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
kopf-fusszeilen.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

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

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

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

In Aufgaben wollen wir das Additum sehen.

```
\LiADDITUMtrue
```

```
\ExplSyntaxOn
```

```
\cs_gset:Npn \stichwoerter_formatiert: {
  \tl_if_empty:NTF \g_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 )
    }
  }
}
```

`\liAufgabenMetadaten`

```
\def\liAufgabenMetadaten #1
{
  \liMetaSetze{#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
  {
    \liLogoTextProjekt
    \bigskip

    \liLogoTextCCLizenz
    \bigskip

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

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

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

```
\liLadePakete{formatierung}
```

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

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

`\liPruefungsNummer`

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

`\liPruefungsTitel`

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

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

`\liTrennSeite`

```
\def\liTrennSeite#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
}
```

`\liTitelSeite`

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

```
  \clearpage
}
```

\liBindePdfEin

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

\AtBeginDocument{
  \thispagestyle{empty}
  \liTitelSeite{
    \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.]
```

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

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

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

Formatierung für die Überschriften setzen.

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

```
\liLadeAllePakete
```

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

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

```
\RequirePackage{standalone}
```

```
\ExplSyntaxOn
```

`\liSetzeExamenThemaNr`

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

`\liSetzeExamenTeilaufgabeNr`

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

`\liBindeAufgabeEin`

```
\def\liBindeAufgabeEin#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
  }
}
```

`\liAufgabenMetadaten` Das Metadaten-Makro überschreiben

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

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

    \vspace{0.5cm}

    Fach~Informatik \par

    \vfill

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

    \vfill

    {
      \bfseries\Huge

      \g_jahreszeit_tl \par

      \g_jahr_tl \par
    }

    \vspace{2cm}

    {\LARGE \g_einzelpruefungs_nr_tl \par}

    \vspace{0.5cm}

    \g_examen_fach_tl \par

    \vspace{3cm}

    Aufgabenstellungen~mit~Lösungsvorschlägen \par

  \end{center}

  \vfill
}
```

```
\cs_new:Npn \inhalts_verzeichnis: {
  % für den Abstand vor den section im Inhaltsverzeichnis
  % https://tex.stackexchange.com/questions/241445/how-to-control-spacing-in-toc-for-different-sections
  \setlength{\cftbeforesecskip}{1.5cm}
  \setlength{\cftbeforesubsecskip}{0.5cm}
  \renewcommand{\cftsubsecafterpnum}{\vspace{\cftbeforesubsecskip}}
  \setlength{\cftbeforesubsubsecskip}{0.1cm}
  \renewcommand{\contentsname}{Aufgabenübersicht}
  \tableofcontents
}
```

```
\setcounter{tocdepth}{4}
\RequirePackage[titles]{tocloft}
```

```
\AddToHook{begindocument}{
  \titel_seite:

  \clearpage

  \inhalts_verzeichnis:

  \vfill

  \liLogoTextProjekt
  \bigskip

  \liLogoTextCCLizenz
  \bigskip

  \clearpage
}
\ExplSyntaxOff
```

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.]
```

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

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

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

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

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

```
\ExplSyntaxOn
```

```
\setze_kopf_fusszeilen:nn {} {}
```

```
\AddToHook { begindocument }
{
  \repariere_kopfzeile_breite:
}
```

```
\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

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

```
\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]
```

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

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

`\liExamensAufgabe` Eine Examensaufgaben mit den Pfad relativ zu Staatsexamen einbinden, z. B. `\liExamensAufgabe{461162/Teilaufgabe-1/Aufgabe-3}`

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

`\liExamensAufgabeTTA`

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

`\liExamensAufgabeTA`

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

`\liExamensAufgabeA`

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

aufgaben-metadaten.sty

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

```
\ExplSyntaxOn
```

`\liMetaSetze` Setze die Metadaten einer Aufgabe.

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

  \tl_clear:N \g_teilaufgabe_nr_tl

  \keys_set:nn { aufgabenmetadaten } {
    #1
  }

  \_setze_relativen_pfad:
}
```

`\liAufgabenMetadaten` Setzen der Aufgaben-Metadaten über eine plist bzw. über key-values.

Die Schlüssel-Werte-Paare sind in der Datei basis.sty definiert. In der Typescript-Datei `.scripts/nodejs/src/aufgaben.ts` gibt es ein entsprechendes Interface `AufgabenMetadaten`.

```
\liAufgabenMetadaten{
  Titel = {Aufgabe 2},
  Thematik = {Petri-Netz},
  Stichwoerter = {Feld (Array), Implementierung in Java}
  ZitatSchluessel = sosy:pu:4,
  ZitatBeschreibung = {Seite 11},
  BearbeitungsStand = OCR,
  Korrektheit = absolut korrekt,
  RelativerPfad = Staatsexamen/46116/2016/03/Thema-2/Teilaufgabe-1/Aufgabe-2.tex,
  ExamenNummer = 46116,
  ExamenJahr = 2016,
  ExamenMonat = 03,
  ExamenThemaNr = 2,
  ExamenTeilaufgabeNr = 1,
  ExamenAufgabeNr = 2,
}
```

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

  \_gib_examen_titel: {}

  \section{\_gib_aufgaben_titel:}
}
```

```
\ExplSyntaxOff
```

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

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

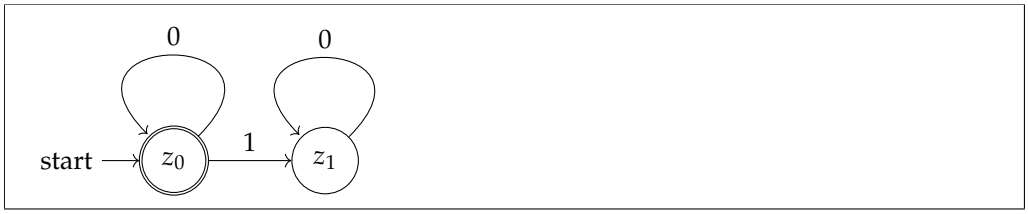
automaten.sty

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

Endlicher Automat

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

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

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




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

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

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

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

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

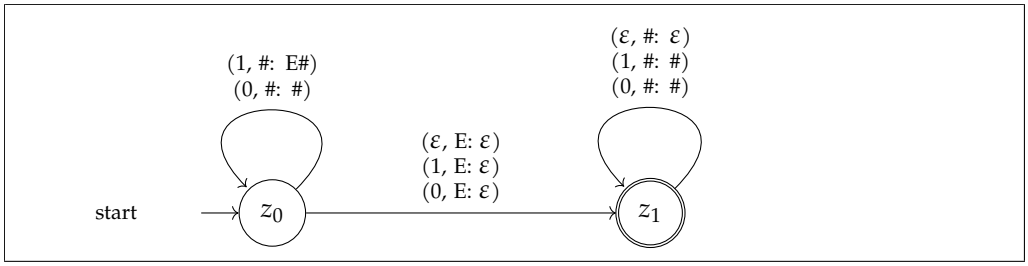
```
    start .code:n = {\tl_set:Nn \l_start_tl {##1}},
    dea .value_forbidden:n = true,
    dea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{DEA}}}},
    nea .value_forbidden:n = true,
    nea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{NEA}}}},
  }

  \keys_set:nn { automat } { #2 }

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

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

```
\def\liAutomatenKante#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) {$z_1$};

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

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

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



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

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

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

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

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

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

  \keys_set:nn { kellerautomat } { #2 }

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

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

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

(a, #: A)
(b, #: ε)

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

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

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

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

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

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

Turingmaschine

```
\RequirePackage{amssymb}
```

`\liTuringLeerzeichen` □

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

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

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

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

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

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

  \keys_set:nn { kellerautomat } { #2 }

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

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

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

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

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

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

Makro-Faulenzer: `\let\t=\liTuringKante`

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

`\liTuringUeberfuehrung`

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

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

basis.sty

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

```
\RequirePackage{xparse}
```

```
\ExplSyntaxOn
```

IFs

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

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

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

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

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

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

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

`\liLadePakete`

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

`\liLadeAllePakete`

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

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

Definition einer Komma-getrennten-Liste mit deren Hilfe die vielen globalen Token-List-Variablen definiert werden können. Die einzelnen Schlüssel sind im Interface `AufgabenMetadaten` in der Typescript-Datei `.scripts/nodejs/src/aufgaben.ts` erklärt.

```
\clist_new:N \g_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
   ]
  }
}
```

```
\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}
```

845

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}
```

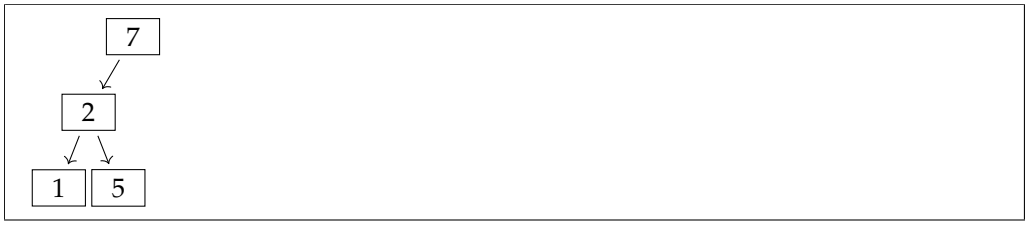
Für b baum

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

Binärbaum

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

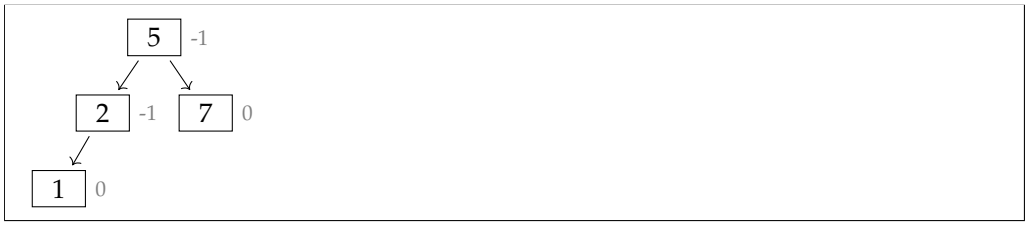
```
\begin{tikzpicture}[li binaer baum]
\Tree
[.7
  [.2
    [.1 ]
    [.5 ]
  ]
  \edge[blank]; \node[blank]{};
]
\end{tikzpicture}
```




