Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends*

September 12, 2021

Contents

Klassen **4**
aufgabe.cls . . . 5
basis.cls . . . 7
examen-scans.cls . . . 8
examen.cls . . . 10
haupt.cls . . . 13
theorie.cls . . . 14

Pakete **15**
abmessung.sty . . . 16
aufgaben-einbinden.sty . . . 17
aufgaben-metadaten.sty . . . 18
automaten.sty . . . 19
Endlicher Automat . . . 19
Kellerautomat . . . 21
Turingmaschine . . . 22
basis.sty . . . 25
IFs . . . 25
baum.sty . . . 31
Binärbaum . . . 32
AVL-Baum . . . 33
B-Baum . . . 34
checkbox.sty . . . 35
chomsky-normalform.sty . . . 36
Faulenzer . . . 36
TeX-Markup-Grundgerüst . . . 36
Konkretes TeX-Markup-Beispiel . . . 36
cpm.sty . . . 39
Faulenzer . . . 39
TeX-Markup-Beispiel: Graph . . . 39
TeX-Markup-Beispiel: Ergebnistabelle . . . 39
TeX-Markup-Beispiel: Nebenrechnungstabelle „Frühester Zeitpunkt“ . . . 39
TeX-Markup-Beispiel: Nebenrechnungstabelle „Spätester Zeitpunkt“ . . . 40
Hilfsmakros für Zeitpunkt-Tabelle . . . 41
cyk-algorithmus.sty . . . 43
Faulenzer . . . 43
TeX-Markup-Beispiel . . . 43
entwurfsmuster.sty . . . 44
Namensschema der Entwurfsmuster-Makros: . . . 44
Reihenfolge . . . 44

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

Abstrakte Fabrik (Abstract Factory) . . . 44
Adapter . . . 46
Beobachter (Observer) . . . 47
Dekorierer (Decorator) . . . 49
Einfache Fabrik (Simple Factory) . . . 50
Einzelstück (Singleton) . . . 51
Erbauer (Builder) . . . 52
Fabrikmethode (Factory Method) . . . 53
Kompositum (Composite) . . . 55
Modell-Präsentation-Steuerung (Model-View-Controller) . . . 55
Stellvertreter (Proxy) . . . 56
Zustand (State) . . . 57
er.sty . . . 59
Tex-Markup-Beispiel: Komplettes Diagramm . . . 59
Tex-Markup-Beispiel: EER Enhanced Entity-Relation-Modell nach Elmasri/Navante . . . 60
Faulenzer . . . 60
formale-sprachen.sty . . . 62
formatierung.sty . . . 65
Schriftarten / Typographie . . . 65
Farben . . . 65
Überschriften . . . 65
Listen . . . 65
Kasten . . . 65
Header . . . 65
Zeilenabstände . . . 65
gantt.sty . . . 67
grafik.sty . . . 68
graph.sty . . . 69
hanoi.sty . . . 71
komplexitaetstheorie.sty . . . 72
Faulenzer . . . 72
kontrollflussgraph.sty . . . 74
Faulenzer . . . 74
TeX-Markup-Beispiel . . . 74
TikZ: pin . . . 74
Umgebungen . . . 75
Makros . . . 76
kopfzeile.sty . . . 77
literatur-dummy.sty . . . 78
literatur.sty . . . 79
makros.sty . . . 80
Umgebungen, die Inhalte aus- und einblenden können . . . 81
master-theorem.sty . . . 85
Faulenzer . . . 85
mathe.sty . . . 89
meta.sty . . . 90
Einfache Makros (Low level) . . . 90
Zusammengesetzte Makros (High level) . . . 91
minimierung.sty . . . 93
normalformen.sty . . . 96
Faulenzer . . . 96
o-notation.sty . . . 99
Faulenzer . . . 99
TeX-Markup-Beispiel: Funktionsgraphen mit pgfplots . . . 99
petri.sty . . . 100
Faulenzer . . . 100
potenzmengen-konstruktion.sty . . . 102

pseudo.sty . . . 104
pumping-lemma.sty . . . 105
quicksort.sty . . . 106
relationale-algebra.sty . . . 109
rmodell.sty . . . 110
Faulenzer . . . 110
sortieren.sty . . . 111
spalten.sty . . . 113
sql.sty . . . 114
Latex-Markup-Beispiel . . . 114
struktogramm.sty . . . 115
syntax.sty . . . 116
Faulenzer . . . 116
syntaxbaum.sty . . . 119
synthese-algorithmus.sty . . . 120
Faulenzer . . . 120
TeX-Markup Grundgerüst . . . 120
TeX-Markup Linksreduktion . . . 120
TeX-Markup Rechtreduktion . . . 120
TeX-Markup Relationen formen . . . 121
tabelle.sty . . . 124
tex-dokumentation.sty . . . 125
typographie.sty . . . 126
uml.sty . . . 128
vollstaendige-induktion.sty . . . 130
Faulenzer . . . 130
wasserfall.sty . . . 132
wpkalkuel.sty . . . 133
Faulenzer . . . 133

Index **134**

Klassen

aufgabe.cls

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{bschlangaul-aufgabe}[2019/10/27 Minimale Klasse zu
Setzen einer Aufgabe]
```

```
\LoadClass{bschlangaul-basis}
```

Formatierung muss vor literatur sein, sonst Option clash

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

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

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

In Aufgaben wollen wir das Additum sehen.

```
\ADDITUMtrue
```

```
\ExplSyntaxOn
```

```
\cs_gset:Npn \stichwoerter_formatiert: {
  \tl_if_empty:NTF \g_stichwoerter_tl {} {
    \textbf{Stichwörter:} ~
    \g_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_thematik_tl {} {
    \textit{
      ( \g_thematik_tl )
    }
  }
}
```

`\bAufgabenMetadaten`

```
\def\bAufgabenMetadaten #1
{
  \bMetaSetze{#1}

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

  {
    \noindent
    \bfseries
```

```
    \Large
    \tl_if_empty:NTF \g_einzelpruefungs_nr_tl
    {
      \g_titel_tl
    }
    {
      \_gib_aufgaben_pfad_trenner:
    }
  }
  \hfill \thematik_formatiert:
  \par

  \medskip

  \noindent
  {\footnotesize\stichwoerter_formatiert:}

  \horizontale_linie:

  \setze_kopfzeile_oben_rechts:n { \gib_aufgaben_pfad_lang_sicher: }

  \bigskip

  \par
  % Keine Einrückung
  \@afterindentfalse
  \@afterheading
}
\setze_kopf_fusszeilen:nn {} {}
\AddToHook { begindocument }
{
  \repariere_kopfzeile_breite:
}
\AddToHook { enddocument }
{
  \vfill
  {
    \bLogoTextProjekt
    \bigskip

    \bLogoTextCCLizenz
    \bigskip

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

      \bMetaQuelltext
      \_gib_github_url_href:
    \end{spacing}
  }
}
\ExplSyntaxOff
```

basis.cls

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{bschlangaul-basis}[2021/09/12 Basis-Klasse, die nur als
Unterklasse benutzt werden soll.]
```

```
\LoadClass[a4paper,oneside,12pt]{book}
```

Damit wir Klassen-Optionen mit LaTeX-3 verwalten könnten.

```
\RequirePackage{l3keys2e}
```

```
\RequirePackage{bschlangaul-basis}
```

Lade Pakete die von allen Unterklassen benötigt werden.

```
\bLadePakete{
  kopfzeile
}
```

```
\ExplSyntaxOn
```

```
\keys_define:nn { klassen-setup }
{
  pakete .code:n = {
    \bLadePakete{#1}
  }
}
```

https://tex.stackexchange.com/questions/544374/passing-options-to-a-class-and-evaluate-using-latex3-interfaces

```
\ProcessKeysOptions { klassen-setup }
```

```
\ExplSyntaxOff
```

examen-scans.cls

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{bschlangaul-examen-scans}[2021/01/08 Zum
Zusammenfügen mehrerer Staatsexamen-Scans zu einer PDF-Datei.]
\LoadClass[a4paper,oneside]{book}
```

```
\LoadClass{bschlangaul-basis}
```

```
\RequirePackage{pdfpages}
```

Formatierung muss vor literatur sein, sonst Option clash

```
\bLadePakete{formatierung}
```

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

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

```
\ExplSyntaxOn
```

`\bPruefungsNummer`

```
\def\bPruefungsNummer#1{
  \str_set:Nn \g_pruefungs_nummer {#1}
}
```

`\bPruefungsTitel`

```
\def\bPruefungsTitel#1{
  \str_set:Nn \g_pruefungs_titel {#1}
}
```

```
\def\li@SansFett#1#2{
  {
    #1
    \bfseries
    \rmfamily
    #2
  }
}
```

`\bTrennSeite`

```
\def\bTrennSeite#1{
  \clearpage
  \strut
  \vfill
  \begin{center}
  {
    \li@SansFett{\LARGE}{\str_use:N \g_pruefungs_nummer} \\
    \vspace{2cm}
    \li@SansFett{\large}{\str_use:N \g_pruefungs_titel} \\
    \vspace{5cm}
    \li@SansFett{\Huge}{#1}
  }
  \end{center}
  \vfill
  \strut
  \clearpage
}
```

`\bTitelSeite`

```
\def\bTitelSeite#1{
  \clearpage
  \strut
  \vfill
  \begin{center}
  #1
  \end{center}
  \vfill
  \strut
```

```
  \clearpage
}
```

\bBindePdfEin

```
\def\bBindePdfEin#1{\includepdf[pages={1-}]{\LehramtInformatikRepository/#1}}

\AtBeginDocument{
  \thispagestyle{empty}
  \bTitelSeite{
    \li@SansFett{\Large}{Sammlung~aller~Staatsexamensaufgaben~der~Prüfungsnummer} \\

    \vspace{4cm}

    \li@SansFett{\Huge}{\str_use:N \g_pruefungs_nummer} \\

    \vspace{4cm}

    \li@SansFett{\LARGE}{\str_use:N \g_pruefungs_titel} \\
  }
}

\ExplSyntaxOff

```

examen.cls

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{bschlangaul-examen}[2021/06/10 Zum Einbinden von
mehreren Aufgaben zu einem Examen.]
```

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

```
\LoadClass[pakete={
  formatierung,
  literatur-dummy,
  makros,
  aufgaben-einbinden,
  aufgaben-metadaten,
  abmessung,
  typographie,
  grafik,
  meta
}]{bschlangaul-basis}
```

Formatierung für die Überschriften setzen.

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

```
\bLadeAllePakete
```

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

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

```
\ExplSyntaxOn
```

`\bSetzeExamenThemaNr`

```
\def\bSetzeExamenThemaNr#1{
  \tl_gset:Nn \g_thema_nr_tl { #1 }
  \section{Thema~Nr.~#1}
}
```

`\bSetzeExamenTeilaufgabeNr`

```
\def\bSetzeExamenTeilaufgabeNr#1{
  \tl_gset:Nn \g_teilaufgabe_nr_tl { #1 }
  \subsection{Teilaufgabe~Nr.~#1}
}
```

`\bBindeAufgabeEin`

```
\def\bBindeAufgabeEin#1{
  \tl_gset:Nn \g_aufgabe_nr_tl { #1 }
  \input{
    \LehramtInformatikRepository /
    Staatsexamen /
    \g_einzelpruefungs_nr_tl /
    \g_jahr_tl /
    \g_monat_tl /
    \tl_if_empty:NTF \g_thema_nr_tl {} {
      Thema - \g_thema_nr_tl /
    }
    \tl_if_empty:NTF \g_teilaufgabe_nr_tl {} {
      Teilaufgabe - \g_teilaufgabe_nr_tl /
    }
    Aufgabe - \g_aufgabe_nr_tl .tex
  }
}
```

`\bAufgabenMetadaten` Das Metadaten-Makro überschreiben

```
\def\bAufgabenMetadaten#1{
  \bMetaSetze{#1}
```

```
  \subsubsection{\_gib_aufgaben_titel:}
}
\cs_new:Npn \titel_seite:
{
  \titel_seite:nn
  {
    {
      \bfseries\Huge

      \g_einzelpruefungs_nr_tl
      \par

      \g_jahreszeit_tl
      \par

      \g_jahr_tl
      \par
    }
  }
  {
    \g_examen_fach_tl
    \par

    \vspace{0.5cm}

    Aufgabenstellungen~mit~Lösungsvorschlägen
    \par
  }

}
\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

  \bLogoTextProjekt
  \bigskip

  \bLogoTextCCLizenz
  \bigskip

  \clearpage
}
\ExplSyntaxOff
```

306

haupt.cls

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{bschlangaul-haupt}[2019/10/27 Klasse zum
Setzen der Haupt-Datei, Bschlangaul-Sammlung.tex d.h. des wichtigsten
Dokuments, das alles zusammenfasst.]
```

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

```
\LoadClass[pakete={
  formatierung,
  literatur-dummy,
  makros,
  aufgaben-einbinden,
  aufgaben-metadaten,
  abmessung,
  typographie,
  grafik,
  meta
}]{bschlangaul-basis}
\bLadeAllePakete
```

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

```
\RequirePackage[ngerman]{babel}
\ExplSyntaxOn
\setze_kopf_fusszeilen:nn {} {}
\AddToHook { begindocument }
{
  \repariere_kopfzeile_breite:
  \titel_seite:nn
  {
    {
      \Huge
      Die~komplette~Sammlung
    }

  }
  {
    Alle~Aufgaben
  }
}
\ExplSyntaxOff

```

theorie.cls

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{bschlangaul-theorie}[2021/09/12 Klasse zum
Setzen von Theorie-Zusammenfassungen]
```

```
\LoadClass{bschlangaul-basis}
```

Paket „formatierung“ muss vor „literatur“ sein, sonst Option clash

```
\bLadePakete{
  formatierung,
  literatur,
  makros,
  aufgaben-metadaten
}
```

```
\RequirePackage{standalone}
```

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

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

Pakete

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,
}

```

aufgaben-einbinden.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-aufgaben-einbinden}[2020/06/13
Binde Aufgaben in ein größeres Dokument ein.]
```

Mit Hilfe des `standalone`-Pakets können eigenständige Dokumente eingebunden werden, die auch uneingebunden kompiliert werden können, wie zum Beispiel Aufgaben

```
\RequirePackage{standalone}
```

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

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

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

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

`\bExamensAufgabeTTA`

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

`\bExamensAufgabeTA`

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

`\bExamensAufgabeA`

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

aufgaben-metadaten.sty

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

```
\ExplSyntaxOn
```

`\bMetaSetze` Setze die Metadaten einer Aufgabe.

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

  \tl_clear:N \g_teilaufgabe_nr_tl

  \keys_set:nn { aufgabenmetadaten } {
    #1
  }

  \_setze_relativen_pfad:
}
```

`\bAufgabenMetadaten` 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`.

