lehramt-informatik

Hermine Bschlangaul *

September 4, 2021

Contents

1 Klassen **4**
1.1 Vorlage Theorie-Teil . . . 5
1.2 Vorlage Aufgabensammlung . . . 6
1.3 Vorlage Aufgabe . . . 7

2 Pakete **8**
2.1 abmessung.sty . . . 9
2.2 aufgaben-einbinden.sty . . . 10
2.3 aufgaben-metadaten.sty . . . 11
2.4 automaten.sty . . . 12
2.4.1 Endlicher Automat . . . 12
2.4.2 Kellerautomat . . . 14
2.4.3 Turingmaschine . . . 15
2.5 basis.sty . . . 18
2.6 baum.sty . . . 22
2.6.1 Binärbaum . . . 23
2.6.2 AVL-Baum . . . 24
2.6.3 B-Baum . . . 25
2.7 checkbox.sty . . . 26
2.8 chomsky-normalform.sty . . . 27
2.9 cpm.sty . . . 29
2.9.1 Makro-Kürzel . . . 29
2.9.2 Hilfsmakros für Zeitpunkt-Tabelle . . . 30
2.10 cyk-algorithmus.sty . . . 32
2.10.1 Makro-Kürzel . . . 32
2.10.2 TeX-Markup-Beispiel . . . 32
2.11 entwurfsmuster.sty . . . 33
2.11.1 Namensschema der Entwurfsmuster-Makros: . . . 33
2.11.2 Reihenfolge . . . 33
2.11.3 Abstrakte Fabrik (Abstract Factory) . . . 33
2.11.4 Adapter . . . 34
2.11.5 Beobachter (Observer) . . . 36
2.11.6 Dekorierer (Decorator) . . . 38
2.11.7 Einfache Fabrik (Simple Factory) . . . 39
2.11.8 Einzelstück (Singleton) . . . 40
2.11.9 Erbauer (Builder) . . . 40
2.11.10 Fabrikmethode (Factory Method) . . . 42
2.11.11 Kompositum (Composite) . . . 43
2.11.12 Modell-Präsentation-Steuerung (Model-View-Controller) . . . 44
2.11.13 Zustand (State) . . . 45
2.12 er.sty . . . 47
2.12.1 Tex-Markup-Beispiel: Komplettes Diagramm . . . 47

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

2.12.2 Tex-Markup-Beispiel: EER Enhanced Entity-Relation-Modell nach Elmasri/Navante . . . 48
2.12.3 Makro-Kürzel . . . 48
2.13 formale-sprachen.sty . . . 50
2.14 formatierung.sty . . . 53
2.14.1 Schriftarten / Typographie . . . 53
2.14.2 Farben . . . 53
2.14.3 Überschriften . . . 53
2.14.4 Listen . . . 53
2.14.5 Kasten . . . 53
2.14.6 Header . . . 53
2.15 gantt.sty . . . 54
2.16 grafik.sty . . . 55
2.17 graph.sty . . . 56
2.18 hanoi.sty . . . 58
2.19 klassen-konfiguration-aufgabe.sty . . . 59
2.20 komplexitaetstheorie.sty . . . 60
2.20.1 Makro-Kürzel . . . 60
2.21 kontrollflussgraph.sty . . . 62
2.21.1 Makro-Kürzel . . . 62
2.21.2 TeX-Markup-Beispiel . . . 62
2.21.3 TikZ: pin . . . 62
2.21.4 Umgebungen . . . 63
2.21.5 Makros . . . 64
2.22 kopf-fusszeilen.sty . . . 65
2.23 literatur-dummy.sty . . . 66
2.24 literatur.sty . . . 67
2.25 makros.sty . . . 68
2.26 master-theorem.sty . . . 72
2.26.1 Makro-Kürzel . . . 72
2.27 mathe.sty . . . 76
2.28 minimierung.sty . . . 77
2.29 normalformen.sty . . . 80
2.29.1 Makro-Kürzel . . . 80
2.30 petri.sty . . . 83
2.30.1 Makro-Kürzel . . . 83
2.31 potenzmengen-konstruktion.sty . . . 85
2.32 pseudo.sty . . . 87
2.33 pumping-lemma.sty . . . 88
2.34 quicksort.sty . . . 89
2.35 relationale-algebra.sty . . . 92
2.36 rmodell.sty . . . 93
2.36.1 Makro-Kürzel . . . 93
2.37 sortieren.sty . . . 94
2.38 spalten.sty . . . 96
2.39 sql.sty . . . 97
2.40 struktogramm.sty . . . 98
2.41 syntax.sty . . . 99
2.41.1 Makro-Kürzel . . . 99
2.42 syntaxbaum.sty . . . 101
2.43 synthese-algorithmus.sty . . . 102
2.43.1 Makro-Kürzel . . . 102
2.43.2 TeX-Markup Grundgerüst . . . 102
2.43.3 TeX-Markup Linksreduktion . . . 102
2.43.4 TeX-Markup Rechtreduktion . . . 102
2.43.5 TeX-Markup Relationen formen . . . 103
2.44 tabelle.sty . . . 106
2.45 typographie.sty . . . 107

2.46 uml.sty 108
2.47 vollstaendige-induktion.sty 110
2.47.1 Makro-Kürzel 110
2.48 wasserfall.sty 112
2.49 wpkalkuel.sty 113
2.49.1 Makro-Kürzel 113

3 Index **114**

1 Klassen

1.1 Vorlage Theorie-Teil

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

\begin{document}

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

\chapter{Thema des Theorie-Teils}

\literatur

\end{document}
```

1.2 Vorlage Aufgabensammlung

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

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

1.3 Vorlage Aufgabe

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

\end{document}
```

2 Pakete

2.1 abmessung.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-baum}[2021/09/04 Einstellung der
Seitenabmessung mit Hilfe des geometry-Pakets.]
\RequirePackage{geometry}
\geometry{
  a4paper,
  margin=2cm,
  includeheadfoot,
  %showframe,
  %showcrop,
  %verbose=true,
}

```

2.2 aufgaben-einbinden.sty

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

`\liAufgabe` Eine Aufgaben mit den Pfad relativ zu Wurzelverzeichnis dieses Repository einbinden.

```
\def\liAufgabe#1{
  \input{\LehramtInformatikRepository/#1.tex}
}
```

`\liExamensAufgabe` Eine Examensaufgaben mit den Pfad relativ zu Staatsexamen einbinden, z. B. `\liExamensAufgabe{46116/2015/09/Thema-2/Teilaufgabe-1/Aufgabe-3}`

```
\def\liExamensAufgabe#1{
  \input{\LehramtInformatikRepository/Staatsexamen/#1.tex}
}
```

`\liExamensAufgabeTTA`

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

`\liExamensAufgabeTA`

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

`\liExamensAufgabeA`

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

2.3 aufgaben-metadaten.sty

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

```
\ExplSyntaxOn
```

`\liAufgabenTitel` Momentan eine dummy Makro das die Thematik enthält.

```
\def\liAufgabenTitel#1{
}
```

`\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,
}
```

```
\def\liAufgabenMetadaten#1{

  \_setze_variablen_zurueck:

  \tl_clear:N \g_auf_examen_teilaufgabe_nr_tl

  \keys_set:nn { aufgabenmetadaten } {
    #1
  }

  \_setze_relativen_pfad:

  \tl_if_empty:NTF \g_auf_titel_tl
  {
    \msg_fatal:nn { aufgabenmetadaten } { kein-titel }
  }
  {
  }

  \_gib_examen_titel: {}

  \section{\_gib_aufgaben_titel:}

  \bigskip
}
```

```
\ExplSyntaxOff
```

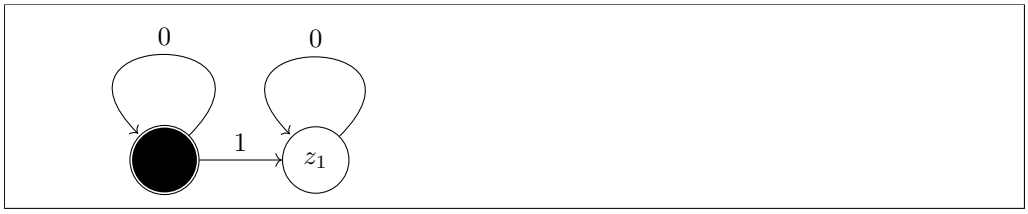
2.4 automaten.sty

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

2.4.1 Endlicher Automat

```
\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}
```




```
\RequirePackage{tikz}
\usetikzlibrary{arrows,automata,positioning}
\liLadePakete{mathe}
\directlua{
  automaten = require('lehramt-informatik-automaten')
}
```

`\liAutomat` `\liAutomat[`⟨*automaten-name*⟩`]{`⟨*zustaende=Z,alphabet=Σ,delta=δ,ende=E,start=z_0*⟩`}`

- `\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}`: $A_{\mathrm{DEA}} = (Z, \Sigma, \delta, E, z_0)$
- `\liAutomat{nea}`: $A_{\mathrm{NEA}} = (Z, \Sigma, \delta, E, z_0)$

```
\ExplSyntaxOn
\NewDocumentCommand {\liAutomat} { O{A} m } {
  \tl_set:Nn \l_zustaende_tl {Z}
  \tl_set:Nn \l_alphabet_tl {\Sigma}
  \tl_set:Nn \l_delta_tl {\delta}
  \tl_set:Nn \l_ende_tl {E}
  \tl_set:Nn \l_start_tl {z\sb{0}}
  \tl_set:Nn \l_typ_tl {}

  \keys_define:nn { automat } {
    zustaende .code:n = {\tl_set:Nn \l_zustaende_tl {\liMenge{##1}}},
    alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
    delta .code:n = {\tl_set:Nn \l_delta_tl {##1}},
    ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
```

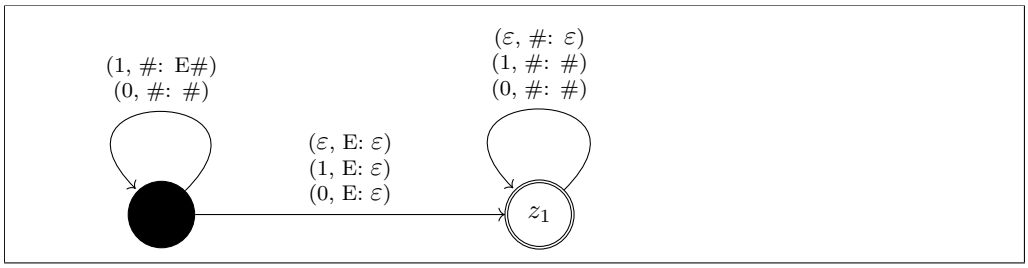
```
    start .code:n = {\tl_set:Nn \l_start_tl {##1}},
    dea .value_forbidden:n = true,
    dea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{DEA}}}},
    nea .value_forbidden:n = true,
    nea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{NEA}}}},
  }

  \keys_set:nn { automat } { #2 }

  $#1 \l_typ_tl = (
    \l_zustaende_tl,
    \l_alphabet_tl,
    \l_delta_tl,
    \l_ende_tl,
    \l_start_tl
  )$
}
\ExplSyntaxOff
```

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

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

\tikzset{
  li automat/.style={
    ->,
    node distance=2cm
  },
}
```

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[`⟨*automaten-name*⟩`]`
`{`⟨*zustaende=Z,alphabet=*Σ*,kelleralphabet=*Γ*,delta=*δ*,start=*z_0*,kellerboden=*$\#$*,ende=E*⟩`}`

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

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

```
\ExplSyntaxOn
\NewDocumentCommand {\liKellerAutomat} { O{K} m } {
  \tl_set:Nn \l_zustaende_tl {Z}
  \tl_set:Nn \l_alphabet_tl {\Sigma}
  \tl_set:Nn \l_kelleralphabet_tl {\Gamma}
  \tl_set:Nn \l_delta_tl {\delta}
  \tl_set:Nn \l_start_tl {z\sb{0}}
  \tl_set:Nn \l_kellerboden_tl {\#}
  \tl_set:Nn \l_ende_tl {E}

  \keys_define:nn { kellerautomat } {
    zustaende .code:n = {\tl_set:Nn \l_zustaende_tl {\liMenge{##1}}},
    alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
    kelleralphabet .code:n = {\tl_set:Nn \l_kelleralphabet_tl {\liMenge{##1}}},
    delta .code:n = {\tl_set:Nn \l_delta_tl {##1}},
    start .code:n = {\tl_set:Nn \l_start_tl {##1}},
```

```
    kellerboden .code:n = {\tl_set:Nn \l_kellerboden_tl {##1}},
    ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
  }

  \keys_set:nn { kellerautomat } { #2 }

  $#1 = (
    \l_zustaende_tl,
    \l_alphabet_tl,
    \l_kelleralphabet_tl,
    \l_delta_tl,
    \l_start_tl,
    \l_kellerboden_tl,
    \l_ende_tl
  )$
}
\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}`

```
\ExplSyntaxOn
\def\liKellerUebergang#1{
  \directlua{automaten.drucke_keller_uebergaenge('#1')}
}
\ExplSyntaxOff
```

`\liKellerKante` `\liKellerKante[`⟨*tikz-optionen*⟩`]{`⟨*von*⟩`}{`⟨*zu*⟩`}{`⟨*übergange*⟩`}`

Let-Abkürzung: `\let\k=\liKellerKante`

```
\NewDocumentCommand{\liKellerKante} { O{above} m m m  } {
  \path (#2) edge[#1] node{\liKellerUebergang{#4}} (#3);
}
```

```
\tikzset{
  li keller knoten/.style={
    text width=2cm,
    align=center,
    font=\footnotesize,
  },
  li kellerautomat/.style={
    li automat,
    every edge/.append style={
      every node/.style={
        li keller knoten
      }
    }
  }
}
```

2.4.3 Turingmaschine

```
\RequirePackage{amssymb}
```

`\liTuringLeerzeichen` □

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

`\liTuringMaschine` `\liTuringMaschine[`⟨*automaten-name*⟩`]`
`{`⟨*zustaende=Z,alphabet=*Σ*,bandalphabet=*Γ*,delta=*δ*,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},
}
```

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

```
\ExplSyntaxOn
\NewDocumentCommand {\liTuringMaschine} { O{TM} m } {
  \tl_set:Nn \l_zustaende_tl {Z}
  \tl_set:Nn \l_alphabet_tl {\Sigma}
  \tl_set:Nn \l_bandalphabet_tl {\Gamma}
  \tl_set:Nn \l_delta_tl {\delta}
  \tl_set:Nn \l_start_tl {z\sb{0}}
  \tl_set:Nn \l_leerzeichen_tl {\liTuringLeerzeichen}
  \tl_set:Nn \l_ende_tl {E}

  \keys_define:nn { kellerautomat } {
    zustaende .code:n = {\tl_set:Nn \l_zustaende_tl {\liMenge{##1}}},
    alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
    bandalphabet .code:n = {\tl_set:Nn \l_bandalphabet_tl {\liMenge{##1}}},
    delta .code:n = {\tl_set:Nn \l_delta_tl {##1}},
    start .code:n = {\tl_set:Nn \l_start_tl {##1}},
    leerzeichen .code:n = {\tl_set:Nn \l_leerzeichen_tl {##1}},
    ende .code:n = {\tl_set:Nn \l_ende_tl {\liMenge{##1}}},
  }

  \keys_set:nn { kellerautomat } { #2 }

  $\text{#1} = (
    \l_zustaende_tl,
    \l_alphabet_tl,
    \l_bandalphabet_tl,
    \l_delta_tl,
    \l_start_tl,
    \l_leerzeichen_tl,
    \l_ende_tl
  )$
}
\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}`: (z_1: $\square$, L)

```
\ExplSyntaxOn
\def\liTuringUebergangZelle#1{
  \directlua{tex.print(automaten.gib_einen_turing_uebergang('#1'))}
}
\ExplSyntaxOff
```

`\liTuringUebergaenge` Nur in den TikZ-Grafiken zu verwenden. Setzt Zeilenumbrüche ans Ende. Nicht für die Tabelle geeignet.
Let-Abkürzung: `\let\t=\liTuringUebergaenge`
(z_1: $\square$, L)
($\square$: $\square$, R)

```
\ExplSyntaxOn
\def\liTuringUebergaenge#1{
  \directlua{automaten.drucke_turing_uebergaenge('#1')}
}
\ExplSyntaxOff
```

\liTuringKante `\liTuringKante[`⟨*tikz-optionen*⟩`]{`⟨*zustand-oder-lese*⟩`}{`⟨*schreibe*⟩`}{`⟨*richtung*⟩`}`

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

```
\NewDocumentCommand{\liTuringKante} { O{above} m m m  } {
  \path (#2) edge[#1] node{\liTuringUebergaenge{#4}} (#3);
}
```

\liTuringUeberfuehrung

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

```
\tikzset{
  li turingmaschine/.style={
    li automat,
    every edge/.append style={
      every node/.style={
        li keller knoten
      }
    }
  }
}
```

2.5 basis.sty

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

\liLadePakete

```
\NewDocumentCommand{\liLadePakete}{ m }
{
  \clist_map_inline:nn { #1 } { \RequirePackage{lehramt-informatik-##1} }
}
```

\liLadeAllePakete

```
\def\liLadeAllePakete{
  \liLadePakete{
    aufgaben-einbinden,
    automaten,
    baum,
    checkbox,
    chomsky-normalform,
    cpm,
    cyk-algorithmus,
    entwurfsmuster,
    er,
    formale-sprachen,
    gantt,
    grafik,
    graph,
    hanoi,
    kontrollflussgraph,
    makros,
    master-theorem,
    mathe,
    minimierung,
    normalformen,
    petri,
    potenzmengen-konstruktion,
    pseudo,
    quicksort,
    relationale-algebra,
    rmodell,
    sortieren,
    spalten,
    struktogramm,
    syntax,
    syntaxbaum,
    synthese-algorithmus,
    tabelle,
    typographie,
    uml,
    vollstaendige-induktion,
    wasserfall,
    wpkalkuel,
  }
}
```

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.

```
\clist_new:N \g_auf_schluessel_clist
\clist_set:Nn \g_auf_schluessel_clist {
```

```
  titel,
  thematik,
  stichwoerter,
  zitat_schluessel,
  zitat_beschreibung,
  %
  bearbeitungs_stand,
  korrektheit,
  %
  relativer_pfad,
  %
  examen_nummer,
  examen_jahr,
  examen_monat,
  examen_thema_nr,
  examen_teilaufgabe_nr,
  examen_aufgabe_nr,
}
```

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

```
\clist_map_inline:Nn \g_auf_schluessel_clist {
  \tl_new:c {g_auf_#1_tl}
}
```

Funktion über alle globalen Token-List-Variablen zurückzusetzten.

```
\cs_new:Npn \_setze_variablen_zurueck: {
  \clist_map_inline:Nn \g_auf_schluessel_clist {
    \tl_clear:c {g_auf_##1_tl}
  }
}
```

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

```
\keys_define:nn { aufgabenmetadaten }
{
  Titel .tl_gset:N = \g_auf_titel_tl,
  Thematik .tl_gset:N = \g_auf_thematik_tl,
  Stichwoerter .tl_gset:N = \g_auf_stichwoerter_tl,
  ZitatSchluessel .tl_gset:N = \g_auf_zitat_beschreibung_tl,
  ZitatBeschreibung .tl_gset:N = \g_auf_zitat_schluessel_tl,
  %
  BearbeitungsStand .tl_gset:N = \g_auf_bearbeitungs_stand_tl,
  Korrektheit .tl_gset:N = \g_auf_korrektheit_tl,
  %
  RelativerPfad .tl_gset:N = \g_auf_relativer_pfad_tl,
  %
  ExamenNummer .tl_gset:N = \g_auf_examen_nummer_tl,
  ExamenJahr .tl_gset:N = \g_auf_examen_jahr_tl,
  ExamenMonat .tl_gset:N = \g_auf_examen_monat_tl,
  ExamenThemaNr .tl_gset:N = \g_auf_examen_thema_nr_tl,
  ExamenTeilaufgabeNr .tl_gset:N = \g_auf_examen_teilaufgabe_nr_tl,
  ExamenAufgabeNr .tl_gset:N = \g_auf_examen_aufgabe_nr_tl,
}
```

```
\cs_gset:Npn \_setze_relativen_pfad: {
  \tl_if_empty:NTF \g_auf_relativer_pfad_tl
  {
    \bool_if:nTF
    {
      ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
      ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
      ! \tl_if_empty_p:N \g_auf_examen_monat_tl
    }
    {
```

```
      \tl_gset:Nn \g_auf_relativer_pfad_tl {
        Staatsexamen /
        \g_auf_examen_nummer_tl /
        \g_auf_examen_jahr_tl /
        \g_auf_examen_monat_tl /
        \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} { Thema - \g_auf_examen_thema_nr_tl / 
        \tl_if_empty:NTF \g_auf_examen_teilaufgabe_nr_tl {} { Teilaufgabe - \g_auf_examen_tei
        \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} { Aufgabe - \g_auf_examen_aufgabe_nr_
      }
    }
    {}
  }
  {}
}
\cs_gset:Npn \_gib_examen_titel: {
  \cs_set:Nn \_trenner: { \, / \, }
  \bool_if:nTF
  {
    ! \tl_if_empty_p:N \g_auf_examen_nummer_tl &&
    ! \tl_if_empty_p:N \g_auf_examen_jahr_tl &&
    ! \tl_if_empty_p:N \g_auf_examen_monat_tl &&
    ! \tl_if_empty_p:N \g_auf_examen_aufgabe_nr_tl
  }
  {
    {
      \footnotesize
      \par
      \noindent
      Staatsexamen ~
      \g_auf_examen_nummer_tl \_trenner:
      \g_auf_examen_jahr_tl \_trenner:

      \tl_case:Nn \g_auf_examen_monat_tl
      {
        { 03 } { Frühjahr }
        { 09 } { Herbst }
      } \_trenner:

      \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} {
        Thema ~ Nr. ~ \g_auf_examen_thema_nr_tl \_trenner:
      }
      \tl_if_empty:NTF \g_auf_examen_teilaufgabe_nr_tl {} {
        Teilaufgabe ~ Nr. ~ \g_auf_examen_teilaufgabe_nr_tl \_trenner:
      }
      \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} {
        Aufgabe ~ Nr. ~ \g_auf_examen_aufgabe_nr_tl
      }
      \par
      \bigskip
    }
  }
}
\cs_new:Npn \_gib_github_url: {
  \tl_if_empty:NTF \g_auf_relativer_pfad_tl {} {
    Github :~\href{
      \LehramtInformatikGithubRawDomain /
      \LehramtInformatikGithubTexRepo /
      \LehramtInformatikGitBranch /
      \g_auf_relativer_pfad_tl
    }{
      \nolinkurl{\g_auf_relativer_pfad_tl}
    }
  }
```

```
}
\cs_new:Npn \_gib_aufgaben_titel: {
  \g_auf_titel_tl

  \tl_if_empty:NTF \g_auf_thematik_tl
  {}
  {
   \, ~ [
    \g_auf_thematik_tl
   ]
  }
}
\msg_new:nnn { aufgabenmetadaten } { kein-titel }
  { Der~Schlüssel~ist~zwingend~notwendig }
```

\def\LehramtInformatikRepository{/pfad/zum/repository}

```
\input /etc/lehramt-informatik.config.tex
```

biblatex not working with lualatex and babel

```
% \RequirePackage{polyglossia}
% \setmainlanguage{german}
```

2.6 baum.sty

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

```
\RequirePackage{tikz}
```

für `li binaer baum`

```
\RequirePackage{tikz-qtree}
```

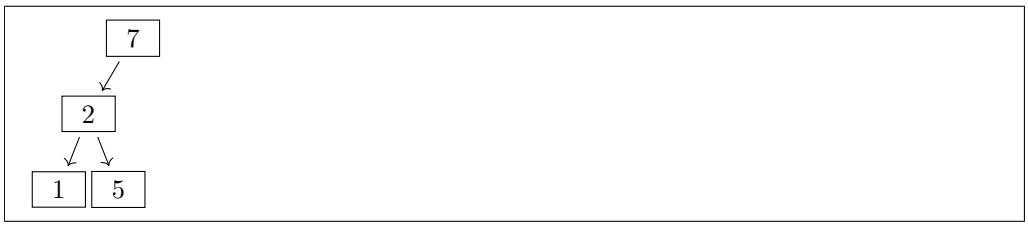
Für b baum

```
\usetikzlibrary{shapes.multipart}
```

2.6.1 Binärbaum

TikZ-Stil: `li binaer baum`: Knoten als Kreise, Kanten als Pfeile.