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

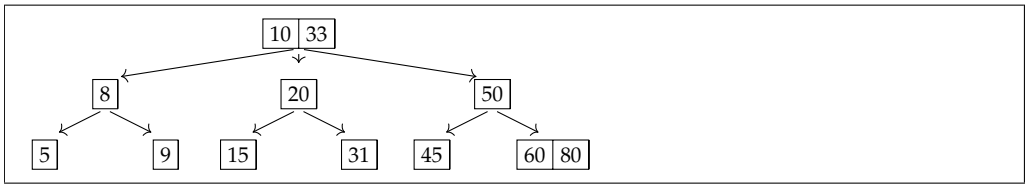
AVL-Baum

```
\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}
```

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

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

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

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

chomsky-normalform.sty

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

```
\ExplSyntaxOn
```

```
\liLadePakete{typographie}
```

Faulenzer

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

TeX-Markup-Grundgerüst

```
\let\schrittE=\liChomskyUeberErklaerung
\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}

\liNichtsZuTun

\item \schrittE{2}

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

\item \schrittE{3}

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

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

```
% S.3 -> T4 T

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

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

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

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

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

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

\ExplSyntaxOff

```

cpm.sty

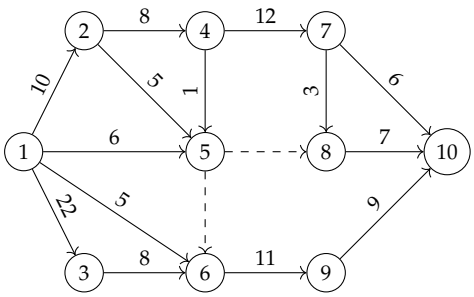
```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-cpm}[2020/09/03]
\RequirePackage{tikz}
```

```
\liLadePakete{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]
\liCpmEreignis{1}{0}{2}
\liCpmEreignis{2}{1}{4}
\liCpmEreignis{3}{1}{0}

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

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

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“

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

```
3   &                          & 18  \\
4   &                          & 7   \\
5   &                          & 19  \\
6   &                          & 26  \\
7   & $\max(19_3, 22_4)$       & 22  \\
8   & $\max(30_5, 30_6, 28_7)$ & 30  \\\hline
\end{tabular}
```

TeX-Markup-Beispiel: Nebenrechnungstabelle „Spätester Zeitpunkt“

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

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

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

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

  \keys_set:nn { cpmEreignis } { #1 }

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

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

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

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

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

  \keys_set:nn { cpmVorgang } { #1 }

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

Hilfsmakros für Zeitpunkt-Tabelle

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

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

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

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

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

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

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

```
\ExplSyntaxOn
```

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

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

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

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

```
  \fi
}
```

\liCpmFruehErklaerung

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

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

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

    \textbf{Erläuterungen:}~

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

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

\liCpmSpaetErklaerung

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

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

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

    \textbf{Erläuterungen:}~

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

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

\ExplSyntaxOff
```

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}
\liWortInSprache{acbcab}
```

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

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

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

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

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

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

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

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

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

Reihenfolge

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

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

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

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

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

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

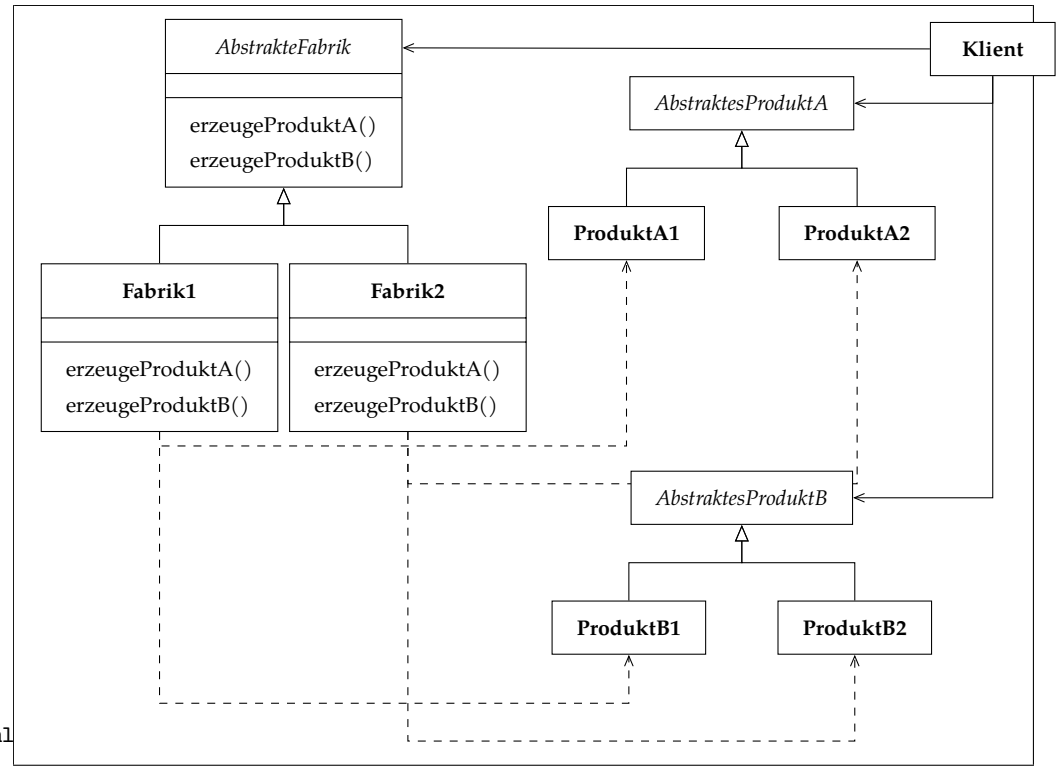
`\li@EntwurfsCodeAllgemein` Allgemeine Code-Beispiele zu den UML-Diagrammen und Stellvertretern

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

Abstrakte Fabrik (Abstract Factory)

`sAbstrakteFabrikBeschreibung`

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



\liEntwurfsAbstrakteFabrikUml

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

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

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

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

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

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

```

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

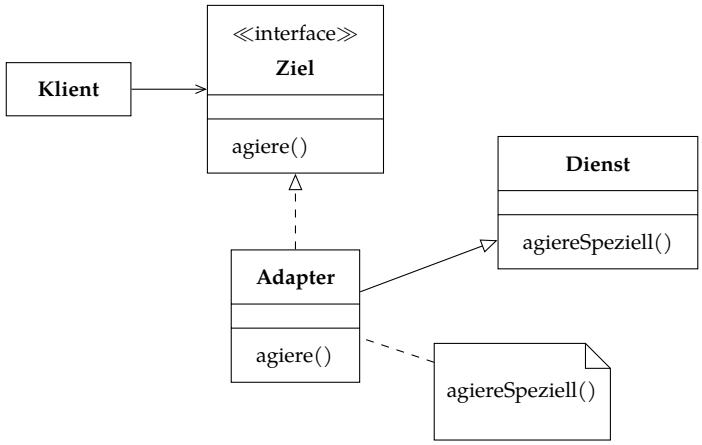
iEntwurfsAbstrakteFabrikCode

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

\liEntwurfsAbstrakteFabrik

```
\def\liEntwurfsAbstrakteFabrik{
  \liEntwurfsAbstrakteFabrikBeschreibung

  \liEntwurfsAbstrakteFabrikUml

  \liEntwurfsAbstrakteFabrikCode
}
```

Adapter



\liEntwurfsAdapterUml

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

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

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

\liEntwurfsAdapterAkteure

Ziel (Target) Das Ziel definiert die Schnittstelle, die der Klient nutzen kann.

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

Dienst (Adaptee) Der Dienst bietet wiederzuverwendende Dienstleistungen mit

fest definierter Schnittstelle an.

Adapter Der Adapter adaptiert die Schnittstelle des Dienstes auf die Schnittstelle zum Klienten.

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

    \item[Ziel (Target)]

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

    \item[Klient (Client)]

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

    \item[Dienst (Adaptee)]

    Der Dienst bietet wiederzuverwendende Dienstleistungen mit fest
    definierter Schnittstelle an.

    \item[Adapter]

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

  \end{description}
}
```

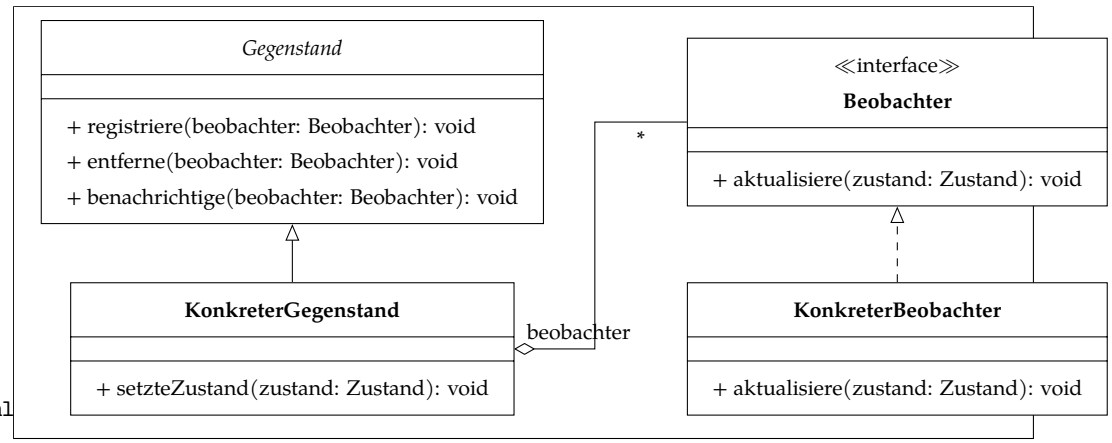
\liEntwurfsAdapterCode

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

\liEntwurfsAdapter

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

Beobachter (Observer)



