

lehramt-informatik

Hermine Bschlangaul *

September 6, 2021

Contents

1	Klassen	4
1.1	Vorlage Theorie-Teil	5
1.2	Vorlage Aufgabensammlung	6
1.3	Vorlage Aufgabe	7
2	Pakete	8
2.1	abmessung.sty	9
2.2	aufgaben-einbinden.sty	10
2.3	aufgaben-metadaten.sty	11
2.4	automaten.sty	12
2.4.1	Endlicher Automat	12
2.4.2	Kellerautomat	14
2.4.3	Turingmaschine	15
2.5	basis.sty	18
2.6	baum.sty	22
2.6.1	Binärbaum	23
2.6.2	AVL-Baum	24
2.6.3	B-Baum	25
2.7	checkbox.sty	26
2.8	chomsky-normalform.sty	27
2.8.1	Makro-Kürzel	27
2.8.2	TeX-Markup-Grundgerüst	27
2.8.3	Konkretes TeX-Markup-Beispiel	27
2.9	cpm.sty	30
2.9.1	Makro-Kürzel	30
2.9.2	Hilfsmakros für Zeitpunkt-Tabelle	31
2.10	cyk-algorithmus.sty	33
2.10.1	Makro-Kürzel	33
2.10.2	TeX-Markup-Beispiel	33
2.11	entwurfsmuster.sty	34
2.11.1	Namensschema der Entwurfsmuster-Makros:	34
2.11.2	Reihenfolge	34
2.11.3	Abstrakte Fabrik (Abstract Factory)	34
2.11.4	Adapter	36
2.11.5	Beobachter (Observer)	37
2.11.6	Dekorierer (Decorator)	39
2.11.7	Einfache Fabrik (Simple Factory)	40
2.11.8	Einzelstück (Singleton)	41
2.11.9	Erbauer (Builder)	42
2.11.10	Fabrikmethode (Factory Method)	43
2.11.11	Kompositum (Composite)	45
2.11.12	Modell-Präsentation-Steuerung (Model-View-Controller)	45

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

2.11.13	Stellvertreter (Proxy)	46
2.11.14	Zustand (State)	47
2.12	er.sty	49
2.12.1	Tex-Markup-Beispiel: Komplettes Diagramm	49
2.12.2	Tex-Markup-Beispiel: EER Enhanced Entity-Relation-Modell nach Elmasri/Navante	50
2.12.3	Makro-Kürzel	50
2.13	formale-sprachen.sty	52
2.14	formatierung.sty	55
2.14.1	Schriftarten / Typographie	55
2.14.2	Farben	55
2.14.3	Überschriften	55
2.14.4	Listen	55
2.14.5	Kasten	55
2.14.6	Header	55
2.15	gantt.sty	56
2.16	grafik.sty	57
2.17	graph.sty	58
2.18	hanoi.sty	60
2.19	klassen-konfiguration-aufgabe.sty	61
2.20	klassen-konfiguration-examen.sty	62
2.21	komplexitaetstheorie.sty	64
2.21.1	Makro-Kürzel	64
2.22	kontrollflussgraph.sty	66
2.22.1	Makro-Kürzel	66
2.22.2	TeX-Markup-Beispiel	66
2.22.3	TikZ: pin	66
2.22.4	Umgebungen	67
2.22.5	Makros	68
2.23	kopf-fusszeilen.sty	69
2.24	literatur-dummy.sty	70
2.25	literatur.sty	71
2.26	makros.sty	72
2.27	master-theorem.sty	76
2.27.1	Makro-Kürzel	76
2.28	mathe.sty	80
2.29	minimierung.sty	81
2.30	normalformen.sty	84
2.30.1	Makro-Kürzel	84
2.31	petri.sty	87
2.31.1	Makro-Kürzel	87
2.32	potenzmengen-konstruktion.sty	89
2.33	pseudo.sty	91
2.34	pumping-lemma.sty	92
2.35	quicksort.sty	93
2.36	relationale-algebra.sty	96
2.37	rmodell.sty	97
2.37.1	Makro-Kürzel	97
2.38	sortieren.sty	98
2.39	spalten.sty	100
2.40	sql.sty	101
2.41	struktogramm.sty	102
2.42	syntax.sty	103
2.42.1	Makro-Kürzel	103
2.43	syntaxbaum.sty	105
2.44	synthese-algorithmus.sty	106
2.44.1	Makro-Kürzel	106
2.44.2	TeX-Markup Grundgerüst	106

2.44.3	TeX-Markup Linksreduktion	106
2.44.4	TeX-Markup Rechtreduktion	106
2.44.5	TeX-Markup Relationen formen	107
2.45	tabelle.sty	110
2.46	typographie.sty	111
2.47	uml.sty	113
2.48	vollstaendige-induktion.sty	115
2.48.1	Makro-Kürzel	115
2.49	wasserfall.sty	117
2.50	wpkalkuel.sty	118
2.50.1	Makro-Kürzel	118
3	Index	119

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 abmessung.sty

```
1 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2 \ProvidesPackage{lehramt-informatik-baum}[2021/09/04 Einstellung der
3 Seitenabmessung mit Hilfe des geometry-Pakets.]
4 \RequirePackage{geometry}
5 \geometry{
6   a4paper,
7   margin=2cm,
8   includeheadfoot,
9   %showframe,
10  %showcrop,
11  %verbose=true,
12 }
13
```

2.2 aufgaben-einbinden.sty

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

\liAufgabe Eine Aufgaben mit den Pfad relativ zu Wurzelverzeichnis dieses Repository einbinden.
16 \def\liAufgabe#1{
17   \input{\LehramtInformatikRepository/#1.tex}
18 }

\liExamensAufgabe Eine Examensaufgaben mit den Pfad relativ zu Staatsexamen einbinden, z. B.
\liExamensAufgabe{46116/2015/09/Thema-2/Teilaufgabe-1/Aufgabe-3}
19 \def\liExamensAufgabe#1{
20   \input{\LehramtInformatikRepository/Staatsexamen/#1.tex}
21 }

\liExamensAufgabeTTA
22 \def\liExamensAufgabeTTA #1 / #2 / #3 : Thema #4 Teilaufgabe #5 Aufgabe #6 {
23   \input{\LehramtInformatikRepository/Staatsexamen/#1/#2/#3/Thema-#4/Teilaufgabe-
      #5/Aufgabe-#6.tex}
24 }

\liExamensAufgabeTA
25 \def\liExamensAufgabeTA #1 / #2 / #3 : Thema #4 Aufgabe #5 {
26   \input{\LehramtInformatikRepository/Staatsexamen/#1/#2/#3/Thema-#4/Aufgabe-#5.tex}
27 }

\liExamensAufgabeA
28 \def\liExamensAufgabeA #1 / #2 / #3 : Aufgabe #4 {
29   \input{\LehramtInformatikRepository/Staatsexamen/#1/#2/#3/Aufgabe-#4.tex}
30 }

31
```

2.3 aufgaben-metadaten.sty

```

32 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
33 \ProvidesPackage{lehramt-informatik-aufgaben-metadaten}[2020/07/07 Makros zum
34 Setzen der Aufgaben-Metadaten.]

35 \ExplSyntaxOn

\liMetaSetze Setze die Metadaten einer Aufgabe.

36 \def\liMetaSetze#1{
37   \_setze_variablen_zurueck:
38
39   \tl_clear:N \g_auf_examen_teilaufgabe_nr_tl
40
41   \keys_set:nn { aufgabenmetadaten } {
42     #1
43   }
44
45   \_setze_relativen_pfad:
46 }

\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,
}

47 \def\liAufgabenMetadaten#1{
48   \liMetaSetze{#1}
49
50   \_gib_examen_titel: {}
51
52   \section{\_gib_aufgaben_titel:}
53 }

54 \ExplSyntaxOff

\liAufgabenTitel Momentan eine dummy Makro das die Thematik enthält.

55 \def\liAufgabenTitel#1{}

56

```

2.4 automaten.sty

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

2.4.1 Endlicher Automat

```
59 \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}
```



```
60 \RequirePackage{tikz}
61 \usetikzlibrary{arrows,automata,positioning}
62 \liLadePakete{mathe}
63 \directlua{
64   automaten = require('lehramt-informatik-automaten')
65 }
```

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

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

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

```

80     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
81     dea .value_forbidden:n = true,
82     dea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{DEA}}}},
83     nea .value_forbidden:n = true,
84     nea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{NEA}}}},
85   }
86
87   \keys_set:nn { automat } { #2 }
88
89   $#1 \l_typ_tl = (
90     \l_zustaende_tl,
91     \l_alphabet_tl,
92     \l_delta_tl,
93     \l_ende_tl,
94     \l_start_tl
95   )$
96 }
97 \ExplSyntaxOff

```

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

```

98 \def\liAutomatenKante#1#2#3#4{
99   \path (#1) edge[#4] node{#3} (#2);
100 }

101 \tikzset{
102   li automat/.style={
103     ->,
104     node distance=2cm
105   },
106 }

```

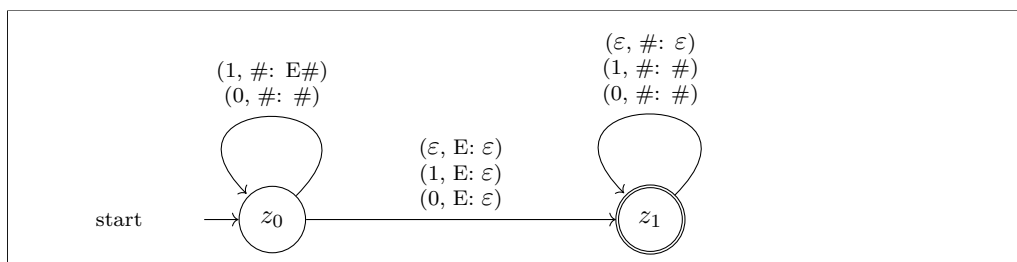
2.4.2 Kellerautomat

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

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

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

```

123     kellerboden .code:n = {\tl_set:Nn \l_kellerboden_tl {##1}},
124     ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
125   }
126
127   \keys_set:nn { kellerautomat } { #2 }
128
129   $#1 = (
130     \l_zustaende_tl,
131     \l_alphabet_tl,
132     \l_kelleralphabet_tl,
133     \l_delta_tl,
134     \l_start_tl,
135     \l_kellerboden_tl,
136     \l_ende_tl
137   )$
138 }
139 \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}`

```

140 \ExplSyntaxOn
141 \def\liKellerUebergang#1{
142   \directlua{automaten.drucke_keller_uebergaenge('#1')}
143 }
144 \ExplSyntaxOff

```

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

```

145 \NewDocumentCommand{\liKellerKante} { 0{above} m m m } {
146   \path (#2) edge[#1] node{\liKellerUebergang{#4}} (#3);
147 }

148 \tikzset{
149   li keller knoten/.style={
150     text width=2cm,
151     align=center,
152     font=\footnotesize,
153   },
154   li kellerautomat/.style={
155     li automat,
156     every edge/.append style={
157       every node/.style={
158         li keller knoten
159       }
160     }
161   }
162 }

```

2.4.3 Turingmaschine

```
163 \RequirePackage{amssymb}
```

\liTuringLeerzeichen

□

```
164 \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},
}

```

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

```

165 \ExplSyntaxOn
166 \NewDocumentCommand {\liTuringMaschine} { O{TM} m } {
167   \tl_set:Nn \l_zustaende_tl {Z}
168   \tl_set:Nn \l_alphabet_tl {\Sigma}
169   \tl_set:Nn \l_bandalphabet_tl {\Gamma}
170   \tl_set:Nn \l_delta_tl {\delta}
171   \tl_set:Nn \l_start_tl {z\sb{0}}
172   \tl_set:Nn \l_leerzeichen_tl {\liTuringLeerzeichen}
173   \tl_set:Nn \l_ende_tl {E}
174
175   \keys_define:nn { kellerautomat } {
176     zustaende .code:n = {\tl_set:Nn \l_zustaende_tl {\liMenge{##1}}},
177     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
178     bandalphabet .code:n = {\tl_set:Nn \l_bandalphabet_tl {\liMenge{##1}}},
179     delta .code:n = {\tl_set:Nn \l_delta_tl {##1}},
180     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
181     leerzeichen .code:n = {\tl_set:Nn \l_leerzeichen_tl {##1}},
182     ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
183   }
184
185   \keys_set:nn { kellerautomat } { #2 }
186
187   $\text{#1} = (
188     \l_zustaende_tl,
189     \l_alphabet_tl,
190     \l_bandalphabet_tl,
191     \l_delta_tl,
192     \l_start_tl,
193     \l_leerzeichen_tl,
194     \l_ende_tl
195   )$
196 }
197 \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)`

```

198 \ExplSyntaxOn
199 \def\liTuringUebergangZelle#1{
200   \directlua{tex.print(automaten.gib_einen_turing_uebergang('#1'))}
201 }
202 \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)`


```

203 \ExplSyntaxOn
204 \def\liTuringUebergaenge#1{
205   \directlua{automaten.drucke_turing_uebergaenge('#1')}
206 }
207 \ExplSyntaxOff

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

208 \NewDocumentCommand{\liTuringKante} { 0{above} m m m } {
209   \path (#2) edge[#1] node{\liTuringUebergaenge{#4}} (#3);
210 }

\liTuringUeberfuehrung

211 \def\liTuringUeberfuehrung{
212    $\delta : Z \times \Gamma \rightarrow Z \times \Gamma \times \{ L, R, N \}$ 
213 }

214 \tikzset{
215   li turingmaschine/.style={
216     li automat,
217     every edge/.append style={
218       every node/.style={
219         li keller knoten
220       }
221     }
222   }
223 }
224

```

2.5 basis.sty

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

227 \RequirePackage{xparse}

228 \ExplSyntaxOn

\liLadePakete

229 \NewDocumentCommand{\liLadePakete}{ m }
230 {
231   \clist_map_inline:nn { #1 } { \RequirePackage{lehramt-informatik-##1} }
232 }

\liLadeAllePakete

233 \def\liLadeAllePakete{
234   \liLadePakete{
235     aufgaben-einbinden,
236     automaten,
237     checkbox,
238     chomsky-normalform,
239     cpm,
240     cyk-algorithmus,
241     entwurfsmuster,
242     er,
243     formale-sprachen,
244     gantt,
245     grafik,
246     graph,
247     hanoi,
248     kontrollflussgraph,
249     komplexitaetstheorie,
250     makros,
251     master-theorem,
252     mathe,
253     minimierung,
254     normalformen,
255     petri,
256     potenzmengen-konstruktion,
257     pumping-lemma,
258     pseudo,
259     quicksort,
260     relationale-algebra,
261     rmodell,
262     sortieren,
263     spalten,
264     struktogramm,
265     syntax,
266     syntaxbaum,
267     synthese-algorithmus,
268     tabelle,
269     typographie,
270     uml,
271     vollstaendige-induktion,
272     wasserfall,
273     wpkalkuel,
274     baum, % am Schluss sonst Fehler: undefined command \edge
275   }
276 }
```

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.

```

277 \clist_new:N \g_auf_schluessel_clist
278 \clist_set:Nn \g_auf_schluessel_clist {
279   titel,
280   thematik,
281   stichwoerter,
282   zitat_schluessel,
283   zitat_beschreibung,
284   %
285   bearbeitungs_stand,
286   korrektheit,
287   %
288   relativer_pfad,
289   identische_aufgabe,
290   %
291   examen_nummer,
292   examen_fach,
293   examen_jahr,
294   examen_monat,
295   examen_jahreszeit,
296   examen_thema_nr,
297   examen_teilaufgabe_nr,
298   examen_aufgabe_nr,
299 }

```

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

```

300 \clist_map_inline:Nn \g_auf_schluessel_clist {
301   \tl_new:c {g_auf_#1_tl}
302 }

```

Funktion über alle globalen Token-List-Variablen zurückzusetzen.

```

303 \cs_new:Npn \setze_variablen_zurueck: {
304   \clist_map_inline:Nn \g_auf_schluessel_clist {
305     \tl_clear:c {g_auf_##1_tl}
306   }
307 }

```

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

```

308 \keys_define:nn { aufgabenmetadaten }
309 {
310   Titel .tl_gset:N = \g_auf_titel_tl,
311   Thematik .tl_gset:N = \g_auf_thematik_tl,
312   Stichwoerter .tl_gset:N = \g_auf_stichwoerter_tl,
313   ZitatSchluessel .tl_gset:N = \g_auf_zitat_beschreibung_tl,
314   ZitatBeschreibung .tl_gset:N = \g_auf_zitat_schluessel_tl,
315   %
316   BearbeitungsStand .tl_gset:N = \g_auf_bearbeitungs_stand_tl,
317   Korrektheit .tl_gset:N = \g_auf_korrektheit_tl,
318   %
319   RelativerPfad .tl_gset:N = \g_auf_relativer_pfad_tl,
320   IdentischeAufgabe .tl_gset:N = \g_auf_identische_aufgabe_tl,
321   %
322   ExamenNummer .tl_gset:N = \g_auf_examen_nummer_tl,
323   ExamenFach .tl_gset:N = \g_auf_examen_fach_tl,
324   ExamenJahr .tl_gset:N = \g_auf_examen_jahr_tl,
325   ExamenMonat .tl_gset:N = \g_auf_examen_monat_tl,
326   ExamenJahreszeit .tl_gset:N = \g_auf_examen_jahreszeit_tl,
327   ExamenThemaNr .tl_gset:N = \g_auf_examen_thema_nr_tl,
328   ExamenTeilaufgabeNr .tl_gset:N = \g_auf_examen_teilaufgabe_nr_tl,
329   ExamenAufgabeNr .tl_gset:N = \g_auf_examen_aufgabe_nr_tl,
330 }
331 \cs_gset:Npn \setze_relativen_pfad: {
332   \tl_if_empty:NTF \g_auf_relativer_pfad_tl

```

```

333 {
334   \bool_if:nTF
335   {
336     ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
337     ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
338     ! \tl_if_empty_p:N \g_auf_examen_monat_tl
339   }
340   {
341     \tl_gset:Nn \g_auf_relativer_pfad_tl {
342       Staatsexamen /
343       \g_auf_examen_nummer_tl /
344       \g_auf_examen_jahr_tl /
345       \g_auf_examen_monat_tl /
346       \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} { Thema - \g_auf_examen_thema_nr_tl / }
347       \tl_if_empty:NTF \g_auf_examen_telaufgabe_nr_tl {} { Teilaufgabe - \g_auf_examen_te
348       \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} { Aufgabe - \g_auf_examen_aufgabe_nr
349     }
350   }
351   {}
352 }
353 {}
354 }

355 \cs_gset:Npn \_gib_examen_titel: {
356   \cs_set:Nn \_trenner: { \, / \, }
357   \bool_if:nTF
358   {
359     ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
360     ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
361     ! \tl_if_empty_p:N \g_auf_examen_monat_tl &&
362     ! \tl_if_empty_p:N \g_auf_examen_aufgabe_nr_tl
363   }
364   {
365     {
366       \footnotesize
367       \par
368       \noindent
369       Staatsexamen ~
370       \g_auf_examen_nummer_tl \_trenner:
371       \g_auf_examen_jahr_tl \_trenner:
372
373       \tl_case:Nn \g_auf_examen_monat_tl
374       {
375         { 03 } { Frühjahr }
376         { 09 } { Herbst }
377       } \_trenner:
378
379       \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} {
380         Thema ~ Nr. ~ \g_auf_examen_thema_nr_tl \_trenner:
381       }
382       \tl_if_empty:NTF \g_auf_examen_telaufgabe_nr_tl {} {
383         Teilaufgabe ~ Nr. ~ \g_auf_examen_telaufgabe_nr_tl \_trenner:
384       }
385       \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} {
386         Aufgabe ~ Nr. ~ \g_auf_examen_aufgabe_nr_tl
387       }
388       \par
389       \bigskip
390     }
391   }
392 }

393 \cs_new:Npn \_gib_github_url: {
394   \tl_if_empty:NTF \g_auf_relativer_pfad_tl {} {
395     Github :~\href{

```

```

396     \LehramtInformatikGithubRawDomain /
397     \LehramtInformatikGithubTexRepo /
398     \LehramtInformatikGitBranch /
399     \g_auf_relativer_pfad_tl
400   }{
401     \nolinkurl{\g_auf_relativer_pfad_tl}
402   }
403 }
404 }

405 \cs_new:Npn \_gib_aufgaben_titel: {
406   \g_auf_titel_tl
407
408   \tl_if_empty:NTF \g_auf_thematik_tl
409   {}
410   {
411     \, ~ [
412       \g_auf_thematik_tl
413     ]
414   }
415 }

416 \msg_new:nnn { aufgabenmetadaten } { kein-titel }
417 { Der-Schlüssel-ist-zwingend-notwendig }

\def\LehramtInformatikRepository{/pfad/zum/repository}
418 \input /etc/lehramt-informatik.config.tex
    biblatex not working with lualatex and babel
419 % \RequirePackage{polyglossia}
420 % \setmainlanguage{german}
421

```

2.6 baum.sty

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

425 \RequirePackage{tikz}

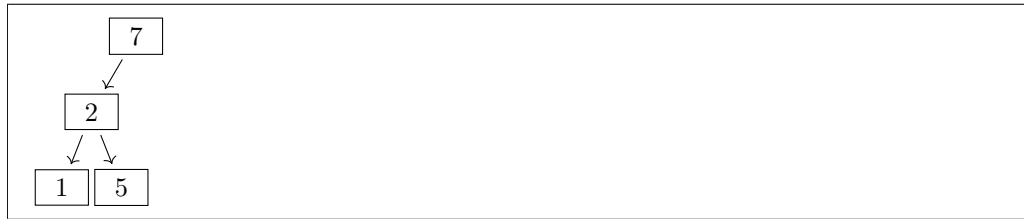
    für li binaer baum
426 \RequirePackage{tikz-qtrees}

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

2.6.1 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}
```



```
428 \tikzset{
429   li binaer baum/.style={
430     shorten <=2pt,
431     shorten >=2pt,
432     ->,
433     every tree node/.style={
434       minimum width=2em,
435       draw,
436       rectangle
437     },
438     blank/.style={
439       draw=none
440     },
441     edge from parent/.style={
442       draw,
443       edge from parent path={(\tikzparentnode) -- (\tikzchildnode)}
444     },
445     level distance=1cm,
446     every label/.style={
447       gray,
448       font=\footnotesize,
449       label position=0,
450       label distance=0cm,
451     }
452   },
453 }
```

2.6.2 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}
```

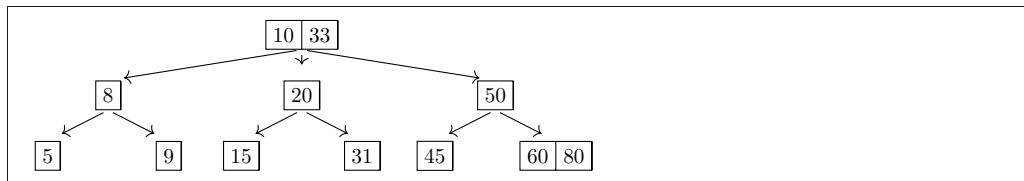


2.6.3 B-Baum

```

\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}

```



```

454 \tikzset{
455   li bbaum knoten/.style={
456     rectangle split parts=10,
457     rectangle split,
458     rectangle split horizontal,
459     rectangle split ignore empty parts,
460     draw,
461     fill=white
462   },
463   li bbaum/.style={
464     every node/.style={
465       li bbaum knoten
466     },
467     level 1/.style={
468       level distance=12mm,
469       sibling distance=25mm,
470     },
471     every child/.style={
472       shorten <= 2pt,
473       shorten >= 6pt,
474       ->,
475     },
476     level 2/.style={
477       level distance=9mm,
478       sibling distance=15mm,
479     },
480   }
481 }
482

```

2.7 checkbox.sty

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

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

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

489
```

2.8 chomsky-normalform.sty

```
490 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
491 \ProvidesPackage{lehramt-informatik-chomsky-normalform}[2021/03/26
492 Hilfsmakros zum Setzen des Algorithmus zum Erreichen der Chomsky-Normalform]

493 \ExplSyntaxOn

494 \liLadePakete{typographie}
```

2.8.1 Makro-Kürzel

```
\let\erklaerung=\liChomskyErklaerung
\let\schritt=\liChomskyUeberschrift
\let\schrittE=\liChomskyUeberErklaerung
```

2.8.2 TeX-Markup-Grundgerüst

```
\let\schrittE=\liChomskyUeberErklaerung
\begin{enumerate}
\item \schrittE{1}
\item \schrittE{2}
\item \schrittE{3}
\item \schrittE{4}
\end{enumerate}
```

2.8.3 Konkretes TeX-Markup-Beispiel

```
\begin{enumerate}
\item \schrittE{1}

\liNichtsZuTun

\item \schrittE{2}

\begin{liProduktionsRegeln}
S -> d S e | a | U c T | S b U,
T -> d S e | a,
U -> d S e | a | U c T,
\end{liProduktionsRegeln}

\item \schrittE{3}

\begin{liProduktionsRegeln}
S -> D S E | a | U C T | S B U,
T -> D S E | a,
U -> D S E | a | U C T,
B -> b,
C -> c,
D -> d,
E -> e,
\end{liProduktionsRegeln}

\item \schrittE{4}
% S -> S S.1 | T2 S.2 | a | U S.3
% T -> T2 S.2 | a
% U -> T2 S.2 | a | U S.3
% T1 -> b
% T2 -> d
% T3 -> e
% T4 -> c
% S.1 -> T1 U
% S.2 -> S T3
% S.3 -> T4 T
```

```

\begin{liProduktionsRegeln}
S -> D S_E | a | U C_T | S B_U, % S -> S S.1 | T2 S.2 | a | U S.3
T -> D S_E | a, % T -> T2 S.2 | a
U -> D S_E | a | U C_T, % U -> T2 S.2 | a | U S.3
B -> b, % T1 -> b
C -> c, % T4 -> c
D -> d, % T2 -> d
E -> e, % T3 -> e
S_E -> S E, % S.2 -> S T3
C_T -> C T, % S.3 -> T4 T
B_U -> B U, % S.1 -> T1 U
\end{liProduktionsRegeln}
\end{enumerate}

```

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

```

495 \def\liChomskyUeberschrift#1{
496 {
497   \bfseries
498   \sffamily
499   \str_case:nn {#1} {
500     {1} {Elimination-der- $\varepsilon$ -Regeln}
501     {2} {Elimination-von-Kettenregeln}
502     {3} {Separation-von-Terminalzeichen}
503     {4} {Elimination-von-mehrelementigen-Nonterminalketten}
504   }
505 }
506 }

