lehramt-informatik

Hermine Bschlangaul *

September 1, 2021

Contents

1 Klassen **3**
1.1 Vorlage Theorie-Teil . . . 4
1.2 Vorlage Aufgabensammlung . . . 5
1.3 Vorlage Aufgabe . . . 6

2 Pakete **7**
2.1 aufgaben-einbinden.sty . . . 8
2.2 aufgaben-metadaten.sty . . . 9
2.3 automaten.sty . . . 10
2.3.1 Endlicher Automat . . . 10
2.3.2 Kellerautomat . . . 12
2.3.3 Turingmaschine . . . 13
2.4 basis.sty . . . 16
2.5 baum.sty . . . 20
2.5.1 Binärbaum . . . 21
2.5.2 AVL-Baum . . . 22
2.5.3 B-Baum . . . 23
2.6 checkbox.sty . . . 24
2.7 chomsky-normalform.sty . . . 25
2.8 cpm.sty . . . 27
2.8.1 Hilfsmakros für Zeitpunkt-Tabelle . . . 28
2.9 cyk-algorithmus.sty . . . 30
2.10 entwurfsmuster.sty . . . 31
2.10.1 Namensschema der Entwurfsmuster-Makros: . . . 31
2.10.2 Reihenfolge . . . 31
2.10.3 Abstrakte Fabrik (Abstract Factory) . . . 31
2.10.4 Adapter . . . 32
2.10.5 Beobachter (Observer) . . . 34
2.10.6 Dekorierer (Decorator) . . . 36
2.10.7 Einfache Fabrik (Simple Factory) . . . 37
2.10.8 Einzelstück (Singleton) . . . 38
2.10.9 Erbauer (Builder) . . . 38
2.10.10 Fabrikmethode (Factory Method) . . . 40
2.10.11 Kompositum (Composite) . . . 41
2.10.12 Modell-Präsentation-Steuerung (Model-View-Controller) . . . 42
2.10.13 Zustand (State) . . . 43
2.11 er.sty . . . 45
2.12 formale-sprachen.sty . . . 47
2.13 formatierung.sty . . . 50
2.13.1 Schriftarten / Typographie . . . 50
2.13.2 Farben . . . 50
2.13.3 Überschriften . . . 50

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

2.13.4 Listen . . . 50
2.13.5 Kasten . . . 50
2.13.6 Header . . . 50
2.14 gantt.sty . . . 51
2.15 grafik.sty . . . 52
2.16 graph.sty . . . 53
2.17 hanoi.sty . . . 55
2.18 klassen-konfiguration-aufgabe.sty . . . 56
2.19 komplexitaetstheorie.sty . . . 57
2.20 kontrollflussgraph.sty . . . 59
2.21 kopf-fusszeilen.sty . . . 61
2.22 literatur-dummy.sty . . . 62
2.23 literatur.sty . . . 63
2.24 makros.sty . . . 64
2.25 master-theorem.sty . . . 68
2.26 mathe.sty . . . 72
2.27 minimierung.sty . . . 73
2.28 normalformen.sty . . . 76
2.29 petri.sty . . . 79
2.30 potenzmengen-konstruktion.sty . . . 81
2.31 pseudo.sty . . . 83
2.32 pumping-lemma.sty . . . 84
2.33 quicksort.sty . . . 85
2.34 relationale-algebra.sty . . . 88
2.35 rmodell.sty . . . 89
2.36 sortieren.sty . . . 90
2.37 spalten.sty . . . 92
2.38 struktogramm.sty . . . 93
2.39 syntax.sty . . . 94
2.40 syntaxbaum.sty . . . 96
2.41 synthese-algorithmus.sty . . . 97
2.42 tabelle.sty . . . 100
2.43 typographie.sty . . . 101
2.44 uml.sty . . . 102
2.45 vollstaendige-induktion.sty . . . 104
2.46 wasserfall.sty . . . 106
2.47 wpkalkuel.sty . . . 107

3 Index **108**

1 Klassen

1.1 Vorlage Theorie-Teil

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

\begin{document}

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

\chapter{Thema des Theorie-Teils}

\literatur

\end{document}
```

1.2 Vorlage Aufgabensammlung

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

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

1.3 Vorlage Aufgabe

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

\end{document}
```

2 Pakete

2.1 aufgaben-einbinden.sty

```
\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}
}
```

2.2 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: {}

  \_gib_github_url:

  \section{\_gib_aufgaben_titel:}

  \bigskip
}
```

```
\ExplSyntaxOff
```

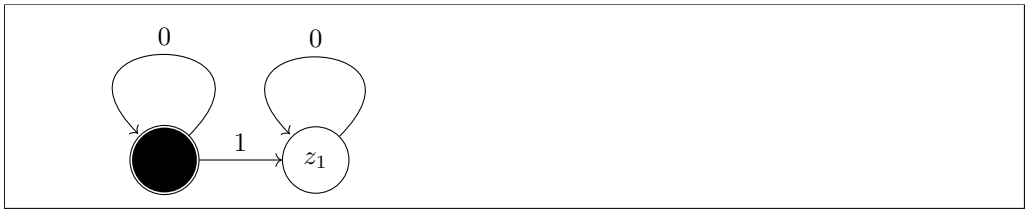
2.3 automaten.sty

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

2.3.1 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}}},
```

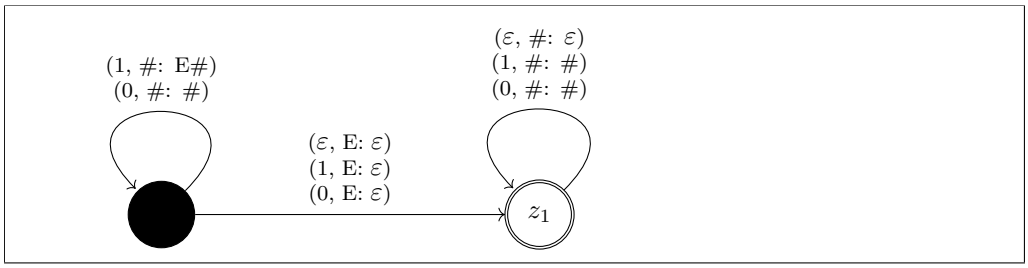
```
    start .code:n = {\tl_set:Nn \l_start_tl {##1}},
    dea .value_forbidden:n = true,
    dea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{DEA}}}},
    nea .value_forbidden:n = true,
    nea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{NEA}}}},
  }

  \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
  },
}
```

2.3.2 Kellerautomat

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

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

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

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



`\liKellerAutomat` `\liKellerAutomat[`⟨*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
      }
    }
  }
}
```

2.3.3 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
      }
    }
  }
}
```

2.4 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
    }{
      \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}
```

2.5 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}
```

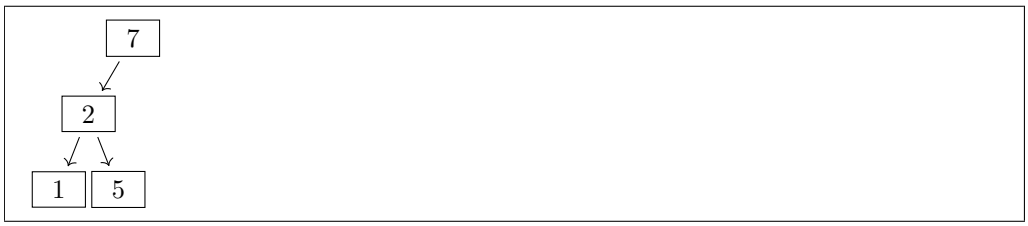
Für b baum

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

2.5.1 Binärbaum

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

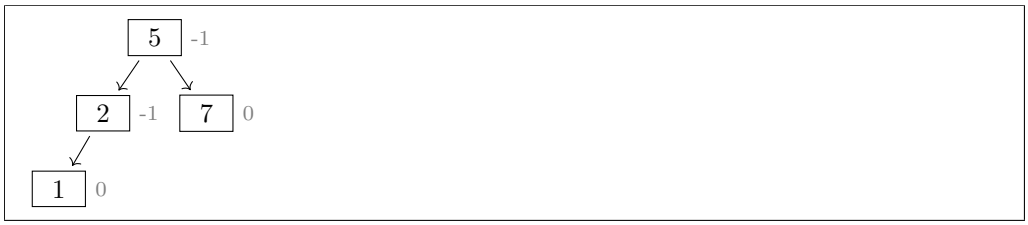
```
\begin{tikzpicture}[li binaer baum]
\Tree
[.7
  [.2
    [.1 ]
    [.5 ]
  ]
  \edge[blank]; \node[blank]{};
]
\end{tikzpicture}
```




```
\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,
    }
  },
}
```

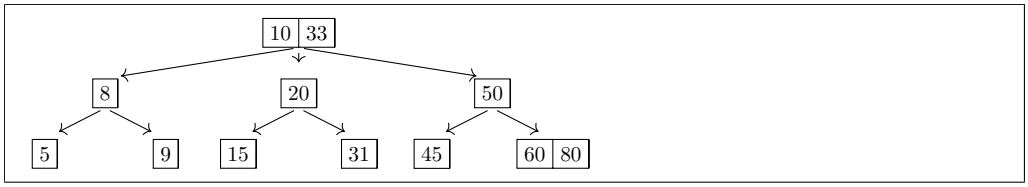
2.5.2 AVL-Baum

```
\begin{tikzpicture}[li binaer baum]
\Tree
[.\node[label=-1]{5};
  [.\node[label=-1]{2};
    [.\node[label=0]{1}; ]
    \edge[blank]; \node[blank]{};
  ]
  [.\node[label=0]{7}; ]
]
\end{tikzpicture}
```