\liEntwurfsBeobachterUml

```
\def\liEntwurfsBeobachterUml{
```

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

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

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

\liEntwurfsBeobachterAkteure

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

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

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

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

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

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

    \item[Beobachter (Observer)]

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

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

```

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

    \item[Konkrete Beobachter (ConcreteObserver)]

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

\liEntwurfsBeobachterCode

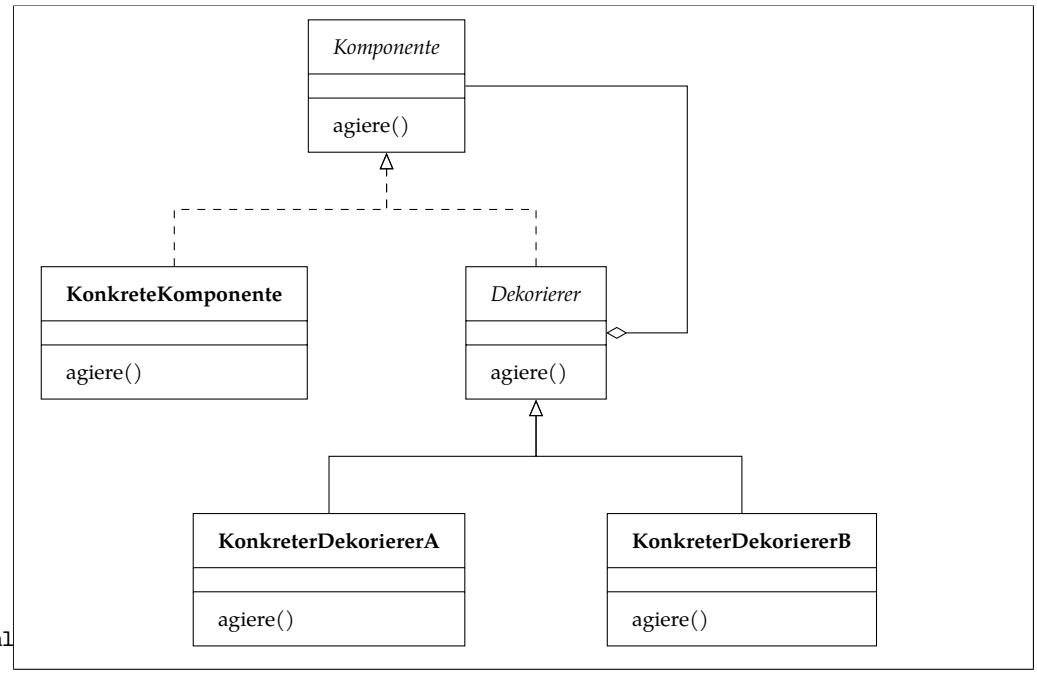
```
\def\liEntwurfsBeobachterCode{
  \li@EntwurfsCode{beobachter}{Gegenstand}
  \li@EntwurfsCode{beobachter}{KonkreterGegenstand}
  \li@EntwurfsCode{beobachter}{Beobachter}
  \li@EntwurfsCode{beobachter}{KonkreterBeobachterA}
  \li@EntwurfsCode{beobachter}{KonkreterBeobachterB}
  \li@EntwurfsCode{beobachter}{Klient}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

\liEntwurfsBeobachter

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

Dekorierer (Decorator)

\liEntwurfsDekoriererUml




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

```
    \umlclass[below right=1.5cm and 0cm of Komponente,type=abstract]{Dekorierer}{}{agiere()}

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

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

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

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

\liEntwurfsDekoriererCode

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

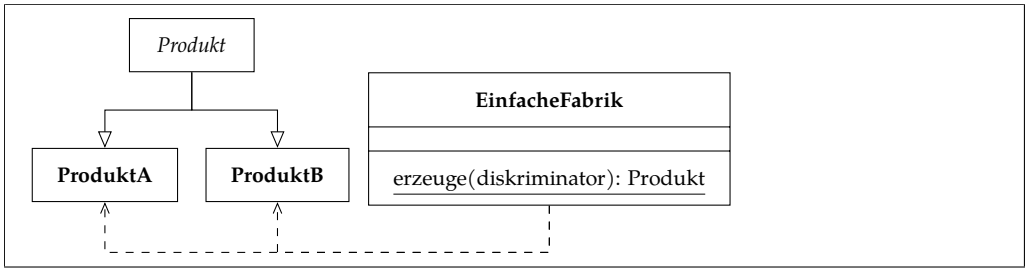
\liEntwurfsDekorierer

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

Einfache Fabrik (Simple Factory)

\liEntwurfsEinfacheFabrikUml

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




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

ntwurfsEinfacheFabrikAkteure

Quelle: frei überstetzt aus GoF

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

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

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

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

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

    \item[Produkt]

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

    \item[KonkretesProdukt]

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

\liEntwurfsEinfacheFabrik

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

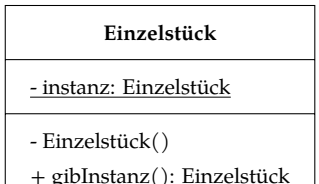
Einzelstück (Singleton)

urfsEinzelstueckBeschreibung

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

\liEntwurfsEinzelstueckUml

Quelle nach der deutschen Wikipedia




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

iEntwurfsEinzelstueckAkteure

Quelle: frei überstetzt aus GoF

Einzelstück (Singleton) stellt eine statische Methode bereit, mit deren Hilfe die Klienten nur auf eine einzige Instanz der Klasse zugreifen können.

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

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

\liEntwurfsEinzelstueckCode

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

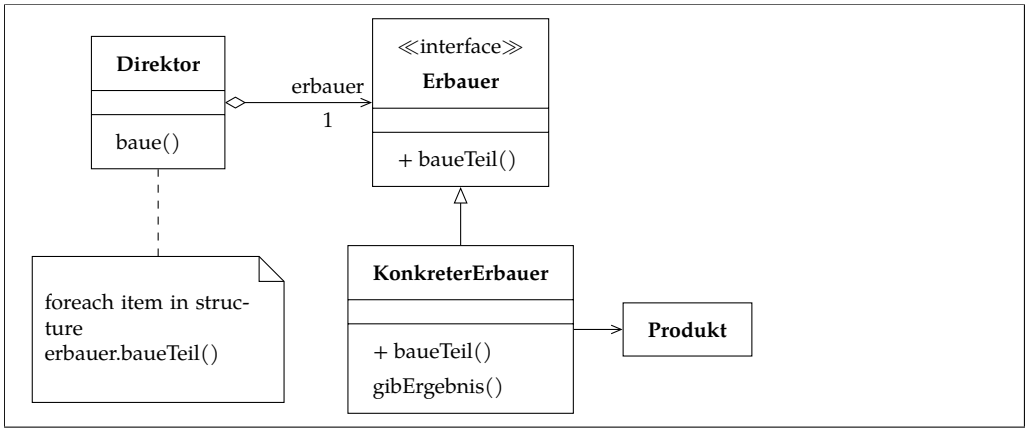
\liEntwurfsEinzelstueck

```
\def\liEntwurfsEinzelstueck{
  \liEntwurfsEinzelstueckBeschreibung

  \liEntwurfsEinzelstueckUml

  \liEntwurfsEinzelstueckAkteure

  \liEntwurfsEinzelstueckCode
}
```

Erbauer (Builder)

\liEntwurfsErbauerUml

Quelle nach der deutschen Wikipedia




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

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

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

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

\liEntwurfsErbauerAkteure

Quelle: deutsche Wikipedia

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

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

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

Produkt Das Produkt repräsentiert das zu konstruierende komplexe Objekt.

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

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

    \item[KonkreterErbauer]

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

    \item[Direktor]

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

    \item[Produkt]

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

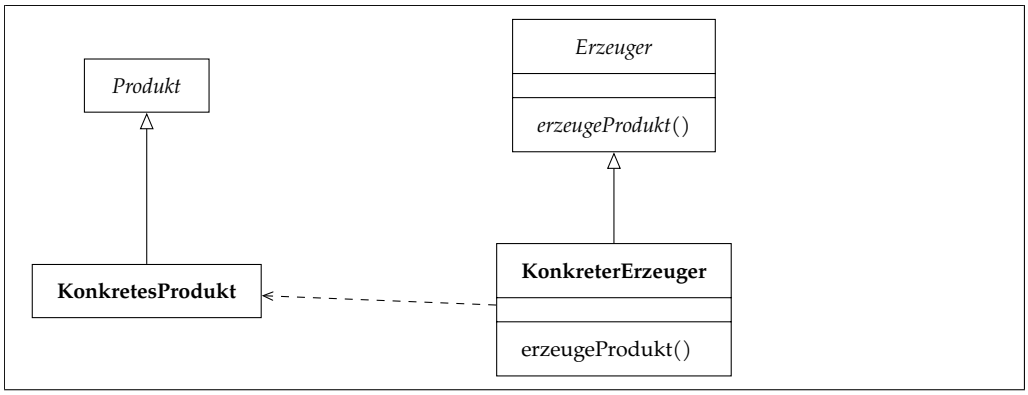
\liEntwurfsErbauer

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

Fabrikmethode (Factory Method)

\liEntwurfsFabrikmethodeUml

Quelle nach der deutschen Wikipedia




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

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

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

`EntwurfsFabrikmethodeAkteure` Quelle: deutsche Wikipedia

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

KonkretesProdukt KonkretesProdukt implementiert die Produkt-Schnittstelle.

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

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

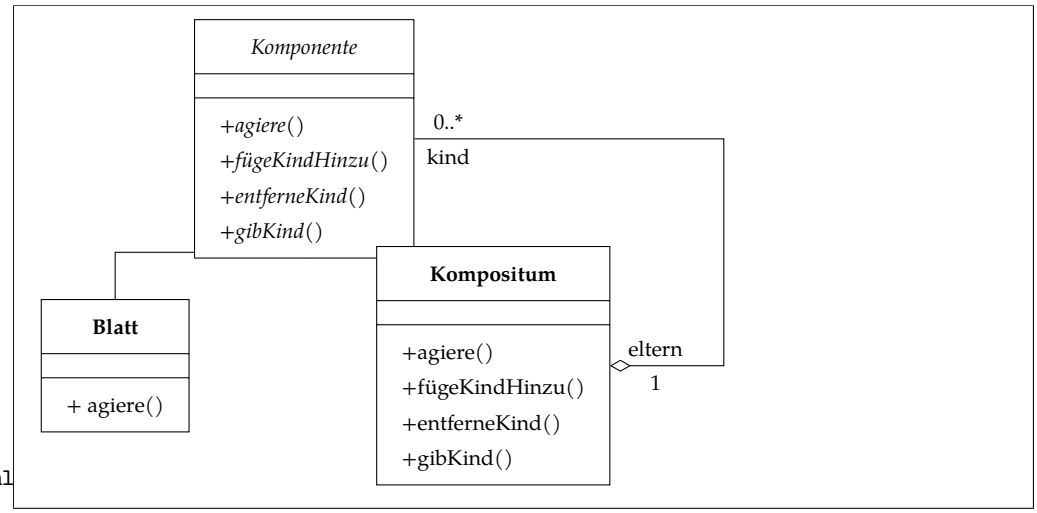
```
\def\liEntwurfsFabrikmethodeAkteure{
  \begin{description}
    \item[Produkt]

