

lehramt-informatik

Hermine Bschlangaul *

September 12, 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
2.5.1	IFs	18
2.6	baum.sty	23
2.6.1	Binärbaum	24
2.6.2	AVL-Baum	25
2.6.3	B-Baum	26
2.7	checkbox.sty	27
2.8	chomsky-normalform.sty	28
2.8.1	Makro-Kürzel	28
2.8.2	TeX-Markup-Grundgerüst	28
2.8.3	Konkretes TeX-Markup-Beispiel	28
2.9	cpm.sty	31
2.9.1	Makro-Kürzel	31
2.9.2	TeX-Markup-Beispiel: Graph	31
2.9.3	TeX-Markup-Beispiel: Ergebnistabelle	31
2.9.4	TeX-Markup-Beispiel: Nebenrechnungstabelle „Frühester Zeitpunkt“	31
2.9.5	TeX-Markup-Beispiel: Nebenrechnungstabelle „Spätester Zeitpunkt“	32
2.9.6	Hilfsmakros für Zeitpunkt-Tabelle	33
2.10	cyk-algorithmus.sty	35
2.10.1	Makro-Kürzel	35
2.10.2	TeX-Markup-Beispiel	35
2.11	entwurfsmuster.sty	36
2.11.1	Namensschema der Entwurfsmuster-Makros:	36
2.11.2	Reihenfolge	36
2.11.3	Abstrakte Fabrik (Abstract Factory)	36
2.11.4	Adapter	38
2.11.5	Beobachter (Observer)	39
2.11.6	Dekorierer (Decorator)	41
2.11.7	Einfache Fabrik (Simple Factory)	42

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

2.11.8 Einzelstück (Singleton)	43
2.11.9 Erbauer (Builder)	44
2.11.10 Fabrikmethode (Factory Method)	45
2.11.11 Kompositum (Composite)	47
2.11.12 Modell-Präsentation-Steuerung (Model-View-Controller)	47
2.11.13 Stellvertreter (Proxy)	48
2.11.14 Zustand (State)	49
2.12 er.sty	51
2.12.1 Tex-Markup-Beispiel: Komplettes Diagramm	51
2.12.2 Tex-Markup-Beispiel: EER Enhanced Entity-Relation-Modell nach Elmasri/Navante	52
2.12.3 Makro-Kürzel	52
2.13 formale-sprachen.sty	54
2.14 formatierung.sty	57
2.14.1 Schriftarten / Typographie	57
2.14.2 Farben	57
2.14.3 Überschriften	57
2.14.4 Listen	57
2.14.5 Kasten	57
2.14.6 Header	57
2.14.7 Zeilenabstände	57
2.15 gantt.sty	59
2.16 grafik.sty	60
2.17 graph.sty	61
2.18 hanoi.sty	63
2.19 klassen-konfiguration-aufgabe.sty	64
2.20 klassen-konfiguration-examen.sty	66
2.21 komplexitaetstheorie.sty	69
2.21.1 Makro-Kürzel	69
2.22 kontrollflussgraph.sty	71
2.22.1 Makro-Kürzel	71
2.22.2 TeX-Markup-Beispiel	71
2.22.3 TikZ: pin	71
2.22.4 Umgebungen	72
2.22.5 Makros	73
2.23 kopf-fusszeilen.sty	74
2.24 literatur-dummy.sty	75
2.25 literatur.sty	76
2.26 makros.sty	77
2.26.1 Umgebungen, die Inhalte aus- und einblenden können	78
2.27 master-theorem.sty	82
2.27.1 Makro-Kürzel	82
2.28 mathe.sty	86
2.29 meta.sty	87
2.29.1 Einfache Makros (Low level)	87
2.29.2 Zusammengesetzte Makros (High level)	88
2.30 minimierung.sty	90
2.31 normalformen.sty	93
2.31.1 Makro-Kürzel	93
2.32 o-notation.sty	96
2.32.1 Makro-Kürzel	96
2.32.2 TeX-Markup-Beispiel: Funktionsgraphen mit pgfplots	96
2.33 petri.sty	97
2.33.1 Makro-Kürzel	97
2.34 potenzmengen-konstruktion.sty	99
2.35 pseudo.sty	101
2.36 pumping-lemma.sty	102
2.37 quicksort.sty	103

2.38	relationale-algebra.sty	106
2.39	rmodell.sty	107
2.39.1	Makro-Kürzel	107
2.40	sortieren.sty	108
2.41	spalten.sty	110
2.42	sql.sty	111
2.42.1	Latex-Markup-Beispiel	111
2.43	struktogramm.sty	112
2.44	syntax.sty	113
2.44.1	Makro-Kürzel	113
2.45	syntaxbaum.sty	116
2.46	synthese-algorithmus.sty	117
2.46.1	Makro-Kürzel	117
2.46.2	TeX-Markup Grundgerüst	117
2.46.3	TeX-Markup Linksreduktion	117
2.46.4	TeX-Markup Rechtsreduktion	117
2.46.5	TeX-Markup Relationen formen	118
2.47	tabelle.sty	121
2.48	typographie.sty	122
2.49	uml.sty	124
2.50	vollstaendige-induktion.sty	126
2.50.1	Makro-Kürzel	126
2.51	wasserfall.sty	128
2.52	wpkalkuel.sty	129
2.52.1	Makro-Kürzel	129

3 Index 130

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}

\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 }

```

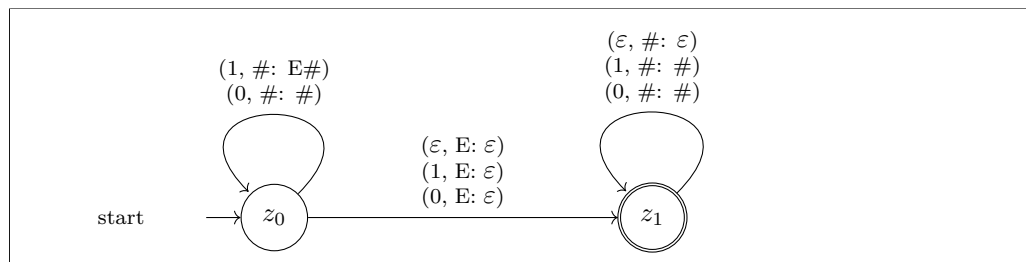
2.4.2 Kellerautomat

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

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

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

```
107 \ExplSyntaxOn
108 \NewDocumentCommand {\liKellerAutomat} { O{K} m } {
109   \tl_set:Nn \l_zustaeende_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     zustaeende .code:n = {\tl_set:Nn \l_zustaeende_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]{\langlezustand-oder-lese\rangle\langleschreibe\rangle\langlerichtung\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
```

2.5.1 IFs

Wir weichen von dem üblichen Namensschema ab und beginnen das `if` mit einem Großbuchstaben und schreiben das letzte Worte komplett in Großbuchstaben, damit die `if`-Befehle schöner lesbar sind, z. B. `\ifLiADDITUM`, `\LiADDITUMtrue` und `\LiADDITUMfalse`.

```
\ifLiADDITUM
\LiADDITUMtrue 229 \newif\ifLiADDITUM
\LiADDITUMfalse 230 \LiADDITUMfalse

\ifLiEXKURS
\LiEXKURStrue 231 \newif\ifLiEXKURS
\LiEXKURfalse 232 \LiEXKURStrue

\ifLiANTWORT
\LiANTWORTtrue 233 \newif\ifLiANTWORT
\LiANTWORTfalse 234 \LiANTWORTtrue

\liLadePakete

235 \NewDocumentCommand{\liLadePakete}{ m }
236 {
237   \clist_map_inline:nn { #1 } { \RequirePackage{lehramt-informatik-##1} }
238 }

\liLadeAllePakete

239 \def\liLadeAllePakete{
240   \liLadePakete{
241     aufgaben-einbinden,
242     automaten,
243     checkbox,
244     chomsky-normalform,
245     cpm,
246     cyk-algorithmus,
247     entwurfsmuster,
248     er,
249     formale-sprachen,
250     gantt,
251     grafik,
252     graph,
253     hanoi,
254     kontrollflussgraph,
255     komplexitaetstheorie,
256     makros,
257     master-theorem,
258     mathe,
259     minimierung,
260     normalformen,
261     petri,
262     potenzmengen-konstruktion,
263     pumping-lemma,
264     pseudo,
265     quicksort,
266     relationale-algebra,
267     rmodell,
268     sortieren,
269     spalten,
```

```

270     struktogramm,
271     sql,
272     syntax,
273     syntaxbaum,
274     synthese-algorithmus,
275     tabelle,
276     typographie,
277     uml,
278     vollstaendige-induktion,
279     wasserfall,
280     wpkalkuel,
281     %
282     baum, % am Schluss sonst Fehler: undefined command \edge
283 }
284 }

```

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.

```

285 \clist_new:N \g_auf_schluessel_clist
286 \clist_set:Nn \g_auf_schluessel_clist {
287     titel,
288     thematik,
289     stichwoerter,
290     zitat_schluessel,
291     zitat_beschreibung,
292     %
293     bearbeitungs_stand,
294     korrektheit,
295     %
296     relativer_pfad,
297     identische_aufgabe,
298     %
299     examen_nummer,
300     examen_fach,
301     examen_jahr,
302     examen_monat,
303     examen_jahreszeit,
304     examen_thema_nr,
305     examen_teilaufgabe_nr,
306     examen_aufgabe_nr,
307 }

```

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

```

308 \clist_map_inline:Nn \g_auf_schluessel_clist {
309     \tl_new:c {g_auf_#1_tl}
310 }

```

Funktion über alle globalen Token-List-Variablen zurückzusetzen.

```

311 \cs_new:Npn \setze_variablen_zurueck: {
312     \clist_map_inline:Nn \g_auf_schluessel_clist {
313         \tl_clear:c {g_auf_##1_tl}
314     }
315 }

```

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

```

316 \keys_define:nn { aufgabenmetadaten }
317 {
318     Titel .tl_gset:N = \g_auf_titel_tl,
319     Thematik .tl_gset:N = \g_auf_thematik_tl,
320     Stichwoerter .tl_gset:N = \g_auf_stichwoerter_tl,

```

```

321 ZitatSchluessel .tl_gset:N = \g_auf_zitat_beschreibung_tl,
322 ZitatBeschreibung .tl_gset:N = \g_auf_zitat_schluessel_tl,
323 %
324 BearbeitungsStand .tl_gset:N = \g_auf_bearbeitungs_stand_tl,
325 Korrektheit .tl_gset:N = \g_auf_korrekttheit_tl,
326 %
327 RelativerPfad .tl_gset:N = \g_auf_relativer_pfad_tl,
328 IdentischeAufgabe .tl_gset:N = \g_auf_identische_aufgabe_tl,
329 %
330 ExamenNummer .tl_gset:N = \g_auf_examen_nummer_tl,
331 ExamenFach .tl_gset:N = \g_auf_examen_fach_tl,
332 ExamenJahr .tl_gset:N = \g_auf_examen_jahr_tl,
333 ExamenMonat .tl_gset:N = \g_auf_examen_monat_tl,
334 ExamenJahreszeit .tl_gset:N = \g_auf_examen_jahreszeit_tl,
335 ExamenThemaNr .tl_gset:N = \g_auf_examen_thema_nr_tl,
336 ExamenTeilaufgabeNr .tl_gset:N = \g_auf_examen_teilaufgabe_nr_tl,
337 ExamenAufgabeNr .tl_gset:N = \g_auf_examen_aufgabe_nr_tl,
338 }

339 \cs_gset:Npn \setze_relativen_pfad: {
340   \tl_if_empty:NTF \g_auf_relativer_pfad_tl
341   {
342     \bool_if:nTF
343     {
344       ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
345       ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
346       ! \tl_if_empty_p:N \g_auf_examen_monat_tl
347     }
348     {
349       \tl_gset:Nn \g_auf_relativer_pfad_tl {
350         Staatsexamen /
351         \g_auf_examen_nummer_tl /
352         \g_auf_examen_jahr_tl /
353         \g_auf_examen_monat_tl /
354         \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} { Thema - \g_auf_examen_thema_nr_tl / }
355         \tl_if_empty:NTF \g_auf_examen_teilaufgabe_nr_tl {} { Teilaufgabe - \g_auf_examen_teilaufgabe_nr_tl / }
356         \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} { Aufgabe - \g_auf_examen_aufgabe_nr_tl / }
357       }
358     }
359   {}
360 }

361 {}
362 }

363 \cs_set:Nn \_trenner: {
364   \, / \,
365 }

366 \cs_gset:Npn \_gib_jahreszeit_durch_monat: #1 {
367   % \str_case funktioniert nicht mit den Tokenlist variablen.
368   \tl_case:Nn { #1 }
369   {
370     { 3 } { Frühjahr }
371     { 03 } { Frühjahr }
372     { 9 } { Herbst }
373     { 09 } { Herbst }
374   }
375 }

Definiert auch in .scripts/nodejs/src/examen.ts funktioniert nicht

376 \cs_gset:Npn \_gib_examen_fach_durch_nummer: #1 {
377   \tl_case:Nn { #1 }
378   {
379     { 46110 } { Grundlagen der Informatik (nicht vertieft) }
380     { 46111 } { Programmentwicklung / Systemprogrammierung / Datenbanksysteme (nicht vertieft) }
381     { 46112 } { Grundlagen der Informatik (nicht vertieft) }

```

```

382 { 46113 } { Theoretische Informatik (nicht vertieft) }
383 { 46114 } { Algorithmen / Datenstrukturen / Programmiermethoden (nicht vertieft) }
384 { 46115 } { Theoretische Informatik / Algorithmen / Datenstrukturen (nicht vertieft) }
385 { 46116 } { Softwaretechnologie / Datenbanksysteme (nicht vertieft) }
386 { 46118 } { Fachdidaktik (Mittelschulen) }
387 { 46119 } { Fachdidaktik (Realschulen) }
388 { 46121 } { Fachdidaktik (berufliche Schulen) }
389 { 66110 } { Automatentheorie, Algorithmische Sprache (vertieft) }
390 { 66111 } { Betriebssysteme / Datenbanksysteme / Rechnerarchitektur (vertieft) }
391 { 66112 } { Automatentheorie / Komplexität / Algorithmen (vertieft) }
392 { 66113 } { Rechnerarchitektur / Datenbanken / Betriebssysteme (vertieft) }
393 { 66114 } { Datenbank- und Betriebssysteme (vertieft) }
394 { 66115 } { Theoretische Informatik / Algorithmen (vertieft) }
395 { 66116 } { Datenbanksysteme / Softwaretechnologie (vertieft) }
396 { 66118 } { Fachdidaktik (Gymnasium) }
397 }
398 }

```

Einzelprüfungsnummer / Jahr / Jahreszeit mit Trennzeichen

```

399 \cs_gset:Npn \_gib_einzelpruefung_trenner: {
400   Staatsexamen ~
401   \g_auf_examen_nummer_tl
402
403   \_trenner:
404
405   \g_auf_examen_jahr_tl
406
407   \_trenner:
408
409   \gib_jahreszeit_durch_monat: \g_auf_examen_monat_tl
410 }

```

Thema Nr.1 / Teilaufgabe Nr. 2 / Aufgabe 3

```

411 \cs_gset:Npn \_gib_aufgaben_pfad_trenner: {
412   \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} {
413     Thema ~ \g_auf_examen_thema_nr_tl \_trenner:
414   }
415   \tl_if_empty:NTF \g_auf_examen_telaufgabe_nr_tl {} {
416     Teilaufgabe ~ \g_auf_examen_telaufgabe_nr_tl \_trenner:
417   }
418   \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} {
419     Aufgabe ~ \g_auf_examen_aufgabe_nr_tl
420   }
421 }

422 \cs_gset:Npn \_gib_examen_titel: {
423   \cs_set:Nn \_trenner: { \, / \, }
424   \bool_if:nTF
425   {
426     ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
427     ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
428     ! \tl_if_empty_p:N \g_auf_examen_monat_tl &&
429     ! \tl_if_empty_p:N \g_auf_examen_aufgabe_nr_tl
430   }
431   {
432     {
433       \footnotesize
434       \par
435       \noindent
436       Staatsexamen ~
437       \g_auf_examen_nummer_tl \_trenner:
438       \g_auf_examen_jahr_tl \_trenner:
439
440       \tl_case:Nn \g_auf_examen_monat_tl
441       {

```

```

442     { 03 } { Frühjahr }
443     { 09 } { Herbst }
444 } \trenner:
445
446 \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} {
447     Thema ~ Nr. ~ \g_auf_examen_thema_nr_tl \trenner:
448 }
449 \tl_if_empty:NTF \g_auf_examen_teilaufgabe_nr_tl {} {
450     Teilaufgabe ~ Nr. ~ \g_auf_examen_teilaufgabe_nr_tl \trenner:
451 }
452 \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} {
453     Aufgabe ~ Nr. ~ \g_auf_examen_aufgabe_nr_tl
454 }
455 \par
456 \bigskip
457 }
458 }
459 }
460 \cs_new:Npn \_gib_github_url: {
461     \LehramtInformatikGithubDomain /
462     \LehramtInformatikGithubTexRepo /
463     blob /
464     \LehramtInformatikGitBranch /
465     \g_auf_relativer_pfad_tl
466 }
467 \cs_new:Npn \_gib_github_url_href: {
468     \tl_if_empty:NTF \g_auf_relativer_pfad_tl {} {
469         \url{ \_gib_github_url: }
470     }
471 }
472 \cs_new:Npn \_gib_aufgaben_titel: {
473     \g_auf_titel_tl
474
475     \tl_if_empty:NTF \g_auf_thematik_tl
476     {}
477     {
478         \, ~ [
479             \g_auf_thematik_tl
480         ]
481     }
482 }
483 \msg_new:nnn { aufgabenmetadaten } { kein-titel }
484 { Der~Schlüssel~ist~zwingend~notwendig }
485
486 \def\LehramtInformatikRepository{/pfad/zum/repository}
487 \input /etc/lehramt-informatik.config.tex
488
489 biblatex not working with lualatex and babel
490 % \RequirePackage{polyglossia}
491 % \setmainlanguage{german}
492
493

```

2.6 baum.sty

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

492 \RequirePackage{tikz}

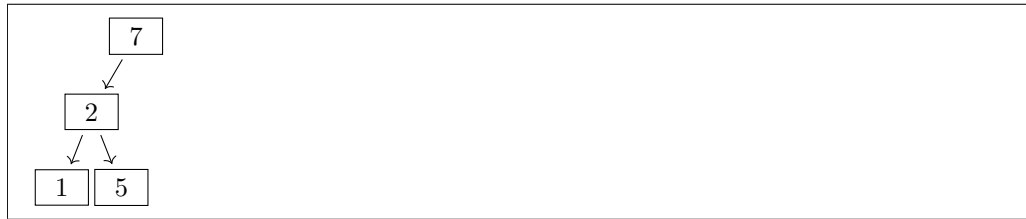
    für li binaer baum
493 \RequirePackage{tikz-qtree}

    Für b baum
494 \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}
```



```
495 \tikzset{
496   li binaer baum/.style={
497     shorten <=2pt,
498     shorten >=2pt,
499     ->,
500     every tree node/.style={
501       minimum width=2em,
502       draw,
503       rectangle
504     },
505     blank/.style={
506       draw=none
507     },
508     edge from parent/.style={
509       draw,
510       edge from parent path={(\tikzparentnode) -- (\tikzchildnode)}
511     },
512     level distance=1cm,
513     every label/.style={
514       gray,
515       font=\footnotesize,
516       label position=0,
517       label distance=0cm,
518     }
519   },
520 }
```


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}
```

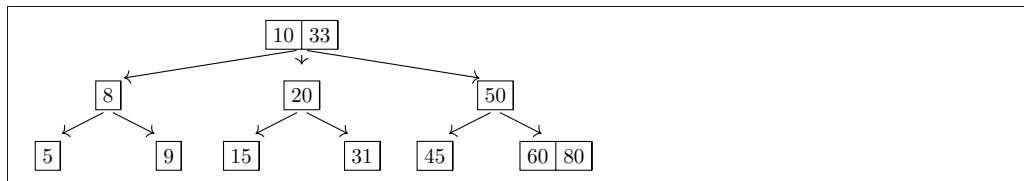


2.6.3 B-Baum

```

\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}

```



```

521 \tikzset{
522   li bbaum knoten/.style={
523     rectangle split parts=10,
524     rectangle split,
525     rectangle split horizontal,
526     rectangle split ignore empty parts,
527     draw,
528     fill=white
529   },
530   li bbaum/.style={
531     every node/.style={
532       li bbaum knoten
533     },
534     level 1/.style={
535       level distance=12mm,
536       sibling distance=25mm,
537     },
538     every child/.style={
539       shorten <= 2pt,
540       shorten >= 6pt,
541       ->,
542     },
543     level 2/.style={
544       level distance=9mm,
545       sibling distance=15mm,
546     },
547   }
548 }
549

```

2.7 checkbox.sty

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

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

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

556
```

2.8 chomsky-normalform.sty

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

560 \ExplSyntaxOn

561 \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

```

562 \def\liChomskyUeberschrift#1{
563   {
564     \bfseries
565     \rmfamily
566     \str_case:nn {#1} {
567       {1} {Elimination-der-\varepsilon-Regeln}
568       {2} {Elimination-von-Kettenregeln}
569       {3} {Separation-von-Terminalzeichen}
570       {4} {Elimination-von-mehrelementigen-Nonterminalketten}
571     }
572   }
573 }

```

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

```

574 \def\liChomsky@erklaerung@texte#1{
575   \str_case:nn {#1} {
576     %
577     {1} {
578       Alle-Regeln-der-Form- $A \rightarrow \varepsilon$ -werden-eliminiert.~
579       Die-Ersetzung-von- $A$ -wird-durch- $\varepsilon$ -in-alen-anderen-
580       Regeln-vorweggenommen.
581     }
582     {2} {
583       Jede-Produktion-der-Form- $A \rightarrow B$ -mit- $A, B$ -in- $S$ -wird-
584       als-Kettenregel-bezeichnet.~Diese-tragen-nicht-zur-Produktion-
585       von-Terminalzeichen-bei-und-lassen-sich-ebenfalls-eliminieren.
586     }
587     {3} {
588       Jedes-Terminalzeichen- $\sigma$ ,~das-in-Kombination-mit-anderen-
589       Symbolen-auftaucht,~wird-durch-ein-neues-Nonterminal-
590        $S_{\sigma}$ -ersetzt-und-die-Menge-der-Produktionen-durch-die-
591       Regel- $S_{\sigma} \rightarrow \sigma$ -ergänzt.
592     }
593     {4} {
594       Alle-Produktionen-der-Form-
595        $A \rightarrow B_{\{1\}} B_{\{2\}} \dots B_{\{n\}}$ -
596       werden-in-die-Produktionen-
597        $A \rightarrow A_{\{n-1\}} B_{\{n\}}, A_{\{n-1\}} \rightarrow A_{\{n-2\}} B_{\{n-1\}}, \dots,$ 
598        $A_{\{2\}} \rightarrow B_{\{1\}} B_{\{2\}}$ -zerteilt.~
599       Nach-der-Ersetzung-sind-alles-längeren-Nonterminalketten-
600       vollständig-heruntergebrochen-und-die-Chomsky-Normalform-erreicht.
601     }
602   }
603 }
604 }

```

```

605 }
606 \def\liChomskyErklaerung#1{
607   {
608     \itshape
609     \footnotesize
610     \liParagraphMitLinien{\li@chomsky@erklaerung@texte{#1}}
611   }
612 }

```

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

```

613 \def\liChomskyUeberErklaerung#1{
614   \liChomskyUeberschrift{#1}\par
615   \liChomskyErklaerung{#1}
616 }

```

```

617 \ExplSyntaxOff
618

```

2.9 cpm.sty

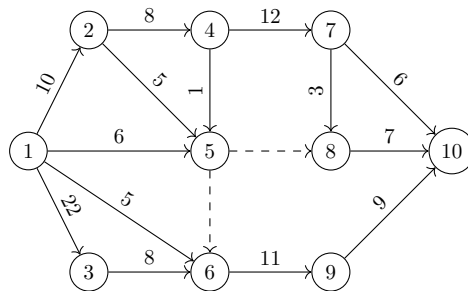
```
619 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
620 \ProvidesPackage{lehramt-informatik-cpm}[2020/09/03]
621 \RequirePackage{tikz}

622 \liLadePakete{mathe,typographie}
```

2.9.1 Makro-Kürzel

```
\let\f=\footnotesize
\let\FZ=\liCpmFruehI
\let\SZ=\liCpmSpaetI
\let\v=\liCpmVon
\let\ vz=\liCpmVonZu
\let\z=\liCpmZu
```

2.9.2 TeX-Markup-Beispiel: Graph



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

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

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

2.9.3 TeX-Markup-Beispiel: Ergebnistabelle

```
\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}
```

2.9.4 TeX-Markup-Beispiel: Nebenrechnungstabelle „Frühester Zeitpunkt“

```
\liCpmFruehErklaerung
\begin{tabular}{|l|l|r|}
\hline
\hline
i & Nebenrechnung & \FZ \\ \hline
1 & & 0 \\
2 & & 5 \\
3 & & 18
\end{tabular}
```

```

4      &                                     & 7    \\
5      &                                     & 19   \\
6      &                                     & 26   \\
7      & $\max(19_3, 22_4)$                 & 22   \\
8      & $\max(30_5, 30_6, 28_7)$ & 30   \\ \\hline
\end{tabular}

```

2.9.5 TeX-Markup-Beispiel: Nebenrechnungstabelle „Spätester Zeitpunkt“

```

\liCpmSpaetErklaerung
% Absteigend nach i sortieren
\begin{tabular}{|l|l|r|}
\hline
$i$ & Nebenrechnung & \SZ \\ \\hline
8      & siehe \FZ[8]    & 30   \\
7      &                 & 24   \\
6      &                 & 26   \\
5      &                 & 19   \\
4      &                 & 9    \\
3      & $\min(18_6, 23_7)$ & 18   \\
2      &                 & 5    \\
1      & $\min(0_2, 0_3, 2_4)$ & 0    \\ \\hline
\end{tabular}

```

```

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

```

```

623 \ExplSyntaxOn
624 \NewDocumentCommand { \liCpmEreignis } { 0{} m m m } {
625   \tl_set:Nn \l_name_tl {}
626
627   \keys_define:nn { cpmEreignis } {
628     name .code:n = {\tl_set:Nn \l_name_tl {##1}},
629   }
630
631   \keys_set:nn { cpmEreignis } { #1 }
632
633   \tl_if_empty:NT \l_name_tl {
634     \tl_set:Nn \l_name_tl {#2}
635   }
636
637   \node[circle,draw] (\l_name_tl) at (#3,#4) {#2};
638 }
639 \ExplSyntaxOff

```

```

\liCpmVorgang \liCpmVORGANG\{((.*)>(.*))\}\{(.*)\}

```

```

640 \ExplSyntaxOn
641 \NewDocumentCommand { \liCpmVorgang } { 0{} m m m } {
642   \tl_set:Nn \l_schein_tl {}
643   \tl_set:Nn \l_kritisch_tl {}
644
645   \keys_define:nn { cpmVorgang } {
646     schein .code:n = {\tl_set:Nn \l_schein_tl {dashed}},
647     kritisch .code:n = {\tl_set:Nn \l_kritisch_tl {very~thick}},
648   }
649
650   \keys_set:nn { cpmVorgang } { #1 }
651
652   \path[->,\l_schein_tl,\l_kritisch_tl] (#2) edge node[auto,sloped] {#4} (#3);
653 }
654 \ExplSyntaxOff

```


2.9.6 Hilfsmakros für Zeitpunkt-Tabelle

```
\begin{tabular}{|1||1|1|1|1|1|1|1|1|1|}
\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 & 0 & 2 & 0 & 0 & 2 & 0 & \\ \hline
\end{tabular}
```

```
\liCpmVonZu Let-Abkürzung: \let\ vz=\liCpmVonZu
\liCpmVonZu{1}(2-3):  $1_{(2 \rightarrow 3)}$ 
655 \def\liCpmVonZuOhneMathe#1(#2-#3){#1_{\scriptscriptstyle(#2\rightarrow#3)}}
656 \def\liCpmVonZu#1(#2-#3){%
657 \ifmmode%
658 \liCpmVonZuOhneMathe{#1}(\#2-\#3)%
659 \else%
660 $\liCpmVonZuOhneMathe{#1}(\#2-\#3)$%
661 \fi%
662 }
```

```
\liCpmVon Let-Abkürzung: \let\ v=\liCpmVon
\liCpmVon{1}(2):  $1_{(\rightarrow 2)}$ 
663 \def\liCpmVonOhneMathe#1(#2){#1_{\scriptscriptstyle(\rightarrow#2)}}
664 \def\liCpmVon#1(#2){%
665 \ifmmode%
666 \liCpmVonOhneMathe{#1}(\#2)%
667 \else%
668 $\liCpmVonOhneMathe{#1}(\#2)$%
669 \fi%
670 }
```

```
\liCpmZu Let-Abkürzung: \let\ z=\liCpmZu
\liCpmZu{1}(2):  $1_{(\leftarrow 2)}$ 
671 \def\liCpmZuOhneMathe#1(#2){#1_{\scriptscriptstyle(\leftarrow#2)}}
672 \def\liCpmZu#1(#2){%
673 \ifmmode%
674 \liCpmZuOhneMathe{#1}(\#2)%
675 \else%
676 $\liCpmZuOhneMathe{#1}(\#2)$%
677 \fi%
678 }
```

```
679 \ExplSyntaxOn
```

\liCpmSpaetI Spätester Zeitpunkt, zu dem Ereignis i eintreten kann

Let-Abkürzung: \let\ SZ=\liCpmSpaetI

```
680 \NewDocumentCommand{ \liCpmSpaetI } { 0{i} } {
681 \ifmmode
682 SZ\sb{#1}
683 \else
684 $SZ\sb{#1}$
685 \fi
686 }
```

\liCpmFruehI Frühester Zeitpunkt, zu dem Ereignis i eintreten kann.