```
\bAufgabenMetadaten{
  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\bAufgabenMetadaten#1{
  \bMetaSetze{#1}

  \_gib_examen_titel: {}

  \section{\_gib_aufgaben_titel:}
}
```

```
\ExplSyntaxOff
```

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

```
\def\bAufgabenTitel#1{}
```

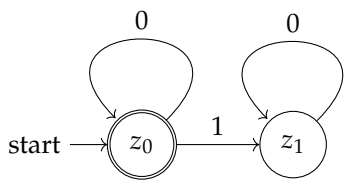
automaten.sty

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

Endlicher Automat

```
\bLadePakete{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}
\bLadePakete{mathe}
\directlua{
  automaten = require('bschlangaul-automaten')
}
```

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

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

```
\ExplSyntaxOn
\NewDocumentCommand {\bAutomat} { 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 {\bMenge{##1}}},
    alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\bMenge{##1}}},
    delta .code:n = {\tl_set:Nn \l_delta_tl {##1}},
    ende .code:n = {\tl_set:Nn \l_ende_tl {\bMenge{##1}}},
```

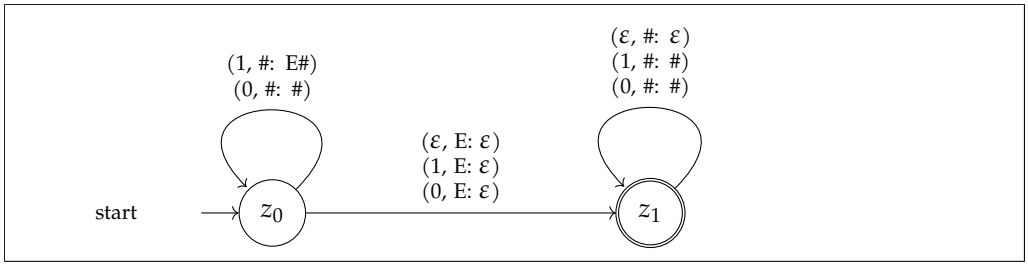
```
    start .code:n = {\tl_set:Nn \l_start_tl {##1}},
    dea .value_forbidden:n = true,
    dea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{DEA}}}},
    nea .value_forbidden:n = true,
    nea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{NEA}}}},
  }

  \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
```

`\bAutomatenKante` **Makro-Faulenzer:** `\let\k=\liAutomatenKante`

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

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

Kellerautomat

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

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

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

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



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

```
\bKellerAutomat{
  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 {\bKellerAutomat} { 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 {\bMenge{##1}}},
    alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\bMenge{##1}}},
    kelleralphabet .code:n = {\tl_set:Nn \l_kelleralphabet_tl {\bMenge{##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 {\bMenge{##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
```

\bKellerUebergang **Makro-Faulenzer:** `\let\u=\liKellerUebergang`

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

(a, #: A)
(b, #: ε)

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

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

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

Makro-Faulenzer: `\let\k=\liKellerKante`

```
\NewDocumentCommand{\bKellerKante} { O{above} m m m  } {
  \path (#2) edge[#1] node{\bKellerUebergang{#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
      }
    }
  }
}
```

Turingmaschine

```
\RequirePackage{amssymb}
```

\bTuringLeerzeichen □

```
\def\bTuringLeerzeichen{\Box}
```

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

```
\bTuringMaschine{
  zustaende={z_0, z_1, z_2},
  alphabet={a, b, c},
  bandalphabet={\bTuringLeerzeichen, 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 {\bTuringMaschine} { 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 {\bTuringLeerzeichen}
  \tl_set:Nn \l_ende_tl {E}

  \keys_define:nn { kellerautomat } {
    zustaende .code:n = {\tl_set:Nn \l_zustaende_tl {\bMenge{##1}}},
    alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\bMenge{##1}}},
    bandalphabet .code:n = {\tl_set:Nn \l_bandalphabet_tl {\bMenge{##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 {\bMenge{##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
```

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

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

`\bTuringUebergaenge` Nur in den TikZ-Grafiken zu verwenden. Setzt Zeilenumbrüche ans Ende. Nicht für die Tabelle geeignet.
Makro-Faulenzer: `\let\t=\liTuringUebergaenge`
$(z_1\colon \square, \mathrm{L})$
$(\square\colon \square, \mathrm{R})$

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

\bTuringKante

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

Makro-Faulenzer: \let\t=\liTuringKante

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

\bTuringUeberfuehrung

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

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

basis.sty

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

```
\ExplSyntaxOn
```

IFs

Wir weichen von dem üblichen Namensschema ab und schreiben das Wort komplett in Großbuchstaben, damit die `if`-Befehle schöner lesbar sind, z. B. `\ifADDITUM`, `\ADDITUMtrue` und `\ADDITUMfalse`.

`\ifADDITUM`
`\ADDITUMtrue`
`\ADDITUMfalse`

```
\newif\ifADDITUM
\ADDITUMfalse
```

`\ifEXKURS`
`\EXKURStrue`
`\EXKURSfalse`

```
\newif\ifEXKURS
\EXKURStrue
```

`\ifANTWORT`
`\ANTWORTtrue`
`\ANTWORTfalse`

```
\newif\ifANTWORT
\ANTWORTtrue
```

`\bLadePakete`

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

`\bLadeAllePakete`

```
\def\bLadeAllePakete{
  \bLadePakete{
    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_schluessel_clist
\clist_set:Nn \g_schluessel_clist {
  titel,
  thematik,
  stichwoerter,
  zitat_schluessel,
  zitat_beschreibung,
  %
  bearbeitungs_stand,
  korrektheit,
  %
  relativer_pfad,
  identische_aufgabe,
  %
  einzelpruefungs_nr,
  examen_fach,
  jahr,
  monat,
  jahreszeit,
  thema_nr,
  teilaufgabe_nr,
  aufgabe_nr,
}
```

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

```
\clist_map_inline:Nn \g_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_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_titel_tl,
  Thematik            .tl_gset:N = \g_thematik_tl,
  Stichwoerter        .tl_gset:N = \g_stichwoerter_tl,
  ZitatSchluessel     .tl_gset:N = \g_zitat_beschreibung_tl,
  ZitatBeschreibung   .tl_gset:N = \g_zitat_schluessel_tl,
```

```
  %
  BearbeitungsStand   .tl_gset:N = \g_bearbeitungs_stand_tl,
  Korrektheit         .tl_gset:N = \g_korrektheit_tl,
  %
  RelativerPfad       .tl_gset:N = \g_relativer_pfad_tl,
  IdentischeAufgabe   .tl_gset:N = \g_identische_aufgabe_tl,
  %
  ExamenNummer        .tl_gset:N = \g_einzelpruefungs_nr_tl,
  ExamenFach          .tl_gset:N = \g_examen_fach_tl,
  ExamenJahr          .tl_gset:N = \g_jahr_tl,
  ExamenMonat         .tl_gset:N = \g_monat_tl,
  ExamenJahreszeit    .tl_gset:N = \g_jahreszeit_tl,
  ExamenThemaNr       .tl_gset:N = \g_thema_nr_tl,
  ExamenTeilaufgabeNr .tl_gset:N = \g_teilaufgabe_nr_tl,
  ExamenAufgabeNr     .tl_gset:N = \g_aufgabe_nr_tl,
}
\cs_gset:Npn \_setze_relativen_pfad: {
  \tl_if_empty:NTF \g_relativer_pfad_tl
  {
    \bool_if:nTF
    {
      ! \tl_if_empty_p:N \g_einzelpruefungs_nr_tl &&
      ! \tl_if_empty_p:N \g_jahr_tl &&
      ! \tl_if_empty_p:N \g_monat_tl
    }
    {
      \tl_gset:Nn \g_relativer_pfad_tl {
        Staatsexamen /
        \g_einzelpruefungs_nr_tl /
        \g_jahr_tl /
        \g_monat_tl /
        \tl_if_empty:NTF \g_thema_nr_tl {} { Thema - \g_thema_nr_tl / }
        \tl_if_empty:NTF \g_teilaufgabe_nr_tl {} { Teilaufgabe - \g_teilaufgabe_nr_tl / }
        \tl_if_empty:NTF \g_aufgabe_nr_tl {} { Aufgabe - \g_aufgabe_nr_tl .tex }
      }
    }
    {}
  }
  {}
}
\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: {
  \g_einzelpruefungs_nr_tl

  \_trenner:

  \g_jahr_tl

  \_trenner:

  \_gib_jahreszeit_durch_monat: \g_monat_tl
}
```

Thema 1 / Teilaufgabe 2 / Aufgabe 3

```
\cs_gset:Npn \_gib_aufgaben_pfad_trenner: {
  \tl_if_empty:NTF \g_thema_nr_tl {} {
    Thema ~ \g_thema_nr_tl \_trenner:
  }
  \tl_if_empty:NTF \g_teilaufgabe_nr_tl {} {
    Teilaufgabe ~ \g_teilaufgabe_nr_tl \_trenner:
  }
  \tl_if_empty:NTF \g_aufgabe_nr_tl {} {
    Aufgabe ~ \g_aufgabe_nr_tl
  }
}
```

Gib den langen Titelpfad einer Aufgabe. Ist die Aufgabe keine Examensaufgabe, wird auch eine Titel zurückgegeben. Für die Kopfzeile gedacht.

```
\cs_new:Npn \gib_aufgaben_pfad_lang_sicher: {
  \tl_if_empty:NTF \g_einzelpruefungs_nr_tl
  {
    \g_titel_tl
  }
  {
    \_gib_einzelpruefung_trenner:
    \_trenner:
    \_gib_aufgaben_pfad_trenner:
  }
}
```

```
\cs_gset:Npn \_gib_examen_titel:
{
  \cs_set:Nn \_trenner: { \, / \, }
  \bool_if:nTF
  {
    ! \tl_if_empty_p:N \g_einzelpruefungs_nr_tl &&
    ! \tl_if_empty_p:N \g_jahr_tl &&
    ! \tl_if_empty_p:N \g_monat_tl &&
    ! \tl_if_empty_p:N \g_aufgabe_nr_tl
```

```
  }
  {
    {
      \footnotesize
      \par
      \noindent
      Staatsexamen ~
      \g_einzelpruefungs_nr_tl \_trenner:
      \g_jahr_tl \_trenner:

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

      \tl_if_empty:NTF \g_thema_nr_tl {} {
        Thema ~ Nr. ~ \g_thema_nr_tl \_trenner:
      }
      \tl_if_empty:NTF \g_teilaufgabe_nr_tl {} {
        Teilaufgabe ~ Nr. ~ \g_teilaufgabe_nr_tl \_trenner:
      }
      \tl_if_empty:NTF \g_aufgabe_nr_tl {} {
        Aufgabe ~ Nr. ~ \g_aufgabe_nr_tl
      }
      \par
      \bigskip
    }
  }
}
\cs_new:Npn \_gib_github_url: {
  \LehramtInformatikGithubDomain /
  \LehramtInformatikGithubTexRepo /
  blob /
  \LehramtInformatikGitBranch /
  \g_relativer_pfad_tl
}
\cs_new:Npn \_gib_github_url_href: {
  \tl_if_empty:NTF \g_relativer_pfad_tl {} {
    \url{ \_gib_github_url: }
  }
}
\cs_new:Npn \_gib_aufgaben_titel: {
  \g_titel_tl

  \tl_if_empty:NTF \g_thematik_tl
  {}
  {
   \, ~ [
    \g_thematik_tl
   ]
  }
}
\cs_new:Npn \titel_seite:nn #1 #2
{
  \pagestyle{empty}

  \strut
  \vspace{1cm}

  \begin{center}
    \bfseries
```

```
    \Large
    #1
    \par
  \end{center}

  \vfill

  \begin{center}
    \large
    #2
    \par
  \end{center}

  \vfill

  \begin{center}
    \bGrafikLogo[width=8cm]
    \par

    \vspace{4cm}

    {
      \bfseries
      \Large
      \bMetaBschlangaulSammlung
    }
    \par

    \medskip

    {
      \large
      \bMetaHermineFriends
    }
    \par
  \end{center}

  \vspace{2cm}

  \strut

  \clearpage
}
```

```
\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}
```

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}
```

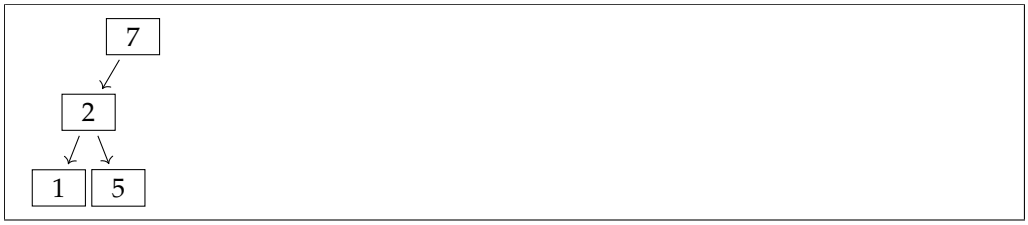
Für b baum

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

Binärbaum

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

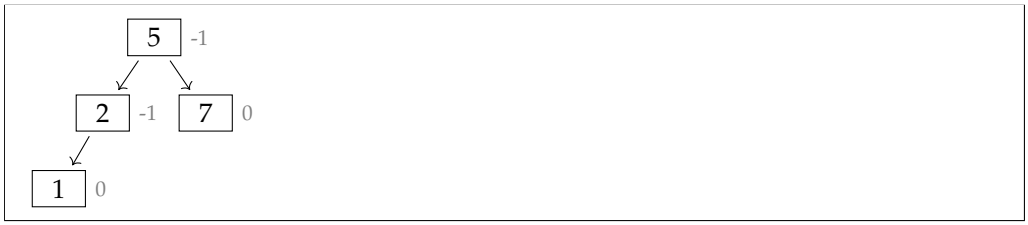
```
\begin{tikzpicture}[li binaer baum]
\Tree
[.7
  [.2
    [.1 ]
    [.5 ]
  ]
  \edge[blank]; \node[blank]{};
]
\end{tikzpicture}
```




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

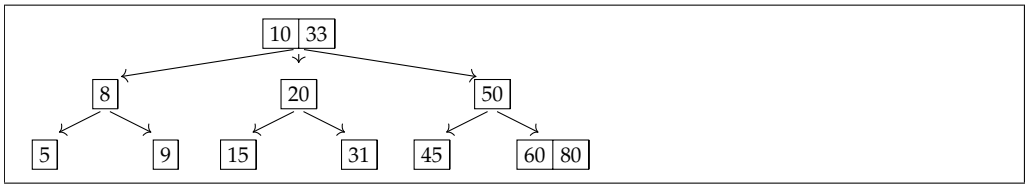
AVL-Baum

```
\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}
```



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,
    },
  }
}
```

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}
```

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

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

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

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

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
```

```
\bLadePakete{typographie}
```

Faulenzer

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

TeX-Markup-Grundgerüst

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

Konkretes TeX-Markup-Beispiel

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

\bNichtsZuTun

\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}
```

\bChomskyUeberschrift **Makro-Faulenzer:** `\let\schritt=\liChomskyUeberschrift`

```
\def\bChomskyUeberschrift#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}
    }
  }
}
```

\bChomskyErklaerung **Makro-Faulenzer:** `\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\bChomskyErklaerung#1{
  {
    \itshape
    \footnotesize
    \bParagraphMitLinien{\li@chomsky@erklaerung@texte{#1}}
  }
}
```

\bChomskyUeberErklaerung **Makro-Faulenzer:** `\let\schrittE=\liChomskyUeberErklaerung`

```
\def\bChomskyUeberErklaerung#1{
  \bChomskyUeberschrift{#1}\par
  \bChomskyErklaerung{#1}
}

\ExplSyntaxOff
```

cpm.sty

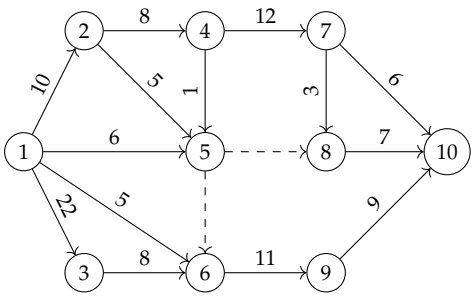
```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-cpm}[2020/09/03]
\RequirePackage{tikz}
```

```
\bLadePakete{mathe,typographie}
```

Faulenzer

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

TeX-Markup-Beispiel: Graph




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

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

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

TeX-Markup-Beispiel: Ergebnistabelle

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

TeX-Markup-Beispiel: Nebenrechnungstabelle „Frühester Zeitpunkt“

```
\bCpmFruehErklaerung
\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}
```

TeX-Markup-Beispiel: Nebenrechnungstabelle „Spätester Zeitpunkt“

```
\bCpmSpaetErklaerung
% 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}
```

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

```
\ExplSyntaxOn
\NewDocumentCommand { \bCpmEreignis } { 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
```

\bCpmVorgang `liCpmVORGANG\((.*)>(.*)\)\{(.*)\}`

```
\ExplSyntaxOn
\NewDocumentCommand { \bCpmVorgang } { 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
```

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}
```

`\bCpmVonZu` **Makro-Faulenzer:** `\let\vz=\liCpmVonZu`
`\bCpmVonZu{1}(2-3)`: $1_{(2\to3)}$

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

`\bCpmVon` **Makro-Faulenzer:** `\let\v=\liCpmVon`
`\bCpmVon{1}(2)`: $1_{(\to2)}$

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

`\bCpmZu` **Makro-Faulenzer:** `\let\z=\liCpmZu`
`\bCpmZu{1}(2)`: $1_{(\leftarrow2)}$

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

```
\ExplSyntaxOn
```

`\bCpmSpaetI` Spätester Zeitpunkt, zu dem Ereignis *i* eintreten kann
Makro-Faulenzer: `\let\SZ=\liCpmSpaetI`

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

`\bCpmFruehI` Frühester Zeitpunkt, zu dem Ereignis *i* eintreten kann.
Makro-Faulenzer: `\let\FZ=\liCpmFruehI`

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

```
  \fi
}
```

\bCpmFruehErklaerung

— 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\bCpmFruehErklaerung{
  \bParagraphMitLinien{
    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~$i$;~\,

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

\bCpmSpaetErklaerung

— 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\bCpmSpaetErklaerung{
  \bParagraphMitLinien{
    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~$i$;~\,

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

\ExplSyntaxOff
```

cyk-algorithmus.sty

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

Faulenzer

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

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}
\bWortInSprache{acbcab}
```

\bKurzeTabellenLinie

Makro-Faulenzer: `\let\l=\liKurzeTabellenLinie`

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

\bWortInSprache

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

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

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

\bWortNichtInSprache

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

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

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

entwurfsmuster.sty

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

Namensschema der Entwurfsmuster-Makros:

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

Reihenfolge

(a) `Beschreibung` Kurze Beschreibung des Entwurfsmusters, z. B `\bEntwurfsEinzelstueckBeschreib`