2.5.3 B-Baum

```
\begin{tikzpicture}[
  scale=0.8,
  transform shape,
  li bbaum,
  level 1/.style={level distance=10mm,sibling distance=32mm},
  level 2/.style={level distance=10mm,sibling distance=20mm},
]
\node {10 \nodepart{two} 33}
  child {node {8}
    child {node {5}}
    child {node {9}}
  }
  child {node {20}
    child {node {15}}
    child {node {31}}
  }
  child {node {50}
    child {node {45}}
    child {node {60 \nodepart{two} 80}}
  }
;
\end{tikzpicture}
```




```
\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,
    },
  }
}
```

2.6 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[$\square$]}
```

2.7 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`

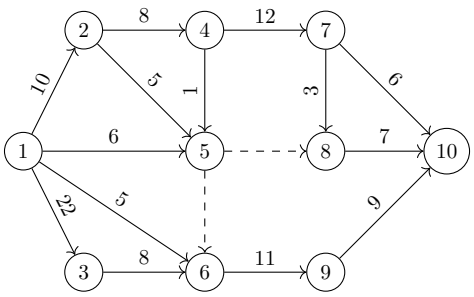
```
\def\liChomskyUeberErklaerung#1{
  \liChomskyUeberschrift{#1}\par
  \liChomskyErklaerung{#1}
}

\ExplSyntaxOff

```

2.8 cpm.sty

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

Let-Abkürzungen

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




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

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

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


\begin{tabular}{|l|l|l|l|l|l|l|l|}
\hline
$i$ & a & b & c  & d  & e  & f  & g  \\\hline\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
```

2.8.1 Hilfsmakros für Zeitpunkt-Tabelle

```
\begin{tabular}{|l||l|l|l|l|l|l|l|l|}
\hline
$i$ & 1 & 2 & 3  & 4 & 5  & 6  & 7  & 8  \\\hline\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): $1_{(2\to3)}$

```
\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): $1_{(\to2)}$

```
\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): $1_{(\leftarrow2)}$

```
\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{$FZ_i$}
```

2.9 cyk-algorithmus.sty

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

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

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

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

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

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

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

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

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

2.10 entwurfsmuster.sty

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

2.10.1 Namensschema der Entwurfsmuster-Makros:

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

2.10.2 Reihenfolge

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

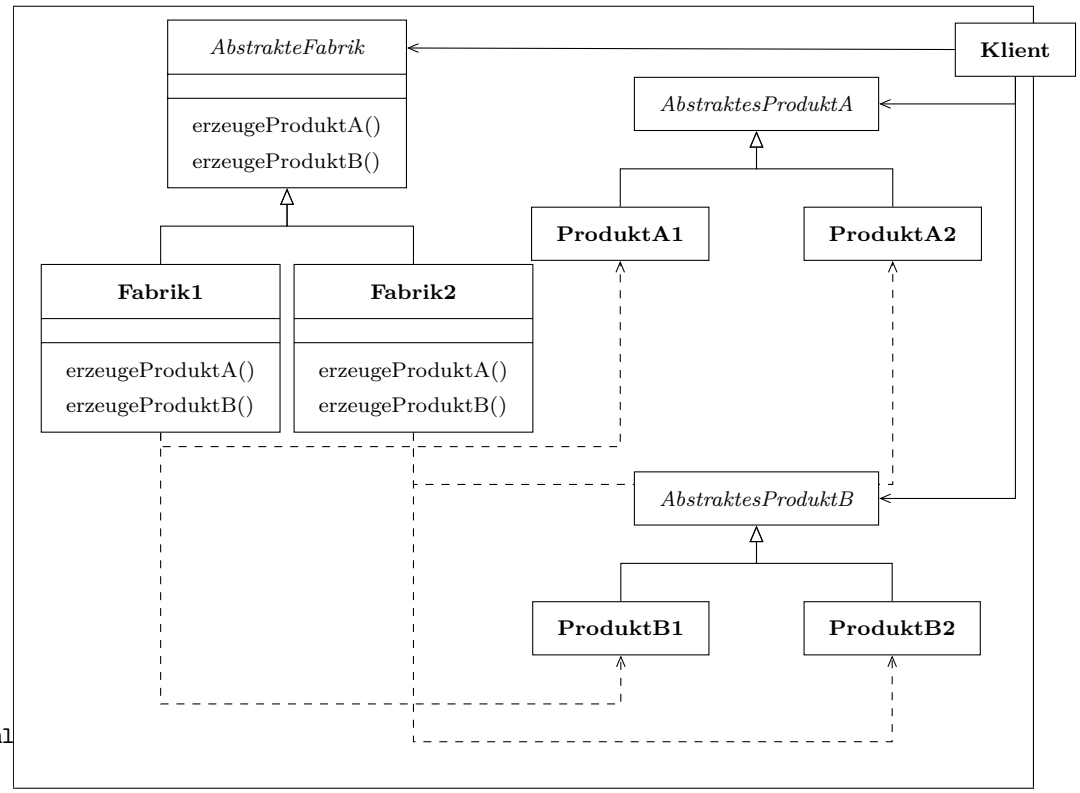
```
\RequirePackage{lehramt-informatik-uml}
```

`\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}
}
```

2.10.3 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

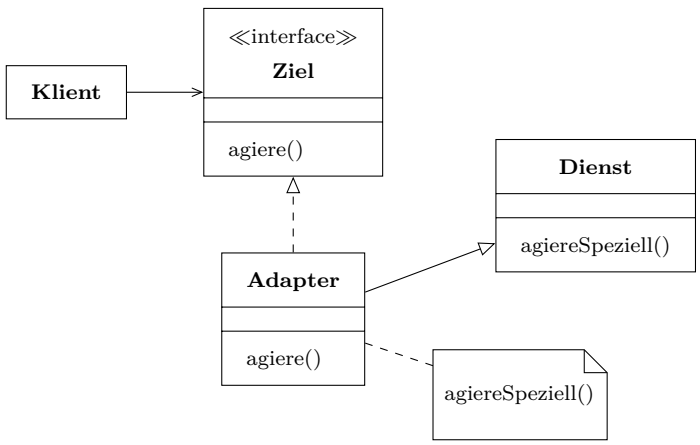
```
\def\liEntwurfsAbstrakteFabrikCode{
  \li@EntwurfsCode{abstrakte_fabrik}{Produkte}
  \li@EntwurfsCode{abstrakte_fabrik}{AbstrakteFabrik}
  \li@EntwurfsCode{abstrakte_fabrik}{Klient}
}
```

\liEntwurfsAbstrakteFabrik

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

2.10.4 Adapter

\liEntwurfsAdapterUml




```
\def\liEntwurfsAdapterUml{
  \begin{tikzpicture}
    \umlsimpleclass[x=1,y=3]{Klient}{}{}
    \umlclass[x=4,y=3,type=interface]{Ziel}{}{agiere()}
    \umlclass[x=4,y=0]{Adapter}{}{agiere()}
    \umlclass[x=8,y=1.5]{Dienst}{}{agiereSpeziell()}

    \umlreal{Adapter}{Ziel}
    \umluniassoc{Klient}{Ziel}
    \umlinherit{Adapter}{Dienst}

    \umlnote[x=7,y=-1,width=2cm]{Adapter}{agiereSpeziell()}
  \end{tikzpicture}
  \footcite[so ähnlich wie GoF]{wiki:adapter}
}
```

\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.

```
\def\liEntwurfsAdapterAkteure{
  \begin{description}

    \item[Ziel (Target)]

    Das Ziel definiert die Schnittstelle, die der Klient nutzen kann.

    \item[Klient (Client)]

    Der Klient nutzt Dienste über inkompatible Schnittstellen und greift
    dabei auf adaptierte Schnittstellen zurück.

    \item[Dienst (Adaptee)]

    Der Dienst bietet wiederzuverwendende Dienstleistungen mit fest
    definierter Schnittstelle an.

    \item[Adapter]

    Der Adapter adaptiert die Schnittstelle des Dienstes auf die
    Schnittstelle zum Klienten.\footcite{wiki:adapter}
```