Let-Abkürzung: \let\ FZ=\liCpmFruehI

```
687 \NewDocumentCommand{ \liCpmFruehI } { 0{i} } {
688 \ifmmode
689 FZ\sb{#1}
690 \else
691 $FZ\sb{#1}$
692 }
```

```

692 \fi
693 }

```

\liCpmFruehErklaerung

— Wir führen eine Vorwärtsterminierung durch und addieren die Dauern. Kann ein Ereignis über mehrere Vorgänge erreicht werden, wählen wir das Maximum aus. **Erläuterungen:** i : Ereignis i ; FZ_i : Frühester Zeitpunkt, zu dem Ereignis i eintreten kann. —

```

694 \def\liCpmFruehErklaerung{
695   \liParagraphMitLinien{
696     Wir~führen~eine~Vorwärtsterminierung~durch~
697     und~addieren~die~Dauern.~
698
699     Kann~ein~Ereignis~über~mehrere~Vorgänge~erreicht~
700     werden,~wählen~wir~das~Maximum~aus.~
701
702     \textbf{Erläuterungen:}~
703
704      $i$ :~
705     Ereignis~ $i$ ;~,
706
707     \liCpmFruehI{}:~
708     Frühester~Zeitpunkt,~zu~dem~Ereignis~ $i$ ~eintreten~kann
709     .
710   }
711 }

```

\liCpmSpaetErklaerung

— Wir führen eine Rückwärtsterminierung durch und subtrahieren die Dauern vom letzten Ereignis aus. Kann ein Ereignis über mehrere Vorgänge erreicht werden, wählen wir das Minimum aus. **Erläuterungen:** i : Ereignis i ; SZ_i : Spätester Zeitpunkt, zu dem Ereignis i eintreten kann. —

```

712 \def\liCpmSpaetErklaerung{
713   \liParagraphMitLinien{
714     Wir~führen~eine~Rückwärtsterminierung~durch~
715     und~subtrahieren~die~Dauern~vom~letzten~Ereignis~aus.~
716
717     Kann~ein~Ereignis~über~mehrere~Vorgänge~erreicht~
718     werden,~wählen~wir~das~Minimum~aus.~
719
720     \textbf{Erläuterungen:}~
721
722      $i$ :~
723     Ereignis~ $i$ ;~,
724
725     \liCpmSpaetI{}:~
726     Spätester~Zeitpunkt,~zu~dem~Ereignis~ $i$ ~eintreten~kann
727     .
728   }
729 }

730 \ExplSyntaxOff
731

```

2.10 cyk-algorithmus.sty

```
732 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
733 \ProvidesPackage{lehramt-informatik-cyk-algorithmus}[2021/06/18 Hilfsmakros
734 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`

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

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

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

```
736 \NewDocumentCommand{ \liWortInSprache } { m O{L(G)} } {
737   \bigskip
738   \noindent
739   $\Rightarrow$ #1 \in #2$
740 }
```

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

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

```
741 \NewDocumentCommand{ \liWortNichtInSprache } { m O{L(G)} } {
742   \bigskip
743   \noindent
744   $\Rightarrow$ #1 \notin #2$
745 }
```

```
746
```

2.11 entwurfsmuster.sty

```
747 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
748 \ProvidesPackage{lehramt-informatik-entwurfsmuster}[2021/05/06
749 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

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

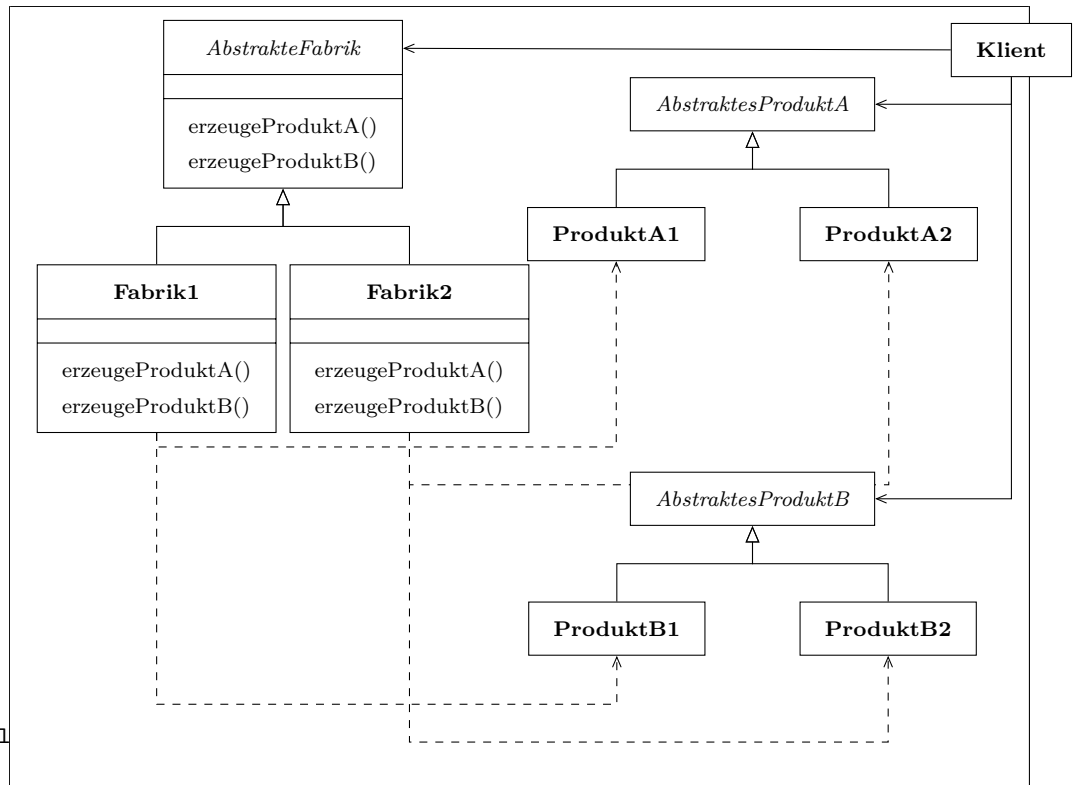
\li@EntwurfsCodeAllgemein Allgemeine Code-Beispiele zu den UML-Diagrammen und Stellvertretern

```
751 \RequirePackage{lehramt-informatik-syntax}
752 \def\li@EntwurfsCode#1#2{
753   \liJavaDatei{entwurfsmuster/#1/allgemein/#2}
754 }
```

2.11.3 Abstrakte Fabrik (Abstract Factory)

sAbstrakteFabrikBeschreibung

```
755 \def\liEntwurfsAbstrakteFabrikBeschreibung{
756   Es wird eine Schnittstelle bereitgestellt, um \emph{Familien
757   verbundener oder abhängiger Objekte} zu erstellen, ohne die konkreten
758   Klassen zu spezifizieren.\footcite[Seite 25]{eilebrecht}
759 }
```



```

760 \def\liEntwurfsAbstrakteFabrikUml{
761   \begin{tikzpicture}
762     \umlclass[type=abstract]{AbstrakteFabrik}{-}{-}{
763       erzeugeProduktA()\
764       erzeugeProduktB()\
765     }
766     \umlclass[below left=1cm and -1.5cm of AbstrakteFabrik]{Fabrik1}{-}{-}{
767       erzeugeProduktA()\
768       erzeugeProduktB()\
769     }
770     \umlclass[below right=1cm and -1.5cm of AbstrakteFabrik]{Fabrik2}{-}{-}{
771       erzeugeProduktA()\
772       erzeugeProduktB()\
773     }
774     \umlVHVinherit{Fabrik1}{AbstrakteFabrik}
775     \umlVHVinherit{Fabrik2}{AbstrakteFabrik}
776
777     \umlsimpleclass[right=3cm of AbstrakteFabrik,type=abstract]{AbstraktesProduktA}
778     \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktA]{ProduktA1}
779     \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktA]{ProduktA2}
780     \umlVHVinherit{ProduktA1}{AbstraktesProduktA}
781     \umlVHVinherit{ProduktA2}{AbstraktesProduktA}
782
783     \umlsimpleclass[above right=0cm and 1cm of AbstraktesProduktA]{Klient}
784
785     \umlsimpleclass[below=4.5cm of AbstraktesProduktA,type=abstract]{AbstraktesProduktB}
786     \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktB]{ProduktB1}
787     \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktB]{ProduktB2}
788     \umlVHVinherit{ProduktB1}{AbstraktesProduktB}
789     \umlVHVinherit{ProduktB2}{AbstraktesProduktB}
790
791     \umlVHVdep[arm1=-1.3cm,arm2=-1cm]{Fabrik1}{ProduktA1}
792     \umlVHVdep[arm2=-1cm]{Fabrik1}{ProduktB1}
793
794     \umlVHVdep[arm1=-1.8cm,arm2=-1.5cm]{Fabrik2}{ProduktA2}
795     \umlVHVdep[arm2=-1.5cm]{Fabrik2}{ProduktB2}
796
  
```

```

797 \umluniassoc[anchor2=25]{Klient}{AbstrakteFabrik}
798 \umlVHuniassoc[arm1=-1cm]{Klient}{AbstraktesProduktA}
799 \umlVHuniassoc{Klient}{AbstraktesProduktB}
800 \end{tikzpicture}
801 }

```

iEntwurfsAbstrakteFabrikCode

```

802 \def\liEntwurfsAbstrakteFabrikCode{
803 \li@EntwurfsCode{abstrakte_fabrik}{Produkte}
804 \li@EntwurfsCode{abstrakte_fabrik}{AbstrakteFabrik}
805 \li@EntwurfsCode{abstrakte_fabrik}{Klient}
806 }

```

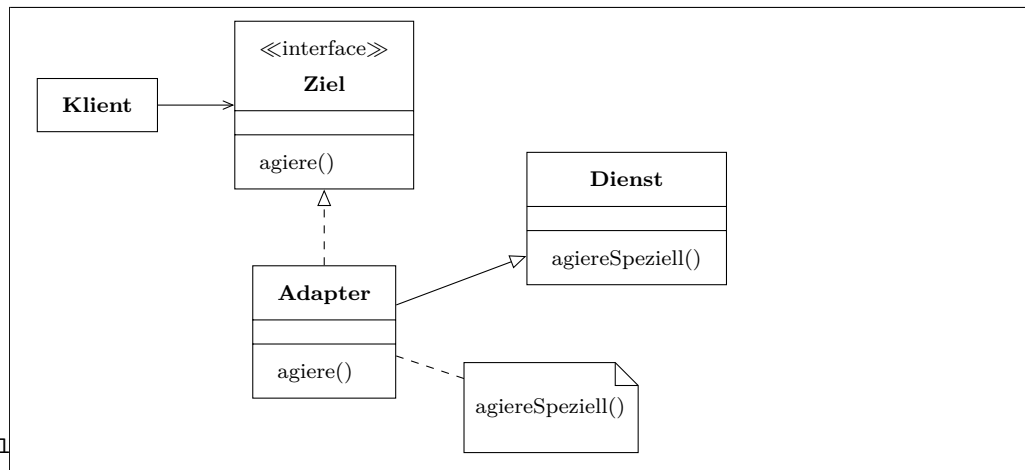
\liEntwurfsAbstrakteFabrik

```

807 \def\liEntwurfsAbstrakteFabrik{
808 \liEntwurfsAbstrakteFabrikBeschreibung
809
810 \liEntwurfsAbstrakteFabrikUml
811
812 \liEntwurfsAbstrakteFabrikCode
813 }

```

2.11.4 Adapter



\liEntwurfsAdapterUml

```

814 \def\liEntwurfsAdapterUml{
815 \begin{tikzpicture}
816 \umlsimpleclass[x=1,y=3]{Klient}{}{}
817 \umlclass[x=4,y=3,type=interface]{Ziel}{}{agiere()}
818 \umlclass[x=4,y=0]{Adapter}{}{agiere()}
819 \umlclass[x=8,y=1.5]{Dienst}{}{agiereSpeziell()}
820
821 \umlreal{Adapter}{Ziel}
822 \umluniassoc{Klient}{Ziel}
823 \umlinherit{Adapter}{Dienst}
824
825 \umlnote[x=7,y=-1,width=2cm]{Adapter}{agiereSpeziell()}
826 \end{tikzpicture}
827 \footcite[so ähnlich wie GoF]{wiki:adapter}
828 }

```

\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.

```

829 \def\liEntwurfsAdapterAkteure{
830   \begin{description}
831
832     \item[Ziel (Target)]
833
834     Das Ziel definiert die Schnittstelle, die der Klient nutzen kann.
835
836     \item[Klient (Client)]
837
838     Der Klient nutzt Dienste über inkompatible Schnittstellen und greift
839     dabei auf adaptierte Schnittstellen zurück.
840
841     \item[Dienst (Adaptee)]
842
843     Der Dienst bietet wiederzuverwendende Dienstleistungen mit fest
844     definierter Schnittstelle an.
845
846     \item[Adapter]
847
848     Der Adapter adaptiert die Schnittstelle des Dienstes auf die
849     Schnittstelle zum Klienten.\footcite{wiki:adapter}
850
851   \end{description}
852 }

```

\liEntwurfsAdapterCode

```

853 \def\liEntwurfsAdapterCode{
854   \li@EntwurfsCode{adapter}{Dienst}
855   \li@EntwurfsCode{adapter}{Ziel}
856   \li@EntwurfsCode{adapter}{Adapter}
857   \li@EntwurfsCode{adapter}{Klient}
858 }

```

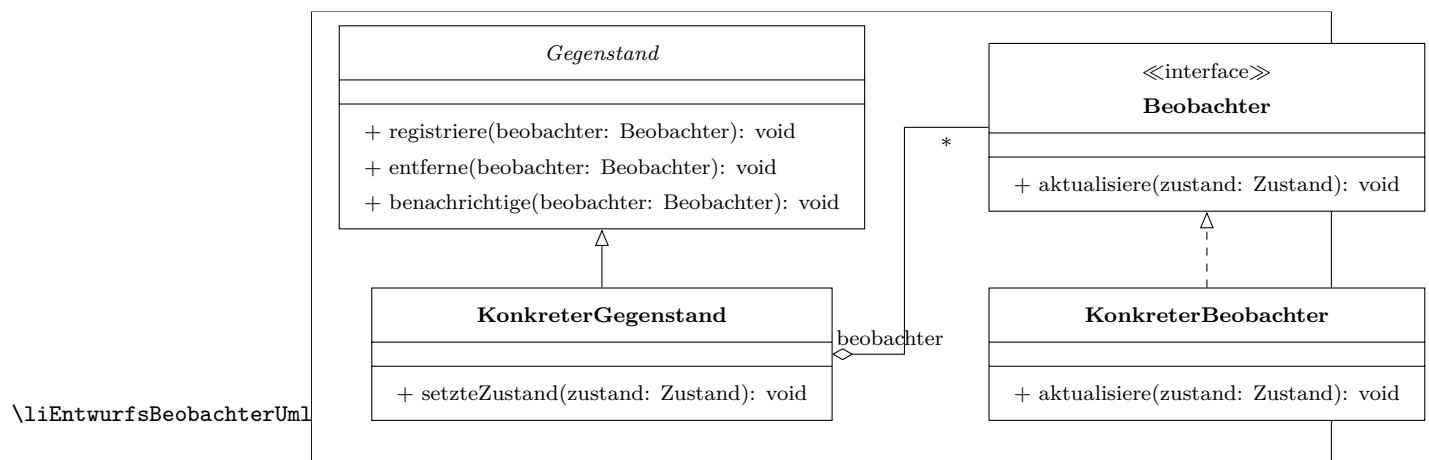
\liEntwurfsAdapter

```

859 \def\liEntwurfsAdapter{
860   \liEntwurfsAdapterUml
861   \liEntwurfsAdapterAkteure
862   \liEntwurfsAdapterCode
863 }

```

2.11.5 Beobachter (Observer)



```

864 \def\liEntwurfsBeobachterUml{
865   \begin{tikzpicture}
866     \umlclass[x=0,y=0,type=abstract]{Gegenstand}{}{

```

```

867     + registriere(beobachter: Beobachter): void\\
868     + entferne(beobachter: Beobachter): void\\
869     + benachrichtige(beobachter: Beobachter): void\\
870 }
871 \umlclass[x=0,y=-3]{KonkreterGegenstand}{-}{
872     + setzeZustand(zustand: Zustand): void
873 }
874 \umlinherit{KonkreterGegenstand}{Gegenstand}
875
876 \umlclass[x=8,y=0,type=interface]{Beobachter}{-}{
877     + aktualisiere(zustand: Zustand): void
878 }
879 \umlclass[x=8,y=-3]{KonkreterBeobachter}{-}{
880     + aktualisiere(zustand: Zustand): void
881 }
882 \umlreal{KonkreterBeobachter}{Beobachter}
883
884 \umlHVHaggreg[arg1=beobachter,pos1=0.8,mult2=*,pos2=2.5]
885 {KonkreterGegenstand}{Beobachter}
886 \end{tikzpicture}
887 }

```

\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.

```

888 \def\liEntwurfsBeobachterAkteure{
889   \begin{description}
890     \item[Gegenstand / Subjekt (Subject / Observable)]
891
892     Ein Subjekt (beobachtbares Objekt, auf Englisch publisher, also
893     „Veröffentlicher“, genannt) hat eine Liste von Beobachtern, ohne
894     deren konkrete Typen zu kennen. Es bietet eine Schnittstelle zur An-
895     und Abmeldung von Beobachtern und eine Schnittstelle zur
896     Benachrichtigung von Beobachtern über Änderungen an.\footcite[Seite
897     251]{gof}
898
899     \item[Beobachter (Observer)]
900
901     Die Beobachter (auf Englisch auch subscriber, also „Abonnent“,
902     genannt) definieren eine Aktualisierungsschnittstelle.
903
904     \item[konkreter/s Gegenstand / Subjekt (ConcreteSubject / ConcreteObservable)]
905
906     Ein konkretes Subjekt (konkretes, beobachtbares Objekt) speichert
907     den relevanten Zustand und benachrichtigt alle Beobachter bei

```



```

908 Zustandsänderungen über deren Aktualisierungsschnittstelle. Es
909 verfügt über eine Schnittstelle zur Erfragung des aktuellen
910 Zustands.
911
912 \item[Konkrete Beobachter (ConcreteObserver)]
913
914 Konkrete Beobachter verwalten die Referenz auf ein konkretes
915 Subjekt, dessen Zustand sie beobachten und speichern und dessen
916 Zustand konsistent ist. Sie implementieren eine
917 Aktualisierungsschnittstelle unter Verwendung der
918 Abfrageschnittstelle des konkreten Subjekts.
919 \footcite{wiki:beobachter}
920 \end{description}
921 }

```

\liEntwurfsBeobachterCode

```

922 \def\liEntwurfsBeobachterCode{
923 \li@EntwurfsCode{beobachter}{Gegenstand}
924 \li@EntwurfsCode{beobachter}{KonkreterGegenstand}
925 \li@EntwurfsCode{beobachter}{Beobachter}
926 \li@EntwurfsCode{beobachter}{KonkreterBeobachterA}
927 \li@EntwurfsCode{beobachter}{KonkreterBeobachterB}
928 \li@EntwurfsCode{beobachter}{Klient}
929 }

```

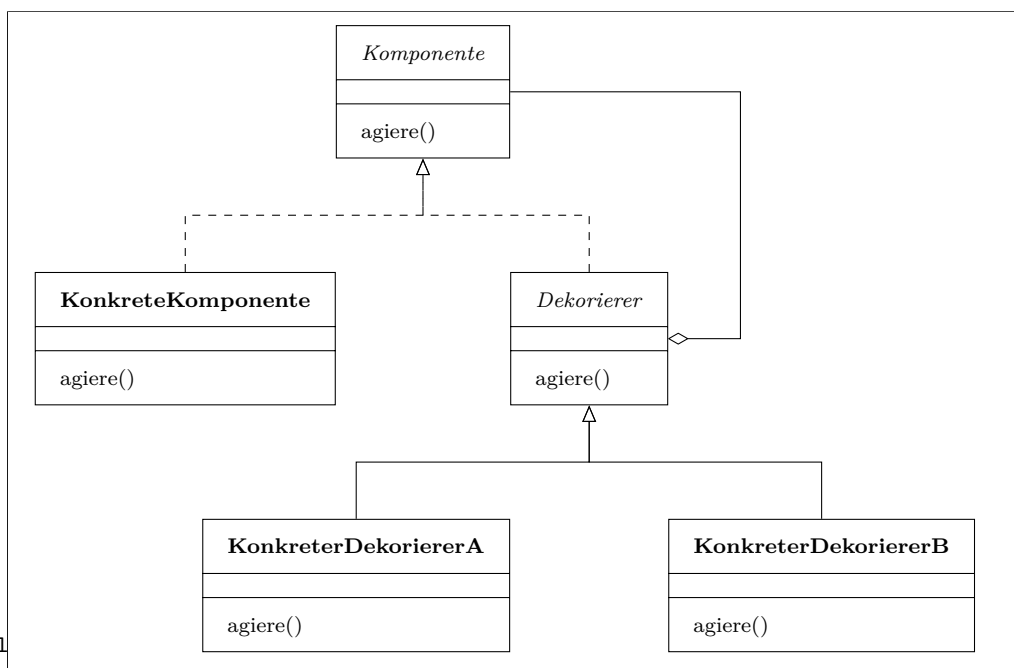
\liEntwurfsBeobachter

```

930 \def\liEntwurfsBeobachter{
931 \liEntwurfsBeobachterUml
932 \liEntwurfsBeobachterAkteure
933 \liEntwurfsBeobachterCode
934 }

```

2.11.6 Dekorierer (Decorator)



\liEntwurfsDekoriererUml

```

935 \def\liEntwurfsDekoriererUml{
936 \begin{tikzpicture}
937 \umlclass[type=abstract]{Komponente}{}{agiere()}
938 \umlclass[below left=1.5cm and 0cm of Komponente]{KonkreteKomponente}{}{agiere()}
939 \umlclass[below right=1.5cm and 0cm of Komponente,type=abstract]{Dekorierer}{}{agiere()}
940

```

```

941 \umlVHVreal{KonkreteKomponente}{Komponente}
942 \umlVHVreal{Dekorierer}{Komponente}
943
944 \umlclass[below left=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererA}{\{agiere()\}}
945 \umlclass[below right=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererB}{\{agiere()\}}
946
947 \umlVHVinherit{KonkreterDekoriererA}{Dekorierer}
948 \umlVHVinherit{KonkreterDekoriererB}{Dekorierer}
949
950 \umlHVHaggreg[arm1=2cm]{Dekorierer}{Komponente}
951 \footcite{wiki:dekorierer}
952 \end{tikzpicture}
953 }

```

\liEntwurfsDekoriererCode

```

954 \def\liEntwurfsDekoriererCode{
955 \li@EntwurfsCode{dekorierer}{Komponente}
956 \li@EntwurfsCode{dekorierer}{KonkreteKomponente}
957 \li@EntwurfsCode{dekorierer}{Dekorierer}
958 \li@EntwurfsCode{dekorierer}{KonkreterDekoriererA}
959 \li@EntwurfsCode{dekorierer}{KonkreterDekoriererB}
960 \li@EntwurfsCode{dekorierer}{Klient}
961 }

```

\liEntwurfsDekorierer

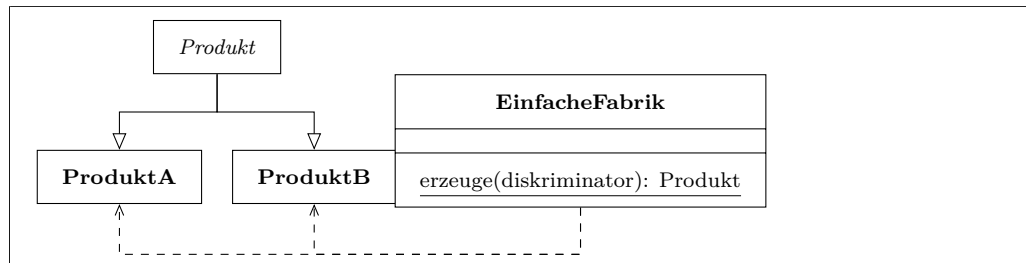
```

962 \def\liEntwurfsDekorierer{
963 \liEntwurfsDekoriererUml
964 \liEntwurfsDekoriererAkteure
965 \liEntwurfsDekoriererCode
966 }