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

    \item[KonkretesProdukt]

    KonkretesProdukt implementiert die Produkt-Schnittstelle.

    \item[Erzeuger]

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

    \item[KonkreterErzeuger]

```

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

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

\liEntwurfsFabrikmethode

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

Kompositum (Composite)



\liEntwurfsKompositumUml

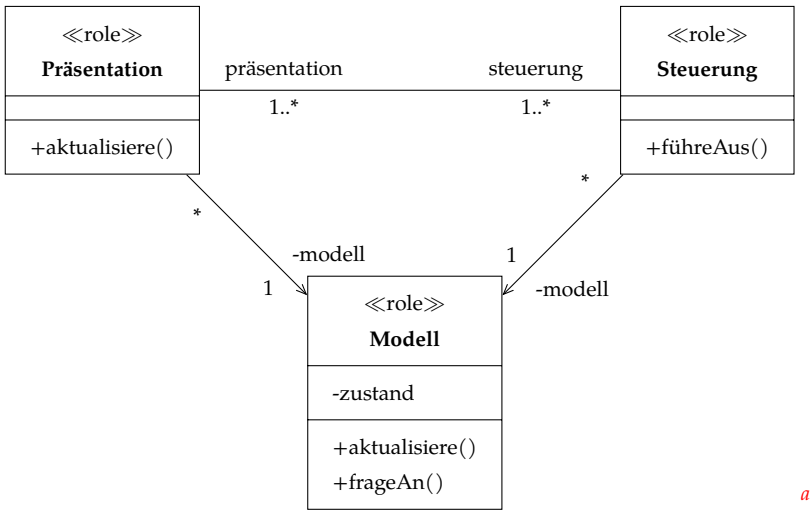
```
\def\liEntwurfsKompositumUml{
  \begin{tikzpicture}
    \umlclass[x=2.5,y=3,type=abstract]{Komponente}{}{
      \textit{+agiere()}\\
      \textit{+fügeKindHinzu()}\\
      \textit{+entferneKind()}\\
      \textit{+gibKind()}
    }
    \umlclass[x=0]{Blatt}{}{+ agiere()}
    \umlclass[x=5]{Kompositum}{}{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

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

\liEntwurfsFabrikmethode

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

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

odellPraesentationSteuerungUml



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

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

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

ModellPraesentationSteuerung

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

Stellvertreter (Proxy)

\liEntwurfsStellvertreterUml

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

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

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

liEntwurfsStellvertreterCode

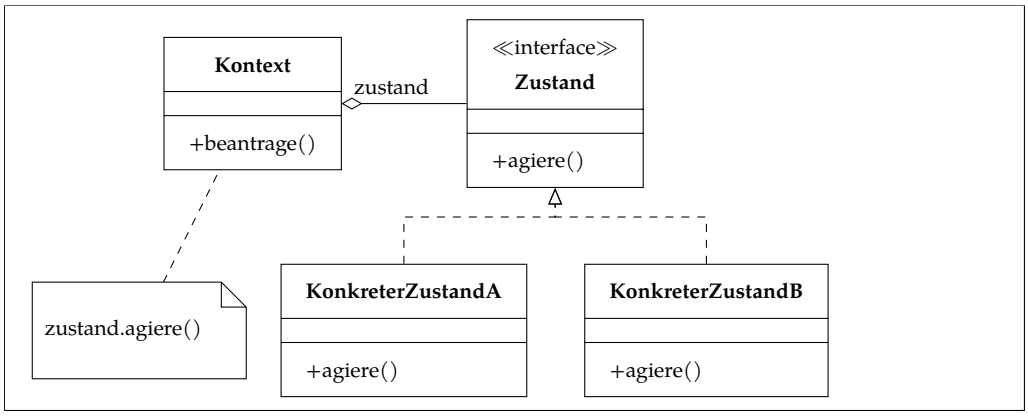
```
\def\liEntwurfsStellvertreterCode{
  \li@EntwurfsCode{stellvertreter}{Subjekt}
  \li@EntwurfsCode{stellvertreter}{KonkretesSubjekt}
  \li@EntwurfsCode{stellvertreter}{Stellvertreter}
  \li@EntwurfsCode{stellvertreter}{Klient}
}
```

`\liEntwurfsStellvertreter`

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

Zustand (State)

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




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

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

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

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

`\liEntwurfsZustandAkteure` Quelle: Deutsche Wikipedia

Kontext (Context) definiert die clientseitige Schnittstelle und verwaltet die separaten Zustandsklassen.

State (Zustand) definiert eine einheitliche Schnittstelle aller Zustandsobjekte und implementiert gegebenenfalls ein Standardverhalten.

KontreterZustand (ConcreteState) implementiert das Verhalten, das mit dem Zustand des Kontextobjektes verbunden ist.

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

    definiert die clientseitige Schnittstelle und verwaltet die separaten
```

```
    Zustandsklassen.

    \item[State (Zustand)]

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

    \item[KontreterZustand (ConcreteState)]

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

\liEntwurfsZustandCode

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

\liEntwurfsZustand

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

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
```

`\liErEntity`

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

`\liErRelationship`

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

`\liErAttribute`

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

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

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

□

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

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

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

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

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

```
\def\liErDatenbankName#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}
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`\liZustandsBuchstabe`

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

`\liZustandsBuchstabeGross`

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

\liZustandsmengeNr

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

\liZustandsmengeNrGross

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

\liZustandsname

\liZustandsname{1}: z_1

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

\liZustandsnameGross

\liZustandsnameGross{1}: Z_1

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

\liAbleitung

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

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

liProduktionsRegeln

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

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

\liProduktionen

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

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

\liZustandsnameTiefgestellt

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

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

```
\ExplSyntaxOn
```

\liAusdruck

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

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

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

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

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

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

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

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

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

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

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

  \keys_set:nn { grammatik } { #2 }

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

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.

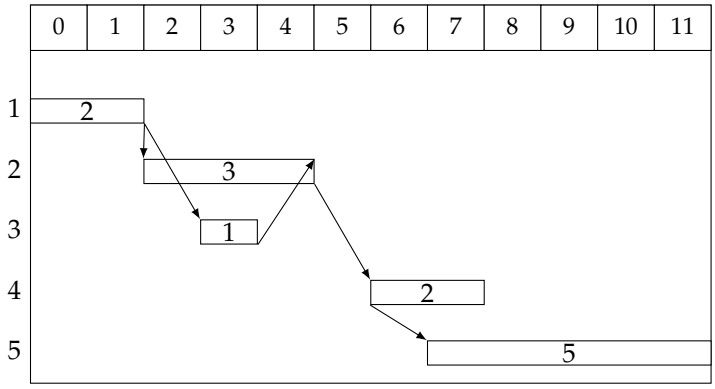
```
\RequirePackage{setspace}
```

1790

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}
```

\liGrafikLogoPfad

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

\liGrafikCCLizenz

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

\liGrafikLogo

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

```
\ExplSyntaxOff
```

graph.sty

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

Für die die Adjazenzliste (xrightarrow)

```
\RequirePackage{amsmath}
```

Für Adjazenz-Matrix

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

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

```
\RequirePackage{blkarray}
```

```
\usetikzlibrary{arrows.meta}
```

```
\begin{tikzpicture}[li graph]
\node (a) at (0,0) {a};
\node (b) at (1,1) {b};
\node (c) at (4,1) {c};
\node (d) at (3,0) {d};
\node (e) at (3,3) {e};

\path[->] (a) edge (b);
\path (b) edge (d);
\path[->] (b) edge node {3} (e);
\path (c) edge (d);
\path (d) edge node {4} (a);
\path[->] (d) edge node {2} (e);
\end{tikzpicture}
```

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

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

```
\begin{liGraphenFormat}
a: 0 0
b: 1 1
c: 4 1
d: 3 0
e: 2 2
a -> b
b -- d
b -> e: 3
c -- d
d -> e: 2
d -- a: 4
\end{liGraphenFormat}
```

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

hanoi.sty

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

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

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

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

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

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

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
```

```
\liLadePakete{mathe}
```

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

```
\RequirePackage{mdframed}
```