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

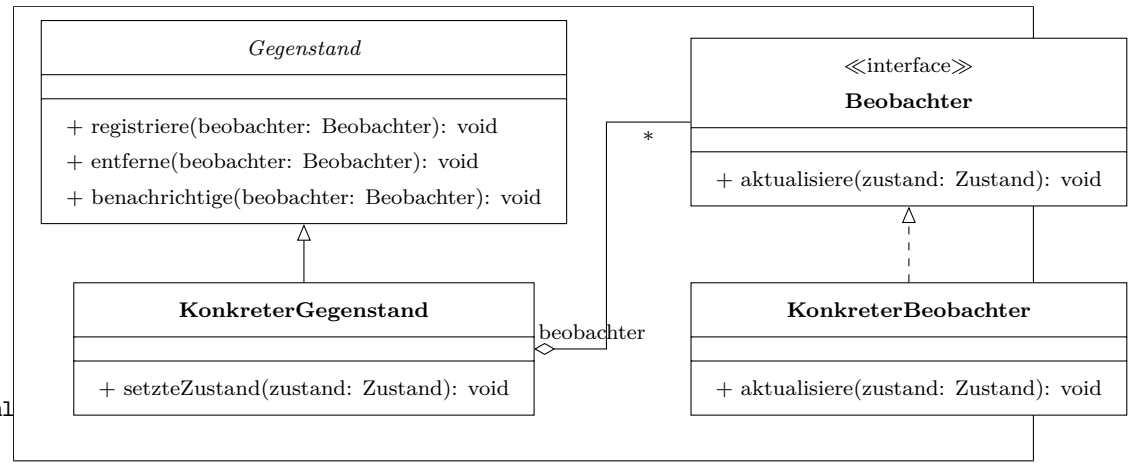
\liEntwurfsAdapterCode

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

\liEntwurfsAdapter

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

2.10.5 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

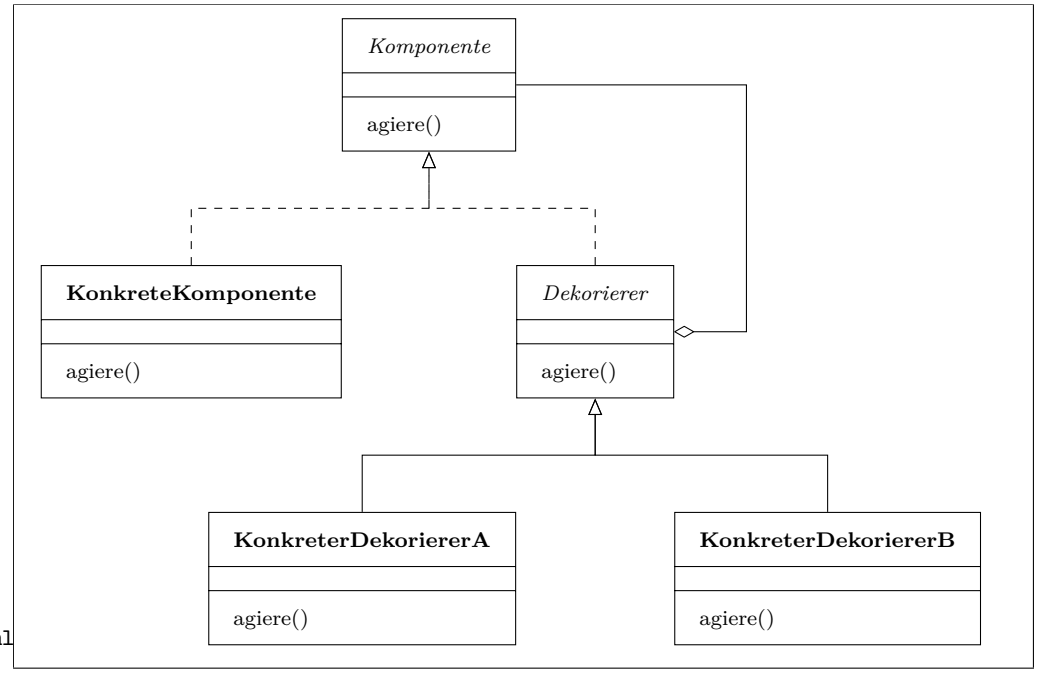
```
\def\liEntwurfsBeobachterCode{
  \li@EntwurfsCode{beobachter}{Gegenstand}
  \li@EntwurfsCode{beobachter}{KonkreterGegenstand}
  \li@EntwurfsCode{beobachter}{Beobachter}
  \li@EntwurfsCode{beobachter}{KonkreterBeobachterA}
  \li@EntwurfsCode{beobachter}{KonkreterBeobachterB}
```

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

\liEntwurfsBeobachter

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

2.10.6 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}
}
```

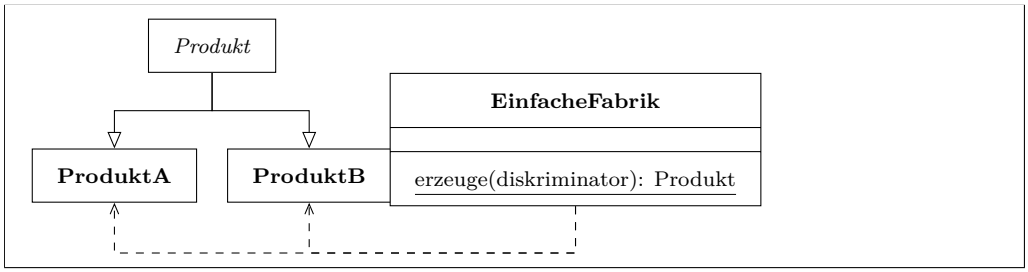
\liEntwurfsDekorierer

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

2.10.7 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}
}
```

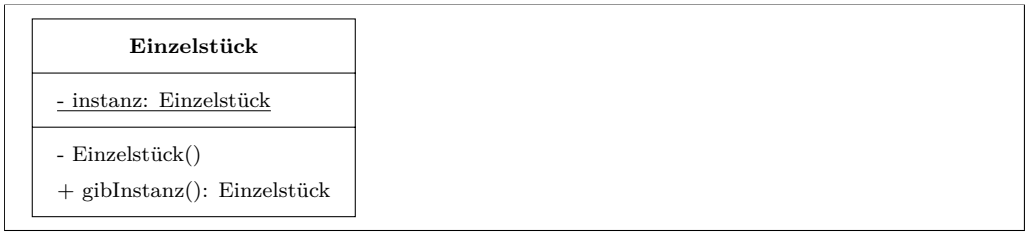
\liEntwurfsEinfacheFabrik

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

2.10.8 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}
}
```

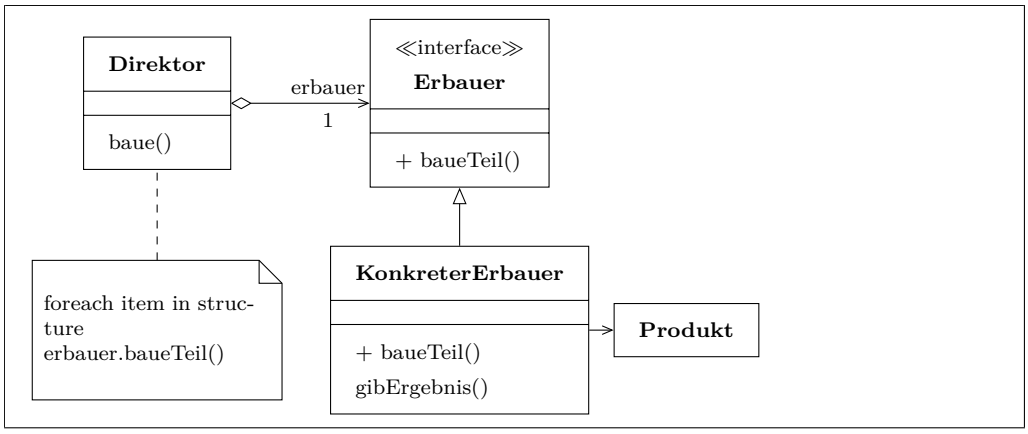
\liEntwurfsEinzelstueck

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

2.10.9 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}
}
```

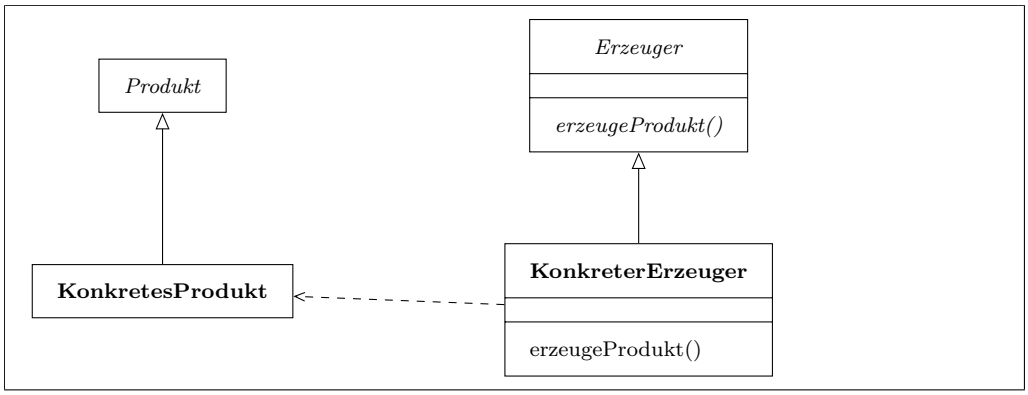
\liEntwurfsErbauer

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

2.10.10 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.

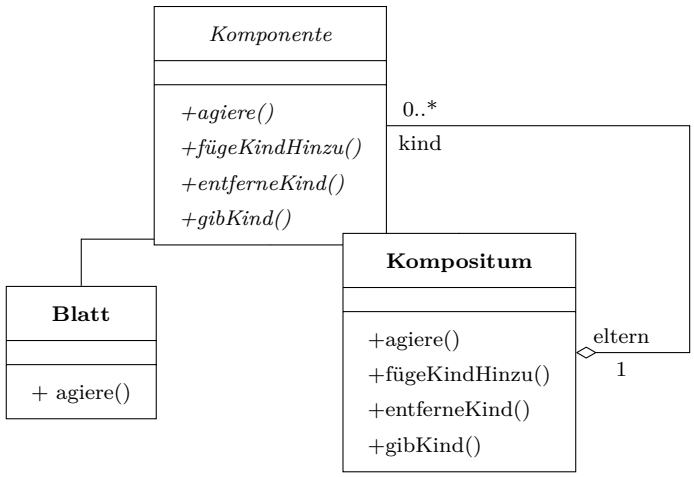
KonkreterErzeuger KonkreterErzeuger überschreibt die Fabrikmethode, um die ihm entsprechenden konkreten Produkte zu erzeugen (z. B. indem er den Konstruktor einer konkreten Produkt-Klasse aufruft).

```
\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
}
```

2.10.11 Kompositum (Composite)



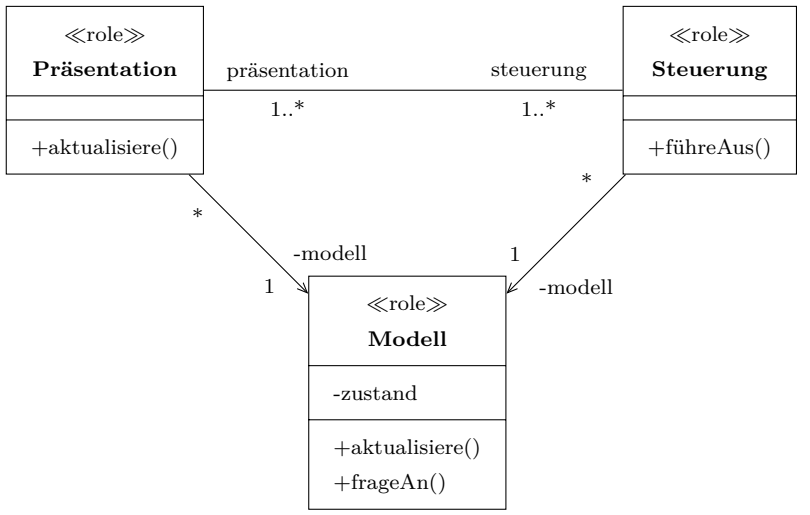
`\liEntwurfsKompositumUml`

```
\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
}
```