```

2.11.7 Einfache Fabrik (Simple Factory)

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



```

967 \def\liEntwurfsEinfacheFabrikUml{
968 \begin{tikzpicture}
969 \umlsimpleclass[type=abstract]{Produkt}
970 \umlsimpleclass[below left=1cm and -0.65cm of Produkt]{ProduktA}
971 \umlsimpleclass[below right=1cm and -0.65cm of Produkt]{ProduktB}
972 \umlVHVinherit{Produkt}{ProduktA}
973 \umlVHVinherit{Produkt}{ProduktB}
974 \umlclass[below right=0cm and 1.5cm of Produkt]{EinfacheFabrik}{
975 }{
976 \umlstatic{erzeuge(diskriminator): Produkt}\}
977 }
978 \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktA}
979 \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktB}
980 \end{tikzpicture}
981 }

```

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.

```

982 \def\liEntwurfsEinfacheFabrikAkteure{
983   \begin{description}
984     \item[EinfacheFabrik]
985
986     Eine Klasse mit einer Erzeugungsmethode, die über eine größere
987     Bedingung verschiedene Objekt instanziert.
988
989     \item[Produkt]
990
991     Eine abstrakte Klasse, die von den konkreten Produkten geerbt wird.
992
993     \item[KonkretesProdukt]
994
995     Ein konkretes Produkt, das von der einfachen Fabrik erzeugt wird.
996   \end{description}
997 }
```

\liEntwurfsEinfacheFabrik

```

998 \def\liEntwurfsEinfacheFabrik{
999   \liEntwurfsEinfacheFabrikUml
1000   \liEntwurfsEinfacheFabrikAkteure
1001 }
```

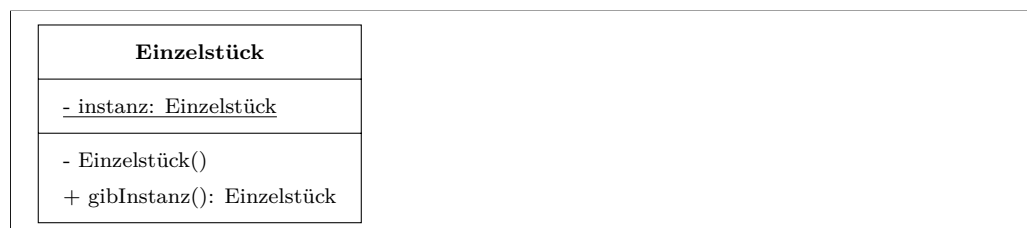
2.11.8 Einzelstück (Singleton)

urfsEinzelstueckBeschreibung

```

1002 \def\liEntwurfsEinzelstueckBeschreibung{
1003   Stellt sicher, dass nur \emph{genau eine Instanz einer Klasse} erzeugt
1004   wird.\footcite[Seite 38]{eilebrecht}
1005 }
1006
```

\liEntwurfsEinzelstueckUml Quelle nach der deutschen Wikipedia



```

1007 \def\liEntwurfsEinzelstueckUml{
1008   \begin{tikzpicture}
1009     \umlclass{Einzelstück}{
1010       \umlstatic{- instanz: Einzelstück}\\
1011     }{
1012       - Einzelstück()\\
1013       + gibInstanz(): Einzelstück
1014     }
1015   \end{tikzpicture}
1016 }
```

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.

```

1017 \def\liEntwurfsEinzelstueckAkteure{
1018   \begin{description}
1019     \item[Einzelstück (Singleton)]
1020
1021     stellt eine statische Methode bereit, mit deren Hilfe die Klienten
1022     nur auf eine einzige Instanz der Klasse zugreifen können.
1023   \end{description}
1024 }

```

\liEntwurfsEinzelstueckCode

```

1025 \def\liEntwurfsEinzelstueckCode{
1026   \li@EntwurfsCode{einzelstueck}{Einzelstueck}
1027 }

```

\liEntwurfsEinzelstueck

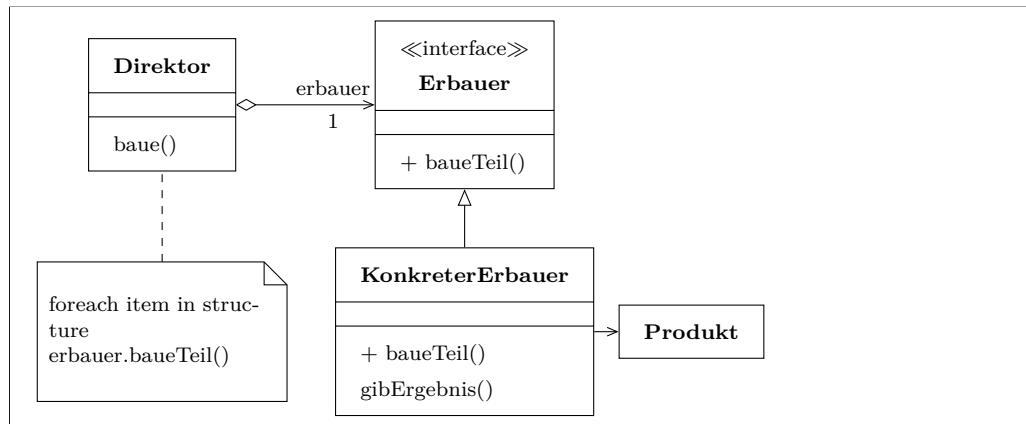
```

1028 \def\liEntwurfsEinzelstueck{
1029   \liEntwurfsEinzelstueckBeschreibung
1030
1031   \liEntwurfsEinzelstueckUml
1032
1033   \liEntwurfsEinzelstueckAkteure
1034
1035   \liEntwurfsEinzelstueckCode
1036 }

```

2.11.9 Erbauer (Builder)

\liEntwurfsErbauerUml Quelle nach der deutschen Wikipedia



```

1037 \def\liEntwurfsErbauerUml{
1038   \begin{tikzpicture}
1039     \umlsimpleclass[x=7,y=0]{Produkt}{}{}
1040     \umlclass[x=4,y=3,type=interface]{Erbauer}{}{+ baueTeil()}
1041     \umlclass[x=4,y=0]{KonkreterErbauer}{}{
1042       + baueTeil()\n
1043       gibErgebnis()}
1044     \umlclass[x=0,y=3]{Direktor}{}{baue()}
1045
1046     \umluniagg[arg2=erbauer,pos2=0.7,mult2=1]{Direktor}{Erbauer}
1047     \umluniassoc{KonkreterErbauer}{Produkt}
1048     \umlinherit{KonkreterErbauer}{Erbauer}
1049
1050     \umlnote[x=0,y=0,width=3cm]{Direktor}{
1051       foreach item in structure\n
1052       erbauer.baueTeil()

```

```

1053 }
1054 \end{tikzpicture}
1055 \footcite{wiki:erbauer}
1056 }

```

\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.

```

1057 \def\liEntwurfsErbauerAkteure{
1058   \begin{description}
1059     \item[Erbauer]
1060
1061     Der Erbauer spezifiziert eine abstrakte Schnittstelle zur Erzeugung der
1062     Teile eines komplexen Objektes.
1063
1064     \item[KonkreterErbauer]
1065
1066     Der konkrete Erbauer erzeugt die Teile des komplexen Objekts durch
1067     Implementierung der Schnittstelle. Außerdem definiert und verwaltet er
1068     die von ihm erzeugte Repräsentation des Produkts. Er bietet auch eine
1069     Schnittstelle zum Auslesen des Produkts.
1070
1071     \item[Direktor]
1072
1073     Der Direktor konstruiert ein komplexes Objekt unter Verwendung der
1074     Schnittstelle des Erbauers. Der Direktor arbeitet eng mit dem Erbauer
1075     zusammen: Er weiß, welche Baureihenfolge der Erbauer verträgt oder
1076     benötigt. Der Direktor entkoppelt somit den Konstruktionsablauf vom
1077     Klienten.
1078
1079     \item[Produkt]
1080
1081     Das Produkt repräsentiert das zu konstruierende komplexe Objekt.
1082     \footcite{wiki:erbauer}
1083   \end{description}
1084 }

```

\liEntwurfsErbauer

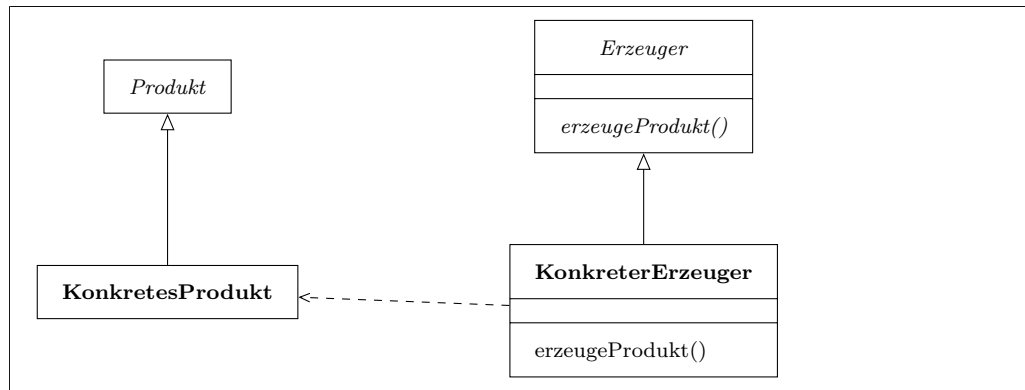
```

1085 \def\liEntwurfsErbauer{
1086   \liEntwurfsErbauerUml
1087   \liEntwurfsErbauerAkteure
1088 }

```

2.11.10 Fabrikmethode (Factory Method)

\liEntwurfsFabrikmethodeUml Quelle nach der deutschen Wikipedia



```

1089 \def\liEntwurfsFabrikmethodeUml{
1090   \begin{tikzpicture}
1091     \umlsimpleclass[type=abstract]{Produkt}
1092     \umlsimpleclass[below=2cm of Produkt]{KonkretesProdukt}
1093     \umlinherit{KonkretesProdukt}{Produkt}
1094
1095     \umlclass[type=abstract,right=4cm of Produkt]{Erzeuger}{\{
1096       \textit{erzeugeProdukt()}\}
1097   }
1098   \umlclass[below=1.2cm of Erzeuger]{KonkreterErzeuger}{\{
1099     erzeugeProdukt()
1100   }
1101   \umlinherit{KonkreterErzeuger}{Erzeuger}
1102
1103   \umldep{KonkreterErzeuger}{KonkretesProdukt}
1104 \end{tikzpicture}
1105 }

```

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).

```

1106 \def\liEntwurfsFabrikmethodeAkteure{
1107   \begin{description}
1108     \item[Produkt]
1109
1110     Das Produkt ist der Basistyp (Klasse oder Schnittstelle) für das
1111     zu erzeugende Produkt.
1112
1113     \item[KonkretesProdukt]
1114
1115     KonkretesProdukt implementiert die Produkt-Schnittstelle.
1116
1117     \item[Erzeuger]
1118
1119     Der Erzeuger deklariert die Fabrikmethode, um ein solches Produkt
1120     zu erzeugen und kann eine Default-Implementierung beinhalten.
1121
1122     \item[KonkreterErzeuger]
1123

```

```

1124     KonkreterErzeuger überschreibt die Fabrikmethode, um die ihm
1125     entsprechenden konkreten Produkte zu erzeugen (z. B. indem er den
1126     Konstruktor einer konkreten Produkt-Klasse aufruft).
1127
1128     \footcite{wiki:fabrikmethode}
1129 \end{description}
1130 }

```

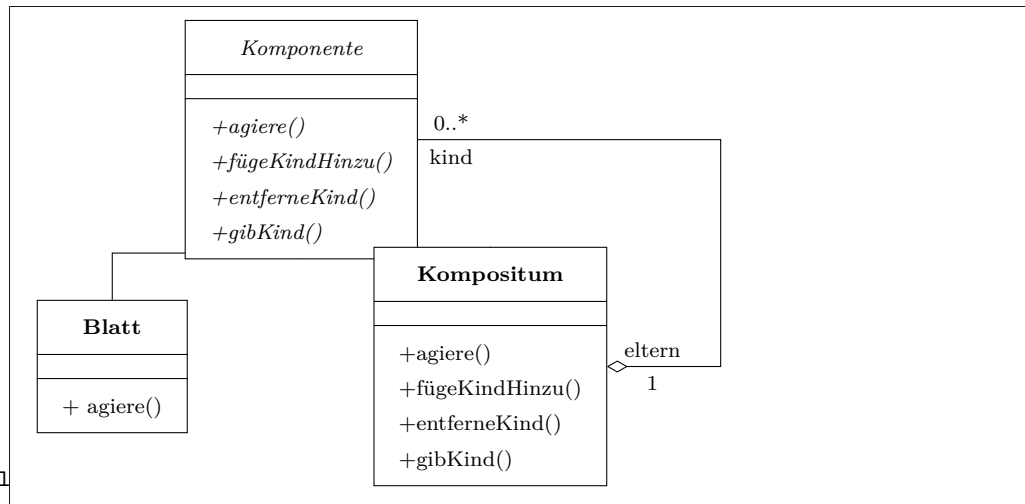
\liEntwurfsFabrikmethode

```

1131 \def\liEntwurfsFabrikmethode{
1132   \liEntwurfsFabrikmethodeUml
1133   \liEntwurfsFabrikmethodeAkteure
1134 }

```

2.11.11 Kompositum (Composite)



\liEntwurfsKompositumUml

```

1135 \def\liEntwurfsKompositumUml{
1136   \begin{tikzpicture}
1137     \umlclass[x=2.5,y=3,type=abstract]{Komponente}{-}{
1138       \textit{+agiere()}\
1139       \textit{+fügeKindHinzu()}\
1140       \textit{+entferneKind()}\
1141       \textit{+gibKind()}
1142     }
1143     \umlclass[x=0]{Blatt}{-}{+ agiere()}
1144     \umlclass[x=5]{Kompositum}{-}{
1145       +agiere()\
1146       +fügeKindHinzu()\
1147       +entferneKind()\
1148       +gibKind()
1149     }
1150
1151     \umlVHVinherit{Kompositum}{Komponente}
1152     \umlVHVinherit{Blatt}{Komponente}
1153     \umlHVVHaggreg[anchor1=east,arm1=1.5cm,arg1=eltern,mult1=1,arg2=kind,mult2=0..*,pos2=2.9,]
1154   \end{tikzpicture}
1155 }

```

\liEntwurfsFabrikmethode

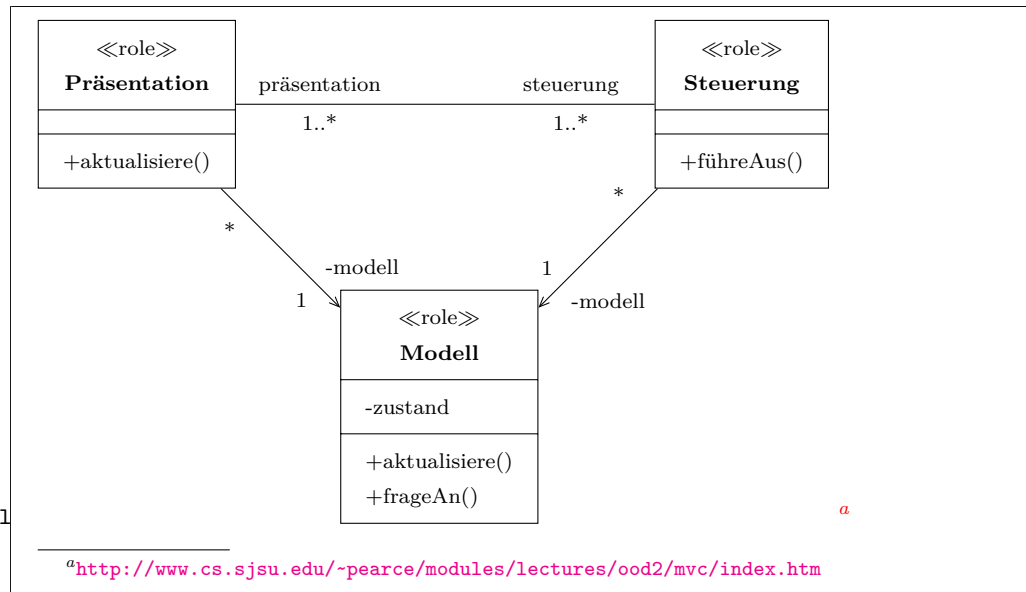
```

1156 \def\liEntwurfsKompositum{
1157   \liEntwurfsKompositumUml
1158   \liEntwurfsKompositumAkteure
1159 }

```

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

ModellPraesentationSteuerungUml



```

1160 \def\liEntwurfsModellPraesentationSteuerungUml{
1161   \begin{tikzpicture}
1162     \umlclass[x=-4cm,type=role]{Präsentation}{+aktualisiere()}
1163     \umlclass[x=4cm,type=role]{Steuerung}{+führeAus()}
1164     \umlclass[y=-4cm,type=role]{Modell}{
1165       -zustand
1166     }{
1167       +aktualisiere()\
1168       +frageAn()
1169     }
1170
1171     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Präsentation}{Modell}
1172     \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Steuerung}{Modell}
1173     \umlassoc[arg1=präsentation,mult1=1..*,arg2=steuerung,mult2=1..*]{Präsentation}{Steuerung}
1174   \end{tikzpicture}
1175   \liFussnoteUrl{http://www.cs.sjsu.edu/~pearce/modules/lectures/ood2/mvc/index.htm}
1176 }

```

ModellPraesentationSteuerung

```

1177 \def\liEntwurfs{
1178   \liEntwurfsModellPraesentationSteuerungUml
1179   \liEntwurfsModellPraesentationSteuerungAkteure
1180 }

```

2.11.13 Stellvertreter (Proxy)

\liEntwurfsStellvertreterUml

```

1181 \def\liEntwurfsStellvertreterUml{
1182   \begin{tikzpicture}
1183     \umlsimpleclass[x=-1,y=2]{Klient}
1184
1185     \umlclass[x=2,y=2]{Subjekt}{+ agiere()}
1186     \umlclass[x=0,y=-1]{KonkretesSubjekt}{+ agiere()}
1187     \umlclass[x=4,y=-1]{Stellvertreter}{+ agiere()}
1188
1189     \umlVHVinherit{KonkretesSubjekt}{Subjekt}
1190     \umlVHVinherit{Stellvertreter}{Subjekt}
1191     \umluniassoc{Stellvertreter}{KonkretesSubjekt}
1192     \umluniassoc{Klient}{Subjekt}
1193   \end{tikzpicture}
1194 }

```

\liEntwurfsStellvertreterCode


```

1195 \def\liEntwurfsStellvertreterCode{
1196   \li@EntwurfsCode{stellvertreter}{Subjekt}
1197   \li@EntwurfsCode{stellvertreter}{KonkretesSubjekt}
1198   \li@EntwurfsCode{stellvertreter}{Stellvertreter}
1199   \li@EntwurfsCode{stellvertreter}{Klient}
1200 }

```

\liEntwurfsStellvertreter

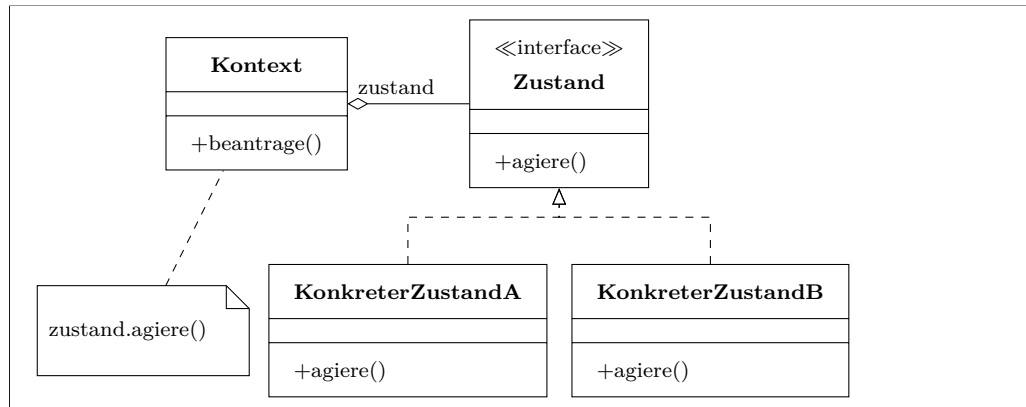
```

1201 \def\liEntwurfsStellvertreter{
1202   \liEntwurfsStellvertreterUml
1203   \liEntwurfsStellvertreterCode
1204 }

```

2.11.14 Zustand (State)

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



```

1205 \def\liEntwurfsZustandUml{
1206   \begin{tikzpicture}
1207     \umlclass[x=-1,y=3]{Kontext}{+beantrage()}
1208     \umlclass[x=3,y=3,type=interface]{Zustand}{+agiere()}
1209     \umlclass[x=1,y=0]{KonkreterZustandA}{+agiere()}
1210     \umlclass[x=5,y=0]{KonkreterZustandB}{+agiere()}
1211
1212     \umlVHVreal{KonkreterZustandA}{Zustand}
1213     \umlVHVreal{KonkreterZustandB}{Zustand}
1214
1215     \umlagg[arg=zustand,pos=0.4]{Kontext}{Zustand}
1216
1217     \umlnote[x=-2.5,y=0,width=2.5cm]{Kontext}{zustand.agiere()}
1218   \end{tikzpicture}
1219 }

```

\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.

```

1220 \def\liEntwurfsZustandAkteure{
1221   \begin{description}
1222     \item[Kontext (Context)]
1223
1224     definiert die clientseitige Schnittstelle und verwaltet die separaten

```

```

1225     Zustandsklassen.
1226
1227     \item[State (Zustand)]
1228
1229     definiert eine einheitliche Schnittstelle aller Zustandsobjekte und
1230     implementiert gegebenenfalls ein Standardverhalten.
1231
1232     \item[KontreterZustand (ConcreteState)]
1233
1234     implementiert das Verhalten, das mit dem Zustand des Kontextobjektes
1235     verbunden ist.
1236 \end{description}
1237 }

```

\liEntwurfsZustandCode

```

1238 \def\liEntwurfsZustandCode{
1239   \li@EntwurfsCode{zustand}{Kontext}
1240   \li@EntwurfsCode{zustand}{Zustand}
1241 }

```

\liEntwurfsZustand

```

1242 \def\liEntwurfsZustand{
1243   \liEntwurfsZustandUml
1244   \liEntwurfsZustandAkteure
1245   \liEntwurfsZustandCode
1246 }

```

```

1247

```

2.12 er.sty

```
1248 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1249 \ProvidesPackage{lehramt-informatik-er}[2020/06/13 Zum Setzen von
1250 ER-Diagrammen]
```

```
1251 \RequirePackage{tikz-er2}
1252 \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);

```

```
1253 \RequirePackage{soul}
```

```
1254 \RequirePackage{fontawesome}
```

2.12.3 Makro-Kürzel

```

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

```

```
1255 \ExplSyntaxOn
```

```
\liErEntity
```

```
1256 \def\liErEntity#1{\textbf{#1}}
```

```
\liErRelationship
```

```
1257 \def\liErRelationship#1{\ul{#1}}
```

```
\liErAttribute
```

```
1258 \def\liErAttribute#1{\emph{#1}}
```

```
\liErMpEntity mp = marginpar
```

Let-Abkürzung: \let\e=\liErMpEntity

```

1259 \def\liErMpEntity#1{
1260   \liErEntity{#1}
1261   \marginpar{
1262     \liErEntity{\tiny\faSquareO{~E:~#1}
1263   }
1264 }

```

□

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

```

1265 \def\liErMpRelationship#1{
1266   \liErRelationship{#1}
1267   \marginpar{
1268     \liErRelationship{\tiny\faGg{~R:~#1}
1269   }
1270 }

```

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

```

1271 \def\liErMpAttribute#1{
1272   \liErAttribute{#1}
1273   \marginpar{
1274     \liErAttribute{\tiny\faCircleThin{~A:~#1}
1275   }
1276 }

```

```

\liErDatenbankName Let-Abkürzung: \let\d=\liErDatenbankName
                    datenbank name
1277 \def\liErDatenbankName#1{
1278   {
1279     \footnotesize\texttt{(#1)}
1280   }
1281 }

1282 \ExplSyntaxOff
1283

```

2.13 formale-sprachen.sty

```

1284 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1285 \ProvidesPackage{lehramt-informatik-formale-sprachen}[2021/02/21 Hilfsmakros
1286 zum Setzen von mathematischen Formeln bei dem Thema Formale Sprachen]

1287 \directlua{
1288   formale_sprachen = require('lehramt-informatik-formale-sprachen')
1289 }

1290 \RequirePackage{hyperref}

1291 \liLadePakete{mathe,typographie}

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

1292 \def\liMengeOhneMathe#1{\{ #1 \}}
1293 \def\liMenge#1{%
1294   \ifmmode%
1295     \liMengeOhneMathe{#1}%
1296   \else%
1297     $\liMengeOhneMathe{#1}$%
1298   \fi%
1299 }

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

1300 \def\liEpsilon{$\varepsilon$}

\liPotenzmenge Umgeben mit geschweiften Klammern in einer Mathematik-Umgebung

1301 \def\erzeuge@tiefgestellt#1{\directlua{formale_sprachen.erzeuge_tiefgestellt('#1')}}
1302 \def\liPotenzmengeOhneMathe#1{\{ \erzeuge@tiefgestellt{#1} \}}
1303 \def\liPotenzmenge#1{$\liPotenzmengeOhneMathe{#1}$}

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

1304 \let\liZustandsmengeOhneMathe=\liPotenzmengeOhneMathe
1305 \let\liZustandsmenge=\liPotenzmenge

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

1306 \def\liUeberfuehrungsFunktionOhneMathe#1{\delta(\erzeuge@tiefgestellt{#1})}
1307 \def\liUeberfuehrungsFunktion#1{
1308   \ifmmode
1309     \liUeberfuehrungsFunktionOhneMathe{#1}
1310   \else
1311     $\liUeberfuehrungsFunktionOhneMathe{#1}$
1312   \fi
1313 }

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

1314 \def\liAlphabet#1{$\Sigma = \{ #1 \}$}

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

1315 \def\liBandAlphabet#1{$\Gamma = \Sigma \cup \{ #1 \}$}

\liZustandsBuchstabe

1316 \def\liZustandsBuchstabe{z}

\liZustandsBuchstabeGross

1317 \def\liZustandsBuchstabeGross{Z}