```

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

```

507 \def\liChomsky@erklaerung@texte#1{
508   \str_case:nn {#1} {
509     %
510     {1} {
511       Alle-Regeln-der-Form- $A \rightarrow \varepsilon$ -werden-eliminiert.~
512       Die-Ersetzung-von- $A$ -wird-durch- $\varepsilon$ -in-allen-anderen-
513       Regeln-vorweggenommen.
514     }
515     {2} {
516       Jede-Produktion-der-Form- $A \rightarrow B$ -mit- $A, B$ -in- $S$ -wird-
517       als-Kettenregel-bezeichnet.~Diese-tragen-nicht-zur-Produktion-
518       von-Terminalzeichen-bei-und-lassen-sich-ebenfalls-eliminieren.
519     }
520     {3} {
521       Jedes-Terminalzeichen- $\sigma$ ,~das-in-Kombination-mit-anderen-
522       Symbolen-auftaucht,~wird-durch-ein-neues-Nonterminal-
523        $S_{\sigma}$ -ersetzt-und-die-Menge-der-Produktionen-durch-die-
524       Regel- $S_{\sigma} \rightarrow \sigma$ -ergänzt.
525     }
526     {4} {
527       Alle-Produktionen-der-Form-
528        $A \rightarrow B_1 B_2 \dots B_n$ -
529       werden-in-die-Produktionen-
530        $A \rightarrow$ 
531        $A_{n-1} B_n, A_{n-1} \rightarrow$ 
532        $A_{n-2} B_{n-1}, \dots,$ 
533        $A_2 \rightarrow B_1 B_2$ -zerteilt.~
534       Nach-der-Ersetzung-sind-alle-längeren-Nonterminalketten-
535       vollständig-heruntergebrochen-und-die-Chomsky-Normalform-erreicht.
536     }
537   }

```

```

538 }
539 \def\liChomskyErklaerung#1{
540   {
541     \itshape
542     \footnotesize
543     \liParagraphMitLinien{\li@chomsky@erklaerung@texte{#1}}
544   }
545 }

```

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

```

546 \def\liChomskyUeberErklaerung#1{
547   \liChomskyUeberschrift{#1}\par
548   \liChomskyErklaerung{#1}
549 }

```

```

550 \ExplSyntaxOff
551

```

2.9 cpm.sty

```

552 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
553 \ProvidesPackage{lehramt-informatik-cpm}[2020/09/03]
554 \RequirePackage{tikz}
555 \liLadePakete{mathe}

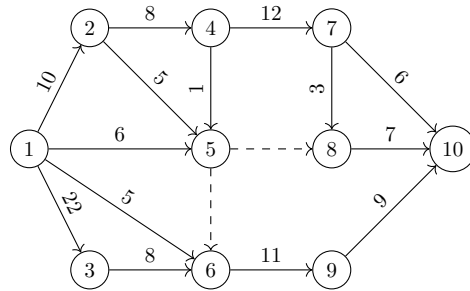
```

2.9.1 Makro-Kürzel

```

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

```



```

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

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

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

```

```

\begin{tabular}{|l|l|l|l|l|l|l|l|}
\hline
\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}

```

```

556 \ExplSyntaxOn
557 \NewDocumentCommand { \liCpmEreignis } { 0{} m m m } {
558   \tl_set:Nn \l_name_tl {}
559
560   \keys_define:nn { cpmEreignis } {
561     name .code:n = {\tl_set:Nn \l_name_tl {##1}},
562   }
563
564   \keys_set:nn { cpmEreignis } { #1 }
565
566   \tl_if_empty:NT \l_name_tl {
567     \tl_set:Nn \l_name_tl {#2}
568   }

```

```

569
570 \node[circle,draw] (\l_name_tl) at (#3,#4) {#2};
571 }
572 \ExplSyntaxOff

\liCpmVorgang liCpmVORGANG\((.*)>(.*))\{(.*)\}
573 \ExplSyntaxOn
574 \NewDocumentCommand { \liCpmVorgang } { 0{ } m m m } {
575   \tl_set:Nn \l_schein_tl {}
576   \tl_set:Nn \l_kritisch_tl {}
577
578   \keys_define:nn { cpmVorgang } {
579     schein .code:n = {\tl_set:Nn \l_schein_tl {dashed}},
580     kritisch .code:n = {\tl_set:Nn \l_kritisch_tl {thick}},
581   }
582
583   \keys_set:nn { cpmVorgang } { #1 }
584
585   \path[->,\l_schein_tl,\l_kritisch_tl] (#2) edge node[auto,sloped] {#4} (#3);
586 }
587 \ExplSyntaxOff

```

2.9.2 Hilfsmakros für Zeitpunkt-Tabelle

```

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

```

```

\liCpmVonZu Let-Abkürzung: \let\vz=\liCpmVonZu
\liCpmVonZu{1}(2-3):  $l_{(2 \rightarrow 3)}$ 
588 \def\liCpmVonZuOhneMathe#1(#2-#3){#1_{\scriptscriptstyle(#2\rightarrow#3)}}
589 \def\liCpmVonZu#1(#2-#3){%
590   \ifmmode%
591     \liCpmVonZuOhneMathe{#1}(\#2-\#3)%
592   \else%
593     $\liCpmVonZuOhneMathe{#1}(\#2-\#3)$%
594   \fi%
595 }

```

```

\liCpmVon Let-Abkürzung: \let\v=\liCpmVon
\liCpmVon{1}(2):  $l_{(\rightarrow 2)}$ 
596 \def\liCpmVonOhneMathe#1(#2){#1_{\scriptscriptstyle(\rightarrow#2)}}
597 \def\liCpmVon#1(#2){%
598   \ifmmode%
599     \liCpmVonOhneMathe{#1}(\#2)%
600   \else%
601     $\liCpmVonOhneMathe{#1}(\#2)$%
602   \fi%
603 }

```

```

\liCpmZu Let-Abkürzung: \let\z=\liCpmZu
\liCpmZu{1}(2):  $l_{(\leftarrow 2)}$ 
604 \def\liCpmZuOhneMathe#1(#2){#1_{\scriptscriptstyle(\leftarrow#2)}}
605 \def\liCpmZu#1(#2){%
606   \ifmmode%
607     \liCpmZuOhneMathe{#1}(\#2)%
608   \else%
609     $\liCpmZuOhneMathe{#1}(\#2)$%

```

```

610 \fi%
611 }

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

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

614

```


2.10 cyk-algorithmus.sty

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

2.10.1 Makro-Kürzel

```
\let\l=\liKurzeTabellenLinie
```

2.10.2 TeX-Markup-Beispiel

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

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

```
\liKurzeTabellenLinie Let-Abkürzung: \let\l=\liKurzeTabellenLinie
```

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

```
\liWortInSprache \liWortInSprache{abc}:  $\Rightarrow abc \in L(G)$ 
```

```
\liWortInSprache{abc}[L(Z)]:  $\Rightarrow abc \in L(Z)$ 
```

```
619 \NewDocumentCommand{ \liWortInSprache } { m O{L(G)} } {
620   \bigskip
621   \noindent
622   $\Rrightarrow$ #1 \in #2$
623 }
```

```
\liWortNichtInSprache \liWortNichtInSprache{abc}:  $\Rightarrow abc \notin L(G)$ 
```

```
\liWortNichtInSprache{abc}[L(Z)]:  $\Rightarrow abc \notin L(Z)$ 
```

```
624 \NewDocumentCommand{ \liWortNichtInSprache } { m O{L(G)} } {
625   \bigskip
626   \noindent
627   $\Rrightarrow$ #1 \notin #2$
628 }
```

```
629
```

2.11 entwurfsmuster.sty

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

2.11.1 Namensschema der Entwurfsmuster-Makros:

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

2.11.2 Reihenfolge

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

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

\li@EntwurfsCodeAllgemein Allgemeine Code-Beispiele zu den UML-Diagrammen und Stellvertretern

```
634 \RequirePackage{lehramt-informatik-syntax}
635 \def\li@EntwurfsCode#1#2{
636   \liJavaDatei{entwurfsmuster/#1/allgemein/#2}
637 }
```

2.11.3 Abstrakte Fabrik (Abstract Factory)

sAbstrakteFabrikBeschreibung

```
638 \def\liEntwurfsAbstrakteFabrikBeschreibung{
639   Es wird eine Schnittstelle bereitgestellt, um \emph{Familien
640   verbundener oder abhängiger Objekte} zu erstellen, ohne die konkreten
641   Klassen zu spezifizieren.\footcite[Seite 25]{eilebrecht}
642 }
```



```

680 \umluniassoc[anchor2=25]{Klient}{AbstrakteFabrik}
681 \umlVHuniassoc[arm1=-1cm]{Klient}{AbstraktesProduktA}
682 \umlVHuniassoc{Klient}{AbstraktesProduktB}
683 \end{tikzpicture}
684 }

```

iEntwurfsAbstrakteFabrikCode

```

685 \def\liEntwurfsAbstrakteFabrikCode{
686 \li@EntwurfsCode{abstrakte_fabrik}{Produkte}
687 \li@EntwurfsCode{abstrakte_fabrik}{AbstrakteFabrik}
688 \li@EntwurfsCode{abstrakte_fabrik}{Klient}
689 }

```

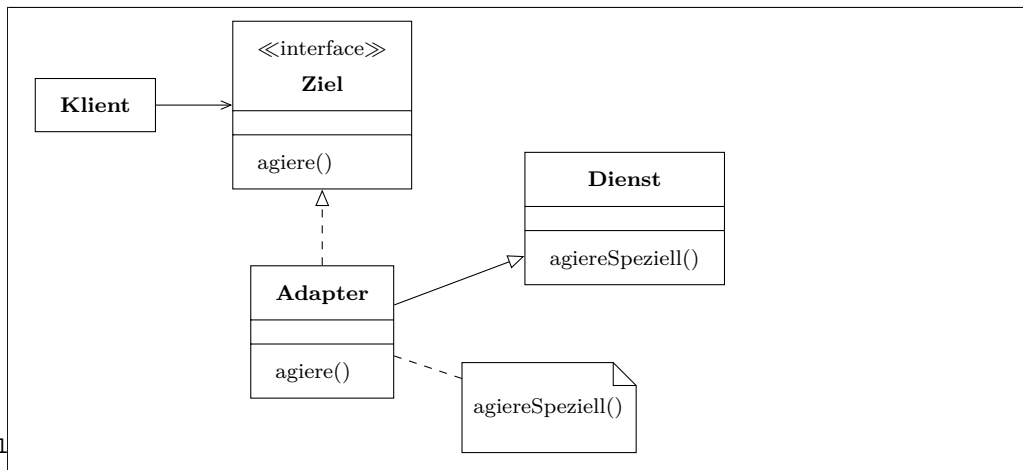
\liEntwurfsAbstrakteFabrik

```

690 \def\liEntwurfsAbstrakteFabrik{
691 \liEntwurfsAbstrakteFabrikBeschreibung
692
693 \liEntwurfsAbstrakteFabrikUml
694
695 \liEntwurfsAbstrakteFabrikCode
696 }

```

2.11.4 Adapter



\liEntwurfsAdapterUml

```

697 \def\liEntwurfsAdapterUml{
698 \begin{tikzpicture}
699 \umlsimpleclass[x=1,y=3]{Klient}{}{}
700 \umlclass[x=4,y=3,type=interface]{Ziel}{}{agiere()}
701 \umlclass[x=4,y=0]{Adapter}{}{agiere()}
702 \umlclass[x=8,y=1.5]{Dienst}{}{agiereSpeziell()}
703
704 \umlreal{Adapter}{Ziel}
705 \umluniassoc{Klient}{Ziel}
706 \umlinherit{Adapter}{Dienst}
707
708 \umlnote[x=7,y=-1,width=2cm]{Adapter}{agiereSpeziell()}
709 \end{tikzpicture}
710 \footcite[so ähnlich wie GoF]{wiki:adapter}
711 }

```

\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.

```

712 \def\liEntwurfsAdapterAkteure{
713   \begin{description}
714
715     \item[Ziel (Target)]
716
717     Das Ziel definiert die Schnittstelle, die der Klient nutzen kann.
718
719     \item[Klient (Client)]
720
721     Der Klient nutzt Dienste über inkompatible Schnittstellen und greift
722     dabei auf adaptierte Schnittstellen zurück.
723
724     \item[Dienst (Adaptee)]
725
726     Der Dienst bietet wiederzuverwendende Dienstleistungen mit fest
727     definierter Schnittstelle an.
728
729     \item[Adapter]
730
731     Der Adapter adaptiert die Schnittstelle des Dienstes auf die
732     Schnittstelle zum Klienten.\footcite{wiki:adapter}
733
734   \end{description}
735 }

```

\liEntwurfsAdapterCode

```

736 \def\liEntwurfsAdapterCode{
737   \li@EntwurfsCode{adapter}{Dienst}
738   \li@EntwurfsCode{adapter}{Ziel}
739   \li@EntwurfsCode{adapter}{Adapter}
740   \li@EntwurfsCode{adapter}{Klient}
741 }

```

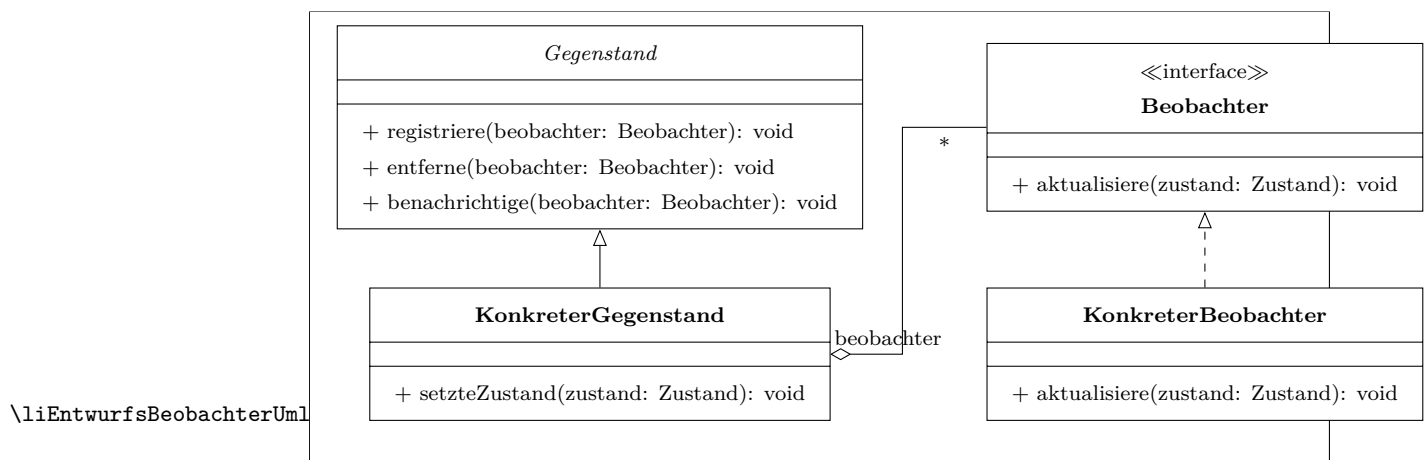
\liEntwurfsAdapter

```

742 \def\liEntwurfsAdapter{
743   \liEntwurfsAdapterUml
744   \liEntwurfsAdapterAkteure
745   \liEntwurfsAdapterCode
746 }

```

2.11.5 Beobachter (Observer)



```

747 \def\liEntwurfsBeobachterUml{
748   \begin{tikzpicture}
749     \umlclass[x=0,y=0,type=abstract]{Gegenstand}{}{

```

```

750     + registriere(beobachter: Beobachter): void\\
751     + entferne(beobachter: Beobachter): void\\
752     + benachrichtige(beobachter: Beobachter): void\\
753 }
754 \umlclass[x=0,y=-3]{KonkreterGegenstand}{-}{
755     + setzeZustand(zustand: Zustand): void
756 }
757 \umlinherit{KonkreterGegenstand}{Gegenstand}
758
759 \umlclass[x=8,y=0,type=interface]{Beobachter}{-}{
760     + aktualisiere(zustand: Zustand): void
761 }
762 \umlclass[x=8,y=-3]{KonkreterBeobachter}{-}{
763     + aktualisiere(zustand: Zustand): void
764 }
765 \umlreal{KonkreterBeobachter}{Beobachter}
766
767 \umlHVHaggreg[arg1=beobachter,pos1=0.8,mult2=*,pos2=2.5]
768 {KonkreterGegenstand}{Beobachter}
769 \end{tikzpicture}
770 }

```

\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.

```

771 \def\liEntwurfsBeobachterAkteure{
772   \begin{description}
773     \item[Gegenstand / Subjekt (Subject / Observable)]
774
775     Ein Subjekt (beobachtbares Objekt, auf Englisch publisher, also
776     „Veröffentlicher“, genannt) hat eine Liste von Beobachtern, ohne
777     deren konkrete Typen zu kennen. Es bietet eine Schnittstelle zur An-
778     und Abmeldung von Beobachtern und eine Schnittstelle zur
779     Benachrichtigung von Beobachtern über Änderungen an.\footcite[Seite
780     251]{gof}
781
782     \item[Beobachter (Observer)]
783
784     Die Beobachter (auf Englisch auch subscriber, also „Abonnent“,
785     genannt) definieren eine Aktualisierungsschnittstelle.
786
787     \item[konkreter/s Gegenstand / Subjekt (ConcreteSubject / ConcreteObservable)]
788
789     Ein konkretes Subjekt (konkretes, beobachtbares Objekt) speichert
790     den relevanten Zustand und benachrichtigt alle Beobachter bei

```

```

791 Zustandsänderungen über deren Aktualisierungsschnittstelle. Es
792 verfügt über eine Schnittstelle zur Erfragung des aktuellen
793 Zustands.
794
795 \item[Konkrete Beobachter (ConcreteObserver)]
796
797 Konkrete Beobachter verwalten die Referenz auf ein konkretes
798 Subjekt, dessen Zustand sie beobachten und speichern und dessen
799 Zustand konsistent ist. Sie implementieren eine
800 Aktualisierungsschnittstelle unter Verwendung der
801 Abfrageschnittstelle des konkreten Subjekts.
802 \footcite{wiki:beobachter}
803 \end{description}
804 }

```

\liEntwurfsBeobachterCode

```

805 \def\liEntwurfsBeobachterCode{
806 \li@EntwurfsCode{beobachter}{Gegenstand}
807 \li@EntwurfsCode{beobachter}{KonkreterGegenstand}
808 \li@EntwurfsCode{beobachter}{Beobachter}
809 \li@EntwurfsCode{beobachter}{KonkreterBeobachterA}
810 \li@EntwurfsCode{beobachter}{KonkreterBeobachterB}
811 \li@EntwurfsCode{beobachter}{Klient}
812 }

```

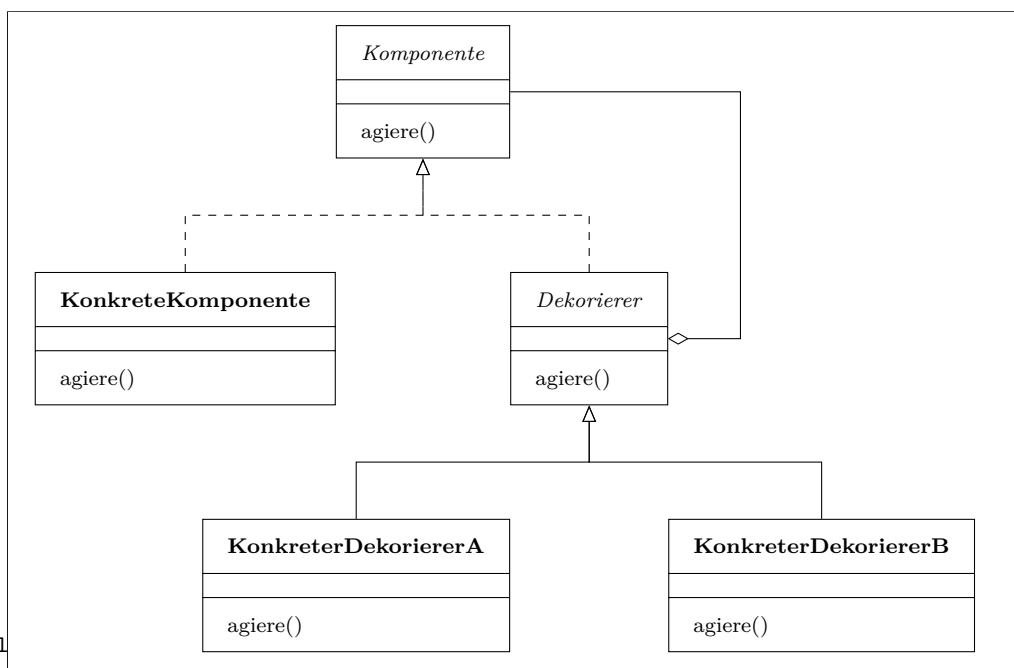
\liEntwurfsBeobachter

```

813 \def\liEntwurfsBeobachter{
814 \liEntwurfsBeobachterUml
815 \liEntwurfsBeobachterAkteure
816 \liEntwurfsBeobachterCode
817 }

```

2.11.6 Dekorierer (Decorator)



\liEntwurfsDekoriererUml

```

818 \def\liEntwurfsDekoriererUml{
819 \begin{tikzpicture}
820 \umlclass[type=abstract]{Komponente}{}{agiere()}
821 \umlclass[below left=1.5cm and 0cm of Komponente]{KonkreteKomponente}{}{agiere()}
822 \umlclass[below right=1.5cm and 0cm of Komponente,type=abstract]{Dekorierer}{}{agiere()}
823

```

```

824 \umlVHVreal{KonkreteKomponente}{Komponente}
825 \umlVHVreal{Dekorierer}{Komponente}
826
827 \umlclass[below left=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererA}{-}{agiere()}
828 \umlclass[below right=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererB}{-}{agiere()}
829
830 \umlVHVinherit{KonkreterDekoriererA}{Dekorierer}
831 \umlVHVinherit{KonkreterDekoriererB}{Dekorierer}
832
833 \umlHVHaggreg[arm1=2cm]{Dekorierer}{Komponente}
834 \footcite{wiki:dekorierer}
835 \end{tikzpicture}
836 }

```

\liEntwurfsDekoriererCode

```

837 \def\liEntwurfsDekoriererCode{
838 \li@EntwurfsCode{dekorierer}{Komponente}
839 \li@EntwurfsCode{dekorierer}{KonkreteKomponente}
840 \li@EntwurfsCode{dekorierer}{Dekorierer}
841 \li@EntwurfsCode{dekorierer}{KonkreterDekoriererA}
842 \li@EntwurfsCode{dekorierer}{KonkreterDekoriererB}
843 \li@EntwurfsCode{dekorierer}{Klient}
844 }

```

\liEntwurfsDekorierer

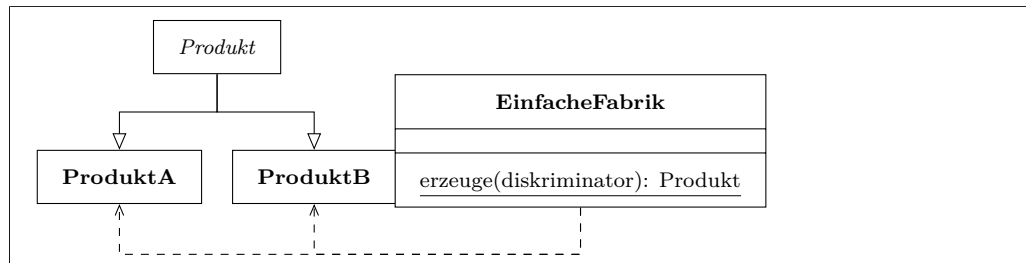
```

845 \def\liEntwurfsDekorierer{
846 \liEntwurfsDekoriererUml
847 \liEntwurfsDekoriererAkteure
848 \liEntwurfsDekoriererCode
849 }

```

2.11.7 Einfache Fabrik (Simple Factory)

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



```

850 \def\liEntwurfsEinfacheFabrikUml{
851 \begin{tikzpicture}
852 \umlsimpleclass[type=abstract]{Produkt}
853 \umlsimpleclass[below left=1cm and -0.65cm of Produkt]{ProduktA}
854 \umlsimpleclass[below right=1cm and -0.65cm of Produkt]{ProduktB}
855 \umlVHVinherit{Produkt}{ProduktA}
856 \umlVHVinherit{Produkt}{ProduktB}
857 \umlclass[below right=0cm and 1.5cm of Produkt]{EinfacheFabrik}{
858 }{
859 \umlstatic{erzeuge(diskriminator): Produkt}}\
860 }
861 \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktA}
862 \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktB}
863 \end{tikzpicture}
864 }

```

ntwurfsEinfacheFabrikAkteure Quelle: frei überstetzt aus GoF

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

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

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

```
865 \def\liEntwurfsEinfacheFabrikAkteure{
866   \begin{description}
867     \item[EinfacheFabrik]
868
869     Eine Klasse mit einer Erzeugungsmethode, die über eine größere
870     Bedingung verschiedene Objekt instanziert.
871
872     \item[Produkt]
873
874     Eine abstrakte Klasse, die von den konkreten Produkten geerbt wird.
875
876     \item[KonkretesProdukt]
877
878     Ein konkretes Produkt, das von der einfachen Fabrik erzeugt wird.
879   \end{description}
880 }
```

\liEntwurfsEinfacheFabrik

```
881 \def\liEntwurfsEinfacheFabrik{
882   \liEntwurfsEinfacheFabrikUml
883   \liEntwurfsEinfacheFabrikAkteure
884 }
```

2.11.8 Einzelstück (Singleton)

urfsEinzelstueckBeschreibung

```
885 \def\liEntwurfsEinzelstueckBeschreibung{
886   Stellt sicher, dass nur \emph{genau eine Instanz einer Klasse} erzeugt
887   wird.\footcite[Seite 38]{eilebrecht}
888 }
889
```

\liEntwurfsEinzelstueckUml Quelle nach der deutschen Wikipedia

Einzelstück
<u>- instanz: Einzelstück</u>
- Einzelstück() + gibInstanz(): Einzelstück

```
890 \def\liEntwurfsEinzelstueckUml{
891   \begin{tikzpicture}
892     \umlclass{Einzelstück}{
893       \umlstatic{- instanz: Einzelstück}\\
894     }{
895       - Einzelstück()\\
896       + gibInstanz(): Einzelstück
897     }
898   \end{tikzpicture}
899 }
```

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.

```

900 \def\liEntwurfsEinzelstueckAkteure{
901   \begin{description}
902     \item[Einzelstück (Singleton)]
903
904     stellt eine statische Methode bereit, mit deren Hilfe die Klienten
905     nur auf eine einzige Instanz der Klasse zugreifen können.
906   \end{description}
907 }

```

\liEntwurfsEinzelstueckCode

```

908 \def\liEntwurfsEinzelstueckCode{
909   \li@EntwurfsCode{einzelstueck}{Einzelstueck}
910 }

