lehramt-informatik

Hermine Bschlangaul *

September 12, 2021

Contents

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

2 Pakete **8**
2.1 abmessung.sty . . . 9
2.2 aufgaben-einbinden.sty . . . 10
2.3 aufgaben-metadaten.sty . . . 11
2.4 automaten.sty . . . 12
2.4.1 Endlicher Automat . . . 12
2.4.2 Kellerautomat . . . 14
2.4.3 Turingmaschine . . . 15
2.5 basis.sty . . . 18
2.5.1 IFs . . . 18
2.6 baum.sty . . . 23
2.6.1 Binärbaum . . . 24
2.6.2 AVL-Baum . . . 25
2.6.3 B-Baum . . . 26
2.7 checkbox.sty . . . 27
2.8 chomsky-normalform.sty . . . 28
2.8.1 Makro-Kürzel . . . 28
2.8.2 TeX-Markup-Grundgerüst . . . 28
2.8.3 Konkretes TeX-Markup-Beispiel . . . 28
2.9 cpm.sty . . . 31
2.9.1 Makro-Kürzel . . . 31
2.9.2 TeX-Markup-Beispiel: Graph . . . 31
2.9.3 TeX-Markup-Beispiel: Ergebnistabelle . . . 31
2.9.4 TeX-Markup-Beispiel: Nebenrechnungstabelle „Frühester Zeitpunkt“ 31
2.9.5 TeX-Markup-Beispiel: Nebenrechnungstabelle „Spätester Zeitpunkt“ 32
2.9.6 Hilfsmakros für Zeitpunkt-Tabelle . . . 33
2.10 cyk-algorithmus.sty . . . 35
2.10.1 Makro-Kürzel . . . 35
2.10.2 TeX-Markup-Beispiel . . . 35
2.11 entwurfsmuster.sty . . . 36
2.11.1 Namensschema der Entwurfsmuster-Makros: . . . 36
2.11.2 Reihenfolge . . . 36
2.11.3 Abstrakte Fabrik (Abstract Factory) . . . 36
2.11.4 Adapter . . . 38
2.11.5 Beobachter (Observer) . . . 39
2.11.6 Dekorierer (Decorator) . . . 41
2.11.7 Einfache Fabrik (Simple Factory) . . . 42

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

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

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

3 Index **130**

1 Klassen

1.1 Vorlage Theorie-Teil

```
\documentclass{bschlangaul-haupt}

\begin{document}

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

\chapter{Thema des Theorie-Teils}

\literatur

\end{document}
```

1.2 Vorlage Aufgabensammlung

```
\documentclass{bschlangaul-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{bschlangaul-aufgabe}

\begin{document}
\liAufgabenTitel{}
\section{
\index{DB}
\footcite{examen:}
}

\end{document}
```

2 Pakete

2.1 abmessung.sty

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

```

2.2 aufgaben-einbinden.sty

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

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

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

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

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

`\liExamensAufgabeTTA`

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

`\liExamensAufgabeTA`

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

`\liExamensAufgabeA`

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

2.3 aufgaben-metadaten.sty

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

```
\ExplSyntaxOn
```

`\liMetaSetze` Setze die Metadaten einer Aufgabe.

```
\def\liMetaSetze#1{
  \_setze_variablen_zurueck:

  \tl_clear:N \g_auf_examen_teilaufgabe_nr_tl

  \keys_set:nn { aufgabenmetadaten } {
    #1
  }

  \_setze_relativen_pfad:
}
```

`\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{
  \liMetaSetze{#1}

  \_gib_examen_titel: {}

  \section{\_gib_aufgaben_titel:}
}
```

```
\ExplSyntaxOff
```

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

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

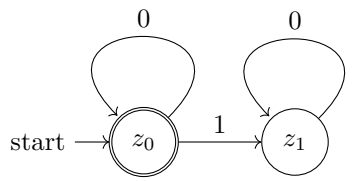
2.4 automaten.sty

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

2.4.1 Endlicher Automat

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

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

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




```
\RequirePackage{tikz}
\usetikzlibrary{arrows,automata,positioning}
\liLadePakete{mathe}
\directlua{
  automaten = require('bschlangaul-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}`: $A_{\text{DEA}} = (Z, \Sigma, \delta, E, z_0)$
- `\liAutomat{nea}`: $A_{\text{NEA}} = (Z, \Sigma, \delta, E, z_0)$

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

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

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

  \keys_set:nn { automat } { #2 }

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

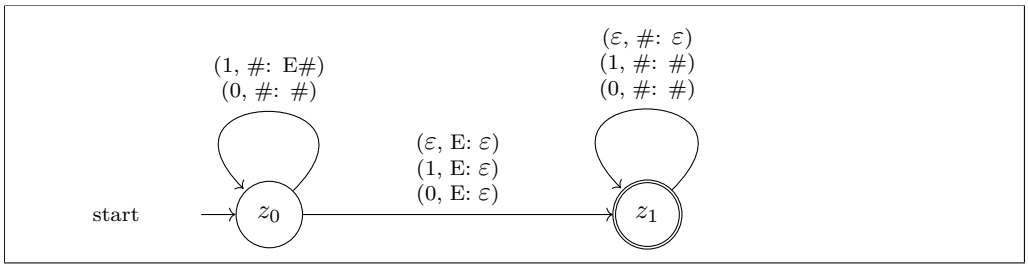
`\liAutomatenKante` **Let-Abkürzung:** `\let\k=\liAutomatenKante`

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

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

2.4.2 Kellerautomat

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

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

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

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



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

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

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

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

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

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

  \keys_set:nn { kellerautomat } { #2 }

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

`\liKellerUebergang` **Let-Abkürzung:** `\let\u=\liKellerUebergang`

`\liKellerUebergang{a, KELLERBODEN, A; b KELLERBODEN, EPSILON}`

(a, #: A)
(b, #: ε)

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

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

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

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

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

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

2.4.3 Turingmaschine

```
\RequirePackage{amssymb}
```

`\liTuringLeerzeichen` □

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

`\liTuringMaschine` `\liTuringMaschine[`⟨*automaten-name*⟩`]`
`{`⟨*zustaende=Z,alphabet=*Σ*,bandalphabet=*Γ*,delta=*δ*,start=*z_0*,leerzeichen=*$\square$*,ende=E*⟩`}`

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

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

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

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

  \keys_set:nn { kellerautomat } { #2 }

  $\text{#1} = (
    \l_zustaende_tl,
    \l_alphabet_tl,
    \l_bandalphabet_tl,
    \l_delta_tl,
    \l_start_tl,
    \l_leerzeichen_tl,
    \l_ende_tl
  )$
}
\ExplSyntaxOff
```

`\liTuringUebergangZelle` Formatiert einen Zustandsübergang für eine Übergangstabelle.
Let-Abkürzung: `\let\t=\liTuringUebergangZelle`
`\liTuringUebergangZelle{z_1, LEER, R}`: (z_1: $\square$, R) `\liTuringUebergangZelle{z1, leer, l}`: (z_1: $\square$, L)

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

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

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

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

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

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

\liTuringUeberfuehrung

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

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

2.5 basis.sty

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

2.5.1 IFs

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

`\ifLiADDITUM`
`\LiADDITUMtrue`
`\LiADDITUMfalse`

```
\newif\ifLiADDITUM
\LiADDITUMfalse
```

`\ifLiEXKURS`
`\LiEXKURStrue`
`\LiEXKURSfalse`

```
\newif\ifLiEXKURS
\LiEXKURStrue
```

`\ifLiANTWORT`
`\LiANTWORTtrue`
`\LiANTWORTfalse`

```
\newif\ifLiANTWORT
\LiANTWORTtrue
```

`\liLadePakete`

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

`\liLadeAllePakete`

```
\def\liLadeAllePakete{
  \liLadePakete{
    aufgaben-einbinden,
    automaten,
    checkbox,
    chomsky-normalform,
    cpm,
    cyk-algorithmus,
    entwurfsmuster,
    er,
    formale-sprachen,
    gantt,
    grafik,
    graph,
    hanoi,
    kontrollflussgraph,
    komplexitaetstheorie,
    makros,
    master-theorem,
    mathe,
    minimierung,
    normalformen,
    petri,
    potenzmengen-konstruktion,
    pumping-lemma,
    pseudo,
    quicksort,
    relationale-algebra,
    rmodell,
    sortieren,
    spalten,
```

```
    struktogramm,
    sql,
    syntax,
    syntaxbaum,
    synthese-algorithmus,
    tabelle,
    typographie,
    uml,
    vollstaendige-induktion,
    wasserfall,
    wpkalkuel,
    %
    baum, % am Schluss sonst Fehler: undefined command \edge
  }
}
```

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,
  identische_aufgabe,
  %
  examen_nummer,
  examen_fach,
  examen_jahr,
  examen_monat,
  examen_jahreszeit,
  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,
  IdentischeAufgabe .tl_gset:N = \g_auf_identische_aufgabe_tl,
  %
  ExamenNummer .tl_gset:N = \g_auf_examen_nummer_tl,
  ExamenFach .tl_gset:N = \g_auf_examen_fach_tl,
  ExamenJahr .tl_gset:N = \g_auf_examen_jahr_tl,
  ExamenMonat .tl_gset:N = \g_auf_examen_monat_tl,
  ExamenJahreszeit .tl_gset:N = \g_auf_examen_jahreszeit_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_set:Nn \_trenner: {
  \, / \,
}
\cs_gset:Npn \_gib_jahreszeit_durch_monat: #1 {
  % \str_case funktioniert nicht mit den Tokenlist variablen.
  \tl_case:Nn { #1 }
  {
    { 3 } { Frühjahr }
    { 03 } { Frühjahr }
    { 9 } { Herbst }
    { 09 } { Herbst }
  }
}
```

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

```
\cs_gset:Npn \_gib_examen_fach_durch_nummer: #1 {
  \tl_case:Nn { #1 }
  {
    { 46110 } { Grundlagen der Informatik (nicht vertieft) }
    { 46111 } { Programmentwicklung / Systemprogrammierung / Datenbanksysteme (nicht vertieft
    { 46112 } { Grundlagen der Informatik (nicht vertieft) }
```

```
    { 46113 } { Theoretische Informatik (nicht vertieft) }
    { 46114 } { Algorithmen / Datenstrukturen / Programmiermethoden (nicht vertieft) }
    { 46115 } { Theoretische Informatik / Algorithmen / Datenstrukturen (nicht vertieft) }
    { 46116 } { Softwaretechnologie / Datenbanksysteme (nicht vertieft) }
    { 46118 } { Fachdidaktik (Mittelschulen) }
    { 46119 } { Fachdidaktik (Realschulen) }
    { 46121 } { Fachdidaktik (berufliche Schulen) }
    { 66110 } { Automatentheorie, Algorithmische Sprache (vertieft) }
    { 66111 } { Betriebssysteme / Datenbanksysteme / Rechnerarchitektur (vertieft) }
    { 66112 } { Automatentheorie / Komplexität / Algorithmen (vertieft) }
    { 66113 } { Rechnerarchitektur / Datenbanken / Betriebssysteme (vertieft) }
    { 66114 } { Datenbank- und Betriebssysteme (vertieft) }
    { 66115 } { Theoretische Informatik / Algorithmen (vertieft) }
    { 66116 } { Datenbanksysteme / Softwaretechnologie (vertieft) }
    { 66118 } { Fachdidaktik (Gymnasium) }
  }
}
```

Einzelprüfungsnummer / Jahr / Jahreszeit mit Trennzeichen

```
\cs_gset:Npn \_gib_einzelpruefung_trenner: {
  Staatsexamen ~
  \g_auf_examen_nummer_tl

  \_trenner:

  \g_auf_examen_jahr_tl

  \_trenner:

  \_gib_jahreszeit_durch_monat: \g_auf_examen_monat_tl
}
```

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

```
\cs_gset:Npn \_gib_aufgaben_pfad_trenner: {
  \tl_if_empty:NTF \g_auf_examen_thema_nr_tl {} {
    Thema ~ \g_auf_examen_thema_nr_tl \_trenner:
  }
  \tl_if_empty:NTF \g_auf_examen_teilaufgabe_nr_tl {} {
    Teilaufgabe ~ \g_auf_examen_teilaufgabe_nr_tl \_trenner:
  }
  \tl_if_empty:NTF \g_auf_examen_aufgabe_nr_tl {} {
    Aufgabe ~ \g_auf_examen_aufgabe_nr_tl
  }
}
```

```
\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: {
  \LehramtInformatikGithubDomain /
  \LehramtInformatikGithubTexRepo /
  blob /
  \LehramtInformatikGitBranch /
  \g_auf_relativer_pfad_tl
}
\cs_new:Npn \_gib_github_url_href: {
  \tl_if_empty:NTF \g_auf_relativer_pfad_tl {} {
    \url{ \_gib_github_url: }
  }
}
\cs_new:Npn \_gib_aufgaben_titel: {
  \g_auf_titel_tl

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

```
\def\LehramtInformatikRepository{/pfad/zum/repository}
\input /etc/lehramt-informatik.config.tex
```

biblatex not working with lualatex and babel

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

2.6 baum.sty

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

```
\RequirePackage{tikz}
```

für `li binaer baum`

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

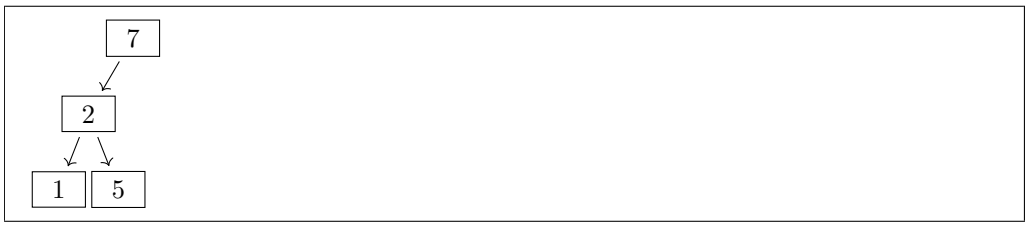
Für b baum

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

2.6.1 Binärbaum

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

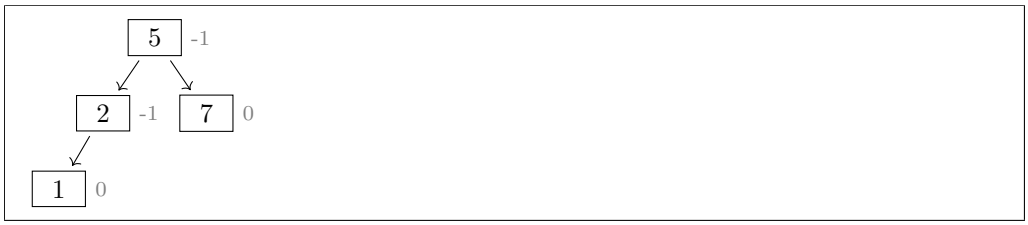
```
\begin{tikzpicture}[li binaer baum]
\Tree
[.7
  [.2
    [.1 ]
    [.5 ]
  ]
  \edge[blank]; \node[blank]{};
]
\end{tikzpicture}
```