```

```

\liZustandsmengeNr
1318 \def\zustandsnamens@liste#1#2{
1319   $
1320   \{
1321     \directlua{formale_sprachen.erzeuge_zustandsnamens_liste('#1', '#2')}
1322   \}
1323   $
1324 }
1325 \def\liZustandsmengeNr#1{\zustandsnamens@liste{\liZustandsBuchstabe}{#1}}

\liZustandsmengeNrGross
1326 \def\liZustandsmengeNrGross#1{\zustandsnamens@liste{\liZustandsBuchstabeGross}{#1}}

\liZustandsname \liZustandsname{1}: $z_1$
1327 \def\liZustandsname#1{\liZustandsBuchstabe_#1$}

\liZustandsnameGross \liZustandsnameGross{1}: $Z_1$
1328 \def\liZustandsnameGross#1{\liZustandsBuchstabeGross_#1$}

\liAbleitung \liAbleitung{S -> aB -> ab}:  $S \vdash aB \vdash ab$ 
1329 \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}

1330 \NewDocumentEnvironment { liProduktionsRegeln }
1331 { 0{P} +b }
1332 {
1333   \liGeschweifteKlammern{#1}
1334   {
1335     \begin{align*}
1336       \directlua{formale_sprachen.produktions_regeln('#2')}
1337     \end{align*}
1338   }{-0.2cm}{-1.5cm}
1339 } {}

\liProduktionen \liProduktionen{S -> A, A -> a}:  $\{S \rightarrow A, A \rightarrow a\}$ 
1340 \def\liProduktionen#1{
1341   \liMenge{\directlua{formale_sprachen.produktions_regeln('#1', true)}}
1342 }

\liZustandsnameTiefgestellt Automatisch tiefgestellte Nummerierung \z1
Let-Abkürzung: \let\z=\liZustandsnameTiefgestellt
1343 \def\liZustandsnameTiefgestellt#1{
1344   \ifmmode
1345     \liZustandsBuchstabe\sb{#1}
1346   \else
1347     $\liZustandsBuchstabe\sb{#1}$
1348   \fi
1349 }

1350 \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}

```

```

1351 \NewDocumentCommand{ \liAusdruck } { O{L} m m } {
1352   $
1353   \tl_if_empty:nTF {#1} {} {#1 =}
1354   \{
1355     \, #2 \,
1356     |
1357     \, #3 \,
1358   \}$
1359 }
1360 \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

1361 \def\liFlaci#1{%
1362   \par
1363   {%
1364     \scriptsize
1365     Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte
1366     Automaten, Compiler und Interpreter) Ein Projekt der Hochschule
1367     Zittau/Görlitz und der Pädagogischen Hochschule Schwyz:
1368     \href{https://flaci.com/#1}{flaci.com/#1}%
1369   }%
1370   \par
1371 }

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

1372 \ExplSyntaxOn
1373 \NewDocumentCommand {\liGrammatik} { O{G} m } {
1374   \tl_set:Nn \l_variablen_tl {V}
1375   \tl_set:Nn \l_alphabet_tl {\Sigma}
1376   \tl_set:Nn \l_produktionen_tl {P}
1377   \tl_set:Nn \l_start_tl {S}
1378
1379   \keys_define:nn { grammatik } {
1380     variablen .code:n = {\tl_set:Nn \l_variablen_tl {\liMenge{##1}}},
1381     alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
1382     produktionen .code:n = {\tl_set:Nn \l_produktionen_tl {\liProduktionen{##1}}},
1383     start .code:n = {\tl_set:Nn \l_start_tl {##1}},
1384   }
1385
1386   \keys_set:nn { grammatik } { #2 }
1387
1388   $#1 = (
1389     \l_variablen_tl,
1390     \l_alphabet_tl,
1391     \l_produktionen_tl,
1392     \l_start_tl
1393   )$
1394 }
1395 \ExplSyntaxOff
1396

```


2.14 formatierung.sty

```
1397 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1398 \ProvidesPackage{lehramt-informatik-formatierung}[2020/11/27]
```

2.14.1 Schriftarten / Typographie

The package mathpazo Loading this package changes the default roman font family to Adobe Palatino, and the virtual ‘mathpazo’ fonts will be used for math. These virtual fonts are made up basically from Palatino Italic, with the missing math symbols coming from the CM and Pazo math fonts.

```
1399 \RequirePackage{mathpazo}
1400 \RequirePackage[no-math]{fontspec}
1401 \setmainfont{texgyrepagella}
```

2.14.2 Farben

```
1402 \RequirePackage{xcolor}
1403 \definecolor{infogray}{rgb}{0.97,0.97,0.97}
```

2.14.3 Überschriften

```
1404 \RequirePackage{titlesec}
1405 \titleformat{\chapter}[display]{\bfseries}{\Opt}{\LARGE}
1406 \titlespacing{\chapter}{\Opt}{\Opt}{*1}
1407 \titleformat{\paragraph}[hang]{\normalsize\bfseries}{\theparagraph}{1em}{\}
1408 \setcounter{secnumdepth}{0}
```

2.14.4 Listen

```
1409 \RequirePackage{paralist}
1410 \renewcommand\labelitemi{-}
1411 \renewcommand\labelitemii{-}
1412 \renewcommand\labelitemiii{-}
1413 \renewcommand\labelitemiv{-}
1414 % Counter: enumi enumii enumiii enumiv
1415 % Styles: \arabic{counter} \alph{counter} \Alph{counter} \roman{counter} \Roman{counter}
1416 \renewcommand{\labelenumi}{(\alph{enumi})}
1417 \renewcommand{\labelenumii}{(\roman{enumii})}
```

2.14.5 Kasten

```
1418 \RequirePackage{mdframed}
1419 \mdfsetup{backgroundcolor=infogray}
```

liKasten

```
1420 \NewDocumentEnvironment { liKasten }{ } {
1421   \begin{mdframed}[backgroundcolor=white!0]
1422 } {
1423   \end{mdframed}
1424 }
```

2.14.6 Header

```
1425 \RequirePackage{fancyhdr}
1426 \fancyhead[L,C,R]{\}
1427 \fancyfoot[L]{\}
1428 \fancyfoot[C]{\}
1429 \fancyfoot[R]{\thepage}
1430 \pagestyle{fancy}
1431 \renewcommand{\headrulewidth}{\Opt}
1432 \renewcommand{\footrulewidth}{\Opt}
```

2.14.7 Zeilenabstände

Werden kleinere Schriften verwendet, passt sich der Zeilenabstand nicht entsprechend an. Mit der Umgebung spacing funktioniert es dann.

```
1433 \RequirePackage{setspace}
```


2.15 gantt.sty

```

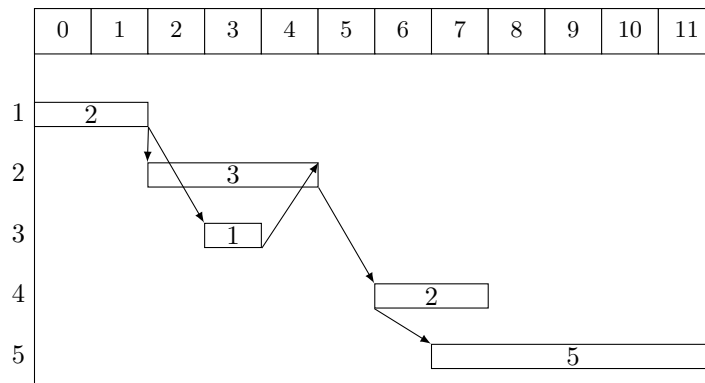
1435 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1436 \ProvidesPackage{lehramt-informatik-gantt}[2020/09/05]

\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}

```



```

1437 \RequirePackage{tikz-uml}
1438 \RequirePackage{pgfgantt}
1439 \setganttlinklabel{f-s}{}
1440 \setganttlinklabel{s-s}{}
1441 \setganttlinklabel{f-f}{}
1442 \setganttlinklabel{s-f}{}

1443

```

2.16 grafik.sty

```
1444 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1445 \ProvidesPackage{lehramt-informatik-grafik}[2020/12/27 Lädt das Paket
1446 TikZ, um Grafiken zeichnen zu können und graphicx um Bilder laden zu können.]
1447 \ExplSyntaxOn
1448 \RequirePackage{tikz}
1449 \RequirePackage{graphicx}

\liGrafikLogoPfad

1450 \def\liGrafikLogoPfad#1{
1451   \LehramtInformatikRepository / .tex / Logo / #1
1452 }

\liGrafikCCLizenz

1453 \NewDocumentCommand{ \liGrafikCCLizenz } { 0{} } {
1454   \includegraphics[#1]{
1455     \liGrafikLogoPfad{CC-by-nc-sa.eps}
1456   }
1457 }

\liGrafikLogo

1458 \NewDocumentCommand{ \liGrafikLogo } { 0{} } {
1459   \includegraphics[#1]{
1460     \liGrafikLogoPfad{Logo_nur-Pfade.eps}
1461   }
1462 }

1463 \ExplSyntaxOff
1464
```

2.17 graph.sty

```

1465 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1466 \ProvidesPackage{lehramt-informatik-graph}[2020/06/09]
1467 \RequirePackage{tikz}

```

Für die die Adjazenzliste (xrightarrow)

```

1468 \RequirePackage{amsmath}

```

Für Adjazenz-Matrix

```

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

```

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

```

1469 \RequirePackage{blkarray}
1470 \usetikzlibrary{arrows.meta}

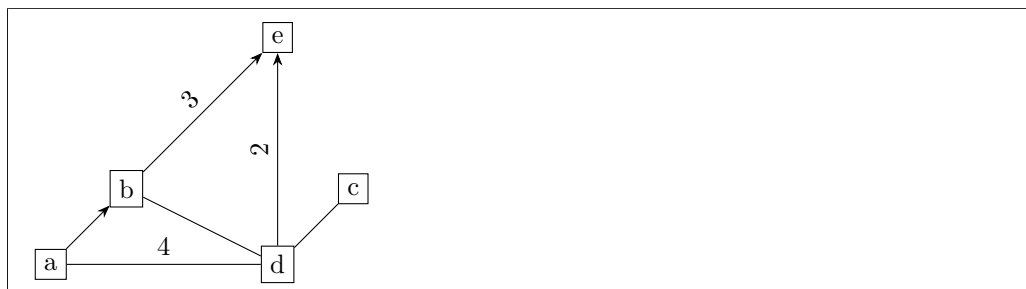
```

```

\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}

```



```

1471 \tikzset{
1472   li graph/.style={
1473     every node/.style={
1474       rectangle,
1475       draw,
1476     },
1477     every edge/.style={
1478       >={Stealth[black]},
1479       draw,
1480     },
1481     every edge/.append style={
1482       every node/.style={
1483         sloped,
1484         auto,
1485       }
1486     }
1487   },
1488   li markierung/.style={
1489     ultra thick,
1490   }
1491 }

```

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}

```

```

1492 \NewDocumentEnvironment { liGraphenFormat }{ +b } {} {}

```

```

1493

```

2.18 hanoi.sty

```
1494 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1495 \ProvidesPackage{lehramt-informatik-hanoi}[2020/12/19 Paket zum Setzen
1496 von Hanoi-Grafiken]
```

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

```
1497 \RequirePackage{tikz}
1498 \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}

1499 \def\li@mset #1[#2]=#3{%
1500 \expandafter\xdef\csname #1#2\endcsname{#3}
1501 }
1502 \def\li@mget #1[#2]{%
1503 \csname #1#2\endcsname
1504 }
1505 \def\li@minc #1[#2] += #3{%
1506 \pgfmathparse{\li@mget #1[#2] + #3}%
1507 \li@mset #1[#2] = \pgfmathresult
1508 }
1509
1510 \def\liHanoi#1#2{
1511   \edef\li@numdiscs{#1}
1512   \def\li@sequence{#2}
1513   \begin{tikzpicture}[line width=4mm,brown!40,line cap=round,xscale=3]
1514     % init colors
1515     \foreach[count=\j] \c in {red,green!80!black,blue,orange,violet,gray,yellow!80!black,purp
1516     \li@mset col[\j]={\c};
1517     % draw poles and init pole counters
1518     \foreach \j in {1,2,3}{
1519       \li@mset pos[\j]=0
1520       \draw (\j,-.5) -- +(0, .5 + .5*\li@numdiscs);
1521     }
1522     % draw base
1523     \draw (.5,-.5) -- +(3,0);
1524     % draw discs
1525     \foreach[count=\k] \i/\j in \li@sequence{
1526       \draw[draw={\li@mget col[\i]}] (\j,\li@mget pos[\j]) +(-.4*\i/\li@numdiscs,0) -- +(.4*
1527       \li@minc pos[\j] += {.5}
1528     }
1529   \end{tikzpicture}
1530 }

1531
```

2.19 klassen-konfiguration-aufgabe.sty

```
1532 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1533 \ProvidesPackage{lehramt-informatik-klassen-konfiguration-aufgabe}[2021/09/01 Die
1534 Klasse Aufgabe konfigurieren, d. h. Laden von einigen Paketen, Registrierung von Hooks]
```

Formatierung muss vor literatur sein, sonst Option clash

```
1535 \liLadePakete{
1536   formatierung,
1537   abmessung,
1538   literatur-dummy,
1539   makros,
1540   aufgaben-metadaten,
1541   kopf-fusszeilen,
1542   mathe,
1543   grafik,
1544   meta
1545 }
```

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

```
1546 \RequirePackage[ngerman]{babel}
```

In Aufgaben wollen wir das Additum sehen.

```
1547 \LiADDITUMtrue
1548 \ExplSyntaxOn

1549 \cs_gset:Npn \stichwoerter_formatiert: {
1550   \tl_if_empty:NTF \g_auf_stichwoerter_tl {} {
1551     \textbf{Stichwörter:} ~
1552     \g_auf_stichwoerter_tl
1553     \par
1554   }
1555 }

1556 \cs_gset:Npn \horizontale_linie: {
1557   \par
1558   \noindent
1559   \rule{\textwidth}{0.8pt}
1560   \par
1561 }

1562 \cs_gset:Npn \thematik_formatiert: {
1563   \tl_if_empty:NTF \g_auf_thematik_tl {} {
1564     \textit{
1565       ( \g_auf_thematik_tl )
1566     }
1567   }
1568 }
```

`\liAufgabenMetadaten`

```
1569 \def\liAufgabenMetadaten#1{
1570   \liMetaSetze{#1}
1571
1572   \tl_if_empty:NTF \g_auf_examen_nummer_tl {} {
1573     {
1574       \noindent
1575       \large
1576       \gib_einzelpruefung_trenner:
1577       \par\medskip
1578     }
1579   }

1580
1581   {
1582     \noindent
1583     \bfseries
1584     \Large
1585     \tl_if_empty:NTF \g_auf_examen_nummer_tl {
```



```

1586     \g_auf_titel_tl
1587   } {
1588     \_gib_aufgaben_pfad_trenner:
1589   }
1590 }
1591 \hfill \thematik_formatiert:
1592 \par
1593
1594 \medskip
1595
1596 \noindent
1597 {\footnotesize\stichwoerter_formatiert:}
1598
1599 \horizontale_linie:
1600
1601 \bigskip
1602
1603 \par
1604 % Keine Einrückung
1605 \@afterindentfalse
1606 \@afterheading
1607 }

1608 \AddToHook{enddocument}{
1609   \vfill
1610   {
1611
1612     \liLogoTextProjekt
1613     \bigskip
1614
1615     \liLogoTextCCLizenz
1616     \bigskip
1617
1618     \begin{spacing}{1}
1619       \tiny
1620       \noindent
1621       \liMetaHilfMit
1622
1623       \liMetaQuelltext
1624       \_gib_github_url_href:
1625     \end{spacing}
1626   }
1627 }

1628 \ExplSyntaxOff
1629

```

2.20 klassen-konfiguration-examen.sty

```
1630 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1631 \ProvidesPackage{lehramt-informatik-klassen-konfiguration-examen}[2021/09/04 Die
1632 Klasse liSetzeExamen konfigurieren, d. h. Laden von einigen Paketen, Registrierung von Hooks]
```

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

```
1633 \liLadePakete{
1634   formatierung,
1635   literatur-dummy,
1636   makros,
1637   aufgaben-metadaten,
1638   abmessung,
1639   typographie,
1640   grafik,
1641   meta
1642 }
```

Formatierung für die Überschriften setzen.

```
1643 \RequirePackage{titlesec}
1644 \titleformat{\section}{\huge\filcenter\bfseries}{\thesection}{1em}{}
1645 \newcommand{\sectionbreak}{\clearpage}
1646 \setcounter{secnumdepth}{0}
1647 \liLadeAllePakete
```

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

```
1648 \RequirePackage[ngerman]{babel}
1649 \RequirePackage{standalone}
1650 \ExplSyntaxOn
```

`\liSetzeExamenThemaNr`

```
1651 \def\liSetzeExamenThemaNr#1{
1652   \tl_gset:Nn \g_auf_examen_thema_nr_tl { #1 }
1653   \section{Thema-Nr.~#1}
1654 }
```

`\liSetzeExamenTeilaufgabeNr`

```
1655 \def\liSetzeExamenTeilaufgabeNr#1{
1656   \tl_gset:Nn \g_auf_examen_teilaufgabe_nr_tl { #1 }
1657   \subsection{Teilaufgabe-Nr.~#1}
1658 }
```

`\liBindeAufgabeEin`

```
1659 \def\liBindeAufgabeEin#1{
1660   \tl_gset:Nn \g_auf_examen_aufgabe_nr_tl { #1 }
1661   \input{
1662     \LehramtInformatikRepository /
1663     Staatsexamen /
1664     \g_auf_examen_nummer_tl /
1665     \g_auf_examen_jahr_tl /
1666     \g_auf_examen_monat_tl /
1667     \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} {
1668       Thema - \g_auf_examen_thema_nr_tl /
1669     }
1670     \tl_if_empty:NTF \g_auf_examen_teilaufgabe_nr_tl {} {
1671       Teilaufgabe - \g_auf_examen_teilaufgabe_nr_tl /
1672     }
1673     Aufgabe - \g_auf_examen_aufgabe_nr_tl .tex
1674   }
1675 }
```

`\liAufgabenMetadaten` Das Metadaten-Makro überschreiben

```
1676 \def\liAufgabenMetadaten#1{
1677   \liMetaSetze{#1}
1678   \subsubsection{\_gib_aufgaben_titel:}
1679 }
```

```

1680 \cs_new:Npn \titel_seite:
1681 {
1682   \pagestyle{empty}
1683   \begin{center}
1684     \large
1685     Erste~Staatsprüfung~für~ein~Lehramt~an~öffentlichen~Schulen \par
1686
1687     \vspace{0.5cm}
1688
1689     Fach~Informatik \par
1690
1691     \vfill
1692
1693     \liGrafikLogo[width=8cm]\par
1694     \bigskip
1695     Die~Bschlangaul-Sammlung \par
1696     {\footnotesize \liMetaHermineBschlangaulAndFriends} \par
1697
1698     \vfill
1699
1700     {
1701       \bfseries\Huge
1702
1703       \g_auf_examen_jahreszeit_tl \par
1704
1705       \g_auf_examen_jahr_tl \par
1706     }
1707
1708     \vspace{2cm}
1709
1710     {\LARGE \g_auf_examen_nummer_tl \par}
1711
1712     \vspace{0.5cm}
1713
1714     \g_auf_examen_fach_tl \par
1715
1716     \vspace{3cm}
1717
1718     Aufgabenstellungen~mit~Lösungsvorschlägen \par
1719
1720   \end{center}
1721
1722   \vfill
1723 }
1724 \cs_new:Npn \inhalts_verzeichnis: {
1725   % für den Abstand vor den section im Inhaltsverzeichnis
1726   % https://tex.stackexchange.com/questions/241445/how-to-control-spacing-in-toc-for-different-sections
1727   \setlength{\cftbeforesecskip}{1.5cm}
1728   \setlength{\cftbeforesubsecskip}{0.5cm}
1729   \renewcommand{\cftsubsecafterpnum}{\vspace{\cftbeforesubsecskip}}
1730   \setlength{\cftbeforesubsubsecskip}{0.1cm}
1731   \renewcommand{\contentsname}{Aufgabenübersicht}
1732   \tableofcontents
1733 }
1734 \setcounter{tocdepth}{4}
1735 \RequirePackage[titles]{tocloft}
1736 \AddToHook{begindocument}{
1737   \titel_seite:
1738
1739   \clearpage
1740
1741   \inhalts_verzeichnis:

```

```
1742
1743 \vfill
1744
1745 \liLogoTextProjekt
1746 \bigskip
1747
1748 \liLogoTextCCLizenz
1749 \bigskip
1750
1751 \clearpage
1752 }
1753 \ExplSyntaxOff
1754
```

2.21 komplexitaetstheorie.sty

```
1755 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1756 \ProvidesPackage{lehramt-informatik-komplexitaetstheorie}[2021/07/08 Zum
1757 Setzen von Karp's NP-vollständigen Problemen, Hilfsmakros für die
1758 Polynomialzeitreduktion.]
```

2.21.1 Makro-Kürzel

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

```
1759 \liLadePakete{mathe}
```

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

```
1760 \RequirePackage{mdframed}
```

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

```
1761 \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`

```
1762 \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`

```
1763 \def\liProblemBeschreibung#1#2#3{
1764   \begin{mdframed}[
1765     userdefinedwidth=9cm,
1766     align=center,
1767     backgroundcolor=white!0,
1768   ]
1769     \centerline{\large\liProblemName{#1}}
1770
1771     \medskip
1772
1773     \begin{description}
1774       \item[Gegeben:] #2
1775       \item[Frage:] #3
1776     \end{description}
1777   \end{mdframed}
1778 }
```

```

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

1779 \NewDocumentCommand{ \liPolynomiellReduzierbar } { m O{p} m } {
1780 \begin{displaymath}
1781   \liProblemName{#1}
1782   \preceq_{#2}
1783   \liProblemName{#3}
1784 \end{displaymath}
1785 }

\liProblemVertexCover

1786 \def\liProblemClique{%
1787 Das \textbf{Cliquesproblem} fragt nach der Existenz einer Clique der
1788 Mindestgröße  $n$  in einem gegebenen Graphen.
1789 \footcite[Seite 76]{theo:fs:4}
1790 Eine Clique ist eine Teilmenge von Knoten in einem ungerichteten
1791 Graphen, bei der \emph{jedes Knotenpaar durch eine Kante} verbunden ist.
1792 \footcite{wiki:cliquenproblem}
1793 }

\liProblemVertexCover

1794 \def\liProblemVertexCover{%
1795 %
1796 Das \textbf{Knotenüberdeckungsproblem} (\liProblemName{Vertex Cover})
1797 fragt, ob zu einem gegebenen einfachen Graphen und einer natürlichen
1798 Zahl  $k$  eine Knotenüberdeckung der Größe von höchstens  $k$  existiert.
1799 \footcite{wiki:knotenueberdeckung}
1800
1801 Das heißt, ob es eine aus maximal  $k$  Knoten bestehende Teilmenge  $U$ 
1802 der Knotenmenge gibt, so dass jede Kante des Graphen mit
1803 mindestens einem Knoten aus  $U$  verbunden ist.
1804 \footcite[Seite 78]{theo:fs:4}%
1805 }

\liProblemSubsetSum Kein Karp-21-Problem

1806 \def\liProblemSubsetSum{%
1807 Das \textbf{Teilsommenproblem} (\liProblemName{Subset Sum} oder
1808 \liProblemName{SSP}) ist ein spezielles Rucksackproblem.
1809 \footcite{wiki:teilsommenproblem}
1810 Gegeben sei eine Menge von ganzen Zahlen  $I = \{w_1, w_2, \dots, w_n\}$ .
1811 Gesucht ist eine Untermenge, deren Elementsumme maximal, aber nicht
1812 größer als eine gegebene obere Schranke  $c$  ist.
1813 \footcite[Seite 74]{theo:fs:4}
1814 }

\liProblemSubsetSum Kein Karp-21-Problem

1815 \def\liProblemSat{%
1816 Das \textbf{Erfüllbarkeitsproblem der Aussagenlogik} \liProblemName{Sat}
1817 und \liProblemName{k-SAT} mit  $k \geq 3$ ,  $k \in \mathbb{N}$  (Satz von
1818 Cook) fragt, ob eine aussagenlogische Formel erfüllbar
1819 ist.\footcite{wiki:sat} Das Erfüllbarkeitsproblem der
1820 \emph{Aussagenlogik} ist in exponentieller Zeit in Abhängigkeit der
1821 Anzahl der Variablen mit Hilfe einer Wahrheitstabelle entscheidbar.
1822 Diese \emph{Wahrheitstabelle} kann nicht in polynomieller Zeit
1823 aufgestellt werden.
1824 \footcite[Seite 71]{theo:fs:4}
1825 }

1826

```

2.22 kontrollflussgraph.sty

1827 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

1828 \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 = ...}]
```

```

1829 \RequirePackage{tikz}
1830 \usetikzlibrary{positioning}
1831 \tikzset{
1832   li kontrollfluss/.style={
1833     knoten/.style={
1834       circle,
1835       draw
1836     },
1837     usebox/.style={
1838       draw,
1839       rectangle,
1840       font=\scriptsize,
1841       anchor=west,
1842       align=left,
1843     },
1844     bedingung/.style={
1845       midway,
1846       draw=none,
1847       font=\scriptsize
1848     },
1849     knotenbeschriftung/.style={
1850       draw,
1851       rectangle,
1852       midway,
1853       font=\scriptsize
1854     },
1855     wahr/.style={
1856       thick
1857     },
1858     falsch/.style={
1859       dashed
1860     },
1861     every node/.style={
1862       circle,
1863       draw,
1864     },
1865     every edge/.append style={
1866       every node/.style={
1867         draw=none,
1868         bedingung,
1869       }
1870     },
1871     every path/.style={
1872       draw,
1873       ->,
1874     },
1875     every pin/.style={
1876       draw,
1877       dotted,
1878       rectangle,
1879       pin position=right
1880     },
1881     every pin edge/.style={
1882       dotted,
1883       arrows=-,
1884     }
1885   }
1886 }

```

2.22.4 Umgebungen

liKontrollflussgraph

```

1887 \NewDocumentEnvironment { liKontrollflussgraph } { 0{} } {

```



```

1888 \begin{tikzpicture}[
1889     li kontrollfluss,
1890     #1
1891 ]
1892 } {
1893 \end{tikzpicture}
1894 }

```

2.22.5 Makros

`\liAnweisung`

```
1895 \def\liAnweisung#1(#2,#3){\node[knoten] at (#2,#3) (#1) {#1};}
```

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

```
1896 \def\liBedingung#1#2{\node[bedingung,#1]{\texttt{#2}}}
```

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

```
1897 \def\liBedingungWahr#1{\node[bedingung,#1]{\texttt{[true]}}}
```

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

```
1898 \def\liBedingungFalsch#1{\node[bedingung,#1]{\texttt{[false]}}}
```

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

```
1899 \def\liKontrollCode#1{{\tiny\texttt{\textcolor{gray}{#1}}}}
```

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

```
1900 \def\liKontrollTextzeileKnoten#1{\raisebox{-2pt}{\tikz[scale=0.5,transform shape] \node[draw
```

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

```

1901 \ExplSyntaxOn
1902 \NewDocumentCommand { \liKontrollKnotenPfad } { m }
1903 {
1904   \seq_set_split:Nnn \l_tmpa_seq { - } { #1 }
1905   \seq_set_map:Nnn \l_tmpa_seq \l_tmpa_seq {\liKontrollTextzeileKnoten{##1}}
1906   \seq_use:Nn \l_tmpa_seq { ~~~~ }
1907 }
1908 \ExplSyntaxOff
1909