```

\liEntwurfsEinzelstueck

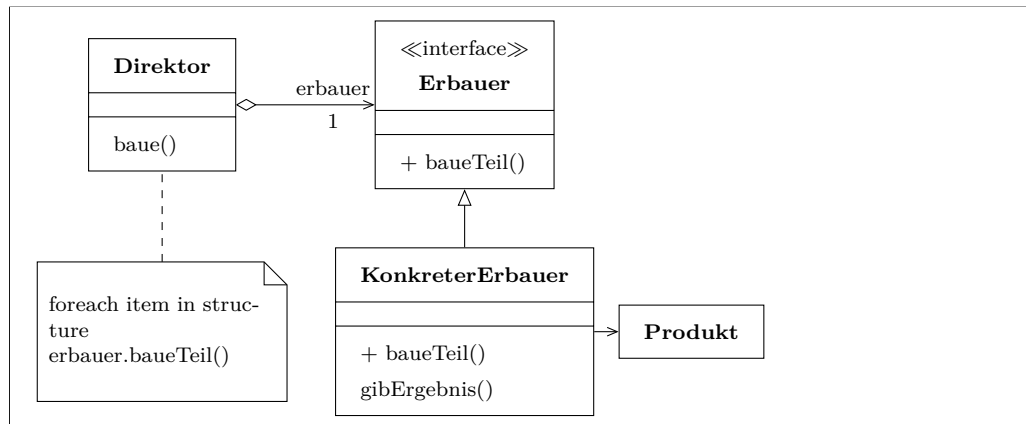
```

911 \def\liEntwurfsEinzelstueck{
912   \liEntwurfsEinzelstueckBeschreibung
913
914   \liEntwurfsEinzelstueckUml
915
916   \liEntwurfsEinzelstueckAkteure
917
918   \liEntwurfsEinzelstueckCode
919 }

```

2.11.9 Erbauer (Builder)

\liEntwurfsErbauerUml Quelle nach der deutschen Wikipedia



```

920 \def\liEntwurfsErbauerUml{
921   \begin{tikzpicture}
922     \umlsimpleclass[x=7,y=0]{Produkt}{}{}
923     \umlclass[x=4,y=3,type=interface]{Erbauer}{}{+ baueTeil()}
924     \umlclass[x=4,y=0]{KonkreterErbauer}{}{
925       + baueTeil()\n
926       gibErgebnis()}
927     \umlclass[x=0,y=3]{Direktor}{}{baue()}
928
929     \umluniagg[arg2=erbauer,pos2=0.7,mult2=1]{Direktor}{Erbauer}
930     \umluniassoc{KonkreterErbauer}{Produkt}
931     \umlinherit{KonkreterErbauer}{Erbauer}
932
933     \umlnote[x=0,y=0,width=3cm]{Direktor}{
934       foreach item in structure\n
935       erbauer.baueTeil()

```

```

936 }
937 \end{tikzpicture}
938 \footcite{wiki:erbauer}
939 }

```

\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.

```

940 \def\liEntwurfsErbauerAkteure{
941   \begin{description}
942     \item[Erbauer]
943
944     Der Erbauer spezifiziert eine abstrakte Schnittstelle zur Erzeugung der
945     Teile eines komplexen Objektes.
946
947     \item[KonkreterErbauer]
948
949     Der konkrete Erbauer erzeugt die Teile des komplexen Objekts durch
950     Implementierung der Schnittstelle. Außerdem definiert und verwaltet er
951     die von ihm erzeugte Repräsentation des Produkts. Er bietet auch eine
952     Schnittstelle zum Auslesen des Produkts.
953
954     \item[Direktor]
955
956     Der Direktor konstruiert ein komplexes Objekt unter Verwendung der
957     Schnittstelle des Erbauers. Der Direktor arbeitet eng mit dem Erbauer
958     zusammen: Er weiß, welche Baureihenfolge der Erbauer verträgt oder
959     benötigt. Der Direktor entkoppelt somit den Konstruktionsablauf vom
960     Klienten.
961
962     \item[Produkt]
963
964     Das Produkt repräsentiert das zu konstruierende komplexe Objekt.
965     \footcite{wiki:erbauer}
966   \end{description}
967 }

```

\liEntwurfsErbauer

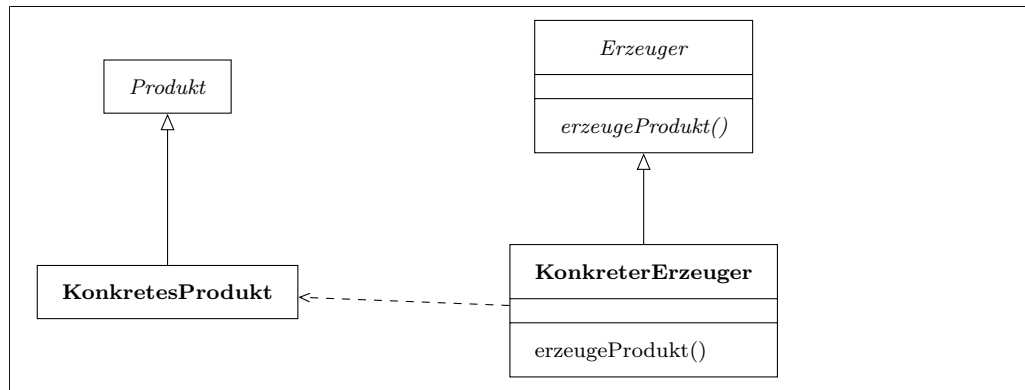
```

968 \def\liEntwurfsErbauer{
969   \liEntwurfsErbauerUml
970   \liEntwurfsErbauerAkteure
971 }

```

2.11.10 Fabrikmethode (Factory Method)

\liEntwurfsFabrikmethodeUml Quelle nach der deutschen Wikipedia



```

972 \def\liEntwurfsFabrikmethodeUml{
973   \begin{tikzpicture}
974     \umlsimpleclass[type=abstract]{Produkt}
975     \umlsimpleclass[below=2cm of Produkt]{KonkretesProdukt}
976     \umlinherit{KonkretesProdukt}{Produkt}
977
978     \umlclass[type=abstract,right=4cm of Produkt]{Erzeuger}{\{
979       \textit{erzeugeProdukt()}\}
980   }
981   \umlclass[below=1.2cm of Erzeuger]{KonkreterErzeuger}{\{
982     erzeugeProdukt()
983   }
984   \umlinherit{KonkreterErzeuger}{Erzeuger}
985
986   \umldep{KonkreterErzeuger}{KonkretesProdukt}
987 \end{tikzpicture}
988 }

```

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).

```

989 \def\liEntwurfsFabrikmethodeAkteure{
990   \begin{description}
991     \item[Produkt]
992
993     Das Produkt ist der Basistyp (Klasse oder Schnittstelle) für das
994     zu erzeugende Produkt.
995
996     \item[KonkretesProdukt]
997
998     KonkretesProdukt implementiert die Produkt-Schnittstelle.
999
1000    \item[Erzeuger]
1001
1002    Der Erzeuger deklariert die Fabrikmethode, um ein solches Produkt
1003    zu erzeugen und kann eine Default-Implementierung beinhalten.
1004
1005    \item[KonkreterErzeuger]
1006

```

```

1007     KonkreterErzeuger überschreibt die Fabrikmethode, um die ihm
1008     entsprechenden konkreten Produkte zu erzeugen (z. B. indem er den
1009     Konstruktor einer konkreten Produkt-Klasse aufruft).
1010
1011     \footcite{wiki:fabrikmethode}
1012 \end{description}
1013 }

```

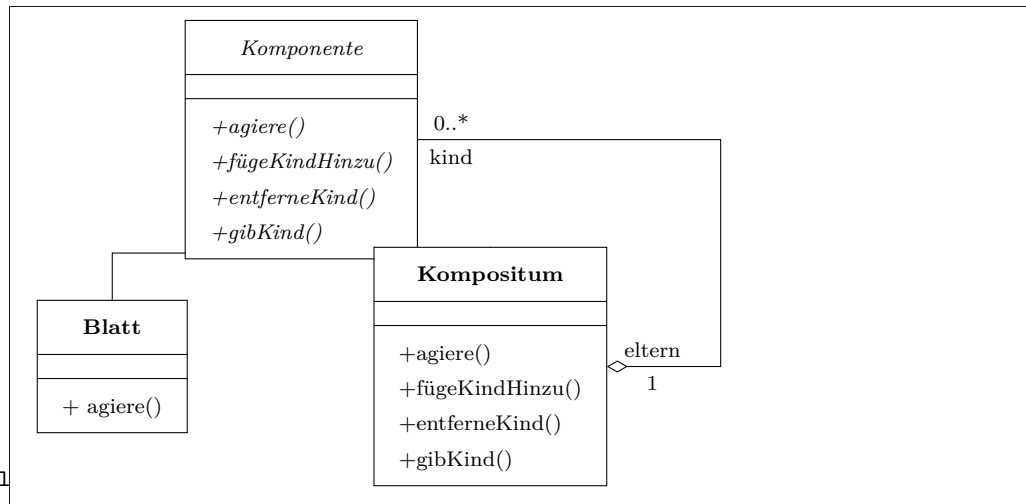
\liEntwurfsFabrikmethode

```

1014 \def\liEntwurfsFabrikmethode{
1015     \liEntwurfsFabrikmethodeUml
1016     \liEntwurfsFabrikmethodeAkteure
1017 }

```

2.11.11 Kompositum (Composite)



\liEntwurfsKompositumUml

```

1018 \def\liEntwurfsKompositumUml{
1019     \begin{tikzpicture}
1020         \umlclass[x=2.5,y=3,type=abstract]{Komponente}{-}{
1021             \textit{+agiere()}\}
1022             \textit{+fügeKindHinzu()}\}
1023             \textit{+entferneKind()}\}
1024             \textit{+gibKind()}\}
1025         }
1026         \umlclass[x=0]{Blatt}{-}{+ agiere()}\}
1027         \umlclass[x=5]{Kompositum}{-}{
1028             +agiere()\}
1029             +fügeKindHinzu()\}
1030             +entferneKind()\}
1031             +gibKind()}\}
1032     }
1033
1034     \umlVHVinherit{Kompositum}{Komponente}
1035     \umlVHVinherit{Blatt}{Komponente}
1036     \umlHVVHaggreg[anchor1=east,arm1=1.5cm,arg1=eltern,mult1=1,arg2=kind,mult2=0..*,pos2=2.9,]
1037 \end{tikzpicture}
1038 }

```

\liEntwurfsFabrikmethode

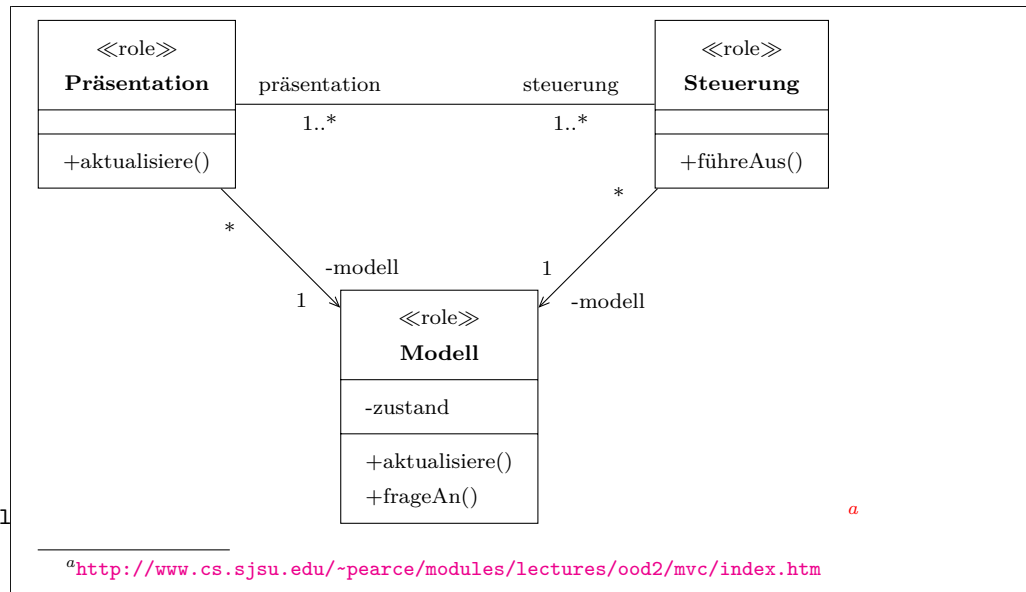
```

1039 \def\liEntwurfsKompositum{
1040     \liEntwurfsKompositumUml
1041     \liEntwurfsKompositumAkteure
1042 }

```

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

ModellPraesentationSteuerungUml



```

1043 \def\liEntwurfsModellPraesentationSteuerungUml{
1044   \begin{tikzpicture}
1045     \umlclass[x=-4cm,type=role]{Präsentation}{+aktualisiere()}
1046     \umlclass[x=4cm,type=role]{Steuerung}{+führeAus()}
1047     \umlclass[y=-4cm,type=role]{Modell}{
1048       -zustand
1049     }{
1050       +aktualisiere()\\
1051       +frageAn()
1052     }
1053
1054     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Präsentation}{Modell}
1055     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Steuerung}{Modell}
1056     \umlassoc[arg1=präsentation,mult1=1..*,arg2=steuerung,mult2=1..*]{Präsentation}{Steuerung}
1057   \end{tikzpicture}
1058   \liFussnoteUrl{http://www.cs.sjsu.edu/~pearce/modules/lectures/ood2/mvc/index.htm}
1059 }

```

ModellPraesentationSteuerung

```

1060 \def\liEntwurfs{
1061   \liEntwurfsModellPraesentationSteuerungUml
1062   \liEntwurfsModellPraesentationSteuerungAkteure
1063 }

```

2.11.13 Stellvertreter (Proxy)

\liEntwurfsStellvertreterUml

```

1064 \def\liEntwurfsStellvertreterUml{
1065   \begin{tikzpicture}
1066     \umlsimpleclass[x=-1,y=2]{Klient}
1067
1068     \umlclass[x=2,y=2]{Subjekt}{+ agiere()}
1069     \umlclass[x=0,y=-1]{KonkretesSubjekt}{+ agiere()}
1070     \umlclass[x=4,y=-1]{Stellvertreter}{+ agiere()}
1071
1072     \umlVHVinherit{KonkretesSubjekt}{Subjekt}
1073     \umlVHVinherit{Stellvertreter}{Subjekt}
1074     \umluniassoc{Stellvertreter}{KonkretesSubjekt}
1075     \umluniassoc{Klient}{Subjekt}
1076   \end{tikzpicture}
1077 }

```

\liEntwurfsStellvertreterCode

```

1078 \def\liEntwurfsStellvertreterCode{
1079   \li@EntwurfsCode{stellvertreter}{Subjekt}
1080   \li@EntwurfsCode{stellvertreter}{KonkretesSubjekt}
1081   \li@EntwurfsCode{stellvertreter}{Stellvertreter}
1082   \li@EntwurfsCode{stellvertreter}{Klient}
1083 }

```

\liEntwurfsStellvertreter

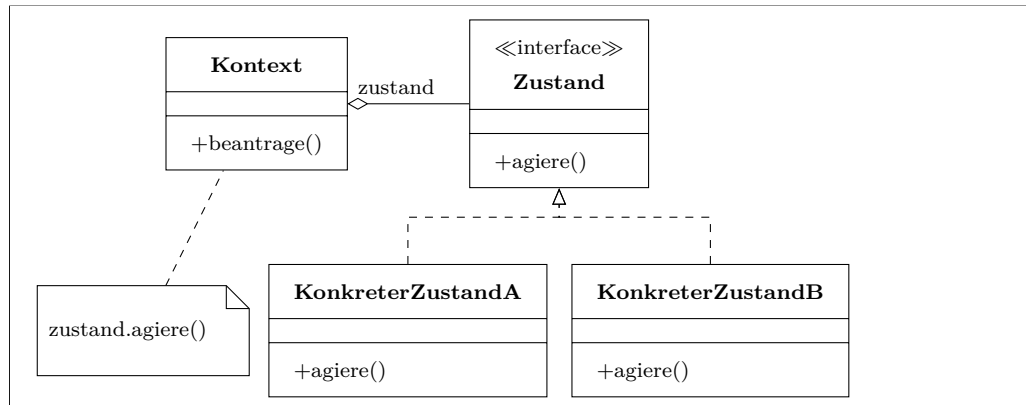
```

1084 \def\liEntwurfsStellvertreter{
1085   \liEntwurfsStellvertreterUml
1086   \liEntwurfsStellvertreterCode
1087 }

```

2.11.14 Zustand (State)

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



```

1088 \def\liEntwurfsZustandUml{
1089   \begin{tikzpicture}
1090     \umlclass[x=-1,y=3]{Kontext}{{+beantrage()}}
1091     \umlclass[x=3,y=3,type=interface]{Zustand}{{+agiere()}}
1092     \umlclass[x=1,y=0]{KonkreterZustandA}{{+agiere()}}
1093     \umlclass[x=5,y=0]{KonkreterZustandB}{{+agiere()}}
1094
1095     \umlVHVreal{KonkreterZustandA}{Zustand}
1096     \umlVHVreal{KonkreterZustandB}{Zustand}
1097
1098     \umlagg[arg=zustand,pos=0.4]{Kontext}{Zustand}
1099
1100     \umlnote[x=-2.5,y=0,width=2.5cm]{Kontext}{zustand.agiere()}
1101   \end{tikzpicture}
1102 }

```

\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.

```

1103 \def\liEntwurfsZustandAkteure{
1104   \begin{description}
1105     \item[Kontext (Context)]
1106
1107     definiert die clientseitige Schnittstelle und verwaltet die separaten

```

```

1108     Zustandsklassen.
1109
1110     \item[State (Zustand)]
1111
1112     definiert eine einheitliche Schnittstelle aller Zustandsobjekte und
1113     implementiert gegebenenfalls ein Standardverhalten.
1114
1115     \item[KontreterZustand (ConcreteState)]
1116
1117     implementiert das Verhalten, das mit dem Zustand des Kontextobjektes
1118     verbunden ist.
1119 \end{description}
1120 }

```

\liEntwurfsZustandCode

```

1121 \def\liEntwurfsZustandCode{
1122   \li@EntwurfsCode{zustand}{Kontext}
1123   \li@EntwurfsCode{zustand}{Zustand}
1124 }

```

\liEntwurfsZustand

```

1125 \def\liEntwurfsZustand{
1126   \liEntwurfsZustandUml
1127   \liEntwurfsZustandAkteure
1128   \liEntwurfsZustandCode
1129 }

```

1130

2.12 er.sty

```
1131 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1132 \ProvidesPackage{lehramt-informatik-er}[2020/06/13 Zum Setzen von
1133 ER-Diagrammen]
```

```
1134 \RequirePackage{tikz-er2}
1135 \usetikzlibrary{positioning}
```

2.12.1 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}
```

2.12.2 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);
```

```
1136 \RequirePackage{soul}
```

```
1137 \RequirePackage{fontawesome}
```

2.12.3 Makro-Kürzel

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

```
1138 \ExplSyntaxOn
```

```
\liErEntity
```

```
1139 \def\liErEntity#1{\textbf{#1}}
```

```
\liErRelationship
```

```
1140 \def\liErRelationship#1{\ul{#1}}
```

```
\liErAttribute
```

```
1141 \def\liErAttribute#1{\emph{#1}}
```

```
\liErMpEntity
```

```
mp = marginpar
Let-Abkürzung: \let\e=\liErMpEntity
```

```
1142 \def\liErMpEntity#1{
1143   \liErEntity{#1}
1144   \marginpar{
1145     \liErEntity{\tiny\faSquareO{~E:~#1}
1146   }
1147 }
```

□

```
\liErMpRelationship
```

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

```
1148 \def\liErMpRelationship#1{
1149   \liErRelationship{#1}
1150   \marginpar{
1151     \liErRelationship{\tiny\faGg{~R:~#1}
1152   }
1153 }
```

```
\liErMpAttribute
```

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

```
1154 \def\liErMpAttribute#1{
1155   \liErAttribute{#1}
1156   \marginpar{
1157     \liErAttribute{\tiny\faCircleThin{~A:~#1}
1158   }
1159 }
```

```

\liErDatenbankName Let-Abkürzung: \let\d=\liErDatenbankName
    datenbank name
1160 \def\liErDatenbankName#1{
1161   {
1162     \footnotesize\texttt{(#1)}
1163   }
1164 }

1165 \ExplSyntaxOff
1166

```

2.13 formale-sprachen.sty

```

1167 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1168 \ProvidesPackage{lehramt-informatik-formale-sprachen}[2021/02/21 Hilfsmakros
1169 zum Setzen von mathematischen Formeln bei dem Thema Formale Sprachen]

1170 \directlua{
1171   formale_sprachen = require('lehramt-informatik-formale-sprachen')
1172 }

1173 \RequirePackage{hyperref}

1174 \liLadePakete{mathe,typographie}

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

1175 \def\liMengeOhneMathe#1{\{ #1 \}}
1176 \def\liMenge#1{%
1177   \ifmmode%
1178     \liMengeOhneMathe{#1}%
1179   \else%
1180     $\liMengeOhneMathe{#1}$%
1181   \fi%
1182 }

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

1183 \def\liEpsilon{$\varepsilon$}

\liPotenzmenge Umgeben mit geschweiften Klammern in einer Mathematik-Umgebung

1184 \def\erzeuge@tiefgestellt#1{\directlua{formale_sprachen.erzeuge_tiefgestellt('#1')}}
1185 \def\liPotenzmengeOhneMathe#1{\{ \erzeuge@tiefgestellt{#1} \}}
1186 \def\liPotenzmenge#1{$\liPotenzmengeOhneMathe{#1}$}

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

1187 \let\liZustandsmengeOhneMathe=\liPotenzmengeOhneMathe
1188 \let\liZustandsmenge=\liPotenzmenge

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

1189 \def\liUeberfuehrungsFunktionOhneMathe#1{\delta(\erzeuge@tiefgestellt{#1})}
1190 \def\liUeberfuehrungsFunktion#1{
1191   \ifmmode
1192     \liUeberfuehrungsFunktionOhneMathe{#1}
1193   \else
1194     $\liUeberfuehrungsFunktionOhneMathe{#1}$
1195   \fi
1196 }

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

1197 \def\liAlphabet#1{$\Sigma = \{ #1 \}$}

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

1198 \def\liBandAlphabet#1{$\Gamma = \Sigma \cup \{ #1 \}$}

\liZustandsBuchstabe

1199 \def\liZustandsBuchstabe{z}

\liZustandsBuchstabeGross

1200 \def\liZustandsBuchstabeGross{Z}

```

```

\liZustandsmengeNr
1201 \def\zustandsnamens@liste#1#2{
1202   $
1203   \{
1204     \directlua{formale_sprachen.erzeuge_zustandsnamens_liste('#1', '#2')}
1205   \}
1206   $
1207 }
1208 \def\liZustandsmengeNr#1{\zustandsnamens@liste{\liZustandsBuchstabe}{#1}}

\liZustandsmengeNrGross
1209 \def\liZustandsmengeNrGross#1{\zustandsnamens@liste{\liZustandsBuchstabeGross}{#1}}

\liZustandsname \liZustandsname{1}: $z_1$
1210 \def\liZustandsname#1{\liZustandsBuchstabe_#1$}

\liZustandsnameGross \liZustandsnameGross{1}: $Z_1$
1211 \def\liZustandsnameGross#1{\liZustandsBuchstabeGross_#1$}

\liAbleitung \liAbleitung{S -> aB -> ab}:  $S \vdash aB \vdash ab$ 
1212 \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}

1213 \NewDocumentEnvironment { liProduktionsRegeln }
1214 { 0{P} +b }
1215 {
1216   \liGeschweifteKlammern{#1}
1217   {
1218     \begin{align*}
1219       \directlua{formale_sprachen.produktions_regeln('#2')}
1220     \end{align*}
1221   }{-0.2cm}{-1.5cm}
1222 } {}

\liProduktionen \liProduktionen{S -> A, A -> a}:  $\{S \rightarrow A, A \rightarrow a\}$ 
1223 \def\liProduktionen#1{
1224   \liMenge{\directlua{formale_sprachen.produktions_regeln('#1', true)}}
1225 }

\liZustandsnameTiefgestellt Automatisch tiefgestellte Nummerierung \z1
Let-Abkürzung: \let\z=\liZustandsnameTiefgestellt
1226 \def\liZustandsnameTiefgestellt#1{
1227   \ifmmode
1228     \liZustandsBuchstabe\sb{#1}
1229   \else
1230     $\liZustandsBuchstabe\sb{#1}$
1231   \fi
1232 }

1233 \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}

```

```

1234 \NewDocumentCommand{ \liAusdruck } { O{L} m m } {
1235   $
1236   \tl_if_empty:nTF {#1} {} {#1 =}
1237   \{
1238     \, #2 \,
1239     |
1240     \, #3 \,
1241   \}$
1242 }
1243 \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

1244 \def\liFlaci#1{%
1245   \par
1246   {%
1247     \scriptsize
1248     Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte
1249     Automaten, Compiler und Interpreter) Ein Projekt der Hochschule
1250     Zittau/Görlitz und der Pädagogischen Hochschule Schwyz:
1251     \href{https://flaci.com/#1}{flaci.com/#1}%
1252   }%
1253   \par
1254 }

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

1255 \ExplSyntaxOn
1256 \NewDocumentCommand {\liGrammatik} { O{G} m } {
1257   \tl_set:Nn \l_variablen_tl {V}
1258   \tl_set:Nn \l_alphabet_tl {\Sigma}
1259   \tl_set:Nn \l_produktionen_tl {P}
1260   \tl_set:Nn \l_start_tl {S}
1261
1262   \keys_define:nn { grammatik } {
1263     variablen .code:n = {\tl_set:Nn \l_variablen_tl {\liMenge{##1}}},
1264     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
1265     produktionen .code:n = {\tl_set:Nn \l_produktionen_tl {\liProduktionen{##1}}},
1266     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
1267   }
1268
1269   \keys_set:nn { grammatik } { #2 }
1270
1271   $#1 = (
1272     \l_variablen_tl,
1273     \l_alphabet_tl,
1274     \l_produktionen_tl,
1275     \l_start_tl
1276   )$
1277 }
1278 \ExplSyntaxOff
1279

```

2.14 formatierung.sty

```
1280 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1281 \ProvidesPackage{lehramt-informatik-formatierung}[2020/11/27]
```

2.14.1 Schriftarten / Typographie

```
1282 \RequirePackage{mathpazo}
1283 \RequirePackage[no-math]{fontspec}
1284 \setmainfont{texgyrepagella}
1285 \setsansfont{QTAncientOlive}
1286 \RequirePackage{sectsty}
1287 \allsectionsfont{\fontspec{QTAncientOlive}}
```

2.14.2 Farben

```
1288 \RequirePackage{xcolor}
1289 \definecolor{infogray}{rgb}{0.97,0.97,0.97}
```

2.14.3 Überschriften

```
1290 \RequirePackage{titlesec}
1291 \titleformat{\chapter}[display]{\sffamily\bfseries}{\Opt}{\LARGE}
1292 \titlespacing{\chapter}{0pt}{0pt}{*1}
1293 \titleformat{\paragraph}[hang]{\sffamily\normalsize\bfseries}{\theparagraph}{1em}{}
1294 \setcounter{secnumdepth}{0}
```

2.14.4 Listen

```
1295 \RequirePackage{paralist}
1296 \renewcommand\labelitemi{-}
1297 \renewcommand\labelitemii{-}
1298 \renewcommand\labelitemiii{-}
1299 \renewcommand\labelitemiv{-}
1300 % Counter: enumi enumii enumiii enumiv
1301 % Styles: \arabic{counter} \alph{counter} \Alph{counter} \roman{counter} \Roman{counter}
1302 \renewcommand{\labelenumi}{(\alph{enumi})}
1303 \renewcommand{\labelenumii}{(\roman{enumii})}
```