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

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

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

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

`\liProblemBeschreibung` Zu setzen von Problem-Beschreibungen:

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

CLIQUE

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

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

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

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

    \medskip

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

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

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

\liProblemVertexCover

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

\liProblemVertexCover

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

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

\liProblemSubsetSum Kein Karp-21-Problem

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

\liProblemSubsetSum Kein Karp-21-Problem

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

```

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

\liAnweisung

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

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

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

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

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

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

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

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

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

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

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

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

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

kopf-fusszeilen.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-kopf-fusszeilen}[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{\liMetaBschlangaulSammlung}}
  \fancyhead[C]{\kopfzeile_schrift:n{#1}}
  \fancyhead[R]{\kopfzeile_schrift:n{#2}}

  \fancyfoot{}
  \fancyfoot[L]{\kopfzeile_schrift:n{\liMetaHermineFriends}}
  \fancyfoot[C]{\kopfzeile_schrift:n{\liMetaEmail}}
  \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`\\}
```

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

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

  \par
  \noindent
  \textbf{#1}

  \medskip

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

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

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

`\hinweis`

```
\def\hinweis#1{{\footnotesize[#1]}}
```

`liProjektSprache` `\begin{liProjektSprache}{NameProjektSprache} \end{liProjektSprache}`: Zum Einbetten von projekteigenen Minisprachen/DSLs (Domain-specific language) (z. B.RelationenSchema). Der Inhalt der Umgebung wird nicht von TeX kompiliert, sondern von dem Java-Kommandozeilen-Programm `didaktik.java` verarbeitet.

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

liEinbettung

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

Umgebungen, die Inhalte aus- und einblenden können

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

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

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

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

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

liAntwort

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

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

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

```
  \fi
  \begin{mdframed}[
    frametitle={\scriptsize\mdseries Lösungshinweise},
    innertopmargin=6pt,
    frametitleaboveskip=-12pt,
    frametitlealignment=\raggedright
  ]
}
{
  \end{mdframed}
  \ifLiANTWORT
  \else
    \egroup
  \fi
}
```

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

```
\NewDocumentEnvironment{ liAdditum } { o }
{
  \ifLiADDITUM
  \else
    \setbox 0 \vbox
    \bgroup
  \fi

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

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

`liExkurs`

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

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

```
\NewDocumentEnvironment{ liExkurs }{o +b}{
  \ifLiEXKURS
    \vspace{0.2cm}%
    \begin{mdframed}[
      backgroundcolor=white,
      bottomline=false,
      innermargin=1cm,
      leftline=true,
```

```
      linecolor=black,
      linewidth=0.1cm,
      outermargin=1cm,
      rightline=false,
      topline=false,
    ]
    \footnotesize
    \noindent%
    \textbf{Exkurs:~#1}\par%
    \noindent%
    #2
    \end{mdframed}
    \vspace{0.2cm}
  \else
  \fi
}{}
```

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

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

Weiterführende Literatur:

- Quelle 1
- Quelle 2

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

`liLernkartei`

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

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

```
\NewDocumentEnvironment { liDiagramm }{ m +b }
{
  \begin{mdframed}[backgroundcolor=white!0]
  \small
  \noindent%
  \textit{#1}:
  \begin{center}
  #2
  \medskip
  \end{center}
  \end{mdframed}
} {}
```

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

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

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

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

`\zB`

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

`\ZB`

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

`\dh`

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

```
\ExplSyntaxOff
```

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


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

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

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


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

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

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


\ExplSyntaxOn
\RequirePackage{amsmath}
```

\liRundeKlammer

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

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

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

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

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

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

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

\liT **Makro-Faulenzer:** \let\T=\liT
\liT{16}{2}: $16 \cdot T\left(\frac{n}{2}\right)$ \liT{}{2}: $T\left(\frac{n}{2}\right)$

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

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

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

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

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

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

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

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

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

```
\ExplSyntaxOff
```

\liMasterVariablen

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

  \begin{itemize}
  \item[$a =$]
  Anzahl der rekursiven Aufrufe, Anzahl der Unterprobleme in der
  Rekursion
  ($a \geq 1$).

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

  \item[$f(n) =$]
  Kosten (Aufwand, Nebenkosten), die durch die Division des Problems und
  die Kombination der Teillösungen entstehen. Eine von $T(n)$
  unabhängige und nicht negative Funktion.
  \end{itemize}
  \footcite{wiki:master-theorem}
  \footcite[Seite 19-35]{aud:fs:2}
}
```

\liMasterFaelle

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

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

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

  \hfill falls \liBedingungZwei

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

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

liMasterVariablenDeklaration

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

    \liRekursionsGleichung

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

    #1

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

```

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

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

    $#3$

    \item[Ergibt folgende Rekursionsgleichung:] \strut

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

\liMasterFallRechnung

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

  #1

  \item[2. Fall:] \liBedingungZwei:

  #2

  \item[3. Fall:] \liBedingungDrei:

  #3
  \end{description}
}
```

\liMasterExkurs

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

  \noindent
  Dann gilt:

  \liMasterFaelle
  \end{liExkurs}
}
```

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

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

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
```

```
\liLadePakete{grafik}
```

Einfache Makros (Low level)

\liMetaBschlangaulSammlung

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

\liMetaHermineFriends

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

\liMetaUeberDasProjekt

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

\liMetaCCLink

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

\liMetaEmail

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

\liMetaEmailLink

```
\def\liMetaEmailLink{
  \href{
    mailto:\liMetaEmail
  }{
    \liMetaEmail
  }
}
```

\liMetaHilfMit

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

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

  Das~ist~ein~Community-Projekt.~

  Verbesserungsvorschläge,~Fehlerkorrekturen,~weitere~Lösungen~sind~
```

```
  herzlich~willkommen~~~egal~wie~~~per~Pull-Request~oder~per~E-Mail~an~
  \liMetaEmailLink.
}
```

\liMetaHilfMit

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

Zusammengesetzte Makros (High level)

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

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

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

\liLogoTextProjekt

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

    \liMetaHermineFriends
    \par

    \medskip

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

\liLogoTextCCLizenz

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

```
    \end{spacing}
  }
}

\ExplSyntaxOff

```

minimierung.sty

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

```
\liLadePakete{typographie}
```

```
\let\z=\liZustandsnameTiefgestellt
\let\f=\liFussnote
\let\l=\liLeereZelle
\let\Z=\liZustandsPaar
\let\erklaerung=\liMinimierungErklaerung

\begin{tabular}{|c||c|c|c|c|c|c|c|c|}
\hline
\z0 & \l  & \l  & \l  & \l  & \l  & \l  & \l  & \l  & \l  \\ \hline
\z1 &     & \l  & \l  & \l  & \l  & \l  & \l  & \l  & \l  \\ \hline
\z2 &     &     & \l  & \l  & \l  & \l  & \l  & \l  & \l  \\ \hline
\z3 &     &     &     & \l  & \l  & \l  & \l  & \l  & \l  \\ \hline
\z4 &     &     &     &     & \l  & \l  & \l  & \l  & \l  \\ \hline
\z5 &     &     &     &     &     & \l  & \l  & \l  & \l  \\ \hline
\z6 &     &     &     &     &     &     & \l  & \l  & \l  \\ \hline
\z7 &     &     &     &     &     &     &     & \l  & \l  \\ \hline
\z8 &     &     &     &     &     &     &     &     & \l  \\ \hline\hline
    & \z0 & \z1 & \z2 & \z3 & \z4 & \z5 & \z6 & \z7 & \z8 \\ \hline
\end{tabular}

\liFussnoten

\begin{liUebergangsTabelle}{0}{1}
\Z01 & \Z10 & \Z23     \\
\Z05 & \Z15 & \Z25 \f2 \\
\Z15 & \Z05 & \Z35 \f2 \\
\Z23 & \Z44 & \Z55     \\
\Z24 & \Z44 & \Z55     \\
\Z34 & \Z44 & \Z55     \\
\end{liUebergangsTabelle}
```

\liFussnote

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

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

\liFussnoteEinsText

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

\liFussnoteZweiText

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

\liFussnoteDreiText

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

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

\liFussnoteVierText

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

\liFussnoten

x_1	Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.
x_2	Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt.
x_3	In weiteren Iterationen markierte Zustände.
x_4	...

```
\def\liFussnoten{
  \bigskip

  \noindent
  \liFussnoteEinsText

  \noindent
  \liFussnoteZweiText

  \noindent
  \liFussnoteDreiText

  \noindent
  \liFussnoteVierText
}
```

\liLeereZelle

\liLeereZelle: $\emptyset$

Makro-Faulenzer: \let\l=\liLeereZelle

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

\liZustandsPaarVariablenName

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

\liZustandsPaar

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

liUebergangsTabelle

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

iUeberschriftDreiecksTabelle

\liUeberschriftDreiecksTabelle:

Minimierungstabelle (Table filling)

```
\ExplSyntaxOn
```

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

\liMinimierungErklaerung

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\liMinimierungErklaerung{
  %\footcite[Seite~19]{koenig}
  \liParagraphMitLinien{
    Der~Minimierungs-Algorithmus~(auch~Table-Filling-Algorithmus~genannt)~
    trägt~in~seinem~Verlauf~eine~Markierung~in~alle~
    diejenigen~Zellen~der~Tabelle~ein,~die~zueinander~nicht~äquivalente~
    Zustände~bezeichnen.~Die~Markierung~„$x\sb{n}$"~in~einer~Tabellenzelle~($i$,~
    $j$)~bedeutet~dabei,~dass~das~Zustandspaar~($i$,~$j$)~in~der~$k$-ten~
    Iteration~des~Algorithmus~markiert~wurde~und~die~Zustände~$i$~und~$j$~
    somit~zueinander~($k~-~1$)-äquivalent,~aber~nicht~$k$-äquivalent~und~
    somit~insbesondere~nicht~äquivalent~sind.~Bleibt~eine~Zelle~bis~zum~Ende~
    unmarkiert,~sind~die~entsprechenden~Zustände~zueinander~äquivalent.
  }
}
\ExplSyntaxOff
```

normalformen.sty

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

Mathe für textit tex etc, typographie für geschweifte Klammer

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

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

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

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

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

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

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

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

`liAHuelle`

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

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

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

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

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

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

\liLinksReduktionInline

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

\liLinksReduktionInline

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

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

\liFunktionaleAbhaengigkeit

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

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

liFunktionaleAbhaengigkeiten

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

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

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

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

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

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

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

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}
```