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

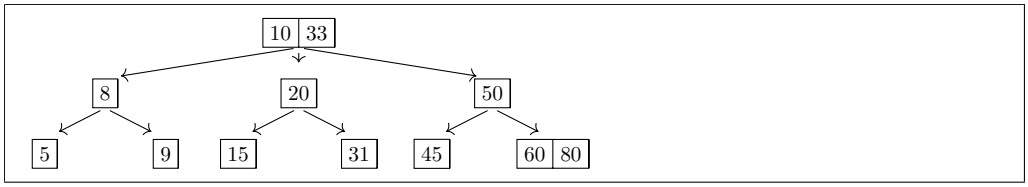
2.6.2 AVL-Baum

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



2.6.3 B-Baum

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




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

2.7 checkbox.sty

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

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

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

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

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

2.8 chomsky-normalform.sty

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

2.8.1 Makro-Kürzel

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

2.8.2 TeX-Markup-Grundgerüst

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

2.8.3 Konkretes TeX-Markup-Beispiel

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

\liNichtsZuTun

\item \schrittE{2}

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

\item \schrittE{3}

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

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

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

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

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

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

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

```
}
\def\liChomskyErklaerung#1{
  {
    \itshape
    \footnotesize
    \liParagraphMitLinien{\li@chomsky@erklaerung@texte{#1}}
  }
}
```

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

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

\ExplSyntaxOff

```

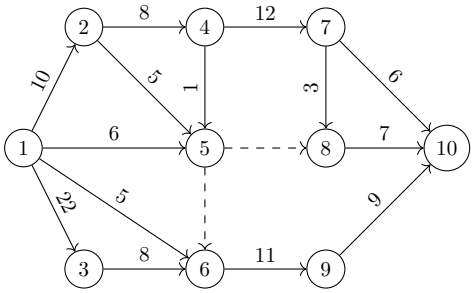
2.9 cpm.sty

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

2.9.1 Makro-Kürzel

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

2.9.2 TeX-Markup-Beispiel: Graph




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

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

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

2.9.3 TeX-Markup-Beispiel: Ergebnistabelle

```
\begin{tabular}{|l|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}
```

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

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

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

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

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

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

```
\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 {very~thick}},
  }

  \keys_set:nn { cpmVorgang } { #1 }

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

2.9.6 Hilfsmakros für Zeitpunkt-Tabelle

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

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

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

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

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

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

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

```
\ExplSyntaxOn
```

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

```
\NewDocumentCommand{ \liCpmSpaetI } { O{i} } {
  \ifmmode
    SZ\sb{#1}
  \else
    $SZ\sb{#1}$
  \fi
}
```

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

```
\NewDocumentCommand{ \liCpmFruehI } { O{i} } {
  \ifmmode
    FZ\sb{#1}
  \else
    $FZ\sb{#1}$
```

```
  \fi
}
```

\liCpmFruehErklaerung

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

```
\def\liCpmFruehErklaerung{
  \liParagraphMitLinien{
    Wir~führen~eine~Vorwärtsterminierung~durch~
    und~addieren~die~Dauern.~

    Kann~ein~Ereignis~über~mehrere~Vorgänge~erreicht~
    werden,~wählen~wir~das~Maximum~aus.~

    \textbf{Erläuterungen:}~

    $i$:~
    Ereignis~$i$;~\,

    \liCpmFruehI{}:~
    Frühester~Zeitpunkt,~zu~dem~Ereignis~$i$~eintreten~kann
    .
  }
}
```

\liCpmSpaetErklaerung

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

```
\def\liCpmSpaetErklaerung{
  \liParagraphMitLinien{
    Wir~führen~eine~Rückwärtsterminierung~durch~
    und~subtrahieren~die~Dauern~vom~letzten~Ereignis~aus.~

    Kann~ein~Ereignis~über~mehrere~Vorgänge~erreicht~
    werden,~wählen~wir~das~Minimum~aus.~

    \textbf{Erläuterungen:}~

    $i$:~
    Ereignis~$i$;~\,

    \liCpmSpaetI{}:~
    Spätester~Zeitpunkt,~zu~dem~Ereignis~$i$~eintreten~kann
    .
  }
}

\ExplSyntaxOff

```

2.10 cyk-algorithmus.sty

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

2.10.1 Makro-Kürzel

`\let\l=\liKurzeTabellenLinie`

2.10.2 TeX-Markup-Beispiel

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

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

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

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

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

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

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

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

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

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

2.11 entwurfsmuster.sty

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

2.11.1 Namensschema der Entwurfsmuster-Makros:

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

2.11.2 Reihenfolge

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

```
\RequirePackage{bschlangaul-uml}
```

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

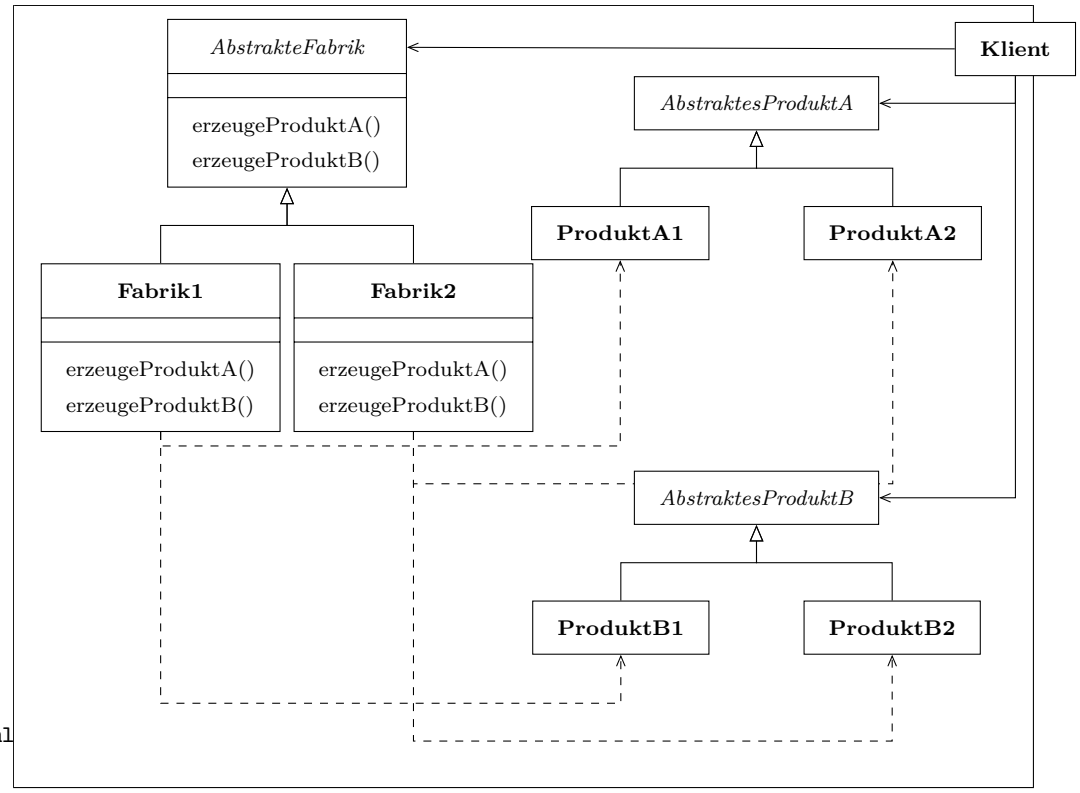
```
\RequirePackage{bschlangaul-syntax}
\def\li@EntwurfsCode#1#2{
  \liJavaDatei{entwurfsmuster/#1/allgemein/#2}
}
```

2.11.3 Abstrakte Fabrik (Abstract Factory)

`sAbstrakteFabrikBeschreibung`

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

\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}
}
```

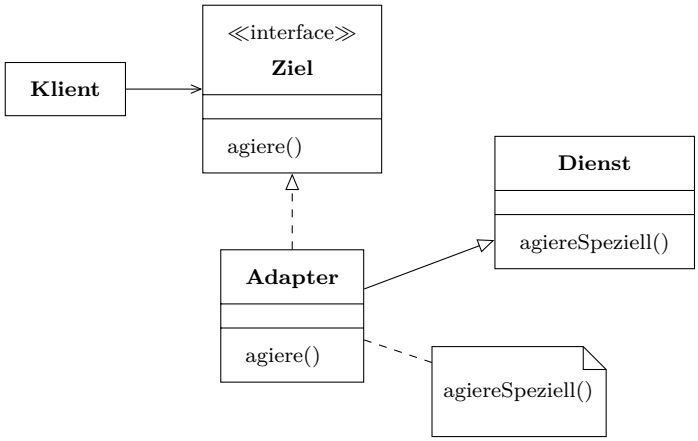
iEntwurfsAbstrakteFabrikCode

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

\liEntwurfsAbstrakteFabrik

```
\def\liEntwurfsAbstrakteFabrik{
  \liEntwurfsAbstrakteFabrikBeschreibung

  \liEntwurfsAbstrakteFabrikUml

  \liEntwurfsAbstrakteFabrikCode
}
```

2.11.4 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}
}
```

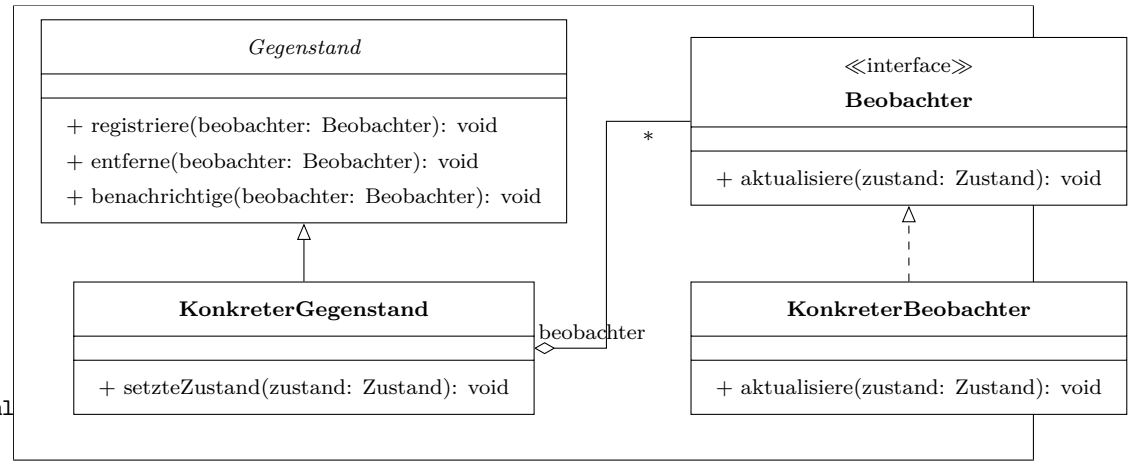
\liEntwurfsAdapterCode

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

\liEntwurfsAdapter

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

2.11.5 Beobachter (Observer)



\liEntwurfsBeobachterUml

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

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

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

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

\liEntwurfsBeobachterAkteure

Gegenstand / Subjekt (Subject / Observable) Ein Subjekt (beobachtbares Objekt, auf Englisch publisher, also „Veröffentlicher“, genannt) hat eine Liste von Beobachtern, ohne deren konkrete Typen zu kennen. Es bietet eine Schnittstelle zur An- und Abmeldung von Beobachtern und eine Schnittstelle zur Benachrichtigung von Beobachtern über Änderungen an.

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

konkreter/s Gegenstand / Subjekt (ConcreteSubject / ConcreteObservable) Ein konkretes Subjekt (konkretes, beobachtbares Objekt) speichert den relevanten Zustand und benachrichtigt alle Beobachter bei Zustandsänderungen über deren Aktualisierungsschnittstelle. Es verfügt über eine Schnittstelle zur Erfragung des aktuellen Zustands.

Konkrete Beobachter (ConcreteObserver) Konkrete Beobachter verwalten die Referenz auf ein konkretes Subjekt, dessen Zustand sie beobachten und speichern und dessen Zustand konsistent ist. Sie implementieren eine Aktualisierungsschnittstelle unter Verwendung der Abfrageschnittstelle des konkreten Subjekts.

```
\def\liEntwurfsBeobachterAkteure{
  \begin{description}
    \item[Gegenstand / Subjekt (Subject / Observable)]

    Ein Subjekt (beobachtbares Objekt, auf Englisch publisher, also
    „Veröffentlicher", genannt) hat eine Liste von Beobachtern, ohne
    deren konkrete Typen zu kennen. Es bietet eine Schnittstelle zur An-
    und Abmeldung von Beobachtern und eine Schnittstelle zur
    Benachrichtigung von Beobachtern über Änderungen an.\footcite[Seite
    251]{gof}

    \item[Beobachter (Observer)]

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

    \item[konkreter/s Gegenstand / Subjekt (ConcreteSubject / ConcreteObservable)]

    Ein konkretes Subjekt (konkretes, beobachtbares Objekt) speichert
    den relevanten Zustand und benachrichtigt alle Beobachter bei
```

```
    Zustandsänderungen über deren Aktualisierungsschnittstelle. Es
    verfügt über eine Schnittstelle zur Erfragung des aktuellen
    Zustands.

    \item[Konkrete Beobachter (ConcreteObserver)]

    Konkrete Beobachter verwalten die Referenz auf ein konkretes
    Subjekt, dessen Zustand sie beobachten und speichern und dessen
    Zustand konsistent ist. Sie implementieren eine
    Aktualisierungsschnittstelle unter Verwendung der
    Abfrageschnittstelle des konkreten Subjekts.
    \footcite{wiki:beobachter}
  \end{description}
}
```