2.10.12 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}
}
```

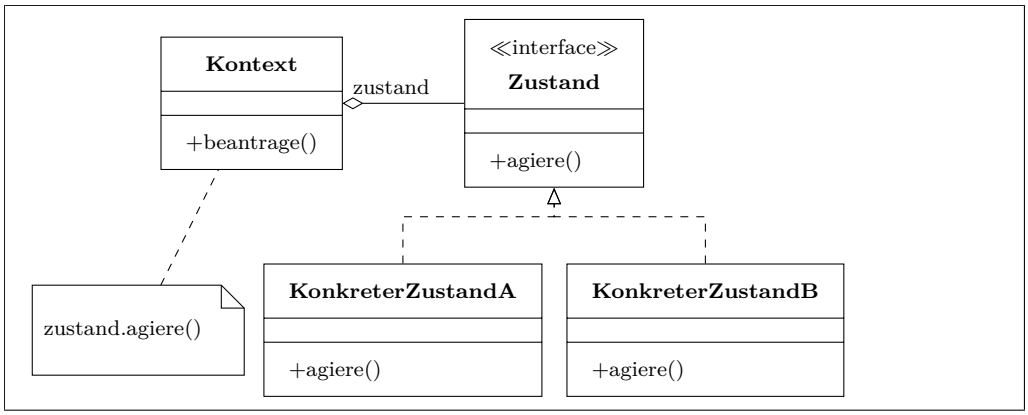
ModellPraesentationSteuerung

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

2.10.13 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
}
```

2.11 er.sty

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

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

```
\begin{tikzpicture}[er2,scale=0.7,transform shape]
% Person
\node[entity] (Person) {Person};
\node[attribute,right=1cm of Person] {\key{E-Mail}} edge (Person);
\node[multi attribute,above left=1cm of Person] {Vornamen} edge (Person);
\node[attribute,left=1cm of Person] {Nachnamen} edge (Person);
\node[attribute,above right=1cm of Person] {Geburtsdatum} edge (Person);

% Kunde
\node[entity,below left=1cm of Person] (Kunde) {Kunde};

% Händler
\node[entity,below right=1cm of Person] (Händler) {Händler};

\node[specialization,below=0.2cm of Person]{is-a}
  edge (Kunde) edge (Händler) edge (Person);

% Transaktion
\node[relationship,below=2cm of Person] (Transaktion) {Transaktion}
  edge node[auto]{1} (Kunde)
  edge node[auto]{1} (Händler);
\node[attribute,below=1cm of Transaktion] {Preis} edge (Transaktion);
\node[attribute,left=1cm of Transaktion,text width=2cm] {Abschlussdatum} edge (Transaktion);
\node[attribute,right=1cm of Transaktion] {Lieferadresse} edge (Transaktion);

% Zahlungsmittel
\node[entity,below=4cm of Händler] (Zahlungsmittel) {Zahlungsmittel}
  edge node[auto]{1} (Transaktion);
\node[attribute,right=1cm of Zahlungsmittel] {Inhaber} edge (Zahlungsmittel);

% Bankverbindung
\node[entity,below left=1cm and 0cm of Zahlungsmittel] (Bankverbindung) {Bankverbindung};
\node[attribute,below left=1cm of Bankverbindung] {\key{IBAN}} edge (Bankverbindung);

% Kreditkarte
\node[entity,below right=3cm and -2cm of Zahlungsmittel]
(Kreditkarte) {Kreditkarte};

\node[attribute,below left=1cm of Kreditkarte]
{\key{Nummer}} edge (Kreditkarte);

\node[attribute,below right=1cm of Kreditkarte,text width=2cm]
{Ablaufdatum} edge (Kreditkarte);

\node[attribute,right=1cm of Kreditkarte]
{Anbieter} edge (Kreditkarte);

\node[generalization,below=0.8cm of Zahlungsmittel]{is-a}
  edge (Zahlungsmittel) edge (Bankverbindung) edge (Kreditkarte);

% Produkt
\node[entity,below=4cm of Kunde] (Produkt) {Produkt}
  edge node[auto]{1} (Transaktion);
\node[attribute,left=1cm of Produkt] {\key{Bezeichnung}} edge (Produkt);
\node[attribute,below left=1cm of Produkt,text width=2cm] {Beschreibungstext} edge (Produkt);
\node[attribute,above left=1cm of Produkt] {Bewertung} edge (Produkt);
```

```
\end{tikzpicture}
```

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

Let-Abkürzungen
`\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
```

2.12 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
```

2.13 formatierung.sty

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

2.13.1 Schriftarten / Typographie

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

2.13.2 Farben

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

2.13.3 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}
```

2.13.4 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})}
```

2.13.5 Kasten

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

liKasten

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

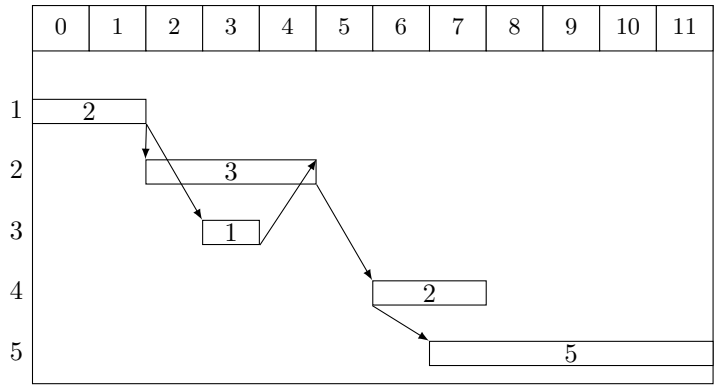
2.13.6 Header

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

2.14 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}{}
```

2.15 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}

```

2.16 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

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

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

```
\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 } {} {}
```

2.17 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}
}
```

2.18 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]
\liLadePakete{mathe}
\ExplSyntaxOn
\AddToHook{enddocument}{
Github \_gib_github_url:
}
\ExplSyntaxOff
```

2.19 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.]
```

Let-Abkürzungen

```
\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}
}
```

2.20 kontrollflussgraph.sty

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

liKontrollflussgraph

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

\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\bWahr=\liBedingungWahr

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

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

2.21 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}
```

```
\ExplSyntaxOff
```

2.22 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 }{}

```

2.23 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]}
```

2.24 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. }
```

2.25 master-theorem.sty

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

Let-Abkürzungen

```
\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}
}

```

2.26 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}{`'}

```

2.27 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
```

2.28 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')
}
```

Let-Abkürzungen

```
\let\ah=\liAttributHuelle
\let\ahL=\liLinksReduktion
\let\ahl=\liLinksReduktionInline
\let\ahr=\liRechtsReduktionInline
\let\fa=\liFunktionaleAbhaengigkeit
\let\FA=\liFunktionaleAbhaengigkeiten
\let\m=\liAttributMenge
\let\r=\liRelation
\let\u=\underline
```

```
\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}`: $\{\,A,\ B\,\} \rightarrow \{\,C,\ D\,\}$ 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;
}
```

$$\mathrm{F} = \Big\{ \begin{array}{l} \{\,M\} \rightarrow \{\,M\}, \\ \{\,M\} \rightarrow \{\,N\}, \\ \{\,V\} \rightarrow \{\,T,\ P,\ PN\}, \\ \{\,P\} \rightarrow \{\,PN\}, \end{array} \Big\}$$

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\,})
}
```

2.29 petri.sty

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

Let-Abkürzungen

```
\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
```

2.30 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

```

2.31 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}
```

2.32 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}
}
```

2.33 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}}
}
```

2.34 relationale-algebra.sty

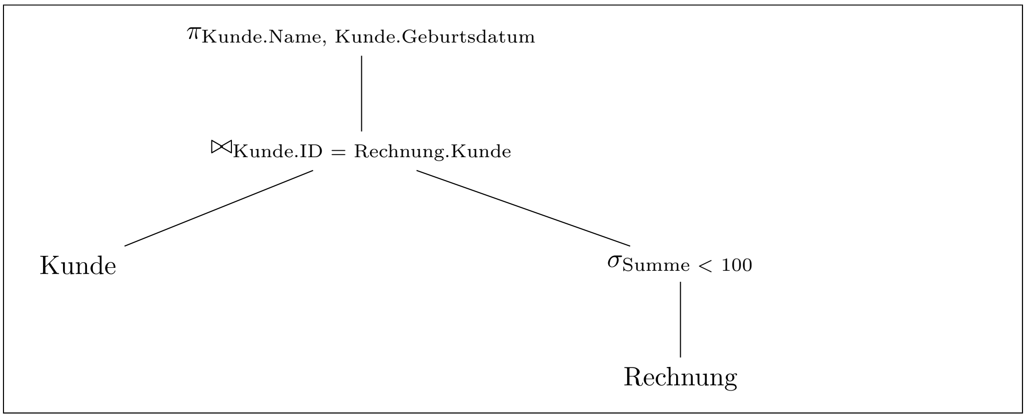
```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{lehramt-informatik-relationale-algebra}[2020/12/11]
```

```
\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}}
```

2.35 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}
```

Let-Abkürzungen

```
\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 } {} {}
```

2.36 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}
```

2.37 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}
```