(b) `Uml`: Uml-Klassendiagramm, z. B `\bEntwurfsEinzelstueckUml`

(c) `Akteure`: Akteure, beteiligte Klassen, z. B `\bEntwurfsEinzelstueckAkteure`

(d) `Code`: Allgemeines Code-Beispiel, z. B `\bEntwurfsEinzelstueckCode`

(e) `ohne`: Ohne Suffix, Bündelung der einzelnen Makros eines Entwurfsmusters `\bEntwurfsEinzelstueckAkteure`

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

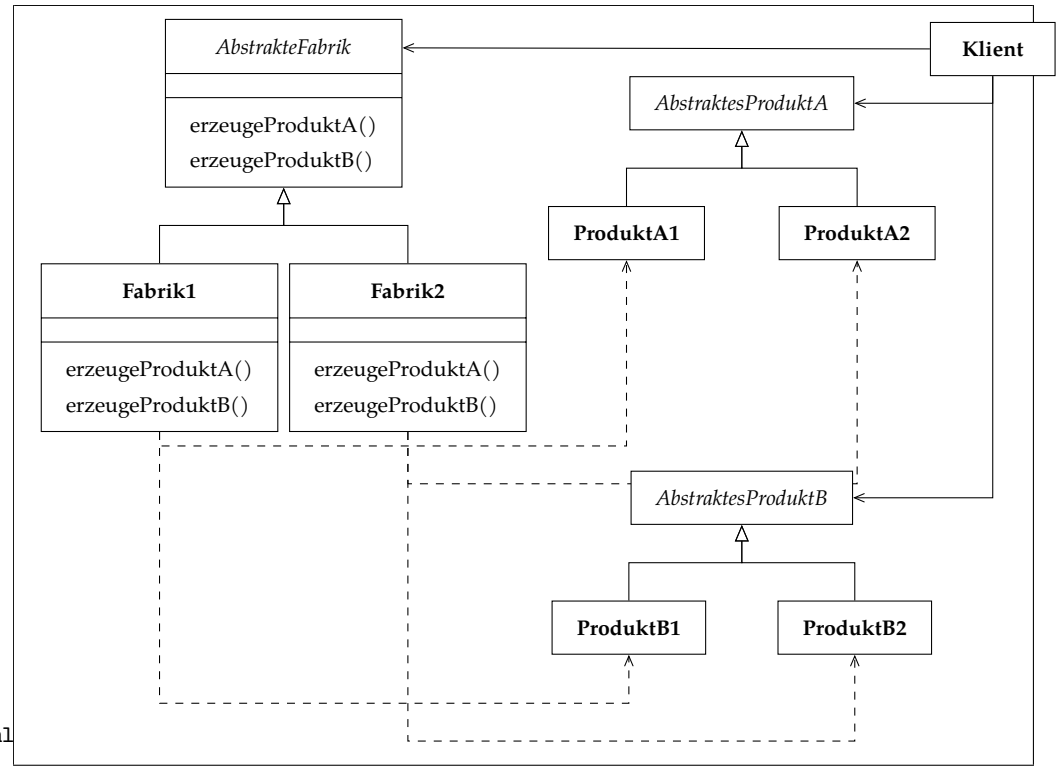
`\li@EntwurfsCodeAllgemein` Allgemeine Code-Beispiele zu den UML-Diagrammen und Stellvertretern

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

Abstrakte Fabrik (Abstract Factory)

`sAbstrakteFabrikBeschreibung`

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



\bEntwurfsAbstrakteFabrikUml

```
\def\bEntwurfsAbstrakteFabrikUml{
 \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}
}
```

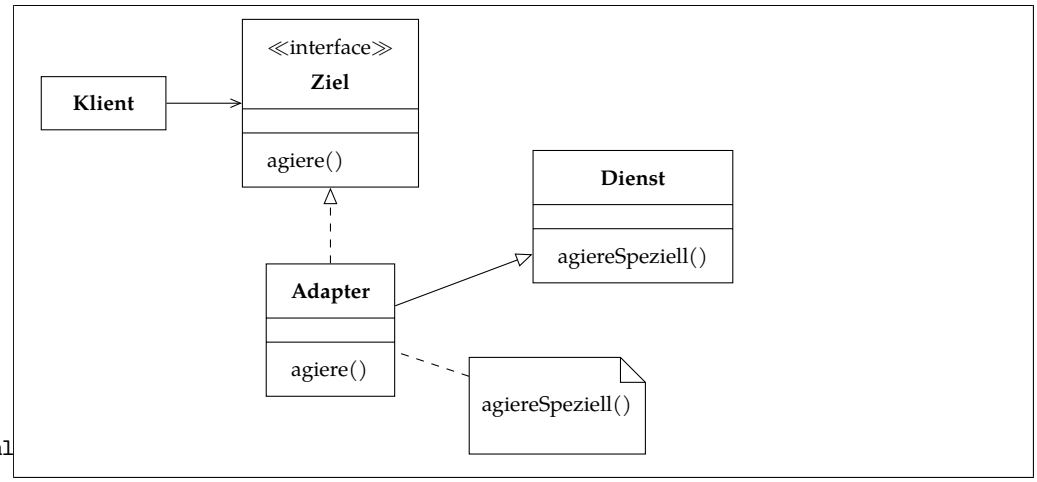
bEntwurfsAbstrakteFabrikCode

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

\bEntwurfsAbstrakteFabrik

```
\def\bEntwurfsAbstrakteFabrik{
  \bEntwurfsAbstrakteFabrikBeschreibung

  \bEntwurfsAbstrakteFabrikUml

  \bEntwurfsAbstrakteFabrikCode
}
```

Adapter



\bEntwurfsAdapterUml

```
\def\bEntwurfsAdapterUml{
  \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}
}
```

\bEntwurfsAdapterAkteure

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\bEntwurfsAdapterAkteure{
  \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}
}
```

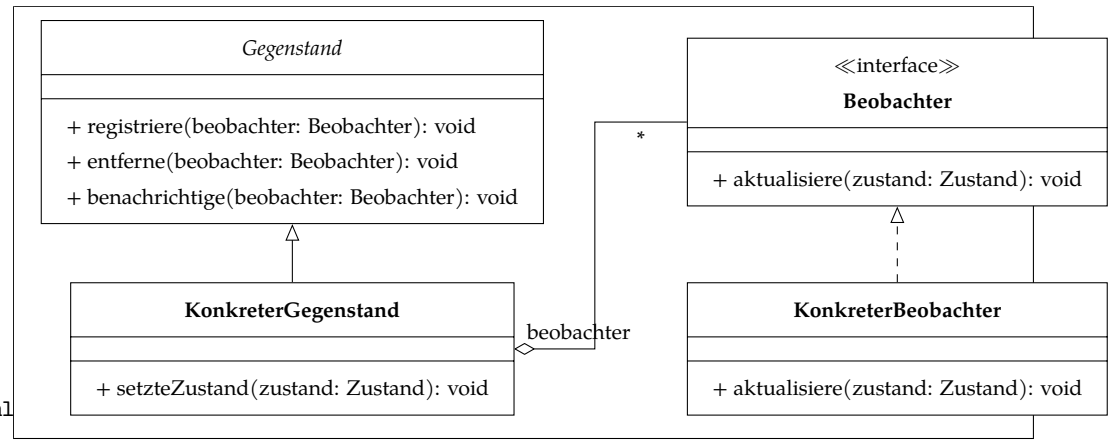
\bEntwurfsAdapterCode

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

\bEntwurfsAdapter

```
\def\bEntwurfsAdapter{
  \bEntwurfsAdapterUml
  \bEntwurfsAdapterAkteure
  \bEntwurfsAdapterCode
}
```

Beobachter (Observer)



\bEntwurfsBeobachterUml

```
\def\bEntwurfsBeobachterUml{
```

```
  \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}
}
```

\bEntwurfsBeobachterAkteure

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\bEntwurfsBeobachterAkteure{
  \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}
}
```

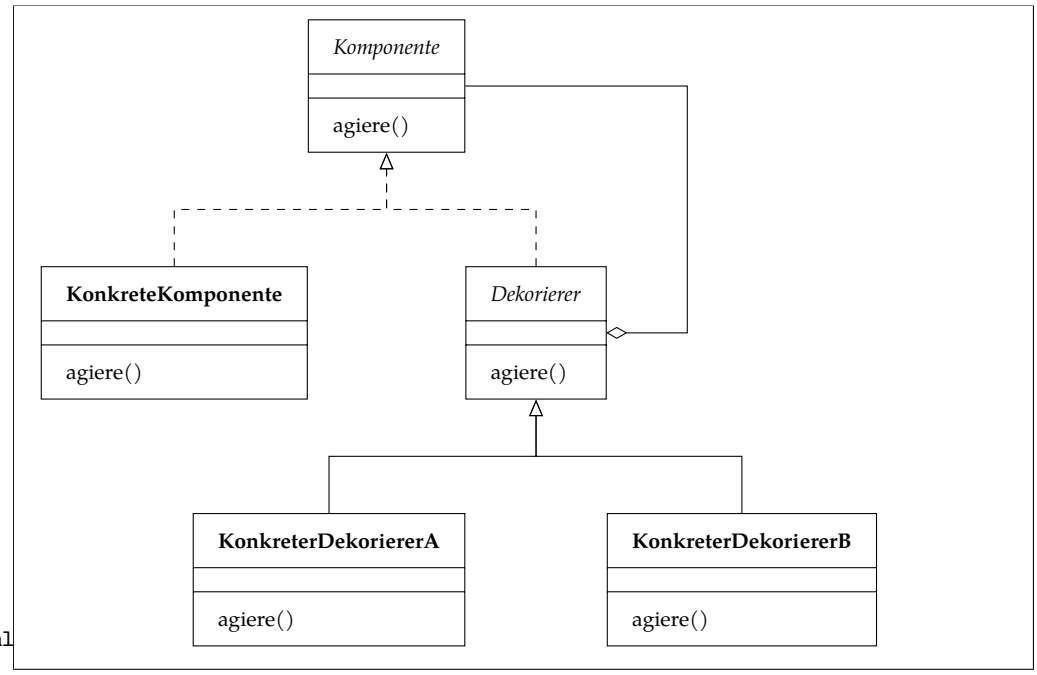
\bEntwurfsBeobachterCode

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

\bEntwurfsBeobachter

```
\def\bEntwurfsBeobachter{
  \bEntwurfsBeobachterUml
  \bEntwurfsBeobachterAkteure
  \bEntwurfsBeobachterCode
}
```

Dekorierer (Decorator)



\bEntwurfsDekoriererUml

```
\def\bEntwurfsDekoriererUml{
  \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}
}
```

\bEntwurfsDekoriererCode

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

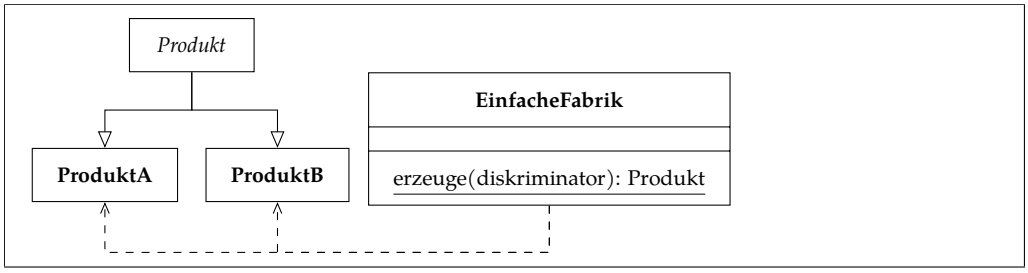
\bEntwurfsDekorierer

```
\def\bEntwurfsDekorierer{
  \bEntwurfsDekoriererUml
  \bEntwurfsDekoriererAkteure
  \bEntwurfsDekoriererCode
}
```

Einfache Fabrik (Simple Factory)

\bEntwurfsEinfacheFabrikUml

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




```
\def\bEntwurfsEinfacheFabrikUml{
  \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\bEntwurfsEinfacheFabrikAkteure{
  \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}
}
```

\bEntwurfsEinfacheFabrik

```
\def\bEntwurfsEinfacheFabrik{
  \bEntwurfsEinfacheFabrikUml
  \bEntwurfsEinfacheFabrikAkteure
}
```

Einzelstück (Singleton)

urfsEinzelstueckBeschreibung

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

```

\bEntwurfsEinzelstueckUml

Quelle nach der deutschen Wikipedia

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

bEntwurfsEinzelstueckAkteure

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\bEntwurfsEinzelstueckAkteure{
  \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}
}
```

\bEntwurfsEinzelstueckCode

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

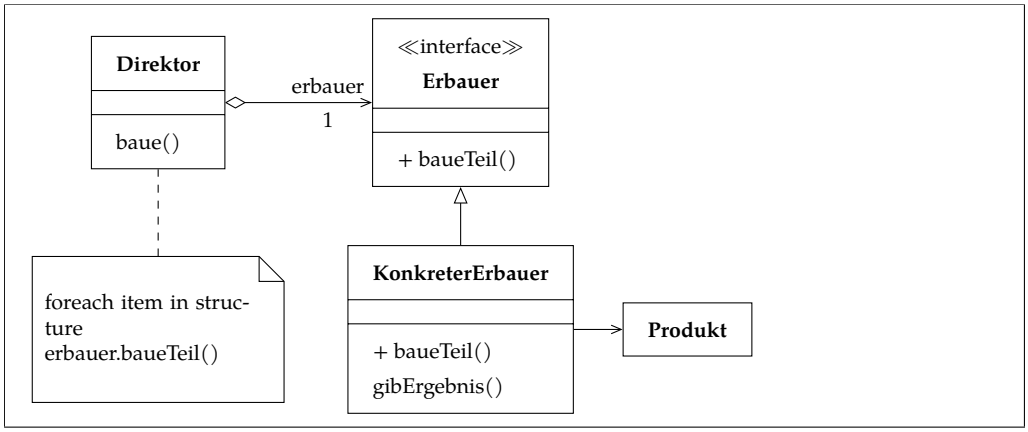
\bEntwurfsEinzelstueck

```
\def\bEntwurfsEinzelstueck{
  \bEntwurfsEinzelstueckBeschreibung

  \bEntwurfsEinzelstueckUml

  \bEntwurfsEinzelstueckAkteure

  \bEntwurfsEinzelstueckCode
}
```

Erbauer (Builder)

\bEntwurfsErbauerUml

Quelle nach der deutschen Wikipedia




```
\def\bEntwurfsErbauerUml{
  \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}
}
```

\bEntwurfsErbauerAkteure

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\bEntwurfsErbauerAkteure{
  \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}
}
```

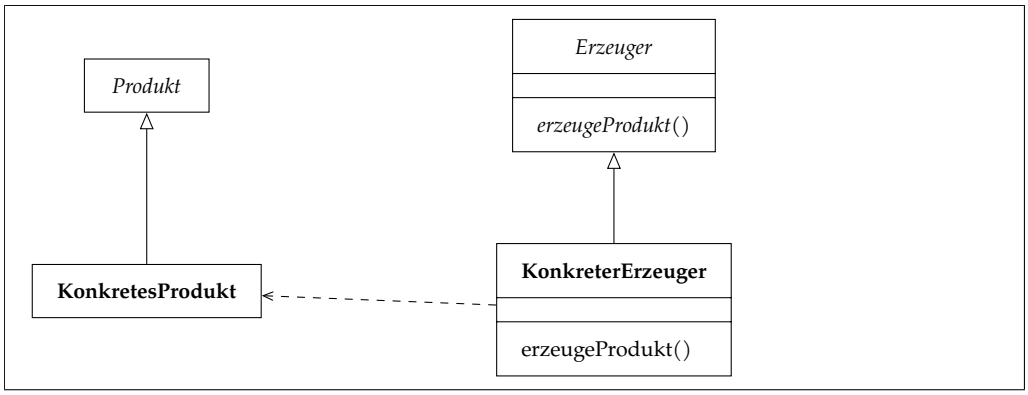
\bEntwurfsErbauer

```
\def\bEntwurfsErbauer{
  \bEntwurfsErbauerUml
  \bEntwurfsErbauerAkteure
}
```

Fabrikmethode (Factory Method)

\bEntwurfsFabrikmethodeUml

Quelle nach der deutschen Wikipedia




```
\def\bEntwurfsFabrikmethodeUml{
  \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).