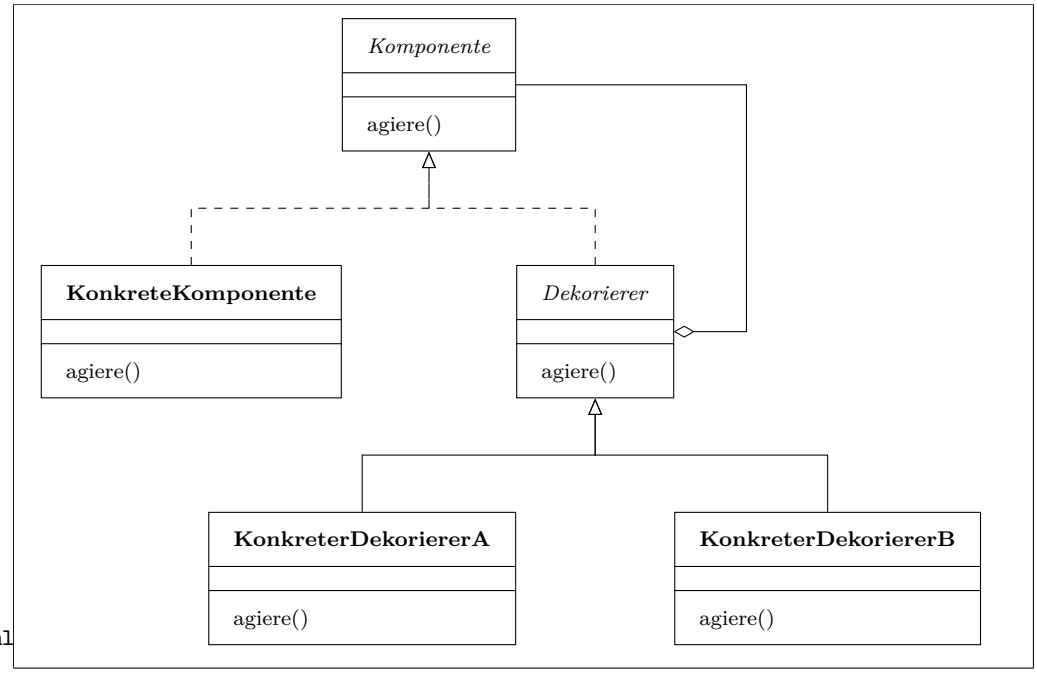
\liEntwurfsBeobachterCode

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

\liEntwurfsBeobachter

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

2.11.6 Dekorierer (Decorator)



\liEntwurfsDekoriererUml

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

```

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

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

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

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

`\liEntwurfsDekoriererCode`

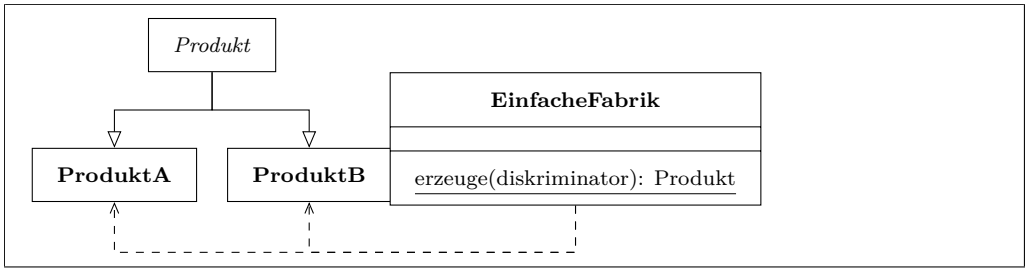
```
\def\liEntwurfsDekoriererCode{
  \li@EntwurfsCode{dekorierer}{Komponente}
  \li@EntwurfsCode{dekorierer}{KonkreteKomponente}
  \li@EntwurfsCode{dekorierer}{Dekorierer}
  \li@EntwurfsCode{dekorierer}{KonkreterDekoriererA}
  \li@EntwurfsCode{dekorierer}{KonkreterDekoriererB}
  \li@EntwurfsCode{dekorierer}{Klient}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

`\liEntwurfsDekorierer`

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

2.11.7 Einfache Fabrik (Simple Factory)

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




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

`ntwurfsEinfacheFabrikAkteure` Quelle: frei überstetzt aus GoF

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

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

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

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

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

    \item[Produkt]

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

    \item[KonkretesProdukt]

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

\liEntwurfsEinfacheFabrik

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

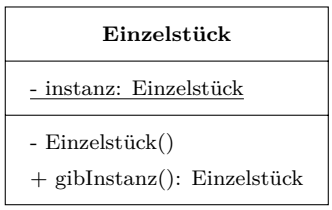
2.11.8 Einzelstück (Singleton)

urfsEinzelstueckBeschreibung

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

```

\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}
}
```

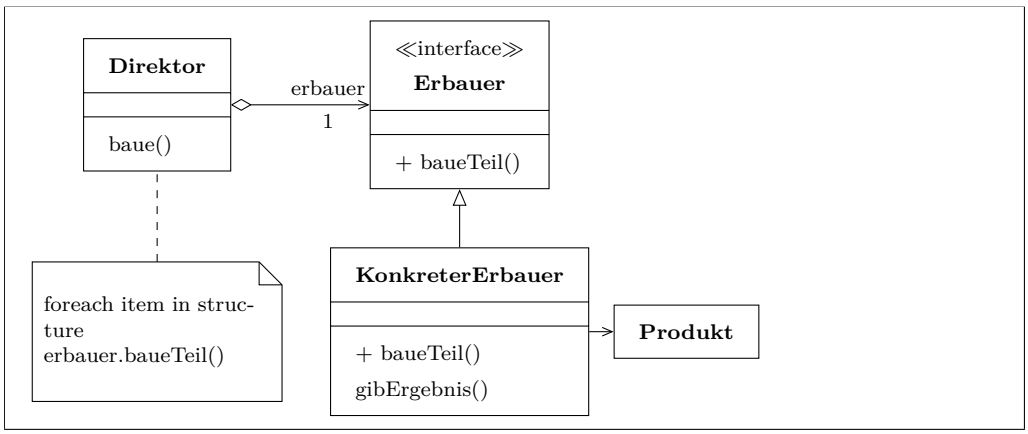
\liEntwurfsEinzelstueck

```
\def\liEntwurfsEinzelstueck{
  \liEntwurfsEinzelstueckBeschreibung

  \liEntwurfsEinzelstueckUml

  \liEntwurfsEinzelstueckAkteure

  \liEntwurfsEinzelstueckCode
}
```

2.11.9 Erbauer (Builder)

\liEntwurfsErbauerUml

Quelle nach der deutschen Wikipedia




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

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

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

```
  }
  \end{tikzpicture}
  \footcite{wiki:erbauer}
}
```

\liEntwurfsErbauerAkteure

Quelle: deutsche Wikipedia

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

KonkreterErbauer Der konkrete Erbauer erzeugt die Teile des komplexen Objekts durch Implementierung der Schnittstelle. Außerdem definiert und verwaltet er die von ihm erzeugte Repräsentation des Produkts. Er bietet auch eine Schnittstelle zum Auslesen des Produkts.

Direktor Der Direktor konstruiert ein komplexes Objekt unter Verwendung der Schnittstelle des Erbauers. Der Direktor arbeitet eng mit dem Erbauer zusammen: Er weiß, welche Baureihenfolge der Erbauer verträgt oder benötigt. Der Direktor entkoppelt somit den Konstruktionsablauf vom Klienten.

Produkt Das Produkt repräsentiert das zu konstruierende komplexe Objekt.

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

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

    \item[KonkreterErbauer]

    Der konkrete Erbauer erzeugt die Teile des komplexen Objekts durch
    Implementierung der Schnittstelle. Außerdem definiert und verwaltet er
    die von ihm erzeugte Repräsentation des Produkts. Er bietet auch eine
    Schnittstelle zum Auslesen des Produkts.

    \item[Direktor]

    Der Direktor konstruiert ein komplexes Objekt unter Verwendung der
    Schnittstelle des Erbauers. Der Direktor arbeitet eng mit dem Erbauer
    zusammen: Er weiß, welche Baureihenfolge der Erbauer verträgt oder
    benötigt. Der Direktor entkoppelt somit den Konstruktionsablauf vom
    Klienten.

    \item[Produkt]

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

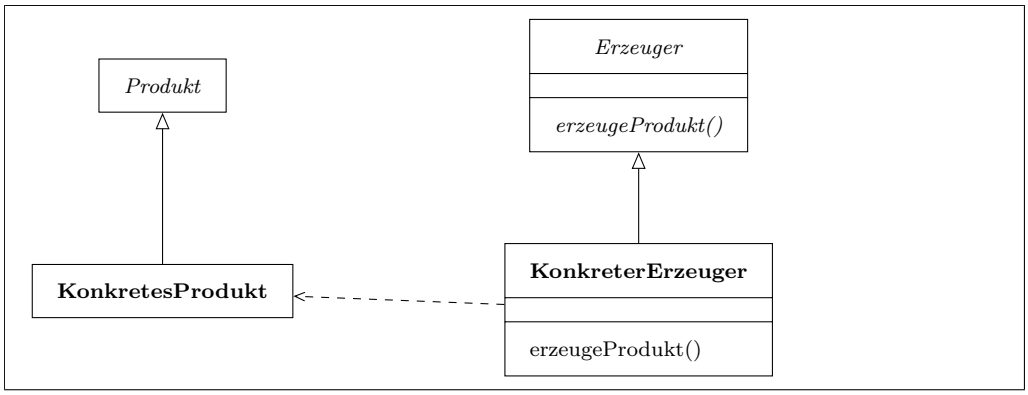
\liEntwurfsErbauer

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

2.11.10 Fabrikmethode (Factory Method)

\liEntwurfsFabrikmethodeUml

Quelle nach der deutschen Wikipedia




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

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

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

`EntwurfsFabrikmethodeAkteure` Quelle: deutsche Wikipedia

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

KonkretesProdukt KonkretesProdukt implementiert die Produkt-Schnittstelle.

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

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

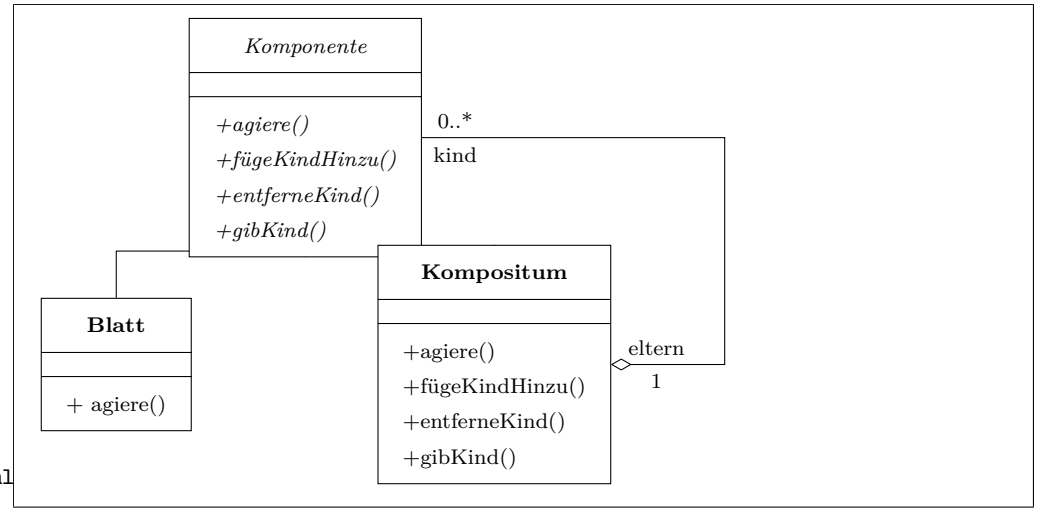
```
\def\liEntwurfsFabrikmethodeAkteure{
  \begin{description}
    \item[Produkt]

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

    \item[KonkretesProdukt]

    KonkretesProdukt implementiert die Produkt-Schnittstelle.

    \item[Erzeuger]

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

    \item[KonkreterErzeuger]

```

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

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

\liEntwurfsFabrikmethode

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

2.11.11 Kompositum (Composite)



\liEntwurfsKompositumUml

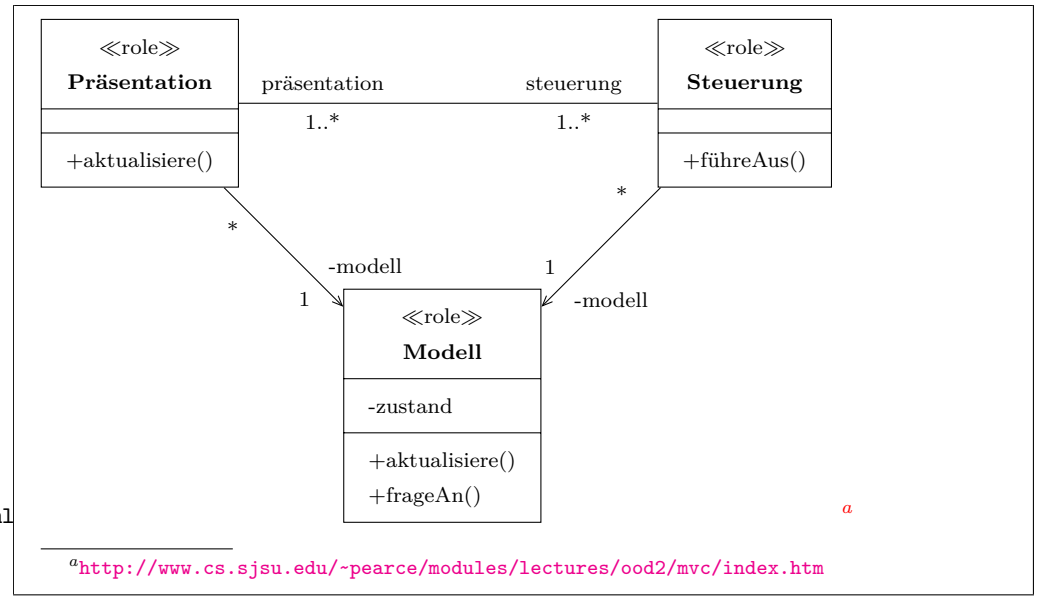
```
\def\liEntwurfsKompositumUml{
  \begin{tikzpicture}
    \umlclass[x=2.5,y=3,type=abstract]{Komponente}{}{
      \textit{+agiere()}\\
      \textit{+fügeKindHinzu()}\\
      \textit{+entferneKind()}\\
      \textit{+gibKind()}
    }
    \umlclass[x=0]{Blatt}{}{+ agiere()}
    \umlclass[x=5]{Kompositum}{}{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

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

\liEntwurfsFabrikmethode

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

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

odellPraesentationSteuerungUml




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

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

ModellPraesentationSteuerung

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

2.11.13 Stellvertreter (Proxy)

\liEntwurfsStellvertreterUml

```
\def\liEntwurfsStellvertreterUml{
  \begin{tikzpicture}
    \umlsimpleclass[x=-1,y=2]{Klient}

    \umlclass[x=2,y=2]{Subjekt}{}{+ agiere()}
    \umlclass[x=0,y=-1]{KonkretesSubjekt}{}{+ agiere()}
    \umlclass[x=4,y=-1]{Stellvertreter}{}{+ agiere()}