2.38 struktogramm.sty

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

```

2.39 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.

Let-Abkürzungen
`\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|}

```

2.40 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
    },
  }
}

```

2.41 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
```

Let-Abkürzungen

`\let\schrittE=\liSyntheseUeberErklaerung`

```
\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}
```

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
```

2.42 tabelle.sty

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

```

2.43 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
}
```

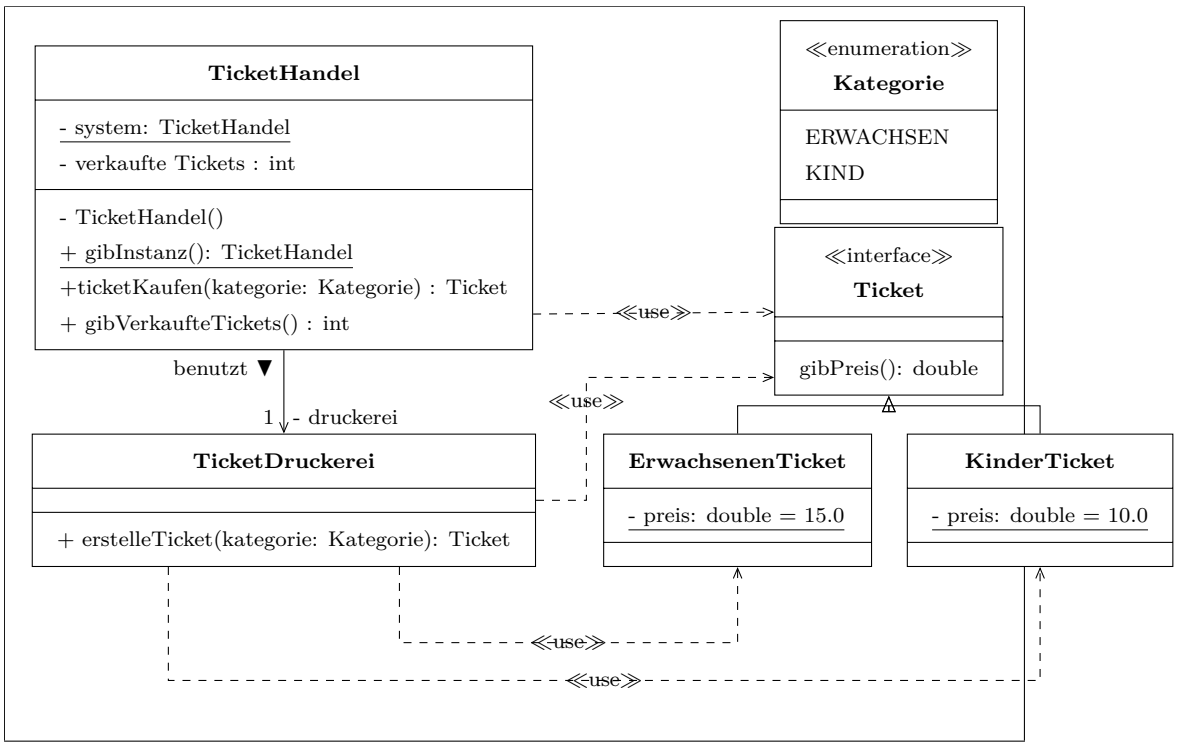
```
\ExplSyntaxOff
```

2.44 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
  };
}
```

2.45 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]
```

Let-Abkürzungen

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

```
\begin{align*}
C_{n+1}
& = \frac
    {(4 \cdot (\m{n + 1} - 1) + 2) \cdot \text{cn}(\m{n + 1} - 1)}
    {\m{n + 1} + 1}
& \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
```

2.46 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}