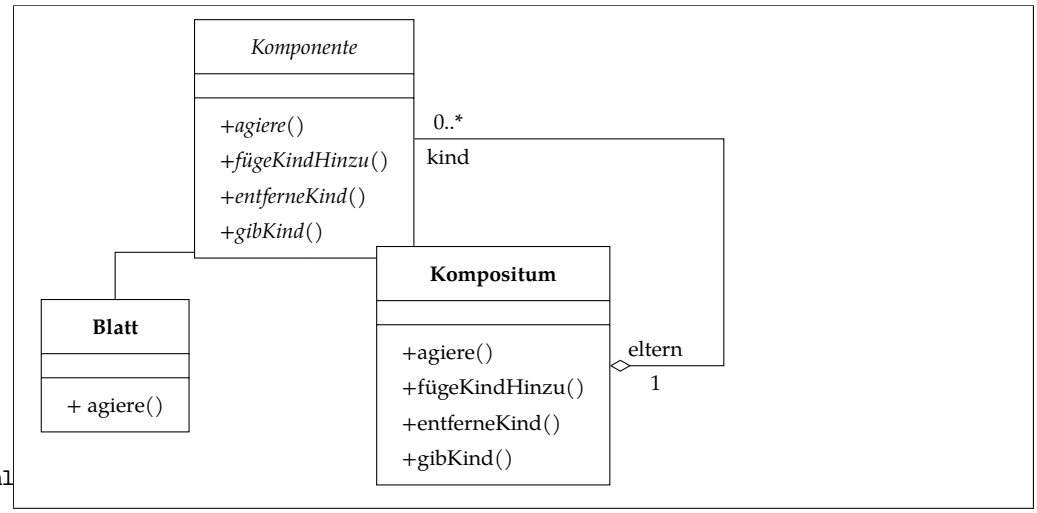
```
\def\bEntwurfsFabrikmethodeAkteure{
  \begin{description}
    \item[Produkt]

    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}
}
```

\bEntwurfsFabrikmethode

```
\def\bEntwurfsFabrikmethode{
  \bEntwurfsFabrikmethodeUml
  \bEntwurfsFabrikmethodeAkteure
}
```

Kompositum (Composite)



\bEntwurfsKompositumUml

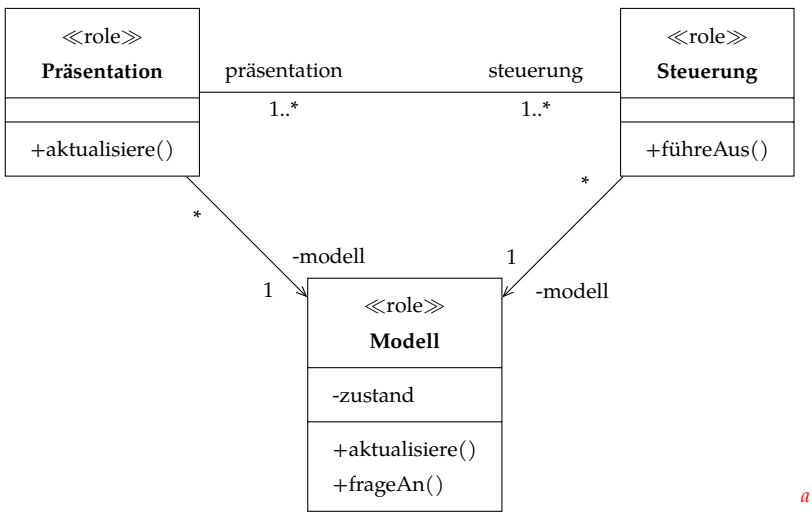
```
\def\bEntwurfsKompositumUml{
  \begin{tikzpicture}
    \umlclass[x=2.5,y=3,type=abstract]{Komponente}{}{
      \textit{+agiere()}\\
      \textit{+fügeKindHinzu()}\\
      \textit{+entferneKind()}\\
      \textit{+gibKind()}
    }
    \umlclass[x=0]{Blatt}{}{+ agiere()}
    \umlclass[x=5]{Kompositum}{}{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

\bEntwurfsFabrikmethode

```
\def\bEntwurfsKompositum{
  \bEntwurfsKompositumUml
  \bEntwurfsKompositumAkteure
}
```

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

odellPraesentationSteuerungUml



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

```
\def\bEntwurfsModellPraesentationSteuerungUml{
  \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}
  \bFussnoteUrl{http://www.cs.sjsu.edu/~pearce/modules/lectures/ood2/mvc/index.htm}
}
```

ModellPraesentationSteuerung

```
\def\bEntwurfs{
  \bEntwurfsModellPraesentationSteuerungUml
  \bEntwurfsModellPraesentationSteuerungAkteure
}
```

Stellvertreter (Proxy)

\bEntwurfsStellvertreterUml

```
\def\bEntwurfsStellvertreterUml{
  \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}
}
```

\bEntwurfsStellvertreterCode

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

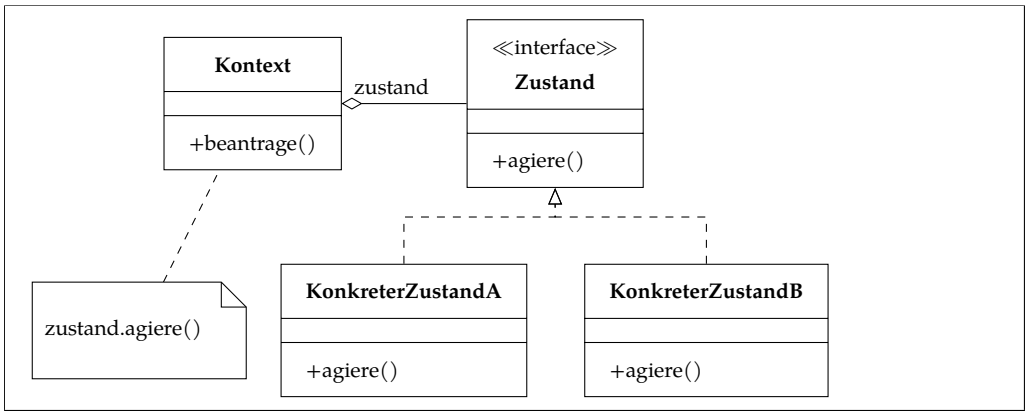
\bEntwurfsStellvertreter

```
\def\bEntwurfsStellvertreter{
  \bEntwurfsStellvertreterUml
  \bEntwurfsStellvertreterCode
}
```

Zustand (State)

\bEntwurfsZustandUml

Quelle: Englische Wikipedia, so ähnlich wie in GoF




```
\def\bEntwurfsZustandUml{
  \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}
}
```

\bEntwurfsZustandAkteure

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\bEntwurfsZustandAkteure{
  \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}
}
```

\bEntwurfsZustandCode

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

\bEntwurfsZustand

```
\def\bEntwurfsZustand{
  \bEntwurfsZustandUml
  \bEntwurfsZustandAkteure
  \bEntwurfsZustandCode
}
```

er.sty

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

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}
```

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}
```

Faulenzer

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

```
\ExplSyntaxOn
```

`\bErEntity`

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

`\bErRelationship`

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

`\bErAttribute`

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

`\bErMpEntity` mp = marginpar
Makro-Faulenzer: `\let\e=\liErMpEntity`

```
\def\bErMpEntity#1{
  \bErEntity{#1}
  \marginpar{
    \bErEntity{\tiny\faSquareO{}~E:~#1}
  }
}
```

□

`\bErMpRelationship` **Makro-Faulenzer:** `\let\r=\liErMpRelationship`

```
\def\bErMpRelationship#1{
  \bErRelationship{#1}
  \marginpar{
    \bErRelationship{\tiny\faGg{}~R:~#1}
  }
}
```

`\bErMpAttribute` **Makro-Faulenzer:** `\let\a=\liErMpAttribute`

```
\def\bErMpAttribute#1{
  \bErAttribute{#1}
  \marginpar{
    \bErAttribute{\tiny\faCircleThin{}~A:~#1}
  }
}
```

`\bErDatenbankName` **Makro-Faulenzer:** `\let\d=\liErDatenbankName`
datenbank name

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

```
\ExplSyntaxOff
```

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}
```

```
\bLadePakete{mathe,typographie}
```

`\bMenge` `$\bMenge{a, b, c}$`: $\{a, b, c\}$
Makro-Faulenzer: `\let\m=\liMenge`

```
\def\bMengeOhneMathe#1{\{ #1 \}}
\def\bMenge#1{%
\ifmmode%
\bMengeOhneMathe{#1}%
\else%
$\bMengeOhneMathe{#1}$%
\fi%
}
```

`\bEpsilon` `\bEpsilon`: ε
Makro-Faulenzer: `\let\e=\liEpsilon`

```
\def\bEpsilon{$\varepsilon$}
```

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

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

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

```
\let\bZustandsmengeOhneMathe=\bPotenzmengeOhneMathe
\let\bZustandsmenge=\bPotenzmenge
```

`\bUeberfuehrungsFunktion` `\bUeberfuehrungsFunktion{z0, a}`: `$\delta(z_0, a)$`
Makro-Faulenzer: `\let\d=\liUeberfuehrungsFunktion`

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

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

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

`\bBandAlphabet` `\bBandAlphabet{\bTuringLeerzeichen}`: $\Gamma = \Sigma \cup \{\square\}$

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

`\bZustandsBuchstabe`

```
\def\bZustandsBuchstabe{z}
```

`\bZustandsBuchstabeGross`

```
\def\bZustandsBuchstabeGross{Z}
```

\bZustandsmengeNr

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

\bZustandsmengeNrGross

```
\def\bZustandsmengeNrGross#1{\zustandsnamens@liste{\bZustandsBuchstabeGross}{#1}}
```

\bZustandsname

\bZustandsname{1}: z_1

```
\def\bZustandsname#1{$\bZustandsBuchstabe_#1$}
```

\bZustandsnameGross

\bZustandsnameGross{1}: Z_1

```
\def\bZustandsnameGross#1{$\bZustandsBuchstabeGross_#1$}
```

\bAbleitung

\bAbleitung{S -> aB -> ab}: $S \vdash aB \vdash ab$

```
\def\bAbleitung#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 }
{
  \bGeschweifteKlammern{#1}
  {
    \begin{align*}
    \directlua{formale_sprachen.produktions_regeln('#2')}
    \end{align*}
  }{-0.2cm}{-1.5cm}
} {}
```

\bProduktionen

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

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

\bZustandsnameTiefgestellt

Automatisch tiefgestellte Nummerierung \z1
Makro-Faulenzer: \let\z=\liZustandsnameTiefgestellt

```
\def\bZustandsnameTiefgestellt#1{
  \ifmmode
    \bZustandsBuchstabe\sb{#1}
  \else
    $\bZustandsBuchstabe\sb{#1}$
  \fi
}
```

```
\ExplSyntaxOn
```

\bAusdruck

\bAusdruck[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 „=“: \bAusdruck[]{x}{y}: $\{ x \mid y \}$
Regulärer Ausdruck zum Konvertieren:

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

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

`\bFlaci` Link zur flaci.com Website: `\bFlaci{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\bFlaci#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
}
```

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

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

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

```
\ExplSyntaxOn
\NewDocumentCommand {\bGrammatik} { 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 {\bMenge{##1}}},
    alphabet .code:n = {\tl_set:Nn \l_alphabet_tl {\bMenge{##1}}},
    produktionen .code:n = {\tl_set:Nn \l_produktionen_tl {\bProduktionen{##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
```

formatierung.sty

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

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}
```

Farben

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

Überschriften

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

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})}
```

Kasten

```
\RequirePackage{mdframed}
```

`liKasten`

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

Header

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

Zeilenabstände

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

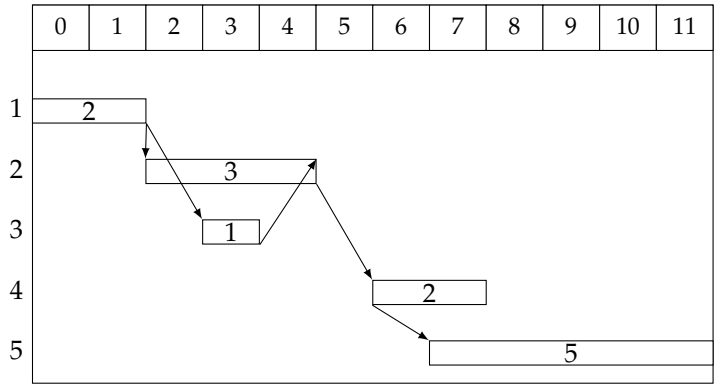
```
\RequirePackage{setspace}
```

1852

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}{}
```

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}
```

\bGrafikLogoPfad

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

\bGrafikCCLizenz

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

\bGrafikLogo

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

```
\ExplSyntaxOff
```

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

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

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

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

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}
```

\bHanoi `\bHanoi{anzahl-scheiben}{gewicht-scheibe/turm-nr,gewicht-scheibe/turm-nr}`, z. B.: `\bHanoi{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\bHanoi#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}
}
```

komplexitaetstheorie.sty

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

Faulenzer

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

```
\bLadePakete{mathe}
```

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

```
\RequirePackage{mdframed}
```

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

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

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

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

`\bProblemBeschreibung` Zu setzen von Problem-Beschreibungen:

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

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?

Makro-Faulenzer: `\let\b=\liProblemBeschreibung`

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

    \medskip

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

\bPolynomiellReduzierbar **Makro-Faulenzer:** \let\r=\liPolynomiellReduzierbar

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

\bProblemVertexCover

```
\def\bProblemClique{%
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}
}
```

\bProblemVertexCover

```
\def\bProblemVertexCover{%
%
Das \textbf{Knotenüberdeckungsproblem} (\bProblemName{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}%
}
```

\bProblemSubsetSum Kein Karp-21-Problem

```
\def\bProblemSubsetSum{%
Das \textbf{Teilsummenproblem} (\bProblemName{Subset Sum} oder
\bProblemName{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}
}
```

\bProblemSubsetSum Kein Karp-21-Problem

```
\def\bProblemSat{%
Das \textbf{Erfüllbarkeitsproblem der Aussagenlogik} \bProblemName{Sat}
und \bProblemName{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}
}

```

kontrollflussgraph.sty

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

Faulenzer

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

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}
```

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=-,
    }
  }
}
```

Umgebungen

liKontrollflussgraph

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

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

Makros

\bAnweisung

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

\bBedingung **Makro-Faulenzer:** \let\b=\liBedingung

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

\bBedingungWahr **Makro-Faulenzer:** \let\w=\liBedingungWahr

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

\bBedingungFalsch **Makro-Faulenzer:** \let\f=\liBedingungFalsch

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

\bKontrollCode **Makro-Faulenzer:** \let\c=\liKontrollCode

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

\bKontrollTextzeileKnoten **Makro-Faulenzer:** \let\k=\liKontrollTextzeileKnoten

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

\bKontrollKnotenPfad **Makro-Faulenzer:** \let\p=\liKontrollKnotenPfad

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

kopfzeile.sty

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

```
\ExplSyntaxOn
```

```
\RequirePackage{bschlangaul-meta}
```

```
\cs_new:Npn \kopfzeile_schrift:n #1
{
  {
    \scriptsize
    #1
  }
}
```

```
\cs_new:Npn \setze_kopf_fusszeilen:nn #1 #2
{
  \fancyhead{}
  \fancyhead[L]{\kopfzeile_schrift:n{\bMetaBschlangaulSammlung}}
  \fancyhead[C]{\kopfzeile_schrift:n{#1}}
  \fancyhead[R]{\kopfzeile_schrift:n{#2}}

  \fancyfoot{}
  \fancyfoot[L]{\kopfzeile_schrift:n{\bMetaHermineFriends}}
  \fancyfoot[C]{\kopfzeile_schrift:n{\bMetaEmail}}
  \fancyfoot[R]{\kopfzeile_schrift:n{\thepage}}

  \renewcommand{\headrulewidth}{0.4pt}
  \renewcommand{\footrulewidth}{0.4pt}
}
```

```
\cs_new:Npn \setze_kopfzeile_oben_rechts:n #1
{
  \fancyhead[R] {
    \kopfzeile_schrift:n { #1 }
  }
}
```

Das Makro darf nicht in der Präambel aufgerufen werden, da es die Textbreite braucht.

```
\cs_new:Npn \repariere_kopfzeile_breite:
{
  \setlength{\headwidth}{\textwidth}
}
```

```
\ExplSyntaxOff
```

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 }{}
```

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]}
```

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`\\}
```

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

```
\newcommand{\bPseudoUeberschrift}[1]{
  \bigskip

  \par
  \noindent
  \textbf{#1}

  \medskip

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

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

```
\newcommand{\bBeschriftung}[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} {}
```

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 }
{
  \ifADDITUM
  \else
    % Alles in eine Box verschieben und die dann ignorieren.
    \setbox 0 \vbox
    \bgroup
  \fi

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

  \ifADDITUM
  \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} }
{
  \ifANTWORT
  \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}
  \ifANTWORT
  \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 }
{
  \ifADDITUM
  \else
    \setbox 0 \vbox
    \bgroup
  \fi

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

  \ifADDITUM
  \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}{
  \ifEXKURS
    \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}
} {}
```

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

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

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

```
\NewDocumentCommand{\bFussnoteLink} { 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
```

master-theorem.sty

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

Faulenzer

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

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

\bMasterFallRechnung
% 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
{$f(n) = 5n^2 \notin \o{n^{\log_2 {8 + \varepsilon}}}$}

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

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

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

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

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

\bRundeKlammer

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

\bTheta

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