2.14.5 Kasten

```
1304 \RequirePackage{mdframed}
1305 \mdfsetup{backgroundcolor=infogray}
```

liKasten

```
1306 \NewDocumentEnvironment { liKasten }{ } {
1307   \begin{mdframed}[backgroundcolor=white!0]
1308 } {
1309   \end{mdframed}
1310 }
```

2.14.6 Header

```
1311 \RequirePackage{fancyhdr}
1312 \fancyhead[L,C,R]{}
1313 \fancyfoot[L]{}
1314 \fancyfoot[C]{}
1315 \fancyfoot[R]{\thepage}
1316 \pagestyle{fancy}
1317 \renewcommand{\headrulewidth}{0pt}
1318 \renewcommand{\footrulewidth}{0pt}
1319
```

2.15 gantt.sty

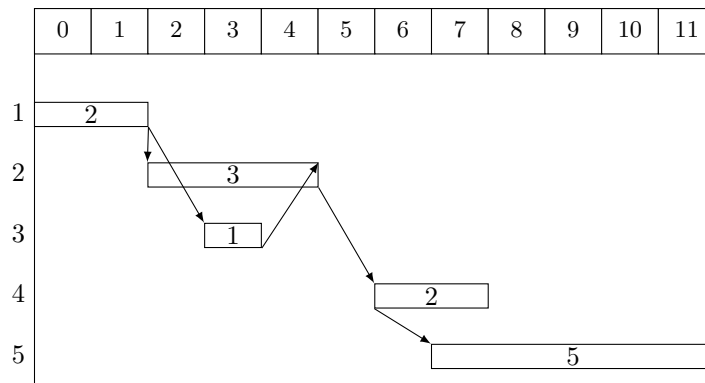
1320 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

1321 \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}
```



```
1322 \RequirePackage{tikz-uml}
1323 \RequirePackage{pgfgantt}
1324 \setganttlinklabel{f-s}{}
1325 \setganttlinklabel{s-s}{}
1326 \setganttlinklabel{f-f}{}
1327 \setganttlinklabel{s-f}{}
1328
```


2.16 grafik.sty

```
1329 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1330 \ProvidesPackage{lehramt-informatik-grafik}[2020/12/27 Lädt das Paket
1331 TikZ, um Grafiken zeichnen zu können.]
1332 \RequirePackage{tikz}
1333
```

2.17 graph.sty

```

1334 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1335 \ProvidesPackage{lehramt-informatik-graph}[2020/06/09]
1336 \RequirePackage{tikz}

```

Für die die Adjazenzliste (xrightarrow)

```

1337 \RequirePackage{amsmath}

```

Für Adjazenz-Matrix

```

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

```

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

```

1338 \RequirePackage{blkarray}
1339 \usetikzlibrary{arrows.meta}

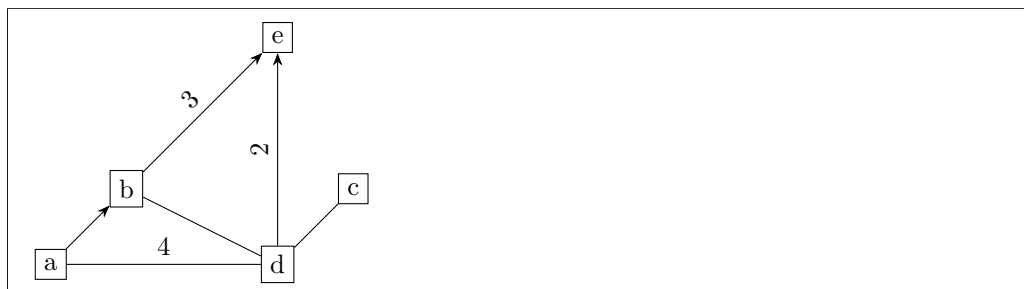
```

```

\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}

```



```

1340 \tikzset{
1341   li graph/.style={
1342     every node/.style={
1343       rectangle,
1344       draw,
1345     },
1346     every edge/.style={
1347       >={Stealth[black]},
1348       draw,
1349     },
1350     every edge/.append style={
1351       every node/.style={
1352         sloped,
1353         auto,
1354       }
1355     }
1356   },
1357   li markierung/.style={
1358     ultra thick,
1359   }
1360 }

```

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}

```

```

1361 \NewDocumentEnvironment { liGraphenFormat }{ +b } {} {}

```

```

1362

```

2.18 hanoi.sty

```
1363 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1364 \ProvidesPackage{lehramt-informatik-hanoi}[2020/12/19 Paket zum Setzen
1365 von Hanoi-Grafiken]
```

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

```
1366 \RequirePackage{tikz}
1367 \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}

1368 \def\li@mset #1[#2]=#3{%
1369 \expandafter\xdef\csname #1#2\endcsname{#3}
1370 }
1371 \def\li@mget #1[#2]{%
1372 \csname #1#2\endcsname
1373 }
1374 \def\li@minc #1[#2]+=#3{%
1375 \pgfmathparse{\li@mget #1[#2]+#3}%
1376 \li@mset #1[#2]=\pgfmathresult
1377 }
1378
1379 \def\liHanoi#1#2{
1380   \edef\li@numdiscs{#1}
1381   \def\li@sequence{#2}
1382   \begin{tikzpicture}[line width=4mm,brown!40,line cap=round,xscale=3]
1383     % init colors
1384     \foreach[count=\j] \c in {red,green!80!black,blue,orange,violet,gray,yellow!80!black,purple!80!black}{
1385       \li@mset col[\j]={\c};
1386     }
1387     % draw poles and init pole counters
1388     \foreach \j in {1,2,3}{
1389       \li@mset pos[\j]=0
1390       \draw (\j,-.5) -- +(0, .5 + .5*\li@numdiscs);
1391     }
1392     % draw base
1393     \draw (.5,-.5) -- +(3,0);
1394     % draw discs
1395     \foreach[count=\k] \i/\j in \li@sequence{
1396       \draw[draw={\li@mget col[\i]}] (\j,\li@mget pos[\j]) +(-.4*\i/\li@numdiscs,0) -- +(.4*\i/\li@numdiscs,0);
1397       \li@minc pos[\j]+=.5
1398     }
1399   \end{tikzpicture}
1400 }
```

2.19 klassen-konfiguration-aufgabe.sty

```
1401 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1402 \ProvidesPackage{lehramt-informatik-klassen-konfiguration-aufgabe}[2021/09/01 Die
1403 Klasse Aufgabe konfigurieren, d. h. Laden von einigen Paketen, Registrierung von Hooks]

    Formatierung muss vor literatur sein, sonst Option clash
1404 \liLadePakete{
1405   formatierung,
1406   abmessung,
1407   literatur-dummy,
1408   makros,
1409   aufgaben-metadaten,
1410   kopf-fusszeilen,
1411   mathe
1412 }
1413
1414 % Komischer Option-Clash deshalb ganz am Ende, für die Silbentrennung
1415 \RequirePackage[ngerman]{babel}
1416 \ExplSyntaxOn
1417 \AddToHook{enddocument}{
1418   \_gib_github_url:
1419 }
1420 \ExplSyntaxOff
1421
```

2.20 klassen-konfiguration-examen.sty

```
1422 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1423 \ProvidesPackage{lehramt-informatik-klassen-konfiguration-examen}[2021/09/04 Die
1424 Klasse liSetzeExamen konfigurieren, d. h. Laden von einigen Paketen, Registrierung von Hooks]
```

Lade die wichtigsten Pakete. Formatierung muss vor literatur sein, sonst Option clash

```
1425 \liLadePakete{
1426   formatierung,
1427   literatur-dummy,
1428   makros,
1429   aufgaben-metadaten,
1430   abmessung,
1431   typographie
1432 }
```

Formatierung für die Überschriften setzen.

```
1433 \RequirePackage{titlesec}
1434 \titleformat{\section}{\sffamily\huge\filcenter\bfseries}{\thesection}{1em}{}
1435 \newcommand{\sectionbreak}{\clearpage}
1436 \setcounter{secnumdepth}{0}
1437 \liLadeAllePakete
```

Komischer Option-Clash deshalb ganz am Ende, für die Silbentrennung

```
1438 \RequirePackage[ngerman]{babel}
1439 \RequirePackage{standalone}
1440 \ExplSyntaxOn
```

`\liSetzeExamenThemaNr`

```
1441 \def\liSetzeExamenThemaNr#1{
1442   \tl_gset:Nn \g_auf_examen_thema_nr_tl { #1 }
1443   \section{Thema-Nr.~#1}
1444 }
```

`\liSetzeExamenTeilaufgabeNr`

```
1445 \def\liSetzeExamenTeilaufgabeNr#1{
1446   \tl_gset:Nn \g_auf_examen_teilaufgabe_nr_tl { #1 }
1447   \subsection{Teilaufgabe-Nr.~#1}
1448 }
```

`\liBindeAufgabeEin`

```
1449 \def\liBindeAufgabeEin#1{
1450   \tl_gset:Nn \g_auf_examen_aufgabe_nr_tl { #1 }
1451   \input{
1452     \LehramtInformatikRepository /
1453     Staatsexamen /
1454     \g_auf_examen_nummer_tl /
1455     \g_auf_examen_jahr_tl /
1456     \g_auf_examen_monat_tl /
1457     \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} {
1458       Thema - \g_auf_examen_thema_nr_tl /
1459     }
1460     \tl_if_empty:NTF \g_auf_examen_teilaufgabe_nr_tl {} {
1461       Teilaufgabe - \g_auf_examen_teilaufgabe_nr_tl /
1462     }
1463     Aufgabe - \g_auf_examen_aufgabe_nr_tl .tex
1464   }
1465 }
```

`\liAufgabenMetadaten` Das Metadaten-Makro überschreiben

```
1466 \def\liAufgabenMetadaten#1{
1467   \liMetaSetze{#1}
1468   \subsubsection{\_gib_aufgaben_titel:}
1469 }
```

```

1470 \setcounter{tocdepth}{4}
1471 \AddToHook{begindocument}{
1472   \begin{center}
1473     \liTypoUeberGross{Erste-Staatsprüfung-für-ein-Lehramt-an-öffentlichen-Schulen}
1474   \end{center}
1475
1476   \vfill
1477
1478   \begin{center}
1479     \liTypoUeberGROSS{\g_auf_examen_nummer_tl}
1480   \end{center}
1481
1482   \begin{center}
1483     \liTypoUeberGROSS{\g_auf_examen_jahr_tl}
1484   \end{center}
1485
1486   \begin{center}
1487     \liTypoUeberGROSS{\g_auf_examen_monat_tl}
1488   \end{center}
1489
1490   \vfill
1491   \tableofcontents
1492   \clearpage
1493 }
1494 \ExplSyntaxOff
1495

```

2.21 komplexitaetstheorie.sty

```
1496 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1497 \ProvidesPackage{lehramt-informatik-komplexitaetstheorie}[2021/07/08 Zum
1498 Setzen von Karp's NP-vollständigen Problemen, Hilfsmakros für die
1499 Polynomialzeitreduktion.]
```

2.21.1 Makro-Kürzel

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

```
1500 \liLadePakete{mathe}
```

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

```
1501 \RequirePackage{mdframed}
```

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

```
1502 \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`

```
1503 \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`

```
1504 \def\liProblemBeschreibung#1#2#3{
1505   \begin{mdframed}[
1506     userdefinedwidth=9cm,
1507     align=center,
1508     backgroundcolor=white!0,
1509   ]
1510   \centerline{\large\liProblemName{#1}}
1511
1512   \medskip
1513
1514   \begin{description}
1515     \item[Gegeben:] #2
1516     \item[Frage:] #3
1517   \end{description}
1518 \end{mdframed}
1519 }
```



```

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

1520 \NewDocumentCommand{ \liPolynomiellReduzierbar } { m O{p} m } {
1521 \begin{displaymath}
1522 \liProblemName{#1}
1523 \preceq_{#2}
1524 \liProblemName{#3}
1525 \end{displaymath}
1526 }

\liProblemVertexCover

1527 \def\liProblemClique{%
1528 Das \textbf{Cliquesproblem} fragt nach der Existenz einer Clique der
1529 Mindestgröße  $n$  in einem gegebenen Graphen.
1530 \footcite[Seite 76]{theo:fs:4}
1531 Eine Clique ist eine Teilmenge von Knoten in einem ungerichteten
1532 Graphen, bei der \emph{jedes Knotenpaar durch eine Kante} verbunden ist.
1533 \footcite{wiki:cliquenproblem}
1534 }

\liProblemVertexCover

1535 \def\liProblemVertexCover{%
1536 %
1537 Das \textbf{Knotenüberdeckungsproblem} (\liProblemName{Vertex Cover})
1538 fragt, ob zu einem gegebenen einfachen Graphen und einer natürlichen
1539 Zahl  $k$  eine Knotenüberdeckung der Größe von höchstens  $k$  existiert.
1540 \footcite{wiki:knotenueberdeckung}
1541
1542 Das heißt, ob es eine aus maximal  $k$  Knoten bestehende Teilmenge  $U$ 
1543 der Knotenmenge gibt, so dass jede Kante des Graphen mit
1544 mindestens einem Knoten aus  $U$  verbunden ist.
1545 \footcite[Seite 78]{theo:fs:4}%
1546 }

\liProblemSubsetSum Kein Karp-21-Problem

1547 \def\liProblemSubsetSum{%
1548 Das \textbf{Teilsommenproblem} (\liProblemName{Subset Sum} oder
1549 \liProblemName{SSP}) ist ein spezielles Rucksackproblem.
1550 \footcite{wiki:teilsommenproblem}
1551 Gegeben sei eine Menge von ganzen Zahlen  $I = \{w_1, w_2, \dots, w_n\}$ .
1552 Gesucht ist eine Untermenge, deren Elementsumme maximal, aber nicht
1553 größer als eine gegebene obere Schranke  $c$  ist.
1554 \footcite[Seite 74]{theo:fs:4}
1555 }

\liProblemSubsetSum Kein Karp-21-Problem

1556 \def\liProblemSat{%
1557 Das \textbf{Erfüllbarkeitsproblem der Aussagenlogik} \liProblemName{Sat}
1558 und \liProblemName{k-SAT} mit  $k \geq 3$ ,  $k \in \mathbb{N}$  (Satz von
1559 Cook) fragt, ob eine aussagenlogische Formel erfüllbar
1560 ist.\footcite{wiki:sat} Das Erfüllbarkeitsproblem der
1561 \emph{Aussagenlogik} ist in exponentieller Zeit in Abhängigkeit der
1562 Anzahl der Variablen mit Hilfe einer Wahrheitstabelle entscheidbar.
1563 Diese \emph{Wahrheitstabelle} kann nicht in polynomieller Zeit
1564 aufgestellt werden.
1565 \footcite[Seite 71]{theo:fs:4}
1566 }

1567

```

2.22 kontrollflussgraph.sty

1568 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

1569 \ProvidesPackage{lehramt-informatik-kontrollflussgraph}[2020/11/07]

2.22.1 Makro-Kürzel

```
\let\b=\liBedingung
\let\c=\liKontrollCode
\let\f=\liBedingungFalsch
\let\k=\liKontrollTextzeileKnoten
\let\p=\liKontrollKnotenPfad
\let\w=\liBedingungWahr
```

2.22.2 TeX-Markup-Beispiel

```
\begin{liKontrollflussgraph}[xscale=1,yscale=-1.6]
\node[knoten] at (0,0) (S) {S};

\node[pin=\c{boolean resultat = false;}] at (0,1) (2) {2};
\node[pin=\c{if (wort != null)}] at (0,2) (3) {3};
\node[pin=\c{int laenge = wort.length;}] at (1,3) (4) {4};
\node[pin=\c{if (laenge >= 2)}] at (1,4) (5) {5};
\node[pin=\c{resultat = true; int i = 0;}] at (2,5) (6) {6};
\node[pin=\c{for (i < laenge / 2;)}] at (2,6) (7) {7};
\node[pin=\c{char c1; char c2 ...}] at (3,7) (8) {8};
\node[pin=\c{if (Char...) }] at (2,8) (10) {10};
\node[pin=\c{resultat = false; break;}] at (2,9) (12) {12};
\node[pin=\c{if (laenge == 1)}] at (1,10) (17) {17};
\node[pin=\c{resultat = true;}] at (1,11) (18) {18};
\node[pin=180:\c{return resultat;}] at (-1,9) (22) {22};
\node[knoten] at (-1,10) (E) {E};

\path (S) -- (2);
\path (2) -- (3);
\path[wahr] (3) -- (4) \b{right}{wort != null};
\path[falsch] (3) -- (22) \b{left}{wort == null};
\path (4) -- (5);
\path[wahr] (5) -- (6) \b{right}{laenge >= 2};
\path[falsch] (5) -- (17) \b{left,rotate=70,pos=0.2}{laenge < 2};
\path (6) -- (7);
\path[wahr] (7) -- (8) \b{right}{i < laenge / 2};
\path[falsch] (7) -- (22) \b{left}{i >= laenge / 2};
\path (8) -- (10);
\path[wahr] (10) -- (12) \b{right}{c != c};
\path[falsch] (10) -- (7) \b{left,rotate=70,pos=0.8}{c == c};
\path (12) -- (22);
\path[wahr] (17) -- (18) \b{right}{laenge == 1};
\path[falsch] (17) -- (22) \b{right,rotate=-20,pos=0.99}{laenge != 1};
\path (18) -- (22);
\path (22) -- (E);
\end{liKontrollflussgraph}
```

2.22.3 TikZ: pin

Die Knoten erhalten sogenannte Pins, in denen Code-Ausschnitte der entsprechenden Anweisung zu sehen ist.

```
pin=[options]angle:text

\node[pin=\c{resultat = true;}] at (1,11) {18};
\node[pin=180:\c{return resultat;}] at (-1,9) {22};
\node[pin=[pin distance=3cm]\c{resultat = ...}]
```

```

1570 \RequirePackage{tikz}
1571 \usetikzlibrary{positioning}
1572 \tikzset{
1573   li kontrollfluss/.style={
1574     knoten/.style={
1575       circle,
1576       draw
1577     },
1578     usebox/.style={
1579       draw,
1580       rectangle,
1581       font=\scriptsize,
1582       anchor=west,
1583       align=left,
1584     },
1585     bedingung/.style={
1586       midway,
1587       draw=none,
1588       font=\scriptsize
1589     },
1590     knotenbeschriftung/.style={
1591       draw,
1592       rectangle,
1593       midway,
1594       font=\scriptsize
1595     },
1596     wahr/.style={
1597       thick
1598     },
1599     falsch/.style={
1600       dashed
1601     },
1602     every node/.style={
1603       circle,
1604       draw,
1605     },
1606     every edge/.append style={
1607       every node/.style={
1608         draw=none,
1609         bedingung,
1610       }
1611     },
1612     every path/.style={
1613       draw,
1614       ->,
1615     },
1616     every pin/.style={
1617       draw,
1618       dotted,
1619       rectangle,
1620       pin position=right
1621     },
1622     every pin edge/.style={
1623       dotted,
1624       arrows=-,
1625     }
1626   }
1627 }

```

2.22.4 Umgebungen

liKontrollflussgraph

```

1628 \NewDocumentEnvironment { liKontrollflussgraph } { 0{} } {

```

```

1629 \begin{tikzpicture}[
1630     li kontrollfluss,
1631     #1
1632 ]
1633 } {
1634 \end{tikzpicture}
1635 }

```

2.22.5 Makros

`\liAnweisung`

```
1636 \def\liAnweisung#1(#2,#3){\node[knoten] at (#2,#3) (#1) {#1};}
```

`\liBedingung` **Let-Abkürzung:** `\let\b=\liBedingung`

```
1637 \def\liBedingung#1#2{\node[bedingung,#1]{\texttt{#2}}}
```

`\liBedingungWahr` **Let-Abkürzung:** `\let\w=\liBedingungWahr`

```
1638 \def\liBedingungWahr#1{\node[bedingung,#1]{\texttt{[true]}}}
```

`\liBedingungFalsch` **Let-Abkürzung:** `\let\f=\liBedingungFalsch`

```
1639 \def\liBedingungFalsch#1{\node[bedingung,#1]{\texttt{[false]}}}
```

`\liKontrollCode` **Let-Abkürzung:** `\let\c=\liKontrollCode`

```
1640 \def\liKontrollCode#1{{\tiny\texttt{\textcolor{gray}{#1}}}}
```

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

```
1641 \def\liKontrollTextzeileKnoten#1{\raisebox{-2pt}{\tikz[scale=0.5,transform shape] \node[draw
```

`\liKontrollKnotenPfad` **Let-Abkürzung:** `\let\p=\liKontrollKnotenPfad`

```
1642 \ExplSyntaxOn
```

```
1643 \NewDocumentCommand { \liKontrollKnotenPfad } { m }
```

```
1644 {
```

```
1645   \seq_set_split:Nnn \l_tmpa_seq { - } { #1 }
```

```
1646   \seq_set_map:NNn \l_tmpa_seq \l_tmpa_seq {\liKontrollTextzeileKnoten{##1}}
```

```
1647   \seq_use:Nn \l_tmpa_seq { ~~~~ }
```

```
1648 }
```

```
1649 \ExplSyntaxOff
```

```
1650
```

2.23 kopf-fusszeilen.sty

```
1651 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1652 \ProvidesPackage{lehramt-informatik-kopf-fusszeilen}[2021/08/20 Kopf-
1653 und Fußzeilen mit Hilfe des Pakets fancyhdr.]

1654 \ExplSyntaxOn

1655 \fancyhead{}
1656 \fancyhead[R0,LE]{\scriptsize\LehramtInformatikTitel}
1657 \fancyhead[L0,RE]{\scriptsize\today}
1658 \fancyfoot{}
1659 \fancyfoot[LE,R0]{\thepage}
1660 \fancyfoot[L0,CE]{\scriptsize\LehramtInformatikAutorName}
1661 \fancyfoot[CO,RE]{\scriptsize\LehramtInformatikAutorEmail}
1662 \renewcommand{\headrulewidth}{0.4pt}
1663 \renewcommand{\footrulewidth}{0.4pt}
1664 \setlength{\headwidth}{\textwidth}

1665 \ExplSyntaxOff

1666
```

2.24 literatur-dummy.sty

```
1667 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1668 \ProvidesPackage{lehramt-informatik-literatur-dummy}[2020/11/27]

\literatur

1669 \def\literatur{}

\footcite

1670 % \RequirePackage[stable,multiple]{footmisc}
1671 \NewDocumentCommand{ \footcite } { o m }{}

1672
```

2.25 literatur.sty

```
1673 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1674 \ProvidesPackage{lehramt-informatik-literatur}[2020/11/27]

1675 \RequirePackage{csquotes}
1676 \RequirePackage[
1677   bibencoding=utf8,
1678   citestyle=authortitle,
1679   backend=biber,
1680 ]{biblatex}
1681 \addbibresource{\LehramtInformatikRepository/Quellen/10_DB.bib}
1682 \addbibresource{\LehramtInformatikRepository/Quellen/20_OOMUP.bib}
1683 \addbibresource{\LehramtInformatikRepository/Quellen/30_AUD.bib}
1684 \addbibresource{\LehramtInformatikRepository/Quellen/40_SOSY.bib}
1685 \addbibresource{\LehramtInformatikRepository/Quellen/60_FUMUP.bib}
1686 \addbibresource{\LehramtInformatikRepository/Quellen/70_THEO.bib}
1687 \addbibresource{\LehramtInformatikRepository/Quellen/80_DDI.bib}
1688 \addbibresource{\LehramtInformatikRepository/Quellen/Allgemein.bib}
1689 \addbibresource{\LehramtInformatikRepository/Quellen/Examen.bib}
1690 \addbibresource{\LehramtInformatikRepository/Quellen/Wikipedia.bib}
1691 % To allow footnotes in the heading
1692 \RequirePackage[stable,multiple]{footmisc}

\literatur

1693 \def\literatur{\printbibliography[heading=subbibliography]}

1694
```

2.26 makros.sty

```
1695 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1696 \ProvidesPackage{lehramt-informatik-makros}[2020/11/27 Eine Sammlung von
1697 häufig verwendeten Makros und Umgebungen, die thematisch zu keinem
1698 anderen Paket passen]
1699 \RequirePackage{hyperref}
1700 \RequirePackage{graphicx}
    Für die Umgebung liQuellen benötigt.
1701 \RequirePackage{paralist}

\inhaltsverzeichnis
1702 \def\inhaltsverzeichnis {
1703   \begin{mdframed}
1704     \begin{group}
1705       \let\clearpage\relax
1706       \tableofcontents
1707     \end{group}
1708   \end{mdframed}
1709 }

\memph \mephm (\marginpar and \emph)
1710 \newcommand{\memph}[1]{\emph{#1}\marginpar{\tiny#1}}

\SLASH
1711 \newcommand\SLASH{\char`\\}

\liPseudoUeberschrift Text, der sich wie eine Überschrift verhält.
1712 \newcommand{\liPseudoUeberschrift}[1]{
1713   \bigskip
1714   \noindent
1715   \textsf{\textbf{#1}}
1716   \noindent
1717 }

\liBeschriftung Ähnlich dem Makro \liPseudoUeberschrift{}. Am Ende des Textes wird ein Doppelpunktzeichen angehängt.
1718 \newcommand{\liBeschriftung}[1]{
1719   \par
1720   \noindent
1721   \medskip
1722   \textbf{#1}:
1723   \medskip
1724   \noindent
1725 }

\hinweis
1726 \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.
1727 \NewDocumentEnvironment { liProjektSprache }{ o +b } {} {}

liEinbettung
1728 \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.
1729 \RequirePackage{xparse}
1730 \ExplSyntaxOn
```



```

1731 \NewDocumentEnvironment{ liAntwort } { 0{standard} }
1732 {
1733   \str_case:nn {#1} {
1734     {standard} {
1735       \def\beschriftung{}
1736       \mdfsetup{backgroundcolor=gray!1,linecolor=gray}
1737     }
1738     {richtig} {
1739       \def\beschriftung{richtig}
1740       \mdfsetup{backgroundcolor=green!5,linecolor=green}
1741     }
1742     {falsch} {
1743       \def\beschriftung{falsch}
1744       \mdfsetup{backgroundcolor=red!3,linecolor=red}
1745     }
1746     {muster} {
1747       \def\beschriftung{Musterlösung}
1748       \mdfsetup{backgroundcolor=green!12,linecolor=green}
1749     }
1750   }
1751   \ifx\beschriftung\empty\else
1752     \noindent
1753     \textbf{\beschriftung{:}}
1754   \fi
1755   \begin{mdframed}
1756 }
1757 {\end{mdframed}}

```

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

```

1758 \NewDocumentEnvironment{ liAdditum } { o }
1759 {
1760   \begin{mdframed}[backgroundcolor=yellow!5]
1761     \IfNoValueTF {#1}
1762     { \liPseudoUeberschrift{Additum} }
1763     { \liPseudoUeberschrift{Additum:~#1} }
1764   }
1765 {\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.