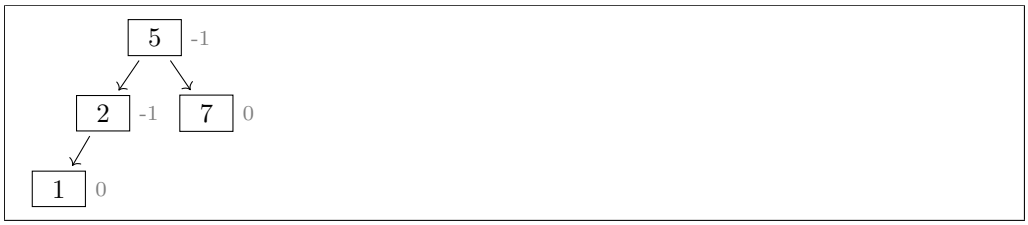
```
\begin{tikzpicture}[li binaer baum]
\Tree
[.7
  [.2
    [.1 ]
    [.5 ]
  ]
  \edge[blank]; \node[blank]{};
]
\end{tikzpicture}
```




```
\tikzset{
  li binaer baum/.style={
    shorten <=2pt,
    shorten >=2pt,
    ->,
    every tree node/.style={
      minimum width=2em,
      draw,
      rectangle
    },
    blank/.style={
      draw=none
    },
    edge from parent/.style={
      draw,
      edge from parent path={(\tikzparentnode) -- (\tikzchildnode)}
    },
    level distance=1cm,
    every label/.style={
      gray,
      font=\footnotesize,
      label position=0,
      label distance=0cm,
    }
  },
}
```

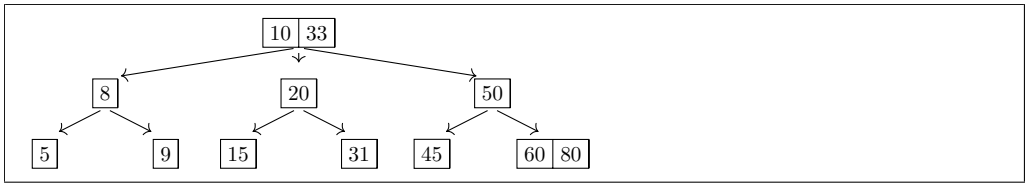
2.6.2 AVL-Baum

```
\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}
```




```
\tikzset{
  li bbaum knoten/.style={
    rectangle split parts=10,
    rectangle split,
    rectangle split horizontal,
    rectangle split ignore empty parts,
    draw,
    fill=white
  },
  li bbaum/.style={
    every node/.style={
      li bbaum knoten
    },
    level 1/.style={
      level distance=12mm,
      sibling distance=25mm,
    },
    every child/.style={
      shorten <= 2pt,
      shorten >= 6pt,
      ->,
    },
    level 2/.style={
      level distance=9mm,
      sibling distance=15mm,
    },
  }
}
```

2.7 checkbox.sty

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

`\liRichtig` Angekreuztes Kästchen (nur innerhalb der `itemize`-Umgebung zu verwenden.

```
\def\liRichtig{\item[$\boxtimes$]}
```

`\liFalsch` Nicht angekreuztes Kästchen (nur innerhalb der `itemize`-Umgebung zu verwenden.

```
\def\liFalsch{\item[$\square$]}
```

2.8 chomsky-normalform.sty

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

\ExplSyntaxOn

\liLadePakete{typographie}
```

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

```
\def\liChomskyUeberschrift#1{
  {
    \bfseries
    \sffamily
    \str_case:nn {#1} {
      {1} {Elimination~der~$\varepsilon$-Regeln}
      {2} {Elimination~von~Kettenregeln}
      {3} {Separation~von~Terminalzeichen}
      {4} {Elimination~von~mehrelementigen~Nonterminalketten}
    }
  }
}
```

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

```
\def\li@chomsky@erklaerung@texte#1{
  \str_case:nn {#1} {
    %
    {1} {
      Alle~Regeln~der~Form~$A~\rightarrow~\varepsilon$~werden~eliminiert.~
      Die~Ersetzung~von~$A$~wird~durch~$\varepsilon$~in~allen~anderen~
      Regeln~vorweggenommen.
    }
    {2} {
      Jede~Produktion~der~Form~$A~\rightarrow~B$~mit~$A,~B~\in~S$~wird~
      als~Kettenregel~bezeichnet.~Diese~tragen~nicht~zur~Produktion~
      von~Terminalzeichen~bei~und~lassen~sich~ebenfalls~eliminieren.
    }
    {3} {
      Jedes~Terminalzeichen~$\sigma$,~das~in~Kombination~mit~anderen~
      Symbolen~auftaucht,~wird~durch~ein~neues~Nonterminal~
      $S\sb{\sigma}$~ersetzt~und~die~Menge~der~Produktionen~durch~die~
      Regel~$S\sb{\sigma}~\rightarrow~\sigma$~ergänzt.
    }
    {4} {
      Alle~Produktionen~der~Form~
      $A\rightarrow~B\sb{1}~B\sb{2}~\dots~B\sb{n}$~
      werden~in~die~Produktionen~
      $A~\rightarrow~
      A\sb{n-1}~B\sb{n},~A\sb{n-1}~\rightarrow~
      A\sb{n-2}~B\sb{n-1},~\dots,~
      A\sb{2}~\rightarrow~B\sb{1}~B\sb{2}$~zerteilt.~
      Nach~der~Ersetzung~sind~alle~längeren~Nonterminalketten~
      vollständig~heruntergebrochen~und~die~Chomsky-Normalform~erreicht.
    }
  }
}
\def\liChomskyErklaerung#1{
  {
    \itshape
    \footnotesize
    \liParagraphMitLinien{\li@chomsky@erklaerung@texte{#1}}
  }
```

```
}
```

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

```
\def\liChomskyUeberErklaerung#1{
  \liChomskyUeberschrift{#1}\par
  \liChomskyErklaerung{#1}
}

\ExplSyntaxOff

```

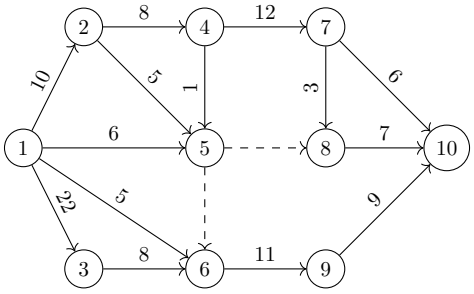
2.9 cpm.sty

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

```
\liLadePakete{mathe}
```

2.9.1 Makro-Kürzel

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




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

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

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


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

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

```
\ExplSyntaxOn
\NewDocumentCommand { \liCpmEreignis } { O{} m m m  } {
  \tl_set:Nn \l_name_tl {}

  \keys_define:nn { cpmEreignis } {
    name .code:n = {\tl_set:Nn \l_name_tl {##1}},
  }

  \keys_set:nn { cpmEreignis } { #1 }

  \tl_if_empty:NT \l_name_tl {
    \tl_set:Nn \l_name_tl {#2}
  }
```

```
  \node[circle,draw] (\l_name_tl) at (#3,#4) {#2};
}
\ExplSyntaxOff
```

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

```
\ExplSyntaxOn
\NewDocumentCommand { \liCpmVorgang } { O{} m m m  } {
  \tl_set:Nn \l_schein_tl {}
  \tl_set:Nn \l_kritisch_tl {}

  \keys_define:nn { cpmVorgang } {
    schein .code:n = {\tl_set:Nn \l_schein_tl {dashed}},
    kritisch .code:n = {\tl_set:Nn \l_kritisch_tl {thick}},
  }

  \keys_set:nn { cpmVorgang } { #1 }

  \path[->,\l_schein_tl,\l_kritisch_tl] (#2) edge node[auto,sloped] {#4} (#3);
}
\ExplSyntaxOff
```

2.9.2 Hilfsmakros für Zeitpunkt-Tabelle

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

\liCpmVonZu **Let-Abkürzung:** \let\vz=\liCpmVonZu
\liCpmVonZu{1}(2-3): $1_{(2\to 3)}$

```
\def\liCpmVonZuOhneMathe#1(#2-#3){#1_{\scriptscriptstyle(#2\rightarrow#3)}}
\def\liCpmVonZu#1(#2-#3){%
  \ifmmode%
    \liCpmVonZuOhneMathe{#1}(#2-#3)%
  \else%
    $\liCpmVonZuOhneMathe{#1}(#2-#3)$%
  \fi%
}
```

\liCpmVon **Let-Abkürzung:** \let\v=\liCpmVon
\liCpmVon{1}(2): $1_{(\to 2)}$

```
\def\liCpmVonOhneMathe#1(#2){#1_{\scriptscriptstyle(\rightarrow#2)}}
\def\liCpmVon#1(#2){%
  \ifmmode%
    \liCpmVonOhneMathe{#1}(#2)%
  \else%
    $\liCpmVonOhneMathe{#1}(#2)$%
  \fi%
}
```

\liCpmZu **Let-Abkürzung:** \let\z=\liCpmZu
\liCpmZu{1}(2): $1_{(\leftarrow 2)}$

```
\def\liCpmZuOhneMathe#1(#2){#1_{\scriptscriptstyle(\leftarrow#2)}}
\def\liCpmZu#1(#2){%
  \ifmmode%
    \liCpmZuOhneMathe{#1}(#2)%
  \else%
    $\liCpmZuOhneMathe{#1}(#2)$%
```

```
  \fi%
}
```

`\liCpmSpaetesterI` Spätester Zeitpunkt, zu dem Ereignis i eintreten kann
Let-Abkürzung: `\let\SZ=\liCpmSpaetesterI`

```
\def\liCpmSpaetesterI{$SZ_i$}
```

`\liCpmFruehesterI` Frühester Zeitpunkt, zu dem Ereignis i eintreten kann.
Let-Abkürzung: `\let\FZ=\liCpmFruehesterI`

```
\def\liCpmFruehesterI{$FZ_i$}
```

2.10 cyk-algorithmus.sty

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

```
\def\liKurzeTabellenLinie#1{\\\cline{1-#1}}
```

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

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

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

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

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

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

2.11 entwurfsmuster.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-entwurfsmuster}[2021/05/06
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. `Uml`: Uml-Klassendiagramm `\liEntwurfsEinzelstueckUml`
2. `Akteure`: Akteure, beteiligte Klassen `\liEntwurfsEinzelstueckAkteure`
3. `Code`: Allgemeines Code-Beispiel `\liEntwurfsEinzelstueckCode`
4. `ohne`: Ohne Suffix, Bündelung der einzelnen Makros eines Entwurfsmusters `\liEntwurfsEinzelstueckAkteure`

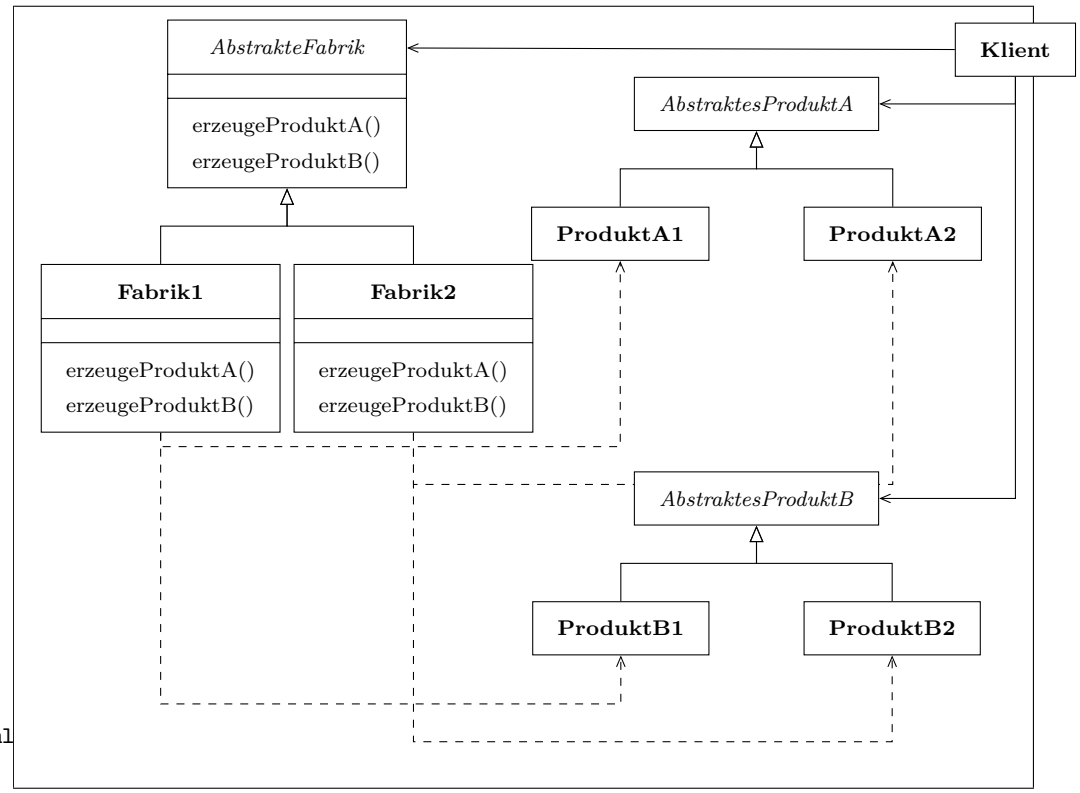
```
\RequirePackage{lehramt-informatik-uml}
```

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

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

2.11.3 Abstrakte Fabrik (Abstract Factory)

`\liEntwurfsAbstrakteFabrikUml`




```
\def\liEntwurfsAbstrakteFabrikUml{
  \begin{tikzpicture}
    \umlclass[type=abstract]{AbstrakteFabrik}{}{
      erzeugeProduktA()\\
      erzeugeProduktB()\\
    }
```

```
    \umlclass[below left=1cm and -1.5cm of AbstrakteFabrik]{Fabrik1}{}{
      erzeugeProduktA()\\
      erzeugeProduktB()\\
    }
    \umlclass[below right=1cm and -1.5cm of AbstrakteFabrik]{Fabrik2}{}{
      erzeugeProduktA()\\
      erzeugeProduktB()\\
    }
    \umlVHVinherit{Fabrik1}{AbstrakteFabrik}
    \umlVHVinherit{Fabrik2}{AbstrakteFabrik}

    \umlsimpleclass[right=3cm of AbstrakteFabrik,type=abstract]{AbstraktesProduktA}
    \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktA]{ProduktA1}
    \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktA]{ProduktA2}
    \umlVHVinherit{ProduktA1}{AbstraktesProduktA}
    \umlVHVinherit{ProduktA2}{AbstraktesProduktA}

    \umlsimpleclass[above right=0cm and 1cm of AbstraktesProduktA]{Klient}

    \umlsimpleclass[below=4.5cm of AbstraktesProduktA,type=abstract]{AbstraktesProduktB}
    \umlsimpleclass[below left=1cm and -1cm of AbstraktesProduktB]{ProduktB1}
    \umlsimpleclass[below right=1cm and -1cm of AbstraktesProduktB]{ProduktB2}
    \umlVHVinherit{ProduktB1}{AbstraktesProduktB}
    \umlVHVinherit{ProduktB2}{AbstraktesProduktB}

    \umlVHVdep[arm1=-1.3cm,arm2=-1cm]{Fabrik1}{ProduktA1}
    \umlVHVdep[arm2=-1cm]{Fabrik1}{ProduktB1}

    \umlVHVdep[arm1=-1.8cm,arm2=-1.5cm]{Fabrik2}{ProduktA2}
    \umlVHVdep[arm2=-1.5cm]{Fabrik2}{ProduktB2}

    \umluniassoc[anchor2=25]{Klient}{AbstrakteFabrik}
    \umlVHuniassoc[arm1=-1cm]{Klient}{AbstraktesProduktA}
    \umlVHuniassoc{Klient}{AbstraktesProduktB}
  \end{tikzpicture}
}
```

iEntwurfsAbstrakteFabrikCode

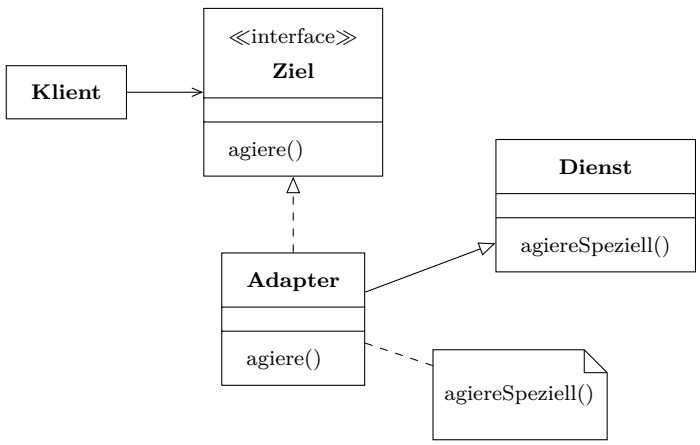
```
\def\liEntwurfsAbstrakteFabrikCode{
  \li@EntwurfsCode{abstrakte_fabrik}{Produkte}
  \li@EntwurfsCode{abstrakte_fabrik}{AbstrakteFabrik}
  \li@EntwurfsCode{abstrakte_fabrik}{Klient}
}
```

\liEntwurfsAbstrakteFabrik

```
\def\liEntwurfsAbstrakteFabrik{
  \liEntwurfsAbstrakteFabrikUml
  \liEntwurfsAbstrakteFabrikCode
}
```

2.11.4 Adapter

\liEntwurfsAdapterUml




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

\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.

```
706 \def\liEntwurfsAdapterAkteure{
707   \begin{description}
708
709     \item[Ziel (Target)]
710
711     Das Ziel definiert die Schnittstelle, die der Klient nutzen kann.
712
713     \item[Klient (Client)]
714
715     Der Klient nutzt Dienste über inkompatible Schnittstellen und greift
716     dabei auf adaptierte Schnittstellen zurück.
717
718     \item[Dienst (Adaptee)]
719
720     Der Dienst bietet wiederzuverwendende Dienstleistungen mit fest
721     definierter Schnittstelle an.
722
723     \item[Adapter]
724
725     Der Adapter adaptiert die Schnittstelle des Dienstes auf die
726     Schnittstelle zum Klienten.\footcite{wiki:adapter}
```

```
  \end{description}
}
```

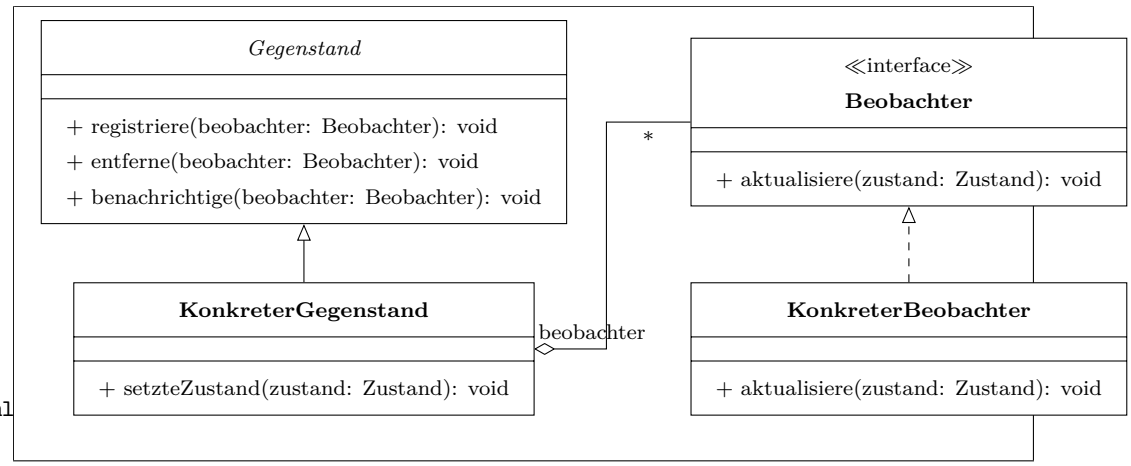
\liEntwurfsAdapterCode

```
\def\liEntwurfsAdapterCode{
  \li@EntwurfsCode{adapter}{Dienst}
  \li@EntwurfsCode{adapter}{Ziel}
  \li@EntwurfsCode{adapter}{Adapter}
  \li@EntwurfsCode{adapter}{Klient}
}
```

\liEntwurfsAdapter

```
\def\liEntwurfsAdapter{
  \liEntwurfsAdapterUml
  \liEntwurfsAdapterAkteure
  \liEntwurfsAdapterCode
}
```

2.11.5 Beobachter (Observer)



\liEntwurfsBeobachterUml

```
\def\liEntwurfsBeobachterUml{
  \begin{tikzpicture}
    \umlclass[x=0,y=0,type=abstract]{Gegenstand}{}{
      + registriere(beobachter: Beobachter): void\\
      + entferne(beobachter: Beobachter): void\\
      + benachrichtige(beobachter: Beobachter): void\\
    }
    \umlclass[x=0,y=-3]{KonkreterGegenstand}{}{
      + setzteZustand(zustand: Zustand): void
    }
    \umlinherit{KonkreterGegenstand}{Gegenstand}

    \umlclass[x=8,y=0,type=interface]{Beobachter}{}{
      + aktualisiere(zustand: Zustand): void
    }
    \umlclass[x=8,y=-3]{KonkreterBeobachter}{}{
      + aktualisiere(zustand: Zustand): void
    }
    \umlreal{KonkreterBeobachter}{Beobachter}

    \umlHVHaggreg[arg1=beobachter,pos1=0.8,mult2=*,pos2=2.5]
    {KonkreterGegenstand}{Beobachter}
  \end{tikzpicture}
}
```

\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.