    \umlVHVinherit{KonkretesSubjekt}{Subjekt}
    \umlVHVinherit{Stellvertreter}{Subjekt}
    \umluniassoc{Stellvertreter}{KonkretesSubjekt}
    \umluniassoc{Klient}{Subjekt}
  \end{tikzpicture}
}
```

liEntwurfsStellvertreterCode

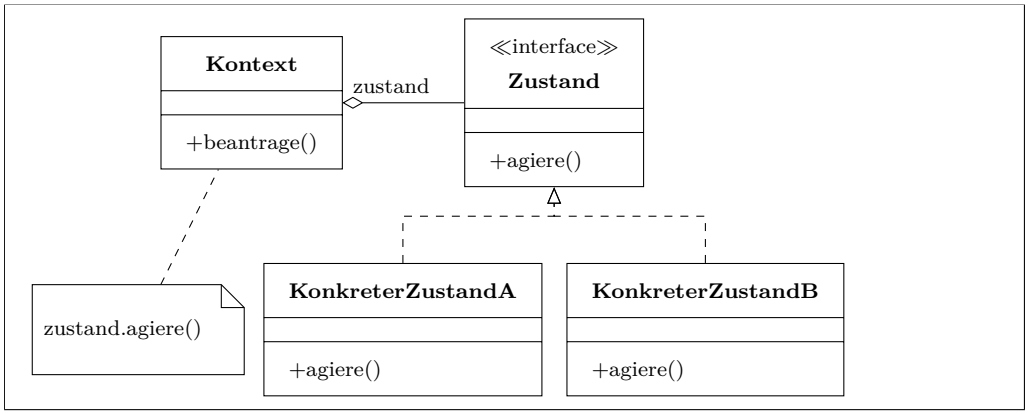
```
\def\liEntwurfsStellvertreterCode{
  \li@EntwurfsCode{stellvertreter}{Subjekt}
  \li@EntwurfsCode{stellvertreter}{KonkretesSubjekt}
  \li@EntwurfsCode{stellvertreter}{Stellvertreter}
  \li@EntwurfsCode{stellvertreter}{Klient}
}
```

`\liEntwurfsStellvertreter`

```
\def\liEntwurfsStellvertreter{
  \liEntwurfsStellvertreterUml
  \liEntwurfsStellvertreterCode
}
```

2.11.14 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}
}
```

\liEntwurfsZustandCode

```
\def\liEntwurfsZustandCode{
  \li@EntwurfsCode{zustand}{Kontext}
  \li@EntwurfsCode{zustand}{Zustand}
}
```

\liEntwurfsZustand

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

2.12 er.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-er}[2020/06/13 Zum Setzen von
ER-Diagrammen]
\RequirePackage{tikz-er2}
\usetikzlibrary{positioning}
```

2.12.1 Tex-Markup-Beispiel: Komplettes Diagramm

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

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

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

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

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

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

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

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

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

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

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

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

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

```
\node[attribute,above left=1cm of Produkt] {Bewertung} edge (Produkt);
\end{tikzpicture}
```

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

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

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

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

2.12.3 Makro-Kürzel

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

```
\ExplSyntaxOn
```

`\liErEntity`

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

`\liErRelationship`

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

`\liErAttribute`

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

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

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

□

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

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

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

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

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

datenbank name

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

```
\ExplSyntaxOff
```

2.13 formale-sprachen.sty

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

```
\directlua{
  formale_sprachen = require('bschlangaul-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}`: $\Gamma = \Sigma \cup \{\square\}$

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

`\liZustandsBuchstabe`

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

`\liZustandsBuchstabeGross`

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

\liZustandsmengeNr

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

\liZustandsmengeNrGross

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

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

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

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

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

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

```
\def\liAbleitung#1{$\directlua{formale_sprachen.formatiere_ableitung('#1')}$}
```

liProduktionsRegeln

```
\begin{liProduktionsRegeln}[P_1]
  S -> S A B | EPSILON,
  B A -> A B,
  A A -> a a,
  B B -> b b
\end{liProduktionsRegeln}
```

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

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

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

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

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

```
\ExplSyntaxOn
```

\liAusdruck `\liAusdruck[L_2]{a_1,a_2,\dots,a_n}{n \in N}`: $L_2 = \{ a_1, a_2, \dots, a_n \mid n \in N \}$
Ohne „=“: `\liAusdruck[]{x}{y}`: $\{ x \mid y \}$
Regulärer Ausdruck zum Konvertieren:

```
\$(.*) += +\\\{ *(.*?)( *\\, *)?\|( *\\, *)?(.*?) *\\\}\$
\\liAusdruck[$1]{$2}{$5}
```

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

\liFlaci Link zur flaci.com Website: `\liFlaci{Grxk1oczg}`:

Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Grxk1oczg

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

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

`\liGrammatik{variablen={}, alphabet={}}`

- `\liGrammatik{}`: $G = (V, \Sigma, P, S)$
- `\liGrammatik[G_1]{}`: $G_1 = (V, \Sigma, P, S)$
- `\liGrammatik{variablen={S,A,B}}`: $G = (\{S, A, B\}, \Sigma, P, S)$
- `\liGrammatik{alphabet={a,b}}`: $G = (V, \{a, b\}, P, S)$
- `\liGrammatik{start=X}`: $G = (V, \Sigma, P, X)$

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

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

  \keys_set:nn { grammatik } { #2 }

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

2.14 formatierung.sty

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

2.14.1 Schriftarten / Typographie

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

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

2.14.2 Farben

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

2.14.3 Überschriften

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

2.14.4 Listen

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

2.14.5 Kasten

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

`liKasten`

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

2.14.6 Header

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

2.14.7 Zeilenabstände

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

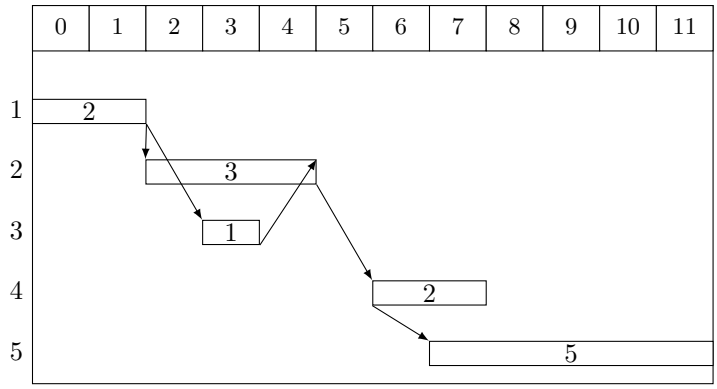
```
\RequirePackage{setspace}
```

1434

2.15 gantt.sty

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

```
\begin{ganttchart}[x unit=0.75cm, y unit chart=0.8cm]{0}{11}
\ganttitlelist{0,...,11}{1} \\
\ganttbar[name=1]{1}{0}{1} \\
\ganttbar[name=2]{2}{2}{4} \\
\ganttbar[name=3]{3}{3}{3} \\
\ganttbar[name=4]{4}{6}{7} \\
\ganttbar[name=5]{5}{7}{11}

\node at (1) {2};
\node at (2) {3};
\node at (3) {1};
\node at (4) {2};
\node at (5) {5};

\ganttlink[link type=f-f]{3}{2}
\ganttlink[link type=f-s]{1}{2}
\ganttlink[link type=f-s]{1}{3}
\ganttlink[link type=f-s]{2}{4}
\ganttlink[link type=s-s]{4}{5}
\end{ganttchart}
```




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

2.16 grafik.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-grafik}[2020/12/27 Lädt das Paket
TikZ, um Grafiken zeichnen zu können und graphicx um Bilder laden zu können.]
\ExplSyntaxOn
\RequirePackage{tikz}
\RequirePackage{graphicx}
```

\liGrafikLogoPfad

```
\def\liGrafikLogoPfad#1{
  \LehramtInformatikRepository / .tex / Logo / #1
}
```

\liGrafikCCLizenz

```
\NewDocumentCommand{ \liGrafikCCLizenz } { O{} } {
  \includegraphics[#1]{
    \liGrafikLogoPfad{CC-by-nc-sa.eps}
  }
}
```

\liGrafikLogo

```
\NewDocumentCommand{ \liGrafikLogo } { O{} } {
  \includegraphics[#1]{
    \liGrafikLogoPfad{Logo_nur-Pfade.eps}
  }
}
```

```
\ExplSyntaxOff
```

2.17 graph.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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 } {} {}
```

2.18 hanoi.sty

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

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

```
\RequirePackage{tikz}
\RequirePackage{xcolor}
```

\liHanoi `\liHanoi{anzahl-scheiben}{gewicht-scheibe/turm-nr,gewicht-scheibe/turm-nr}`, z. B.: `\liHanoi{4}{4/1,3/1,2/3,1/2}`

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

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

2.19 klassen-konfiguration-aufgabe.sty

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

Formatierung muss vor literatur sein, sonst Option clash

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

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

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

In Aufgaben wollen wir das Additum sehen.

```
\LiADDITUMtrue
```

```
\ExplSyntaxOn
```

```
\cs_gset:Npn \stichwoerter_formatiert: {
  \tl_if_empty:NTF \g_auf_stichwoerter_tl {} {
    \textbf{Stichwörter:} ~
    \g_auf_stichwoerter_tl
    \par
  }
}
```

```
\cs_gset:Npn \horizontale_linie: {
  \par
  \noindent
  \rule{\textwidth}{0.8pt}
  \par
}
```

```
\cs_gset:Npn \thematik_formatiert: {
  \tl_if_empty:NTF \g_auf_thematik_tl {} {
    \textit{
      ( \g_auf_thematik_tl )
    }
  }
}
```

`\liAufgabenMetadaten`

```
\def\liAufgabenMetadaten#1{
  \liMetaSetze{#1}

  \tl_if_empty:NTF \g_auf_examen_nummer_tl {} {
    {
      \noindent
      \large
      \_gib_einzelpruefung_trenner:
      \par\medskip
    }
  }

  {
    \noindent
    \bfseries
    \Large
    \tl_if_empty:NTF \g_auf_examen_nummer_tl {
```

```
      \g_auf_titel_tl
    } {
      \_gib_aufgaben_pfad_trenner:
    }
  }
  \hfill \thematik_formatiert:
  \par

  \medskip

  \noindent
  {\footnotesize\stichwoerter_formatiert:}

  \horizontale_linie:

  \bigskip

  \par
  % Keine Einrückung
  \@afterindentfalse
  \@afterheading
}

\AddToHook{enddocument}{
  \vfill
  {

    \liLogoTextProjekt
    \bigskip

    \liLogoTextCCLizenz
    \bigskip

    \begin{spacing}{1}
      \tiny
      \noindent
      \liMetaHilfMit

      \liMetaQuelltext
      \_gib_github_url_href:
     \end{spacing}
  }
}

\ExplSyntaxOff

```

2.20 klassen-konfiguration-examen.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-klassen-konfiguration-examen}[2021/09/04 Die
Klasse examen konfigurieren, d. h. Laden von einigen Paketen, Registierung von Hooks]
```

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

```
\liLadePakete{
  formatierung,
  literatur-dummy,
  makros,
  aufgaben-metadaten,
  abmessung,
  typographie,
  grafik,
  meta
}
```

Formatierung für die Überschriften setzen.

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

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

```
\RequirePackage[ngerman]{babel}
\RequirePackage{standalone}
\ExplSyntaxOn
```

`\liSetzeExamenThemaNr`

```
\def\liSetzeExamenThemaNr#1{
  \tl_gset:Nn \g_auf_examen_thema_nr_tl { #1 }
  \section{Thema~Nr.~#1}
}
```

`\liSetzeExamenTeilaufgabeNr`

```
\def\liSetzeExamenTeilaufgabeNr#1{
  \tl_gset:Nn \g_auf_examen_teilaufgabe_nr_tl { #1 }
  \subsection{Teilaufgabe~Nr.~#1}
}
```

`\liBindeAufgabeEin`

```
\def\liBindeAufgabeEin#1{
  \tl_gset:Nn \g_auf_examen_aufgabe_nr_tl { #1 }
  \input{
    \LehramtInformatikRepository /
    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_teilaufgabe_nr_tl /
    }
    Aufgabe - \g_auf_examen_aufgabe_nr_tl .tex
  }
}
```

`\liAufgabenMetadaten` Das Metadaten-Makro überschreiben

```
\def\liAufgabenMetadaten#1{
  \liMetaSetze{#1}
  \subsubsection{\_gib_aufgaben_titel:}
}
```

```
\cs_new:Npn \titel_seite:
{
  \pagestyle{empty}
  \begin{center}
    \large
    Erste~Staatsprüfung~für~ein~Lehramt~an~öffentlichen~Schulen \par

    \vspace{0.5cm}

    Fach~Informatik \par

    \vfill

    \liGrafikLogo[width=8cm]\par
    \bigskip
    Die~Bschlangaul-Sammlung \par
    {\footnotesize \liMetaHermineBschlangaulAndFriends} \par

    \vfill

    {
      \bfseries\Huge

      \g_auf_examen_jahreszeit_tl \par

      \g_auf_examen_jahr_tl \par
    }

    \vspace{2cm}

    {\LARGE \g_auf_examen_nummer_tl \par}

    \vspace{0.5cm}

    \g_auf_examen_fach_tl \par

    \vspace{3cm}

    Aufgabenstellungen~mit~Lösungsvorschlägen \par

  \end{center}

  \vfill
}
\cs_new:Npn \inhalts_verzeichnis: {
  % für den Abstand vor den section im Inhaltsverzeichnis
  % https://tex.stackexchange.com/questions/241445/how-to-control-spacing-in-toc-for-different-sections
  \setlength{\cftbeforesecskip}{1.5cm}
  \setlength{\cftbeforesubsecskip}{0.5cm}
  \renewcommand{\cftsubsecafterpnum}{\vspace{\cftbeforesubsecskip}}
  \setlength{\cftbeforesubsubsecskip}{0.1cm}
  \renewcommand{\contentsname}{Aufgabenübersicht}
  \tableofcontents
}
\setcounter{tocdepth}{4}
\RequirePackage[titles]{tocloft}
\AddToHook{begindocument}{
  \titel_seite:

  \clearpage

  \inhalts_verzeichnis:
```

```

  \vfill

  \liLogoTextProjekt
  \bigskip

  \liLogoTextCCLizenz
  \bigskip

  \clearpage
}
\ExplSyntaxOff

```

2.21 komplexitaetstheorie.sty

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

2.21.1 Makro-Kürzel

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

```
\liLadePakete{mathe}
```

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

```
\RequirePackage{mdframed}
```

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

```
\def\liStrich#1{#1^\prime}
```

`\liProblemName` Zu Setzen von Problemnamen wie zum Beispiel SAT, COL, VERTEX COVER
Let-Abkürzung: `\let\n=\liProblemName`
`\liProblemName`: SAT VERTEX COVER