```

1766 \NewDocumentEnvironment{ liExkurs }{o +b}{
1767   \vspace{0.2cm}%
1768   \begin{mdframed}[
1769     backgroundcolor=white,
1770     bottomline=false,
1771     innermargin=1cm,
1772     leftline=true,
1773     linecolor=black,
1774     linewidth=0.1cm,
1775     outermargin=1cm,
1776     rightline=false,
1777     topline=false,
1778   ]

```

```

1779 \footnotesize
1780 \noindent%
1781 \textbf{Exkurs:~\#1}\par%
1782 \noindent%
1783 #2
1784 \end{mdframed}
1785 \vspace{0.2cm}
1786 }{}

```

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

```

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

```

Weiterführende Literatur:

- Quelle 1
- Quelle 2

```

1787 \cs_new:Npn \listen@punkt #1 {\item #1}
1788 \NewDocumentEnvironment { liQuellen }{ +b }
1789 {
1790   \seq_clear_new:N \l_quellen
1791   \seq_set_split:Nnn \l_quellen {\item} {#1}
1792   \seq_remove_all:Nn \l_quellen {}
1793   \begin{mdframed}[backgroundcolor=white!0]
1794     \footnotesize
1795     \noindent
1796     \textsf{\textbf{Weiterführende~Literatur:}}
1797     \medskip
1798     \begin{compactitem}
1799       \seq_map_function:NN \l_quellen {\listen@punkt}
1800     \end{compactitem}
1801   \end{mdframed}
1802   %
1803   \makeatletter
1804   \par\@afterindentfalse\@afterheading
1805   \makeatother
1806 } {}

```

liLernkartei

```

1807 \NewDocumentEnvironment { liLernkartei }{ m +b }
1808 {
1809   \begin{mdframed}
1810     \footnotesize
1811     \noindent%
1812     \textbf{Lernkarteikarte:~\#1}\par%
1813     \noindent%
1814     #2
1815   \end{mdframed}
1816 } {}

```

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

```

1817 \NewDocumentEnvironment { liDiagramm }{ m +b }
1818 {
1819   \begin{mdframed}[backgroundcolor=white!0]
1820     \small
1821     \noindent%
1822     \textit{#1}:
1823     \begin{center}

```

```

1824 #2
1825 \medskip
1826 \end{center}
1827 \end{mdframed}
1828 } {}
1829 \ExplSyntaxOff

\liFussnoteUrl \liFussnoteUrl[\langle zusätzlicher-text \rangle]{\langle url \rangle} \liFussnoteUrl[zusätzlicher Text]{url}:
Eine HTTP-URL als Fußnote setzen.
1830 \NewDocumentCommand{\liFussnoteUrl} { o m } {
1831 \footnote{\url{#2}\IfNoValueTF{#1}{}{ ( #1 )}}
1832 }
1833

\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.
1834 \NewDocumentCommand{\liFussnoteLink} { o m m } {
1835 \footnote{\href{#3}{#2}\IfNoValueTF{#1}{}{ ( #1 )}}
1836 }

\zB
1837 \def\zB{z.\,B. }

\ZB
1838 \def\ZB{Z.\,B. }

\dh
1839 \def\dh{d.\,h. }

1840

```

2.27 master-theorem.sty

1841 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

1842 \ProvidesPackage{lehramt-informatik-master-theorem}[2021/04/13]

2.27.1 Makro-Kürzel

\let\O=\liO

\let\o=\liOmega

\let\T=\liT

\let\t=\liTheta

\liMasterVariablenDeklaration

{3} % a

{3} % b

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

\liMasterFallRechnung

% 1. Fall

{für $\varepsilon = 4$: \}

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

% 2. Fall

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

% 3. Fall

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

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

\liMasterVariablenDeklaration

{ } % a

{ } % b

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

\liMasterFallRechnung

% 1. Fall

{ }

% 2. Fall

{ }

% 3. Fall

{ }

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

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

1843 \ExplSyntaxOn

1844 \RequirePackage{amsmath}

\liRundeKlammer

1845 \def\liRundeKlammer#1{

1846 \negthinspace \left(#1 \right)

1847 }

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

1848 \def\liThetaOhneMathe#1{

1849 \Theta \liRundeKlammer{#1}

1850 }

1851 \def\liTheta#1{

1852 \ifmmode

1853 \liThetaOhneMathe{#1}

1854 \else

1855 \$\liThetaOhneMathe{#1}\$

1856 \fi

1857 }

```

\liOmega \liOmega{n^2}:  $\Omega(n^2)$ 
1858 \def\liOmegaOhneMathe#1{
1859 \Omega \liRundeKlammer{#1}
1860 }
1861 \def\liOmega#1{
1862 \ifmmode
1863 \liOmegaOhneMathe{#1}
1864 \else
1865 $\liOmegaOhneMathe{#1}$
1866 \fi
1867 }

\liO \liO{n^2}:  $\mathcal{O}(n^2)$ 
1868 \def\liOOhneMathe#1{
1869 \mathcal{O} \liRundeKlammer{#1}
1870 }
1871 \def\liO#1{
1872 \ifmmode
1873 \liOOhneMathe{#1}
1874 \else
1875 $\liOOhneMathe{#1}$
1876 \fi
1877 }

\liT Let-Abkürzung: \let\T=\liT
\liT{16}{2}:  $16 \cdot T\left(\frac{n}{2}\right)$  \liT{}{2}:  $T\left(\frac{n}{2}\right)$ 
1878 \def\liTOhneMathe#1#2{
1879 \tl_if_blank:nTF {#1}
1880 {}
1881 {#1 \cdot }
1882 T
1883 \liRundeKlammer{\frac{n}{#2}}
1884 }
1885 \def\liT#1#2{
1886 \ifmmode
1887 \liTOhneMathe{#1}{#2}
1888 \else
1889 $\liTOhneMathe{#1}{#2}$
1890 \fi
1891 }

\liRekursionsGleichung \liRekursionsGleichung:  $T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$ 
1892 \def\liRekursionsGleichung{
1893 $T(n) = \liT{a}{b} + f(n)$
1894 }

\liBedingungEins \liBedingungEins:  $f(n) \in \mathcal{O}(n^{\log_b a - \varepsilon})$ 
1895 \def\liBedingungEins{
1896 $f(n) \in \liO{n^{\log\sb{b}a - \varepsilon}}$
1897 }

\liBedingungZwei \liBedingungZwei:  $f(n) \in \Theta(n^{\log_b a})$ 
1898 \def\liBedingungZwei{
1899 $f(n) \in \liTheta{n^{\log\sb{b}a}}$
1900 }

\liBedingungDrei \liBedingungDrei:  $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ 
1901 \def\liBedingungDrei{
1902 $f(n) \in \liOmega{n^{\log\sb{b}a + \varepsilon}}$
1903 }

1904 \ExplSyntaxOff

```

\liMasterVariablen

```

1905 \def\liMasterVariablen{
1906   \begin{displaymath}
1907     T(n) = \liT{a}{b} + f(n)
1908   \end{displaymath}
1909
1910   \begin{itemize}
1911     \item[$a = $]
1912       Anzahl der rekursiven Aufrufe, Anzahl der Unterprobleme in der
1913       Rekursion
1914       ($a \geq 1$).
1915
1916     \item[$\textstyle{\frac{1}{b}} = $]
1917       Teil des Originalproblems, welches wiederum durch alle Unterprobleme
1918       repräsentiert wird, Anteil an der Verkleinerung des Problems ($b > 1$).
1919
1920     \item[$f(n) = $]
1921       Kosten (Aufwand, Nebenkosten), die durch die Division des Problems und
1922       die Kombination der Teillösungen entstehen. Eine von $T(n)$
1923       unabhängige und nicht negative Funktion.
1924   \end{itemize}
1925   \footcite{wiki:master-theorem}
1926   \footcite[Seite 19-35]{aud:fs:2}
1927 }
```

\liMasterFaelle

```

1928 \def\liMasterFaelle{
1929   \begin{description}
1930     \item[1. Fall:]
1931       $T(n) \in \liTheta{n^{\log\sb{b}a}}$
1932
1933       \hfill falls \liBedingungEins
1934       für $\varepsilon > 0$
1935
1936     \item[2. Fall:]
1937       $T(n) \in \liTheta{n^{\log\sb{b}a} \cdot \log n}$
1938
1939       \hfill falls \liBedingungZwei
1940
1941     \item[3. Fall:]
1942       $T(n) \in \liTheta{f(n)}$
1943
1944       \hfill falls \liBedingungDrei
1945       für $\varepsilon > 0$
1946       und ebenfalls für ein $c$ mit $0 < c < 1$ und alle hinreichend großen $n$
1947       gilt:
1948       $a \cdot f(\textstyle{\frac{n}{b}}) \leq c \cdot f(n)$
1949   \end{description}
1950 }
```

\liMasterVariablenDeklaration

```

1951 \def\liMasterVariablenDeklaration#1#2#3{
1952   \begin{description}
1953     \item[Allgemeine Rekursionsgleichung:] \strut
1954
1955     \liRekursionsGleichung
1956
1957     \item[Anzahl der rekursiven Aufrufe ($a$):] \strut
1958
1959     #1
1960
1961     \item[Anteil Verkleinerung des Problems ($b$):] \strut
1962   \end{description}
}
```

```

1963     um  $\frac{1}{b}$  also  $b = \frac{1}{a}$ 
1964
1965     \item[Laufzeit der rekursiven Funktion ( $f(n)$ ):] \strut
1966
1967      $T(n) = \frac{1}{b} + T(\frac{n}{b})$ 
1968
1969     \item[Ergibt folgende Rekursionsgleichung:] \strut
1970
1971      $T(n) = \frac{1}{b} + T(\frac{n}{b})$ 
1972 \end{description}
1973 }

\liMasterFallRechnung
1974 \def\liMasterFallRechnung#1#2#3{
1975   \begin{description}
1976     \item[1. Fall:] \liBedingungEins:
1977
1978     #1
1979
1980     \item[2. Fall:] \liBedingungZwei:
1981
1982     #2
1983
1984     \item[3. Fall:] \liBedingungDrei:
1985
1986     #3
1987   \end{description}
1988 }

\liMasterExkurs
1989 \def\liMasterExkurs{
1990   \begin{liExkurs}[Master-Theorem]
1991     \liMasterVariablen
1992
1993     \noindent
1994     Dann gilt:
1995
1996     \liMasterFaelle
1997   \end{liExkurs}
1998 }

\liMasterWolframLink Link zu Wolframalpha (+ durch Prozent 2B ersetzen)
1999 \def\liMasterWolframLink#1{
2000   Berechne die Rekursionsgleichung auf WolframAlpha:
2001   \href{https://www.wolframalpha.com/input/?i=#1}{WolframAlpha}
2002 }

2003

```

2.28 mathe.sty

```
2004 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2005 \ProvidesPackage{lehramt-informatik-mathe}[2020/06/10]
2006
2007 % for example \ltimes \rtimes
2008 %\RequirePackage{amssymb}
2009 \RequirePackage{amsmath}
2010
2011 %%
2012 % \mlq \mrq
2013 %%
2014 \DeclareMathSymbol{\mlq}{\mathord}{operators}{``}
2015 \DeclareMathSymbol{\mrq}{\mathord}{operators}{`}
2016
```


2.29 minimierung.sty

```

2017 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2018 \ProvidesPackage{lehramt-informatik-mathe}[2021/03/13 Für den
2019 Minimierungsalgorithmus von einem NEA zu einem DEA]
2020 \liLadePakete{typographie}

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

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

\liFussnoten

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

\liFussnote

2021 \def\liFussnote#1{\$x_{#1}\$}

2022 \def\li@fussnote@text#1#2{
2023 \liFussnote{#1}
2024 \quad
2025 {\footnotesize #2}
2026 }

\liFussnoteEinsText
2027 \def\liFussnoteEinsText{
2028 \li@fussnote@text{1}
2029 {Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.}
2030 }

\liFussnoteZweiText
2031 \def\liFussnoteZweiText{
2032 \li@fussnote@text{2}
2033 {Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt.}
2034 }

\liFussnoteDreiText
2035 \def\liFussnoteDreiText{
2036 \li@fussnote@text{3}

```

```

2037 {In weiteren Iterationen markierte Zustände.}
2038 }

\liFussnoteVierText
2039 \def\liFussnoteVierText{
2040   \li@fussnote@text{4}
2041   {...}
2042 }

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



2043 \def\liFussnoten{
2044   \bigskip
2045   \noindent
2046   \liFussnoteEinsText
2047   \noindent
2048   \liFussnoteZweiText
2049   \noindent
2050   \liFussnoteDreiText
2051   \noindent
2052   \liFussnoteVierText
2053   \noindent
2054   \liFussnoteVierText
2055   \noindent
2056   \liFussnoteVierText
2057 }

\liLeereZelle \liLeereZelle:  $\emptyset$ 
Let-Abkürzung: \let\l=\liLeereZelle
2058 \def\liLeereZelle{\emptyset}

\liZustandsPaarVariablenName
2059 \def\liZustandsPaarVariablenName{z}

\liZustandsPaar
2060 \def\liZustandsPaar#1#2{
2061   $(
2062     \liZustandsPaarVariablenName_#1,
2063     \liZustandsPaarVariablenName_#2
2064   )$
2065 }

liUebergangsTabelle
2066 \renewcommand{\arraystretch}{1.4}
2067 \NewDocumentEnvironment{ liUebergangsTabelle } { m m } {
2068   \liPseudoUeberschrift{Übergangstabelle}
2069   \begin{center}
2070     \begin{tabular}{r|l|l}
2071       \textbf{Zustandspaar} & \textbf{\#1} & \textbf{\#2} \\ \hline
2072     \end{tabular}
2073   \end{center}
2074 }

liUeberschriftDreiecksTabelle \liUeberschriftDreiecksTabelle: Minimierungstabelle (Table filling)

2076 \ExplSyntaxOn
2077 \def\liUeberschriftDreiecksTabelle{
2078   \liPseudoUeberschrift{Minimierungstabelle~(Table~filling)}
2079 }

```

\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. —

```

2080 \def\liMinimierungErklaerung{
2081   %\footcite[Seite~19]{koenig}
2082   \liParagraphMitLinien{
2083     Der~Minimierungs-Algorithmus~(auch~Table-Filling-Algorithmus-genannt)~
2084     trägt~in~seinem~Verlauf~eine~Markierung~in~alle~
2085     diejenigen~Zellen~der~Tabelle~ein,~die~zueinander~nicht~äquivalente~
2086     Zustände~bezeichnen.~Die~Markierung~„ $x_{\sb{n}}$ “~in~einer~Tabellenzelle~( $i$ ,~
2087      $j$ )~bedeutet~dabei,~dass~das~Zustandspaar~( $i$ ,~ $j$ )~in~der~ $k$ -ten~
2088     Iteration~des~Algorithmus~markiert~wurde~und~die~Zustände~ $i$ ~und~ $j$ ~
2089     somit~zueinander~( $k - 1$ )~äquivalent,~aber~nicht~ $k$ -äquivalent~und~
2090     somit~insbesondere~nicht~äquivalent~sind.~Bleibt~eine~Zelle~bis~zum~Ende~
2091     unmarkiert,~sind~die~entsprechenden~Zustände~zueinander~äquivalent.
2092   }
2093 }
2094 \ExplSyntaxOff
2095

```

2.30 normalformen.sty

```
2096 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2097 \ProvidesPackage{lehramt-informatik-normalformen}[2020/12/10]
2098 Hilfsmakros zum Setzen von Normalformen, Funktionalen Abhängigkeiten,
2099 Attributhülle]
```

Mathe für textit tex etc, typographie für geschweifte Klammer

```
2100 \liLadePakete{mathe,typographie}
2101 \directlua{
2102   helfer = require('lehramt-informatik-helfer')
2103   normalformen = require('lehramt-informatik-normalformen')
2104 }
```

2.30.1 Makro-Kürzel

```
\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
```

```
2105 \def\liTeilen#1{
2106   \directlua{tex.print(normalformen.teilen('#1'))}
2107 }
```

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

```
2108 \def\liAttributHuelleOhneMathe#1{\text{AttrHülle}(\#1)}
2109 \def\liAttributHuelle#1{
2110   \ifmmode
2111     \liAttributHuelleOhneMathe{\#1}
2112   \else
2113     $\liAttributHuelleOhneMathe{\#1}$
2114   \fi
2115 }
```

\liAttributMenge Let-Abkürzung: $\text{let}\ \text{m}=\text{liAttributMenge}$
2116 $\text{def}\ \text{liAttributMenge}\#1\{\backslash\{\ \text{textit}\{\#1\}\ \backslash\}$

liAHuelle

```
2117 \NewDocumentEnvironment{ liAHuelle } { +b } {
2118   \begingroup
2119   \footnotesize
2120   \begin{multline*}
2121     \#1
2122   \end{multline*}
2123   \endgroup
2124 } { }
```

\liLinksReduktion Nur innerhalb von liAHuelle zu verwenden bzw. multline

Let-Abkürzung: $\text{let}\ \text{ahL}=\text{liLinksReduktion}$
 $\text{ahL}\{\text{ursprüngliche linke Attributmenge}\{\text{ohne dieses Attribut}\}\{\text{Ergebnis}\}$
2125 $\text{def}\ \text{liLinksReduktion}\#1\#2\#3\{$
2126 $\text{shoveleft}\{$
2127 $\text{liAttributHuelleOhneMathe}\{\text{FA},$
2128 $\text{liAttributMenge}\{\#1\ \text{string}\ \#2\}\} =$
2129 $\}\ \backslash\backslash$

```

2130 \shoveright{
2131   \liAttributMenge{#3}
2132 } \\\
2133 }

```

```

\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( $F, \{A, E \setminus E\}$ ) =  $\{A, E, F, B, D\}$ 
2134 \def\liLinksReduktionInline#1#2#3{%
2135   {%
2136     \footnotesize%
2137     $\liAttributHuelleOhneMathe{F,
2138       \liAttributMenge{#1} \string\ #2}} =
2139     \liAttributMenge{#3}$
2140   }
2141 }

```

```

\liLinksReduktionInline Let-Abkürzung: \let\ahr=\liLinksReduktionInline
    \ahr{gelöschte FA}{neue FA ohne rechts Attribut}{gegebene Attribute}{Ergebnis}
2142 \def\liRechtsReduktionInline#1#2#3#4{%
2143   {%
2144     \footnotesize%
2145     $\liAttributHuelleOhneMathe{
2146       F \setminus
2147       \liFunktionaleAbhaengigkeit{#1}
2148       \def\tmp{#2}\ifx\tmp\empty
2149       \else
2150         \cup \liFunktionaleAbhaengigkeit{#2}
2151       \fi
2152       ,
2153       \liAttributMenge{#3}
2154     } =
2155     \liAttributMenge{#4}$
2156   }
2157 }

```

```

\liFunktionaleAbhaengigkeit Let-Abkürzung: \let\fa=\liFunktionaleAbhaengigkeit
    \fa{A, B -> C, D}:  $\{A, B\} \rightarrow \{C, D\}$  Regulärer Ausdruck zum Konvertieren:
    \$(.*?) \\\rightarrow (.*)\$ \fa{$1 -> $2}
2158 \def\liFunktionaleAbhaengigkeit#1{%
2159   \directlua{normalformen.drucke_funk_abhaengigkeit('#1', false)}%
2160 }

```

```

\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;

```

```

2161 \NewDocumentCommand {\liFunktionaleAbhaengigkeiten} { O{FA} m } {
2162   \liGeschweifteKlammern
2163   {#1}
2164   {
2165     \begin{align*}
2166       \directlua{normalformen.drucke_funk_abhaengigkeiten('#2')}
2167     \end{align*}
2168   }
2169   {-0.5cm}
2170   {-1.7cm}
2171 }

\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}

2172 \NewDocumentCommand {\liRelation} { O{R} m } {
2173   $\directlua{
2174     local name = helper.konvertiere_tiefgestellt('#1')
2175     tex.print(name)
2176   }$(\textit{\, #2\,})
2177 }

2178

```

2.31 petri.sty

2179 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2180 \ProvidesPackage{lehramt-informatik-petri}[2020/12/03]

2.31.1 Makro-Kürzel

\let\t=\liPetriTransitionsName

\let\tp=\liPetriTransPfeile

\let\k=\liPetriErreichKnotenDrei

2181 \RequirePackage{tikz}

2182 \usetikzlibrary{petri,arrows.meta}

Für die Darstellungsmatrix

2183 \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

```
2184 \def\liPetriSetzeSchluessel{%
2185   \def\TmpTransitionOne{}%
2186   \def\TmpTransitionTwo{}%
2187   \def\TmpTransitionThree{}%
2188   \def\TmpTransitionFour{}%
2189   \def\TmpTransitionFive{}%
2190   \def\TmpTransitionSix{}%
2191   \def\TmpTransitionSeven{}%
2192   \def\TmpTransitionEight{}%
2193   \def\TmpTransitionNine{}%
2194   \def\TmpTransitionTen{}%
2195   \pgfkeys{/petri/.cd,
2196     p1/.store in=\TmpPlaceOne,p1/.default=0,p1,
2197     p2/.store in=\TmpPlaceTwo,p2/.default=0,p2,
2198     p3/.store in=\TmpPlaceThree,p3/.default=0,p3,
2199     p4/.store in=\TmpPlaceFour,p4/.default=0,p4,
```

```

2200 p5/.store in=\TmpPlaceFive,p5/.default=0,p5,
2201 p6/.store in=\TmpPlaceSix,p6/.default=0,p6,
2202 p7/.store in=\TmpPlaceSeven,p7/.default=0,p7,
2203 p8/.store in=\TmpPlaceEight,p8/.default=0,p8,
2204 p9/.store in=\TmpPlaceNine,p9/.default=0,p9,
2205 p10/.store in=\TmpPlaceTen,p10/.default=0,p10,
2206 t1/.store in=\TmpTransitionOne,t1/.default=activated,
2207 t2/.store in=\TmpTransitionTwo,t2/.default=activated,
2208 t3/.store in=\TmpTransitionThree,t3/.default=activated,
2209 t4/.store in=\TmpTransitionFour,t4/.default=activated,
2210 t5/.store in=\TmpTransitionFive,t5/.default=activated,
2211 t6/.store in=\TmpTransitionSix,t6/.default=activated,
2212 t7/.store in=\TmpTransitionSeven,t7/.default=activated,
2213 t8/.store in=\TmpTransitionEight,t8/.default=activated,
2214 t9/.store in=\TmpTransitionNine,t9/.default=activated,
2215 t10/.store in=\TmpTransitionTen,t10/.default=activated,
2216 scale/.store in=\TmpScale,scale/.default=0.5,
2217 x/.store in=\TmpX,x/.default=5,
2218 y/.store in=\TmpY,y/.default=5,
2219 }%
2220 }

```

```

2221 \tikzset{
2222   li petri/.style={
2223     activated/.style={
2224       very thick
2225     },
2226     inhibitor/.style={
2227       {Circle[open,length=2mm,fill=white]}-
2228     }
2229   }
2230 }

```

\liPetriTransitionsName **Let-Abkürzung:** \let\t=\liPetriTransitionsName
 \t_{\d+}\\$ \t\$1

```

2231 \def\liPetriTransitionsNameOhneMathe#1{t\sb{#1}}
2232 \def\liPetriTransitionsName#1{
2233   \ifmmode
2234     \liPetriTransitionsNameOhneMathe{#1}
2235   \else
2236     $\liPetriTransitionsNameOhneMathe{#1}$
2237   \fi
2238 }

```

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

```

2239 \NewDocumentCommand{ \liPetriErreichTransition } { m m m O{} O{} } {
2240   \draw[->] (#1) edge[#4] node[pos=0.5,auto,sloped,#5]{t\sb{#3}$} (#2);
2241 }

```

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

```

2242 \def\liPetriErreichKnotenDrei#1#2#3{(#1,#2,#3)}

```

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

```

2243 \def\liPetriTransPfeile#1{ $\rightarrow \hspace{0.4cm} \liPetriTransitionsName{#1} \hspace{0.3cm}
2244 }

```


2.32 potenzmengen-konstruktion.sty

```
2245 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2246 \ProvidesPackage{lehramt-informatik-potenzmengen-konstruktion}[2021/02/21 Hilfsmakros
2247 einzusetzen bei der Potenzmengen-Konstruktion (Potenzmengen-Algorithmus)]
```

```
2248 \liLadePakete{formale-sprachen}
2249 \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}
    }
  }
}
```

```
2250 \def\liZustandsMengenSammlung#1#2{
2251   \liZustandsnameGross{#1}
2252   {
2253     \footnotesize
2254     \liPotenzmenge{
2255       \str_case:nn {#1} {#2
2256       }
2257     }
2258 }
```

```
\liZustandsMengenSammlungNr
```

```
2259 \def\liZustandsMengenSammlungNr#1#2{
2260   \liZustandsnameGross{#1}
2261   {
```

```
2262     \footnotesize
2263     \liZustandsmengeNr{
2264         \str_case:nn {#1} #2
2265     }
2266 }
2267 }

2268 \ExplSyntaxOff
2269
```

2.33 pseudo.sty

2270 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2271 \ProvidesPackage{lehramt-informatik-pseudo}[2020/12/30 Paket zum Setzen
 2272 von Pseudo-Code, Hüll-Paket um algorithm2e]

```
\begin{algorithm}[H]
\KwData{$G = (V, E, w)$: ein zusammenhängender, ungerichteter,
kantengewichteter Graph kruskal(G)}
$E' \leftarrow \emptyset$;
$L \leftarrow E$;
Sortiere die Kanten in $L$ aufsteigend nach ihrem Kantengewicht.;
\While{$L \neq \emptyset$}{
  wähle eine Kante $e \in L$ mit kleinstem Kantengewicht;
  entferne die Kante $e$ aus $L$;
  \If{der Graph $(V, E' \cup \{e\})$ keinen Kreis enthält}{
    $E' \leftarrow E' \cup \{e\}$;
  }
}
\KwResult{$M = (V, E')$ ist ein minimaler Spannbaum von $G$.}
\caption{Minimaler Spannbaum nach Kruskal\footcite{wiki:kruskal}}
\end{algorithm}
```

Algorithmus 1: Minimaler Spannbaum nach Kruskal

<p>Data: $G = (V, E, w)$: ein zusammenhängender, ungerichteter, kantengewichteter Graph kruskal(G)</p> <p>$E' \leftarrow \emptyset$; $L \leftarrow E$; Sortiere die Kanten in L aufsteigend nach ihrem Kantengewicht.;</p> <p>while $L \neq \emptyset$ do</p> <div style="margin-left: 20px;"> <p>wähle eine Kante $e \in L$ mit kleinstem Kantengewicht;</p> <p>entferne die Kante e aus L;</p> <p>if der Graph $(V, E' \cup \{e\})$ keinen Kreis enthält then</p> <div style="margin-left: 20px;"> <p>$E' \leftarrow E' \cup \{e\}$;</p> </div> <p>end</p> </div> <p>end</p> <p>Result: $M = (V, E')$ ist ein minimaler Spannbaum von G.</p>

2273 \RequirePackage[german,boxruled]{algorithm2e}

2274

2.34 pumping-lemma.sty

2275 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2276 \ProvidesPackage{lehramt-informatik-pumping-lemma}[2021/08/11 Enthält die
 2277 Definitionen für das Pumping-Lemma in der Regulären Sprache und
 2278 in der Kontextfreien Sprache]

\liPumpingRegulaer