```
\def\liEntwurfsBeobachterAkteure{
  \begin{description}
    \item[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.\footcite[Seite
    251]{gof}

    \item[Beobachter (Observer)]

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

    \item[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.

    \item[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.
    \footcite{wiki:beobachter}
  \end{description}
}
```

\liEntwurfsBeobachterCode

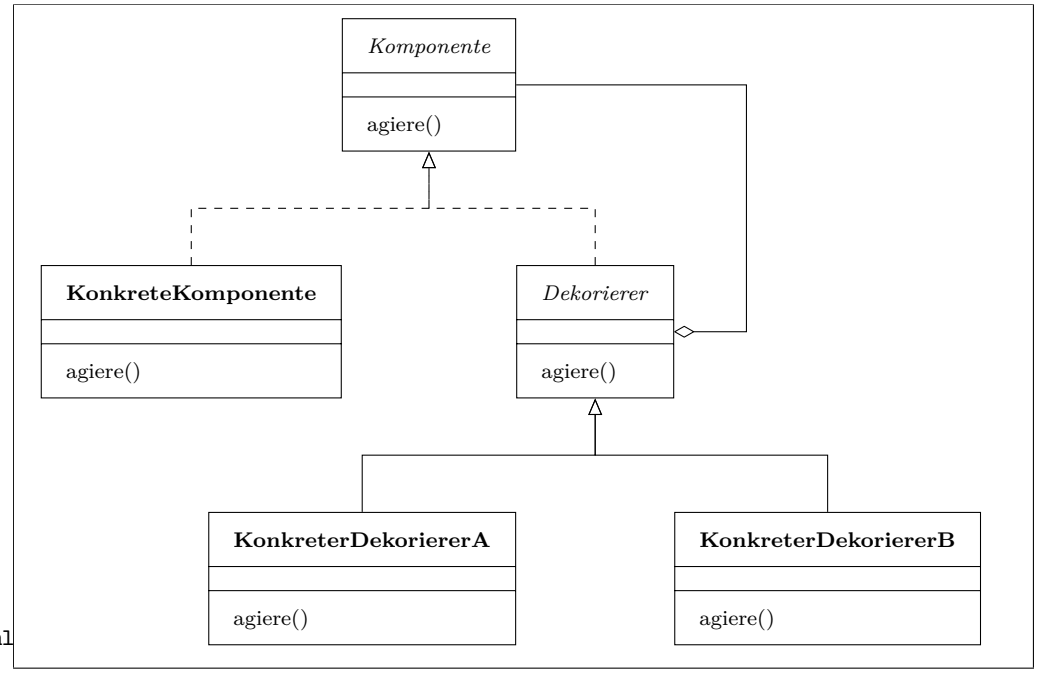
```
\def\liEntwurfsBeobachterCode{
  \li@EntwurfsCode{beobachter}{Gegenstand}
  \li@EntwurfsCode{beobachter}{KonkreterGegenstand}
  \li@EntwurfsCode{beobachter}{Beobachter}
  \li@EntwurfsCode{beobachter}{KonkreterBeobachterA}
  \li@EntwurfsCode{beobachter}{KonkreterBeobachterB}
```

```
  \li@EntwurfsCode{beobachter}{Klient}
}
```

\liEntwurfsBeobachter

```
\def\liEntwurfsBeobachter{
  \liEntwurfsBeobachterUml
  \liEntwurfsBeobachterAkteure
  \liEntwurfsBeobachterCode
}
```

2.11.6 Dekorierer (Decorator)



\liEntwurfsDekoriererUml

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

    \umlVHVreal{KonkreteKomponente}{Komponente}
    \umlVHVreal{Dekorierer}{Komponente}

    \umlclass[below left=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererA}{}{agiere()}
    \umlclass[below right=1.5cm and 0cm of Dekorierer]{KonkreterDekoriererB}{}{agiere()}

    \umlVHVinherit{KonkreterDekoriererA}{Dekorierer}
    \umlVHVinherit{KonkreterDekoriererB}{Dekorierer}

    \umlHVHaggreg[arm1=2cm]{Dekorierer}{Komponente}
    \footcite{wiki:dekorierer}
  \end{tikzpicture}
}
```

\liEntwurfsDekoriererCode

```
\def\liEntwurfsDekoriererCode{
  \li@EntwurfsCode{dekorierer}{Komponente}
  \li@EntwurfsCode{dekorierer}{KonkreteKomponente}
  \li@EntwurfsCode{dekorierer}{Dekorierer}
  \li@EntwurfsCode{dekorierer}{KonkreterDekoriererA}
  \li@EntwurfsCode{dekorierer}{KonkreterDekoriererB}
  \li@EntwurfsCode{dekorierer}{Klient}
}
```

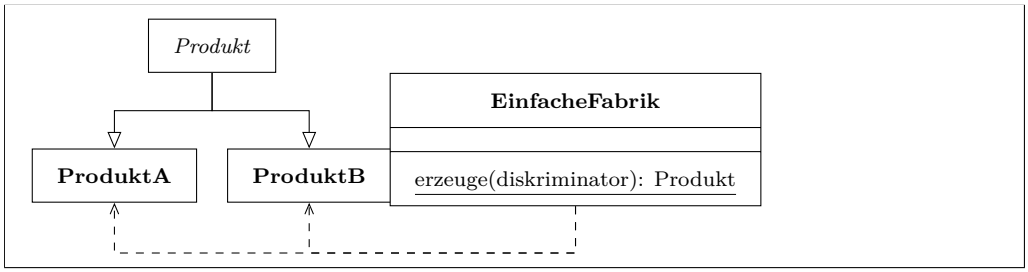
\liEntwurfsDekorierer

```
\def\liEntwurfsDekorierer{
  \liEntwurfsDekoriererUml
  \liEntwurfsDekoriererAkteure
  \liEntwurfsDekoriererCode
}
```

2.11.7 Einfache Fabrik (Simple Factory)

\liEntwurfsEinfacheFabrikUml

Quelle: https://refactoring.guru/design-patterns/factory-comparison




```
\def\liEntwurfsEinfacheFabrikUml{
  \begin{tikzpicture}
    \umlsimpleclass[type=abstract]{Produkt}
    \umlsimpleclass[below left=1cm and -0.65cm of Produkt]{ProduktA}
    \umlsimpleclass[below right=1cm and -0.65cm of Produkt]{ProduktB}
    \umlVHVinherit{Produkt}{ProduktA}
    \umlVHVinherit{Produkt}{ProduktB}
    \umlclass[below right=0cm and 1.5cm of Produkt]{EinfacheFabrik}{
    }{
      \umlstatic{erzeuge(diskriminator): Produkt}\\
    }
    \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktA}
    \umlVHVdep[arm1=-1.5cm]{EinfacheFabrik}{ProduktB}
  \end{tikzpicture}
}
```

ntwurfsEinfacheFabrikAkteure

Quelle: frei überstetzt aus GoF

EinfacheFabrik Eine Klasse mit einer Erzeugunsmethode, 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.

```
\def\liEntwurfsEinfacheFabrikAkteure{
  \begin{description}
    \item[EinfacheFabrik]

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

    \item[Produkt]

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

    \item[KonkretesProdukt]

    Ein konkretes Produkt, das von der einfachen Fabrik erzeugt wird.
  \end{description}
}
```

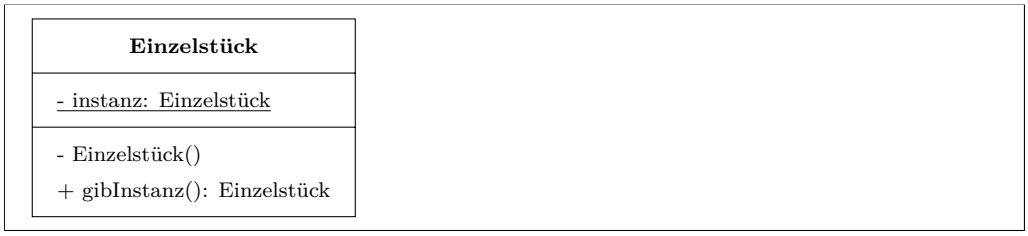
\liEntwurfsEinfacheFabrik

```
\def\liEntwurfsEinfacheFabrik{
  \liEntwurfsEinfacheFabrikUml
  \liEntwurfsEinfacheFabrikAkteure
}
```

2.11.8 Einzelstück (Singleton)

\liEntwurfsEinzelstueckUml

Quelle nach der deutschen Wikipedia




```
\def\liEntwurfsEinzelstueckUml{
  \begin{tikzpicture}
    \umlclass{Einzelstück}{
    \umlstatic{- instanz: Einzelstück}\\
    }{
    - Einzelstück()\\
    + gibInstanz(): Einzelstück
    }
  \end{tikzpicture}
}
```

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.

```
\def\liEntwurfsEinzelstueckAkteure{
  \begin{description}
    \item[Einzelstück (Singleton)]

    stellt eine statische Methode bereit, mit deren Hilfe die Klienten
    nur auf eine einzige Instanz der Klasse zugreifen können.
  \end{description}
}
```

\liEntwurfsEinzelstueckCode

```
\def\liEntwurfsEinzelstueckCode{
  \li@EntwurfsCode{einzelstueck}{Einzelstueck}
}
```

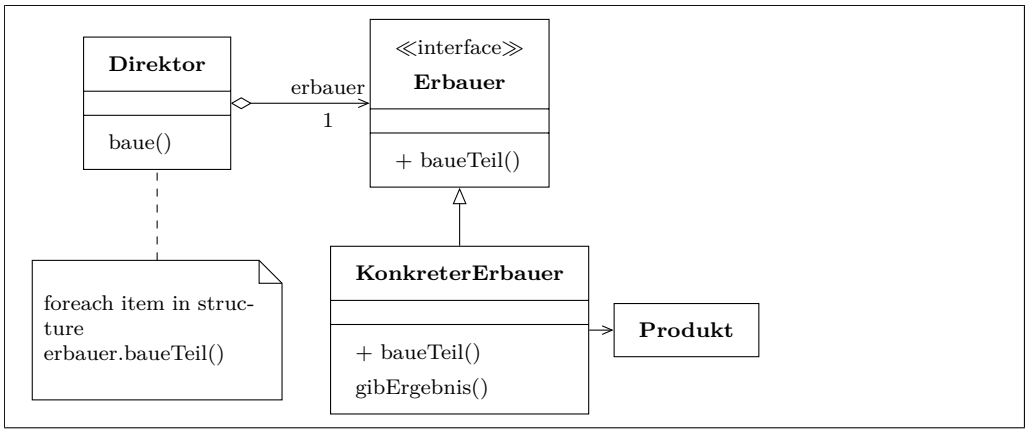
\liEntwurfsEinzelstueck

```
\def\liEntwurfsEinzelstueck{
  \liEntwurfsEinzelstueckUml
  \liEntwurfsEinzelstueckAkteure
  \liEntwurfsEinzelstueckCode
}
```

2.11.9 Erbauer (Builder)

\liEntwurfsErbauerUml

Quelle nach der deutschen Wikipedia




```
\def\liEntwurfsErbauerUml{
  \begin{tikzpicture}
    \umlsimpleclass[x=7,y=0]{Produkt}{}{}
    \umlclass[x=4,y=3,type=interface]{Erbauer}{}{+ baueTeil()}
    \umlclass[x=4,y=0]{KonkreterErbauer}{}{
      + baueTeil()\\
      gibErgebnis()}
    \umlclass[x=0,y=3]{Direktor}{}{baue()}

    \umluniaggreg[arg2=erbauer,pos2=0.7,mult2=1]{Direktor}{Erbauer}
    \umluniassoc{KonkreterErbauer}{Produkt}
    \umlinherit{KonkreterErbauer}{Erbauer}

  \umlnote[x=0,y=0,width=3cm]{Direktor}{
    foreach item in structure\\
    erbauer.baueTeil()
  }
  \end{tikzpicture}
  \footcite{wiki:erbauer}
}
```

`\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.

```
\def\liEntwurfsErbauerAkteure{
  \begin{description}
    \item[Erbauer]

    Der Erbauer spezifiziert eine abstrakte Schnittstelle zur Erzeugung der
    Teile eines komplexen Objektes.

    \item[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.

    \item[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.

    \item[Produkt]

    Das Produkt repräsentiert das zu konstruierende komplexe Objekt.
    \footcite{wiki:erbauer}
  \end{description}
}
```

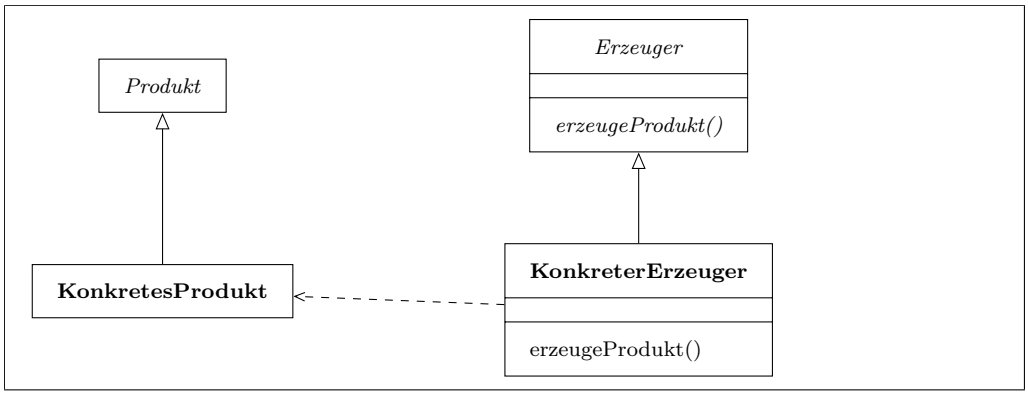
\liEntwurfsErbauer

```
\def\liEntwurfsErbauer{
  \liEntwurfsErbauerUml
  \liEntwurfsErbauerAkteure
}
```

2.11.10 Fabrikmethode (Factory Method)

\liEntwurfsFabrikmethodeUml

Quelle nach der deutschen Wikipedia




```
\def\liEntwurfsFabrikmethodeUml{
  \begin{tikzpicture}
    \umlsimpleclass[type=abstract]{Produkt}
    \umlsimpleclass[below=2cm of Produkt]{KonkretesProdukt}
    \umlinherit{KonkretesProdukt}{Produkt}

    \umlclass[type=abstract,right=4cm of Produkt]{Erzeuger}{}{
      \textit{erzeugeProdukt()}\\
    }
    \umlclass[below=1.2cm of Erzeuger]{KonkreterErzeuger}{}{
    erzeugeProdukt()
    }
    \umlinherit{KonkreterErzeuger}{Erzeuger}

    \umldep{KonkreterErzeuger}{KonkretesProdukt}
  \end{tikzpicture}
}
```

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.

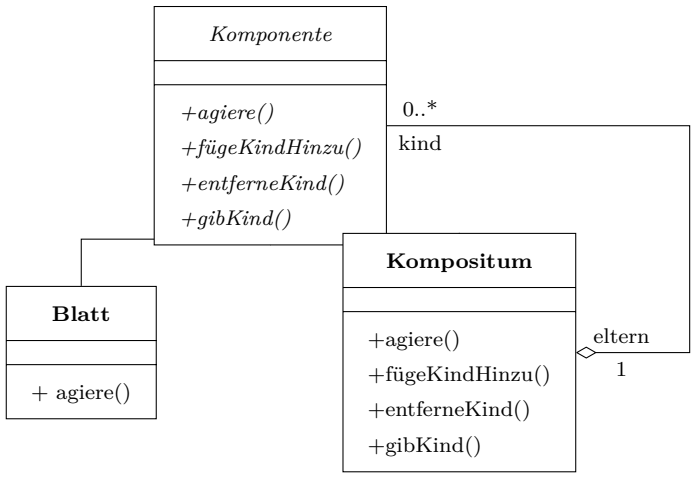
KonkreterErzeuger KonkreterErzeuger überschreibt die Fabrikmethode, um die ihm entsprechenden konkreten Produkte zu erzeugen (z. B. indem er den Konstruktor einer konkreten Produkt-Klasse aufruft).

```
\def\liEntwurfsFabrikmethodeAkteure{
  \begin{description}
    \item[Produkt]

    Das Produkt ist der Basistyp (Klasse oder Schnittstelle) für das
    zu erzeugende Produkt.

    \item[KonkretesProdukt]

    KonkretesProdukt implementiert die Produkt-Schnittstelle.

    \item[Erzeuger]

    Der Erzeuger deklariert die Fabrikmethode, um ein solches Produkt
    zu erzeugen und kann eine Default-Implementierung beinhalten.

    \item[KonkreterErzeuger]

    KonkreterErzeuger überschreibt die Fabrikmethode, um die ihm
    entsprechenden konkreten Produkte zu erzeugen (z. B. indem er den
    Konstruktor einer konkreten Produkt-Klasse aufruft).

    \footcite{wiki:fabrikmethode}
  \end{description}
}
```

`\liEntwurfsFabrikmethode`

```
\def\liEntwurfsFabrikmethode{
  \liEntwurfsFabrikmethodeUml
  \liEntwurfsFabrikmethodeAkteure
}
```

2.11.11 Kompositum (Composite)



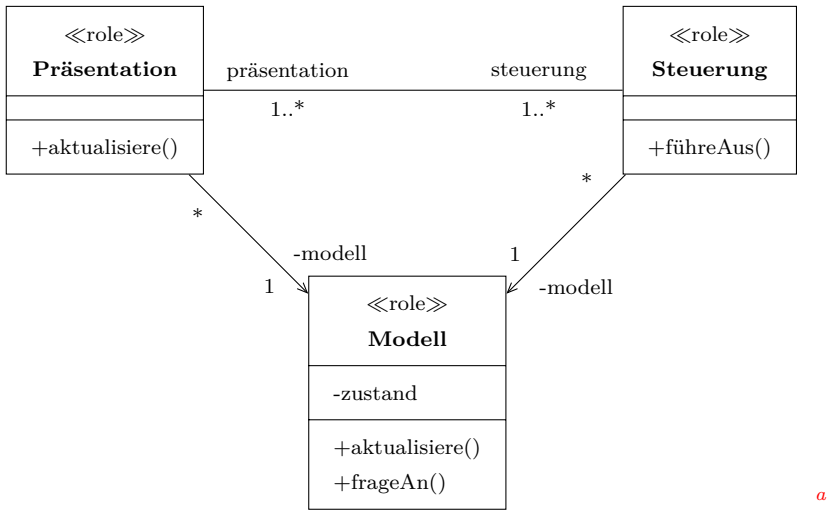
`\liEntwurfsKompositumUml`

```
\def\liEntwurfsKompositumUml{
  \begin{tikzpicture}
    \umlclass[x=2.5,y=3,type=abstract]{Komponente}{}{
      \textit{+agiere()}\\
      \textit{+fügeKindHinzu()}\\
      \textit{+entferneKind()}\\
      \textit{+gibKind()}
    }
    \umlclass[x=0]{Blatt}{}{+ agiere()}
    \umlclass[x=5]{Kompositum}{}{
      +agiere()\\
      +fügeKindHinzu()\\
      +entferneKind()\\
      +gibKind()
    }

    \umlVHVinherit{Kompositum}{Komponente}
    \umlVHVinherit{Blatt}{Komponente}
    \umlHVHaggreg[anchor1=east,arm1=1.5cm,arg1=eltern,mult1=1,arg2=kind,mult2=0..*,pos2=2.9,p
  \end{tikzpicture}
}
```

\liEntwurfsFabrikmethode

```
\def\liEntwurfsKompositum{
  \liEntwurfsKompositumUml
  \liEntwurfsKompositumAkteure
}
```

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



\liEntwurfsModellPraesentationSteuerungUml

[a] http://www.cs.sjsu.edu/~pearce/modules/lectures/ood2/mvc/index.htm

```
\def\liEntwurfsModellPraesentationSteuerungUml{
  \begin{tikzpicture}
    \umlclass[x=-4cm,type=role]{Präsentation}{}{+aktualisiere()}
    \umlclass[x=4cm,type=role]{Steuerung}{}{+führeAus()}
    \umlclass[y=-4cm,type=role]{Modell}{
      -zustand
    }{
      +aktualisiere()\\
      +frageAn()
    }

    \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Präsentation}{Modell}
    \umluniassoc[arg2=-modell,mult2=1,mult1=*]{Steuerung}{Modell}
```

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

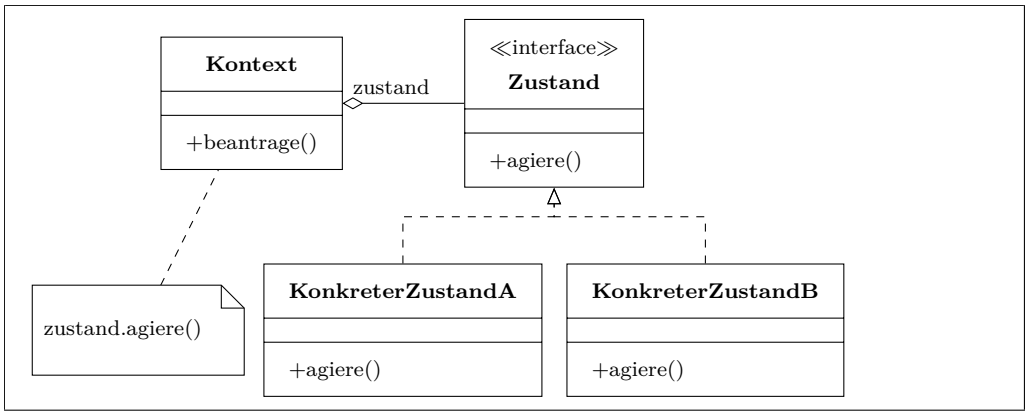
ModellPraesentationSteuerung

```
\def\liEntwurfs{
  \liEntwurfsModellPraesentationSteuerungUml
  \liEntwurfsModellPraesentationSteuerungAkteure
}
```

2.11.13 Zustand (State)

\liEntwurfsZustandUml

Quelle: Englische Wikipedia, so ähnlich wie in GoF




```
\def\liEntwurfsZustandUml{
  \begin{tikzpicture}
    \umlclass[x=-1,y=3]{Kontext}{}{+beantrage()}
    \umlclass[x=3,y=3,type=interface]{Zustand}{}{+agiere()}
    \umlclass[x=1,y=0]{KonkreterZustandA}{}{+agiere()}
    \umlclass[x=5,y=0]{KonkreterZustandB}{}{+agiere()}

    \umlVHVreal{KonkreterZustandA}{Zustand}
    \umlVHVreal{KonkreterZustandB}{Zustand}

    \umlaggreg[arg=zustand,pos=0.4]{Kontext}{Zustand}

    \umlnote[x=-2.5,y=0,width=2.5cm]{Kontext}{zustand.agiere()}
  \end{tikzpicture}
}
```

\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.

```
\def\liEntwurfsZustandAkteure{
  \begin{description}
    \item[Kontext (Context)]

    definiert die clientseitige Schnittstelle und verwaltet die separaten
    Zustandsklassen.