\liRundeKlammer

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

\liONotationO

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

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

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

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

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

\liPetriSetzeSchluessel

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

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

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

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

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

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

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

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

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

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

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

potenzmengen-konstruktion.sty

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

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

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

\begin{tabular}{l|l|l}
Zustandsmenge & Eingabe $a$ & Eingabe $b$ \\\hline
\z0 & \z0 & \z1 \\
\z1 & \z2 & \z1 \\
\z2 & \z0 & \z3 \\
\z3 & \z4 & \z3 \\
\z4 & \z5 & \z3 \\
\z5 & \z5 & \z3\\
\end{tabular}
```

`\liZustandsMengenSammlung` `\liZustandsMengenSammlung{nummer}{latex3 str_case:nn}`

```
\def\z#1{
  \liZustandsMengenSammlung{#1}{
    {
      {0} {z0}
      {1} {z0, z1}
      {2} {z0, z1, z2}
      {3} {z0, z2}
      {4} {z0, z1, z2, z3}
      {5} {z0, z3}
      {6} {z0, z2, z3}
      {7} {z0, z1, z3}
    }
  }
}
```

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

`\liZustandsMengenSammlungNr`

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

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

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]
```

\liPumpingRegulaer

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

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

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

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

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

\liPumpingKontextfrei

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

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

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

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

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

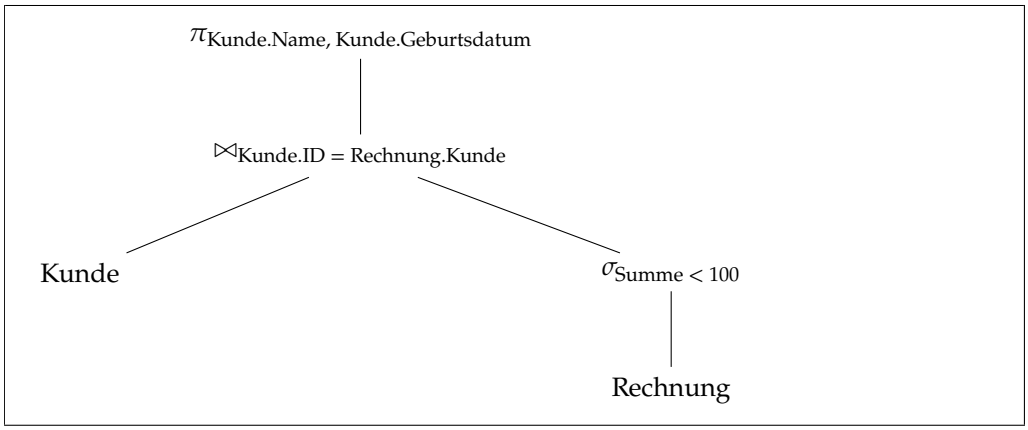
```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-relationale-algebra}[2020/12/11]
```

```
\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
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

sortieren.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-sortieren}[2020/06/10 Schaubilder
für händisches Sortieren bzw. einen Schreibtischlauf setzen]
```

```
\tikz[
  rectangle split parts=5,
]{
  \node[li sortierung zahlenreihe] (reihe) {\nodepart{one} 2 \nodepart{two} 1 \nodepart{three} 3
  \liSortierPfeil{one}{two}
  \liSortierPfeil{two}{three}
  \liSortierMarkierung{two split south}{three split north}
  \liSortierPfeilUnten{three}{one}
}
```

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

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

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

`\liSortierPfeil`

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

`\liSortierPfeilUnten`

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

`\liSortierMarkierung`

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

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

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

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

\def\myNodes{}

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

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

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

spalten.sty

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

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

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

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}
```

```
\liLadePakete{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
}
```

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

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

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

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

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

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

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

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

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

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

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

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

`\liAssemblerCode`

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

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

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

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

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

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

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

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

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

```
\ExplSyntaxOff
```

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

```
\def\liSqlCode#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]
```

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

Faulenzer

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

TeX-Markup Grundgerüst

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

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

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

TeX-Markup Linksreduktion

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

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

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

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

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

TeX-Markup Rechtreduktion

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

\liPseudoUeberschrift{F}

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

\liPseudoUeberschrift{A}

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

TeX-Markup Relationen formen

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

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

(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. $R_\alpha \subseteq R_{\alpha'}$.*

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

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

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

```
}
```

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

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

`\liErledigt`

`\liErledigt`: ☑

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

`\liNichtsZuTun`

`\liNichtsZuTun`: ∅ Nichts zu tun

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

`\liParagraphMitLinien`

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

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

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

`\liGeschweifteKlammern`

Große geschweifte Klammer mit Istgleich-Zeichen.

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

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

`\liTypoUeberschrift`

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

\liTypoUeberGross

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

\liTypoUeberGross

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

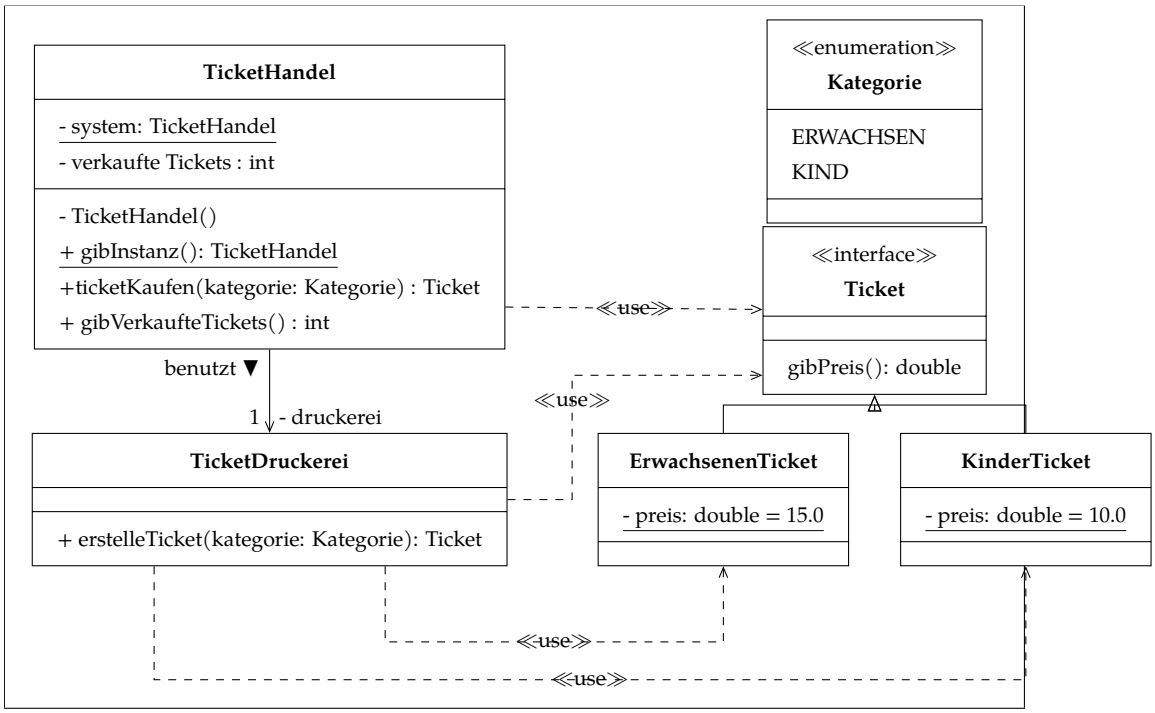
```
\ExplSyntaxOff
```

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,
}
```



\liUmlLeserichtung

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

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

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

  \def\@liDistance{0cm}
```

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

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

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

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
```

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

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

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

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

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

`\liInduktionAnfang`

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

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

`\liInduktionVoraussetzung`

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

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

`\liInduktionSchritt`

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

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

```
\ExplSyntaxOff
```

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
```

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

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

\MatheEnv

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

  \hspace{1em}#1

  \medskip
}
```

\Mathe

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

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

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

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

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

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

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

\liWpErklaerungVerzweigung

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

```
\ExplSyntaxOff
```

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

\# 460
\, . . 710, 780, 835, 1062, 1080, 1712, 1714, 2295, 2296, 2297, 2737, 3238, 3245, 3478
\@Skip@Erklaerung@Reset 3631, 3633, 3644
\@afterheading 82, 2140, 2264
\@afterindentfalse 81, 2139, 2263
\@liDirLeft 3526, 3531, 3543
\@liDirRight 3527, 3529, 3530, 3531, 3532, 3543
\@liDistance 3537, 3538, 3542
\@liPos . . . 3534, 3535, 3542
\\ . . . 146, 148, 172, 176, 180, 1092, 1120, 1121, 1124, 1125, 1128, 1129, 1224, 1225, 1226, 1333, 1367, 1369, 1399, 1408, 1453, 1495, 1496, 1497, 1502, 1503, 1504, 1524, 2127, 2632, 2690, 2693
\{ 558, 1649, 1659, 1671, 1672, 1677, 1711, 1943, 2677, 3102, 3478, 3647
\} 558, 1649, 1659, 1671, 1672, 1679, 1715, 1944, 2677, 3102, 3482, 3647
_ . 48, 62, 105, 234, 383, 391, 396, 398, 657, 685, 709, 712, 722, 745, 748, 752, 754, 756, 758, 761, 773, 774, 775, 778, 780, 794, 795, 801, 804, 807, 817, 824, 826, 829
\␣ 2689, 2699