```

2.23 kopf-fusszeilen.sty

```
1910 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1911 \ProvidesPackage{lehramt-informatik-kopf-fusszeilen}[2021/08/20 Kopf-
1912 und Fußzeilen mit Hilfe des Pakets fancyhdr.]

1913 \ExplSyntaxOn

1914 \fancyhead{}
1915 \fancyhead[R0,LE]{\scriptsize\LehramtInformatikTitel}
1916 \fancyhead[L0,RE]{\scriptsize\today}
1917 \fancyfoot{}
1918 \fancyfoot[LE,R0]{\thepage}
1919 \fancyfoot[L0,CE]{\scriptsize\LehramtInformatikAutorName}
1920 \fancyfoot[CO,RE]{\scriptsize\LehramtInformatikAutorEmail}
1921 \renewcommand{\headrulewidth}{0.4pt}
1922 \renewcommand{\footrulewidth}{0.4pt}
1923 \setlength{\headwidth}{\textwidth}

1924 \ExplSyntaxOff

1925
```

2.24 literatur-dummy.sty

```
1926 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1927 \ProvidesPackage{lehramt-informatik-literatur-dummy}[2020/11/27]

\literatur

1928 \def\literatur{}

\footcite

1929 % \RequirePackage[stable,multiple]{footmisc}
1930 \NewDocumentCommand{ \footcite } { o m }{}

1931
```

2.25 literatur.sty

```
1932 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1933 \ProvidesPackage{lehramt-informatik-literatur}[2020/11/27]

1934 \RequirePackage{csquotes}
1935 \RequirePackage[
1936   bibencoding=utf8,
1937   citestyle=authortitle,
1938   backend=biber,
1939 ]{biblatex}
1940 \addbibresource{\LehramtInformatikRepository/Quellen/10_DB.bib}
1941 \addbibresource{\LehramtInformatikRepository/Quellen/20_OOMUP.bib}
1942 \addbibresource{\LehramtInformatikRepository/Quellen/30_AUD.bib}
1943 \addbibresource{\LehramtInformatikRepository/Quellen/40_SOSY.bib}
1944 \addbibresource{\LehramtInformatikRepository/Quellen/60_FUMUP.bib}
1945 \addbibresource{\LehramtInformatikRepository/Quellen/70_THEO.bib}
1946 \addbibresource{\LehramtInformatikRepository/Quellen/80_DDI.bib}
1947 \addbibresource{\LehramtInformatikRepository/Quellen/Allgemein.bib}
1948 \addbibresource{\LehramtInformatikRepository/Quellen/Examen.bib}
1949 \addbibresource{\LehramtInformatikRepository/Quellen/Wikipedia.bib}
1950 % To allow footnotes in the heading
1951 \RequirePackage[stable,multiple]{footmisc}

\literatur

1952 \def\literatur{\printbibliography[heading=subbibliography]}

1953
```

2.26 makros.sty

```
1954 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
1955 \ProvidesPackage{lehramt-informatik-makros}[2020/11/27 Eine Sammlung von
1956 häufig verwendeten Makros und Umgebungen, die thematisch zu keinem
1957 anderen Paket passen]
1958 \RequirePackage{hyperref}
1959 \RequirePackage{graphicx}
    Für die Umgebung liQuellen benötigt.
1960 \RequirePackage{paralist}
1961 \ExplSyntaxOn

\inhaltsverzeichnis
1962 \def\inhaltsverzeichnis {
1963   \begin{mdframed}
1964     \begin{group}
1965       \let\clearpage\relax
1966       \tableofcontents
1967     \end{group}
1968   \end{mdframed}
1969 }

\memph \mephm (\marginpar and \emph)
1970 \newcommand{\memph}[1]{\emph{#1}\marginpar{\tiny#1}}

\SLASH
1971 \newcommand\SLASH{\char`\\}

\liPseudoUeberschrift Text, der sich wie eine Überschrift verhält.
1972 \newcommand{\liPseudoUeberschrift}[1]{
1973   \bigskip
1974
1975   \par
1976   \noindent
1977   \textbf{#1}
1978
1979   \medskip
1980
1981   \par
1982   % Keine Einrückung
1983   \@afterindentfalse
1984   \@afterheading
1985 }

\liBeschriftung Ähnlich dem Makro \liPseudoUeberschrift{}. Am Ende des Textes wird ein Doppelpunktzeichen angehängt.
1986 \newcommand{\liBeschriftung}[1]{
1987   \par
1988   \noindent
1989   \medskip
1990   \textbf{#1}:
1991   \medskip
1992   \noindent
1993 }

\hinweis
1994 \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.
1995 \NewDocumentEnvironment { liProjektSprache }{ o +b } {} {}
```

liEinbettung

```
1996 \NewDocumentEnvironment { liEinbettung }{ o +b } {#2} {}
```

2.26.1 Umgebungen, die Inhalte aus- und einblenden können

Die einfachste Möglichkeit, um den kompletten Inhalt einer Umgebung auszublenden ist die Verwendung `+b` in einer `xparse` erzeugten Umgebung. Manchmal funktioniert diese Methode nicht. In der Dokumentation von `xparse` steht, dass dieses Feature etwas experimentell ist.

Eine andere Methode verwendet das `exam`-Package. Die Inhalt wird in eine Box verschoben, die dann einfach ignoriert wird.

```
\NewDocumentEnvironment{ liAdditum } { o }
{
  \ifLiADDITUM
  \else
    % Alles in eine Box verschieben und die dann ignorieren.
    \setbox 0 \vbox
    \bgroup
    \fi

    \begin{frame}
  } {
    \end{frame}

    \ifLiADDITUM
    \else
      \egroup
    \fi
}
```

liAntwort Zum Setzen von Antworten. Sie werden mit einem Kasten umgeben. Könnten eventuell ausgeblendet werden, wenn man nur die Lösung sehen will.

```
1997 \NewDocumentEnvironment{ liAntwort } { O{standard} }
1998 {
1999   \ifLiANTWORT
2000   \else
2001     \setbox 0 \vbox
2002     \bgroup
2003     \fi
2004
2005   \str_case:nn {#1} {
2006     {standard} {
2007       \def\beschriftung{}
2008       \mdfsetup{backgroundcolor=gray!1}
2009     }
2010     {richtig} {
2011       \def\beschriftung{richtig}
2012       \mdfsetup{backgroundcolor=green!5,linecolor=green}
2013     }
2014     {falsch} {
2015       \def\beschriftung{falsch}
2016       \mdfsetup{backgroundcolor=red!3,linecolor=red}
2017     }
2018     {muster} {
2019       \def\beschriftung{Musterlösung}
2020       \mdfsetup{backgroundcolor=green!12,linecolor=green}
2021     }
2022   }
2023   \ifx\beschriftung\empty\else
2024     \noindent
2025     \textbf{\beschriftung{}}
  }
```

```

2026 \fi
2027 \begin{mdframed}[
2028   frametitle={\scriptsize\mdseries Lösungshinweise},
2029   innertopmargin=6pt,
2030   frametitleaboveskip=-12pt,
2031   frametitlealignment=\raggedright
2032 ]
2033 }
2034 {
2035   \end{mdframed}
2036   \ifLiANTWORT
2037   \else
2038     \egroup
2039   \fi
2040 }

```

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

```

2041 \NewDocumentEnvironment{ liAdditum } { o }
2042 {
2043   \ifLiADDITUM
2044   \else
2045     \setbox 0 \vbox
2046     \bgroup
2047   \fi
2048
2049   \begin{mdframed}[backgroundcolor=yellow!5]
2050     \IfNoValueTF {#1}
2051     {
2052       \liPseudoUeberschrift{Additum}
2053     }
2054     {
2055       \liPseudoUeberschrift{Additum:~#1}
2056     }
2057   }
2058 {
2059   \end{mdframed}
2060
2061   \ifLiADDITUM
2062   \else
2063     \egroup
2064   \fi
2065 }

```

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.

```

2066 \NewDocumentEnvironment{ liExkurs } { o +b } {
2067   \ifLiEXKURS
2068     \vspace{0.2cm}%
2069     \begin{mdframed}[
2070       backgroundcolor=white,
2071       bottomline=false,
2072       innermargin=1cm,
2073       leftline=true,

```

```

2074     linecolor=black,
2075     linewidth=0.1cm,
2076     outermargin=1cm,
2077     rightline=false,
2078     topline=false,
2079 ]
2080 \footnotesize
2081 \noindent%
2082 \textbf{Exkurs:~#1}\par%
2083 \noindent%
2084 #2
2085 \end{mdframed}
2086 \vspace{0.2cm}
2087 \else
2088 \fi
2089 }{}

```

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

```

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

```

Weiterführende Literatur:

- Quelle 1
- Quelle 2

```

2090 \cs_new:Npn \listen@punkt #1 {\item #1}
2091 \NewDocumentEnvironment { liQuellen }{ +b }
2092 {
2093   \seq_clear_new:N \l_quellen
2094   \seq_set_split:Nnn \l_quellen {\item} {#1}
2095   \seq_remove_all:Nn \l_quellen {}
2096   \begin{mdframed}[backgroundcolor=white!0]
2097     \footnotesize
2098     \noindent
2099     \textsf{\textbf{Weiterführende~Literatur:}}
2100     \medskip
2101     \begin{compactitem}
2102       \seq_map_function:NN \l_quellen {\listen@punkt}
2103     \end{compactitem}
2104   \end{mdframed}
2105   %
2106   \par
2107   \@afterindentfalse
2108   \@afterheading
2109 } {}

```

liLernkartei

```

2110 \NewDocumentEnvironment { liLernkartei }{ m +b }
2111 {
2112   \begin{mdframed}
2113     \footnotesize
2114     \noindent%
2115     \textbf{Lernkarteikarte:~#1}\par%
2116     \noindent%
2117     #2
2118   \end{mdframed}
2119 } {}

```

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


```

2120 \NewDocumentEnvironment { liDiagramm }{ m +b }
2121 {
2122   \begin{mdframed}[backgroundcolor=white!0]
2123   \small
2124   \noindent%
2125   \textit{#1}:
2126   \begin{center}
2127     #2
2128   \medskip
2129   \end{center}
2130   \end{mdframed}
2131 } {}

\liFussnoteUrl \liFussnoteUrl[\langle zusätzlicher-text \rangle]{\langle url \rangle} \liFussnoteUrl[zusätzlicher Text]{url}:
Eine HTTP-URL als Fußnote setzen.
2132 \NewDocumentCommand{\liFussnoteUrl} { o m } {
2133   \footnote{\url{#2}\IfNoValueTF{#1}{}{ ( #1 )}}
2134 }
2135

\liFussnoteLink \liFussnoteLink[\langle zusätzlicher-text \rangle]{\langle link-text \rangle}{\langle url \rangle} \liFussnoteLink[zusätzlicher Text]{text}
Einen Link, d. h. einen Link-Text und eine URL als Fußnote setzen.
2136 \NewDocumentCommand{\liFussnoteLink} { o m m } {
2137   \footnote{\href{#3}{#2}\IfNoValueTF{#1}{}{ ( #1 )}}
2138 }

\zB
2139 \def\zB{z.\.,B. }

\ZB
2140 \def\ZB{Z.\.,B. }

\dh
2141 \def\dh{d.\.,h. }

2142 \ExplSyntaxOff
2143

```

2.27 master-theorem.sty

2144 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2145 \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}

2146 \ExplSyntaxOn

2147 \RequirePackage{amsmath}

\liRundeKlammer

2148 \def\liRundeKlammer#1{

2149 \negthinspace \left(#1 \right)

2150 }

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

2151 \def\liThetaOhneMathe#1{

2152 \Theta \liRundeKlammer{#1}

2153 }

2154 \def\liTheta#1{

2155 \ifmmode

2156 \liThetaOhneMathe{#1}

2157 \else

2158 \$\liThetaOhneMathe{#1}\$

2159 \fi

2160 }

```

\liOmega \liOmega{n^2}:  $\Omega(n^2)$ 
2161 \def\liOmegaOhneMathe#1{
2162   \Omega \liRundeKlammer{#1}
2163 }
2164 \def\liOmega#1{
2165   \ifmmode
2166     \liOmegaOhneMathe{#1}
2167   \else
2168     $\liOmegaOhneMathe{#1}$
2169   \fi
2170 }

\liO \liO{n^2}:  $\mathcal{O}(n^2)$ 
2171 \def\liOOhneMathe#1{
2172   \mathcal{O} \liRundeKlammer{#1}
2173 }
2174 \def\liO#1{
2175   \ifmmode
2176     \liOOhneMathe{#1}
2177   \else
2178     $\liOOhneMathe{#1}$
2179   \fi
2180 }

\liT Let-Abkürzung: \let\T=\liT
      \liT{16}{2}:  $16 \cdot T\left(\frac{n}{2}\right)$  \liT{}{2}:  $T\left(\frac{n}{2}\right)$ 
2181 \def\liTOhneMathe#1#2{
2182   \tl_if_blank:nTF {#1}
2183   {}
2184   {#1 \cdot }
2185   T
2186   \liRundeKlammer{\frac{n}{#2}}
2187 }
2188 \def\liT#1#2{
2189   \ifmmode
2190     \liTOhneMathe{#1}{#2}
2191   \else
2192     $\liTOhneMathe{#1}{#2}$
2193   \fi
2194 }

\liRekursionsGleichung \liRekursionsGleichung:  $T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$ 
2195 \def\liRekursionsGleichung{
2196   $T(n) = \liT{a}{b} + f(n)$
2197 }

\liBedingungEins \liBedingungEins:  $f(n) \in \mathcal{O}(n^{\log_b a - \varepsilon})$ 
2198 \def\liBedingungEins{
2199   $f(n) \in \liO{n^{\log\sb{b}a - \varepsilon}}$
2200 }

\liBedingungZwei \liBedingungZwei:  $f(n) \in \Theta(n^{\log_b a})$ 
2201 \def\liBedingungZwei{
2202   $f(n) \in \liTheta{n^{\log\sb{b}a}}$
2203 }

\liBedingungDrei \liBedingungDrei:  $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ 
2204 \def\liBedingungDrei{
2205   $f(n) \in \liOmega{n^{\log\sb{b}a + \varepsilon}}$
2206 }

2207 \ExplSyntaxOff

```

\liMasterVariablen

```

2208 \def\liMasterVariablen{
2209   \begin{displaymath}
2210     T(n) = \liT{a}{b} + f(n)
2211   \end{displaymath}
2212
2213   \begin{itemize}
2214     \item[$a = $]
2215       Anzahl der rekursiven Aufrufe, Anzahl der Unterprobleme in der
2216       Rekursion
2217       ($a \geq 1$).
2218
2219     \item[$\textstyle{\frac{1}{b}} = $]
2220       Teil des Originalproblems, welches wiederum durch alle Unterprobleme
2221       repräsentiert wird, Anteil an der Verkleinerung des Problems ($b > 1$).
2222
2223     \item[$f(n) = $]
2224       Kosten (Aufwand, Nebenkosten), die durch die Division des Problems und
2225       die Kombination der Teillösungen entstehen. Eine von $T(n)$
2226       unabhängige und nicht negative Funktion.
2227   \end{itemize}
2228   \footcite{wiki:master-theorem}
2229   \footcite[Seite 19-35]{aud:fs:2}
2230 }
```

\liMasterFaelle

```

2231 \def\liMasterFaelle{
2232   \begin{description}
2233     \item[1. Fall:]
2234       $T(n) \in \liTheta{n^{\log\sb{b}a}}$
2235
2236       \hfill falls \liBedingungEins
2237       für $\varepsilon > 0$
2238
2239     \item[2. Fall:]
2240       $T(n) \in \liTheta{n^{\log\sb{b}a} \cdot \log n}$
2241
2242       \hfill falls \liBedingungZwei
2243
2244     \item[3. Fall:]
2245       $T(n) \in \liTheta{f(n)}$
2246
2247       \hfill falls \liBedingungDrei
2248       für $\varepsilon > 0$
2249       und ebenfalls für ein $c$ mit $0 < c < 1$ und alle hinreichend großen $n$
2250       gilt:
2251       $a \cdot f(\textstyle{\frac{n}{b}}) \leq c \cdot f(n)$
2252   \end{description}
2253 }
```

\liMasterVariablenDeklaration

```

2254 \def\liMasterVariablenDeklaration#1#2#3{
2255   \begin{description}
2256     \item[Allgemeine Rekursionsgleichung:] \strut
2257
2258     \liRekursionsGleichung
2259
2260     \item[Anzahl der rekursiven Aufrufe ($a$):] \strut
2261
2262     #1
2263
2264     \item[Anteil Verkleinerung des Problems ($b$):] \strut
2265   \end{description}
}
```

```

2266     um  $\frac{1}{b}$  also  $b = \frac{1}{a}$ 
2267
2268     \item[Laufzeit der rekursiven Funktion ( $f(n)$ ):] \strut
2269
2270      $T(n) = \frac{1}{b} + T(\frac{n}{b})$ 
2271
2272     \item[Ergibt folgende Rekursionsgleichung:] \strut
2273
2274      $T(n) = \frac{1}{b} + T(\frac{n}{b})$ 
2275 \end{description}
2276 }

\liMasterFallRechnung
2277 \def\liMasterFallRechnung#1#2#3{
2278   \begin{description}
2279     \item[1. Fall:] \liBedingungEins:
2280
2281     #1
2282
2283     \item[2. Fall:] \liBedingungZwei:
2284
2285     #2
2286
2287     \item[3. Fall:] \liBedingungDrei:
2288
2289     #3
2290   \end{description}
2291 }

\liMasterExkurs
2292 \def\liMasterExkurs{
2293   \begin{liExkurs}[Master-Theorem]
2294     \liMasterVariablen
2295
2296     \noindent
2297     Dann gilt:
2298
2299     \liMasterFaelle
2300   \end{liExkurs}
2301 }

\liMasterWolframLink Link zu Wolframalpha (+ durch Prozent 2B ersetzen)
2302 \def\liMasterWolframLink#1{
2303   Berechne die Rekursionsgleichung auf WolframAlpha:
2304   \href{https://www.wolframalpha.com/input/?i=#1}{WolframAlpha}
2305 }

2306

```

2.28 mathe.sty

```
2307 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2308 \ProvidesPackage{lehramt-informatik-mathe}[2020/06/10]
2309
2310 % for example \ltimes \rtimes
2311 %\RequirePackage{amssymb}
2312 \RequirePackage{amsmath}
2313
2314 %%
2315 % \mlq \mrq
2316 %%
2317 \DeclareMathSymbol{\mlq}{\mathord}{operators}{``}
2318 \DeclareMathSymbol{\mrq}{\mathord}{operators}{`}
2319
```

2.29 meta.sty

```
2320 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2321 \ProvidesPackage{lehramt-informatik-klassen-meta}[2021/09/10 Sammlung
2322 von Textschnipseln, die das Projekt beschreiben]
```

```
2323 \ExplSyntaxOn
```

```
2324 \liLadePakete{grafik}
```

2.29.1 Einfache Makros (Low level)

```
\liMetaBschlangaulSammlung
```

```
2325 \def\liMetaBschlangaulSammlung{
2326   Die~Bschlangaul-Sammlung
2327 }
```

```
HermineBschlangaulAndFriends
```

```
2328 \def\liMetaHermineBschlangaulAndFriends{
2329   Hermine~Bschlangaul~and~Friends
2330 }
```

```
\liMetaUeberDasProjekt
```

```
2331 \def\liMetaUeberDasProjekt{
2332   Eine~freie~Aufgabensammlung~mit~Lösungen~
2333   von~Studierenden~für~Studierende~
2334   zur~Vorbereitung~auf~die~1.~Staatsexamensprüfungen~
2335   des~Lehramts~Informatik~in~Bayern.
2336 }
```

```
\liMetaCCLink
```

```
2337 \def\liMetaCCLink{
2338   Diese~Materialsammlung~unterliegt~den~Bestimmungen~der~
2339   \href{
2340     https://creativecommons.org/licenses/by-nc-sa/4.0/deed.de
2341   }
2342   {
2343     Creative~Commons~Namensnennung~Nicht~kommerziell~Share~Alike~4.0~
2344     International~Lizenz
2345   }.
2346 }
```

```
\liMetaEmailLink
```

```
2347 \def\liMetaEmailLink{
2348   \href{
2349     mailto:hermine.bschlangaul@gmx.net
2350   }{
2351     hermine.bschlangaul@gmx.net
2352   }
2353 }
```

```
\liMetaHilfMit
```

```
2354 \def\liMetaHilfMit{
2355   Hilf~mit!~
2356
2357   Die~Hermine~schafft~das~nicht~alleine!~
2358
2359   Das~ist~ein~Community-Projekt.~
2360
2361   Verbesserungsvorschläge,~Fehlerkorrekturen,~weitere~Lösungen~sind~
2362   herzlich~willkommen~~~egal~wie~~~per~Pull-Request~oder~per~E-Mail~an~
2363   \liMetaEmailLink.
2364 }
```

`\liMetaHilfMit`

```
2365 \def\liMetaQuelltext{
2366   Der~\TeX-Quelltext~dieses~Dokuments~kann~unter~folgender~
2367   URL~aufgerufen~werden:~
2368 }
```

2.29.2 Zusammengesetzte Makros (High level)

Plaziert zwei Minipages nebeneinander. Die erste Umgebung ist für eine Logo gedacht, die zweite für einen Text

```
2369 \cs_new:Npn \logo_dann_text:nn #1 #2 {
2370   \begin{center}
2371     \begin{minipage}[c]{5.5cm}
2372       #1
2373     \end{minipage}
2374
2375     \begin{minipage}[c]{10cm}
2376       #2
2377     \end{minipage}
2378   \end{center}
2379 }
```

`\liLogoTextProjekt`

```
2380 \def\liLogoTextProjekt
2381 {
2382   \logo_dann_text:nn
2383   {
2384     \liGrafikLogo[width=5cm]
2385   }
2386   {
2387     {
2388       \bfseries
2389       \liMetaBschlangaulSammlung
2390     }
2391     \par
2392
2393     \liMetaHermineBschlangaulAndFriends
2394     \par
2395
2396     \medskip
2397
2398     \begin{spacing}{1}
2399       \footnotesize
2400       \liMetaUeberDasProjekt
2401     \end{spacing}
2402   }
2403 }
```

`\liLogoTextCCLizenz`

```
2404 \def\liLogoTextCCLizenz
2405 {
2406   \logo_dann_text:nn
2407   {
2408     \centerline{\liGrafikCCLizenz[width=3cm]}
2409   }
2410   {
2411     \begin{spacing}{1}
2412       \scriptsize
2413       \liMetaCCLink
2414     \end{spacing}
2415   }
2416 }
```


2417 \ExplSyntaxOff

2418

2.30 minimierung.sty

```

2419 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2420 \ProvidesPackage{lehramt-informatik-mathe}[2021/03/13 Für den
2421 Minimierungsalgorithmus von einem NEA zu einem DEA]

2422 \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

2423 \def\liFussnote#1{\$x_{#1}\$}

2424 \def\li@fussnote@text#1#2{
2425 \liFussnote{#1}
2426 \quad
2427 {\footnotesize #2}
2428 }

\liFussnoteEinsText

2429 \def\liFussnoteEinsText{
2430 \li@fussnote@text{1}
2431 {Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.}
2432 }

\liFussnoteZweiText

2433 \def\liFussnoteZweiText{
2434 \li@fussnote@text{2}
2435 {Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt.}
2436 }

\liFussnoteDreiText

2437 \def\liFussnoteDreiText{
2438 \li@fussnote@text{3}

```

```

2439 {In weiteren Iterationen markierte Zustände.}
2440 }

\liFussnoteVierText
2441 \def\liFussnoteVierText{
2442   \li@fussnote@text{4}
2443   {...}
2444 }

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



2445 \def\liFussnoten{
2446   \bigskip
2447
2448   \noindent
2449   \liFussnoteEinsText
2450
2451   \noindent
2452   \liFussnoteZweiText
2453
2454   \noindent
2455   \liFussnoteDreiText
2456
2457   \noindent
2458   \liFussnoteVierText
2459 }

\liLeereZelle \liLeereZelle:  $\emptyset$ 
Let-Abkürzung: \let\l=\liLeereZelle
2460 \def\liLeereZelle{\$\emptyset\$}

\liZustandsPaarVariablenName
2461 \def\liZustandsPaarVariablenName{z}

\liZustandsPaar
2462 \def\liZustandsPaar#1#2{
2463   $(
2464     \liZustandsPaarVariablenName_#1,
2465     \liZustandsPaarVariablenName_#2
2466   )$
2467 }

liUebergangsTabelle
2468 \renewcommand{\arraystretch}{1.4}
2469 \NewDocumentEnvironment{ liUebergangsTabelle } { m m } {
2470   \liPseudoUeberschrift{Übergangstabelle}
2471   \begin{center}
2472     \begin{tabular}{r|l|l}
2473       \textbf{Zustandspaar} & \textbf{\#1} & \textbf{\#2} \\ \hline
2474     \end{tabular}
2475   \end{center}
2476 }

liUeberschriftDreiecksTabelle \liUeberschriftDreiecksTabelle:

Minimierungstabelle (Table filling)
2478 \ExplSyntaxOn

```

```

2479 \def\liUeberschriftDreiecksTabelle{
2480   \liPseudoUeberschrift{Minimierungstabelle~(Table~filling)}
2481 }

```

\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. —

```

2482 \def\liMinimierungErklaerung{
2483   %\footcite[Seite~19]{koenig}
2484   \liParagraphMitLinien{
2485     Der~Minimierungs-Algorithmus~(auch~Table-Filling-Algorithmus-genannt)~
2486     trägt~in~seinem~Verlauf~eine~Markierung~in~alle~
2487     diejenigen~Zellen~der~Tabelle~ein,~die~zueinander~nicht~äquivalente~
2488     Zustände~bezeichnen.~Die~Markierung~„ $x_{\sb{n}}$ “~in~einer~Tabellenzelle~( $i$ ,~
2489      $j$ )~bedeutet~dabei,~dass~das~Zustandspaar~( $i$ ,~ $j$ )~in~der~ $k$ -ten~
2490     Iteration~des~Algorithmus~markiert~wurde~und~die~Zustände~ $i$ ~und~ $j$ ~
2491     somit~zueinander~( $k-1$ )-äquivalent,~aber~nicht~ $k$ -äquivalent~und~
2492     somit~insbesondere~nicht~äquivalent~sind.~Bleibt~eine~Zelle~bis~zum~Ende~
2493     unmarkiert,~sind~die~entsprechenden~Zustände~zueinander~äquivalent.
2494   }
2495 }
2496 \ExplSyntaxOff

```

2497

2.31 normalformen.sty

```
2498 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2499 \ProvidesPackage{lehramt-informatik-normalformen}[2020/12/10
2500 Hilfsmakros zum Setzen von Normalformen, Funktionalen Abhängigkeiten,
2501 Attributhülle]
```

Mathe für textit tex etc, typographie für geschweifte Klammer

```
2502 \liLadePakete{mathe,typographie}
```

```
2503 \directlua{
2504   helfer = require('lehramt-informatik-helfer')
2505   normalformen = require('lehramt-informatik-normalformen')
2506 }
```

2.31.1 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
```

```
2507 \def\liTeilen#1{
2508   \directlua{tex.print(normalformen.teilen('#1'))}
2509 }
```

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

```
2510 \def\liAttributHuelleOhneMathe#1{\text{AttrHülle}(\#1)}
2511 \def\liAttributHuelle#1{
2512   \ifmmode
2513     \liAttributHuelleOhneMathe{\#1}
2514   \else
2515     $\liAttributHuelleOhneMathe{\#1}$
2516   \fi
2517 }
```

\liAttributMenge Let-Abkürzung: `\let\m=\liAttributMenge`
2518 `\def\liAttributMenge#1{\{ \textit{\#1} \}}`

liAHuelle