```

2.47 wpkalkuel.sty

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

Let-Abkürzungen

```
\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

\# 111
\, ... 345, 400, 1183, 1185, 1696, 1697, 1698, 2032, 2502, 2688
\@Skip@Erklaerung@Reset . . . 2823, 2825, 2836
\@afterheading 1663
\@afterindentfalse . 1663
\@liDirLeft 2718, 2723, 2735
\@liDirRight 2719, 2721, 2722, 2723, 2724, 2735
\@liDistance 2729, 2730, 2734
\@liPos . . 2726, 2727, 2734
\\ 609, 632, 633, 636, 637, 640, 641, 733, 734, 735, 842, 871, 873, 899, 908, 953, 995, 996, 997, 1002, 1003, 1004, 1024, 1570, 1927, 1985, 1988
\{ 209, 1120, 1130, 1142, 1143, 1148, 1182, 1411, 1972, 2377, 2688, 2839
\} 209, 1120, 1130, 1142, 1143, 1150, 1186, 1412, 1972, 2377, 2692, 2839
_ . . 27, 35, 44, 46, 48, 295, 320, 344, 345, 359, 360, 366, 369, 372, 382, 394, 1352
\␣ 1984, 1994

A

\addbibresource 1540, 1541, 1542, 1543, 1544, 1545, 1546, 1547, 1548, 1549
\AddToHook 1351
\advance 2426
\AfterEndEnvironment 2495
\allsectionsfont . . . 1232
\Alph 1246
\alph 1246, 1247
\alpha 2602, 2604, 2605, 2608, 2610, 2611, 2612, 2613, 2614, 2618, 2624, 2625, 2630, 2631, 2632, 2635, 2643, 2644, 2645
\arabic 1246, 2221, 2226, 2231, 2237, 2243, 2249
\arraystretch 1922

B

\BeforeBeginEnvironment 2494
\begin . . 630, 681, 696, 731, 755, 802, 834, 849, 869, 879, 895, 915, 947, 964, 993, 1018, 1039, 1054, 1163, 1252, 1327, 1365, 1374, 1381, 1489, 1562, 1614, 1619, 1627, 1652, 1657, 1668, 1678, 1682, 1765, 1769, 1785, 1808, 1831, 1846, 1925, 1926, 1976, 2021, 2141, 2161, 2302, 2324, 2338, 2494, 2505, 2692
\begingroup 1563, 1974, 2422
\beschriftung 1594, 1598, 1602, 1606, 1610, 1612
\beta 2602, 2605, 2609, 2610, 2611, 2614, 2624, 2625, 2626, 2631, 2632
\bf 2199, 2200, 2201
\bfseries . . 488, 1236, 1238, 2199, 2205, 2207, 2209, 2210, 2579
\Bigl 2688
\Bigr 2692
\bigskip 50, 378, 611, 616, 1572, 1900
\bool 323, 346
\bowtie 2347, 2350, 2351, 2352
\Box 161
\boxtimes 476

C

\c 1329, 1330
\cdot 1740, 1793, 1804
\centerline 1370, 2300, 2322, 2337
\chapter 1236, 1237
\char 1570
\clearpage 1564
\cline 609
\clist 228, 272, 273, 292, 296, 2434
\columnbreak 2469
\cs 295, 320, 344, 345, 382, 394, 1646
\csname 1314, 1317
\cup 1143, 2006, 2611, 2625, 2632

D

\DeclareMathSymbol 1870, 1871
\DecoINERT 2224, 2297, 2298, 2320
\DecoINERTwithPivot 2240, 2319
\DecoLEFT 2219, 2318
\DecoLEFTwithPivot 2234, 2296
\DecoRIGHT 2229, 2321, 2331
\DecoRIGHTwithPivot 2246, 2299

\definecolor 1234
\delta 67, 109, 167, 209, 1134
\dh 1698, 2613
\directlua .....
60, 139, 197, 202, 1115, 1129, 1149, 1157, 1164, 1169, 1957, 1962, 2015, 2022, 2029, 2389, 2481, 2513, 2518, 2519, 2523, 2528, 2529, 2533, 2539, 2540
\do 2220, 2225, 2230, 2235, 2241, 2247
\dots 519, 523, 1411, 2148, 2624, 2625
\DOWNarrow 2722
\draw 1334, 1337, 1340, 2096, 2395, 2398

E

\edef 1325, 2313, 2317, 2329, 2330
\else 583, 591, 599, 1124, 1138, 1174, 1610, 1713, 1723, 1733, 1747, 1968, 2005, 2091, 2259, 2452, 2454, 2806
\emph 1086, 1392, 1421, 1423, 1569
\empty 1610, 2004
\emptyset .....
1914, 2619, 2640, 2673
\end 669, 692, 717, 752, 786, 818, 846, 862, 876, 884, 911, 940, 961, 986, 1011, 1031, 1051, 1069, 1165, 1254, 1343, 1377, 1378, 1385, 1494, 1567, 1616, 1624, 1643, 1659, 1660, 1674, 1685, 1686, 1767, 1780, 1805, 1828, 1843, 1853, 1929, 1930, 1978, 2023, 2151, 2171, 2302, 2324, 2338, 2495, 2509, 2692
\endcsname 1314, 1317
\endgroup 1566, 1979, 2427
\enspace 2677, 2679
environments:
liAdditum 1617
liAHuelle 1973
liAntwort 1588
liDiagramm 1676
liEinbettung 1587
liExkurs 1625
liGraphenFormat . 1306
liKasten 1251
liKontrollflussgraph
..... 1488
liLernkartei 1666
liProduktionsRegeln
..... 1158
liProjektSprache 1586
liQuellen 1646
liRelationenSchemaFormat
..... 2381
liRmodell 2361
liUebergangsTabelle
..... 1922
\equiv 2821, 2840
\erzeuge@tiefgestellt
... 1129, 1130, 1134
\expandafter .....
1314, 2255, 2257, 2258, 2259, 2267, 2425
\ExplSyntaxOff .....
. 52, 94, 136, 141, 194, 199, 204, 541, 563, 578, 1110, 1188, 1223, 1354, 1509, 1524, 1688, 1763, 1950, 2124, 2374, 2435, 2553, 2660, 2695, 2774, 2845
\ExplSyntaxOn 22, 63, 104, 137, 162, 195, 200, 225, 483, 547, 564, 1083, 1178, 1200, 1350, 1502, 1514, 1589, 1702, 1932, 2105, 2362, 2432, 2480, 2576, 2670, 2746, 2799

F

\faCheckSquareO 2672
\faCircleThin 1102
\faGg 1096
\fancyfoot .....
1258, 1259, 1260, 1518, 1519, 1520, 1521
\fancyhead .....
1257, 1515, 1516, 1517
\faSquareO 1090
\fi 585, 593, 601, 1126, 1140, 1176, 1613, 1715, 1725, 1735, 1749, 1970, 2007, 2093, 2260, 2452, 2453, 2456, 2458, 2808
\fontspec 1232
\footcite .....
693, 715, 762, 785, 817, 912, 939, 985, 1390, 1393, 1400, 1405, 1410, 1414, 1420, 1425, 1529, 1781, 1782, 1937, 2154
\footnote 1690, 1694
\footnotesize .....
149, 355, 437, 533, 1107, 1585, 1638, 1653, 1669, 1881, 1975, 1992, 2000, 2109, 2118, 2361, 2500, 2652, 2709, 2735
\footrulewidth . 1263, 1523
\foreach . 1329, 1332, 1339
\forestFirst .. 2449, 2452
\forestLast ... 2450, 2452
\forestOget ... 2449, 2450
\forestOnes 2462
\forestOv 2451, 2452, 2455
\forestov . 2441, 2445, 2446, 2449, 2450, 2451, 2452, 2454, 2455
\forestset 2437, 2460
\forestSortLevel ....
2439, 2447, 2461, 2462
\frac 1742, 1773, 1804, 1819
\fullouterjoin 2352

G

\g 29, 37, 272, 273, 292, 296, 302, 303, 304, 305, 306, 308, 309, 311, 313, 314, 315, 316, 317, 318, 321, 325, 326, 327, 330, 332, 333, 334, 335, 336, 337, 348, 349, 350, 351, 359, 360, 362, 368, 369, 371, 372, 374, 375, 383, 388, 390, 395, 397, 401
\Gamma . 108, 166, 209, 1143
\gappto 2457
\geq 1418, 2137, 2142, 2158, 2162

H

\hbox 2347
\headrulewidth . 1262, 1522
\hfill 1789, 1795, 1800, 2680
\hinweis 1585
\hline 1927
\href 384, 1196, 1694, 1857, 2508
\hspace .. 2099, 2813, 2821
\ht 2348

I

\i 1339, 1340
\ifcase 2256
\ifmmode 581, 589, 597, 1122, 1136, 1172, 1711, 1721, 1731, 1745, 1966, 2089, 2804
\IfNoValueTF .....
... 1620, 1690, 1694
\ifnum 2183, 2424, 2440, 2448, 2454
\ifx 1610, 2004, 2452
\in 507, 613, 1418, 1755, 1758, 1761, 1787, 1793, 1798, 2137, 2148,

2158, 2168, 2602, 2604, 2610, 2631, 2762
\inhaltsverzeichnis 1561
\input . 4, 7, 10, 13, 16, 407
\inputminted 2512, 2522, 2532, 2544, 2547, 2551
\int 2434
\item 476, 477, 698, 702, 707, 712, 756, 765, 770, 778, 850, 855, 859, 880, 916, 921, 928, 936, 965, 970, 974, 979, 1055, 1060, 1065, 1375, 1376, 1646, 1650, 1770, 1773, 1777, 1786, 1792, 1797, 1809, 1813, 1817, 1821, 1825, 1832, 1836, 1840, 2142, 2145, 2148, 2162, 2165, 2168
\itshape 532, 2651

J

\j 1329, 1330, 1332, 1333, 1334, 1339, 1340, 1341

K

\k 1339
\keys 31, 72, 84, 114, 124, 172, 182, 300, 551, 555, 569, 574, 1207, 1214

L

\l 65, 66, 67, 68, 69, 70, 73, 74, 75, 76, 77, 79, 81, 86, 87, 88, 89, 90, 91, 106, 107, 108, 109, 110, 111, 112, 115, 116, 117, 118, 119, 120, 121, 127, 128, 129, 130, 131, 132, 133, 164, 165, 166, 167, 168, 169, 170, 173, 174, 175, 176, 177, 178, 179, 185, 186, 187, 188, 189, 190, 191, 549, 552, 557, 558, 561, 566, 567, 570, 571, 576, 1202, 1203, 1204, 1205, 1208, 1209, 1210, 1211, 1217, 1218, 1219, 1220, 1505, 1506, 1507, 1649, 1650, 1651, 1658
\labelenumi 1247
\labelenumii 1248
\labelitemi 1241
\labelitemii 1242
\labelitemiii 1243
\labelitemiv 1244
\land 2841, 2843
\LARGE 1236
\large 1370, 2411
\leaders 2680
\left 1705
\LEFTarrow 2723
\leftarrow 595
\leftouterjoin 2350
\leftskip 2825, 2826, 2836
\LehramtInformatikAutorEmail 1521
\LehramtInformatikAutorName 1520
\LehramtInformatikGitBranch 387, 2488
\LehramtInformatikGithubCodeRepo 2487
\LehramtInformatikGithubDomain 2484
\LehramtInformatikGithubRawDomain 385, 2485
\LehramtInformatikGithubTexRepo 386, 2486
\LehramtInformatikRepository 4, 7, 10, 13, 16, 1540, 1541, 1542, 1543, 1544, 1545, 1546, 1547, 1548, 1549, 2483
\LehramtInformatikTitel 1516
\leq 1804, 2145, 2165
\let 1132, 1133, 1564, 2296, 2297, 2298, 2299, 2312, 2314, 2315, 2316, 2318, 2319, 2320, 2321, 2331, 2423, 2461, 2462, 2672