```
\def\liProblemName#1{\texttt{\textsc{#1}}}
```

`\liProblemBeschreibung` Zu setzen von Problem-Beschreibungen:

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

CLIQUE

Gegeben: Ein ungerichteter Graph $G = (V, E)$, eine Zahl $k \in \mathcal{N}$

Frage: Gibt es eine Menge $S \subseteq V$ mit $\mathtt{S} = k$, sodass für alle Knoten $u \neq v \in V$ gilt, dass $\{u, v\}$ eine Kante in E ist?

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

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

    \medskip

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

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

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

\liProblemVertexCover

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

\liProblemVertexCover

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

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

\liProblemSubsetSum Kein Karp-21-Problem

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

\liProblemSubsetSum Kein Karp-21-Problem

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

2.22 kontrollflussgraph.sty

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

2.22.1 Makro-Kürzel

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

2.22.2 TeX-Markup-Beispiel

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

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

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

2.22.3 TikZ: pin

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

`pin={[options]angle:text}`

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

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

2.22.4 Umgebungen

liKontrollflussgraph

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

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

2.22.5 Makros

\liAnweisung

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

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

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

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

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

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

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

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

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

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

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

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

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

2.23 kopf-fusszeilen.sty

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

```
\ExplSyntaxOn
```

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

```
\ExplSyntaxOff
```

2.24 literatur-dummy.sty

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

\literatur

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

\footcite

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

```

2.25 literatur.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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]}
```

2.26 makros.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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}
```

```
\ExplSyntaxOn
```

`\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

  \par
  \noindent
  \textbf{#1}

  \medskip

  \par
  % Keine Einrückung
  \@afterindentfalse
  \@afterheading
}
```

`\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} {}
```

2.26.1 Umgebungen, die Inhalte aus- und einblenden können

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

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

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

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

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

liAntwort

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

```
\NewDocumentEnvironment{ liAntwort } { O{standard} }
{
  \ifLiANTWORT
  \else
    \setbox 0 \vbox
    \bgroup
  \fi

  \str_case:nn {#1} {
    {standard} {
      \def\beschriftung{}
      \mdfsetup{backgroundcolor=gray!1}
    }
    {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}[
    frametitle={\scriptsize\mdseries Lösungshinweise},
    innertopmargin=6pt,
    frametitleaboveskip=-12pt,
    frametitlealignment=\raggedright
  ]
}
{
  \end{mdframed}
  \ifLiANTWORT
  \else
    \egroup
  \fi
}
```

`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 }
{
  \ifLiADDITUM
  \else
    \setbox 0 \vbox
    \bgroup
  \fi

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

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

`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}{
  \ifLiEXKURS
    \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}
  \else
  \fi
}{}
```

`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}
  %
  \par
  \@afterindentfalse
  \@afterheading
} {}
```

`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}
} {}
```

`\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. }
```

```
\ExplSyntaxOff
```

2.27 master-theorem.sty

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

2.27.1 Makro-Kürzel

```
\let\O=\liO
\let\o=\liOmega
\let\T=\liT
\let\t=\liTheta


 \liMasterVariablenDeklaration
 {3} % a
 {3} % b
 {\mathcal{O}(1)} % f(n)

 \liMasterFallRechnung
 % 1. Fall
 {für $\varepsilon = 4$: \\
 $f(n) = 5n^2 \in \O{n^{\log_2 {8 - 4}}} = \O{n^{\log_2 4}} = \O{n^2}$}
 % 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 rekursiven Aufrufe, Anzahl der Unterprobleme in der
  Rekursion
  ($a \geq 1$).

  \item[$\textstyle{\frac{1}{b}} =$]
  Teil des Originalproblems, welches wiederum durch alle Unterprobleme
  repräsentiert wird, Anteil an der Verkleinerung des Problems ($b > 1$).

  \item[$f(n) =$]
  Kosten (Aufwand, Nebenkosten), die durch die Division des Problems und
  die Kombination der Teillösungen entstehen. Eine von $T(n)$
  unabhängige und nicht negative Funktion.
  \end{itemize}
  \footcite{wiki:master-theorem}
  \footcite[Seite 19-35]{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}
}
```

2.28 mathe.sty

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

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

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

```

2.29 meta.sty

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

```
\ExplSyntaxOn
```

```
\liLadePakete{grafik}
```

2.29.1 Einfache Makros (Low level)

\liMetaBschlangaulSammlung

```
\def\liMetaBschlangaulSammlung{
  Die~Bschlangaul-Sammlung
}
```

HermineBschlangaulAndFriends

```
\def\liMetaHermineBschlangaulAndFriends{
  Hermine~Bschlangaul~and~Friends
}
```

\liMetaUeberDasProjekt

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

\liMetaCCLink

```
\def\liMetaCCLink{
  Diese~Materialsammlung~unterliegt~den~Bestimmungen~der~
  \href{
    https://creativecommons.org/licenses/by-nc-sa/4.0/deed.de
  }
  {
    Creative~Commons~Namensnennung-Nicht~kommerziell-Share~Alike~4.0~
    International-Lizenz
  }.
}
```

\liMetaEmailLink

```
\def\liMetaEmailLink{
  \href{
    mailto:hermine.bschlangaul@gmx.net
  }{
    hermine.bschlangaul@gmx.net
  }
}
```

\liMetaHilfMit

```
\def\liMetaHilfMit{
  Hilf~mit!~

  Die~Hermine~schafft~das~nicht~alleine!~

  Das~ist~ein~Community-Projekt.~

  Verbesserungsvorschläge,~Fehlerkorrekturen,~weitere~Lösungen~sind~
  herzlich~willkommen~-~egal~wie~-~per~Pull-Request~oder~per~E-Mail~an~
  \liMetaEmailLink.
}
```

\liMetaHilfMit

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

2.29.2 Zusammengesetzte Makros (High level)

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

```
\cs_new:Npn \logo_dann_text:nn #1 #2 {
  \begin{center}
    \begin{minipage}[c]{5.5cm}
      #1
    \end{minipage}

    \begin{minipage}[c]{10cm}
      #2
    \end{minipage}
  \end{center}
}
```

\liLogoTextProjekt

```
\def\liLogoTextProjekt
{
  \logo_dann_text:nn
  {
    \liGrafikLogo[width=5cm]
  }
  {
    {
      \bfseries
      \liMetaBschlangaulSammlung
    }
    \par

    \liMetaHermineBschlangaulAndFriends
    \par

    \medskip

    \begin{spacing}{1}
      \footnotesize
      \liMetaUeberDasProjekt
    \end{spacing}
  }
}
```

\liLogoTextCCLizenz

```
\def\liLogoTextCCLizenz
{
  \logo_dann_text:nn
  {
    \centerline{\liGrafikCCLizenz[width=3cm]}
  }
  {
    \begin{spacing}{1}
      \scriptsize
      \liMetaCCLink
    \end{spacing}
  }
}
```

```
\ExplSyntaxOff
```

2.30 minimierung.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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
```

2.31 normalformen.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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('bschlangaul-helfer')
  normalformen = require('bschlangaul-normalformen')
}
```

2.31.1 Makro-Kürzel

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

```
\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;
}
```

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

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

Regulärer Ausdruck zum Konvertieren: `\\item \$(.*) \\rightarrow (.*)\$ \$(.*) \\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\,})
}
```

2.32 o-notation.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-o-notation}[2021/09/08]
```

2.32.1 Makro-Kürzel

```
\let\O=\liONotationO
```

2.32.2 TeX-Markup-Beispiel: Funktionsgraphen mit pgfplots

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

```
\ExplSyntaxOn
\RequirePackage{amssymb}
\RequirePackage{pgfplots}
```

Für echte Teilmenge \subsetneq: $\subsetneq$

```
\RequirePackage{amssymb}
```

\liRundeKlammer

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

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

```
\cs_new:Npn \o_notation_O:n #1 {
  \mathcal{O} \liRundeKlammer{#1}
}
\def\liONotationO#1{
  \ifmmode
    \o_notation_O:n { #1 }
  \else
    $ \o_notation_O:n { #1 } $
  \fi
}
```

2.33 petri.sty

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

2.33.1 Makro-Kürzel

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

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

Für die Darstellungsmatrix

```
\RequirePackage{blkarray}
```

```
\def\TmpA#1{
  \liPetriSetzeSchluessel%
  \pgfkeys{/petri/.cd,#1}%
  \begin{tikzpicture}[li petri]
  \node at (-0.25,-0.25) {};
  \node at (\TmpX,\TmpY) {};

  \begin{scope}[transform canvas={scale=\TmpScale},x=2cm,y=2cm,]
    \node[place,tokens=\TmpPlaceOne,label=$p_1$] at (0,1) (p1) {};
    \node[place,tokens=\TmpPlaceTwo,label=$p_2$] at (2,2) (p2) {};
    \node[place,tokens=\TmpPlaceThree,label=east:$p_3$] at (2,0) (p3) {};

    \node[transition,label=east:$t_1$,\TmpTransitionOne] at (2,1) {}
      edge[pre] (p2)
      edge[post] (p3);
    \node[transition,label=$t_2$,\TmpTransitionTwo] at (1,1.5) {}
      edge[pre] (p1)
      edge[post] (p2);
    \node[transition,label=$t_3$,\TmpTransitionThree] at (1,0.5) {}
      edge[pre] (p3)
      edge[post] (p1);
    \node[transition,label=$t_4$,\TmpTransitionFour] at (1,1) {}
      edge[pre] (p2)
      edge[pre] (p3)
      edge[post] (p1);
  \end{scope}
  \end{tikzpicture}
}
```

\liPetriSetzeSchluessel

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

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

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

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

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

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

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

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

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

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

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

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

2.34 potenzmengen-konstruktion.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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
```

2.35 pseudo.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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}
```

2.36 pumping-lemma.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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}
}
```

2.37 quicksort.sty

```
% https://tex.stackexchange.com/a/142634
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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}}
}
```

2.38 relationale-algebra.sty

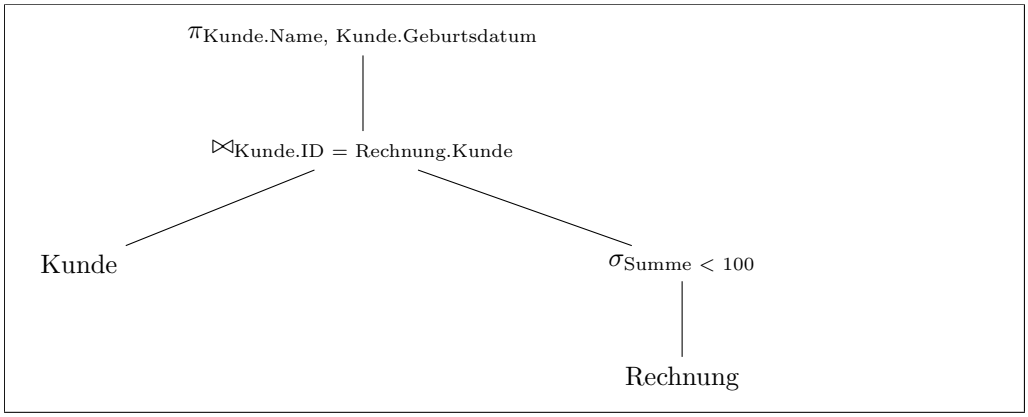
```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-relationale-algebra}[2020/12/11]
```

```
\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: $A \bowtie\!\!\!\!\!\!\;\;\; B$

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

`\rightouterjoin` A `\rightouterjoin` B: $A \;\;\!\!\!\!\!\!\bowtie B$

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

`\fullouterjoin` A `\fullouterjoin` B: $A \;\;\!\!\!\!\!\!\bowtie\!\!\!\!\!\!\;\;\; B$

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

2.39 rmodell.sty

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

2.39.1 Makro-Kürzel

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

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

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

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

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

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

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

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

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

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

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

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

```
\def\liAttribut#1{{\li@Rmodell@Schrift#1}}
```

`liRelationenSchemaFormat` Dummy-Umgebung, zum Parsen durch Java gedacht.

```
\begin{liRelationenSchemaFormat}
Springer(Startnummer*, Nachname, Vorname, Geburtsdatum, Körpergröße)
Sprung(SID*, Beschreibung, Schwierigkeit)
springt(SID[Sprung], Startnummer[Springer], Durchgang)
\end{liRelationenSchemaFormat}
```

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

2.40 sortieren.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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('bschlangaul-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}
```

2.41 spalten.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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}
```

2.42 sql.sty

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

2.42.1 Latex-Markup-Beispiel

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

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

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

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

```
\liLadePakete{syntax}
```

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

2.43 struktogramm.sty

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

```

2.44 syntax.sty

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

Um das Paket `minted` benutzen zu können, muss `pygmentize` installiert sein. Außerdem müssen die TeX-Dateien mit der Kommandozeilen-Option `--shell-escape` kompiliert werden.

2.44.1 Makro-Kürzel

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

```
\ExplSyntaxOn
```

```
\directlua{
  syntax = require('bschlangaul-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=false,
  fontsize=\footnotesize,
}
```

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

```
\newminted[liJavaAngabe]{java}{
  xleftmargin=1cm
}
```

`\liJavaCode` Im Zeilenfluss einen kurzen Java-Code-Ausschnitt setzen. Es werden automatische Zeilenumbrüche gemacht.
Let-Abkürzung: `\let\j=\liJavaCode`

```
\def\liJavaCode#1{
  \,
  \textcolor{blue}{
    \mintinline[
      fontsize=\normalsize,
      breakanywhere % https://github.com/gpoore/minted/issues/31#issuecomment-458640242
    ]{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|}

```

2.45 syntaxbaum.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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
    },
  }
}

```

2.46 synthese-algorithmus.sty

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

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

2.46.1 Makro-Kürzel

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

2.46.2 TeX-Markup Grundgerüst

```
\let\schrittE=\liSyntheseUeberErklaerung
\begin{enumerate}
\item \schrittE{1}

\begin{enumerate}
\item \schrittE{1-1}
\item \schrittE{1-2}
\item \schrittE{1-3}
\item \schrittE{1-4}
\end{enumerate}

\item \schrittE{2}
\item \schrittE{3}
\item \schrittE{4}
\end{enumerate}
```

2.46.3 TeX-Markup Linksreduktion

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

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

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

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

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

2.46.4 TeX-Markup Rechtreduktion

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

\liPseudoUeberschrift{F}

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

\liPseudoUeberschrift{A}

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

2.46.5 TeX-Markup Relationen formen

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

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

1. **Kanonische Überdeckung**

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

   (a) **Linksreduktion**

   — *Führe für jede funktionale 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
   \rmfamily
   \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
```

2.47 tabelle.sty

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

```

2.48 typographie.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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.

Variable $= \Big\{$

Inhalt

$\Big\}$

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

`\liTypoUeberschrift`

```
\def\liTypoUeberschrift#1{
  {
    \bfseries\rmfamily
    #1
  }
}
```

\liTypoUeberGross

```
\def\liTypoUeberGross#1{
  {
    \huge
    \liTypoUeberschrift{#1}
  }
}
```

\liTypoUeberGross