```
2279 \def\liPumpingRegulaer{%
2280   Es sei  $L$  eine reguläre Sprache. Dann gibt es eine Zahl  $j$ , sodass für
2281   alle Wörter  $\omega$  in  $L$  mit  $|\omega| \geq j$  (jedes Wort  $\omega$  in
2282    $L$  mit Mindestlänge  $j$ ) jeweils eine Zerlegung  $\omega = uvw$ 
2283   existiert, sodass die folgenden Eigenschaften erfüllt sind:
2284
2285   \begin{enumerate}
2286     \item  $|v| \geq 1$ 
2287     (Das Wort  $v$  ist nicht leer.)
2288
2289     \item  $|uv| \leq j$ 
2290     (Die beiden Wörter  $u$  und  $v$  haben zusammen höchstens die Länge  $j$ .)
2291
2292     \item Für alle  $i = 0, 1, 2, \dots$  gilt  $uv^i w$  in  $L$ 
2293     (Für jede natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^i w$  in der
2294     Sprache  $L$ )
2295   \end{enumerate}
2296
2297   Die kleinste Zahl  $j$ , die diese Eigenschaften erfüllt, wird
2298   Pumping-Zahl der Sprache  $L$  genannt.\footcite{wiki:pumping-lemma}
2299 }
```

\liPumpingKontextfrei

```
2300 \def\liPumpingKontextfrei{%
2301   Es sei  $L$  eine kontextfreie Sprache. Dann gibt es eine Zahl  $j$ , sodass
2302   sich alle Wörter  $\omega$  in  $L$  mit  $|\omega| \geq j$  zerlegen lassen in
2303    $\omega = uvwxy$ , sodass die folgenden Eigenschaften erfüllt sind:
2304
2305   \begin{enumerate}
2306     \item  $|vx| \geq 1$ 
2307     (Die Wörter  $v$  und  $x$  sind nicht leer.)
2308
2309     \item  $|vwx| \leq j$ 
2310     (Die Wörter  $v$ ,  $w$  und  $x$  haben zusammen höchstens die Länge  $j$ .)
2311
2312     \item Für alle  $i$  in  $\mathbb{N}_0$  gilt  $uv^iwx^iy$  in  $L$  (Für jede
2313     natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^iwx^iy$  in der
2314     Sprache  $L$ )
2315   \end{enumerate}
2316 }
2317
```

2.35 quicksort.sty

```

2318 % https://tex.stackexchange.com/a/142634
2319 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2320 \ProvidesPackage{lehramt-informatik-quicksort}[2020/06/12]
2321
2322 %-----
2323 % USAGE:
2324 % \QSinitialize{comma, separated, numerical, values}
2325 % \loop
2326 % \QSpivotStep
2327 % \ifnum\value{pivotcount}>0
2328 %   \QSSortStep
2329 % \repeat
2330 %-----
2331
2332 % xintfrac does not load xinttools, this must be done explicitly if needed as here.
2333 \RequirePackage{xintfrac, xinttools}
2334
2335 \RequirePackage{tikz}
2336
2337 %-----
2338 % FIRST PART: TikZ styles and macros for the actual drawing
2339 \newcounter{cellcount}% used for coordinates of the node
2340 \newcounter{pivotcount}% when it will remain at zero, will signal the sort is finished.
2341
2342 % Styles defined by Tom Bombaldi. (modified: all share the same size)
2343 % (re-modified \bf -> \bfseries due to extremely annoying warnings from
2344 % KOMA-script which are truly a pain and do not make any sense regarding \bf:
2345 % if I want to use \bf, and know what I am doing, why should I get HARASSED
2346 % by police of LaTeX good conduct ? )
2347 \tikzset{!/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=lime!70!gray},
2348          o/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=olive!50},
2349          r/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2350 % this is the "b" style as used in the image below
2351          b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2352 % nicer:
2353          b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=white, text=magenta},
2354          g/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=gray, text=white, text=white},
2355
2356 % NOTE the b style was originally the same as the r(aised) style apart from
2357 % not being raised, but I find it nicer with a somewhat different
2358 % specification. I have not updated the images though.
2359
2360 % How the nodes are drawn depending on whether on the left of the pivot value
2361 % or on the right, or is a pivot value, or a raised pivot during selection phase.
2362
2363 \def\DecoLEFT #1{%
2364   \xintFor* ##1 in {#1} \do
2365     {\stepcounter{cellcount}\node[o] at (\arabic{cellcount},0) {##1};}%
2366 }
2367
2368 \def\DecoINERT #1{%
2369   \xintFor* ##1 in {#1} \do
2370     {\stepcounter{cellcount}\node[g] at (\arabic{cellcount},0) {##1};}%
2371 }
2372
2373 \def\DecoRIGHT #1{%
2374   \xintFor* ##1 in {#1} \do
2375     {\stepcounter{cellcount}\node[l] at (\arabic{cellcount},0) {##1};}%
2376 }
2377
2378 \def\DecoLEFTwithPivot #1{\stepcounter{pivotcount}%
2379   \xintFor* ##1 in {#1} \do

```

```

2380     {\stepcounter{cellcount}}%
2381     \xintifForLast {\node[r]}\node[o]} at (\arabic{cellcount},0) {##1};}%
2382 }
2383
2384 \def\DecoINERTwithPivot #1{\stepcounter{pivotcount}%
2385     \xintFor* ##1 in {#1} \do
2386     {\stepcounter{cellcount}}%
2387     \xintifForLast {\node[b]}\node[g]} at (\arabic{cellcount},0) {##1};}%
2388 }
2389
2390 \def\DecoRIGHTwithPivot #1{\stepcounter{pivotcount}%
2391     \xintFor* ##1 in {#1} \do
2392     {\stepcounter{cellcount}}%
2393     \xintifForLast {\node[r]}\node[l]} at (\arabic{cellcount},0) {##1};}%
2394 }
2395
2396 %-----
2397 % SECOND PART: the actual sorting routines.
2398
2399 \def\QS@sort@a #1{\expandafter \QS@sort@b \expandafter {\xintLength {#1}}{#1}}
2400 \def\QS@sort@b #1{\ifcase #1
2401     \expandafter\QS@sort@empty
2402     \or\expandafter\QS@sort@single
2403     \else\expandafter\QS@sort@c
2404     \fi
2405 }%
2406 \def\QS@sort@empty #1{}
2407 \def\QS@sort@single #1{\QSIr {#1}}
2408
2409 % This step is to pick the last as pivot.
2410 \def\QS@sort@c #1%
2411     {\expandafter\QS@sort@d\expandafter {\romannumeral0\xintnthelt {-1}{#1}}{#1}}%
2412
2413 % Here \QSLr, \QSIr, \QSR have been let to \relax.
2414 % The trick with \xintApplyUnbraced is that for example when selecting
2415 % the elements smaller than pivot, if we had been using \xintApply we
2416 % would have had at the minimum an empty brace pair. Thus we use the
2417 % "unbraced" variant, but then the \QS@select@smaller has added in
2418 % anticipation a level of braces.
2419 \def\QS@sort@d #1#2{%
2420     \QSLr {\xintApplyUnbraced {\QS@select@smaller {#1}}{#2}}}%
2421     \QSIr {\xintApplyUnbraced {\QS@select@equal {#1}}{#2}}}%
2422     \QSRr {\xintApplyUnbraced {\QS@select@greater {#1}}{#2}}}%
2423 }%
2424 \def\QS@select@smaller #1#2{\xintifLt {#2}{#1}{#{#2}}{ }}% space will stop a f-
    expansion
2425 \def\QS@select@equal #1#2{\xintifEq {#2}{#1}{#{#2}}{ }}% space will stop a f-
    expansion
2426 \def\QS@select@greater #1#2{\xintifGt {#2}{#1}{#{#2}}{ }}% space will stop a f-
    expansion
2427
2428 %
2429 % NOTE 1: thus, each comparison with the pivot is done three (!) times.
2430 %
2431 % NOTE 2: we may well end up with \QSLr {<empty>} situations. This is handled
2432 % silently by the \xintFor loops, and also when \QSLr becomes \QS@sort@a, the
2433 % latter must handle correctly an empty argument.
2434
2435 %-----
2436 % THIRD PART: the main macros \QSpivotStep, \QSsortStep and \QSinitialize.
2437
2438 % This draws all with suitable highlighting for the newly chosen pivots
2439 % (which will be shown raised)

```

```

2440 \def\QSpivotStep {\let\QSLr\DecoLEFTwithPivot
2441             \let\QSIr\DecoINERT
2442             \let\QSIrr\DecoINERT
2443             \let\QSRr\DecoRIGHTwithPivot
2444 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2445             \setcounter{cellcount}{0}\setcounter{pivotcount}{0}%
2446             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2447 }
2448
2449 % This sorts and then draws, showing where the pivot chosen in the previous
2450 % step go. Next time they will have become "inert". If pivotcount is still at
2451 % zero on exit from \QSpivotStep, then this is the signal to stop before
2452 % executing \QSsortStep.
2453 \def\QSsortStep {\def\QSLr {\noexpand\QS@sort@a}%
2454             \def\QSRr {\noexpand\QS@sort@a}%
2455             \def\QSIr {\noexpand\QSIrr}%
2456             \let\QSIrr\relax
2457             \edef\QS@list{\QS@list}%
2458             \let\QSLr\relax
2459             \let\QSRr\relax
2460             \let\QSIr\relax
2461             \edef\QS@list{\QS@list}%
2462             \let\QSLr\DecoLEFT
2463             \let\QSIr\DecoINERTwithPivot
2464             \let\QSIrr\DecoINERT
2465             \let\QSRr\DecoRIGHT
2466 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2467             \setcounter{cellcount}{0}%
2468             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2469 }
2470
2471 \def\QSinitialize #1{%
2472     % first, we convert the comma separated values into a list of braced items
2473     % we use an \edef, and anyhow many \edef's will be used later
2474     \edef\QS@list {\noexpand\QSRr {\xintCSVtoList {#1}}}%
2475     \let\QSRr\DecoRIGHT
2476     % The \QSRr marker mutated to draw the last element as
2477     % pivot and the earlier ones with the suitable style.
2478     %
2479     % The list of marked braced items \QS@list is used both for drawing
2480     % (as here) and for doing the exchange of elements during sort.
2481     \par\centerline{\rule[1.5mm]{0pt}{8mm}\setcounter{cellcount}{0}%
2482             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2483 }
2484

```

2.36 relationale-algebra.sty

```

2485 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2486 \ProvidesPackage{lehrant-informatik-relationale-algebra}[2020/12/11]
2487 \RequirePackage{amsmath}
2488 \RequirePackage{amssymb}

```

```

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

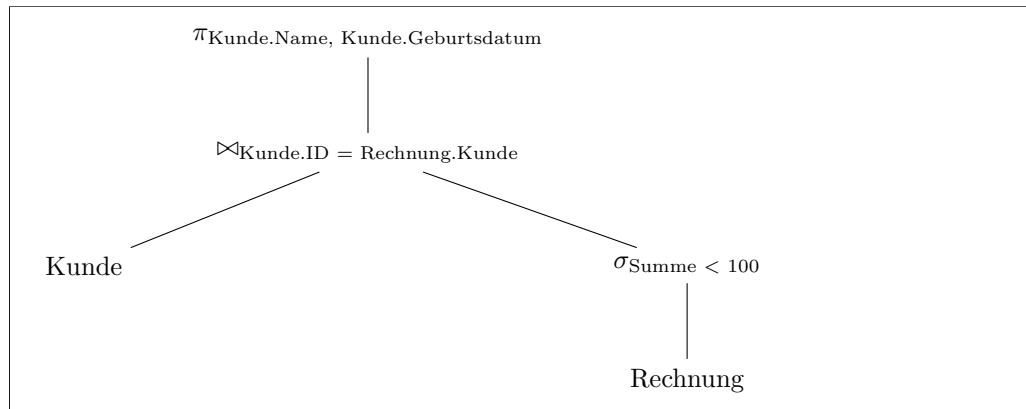
  \node[below=of pi]
    (theta join) {\bowtie_{\text{Kunde.ID = Rechnung.Kunde}}};
    edge (pi);

  \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}

```



```

2489 \RequirePackage{tikz}
2490 \usetikzlibrary{positioning}

    Privates Makros, das zwei Querstriche erzeugt.
2491 \def\o@join{\setbox0=\hbox{\bowtie}}%
2492   \rule[-.02ex]{.25em}{.4pt}\llap{\rule[\ht0]{.25em}{.4pt}}%
2493 }

\leftouterjoin A \leftouterjoin B: A ⋈ B
2494 \def\leftouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie}}

\rightouterjoin A \rightouterjoin B: A ⋉ B
2495 \def\rightouterjoin{\mathbin{\bowtie\mkern-5.8mu\o@join}}

\fullouterjoin A \fullouterjoin B: A ⋈⋉ B
2496 \def\fullouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie\mkern-5.8mu\o@join}}
2497

```


2.37 rmodell.sty

```
2498 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2499 \ProvidesPackage{lehramt-informatik-rmodell}[2020/09/01]
2500 Makros und Umgebungen zum Setzen des Relationenmodells beim Thema
2501 Datenbanken.]
2502 \RequirePackage{soul}
```

2.37.1 Makro-Kürzel

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

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

```
2503 \def\liPrimaer#1{\ul{#1}}
```

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

```
2504 \def\liFremd#1{{\setul{-0.9em}{}\ul{#1}}}
```

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

```
2505 \def\li@Rmodell@Schrift{\footnotesize\ttfamily}
2506 \ExplSyntaxOn
2507 \NewDocumentEnvironment { liRmodell }
2508 { +b }
2509 {
2510   \medskip
2511   {
2512     \linespread{2}
2513     \setlength{\parindent}{0pt}
2514     \li@Rmodell@Schrift#1
2515   }
2516   \medskip
2517 } {}
2518 \ExplSyntaxOff
```

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

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

```
2519 \def\liRelationMenge#1#2{
2520 \noindent
2521 #1 : \[ #2 ]\}
2522 \par
2523 }
```

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

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

```
2524 \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}
```

```
2525 \NewDocumentEnvironment { liRelationenSchemaFormat }{ +b } {} {}
```

```
2526
```

2.38 sortieren.sty

```
2527 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2528 \ProvidesPackage{lehramt-informatik-sortieren}[2020/06/10 Schaubilder
2529 für händisches Sortieren bzw. einen Schreibtischlauf setzen]
```

```
\tikz[
  rectangle split parts=5,
]{
  \node[li sortierung zahlenreihe] (reihe) {\nodepart{one} 2 \nodepart{two} 1 \nodepart{three} 3 \nodepart{four} 4 \nodepart{five} 5}
  \liSortierPfeil{one}{two}
  \liSortierPfeil{two}{three}
  \liSortierMarkierung{two split south}{three split north}
  \liSortierPfeilUnten{three}{one}
}
```



```
2530 \RequirePackage{tikz}
2531 \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.

```
2532 \def\liVertauschen#1{
2533   \directlua{
2534     local sortieren = require('lehramt-informatik-sortieren')
2535     sortieren('#1')
2536   }
2537 }
```

`\liSortierPfeil`

```
2538 \def\liSortierPfeil#1#2{
2539   \draw[-latex] ([xshift=1mm]reihe.#1 north) -- ++(0,0.25) -| ([xshift=-1mm]reihe.#2 north);
2540 }
```

`\liSortierPfeilUnten`

```
2541 \def\liSortierPfeilUnten#1#2{
2542   \draw[-latex] ([xshift=1mm]reihe.#1 south) -- ++(0,-0.25) -| ([xshift=-1mm]reihe.#2 south);
2543 }
```

`\liSortierMarkierung`

```
2544 \def\liSortierMarkierung#1#2{\node[
2545   draw,
2546   very thick,
2547   fit=(reihe.#1) (reihe.#2),
2548   inner sep=0pt
2549 ] {}
2550 }

2551 \tikzset{
2552   li sortierung zahlenreihe/.style={
2553     draw,
2554     thin,
2555     font=\large,
2556     rectangle split horizontal,
2557     rectangle split,
2558   }
2559 }
```

```

2560 % https://tex.stackexchange.com/a/140895
2561 \RequirePackage{forest,xstring}
2562 \usetikzlibrary{calc}
2563
2564 \makeatletter
2565 \pgfmathdeclarefunction{strrepeat}{2}{%
2566   \begingroup\pgfmathint{#2}\pgfmath@count\pgfmathresult
2567   \let\pgfmathresult\pgfutil@empty
2568   \pgfutil@loop\ifnum\pgfmath@count>0\relax
2569     \expandafter\def\expandafter\pgfmathresult\expandafter{\pgfmathresult#1}%
2570     \advance\pgfmath@count-1\relax
2571   \pgfutil@repeat\pgfmath@smuggleone\pgfmathresult\endgroup}
2572 \makeatother
2573
2574 \def\myNodes{}
2575
2576 \ExplSyntaxOn
2577 \newcommand*\sortList[1]{%
2578   \clist_sort:Nn#1{\int_compare:nNnTF{##1}>{##2}\sort_return_swapped:\sort_return_same:}}
2579 \ExplSyntaxOff
2580
2581 \forestset{
2582   sort/.code={%
2583     \pgfmathparse{level()>\forestSortLevel}%
2584     \ifnum\pgfmathresult=0
2585       \StrSubstitute{\forestov{content}}{ }{,}[\myList]%
2586       \sortList\myList
2587       \StrSubstitute{\myList}{,}{ }[\myList]%
2588       \pgfmathparse{strrepeat("1",level())}%
2589       \xappto\myNodes{\noexpand\node at ({\forestov{name}}|-m)!-!({\forestov{name}}$)
2590         (m\forestov{name}) {\myList}}%
2591       \pgfmathparse{level()=\forestSortLevel}%
2592       \ifnum\pgfmathresult=1
2593         \forestOget{\forestov{@first}}{name}\forestFirst
2594         \forestOget{\forestov{@last}}{name}\forestLast
2595         \xappto\myNodes{{[<-]edge (\forestOv{\forestov{@first}}{name})
2596           \ifx\forestFirst\forestLast\else edge (\forestOv{\forestov{@last}}{name})\fi}}%
2597       \fi
2598       \ifnum\forestov{@parent}=0\else
2599         \xappto\myNodes{edge (m\forestOv{\forestov{@parent}}{name})}%
2600       \fi
2601       \gappto\myNodes{;}%
2602     \fi}}
2603
2604 \forestset{sort level/.code=%
2605   \pgfmathparse{#1}\let\forestSortLevel\pgfmathresult
2606   \pgfmathparse{strrepeat("1",\forestSortLevel+1)}\let\forestOnes\pgfmathresult}
2607

```

2.39 spalten.sty

```
2608 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2609 \ProvidesPackage{lehramt-informatik-spalten}[2020/12/07 Lädt das Paket
2610 „multicol“, damit mehrspaltiger Satz mit Hilfe der Umgebung „multicols“
2611 realisiert werden kann.]
2612 \RequirePackage{multicol}
```

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

```
2613 \def\liSpaltenUmbruch{\vfill\strut\columnbreak}
```

```
2614
```

2.40 sql.sty

```
2615 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2616 \ProvidesPackage{lehramt-informatik-tabelle}[2021/09/02 Zu Setzen von SQL]
2617 \liLadePakete{syntax}
2618 \RequirePackage{fancyvrb}
2619 \DefineVerbatimEnvironment{liSqlErgebnis}{Verbatim}
2620 {fontsize=\footnotesize}
2621
```

2.41 struktogramm.sty

```
2622 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2623 \ProvidesPackage{lehramt-informatik-struktogramm}[2021/01/31 Lädt das
2624 Paket struktex zum Setzen von Struktogrammen]
2625 \RequirePackage{struktex}
2626
```

2.42 syntax.sty

```
2627 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2628 \ProvidesPackage{lehramt-informatik-syntax}[2020/06/10 Ein Hüll-Paket um
2629 `minted`, das sich um die Syntax-Hervorhebung des Java-Codes kümmert.]
2630 \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.

2.42.1 Makro-Kürzel

```
\let\j=\liJavaCode
\let\s=\liSqlCode
```

```
2631 \ExplSyntaxOn
2632 \directlua{
2633   syntax = require('lehramt-informatik-syntax')
2634   syntax.importiere_konfiguration('tex_repo_lokaler_pfad', '\LehramtInformatikRepository')
2635   syntax.importiere_konfiguration('github_domain', '\LehramtInformatikGithubDomain')
2636   syntax.importiere_konfiguration('github_raw_domain', '\LehramtInformatikGithubRawDomain')
2637   syntax.importiere_konfiguration('github_tex_repo', '\LehramtInformatikGithubTexRepo')
2638   syntax.importiere_konfiguration('github_code_repo', '\LehramtInformatikGithubCodeRepo')
2639   syntax.importiere_konfiguration('git_branch', '\LehramtInformatikGitBranch')
2640 }
2641 \RequirePackage{hyperref}
2642 \RequirePackage{minted}
2643 % pygmentize -L styles
2644 \usemintedstyle{colorful}
2645 %\BeforeBeginEnvironment{minted}{\begin{mdframed}}
2646 %\AfterEndEnvironment{minted}{\end{mdframed}}
2647 %\setminted{breaklines=true,linenos}
2648 \setminted{
2649   breaklines=true,
2650   linenos,
2651   fontsize=\footnotesize,
2652 }
```

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

Let-Abkürzung: `\let\j=\liJavaCode`

```
2653 \def\liJavaCode#1{\,\mintinline{java}|#1|\,}
```

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

```
2654 \def\liLatexCode#1{\mintinline{latex}|#1|}
```

```
2655 \def\li@GithubLink#1#2{
2656   \begin{flushright}
2657     \tiny
2658     Code-Beispiel~auf~Github~ansehen:~
2659     \href{#1}{\nolinkurl{#2}}
2660   \end{flushright}
2661 }
```

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

```
2662 \NewDocumentCommand{\liJavaDatei}{0{firstline=3} m }{
2663   \inputminted[#1]{java}{
2664     \directlua{
2665       syntax.drucke_absoluten_pfad('#2', false)
2666     }
2667   }
2668   \li@GithubLink
```

```

2669     {\directlua{syntax.drucke_github_url('#2', false)}}
2670     {\directlua{syntax.drucke_relativen_repo_pfad('#2', false)}}
2671 }

\liJavaTestDatei Eine komplette Java-Test-Datei einbinden, die Verzeichnis ./Code/src/test/java/org/bschlangaul
liegt.
2672 \NewDocumentCommand{\liJavaTestDatei}{ 0{firstline=3} m }{
2673   \inputminted[#1]{java}{
2674     \directlua{
2675       syntax.drucke_absoluten_pfad('#2', true)
2676     }
2677   }
2678   \li@GithubLink
2679   {\directlua{syntax.drucke_github_url('#2', true)}}
2680   {\directlua{syntax.drucke_relativen_repo_pfad('#2', true)}}
2681 }

\liJavaExamen \liJavaExamen{66116}{2015}{03}{Kunde} Regulärer Ausdruck zum Konvertieren: \liJavaDatei([.
\liJavaExamen$1{$2}{$3}{$4}{$5}
2682 \NewDocumentCommand{\liJavaExamen}{ 0{firstline=3} m m m m }{
2683   \inputminted[#1]{java}{
2684     \directlua{
2685       syntax.drucke_absoluten_examens_pfad('#2', '#3', '#4', '#5')
2686     }
2687   }
2688
2689   \li@GithubLink
2690   {\directlua{syntax.drucke_github_examens_url('#2', '#3', '#4', '#5')}}
2691   {\directlua{syntax.drucke_relativen_examens_repo_pfad('#2', '#3', '#4', '#5')}}
2692 }

\liAssemblerCode
2693 \def\liAssemblerCode#1{\mintinline{asm}|#1|}

\liAssemblerDatei \liAssemblerDatei{relativer-pfad}: Relativer Pfad zu einer Assembler-Datei.
2694 \NewDocumentCommand{\liAssemblerDatei}{ m }{
2695   \inputminted{asm}{#1}
2696 }

\liMinispracheDatei \liMinispracheDatei{relativer-pfad}: Relativer Pfad zu einer Minisprachen-Datei
(Hochsprache für die Minimaschine von Albert Wiedemann).
2697 \NewDocumentCommand{\liMinispracheDatei}{ m }{
2698   \inputminted{componentpascal}{#1}
2699 }

\liHaskellCode \liHaskellCode{haskell}: Zum Setzen von Haskell-Code.
2700 \def\liHaskellCode#1{\mintinline{haskell}|#1|}

\liHaskellDatei \liHaskellDatei{relativer-pfad}: Relativer Pfad zu einer Haskell-Datei.
2701 \NewDocumentCommand{\liHaskellDatei}{ m }{
2702   \inputminted{haskell}{#1}
2703 }

2704 \ExplSyntaxOff

\liSqlCode \liHaskellCode{sql}: Zum Setzen von SQL-Code.
Let-Abkürzung: \let\s=\liSqlCode
2705 \def\liSqlCode#1{\mintinline{sql}|#1|}
2706

```


2.43 syntaxbaum.sty

```
2707 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2708 \ProvidesPackage{lehramt-informatik-syntaxbaum}[2021/02/14 Zum Setzen von
2709 Syntaxbäumen mit Hilfe des Pakets tikz-qtrees]
2710 \RequirePackage{tikz-qtrees}
2711
2712 \tikzset{li parsetree/.style={
2713     every internal node/.style={
2714         draw,circle
2715     },
2716     every leaf node/.style={
2717         draw,rectangle
2718     },
2719 }
2720 }
2721
```

2.44 synthese-algorithmus.sty

```
2722 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2723 \ProvidesPackage{lehramt-informatik-synthese-algorithmus}[2021/03/19
2724 Hilfsmakros zum Setzen des Synthese-Algorithmuses zur Umformung einer
2725 Relation in die 3. Normalform]

2726 \liLadePakete{normalformen,mathe,typographie}
2727 \ExplSyntaxOn
```

2.44.1 Makro-Kürzel

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

2.44.2 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}
```

2.44.3 TeX-Markup Linksreduktion

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

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

 $\mathbf{D, X} \in \mathbf{ahl}\{C, E\}\{A, C, B, \mathbf{D, X}\} \setminus$ 
 $\mathbf{D, X} \notin \mathbf{ahl}\{C, E\}\{E, F\}$ 