\li@chomsky@erklaerung@texte 498, 534
\li@EntwurfsCode 626, 672, 673, 674, 720, 721, 722, 723, 789, 790, 791, 792, 793, 794, 821, 822, 823, 824, 825, 826, 887
\li@EntwurfsCodeAllgemein 625
\li@fussnote@text 1878, 1884, 1888, 1892, 1896
\li@GithubLink 2504, 2517, 2527, 2538
\li@mget . 1316, 1320, 1340
\li@minc 1319, 1341
\li@mset 1313, 1321, 1330, 1333
\li@numdiscs 1325, 1334, 1340
\li@Rmodell@Schrift 2361, 2370, 2380
\li@sequence . . 1326, 1339
\li@synthese@erklaerung@texte 2593, 2653
\liAbleitung 1157
liAdditum (environment) 1617
liAHuelle (environment) 1973
\liAlphabet 1142
liAntwort (environment) 1588
\liAnweisung 1496
\liAssemblerCode . . . 2542
\liAssemblerDatei . . 2543
\liAttribut 2380
\liAttributHuelle 1964, 2605, 2610
\liAttributHuelleOhneMathe 1964, 1967, 1969, 1983, 1993, 2001
\liAttributMenge 1972, 1984, 1987, 1994, 1995, 2009, 2011
\liAufgabe 3
\liAufgabenMetadaten . 25
\liAufgabenTitel 23
\liAusdruck 1179
\liAutomat 63
\liAutomatenKante 95
\liBandAlphabet 1143
\liBedingung 1497
\liBedingungDrei 1760, 1800, 1840
\liBedingungEins 1754, 1789, 1832
\liBedingungFalsch . 1499
\liBedingungWahr . . . 1498
\liBedingungZwei 1757, 1795, 1836
\liBeschriftung 1577
\liChomskyErklaerung 498, 539
\liChomskyUeberErklaerung 537
\liChomskyUeberschrift 486, 538
\liCpmEreignis 547
\liCpmFruehesterI . . . 604
\liCpmSpaetesterI . . . 603
\liCpmVon 587
\liCpmVonOhneMathe 587, 590, 592
\liCpmVonZu 579
\liCpmVonZuOhneMathe 579, 582, 584
\liCpmVorgang 564
\liCpmZu 595
\liCpmZuOhneMathe 595, 598, 600
liDiagramm (environment) 1676
liEinbettung (environment) 1587
\liEntwurfs 1034

\liEntwurfsAbstrakteFabrik 676
\liEntwurfsAbstrakteFabrikCode 671, 678
\liEntwurfsAbstrakteFabrikUml 629, 677
\liEntwurfsAdapter . . 725
\liEntwurfsAdapterAkteure 695, 727
\liEntwurfsAdapterCode 719, 728
\liEntwurfsAdapterUml 680, 726
\liEntwurfsBeobachter 796
\liEntwurfsBeobachterAkteure 754, 798
\liEntwurfsBeobachterCode 788, 799
\liEntwurfsBeobachterUml 730, 797
\liEntwurfsDekorierer 828
\liEntwurfsDekoriererAkteure 830
\liEntwurfsDekoriererCode 820, 831
\liEntwurfsDekoriererUml 801, 829
\liEntwurfsEinfacheFabrik 864
\liEntwurfsEinfacheFabrikAkteure 848, 866
\liEntwurfsEinfacheFabrikUml 833, 865
\liEntwurfsEinzelstueck 889
\liEntwurfsEinzelstueckAkteure 878, 891
\liEntwurfsEinzelstueckCode 886, 892
\liEntwurfsEinzelstueckUml 868, 890
\liEntwurfsErbauer . . 942
\liEntwurfsErbauerAkteure 914, 944
\liEntwurfsErbauerUml 894, 943
\liEntwurfsFabrikmethode 988, 1013
\liEntwurfsFabrikmethodeAkteure 963, 990
\liEntwurfsFabrikmethodeUml 946, 989
\liEntwurfsKompositum 1013
\liEntwurfsKompositumAkteure 1015
\liEntwurfsKompositumUml 992, 1014
\liEntwurfsModellPraesentation 1034
\liEntwurfsModellPraesentationAkteure 1036
\liEntwurfsModellPraesentationUml 1017, 1035
\liEntwurfsZustand . 1071
\liEntwurfsZustandAkteure 1053, 1073
\liEntwurfsZustandUml 1038, 1072
\liEpsilon 1128
\liErAttribute 1086, 1100, 1102
\liErDatenbankName . 1105
\liErEntity 1084, 1088, 1090
\liErledigt 2672
\liErMpAttribute . . . 1099
\liErMpEntity 1087
\liErMpRelationship 1093
\liErRelationship 1085, 1094, 1096
\liExamensAufgabe 6
\liExamensAufgabeA . . . 15
\liExamensAufgabeTA . . 12
\liExamensAufgabeTTA . . 9
liExkurs (environment) 1625
\liFalsch 477
\liFlaci 1189
\liFremd 2360
\liFunktionaleAbhaengigkeit . . . 2003, 2006, 2014
\liFunktionaleAbhaengigkeiten 2017
\liFussnote . . . 1877, 1879
\liFussnoteDreiText 1891, 1909
\liFussnoteEinsText 1883, 1903
\liFussnoteLink 1693
\liFussnoten 1899
\liFussnoteUrl . 1032, 1689
\liFussnoteVierText 1895, 1912
\liFussnoteZweiText 1887, 1906
\liGeschweifteKlammern . . . 1161, 2018, 2684
\liGrammatik 1200
liGraphenFormat (environment) 1306
\liHanoi 1313
\liHaskellCode 2549
\liHaskellDatei 2550
\liInduktionAnfang . 2749
\liInduktionErklaerung 2748
\liInduktionMarkierung 2747
\liInduktionSchritt 2765
\liInduktionVoraussetzung 2757
\liJavaCode 2502
\liJavaDatei . . 627, 2511
\liJavaExamen 2531
\liJavaTestAkteure . . 2521
liKasten (environment) 1251
\liKellerAutomat 104
\liKellerKante 142
\liKellerUebergang 137, 143
\liKontrollCode 1500
liKontrollflussgraph (environment) 1488
\liKontrollKnotenPfad 1502
\liKontrollTextzeileKnoten 1501, 1506
\liKurzeTabellenLinie 609
\liLadeAllePakete . . . 230
\liLadePakete . . 56, 59, 226, 231, 485, 546, 1119, 1349, 1360, 1876, 1956, 2104, 2575
\liLatexCode 2503
\liLeereZelle 1914
liLernkartei (environment) 1666
\liLinksReduktion . . 1981
\liLinksReduktionInline 1990, 1998
\liMasterExkurs 1845
\liMasterFaelle 1784, 1852
\liMasterFallRechnung 1830
\liMasterVariablen 1764, 1847
\liMasterVariablenDeklaration 1807
\liMasterWolframLink 1855
\liMenge . . . 73, 74, 76, 115, 116, 117, 121, 173, 174, 175, 179, 1120, 1169, 1208, 1209
\liMengeOhneMathe 1120, 1123, 1125
\liMinimierungErklaerung 1936
\liMinispracheDatei 2546
\linespread 2368
\liNichtsZuTun 2673
\liO 1727, 1755
\liOmega 1717, 1761
\liOmegaOhneMathe 1717, 1722, 1724
\liOOhneMathe 1727, 1732, 1734
\liParagraphMitLinien . 534, 1938, 2653, 2674, 2753, 2761, 2769
\liPetriErreichKnotenDrei 2098
\liPetriErreichTransition 2095
\liPetriSetzeSchluessel 2040
\liPetriTransitionsName 2087, 2099
\liPetriTransitionsNameOhneMathe . . . 2087, 2090, 2092

\liPetriTransPfeile 2099
\liPolynomiellReduzierbar 1380
\liPotenzmenge 1129, 1133, 2110
\liPotenzmengeOhneMathe . . . 1130, 1131, 1132
\liPrimaer 2359
\liProblemBeschreibung 1364
\liProblemClique . . . 1387
\liProblemName 1363, 1370, 1382, 1384, 1397, 1408, 1409, 1417, 1418
\liProblemSat 1416
\liProblemSubsetSum 1407, 1416
\liProblemVertexCover 1387, 1395
\liProduktionen 1168, 1210
liProduktionsRegeln (environment) 1158
liProjektSprache (environment) 1586
\liPseudoUeberschrift 1571, 1621, 1622, 1924, 1934, 2750, 2758, 2766
\liPumpingKontextfrei 2156
\liPumpingRegulaer . 2135
liQuellen (environment) 1646
\liRechtsReduktionInline 1998
\liRekursionsGleichung 1751, 1811
\liRelation 2028
liRelationenSchemaFormat (environment) 2381
\liRelationMenge . . . 2375
\liRichtig 476
liRmodell (environment) 2361
\liRundeKlammer . 1704, 1708, 1718, 1728, 1742
\liSortierMarkierung 2400
\liSortierPfeil 2394
\liSortierPfeilUnten 2397
\liSpaltenUmbruch . . 2469
\liSqlCode 2554
\listen@punkt . . 1646, 1658
\liStrich 1362
\liSyntheseErklaerung 2593, 2658
\liSyntheseUeberErklaerung 2656
\liSyntheseUeberschrift 2577, 2657
\liT 1737, 1752, 1766, 1827
\liTeilen 1961
\literatur 1528, 1552
\liTheta 1707, 1758, 1787, 1793, 1798
\liThetaOhneMathe 1707, 1712, 1714
\liTOhneMathe 1737, 1746, 1748
\liTuringKante 205
\liTuringLeerzeichen 161, 169
\liTuringMaschine . . . 162
\liTuringUeberfuehrung 208
\liTuringUebergaenge 200, 206
\liTuringUebergangZelle 195
\liUeberfuehrungsFunktion 1134
\liUeberfuehrungsFunktionOhneMathe . . . 1134, 1137, 1139
liUebergangsTabelle (environment) 1922
\liUeberschriftDreiecksTabelle 1932
\liUmlLeserichtung . 2717
\liVertauschen 2388
\liWortInSprache 610
\liWortNichtInSprache 615
\liWpEquivalent 2820
\liWpErklaerung 2823
\liWpErklaerungVerzweigung 2838
\liWpKalkuel 2800
\liWpKalkuelOhneMathe 2800, 2805, 2807, 2839, 2841, 2843
\liZustandsBuchstabe 1144, 1153, 1155, 1173, 1175
\liZustandsBuchstabeGross . . . 1145, 1154, 1156
\liZustandsmenge . . . 1132
\liZustandsmengeNr 1146, 2119
\liZustandsmengeNrGross 1154
\liZustandsMengenSammlung 2106
\liZustandsMengenSammlungNr 2115
\liZustandsmengeOhneMathe 1132
\liZustandsname 1155
\liZustandsnameGross . . . 1156, 2107, 2116
\liZustandsnameTiefgestellt 1171
\liZustandsPaar 1916
\liZustandsPaarVariablenName . . . 1915, 1918, 1919
\llap 2348
\log 1755, 1758, 1761, 1787, 1793
\loop 2181
\lor 2842
\ltimes 1863

M

\makeatletter . . 1662, 2420
\makeatother . . 1664, 2428
\marginpar 1089, 1095, 1101, 1569
\mathbb . . 1418, 2168, 2762
\mathbin . 2350, 2351, 2352
\mathcal . . 1728, 2631, 2636, 2638, 2639, 2640
