	<code>\liUeberfuehrungsFunktion</code> 1132	<code>\mathcal</code> .. 1726 , 2629 , 2634 , 2636 , 2637 , 2638	
<code>\liProblemVertexCover</code> 1385 , 1393	<code>\liUeberfuehrungsFunktionOhneMathe</code> ... 1132 , 1135 , 1137	<code>\Mathe</code> 2815	
<code>\liProduktionen</code> 1166 , 1208	<code>liUebergangsTabelle</code> (environment) 1920	<code>\MatheEnv</code> 2808 , 2816 , 2819	
<code>liProduktionsRegeln</code> (environment) 1156	<code>\liUeberschriftDreiecksTabelle</code> 1930	<code>\mathord</code> 1868 , 1869	
<code>liProjektSprache</code> (envi- ronment) 1584	<code>\liUmlLeserichtung</code> . 2715	<code>\mdfsetup</code> 1248 , 1593 , 1597 , 1601 , 1605	
<code>\liPseudoUeberschrift</code> 1569 , 1619 , 1620 , 1922 , 1932 , 2748 , 2756 , 2764	<code>\liVertauschen</code> 2386	<code>\medskip</code> 1370 , 1578 , 1580 , 1654 , 1682 , 2364 , 2370 , 2680 , 2684 , 2809 , 2813	
<code>\liPumpingKontextfrei</code> 2154	<code>\liWortInSprache</code> 608	<code>\emph</code> 1567	
<code>\liPumpingRegulaer</code> . 2133	<code>\liWortNichtInSprache</code> 613	<code>\mintinline</code> 2500 , 2501 , 2540 , 2547 , 2552	
<code>liQuellen</code> (environment) 1644	<code>\liWpEquivalent</code> 2818	<code>\mkern</code> ... 2348 , 2349 , 2350	
<code>\liRechtsReduktionInline</code> 1996	<code>\liWpErklaerung</code> 2821	<code>\mlq</code> 1866 , 1868	
<code>\liRekursionsGleichung</code> 1749 , 1809	<code>\liWpErklaerungVerzweigung</code> 2836	<code>\mrq</code> 1866 , 1869	
<code>\liRelation</code> 2026	<code>\liWpKalkuel</code> 2798	<code>\msg</code> 39 , 403	
<code>liRelationenSchemaFormat</code> (environment) 2379	<code>\liWpKalkuelOhneMathe</code> 2798 , 2803 , 2805 , 2837 , 2839 , 2841	<code>\myList</code> 2439 , 2440 , 2441 , 2444	
<code>\liRelationMenge</code> ... 2373	<code>\liZustandsBuchstabe</code> 1142 , 1151 , 1153 , 1171 , 1173	<code>\myNodes</code> 2428 , 2443 , 2449 , 2453 , 2455	
<code>\liRichtig</code> 474	<code>\liZustandsBuchstabeGross</code> ... 1143 , 1152 , 1154	N	
<code>liRmodell</code> (environment) 2359	<code>\liZustandsmenge</code> ... 1130	<code>\NeedsTeXFormat</code> . 1 , 19 , 52 , 220 , 409 , 470 , 477 , 541 , 604 , 619 , 1074 , 1110 , 1223 , 1263 , 1272 , 1277 , 1306 , 1344 , 1354 , 1426 , 1509 , 1524 , 1530 , 1552 , 1698 , 1858 , 1871 , 1950 , 2033 , 2099 , 2124 , 2129 , 2173 , 2339 , 2352 , 2381 , 2462 , 2469 , 2474 , 2554 , 2569 , 2660 , 2664 , 2695 , 2737 , 2774 , 2794	
<code>\liRundeKlammer</code> . 1702 , 1706 , 1716 , 1726 , 1740	<code>\liZustandsmengeNr</code> 1144 , 2117	<code>\neg</code> 2841	
<code>\liSortierMarkierung</code> 2398	<code>\liZustandsmengeNrGross</code> 1152	<code>\negthinspace</code> 1703	
<code>\liSortierPfeil</code> 2392	<code>\liZustandsMengenSammlung</code> 2104	<code>\newcounter</code> ... 2193 , 2194	
<code>\liSortierPfeilUnten</code> 2395	<code>\liZustandsMengenSammlungNr</code> 2113	<code>\NewDocumentCommand</code> . 62 , 103 , 140 , 161 , 203 , 224 , 546 , 563 , 608 , 613 , 1177 , 1199 , 1378 , 1501 , 1528 , 1687 , 1691 , 2015 , 2026 , 2093 , 2509 , 2519 , 2529 , 2541 , 2544 , 2548 , 2715	
<code>\liSpaltenUmbruch</code> .. 2467	<code>\liZustandsmengeOhneMathe</code> 1130		
<code>\liSqlCode</code> 2552	<code>\liZustandsname</code> 1153		
<code>\listen@punkt</code> .. 1644 , 1656	<code>\liZustandsnameGross</code> ... 1154 , 2105 , 2114		
<code>\liStrich</code> 1360	<code>\liZustandsnameTiefgestellt</code> 1169		
<code>\liSyntheseErklaerung</code> 2591 , 2656	<code>\liZustandsPaar</code> 1914		
<code>\liSyntheseUeberErklaerung</code> 2654	<code>\liZustandsPaarVariablenName</code> ... 1913 , 1916 , 1917		
<code>\liSyntheseUeberschrift</code> 2575 , 2655	<code>\llap</code> 2346		
<code>\liT</code> 1735 , 1750 , 1764 , 1825	<code>\log</code> 1753 , 1756 , 1759 , 1785 , 1791	<code>\NewDocumentEnvironment</code> 1156 , 1249 , 1304 ,	
<code>\liTeilen</code> 1959			
<code>\literatur</code> 1526 , 1550			

\xintLength	2253		Z		\zustandsnamens@liste
			\ZB	<u>1695</u>	... 1144, 1151, 1152
\xintnthelt	2265	\zB	<u>1694</u>	