```
\def\bThetaOhneMathe#1{
  \Theta \bRundeKlammer{#1}
}
\def\bTheta#1{
  \ifmmode
    \bThetaOhneMathe{#1}
  \else
    $\bThetaOhneMathe{#1}$
  \fi
}
```

\bOmega \bOmega{n^2}: $\Omega\left(n^2\right)$

```
\def\bOmegaOhneMathe#1{
  \Omega \bRundeKlammer{#1}
}
\def\bOmega#1{
  \ifmmode
    \bOmegaOhneMathe{#1}
  \else
    $\bOmegaOhneMathe{#1}$
  \fi
}
```

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

```
\def\bOOhneMathe#1{
  \mathcal{O} \bRundeKlammer{#1}
}
\def\bO#1{
  \ifmmode
    \bOOhneMathe{#1}
  \else
    $\bOOhneMathe{#1}$
  \fi
}
```

\bT **Makro-Faulenzer:** `\let\T=\liT`

`\bT{16}{2}`: $16 \cdot T\left(\frac{n}{2}\right)$ `\bT{}{2}`: $T\left(\frac{n}{2}\right)$

```
\def\bTOhneMathe#1#2{
  \tl_if_blank:nTF {#1}
  {}
  {#1 \cdot }
  T
  \bRundeKlammer{\frac{n}{#2}}
}
\def\bT#1#2{
  \ifmmode
    \bTOhneMathe{#1}{#2}
  \else
    $\bTOhneMathe{#1}{#2}$
  \fi
}
```

\bRekursionsGleichung \bRekursionsGleichung: $T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$

```
\def\bRekursionsGleichung{
  $T(n) = \bT{a}{b} + f(n)$
}
```

\bBedingungEins \bBedingungEins: $f(n) \in \mathcal{O}\left(n^{\log_b a - \varepsilon}\right)$

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

\bBedingungZwei \bBedingungZwei: $f(n) \in \Theta\left(n^{\log_b a}\right)$

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

\bBedingungDrei \bBedingungDrei: $f(n) \in \Omega\left(n^{\log_b a + \varepsilon}\right)$

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

```
\ExplSyntaxOff
```

\bMasterVariablen

```
\def\bMasterVariablen{
  \begin{displaymath}
  T(n) = \bT{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}
}
```

\bMasterFaelle

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

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

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

  \hfill falls \bBedingungZwei

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

  \hfill falls \bBedingungDrei
  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}
}
```

\bMasterVariablenDeklaration

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

    \bRekursionsGleichung

    \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

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

\bMasterFallRechnung

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

  #1

  \item[2. Fall:] \bBedingungZwei:

  #2

  \item[3. Fall:] \bBedingungDrei:

  #3
  \end{description}
}
```

\bMasterExkurs

```
\def\bMasterExkurs{
  \begin{liExkurs}[Master-Theorem]
  \bMasterVariablen

  \noindent
  Dann gilt:

  \bMasterFaelle
  \end{liExkurs}
}
```

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

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

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}{`'}

```

meta.sty

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

```
\ExplSyntaxOn
```

```
\bLadePakete{grafik}
```

Einfache Makros (Low level)

\bMetaBschlangaulSammlung

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

\bMetaHermineFriends

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

\bMetaUeberDasProjekt

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

\bMetaCCLink

```
\def\bMetaCCLink{
  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
  }.
}
```

\bMetaEmail

```
\def\bMetaEmail{
  hermine.bschlangaul@gmx.net
}
```

\bMetaEmailLink

```
\def\bMetaEmailLink{
  \href{
    mailto:\bMetaEmail
  }{
    \bMetaEmail
  }
}
```

\bMetaHilfMit

```
\def\bMetaHilfMit{
  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~
  \bMetaEmailLink.
}
```

\bMetaHilfMit

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

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}
}
```

\bLogoTextProjekt

```
\def\bLogoTextProjekt
{
  \logo_dann_text:nn
  {
    \bGrafikLogo[width=5cm]
  }
  {
    {
      \bfseries
      \bMetaBschlangaulSammlung
    }
    \par

    \bMetaHermineFriends
    \par

    \medskip

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

\bLogoTextCCLizenz

```
\def\bLogoTextCCLizenz
{
  \logo_dann_text:nn
  {
    \centerline{\bGrafikCCLizenz[width=3cm]}
  }
  {
    \begin{spacing}{1}
      \scriptsize
      \bMetaCCLink
```

```
    \end{spacing}
  }
}
\ExplSyntaxOff
```

minimierung.sty

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

```
\bLadePakete{typographie}
```

```
\let\z=\bZustandsnameTiefgestellt
\let\f=\bFussnote
\let\l=\bLeereZelle
\let\Z=\bZustandsPaar
\let\erklaerung=\bMinimierungErklaerung

\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}

\bFussnoten

\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}
```

\bFussnote

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

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

\bFussnoteEinsText

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

\bFussnoteZweiText

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

\bFussnoteDreiText

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

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

\bFussnoteVierText

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

\bFussnoten

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\bFussnoten{
  \bigskip

  \noindent
  \bFussnoteEinsText

  \noindent
  \bFussnoteZweiText

  \noindent
  \bFussnoteDreiText

  \noindent
  \bFussnoteVierText
}
```

\bLeereZelle

\bLeereZelle: ∅

Makro-Faulenzer: \let\l=\liLeereZelle

```
\def\bLeereZelle{$\emptyset$}
```

\bZustandsPaarVariablenName

```
\def\bZustandsPaarVariablenName{z}
```

\bZustandsPaar

```
\def\bZustandsPaar#1#2{
  $(
    \bZustandsPaarVariablenName_#1,
    \bZustandsPaarVariablenName_#2
  )$
}
```

liUebergangsTabelle

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

bUeberschriftDreiecksTabelle

\bUeberschriftDreiecksTabelle:

Minimierungstabelle (Table filling)

```
\ExplSyntaxOn
```

```
\def\bUeberschriftDreiecksTabelle{
  \bPseudoUeberschrift{Minimierungstabelle~(Table~filling)}
}
```

\bMinimierungErklaerung

Makro-Faulenzer: `\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\bMinimierungErklaerung{
  %\footcite[Seite~19]{koenig}
  \bParagraphMitLinien{
    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
```

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

```
\bLadePakete{mathe,typographie}
```

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

Faulenzer

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

`\bAttributHuelle` **Makro-Faulenzer:** `\let\ah=\liAttributHuelle`

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

```
\def\bAttributHuelleOhneMathe#1{\text{AttrHülle}(#1)}
\def\bAttributHuelle#1{
\ifmmode
  \bAttributHuelleOhneMathe{#1}
\else
  $\bAttributHuelleOhneMathe{#1}$
\fi
}
```

`\bAttributMenge` **Makro-Faulenzer:** `\let\m=\liAttributMenge`

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

`liAHuelle`

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

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

Makro-Faulenzer: `\let\ahL=\liLinksReduktion`

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

```
\def\bLinksReduktion#1#2#3{
  \shoveleft{
    \bAttributHuelleOhneMathe{FA,
    \bAttributMenge{#1 \string\ #2}} =
    } \\
```

```
  \shoveright{
    \bAttributMenge{#3}
  } \\
}
```

\bLinksReduktionInline

Makro-Faulenzer: `\let\ahL=\liLinksReduktionInline`
`\ahl{ursprüngliche linke Attributmenge}{ohne dieses Attribut}{Ergebnis}`
`\ahl{A, E}{E}{A, E, F, B, \textbf{D}}`: $\mathrm{AttrHülle}(F, \{A, E \setminus E\}) = \{A, E, F, B, \boldsymbol{D}\}$

```
\def\bLinksReduktionInline#1#2#3{%
  {%
    \footnotesize%
    $\bAttributHuelleOhneMathe{F,
    \bAttributMenge{#1 \string\ #2}} =
    \bAttributMenge{#3}$
  }
}
```

\bLinksReduktionInline

Makro-Faulenzer: `\let\ahr=\liLinksReduktionInline`
`\ahr{gelöschte FA}{neue FA ohne rechts Attribut}{gegebene Attribute}{Ergebnis}`

```
\def\bRechtsReduktionInline#1#2#3#4{%
  {%
    \footnotesize%
    $\bAttributHuelleOhneMathe{
      F \setminus
      \bFunktionaleAbhaengigkeit{#1}
      \def\tmp{#2}\ifx\tmp\empty
      \else
        \cup \bFunktionaleAbhaengigkeit{#2}
      \fi
      ,
      \bAttributMenge{#3}
    } =
    \bAttributMenge{#4}$
  }
}
```

\bFunktionaleAbhaengigkeit

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

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

\bFunktionaleAbhaengigkeiten

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

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

Makro-Faulenzer: `\let\FA=\liFunktionaleAbhaengigkeiten`
Regulärer Ausdruck zum Konvertieren: `\\item \$(.*) \\rightarrow (.*)\$ \$(.*) \\rightarrow`
`$1 -> $2;`

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

\bRelation **Makro-Faulenzer:** `\let\r=\liRelation`

`\r[R3]{\u{A}, B, C}`: $R_3(\underline{A}, B, C)$ Regulärer Ausdruck zum Konvertieren: `\$(R.*)\((.*)\)\$` `\\bRelation[$1]{$2}`

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

o-notation.sty

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

Faulenzer

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

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}
```

\bRundeKlammer

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

\bONotationO

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

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

petri.sty

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

Faulenzer

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

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

Für die Darstellungsmatrix

```
\RequirePackage{blkarray}
```

```
\def\TmpA#1{
  \bPetriSetzeSchluessel%
  \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}
}
```

\bPetriSetzeSchluessel

```
\def\bPetriSetzeSchluessel{%
  \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]}-
    }
  }
}
```

\bPetriTransitionsName **Makro-Faulenzer:** `\let\t=\liPetriTransitionsName`
`\$t_(\d+)\$ \t$1`

```
\def\bPetriTransitionsNameOhneMathe#1{t\sb{#1}}
\def\bPetriTransitionsName#1{
  \ifmmode
    \bPetriTransitionsNameOhneMathe{#1}
  \else
    $\bPetriTransitionsNameOhneMathe{#1}$
  \fi
}
```

\bPetriErreichTransition **Makro-Faulenzer:** `\let\t=\liPetriErreichTransition`

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

\bPetriErreichKnotenDrei **Makro-Faulenzer:** `\let\k=\liPetriErreichKnotenDrei`

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

\bPetriTransPfeile **Makro-Faulenzer:** `\let\tp=\liPetriTransPfeile`

```
\def\bPetriTransPfeile#1{$\rightarrow \hspace{0.4cm} \bPetriTransitionsName{#1} \hspace{0.35c
```

potenzmengen-konstruktion.sty

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

```
\bLadePakete{formale-sprachen}
\ExplSyntaxOn
```

```
\def\z#1{
  \bZustandsMengenSammlungNr{#1}{
    {
      {0} {0}
      {1} {0,1}
      {2} {0,2}
      {3} {0,1,3}
      {4} {0,2,3}
      {5} {0,3}
    }
  }
}
\let\s=\bZustandsnameGross

\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}
```

\bZustandsMengenSammlung

\bZustandsMengenSammlung{nummer}{latex3 str_case:nn}

```
\def\z#1{
  \bZustandsMengenSammlung{#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\bZustandsMengenSammlung#1#2{
  \bZustandsnameGross{#1}
  {
    \footnotesize
    \bPotenzmenge{
      \str_case:nn {#1} #2
    }
  }
}
```

\bZustandsMengenSammlungNr

```
\def\bZustandsMengenSammlungNr#1#2{
  \bZustandsnameGross{#1}
  {
```

```
    \footnotesize
    \bZustandsmengeNr{
      \str_case:nn {#1} #2
    }
  }
}
\ExplSyntaxOff
```

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}
```

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]
```

\bPumpingRegulaer

```
\def\bPumpingRegulaer{%
  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}
}
```

\bPumpingKontextfrei

```
\def\bPumpingKontextfrei{%
  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}
}
```

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)
```

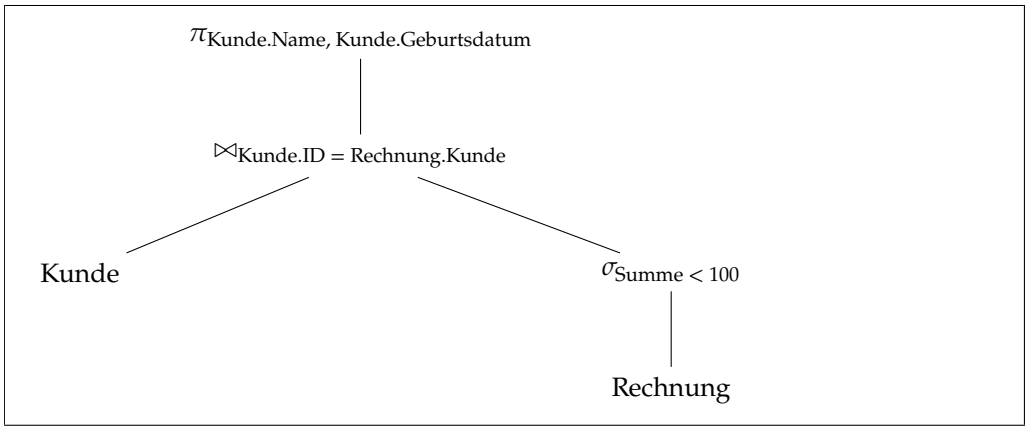
```
\def\QSpivotStep {\let\QSLr\DecoLEFTwithPivot
                \let\QSIr\DecoINERT
                \let\QSIrr\DecoINERT
                \let\QSRr\DecoRIGHTwithPivot
\par\centerline{\rule[1.5mm]{0pt}{8mm}%
            \setcounter{cellcount}{0}\setcounter{pivotcount}{0}%
            \begin{tikzpicture}\QS@list\end{tikzpicture}}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

% 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}}
}
```

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`: $A \mathbin{\o@join\mkern-5.8mu\bowtie} B$

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

`\rightouterjoin` `A \rightouterjoin B`: $A \mathbin{\bowtie\mkern-5.8mu\o@join} B$

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

`\fullouterjoin` `A \fullouterjoin B`: $A \mathbin{\o@join\mkern-5.8mu\bowtie\mkern-5.8mu\o@join} B$

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

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}
```

Faulenzer

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

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

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

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

```
\def\bFremd#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
```

`\bRelationMenge` **Makro-Faulenzer:** `\let\r=\liRelationMenge`

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

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

`\bAttribut` **Makro-Faulenzer:** `\let\a=\liAttribut`

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

```
\def\bAttribut#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 } {} {}
```

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
  \bSortierPfeil{one}{two}
  \bSortierPfeil{two}{three}
  \bSortierMarkierung{two split south}{three split north}
  \bSortierPfeilUnten{three}{one}
}
```

```
\RequirePackage{tikz}
\usetikzlibrary{shapes.multipart,positioning}
```

`\bVertauschen` `\bVertauschen{1 2 >4 <3 5}`: Setze ein Schaubild mit Hilfe von TikZ. < und > werden dazu verwendet, um den Vertauschprozess zu visualisieren.

```
\def\bVertauschen#1{
  \directlua{
    local sortieren = require('bschlangaul-sortieren')
    sortieren('#1')
  }
}
```

`\bSortierPfeil`

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

`\bSortierPfeilUnten`

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

`\bSortierMarkierung`

```
\def\bSortierMarkierung#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}
```

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}
```

`\bSpaltenUmbruch`

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

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

sql.sty

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

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}
```

```
\bLadePakete{syntax}
```

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

struktogramm.sty

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

```

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.

Faulenzer

```
\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
}
```

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

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

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

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

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

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

```
\NewDocumentCommand{\bJavaDatei}{ 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)}}
}
```

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

```
\NewDocumentCommand{\bJavaTestDatei}{ 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)}}
}
```

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

```
\NewDocumentCommand{\bJavaExamen}{ 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')}}
}
```

\bAssemblerCode

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

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

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

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

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

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

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

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

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

```
\ExplSyntaxOff
```

`\bSqlCode` `\bHaskellCode{sql}`: Zum Setzen von SQL-Code.
Makro-Faulenzer: `\let\s=\liSqlCode`

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

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
    },
  }
}

```

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]
```

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

Faulenzer

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

TeX-Markup Grundgerüst

```
\let\schrittE=\bSyntheseUeberErklaerung
\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}
```

TeX-Markup Linksreduktion

```
\let\ahl=\bLinksReduktionInline
\let\fa=\bFunktionaleAbhaengigkeit
\let\m=\bAttributMenge
\let\b=\textbf

\bPseudoUeberschrift{\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}

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

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

TeX-Markup Rechtreduktion

```
\let\ahr=\bRechtsReduktionInline
\let\fa=\bFunktionaleAbhaengigkeit

\bPseudoUeberschrift{F}

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

\bPseudoUeberschrift{A}

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

TeX-Markup Relationen formen

```
\let\r=\bRelation
\let\u=\underline

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

(a) **Kanonische Überdeckung**

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

(i) **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 $\beta \subseteq AttrHülle(F, \alpha - A)$.*