```
\def\liTypoUeberGROSS#1{
  {
    \Huge
    \liTypoUeberschrift{#1}
  }
}
```

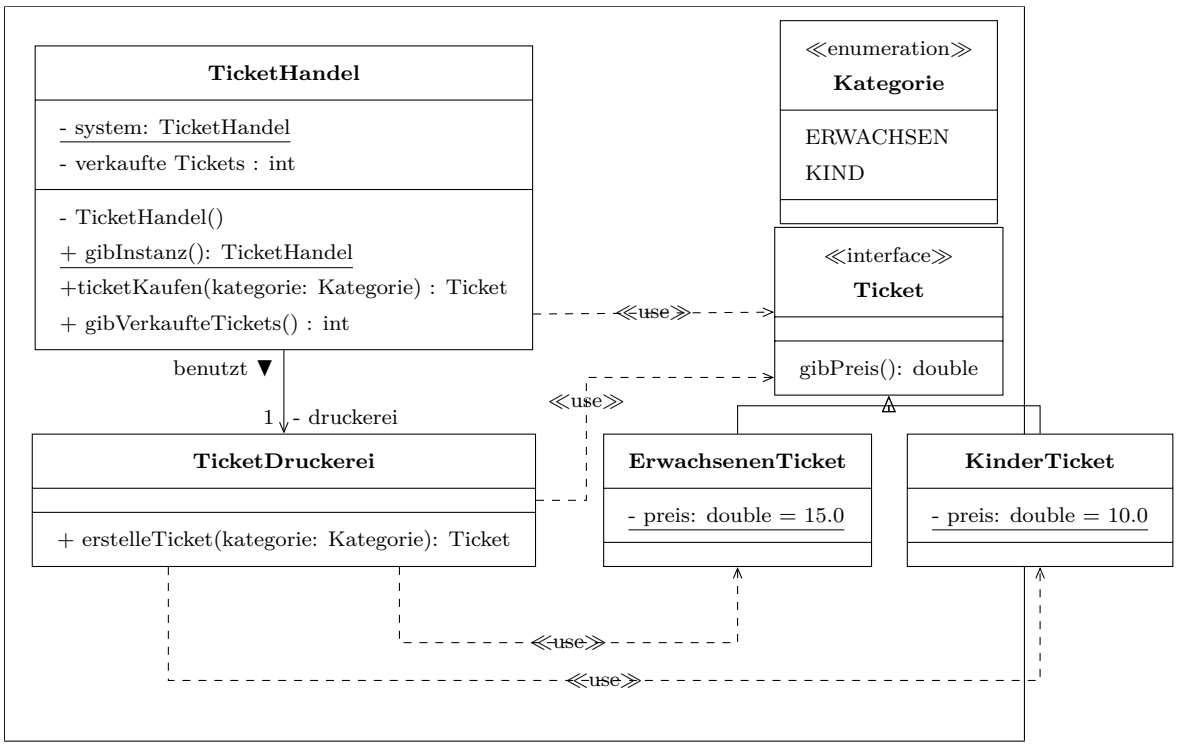
```
\ExplSyntaxOff
```

2.49 uml.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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
  };
}
```

2.50 vollstaendige-induktion.sty

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

2.50.1 Makro-Kürzel

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

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

2.51 wasserfall.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-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}

```

2.52 wpkalkuel.sty

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

2.52.1 Makro-Kürzel

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

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

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

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

\MatheEnv

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

  \hspace{1em}#1

  \medskip
}
```