```

```
    \item[State (Zustand)]

    definiert eine einheitliche Schnittstelle aller Zustandsobjekte und
    implementiert gegebenenfalls ein Standardverhalten.

    \item[KontreterZustand (ConcreteState)]

    implementiert das Verhalten, das mit dem Zustand des Kontextobjektes
    verbunden ist.
  \end{description}
}
```

\liEntwurfsZustand

```
\def\liEntwurfsZustand{
  \liEntwurfsZustandUml
  \liEntwurfsZustandAkteure
}
```

2.12 er.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-er}[2020/06/13 Zum Setzen von
ER-Diagrammen]
\RequirePackage{tikz-er2}
\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);
```

```
\RequirePackage{soul}
\RequirePackage{fontawesome}
```

2.12.3 Makro-Kürzel

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

```
\ExplSyntaxOn
```

`\liErEntity`

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

`\liErRelationship`

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

`\liErAttribute`

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

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

```
\def\liErMpEntity#1{
  \liErEntity{#1}
  \marginpar{
    \liErEntity{\tiny\faSquareO{}~E:~#1}
  }
}
```

□

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

```
\def\liErMpRelationship#1{
  \liErRelationship{#1}
  \marginpar{
    \liErRelationship{\tiny\faGg{}~R:~#1}
  }
}
```

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

```
\def\liErMpAttribute#1{
  \liErAttribute{#1}
  \marginpar{
    \liErAttribute{\tiny\faCircleThin{}~A:~#1}
  }
}
```

`\liErDatenbankName` **Let-Abkürzung:** `\let\d=\liErDatenbankName`

datenbank name

```
\def\liErDatenbankName#1{
  {
    \footnotesize\texttt{(#1)}
  }
}
```

```
\ExplSyntaxOff
```

2.13 formale-sprachen.sty

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

```
\directlua{
  formale_sprachen = require('lehramt-informatik-formale-sprachen')
}
```

```
\RequirePackage{hyperref}
```

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

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

```
\def\liMengeOhneMathe#1{\{ #1 \}}
\def\liMenge#1{%
\ifmmode%
\liMengeOhneMathe{#1}%
\else%
$\liMengeOhneMathe{#1}$%
\fi%
}
```

`\liEpsilon` `\liEpsilon`: ε
Let-Abkürzung: `\let\e=\liEpsilon`

```
\def\liEpsilon{$\varepsilon$}
```

`\liPotenzmenge` Umgeben mit geschweiften Klammern in einer Mathematik-Umgebung

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

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

```
\let\liZustandsmengeOhneMathe=\liPotenzmengeOhneMathe
\let\liZustandsmenge=\liPotenzmenge
```

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

```
\def\liUeberfuehrungsFunktionOhneMathe#1{\delta(\erzeuge@tiefgestellt{#1})}
\def\liUeberfuehrungsFunktion#1{
\ifmmode
  \liUeberfuehrungsFunktionOhneMathe{#1}
\else
  $\liUeberfuehrungsFunktionOhneMathe{#1}$
\fi
}
```

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

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

`\liBandAlphabet` `\liBandAlphabet{\liTuringLeerzeichen}`: $\Gamma = \Sigma \cup \{\Box\}$

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

`\liZustandsBuchstabe`

```
\def\liZustandsBuchstabe{z}
```

`\liZustandsBuchstabeGross`

```
\def\liZustandsBuchstabeGross{Z}
```

\liZustandsmengeNr

```
\def\zustandsnamens@liste#1#2{
  $
    \{
      \directlua{formale_sprachen.erzeuge_zustandsnamens_liste('#1', '#2')}
    \}
  $
}
\def\liZustandsmengeNr#1{\zustandsnamens@liste{\liZustandsBuchstabe}{#1}}
```

\liZustandsmengeNrGross

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

\liZustandsname `\liZustandsname{1}`: `$z_1$`

```
\def\liZustandsname#1{$\liZustandsBuchstabe_#1$}
```

\liZustandsnameGross `\liZustandsnameGross{1}`: `$Z_1$`

```
\def\liZustandsnameGross#1{$\liZustandsBuchstabeGross_#1$}
```

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

```
\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}
```

```
\NewDocumentEnvironment { liProduktionsRegeln }
{ O{P} +b }
{
  \liGeschweifteKlammern{#1}
  {
    \begin{align*}
    \directlua{formale_sprachen.produktions_regeln('#2')}
    \end{align*}
  }{-0.2cm}{-1.5cm}
} {}
```

\liProduktionen `\liProduktionen{S -> A, A -> a}`: $\{S \to A, A \to a\}$

```
\def\liProduktionen#1{
  \liMenge{\directlua{formale_sprachen.produktions_regeln('#1', true)}}
}
```

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

```
\def\liZustandsnameTiefgestellt#1{
  \ifmmode
    \liZustandsBuchstabe\sb{#1}
  \else
    $\liZustandsBuchstabe\sb{#1}$
  \fi
}
```

```
\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}
```

```
\NewDocumentCommand{ \liAusdruck } { O{L} m m } {
  $
   \tl_if_empty:nTF {#1} {} {#1 =}
   \{
    \, #2 \,
    |
    \, #3 \,
  \}$
}
\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

```
\def\liFlaci#1{%
  \par
  {%
    \scriptsize
    Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte
    Automaten, Compiler und Interpreter) Ein Projekt der Hochschule
    Zittau/Görlitz und der Pädagogischen Hochschule Schwyz:
    \href{https://flaci.com/#1}{flaci.com/#1}%
  }%
  \par
}
```

`\liGrammatik` `\liGrammatik[`⟨*grammtik-name*⟩`]{`⟨*variablen=V,alphabet=Σ,produktionen=P,start=S*⟩`}`

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

```
\ExplSyntaxOn
\NewDocumentCommand {\liGrammatik} { O{G} m } {
  \tl_set:Nn \l_variablen_tl {V}
  \tl_set:Nn \l_alphabet_tl {\Sigma}
  \tl_set:Nn \l_produktionen_tl {P}
  \tl_set:Nn \l_start_tl {S}

  \keys_define:nn { grammatik } {
    variablen .code:n = {\tl_set:Nn \l_variablen_tl {\liMenge{##1}}},
    alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\liMenge{##1}}},
    produktionen .code:n = {\tl_set:Nn \l_produktionen_tl {\liProduktionen{##1}}},
    start .code:n = {\tl_set:Nn \l_start_tl {##1}},
  }

  \keys_set:nn { grammatik } { #2 }

  $#1 = (
    \l_variablen_tl,
    \l_alphabet_tl,
    \l_produktionen_tl,
    \l_start_tl
  )$
}
\ExplSyntaxOff
```

2.14 formatierung.sty

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

2.14.1 Schriftarten / Typographie

```
\RequirePackage{mathpazo}
\RequirePackage[no-math]{fontspec}
\setmainfont{texgyrepagella}
\setsansfont{QTAncientOlive}
\RequirePackage{sectsty}
\allsectionsfont{\fontspec{QTAncientOlive}}
```

2.14.2 Farben

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

2.14.3 Überschriften

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

2.14.4 Listen

```
\RequirePackage{paralist}
\renewcommand\labelitemi{-}
\renewcommand\labelitemii{-}
\renewcommand\labelitemiii{-}
\renewcommand\labelitemiv{-}
% Counter: enumi enumii enumiii enumiv
% Styles: \arabic{counter} \alph{counter} \Alph{counter} \roman{counter} \Roman{counter}
\renewcommand{\labelenumi}{(\alph{enumi})}
\renewcommand{\labelenumii}{(\roman{enumii})}
```

2.14.5 Kasten

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

liKasten

```
\NewDocumentEnvironment { liKasten }{ } {
  \begin{mdframed}[backgroundcolor=white!0]
} {
  \end{mdframed}
}
```

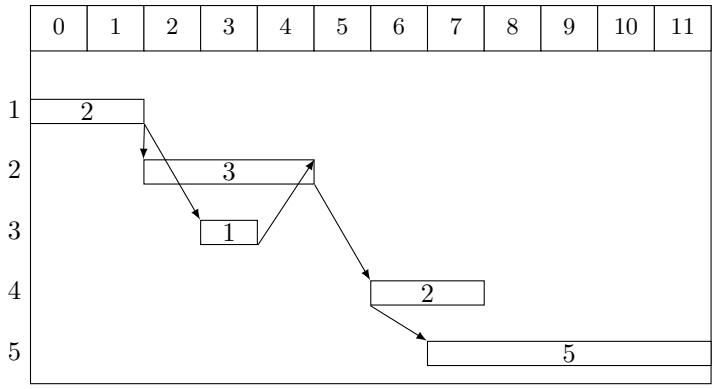
2.14.6 Header

```
\RequirePackage{fancyhdr}
\fancyhead[L,C,R]{}
\fancyfoot[L]{}
\fancyfoot[C]{}
\fancyfoot[R]{\thepage}
\pagestyle{fancy}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}
```

2.15 gantt.sty

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

```
\begin{ganttchart}[x unit=0.75cm, y unit chart=0.8cm]{0}{11}
\ganttitlelist{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}
```




```
\RequirePackage{tikz-uml}
\RequirePackage{pgfgantt}
\setganttlinklabel{f-s}{}
\setganttlinklabel{s-s}{}
\setganttlinklabel{f-f}{}
\setganttlinklabel{s-f}{}
```

2.16 grafik.sty

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

```

2.17 graph.sty

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

Für die die Adjazenzliste (xrightarrow)

```
\RequirePackage{amsmath}
```

Für Adjazenz-Matrix

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

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

```
\RequirePackage{blkarray}
\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}
```

```
\tikzset{
  li graph/.style={
    every node/.style={
      rectangle,
      draw,
    },
    every edge/.style={
      >={Stealth[black]},
      draw,
    },
    every edge/.append style={
      every node/.style={
        sloped,
        auto,
      }
    }
  },
  li markierung/.style={
    ultra thick,
  }
}
```

`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}
```

```
\NewDocumentEnvironment { liGraphenFormat }{ +b } {} {}
```

2.18 hanoi.sty

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

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

```
\RequirePackage{tikz}
\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}`

```
\def\li@mset #1[#2]=#3{%
\expandafter\xdef\csname #1#2\endcsname{#3}
}
\def\li@mget #1[#2]{%
\csname #1#2\endcsname
}
\def\li@minc #1[#2]+=#3{%
\pgfmathparse{\li@mget #1[#2]+#3}%
\li@mset #1[#2]=\pgfmathresult
}

\def\liHanoi#1#2{
  \edef\li@numdiscs{#1}
  \def\li@sequence{#2}
  \begin{tikzpicture}[line width=4mm,brown!40,line cap=round,xscale=3]
    % init colors
    \foreach[count=\j] \c in {red,green!80!black,blue,orange,violet,gray,yellow!80!black,purp
    \li@mset col[\j]={\c};
    % draw poles and init pole counters
    \foreach \j in {1,2,3}{
      \li@mset pos[\j]=0
      \draw (\j,-.5) -- +(0, .5 + .5*\li@numdiscs);
    }
    % draw base
    \draw (.5,-.5) -- +(3,0);
    % draw discs
    \foreach[count=\k] \i/\j in \li@sequence{
      \draw[draw={\li@mget col[\i]}] (\j,\li@mget pos[\j]) +(-.4*\i/\li@numdiscs,0) -- +(.4*\
      \li@minc pos[\j]+={.5}
    }
  \end{tikzpicture}
}
```

2.19 klassen-konfiguration-aufgabe.sty

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

Formatierung muss vor literatur sein, sonst Option clash

```
\liLadePakete{
  formatierung,
  abmessung,
  literatur-dummy,
  makros,
  aufgaben-metadaten,
  kopf-fusszeilen,
  mathe
}

% Komischer Option-Clash deshalb ganz am Ende, für die Silbentrennung
\RequirePackage[ngerman]{babel}
\ExplSyntaxOn
\AddToHook{enddocument}{
  \_gib_github_url:
}
\ExplSyntaxOff

```

2.20 komplexitaetstheorie.sty

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

2.20.1 Makro-Kürzel

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

```
\liLadePakete{mathe}
```

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

```
\RequirePackage{mdframed}
```

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

```
\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

```
\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 $\mathtt{S} = k$, sodass für alle Knoten $u \neq v \in V$ gilt, dass $\{u, v\}$ eine Kante in E ist?

Let-Abkürzung: `\let\b=\liProblemBeschreibung`

```
\def\liProblemBeschreibung#1#2#3{
  \begin{mdframed}[
    userdefinedwidth=9cm,
    align=center,
    backgroundcolor=white!0,
  ]
    \centerline{\large\liProblemName{#1}}

    \medskip

    \begin{description}
    \item[Gegeben:] #2
    \item[Frage:] #3
    \end{description}
  \end{mdframed}
}
```

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

```
\NewDocumentCommand{ \liPolynomiellReduzierbar } { m O{p}  m } {
\begin{displaymath}
  \liProblemName{#1}
  \preceq_{#2}
  \liProblemName{#3}
\end{displaymath}
}
```

\liProblemVertexCover

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

\liProblemVertexCover

```
\def\liProblemVertexCover{%
%
Das \textbf{Knotenüberdeckungsproblem} (\liProblemName{Vertex Cover})
fragt, ob zu einem gegebenen einfachen Graphen und einer natürlichen
Zahl $k$ eine Knotenüberdeckung der Größe von höchstens $k$ existiert.
\footcite{wiki:knotenueberdeckung}

Das heißt, ob es eine aus maximal $k$ Knoten bestehende Teilmenge $U$
der Knotenmenge gibt, so dass jede Kante des Graphen mit
mindestens einem Knoten aus $U$ verbunden ist.
\footcite[Seite 78]{theo:fs:4}%
}
```

\liProblemSubsetSum Kein Karp-21-Problem

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

\liProblemSubsetSum Kein Karp-21-Problem

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

```

2.21 kontrollflussgraph.sty

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

2.21.1 Makro-Kürzel

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

2.21.2 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}
```

2.21.3 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 = ...}}]
```

```
\RequirePackage{tikz}
\usetikzlibrary{positioning}
\tikzset{
  li kontrollfluss/.style={
    knoten/.style={
      circle,
      draw
    },
    usebox/.style={
      draw,
      rectangle,
      font=\scriptsize,
      anchor=west,
      align=left,
    },
    bedingung/.style={
      midway,
      draw=none,
      font=\scriptsize
    },
    knotenbeschriftung/.style={
      draw,
      rectangle,
      midway,
      font=\scriptsize
    },
    wahr/.style={
      thick
    },
    falsch/.style={
      dashed
    },
    every node/.style={
      circle,
      draw,
    },
    every edge/.append style={
      every node/.style={
        draw=none,
        bedingung,
      }
    },
    every path/.style={
      draw,
      ->,
    },
    every pin/.style={
      draw,
      dotted,
      rectangle,
      pin position=right
    },
    every pin edge/.style={
      dotted,
      arrows=-,
    }
  }
}
```

2.21.4 Umgebungen

liKontrollflussgraph

```
\NewDocumentEnvironment { liKontrollflussgraph } { O{} } {
```

```
  \begin{tikzpicture}[
    li kontrollfluss,
    #1
  ]
} {
  \end{tikzpicture}
}
```

2.21.5 Makros

\liAnweisung

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

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

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

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

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

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

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

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

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

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

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

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

```
\ExplSyntaxOn
\NewDocumentCommand { \liKontrollKnotenPfad }{ m }
{
  \seq_set_split:Nnn \l_tmpa_seq { - } { #1 }
  \seq_set_map:NNn \l_tmpa_seq \l_tmpa_seq {\liKontrollTextzeileKnoten{##1}}
  \seq_use:Nn \l_tmpa_seq { ~~~~ }
}
\ExplSyntaxOff
```

2.22 kopf-fusszeilen.sty

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

```
\ExplSyntaxOn
```

```
\fancyhead{}
\fancyhead[RO,LE]{{\scriptsize\LehramtInformatikTitel}}
\fancyhead[LO,RE]{{\scriptsize\today}}
\fancyfoot{}
\fancyfoot[LE,RO]{\thepage}
\fancyfoot[LO,CE]{{\scriptsize\LehramtInformatikAutorName}}
\fancyfoot[CO,RE]{{\scriptsize\LehramtInformatikAutorEmail}}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
\setlength{\headwidth}{\textwidth}
```

```
\ExplSyntaxOff
```

2.23 literatur-dummy.sty

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

\literatur

```
\def\literatur{}
```

\footcite

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

```

2.24 literatur.sty

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

```
\RequirePackage{csquotes}
\RequirePackage[
  bibencoding=utf8,
  citestyle=authortitle,
  backend=biber,
]{biblatex}
\addbibresource{\LehramtInformatikRepository/Quellen/10_DB.bib}
\addbibresource{\LehramtInformatikRepository/Quellen/20_OOMUP.bib}
\addbibresource{\LehramtInformatikRepository/Quellen/30_AUD.bib}
\addbibresource{\LehramtInformatikRepository/Quellen/40_SOSY.bib}
\addbibresource{\LehramtInformatikRepository/Quellen/60_FUMUP.bib}
\addbibresource{\LehramtInformatikRepository/Quellen/70_THEO.bib}
\addbibresource{\LehramtInformatikRepository/Quellen/80_DDI.bib}
\addbibresource{\LehramtInformatikRepository/Quellen/Allgemein.bib}
\addbibresource{\LehramtInformatikRepository/Quellen/Examen.bib}
\addbibresource{\LehramtInformatikRepository/Quellen/Wikipedia.bib}
% To allow footnotes in the heading
\RequirePackage[stable,multiple]{footmisc}
```

\literatur

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

2.25 makros.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-makros}[2020/11/27 Eine Sammlung von
häufig verwendeten Makros und Umgebungen, die thematisch zu keinem
anderen Paket passen]
```

```
\RequirePackage{hyperref}
```

```
\RequirePackage{graphicx}
```

Für die Umgebung `liQuellen` benötigt.

```
\RequirePackage{paralist}
```

`\inhaltsverzeichnis`

```
\def\inhaltsverzeichnis {
  \begin{mdframed}
    \begingroup
    \let\clearpage\relax
    \tableofcontents
    \endgroup
  \end{mdframed}
}
```

`\memph` `\mephm` (`\marginpar` and `\emph`)

```
\newcommand{\memph}[1]{\emph{#1}\marginpar{\tiny#1}}
```

`\SLASH`

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

`\liPseudoUeberschrift` Text, der sich wie eine Überschrift verhält.

```
\newcommand{\liPseudoUeberschrift}[1]{
\bigskip
\noindent
\textsf{\textbf{#1}}
\noindent
}
```

`\liBeschriftung` Ähnlich dem Makro `\liPseudoUeberschrift{}`. Am Ende des Textes wird ein Doppelpunktzeichen angehängt.

```
\newcommand{\liBeschriftung}[1]{
\par
\noindent
\medskip
\textbf{#1}:
\medskip
\noindent
}
```

`\hinweis`

```
\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.

```
\NewDocumentEnvironment { liProjektSprache }{ o +b } {} {}
```

`liEinbettung`

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

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

```
\RequirePackage{xparse}
\ExplSyntaxOn
```

```
\NewDocumentEnvironment{ liAntwort } { O{standard} }
{
  \str_case:nn {#1} {
    {standard} {
      \def\beschriftung{}
      \mdfsetup{backgroundcolor=gray!1,linecolor=gray}
    }
    {richtig} {
      \def\beschriftung{richtig}
      \mdfsetup{backgroundcolor=green!5,linecolor=green}
    }
    {falsch} {
      \def\beschriftung{falsch}
      \mdfsetup{backgroundcolor=red!3,linecolor=red}
    }
    {muster} {
      \def\beschriftung{Musterlösung}
      \mdfsetup{backgroundcolor=green!12,linecolor=green}
    }
  }
  \ifx\beschriftung\empty\else
  \noindent
  \textbf{\beschriftung{}:}
  \fi
  \begin{mdframed}
}
{\end{mdframed}}
```

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

```
\NewDocumentEnvironment{ liAdditum } { o }
{
  \begin{mdframed}[backgroundcolor=yellow!5]
  \IfNoValueTF {#1}
  { \liPseudoUeberschrift{Additum} }
  { \liPseudoUeberschrift{Additum:~#1} }
}
{\end{mdframed}}
```

`liExkurs`

```
\begin{liExkurs}[Linear rekursiv]
Die häufigste Rekursionsform ist die lineare Rekursion, bei der in jedem
Fall der rekursiven Definition höchstens ein rekursiver Aufruf vorkommen
darf.
\end{liExkurs}
```

> **Exkurs: Linear rekursiv**
> Die häufigste Rekursionsform ist die lineare Rekursion, bei der in jedem Fall der rekursiven Definition höchstens ein rekursiver Aufruf vorkommen darf.

```
\NewDocumentEnvironment{ liExkurs }{o +b}{
  \vspace{0.2cm}%
  \begin{mdframed}[
    backgroundcolor=white,
    bottomline=false,
    innermargin=1cm,
    leftline=true,
    linecolor=black,
    linewidth=0.1cm,
    outermargin=1cm,
    rightline=false,
    topline=false,
  ]
```

```
  \footnotesize
  \noindent%
  \textbf{Exkurs:~#1}\par%
  \noindent%
  #2
  \end{mdframed}
  \vspace{0.2cm}
}{}
```

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

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

Weiterführende Literatur:

- Quelle 1
- Quelle 2

```
\cs_new:Npn \listen@punkt #1 {\item #1}
\NewDocumentEnvironment { liQuellen }{ +b }
{
  \seq_clear_new:N \l_quellen
  \seq_set_split:Nnn \l_quellen {\item} {#1}
  \seq_remove_all:Nn \l_quellen {}
  \begin{mdframed}[backgroundcolor=white!0]
  \footnotesize
  \noindent
  \textsf{\textbf{Weiterführende~Literatur:}}
  \medskip
  \begin{compactitem}
  \seq_map_function:NN \l_quellen {\listen@punkt}
  \end{compactitem}
  \end{mdframed}
  %
  \makeatletter
  \par\@afterindentfalse\@afterheading
  \makeatother
} {}
```

liLernkartei

```
\NewDocumentEnvironment { liLernkartei }{ m +b }
{
  \begin{mdframed}
  \footnotesize
  \noindent%
  \textbf{Lernkarteikarte:~#1}\par%
  \noindent%
  #2
  \end{mdframed}
} {}
```

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

```
\NewDocumentEnvironment { liDiagramm }{ m +b }
{
  \begin{mdframed}[backgroundcolor=white!0]
  \small
  \noindent%
  \textit{#1}:
  \begin{center}
```

```
  #2
  \medskip
  \end{center}
  \end{mdframed}
} {}
\ExplSyntaxOff
```

\liFussnoteUrl

\liFussnoteUrl[⟨*zusätzlicher-text*⟩]{⟨*url*⟩} \liFussnoteUrl[zusätzlicher Text]{url}: Eine HTTP-URL als Fußnote setzen.