A

\addbibresource 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105
\AddToHook . 85, 89, 292, 327
\advance 3151
\AfterEndEnvironment 3227
\Alph 1772
\alph 1772, 1773
\alpha . 3346, 3348, 3349, 3352, 3354, 3355, 3356, 3357, 3358, 3362, 3368, 3369, 3374, 3375, 3376, 3379, 3387, 3388, 3389
\arabic 1772, 2946, 2951, 2956, 2962, 2968, 2974
\arraystretch 2627
\AtBeginDocument 169

B

\BeforeBeginEnvironment 3226
\begin 99, 144, 161, 239, 1118, 1172, 1187, 1222, 1246, 1293, 1325, 1340, 1365, 1375, 1395, 1415, 1447, 1464, 1493, 1518, 1539, 1563, 1578, 1692, 1777, 1869, 1897, 1906, 1913, 2021, 2119, 2183, 2205, 2225, 2252, 2257, 2268, 2278, 2282, 2365, 2369, 2388, 2411, 2434, 2449, 2529, 2530, 2534, 2557, 2570, 2630, 2631, 2681, 2726, 2866, 2886, 3027, 3049, 3063, 3226, 3249, 3417, 3482
\begingroup 2120, 2679, 3147
\beschriftung 2163, 2167, 2171, 2175, 2179, 2181
\beta 3346, 3349, 3353, 3354, 3355, 3358, 3368, 3369, 3370, 3375, 3376
\bf 2924, 2925, 2926
\bFaulenzer 3441
\bfseries 55, 135, 200, 257, 921, 1762, 1764, 2547, 2924, 2930, 2932, 2934, 2935, 3323, 3487
\bgroup 2158, 2202
\Bigl 3478
\Bigr 3482
\bigskip 77, 94, 97, 250, 302, 305, 813, 1094, 1099, 2129, 2605, 3452
\bMakroFaulenzer . . . 3422
\bool 688, 781
\bowtie 3072, 3075, 3076, 3077
\Box 510
\boxtimes 911

C

\c 1871, 1872
\cdot 2340, 2396, 2407
\centerline 1902, 2567, 3025, 3047, 3062
\cftbeforesecskip . . . 283
\cftbeforesubsecskip 284, 285
\cftbeforesubsubsecskip 286

\cftsubsecafterpnum . 285
\chapter 1762, 1763
\char 2127
\clearpage 141, 155, 158, 166, 201, 295, 307, 2121
\cline 1092
\clist 583, 631, 632, 654, 658, 3159
\columnbreak 3194
\contentsname 287
\cs 19, 26, 32, 236, 280, 657, 685, 709, 712, 722, 745, 756, 767, 778, 780, 817, 824, 829, 2048, 2055, 2070, 2076, 2246, 2528, 2749, 3422
\csname 1856, 1859
\cup 1672, 2711, 3355, 3369, 3376

D

\DeclareMathSymbol 2473, 2474
\DecoINERT 2949, 3022, 3023, 3045
\DecoINERTwithPivot 2965, 3044
\DecoLEFT 2944, 3043
\DecoLEFTwithPivot 2959, 3021
\DecoRIGHT 2954, 3046, 3056
\DecoRIGHTwithPivot 2971, 3024
\definecolor 1760
\DefineVerbatimEnvironment 3200
\delta 416, 458, 516, 558, 1663
\dh 2297, 3357
\directlua 409, 488, 546, 551, 1644, 1658, 1678, 1686, 1693, 1698, 2662, 2667, 2720, 2727, 2734, 3114, 3213, 3257, 3262, 3263, 3267, 3272, 3273, 3277, 3283, 3284
\do 2945, 2950, 2955, 2960, 2966, 2972
\dots 952, 956, 1943, 2873, 3368, 3369
\DOWNarrow 3530
\draw 1876, 1879, 1882, 2821, 3120, 3123

E

\edef 1867, 3038, 3042, 3054, 3055
\edge 628
\egroup 2194, 2219
\else 1016, 1024, 1032, 1040, 1047, 1653, 1667, 1703, 2156, 2179, 2193, 2200, 2218, 2243, 2313, 2323, 2333, 2347, 2673, 2710, 2755, 2816, 2984, 3177, 3179, 3614
\emph . . 1113, 1360, 1615, 1924, 1953, 1955, 2126
\empty 2179, 2709
\emptyset 2619, 3363, 3384, 3463
\end 106, 152, 163, 276, 1157, 1183, 1208, 1243, 1277, 1309, 1337, 1353, 1372, 1380, 1411, 1440, 1461, 1486, 1511, 1531, 1550, 1575, 1593, 1694, 1779, 1885, 1909, 1910, 1917, 2026, 2124, 2191, 2215, 2241, 2259, 2260, 2274, 2285, 2286, 2367, 2383, 2408, 2431, 2446, 2456, 2532, 2536, 2537, 2560, 2573, 2634, 2635, 2683, 2728, 2876, 2896, 3027, 3049, 3063, 3227, 3253, 3420, 3482
\endcsname 1856, 1859
\endgroup . 2123, 2684, 3152
\enspace 3467, 3469
environments:
liAdditum 2197
liAHuelle 2678
liAntwort 2153
liDiagramm 2276
liEinbettung 2152
liExkurs 2222
liGraphenFormat . 1848
liJavaAngabe 3234
liKasten 1776
liKontrollflussgraph 2020
liLernkartei 2266
liProduktionsRegeln 1687
liProjektSprache 2151
liQuellen 2246
liRelationenSchemaFormat 3106
liRmodell 3086
liUebergangsTabelle 2627
\equiv 3629, 3648
\erzeuge@tiefgestellt 1658, 1659, 1663
\expandafter 1856, 2980, 2982, 2983, 2984, 2992, 3150
\ExplSyntaxOff . . . 109, 183, 309, 331, 400, 443, 485, 490, 543, 548, 553, 974, 996, 1011, 1087, 1639, 1717, 1752, 1819, 2041, 2080, 2298, 2363, 2576, 2655, 2849, 3099, 3160, 3297, 3404, 3454, 3503, 3582, 3653
\ExplSyntaxOn . 18, 125, 206, 325, 381, 412, 453, 486, 511, 544, 549, 574, 917, 980, 997, 1036, 1612, 1707, 1729, 1803, 2034, 2046, 2117, 2302, 2479, 2637, 2742, 2830, 3087, 3157, 3212, 3320, 3413, 3460, 3554, 3607

F

\faCheckSquareO 3462
\faCircleThin 1631
\faGg 1625
\fancyfoot 1783, 1784, 1785, 2062, 2063, 2064, 2065
\fancyhead . 1782, 2057, 2058, 2059, 2060, 2072
\faSquareO 1619
\fi . . . 1018, 1026, 1034, 1042, 1049, 1655, 1669, 1705, 2159, 2182, 2195, 2203, 2220, 2244, 2315, 2325, 2335, 2349, 2675, 2712, 2757, 2818, 2985, 3177, 3178, 3181, 3183, 3616
\filcenter 200
\footcite 1115, 1184, 1206, 1253, 1276, 1308, 1361, 1412, 1439, 1485, 1922, 1925, 1932, 1937, 1942, 1946, 1952, 1957, 2085, 2384, 2385, 2642, 2879
\footnote 2289, 2293
\footnotesize 71, 252, 498, 790, 872, 966, 1636, 2150, 2236, 2253, 2269, 2558, 2586, 2680, 2697, 2705, 2834, 2843, 3086, 3201, 3232, 3396, 3517, 3543
\footrulewidth . 1788, 2068
\foreach . . 1871, 1874, 1881
\forestFirst . . . 3174, 3177
\forestLast . . . 3175, 3177

\forestOget ... 3174, 3175
\forestOnes 3187
\forestOv . 3176, 3177, 3180
\forestov .. 3166, 3170, 3171, 3174, 3175, 3176, 3177, 3179, 3180
\forestset 3162, 3185
\forestSortLevel 3164, 3172, 3186, 3187
\frac . 2342, 2375, 2407, 2422
\fullouterjoin 3077

G

\g 20, 22, 33, 35, 43, 57, 59, 127, 130, 146, 148, 176, 180, 208, 212, 216, 220, 221, 222, 223, 224, 226, 227, 229, 259, 261, 266, 270, 385, 631, 632, 654, 658, 664, 665, 666, 667, 668, 670, 671, 673, 674, 676, 677, 678, 679, 680, 681, 682, 683, 686, 690, 691, 692, 695, 697, 698, 699, 700, 701, 702, 746, 750, 754, 757, 758, 760, 761, 763, 764, 768, 770, 783, 784, 785, 786, 794, 795, 797, 803, 804, 806, 807, 809, 810, 822, 825, 830, 832, 836
\Gamma .. 457, 515, 558, 1672
\gappto 3182
\geometry 351
\geq 1950, 2373, 2862, 2867, 2883, 2887
\gib 75, 767

H

\hbox 3072
\headrulewidth . 1787, 2067
\headwidth 2078
\hfill 65, 2392, 2398, 2403, 3470
\hinweis 2150
\hline 2632
\horizontale 26, 73
\href 1725, 2293, 2460, 2495, 2507, 3252
\hspace ... 2824, 3621, 3629
\ht 3073
\Huge ... 150, 176, 257, 3499
\huge 200, 3493