\Mathe

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

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

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

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

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

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

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

\liWpErklaerungVerzweigung

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

\ExplSyntaxOff
```

3 Index

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

Symbols

\# 114
\, . . 364, 423, 478, 705, 723, 1355, 1357, 2139, 2140, 2141, 2578, 3079, 3086, 3273
\@Skip@Erklaerung@Reset . . . 3426, 3428, 3439
\@afterheading 1606, 1984, 2108
\@afterindentfalse 1605, 1983, 2107
\@liDirLeft 3321, 3326, 3338
\@liDirRight 3322, 3324, 3325, 3326, 3327, 3338
\@liDistance 3332, 3333, 3337
\@liPos . . 3329, 3330, 3337
\\ 735, 763, 764, 767, 768, 771, 772, 867, 868, 869, 976, 1010, 1012, 1042, 1051, 1096, 1138, 1139, 1140, 1145, 1146, 1147, 1167, 1971, 2473, 2531, 2534
\{ 212, 1292, 1302, 1314, 1315, 1320, 1354, 1810, 2518, 2943, 3273, 3442
\} 212, 1292, 1302, 1314, 1315, 1322, 1358, 1811, 2518, 2943, 3277, 3442
_ 37, 45, 50, 52, 311, 339, 363, 366, 376, 399, 403, 407, 409, 411, 413, 416, 422, 423, 437, 438, 444, 447, 450, 460, 467, 469, 472, 1576, 1588, 1624, 1678
\␣ 2530, 2540

A

\addbibresource 1940, 1941, 1942, 1943, 1944, 1945, 1946, 1947, 1948, 1949
\AddToHook 1608, 1736
\advance 2992
\AfterEndEnvironment 3068
\Alph 1415
\alph 1415, 1416
\alpha 3187, 3189, 3190, 3193, 3195, 3196, 3197, 3198, 3199, 3203, 3209, 3210, 3215, 3216, 3217, 3220, 3228, 3229, 3230
\arabic 1415, 2787, 2792, 2797, 2803, 2809, 2815
\arraystretch 2468

B

\BeforeBeginEnvironment 3067
\begin 761, 815, 830, 865, 889, 936, 968, 983, 1008, 1018, 1038, 1058, 1090, 1107, 1136, 1161, 1182, 1206, 1221, 1335, 1421, 1513, 1618, 1683, 1764, 1773, 1780, 1888, 1963, 2027, 2049, 2069, 2096, 2101, 2112, 2122, 2126, 2209, 2213, 2232, 2255, 2278, 2293, 2370, 2371, 2375, 2398, 2411, 2471, 2472, 2522, 2567, 2707, 2727, 2868, 2890, 2904, 3067, 3090, 3277
\begingroup 1964, 2520, 2988
\beschriftung 2007, 2011, 2015, 2019, 2023, 2025
\beta 3187, 3190, 3194, 3195, 3196, 3199, 3209, 3210, 3211, 3216, 3217
\bf 2765, 2766, 2767
\bfseries 564, 1405, 1407, 1583, 1644, 1701, 2388, 2765, 2771, 2773, 2775, 2776, 3164, 3282
\bgroup 2002, 2046
\Bigl 3273
\Bigr 3277
\bigskip 456, 737, 742, 1601, 1613, 1616, 1694, 1746, 1749, 1973, 2446
\bool 342, 424
\bowtie 2913, 2916, 2917, 2918
\Box 164
\boxtimes 554

C

\c 1515, 1516
\cdot 2184, 2240, 2251
\centerline 1769, 2408, 2866, 2888, 2903
\cftbeforesecskip . . 1727
\cftbeforesubsecskip 1728, 1729
\cftbeforesubsubsecskip 1730
\cftsubsecafterpnum 1729
\chapter 1405, 1406
\char 1971
\clearpage 1645, 1739, 1751, 1965

\cline 735
\clist 237, 285, 286, 308, 312, 3000
\columnbreak 3035
\contentsname 1731
\cs 311, 339, 363, 366, 376, 399, 411, 422, 423, 460, 467, 472, 1549, 1556, 1562, 1680, 1724, 2090, 2369, 2590
\csname 1500, 1503
\cup 1315, 2552, 3196, 3210, 3217

D

\DeclareMathSymbol 2317, 2318
\DecoINERT 2790, 2863, 2864, 2886
\DecoINERTwithPivot 2806, 2885
\DecoLEFT 2785, 2884
\DecoLEFTwithPivot 2800, 2862
\DecoRIGHT 2795, 2887, 2897
\DecoRIGHTwithPivot 2812, 2865
\definecolor 1403
\DefineVerbatimEnvironment 3041
\delta 70, 112, 170, 212, 1306
\dh 2141, 3198
\directlua 63, 142, 200, 205, 1287, 1301, 1321, 1329, 1336, 1341, 2503, 2508, 2561, 2568, 2575, 2955, 3054, 3098, 3103, 3104, 3108, 3113, 3114, 3118, 3124, 3125
\do 2786, 2791, 2796, 2801, 2807, 2813
\dots 595, 599, 1810, 2714, 3209, 3210
\DOWNarrow 3325
\draw 1520, 1523, 1526, 2662, 2961, 2964

E

\edef 1511, 2879, 2883, 2895, 2896
\edge 282
\egroup 2038, 2063
\else 659, 667, 675, 683, 690, 1296, 1310, 1346, 2000, 2023, 2037, 2044, 2062, 2087, 2157, 2167, 2177, 2191, 2514, 2551, 2596, 2657, 2825, 3018, 3020, 3409
\emph . . 756, 1003, 1258, 1791, 1820, 1822, 1970
\empty 2023, 2550
\emptyset 2460, 3204, 3225, 3258
\end 800, 826, 851, 886, 920, 952, 980, 996, 1015, 1023, 1054, 1083, 1104, 1129, 1154, 1174, 1193, 1218, 1236, 1337, 1423, 1529, 1625, 1720, 1776, 1777, 1784, 1893, 1968, 2035, 2059, 2085, 2103, 2104, 2118, 2129, 2130, 2211, 2227, 2252, 2275, 2290, 2300, 2373, 2377, 2378, 2401, 2414, 2475, 2476, 2524, 2569, 2717, 2737, 2868, 2890, 2904, 3068, 3094, 3277
\endcsname 1500, 1503
\endgroup 1967, 2525, 2993
\enspace 3262, 3264
environments:
liAdditum 2041
liAHuelle 2519
liAntwort 1997
liDiagramm 2120
liEinbettung 1996
liExkurs 2066
liGraphenFormat . 1492
liJavaAngabe 3075
liKasten 1420
liKontrollflussgraph 1887
liLernkartei 2110
liProduktionsRegeln 1330
liProjektSprache 1995
liQuellen 2090
liRelationenSchemaFormat 2947
liRmodell 2927
liUebergangsTabelle 2468
\equiv 3424, 3443
\erzeuge@tiefgestellt . . . 1301, 1302, 1306
\expandafter 1500, 2821, 2823, 2824, 2825, 2833, 2991
\ExplSyntaxOff . . . 54, 97, 139, 144, 197, 202, 207, 617, 639, 654, 730, 1282, 1360, 1395, 1463, 1628, 1753, 1908, 1924, 2142, 2207, 2417, 2496, 2690, 2940, 3001, 3138, 3245, 3298, 3377, 3448
\ExplSyntaxOn 35, 66, 107, 140, 165, 198, 203, 228, 560, 623, 640, 679, 1255, 1350, 1372, 1447, 1548, 1650, 1901, 1913, 1961, 2146, 2323, 2478, 2583, 2671, 2928, 2998, 3053, 3161, 3255, 3349, 3402

F

\faCheckSquareO 3257
\faCircleThin 1274
\faGg 1268
\fancyfoot 1427, 1428, 1429, 1917, 1918, 1919, 1920
\fancyhead 1426, 1914, 1915, 1916
\faSquareO 1262
\fi 661, 669, 677, 685, 692, 1298, 1312, 1348, 2003, 2026, 2039, 2047, 2064, 2088, 2159, 2169, 2179, 2193, 2516, 2553, 2598, 2659, 2826, 3018, 3019, 3022, 3024, 3411
\filcenter 1644
\footcite 758, 827, 849, 896, 919, 951, 1004, 1055, 1082, 1128, 1789, 1792, 1799, 1804, 1809, 1813, 1819, 1824, 1929, 2228, 2229, 2483, 2720
\footnote 2133, 2137
\footnotesize 152, 433, 515, 609, 1279, 1597, 1696, 1994, 2080, 2097, 2113, 2399, 2427, 2521, 2538, 2546, 2675, 2684, 2927, 3042, 3073, 3237, 3312, 3338
\footrulewidth . 1432, 1922
\foreach . 1515, 1518, 1525
\forestFirst . . 3015, 3018
\forestLast . . . 3016, 3018
\forestOget . . . 3015, 3016
\forestOnes 3028
\forestOv 3017, 3018, 3021
\forestov . 3007, 3011, 3012, 3015, 3016, 3017, 3018, 3020, 3021
\forestset 3003, 3026
\forestSortLevel 3005, 3013, 3027, 3028

\frac 2186, 2219, 2251, 2266
\fullouterjoin 2918

G

\g ... 39, 285, 286, 308, 312, 318, 319, 320, 321, 322, 324, 325, 327, 328, 330, 331, 332, 333, 334, 335, 336, 337, 340, 344, 345, 346, 349, 351, 352, 353, 354, 355, 356, 401, 405, 409, 412, 413, 415, 416, 418, 419, 426, 427, 428, 429, 437, 438, 440, 446, 447, 449, 450, 452, 453, 465, 468, 473, 475, 479, 1550, 1552, 1563, 1565, 1572, 1585, 1586, 1652, 1656, 1660, 1664, 1665, 1666, 1667, 1668, 1670, 1671, 1673, 1703, 1705, 1710, 1714
\Gamma . 111, 169, 212, 1315
\gappto 3023
\geometry 5
\geq 1817, 2217, 2703, 2708, 2724, 2728

H

\hbox 2913
\headrulewidth . 1431, 1921
\headwidth 1923
\hfill 1591, 2236, 2242, 2247, 3265
\hinweis 1994
\hline 2473
\horizontale .. 1556, 1599
\href 1368, 2137, 2304, 2339, 2348, 3093
\hspace .. 2665, 3416, 3424
\ht 2914
\Huge 1701, 3294
\huge 1644, 3288

I

\i 1525, 1526
\ifcase 2822
\ifLiADDITUM 229, 2043, 2061
\ifLiANTWORT 233, 1999, 2036
\ifLiEXKURS ... 231, 2067
\ifmmode 657, 665, 673, 681, 688, 1294, 1308, 1344, 2155, 2165, 2175, 2189, 2512, 2594, 2655, 3407
\IfNoValueTF 2050, 2133, 2137
\ifnum 2749, 2990, 3006, 3014, 3020
\ifx 2023, 2550, 3018
\in 583, 739, 1817, 2199, 2202, 2205, 2234, 2240, 2245, 2703, 2714, 2724, 2734, 3187, 3189, 3195, 3216, 3365
\includegraphics 1454, 1459
\inhalts 1724, 1741
\inhaltsverzeichnis 1962
\input 17, 20, 23, 26, 29, 485, 1661
\inputminted 3097, 3107, 3117, 3129, 3132, 3136
\int 3000
\item 554, 555, 832, 836, 841, 846, 890, 899, 904, 912, 984, 989, 993, 1019, 1059, 1064, 1071, 1079, 1108, 1113, 1117, 1122, 1222, 1227, 1232, 1774, 1775, 2090, 2094, 2214, 2219, 2223, 2233, 2239, 2244, 2256, 2260, 2264, 2268, 2272, 2279, 2283, 2287, 2708, 2711, 2714, 2728, 2731, 2734
\itshape 608, 3236

J

\j 1515, 1516, 1518, 1519, 1520, 1525, 1526, 1527

K

\k 1525
\keys 41, 75, 87, 117, 127, 175, 185, 316, 627, 631, 645, 650, 1379, 1386

L

\l 68, 69, 70, 71, 72, 73, 76, 77, 78, 79, 80, 82, 84, 89, 90, 91, 92, 93, 94, 109, 110, 111, 112, 113, 114, 115, 118, 119, 120, 121, 122, 123, 124, 130, 131, 132, 133, 134, 135, 136, 167, 168, 169, 170, 171, 172, 173, 176, 177, 178, 179, 180, 181, 182, 188, 189, 190, 191, 192, 193, 194, 625, 628, 633, 634, 637, 642, 643, 646, 647, 652, 1374, 1375, 1376, 1377, 1380, 1381, 1382, 1383, 1389, 1390, 1391, 1392, 1904, 1905, 1906, 2093, 2094, 2095, 2102
\labelenumi 1416
\labelenumii 1417
\labelitemi 1410
\labelitemii 1411
\labelitemiii 1412
\labelitemiv 1413
\land 3444, 3446
\LARGE 1405, 1710
\Large 1584
\large 1575, 1684, 1769, 2977
\leaders 3265
\left 2149, 2588
\LEFTarrow 3326
\leftarrow 671
\leftouterjoin 2916
\leftskip 3428, 3429, 3439
\LehramtInformatikAutorEmail 1920
\LehramtInformatikAutorName 1919
\LehramtInformatikGitBranch 464, 3061
\LehramtInformatikGithubCodeRepo 3060
\LehramtInformatikGithubDomain 461, 3057
\LehramtInformatikGithubRawDomain 3058
\LehramtInformatikGithubTexRepo 462, 3059
\LehramtInformatikRepository 17, 20, 23, 26, 29, 1451, 1662, 1940, 1941, 1942, 1943, 1944, 1945, 1946, 1947, 1948, 1949, 3056
\LehramtInformatikTitel 1915
\leq 2251, 2711, 2731
\let 1304, 1305, 1965, 2862, 2863, 2864, 2865, 2878, 2880, 2881, 2882, 2884, 2885, 2886, 2887, 2897, 2989, 3027, 3028, 3257
\li@chomsky@erklaerung@texte 574, 610
\li@EntwurfsCode 752, 803, 804, 805, 854, 855, 856, 857, 923, 924, 925, 926, 927, 928, 955, 956, 957, 958, 959, 960, 1026, 1196, 1197, 1198, 1199, 1239, 1240

\li@EntwurfsCodeAllgemein 751
\li@fussnote@text 2424, 2430, 2434, 2438, 2442
\li@GithubLink 3089, 3102, 3112, 3123
\li@mget . 1502, 1506, 1526
\li@minc 1505, 1527
\li@mset 1499, 1507, 1516, 1519
\li@numdiscs 1511, 1520, 1526
\li@Rmodell@Schrift 2927, 2936, 2946
\li@sequence . . 1512, 1525
\li@synthese@erklaerung@texte 3178, 3238
\liAbleitung 1329
liAdditum (environment) 2041
\LiADDITUMfalse 229
\LiADDITUMtrue . 229, 1547
liAHuelle (environment) 2519
\liAlphabet 1314
liAntwort (environment) 1997
\LiANTWORTfalse 233
\LiANTWORTtrue 233
\liAnweisung 1895
\liAssemblerCode . . . 3127
\liAssemblerDatei . . 3128
\liAttribut 2946
\liAttributHuelle 2510, 3190, 3195
\liAttributHuelleOhneMathe 2510, 2513, 2515, 2529, 2539, 2547
\liAttributMenge 2518, 2530, 2533, 2540, 2541, 2555, 2557
\liAufgabe 16
\liAufgabenMetadaten 47, 1569, 1676
\liAufgabenTitel 55
\liAusdruck 1351
\liAutomat 66
\liAutomatenKante 98
\liBandAlphabet 1315
\liBedingung 1896
\liBedingungDrei 2204, 2247, 2287
\liBedingungEins 2198, 2236, 2279
\liBedingungFalsch . 1898
\liBedingungWahr . . . 1897
\liBedingungZwei 2201, 2242, 2283
\liBeschriftung 1986
\liBindeAufgabeEin . 1659
\liChomskyErklaerung 574, 615
\liChomskyUeberErklaerung 613
\liChomskyUeberschrift 562, 614
\liCpmEreignis 623
\liCpmFruehErklaerung 694
\liCpmFruehI . . . 687, 707
\liCpmSpaetErklaerung 712
\liCpmSpaetI . . . 680, 725
\liCpmVon 663
\liCpmVonOhneMathe 663, 666, 668
\liCpmVonZu 655
\liCpmVonZuOhneMathe 655, 658, 660
\liCpmVorgang 640
\liCpmZu 671
\liCpmZuOhneMathe 671, 674, 676
liDiagramm (environment) 2120
liEinbettung (environment) 1996
\liEntwurfs 1177
\liEntwurfsAbstrakteFabrik 807
\liEntwurfsAbstrakteFabrikBeschreibung 755, 808
\liEntwurfsAbstrakteFabrikCode 802, 812
\liEntwurfsAbstrakteFabrikUml 760, 810
\liEntwurfsAdapter . . 859
\liEntwurfsAdapterAkteure 829, 861
\liEntwurfsAdapterCode 853, 862
\liEntwurfsAdapterUml 814, 860
\liEntwurfsBeobachter 930
\liEntwurfsBeobachterAkteure 888, 932
\liEntwurfsBeobachterCode 922, 933
\liEntwurfsBeobachterUml 864, 931
\liEntwurfsDekorierer 962
\liEntwurfsDekoriererAkteure 964
\liEntwurfsDekoriererCode 954, 965
\liEntwurfsDekoriererUml 935, 963
\liEntwurfsEinfacheFabrik 998
\liEntwurfsEinfacheFabrikAkteure 982, 1000
\liEntwurfsEinfacheFabrikUml 967, 999
\liEntwurfsEinzelstueck 1028
\liEntwurfsEinzelstueckAkteure 1017, 1033
\liEntwurfsEinzelstueckBeschreibung 1002, 1029
\liEntwurfsEinzelstueckCode 1025, 1035
\liEntwurfsEinzelstueckUml 1007, 1031
\liEntwurfsErbauer . 1085
\liEntwurfsErbauerAkteure 1057, 1087
\liEntwurfsErbauerUml 1037, 1086
\liEntwurfsFabrikmethode 1131, 1156
\liEntwurfsFabrikmethodeAkteure 1106, 1133
\liEntwurfsFabrikmethodeUml 1089, 1132
\liEntwurfsKompositum 1156
\liEntwurfsKompositumAkteure 1158
\liEntwurfsKompositumUml 1135, 1157
\liEntwurfsModellPraesentationSteuerung 1177
\liEntwurfsModellPraesentationSteuerungAk 1179
\liEntwurfsModellPraesentationSteuerungUm 1160, 1178
\liEntwurfsStellvertreter 1201
\liEntwurfsStellvertreterCode 1195, 1203
\liEntwurfsStellvertreterUml 1181, 1202
\liEntwurfsZustand . 1242
\liEntwurfsZustandAkteure 1220, 1244
\liEntwurfsZustandCode 1238, 1245
\liEntwurfsZustandUml 1205, 1243
\liEpsilon 1300
\liErAttribute 1258, 1272, 1274
\liErDatenbankName . 1277
\liErEntity 1256, 1260, 1262
\liErledigt 3257
\liErMpAttribute . . . 1271
\liErMpEntity 1259
\liErMpRelationship 1265
\liErRelationship 1257, 1266, 1268
\liExamensAufgabe 19
\liExamensAufgabeA . . . 28
\liExamensAufgabeTA . . 25
\liExamensAufgabeTTA . 22
liExkurs (environment) 2066
\LiEXKURSfalse 231
\LiEXKURStrue 231
\liFalsch 555
\liFlaci 1361
\liFremd 2926

\liFunktionaleAbhaengigkeit ... 2549, 2552, 2560
\liFunktionaleAbhaengigkeiten ... 2563
\liFussnote ... 2423, 2425
\liFussnoteDreiText ... 2437, 2455
\liFussnoteEinsText ... 2429, 2449
\liFussnoteLink ... 2136
\liFussnoten ... 2445
\liFussnoteUrl . 1175, 2132
\liFussnoteVierText ... 2441, 2458
\liFussnoteZweiText ... 2433, 2452
\liGeschweifteKlammern ... 1333, 2564, 3269
\liGrafikCCLizenz ... 1453, 2408
\liGrafikLogo ... 1458, 1693, 2384