```
\NewDocumentCommand{\liFussnoteUrl} { o m } {
  \footnote{\url{#2}\IfNoValueTF{#1}{}{ (#1)}}
}
```

\liFussnoteLink

\liFussnoteLink[⟨*zusätzlicher-text*⟩]{⟨*link-text*⟩}{⟨*url*⟩} \liFussnoteLink[zusätzlicher Text]{tex
Einen Link, d. h. einen Link-Text und eine URL als Fußnote setzen.

```
\NewDocumentCommand{\liFussnoteLink} { o m m } {
  \footnote{\href{#3}{#2}\IfNoValueTF{#1}{}{ (#1)}}
}
```

\zB

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

\ZB

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

\dh

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

2.26 master-theorem.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-master-theorem}[2021/04/13]
```

2.26.1 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 $\varepsilon = 4$: \\
$f(n) = 5n^2 \in \O{n^{\log_2 {8 - 4}}} = \O{n^{\log_2 4}} = \O{n^2}$}
% 2. Fall
{$f(n) = 5n^2 \notin \t{n^{\log_2 {8}}} = \t{n^3}$}
% 3. Fall
{$f(n) = 5n^2 \notin \o{n^{\log_2 {8 + \varepsilon}}}$}

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

```
\liMasterVariablenDeklaration
{} % a
{} % b
{} % f(n) ohne $mathe$

\liMasterFallRechnung
% 1. Fall
{}
% 2. Fall
{}
% 3. Fall
{}

$\Rightarrow T(n) \in \t{n^2 \cdot \log n}$
\liMasterWolframLink{T[n]=9T[n/3]\%2B5n^2}
```

```
\ExplSyntaxOn
\RequirePackage{amsmath}
```

\liRundeKlammer

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

\liTheta

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

```
\def\liThetaOhneMathe#1{
  \Theta \liRundeKlammer{#1}
}
\def\liTheta#1{
  \ifmmode
    \liThetaOhneMathe{#1}
  \else
    $\liThetaOhneMathe{#1}$
  \fi
}
```

\liOmega \liOmega{n^2}: $\Omega\big(n^2\big)$

```
\def\liOmegaOhneMathe#1{
  \Omega \liRundeKlammer{#1}
}
\def\liOmega#1{
  \ifmmode
    \liOmegaOhneMathe{#1}
  \else
    $\liOmegaOhneMathe{#1}$
  \fi
}
```

\liO \liO{n^2}: $\mathcal{O}\big(n^2\big)$

```
\def\liOOhneMathe#1{
  \mathcal{O} \liRundeKlammer{#1}
}
\def\liO#1{
  \ifmmode
    \liOOhneMathe{#1}
  \else
    $\liOOhneMathe{#1}$
  \fi
}
```

\liT **Let-Abkürzung:** \let\T=\liT
\liT{16}{2}: $16 \cdot T\big(\frac{n}{2}\big)$ \liT{}{2}: $T\big(\frac{n}{2}\big)$

```
\def\liTOhneMathe#1#2{
  \tl_if_blank:nTF {#1}
  {}
  {#1 \cdot }
  T
  \liRundeKlammer{\frac{n}{#2}}
}
\def\liT#1#2{
  \ifmmode
    \liTOhneMathe{#1}{#2}
  \else
    $\liTOhneMathe{#1}{#2}$
  \fi
}
```

\liRekursionsGleichung \liRekursionsGleichung: $T(n) = a \cdot T\big(\frac{n}{b}\big) + f(n)$

```
\def\liRekursionsGleichung{
  $T(n) = \liT{a}{b} + f(n)$
}
```

\liBedingungEins \liBedingungEins: $f(n) \in \mathcal{O}\big(n^{\log_b a-\varepsilon}\big)$

```
\def\liBedingungEins{
  $f(n) \in \liO{n^{\log\sb{b}a - \varepsilon}}$
}
```

\liBedingungZwei \liBedingungZwei: $f(n) \in \Theta\big(n^{\log_b a}\big)$

```
\def\liBedingungZwei{
  $f(n) \in \liTheta{n^{\log\sb{b}a}}$
}
```

\liBedingungDrei \liBedingungDrei: $f(n) \in \Omega\big(n^{\log_b a+\varepsilon}\big)$

```
\def\liBedingungDrei{
  $f(n) \in \liOmega{n^{\log\sb{b}a + \varepsilon}}$
}
```

```
\ExplSyntaxOff
```

\liMasterVariablen

```
\def\liMasterVariablen{
  \begin{displaymath}
  T(n) = \liT{a}{b} + f(n)
  \end{displaymath}

  \begin{itemize}
  \item[$a =$]
  Anzahl der Unterprobleme in der Rekursion

  \item[$\textstyle{\frac{1}{b}} =$]
  Teil des Originalproblems, welches wiederum durch alle Unterprobleme
  repräsentiert wird

  \item[$f(n) =$]
  Kosten (Aufwand, Nebenkosten), die durch die Division des Problems und
  die Kombination der Teillösungen entstehen
  \end{itemize}
  \footcite{wiki:master-theorem}
  \footcite[Seite 19-35 (PDF 11-24)]{aud:fs:2}
}
```

\liMasterFaelle

```
\def\liMasterFaelle{
  \begin{description}
  \item[1. Fall:]
  $T(n) \in \liTheta{n^{\log\sb{b}a}}$

  \hfill falls \liBedingungEins
  für $\varepsilon > 0$

  \item[2. Fall:]
  $T(n) \in \liTheta{n^{\log\sb{b}a} \cdot \log n}$

  \hfill falls \liBedingungZwei

  \item[3. Fall:]
  $T(n) \in \liTheta{f(n)}$

  \hfill falls \liBedingungDrei
  für $\varepsilon > 0$
  und ebenfalls für ein $c$ mit $0 < c < 1$ und alle hinreichend großen $n$
  gilt:
  $a \cdot f(\textstyle {\frac {n}{b}})\leq c \cdot f(n)$
  \end{description}
}
```

liMasterVariablenDeklaration

```
\def\liMasterVariablenDeklaration#1#2#3{
  \begin{description}
    \item[Allgemeine Rekursionsgleichung:] \strut

    \liRekursionsGleichung

    \item[Anzahl der rekursiven Aufrufe ($a$):] \strut

    #1

    \item[Anteil Verkleinerung des Problems ($b$):] \strut

    um $\frac{1}{#2}$ also $b = #2$

    \item[Laufzeit der rekursiven Funktion ($f(n)$):] \strut
```

```

    $#3$

    \item[Ergibt folgende Rekursionsgleichung:] \strut

    $T(n) = \liT{#1}{#2} + #3$
  \end{description}
}
```

\liMasterFallRechnung

```
\def\liMasterFallRechnung#1#2#3{
  \begin{description}
  \item[1. Fall:] \liBedingungEins:

  #1

  \item[2. Fall:] \liBedingungZwei:

  #2

  \item[3. Fall:] \liBedingungDrei:

  #3
  \end{description}
}
```

\liMasterExkurs

```
\def\liMasterExkurs{
  \begin{liExkurs}[Master-Theorem]
  \liMasterVariablen

  \noindent
  Dann gilt:

  \liMasterFaelle
  \end{liExkurs}
}
```

\liMasterWolframLink Link zu Wolframalpha (+ durch Prozent 2B ersetzen)

```
\def\liMasterWolframLink#1{
  Berechne die Rekursionsgleichung auf WolframAlpha:
  \href{https://www.wolframalpha.com/input/?i=#1}{WolframAlpha}
}

```

2.27 mathe.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-mathe}[2020/06/10]

% for example \ltimes \rtimes
%\RequirePackage{amssymb}
\RequirePackage{amsmath}

%%
% \mlq \mrq
%%
\DeclareMathSymbol{\mlq}{\mathord}{operators}{``}
\DeclareMathSymbol{\mrq}{\mathord}{operators}{`'}

```

2.28 minimierung.sty

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

```
\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|}
\hline
\z0 & \l  & \l  & \l  & \l  & \l  & \l  & \l  & \l  & \l  \\ \hline
\z1 &     & \l  & \l  & \l  & \l  & \l  & \l  & \l  & \l  \\ \hline
\z2 &     &     & \l  & \l  & \l  & \l  & \l  & \l  & \l  \\ \hline
\z3 &     &     &     & \l  & \l  & \l  & \l  & \l  & \l  \\ \hline
\z4 &     &     &     &     & \l  & \l  & \l  & \l  & \l  \\ \hline
\z5 &     &     &     &     &     & \l  & \l  & \l  & \l  \\ \hline
\z6 &     &     &     &     &     &     & \l  & \l  & \l  \\ \hline
\z7 &     &     &     &     &     &     &     & \l  & \l  \\ \hline
\z8 &     &     &     &     &     &     &     &     & \l  \\ \hline\hline
    & \z0 & \z1 & \z2 & \z3 & \z4 & \z5 & \z6 & \z7 & \z8 \\ \hline
\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

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

```
\def\li@fussnote@text#1#2{
  \liFussnote{#1}
  \quad
  {\footnotesize #2}
}
```

\liFussnoteEinsText

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

\liFussnoteZweiText

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

\liFussnoteDreiText

```
\def\liFussnoteDreiText{
  \li@fussnote@text{3}
```

```
  {In weiteren Iterationen markierte Zustände.}
}
```

\liFussnoteVierText

```
\def\liFussnoteVierText{
  \li@fussnote@text{4}
  {…}
}
```

\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 …

```
\def\liFussnoten{
  \bigskip

  \noindent
  \liFussnoteEinsText

  \noindent
  \liFussnoteZweiText

  \noindent
  \liFussnoteDreiText

  \noindent
  \liFussnoteVierText
}
```

\liLeereZelle

\liLeereZelle: $\emptyset$
Let-Abkürzung: \let\l=\liLeereZelle

```
\def\liLeereZelle{$\emptyset$}
```

\liZustandsPaarVariablenName

```
\def\liZustandsPaarVariablenName{z}
```

\liZustandsPaar

```
\def\liZustandsPaar#1#2{
  $(
    \liZustandsPaarVariablenName_#1,
    \liZustandsPaarVariablenName_#2
  )$
}
```

liUebergangsTabelle

```
\renewcommand{\arraystretch}{1.4}
\NewDocumentEnvironment{ liUebergangsTabelle } { m m } {
  \liPseudoUeberschrift{Übergangstabelle}
  \begin{center}
  \begin{tabular}{r||l|l}
  \textbf{Zustandspaar} & \textbf{#1} & \textbf{#2} \\\hline
} {
  \end{tabular}
  \end{center}
}
```

iUeberschriftDreiecksTabelle

\liUeberschriftDreiecksTabelle: **Minimierungstabelle (Table filling)**

```
\ExplSyntaxOn
\def\liUeberschriftDreiecksTabelle{
  \liPseudoUeberschrift{Minimierungstabelle~(Table~filling)}
}
```

`\liMinimierungErklaerung` **Let-Abkürzung:** `\let\erklaerung=\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. ————

```
\def\liMinimierungErklaerung{
  %\footcite[Seite~19]{koenig}
  \liParagraphMitLinien{
    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~„$x\sb{n}$"~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.
  }
}
\ExplSyntaxOff
```

2.29 normalformen.sty

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

Mathe für textit tex etc, typographie für geschweifte Klammer

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

```
\directlua{
  helfer = require('lehramt-informatik-helfer')
  normalformen = require('lehramt-informatik-normalformen')
}
```

2.29.1 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
```

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

`\liAttributHuelle` **Let-Abkürzung:** `\let\ah=\liAttributHuelle`

`\ah{F, \m{A, B}}` AttrHülle$(F, \{A, B\})$ Regulärer Ausdruck zum Konvertieren `AttrHülle\((.*)\) \\ah{$1}`

```
\def\liAttributHuelleOhneMathe#1{\text{AttrHülle}(#1)}
\def\liAttributHuelle#1{
\ifmmode
  \liAttributHuelleOhneMathe{#1}
\else
  $\liAttributHuelleOhneMathe{#1}$
\fi
}
```

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

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

`liAHuelle`

```
\NewDocumentEnvironment{ liAHuelle } { +b } {
  \begingroup
  \footnotesize
  \begin{multline*}
    #1
  \end{multline*}
  \endgroup
} { }
```

`\liLinksReduktion` Nur innerhalb von `liAHuelle` zu verwenden bzw. `multline`

Let-Abkürzung: `\let\ahL=\liLinksReduktion`

`\ahL{ursprüngliche linke Attributmenge}{ohne dieses Attribut}{Ergebnis}`

```
\def\liLinksReduktion#1#2#3{
  \shoveleft{
    \liAttributHuelleOhneMathe{FA,
    \liAttributMenge{#1 \string\ #2}} =
    } \\
```

```
  \shoveright{
    \liAttributMenge{#3}
  } \\
}
```

\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$(F, \{A,\ E \mid E\}) = \{A,\ E,\ F,\ B,\ \boldsymbol{D}\}$

```
\def\liLinksReduktionInline#1#2#3{%
  {%
    \footnotesize%
    $\liAttributHuelleOhneMathe{F,
    \liAttributMenge{#1 \string\ #2}} =
    \liAttributMenge{#3}$
  }
}
```

\liLinksReduktionInline

Let-Abkürzung: `\let\ahr=\liLinksReduktionInline`

`\ahr{gelöschte FA}{neue FA ohne rechts Attribut}{gegebene Attribute}{Ergebnis}`

```
\def\liRechtsReduktionInline#1#2#3#4{%
  {%
    \footnotesize%
    $\liAttributHuelleOhneMathe{
      F \setminus
      \liFunktionaleAbhaengigkeit{#1}
      \def\tmp{#2}\ifx\tmp\empty
      \else
        \cup \liFunktionaleAbhaengigkeit{#2}
      \fi
      ,
      \liAttributMenge{#3}
    } =
    \liAttributMenge{#4}$
  }
}
```

\liFunktionaleAbhaengigkeit

Let-Abkürzung: `\let\fa=\liFunktionaleAbhaengigkeit`

`\fa{A, B -> C, D}`: $\{\,A,\ B\,\} \to \{\,C,\ D\,\}$ Regulärer Ausdruck zum Konvertieren: `\$(.*?) \\rightarrow (.*?)\$ \\fa{$1 -> $2}`

```
\def\liFunktionaleAbhaengigkeit#1{%
  \directlua{normalformen.drucke_funk_abhaengigkeit('#1', false)}%
}
```

liFunktionaleAbhaengigkeiten

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

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

Let-Abkürzung: `\let\FA=\liFunktionaleAbhaengigkeiten`

Regulärer Ausdruck zum Konvertieren: `\\item \$(.*) \\rightarrow (.*)\$ \$(.*) \\rightarro`
`$1 -> $2;`

```
\NewDocumentCommand {\liFunktionaleAbhaengigkeiten} { O{FA} m } {
  \liGeschweifteKlammern
  {#1}
  {
    \begin{align*}
    \directlua{normalformen.drucke_funk_abhaengigkeiten('#2')}
    \end{align*}
  }
  {-0.5cm}
  {-1.7cm}
}
```

`\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}`

```
\NewDocumentCommand {\liRelation} { O{R} m } {
  $\directlua{
    local name = helfer.konvertiere_tiefgestellt('#1')
    tex.print(name)
  }$(\textit{\,#2\,})
}
```

2.30 petri.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-petri}[2020/12/03]
```

2.30.1 Makro-Kürzel

```
\let\t=\liPetriTransitionsName
\let\tp=\liPetriTransPfeile
\let\k=\liPetriErreichKnotenDrei
```

```
\RequirePackage{tikz}
\usetikzlibrary{petri,arrows.meta}
```

Für die Darstellungsmatrix

```
\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

```
\def\liPetriSetzeSchluessel{%
  \def\TmpTransitionOne{}%
  \def\TmpTransitionTwo{}%
  \def\TmpTransitionThree{}%
  \def\TmpTransitionFour{}%
  \def\TmpTransitionFive{}%
  \def\TmpTransitionSix{}%
  \def\TmpTransitionSeven{}%
  \def\TmpTransitionEight{}%
  \def\TmpTransitionNine{}%
  \def\TmpTransitionTen{}%
  \pgfkeys{/petri/.cd,
    p1/.store in=\TmpPlaceOne,p1/.default=0,p1,
    p2/.store in=\TmpPlaceTwo,p2/.default=0,p2,
    p3/.store in=\TmpPlaceThree,p3/.default=0,p3,
    p4/.store in=\TmpPlaceFour,p4/.default=0,p4,
```

```
    p5/.store in=\TmpPlaceFive,p5/.default=0,p5,
    p6/.store in=\TmpPlaceSix,p6/.default=0,p6,
    p7/.store in=\TmpPlaceSeven,p7/.default=0,p7,
    p8/.store in=\TmpPlaceEight,p8/.default=0,p8,
    p9/.store in=\TmpPlaceNine,p9/.default=0,p9,
    p10/.store in=\TmpPlaceTen,p10/.default=0,p10,
    t1/.store in=\TmpTransitionOne,t1/.default=activated,
    t2/.store in=\TmpTransitionTwo,t2/.default=activated,
    t3/.store in=\TmpTransitionThree,t3/.default=activated,
    t4/.store in=\TmpTransitionFour,t4/.default=activated,
    t5/.store in=\TmpTransitionFive,t5/.default=activated,
    t6/.store in=\TmpTransitionSix,t6/.default=activated,
    t7/.store in=\TmpTransitionSeven,t7/.default=activated,
    t8/.store in=\TmpTransitionEight,t8/.default=activated,
    t9/.store in=\TmpTransitionNine,t9/.default=activated,
    t10/.store in=\TmpTransitionTen,t10/.default=activated,
    scale/.store in=\TmpScale,scale/.default=0.5,
    x/.store in=\TmpX,x/.default=5,
    y/.store in=\TmpY,y/.default=5,
  }%
}
```

```
\tikzset{
  li petri/.style={
    activated/.style={
      very thick
    },
    inhibitor/.style={
      {Circle[open,length=2mm,fill=white]}-
    }
  }
}
```

`\liPetriTransitionsName` **Let-Abkürzung:** `\let\t=\liPetriTransitionsName`

```
\$t_(\d+)\$ \t$1
```

```
\def\liPetriTransitionsNameOhneMathe#1{t\sb{#1}}
\def\liPetriTransitionsName#1{
  \ifmmode
    \liPetriTransitionsNameOhneMathe{#1}
  \else
    $\liPetriTransitionsNameOhneMathe{#1}$
  \fi
}
```

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

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

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

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

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

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

2.31 potenzmengen-konstruktion.sty

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

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

```
\def\liZustandsMengenSammlung#1#2{
  \liZustandsnameGross{#1}
  {
    \footnotesize
    \liPotenzmenge{
      \str_case:nn {#1} #2
    }
  }
}
```

\liZustandsMengenSammlungNr

```
\def\liZustandsMengenSammlungNr#1#2{
  \liZustandsnameGross{#1}
  {
```

```
    \footnotesize
    \liZustandsmengeNr{
      \str_case:nn {#1} #2
    }
  }
}

\ExplSyntaxOff

```

2.32 pseudo.sty

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

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

Data: $G = (V, E, w)$: ein zusammenhängender, ungerichteter, kantengewichteter Graph kruskal(G)
$E' \leftarrow \emptyset$;
$L \leftarrow E$;
Sortiere die Kanten in L aufsteigend nach ihrem Kantengewicht.;
while $L \neq \emptyset$ **do**
  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
end
Result: $M = (V, E')$ ist ein minimaler Spannbaum von G.

```
\RequirePackage[german,boxruled]{algorithm2e}
```

2.33 pumping-lemma.sty

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

\liPumpingRegulaer

```
\def\liPumpingRegulaer{%
  Es sei $L$ eine reguläre Sprache. Dann gibt es eine Zahl $j$, sodass für
  alle Wörter $\omega \in L$ mit $|\omega| \geq j$ (jedes Wort $\omega$ in
  $L$ mit Mindestlänge $j$) jeweils eine Zerlegung $\omega = uvw$
  existiert, sodass die folgenden Eigenschaften erfüllt sind:

  \begin{enumerate}
  \item $|v| \geq 1$
  (Das Wort $v$ ist nicht leer.)

  \item $|uv| \leq j$
  (Die beiden Wörter $u$ und $v$ haben zusammen höchstens die Länge $j$.)

  \item Für alle $i = 0, 1, 2, \dots$ gilt $uv^iw \in L$
  (Für jede natürliche Zahl (mit $0$) $i$ ist das Wort $uv^{i}w$ in der
  Sprache $L$)
  \end{enumerate}

  Die kleinste Zahl $j$, die diese Eigenschaften erfüllt, wird
  Pumping-Zahl der Sprache $L$ genannt.\footcite{wiki:pumping-lemma}
}
```

\liPumpingKontextfrei

```
\def\liPumpingKontextfrei{%
  Es sei $L$ eine kontextfreie Sprache. Dann gibt es eine Zahl $j$, sodass
  sich alle Wörter $\omega \in L$ mit $|\omega| \geq j$ zerlegen lassen in
  $\omega = uvwxy$, sodass die folgenden Eigenschaften erfüllt sind:

  \begin{enumerate}
  \item $|vx| \geq 1$
  (Die Wörter $v$ und $x$ sind nicht leer.)

  \item $|vwx| \leq j$
  (Die Wörter $v$, $w$ und $x$ haben zusammen höchstens die Länge $j$.)

  \item Für alle $i \in \mathbb{N}_0$ gilt $u v^i w x^i y \in L$ (Für jede
  natürliche Zahl (mit $0$) $i$ ist das Wort $u v^i w x^i y$ in der
  Sprache $L$)
  \end{enumerate}
}
```

2.34 quicksort.sty

```
% https://tex.stackexchange.com/a/142634
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-quicksort}[2020/06/12]

%---------------------------------------------------
% USAGE:
% \QSinitialize{comma, separated, numerical, values}
% \loop
% \QSpivotStep
% \ifnum\value{pivotcount}>0
%   \QSsortStep
% \repeat
%---------------------------------------------------

% xintfrac does not load xinttools, this must be done explicitely if needed as here.
\RequirePackage{xintfrac, xinttools}

\RequirePackage{tikz}

%-----------------------------------------------------------------
% FIRST PART: TikZ styles and macros for the actual drawing
\newcounter{cellcount}%  used for coordinates of the node
\newcounter{pivotcount}% when it will remain at zero, will signal the sort is finished.