```
2519 \NewDocumentEnvironment{ liAHuelle } { +b } {
2520   \begingroup
2521   \footnotesize
2522   \begin{multline*}
2523     \#1
2524   \end{multline*}
2525   \endgroup
2526 } { }
```

\liLinksReduktion Nur innerhalb von liAHuelle zu verwenden bzw. multline

Let-Abkürzung: `\let\ahL=\liLinksReduktion`
`\ahL{ursprüngliche linke Attributmenge}{ohne dieses Attribut}{Ergebnis}`
2527 `\def\liLinksReduktion#1#2#3{`
2528 `\shoveleft{`
2529 `\liAttributHuelleOhneMathe{FA,`
2530 `\liAttributMenge{\#1 \string\ #2}} =`
2531 `} \\\`

```

2532 \shoveright{
2533   \liAttributMenge{#3}
2534 } \\\
2535 }

```

```

\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( $F, \{A, E \setminus E\}$ ) =  $\{A, E, F, B, D\}$ 
2536 \def\liLinksReduktionInline#1#2#3{%
2537   {%
2538     \footnotesize%
2539     $\liAttributHuelleOhneMathe{F,
2540       \liAttributMenge{#1} \string\ #2}} =
2541     \liAttributMenge{#3}$
2542   }
2543 }

```

```

\liLinksReduktionInline Let-Abkürzung: \let\ahr=\liLinksReduktionInline
    \ahr{gelöschte FA}{neue FA ohne rechts Attribut}{gegebene Attribute}{Ergebnis}
2544 \def\liRechtsReduktionInline#1#2#3#4{%
2545   {%
2546     \footnotesize%
2547     $\liAttributHuelleOhneMathe{
2548       F \setminus
2549       \liFunktionaleAbhaengigkeit{#1}
2550       \def\tmp{#2}\ifx\tmp\empty
2551       \else
2552         \cup \liFunktionaleAbhaengigkeit{#2}
2553       \fi
2554       ,
2555       \liAttributMenge{#3}
2556     } =
2557     \liAttributMenge{#4}$
2558   }
2559 }

```

```

\liFunktionaleAbhaengigkeit Let-Abkürzung: \let\fa=\liFunktionaleAbhaengigkeit
    \fa{A, B -> C, D}:  $\{A, B\} \rightarrow \{C, D\}$  Regulärer Ausdruck zum Konvertieren:
    \$(.*) \rightarrow (.*)\$ \fa{$1 -> $2}
2560 \def\liFunktionaleAbhaengigkeit#1{%
2561   \directlua{normalformen.drucke_funk_abhaengigkeit('#1', false)}%
2562 }

```

```

\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 (.*)\$ \$(.*) \rightarrow
    $1 -> $2;

```

```

2563 \NewDocumentCommand {\liFunktionaleAbhaengigkeiten} { O{FA} m } {
2564   \liGeschweifteKlammern
2565   {#1}
2566   {
2567     \begin{align*}
2568       \directlua{normalformen.drucke_funk_abhaengigkeiten('#2')}
2569     \end{align*}
2570   }
2571   {-0.5cm}
2572   {-1.7cm}
2573 }

\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}

2574 \NewDocumentCommand {\liRelation} { O{R} m } {
2575   $\directlua{
2576     local name = helper.konvertiere_tiefgestellt('#1')
2577     tex.print(name)
2578   }$(\textit{\, #2\,})
2579 }

2580

```

2.32 o-notation.sty

2581 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2582 \ProvidesPackage{lehramt-informatik-o-notation}[2021/09/08]

2.32.1 Makro-Kürzel

\let\O=\liONotationO

2.32.2 TeX-Markup-Beispiel: Funktionsgraphen mit pgfplots

```
\begin{tikzpicture}
  \begin{axis}[
    xlabel=$n$,
    legend entries={\f a, \f b, \f c, \f d, \f e},
    ymax=500,
    xmin=0,
    xmax=7,
    legend pos=north west,
    domain=0:7
  ]
    \addplot{sqrt(x^5) + (4 * x) - 5};
    \addplot{log2(log2(x))};
    \addplot{2^x};
    \addplot{x^2 * log10(x) + (2 * x)};
    \addplot{4^x / (log2(x))};
  \end{axis}
\end{tikzpicture}
```

2583 \ExplSyntaxOn

2584 \RequirePackage{amssymb}

2585 \RequirePackage{pgfplots}

Für echte Teilmenge \subsetneq: \subsetneq

2586 \RequirePackage{amssymb}

\liRundeKlammer

```
2587 \def\liRundeKlammer#1{
2588   \negthinspace \left( #1 \right)
2589 }
```

\liONotationO $\mathcal{O}(n^2)$

```
2590 \cs_new:Npn \o_notation_O:n #1 {
2591   \mathcal{O} \liRundeKlammer{#1}
2592 }
2593 \def\liONotationO#1{
2594   \ifmmode
2595     \o_notation_O:n { #1 }
2596   \else
2597     $ \o_notation_O:n { #1 } $
2598   \fi
2599 }
```

2600

2.33 petri.sty

2601 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

2602 \ProvidesPackage{lehramt-informatik-petri}[2020/12/03]

2.33.1 Makro-Kürzel

\let\t=\liPetriTransitionsName

\let\tp=\liPetriTransPfeile

\let\k=\liPetriErreichKnotenDrei

2603 \RequirePackage{tikz}

2604 \usetikzlibrary{petri,arrows.meta}

Für die Darstellungsmatrix

2605 \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

```
2606 \def\liPetriSetzeSchluessel{%
2607   \def\TmpTransitionOne{}%
2608   \def\TmpTransitionTwo{}%
2609   \def\TmpTransitionThree{}%
2610   \def\TmpTransitionFour{}%
2611   \def\TmpTransitionFive{}%
2612   \def\TmpTransitionSix{}%
2613   \def\TmpTransitionSeven{}%
2614   \def\TmpTransitionEight{}%
2615   \def\TmpTransitionNine{}%
2616   \def\TmpTransitionTen{}%
2617   \pgfkeys{/petri/.cd,
2618     p1/.store in=\TmpPlaceOne,p1/.default=0,p1,
2619     p2/.store in=\TmpPlaceTwo,p2/.default=0,p2,
2620     p3/.store in=\TmpPlaceThree,p3/.default=0,p3,
2621     p4/.store in=\TmpPlaceFour,p4/.default=0,p4,
```

```

2622 p5/.store in=\TmpPlaceFive,p5/.default=0,p5,
2623 p6/.store in=\TmpPlaceSix,p6/.default=0,p6,
2624 p7/.store in=\TmpPlaceSeven,p7/.default=0,p7,
2625 p8/.store in=\TmpPlaceEight,p8/.default=0,p8,
2626 p9/.store in=\TmpPlaceNine,p9/.default=0,p9,
2627 p10/.store in=\TmpPlaceTen,p10/.default=0,p10,
2628 t1/.store in=\TmpTransitionOne,t1/.default=activated,
2629 t2/.store in=\TmpTransitionTwo,t2/.default=activated,
2630 t3/.store in=\TmpTransitionThree,t3/.default=activated,
2631 t4/.store in=\TmpTransitionFour,t4/.default=activated,
2632 t5/.store in=\TmpTransitionFive,t5/.default=activated,
2633 t6/.store in=\TmpTransitionSix,t6/.default=activated,
2634 t7/.store in=\TmpTransitionSeven,t7/.default=activated,
2635 t8/.store in=\TmpTransitionEight,t8/.default=activated,
2636 t9/.store in=\TmpTransitionNine,t9/.default=activated,
2637 t10/.store in=\TmpTransitionTen,t10/.default=activated,
2638 scale/.store in=\TmpScale,scale/.default=0.5,
2639 x/.store in=\TmpX,x/.default=5,
2640 y/.store in=\TmpY,y/.default=5,
2641 }%
2642 }

```

```

2643 \tikzset{
2644   li petri/.style={
2645     activated/.style={
2646       very thick
2647     },
2648     inhibitor/.style={
2649       {Circle[open,length=2mm,fill=white]}-
2650     }
2651   }
2652 }

```

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

```

2653 \def\liPetriTransitionsNameOhneMathe#1{t\sb{#1}}
2654 \def\liPetriTransitionsName#1{
2655   \ifmmode
2656     \liPetriTransitionsNameOhneMathe{#1}
2657   \else
2658     $\liPetriTransitionsNameOhneMathe{#1}$
2659   \fi
2660 }

```

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

```

2661 \NewDocumentCommand{ \liPetriErreichTransition } { m m m O{} O{} } {
2662   \draw[->] (#1) edge[#4] node[pos=0.5,auto,sloped,#5]{t\sb{#3}$} (#2);
2663 }

```

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

```

2664 \def\liPetriErreichKnotenDrei#1#2#3{(#1,#2,#3)}

```

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

```

2665 \def\liPetriTransPfeile#1{ $\rightarrow \hspace{0.4cm} \liPetriTransitionsName{#1} \hspace{0.3cm}
2666 }

```

2.34 potenzmengen-konstruktion.sty

```

2667 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2668 \ProvidesPackage{lehramt-informatik-potenzmengen-konstruktion}[2021/02/21 Hilfsmakros
2669 einzusetzen bei der Potenzmengen-Konstruktion (Potenzmengen-Algorithmus)]

2670 \liLadePakete{formale-sprachen}
2671 \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}
    }
  }
}

2672 \def\liZustandsMengenSammlung#1#2{
2673   \liZustandsnameGross{#1}
2674   {
2675     \footnotesize
2676     \liPotenzmenge{
2677       \str_case:nn {#1} #2
2678     }
2679   }
2680 }

\liZustandsMengenSammlungNr

2681 \def\liZustandsMengenSammlungNr#1#2{
2682   \liZustandsnameGross{#1}
2683   {

```

```
2684     \footnotesize
2685     \liZustandsmengeNr{
2686         \str_case:nn {#1} #2
2687     }
2688 }
2689 }

2690 \ExplSyntaxOff
2691
```

2.35 pseudo.sty

```

2692 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2693 \ProvidesPackage{lehramt-informatik-pseudo}[2020/12/30 Paket zum Setzen
2694 von Pseudo-Code, Hüll-Paket um algorithm2e]

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

```

Algorithmus 1: Minimaler Spannbaum nach Kruskal

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

```

2695 \RequirePackage[german,boxruled]{algorithm2e}

```

```

2696

```

2.36 pumping-lemma.sty

2697 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2698 \ProvidesPackage{lehramt-informatik-pumping-lemma}[2021/08/11 Enthält die
 2699 Definitionen für das Pumping-Lemma in der Regulären Sprache und
 2700 in der Kontextfreien Sprache]

\liPumpingRegulaer

```
2701 \def\liPumpingRegulaer{%
2702   Es sei  $L$  eine reguläre Sprache. Dann gibt es eine Zahl  $j$ , sodass für
2703   alle Wörter  $\omega$  in  $L$  mit  $|\omega| \geq j$  (jedes Wort  $\omega$  in
2704    $L$  mit Mindestlänge  $j$ ) jeweils eine Zerlegung  $\omega = uvw$ 
2705   existiert, sodass die folgenden Eigenschaften erfüllt sind:
2706
2707   \begin{enumerate}
2708     \item  $|v| \geq 1$ 
2709     (Das Wort  $v$  ist nicht leer.)
2710
2711     \item  $|uv| \leq j$ 
2712     (Die beiden Wörter  $u$  und  $v$  haben zusammen höchstens die Länge  $j$ .)
2713
2714     \item Für alle  $i = 0, 1, 2, \dots$  gilt  $uv^i w$  in  $L$ 
2715     (Für jede natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^i w$  in der
2716     Sprache  $L$ )
2717   \end{enumerate}
2718
2719   Die kleinste Zahl  $j$ , die diese Eigenschaften erfüllt, wird
2720   Pumping-Zahl der Sprache  $L$  genannt.\footcite{wiki:pumping-lemma}
2721 }
```

\liPumpingKontextfrei

```
2722 \def\liPumpingKontextfrei{%
2723   Es sei  $L$  eine kontextfreie Sprache. Dann gibt es eine Zahl  $j$ , sodass
2724   sich alle Wörter  $\omega$  in  $L$  mit  $|\omega| \geq j$  zerlegen lassen in
2725    $\omega = uvwxy$ , sodass die folgenden Eigenschaften erfüllt sind:
2726
2727   \begin{enumerate}
2728     \item  $|vx| \geq 1$ 
2729     (Die Wörter  $v$  und  $x$  sind nicht leer.)
2730
2731     \item  $|vwx| \leq j$ 
2732     (Die Wörter  $v$ ,  $w$  und  $x$  haben zusammen höchstens die Länge  $j$ .)
2733
2734     \item Für alle  $i$  in  $\mathbb{N}_0$  gilt  $uv^iwx^iy$  in  $L$  (Für jede
2735     natürliche Zahl (mit  $0$ )  $i$  ist das Wort  $uv^iwx^iy$  in der
2736     Sprache  $L$ )
2737   \end{enumerate}
2738 }
2739
```

2.37 quicksort.sty

```

2740 % https://tex.stackexchange.com/a/142634
2741 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2742 \ProvidesPackage{lehramt-informatik-quicksort}[2020/06/12]
2743
2744 %-----
2745 % USAGE:
2746 % \QSinitialize{comma, separated, numerical, values}
2747 % \loop
2748 % \QSpivotStep
2749 % \ifnum\value{pivotcount}>0
2750 %   \QSSortStep
2751 % \repeat
2752 %-----
2753
2754 % xintfrac does not load xinttools, this must be done explicitly if needed as here.
2755 \RequirePackage{xintfrac, xinttools}
2756
2757 \RequirePackage{tikz}
2758
2759 %-----
2760 % FIRST PART: TikZ styles and macros for the actual drawing
2761 \newcounter{cellcount}% used for coordinates of the node
2762 \newcounter{pivotcount}% when it will remain at zero, will signal the sort is finished.
2763
2764 % Styles defined by Tom Bombaldi. (modified: all share the same size)
2765 % (re-modified \bf -> \bfseries due to extremely annoying warnings from
2766 % KOMA-script which are truly a pain and do not make any sense regarding \bf:
2767 % if I want to use \bf, and know what I am doing, why should I get HARASSED
2768 % by police of LaTeX good conduct ? )
2769 \tikzset{
2770   l/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=lime!70!gray},
2771   o/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=olive!50},
2772   r/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2773   b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, text=white},
2774   n/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=white, text=magenta},
2775   g/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=gray, text=white, text=white},
2776
2777
2778 % NOTE the b style was originally the same as the r(aised) style apart from
2779 % not being raised, but I find it nicer with a somewhat different
2780 % specification. I have not updated the images though.
2781
2782 % How the nodes are drawn depending on whether on the left of the pivot value
2783 % or on the right, or is a pivot value, or a raised pivot during selection phase.
2784
2785 \def\DecoLEFT #1{%
2786   \xintFor* ##1 in {#1} \do
2787   {\stepcounter{cellcount}\node[o] at (\arabic{cellcount},0) {##1};}%
2788 }
2789
2790 \def\DecoINERT #1{%
2791   \xintFor* ##1 in {#1} \do
2792   {\stepcounter{cellcount}\node[g] at (\arabic{cellcount},0) {##1};}%
2793 }
2794
2795 \def\DecoRIGHT #1{%
2796   \xintFor* ##1 in {#1} \do
2797   {\stepcounter{cellcount}\node[l] at (\arabic{cellcount},0) {##1};}%
2798 }
2799
2800 \def\DecoLEFTwithPivot #1{\stepcounter{pivotcount}%
2801   \xintFor* ##1 in {#1} \do

```

```

2802     {\stepcounter{cellcount}}%
2803     \xintifForLast {\node[r]}\node[o]} at (\arabic{cellcount},0) {##1};}%
2804 }
2805
2806 \def\DecoINERTwithPivot #1{\stepcounter{pivotcount}%
2807     \xintFor* ##1 in {#1} \do
2808     {\stepcounter{cellcount}}%
2809     \xintifForLast {\node[b]}\node[g]} at (\arabic{cellcount},0) {##1};}%
2810 }
2811
2812 \def\DecoRIGHTwithPivot #1{\stepcounter{pivotcount}%
2813     \xintFor* ##1 in {#1} \do
2814     {\stepcounter{cellcount}}%
2815     \xintifForLast {\node[r]}\node[l]} at (\arabic{cellcount},0) {##1};}%
2816 }
2817
2818 %-----
2819 % SECOND PART: the actual sorting routines.
2820
2821 \def\QS@sort@a #1{\expandafter \QS@sort@b \expandafter {\xintLength {#1}}{#1}}
2822 \def\QS@sort@b #1{\ifcase #1
2823     \expandafter\QS@sort@empty
2824     \or\expandafter\QS@sort@single
2825     \else\expandafter\QS@sort@c
2826     \fi
2827 }%
2828 \def\QS@sort@empty #1{}
2829 \def\QS@sort@single #1{\QS@Ir {#1}}
2830
2831 % This step is to pick the last as pivot.
2832 \def\QS@sort@c #1%
2833     {\expandafter\QS@sort@d\expandafter {\romannumeral0\xintnthelt {-1}{#1}}{#1}}%
2834
2835 % Here \QSLr, \QSIr, \QSR have been let to \relax.
2836 % The trick with \xintApplyUnbraced is that for example when selecting
2837 % the elements smaller than pivot, if we had been using \xintApply we
2838 % would have had at the minimum an empty brace pair. Thus we use the
2839 % "unbraced" variant, but then the \QS@select@smaller has added in
2840 % anticipation a level of braces.
2841 \def\QS@sort@d #1#2{%
2842     \QSLr {\xintApplyUnbraced {\QS@select@smaller {#1}}{#2}}}%
2843     \QSIr {\xintApplyUnbraced {\QS@select@equal {#1}}{#2}}}%
2844     \QSRr {\xintApplyUnbraced {\QS@select@greater {#1}}{#2}}}%
2845 }%
2846 \def\QS@select@smaller #1#2{\xintifLt {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2847 \def\QS@select@equal #1#2{\xintifEq {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2848 \def\QS@select@greater #1#2{\xintifGt {#2}{#1}{{#2}}{ }}% space will stop a f-
    expansion
2849
2850 %
2851 % NOTE 1: thus, each comparison with the pivot is done three (!) times.
2852 %
2853 % NOTE 2: we may well end up with \QSLr {<empty>} situations. This is handled
2854 % silently by the \xintFor loops, and also when \QSLr becomes \QS@sort@a, the
2855 % latter must handle correctly an empty argument.
2856
2857 %-----
2858 % THIRD PART: the main macros \QSpivotStep, \QSsortStep and \QS@initialize.
2859
2860 % This draws all with suitable highlighting for the newly chosen pivots
2861 % (which will be shown raised)

```



```

2862 \def\QSpivotStep {\let\QSLr\DecoLEFTwithPivot
2863             \let\QSIr\DecoINERT
2864             \let\QSIrr\DecoINERT
2865             \let\QSRr\DecoRIGHTwithPivot
2866 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2867             \setcounter{cellcount}{0}\setcounter{pivotcount}{0}%
2868             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2869 }
2870
2871 % This sorts and then draws, showing where the pivot chosen in the previous
2872 % step go. Next time they will have become "inert". If pivotcount is still at
2873 % zero on exit from \QSpivotStep, then this is the signal to stop before
2874 % executing \QSsortStep.
2875 \def\QSsortStep {\def\QSLr {\noexpand\QS@sort@a}%
2876             \def\QSRr {\noexpand\QS@sort@a}%
2877             \def\QSIr {\noexpand\QSIrr}%
2878             \let\QSIrr\relax
2879             \edef\QS@list{\QS@list}%
2880             \let\QSLr\relax
2881             \let\QSRr\relax
2882             \let\QSIr\relax
2883             \edef\QS@list{\QS@list}%
2884             \let\QSLr\DecoLEFT
2885             \let\QSIr\DecoINERTwithPivot
2886             \let\QSIrr\DecoINERT
2887             \let\QSRr\DecoRIGHT
2888 \par\centerline{\rule[1.5mm]{0pt}{8mm}}%
2889             \setcounter{cellcount}{0}%
2890             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2891 }
2892
2893 \def\QSinitialize #1{%
2894     % first, we convert the comma separated values into a list of braced items
2895     % we use an \edef, and anyhow many \edef's will be used later
2896     \edef\QS@list {\noexpand\QSRr {\xintCSVtoList {#1}}}%
2897     \let\QSRr\DecoRIGHT
2898     % The \QSRr marker mutated to draw the last element as
2899     % pivot and the earlier ones with the suitable style.
2900     %
2901     % The list of marked braced items \QS@list is used both for drawing
2902     % (as here) and for doing the exchange of elements during sort.
2903     \par\centerline{\rule[1.5mm]{0pt}{8mm}\setcounter{cellcount}{0}%
2904             \begin{tikzpicture}\QS@list\end{tikzpicture}}
2905 }
2906

```

2.38 relationale-algebra.sty

2907 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 2908 \ProvidesPackage{lehramt-informatik-relationale-algebra}[2020/12/11]

2909 \RequirePackage{amsmath}

2910 \RequirePackage{amssymb}

Zum Zeichnen von Operatorenbäumen verwenden wir TikZ

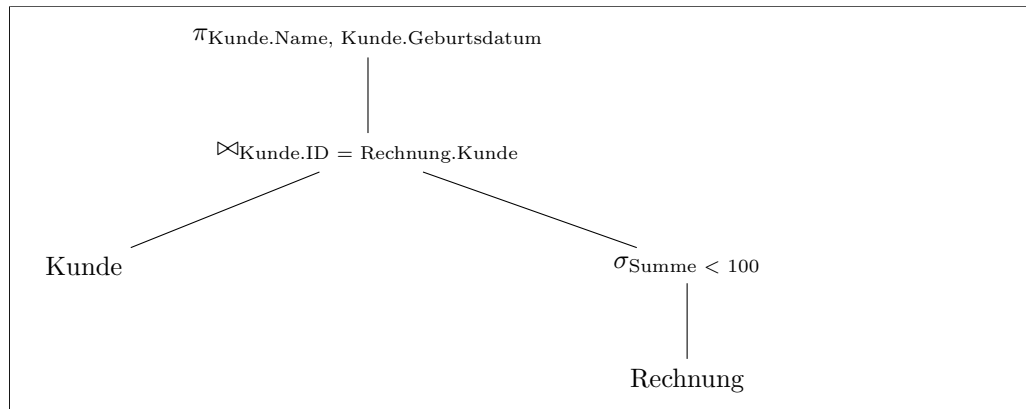
```
\begin{tikzpicture}
  \node
    (pi) {\pi_{\text{Kunde.Name, Kunde.Geburtsdatum}}};

  \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}
```



2911 \RequirePackage{tikz}

2912 \usetikzlibrary{positioning}

Privates Makros, das zwei Querstriche erzeugt.

```
2913 \def\o@join{\setbox0=\hbox{\bowtie}}%
2914 \rule[-.02ex]{.25em}{.4pt}\llap{\rule[\ht0]{.25em}{.4pt}}%
2915 }
```

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

2916 \def\leftouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie}}

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

2917 \def\rightouterjoin{\mathbin{\bowtie\mkern-5.8mu\o@join}}

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

2918 \def\fullouterjoin{\mathbin{\o@join\mkern-5.8mu\bowtie\mkern-5.8mu\o@join}}

2919

2.39 rmodell.sty

```
2920 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2921 \ProvidesPackage{lehramt-informatik-rmodell}[2020/09/01]
2922 Makros und Umgebungen zum Setzen des Relationenmodells beim Thema
2923 Datenbanken.]
2924 \RequirePackage{soul}
```

2.39.1 Makro-Kürzel

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

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

```
2925 \def\liPrimaer#1{\ul{#1}}
```

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

```
2926 \def\liFremd#1{{\setul{-0.9em}{}}\ul{#1}}
```

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

```
2927 \def\li@Rmodell@Schrift{\footnotesize\ttfamily}
2928 \ExplSyntaxOn
2929 \NewDocumentEnvironment { liRmodell }
2930 { +b }
2931 {
2932   \medskip
2933   {
2934     \linespread{2}
2935     \setlength{\parindent}{0pt}
2936     \li@Rmodell@Schrift#1
2937   }
2938   \medskip
2939 } {}
2940 \ExplSyntaxOff
```

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

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

```
2941 \def\liRelationMenge#1#2{
2942 \noindent
2943 #1 : \[ #2 ]\}
2944 \par
2945 }
```

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

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

```
2946 \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}
```

```
2947 \NewDocumentEnvironment { liRelationenSchemaFormat }{ +b } {} {}
```

```
2948
```

2.40 sortieren.sty

```
2949 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
2950 \ProvidesPackage{lehramt-informatik-sortieren}[2020/06/10 Schaubilder
2951 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}
}
```



```
2952 \RequirePackage{tikz}
2953 \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.

```
2954 \def\liVertauschen#1{
2955   \directlua{
2956     local sortieren = require('lehramt-informatik-sortieren')
2957     sortieren('#1')
2958   }
2959 }
```

`\liSortierPfeil`

```
2960 \def\liSortierPfeil#1#2{
2961   \draw[-latex] ([xshift=1mm]reihe.#1 north) -- ++(0,0.25) -| ([xshift=-1mm]reihe.#2 north);
2962 }
```

`\liSortierPfeilUnten`

```
2963 \def\liSortierPfeilUnten#1#2{
2964   \draw[-latex] ([xshift=1mm]reihe.#1 south) -- ++(0,-0.25) -| ([xshift=-1mm]reihe.#2 south);
2965 }
```