\liGrafikLogoPfad ... 1450, 1455, 1460
\liGrammatik ... 1372
liGraphenFormat (environment) ... 1492
\liHanoi ... 1499
\liHaskellCode ... 3134
\liHaskellDatei ... 3135
\liInduktionAnfang . 3352
\liInduktionErklaerung ... 3351
\liInduktionMarkierung ... 3350
\liInduktionSchritt 3368
\liInduktionVoraussetzung ... 3360
liJavaAngabe (environment) ... 3075
\liJavaCode ... 3078
\liJavaDatei .. 753, 3096
\liJavaExamen ... 3116
\liJavaTestDatei ... 3106
liKasten (environment) 1420
\liKellerAutomat ... 107
\liKellerKante ... 145
\liKellerUebergang 140, 146
\liKontrollCode ... 1899
liKontrollflussgraph (environment) 1887
\liKontrollKnotenPfad ... 1901
\liKontrollTextzeileKnoten ... 1900, 1905
\liKurzeTabellenLinie 735
\liLadeAllePakete 239, 1647
\liLadePakete .. 59, 62, 235, 240, 561, 622, 1291, 1535, 1633, 1759, 2324, 2422, 2502, 2670, 3039, 3160
\liLatexCode ... 3088
\liLeereZelle ... 2460
liLernkartei (environment) ... 2110
\liLinksReduktion .. 2527
\liLinksReduktionInline ... 2536, 2544
\liLogoTextCCLizenz ... 1615, 1748, 2404
\liLogoTextProjekt 1612, 1745, 2380
\liMasterExkurs ... 2292
\liMasterFaelle 2231, 2299
\liMasterFallRechnung ... 2277
\liMasterVariablen 2208, 2294
\liMasterVariablenDeklaration ... 2254
\liMasterWolframLink 2302
\liMenge ... 76, 77, 79, 118, 119, 120, 124, 176, 177, 178, 182, 1292, 1341, 1380, 1381
\liMengeOhneMathe 1292, 1295, 1297
\liMetaBschlangaulSammlung ... 2325, 2389
\liMetaCCLink .. 2337, 2413
\liMetaEmailLink 2347, 2363
\liMetaHermineBschlangaulAndFriends ... 1696, 2328, 2393
\liMetaHilfMit 1621, 2354, 2365
\liMetaQuelltext 1623, 2365
\liMetaSetze 36, 48, 1570, 1677
\liMetaUeberDasProjekt ... 2331, 2400
\liMinimierungErklaerung ... 2482
\liMinispracheDatei 3131
\linespread ... 2934
\liNichtsZuTun ... 3258
\liO ... 2171, 2199
\liOmega ... 2161, 2205
\liOmegaOhneMathe 2161, 2166, 2168
\liONotationO ... 2590
\liOOhneMathe 2171, 2176, 2178
\liParagraphMitLinien ... 610, 695, 713, 2484, 3238, 3259, 3356, 3364, 3372
\liPetriErreichKnotenDrei ... 2664
\liPetriErreichTransition ... 2661
\liPetriSetzeSchluessel ... 2606
\liPetriTransitionsName ... 2653, 2665
\liPetriTransitionsNameOhneMathe ... 2653, 2656, 2658
\liPetriTransPfeile 2665
\liPolynomiellReduzierbar ... 1779
\liPotenzmenge 1301, 1305, 2676
\liPotenzmengeOhneMathe ... 1302, 1303, 1304
\liPrimaer ... 2925
\liProblemBeschreibung ... 1763
\liProblemClique ... 1786
\liProblemName 1762, 1769, 1781, 1783, 1796, 1807, 1808, 1816, 1817
\liProblemSat ... 1815
\liProblemSubsetSum ... 1806, 1815
\liProblemVertexCover ... 1786, 1794
\liProduktionen 1340, 1382
liProduktionsRegeln (environment) 1330
liProjektSprache (environment) ... 1995
\liPseudoUeberschrift ... 1972, 2052, 2055, 2470, 2480, 3353, 3361, 3369
\liPumpingKontextfrei ... 2722
\liPumpingRegulaer . 2701
liQuellen (environment) ... 2090
\liRechtsReduktionInline ... 2544
\liRekursionsGleichung ... 2195, 2258
\liRelation ... 2574
liRelationenSchemaFormat (environment) 2947
\liRelationMenge ... 2941
\liRichtig ... 554
liRmodell (environment) ... 2927
\liRundeKlammer ... 2148, 2152, 2162, 2172, 2186, 2587, 2591
\liSetzeExamenTeilaufgabeNr ... 1655
\liSetzeExamenThemaNr ... 1651
\liSortierMarkierung 2966
\liSortierPfeil ... 2960
\liSortierPfeilUnten 2963
\liSpaltenUmbruch .. 3035
\liSqlCode ... 3139

\listen@punkt .. 2090, 2102
\liStrich 1761
\liSyntheseErklaerung 3178, 3243
\liSyntheseUeberErklaerung 3241
\liSyntheseUeberschrift 3162, 3242
\liT 2181, 2196, 2210, 2274
\liTeilen 2507
\literatur 1928, 1952
\liTheta 2151, 2202, 2234, 2240, 2245
\liThetaOhneMathe 2151, 2156, 2158
\liTOhneMathe 2181, 2190, 2192
\liTuringKante 208
\liTuringLeerzeichen 164, 172
\liTuringMaschine ... 165
\liTuringUeberfuehrung 211
\liTuringUebergaenge 203, 209
\liTuringUebergangZelle 198
\liTypoUeberGROSS .. 3292
\liTypoUeberGross 3286, 3292
\liTypoUeberschrift 3280, 3289, 3295
\liUeberfuehrungsFunktion 1306
\liUeberfuehrungsFunktionOhneMathe ... 1306, 1309, 1311
liUebergangsTabelle (environment) 2468
\liUeberschriftDreiecksTabelle 2478
\liUmlLeserichtung . 3320
\liVertauschen 2954
\liWortInSprache 736
\liWortNichtInSprache 741
\liWpEquivalent 3423
\liWpErklaerung 3426
\liWpErklaerungVerzweigung 3441
\liWpKalkuel 3403
\liWpKalkuelOhneMathe 3403, 3408, 3410, 3442, 3444, 3446
\liZustandsBuchstabe 1316, 1325, 1327, 1345, 1347
\liZustandsBuchstabeGross ... 1317, 1326, 1328
\liZustandsmenge ... 1304
\liZustandsmengeNr 1318, 2685
\liZustandsmengeNrGross 1326
\liZustandsMengenSammlung 2672
\liZustandsMengenSammlungNr 2681
\liZustandsmengeOhneMathe 1304
\liZustandsname 1327
\liZustandsnameGross ... 1328, 2673, 2682
\liZustandsnameTiefgestellt 1343
\liZustandsPaar 2462
\liZustandsPaarVariablenName ... 2461, 2464, 2465
\llap 2914
\log 2199, 2202, 2205, 2234, 2240
\logo 2369, 2382, 2406
\loop 2747
\lor 3445
\ltimes 2310

M

\makeatletter 2986
\makeatother 2994
\marginpar 1261, 1267, 1273, 1970
\mathbb .. 1817, 2734, 3365
\mathbin . 2916, 2917, 2918
\mathcal 2172, 2591, 3216, 3221, 3223, 3224, 3225
\Mathe 3420
\MatheEnv 3413, 3421, 3424
\mathord 2317, 2318
\mdfsetup 1419, 2008, 2012, 2016, 2020
\mdseries 2028
\medskip .. 1577, 1594, 1771, 1979, 1989, 1991, 2100, 2128, 2396, 2932, 2938, 3267, 3271, 3414, 3418
\memph 1970
\mintinline 3081, 3088, 3127, 3134, 3139
\mkern ... 2916, 2917, 2918
\mlq 2315, 2317
\mrq 2315, 2318
\msg 483
\myList 3007, 3008, 3009, 3012
\myNodes 2996, 3011, 3017, 3021, 3023

N

\NeedsTeXFormat 1, 14, 32, 57, 225, 489, 550, 557, 619, 732, 747, 1248, 1284, 1397, 1435, 1444, 1465, 1494, 1532, 1630, 1755, 1827, 1910, 1926, 1932, 1954, 2144, 2307, 2320, 2419, 2498, 2581, 2601, 2667, 2692, 2697, 2741, 2907, 2920, 2949, 3030, 3037, 3044, 3049, 3141, 3156, 3247, 3251, 3300, 3342, 3379, 3399
\neg 3446
\negthinspace .. 2149, 2588
\newcounter ... 2761, 2762
\NewDocumentCommand . 67, 108, 145, 166, 208, 235, 624, 641, 680, 687, 736, 741, 1351, 1373, 1453, 1458, 1779, 1902, 1930, 2132, 2136, 2563, 2574, 2661, 3096, 3106, 3116, 3128, 3131, 3135, 3320
\NewDocumentEnvironment 1330, 1420, 1492, 1887, 1995, 1996, 1997, 2041, 2066, 2091, 2110, 2120, 2469, 2519, 2929, 2947
\newif 229, 231, 233
\newlength 3426
\newminted 3075
\node 637, 1895, 1900, 2787, 2792, 2797, 2803, 2809, 2815, 2966, 3011, 3337
\noexpand 2875, 2876, 2877, 2896, 3011
\noindent ... 435, 738, 743, 1558, 1574, 1582, 1596, 1620, 1976, 1988, 1992, 2024, 2081, 2083, 2098, 2114, 2116, 2124, 2296, 2448, 2451, 2454, 2457, 2942, 3260, 3272, 3432
\nolinkurl 3093
\normalsize ... 1407, 3082
\notin 744
\null 3265

O

\o 2590, 2595, 2597
\o@join 2913, 2916, 2917, 2918
\Omega 2162
\omega 2703, 2704, 2724, 2725
\or 2824

P

\pagestyle 1430, 1682
\par 434, 455, 614, 1362, 1370, 1553, 1557, 1560,

1577, 1592, 1603, 1685, 1689, 1693, 1695, 1696, 1703, 1705, 1710, 1714, 1718, 1975, 1981, 1987, 2082, 2106, 2115, 2391, 2394, 2866, 2888, 2903, 2944, 3242, 3266, 3270, 3278, 3431, 3437
\paragraph 1407
\parindent 2935
\path 99, 146, 209, 652
\pgfkeys . . 2617, 3323, 3324, 3325, 3326, 3327, 3330, 3333, 3335
\pgfmath@count 2988, 2990, 2992
\pgfmath@smuggleone 2993
\pgfmathdeclarefunction 2987
\pgfmathint 2988
\pgfmathparse 1506, 3005, 3010, 3013, 3027, 3028
\pgfmathresult 1507, 2988, 2989, 2991, 2993, 3006, 3014, 3027, 3028
\pgfutil@empty 2989
\pgfutil@loop 2990
\pgfutil@repeat 2993
\preceq 1782
\prime 1761
\printbibliography . 1952
\ProvidesPackage 2, 15, 33, 58, 226, 490, 551, 558, 620, 733, 748, 1249, 1285, 1398, 1436, 1445, 1466, 1495, 1533, 1631, 1756, 1828, 1911, 1927, 1933, 1955, 2145, 2308, 2321, 2420, 2499, 2582, 2602, 2668, 2693, 2698, 2742, 2908, 2921, 2950, 3031, 3038, 3045, 3050, 3142, 3157, 3248, 3252, 3301, 3343, 3380, 3400

Q

\QS@list 2868, 2879, 2883, 2890, 2896, 2901, 2904
\QS@select@equal 2843, 2847
\QS@select@greater 2844, 2848
\QS@select@smaller 2839, 2842, 2846
\QS@sort@a 2821, 2854, 2875, 2876
\QS@sort@b 2821, 2822
\QS@sort@c 2825, 2832
\QS@sort@d 2833, 2841
\QS@sort@empty . 2823, 2828
\QS@sort@single 2824, 2829
\QSinitialize 2746, 2858, 2893
\QSIr . 2829, 2835, 2843, 2863, 2877, 2882, 2885
\QSIrr 2864, 2877, 2878, 2886
\QSLr 2835, 2842, 2853, 2854, 2862, 2875, 2880, 2884
\QSpivotStep 2748, 2858, 2862, 2873
\QSr 2835
\QSRr 2844, 2865, 2876, 2881, 2887, 2896, 2897, 2898
\QSsortStep 2750, 2858, 2874, 2875
\quad 2426

R

\raggedright 2031
\raisebox 1900
\relax 1965, 2835, 2878, 2880, 2881, 2882, 2990, 2992
\renewcommand 1410, 1411, 1412, 1413, 1416, 1417, 1431, 1432, 1729, 1731, 1921, 1922, 2468
\repeat 2751
\RequirePackage 4, 60, 163, 227, 237, 486, 492, 493, 553, 621, 750, 751, 1251, 1253, 1254, 1290, 1399, 1400, 1402, 1404, 1409, 1418, 1425, 1433, 1437, 1438, 1448, 1449, 1467, 1468, 1469, 1497, 1498, 1546, 1643, 1648, 1649, 1735, 1760, 1829, 1929, 1934, 1935, 1951, 1958, 1959, 1960, 2147, 2311, 2312, 2584, 2585, 2586, 2603, 2605, 2695, 2755, 2757, 2909, 2910, 2911, 2924, 2952, 2983, 3034, 3040, 3047, 3052, 3063, 3064, 3144, 3249, 3256, 3304, 3305, 3307, 3308, 3346, 3347, 3348, 3381, 3401
\right 2149, 2588
\RIGHTarrow . . . 3322, 3327
\Rightarrow 739, 744
\rightarrow 212, 578, 583, 591, 595, 597, 598, 600, 655, 663, 2665, 3187, 3194, 3196, 3199, 3204, 3209, 3210, 3215
\rightouterjoin 2917
\rmfamily 565, 1526, 3165, 3282
\Roman 1415
\roman 1415, 1417
\romannumeral 2833
\rtimes 2310
\rule 1559, 2866, 2888, 2903, 2914

S

\sb 72, 82, 84, 113, 171, 590, 591, 595, 598, 599, 600, 682, 684, 689, 691, 1345, 1347, 2199, 2202, 2205, 2234, 2240, 2488, 2653, 2662, 3209, 3210, 3211, 3216, 3220, 3221, 3224, 3225, 3228, 3229, 3230
\scriptscriptstyle 655, 663, 671
\scriptsize 1364, 1840, 1847, 1853, 1915, 1916, 1919, 1920, 2028, 2412, 3351, 3404, 3434
\section . . . 52, 1644, 1653
\sectionbreak 1645
\seq . 1904, 1905, 1906, 2093, 2094, 2095, 2102
\setbox . . 2001, 2045, 2913
\setcounter 1408, 1646, 1734, 2867, 2889, 2903
\setganttlinklabel . . 1439, 1440, 1441, 1442
\setlength 1727, 1728, 1730, 1923, 2935, 3428, 3429, 3439
\setmainfont 1401
\setmainlanguage 487
\setminted 3069, 3070
\setminus 2548
\setul 2926
\shoveleft 2528
\shoveright 2532
\Sigma 69, 110, 168, 1314, 1315, 1375
\sigma 588, 590, 591
\SLASH 1971
\small 2123
\sort 3000
\sortList 2999, 3008

\square 555
\stepcounter 2787, 2792, 2797, 2800, 2802, 2806, 2808, 2812, 2814
\stichwoerter . . 1549, 1597
\str 367, 566, 575, 2005, 2677, 2686, 3166, 3179
\string 2530, 2540
\StrSubstitute . 3007, 3009
\strut 2256, 2260, 2264, 2268, 2272, 3035
\subsection 1657
\subseteq 3190, 3223, 3230
\subsubsection 1678

T

\tableofcontents 1732, 1966
\TeX 2366
\text 82, 84, 187, 2510, 3351, 3404
\textbf 702, 720, 1256, 1551, 1787, 1796, 1807, 1816, 1977, 1990, 2025, 2082, 2099, 2115, 2473
\textcolor 1899, 3080, 3350
\textit . . . 1096, 1138, 1139, 1140, 1141, 1564, 2125, 2518, 2578
\textsc 1762
\textsf 2099
\textstyle 2219, 2251
\texttt 1279, 1762, 1896, 1897, 1898, 1899, 3404
\textwidth 1559, 1923
\thematik 1562, 1591
\thepage 1429, 1918
\theparagraph 1407
\thesection 1644
\Theta 2152
\thinspace 3404
\tikz 1900
tikz: bbaum 26
tikz: li binaer baum 24
\tikzchildnode 510
\tikzparentnode 510
\tikzset 101, 148, 214, 495, 521, 1471, 1831, 2643, 2769, 2973, 3146, 3382
\tikzumlset 3310
\times 212
\tiny . 1262, 1268, 1274, 1619, 1899, 1970, 3091
\titel 1680, 1737
\titleformat 1405, 1407, 1644
\titlespacing 1406
\tl . 39, 68, 69, 70, 71, 72, 73, 76, 77, 78, 79, 80, 82, 84, 109, 110, 111, 112, 113, 114, 115, 118, 119, 120, 121, 122, 123, 124, 167, 168, 169, 170, 171, 172, 173, 176, 177, 178, 179, 180, 181, 182, 309, 313, 340, 344, 345, 346, 349, 354, 355, 356, 368, 377, 412, 415, 418, 426, 427, 428, 429, 440, 446, 449, 452, 468, 475, 625, 628, 633, 634, 642, 643, 646, 647, 1353, 1374, 1375, 1376, 1377, 1380, 1381, 1382, 1383, 1550, 1563, 1572, 1585, 1652, 1656, 1660, 1667, 1670, 2182
\tmp 2550
\TmpPlaceEight 2625
\TmpPlaceFive 2622
\TmpPlaceFour 2621
\TmpPlaceNine 2626
\TmpPlaceOne 2618
\TmpPlaceSeven 2624
\TmpPlaceSix 2623
\TmpPlaceTen 2627
\TmpPlaceThree 2620
\TmpPlaceTwo 2619
\TmpScale 2638
\TmpTransitionEight 2614, 2635
\TmpTransitionFive 2611, 2632
\TmpTransitionFour 2610, 2631
\TmpTransitionNine 2615, 2636
\TmpTransitionOne 2607, 2628
\TmpTransitionSeven 2613, 2634
\TmpTransitionSix 2612, 2633
\TmpTransitionTen 2616, 2637
\TmpTransitionThree 2609, 2630
\TmpTransitionTwo 2608, 2629
\TmpX 2639
\TmpY 2640
\today 1916
\ttfamily 2927

U

\ul 1257, 2925, 2926
\umlaggreg 1215
\umlassoc 1173
\umlclass 762, 766, 770, 817, 818, 819, 866, 871, 876, 879, 937, 938, 939, 944, 945, 974, 1009, 1040, 1041, 1044, 1095, 1098, 1137, 1143, 1144, 1162, 1163, 1164, 1185, 1186, 1187, 1207, 1208, 1209, 1210
\umldep 1103
\umlHVHaggreg 884, 950, 1153
\umlinherit 823, 874, 1048, 1093, 1101
\umlnote . . 825, 1050, 1217
\umlreal 821, 882
\umlsimpleclass 777, 778, 779, 783, 785, 786, 787, 816, 969, 970, 971, 1039, 1091, 1092, 1183
\umlstatic 976, 1010
\umluniaggreg 1046
\umluniassoc 797, 822, 1047, 1171, 1172, 1191, 1192
\umlVHuniassoc . . 798, 799
\umlVHVdep 791, 792, 794, 795, 978, 979
\umlVHVinherit 774, 775, 780, 781, 788, 789, 947, 948, 972, 973, 1151, 1152, 1189, 1190
\umlVHVreal 941, 942, 1212, 1213
\UParrow 3324
\url 469, 2133
\usemintedstyle 3066
\usetikzlibrary . . 61, 494, 1252, 1470, 1830, 2604, 2912, 2953, 2984, 3309, 3397

V

\value 2749
\varepsilon 567, 578, 579, 1300, 2199, 2205, 2237, 2248
\vbox 2001, 2045
\vfill 1609, 1691, 1698, 1722, 1743, 3035
\vrule 3261, 3265
\vspace . . . 1687, 1708, 1712, 1716, 1729, 2068, 2086, 3274, 3276

X

\xappto . . 3011, 3017, 3021
\xdef 1500
\xintApply 2837
\xintApplyUnbraced . . 2836, 2842, 2843, 2844
\xintCSVtoList 2896
\xintFor 2786, 2791, 2796, 2801, 2807, 2813, 2854

\xintifEq 2847
\xintifForLast 2803, 2809, 2815
\xintifGt 2848
\xintifLt 2846
\xintLength 2821
\xintnthelt 2833

Z

\ZB 2140
\zB 2139
\zustandsnamens@liste ... 1318, 1325, 1326