% Styles defined by Tom Bombaldi. (modified: all share the same size)
% (re-modified \bf -> \bfseries due to extremely annoying warnings from
% KOMA-script which are truly a pain and do not make any sense regarding \bf:
% if I want to use \bf, and know what I am doing, why should I get HARASSED
% by police of LaTeX good conduct ? )
\tikzset{l/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=lime!70!gray},
        o/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=olive!50},
        r/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black, t
% this is the "b" style as used in the image below
%        b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=magenta!50!black,
% nicer:
        b/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=white, text=magenta
        g/.style={minimum width=6mm, minimum height=6mm, draw=black, fill=gray, text=white, f

% NOTE the b style was originally the same as the r(aised) style apart from
% not being raised, but I find it nicer with a somewhat different
% specification. I have not updated the images though.

% How the nodes are drawn depending on whether on the left of the pivot value
% or on the right, or is a pivot value, or a raised pivot during selection phase.

\def\DecoLEFT #1{%
   \xintFor* ##1 in {#1} \do
   {\stepcounter{cellcount}\node[o] at (\arabic{cellcount},0) {##1};}%
}

\def\DecoINERT #1{%
   \xintFor* ##1 in {#1} \do
   {\stepcounter{cellcount}\node[g] at (\arabic{cellcount},0) {##1};}%
}

\def\DecoRIGHT #1{%
   \xintFor* ##1 in {#1} \do
   {\stepcounter{cellcount}\node[l] at (\arabic{cellcount},0) {##1};}%
}

\def\DecoLEFTwithPivot #1{\stepcounter{pivotcount}%
     \xintFor* ##1 in {#1} \do
```

```
     {\stepcounter{cellcount}%
      \xintifForLast {\node[r]}{\node[o]} at (\arabic{cellcount},0) {##1};}%
}

\def\DecoINERTwithPivot #1{\stepcounter{pivotcount}%
     \xintFor* ##1 in {#1} \do
     {\stepcounter{cellcount}%
      \xintifForLast {\node[b]}{\node[g]} at (\arabic{cellcount},0) {##1};}%
}

\def\DecoRIGHTwithPivot #1{\stepcounter{pivotcount}%
     \xintFor* ##1 in {#1} \do
     {\stepcounter{cellcount}%
      \xintifForLast {\node[r]}{\node[l]} at (\arabic{cellcount},0) {##1};}%
}

%-----------------------------------------------------------------
% SECOND PART: the actual sorting routines.

\def\QS@sort@a #1{\expandafter \QS@sort@b \expandafter {\xintLength {#1}}{#1}}
\def\QS@sort@b #1{\ifcase #1
                      \expandafter\QS@sort@empty
                    \or\expandafter\QS@sort@single
                  \else\expandafter\QS@sort@c
                  \fi
}%
\def\QS@sort@empty  #1{}
\def\QS@sort@single #1{\QSIr {#1}}

% This step is to pick the last as pivot.
\def\QS@sort@c #1%
   {\expandafter\QS@sort@d\expandafter {\romannumeral0\xintnthelt {-1}{#1}}{#1}}%

% Here \QSLr, \QSIr, \QSr have been let to \relax.
% The trick with \xintApplyUnbraced is that for example when selecting
% the elements smaller than pivot, if we had been using \xintApply we
% would have had at the minimum an empty brace pair. Thus we use the
% "unbraced" variant, but then the \QS@select@smaller has added in
% anticipation a level of braces.
\def\QS@sort@d #1#2{%
    \QSLr {\xintApplyUnbraced {\QS@select@smaller  {#1}}{#2}}%
    \QSIr {\xintApplyUnbraced {\QS@select@equal    {#1}}{#2}}%
    \QSRr {\xintApplyUnbraced {\QS@select@greater {#1}}{#2}}%
}%
\def\QS@select@smaller #1#2{\xintifLt {#2}{#1}{{#2}}{ }}% space will stop a f-expansion
\def\QS@select@equal   #1#2{\xintifEq {#2}{#1}{{#2}}{ }}% space will stop a f-expansion
\def\QS@select@greater #1#2{\xintifGt {#2}{#1}{{#2}}{ }}% space will stop a f-expansion

%
% NOTE 1: thus, each comparison with the pivot is done three (!) times.
%
% NOTE 2: we may well end up with \QSLr {<empty>} situations. THis is handled
% silently by the \xintFor loops, and also when \QSLr becomes \QS@sort@a, the
% latter must handle correctly an empty argument.

%-----------------------------------------------------------------
% THIRD PART: the main macros \QSpivotStep, \QSsortStep and \QSinitialize.

% This draws all with suitable highlighting for the newly chosen pivots
% (which will be shown raised)
```

```
\def\QSpivotStep {\let\QSLr\DecoLEFTwithPivot
                \let\QSIr\DecoINERT
                \let\QSIrr\DecoINERT
                \let\QSRr\DecoRIGHTwithPivot
\par\centerline{\rule[1.5mm]{0pt}{8mm}%
            \setcounter{cellcount}{0}\setcounter{pivotcount}{0}%
            \begin{tikzpicture}\QS@list\end{tikzpicture}}
}

% This sorts and then draws, showing where the pivot chosen in the previous
% step go. Next time they will have become "inert". If pivotcount is still at
% zero on exit from \QSpivotStep, then this is the signal to stop before
% executing \QSsortStep.
\def\QSsortStep {\def\QSLr {\noexpand\QS@sort@a}%
                 \def\QSRr {\noexpand\QS@sort@a}%
                 \def\QSIr {\noexpand\QSIrr}%
                 \let\QSIrr\relax
                    \edef\QS@list{\QS@list}%
                \let\QSLr\relax
                \let\QSRr\relax
                \let\QSIr\relax
                    \edef\QS@list{\QS@list}%
                \let\QSLr\DecoLEFT
                \let\QSIr\DecoINERTwithPivot
                \let\QSIrr\DecoINERT
                \let\QSRr\DecoRIGHT
\par\centerline{\rule[1.5mm]{0pt}{8mm}%
            \setcounter{cellcount}{0}%
            \begin{tikzpicture}\QS@list\end{tikzpicture}}
}

\def\QSinitialize #1{%
    % first, we convert the comma separated values into a list of braced items
    % we use an \edef, and anyhow many \edef's will be used later
    \edef\QS@list {\noexpand\QSRr {\xintCSVtoList {#1}}}%
    \let\QSRr\DecoRIGHT
    % The \QSRr marker mutated to draw the last element as
    % pivot and the earlier ones with the suitable style.
    %
    % The list of marked braced items \QS@list is used both for drawing
    % (as here) and for doing the exchange of elements during sort.
    \par\centerline{\rule[1.5mm]{0pt}{8mm}\setcounter{cellcount}{0}%
                \begin{tikzpicture}\QS@list\end{tikzpicture}}
}
```

2.35 relationale-algebra.sty

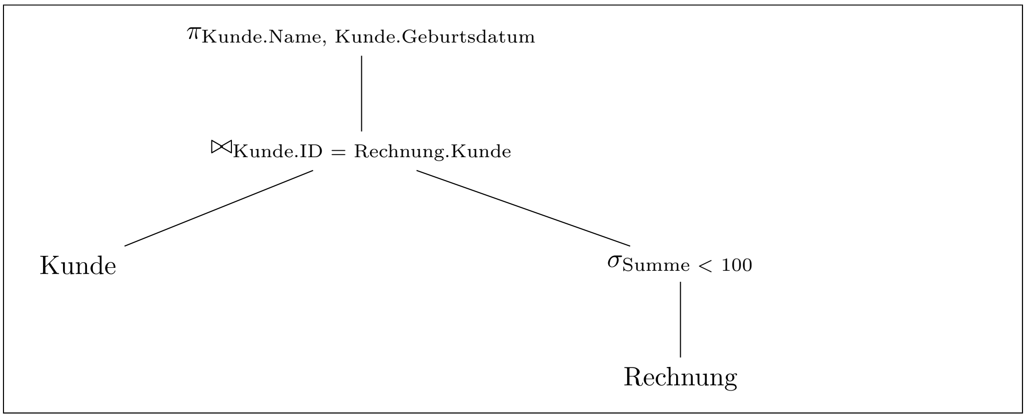
```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-relationale-algebra}[2020/12/11]
```

```
\RequirePackage{amsmath}
```

```
\RequirePackage{amssymb}
```

Zum Zeichen 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}
```




```
\RequirePackage{tikz}
\usetikzlibrary{positioning}
```

Privates Makros, das zwei Querstriche erzeugt.

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

`\leftouterjoin` A \leftouterjoin B: $A ⟕ B$

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

`\rightouterjoin` A \rightouterjoin B: $A ⟖ B$

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

`\fullouterjoin` A \fullouterjoin B: $A ⟗ B$

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

2.36 rmodell.sty

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

2.36.1 Makro-Kürzel

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

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

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

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

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

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

```
\def\li@Rmodell@Schrift{\footnotesize\ttfamily}
\ExplSyntaxOn
\NewDocumentEnvironment { liRmodell }
{ +b }
{
  \medskip
  {
    \linespread{2}
    \setlength{\parindent}{0pt}
    \li@Rmodell@Schrift#1
  }
  \medskip
} {}
\ExplSyntaxOff
```

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

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

```
\def\liRelationMenge#1#2{
\noindent
#1 : \{[ #2 ]\}
\par
}
```

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

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

```
\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}
```

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

2.37 sortieren.sty

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

```
\RequirePackage{tikz}
\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.

```
\def\liVertauschen#1{
  \directlua{
    local sortieren = require('lehramt-informatik-sortieren')
    sortieren('#1')
  }
}
```

`\liSortierPfeil`

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

`\liSortierPfeilUnten`

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

`\liSortierMarkierung`

```
\def\liSortierMarkierung#1#2{\node[
  draw,
  very thick,
  fit=(reihe.#1) (reihe.#2),
  inner sep=0pt
] {};
}
```

```
\tikzset{
  li sortierung zahlenreihe/.style={
    draw,
    thin,
    font=\large,
    rectangle split horizontal,
    rectangle split,
  }
}
```

```
% https://tex.stackexchange.com/a/140895
\RequirePackage{forest,xstring}
\usetikzlibrary{calc}

\makeatletter
\pgfmathdeclarefunction{strrepeat}{2}{%
  \begingroup\pgfmathint{#2}\pgfmath@count\pgfmathresult
    \let\pgfmathresult\pgfutil@empty
    \pgfutil@loop\ifnum\pgfmath@count>0\relax
      \expandafter\def\expandafter\pgfmathresult\expandafter{\pgfmathresult#1}%
      \advance\pgfmath@count-1\relax
    \pgfutil@repeat\pgfmath@smuggleone\pgfmathresult\endgroup}
\makeatother

\def\myNodes{}

\ExplSyntaxOn
\newcommand*\sortList[1]{%
  \clist_sort:Nn#1{\int_compare:nNnTF{##1}>{##2}\sort_return_swapped:\sort_return_same:}}
\ExplSyntaxOff

\forestset{
  sort/.code={%
    \pgfmathparse{level()>\forestSortLevel}%
    \ifnum\pgfmathresult=0
      \StrSubstitute{\forestov{content}}{ }{,}[\myList]%
      \sortList\myList
      \StrSubstitute{\myList}{,}{ }[\myList]%
      \pgfmathparse{strrepeat("1",level())}%
      \xappto\myNodes{\noexpand\node at ($(\forestov{name}|-m)!-1!(\forestov{name})$)
        (m\forestov{name}) {\myList}}%
      \pgfmathparse{level()==\forestSortLevel}%
      \ifnum\pgfmathresult=1
        \forestOget{\forestov{@first}}{name}\forestFirst
        \forestOget{\forestov{@last}}{name}\forestLast
        \xappto\myNodes{{[<-]edge (\forestOv{\forestov{@first}}{name})
          \ifx\forestFirst\forestLast\else edge (\forestOv{\forestov{@last}}{name})\fi}}%
      \fi
      \ifnum\forestov{@parent}=0\else
        \xappto\myNodes{edge (m\forestOv{\forestov{@parent}}{name})}%
      \fi
      \gappto\myNodes{;}%
    \fi}}

\forestset{sort level/.code=%
  \pgfmathparse{#1}\let\forestSortLevel\pgfmathresult
  \pgfmathparse{strrepeat("1",\forestSortLevel+1)}\let\forestOnes\pgfmathresult}
```

2.38 spalten.sty

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

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

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

2.39 sql.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-tabelle}[2021/09/02 Zu Setzen von SQL]

\liLadePakete{syntax}

\RequirePackage{fancyvrb}
\DefineVerbatimEnvironment{liSqlErgebnis}{Verbatim}
{fontsize=\footnotesize}

```

2.40 struktogramm.sty

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

```

2.41 syntax.sty

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

2.41.1 Makro-Kürzel

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

```
\ExplSyntaxOn
```

```
\directlua{
  syntax = require('lehramt-informatik-syntax')
  syntax.importiere_konfiguration('tex_repo_lokaler_pfad', '\LehramtInformatikRepository')
  syntax.importiere_konfiguration('github_domain', '\LehramtInformatikGithubDomain')
  syntax.importiere_konfiguration('github_raw_domain', '\LehramtInformatikGithubRawDomain')
  syntax.importiere_konfiguration('github_tex_repo', '\LehramtInformatikGithubTexRepo')
  syntax.importiere_konfiguration('github_code_repo', '\LehramtInformatikGithubCodeRepo')
  syntax.importiere_konfiguration('git_branch', '\LehramtInformatikGitBranch')
}
```

```
\RequirePackage{hyperref}
```

```
\RequirePackage{minted}
% pygmentize -L styles
\usemintedstyle{colorful}
%\BeforeBeginEnvironment{minted}{\begin{mdframed}}
%\AfterEndEnvironment{minted}{\end{mdframed}}
%\setminted{breaklines=true,linenos}
\setminted{
  breaklines=true,
  linenos,
  fontsize=\footnotesize,
}
```

`\liJavaCode` Im Zeilenfluss einen kurzen Java-Code-Ausschnitt setzen.
Let-Abkürzung: `\let\j=\liJavaCode`

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

`\liLatexCode` Im Zeilenfluss einen kurzen LaTeX-Code-Ausschnitt setzen.

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

```
\def\li@GithubLink#1#2{
  \begin{flushright}
    \tiny
    Code-Beispiel~auf~Github~ansehen:~
    \href{#1}{\nolinkurl{#2}}
  \end{flushright}
}
```

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

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

```
    {\directlua{syntax.drucke_github_url('#2', false)}}
    {\directlua{syntax.drucke_relativen_repo_pfad('#2', false)}}
}
```

\liJavaTestDatei Eine komplette Java-Test-Datei einbinden, die Verzeichnis `./Code/src/test/java/org/bschlangaul` liegt.

```
\NewDocumentCommand{\liJavaTestDatei}{ O{firstline=3} m }{
  \inputminted[#1]{java}{
    \directlua{
      syntax.drucke_absoluten_pfad('#2', true)
    }
  }
  \li@GithubLink
    {\directlua{syntax.drucke_github_url('#2', true)}}
    {\directlua{syntax.drucke_relativen_repo_pfad('#2', true)}}
}
```

\liJavaExamen `\liJavaExamen{66116}{2015}{03}{Kunde}` Regulärer Ausdruck zum Konvertieren: `\\liJavaDatei(\[.` `\\liJavaExamen$1{$2}{$3}{$4}{$5}`

```
\NewDocumentCommand{\liJavaExamen}{ O{firstline=3} m m m m }{
  \inputminted[#1]{java}{
    \directlua{
      syntax.drucke_absoluten_examens_pfad('#2', '#3', '#4', '#5')
    }
  }

  \li@GithubLink
  {\directlua{syntax.drucke_github_examens_url('#2', '#3', '#4', '#5')}}
  {\directlua{syntax.drucke_relativen_examens_repo_pfad('#2', '#3', '#4', '#5')}}
}
```

\liAssemblerCode

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

\liAssemblerDatei `\liAssemblerDatei{relativer-pfad}`: Relativer Pfad zu einer Assembler-Datei.

```
\NewDocumentCommand{\liAssemblerDatei}{ m }{
  \inputminted{asm}{#1}
}
```

\liMinispracheDatei `\liMinispracheDatei{relativer-pfad}`: Relativer Pfad zu einer Minisprachen-Datei (Hochsprache für die Minimaschine von Albert Wiedemann).

```
\NewDocumentCommand{\liMinispracheDatei}{ m }{
  \inputminted{componentpascal}{#1}
}
```

\liHaskellCode `\liHaskellCode{haskell}`: Zum Setzen von Haskell-Code.

```
\def\liHaskellCode#1{\mintinline{haskell}|#1|}
```

\liHaskellDatei `\liHaskellDatei{relativer-pfad}`: Relativer Pfad zu einer Haskell-Datei.

```
\NewDocumentCommand{\liHaskellDatei}{ m }{
  \inputminted{haskell}{#1}
}

\ExplSyntaxOff
```

\liSqlCode `\liHaskellCode{sql}`: Zum Setzen von SQL-Code.
Let-Abkürzung: `\let\s=\liSqlCode`

```
\def\liSqlCode#1{\mintinline{sql}|#1|}

```

2.42 syntaxbaum.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-syntaxbaum}[2021/02/14 Zum Setzen von
Syntaxbäumen mit Hilfe des Pakets tikz-qtree]
\RequirePackage{tikz-qtree}

\tikzset{li parsetree/.style={
    every internal node/.style={
      draw,circle
    },
    every leaf node/.style={
      draw,rectangle
    },
  }
}

```

2.43 synthese-algorithmus.sty

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

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

2.43.1 Makro-Kürzel

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

2.43.2 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}
```

2.43.3 TeX-Markup Linksreduktion

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

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

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

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

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

2.43.4 TeX-Markup Rechtreduktion

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

\liPseudoUeberschrift{F}

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

\liPseudoUeberschrift{A}

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

2.43.5 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 Anhängigkeit $\alpha \to \beta \in F$ die Linksreduktion durch, überprüfe also für alle $A \in \alpha$, ob A überflüssig ist, d. h. ob $\beta \subseteq AttrHülle(F, \alpha - A)$.*

   (b) **Rechtsreduktion**

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

   (c) **Löschen leerer Klauseln**

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

   (d) **Vereinigung**

   — *Fasse mittels der Vereinigungsregel funktionale Abhängigkeiten der Form $\alpha \to \beta_1, \ldots, \alpha \to \beta_n$, so dass $\alpha \to \beta_1 \cup \cdots \cup \beta_n$ verbleibt.* —

2. **Relationsschemata formen**

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

3. **Schlüssel hinzufügen**

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

4. **Entfernung überflüssiger Teilschemata**

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

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

```
\def\liSyntheseUeberschrift#1{
  {
    \bfseries
    \sffamily
    \str_case:nn {#1} {
      {1} {Kanonische~Überdeckung}
      {1-1} {Linksreduktion}
      {1-2} {Rechtsreduktion}
      {1-3} {Löschen~leerer~Klauseln}
      {1-4} {Vereinigung}
      {2} {Relationsschemata~formen}
      {3} {Schlüssel~hinzufügen}
      {4} {Entfernung~überflüssiger~Teilschemata}
    }
```

```
  }
}
```

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

```
\def\li@synthese@erklaerung@texte#1{
  \str_case:nn {#1} {
    {1} {
      Die~kanonische~Überdeckung~--~also~die~kleinst~mögliche~noch~
      äquivalente~Menge~von~funktionalen~Abhängigkeiten~kann~in~vier~
      Schritten~erreicht~werden.
    }
    {1-1} {
      Führe~für~jede~funktionale~Anhängigkeit~
      $\alpha~\rightarrow~\beta~\in~F$~die~Linksreduktion~durch,~
      überprüfe~also~für~alle~
      $A~\in~\alpha$,~ob~$A$~überflüssig~ist,~d.~h.~ob~
      $\beta~\subseteq~\liAttributHuelle{F,~\alpha~--A}.$
    }
    {1-2} {
      Führe~für~jede~(verbliebene)~funktionale~Abhängigkeit~$\alpha~
      \rightarrow~\beta$~die~Rechtsreduktion~durch,~überprüfe~also~für~
      alle~$B~\in~\beta$,~ob~$B~\in~\liAttributHuelle{F~--(\alpha~
      \rightarrow~\beta)~\cup~(\alpha~\rightarrow~(\beta~--B)),~
      \alpha}$~gilt.~In~diesem~Fall~ist~B~auf~der~rechten~Seite~
      überflüssig~und~kann~eleminiert~werden,~\dh~$\alpha~
      \rightarrow~\beta$~wird~durch~$\alpha~\rightarrow~(\beta~--B)$~
      ersetzt.
    }
    {1-3} {
      Entferne~die~funktionalen~Abhängigkeiten~der~Form~$\alpha~
      \rightarrow~\emptyset$,~die~im~2.~Schritt~möglicherweise~
      entstanden~sind.
    }
    {1-4} {
      Fasse~mittels~der~Vereinigungsregel~funktionale~Abhängigkeiten~
      der~Form~$\alpha~\rightarrow~\beta\sb{1},~\dots,~\alpha~\rightarrow~
      \beta\sb{n}$,~so~dass~$\alpha~\rightarrow~\beta\sb{1}~\cup~\dots~\cup~
      \beta\sb{n}$~verbleibt.
    }
    % Kemper Seite 197
    {2} {
      Erzeuge~für~jede~funktionale~Abhängigkeit~$\alpha~\rightarrow~
      \beta~\in~F\sb{c}$~ein~Relationenschema~$\mathcal{R}\sb{\alpha}~
      :=~\alpha~\cup~\beta$.
    }
    {3} {
      Falls~eines~der~in~Schritt~2.~erzeugten~Schemata~$R\sb{\alpha}$~
      einen~Schlüsselkandidaten~von~$\mathcal{R}$~bezüglich~$F\sb{c}$~
      enthält,~sind~wir~fertig,~sonst~wähle~einen~Schlüsselkandidaten~
      $\mathcal{K}~\subseteq~\mathcal{R}$~aus~und~definiere~folgendes~
      zusätzliche~Schema:~$\mathcal{R}\sb{\mathcal{K}}~:=~\mathcal{K}$~
      und~$\mathcal{F}\sb{\mathcal{K}}~:=~\emptyset$
    }
    {4} {
      Eliminiere~diejenigen~Schemata~$R\sb{\alpha}$,~die~in~einem~
      anderen~Relationenschema~$R\sb{\alpha'}$~enthalten~sind,~d.~h.~
      $R\sb{\alpha}~\subseteq~R\sb{\alpha'}$.
    }
  }
}
\def\liSyntheseErklaerung#1{
  {
    \itshape
    \footnotesize
```