(ii) **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.*

(iii) **Löschen leerer Klauseln**

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

(iv) **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.*

(b) **Relationsschemata formen**

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

(c) **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 $\mathcal{F}_\mathcal{K} := \emptyset$*

(d) **Entfernung überflüssiger Teilschemata**

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

`\bSyntheseUeberschrift` **Makro-Faulenzer:** `\let\schritt=\liSyntheseUeberschrift`

```
\def\bSyntheseUeberschrift#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}
    }
  }
}
```

\bSyntheseErklaerung **Makro-Faulenzer:** \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~
      $\beta~\subseteq~\bAttributHuelle{F,~\alpha~--A}.$
    }
    {1-2} {
      Führe~für~jede~(verbliebene)~funktionale~Abhängigkeit~$\alpha~
      \rightarrow~\beta$~die~Rechtsreduktion~durch,~überprüfe~also~für~
      alle~$B~\in~\beta$,~ob~$B~\in~\bAttributHuelle{F~--(\alpha~
      \rightarrow~\beta)~\cup~(\alpha~\rightarrow~(\beta~--B)),~
      \alpha}$~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\bSyntheseErklaerung#1{
  {
    \itshape
    \footnotesize
    \bParagraphMitLinien{\li@synthese@erklaerung@texte{#1}}
  }
```

```
}
```

\bSyntheseUeberErklaerung **Makro-Faulenzer:** `\let\schrittE=\liSyntheseUeberErklaerung`

```
\def\bSyntheseUeberErklaerung#1{
  \bSyntheseUeberschrift{#1}\par
  \bSyntheseErklaerung{#1}
}

\ExplSyntaxOff
```

tabelle.sty

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

```

tex-dokumentation.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-tex-dokumentation}[2021/09/12 Hilfsmakros
für die DTX-Dokumentation]
```

```
\ExplSyntaxOn
```

```
\RequirePackage{mdframed}
```

```
\newenvironment{liBeispiel}
{
  \begin{mdframed}
}
{
  \end{mdframed}
}
```

\bMakroFaulenzer `\bMakroFaulenzer{`⟨*Abkürzung*⟩`}{`⟨*Makroname ohne Schrägstrich*⟩`}`

```
\cs_new:Npn \let_abkuerzung:nn #1 #2
{
  \texttt
  {
    \textbackslash let
    \textbackslash#1
    =
    \textbackslash#2
  }
}
\def\bMakroFaulenzer #1 #2
{
  \par
  \noindent
  \textbf{Makro-Faulenzer:~}
  \let_abkuerzung:nn {#1}{#2}
  \par
}
```

```
\prop_new:N \l_faulenzer_prop
```

\bFaulenzer

```
\def\bFaulenzer#1
{
  \prop_clear:N \l_faulenzer_prop
  \prop_put_from_keyval:Nn \l_faulenzer_prop {#1}
  \subsubsection{Faulenzer}
  \prop_map_inline:Nn \l_faulenzer_prop
  {
    \noindent
    \let_abkuerzung:nn {##1} {##2}
    \par
  }
  \bigskip
}
```

```
\ExplSyntaxOff
```

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}
```

`\bErledigt` `\bErledigt`: ☑

```
\let\bErledigt=\faCheckSquareO
```

`\bNichtsZuTun` `\bNichtsZuTun`: ∅ Nichts zu tun

```
\def\bNichtsZuTun{$\emptyset$~Nichts~zu~tun}
```

`\bParagraphMitLinien` `\bParagraphMitLinien{Lorem ipsum...}`:

— Lorem ipsum dolor sit, ipsum dolor sit, ipsum dolor sit, ipsum dolor sit, ipsum dolor sit, ipsum dolor sit ———————————————————————————————————————

```
\def\bParagraphMitLinien#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
}
```

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

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

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

`\bTypoUeberschrift`

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

\bTypoUeberGross

```
\def\bTypoUeberGross#1{
  {
    \huge
    \bTypoUeberschrift{#1}
  }
}
```

\bTypoUeberGross

```
\def\bTypoUeberGROSS#1{
  {
    \Huge
    \bTypoUeberschrift{#1}
  }
}
```

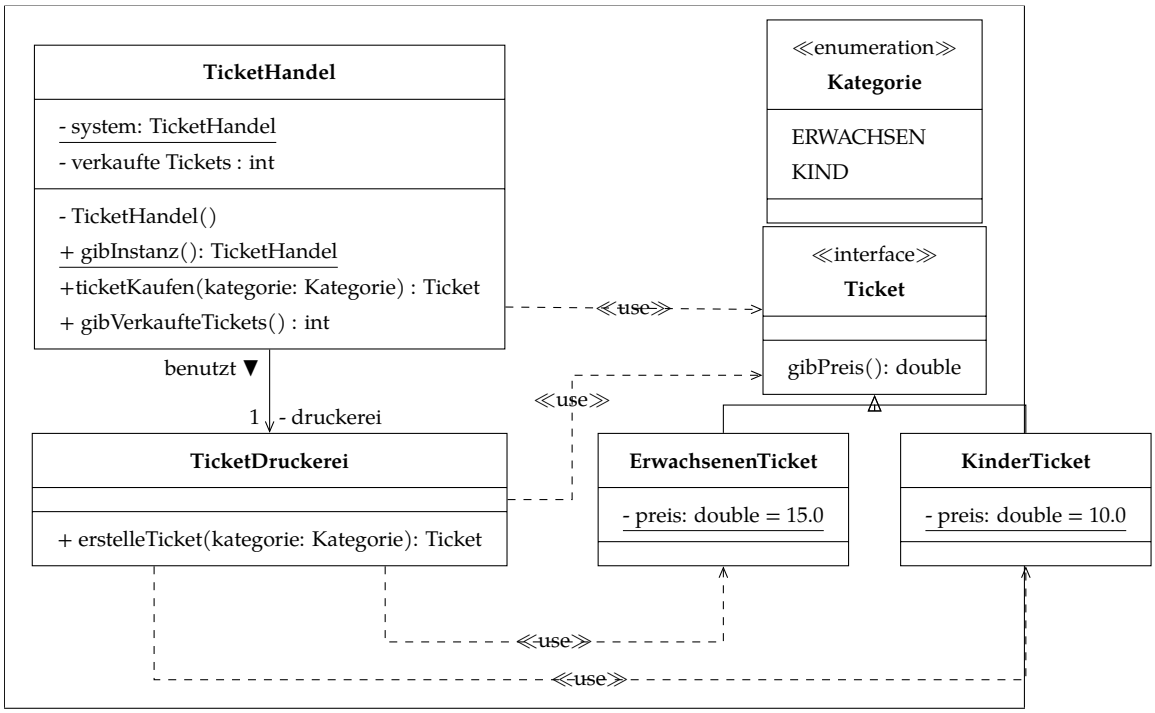
```
\ExplSyntaxOff
```

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,
}
```



\bUmlLeserichtung

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

```
\NewDocumentCommand{ \bUmlLeserichtung } { 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
  };
}
```

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]
```

Faulenzer

```
\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
```

`\bInduktionMarkierung` Hilfsmakro um Teile von mathematischen Formeln markieren zu können.
Makro-Faulenzer: `\let\m=\liInduktionMarkierung`

```
\def\bInduktionMarkierung#1{\textcolor{violet}{#1}}
```

`\bInduktionErklaerung` Gedacht für die rechte Spalte in der align-Umgebung. Das text-Makro ist dann nicht mehr nötig.

Makro-Faulenzer: `\let\e=\liInduktionErklaerung`

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

`\bInduktionAnfang`

```
\def\bInduktionAnfang{
  \bPseudoUeberschrift{Induktionsanfang}

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

`\bInduktionVoraussetzung`

```
\def\bInduktionVoraussetzung{
  \bPseudoUeberschrift{Induktionsvoraussetzung}

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

`\bInduktionSchritt`

```
\def\bInduktionSchritt{
  \bPseudoUeberschrift{Induktionsschritt}

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

```
\ExplSyntaxOff
```

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}

```

wpkalkuel.sty

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

Faulenzer

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

```
\RequirePackage{amsmath}
```

```
\ExplSyntaxOn
```

\bWpKalkuel **Makro-Faulenzer:** `\let\wp=\liWpKalkuel`

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

\MatheEnv

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

  \hspace{1em}#1

  \medskip
}
```

\Mathe

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

\bWpEquivalent **Makro-Faulenzer:** `\let\equivalent=\liWpEquivalent`

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

\bWpErklaerung **Makro-Faulenzer:** `\let\erklaerung=\liWpErklaerung`

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

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

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

\bWpErklaerungVerzweigung

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

\ExplSyntaxOff
```

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

\# 471
\, . . 720, 790, 845, 1124, 1142, 1774, 1776, 2357, 2358, 2359, 2799, 3300, 3307, 3540
\@Skip@Erklaerung@Reset 3693, 3695, 3706
\@afterheading 81, 2202, 2326
\@afterindentfalse 80, 2201, 2325
\@liDirLeft 3588, 3593, 3605
\@liDirRight 3589, 3591, 3592, 3593, 3594, 3605
\@liDistance 3599, 3600, 3604
\@liPos . . . 3596, 3597, 3604
\\ . . . 158, 160, 184, 188, 192, 1154, 1182, 1183, 1186, 1187, 1190, 1191, 1286, 1287, 1288, 1395, 1429, 1431, 1461, 1470, 1515, 1557, 1558, 1559, 1564, 1565, 1566, 1586, 2189, 2694, 2752, 2755
\{ 569, 1711, 1721, 1733, 1734, 1739, 1773, 2005, 2739, 3164, 3540, 3709
\} 569, 1711, 1721, 1733, 1734, 1741, 1777, 2006, 2739, 3164, 3544, 3709
_ . 47, 61, 104, 245, 394, 402, 407, 409, 667, 695, 719, 722, 732, 755, 758, 762, 764, 766, 768, 771, 783, 784, 785, 788, 790, 804, 805, 811, 814, 817, 827, 834, 836, 839
\␣ 2751, 2761

A

\addbibresource 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167
\ADDITUMfalse 585
\ADDITUMtrue 16, 585
\AddToHook . 84, 88, 287, 326
\advance 3213
\AfterEndEnvironment 3289
\Alph 1834
\alph 1834, 1835
\alpha . 3408, 3410, 3411, 3414, 3416, 3417, 3418, 3419, 3420, 3424, 3430, 3431, 3436, 3437, 3438, 3441, 3449, 3450, 3451
\ANTWORTfalse 589
\ANTWORTtrue 589
\arabic 1834, 3008, 3013, 3018, 3024, 3030, 3036
\arraystretch 2689
\AtBeginDocument 181

B

\bAbleitung 1748
\bAlphabet 1733
\bAnweisung 2090
\bAssemblerCode 3348
\bAssemblerDatei . . . 3349
\bAttribut 3167
\bAttributHuelle 2731, 3411, 3416
\bAttributHuelleOhneMathe 2731, 2734, 2736, 2750, 2760, 2768
\bAttributMenge 2739, 2751, 2754, 2761, 2762, 2776, 2778
\bAufgabe 373
\bAufgabenMetadaten 38, 243, 404
\bAufgabenTitel 412
\bAusdruck 1770
\bAutomat 423
\bAutomatenKante 455
\bBandAlphabet 1734
\bBedingung 2091
\bBedingungDrei 2422, 2465, 2505
\bBedingungEins 2416, 2454, 2497
\bBedingungFalsch . . 2093
\bBedingungWahr 2092
\bBedingungZwei 2419, 2460, 2501
\bBeschriftung 2204
\bBindeAufgabeEin . . . 226
\bBindePdfEin 180
\bChomskyErklaerung 993, 1034
\bChomskyUeberErklaerung 1032
\bChomskyUeberschrift 981, 1033
\bCpmEreignis 1042
\bCpmFruehErklaerung 1113
\bCpmFruehI . . . 1106, 1126
\bCpmSpaetErklaerung 1131
\bCpmSpaetI . . . 1099, 1144
\bCpmVon 1082
\bCpmVonOhneMathe 1082, 1085, 1087
\bCpmVonZu 1074
\bCpmVonZuOhneMathe 1074, 1077, 1079
\bCpmVorgang 1059
\bCpmZu 1090

\bCpmZuOhneMathe 1090, 1093, 1095
\BeforeBeginEnvironment 3288
\begin 98, 156, 173, 857, 866, 874, 1180, 1234, 1249, 1284, 1308, 1355, 1387, 1402, 1427, 1437, 1457, 1477, 1509, 1526, 1555, 1580, 1601, 1625, 1640, 1754, 1839, 1931, 1959, 1968, 1975, 2083, 2181, 2245, 2267, 2287, 2314, 2319, 2330, 2340, 2344, 2427, 2431, 2450, 2473, 2496, 2511, 2591, 2592, 2596, 2619, 2632, 2692, 2693, 2743, 2788, 2928, 2948, 3089, 3111, 3125, 3288, 3311, 3479, 3544
\begingroup 2182, 2741, 3209
\bEntwurfs 1596
\bEntwurfsAbstrakteFabrik 1226
\bEntwurfsAbstrakteFabrikBeschreibung 1174, 1227
\bEntwurfsAbstrakteFabrikCode 1221, 1231
\bEntwurfsAbstrakteFabrikUml 1179, 1229
\bEntwurfsAdapter .. 1278
\bEntwurfsAdapterAkteure 1248, 1280
\bEntwurfsAdapterCode 1272, 1281
\bEntwurfsAdapterUml 1233, 1279
\bEntwurfsBeobachter 1349
\bEntwurfsBeobachterAkteure 1307, 1351
\bEntwurfsBeobachterCode 1341, 1352
\bEntwurfsBeobachterUml 1283, 1350
\bEntwurfsDekorierer 1381
\bEntwurfsDekoriererAkteure 1383
\bEntwurfsDekoriererCode 1373, 1384
\bEntwurfsDekoriererUml 1354, 1382
\bEntwurfsEinfacheFabrik 1417
\bEntwurfsEinfacheFabrikAkteure 1401, 1419
\bEntwurfsEinfacheFabrikUml 1386, 1418
\bEntwurfsEinzelstueck 1447
\bEntwurfsEinzelstueckAkteure 1436, 1452
\bEntwurfsEinzelstueckBeschreibung 1421, 1448
\bEntwurfsEinzelstueckCode 1444, 1454
\bEntwurfsEinzelstueckUml 1426, 1450
\bEntwurfsErbauer .. 1504
\bEntwurfsErbauerAkteure 1476, 1506
\bEntwurfsErbauerUml 1456, 1505
\bEntwurfsFabrikmethode 1550, 1575
\bEntwurfsFabrikmethodeAkteure 1525, 1552
\bEntwurfsFabrikmethodeUml 1508, 1551
\bEntwurfsKompositum 1575
\bEntwurfsKompositumAkteure 1577
\bEntwurfsKompositumUml 1554, 1576
\bEntwurfsModellPraesentation 1596
\bEntwurfsModellPraesentationBeschreibung 1598
\bEntwurfsModellPraesentationUml 1579, 1597
\bEntwurfsStellvertreter 1620
\bEntwurfsStellvertreterCode 1614, 1622
\bEntwurfsStellvertreterUml 1600, 1621
\bEntwurfsZustand .. 1661
\bEntwurfsZustandAkteure 1639, 1663
\bEntwurfsZustandCode 1657, 1664
\bEntwurfsZustandUml 1624, 1662
\bEpsilon 1719
\bErAttribute 1677, 1691, 1693
\bErDatenbankName .. 1696
\bErEntity 1675, 1679, 1681
\bErledigt 3524
\bErMpAttribute 1690
\bErMpEntity 1678
\bErMpRelationship . 1684
\bErRelationship 1676, 1685, 1687
\beschriftung 2225, 2229, 2233, 2237, 2241, 2243
\beta 3408, 3411, 3415, 3416, 3417, 3420, 3430, 3431, 3432, 3437, 3438
\bExamensAufgabe 376
\bExamensAufgabeA ... 385
\bExamensAufgabeTA .. 382
\bExamensAufgabeTTA . 379
\bFalsch 974
\bFaulenzer 3503
\bFlaci 1780
\bFremd 3147
\bfseries . 54, 147, 212, 252, 858, 881, 983, 1824, 1826, 2609, 2986, 2992, 2994, 2996, 2997, 3385, 3549
\bFunktionaleAbhaengigkeit 2770, 2773, 2781
\bFunktionaleAbhaengigkeiten 2784
\bFussnote 2644, 2646
\bFussnoteDreiText 2658, 2676
\bFussnoteEinsText 2650, 2670
\bFussnoteLink 2354
\bFussnoten 2666
\bFussnoteUrl .. 1594, 2350
\bFussnoteVierText 2662, 2679
\bFussnoteZweiText 2654, 2673
\bGeschweifteKlammern 1752, 2785, 3536
\bGrafikCCLizenz 1871, 2629
\bGrafikLogo 875, 1876, 2605
\bGrafikLogoPfad 1868, 1873, 1878
\bGrammatik 1791
\bgroup 2220, 2264
\bHanoi 1917
\bHaskellCode 3355
\bHaskellDatei 3356
\Bigl 3540
\Bigr 3544
\bigskip 76, 93, 96, 298, 301, 823, 1156, 1161, 2191, 2667, 3514
\bInduktionAnfang .. 3619
\bInduktionErklaerung 3618
\bInduktionMarkierung 3617
\bInduktionSchritt . 3635
\bInduktionVoraussetzung 3627
\bJavaCode 3299
\bJavaDatei ... 1172, 3317
\bJavaExamen 3337
\bJavaTestDatei 3327
\bKellerAutomat 464
\bKellerKante 502
\bKellerUebergang 497, 503
\bKontrollCode 2094
\bKontrollKnotenPfad 2096
\bKontrollTextzeileKnoten 2095, 2100