`\liSortierMarkierung`

```
2966 \def\liSortierMarkierung#1#2{\node[
2967   draw,
2968   very thick,
2969   fit=(reihe.#1) (reihe.#2),
2970   inner sep=0pt
2971 ] {}
2972 }

2973 \tikzset{
2974   li sortierung zahlenreihe/.style={
2975     draw,
2976     thin,
2977     font=\large,
2978     rectangle split horizontal,
2979     rectangle split,
2980   }
2981 }
```

```

2982 % https://tex.stackexchange.com/a/140895
2983 \RequirePackage{forest,xstring}
2984 \usetikzlibrary{calc}
2985
2986 \makeatletter
2987 \pgfmathdeclarefunction{strrepeat}{2}{%
2988   \begingroup\pgfmathint{#2}\pgfmath@count\pgfmathresult
2989   \let\pgfmathresult\pgfutil@empty
2990   \pgfutil@loop\ifnum\pgfmath@count>0\relax
2991     \expandafter\def\expandafter\pgfmathresult\expandafter{\pgfmathresult#1}%
2992     \advance\pgfmath@count-1\relax
2993   \pgfutil@repeat\pgfmath@smuggleone\pgfmathresult\endgroup}
2994 \makeatother
2995
2996 \def\myNodes{}
2997
2998 \ExplSyntaxOn
2999 \newcommand*\sortList[1]{%
3000   \clist_sort:Nn#1{\int_compare:nNnTF{##1}>{##2}\sort_return_swapped:\sort_return_same:}}
3001 \ExplSyntaxOff
3002
3003 \forestset{
3004   sort/.code={%
3005     \pgfmathparse{level()>\forestSortLevel}%
3006     \ifnum\pgfmathresult=0
3007       \StrSubstitute{\forestov{content}}{ }{,}{\myList}%
3008       \sortList\myList
3009       \StrSubstitute{\myList}{,}{ }{\myList}%
3010       \pgfmathparse{strrepeat("1",level())}%
3011       \xappto\myNodes{\noexpand\node at ({\forestov{name}}|-m)!-1!({\forestov{name}}$)
3012         (m\forestov{name}) {\myList}}%
3013       \pgfmathparse{level()=\forestSortLevel}%
3014       \ifnum\pgfmathresult=1
3015         \forestOget{\forestov{@first}}{name}\forestFirst
3016         \forestOget{\forestov{@last}}{name}\forestLast
3017         \xappto\myNodes{{[<-]edge (\forestOv{\forestov{@first}}{name})
3018           \ifx\forestFirst\forestLast\else edge (\forestOv{\forestov{@last}}{name})\fi}}%
3019       \fi
3020       \ifnum\forestov{@parent}=0\else
3021         \xappto\myNodes{edge (m\forestOv{\forestov{@parent}}{name})}%
3022       \fi
3023       \gappto\myNodes{;}%
3024     \fi}}
3025
3026 \forestset{sort level/.code=%
3027   \pgfmathparse{#1}\let\forestSortLevel\pgfmathresult
3028   \pgfmathparse{strrepeat("1",\forestSortLevel+1)}\let\forestOnes\pgfmathresult}
3029

```

2.41 spalten.sty

```
3030 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3031 \ProvidesPackage{lehramt-informatik-spalten}[2020/12/07 Lädt das Paket
3032 „multicol“, damit mehrspaltiger Satz mit Hilfe der Umgebung „multicols“
3033 realisiert werden kann.]
3034 \RequirePackage{multicol}
```

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

```
3035 \def\liSpaltenUmbruch{\vfill\strut\columnbreak}
```

```
3036
```

2.42 sql.sty

3037 \NeedsTeXFormat{LaTeX2e}[1995/12/01]

3038 \ProvidesPackage{lehramt-informatik-tabelle}[2021/09/02 Zu Setzen von SQL]

2.42.1 Latex-Markup-Beispiel

```
\begin{liAdditum}[Übungsdatenbank]
% Datenbankname: Personalverwaltung
\begin{minted}{sql}
CREATE TABLE Abteilung(
  AbteilungsID INTEGER PRIMARY KEY,
  Bezeichnung VARCHAR(30)
);

CREATE TABLE Mitarbeiter(
  MitarbeiterID INTEGER PRIMARY KEY,
  Vorname VARCHAR(30),
  Vorgesetzter INTEGER REFERENCES Mitarbeiter(MitarbeiterID),
  Telefonnummer VARCHAR(50),
  Gehalt DOUBLE PRECISION
);

INSERT INTO Abteilung VALUES
  (1, 'Buchhaltung');

INSERT INTO Mitarbeiter
  (MitarbeiterID, Vorname, Nachname, Vorgesetzter, AbteilungsID, Telefonnummer, Gehalt)
VALUES
  (1, 'Hans', 'Meier', 11, 4, '023/13432', 2335),
  (2, 'Fred', 'Wolitz', 11, 2, '0233/413432', 1233);
\end{minted}
\index{SQL mit Übungsdatenbank}
\end{liAdditum}
```

3039 \liLadePakete{syntax}

3040 \RequirePackage{fancyvrb}

3041 \DefineVerbatimEnvironment{liSqlErgebnis}{Verbatim}

3042 {fontsize=\footnotesize}

3043

2.43 struktogramm.sty

```
3044 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3045 \ProvidesPackage{lehramt-informatik-struktogramm}[2021/01/31 Lädt das
3046 Paket struktex zum Setzen von Struktogrammen]
3047 \RequirePackage{struktex}
3048
```


2.44 syntax.sty

```

3049 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3050 \ProvidesPackage{lehramt-informatik-syntax}[2020/06/10 Ein Hüll-Paket um
3051 `minted`, das sich um die Syntax-Hervorhebung des Java-Codes kümmert.]
3052 \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.

2.44.1 Makro-Kürzel

```

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

```

```

3053 \ExplSyntaxOn
3054 \directlua{
3055   syntax = require('lehramt-informatik-syntax')
3056   syntax.importiere_konfiguration('tex_repo_lokaler_pfad', '\LehramtInformatikRepository')
3057   syntax.importiere_konfiguration('github_domain', '\LehramtInformatikGithubDomain')
3058   syntax.importiere_konfiguration('github_raw_domain', '\LehramtInformatikGithubRawDomain')
3059   syntax.importiere_konfiguration('github_tex_repo', '\LehramtInformatikGithubTexRepo')
3060   syntax.importiere_konfiguration('github_code_repo', '\LehramtInformatikGithubCodeRepo')
3061   syntax.importiere_konfiguration('git_branch', '\LehramtInformatikGitBranch')
3062 }
3063 \RequirePackage{hyperref}
3064 \RequirePackage{minted}
3065 % pygmentize -L styles
3066 \usemintedstyle{colorful}
3067 %\BeforeBeginEnvironment{minted}{\begin{mdframed}}
3068 %\AfterEndEnvironment{minted}{\end{mdframed}}
3069 %\setminted{breaklines=true,linenos}
3070 \setminted{
3071   breaklines=true,
3072   linenos=false,
3073   fontsize=\footnotesize,
3074 }

```

liJavaAngabe Eine Umgebung für Java-Code, ohne Zeilennummer und etwas eingerückt um den Java-Code in Angaben / Aufgabestellungen zu setzen.

```

3075 \newminted[liJavaAngabe]{java}{
3076   xleftmargin=1cm
3077 }

```

\liJavaCode Im Zeilenfluss einen kurzen Java-Code-Ausschnitt setzen. Es werden automatische Zeilenumbrüche gemacht.

Let-Abkürzung: `\let\j=\liJavaCode`

```

3078 \def\liJavaCode#1{
3079   \,
3080   \textcolor{blue}{
3081     \mintinline[
3082       fontsize=\normalsize,
3083       breakanywhere % https://github.com/gpoore/minted/issues/31#issuecomment-
458640242
3084     ]{java}|#1|
3085   }
3086   \,
3087 }

```

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

```

3088 \def\liLatexCode#1{\mintinline{latex}|#1|}

```

```

3089 \def\li@GithubLink#1#2{
3090   \begin{flushright}
3091     \tiny
3092     Code-Beispiel~auf~Github~ansehen:~
3093     \href{#1}{\nolinkurl{#2}}
3094   \end{flushright}
3095 }

\liJavaDatei Eine komplette Java-Datei einbinden, die Verzeichnis ./Code/src/main/java/org/bschlangaul
liegt.
3096 \NewDocumentCommand{\liJavaDatei}{ O{firstline=3} m }{
3097   \inputminted[#1]{java}{
3098     \directlua{
3099       syntax.drucke_absoluten_pfad('#2', false)
3100     }
3101   }
3102   \li@GithubLink
3103   {\directlua{syntax.drucke_github_url('#2', false)}}
3104   {\directlua{syntax.drucke_relativen_repo_pfad('#2', false)}}
3105 }

\liJavaTestDatei Eine komplette Java-Test-Datei einbinden, die Verzeichnis ./Code/src/test/java/org/bschlangaul
liegt.
3106 \NewDocumentCommand{\liJavaTestDatei}{ O{firstline=3} m }{
3107   \inputminted[#1]{java}{
3108     \directlua{
3109       syntax.drucke_absoluten_pfad('#2', true)
3110     }
3111   }
3112   \li@GithubLink
3113   {\directlua{syntax.drucke_github_url('#2', true)}}
3114   {\directlua{syntax.drucke_relativen_repo_pfad('#2', true)}}
3115 }

\liJavaExamen \liJavaExamen{66116}{2015}{03}{Kunde} Regulärer Ausdruck zum Konvertieren: \liJavaDatei([
\liJavaExamen$1{$2}{$3}{$4}{$5}
3116 \NewDocumentCommand{\liJavaExamen}{ O{firstline=3} m m m m }{
3117   \inputminted[#1]{java}{
3118     \directlua{
3119       syntax.drucke_absoluten_examens_pfad('#2', '#3', '#4', '#5')
3120     }
3121   }
3122   \li@GithubLink
3123   {\directlua{syntax.drucke_github_examens_url('#2', '#3', '#4', '#5')}}
3124   {\directlua{syntax.drucke_relativen_examens_repo_pfad('#2', '#3', '#4', '#5')}}
3125 }

\liAssemblerCode
3127 \def\liAssemblerCode#1{\mintinline{asm}|#1|}

\liAssemblerDatei \liAssemblerDatei{relativer-pfad}: Relativer Pfad zu einer Assembler-Datei.
3128 \NewDocumentCommand{\liAssemblerDatei}{ m }{
3129   \inputminted{asm}{#1}
3130 }

\liMinispracheDatei \liMinispracheDatei{relativer-pfad}: Relativer Pfad zu einer Minisprachen-Datei
(Hochsprache für die Minimaschine von Albert Wiedemann).
3131 \NewDocumentCommand{\liMinispracheDatei}{ m }{
3132   \inputminted{componentpascal}{#1}
3133 }

```

```

\liHaskellCode \liHaskellCode{haskell}: Zum Setzen von Haskell-Code.
3134 \def\liHaskellCode#1{\mintinline{haskell}|#1|}

\liHaskellDatei \liHaskellDatei{relativer-pfad}: Relativer Pfad zu einer Haskell-Datei.
3135 \NewDocumentCommand{\liHaskellDatei}{ m }{
3136   \inputminted{haskell}{#1}
3137 }

3138 \ExplSyntaxOff

\liSqlCode \liHaskellCode{sql}: Zum Setzen von SQL-Code.
Let-Abkürzung: \let\s=\liSqlCode
3139 \def\liSqlCode#1{\mintinline{sql}|#1|}

3140

```

2.45 syntaxbaum.sty

```
3141 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3142 \ProvidesPackage{lehramt-informatik-syntaxbaum}[2021/02/14 Zum Setzen von
3143 Syntaxbäumen mit Hilfe des Pakets tikz-qtreet]
3144 \RequirePackage{tikz-qtreet}
3145
3146 \tikzset{li parsetree/.style={
3147     every internal node/.style={
3148         draw,circle
3149     },
3150     every leaf node/.style={
3151         draw,rectangle
3152     },
3153 }
3154 }
3155
```

2.46 synthese-algorithmus.sty

```
3156 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3157 \ProvidesPackage{lehramt-informatik-synthese-algorithmus}[2021/03/19
3158 Hilfsmakros zum Setzen des Synthese-Algorithmuses zur Umformung einer
3159 Relation in die 3. Normalform]

3160 \liLadePakete{normalformen,mathe,typographie}
3161 \ExplSyntaxOn
```

2.46.1 Makro-Kürzel

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

2.46.2 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}
```

2.46.3 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}}

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

2.46.4 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}

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

2.46.5 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 $\mathcal{R}_\alpha := \alpha \cup \beta$. —

3. Schlüssel hinzufügen

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

4. Entfernung überflüssiger Teilschemata

— Eliminiere diejenigen Schemata \mathcal{R}_α , die in einem anderen Relationenschema $\mathcal{R}_{\alpha'}$ enthalten sind, d. h. $\mathcal{R}_\alpha \subseteq \mathcal{R}_{\alpha'}$. —

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

```
3162 \def\liSyntheseUeberschrift#1{
3163   {
3164     \bfseries
3165     \rmfamily
3166     \str_case:nn {#1} {
3167       {1} {Kanonische-Überdeckung}
3168       {1-1} {Linksreduktion}
3169       {1-2} {Rechtsreduktion}
3170       {1-3} {Löschen-leerer-Klauseln}
3171       {1-4} {Vereinigung}
3172       {2} {Relationsschemata-formen}
3173       {3} {Schlüssel-hinzufügen}
3174       {4} {Entfernung-überflüssiger-Teilschemata}
3175     }
```

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

```

3178 \def\li@synthese@erklaerung@texte#1{
3179   \str_case:nn {#1} {
3180     {1} {
3181       Die-kanonische-Überdeckung---also-die-kleinst-mögliche-noch-
3182       äquivalente-Menge-von-funktionalen-Abhängigkeiten-kann-in-vier-
3183       Schritten-erreicht-werden.
3184     }
3185     {1-1} {
3186       Führe-für-jede-funktionale-Anhängigkeit~
3187        $\alpha \rightarrow \beta$ -die-Linksreduktion-durch,~
3188       überprüfe-also-für-alle~
3189        $A \in \alpha$ ,~ob- $A$ -überflüssig-ist,~d.h.-ob-
3190        $\beta \subseteqq \text{AttributHuelle}\{F, \alpha\}$ .
3191     }
3192     {1-2} {
3193       Führe-für-jede-(verbliebene)-funktionale-Abhängigkeit- $\alpha \rightarrow \beta$ -die-Rechtsreduktion-durch,~überprüfe-also-für-
3194       alle- $B \in \beta$ ,~ob- $B \in \text{AttributHuelle}\{F, (\alpha \rightarrow \beta)\}$ ,~
3195        $\alpha \rightarrow \beta$ - $\cup (\alpha \rightarrow B)$ ,~
3196        $\alpha$ -gilt.~In-diesem-Fall-ist- $B$ -auf-der-rechten-Seite-
3197       überflüssig-und-kann-eliminiert-werden,~dh- $\alpha \rightarrow \beta$ -wird-durch- $\alpha \rightarrow (\beta \cup B)$ -
3198       ersetzt.
3199     }
3200     {1-3} {
3201       Entferne-die-funktionalen-Abhängigkeiten-der-Form- $\alpha \rightarrow \emptyset$ ,~die-im-2.-Schritt-möglicherweise-
3202       entstanden-sind.
3203     }
3204     {1-4} {
3205       Fasse-mittels-der-Vereinigungsregel-funktionale-Abhängigkeiten-
3206       der-Form- $\alpha \rightarrow \beta_1, \dots, \alpha \rightarrow \beta_n$ ,~so-dass- $\alpha \rightarrow \beta_1 \cup \dots \cup \beta_n$ -
3207       verbleibt.
3208     }
3209     % Kemper Seite 197
3210     {2} {
3211       Erzeuge-für-jede-funktionale-Abhängigkeit- $\alpha \rightarrow \beta$ -ein-Relationenschema- $\mathcal{R} \subseteq \mathcal{A}$ -
3212        $:= \alpha \cup \beta$ .
3213     }
3214     {3} {
3215       Falls-eines-der-in-Schritt-2.-erzeugten-Schemata- $\mathcal{R} \subseteq \mathcal{A}$ -einen-Schlüsselkandidaten-von- $\mathcal{R}$ -bezüglich- $\mathcal{F} \subseteq \mathcal{C}$ -enthält,~sind-wir-fertig,~sonst-wähle-einen-Schlüsselkandidaten-
3216        $\mathcal{K} \subseteq \mathcal{A}$ -aus-und-definiere-folgendes-
3217       zusätzliche-Schema- $\mathcal{R} \subseteq \mathcal{K} := \mathcal{K}$ -und- $\mathcal{F} \subseteq \mathcal{K} := \emptyset$ 
3218     }
3219     {4} {
3220       Eliminiere-diejenigen-Schemata- $\mathcal{R} \subseteq \mathcal{A}$ ,~die-in-einem-
3221       anderen-Relationenschema- $\mathcal{R} \subseteq \mathcal{A}$ -enthalten-sind,~d.h.-
3222        $\mathcal{R} \subseteqq \mathcal{R} \subseteq \mathcal{A}$ .
3223     }
3224   }
3225 }
3226 \def\liSyntheseErklaerung#1{
3227   {
3228     \itshape
3229     \footnotesize
3230   }
3231 }

```

```

3238     \liParagraphMitLinien{\li@synthese@erklaerung@texte{#1}}
3239   }
3240 }

```

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

```

3241 \def\liSyntheseUeberErklaerung#1{
3242   \liSyntheseUeberschrift{#1}\par
3243   \liSyntheseErklaerung{#1}
3244 }

```

```

3245 \ExplSyntaxOff
3246

```


2.47 tabelle.sty

```
3247 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3248 \ProvidesPackage{lehramt-informatik-tabelle}[2020/12/05 Lädt das Paket tabluarx]
3249 \RequirePackage{tabularx}
3250
```

2.48 typography.sty

```
3251 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3252 \ProvidesPackage{lehramt-informatik-typographie}[2021/03/20 Typographische Makros,
3253 die das Erscheinungsbild verändern. Die Schriftdefinition sind in
3254 formatierung.sty definiert.]
```

```
3255 \ExplSyntaxOn
```

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

```
3256 \RequirePackage{fontawesome}
```

`\liErledigt` `\liErledigt:` ☑

```
3257 \let\liErledigt=\faCheckSquareO
```

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

```
3258 \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 —

```
3259 \def\liParagraphMitLinien#1{
3260   \noindent
3261   \vrule height 2pt depth -1.6pt width 0.4cm
3262   \enspace
3263   #1
3264   \enspace
3265   \leaders\vrule height 2pt depth -1.6pt \hfill \null
3266   \par
3267   \medskip
3268 }
```

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

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

```
3269 \def\liGeschweifteKlammern#1#2#3#4{
3270   \par
3271   \medskip
3272   \noindent
3273   #1 \, $= \Bigl\{ \$
3274   \vspace{#3}
3275   #2
3276   \vspace{#4}
3277   \begin{flushright} \$\Bigr\} \$\end{flushright}
3278   \par
3279 }
```

`\liTypoUeberschrift`

```
3280 \def\liTypoUeberschrift#1{
3281   {
3282     \bfseries\rmfamily
3283     #1
3284   }
3285 }
```

\liTypoUeberGross

```
3286 \def\liTypoUeberGross#1{
3287   {
3288     \huge
3289     \liTypoUeberschrift{#1}
3290   }
3291 }
```

\liTypoUeberGross

```
3292 \def\liTypoUeberGROSS#1{
3293   {
3294     \Huge
3295     \liTypoUeberschrift{#1}
3296   }
3297 }
```

3298 \ExplSyntaxOff

3299

2.49 uml.sty

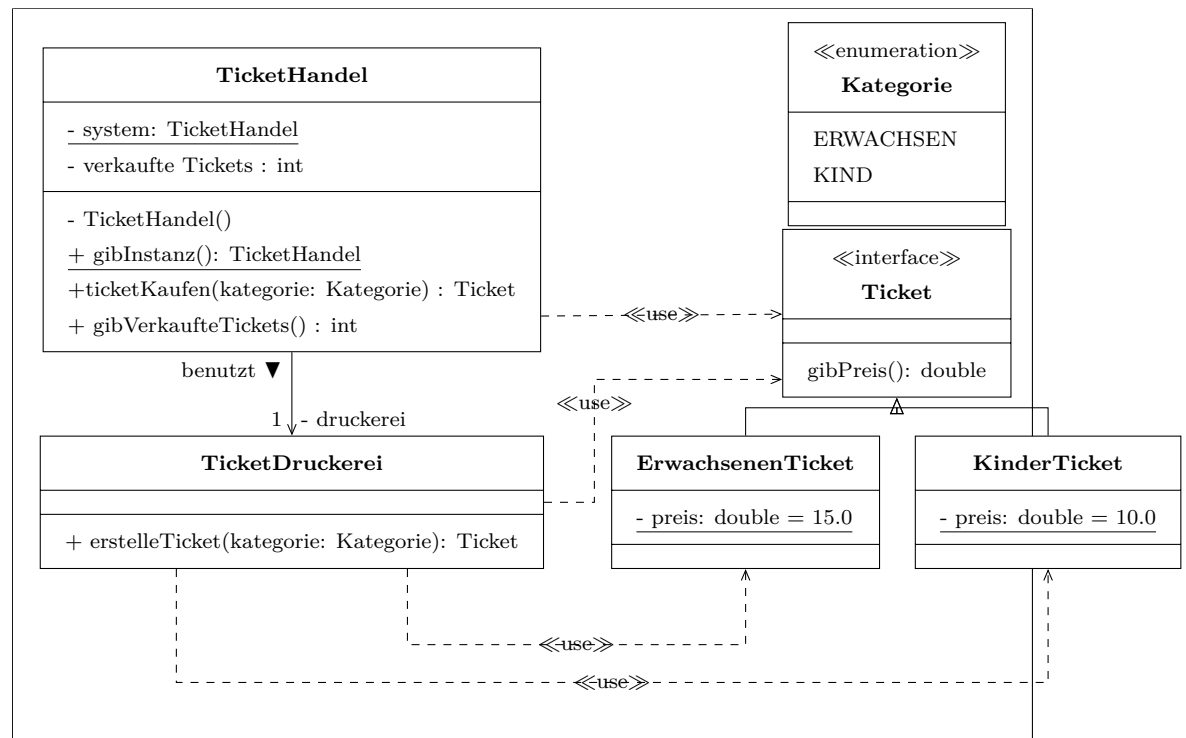
```

3300 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3301 \ProvidesPackage{lehramt-informatik-uml}[2020/06/13 Ein Hüll-Paket um
3302 `tikz-uml', das einige Design-Einstellungen vornimmt und manche
3303 Erweiterung bereitstellt]

3304 \RequirePackage{tikz-uml}
3305 \RequirePackage{tikz-uml-activity}
3306 % Not compatible with wasysym
3307 %\RequirePackage{mathabx}
3308 \RequirePackage{wasysym}
3309 \usetikzlibrary{positioning}

3310 \tikzumlset{
3311   fill class=white!0,
3312   font=\footnotesize,
3313   fill object=white!0,
3314   fill note=white!0,
3315   fill state=white!0,
3316   % Use case
3317   fill usecase=white!0,
3318   fill system=white!0,
3319 }

```



```

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

```

```

3320 \NewDocumentCommand{ \liUmlLeserichtung } { 0{dir=right} m } {
3321   \def\@liDirLeft{}
3322   \def\@liDirRight{ \RIGHTarrow}
3323   \pgfkeys{/lese/dir/.is choice}
3324   \pgfkeys{/lese/dir/up/.code={\def\@liDirRight{ \UParrow}}}
3325   \pgfkeys{/lese/dir/down/.code={\def\@liDirRight{ \DOWNarrow}}}
3326   \pgfkeys{/lese/dir/left/.code={\def\@liDirRight{}\def\@liDirLeft{\LEFTarrow }}}
3327   \pgfkeys{/lese/dir/right/.code={\def\@liDirRight{ \RIGHTarrow}}}
3328
3329   \def\@liPos{above}
3330   \pgfkeys{/lese/pos/.code={\def\@liPos{##1}}}
3331

```

```

3332 \def\@liDistance{0cm}
3333 \pgfkeys{/lese/distance/.code={\def\@liDistance{##1}}}}
3334
3335 \pgfkeys{/lese/.cd,#1}
3336
3337 \node[\@liPos = \@liDistance of #2-middle] {
3338   \@liDirLeft{\footnotesize#2}\@liDirRight
3339 };
3340 }
3341

```

2.50 vollstaendige-induktion.sty

3342 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
 3343 \ProvidesPackage{lehramt-informatik-vollstaendige-induktion}[2021/07/01
 3344 Hilfsmakros zum Setzen der Vollständigen Induktion, vor allem die
 3345 Überschriften für die einzelnen Schritte]

2.50.1 Makro-Kürzel

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

```
\begin{align*}
C_{n+1}
&= \frac{
  \{(4 \cdot (\{m\{n + 1\} - 1) + 2) \cdot \text{\textcolor{violet}{cn}}(\{m\{n + 1\} - 1)\}
  \{m\{n + 1\} + 1\}
}{
  \{e\{Java nach Mathe}\} \\\
\%
&= \frac{
  \{(4\{m\{n\} + 2) \cdot \text{\textcolor{violet}{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

3346 \RequirePackage{lehramt-informatik-typographie}
 3347 \RequirePackage{lehramt-informatik-mathe}
 3348 \RequirePackage{lehramt-informatik-syntax}
 3349 \ExplSyntaxOn

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

Let-Abkürzung: \let\m=\liInduktionMarkierung

3350 \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

```
3351 \def\liInduktionErklaerung#1{\scriptsize\text{#1}}
```

\liInduktionAnfang

```
3352 \def\liInduktionAnfang{
```

```
3353   \liPseudoUeberschrift{Induktionsanfang}
```

```
3354
```

```
3355   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
3356   \liParagraphMitLinien{
```

```
3357     Beweise,~dass~ $A(1)$ ~eine~wahre~Aussage~ist.
```

```
3358   }
```

```
3359 }
```

\liInduktionVoraussetzung

```
3360 \def\liInduktionVoraussetzung{
```

```
3361   \liPseudoUeberschrift{Induktionsvoraussetzung}
```

```
3362
```

```
3363   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
3364   \liParagraphMitLinien{
```

```
3365     Die~Aussage~ $A(k)$ ~ist~wahr~für~ein~beliebiges~ $k \in \mathbb{N}$ ~.
```

```
3366   }
```

```
3367 }
```

\liInduktionSchritt

```
3368 \def\liInduktionSchritt{
```

```
3369   \liPseudoUeberschrift{Induktionsschritt}
```

```
3370
```

```
3371   % https://de.wikibooks.org/wiki/Mathe\_für\_Nicht-Freaks:\_Vollständige\_Induktion
```

```
3372   \liParagraphMitLinien{
```

```
3373     Beweise,~dass~wenn~ $A(n=k)$ ~wahr~ist,~
```

```
3374     auch~ $A(n=k+1)$ ~wahr~sein~muss.
```

```
3375   }
```

```
3376 }
```

```
3377 \ExplSyntaxOff
```

```
3378
```

2.51 wasserfall.sty

```
3379 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3380 \ProvidesPackage{lehramt-informatik-wasserfall}[2020/06/10]
3381 \RequirePackage{tikz}
3382 \tikzset{wasserfall/.style={
3383   >=stealth,
3384   node distance = 2mm and -8mm,
3385   start chain = A going below right,
3386   every node/.style = {
3387     draw,
3388     text width=24mm,
3389     minimum height=12mm,
3390     align=center,
3391     inner sep=1mm,
3392     fill=white,
3393     drop shadow={fill=black},
3394     on chain=A
3395   },
3396 }}
3397 \usetikzlibrary{chains,positioning,shadows}
3398
```


2.52 wpkalkuel.sty

```
3399 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3400 \ProvidesPackage{lehrant-informatik-wpkalkuel}[2020/06/13]
```

2.52.1 Makro-Kürzel

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

```
3401 \RequirePackage{amsmath}
3402 \ExplSyntaxOn
```

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

```
3403 \def\liWpKalkuelOhneMathe#1#2{
3404   \text{wp}(\texttt{\scriptsize"#1"},\thinspace #2)
3405 }
3406 \def\liWpKalkuel#1#2{
3407   \ifmmode
3408     \liWpKalkuelOhneMathe{#1}{#2}
3409   \else
3410     $\liWpKalkuelOhneMathe{#1}{#2}$
3411   \fi
3412 }
```

\MatheEnv

```
3413 \def\MatheEnv#1{
3414   \medskip
3415
3416   \hspace{1em}#1
3417
3418   \medskip
3419 }
```