```
    \liParagraphMitLinien{\li@synthese@erklaerung@texte{#1}}
  }
}
```

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

```
\def\liSyntheseUeberErklaerung#1{
  \liSyntheseUeberschrift{#1}\par
  \liSyntheseErklaerung{#1}
}

\ExplSyntaxOff
```

2.44 tabelle.sty

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

```

2.45 typographie.sty

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

```
\ExplSyntaxOn
```

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

```
\RequirePackage{fontawesome}
```

`\liErledigt` `\liErledigt`: ☑

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

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

```
\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 ———————————————————————————————

```
\def\liParagraphMitLinien#1{
  \noindent
  \vrule height 2pt depth -1.6pt width 0.4cm
  \enspace
  #1
  \enspace
  \leaders\vrule height 2pt depth -1.6pt \hfill \null
  \par
  \medskip
}
```

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

$$\text{Variable} = \Bigl\{ \qquad \text{Inhalt} \qquad \Bigr\}$$

```
\def\liGeschweifteKlammern#1#2#3#4{
  \par
  \medskip
  \noindent
  #1 \, $= \Bigl\{$
  \vspace{#3}
  #2
  \vspace{#4}
  \begin{flushright}$\Bigr\}$\end{flushright}
  \par
}
```

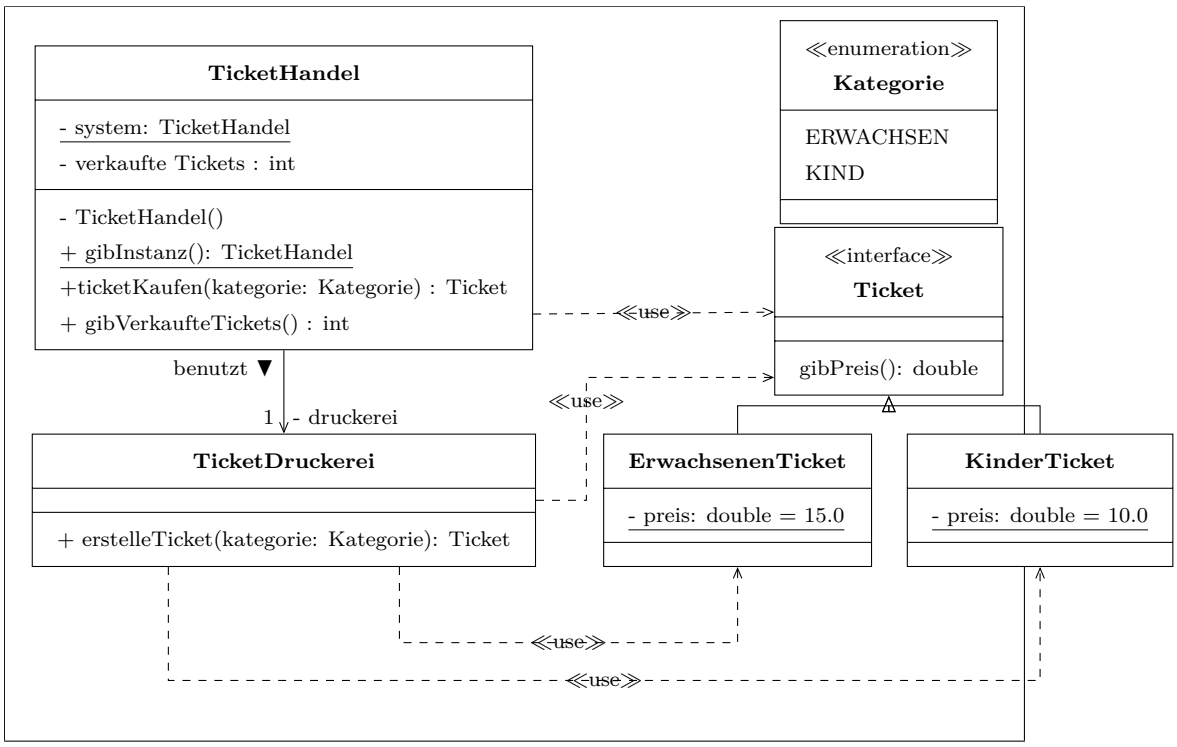
```
\ExplSyntaxOff
```

2.46 uml.sty

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

```
\RequirePackage{tikz-uml}
\RequirePackage{tikz-uml-activity}
% Not compatible with wasysym
%\RequirePackage{mathabx}
\RequirePackage{wasysym}
\usetikzlibrary{positioning}
```

```
\tikzumlset{
  fill class=white!0,
  font=\footnotesize,
  fill object=white!0,
  fill note=white!0,
  fill state=white!0,
  % Use case
  fill usecase=white!0,
  fill system=white!0,
}
```



\liUmlLeserichtung

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

```
\NewDocumentCommand{ \liUmlLeserichtung } { O{dir=right} m } {
  \def\@liDirLeft{}
  \def\@liDirRight{ \RIGHTarrow}
  \pgfkeys{/lese/dir/.is choice}
  \pgfkeys{/lese/dir/up/.code={\def\@liDirRight{ \UParrow}}}
  \pgfkeys{/lese/dir/down/.code={\def\@liDirRight{ \DOWNarrow}}}
  \pgfkeys{/lese/dir/left/.code={\def\@liDirRight{}\def\@liDirLeft{\LEFTarrow }}}
  \pgfkeys{/lese/dir/right/.code={\def\@liDirRight{ \RIGHTarrow}}}

  \def\@liPos{above}
  \pgfkeys{/lese/pos/.code={\def\@liPos{##1}}}

```

```
  \def\@liDistance{0cm}
  \pgfkeys{/lese/distance/.code={\def\@liDistance{##1}}}

  \pgfkeys{/lese/.cd,#1}

  \node[\@liPos = \@liDistance of #2-middle] {
    \@liDirLeft{\footnotesize#2}\@liDirRight
  };
}
```

2.47 vollstaendige-induktion.sty

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

2.47.1 Makro-Kürzel

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

```
\begin{align*}
C_{n+1}
& = \frac
    {(4 \cdot (\m{n + 1} - 1) + 2) \cdot \text{cn}(\m{n + 1} - 1)}
    {\m{n + 1} + 1}
& \e{Java nach Mathe}\\
%
& = \frac
    {(4\m{n} + 2) \cdot \text{cn}(\m{n})}
    {\m{n + 2}}
& \e{addiert, subtrahiert}\\
%
& = \frac
    {(4n + 2) \cdot \m{(2n)!}}
    {(n + 2) \cdot \m{(n + 1)! \cdot n!}}
& \e{für cn(n) Formel eingesetzt}\\
%
& = \frac
    {(4n + 2) \cdot (2n)! \m{\cdot (n + 1)}}
    {(n + 2) \cdot (n + 1)! \cdot n! \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

```
\RequirePackage{lehramt-informatik-typographie}
\RequirePackage{lehramt-informatik-mathe}
\RequirePackage{lehramt-informatik-syntax}
```

```
\ExplSyntaxOn
```

`\liInduktionMarkierung` Hilfsmakro um Teile von mathematischen Formeln markieren zu können.
Let-Abkürzung: `\let\m=\liInduktionMarkierung`

```
\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`

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

`\liInduktionAnfang`

```
\def\liInduktionAnfang{
  \liPseudoUeberschrift{Induktionsanfang}

  % https://de.wikibooks.org/wiki/Mathe_für_Nicht-Freaks:_Vollständige_Induktion
  \liParagraphMitLinien{
    Beweise,~dass~$A(1)$~eine~wahre~Aussage~ist.
  }
}
```

`\liInduktionVoraussetzung`

```
\def\liInduktionVoraussetzung{
  \liPseudoUeberschrift{Induktionsvoraussetzung}

  % https://de.wikibooks.org/wiki/Mathe_für_Nicht-Freaks:_Vollständige_Induktion
  \liParagraphMitLinien{
    Die~Aussage~$A(k)$~ist~wahr~für~ein~beliebiges~$k \in \mathbb{N}$.
  }
}
```

`\liInduktionSchritt`

```
\def\liInduktionSchritt{
  \liPseudoUeberschrift{Induktionsschritt}

  % https://de.wikibooks.org/wiki/Mathe_für_Nicht-Freaks:_Vollständige_Induktion
  \liParagraphMitLinien{
    Beweise,~dass~wenn~$A(n=k)$~wahr~ist,~
    auch~$A(n=k+1)$~wahr~sein~muss.
  }
}
```

```
\ExplSyntaxOff
```

2.48 wasserfall.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-wasserfall}[2020/06/10]
\RequirePackage{tikz}
\tikzset{wasserfall/.style={
  >=stealth,
  node distance = 2mm and -8mm,
  start chain = A going below right,
  every node/.style = {
    draw,
    text width=24mm,
    minimum height=12mm,
    align=center,
    inner sep=1mm,
    fill=white,
    drop shadow={fill=black},
    on chain=A
  },
}}
\usetikzlibrary{chains,positioning,shadows}

```

2.49 wpkalkuel.sty

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

2.49.1 Makro-Kürzel

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

```
\RequirePackage{amsmath}
\ExplSyntaxOn
```

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

```
\def\liWpKalkuelOhneMathe#1#2{
  \text{wp}(\texttt{\scriptsize"#1"},\thinspace #2)
}
\def\liWpKalkuel#1#2{
  \ifmmode
    \liWpKalkuelOhneMathe{#1}{#2}
  \else
    $\liWpKalkuelOhneMathe{#1}{#2}$
  \fi
}
```

\MatheEnv

```
\def\MatheEnv#1{
  \medskip

  \hspace{1em}#1

  \medskip
}
```

\Mathe

```
\def\Mathe#1{
  \MatheEnv{$#1$}
}
```

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

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

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

```
\newlength{\@Skip@Erklaerung@Reset}
\def\liWpErklaerung#1{
  \setlength{\@Skip@Erklaerung@Reset}{\leftskip}
  \setlength{\leftskip}{0.5cm}

  \par
  \noindent
  {
    \scriptsize
    #1
  }
  \par

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

\liWpErklaerungVerzweigung

```
\def\liWpErklaerungVerzweigung{
  $\liWpKalkuelOhneMathe{if~\{~b~\}~then~\{~a1~\}~else~\{~a2~\}}{Q}
  \equiv
  (b \land \liWpKalkuelOhneMathe{a1}{Q})
  \lor
  (\neg b \land \liWpKalkuelOhneMathe{a2}{Q})$
}
```