\bKurzeTabellenLinie 1154
\bLadeAllePakete 215, 322, 595
\bLadePakete 5, 116, 123, 135, 347, 416, 419, 591, 596, 980, 1041, 1710, 1954, 2542, 2643, 2723, 2891, 3260, 3381
\bLatexCode 3309
\bLeereZelle 2681
\bLinksReduktion ... 2748
\bLinksReduktionInline 2757, 2765
\bLogoTextCCLizenz 95, 300, 2625
\bLogoTextProjekt 92, 297, 2601
\bMakroFaulenzer ... 3484
\bMasterExkurs 2510
\bMasterFaelle . 2449, 2517
\bMasterFallRechnung 2495
\bMasterVariablen 2426, 2512
\bMasterVariablenDeklaration 2472
\bMasterWolframLink 2520
\bMenge .. 433, 434, 436, 475, 476, 477, 481, 533, 534, 535, 539, 1711, 1760, 1799, 1800
\bMengeOhneMathe 1711, 1714, 1716
\bMetaBschlangaulSammlung . 883, 2120, 2543, 2610
\bMetaCCLink ... 2555, 2634
\bMetaEmail 2126, 2565, 2570, 2572
\bMetaEmailLink 2568, 2584
\bMetaHermineFriends . . 891, 2125, 2546, 2614
\bMetaHilfMit 101, 2575, 2586
\bMetaQuelltext . 103, 2586
\bMetaSetze 40, 244, 393, 405
\bMetaUeberDasProjekt 2549, 2621
\bMinimierungErklaerung 2703
\bMinispracheDatei . 3352
\bNichtsZuTun 3525
\bO 2389, 2417
\bOmega 2379, 2423
\bOmegaOhneMathe 2379, 2384, 2386
\bONotationO 2811
\bOOhneMathe 2389, 2394, 2396
\bool 698, 791
\bowtie 3134, 3137, 3138, 3139
\Box 521
\boxtimes 973
\bParagraphMitLinien 1029, 1114, 1132, 2705, 3459, 3526, 3623, 3631, 3639
\bPetriErreichKnotenDrei 2885
\bPetriErreichTransition 2882
\bPetriSetzeSchluessel 2827
\bPetriTransitionsName 2874, 2886
\bPetriTransitionsNameOhneMathe 2874, 2877, 2879
\bPetriTransPfeile . 2886
\bPolynomiellReduzierbar 1974
\bPotenzmenge 1720, 1724, 2897
\bPotenzmengeOhneMathe 1721, 1722, 1723
\bPrimaer 3146
\bProblemBeschreibung 1958
\bProblemClique 1981
\bProblemName 1957, 1964, 1976, 1978, 1991, 2002, 2003, 2011, 2012
\bProblemSat 2010
\bProblemSubsetSum 2001, 2010
\bProblemVertexCover 1981, 1989
\bProduktionen . 1759, 1801
\bPruefungsNummer ... 138
\bPruefungsTitel 141
\bPseudoUeberschrift 2190, 2270, 2273, 2691, 2701, 3620, 3628, 3636
\bPumpingKontextfrei 2943
\bPumpingRegulaer .. 2922
\bRechtsReduktionInline 2765
\bRekursionsGleichung 2413, 2476
\bRelation 2795
\bRelationMenge 3162
\bRichtig 973
\bRundeKlammer 2366, 2370, 2380, 2390, 2404, 2808, 2812
\bSetzeExamenTeilaufgabeNr 222
\bSetzeExamenThemaNr . 218
\bSortierMarkierung 3187
\bSortierPfeil 3181
\bSortierPfeilUnten 3184
\bSpaltenUmbruch ... 3256
\bSqlCode 3360
\bStrich 1956
\bSyntheseErklaerung 3399, 3464
\bSyntheseUeberErklaerung 3462
\bSyntheseUeberschrift 3383, 3463
\bT .. 2399, 2414, 2428, 2492
\bTeilen 2728
\bTheta 2369, 2420, 2452, 2458, 2463
\bThetaOhneMathe 2369, 2374, 2376
\bTitelSeite 169, 183
\bTOhneMathe 2399, 2408, 2410
\bTrennSeite 152
\bTuringKante 565
\bTuringLeerzeichen 521, 529
\bTuringMaschine 522
\bTuringUeberfuehrung 568
\bTuringUebergaenge 560, 566
\bTuringUebergangZelle 555
\bTypoUeberGROSS ... 3559
\bTypoUeberGross 3553, 3559
\bTypoUeberschrift 3547, 3556, 3562
\bUeberfuehrungsFunktion 1725
\bUeberfuehrungsFunktionOhneMathe 1725, 1728, 1730
\bUeberschriftDreiecksTabelle 2699
\bUmlLeserichtung .. 3587
\bVertauschen 3175
\bWortInSprache 1155
\bWortNichtInSprache 1160
\bWpEquivalent 3690
\bWpErklaerung 3693
\bWpErklaerungVerzweigung 3708
\bWpKalkuel 3670
\bWpKalkuelOhneMathe 3670, 3675, 3677, 3709, 3711, 3713
\bZustandsBuchstabe 1735, 1744, 1746, 1764, 1766
\bZustandsBuchstabeGross 1736, 1745, 1747
\bZustandsmenge 1723
\bZustandsmengeNr 1737, 2906
\bZustandsmengeNrGross 1745
\bZustandsMengenSammlung 2893
\bZustandsMengenSammlungNr 2902
\bZustandsmengeOhneMathe 1723
\bZustandsname 1746
\bZustandsnameGross 1747, 2894, 2903

\bZustandsnameTiefgestellt 1762
\bZustandsPaar 2683
\bZustandsPaarVariablenName 2682, 2685, 2686

C

\c 1933, 1934
\cdot 2402, 2458, 2469
\centerline 1964, 2629, 3087, 3109, 3124
\cftbeforesecskip . . . 278
\cftbeforesubsecskip 279, 280
\cftbeforesubsubsecskip 281
\cftsubsecafterpnum . 280
\chapter 1824, 1825
\char 2189
\clearpage 153, 167, 170, 178, 213, 291, 303, 900, 2183
\cline 1154
\clist 593, 641, 642, 664, 668, 3221
\columnbreak 3256
\contentsname 282
\cs 18, 25, 31, 247, 275, 667, 695, 719, 722, 732, 755, 766, 777, 788, 790, 827, 834, 839, 850, 2110, 2117, 2132, 2138, 2308, 2590, 2811, 3484
\csname 1918, 1921
\cup 1734, 2773, 3417, 3431, 3438

D

\DeclareMathSymbol 2535, 2536
\DecoINERT 3011, 3084, 3085, 3107
\DecoINERTwithPivot 3027, 3106
\DecoLEFT 3006, 3105
\DecoLEFTwithPivot 3021, 3083
\DecoRIGHT 3016, 3108, 3118
\DecoRIGHTwithPivot 3033, 3086
\definecolor 1822
\DefineVerbatimEnvironment 3262
\delta 427, 469, 527, 569, 1725
\dh 2359, 3419
\directlua 420, 499, 557, 562, 1706, 1720, 1740, 1748, 1755, 1760, 2724, 2729, 2782, 2789, 2796, 3176, 3275, 3319, 3324, 3325, 3329, 3334, 3335, 3339, 3345, 3346
\do 3007, 3012, 3017, 3022, 3028, 3034
\dots 1014, 1018, 2005, 2935, 3430, 3431
\DOWNarrow 3592
\draw 1938, 1941, 1944, 2883, 3182, 3185

E

\edef 1929, 3100, 3104, 3116, 3117
\edge 638
\egroup 2256, 2281
\else 1078, 1086, 1094, 1102, 1109, 1715, 1729, 1765, 2218, 2241, 2255, 2262, 2280, 2305, 2375, 2385, 2395, 2409, 2735, 2772, 2817, 2878, 3046, 3239, 3241, 3676
\emph . . 1175, 1422, 1677, 1986, 2015, 2017, 2188
\empty 2241, 2771
\emptyset 2681, 3425, 3446, 3525
\end 105, 164, 175, 862, 870, 894, 1219, 1245, 1270, 1305, 1339, 1371, 1399, 1415, 1434, 1442, 1473, 1502, 1523, 1548, 1573, 1593, 1612, 1637, 1655, 1756, 1841, 1947, 1971, 1972, 1979, 2088, 2186, 2253, 2277, 2303, 2321, 2322, 2336, 2347, 2348, 2429, 2445, 2470, 2493, 2508, 2518, 2594, 2598, 2599, 2622, 2635, 2696, 2697, 2745, 2790, 2938, 2958, 3089, 3111, 3125, 3289, 3315, 3482, 3544
\endcsname 1918, 1921
\endgroup . 2185, 2746, 3214
\enspace 3529, 3531
environments:
liAdditum 2259
liAHuelle 2740
liAntwort 2215
liDiagramm 2338
liEinbettung 2214
liExkurs 2284
liGraphenFormat . 1910
liJavaAngabe 3296
liKasten 1838
liKontrollflussgraph 2082
liLernkartei 2328
liProduktionsRegeln 1749
liProjektSprache 2213
liQuellen 2308
liRelationenSchemaFormat 3168
liRmodell 3148
liUebergangsTabelle 2689
\equiv 3691, 3710
\erzeuge@tiefgestellt 1720, 1721, 1725
\EXKURSfalse 587
\EXKURStrue 587
\expandafter 1918, 3042, 3044, 3045, 3046, 3054, 3212
\ExplSyntaxOff . . . 108, 127, 195, 305, 341, 411, 454, 496, 501, 554, 559, 564, 1036, 1058, 1073, 1149, 1701, 1779, 1814, 1881, 2103, 2142, 2360, 2425, 2638, 2717, 2911, 3161, 3222, 3359, 3466, 3516, 3565, 3644, 3715
\ExplSyntaxOn 17, 119, 137, 217, 324, 392, 423, 464, 497, 522, 555, 560, 584, 979, 1042, 1059, 1098, 1674, 1769, 1791, 1865, 2096, 2108, 2179, 2364, 2541, 2699, 2804, 2892, 3149, 3219, 3274, 3382, 3475, 3522, 3616, 3669

F

\faCheckSquareO 3524
\faCircleThin 1693
\faGg 1687
\fancyfoot 1845, 1846, 1847, 2124, 2125, 2126, 2127
\fancyhead . 1844, 2119, 2120, 2121, 2122, 2134
\faSquareO 1681
\fi . . . 1080, 1088, 1096, 1104, 1111, 1717, 1731, 1767, 2221, 2244, 2257, 2265, 2282, 2306, 2377, 2387, 2397, 2411, 2737, 2774, 2819, 2880, 3047, 3239, 3240, 3243, 3245, 3678
\filcenter 212

\footcite 1177, 1246, 1268, 1315, 1338, 1370, 1423, 1474, 1501, 1547, 1984, 1987, 1994, 1999, 2004, 2008, 2014, 2019, 2147, 2446, 2447, 2704, 2941
\footnote 2351, 2355
\footnotesize 70, 509, 800, 934, 1028, 1698, 2212, 2298, 2315, 2331, 2620, 2648, 2742, 2759, 2767, 2896, 2905, 3148, 3263, 3294, 3458, 3579, 3605
\footrulewidth . 1850, 2130
\foreach .. 1933, 1936, 1943
\forestFirst ... 3236, 3239
\forestLast ... 3237, 3239
\forestOget ... 3236, 3237
\forestOnes 3249
\forestOv . 3238, 3239, 3242
\forestov .. 3228, 3232, 3233, 3236, 3237, 3238, 3239, 3241, 3242
\forestset 3224, 3247
\forestSortLevel 3226, 3234, 3248, 3249
\frac . 2404, 2437, 2469, 2484
\fullouterjoin 3139

G

\g 19, 21, 32, 34, 42, 56, 58, 139, 142, 158, 160, 188, 192, 219, 223, 227, 231, 232, 233, 234, 235, 237, 238, 240, 254, 257, 260, 265, 396, 641, 642, 664, 668, 674, 675, 676, 677, 678, 680, 681, 683, 684, 686, 687, 688, 689, 690, 691, 692, 693, 696, 700, 701, 702, 705, 707, 708, 709, 710, 711, 712, 756, 760, 764, 767, 768, 770, 771, 773, 774, 778, 780, 793, 794, 795, 796, 804, 805, 807, 813, 814, 816, 817, 819, 820, 832, 835, 840, 842, 846
\Gamma .. 468, 526, 569, 1734
\gappto 3244
\geometry 360
\geq 2012, 2435, 2924, 2929, 2945, 2949
\gib 74, 777

H

\hbox 3134
\headrulewidth . 1849, 2129
\headwidth 2140
\hfill 64, 2454, 2460, 2465, 3532
\hinweis 2212
\hline 2694
\horizontale 25, 72
\href 1787, 2355, 2522, 2557, 2569, 3314
\hspace ... 2886, 3683, 3691
\ht 3135
\Huge 162, 188, 252, 332, 3561
\huge 212, 3555

I

\i 1943, 1944
\ifADDITUM . 585, 2261, 2279
\ifANTWORT . 589, 2217, 2254
\ifcase 3043
\ifEXKURS 587, 2285
\ifmmode 1076, 1084, 1092, 1100, 1107, 1713, 1727, 1763, 2373, 2383, 2393, 2407, 2733, 2815, 2876, 3674
\IfNoValueTF 2268, 2351, 2355
\ifnum 2970, 3211, 3227, 3235, 3241
\ifx 2241, 2771, 3239
\in 1002, 1158, 2012, 2417, 2420, 2423, 2452, 2458, 2463, 2924, 2935, 2945, 2955, 3408, 3410, 3416, 3437, 3632
\includegraphics 1872, 1877
\includepdf 180
\inhalts 275, 293
\inhaltsverzeichnis 2180
\input 228, 374, 377, 380, 383, 386, 904
\inputminted 3318, 3328, 3338, 3350, 3353, 3357
\int 3221
\item ... 973, 974, 1251, 1255, 1260, 1265, 1309, 1318, 1323, 1331, 1403, 1408, 1412, 1438, 1478, 1483, 1490, 1498, 1527, 1532, 1536, 1541, 1641, 1646, 1651, 1969, 1970, 2308, 2312, 2432, 2437, 2441, 2451, 2457, 2462, 2474, 2478, 2482, 2486, 2490, 2497, 2501, 2505, 2929, 2932, 2935, 2949, 2952, 2955
\itshape 1027, 3457

J

\j 1933, 1934, 1936, 1937, 1938, 1943, 1944, 1945

K

\k 1943
\keys . 120, 398, 432, 444, 474, 484, 532, 542, 672, 1046, 1050, 1064, 1069, 1798, 1805
\kopfzeile 2110, 2120, 2121, 2122, 2125, 2126, 2127, 2135

L

\l ... 425, 426, 427, 428, 429, 430, 433, 434, 435, 436, 437, 439, 441, 446, 447, 448, 449, 450, 451, 466, 467, 468, 469, 470, 471, 472, 475, 476, 477, 478, 479, 480, 481, 487, 488, 489, 490, 491, 492, 493, 524, 525, 526, 527, 528, 529, 530, 533, 534, 535, 536, 537, 538, 539, 545, 546, 547, 548, 549, 550, 551, 1044, 1047, 1052, 1053, 1056, 1061, 1062, 1065, 1066, 1071, 1793, 1794, 1795, 1796, 1799, 1800, 1801, 1802, 1808, 1809, 1810, 1811, 2099, 2100, 2101, 2311, 2312, 2313, 2320, 3502, 3505, 3506, 3508
\labelenumi 1835
\labelenumii 1836
\labelitemi 1829
\labelitemii 1830
\labelitemiii 1831
\labelitemiv 1832
\land 3711, 3713
\LARGE 158, 192, 1824
\Large 55, 184, 859, 882
\large 46, 160, 867, 890, 1964, 3198
\leaders 3532
\left 2367, 2809
\LEFTarrow 3593
\leftarrow 1090
\leftouterjoin 3137
\leftskip . 3695, 3696, 3706
\LehramtInformatikGitBranch 831, 3282
\LehramtInformatikGithubCodeRepo 3281
\LehramtInformatikGithubDomain 828, 3278