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

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

2.44.4 TeX-Markup Rechtsreduktion

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

\liPseudoUeberschrift{F}

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

\liPseudoUeberschrift{A}

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

2.44.5 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

```
2728 \def\liSyntheseUeberschrift#1{
2729   {
2730     \bfseries
2731     \sffamily
2732     \str_case:nn {#1} {
2733       {1} {Kanonische-Überdeckung}
2734       {1-1} {Linksreduktion}
2735       {1-2} {Rechtsreduktion}
2736       {1-3} {Löschen-leerer-Klauseln}
2737       {1-4} {Vereinigung}
2738       {2} {Relationsschemata-formen}
2739       {3} {Schlüssel-hinzufügen}
2740       {4} {Entfernung-überflüssiger-Teilschemata}
2741     }
```

```

2742 }
2743 }

\liSyntheseErklaerung Let-Abkürzung: \let\erklaerung=\liSyntheseErklaerung
2744 \def\li@synthese@erklaerung@texte#1{
2745   \str_case:nn {#1} {
2746     {1} {
2747       Die~kanonische~Überdeckung~---also~die~kleinst~mögliche~noch~
2748       äquivalente~Menge~von~funktionalen~Abhängigkeiten~kann~in~vier~
2749       Schritten~erreicht~werden.
2750     }
2751     {1-1} {
2752       Führe~für~jede~funktionale~Anhängigkeit~
2753        $\alpha \rightarrow \beta$ ~in~F~die~Linksreduktion~durch,~
2754       überprüfe~also~für~alle~
2755        $A \in \alpha$ ,~ob~ $A$ ~überflüssig~ist,~d.h.~ob~
2756        $\beta \subseteqq \text{\liAttributHuelle}\{F, \alpha \cup A\}$ .
2757     }
2758     {1-2} {
2759       Führe~für~jede~(verbliebene)~funktionale~Abhängigkeit~ $\alpha \rightarrow \beta$ ~die~Rechtsreduktion~durch,~überprüfe~also~für~
2760       alle~ $B \in \beta$ ,~ob~ $B \in \text{\liAttributHuelle}\{F, (\alpha \rightarrow \beta) \cup (\alpha \rightarrow B)\}$ ,~
2761        $\alpha$ ~gilt.~In~diesem~Fall~ist~B~auf~der~rechten~Seite~
2762       überflüssig~und~kann~eliminiert~werden,~d.h.~ $\alpha \rightarrow \beta$ ~wird~durch~ $\alpha \rightarrow (\beta \cup B)$ ~
2763       ersetzt.
2764     }
2765     {1-3} {
2766       Entferne~die~funktionalen~Abhängigkeiten~der~Form~ $\alpha \rightarrow \emptyset$ ,~die~im~2.~Schritt~möglicherweise~
2767       entstanden~sind.
2768     }
2769     {1-4} {
2770       Fasse~mittels~der~Vereinigungsregel~funktionale~Abhängigkeiten~
2771       der~Form~ $\alpha \rightarrow \beta_{\{1\}}, \dots, \alpha \rightarrow \beta_{\{n\}}$ ,~so~dass~ $\alpha \rightarrow \beta_{\{1\}} \cup \dots \cup \beta_{\{n\}}$ ~
2772       verbleibt.
2773     }
2774     % Kemper Seite 197
2775     {2} {
2776       Erzeuge~für~jede~funktionale~Abhängigkeit~ $\alpha \rightarrow \beta$ ~in~F~ein~Relationenschema~ $\mathcal{R}_{\alpha}$ ~
2777        $:= \alpha \cup \beta$ .
2778     }
2779     {3} {
2780       Falls~eines~der~in~Schritt~2.~erzeugten~Schemata~ $\mathcal{R}_{\alpha}$ ~
2781       einen~Schlüsselkandidaten~von~ $\mathcal{R}$ ~bezüglich~ $F$ ~
2782       enthält,~sind~wir~fertig,~sonst~wähle~einen~Schlüsselkandidaten~
2783        $\mathcal{K} \subseteqq \mathcal{R}$ ~aus~und~definiere~folgendes~
2784       zusätzliche~Schema:~ $\mathcal{R}_{\mathcal{K}} := \mathcal{K}$ ~
2785       und~ $\mathcal{F}_{\mathcal{K}} := \emptyset$ 
2786     }
2787     {4} {
2788       Eliminiere~diejenigen~Schemata~ $\mathcal{R}_{\alpha}$ ,~die~in~einem~
2789       anderen~Relationenschema~ $\mathcal{R}_{\alpha'}$ ~enthalten~sind,~d.h.~
2790        $\mathcal{R}_{\alpha} \subseteqq \mathcal{R}_{\alpha'}$ .
2791     }
2792   }
2793 }
2794 \def\liSyntheseErklaerung#1{
2795   {
2796     \itshape
2797     \footnotesize

```

```

2804     \liParagraphMitLinien{\li@synthese@erklaerung@texte{#1}}
2805   }
2806 }

```

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

```

2807 \def\liSyntheseUeberErklaerung#1{
2808   \liSyntheseUeberschrift{#1}\par
2809   \liSyntheseErklaerung{#1}
2810 }

```

```

2811 \ExplSyntaxOff
2812

```

2.45 tabelle.sty

2813 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2814 \ProvidesPackage{lehramt-informatik-tabelle}[2020/12/05 Lädt das Paket tabluarx]

2815 \RequirePackage{tabularx}

2816

2.46 typographie.sty

```
2817 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2818 \ProvidesPackage{lehramt-informatik-typographie}[2021/03/20 Typographische Makros,
2819 die das Erscheinungsbild verändern. Die Schriftdefinition sind in
2820 formatierung.sty definiert.]
```

```
2821 \ExplSyntaxOn
```

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

```
2822 \RequirePackage{fontawesome}
```

```
\liErledigt \liErledigt: ☑
```

```
2823 \let\liErledigt=\faCheckSquareO
```

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

```
2824 \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 —

```
2825 \def\liParagraphMitLinien#1{
```

```
2826 \noindent
```

```
2827 \vrule height 2pt depth -1.6pt width 0.4cm
```

```
2828 \enspace
```

```
2829 #1
```

```
2830 \enspace
```

```
2831 \leaders\vrule height 2pt depth -1.6pt \hfill \null
```

```
2832 \par
```

```
2833 \medskip
```

```
2834 }
```

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

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

```
2835 \def\liGeschweifteKlammern#1#2#3#4{
```

```
2836 \par
```

```
2837 \medskip
```

```
2838 \noindent
```

```
2839 #1 \, $= \Bigl\{ $\
```

```
2840 \vspace{#3}
```

```
2841 #2
```

```
2842 \vspace{#4}
```

```
2843 \begin{flushright} $\Bigr\} $\end{flushright}
```

```
2844 \par
```

```
2845 }
```

```
\liTypoUeberschrift
```

```
2846 \def\liTypoUeberschrift#1{
```

```
2847 {
```

```
2848 \bfseries\sffamily
```

```
2849 #1
```

```
2850 }
```

```
2851 }
```

\liTypoUeberGross

```
2852 \def\liTypoUeberGross#1{
2853   {
2854     \huge
2855     \liTypoUeberschrift{#1}
2856   }
2857 }
```

\liTypoUeberGross

```
2858 \def\liTypoUeberGROSS#1{
2859   {
2860     \Huge
2861     \liTypoUeberschrift{#1}
2862   }
2863 }
```

2864 \ExplSyntaxOff

2865

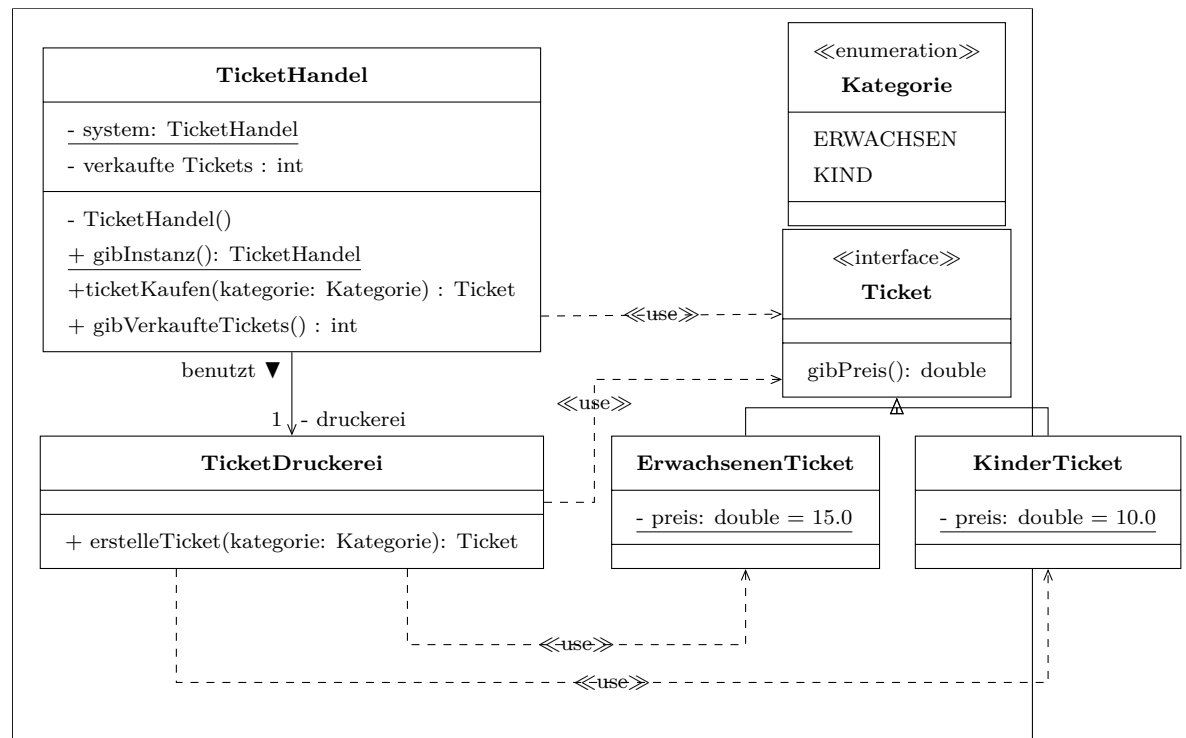
2.47 uml.sty

```

2866 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2867 \ProvidesPackage{lehramt-informatik-uml}[2020/06/13 Ein Hüll-Paket um
2868 `tikz-uml', das einige Design-Einstellungen vornimmt und manche
2869 Erweiterung bereitstellt]

2870 \RequirePackage{tikz-uml}
2871 \RequirePackage{tikz-uml-activity}
2872 % Not compatible with wasysym
2873 %\RequirePackage{mathabx}
2874 \RequirePackage{wasysym}
2875 \usetikzlibrary{positioning}

2876 \tikzumlset{
2877   fill class=white!0,
2878   font=\footnotesize,
2879   fill object=white!0,
2880   fill note=white!0,
2881   fill state=white!0,
2882   % Use case
2883   fill usecase=white!0,
2884   fill system=white!0,
2885 }
```



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

```

2886 \NewDocumentCommand{ \liUmlLeserichtung } { 0{dir=right} m } {
2887   \def\@liDirLeft{}
2888   \def\@liDirRight{ \RIGHTarrow}
2889   \pgfkeys{/lese/dir/.is choice}
2890   \pgfkeys{/lese/dir/up/.code={\def\@liDirRight{ \UParrow}}}
2891   \pgfkeys{/lese/dir/down/.code={\def\@liDirRight{ \DOWNarrow}}}
2892   \pgfkeys{/lese/dir/left/.code={\def\@liDirRight{}\def\@liDirLeft{ \LEFTarrow }}}
2893   \pgfkeys{/lese/dir/right/.code={\def\@liDirRight{ \RIGHTarrow}}}
2894
2895   \def\@liPos{above}
2896   \pgfkeys{/lese/pos/.code={\def\@liPos{##1}}}
2897

```

```

2898 \def\@liDistance{0cm}
2899 \pgfkeys{/lese/distance/.code={\def\@liDistance{##1}}}}
2900
2901 \pgfkeys{/lese/.cd,#1}
2902
2903 \node[\@liPos = \@liDistance of #2-middle] {
2904   \@liDirLeft{\footnotesize#2}\@liDirRight
2905 };
2906 }
2907

```

2.48 vollstaendige-induktion.sty

2908 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2909 \ProvidesPackage{lehramt-informatik-vollstaendige-induktion}[2021/07/01
 2910 Hilfsmakros zum Setzen der Vollständigen Induktion, vor allem die
 2911 Überschriften für die einzelnen Schritte]

2.48.1 Makro-Kürzel

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

Lade häufig benötigte Pakete

2912 \RequirePackage{lehramt-informatik-typographie}
 2913 \RequirePackage{lehramt-informatik-mathe}
 2914 \RequirePackage{lehramt-informatik-syntax}
 2915 \ExplSyntaxOn

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

Let-Abkürzung: \let\m=\liInduktionMarkierung

2916 \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

```
2917 \def\liInduktionErklaerung#1{\scriptsize\text{#1}}
```

\liInduktionAnfang

```
2918 \def\liInduktionAnfang{
```

```
2919   \liPseudoUeberschrift{Induktionsanfang}
```

```
2920
```

```
2921   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
2922   \liParagraphMitLinien{
```

```
2923     Beweise,~dass~ $A(1)$ ~eine~wahre~Aussage~ist.
```

```
2924   }
```

```
2925 }
```

\liInduktionVoraussetzung

```
2926 \def\liInduktionVoraussetzung{
```

```
2927   \liPseudoUeberschrift{Induktionsvoraussetzung}
```

```
2928
```

```
2929   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
2930   \liParagraphMitLinien{
```

```
2931     Die~Aussage~ $A(k)$ ~ist~wahr~für~ein~beliebiges~ $k \in \mathbb{N}$ .
```

```
2932   }
```

```
2933 }
```

\liInduktionSchritt

```
2934 \def\liInduktionSchritt{
```

```
2935   \liPseudoUeberschrift{Induktionsschritt}
```

```
2936
```

```
2937   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
2938   \liParagraphMitLinien{
```

```
2939     Beweise,~dass~wenn~ $A(n=k)$ ~wahr~ist,~
```

```
2940     auch~ $A(n=k+1)$ ~wahr~sein~muss.
```

```
2941   }
```

```
2942 }
```

```
2943 \ExplSyntaxOff
```

```
2944
```

2.49 wasserfall.sty

```
2945 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2946 \ProvidesPackage{lehramt-informatik-wasserfall}[2020/06/10]
2947 \RequirePackage{tikz}
2948 \tikzset{wasserfall/.style={
2949   >=stealth,
2950   node distance = 2mm and -8mm,
2951   start chain = A going below right,
2952   every node/.style = {
2953     draw,
2954     text width=24mm,
2955     minimum height=12mm,
2956     align=center,
2957     inner sep=1mm,
2958     fill=white,
2959     drop shadow={fill=black},
2960     on chain=A
2961   },
2962 }}
2963 \usetikzlibrary{chains,positioning,shadows}
2964
```

2.50 wpkalkuel.sty

```
2965 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2966 \ProvidesPackage{lehramt-informatik-wpkalkuel}[2020/06/13]
```

2.50.1 Makro-Kürzel

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

```
2967 \RequirePackage{amsmath}
2968 \ExplSyntaxOn
```

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

```
2969 \def\liWpKalkuelOhneMathe#1#2{
2970   \text{wp}(\texttt{\scriptsize"#1"},\thinspace #2)
2971 }
2972 \def\liWpKalkuel#1#2{
2973   \ifmmode
2974     \liWpKalkuelOhneMathe{#1}{#2}
2975   \else
2976     $\liWpKalkuelOhneMathe{#1}{#2}$
2977   \fi
2978 }
```

\MatheEnv

```
2979 \def\MatheEnv#1{
2980   \medskip
2981
2982   \hspace{1em}#1
2983
2984   \medskip
2985 }
```

\Mathe

```
2986 \def\Mathe#1{
2987   \MatheEnv{${#1}$}
2988 }
```

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

```
2989 \def\liWpEquivalent#1{
2990   \MatheEnv{${equiv$\hspace{1em}${#1}$}
2991 }
```

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

```
2992 \newlength{\@Skip@Erklaerung@Reset}
2993 \def\liWpErklaerung#1{
2994   \setlength{\@Skip@Erklaerung@Reset}{\leftskip}
2995   \setlength{\leftskip}{0.5cm}
2996
2997   \par
2998   \noindent
2999   {
3000     \scriptsize
3001     #1
3002   }
3003   \par
3004
3005   \setlength{\leftskip}{\@Skip@Erklaerung@Reset}
3006 }
```

\liWpErklaerungVerzweigung

```

3007 \def\liWpErklaerungVerzweigung{
3008   $\liWpKalkuelOhneMathe{if~\{-b~\}~then~\{-a1~\}~else~\{-a2~\}}{Q}
3009   \equiv
3010   (b \land \liWpKalkuelOhneMathe{a1}{Q})
3011   \lor
3012   (\neg b \land \liWpKalkuelOhneMathe{a2}{Q})$
3013 }