```
\ExplSyntaxOff
```

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

\# 122
\, ... 356, 411, 1194, 1196, 1719, 1720, 1721, 2055, 2532, 2718
\@Skip@Erklaerung@Reset . . . 2853, 2855, 2866
\@afterheading 1686
\@afterindentfalse . 1686
\@liDirLeft 2748, 2753, 2765
\@liDirRight 2749, 2751, 2752, 2753, 2754, 2765
\@liDistance 2759, 2760, 2764
\@liPos . . 2756, 2757, 2764
\\ 620, 643, 644, 647, 648, 651, 652, 744, 745, 746, 853, 882, 884, 910, 919, 964, 1006, 1007, 1008, 1013, 1014, 1015, 1035, 1593, 1950, 2008, 2011
\{ 220, 1131, 1141, 1153, 1154, 1159, 1193, 1433, 1995, 2400, 2718, 2869
\} 220, 1131, 1141, 1153, 1154, 1161, 1197, 1434, 1995, 2400, 2722, 2869
_ 40, 48, 57, 59, 306, 331, 355, 356, 370, 371, 377, 380, 383, 393, 405, 1374
\␣ 2007, 2017

A

\addbibresource 1563, 1564, 1565, 1566, 1567, 1568, 1569, 1570, 1571, 1572
\AddToHook 1373
\advance 2449
\AfterEndEnvironment 2525
\allsectionsfont . . . 1243
\Alph 1257
\alph 1257, 1258
\alpha 2632, 2634, 2635, 2638, 2640, 2641, 2642, 2643, 2644, 2648, 2654, 2655, 2660, 2661, 2662, 2665, 2673, 2674, 2675
\arabic 1257, 2244, 2249, 2254, 2260, 2266, 2272
\arraystretch 1945

B

\BeforeBeginEnvironment 2524
\begin 641, 692, 707, 742, 766, 813, 845, 860, 880, 890, 906, 926, 958, 975, 1004, 1029, 1050, 1065, 1174, 1263, 1338, 1387, 1396, 1403, 1511, 1585, 1637, 1642, 1650, 1675, 1680, 1691, 1701, 1705, 1788, 1792, 1808, 1831, 1854, 1869, 1948, 1949, 1999, 2044, 2164, 2184, 2325, 2347, 2361, 2524, 2535, 2722
\begingroup 1586, 1997, 2445
\beschriftung 1617, 1621, 1625, 1629, 1633, 1635
\beta 2632, 2635, 2639, 2640, 2641, 2644, 2654, 2655, 2656, 2661, 2662
\bf 2222, 2223, 2224
\bfseries . . 499, 1247, 1249, 2222, 2228, 2230, 2232, 2233, 2609
\Bigl 2718
\Bigr 2722
\bigskip 61, 389, 622, 627, 1595, 1923
\bool 334, 357
\bowtie 2370, 2373, 2374, 2375
\Box 172
\boxtimes 487

C

\c 1340, 1341
\cdot 1763, 1816, 1827
\centerline 1392, 2323, 2345, 2360
\chapter 1247, 1248
\char 1593
\clearpage 1587
\cline 620
\clist 239, 283, 284, 303, 307, 2457
\columnbreak 2492
\cs 306, 331, 355, 356, 393, 405, 1669
\csname 1325, 1328
\cup 1154, 2029, 2641, 2655, 2662

D

\DeclareMathSymbol 1893, 1894
\DecoINERT 2247, 2320, 2321, 2343
\DecoINERTwithPivot 2263, 2342
\DecoLEFT 2242, 2341
\DecoLEFTwithPivot 2257, 2319
\DecoRIGHT 2252, 2344, 2354
\DecoRIGHTwithPivot 2269, 2322

\definecolor 1245
\DefineVerbatimEnvironment 2498
\delta 78, 120, 178, 220, 1145
\dh 1721, 2643
\directlua 71, 150, 208, 213, 1126, 1140, 1160, 1168, 1175, 1180, 1980, 1985, 2038, 2045, 2052, 2412, 2511, 2543, 2548, 2549, 2553, 2558, 2559, 2563, 2569, 2570
\do 2243, 2248, 2253, 2258, 2264, 2270
\dots 530, 534, 1433, 2171, 2654, 2655
\DOWNarrow 2752
\draw 1345, 1348, 1351, 2119, 2418, 2421

E

\edef 1336, 2336, 2340, 2352, 2353
\else 594, 602, 610, 1135, 1149, 1185, 1633, 1736, 1746, 1756, 1770, 1991, 2028, 2114, 2282, 2475, 2477, 2836
\emph 1097, 1414, 1443, 1445, 1592
\empty 1633, 2027
\emptyset 1937, 2649, 2670, 2703
\end 680, 703, 728, 763, 797, 829, 857, 873, 887, 895, 922, 951, 972, 997, 1022, 1042, 1062, 1080, 1176, 1265, 1354, 1399, 1400, 1407, 1516, 1590, 1639, 1647, 1666, 1682, 1683, 1697, 1708, 1709, 1790, 1803, 1828, 1851, 1866, 1876, 1952, 1953, 2001, 2046, 2174, 2194, 2325, 2347, 2361, 2525, 2539, 2722
\endcsname 1325, 1328
\endgroup 1589, 2002, 2450
\enspace 2707, 2709
environments:
liAdditum 1640
liAHuelle 1996
liAntwort 1611
liDiagramm 1699
liEinbettung 1610
liExkurs 1648
liGraphenFormat . 1317
liKasten 1262
liKontrollflussgraph 1510
liLernkartei 1689
liProduktionsRegeln 1169
liProjektSprache 1609
liQuellen 1669
liRelationenSchemaFormat 2404
liRmodell 2384
liUebergangsTabelle 1945
\equiv 2851, 2870
\erzeuge@tiefgestellt ... 1140, 1141, 1145
\expandafter 1325, 2278, 2280, 2281, 2282, 2290, 2448
\ExplSyntaxOff 63, 105, 147, 152, 205, 210, 215, 552, 574, 589, 1121, 1199, 1234, 1376, 1531, 1547, 1711, 1786, 1973, 2147, 2397, 2458, 2583, 2690, 2725, 2804, 2875
\ExplSyntaxOn 35, 74, 115, 148, 173, 206, 211, 236, 494, 558, 575, 1094, 1189, 1211, 1372, 1524, 1536, 1612, 1725, 1955, 2128, 2385, 2455, 2510, 2606, 2700, 2776, 2829

F

\faCheckSquareO 2702
\faCircleThin 1113
\faGg 1107
\fancyfoot 1269, 1270, 1271, 1540, 1541, 1542, 1543
\fancyhead 1268, 1537, 1538, 1539
\faSquareO 1101
\fi 596, 604, 612, 1137, 1151, 1187, 1636, 1738, 1748, 1758, 1772, 1993, 2030, 2116, 2283, 2475, 2476, 2479, 2481, 2838
\fontspec 1243
\footcite 704, 726, 773, 796, 828, 923, 950, 996, 1412, 1415, 1422, 1427, 1432, 1436, 1442, 1447, 1552, 1804, 1805, 1960, 2177
\footnote 1713, 1717
\footnotesize 160, 366, 448, 544, 1118, 1608, 1661, 1676, 1692, 1904, 1998, 2015, 2023, 2132, 2141, 2384, 2499, 2530, 2682, 2739, 2765
\footrulewidth . 1274, 1545
\foreach . 1340, 1343, 1350
\forestFirst .. 2472, 2475
\forestLast ... 2473, 2475
\forestOget ... 2472, 2473
\forestOnes 2485
\forestOv 2474, 2475, 2478
\forestov . 2464, 2468, 2469, 2472, 2473, 2474, 2475, 2477, 2478
\forestset 2460, 2483
\forestSortLevel 2462, 2470, 2484, 2485
\frac 1765, 1796, 1827, 1842
\fullouterjoin 2375

G

\g 42, 50, 283, 284, 303, 307, 313, 314, 315, 316, 317, 319, 320, 322, 324, 325, 326, 327, 328, 329, 332, 336, 337, 338, 341, 343, 344, 345, 346, 347, 348, 359, 360, 361, 362, 370, 371, 373, 379, 380, 382, 383, 385, 386, 394, 399, 401, 406, 408, 412
\Gamma . 119, 177, 220, 1154
\gappto 2480
\geometry 5
\geq 1440, 2160, 2165, 2181, 2185

H

\hbox 2370
\headrulewidth . 1273, 1544
\headwidth 1546
\hfill 1812, 1818, 1823, 2710
\hinweis 1608
\hline 1950
\href 395, 1207, 1717, 1880, 2538
\hspace .. 2122, 2843, 2851
\ht 2371

I

\i 1350, 1351
\ifcase 2279
\ifmmode 592, 600, 608, 1133, 1147, 1183, 1734, 1744, 1754, 1768, 1989, 2112, 2834
\IfNoValueTF 1643, 1713, 1717
\ifnum 2206, 2447, 2463, 2471, 2477
\ifx 1633, 2027, 2475

\in 518, 624, 1440, 1778, 1781, 1784, 1810, 1816, 1821, 2160, 2171, 2181, 2191, 2632, 2634, 2640, 2661, 2792
\inhaltsverzeichnis 1584
\input 17, 20, 23, 26, 29, 418
\inputminted 2542, 2552, 2562, 2574, 2577, 2581
\int 2457
\item 487, 488, 709, 713, 718, 723, 767, 776, 781, 789, 861, 866, 870, 891, 927, 932, 939, 947, 976, 981, 985, 990, 1066, 1071, 1076, 1397, 1398, 1669, 1673, 1793, 1796, 1800, 1809, 1815, 1820, 1832, 1836, 1840, 1844, 1848, 1855, 1859, 1863, 2165, 2168, 2171, 2185, 2188, 2191
\itshape 543, 2681

J

\j 1340, 1341, 1343, 1344, 1345, 1350, 1351, 1352

K

\k 1350
\keys 44, 83, 95, 125, 135, 183, 193, 311, 562, 566, 580, 585, 1218, 1225

L

\l . . 76, 77, 78, 79, 80, 81, 84, 85, 86, 87, 88, 90, 92, 97, 98, 99, 100, 101, 102, 117, 118, 119, 120, 121, 122, 123, 126, 127, 128, 129, 130, 131, 132, 138, 139, 140, 141, 142, 143, 144, 175, 176, 177, 178, 179, 180, 181, 184, 185, 186, 187, 188, 189, 190, 196, 197, 198, 199, 200, 201, 202, 560, 563, 568, 569, 572, 577, 578, 581, 582, 587, 1213, 1214, 1215, 1216, 1219, 1220, 1221, 1222, 1228, 1229, 1230, 1231, 1527, 1528, 1529, 1672, 1673, 1674, 1681
\labelenumi 1258
\labelenumii 1259
\labelitemi 1252
\labelitemii 1253
\labelitemiii 1254
\labelitemiv 1255
\land 2871, 2873
\LARGE 1247
\large 1392, 2434
\leaders 2710
\left 1728
\LEFTarrow 2753
\leftarrow 606
\leftouterjoin 2373
\leftskip 2855, 2856, 2866
\LehramtInformatikAutorEmail 1543
\LehramtInformatikAutorName 1542
\LehramtInformatikGitBranch 398, 2518
\LehramtInformatikGithubCodeRepo 2517
\LehramtInformatikGithubDomain 2514
\LehramtInformatikGithubRawDomain 396, 2515
\LehramtInformatikGithubTexRepo 397, 2516
\LehramtInformatikRepository 17, 20, 23, 26, 29, 1563, 1564, 1565, 1566, 1567, 1568, 1569, 1570, 1571, 1572, 2513
\LehramtInformatikTitel 1538
\leq 1827, 2168, 2188
\let 1143, 1144, 1587, 2319, 2320, 2321, 2322, 2335, 2337, 2338, 2339, 2341, 2342, 2343, 2344, 2354, 2446, 2484, 2485, 2702
\li@chomsky@erklaerung@texte 509, 545
\li@EntwurfsCode 637, 683, 684, 685, 731, 732, 733, 734, 800, 801, 802, 803, 804, 805, 832, 833, 834, 835, 836, 837, 898
\li@EntwurfsCodeAllgemein 636
\li@fussnote@text 1901, 1907, 1911, 1915, 1919
\li@GithubLink 2534, 2547, 2557, 2568
\li@mget . 1327, 1331, 1351
\li@minc 1330, 1352
\li@mset 1324, 1332, 1341, 1344
\li@numdiscs 1336, 1345, 1351
\li@Rmodell@Schrift 2384, 2393, 2403
\li@sequence . . 1337, 1350
\li@synthese@erklaerung@texte 2623, 2683
\liableitung 1168
liAdditum (environment) 1640
liAHuelle (environment) 1996
\liAlphabet 1153
liAntwort (environment) 1611
\liAnweisung 1518
\liAssemblerCode . . . 2572
\liAssemblerDatei . . 2573
\liAttribut 2403
\liAttributHuelle 1987, 2635, 2640
\liAttributHuelleOhneMathe 1987, 1990, 1992, 2006, 2016, 2024
\liAttributMenge 1995, 2007, 2010, 2017, 2018, 2032, 2034
\liAufgabe 16
\liAufgabenMetadaten . 38
\liAufgabenTitel 36
\liAusdruck 1190
\liAutomat 74
\liAutomatenKante . . . 106
\liBandAlphabet 1154
\liBedingung 1519
\liBedingungDrei 1783, 1823, 1863
\liBedingungEins 1777, 1812, 1855
\liBedingungFalsch . 1521
\liBedingungWahr . . . 1520
\liBedingungZwei 1780, 1818, 1859
\liBeschriftung 1600
\liChomskyErklaerung 509, 550
\liChomskyUeberErklaerung 548
\liChomskyUeberschrift 497, 549
\liCpmEreignis 558
\liCpmFruehesterI . . . 615
\liCpmSpaetesterI . . . 614
\liCpmVon 598
\liCpmVonOhneMathe 598, 601, 603
\liCpmVonZu 590
\liCpmVonZuOhneMathe 590, 593, 595
\liCpmVorgang 575
\liCpmZu 606
\liCpmZuOhneMathe 606, 609, 611

liDiagramm (environment) 1699
liEinbettung (environment) 1610
\liEntwurfs 1045
\liEntwurfsAbstrakteFabrik 687
\liEntwurfsAbstrakteFabrikCode 682, 689
\liEntwurfsAbstrakteFabrikUml 640, 688
\liEntwurfsAdapter . . 736
\liEntwurfsAdapterAkteure 706, 738
\liEntwurfsAdapterCode 730, 739
\liEntwurfsAdapterUml 691, 737
\liEntwurfsBeobachter 807
\liEntwurfsBeobachterAkteure 765, 809
\liEntwurfsBeobachterCode 799, 810
\liEntwurfsBeobachterUml 741, 808
\liEntwurfsDekorierer 839
\liEntwurfsDekoriererAkteure 841
\liEntwurfsDekoriererCode 831, 842
\liEntwurfsDekoriererUml 812, 840
\liEntwurfsEinfacheFabrik 875
\liEntwurfsEinfacheFabrikAkteure 859, 877
\liEntwurfsEinfacheFabrikUml 844, 876
\liEntwurfsEinzelstueck 900
\liEntwurfsEinzelstueckAkteure 889, 902
\liEntwurfsEinzelstueckCode 897, 903
\liEntwurfsEinzelstueckUml 879, 901
\liEntwurfsErbauer . . 953
\liEntwurfsErbauerAkteure 925, 955
\liEntwurfsErbauerUml 905, 954
\liEntwurfsFabrikmethode 999, 1024
\liEntwurfsFabrikmethodeAkteure 974, 1001
\liEntwurfsFabrikmethodeUml 957, 1000
\liEntwurfsKompositum 1024
\liEntwurfsKompositumAkteure 1026
\liEntwurfsKompositumUml 1003, 1025
\liEntwurfsModellPraesentation 1045
\liEntwurfsModellPraesentationAkteure 1047
\liEntwurfsModellPraesentationSteuerungUml 1028, 1046
\liEntwurfsZustand . 1082
\liEntwurfsZustandAkteure 1064, 1084
\liEntwurfsZustandUml 1049, 1083
\liEpsilon 1139
\liErAttribute 1097, 1111, 1113
\liErDatenbankName . 1116
\liErEntity 1095, 1099, 1101
\liErledigt 2702
\liErMpAttribute . . . 1110
\liErMpEntity 1098
\liErMpRelationship 1104
\liErRelationship 1096, 1105, 1107
\liExamensAufgabe 19
\liExamensAufgabeA . . . 28
\liExamensAufgabeTA . . 25
\liExamensAufgabeTTA . 22
liExkurs (environment) 1648
\liFalsch 488
\liFlaci 1200
\liFremd 2383
\liFunktionaleAbhaengigkeit . . . 2026, 2029, 2037
\liFunktionaleAbhaengigkeiten 2040
\liFussnote . . . 1900, 1902
\liFussnoteDreiText 1914, 1932
\liFussnoteEinsText 1906, 1926
\liFussnoteLink 1716
\liFussnoten 1922
\liFussnoteUrl . 1043, 1712
\liFussnoteVierText 1918, 1935
\liFussnoteZweiText 1910, 1929
\liGeschweifteKlammern . . . 1172, 2041, 2714
\liGrammatik 1211
liGraphenFormat (environment) 1317
\liHanoi 1324
\liHaskellCode 2579
\liHaskellDatei 2580
\liInduktionAnfang . 2779
\liInduktionErklaerung 2778
\liInduktionMarkierung 2777
\liInduktionSchritt 2795
\liInduktionVoraussetzung 2787
\liJavaCode 2532
\liJavaDatei . . 638, 2541
\liJavaExamen 2561
\liJavaTestDatei . . 2551
liKasten (environment) 1262
\liKellerAutomat 115
\liKellerKante 153
\liKellerUebergang 148, 154
\liKontrollCode 1522
liKontrollflussgraph (environment) 1510
\liKontrollKnotenPfad 1524
\liKontrollTextzeileKnoten 1523, 1528
\liKurzeTabellenLinie 620
\liLadeAllePakete . . . 241
\liLadePakete 67, 70, 237, 242, 496, 557, 1130, 1360, 1382, 1899, 1979, 2127, 2496, 2605
\liLatexCode 2533
\liLeereZelle 1937
liLernkartei (environment) 1689
\liLinksReduktion . . 2004
\liLinksReduktionInline 2013, 2021
\liMasterExkurs 1868
\liMasterFaelle 1807, 1875
\liMasterFallRechnung 1853
\liMasterVariablen 1787, 1870
\liMasterVariablenDeklaration 1830
\liMasterWolframLink 1878
\liMenge . . . 84, 85, 87, 126, 127, 128, 132, 184, 185, 186, 190, 1131, 1180, 1219, 1220
\liMengeOhneMathe 1131, 1134, 1136
\liMinimierungErklaerung 1959
\liMinispracheDatei 2576
\linespread 2391
\liNichtsZuTun 2703
\liO 1750, 1778
\liOmega 1740, 1784
\liOmegaOhneMathe 1740, 1745, 1747
\liOOhneMathe 1750, 1755, 1757
\liParagraphMitLinien . 545, 1961, 2683, 2704, 2783, 2791, 2799
\liPetriErreichKnotenDrei 2121
\liPetriErreichTransition 2118

\liPetriSetzeSchluessel 2063
\liPetriTransitionsName 2110, 2122
\liPetriTransitionsNameOhneMathe . . . 2110, 2113, 2115
\liPetriTransPfeile 2122
\liPolynomiellReduzierbar 1402
\liPotenzmenge 1140, 1144, 2133
\liPotenzmengeOhneMathe . . . 1141, 1142, 1143
\liPrimaer 2382
\liProblemBeschreibung 1386
\liProblemClique . . . 1409
\liProblemName 1385, 1392, 1404, 1406, 1419, 1430, 1431, 1439, 1440
\liProblemSat 1438
\liProblemSubsetSum 1429, 1438
\liProblemVertexCover 1409, 1417
\liProduktionen 1179, 1221
liProduktionsRegeln (environment) 1169
liProjektSprache (environment) 1609
\liPseudoUeberschrift 1594, 1644, 1645, 1947, 1957, 2780, 2788, 2796
\liPumpingKontextfrei 2179
\liPumpingRegulaer . 2158
liQuellen (environment) 1669
\liRechtsReduktionInline 2021
\liRekursionsGleichung 1774, 1834
\liRelation 2051
liRelationenSchemaFormat (environment) 2404
\liRelationMenge . . . 2398
\liRichtig 487
liRmodell (environment) 2384
\liRundeKlammer . 1727, 1731, 1741, 1751, 1765
\liSortierMarkierung 2423
\liSortierPfeil 2417
\liSortierPfeilUnten 2420
\liSpaltenUmbruch . . 2492
\liSqlCode 2584
\listen@punkt . . 1669, 1681
\liStrich 1384
\liSyntheseErklaerung 2623, 2688
\liSyntheseUeberErklaerung 2686
\liSyntheseUeberschrift 2607, 2687
\liT 1760, 1775, 1789, 1850
\liTeilen 1984
\literatur 1551, 1575
\liTheta 1730, 1781, 1810, 1816, 1821
\liThetaOhneMathe 1730, 1735, 1737
\liTOhneMathe 1760, 1769, 1771
\liTuringKante 216
\liTuringLeerzeichen 172, 180
\liTuringMaschine . . . 173
\liTuringUeberfuehrung 219
\liTuringUebergaenge 211, 217
\liTuringUebergangZelle 206
\liUeberfuehrungsFunktion 1145
\liUeberfuehrungsFunktionOhneMathe . . . 1145, 1148, 1150
liUebergangsTabelle (environment) 1945
\liUeberschriftDreiecksTabelle 1955
\liUmlLeserichtung . 2747
\liVertauschen 2411
\liWortInSprache 621
\liWortNichtInSprache 626
\liWpEquivalent 2850
\liWpErklaerung 2853
\liWpErklaerungVerzweigung 2868
\liWpKalkuel 2830
\liWpKalkuelOhneMathe 2830, 2835, 2837, 2869, 2871, 2873
\liZustandsBuchstabe 1155, 1164, 1166, 1184, 1186
\liZustandsBuchstabeGross . . . 1156, 1165, 1167
\liZustandsmenge . . . 1143
\liZustandsmengeNr 1157, 2142
\liZustandsmengeNrGross 1165
\liZustandsMengenSammlung 2129
\liZustandsMengenSammlungNr 2138
\liZustandsmengeOhneMathe 1143
\liZustandsname 1166
\liZustandsnameGross . . . 1167, 2130, 2139
\liZustandsnameTiefgestellt 1182
\liZustandsPaar 1939
\liZustandsPaarVariablenName . . . 1938, 1941, 1942
\llap 2371
\log 1778, 1781, 1784, 1810, 1816
\loop 2204
\lor 2872
\ltimes 1886

M

\makeatletter . . 1685, 2443
\makeatother . . 1687, 2451
\marginpar 1100, 1106, 1112, 1592
\mathbb . . 1440, 2191, 2792
\mathbin . 2373, 2374, 2375
\mathcal . . 1751, 2661, 2666, 2668, 2669, 2670
\Mathe 2847
\MatheEnv 2840, 2848, 2851
\mathord 1893, 1894
\mdfsetup 1261, 1618, 1622, 1626, 1630
\medskip 1394, 1603, 1605, 1679, 1707, 2389, 2395, 2712, 2716, 2841, 2845
\memph 1592
\mintinline 2532, 2533, 2572, 2579, 2584
\mkern . . . 2373, 2374, 2375
\mlq 1891, 1893
\mrq 1891, 1894
\msg 52, 416
\myList 2464, 2465, 2466, 2469
\myNodes 2453, 2468, 2474, 2478, 2480

N

\NeedsTeXFormat 1, 14, 32, 65, 233, 422, 483, 490, 554, 617, 632, 1087, 1123, 1236, 1276, 1285, 1290, 1319, 1357, 1378, 1450, 1533, 1549, 1555, 1577, 1723, 1883, 1896, 1975, 2058, 2124, 2149, 2154, 2198, 2364, 2377, 2406, 2487, 2494, 2501, 2506, 2586, 2601, 2692, 2696, 2727, 2769, 2806, 2826
\neg 2873
\negthinspace 1728
\newcounter . . . 2218, 2219
\NewDocumentCommand . 75, 116, 153, 174,

216, 237, 559, 576, 621, 626, 1190, 1212, 1402, 1525, 1553, 1712, 1716, 2040, 2051, 2118, 2541, 2551, 2561, 2573, 2576, 2580, 2747
\NewDocumentEnvironment 1169, 1262, 1317, 1510, 1609, 1610, 1613, 1640, 1648, 1670, 1689, 1699, 1946, 1996, 2386, 2404
\newlength 2853
\node 572, 1518, 1523, 2244, 2249, 2254, 2260, 2266, 2272, 2423, 2468, 2764
\noexpand 2332, 2333, 2334, 2353, 2468
\noindent 368, 623, 628, 1596, 1598, 1602, 1606, 1634, 1662, 1664, 1677, 1693, 1695, 1703, 1872, 1925, 1928, 1931, 1934, 2399, 2705, 2717, 2859
\nolinkurl 401, 2538
\normalsize 1249
\notin 629
\null 2710

O

\o@join 2370, 2373, 2374, 2375
\Omega 1741
\omega 2160, 2161, 2181, 2182
\or 2281

P

\pagestyle 1272
\par . . . 367, 388, 549, 1201, 1209, 1601, 1663, 1686, 1694, 2323, 2345, 2360, 2401, 2687, 2711, 2715, 2723, 2858, 2864
\paragraph 1249
\parindent 2392
\path . . . 107, 154, 217, 587
\pgfkeys . . 2074, 2750, 2751, 2752, 2753, 2754, 2757, 2760, 2762
\pgfmath@count 2445, 2447, 2449
\pgfmath@smuggleone 2450
\pgfmathdeclarefunction 2444
\pgfmathint 2445
\pgfmathparse 1331, 2462, 2467, 2470, 2484, 2485
\pgfmathresult 1332, 2445, 2446, 2448, 2450, 2463, 2471, 2484, 2485
\pgfutil@empty 2446
\pgfutil@loop 2447
\pgfutil@repeat 2450
\preceq 1405
\prime 1384
\printbibliography . 1575
\ProvidesPackage 2, 15, 33, 66, 234, 423, 484, 491, 555, 618, 633, 1088, 1124, 1237, 1277, 1286, 1291, 1320, 1358, 1379, 1451, 1534, 1550, 1556, 1578, 1724, 1884, 1897, 1976, 2059, 2125, 2150, 2155, 2199, 2365, 2378, 2407, 2488, 2495, 2502, 2507, 2587, 2602, 2693, 2697, 2728, 2770, 2807, 2827

Q

\QS@list 2325, 2336, 2340, 2347, 2353, 2358, 2361
\QS@select@equal 2300, 2304
\QS@select@greater 2301, 2305
\QS@select@smaller 2296, 2299, 2303
\QS@sort@a 2278, 2311, 2332, 2333
\QS@sort@b 2278, 2279
\QS@sort@c 2282, 2289
\QS@sort@d 2290, 2298
\QS@sort@empty . 2280, 2285
\QS@sort@single 2281, 2286
\QSinitialize 2203, 2315, 2350
\QSIr . 2286, 2292, 2300, 2320, 2334, 2339, 2342
\QSIrr 2321, 2334, 2335, 2343
\QSLr 2292, 2299, 2310, 2311, 2319, 2332, 2337, 2341
\QSpivotStep 2205, 2315, 2319, 2330
\QSr 2292
\QSRr 2301, 2322, 2333, 2338, 2344, 2353, 2354, 2355
\QSsortStep 2207, 2315, 2331, 2332
\quad 1903

R

\raisebox 1523
\relax 1587, 2292, 2335, 2337, 2338, 2339, 2447, 2449
\renewcommand . . . 1252, 1253, 1254, 1255, 1258, 1259, 1273, 1274, 1544, 1545, 1945
\repeat 2208
\RequirePackage 4, 68, 171, 235, 239, 419, 425, 426, 486, 556, 635, 636, 1090, 1092, 1093, 1129, 1238, 1239, 1242, 1244, 1246, 1251, 1260, 1267, 1278, 1279, 1288, 1292, 1293, 1294, 1322, 1323, 1371, 1383, 1452, 1552, 1557, 1558, 1574, 1581, 1582, 1583, 1611, 1726, 1887, 1888, 2060, 2062, 2152, 2212, 2214, 2366, 2367, 2368, 2381, 2409, 2440, 2491, 2497, 2504, 2509, 2520, 2521, 2589, 2694, 2701, 2731, 2732, 2734, 2735, 2773, 2774, 2775, 2808, 2828
\right 1728
\RIGHTarrow . . . 2749, 2754
\Rightarrow 624, 629
\rightarrow 220, 513, 518, 526, 530, 532, 533, 535, 590, 598, 2122, 2632, 2639, 2641, 2644, 2649, 2654, 2655, 2660
\rightouterjoin 2374
\Roman 1257
\roman 1257, 1259
\romannumeral 2290
\rtimes 1886
\rule 2323, 2345, 2360, 2371

S

\sb 80, 90, 92, 121, 179, 525, 526, 530, 533, 534, 535, 1184, 1186, 1778, 1781, 1784, 1810, 1816, 1965, 2110, 2119, 2654, 2655, 2656, 2661, 2665, 2666, 2669, 2670, 2673, 2674, 2675
\scriptscriptstyle 590, 598, 606
\scriptsize 1203, 1463, 1470, 1476,

1538, 1539, 1542, 1543, 2778, 2831, 2861
\section 59
\seq . 1527, 1528, 1529, 1672, 1673, 1674, 1681
\setbox 2370
\setcounter 1250, 2324, 2346, 2360
\setganttlinklabel . . 1280, 1281, 1282, 1283
\setlength 1546, 2392, 2855, 2856, 2866
\setmainfont 1240
\setmainlanguage 420
\setminted 2526, 2527
\setminus 2025
\setsansfont 1241
\setul 2383
\sffamily 500, 1247, 1249, 1351, 2610
\shoveleft 2005
\shoveright 2009
\Sigma 77, 118, 176, 1153, 1154, 1214
\sigma 523, 525, 526
\SLASH 1593
\small 1702
\sort 2457
\sortList 2456, 2465
\square 488
\stepcounter 2244, 2249, 2254, 2257, 2259, 2263, 2265, 2269, 2271
\str . . 501, 510, 1615, 2134, 2143, 2611, 2624
\string 2007, 2017
\StrSubstitute . 2464, 2466
\strut 1832, 1836, 1840, 1844, 1848, 2492
\subseteq 2635, 2668, 2675

T

\tableofcontents . . . 1588
\text 90, 92, 195, 1987, 2778, 2831
\textbf . . . 1095, 1410, 1419, 1430, 1439, 1597, 1604, 1635, 1663, 1678, 1694, 1950
\textcolor 1522, 2777
\textit 964, 1006, 1007, 1008, 1009, 1704, 1995, 2055
\textsc 1385
\textsf 1597, 1678
\textstyle 1796, 1827
\texttt 1118, 1385, 1519, 1520, 1521, 1522, 2831
\textwidth 1546
\thepage 1271, 1541
\theparagraph 1249
\Theta 1731
\thinspace 2831
\tikz 1523
tikz: bbaum 25
tikz: li binaer baum 23
\tikzchildnode 443
\tikzparentnode 443
\tikzset 109, 156, 222, 428, 454, 1296, 1454, 2100, 2226, 2430, 2591, 2809
\tikzumlset 2737
\times 220
\tiny 1101, 1107, 1113, 1522, 1592, 2536
\titleformat . . 1247, 1249
\titlespacing 1248
\tl . 42, 50, 76, 77, 78, 79, 80, 81, 84, 85, 86, 87, 88, 90, 92, 117, 118, 119, 120, 121, 122, 123, 126, 127, 128, 129, 130, 131, 132, 175, 176, 177, 178, 179, 180, 181, 184, 185, 186, 187, 188, 189, 190, 304, 308, 332, 336, 337, 338, 341, 346, 347, 348, 359, 360, 361, 362, 373, 379, 382, 385, 394, 408, 560, 563, 568, 569, 577, 578, 581, 582, 1192, 1213, 1214, 1215, 1216, 1219, 1220, 1221, 1222, 1761
\tmp 2027
\TmpPlaceEight 2082
\TmpPlaceFive 2079
\TmpPlaceFour 2078
\TmpPlaceNine 2083
\TmpPlaceOne 2075
\TmpPlaceSeven 2081
\TmpPlaceSix 2080
\TmpPlaceTen 2084
\TmpPlaceThree 2077
\TmpPlaceTwo 2076
\TmpScale 2095
\TmpTransitionEight 2071, 2092
\TmpTransitionFive 2068, 2089
\TmpTransitionFour 2067, 2088
\TmpTransitionNine 2072, 2093
\TmpTransitionOne 2064, 2085
\TmpTransitionSeven 2070, 2091
\TmpTransitionSix 2069, 2090
\TmpTransitionTen 2073, 2094
\TmpTransitionThree 2066, 2087
\TmpTransitionTwo 2065, 2086
\TmpX 2096
\TmpY 2097
\today 1539
\ttfamily 2384

U

\ul 1096, 2382, 2383
\umlaggreg 1059
\umlassoc 1041
\umlclass . . . 642, 646, 650, 694, 695, 696, 743, 748, 753, 756, 814, 815, 816, 821, 822, 851, 881, 908, 909, 912, 963, 966, 1005, 1011, 1012, 1030, 1031, 1032, 1051, 1052, 1053, 1054
\umldep 971
\umlHVHaggreg 761, 827, 1021
\umlinherit 700, 751, 916, 961, 969
\umlnote . . . 702, 918, 1061
\umlreal 698, 759
\umlsimpleclass . . 657, 658, 659, 663, 665, 666, 667, 693, 846, 847, 848, 907, 959, 960
\umlstatic 853, 882
\umluniaggreg 914
\umluniassoc 677, 699, 915, 1039, 1040
\umlVHuniassoc . . 678, 679
\umlVHVdep 671, 672, 674, 675, 855, 856
\umlVHVinherit 654, 655, 660, 661, 668, 669, 824, 825, 849, 850, 1019, 1020
\umlVHVreal 818, 819, 1056, 1057
\UParrow 2751
\url 1713
\usemintedstyle 2523
\usetikzlibrary . . 69, 427, 1091, 1295, 1453, 2061, 2369, 2410, 2441, 2736, 2824

V

\value 2206
\varepsilon 502, 513, 514, 1139, 1778, 1784, 1813, 1824
\vfill 2492
\vrule 2706, 2710
\vspace 1649, 1667, 2719, 2721

X

\xappto .. 2468, 2474, 2478
\xdef 1325
\xintApply 2294
\xintApplyUnbraced ..
2293, 2299, 2300, 2301
\xintCSVtoList 2353
\xintFor .....
2243, 2248, 2253,
2258, 2264, 2270, 2311
\xintifEq 2304
\xintifForLast .....
... 2260, 2266, 2272
\xintifGt 2305
\xintifLt 2303
\xintLength 2278
\xintnthelt 2290

Z

\ZB 1720
\zB 1719
\zustandsnamens@liste
... 1157, 1164, 1165