\LehramtInformatikGithubRawDownload 3279
\LehramtInformatikGithubTexRepo 829, 3280
\LehramtInformatikRepository 180, 229, 374, 377, 380, 383, 386, 1869, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 3277
\leq 2469, 2932, 2952
\let 1723, 1724, 2183, 3083, 3084, 3085, 3086, 3099, 3101, 3102, 3103, 3105, 3106, 3107, 3108, 3118, 3210, 3248, 3249, 3484, 3499, 3511, 3524
\li@chomsky@erklaerung@texte 993, 1029
\li@EntwurfsCode 1171, 1222, 1223, 1224, 1273, 1274, 1275, 1276, 1342, 1343, 1344, 1345, 1346, 1347, 1374, 1375, 1376, 1377, 1378, 1379, 1445, 1615, 1616, 1617, 1618, 1658, 1659
\li@EntwurfsCodeAllgemein 1170
\li@fussnote@text 2645, 2651, 2655, 2659, 2663
\li@GithubLink 3310, 3323, 3333, 3344
\li@mget . . 1920, 1924, 1944
\li@minc 1923, 1945
\li@mset 1917, 1925, 1934, 1937
\li@numdiscs 1929, 1938, 1944
\li@Rmodell@Schrift 3148, 3157, 3167
\li@SansFett . . 144, 158, 160, 162, 184, 188, 192
\li@sequence . . . 1930, 1943
\li@synthese@erklaerung@texte 3399, 3459
liAdditum (environment) 2259
liAHuelle (environment) 2740
liAntwort (environment) 2215
liDiagramm (environment) 2338
liEinbettung (environment) 2214
liExkurs (environment) 2284
liGraphenFormat (environment) 1910
liJavaAngabe (environment) 3296
liKasten (environment) 1838
liKontrollflussgraph (environment) . 2082
liLernkartei (environment) 2328
\linespread 3155
liProduktionsRegeln (environment) . 1749
liProjektSprache (environment) 2213
liQuellen (environment) 2308
liRelationenSchemaFormat (environment) . 3168
liRmodell (environment) 3148
\listen@punkt . . 2308, 2320
\literatur 2146, 2170
liUebergangsTabelle (environment) . 2689
\llap 3135
\LoadClass 4, 113, 132, 133, 200, 311, 346
\log 2417, 2420, 2423, 2452, 2458
\logo 2590, 2603, 2627
\loop 2968
\lor 3712
\ltimes 2528

M

\makeatletter 3207
\makeatother 3215
\marginpar 1680, 1686, 1692, 2188
\mathbb . . . 2012, 2955, 3632
\mathbin . . 3137, 3138, 3139
\mathcal 2390, 2812, 3437, 3442, 3444, 3445, 3446
\Mathe 3687
\MatheEnv . 3680, 3688, 3691
\mathord 2535, 2536
\mdfsetup 2226, 2230, 2234, 2238
\mdseries 2246
\medskip . . . 48, 67, 887, 1966, 2197, 2207, 2209, 2318, 2346, 2617, 3153, 3159, 3534, 3538, 3681, 3685
\memph 2188
\mintinline 3302, 3309, 3348, 3355, 3360
\mkern 3137, 3138, 3139
\mlq 2533, 2535
\mrq 2533, 2536
\msg 902
\myList 3228, 3229, 3230, 3233
\myNodes 3217, 3232, 3238, 3242, 3244

N

\NeedsTeXFormat 1, 110, 129, 197, 307, 343, 356, 369, 389, 414, 582, 908, 969, 976, 1038, 1151, 1166, 1667, 1703, 1816, 1853, 1862, 1883, 1912, 1950, 2022, 2105, 2144, 2150, 2172, 2362, 2525, 2538, 2640, 2719, 2802, 2822, 2888, 2913, 2918, 2962, 3128, 3141, 3170, 3251, 3258, 3265, 3270, 3362, 3377, 3468, 3472, 3518, 3567, 3609, 3646, 3666
\neg 3713
\negthinspace . . 2367, 2809
\newcounter . . . 2982, 2983
\NewDocumentCommand 424, 465, 502, 523, 565, 591, 1043, 1060, 1099, 1106, 1155, 1160, 1770, 1792, 1871, 1876, 1974, 2097, 2148, 2350, 2354, 2784, 2795, 2882, 3317, 3327, 3337, 3349, 3352, 3356, 3587
\NewDocumentEnvironment . 1749, 1838, 1910, 2082, 2213, 2214, 2215, 2259, 2284, 2309, 2328, 2338, 2690, 2740, 3150, 3168
\newif 585, 587, 589
\newlength 3693
\newminted 3296
\node 1056, 2090, 2095, 3008, 3013, 3018, 3024, 3030, 3036, 3187, 3232, 3604
\noexpand 3096, 3097, 3098, 3117, 3232
\noindent . . . 27, 45, 53, 69, 100, 802, 1157, 1162, 2194, 2206, 2210, 2242, 2299, 2301, 2316, 2332, 2334, 2342, 2514, 2669, 2672, 2675, 2678, 3163, 3497, 3510, 3527, 3539, 3699
\nolinkurl 3314
\normalsize . . . 1826, 3303
\notin 1163
\null 3532

O

\o 2811, 2816, 2818

\o@join 3134, 3137, 3138, 3139
\Omega 2380
\omega 2924, 2925, 2945, 2946
\or 3045

P

\pagestyle 852, 1848
\par . . . 22, 26, 29, 48, 65, 78, 255, 258, 261, 266, 271, 801, 822, 861, 869, 876, 885, 893, 1033, 1781, 1789, 2193, 2199, 2205, 2300, 2324, 2333, 2612, 2615, 3087, 3109, 3124, 3165, 3463, 3496, 3500, 3512, 3533, 3537, 3545, 3698, 3704
\paragraph 1826
\parindent 3156
\path . . . 456, 503, 566, 1071
\pgfkeys . . . 2838, 3590, 3591, 3592, 3593, 3594, 3597, 3600, 3602
\pgfmath@count 3209, 3211, 3213
\pgfmath@smuggleone 3214
\pgfmathdeclarefunction 3208
\pgfmathint 3209
\pgfmathparse 1924, 3226, 3231, 3234, 3248, 3249
\pgfmathresult 1925, 3209, 3210, 3212, 3214, 3227, 3235, 3248, 3249
\pgfutil@empty 3210
\pgfutil@loop 3211
\pgfutil@repeat 3214
\preceq 1977
\prime 1956
\printbibliography . 2170
\ProcessKeysOptions . 126
\prop . 3502, 3505, 3506, 3508
\ProvidesClass 2, 111, 130, 198, 308, 344
\ProvidesPackage 357, 370, 390, 415, 583, 909, 970, 977, 1039, 1152, 1167, 1668, 1704, 1817, 1854, 1863, 1884, 1913, 1951, 2023, 2106, 2145, 2151, 2173, 2363, 2526, 2539, 2641, 2720, 2803, 2823, 2889, 2914, 2919, 2963, 3129, 3142, 3171, 3252, 3259, 3266, 3271, 3363, 3378, 3469, 3473, 3519, 3568, 3610, 3647, 3667

Q

\QS@list 3089, 3100, 3104, 3111, 3117, 3122, 3125
\QS@select@equal 3064, 3068
\QS@select@greater 3065, 3069
\QS@select@smaller 3060, 3063, 3067
\QS@sort@a 3042, 3075, 3096, 3097
\QS@sort@b 3042, 3043
\QS@sort@c 3046, 3053
\QS@sort@d 3054, 3062
\QS@sort@empty . 3044, 3049
\QS@sort@single 3045, 3050
\QSinitialize 2967, 3079, 3114
\QSIr . . 3050, 3056, 3064, 3084, 3098, 3103, 3106
\QSIrr 3085, 3098, 3099, 3107
\QSLr 3056, 3063, 3074, 3075, 3083, 3096, 3101, 3105
\QSpivotStep 2969, 3079, 3083, 3094
\QSr 3056
\QSRr 3065, 3086, 3097, 3102, 3108, 3117, 3118, 3119
\QSsortStep 2971, 3079, 3095, 3096
\quad 2647

R

\raggedright 2249
\raisebox 2095
\relax 2183, 3056, 3099, 3101, 3102, 3103, 3211, 3213
\renewcommand 280, 282, 1829, 1830, 1831, 1832, 1835, 1836, 1849, 1850, 2129, 2130, 2689
\repariere . . 86, 328, 2138
\repeat 2972
\RequirePackage 15, 114, 115, 134, 136, 211, 216, 286, 323, 353, 354, 359, 372, 417, 520, 593, 905, 911, 912, 972, 1040, 1169, 1170, 1670, 1672, 1673, 1709, 1818, 1819, 1821, 1823, 1828, 1837, 1843, 1851, 1855, 1856, 1866, 1867, 1885, 1886, 1887, 1915, 1916, 1955, 2024, 2109, 2147, 2152, 2153, 2169, 2176, 2177, 2178, 2365, 2529, 2530, 2805, 2806, 2807, 2824, 2826, 2916, 2976, 2978, 3130, 3131, 3132, 3145, 3173, 3204, 3255, 3261, 3268, 3273, 3284, 3285, 3365, 3470, 3476, 3523, 3571, 3572, 3574, 3575, 3613, 3614, 3615, 3648, 3668
\right 2367, 2809
\RIGHTarrow . . . 3589, 3594
\Rightarrow . . . 1158, 1163
\rightarrow 569, 997, 1002, 1010, 1014, 1016, 1017, 1019, 1074, 1082, 2886, 3408, 3415, 3417, 3420, 3425, 3430, 3431, 3436
\rightouterjoin 3138
\rmfamily 148, 984, 1944, 3386, 3549
\Roman 1834
\roman 1834, 1836
\romannumeral 3054
\rtimes 2528
\rule 28, 3087, 3109, 3124, 3135

S

\sb 429, 439, 441, 470, 528, 1009, 1010, 1014, 1017, 1018, 1019, 1101, 1103, 1108, 1110, 1764, 1766, 2417, 2420, 2423, 2452, 2458, 2709, 2874, 2883, 3430, 3431, 3432, 3437, 3441, 3442, 3445, 3446, 3449, 3450, 3451
\scriptscriptstyle 1074, 1082, 1090
\scriptsize 1783, 2035, 2042, 2048, 2113, 2246, 2633, 3618, 3671, 3701
\section 212, 220, 409
\sectionbreak 213
\seq . . . 2099, 2100, 2101, 2311, 2312, 2313, 2320
\setbox . . . 2219, 2263, 3134
\setcounter . . 214, 285, 1827, 3088, 3110, 3124
\setganttlinklabel . . 1857, 1858, 1859, 1860
\setlength 278, 279, 281, 2140, 3156, 3695, 3696, 3706

\setmainfont 1820
\setmainlanguage 906
\setminted 3290, 3291
\setminus 2769
\setul 3147
\setze 74, 83, 325, 2117, 2132
\shoveleft 2749
\shoveright 2753
\Sigma 426, 467, 525, 1733, 1734, 1794
\sigma 1007, 1009, 1010
\SLASH 2189
\small 2341
\sort 3221
\sortList 3220, 3229
\square 974
\stepcounter 3008, 3013, 3018, 3021, 3023, 3027, 3029, 3033, 3035
\stichwoerter 18, 70
\str 139, 142, 158, 160, 188, 192, 723, 985, 994, 2223, 2898, 2907, 3387, 3400
\string 2751, 2761
\StrSubstitute . 3228, 3230
\strut 154, 166, 171, 177, 854, 898, 2474, 2478, 2482, 2486, 2490, 3256
\subsection 224
\subseteq . 3411, 3444, 3451
\subsubsection .. 245, 3507

T

\tableofcontents 283, 2184
\TeX 2587
\text 439, 441, 544, 2731, 3618, 3671
\textbackslash 3488, 3489, 3491
\textbf . 20, 1121, 1139, 1675, 1982, 1991, 2002, 2011, 2195, 2208, 2243, 2300, 2317, 2333, 2694, 3498
\textcolor 2094, 3301, 3617
\textit 33, 1515, 1557, 1558, 1559, 1560, 2343, 2739, 2799
\textsc 1957
\textsf 2317
\textstyle 2437, 2469
\texttt 1698, 1957, 2091, 2092, 2093, 2094, 3486, 3671
\textwidth 28, 2140
\thematik 31, 64
\thepage 1847, 2127
\theparagraph 1826
\thesection 212
\Theta 2370
\thinspace 3671
\thispagestyle 182
\tikz 2095
tikz: bbaum 34
tikz: li binaer baum 32
\tikzchildnode 929
\tikzparentnode 929
\tikzset 458, 505, 571, 914, 940, 1889, 2026, 2864, 2990, 3194, 3367, 3649
\tikzumlset 3577
\times 569
\tiny ... 99, 1681, 1687, 1693, 2094, 2188, 3312
\titel 247, 249, 289, 329, 850
\titleformat 212, 1824, 1826
\titlespacing 1825
\tl 19, 32, 42, 56, 219, 223, 227, 234, 237, 396, 425, 426, 427, 428, 429, 430, 433, 434, 435, 436, 437, 439, 441, 466, 467, 468, 469, 470, 471, 472, 475, 476, 477, 478, 479, 480, 481, 524, 525, 526, 527, 528, 529, 530, 533, 534, 535, 536, 537, 538, 539, 665, 669, 696, 700, 701, 702, 705, 710, 711, 712, 724, 733, 767, 770, 773, 778, 793, 794, 795, 796, 807, 813, 816, 819, 835, 842, 1044, 1047, 1052, 1053, 1061, 1062, 1065, 1066, 1772, 1793, 1794, 1795, 1796, 1799, 1800, 1801, 1802, 2400
\tmp 2771
\TmpPlaceEight 2846
\TmpPlaceFive 2843
\TmpPlaceFour 2842
\TmpPlaceNine 2847
\TmpPlaceOne 2839
\TmpPlaceSeven 2845
\TmpPlaceSix 2844
\TmpPlaceTen 2848
\TmpPlaceThree 2841
\TmpPlaceTwo 2840
\TmpScale 2859
\TmpTransitionEight 2835, 2856
\TmpTransitionFive 2832, 2853
\TmpTransitionFour 2831, 2852
\TmpTransitionNine 2836, 2857
\TmpTransitionOne 2828, 2849
\TmpTransitionSeven 2834, 2855
\TmpTransitionSix 2833, 2854
\TmpTransitionTen 2837, 2858
\TmpTransitionThree 2830, 2851
\TmpTransitionTwo 2829, 2850
\TmpX 2860
\TmpY 2861
\ttfamily 3148

U

\ul 1676, 3146, 3147
\umlaggreg 1634
\umlassoc 1592
\umlclass 1181, 1185, 1189, 1236, 1237, 1238, 1285, 1290, 1295, 1298, 1356, 1357, 1358, 1363, 1364, 1393, 1428, 1459, 1460, 1463, 1514, 1517, 1556, 1562, 1563, 1581, 1582, 1583, 1604, 1605, 1606, 1626, 1627, 1628, 1629
\umldep 1522
\umlHVHaggreg 1303, 1369, 1572
\umlinherit 1242, 1293, 1467, 1512, 1520
\umlnote .. 1244, 1469, 1636
\umlreal 1240, 1301
\umlsimpleclass 1196, 1197, 1198, 1202, 1204, 1205, 1206, 1235, 1388, 1389, 1390, 1458, 1510, 1511, 1602
\umlstatic 1395, 1429
\umluniaggreg 1465
\umluniassoc 1216, 1241, 1466, 1590, 1591, 1610, 1611
\umlVHuniassoc . 1217, 1218
\umlVHVdep . 1210, 1211, 1213, 1214, 1397, 1398
\umlVHVinherit .. 1193, 1194, 1199, 1200, 1207, 1208, 1366, 1367, 1391, 1392, 1570, 1571, 1608, 1609
\umlVHVreal 1360, 1361, 1631, 1632
\UParrow 3591
\url 836, 2351
\usemintedstyle 3287
\usetikzlibrary .. 418, 913, 1671, 1888,

2025, 2825, 3133, 3174, 3205, 3576, 3664

V

\value 2970
\varepsilon 986, 997, 998, 1719, 2417, 2423, 2455, 2466
\vbox 2219, 2263
\vfill . 90, 155, 165, 172, 176, 295, 864, 872, 3256
\vrule 3528, 3532
\vspace 159, 161, 186, 190, 268, 280, 855, 878, 896, 2286, 2304, 3541, 3543

X

\xappto . . . 3232, 3238, 3242
\xdef 1918
\xintApply 3058
\xintApplyUnbraced . . 3057, 3063, 3064, 3065
\xintCSVtoList 3117
\xintFor 3007, 3012, 3017, 3022, 3028, 3034, 3075
\xintifEq 3068
\xintifForLast 3024, 3030, 3036
\xintifGt 3069
\xintifLt 3067
\xintLength 3042
\xintnthelt 3054

Z

\ZB 2358
\zB 2357
\zustandsnamens@liste 1737, 1744, 1745