3014 \ExplSyntaxOff

3015

```

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			
\#	114	\AddToHook	1417, 1471, 2762, 2765, 2775, 2776, 2777, 2782, 2783
\,	356, 411, 1238, 1240, 1837, 1838, 1839, 2176, 2653, 2839	\advance	2570
\@Skip@Erklaerung@Reset	2992, 2994, 3005	\AfterEndEnvironment	2646
\@afterheading	1804	\allsectionsfont	1287
\@afterindentfalse	1804	\Alph	1301
\@liDirLeft	2887, 2892, 2904	\alph	1301, 1302
\@liDirRight	2888, 2890, 2891, 2892, 2893, 2904	\alpha	2753, 2755, 2756, 2759, 2761, 2762, 2763, 2764, 2765, 2769, 2775, 2776, 2781, 2782, 2783, 2786, 2794, 2795, 2796
\@liDistance	2898, 2899, 2903	\arabic	1301, 2365, 2370, 2375, 2381, 2387, 2393
\@liPos	2895, 2896, 2903	\arraystretch	2066
\\	618, 646, 647, 650, 651, 654, 655, 750, 751, 752, 859, 893, 895, 925, 934, 979, 1021, 1022, 1023, 1028, 1029, 1030, 1050, 1711, 2071, 2129, 2132		
\{	212, 1175, 1185, 1197, 1198, 1203, 1237, 1551, 2116, 2521, 2839, 3008	B	
\}	212, 1175, 1185, 1197, 1198, 1205, 1241, 1552, 2116, 2521, 2843, 3008	\BeforeBeginEnvironment	2645
_	37, 45, 50, 52, 303, 331, 355, 356, 370, 371, 377, 380, 383, 393, 405, 1418, 1468	\begin	644, 698, 713, 748, 772, 819, 851, 866, 891, 901, 921, 941, 973, 990, 1019, 1044, 1065, 1089, 1104, 1218, 1307, 1382, 1472, 1478, 1482, 1486, 1505, 1514, 1521, 1629, 1703, 1755, 1760, 1768, 1793, 1798, 1809, 1819, 1823, 1906, 1910, 1929, 1952, 1975, 1990, 2069, 2070, 2120, 2165, 2285, 2305, 2446, 2468, 2482, 2645, 2656, 2843
_	2128, 2138	\begingroup	1704, 2118, 2566
		\beschriftung	1735, 1739, 1743, 1747, 1751, 1753
A		\beta	2753, 2756, 2760, 2761,
\addbibresource	1681, 1682, 1683, 1684, 1685, 1686, 1687, 1688, 1689, 1690		
		C	
		\c	1384, 1385
		\cdot	1881, 1937, 1948
		\centerline	1510, 2444, 2466, 2481
		\chapter	1291, 1292
		\char	1711
		\clearpage	1435, 1492, 1705
		\cline	618
		\clist	231, 277, 278, 300, 304, 2578
		\columnbreak	2613
		\cs	303, 331, 355, 356, 393, 405, 1787
		\csname	1369, 1372
		\cup	1198, 2150, 2762, 2776, 2783
		D	
		\DeclareMathSymbol	2014, 2015
		\DecoINERT	2368, 2441, 2442, 2464
		\DecoINERTwithPivot	2384, 2463
		\DecoLEFT	2363, 2462

<code>\DecoLEFTwithPivot</code> ..	environments:	<code>\footcite</code> 641, 710, 732,
..... 2378, 2440	liAdditum 1758	779, 802, 834, 887,
<code>\DecoRIGHT</code> 2373, 2465, 2475	liAHuelle 2117	938, 965, 1011,
<code>\DecoRIGHTwithPivot</code> .	liAntwort 1729	1530, 1533, 1540,
..... 2390, 2443	liDiagramm 1817	1545, 1550, 1554,
<code>\definecolor</code> 1289	liEinbettung 1728	1560, 1565, 1670,
<code>\DefineVerbatimEnvironment</code>	liExkurs 1766	1925, 1926, 2081, 2298
..... 2619	liGraphenFormat . 1361	<code>\footnote</code> 1831, 1835
<code>\delta</code> 70, 112, 170, 212, 1189	liKasten 1306	<code>\footnotesize</code> 152, 366,
<code>\dh</code> 1839, 2764	liKontrollflussgraph	448, 542, 1162,
<code>\directlua</code> 1628	1726, 1779, 1794,
63, 142, 200, 205,	liLernkartei 1807	1810, 2025, 2119,
1170, 1184, 1204,	liProduktionsRegeln	2136, 2144, 2253,
1212, 1219, 1224, 1213	2262, 2505, 2620,
2101, 2106, 2159,	liProjektSprache 1727	2651, 2803, 2878, 2904
2166, 2173, 2533,	liQuellen 1787	<code>\footrulewidth</code> . 1318, 1663
2632, 2664, 2669,	liRelationenSchemaFormat	<code>\foreach</code> . 1384, 1387, 1394
2670, 2674, 2679, 2525	<code>\forestFirst</code> .. 2593, 2596
2680, 2684, 2690, 2691	liRmodell 2505	<code>\forestLast</code> ... 2594, 2596
<code>\do</code> 2364, 2369,	liUebergangsTabelle	<code>\forestOget</code> ... 2593, 2594
2374, 2379, 2385, 2391 2066	<code>\forestOnes</code> 2606
<code>\dots</code> 528, 532,	<code>\equiv</code> 2990, 3009	<code>\forestOv</code> 2595, 2596, 2599
1551, 2292, 2775, 2776	<code>\erzeuge@tiefgestellt</code>	<code>\forestov</code> . 2585, 2589,
<code>\DOWNarrow</code> 2891	... 1184, 1185, 1189	2590, 2593, 2594,
<code>\draw</code> 1389, 1392,	<code>\expandafter</code>	2595, 2596, 2598, 2599
1395, 2240, 2539, 2542	1369, 2399, 2401,	<code>\forestset</code> 2581, 2604
E	2402, 2403, 2411, 2569	<code>\forestSortLevel</code>
<code>\edef</code> 1380,	<code>\ExplSyntaxOff</code> ... 54,	2583, 2591, 2605, 2606
2457, 2461, 2473, 2474	97, 139, 144, 197,	<code>\frac</code> 1883, 1916, 1948, 1963
<code>\edge</code> 274	202, 207, 550, 572,	<code>\fullouterjoin</code> 2496
<code>\else</code> 592, 600,	587, 1165, 1243,	
608, 1179, 1193,	1278, 1420, 1494,	G
1229, 1751, 1854,	1649, 1665, 1829,	<code>\g</code> 39, 277,
1864, 1874, 1888,	1904, 2094, 2268,	278, 300, 304, 310,
2112, 2149, 2235,	2518, 2579, 2704,	311, 312, 313, 314,
2403, 2596, 2598, 2975	2811, 2864, 2943, 3014	316, 317, 319, 320,
<code>\emph</code> .. 639, 886, 1141,	<code>\ExplSyntaxOn</code> .. 35, 66,	322, 323, 324, 325,
1532, 1561, 1563, 1710	107, 140, 165, 198,	326, 327, 328, 329,
<code>\empty</code> 1751, 2148	203, 228, 493, 556,	332, 336, 337, 338,
<code>\emptyset</code>	573, 1138, 1233,	341, 343, 344, 345,
2058, 2770, 2791, 2824	1255, 1416, 1440,	346, 347, 348, 359,
<code>\end</code> ... 683, 709, 734,	1642, 1654, 1730,	360, 361, 362, 370,
769, 803, 835, 863,	1843, 2076, 2249,	371, 373, 379, 380,
879, 898, 906, 937,	2506, 2576, 2631,	382, 383, 385, 386,
966, 987, 1012,	2727, 2821, 2915, 2968	394, 399, 401, 406,
1037, 1057, 1076,	F	408, 412, 1442,
1101, 1119, 1220,	<code>\faCheckSquare0</code> 2823	1446, 1450, 1454,
1309, 1398, 1474,	<code>\faCircleThin</code> 1157	1455, 1456, 1457,
1480, 1484, 1488,	<code>\faGg</code> 1151	1458, 1460, 1461,
1517, 1518, 1525,	<code>\fancyfoot</code>	1463, 1479, 1483, 1487
1634, 1708, 1757,	1313, 1314, 1315,	<code>\Gamma</code> . 111, 169, 212, 1198
1765, 1784, 1800,	1658, 1659, 1660, 1661	<code>\gappto</code> 2601
1801, 1815, 1826,	<code>\fancyhead</code>	<code>\geometry</code> 5
1827, 1908, 1924,	1312, 1655, 1656, 1657	<code>\geq</code> 1558, 1914,
1949, 1972, 1987,	<code>\faSquare0</code> 1145	2281, 2286, 2302, 2306
1997, 2073, 2074,	<code>\fi</code> 594, 602, 610, 1181,	H
2122, 2167, 2295,	1195, 1231, 1754,	<code>\hbox</code> 2491
2315, 2446, 2468,	1856, 1866, 1876,	<code>\headrulewidth</code> . 1317, 1662
2482, 2646, 2660, 2843	1890, 2114, 2151,	<code>\headwidth</code> 1664
<code>\endcsname</code> 1369, 1372	2237, 2404, 2596,	<code>\hfill</code> 1933, 1939, 1944, 2831

<code>\href</code>	395, 1251, 1835, 2001, 2659	109, 110, 111, 112, 113, 114, 115, 118, 119, 120, 121, 122, 123, 124, 130, 131, 132, 133, 134, 135, 136, 167, 168, 169, 170, 171, 172, 173, 176, 177, 178, 179, 180, 181, 182, 188, 189, 190, 191, 192, 193, 194, 558, 561, 566, 567, 570, 575, 576, 579, 580, 585, 1257, 1258, 1259, 1260, 1263, 1264, 1265, 1266, 1272, 1273, 1274, 1275, 1645, 1646, 1647, 1790, 1791, 1792, 1799	2464, 2465, 2475, 2567, 2605, 2606, 2823
<code>\hspace</code> ..	2243, 2982, 2990		<code>\li@chomsky@erklaerung@texte</code>
<code>\ht</code>	2492		507, 543
<code>\Huge</code>	2860		<code>\li@EntwurfsCode</code>
<code>\huge</code>	1434, 2854		635, 686, 687, 688, 737, 738, 739, 740, 806, 807, 808, 809, 810, 811, 838, 839, 840, 841, 842, 843, 909, 1079, 1080, 1081, 1082, 1122, 1123
I			
<code>\i</code>	1394, 1395		<code>\li@EntwurfsCodeAllgemein</code>
<code>\ifcase</code>	2400		634
<code>\ifmmode</code>	590, 598, 606, 1177, 1191, 1227, 1852, 1862, 1872, 1886, 2110, 2233, 2973		<code>\li@fussnote@text</code> 2022, 2028, 2032, 2036, 2040
<code>\IfNoValueTF</code>			<code>\li@GithubLink</code>
... 1761, 1831, 1835			2655, 2668, 2678, 2689
<code>\ifnum</code>	2327, 2568, 2584, 2592, 2598		<code>\li@mget</code> . 1371, 1375, 1395
<code>\ifx</code>	1751, 2148, 2596		<code>\li@minc</code> 1374, 1396
<code>\in</code>	516, 622, 1558, 1896, 1899, 1902, 1931, 1937, 1942, 2281, 2292, 2302, 2312, 2753, 2755, 2761, 2782, 2931	<code>\labelenumi</code> 1302 <code>\labelenumii</code> 1303 <code>\labelitemi</code> 1296 <code>\labelitemii</code> 1297 <code>\labelitemiii</code> 1298 <code>\labelitemiv</code> 1299	<code>\li@mset</code> 1368, 1376, 1385, 1388
<code>\inhaltsverzeichnis</code>	1702	<code>\land</code> 3010, 3012 <code>\LARGE</code> 1291 <code>\large</code> 1510, 2555 <code>\leaders</code> 2831 <code>\left</code> 1846 <code>\LEFTarrow</code> 2892 <code>\leftarrow</code> 604 <code>\leftouterjoin</code> 2494 <code>\leftskip</code> 2994, 2995, 3005 <code>\LehramtInformatikAutorEmail</code> 1661 <code>\LehramtInformatikAutorName</code> 1660 <code>\LehramtInformatikGitBranch</code> 398, 2639 <code>\LehramtInformatikGithubCodeRepo</code> 2638 <code>\LehramtInformatikGithubDomain</code> 2635 <code>\LehramtInformatikGithubRawDomain</code> 2113, 2127, 2137, 2145 396, 2636 <code>\LehramtInformatikGithubTexRepo</code> 397, 2637	<code>\li@numdiscs</code> 1380, 1389, 1395 <code>\li@Rmodell@Schrift</code> 2505, 2514, 2524 <code>\li@sequence</code> .. 1381, 1394 <code>\li@synthese@erklaerung@texte</code> 2744, 2804 <code>\li@Ableitung</code> 1212 <code>liAdditum</code> (environment) 1758 <code>liAHuelle</code> (environment) 2117 <code>\li@Alphabet</code> 1197 <code>liAntwort</code> (environment) 1729 <code>\li@Anweisung</code> 1636 <code>\li@AssemblerCode</code> ... 2693 <code>\li@AssemblerDatei</code> .. 2694 <code>\li@Attribut</code> 2524 <code>\li@AttributHuelle</code> 2108, 2756, 2761 <code>\li@AttributHuelleOhneMathe</code> 2108, 2111, 2138, 2139, 2153, 2155 <code>\li@AttributMenge</code>
<code>\input</code>	17, 20, 23, 26, 29, 418, 1451		<code>\li@Aufgabe</code> 16 17, 20, 23, 26, 29, 1452, 1681, 1682, 1683, 1684, 1685, 1686, 1687, 1688, 1689, 1690, 2634
<code>\inputminted</code>	2663, 2673, 2683, 2695, 2698, 2702	<code>\LehramtInformatikTitel</code> 1656 <code>\leq</code> 1948, 2289, 2309 <code>\let</code> 1187, 1188, 1705, 2440, 2441, 2442, 2443, 2456, 2458, 2459, 2460, 2462, 2463,	<code>\li@AufgabenMetadaten</code> 47, 1466 <code>\li@AufgabenTitel</code> 55 <code>\li@Ausdruck</code> 1234 <code>\li@Automat</code> 66 <code>\li@AutomatenKante</code> 98 <code>\li@BandAlphabet</code> 1198 <code>\li@Bedingung</code> 1637 <code>\li@BedingungDrei</code> 1901, 1944, 1984 <code>\li@BedingungEins</code> 1895, 1933, 1976 <code>\li@BedingungFalsch</code> . 1639
<code>\int</code>	2578		
<code>\item</code>	487, 488, 715, 719, 724, 729, 773, 782, 787, 795, 867, 872, 876, 902, 942, 947, 954, 962, 991, 996, 1000, 1005, 1105, 1110, 1115, 1515, 1516, 1787, 1791, 1911, 1916, 1920, 1930, 1936, 1941, 1953, 1957, 1961, 1965, 1969, 1976, 1980, 1984, 2286, 2289, 2292, 2306, 2309, 2312		
<code>\itshape</code>	541, 2802		
J			
<code>\j</code>	1384, 1385, 1387, 1388, 1389, 1394, 1395, 1396		
K			
<code>\k</code>	1394		
<code>\keys</code>	41, 75, 87, 117, 127, 175, 185, 308, 560, 564, 578, 583, 1262, 1269		
L			
<code>\l</code>	68, 69, 70, 71, 72, 73, 76, 77, 78, 79, 80, 82, 84, 89, 90, 91, 92, 93, 94,		

\liBedingungWahr ...	1638	\liEntwurfsEinfacheFabrikUml	liExkurs (environment)	1766
\liBedingungZwei ...	1898, 1939, 1980	\liEntwurfsEinzelstueck	\liFalsch	488
\liBeschriftung	1718	\liEntwurfsEinzelstueckAkteure	\liFlaci	1244
\liBindeAufgabeEin	1449	\liEntwurfsEinzelstueckBeschreibung	\liFremd	2504
\liChomskyErklaerung	507, 548	\liEntwurfsEinzelstueckCode	\liFunktionaleAbhaengigkeit	2147, 2150, 2158
\liChomskyUeberErklaerung	546	\liEntwurfsEinzelstueckUml	\liFunktionaleAbhaengigkeiten	2161
\liChomskyUeberschrift	495, 547	\liEntwurfsErbauer	\liFussnote	2021, 2023
\liCpmEreignis	556	\liEntwurfsErbauerAkteure	\liFussnoteDreiText	2035, 2053
\liCpmFruehesterI	613	\liEntwurfsErbauerUml	\liFussnoteEinsText	2027, 2047
\liCpmSpaetesterI	612	\liEntwurfsFabrikmethode	\liFussnoteLink	1834
\liCpmVon	596	\liEntwurfsFabrikmethodeAkteure	\liFussnoten	2043
\liCpmVonOhneMathe	596, 599, 601	\liEntwurfsFabrikmethodeUml	\liFussnoteUrl	1058, 1830
\liCpmVonZu	588	\liEntwurfsKompositum	\liFussnoteVierText	2039, 2056
\liCpmVonZuOhneMathe	588, 591, 593	\liEntwurfsKompositumAkteure	\liFussnoteZweiText	2031, 2050
\liCpmVorgang	573	\liEntwurfsKompositumUml	\liGeschweifteKlammern	1216, 2162, 2835
\liCpmZu	604	\liEntwurfsModellPraesentation	\liGrammatik	1255
\liCpmZuOhneMathe	604, 607, 609	\liEntwurfsModellPraesentationAkteure	liGraphenFormat (environment)	1361
liDiagramm (environment)	1817	\liEntwurfsModellPraesentationUml	\liHanoi	1368
liEinbettung (environment)	1728	\liEntwurfsModellSteuerung	\liHaskellCode	2700
\liEntwurfs	1060	\liEntwurfsModellSteuerungAkteure	\liHaskellDatei	2701
\liEntwurfsAbstrakteFabrik	690	\liEntwurfsModellSteuerungUml	\liInduktionAnfang	2918
\liEntwurfsAbstrakteFabrikBeschreibung	638, 691	\liEntwurfsModellSteuerungUml	\liInduktionErklaerung	2917
\liEntwurfsAbstrakteFabrikCode	685, 695	\liEntwurfsModellSteuerungUml	\liInduktionMarkierung	2916
\liEntwurfsAbstrakteFabrikUml	643, 693	\liEntwurfsModellSteuerungUml	\liInduktionSchritt	2934
\liEntwurfsAdapter	742	\liEntwurfsModellSteuerungUml	\liInduktionVoraussetzung	2926
\liEntwurfsAdapterAkteure	712, 744	\liEntwurfsModellSteuerungUml	\liJavaCode	2653
\liEntwurfsAdapterCode	736, 745	\liEntwurfsModellSteuerungUml	\liJavaDatei	636, 2662
\liEntwurfsAdapterUml	697, 743	\liEntwurfsModellSteuerungUml	\liJavaExamen	2682
\liEntwurfsBeobachter	813	\liEntwurfsModellSteuerungUml	\liJavaTestDatei	2672
\liEntwurfsBeobachterAkteure	771, 815	\liEntwurfsModellSteuerungUml	liKasten (environment)	1306
\liEntwurfsBeobachterCode	805, 816	\liEntwurfsModellSteuerungUml	\liKellerAutomat	107
\liEntwurfsBeobachterUml	747, 814	\liEntwurfsModellSteuerungUml	\liKellerKante	145
\liEntwurfsDekorierer	845	\liEntwurfsModellSteuerungUml	\liKellerUebergang	140, 146
\liEntwurfsDekoriererAkteure	847	\liEntwurfsModellSteuerungUml	\liKontrollCode	1640
\liEntwurfsDekoriererCode	837, 848	\liEntwurfsModellSteuerungUml	liKontrollflussgraph (environment)	1628
\liEntwurfsDekoriererUml	818, 846	\liEntwurfsModellSteuerungUml	\liKontrollKnotenPfad	1642
\liEntwurfsEinfacheFabrik	881	\liEntwurfsModellSteuerungUml	\liKontrollTextzeileKnoten	1641, 1646
\liEntwurfsEinfacheFabrikAkteure	865, 883	\liEntwurfsModellSteuerungUml	\liKurzeTabellenLinie	618
\liEntwurfsEinfacheFabrikUml	865, 883	\liEntwurfsModellSteuerungUml	\liLadeAllePakete	233, 1437
\liEntwurfsEinfacheFabrikUml	865, 883	\liEntwurfsModellSteuerungUml	\liLadePakete	59, 62, 229, 234, 494, 555, 1174, 1404, 1425, 1500, 2020, 2100, 2248, 2617, 2726
\liEntwurfsEinfacheFabrikUml	865, 883	\liEntwurfsModellSteuerungUml	\liLatexCode	2654
\liEntwurfsEinfacheFabrikUml	865, 883	\liEntwurfsModellSteuerungUml	\liLeereZelle	2058

\liLernkartei (environ- ment)	1807	\liProblemVertexCover	1527 , 1535	\liTuringUebergangZelle	198
\liLinksReduktion ..	2125	\liProduktionen	1223 , 1265	\liTypoUeberGROSS ..	1479 , 1483 , 1487 , 2858
\liLinksReduktionInline	2134 , 2142	liProduktionsRegeln (environment)	1213	\liTypoUeberGross ..	1473 , 2852 , 2858
\liMasterExkurs	1989	liProjektSprache (envi- ronment)	1727	\liTypoUeberschrift	2846 , 2855 , 2861
\liMasterFaelle	1928 , 1996	\liPseudoUeberschrift	1712 , 1762 , 1763 , 2068 , 2078 , 2919 , 2927 , 2935	\liUeberfuehrungsFunktion	1189
\liMasterFallRechnung	1974	\liPumpingKontextfrei	2300	\liUeberfuehrungsFunktionOhneMathe	1189 , 1192 , 1194
\liMasterVariablen ..	1905 , 1991	\liPumpingRegulaer .	2279	liUebergangsTabelle (environment)	2066
\liMasterVariablenDeklaration	1951	liQuellen (environment)	1787	\liUeberschriftDreiecksTabelle	2076
\liMasterWolframLink	1999	\liRechtsReduktionInline	2142	\liUmlLeserichtung .	2886
\liMenge	76 , 77 , 79 , 118 , 119 , 120 , 124 , 176 , 177 , 178 , 182 , 1175 , 1224 , 1263 , 1264	\liRekursionsGleichung	1892 , 1955	\liVertauschen	2532
\liMengeOhneMathe	1175 , 1178 , 1180	\liRelation	2172	\liWortInSprache	619
\liMetaSetze	36 , 48 , 1467	liRelationenSchemaFormat (environment)	2525	\liWortNichtInSprache	624
\liMinimierungErklaerung	2080	\liRelationMenge . . .	2519	\liWpEquivalent	2989
\liMinispracheDatei	2697	\liRichtig	487	\liWpErklaerung	2992
\linespread	2512	liRmodell (environment)	2505	\liWpErklaerungVerzweigung	3007
\liNichtsZuTun	2824	\liRundeKlammer .	1845 , 1849 , 1859 , 1869 , 1883	\liWpKalkuel	2969
\liO	1868 , 1896	\liSetzeExamenTeilaufgabeNr	1445	\liWpKalkuelOhneMathe	2969 , 2974 , 2976 , 3008 , 3010 , 3012
\liOmega	1858 , 1902	\liSetzeExamenThemaNr	1441	\liZustandsBuchstabe	1199 , 1208 , 1210 , 1228 , 1230
\liOmegaOhneMathe	1858 , 1863 , 1865	\liSortierMarkierung	2544	\liZustandsBuchstabeGross	1200 , 1209 , 1211
\liOOhneMathe	1868 , 1873 , 1875	\liSortierPfeil	2538	\liZustandsmenge . . .	1187
\liParagraphMitLinien	543 , 2082 , 2804 , 2825 , 2922 , 2930 , 2938	\liSortierPfeilUnten	2541	\liZustandsmengeNr ..	1201 , 2263
\liPetriErreichKnotenDrei	2242	\liSpaltenUmbruch . .	2613	\liZustandsmengeNrGross	1209
\liPetriErreichTransition	2239	\liSqlCode	2705	\liZustandsMengenSammlung	2250
\liPetriSetzeSchluessel	2184	\listen@punkt ..	1787 , 1799	\liZustandsMengenSammlungNr	2259
\liPetriTransitionsName	2231 , 2243	\liStrich	1502	\liZustandsmengeOhneMathe	1187
\liPetriTransitionsNameOhneMathe	2231 , 2234 , 2236	\liSyntheseErklaerung	2744 , 2809	\liZustandsname	1210
\liPetriTransPfeile	2243	\liSyntheseUeberErklaerung	2728 , 2808	\liZustandsnameGross	1211 , 2251 , 2260
\liPolynomiellReduzierbar	1520	\liT	1878 , 1893 , 1907 , 1971	\liZustandsnameTiefgestellt	1226
\liPotenzmenge	1184 , 1188 , 2254	\liTeilen	2105	\liZustandsPaar	2060
\liPotenzmengeOhneMathe	1185 , 1186 , 1187	\liLiteratur	1669 , 1693	\liZustandsPaarVariablenName	2059 , 2062 , 2063
\liPrimaer	2503	\liTheta	1848 , 1899 , 1931 , 1937 , 1942	\llap	2492
\liProblemBeschreibung	1504	\liThetaOhneMathe	1848 , 1853 , 1855	\log	1896 , 1899 , 1902 , 1931 , 1937
\liProblemClique . . .	1527	\liTOhneMathe	1878 , 1887 , 1889	\loop	2325
\liProblemName	1503 , 1510 , 1522 , 1524 , 1537 , 1548 , 1549 , 1557 , 1558	\liTuringKante	208	\lor	3011
\liProblemSat	1556	\liTuringLeerzeichen	164 , 172	\ltimes	2007
\liProblemSubsetSum	1547 , 1556	\liTuringMaschine . . .	165		
		\liTuringUeberfuehrung	211		
		\liTuringUebergaenge	203 , 209		

M

\makeatletter ..	1803 , 2564
\makeatother ..	1805 , 2572
\marginpar	1144 , 1150 , 1156 , 1710

<code>\mathbb</code>	1558, 2312, 2931	2375, 2381, 2387,	1652, 1668, 1674,
<code>\mathbin</code>	2494, 2495, 2496	2393, 2544, 2589, 2903	1696, 1842, 2005,
<code>\mathcal</code>	1869, 2782,	<code>\noexpand</code>	2453,
	2787, 2789, 2790, 2791		2454, 2455, 2474, 2589
<code>\Mathe</code>	2986	<code>\noindent</code>	368,
<code>\MatheEnv</code>	2979, 2987, 2990		621, 626, 1714,
<code>\mathord</code>	2014, 2015		1716, 1720, 1724,
<code>\mdfsetup</code>	1305,		1752, 1780, 1782,
	1736, 1740, 1744, 1748		1795, 1811, 1813,
<code>\medskip</code>	1512,		1821, 1993, 2046,
	1721, 1723, 1797,		2049, 2052, 2055,
	1825, 2510, 2516,		2520, 2826, 2838, 2998
	2833, 2837, 2980, 2984	<code>\nolinkurl</code>	401, 2659
<code>\memph</code>	1710	<code>\normalsize</code>	1293
<code>\mintinline</code>	2653,	<code>\notin</code>	627
	2654, 2693, 2700, 2705	<code>\null</code>	2831
<code>\mkern</code>	2494, 2495, 2496	O	
<code>\mlq</code>	2012, 2014		
<code>\mrq</code>	2012, 2015	<code>\o@join</code>	
<code>\msg</code>	416		2491, 2494, 2495, 2496
<code>\myList</code>		<code>\Omega</code>	1859
	2585, 2586, 2587, 2590	<code>\omega</code>	2281, 2282, 2302, 2303
<code>\myNodes</code>	2574,	<code>\or</code>	2402
	2589, 2595, 2599, 2601	P	
N			
<code>\pagestyle</code>	1316		
<code>\NeedsTeXFormat</code>	1, 14,	<code>\par</code>	367, 388, 547,
	32, 57, 225, 422,		1245, 1253, 1719,
	483, 490, 552, 615,		1781, 1804, 1812,
	630, 1131, 1167,		2444, 2466, 2481,
	1280, 1320, 1329,		2522, 2808, 2832,
	1334, 1363, 1401,		2836, 2844, 2997, 3003
	1422, 1496, 1568,	<code>\paragraph</code>	1293
	1651, 1667, 1673,	<code>\parindent</code>	2513
	1695, 1841, 2004,	<code>\path</code>	99, 146, 209, 585
	2017, 2096, 2179,	<code>\pgfkeys</code>	2195, 2889,
	2245, 2270, 2275,		2890, 2891, 2892,
	2319, 2485, 2498,		2893, 2896, 2899, 2901
	2527, 2608, 2615,	<code>\pgfmath@count</code>	
	2622, 2627, 2707,		2566, 2568, 2570
	2722, 2813, 2817,	<code>\pgfmath@smuggleone</code>	2571
	2866, 2908, 2945, 2965	<code>\pgfmathdeclarefunction</code>	2565
<code>\neg</code>	3012	<code>\pgfmathhint</code>	2566
<code>\negthinspace</code>	1846	<code>\pgfmathparse</code>	
<code>\newcounter</code>	2339, 2340		1375, 2583,
<code>\NewDocumentCommand</code>			2588, 2591, 2605, 2606
	67, 108, 145, 166,	<code>\pgfmathresult</code>	
	208, 229, 557, 574,		1376, 2566,
	619, 624, 1234,		2567, 2569, 2571,
	1256, 1520, 1643,		2584, 2592, 2605, 2606
	1671, 1830, 1834,	<code>\pgfutil@empty</code>	2567
	2161, 2172, 2239,	<code>\pgfutil@loop</code>	2568
	2662, 2672, 2682,	<code>\pgfutil@repeat</code>	2571
	2694, 2697, 2701, 2886	<code>\preceq</code>	1523
<code>\NewDocumentEnvironment</code>		<code>\prime</code>	1502
	1213, 1306, 1361,	<code>\printbibliography</code>	1693
	1628, 1727, 1728,	<code>\ProvidesPackage</code>	2, 15,
	1731, 1758, 1766,		33, 58, 226, 423,
	1788, 1807, 1817,		484, 491, 553, 616,
	2067, 2117, 2507, 2525		631, 1132, 1168,
<code>\newlength</code>	2992		1281, 1321, 1330,
<code>\node</code>	570, 1636,		1335, 1364, 1402,
	1641, 2365, 2370,		1423, 1497, 1569,
Q		Q	
<code>\QS@list</code>			2446, 2457, 2461,
		</	

1332, 1336, 1337, 1338, 1366, 1367, 1415, 1433, 1438, 1439, 1501, 1570, 1670, 1675, 1676, 1692, 1699, 1700, 1701, 1729, 1844, 2008, 2009, 2181, 2183, 2273, 2333, 2335, 2487, 2488, 2489, 2502, 2530, 2561, 2612, 2618, 2625, 2630, 2641, 2642, 2710, 2815, 2822, 2870, 2871, 2873, 2874, 2912, 2913, 2914, 2947, 2967	\right 1846 \RIGHTarrow ... 2888, 2893 \Rightarrow 622, 627 \rightarrow 212, 511, 516, 524, 528, 530, 531, 533, 588, 596, 2243, 2753, 2760, 2762, 2765, 2770, 2775, 2776, 2781	\setminus 2146 \setsansfont 1285 \setul 2504 \sffamily 498, 1291, 1293, 1395, 1434, 2731, 2848 \shoveleft 2126 \shoveright 2130 \Sigma 69, 110, 168, 1197, 1198, 1258 \sigma 521, 523, 524 \SLASH 1711 \small 1820 \sort 2578 \sortList 2577, 2586 \square 488 \stepcounter 2365, 2370, 2375, 2378, 2380, 2384, 2386, 2390, 2392 \str .. 499, 508, 1733, 2255, 2264, 2732, 2745 \string 2128, 2138 \StrSubstitute . 2585, 2587 \strut 1953, 1957, 1961, 1965, 1969, 2613 \subsection 1447 \subseql 2756, 2789, 2796 \subsubsection 1468	\tiny 1145, 1151, 1157, 1640, 1710, 2657 \titleformat 1291, 1293, 1434 \titlespacing 1292 \tl 39, 68, 69, 70, 71, 72, 73, 76, 77, 78, 79, 80, 82, 84, 109, 110, 111, 112, 113, 114, 115, 118, 119, 120, 121, 122, 123, 124, 167, 168, 169, 170, 171, 172, 173, 176, 177, 178, 179, 180, 181, 182, 301, 305, 332, 336, 337, 338, 341, 346, 347, 348, 359, 360, 361, 362, 373, 379, 382, 385, 394, 408, 558, 561, 566, 567, 575, 576, 579, 580, 1236, 1257, 1258, 1259, 1260, 1263, 1264, 1265, 1266, 1442, 1446, 1450, 1457, 1460, 1879 \tmp 2148 \TmpPlaceEight 2203 \TmpPlaceFive 2200 \TmpPlaceFour 2199 \TmpPlaceNine 2204 \TmpPlaceOne 2196 \TmpPlaceSeven 2202 \TmpPlaceSix 2201 \TmpPlaceTen 2205 \TmpPlaceThree 2198 \TmpPlaceTwo 2197 \TmpScale 2216 \TmpTransitionEight 2192, 2213 \TmpTransitionFive 2189, 2210 \TmpTransitionFour 2188, 2209 \TmpTransitionNine 2193, 2214 \TmpTransitionOne 2185, 2206 \TmpTransitionSeven 2191, 2212 \TmpTransitionSix 2190, 2211 \TmpTransitionTen 2194, 2215 \TmpTransitionThree 2187, 2208 \TmpTransitionTwo 2186, 2207 \TmpX 2217 \TmpY 2218 \today 1657 \ttfamily 2505
S			
\sb 72, 82, 84, 113, 171, 523, 524, 528, 531, 532, 533, 1228, 1230, 1896, 1899, 1902, 1931, 1937, 2086, 2231, 2240, 2775, 2776, 2777, 2782, 2786, 2787, 2790, 2791, 2794, 2795, 2796			
\scriptscriptstyle 588, 596, 604			
\scriptsize 1247, 1581, 1588, 1594, 1656, 1657, 1660, 1661, 2917, 2970, 3000			
\section ... 52, 1434, 1443			
\sectionbreak 1435			
\seq . 1645, 1646, 1647, 1790, 1791, 1792, 1799			
\setbox 2491			
\setcounter 1294, 1436, 1470, 2445, 2467, 2481			
\setganttlinklabel .. 1324, 1325, 1326, 1327			
\setlength 1664, 2513, 2994, 2995, 3005			
\setmainfont 1284			
\setmainlanguage 420			
\setminted 2647, 2648			
T			
\tableofcontents 1491, 1706			
\text 82, 84, 187, 2108, 2917, 2970			
\textbf ... 1139, 1528, 1537, 1548, 1557, 1715, 1722, 1753, 1781, 1796, 1812, 2071			
\textcolor 1640, 2916			
\textit 979, 1021, 1022, 1023, 1024, 1822, 2116, 2176			
\textsc 1503			
\textsf 1715, 1796			
\textstyle 1916, 1948			
\texttt 1162, 1503, 1637, 1638, 1639, 1640, 2970			
\textwidth 1664			
\thepage 1315, 1659			
\theparagraph 1293			
\thesection 1434			
\Theta 1849			
\thinspace 2970			
\tikz 1641			
tikz: bbaum 25			
tikz: li binaer baum 23			
\tikzchildnode 443			
\tikzparentnode 443			
\tikzset 101, 148, 214, 428, 454, 1340, 1572, 2221, 2347, 2551, 2712, 2948			
\tikzumset 2876			
\times 212			

U		
<code>\ul</code>	1140, 2503, 2504	
<code>\umlaggreg</code>	1098	
<code>\umlassoc</code>	1056	
<code>\umlclass</code> . . .	645, 649, 653, 700, 701, 702, 749, 754, 759, 762, 820, 821, 822, 827, 828, 857, 892, 923, 924, 927, 978, 981, 1020, 1026, 1027, 1045, 1046, 1047, 1068, 1069, 1070, 1090, 1091, 1092, 1093	
<code>\umldep</code>	986	
<code>\umlHVHaggreg</code>	767, 833, 1036	
<code>\umlinherit</code>	706, 757, 931, 976, 984	
<code>\umlnote</code> . . .	708, 933, 1100	
<code>\umlreal</code>	704, 765	
<code>\umlsimpleclass</code>	660, 661, 662, 666, 668, 669, 670, 699, 852, 853, 854, 922, 974, 975, 1066	
<code>\umlstatic</code>	859, 893	
<code>\umluniaggreg</code>	929	
<code>\umluniassoc</code>	680, 705, 930, 1054, 1055, 1074, 1075	
<code>\umlVHuniassoc</code> . .	681, 682	
<code>\umlVHVdep</code>	674, 675, 677, 678, 861, 862	
<code>\umlVHVinherit</code> .	657, 658, 663, 664, 671, 672, 830, 831, 855, 856, 1034, 1035, 1072, 1073	
<code>\umlVHVreal</code>	824, 825, 1095, 1096	
<code>\UParrow</code>	2890	
<code>\url</code>	1831	
<code>\usemintedstyle</code>	2644	
<code>\usetikzlibrary</code> . .	61, 427, 1135, 1339, 1571, 2182, 2490, 2531, 2562, 2875, 2963	
V		
<code>\value</code>	2327	
<code>\varepsilon</code>	500, 511, 512, 1183, 1896, 1902, 1934, 1945	
<code>\vfill</code> . . .	1476, 1490, 2613	
<code>\vrule</code>	2827, 2831	
<code>\vspace</code>	1767, 1785, 2840, 2842	
X		
<code>\xappto</code> . .	2589, 2595, 2599	
<code>\xdef</code>	1369	
<code>\xintApply</code>	2415	
<code>\xintApplyUnbraced</code> . .	2414, 2420, 2421, 2422	
<code>\xintCSVtoList</code>	2474	
<code>\xintFor</code>	2364, 2369, 2374, 2379, 2385, 2391, 2432	
<code>\xintIfEq</code>	2425	
<code>\xintIfForLast</code>	2381, 2387, 2393	
<code>\xintIfGt</code>	2426	
<code>\xintIfLt</code>	2424	
<code>\xintLength</code>	2399	
<code>\xintNthelt</code>	2411	
Z		
<code>\ZB</code>	1838	
<code>\zB</code>	1837	
<code>\zustandsnamens@liste</code>	1201, 1208, 1209	