\Mathe

```
3420 \def\Mathe#1{
3421   \MatheEnv{${#1}$}
3422 }
```

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

```
3423 \def\liWpEquivalent#1{
3424   \MatheEnv{${\equiv}\hspace{1em}${#1}$}
3425 }
```

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

```
3426 \newlength{\@Skip@Erklaerung@Reset}
3427 \def\liWpErklaerung#1{
3428   \setlength{\@Skip@Erklaerung@Reset}{\leftskip}
3429   \setlength{\leftskip}{0.5cm}
3430
3431   \par
3432   \noindent
3433   {
3434     \scriptsize
3435     #1
3436   }
3437   \par
3438
3439   \setlength{\leftskip}{\@Skip@Erklaerung@Reset}
3440 }
```

\liWpErklaerungVerzweigung

```

3441 \def\liWpErklaerungVerzweigung{
3442   $\liWpKalkuelOhneMathe{if~\{-b~\}~then~\{-a1~\}~else~\{-a2~\}}{Q}
3443   \equiv
3444   (b \land \liWpKalkuelOhneMathe{a1}{Q})
3445   \lor
3446   (\neg b \land \liWpKalkuelOhneMathe{a2}{Q})$
3447 }

3448 \ExplSyntaxOff

3449

```

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	\sqcup 2530, 2540
\,	364, 423, 478, 705, 723, 1355, 1357, 2139, 2140, 2141, 2578, 3079, 3086, 3273	
\@Skip@Erklaerung@Reset	3426, 3428, 3439	
\@afterheading	1606, 1984, 2108	
\@afterindentfalse	1605, 1983, 2107	
\@liDirLeft	3321, 3326, 3338	
\@liDirRight	3322, 3324, 3325, 3326, 3327, 3338	
\@liDistance	3332, 3333, 3337	
\@liPos	3329, 3330, 3337	
\\	735, 763, 764, 767, 768, 771, 772, 867, 868, 869, 976, 1010, 1012, 1042, 1051, 1096, 1138, 1139, 1140, 1145, 1146, 1147, 1167, 1971, 2473, 2531, 2534	
\{	212, 1292, 1302, 1314, 1315, 1320, 1354, 1810, 2518, 2943, 3273, 3442	
\}	212, 1292, 1302, 1314, 1315, 1322, 1358, 1811, 2518, 2943, 3277, 3442	
_	37, 45, 50, 52, 311, 339, 363, 366, 376, 399, 403, 407, 409, 411, 413, 416, 422, 423, 437, 438, 444, 447, 450, 460, 467, 469, 472, 1576, 1588, 1624, 1678	
		\bgroup 1964, 2520, 2988
		\beschriftung 2007, 2011, 2015, 2019, 2023, 2025
		\beta 3187, 3190, 3194, 3195, 3196, 3199, 3209, 3210, 3211, 3216, 3217
		\bf 2765, 2766, 2767
		\bfseries 564, 1405, 1407, 1583, 1644, 1701, 2388, 2765, 2771, 2773, 2775, 2776, 3164, 3282
		\bgroup 2002, 2046
		\Bigl 3273
		\Bigr 3277
		\bigskip 456, 737, 742, 1601, 1613, 1616, 1694, 1746, 1749, 1973, 2446
		\bool 342, 424
		\bowtie 2913, 2916, 2917, 2918
		\Box 164
		\boxtimes 554
		C
		\c 1515, 1516
		\cdot 2184, 2240, 2251
		\centerline 1769, 2408, 2866, 2888, 2903
		\cftbeforesecskip 1727
		\cftbeforesubsecskip 1728, 1729
		\cftbeforesubsubsecskip 1730
		\cftsubsecafterpnum 1729
		\chapter 1405, 1406
		\char 1971
		\clearpage 1645, 1739, 1751, 1965

<code>\frac</code>	2186, 2219, 2251, 2266	<code>\ifnum</code>	2749, 1374, 1375, 1376,
<code>\fullouterjoin</code>	2918		1377, 1380, 1381,
G			
<code>\g</code>	39, 285, 286, 308, 312, 318, 319, 320, 321, 322, 324, 325, 327, 328, 330, 331, 332, 333, 334, 335, 336, 337, 340, 344, 345, 346, 349, 351, 352, 353, 354, 355, 356, 401, 405, 409, 412, 413, 415, 416, 418, 419, 426, 427, 428, 429, 437, 438, 440, 446, 447, 449, 450, 452, 453, 465, 468, 473, 475, 479, 1550, 1552, 1563, 1565, 1572, 1585, 1586, 1652, 1656, 1660, 1664, 1665, 1666, 1667, 1668, 1670, 1671, 1673, 1703, 1705, 1710, 1714	<code>\ifx</code>	2023, 2550, 3018
		<code>\in</code>	583, 739, 1817, 2199, 2202, 2205, 2234, 2240, 2245, 2703, 2714, 2724, 2734, 3187, 3189, 3195, 3216, 3365
		<code>\includegraphics</code>	1454, 1459
		<code>\inhalts</code>	1724, 1741
		<code>\inhaltsverzeichnis</code>	1962
		<code>\input</code>	17, 20, 23, 26, 29, 485, 1661
		<code>\inputminted</code>	3097, 3107, 3117, 3129, 3132, 3136
		<code>\int</code>	3000
		<code>\item</code>	554, 555, 832, 836, 841, 846, 890, 899, 904, 912, 984, 989, 993, 1019, 1059, 1064, 1071, 1079, 1108, 1113, 1117, 1122, 1222, 1227, 1232, 1774, 1775, 2090, 2094, 2214, 2219, 2223, 2233, 2239, 2244, 2256, 2260, 2264, 2268, 2272, 2279, 2283, 2287, 2708, 2711, 2714, 2728, 2731, 2734
<code>\Gamma</code>	111, 169, 212, 1315	<code>\itshape</code>	608, 3236
J			
<code>\gappto</code>	3023	<code>\j</code>	1515, 1516, 1518, 1519, 1520, 1525, 1526, 1527
<code>\geometry</code>	5	K	
<code>\geq</code>	1817, 2217, 2703, 2708, 2724, 2728	<code>\k</code>	1525
H			
<code>\hbox</code>	2913	<code>\keys</code>	41, 75, 87, 117, 127, 175, 185, 316, 627, 631, 645, 650, 1379, 1386
<code>\headrulewidth</code>	1431, 1921	L	
<code>\headwidth</code>	1923	<code>\l</code>	68, 69, 70, 71, 72, 73, 76, 77, 78, 79, 80, 82, 84, 89, 90, 91, 92, 93, 94, 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, 625, 628, 633, 634, 637, 642, 643, 646, 647, 652,
<code>\hfill</code>	1591, 2236, 2242, 2247, 3265		
<code>\hinweis</code>	1994		
<code>\hline</code>	2473		
<code>\horizontale</code>	1556, 1599		
<code>\href</code>	1368, 2137, 2304, 2339, 2348, 3093		
<code>\hspace</code>	2665, 3416, 3424		
<code>\ht</code>	2914		
<code>\Huge</code>	1701, 3294		
<code>\huge</code>	1644, 3288		
I			
<code>\i</code>	1525, 1526		
<code>\ifcase</code>	2822		
<code>\ifLiADDITUM</code>	229, 2043, 2061		
<code>\ifLiANTWORT</code>	233, 1999, 2036		
<code>\ifLiEXKURS</code>	231, 2067		
<code>\ifmmode</code>	657, 665, 673, 681, 688, 1294, 1308, 1344, 2155, 2165, 2175, 2189, 2512, 2594, 2655, 3407		
<code>\IfNoValueTF</code>	2050, 2133, 2137		

<code>\li@EntwurfsCodeAllgemein</code>	<code>\liChomskyUeberErklaerung</code>	<code>\liEntwurfsEinzelstueckBeschreibung</code>
..... 751 613 1002 , 1029
<code>\li@fussnote@text</code> 2424 ,	<code>\liChomskyUeberschrift</code>	<code>\liEntwurfsEinzelstueckCode</code>
2430 , 2434 , 2438 , 2442 562 , 614 1025 , 1035
<code>\li@GithubLink</code>	<code>\liCpmEreignis</code>	<code>\liEntwurfsEinzelstueckUml</code>
3089 , 3102 , 3112 , 3123	623 1007 , 1031
<code>\li@mget</code> . 1502 , 1506 , 1526	<code>\liCpmFruehErklaerung</code> 694	<code>\liEntwurfsErbauer</code> . 1085
<code>\li@minc</code>	<code>\liCpmFruehI</code> ... 687 , 707	<code>\liEntwurfsErbauerAkteure</code>
1505 , 1527	<code>\liCpmSpaetErklaerung</code> 712 1057 , 1087
<code>\li@mset</code>	<code>\liCpmSpaetI</code> ... 680 , 725	<code>\liEntwurfsErbauerUml</code>
1499 , 1507 , 1516 , 1519	<code>\liCpmVon</code> 1037 , 1086
<code>\li@numdiscs</code>	<code>\liCpmVonOhneMathe</code> ..	<code>\liEntwurfsFabrikmethode</code>
... 1511 , 1520 , 1526 663 , 666 , 668 1131 , 1156
<code>\li@Rmodell@Schrift</code> .	<code>\liCpmVonZu</code>	<code>\liEntwurfsFabrikmethodeAkteure</code>
... 2927 , 2936 , 2946	655 1106 , 1133
<code>\li@sequence</code> .. 1512 , 1525	<code>\liCpmVonZuOhneMathe</code>	<code>\liEntwurfsFabrikmethodeUml</code>
<code>\li@synthese@erklaerung@text</code> 655 , 658 , 660 1089 , 1132
..... 3178 , 3238	<code>\liCpmVorgang</code>	<code>\liEntwurfsKompositum</code>
<code>\liAbleitung</code>	<code>\liCpmZu</code> 1156
1329	<code>\liCpmZuOhneMathe</code> ...	<code>\liEntwurfsKompositumAkteure</code>
<code>liAdditum (environment)</code> 671 , 674 , 676 1158
..... 2041	<code>liDiagramm (environ-</code>	<code>\liEntwurfsKompositumUml</code>
<code>\LiADDITUMfalse</code>	<code>ment)</code> 1135 , 1157
229	2120	<code>\liEntwurfsModellPraesentationSteuerung</code>
<code>\LiADDITUMtrue</code> . 229 , 1547	<code>liEinbettung (environ-</code>	<code>\liEntwurfsModellPraesentationSteuerungAk</code>
<code>liAHuelle (environment)</code>	<code>ment)</code>	<code>\liEntwurfsModellPraesentationSteuerungUml</code>
..... 2519	1996 1160 , 1178
<code>\liAlphabet</code>	<code>\liEntwurfs</code>	<code>\liEntwurfsStellvertreter</code>
1314	1177 1201
<code>liAntwort (environment)</code>	<code>\liEntwurfsAbstrakteFabrik</code>	<code>\liEntwurfsStellvertreterCode</code>
..... 1997 807 1195 , 1203
<code>\LiANTWORTfalse</code>	<code>\liEntwurfsAbstrakteFabrikBeschreibung</code>	<code>\liEntwurfsStellvertreterUml</code>
233 755 , 808 1181 , 1202
<code>\LiANTWORTtrue</code>	<code>\liEntwurfsAbstrakteFabrikCode</code>	<code>\liEntwurfsZustand</code> . 1242
233 802 , 812	<code>\liEntwurfsZustandAkteure</code>
<code>\liAnweisung</code>	<code>\liEntwurfsAbstrakteFabrikUml</code> 1220 , 1244
1895 760 , 810	<code>\liEntwurfsZustandCode</code>
<code>\liAssemblerCode</code> ... 3127	<code>\liEntwurfsAdapter</code> .. 859 1238 , 1245
<code>\liAssemblerDatei</code> .. 3128	<code>\liEntwurfsAdapterAkteure</code>	<code>\liEntwurfsZustandUml</code>
<code>\liAttribut</code> 829 , 861 1205 , 1243
2946	<code>\liEntwurfsAdapterCode</code>	<code>\liEpsilon</code>
<code>\liAttributHuelle</code> 853 , 862	1300
... 2510 , 3190 , 3195	<code>\liEntwurfsAdapterUml</code>	<code>\liErAttribute</code>
<code>\liAttributHuelleOhneMathe</code> 814 , 860	... 1258 , 1272 , 1274
..... 2510 , 2513 ,	<code>\liEntwurfsBeobachter</code> 930	<code>\liErDatenbankName</code> . 1277
2515 , 2529 , 2539 , 2547	<code>\liEntwurfsBeobachterAkteure</code>	<code>\liErEntity</code> 1256 , 1260 , 1262
<code>\liAttributMenge</code> 888 , 932	<code>\liErledigt</code>
2518 , 2530 , 2533 ,	<code>\liEntwurfsBeobachterCode</code>	3257
2540 , 2541 , 2555 , 2557 922 , 933	<code>\liErMpAttribute</code> ... 1271
<code>\liAufgabe</code>	<code>\liEntwurfsBeobachterUml</code>	<code>\liErMpEntity</code>
16 864 , 931	1259
<code>\liAufgabenMetadaten</code>	<code>\liEntwurfsDekorierer</code> 962	<code>\liErMpRelationship</code> 1265
..... 47 , 1569 , 1676	<code>\liEntwurfsDekoriererAkteure</code>	<code>\liErRelationship</code> ...
<code>\liAufgabenTitel</code> 55 964	... 1257 , 1266 , 1268
<code>\liAusdruck</code>	<code>\liEntwurfsDekoriererCode</code>	<code>\liExamensAufgabe</code> 19
1351 954 , 965	<code>\liExamensAufgabeA</code> ... 28
<code>\liAutomat</code>	<code>\liEntwurfsDekoriererUml</code>	<code>\liExamensAufgabeTA</code> .. 25
66 935 , 963	<code>\liExamensAufgabeTTA</code> . 22
<code>\liAutomatenKante</code> 98	<code>\liEntwurfsEinfacheFabrik</code>	<code>\liExkurs (environment)</code> 2066
<code>\liBandAlphabet</code> 1315 998	<code>\LiEXKURSfalse</code>
<code>\liBedingung</code>	<code>\liEntwurfsEinfacheFabrikAkteure</code>	231
1896 982 , 1000	<code>\LiEXKURStrue</code>
<code>\liBedingungDrei</code>	<code>\liEntwurfsEinfacheFabrikUml</code>	555
... 2204 , 2247 , 2287 967 , 999	<code>\liFlaci</code>
<code>\liBedingungEins</code>	<code>\liEntwurfsEinzelstueck</code>	1361
... 2198 , 2236 , 2279 1028	<code>\liFremd</code>
<code>\liBedingungFalsch</code> . 1898	<code>\liEntwurfsEinzelstueckAkteure</code>	2926
<code>\liBedingungWahr</code> ... 1897 1017 , 1033	
<code>\liBedingungZwei</code>		
... 2201 , 2242 , 2283		
<code>\liBeschriftung</code> 1986		
<code>\liBindeAufgabeEin</code> . 1659		
<code>\liChomskyErklaerung</code>		
..... 574 , 615		

<code>\liFunktionaleAbhaengigkeit</code>	1759, 2324, 2422, ... 2549, 2552, 2560	<code>\liPetriSetzeSchluessel</code>	2606
<code>\liFunktionaleAbhaengigkeiten</code>	2563	<code>\liPetriTransitionsName</code>	2653, 2665
<code>\liFussnote</code>	2423, 2425	<code>\liPetriTransitionsNameOhneMathe</code>	2653, 2656, 2658
<code>\liFussnoteDreiText</code>	2437, 2455	<code>\liPetriTransPfeile</code>	2665
<code>\liFussnoteEinsText</code>	2429, 2449	<code>\liPolynomiellReduzierbar</code>	1779
<code>\liFussnoteLink</code>	2136	<code>\liPotenzmenge</code>	1301, 1305, 2676
<code>\liFussnoten</code>	2445	<code>\liPotenzmengeOhneMathe</code>	1302, 1303, 1304
<code>\liFussnoteUrl</code>	1175, 2132	<code>\liPrimaer</code>	2925
<code>\liFussnoteVierText</code>	2441, 2458	<code>\liProblemBeschreibung</code>	1763
<code>\liFussnoteZweiText</code>	2433, 2452	<code>\liProblemClique</code>	1786
<code>\liGeschweifteKlammern</code>	1333, 2564, 3269	<code>\liProblemName</code>	1762, 1769, 1781, 1783, 1796, 1807, 1808, 1816, 1817
<code>\liGrafikCCLizenz</code>	1453, 2408	<code>\liProblemSat</code>	1815
<code>\liGrafikLogo</code>	1458, 1693, 2384	<code>\liProblemSubsetSum</code>	1806, 1815
<code>\liGrafikLogoPfad</code>	1450, 1455, 1460	<code>\liProblemVertexCover</code>	1786, 1794
<code>\liGrammatik</code>	1372	<code>\liProduktionen</code>	1340, 1382
<code>liGraphenFormat</code> (environment)	1492	<code>liProduktionsRegeln</code> (environment)	1330
<code>\liHanoi</code>	1499	<code>liProjektSprache</code> (environment)	1995
<code>\liHaskellCode</code>	3134	<code>\liPseudoUeberschrift</code>	1972, 2052, 2055, 2470, 2480, 3353, 3361, 3369
<code>\liHaskellDatei</code>	3135	<code>\liPumpingKontextfrei</code>	2722
<code>\liInduktionAnfang</code>	3352	<code>\liPumpingRegulaer</code>	2701
<code>\liInduktionErklaerung</code>	3351	<code>liQuellen</code> (environment)	2090
<code>\liInduktionMarkierung</code>	3350	<code>\liRechtsReduktionInline</code>	2544
<code>\liInduktionSchritt</code>	3368	<code>\liRekursionsGleichung</code>	2195, 2258
<code>\liInduktionVoraussetzung</code>	3360	<code>\liRelation</code>	2574
<code>liJavaAngabe</code> (environment)	3075	<code>liRelationenSchemaFormat</code> (environment)	2947
<code>\liJavaCode</code>	3078	<code>\liRelationMenge</code>	2941
<code>\liJavaDatei</code>	753, 3096	<code>\liRichtig</code>	554
<code>\liJavaExamen</code>	3116	<code>liRmodell</code> (environment)	2927
<code>\liJavaTestDatei</code>	3106	<code>\liRundeKlammer</code>	2148, 2152, 2162, 2172, 2186, 2587, 2591
<code>liKasten</code> (environment)	1420	<code>\liSetzeExamenTeilaufgabeNr</code>	1655
<code>\liKellerAutomat</code>	107	<code>\liSetzeExamenThemaNr</code>	1651
<code>\liKellerKante</code>	145	<code>\liSortierMarkierung</code>	2966
<code>\liKellerUebergang</code>	140, 146	<code>\liSortierPfeil</code>	2960
<code>\liKontrollCode</code>	1899	<code>\liSortierPfeilUnten</code>	2963
<code>liKontrollflussgraph</code> (environment)	1887	<code>\liSpaltenUmbruch</code>	3035
<code>\liKontrollKnotenPfad</code>	1901	<code>\liSqlCode</code>	3139
<code>\liKontrollTextzeileKnoten</code>	1900, 1905		
<code>\liKurzeTabellenLinie</code>	735		
<code>\liLadeAllePakete</code>	239, 1647		
<code>\liLadePakete</code>	59, 62, 235, 240, 561, 622, 1291, 1535, 1633,		
<code>\liLatexCode</code>	3088		
<code>\liLeereZelle</code>	2460		
<code>liLernkartei</code> (environment)	2110		
<code>\liLinksReduktion</code>	2527		
<code>\liLinksReduktionInline</code>	2536, 2544		
<code>\liLogoTextCCLizenz</code>	1615, 1748, 2404		
<code>\liLogoTextProjekt</code>	1612, 1745, 2380		
<code>\liMasterExkurs</code>	2292		
<code>\liMasterFaelle</code>	2231, 2299		
<code>\liMasterFallRechnung</code>	2277		
<code>\liMasterVariablen</code>	2208, 2294		
<code>\liMasterVariablenDeklaration</code>	2254		
<code>\liMasterWolframLink</code>	2302		
<code>\liMenge</code>	76, 77, 79, 118, 119, 120, 124, 176, 177, 178, 182, 1292, 1341, 1380, 1381		
<code>\liMengeOhneMathe</code>	1292, 1295, 1297		
<code>\liMetaBsclangaulSammlung</code>	2325, 2389		
<code>\liMetaCCLink</code>	2337, 2413		
<code>\liMetaEmailLink</code>	2347, 2363		
<code>\liMetaHermineBsclangaulAndFriends</code>	1696, 2328, 2393		
<code>\liMetaHilfMit</code>	1621, 2354, 2365		
<code>\liMetaQuelltext</code>	1623, 2365		
<code>\liMetaSetze</code>	36, 48, 1570, 1677		
<code>\liMetaUeberDasProjekt</code>	2331, 2400		
<code>\liMinimierungErklaerung</code>	2482		
<code>\liMinispracheDatei</code>	3131		
<code>\linespread</code>	2934		
<code>\liNichtsZuTun</code>	3258		
<code>\liO</code>	2171, 2199		
<code>\liOmega</code>	2161, 2205		
<code>\liOmegaOhneMathe</code>	2161, 2166, 2168		
<code>\liONotationO</code>	2590		
<code>\liOOhneMathe</code>	2171, 2176, 2178		
<code>\liParagraphMitLinien</code>	610, 695, 713, 2484, 3238, 3259, 3356, 3364, 3372		
<code>\liPetriErreichKnotenDrei</code>	2664		
<code>\liPetriErreichTransition</code>	2661		

1577, 1592, 1603, 1685, 1689, 1693, 1695, 1696, 1703, 1705, 1710, 1714, 1718, 1975, 1981, 1987, 2082, 2106, 2115, 2391, 2394, 2866, 2888, 2903, 2944, 3242, 3266, 3270, 3278, 3431, 3437	\QS@sort@a 2821, 2854, 2875, 2876	\right 2149, 2588
\paragraph 1407	\QS@sort@b 2821, 2822	\RIGHTarrow ... 3322, 3327
\parindent 2935	\QS@sort@c 2825, 2832	\Rightarrow 739, 744
\path 99, 146, 209, 652	\QS@sort@d 2833, 2841	\rightarrow 212, 578, 583, 591, 595, 597, 598, 600, 655, 663, 2665, 3187, 3194, 3196, 3199, 3204, 3209, 3210, 3215
\pgfkeys .. 2617, 3323, 3324, 3325, 3326, 3327, 3330, 3333, 3335	\QS@sort@empty . 2823, 2828	\rightouterjoin 2917
\pgfmath@count 2988, 2990, 2992	\QS@sort@single 2824, 2829	\rmfamily 565, 1526, 3165, 3282
\pgfmath@smuggleone 2993	\QSinitialize 2746, 2858, 2893	\Roman 1415
\pgfmathdeclarefunction 2987	\QSIr . 2829, 2835, 2843, 2863, 2877, 2882, 2885	\roman 1415, 1417
\pgfmathhint 2988	\QSIrr 2864, 2877, 2878, 2886	\romannumeral 2833
\pgfmathparse 1506, 3005, 3010, 3013, 3027, 3028	\QSLr 2835, 2842, 2853, 2854, 2862, 2875, 2880, 2884	\rtimes 2310
\pgfmathresult 1507, 2988, 2989, 2991, 2993, 3006, 3014, 3027, 3028	\QSpivotStep 2748, 2858, 2862, 2873	\rule 1559, 2866, 2888, 2903, 2914
\pgfutil@empty 2989	\QSr 2835	
\pgfutil@loop 2990	\QSRr 2844, 2865, 2876, 2881, 2887, 2896, 2897, 2898	S
\pgfutil@repeat 2993	\QSSortStep 2750, 2858, 2874, 2875	\sb 72, 82, 84, 113, 171, 590, 591, 595, 598, 599, 600, 682, 684, 689, 691, 1345, 1347, 2199, 2202, 2205, 2234, 2240, 2488, 2653, 2662, 3209, 3210, 3211, 3216, 3220, 3221, 3224, 3225, 3228, 3229, 3230
\preceq 1782	\quad 2426	\scriptscriptstyle 655, 663, 671
\prime 1761	R	\scriptsize 1364, 1840, 1847, 1853, 1915, 1916, 1919, 1920, 2028, 2412, 3351, 3404, 3434
\printbibliography . 1952	\raggedright 2031	\section ... 52, 1644, 1653
\ProvidesPackage 2, 15, 33, 58, 226, 490, 551, 558, 620, 733, 748, 1249, 1285, 1398, 1436, 1445, 1466, 1495, 1533, 1631, 1756, 1828, 1911, 1927, 1933, 1955, 2145, 2308, 2321, 2420, 2499, 2582, 2602, 2668, 2693, 2698, 2742, 2908, 2921, 2950, 3031, 3038, 3045, 3050, 3142, 3157, 3248, 3252, 3301, 3343, 3380, 3400	\raisebox 1900	\sectionbreak 1645
	\relax 1965, 2835, 2878, 2880, 2881, 2882, 2990, 2992	\seq . 1904, 1905, 1906, 2093, 2094, 2095, 2102
	\renewcommand 1410, 1411, 1412, 1413, 1416, 1417, 1431, 1432, 1729, 1731, 1921, 1922, 2468	\setbox .. 2001, 2045, 2913
	\repeat 2751	\setcounter 1408, 1646, 1734, 2867, 2889, 2903
	\RequirePackage 4, 60, 163, 227, 237, 486, 492, 493, 553, 621, 750, 751, 1251, 1253, 1254, 1290, 1399, 1400, 1402, 1404, 1409, 1418, 1425, 1433, 1437, 1438, 1448, 1449, 1467, 1468, 1469, 1497, 1498, 1546, 1643, 1648, 1649, 1735, 1760, 1829, 1929, 1934, 1935, 1951, 1958, 1959, 1960, 2147, 2311, 2312, 2584, 2585, 2586, 2603, 2605, 2695, 2755, 2757, 2909, 2910, 2911, 2924, 2952, 2983, 3034, 3040, 3047, 3052, 3063, 3064, 3144, 3249, 3256, 3304, 3305, 3307, 3308, 3346, 3347, 3348, 3381, 3401	\setganttlinklabel .. 1439, 1440, 1441, 1442
		\setlength 1727, 1728, 1730, 1923, 2935, 3428, 3429, 3439
Q		\setmainfont 1401
\QS@list 2868, 2879, 2883, 2890, 2896, 2901, 2904		\setmainlanguage 487
\QS@select@equal 2843, 2847		\setminted 3069, 3070
\QS@select@greater 2844, 2848		\setminus 2548
\QS@select@smaller 2839, 2842, 2846		\setul 2926
		\shoveleft 2528
		\shoveright 2532
		\Sigma 69, 110, 168, 1314, 1315, 1375
		\sigma 588, 590, 591
		\SLASH 1971
		\small 2123
		\sort 3000
		\sortList 2999, 3008

\xintifEq	2847	\xintLength	2821	\zB	<u>2139</u>
\xintifForLast		\xintnthelt	2833	\zustandsnamens@liste	
...	2803, 2809, 2815			...	1318, 1325, 1326
\xintifGt	2848				
					Z
\xintifLt	2846	\ZB	<u>2140</u>		