\Mathe 2817
\MatheEnv 2810, 2818, 2821
\mathord 1870, 1871
\mdfsetup 1250, 1595, 1599, 1603, 1607
\medskip 1372, 1580, 1582, 1656, 1684, 2366, 2372, 2682, 2686, 2811, 2815
\memph 1569
\mintinline 2502, 2503, 2542, 2549, 2554
\mkern . . . 2350, 2351, 2352
\mlq 1868, 1870
\mrq 1868, 1871
\msg 39, 405
\myList 2441, 2442, 2443, 2446
\myNodes 2430, 2445, 2451, 2455, 2457

N

\NeedsTeXFormat . 1, 19, 54, 222, 411, 472, 479, 543, 606, 621, 1076, 1112, 1225, 1265, 1274, 1279, 1308, 1346, 1356, 1428, 1511, 1526, 1532, 1554, 1700, 1860, 1873, 1952, 2035, 2101, 2126, 2131, 2175, 2341, 2354, 2383, 2464, 2471, 2476, 2556, 2571, 2662, 2666, 2697, 2739, 2776, 2796
\neg 2843
\negthinspace 1705
\newcounter . . . 2195, 2196
\NewDocumentCommand . 64, 105, 142, 163, 205, 226, 548, 565, 610, 615, 1179, 1201, 1380, 1503, 1530, 1689, 1693, 2017, 2028, 2095, 2511, 2521, 2531, 2543, 2546, 2550, 2717
\NewDocumentEnvironment 1158, 1251, 1306,

1488, 1586, 1587, 1590, 1617, 1625, 1647, 1666, 1676, 1923, 1973, 2363, 2381
\newlength 2823
\node 561, 1496, 1501, 2221, 2226, 2231, 2237, 2243, 2249, 2400, 2445, 2734
\noexpand 2309, 2310, 2311, 2330, 2445
\noindent 357, 612, 617, 1573, 1575, 1579, 1583, 1611, 1639, 1641, 1654, 1670, 1672, 1680, 1849, 1902, 1905, 1908, 1911, 2376, 2675, 2687, 2829
\nolinkurl 2508
\normalsize 1238
\notin 618
\null 2680

O

\o@join 2347, 2350, 2351, 2352
\Omega 1718
\omega 2137, 2138, 2158, 2159
\or 2258

P

\pagestyle 1261
\par . . . 356, 377, 538, 1190, 1198, 1578, 1640, 1663, 1671, 2300, 2322, 2337, 2378, 2657, 2681, 2685, 2693, 2828, 2834
\paragraph 1238
\parindent 2369
\path 96, 143, 206, 576
\pgfkeys . . 2051, 2720, 2721, 2722, 2723, 2724, 2727, 2730, 2732
\pgfmath@count 2422, 2424, 2426
\pgfmath@smuggleone 2427
\pgfmathdeclarefunction 2421
\pgfmathint 2422
\pgfmathparse 1320, 2439, 2444, 2447, 2461, 2462
\pgfmathresult 1321, 2422, 2423, 2425, 2427, 2440, 2448, 2461, 2462
\pgfutil@empty 2423
\pgfutil@loop 2424
\pgfutil@repeat 2427
\preceq 1383
\prime 1362
\printbibliography . 1552
\ProvidesPackage 2, 20, 55, 223, 412, 473, 480, 544, 607, 622, 1077, 1113, 1226, 1266, 1275, 1280, 1309, 1347, 1357, 1429, 1512, 1527, 1533, 1555, 1701, 1861, 1874, 1953, 2036, 2102, 2127, 2132, 2176, 2342, 2355, 2384, 2465, 2472, 2477, 2557, 2572, 2663, 2667, 2698, 2740, 2777, 2797

Q

\QS@list 2302, 2313, 2317, 2324, 2330, 2335, 2338
\QS@select@equal 2277, 2281
\QS@select@greater 2278, 2282
\QS@select@smaller 2273, 2276, 2280
\QS@sort@a 2255, 2288, 2309, 2310
\QS@sort@b 2255, 2256
\QS@sort@c 2259, 2266
\QS@sort@d 2267, 2275
\QS@sort@empty . 2257, 2262
\QS@sort@single 2258, 2263
\QSinitialize 2180, 2292, 2327
\QSIr . 2263, 2269, 2277, 2297, 2311, 2316, 2319
\QSIrr 2298, 2311, 2312, 2320
\QSLr 2269, 2276, 2287, 2288, 2296, 2309, 2314, 2318
\QSpivotStep 2182, 2292, 2296, 2307
\QSr 2269
\QSRr 2278, 2299, 2310, 2315, 2321, 2330, 2331, 2332
\QSsortStep 2184, 2292, 2308, 2309
\quad 1880

R

\raisebox 1501
\relax 1564, 2269, 2312, 2314, 2315, 2316, 2424, 2426
\renewcommand . . . 1241, 1242, 1243, 1244, 1247, 1248, 1262, 1263, 1522, 1523, 1922
\repeat 2185
\RequirePackage 57, 160, 224, 228, 408, 414, 415, 475, 545, 624, 625, 1079, 1081, 1082, 1118, 1227, 1228, 1231, 1233, 1235, 1240, 1249, 1256, 1267, 1268, 1277, 1281, 1282, 1283, 1311, 1312, 1361, 1430, 1529, 1534, 1535, 1551, 1558, 1559, 1560, 1588, 1703, 1864, 1865, 2037, 2039, 2129, 2189, 2191, 2343, 2344, 2345, 2358, 2386, 2417, 2468, 2474, 2479, 2490, 2491, 2559, 2664, 2671, 2701, 2702, 2704, 2705, 2743, 2744, 2745, 2778, 2798
\right 1705
\RIGHTarrow . . . 2719, 2724
\Rightarrow 613, 618
\rightarrow 209, 502, 507, 515, 519, 521, 522, 524, 579, 587, 2099, 2602, 2609, 2611, 2614, 2619, 2624, 2625, 2630
\rightouterjoin 2351
\Roman 1246
\roman 1246, 1248
\romannumeral 2267
\rtimes 1863
\rule 2300, 2322, 2337, 2348

S

\sb 69, 79, 81, 110, 168, 514, 515, 519, 522, 523, 524, 1173, 1175, 1755, 1758, 1761, 1787, 1793, 1942, 2087, 2096, 2624, 2625, 2626, 2631, 2635, 2636, 2639, 2640, 2643, 2644, 2645
\scriptscriptstyle 579, 587, 595
\scriptsize 1192, 1441, 1448, 1454, 1516, 1517, 1520, 1521, 2748, 2801, 2831
\section 48
\seq . 1505, 1506, 1507, 1649, 1650, 1651, 1658
\setbox 2347
\setcounter 1239, 2301, 2323, 2337
\setganttlinklabel . . 1269, 1270, 1271, 1272
\setlength 2369, 2825, 2826, 2836

\setmainfont 1229
\setmainlanguage 409
\setminted 2496, 2497
\setminus 2002
\setsansfont 1230
\setul 2360
\sffamily 489, 1236, 1238, 1340, 2580
\shoveleft 1982
\shoveright 1986
\Sigma 66, 107, 165, 1142, 1143, 1203
\sigma 512, 514, 515
\SLASH 1570
\small 1679
\sort 2434
\sortList 2433, 2442
\square 477
\stepcounter 2221, 2226, 2231, 2234, 2236, 2240, 2242, 2246, 2248
\str .. 490, 499, 1592, 2111, 2120, 2581, 2594
\string 1984, 1994
\StrSubstitute . 2441, 2443
\strut 1809, 1813, 1817, 1821, 1825, 2469
\subseteq 2605, 2638, 2645

T

\tableofcontents ... 1565
\text 79, 81, 184, 1964, 2748, 2801
\textbf ... 1084, 1388, 1397, 1408, 1417, 1574, 1581, 1612, 1640, 1655, 1671, 1927
\textcolor 1500, 2747
\textit 953, 995, 996, 997, 998, 1681, 1972, 2032
\textsc 1363
\textsf 1574, 1655
\textstyle 1773, 1804
\texttt 1107, 1363, 1497, 1498, 1499, 1500, 2801
\thepage 1260, 1519
\theparagraph 1238
\Theta 1708
\thinspace 2801
\tikz 1501
tikz: bbaum 23
tikz: li binaer baum 21
\tikzchildnode 432
\tikzparentnode 432
\tikzset 98, 145, 211, 417, 443, 1285, 1432, 2077, 2203, 2407, 2561, 2779
\tikzumlset 2707
\times 209
\tiny 1090, 1096, 1102, 1500, 1569, 2506
\titleformat .. 1236, 1238
\titlespacing 1237
\tl . 29, 37, 65, 66, 67, 68, 69, 70, 73, 74, 75, 76, 77, 79, 81, 106, 107, 108, 109, 110, 111, 112, 115, 116, 117, 118, 119, 120, 121, 164, 165, 166, 167, 168, 169, 170, 173, 174, 175, 176, 177, 178, 179, 293, 297, 321, 325, 326, 327, 330, 335, 336, 337, 348, 349, 350, 351, 362, 368, 371, 374, 383, 397, 549, 552, 557, 558, 566, 567, 570, 571, 1181, 1202, 1203, 1204, 1205, 1208, 1209, 1210, 1211, 1738
\tmp 2004
\TmpPlaceEight 2059
\TmpPlaceFive 2056
\TmpPlaceFour 2055
\TmpPlaceNine 2060
\TmpPlaceOne 2052
\TmpPlaceSeven 2058
\TmpPlaceSix 2057
\TmpPlaceTen 2061
\TmpPlaceThree 2054
\TmpPlaceTwo 2053
\TmpScale 2072
\TmpTransitionEight 2048, 2069
\TmpTransitionFive 2045, 2066
\TmpTransitionFour 2044, 2065
\TmpTransitionNine 2049, 2070
\TmpTransitionOne 2041, 2062
\TmpTransitionSeven 2047, 2068
\TmpTransitionSix 2046, 2067
\TmpTransitionTen 2050, 2071
\TmpTransitionThree 2043, 2064
\TmpTransitionTwo 2042, 2063
\TmpX 2073
\TmpY 2074
\today 1517
\ttfamily 2361

U

\ul 1085, 2359, 2360
\umlaggreg 1048
\umlassoc 1030
\umlclass ... 631, 635, 639, 683, 684, 685, 732, 737, 742, 745, 803, 804, 805, 810, 811, 840, 870, 897, 898, 901, 952, 955, 994, 1000, 1001, 1019, 1020, 1021, 1040, 1041, 1042, 1043
\umldep 960
\umlHVHaggreg 750, 816, 1010
\umlinherit 689, 740, 905, 950, 958
\umlnote ... 691, 907, 1050
\umlreal 687, 748
\umlsimpleclass .. 646, 647, 648, 652, 654, 655, 656, 682, 835, 836, 837, 896, 948, 949
\umlstatic 842, 871
\umluniaggreg 903
\umluniassoc 666, 688, 904, 1028, 1029
\umlVHuniassoc .. 667, 668
\umlVHVdep 660, 661, 663, 664, 844, 845
\umlVHVinherit 643, 644, 649, 650, 657, 658, 813, 814, 838, 839, 1008, 1009
\umlVHVreal 807, 808, 1045, 1046
\UParrow 2721
\url 1690
\usemintedstyle 2493
\usetikzlibrary .. 58, 416, 1080, 1284, 1431, 2038, 2346, 2387, 2418, 2706, 2794

V

\value 2183
\varepsilon 491, 502, 503, 1128, 1755, 1761, 1790, 1801
\vfill 2469
\vrule 2676, 2680
\vspace 1626, 1644, 2689, 2691

X

\xappto .. 2445, 2451, 2455
\xdef 1314
\xintApply 2271
\xintApplyUnbraced .. 2270, 2276, 2277, 2278
\xintCSVtoList 2330
\xintFor 2220, 2225, 2230, 2235, 2241, 2247, 2288
\xintifEq 2281
\xintifForLast 2237, 2243, 2249
\xintifGt 2282
\xintifLt 2280

\xintLength 2255
\xintnthelt 2267

Z

\ZB 1697
\zB 1696
\zustandsnamens@liste ... 1146, 1153, 1154