I

\i 1881, 1882
\ifcase 2981
\ifLiADDITUM 575, 2199, 2217
\ifLiANTWORT 579, 2155, 2192
\ifLiEXKURS 577, 2223
\ifmmode 1014, 1022, 1030, 1038, 1045, 1651, 1665, 1701, 2311, 2321, 2331, 2345, 2671, 2753, 2814, 3612
\IfNoValueTF 2206, 2289, 2293
\ifnum 2908, 3149, 3165, 3173, 3179
\ifx 2179, 2709, 3177
\in 940, 1096, 1950, 2355, 2358, 2361, 2390, 2396, 2401, 2862, 2873, 2883, 2893, 3346, 3348, 3354, 3375, 3570
\includegraphics 1810, 1815
\includepdf 168
\inhalts 280, 297
\inhaltsverzeichnis 2118
\input 217, 363, 366, 369, 372, 375, 842
\inputminted 3256, 3266, 3276, 3288, 3291, 3295
\int 3159
\item ... 911, 912, 1189, 1193, 1198, 1203, 1247, 1256, 1261, 1269, 1341, 1346, 1350, 1376, 1416, 1421, 1428, 1436, 1465, 1470, 1474, 1479, 1579, 1584, 1589, 1907, 1908, 2246, 2250, 2370, 2375, 2379, 2389, 2395, 2400, 2412, 2416, 2420, 2424, 2428, 2435, 2439, 2443, 2867, 2870, 2873, 2887, 2890, 2893
\itshape 965, 3395

J

\j 1871, 1872, 1874, 1875, 1876, 1881, 1882, 1883

K

\k 1881
\keys 387, 421, 433, 463, 473, 521, 531, 662, 984, 988, 1002, 1007, 1736, 1743
\kopfzeile 2048, 2058, 2059, 2060, 2063, 2064, 2065, 2073

L

\l 414, 415, 416, 417, 418, 419, 422, 423, 424, 425, 426, 428, 430, 435, 436, 437, 438, 439, 440, 455, 456, 457, 458, 459, 460, 461, 464, 465, 466, 467, 468, 469, 470, 476, 477, 478, 479, 480, 481, 482, 513, 514, 515, 516, 517, 518, 519, 522, 523, 524, 525, 526, 527, 528, 534, 535, 536, 537, 538, 539, 540, 982, 985, 990, 991, 994, 999, 1000, 1003, 1004, 1009, 1731, 1732, 1733, 1734, 1737, 1738, 1739, 1740, 1746, 1747, 1748, 1749, 2037, 2038, 2039, 2249, 2250, 2251, 2258, 3440, 3443, 3444, 3446
\labelenumi 1773
\labelenumii 1774
\labelitemi 1767
\labelitemii 1768
\labelitemiii 1769
\labelitemiv 1770
\land 3649, 3651
\LARGE .. 146, 180, 266, 1762
\Large 56, 172
\large 47, 148, 240, 1902, 3136
\leaders 3470
\left 2305, 2747
\LEFTarrow 3531
\leftarrow 1028
\leftouterjoin 3075
\leftskip . 3633, 3634, 3644
\LehramtInformatikGitBranch 821, 3220
\LehramtInformatikGithubCodeRepo 3219
\LehramtInformatikGithubDomain 818, 3216
\LehramtInformatikGithubRawDomain 3217
\LehramtInformatikGithubTexRepo 819, 3218
\LehramtInformatikRepository 168, 218, 363, 366, 369, 372, 375, 1807, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 3215
\leq 2407, 2870, 2890
\let 1661, 1662, 2121, 3021, 3022, 3023, 3024, 3037, 3039, 3040, 3041, 3043, 3044, 3045, 3046, 3056, 3148, 3186, 3187, 3422, 3437, 3449, 3462

\li@chomsky@erklaerung@texte 931, 967
\li@EntwurfsCode 1109, 1160, 1161, 1162, 1211, 1212, 1213, 1214, 1280, 1281, 1282, 1283, 1284, 1285, 1312, 1313, 1314, 1315, 1316, 1317, 1383, 1553, 1554, 1555, 1556, 1596, 1597
\li@EntwurfsCodeAllgemein 1108
\li@fussnote@text 2583, 2589, 2593, 2597, 2601
\li@GithubLink 3248, 3261, 3271, 3282
\li@mget .. 1858, 1862, 1882
\li@minc 1861, 1883
\li@mset 1855, 1863, 1872, 1875
\li@numdiscs 1867, 1876, 1882
\li@Rmodell@Schrift 3086, 3095, 3105
\li@SansFett .. 132, 146, 148, 150, 172, 176, 180
\li@sequence ... 1868, 1881
\li@synthese@erklaerung@texte 3337, 3397
\liAbleitung 1686
liAdditum (environment) 2197
\LiADDITUMfalse 575
\LiADDITUMtrue 17, 575
liAHuelle (environment) 2678
\liAlphabet 1671
liAntwort (environment) 2153
\LiANTWORTfalse 579
\LiANTWORTtrue 579
\liAnweisung 2028
\liAssemblerCode ... 3286
\liAssemblerDatei .. 3287
\liAttribut 3105
\liAttributHuelle 2669, 3349, 3354
\liAttributHuelleOhneMathe 2669, 2672, 2674, 2688, 2698, 2706
\liAttributMenge 2677, 2689, 2692, 2699, 2700, 2714, 2716
\liAufgabe 362
\liAufgabenMetadaten 39, 232, 393
\liAufgabenTitel 401
\liAusdruck 1708
\liAutomat 412
\liAutomatenKante ... 444
\liBandAlphabet 1672
\liBedingung 2029
\liBedingungDrei 2360, 2403, 2443
\liBedingungEins 2354, 2392, 2435
\liBedingungFalsch . 2031
\liBedingungWahr ... 2030
\liBedingungZwei 2357, 2398, 2439
\liBeschriftung 2142
\liBindeAufgabeEin .. 215
\liBindePdfEin 168
\liChomskyErklaerung 931, 972
\liChomskyUeberErklaerung 970
\liChomskyUeberschrift 919, 971
\liCpmEreignis 980
\liCpmFruehErklaerung 1051
\liCpmFruehI ... 1044, 1064
\liCpmSpaetErklaerung 1069
\liCpmSpaetI ... 1037, 1082
\liCpmVon 1020
\liCpmVonOhneMathe 1020, 1023, 1025
\liCpmVonZu 1012
\liCpmVonZuOhneMathe 1012, 1015, 1017
\liCpmVorgang 997
\liCpmZu 1028
\liCpmZuOhneMathe 1028, 1031, 1033
liDiagramm (environment) 2276
liEinbettung (environment) 2152
\liEntwurfs 1534
\liEntwurfsAbstrakteFabrik 1164
\liEntwurfsAbstrakteFabrikBeschreibung 1112, 1165
\liEntwurfsAbstrakteFabrikCode 1159, 1169
\liEntwurfsAbstrakteFabrikUml 1117, 1167
\liEntwurfsAdapter . 1216
\liEntwurfsAdapterAkteure 1186, 1218
\liEntwurfsAdapterCode 1210, 1219
\liEntwurfsAdapterUml 1171, 1217
\liEntwurfsBeobachter 1287
\liEntwurfsBeobachterAkteure 1245, 1289
\liEntwurfsBeobachterCode 1279, 1290
\liEntwurfsBeobachterUml 1221, 1288
\liEntwurfsDekorierer 1319
\liEntwurfsDekoriererAkteure 1321
\liEntwurfsDekoriererCode 1311, 1322
\liEntwurfsDekoriererUml 1292, 1320
\liEntwurfsEinfacheFabrik 1355
\liEntwurfsEinfacheFabrikAkteure 1339, 1357
\liEntwurfsEinfacheFabrikUml 1324, 1356
\liEntwurfsEinzelstueck 1385
\liEntwurfsEinzelstueckAkteure 1374, 1390
\liEntwurfsEinzelstueckBeschreibung 1359, 1386
\liEntwurfsEinzelstueckCode 1382, 1392
\liEntwurfsEinzelstueckUml 1364, 1388
\liEntwurfsErbauer . 1442
\liEntwurfsErbauerAkteure 1414, 1444
\liEntwurfsErbauerUml 1394, 1443
\liEntwurfsFabrikmethode 1488, 1513
\liEntwurfsFabrikmethodeAkteure 1463, 1490
\liEntwurfsFabrikmethodeUml 1446, 1489
\liEntwurfsKompositum 1513
\liEntwurfsKompositumAkteure 1515
\liEntwurfsKompositumUml 1492, 1514
\liEntwurfsModellPraesentationSteuerung 1534
\liEntwurfsModellPraesentationSteuerungAk 1536
\liEntwurfsModellPraesentationSteuerungUm 1517, 1535
\liEntwurfsStellvertreter 1558
\liEntwurfsStellvertreterCode 1552, 1560
\liEntwurfsStellvertreterUml 1538, 1559
\liEntwurfsZustand . 1599
\liEntwurfsZustandAkteure 1577, 1601
\liEntwurfsZustandCode 1595, 1602
\liEntwurfsZustandUml 1562, 1600
\liEpsilon 1657
\liErAttribute 1615, 1629, 1631
\liErDatenbankName . 1634
\liErEntity 1613, 1617, 1619
\liErledigt 3462
\liErMpAttribute ... 1628
\liErMpEntity 1616

\liErMpRelationship 1622
\liErRelationship ... 1614, 1623, 1625
\liExamensAufgabe ... 365
\liExamensAufgabeA .. 374
\liExamensAufgabeTA . 371
\liExamensAufgabeTTA . 368
liExkurs (environment) 2222
\LiEXKURSfalse 577
\LiEXKURStrue 577
\liFalsch 912
\liFlaci 1718
\liFremd 3085
\liFunktionaleAbhaengigkeit 2708, 2711, 2719
\liFunktionaleAbhaengigkeiten 2722
\liFussnote ... 2582, 2584
\liFussnoteDreiText 2596, 2614
\liFussnoteEinsText 2588, 2608
\liFussnoteLink 2292
\liFussnoten 2604
\liFussnoteUrl . 1532, 2288
\liFussnoteVierText 2600, 2617
\liFussnoteZweiText 2592, 2611
\liGeschweifteKlammern 1690, 2723, 3474
\liGrafikCCLizenz 1809, 2567
\liGrafikLogo 249, 1814, 2543
\liGrafikLogoPfad 1806, 1811, 1816
\liGrammatik 1729
liGraphenFormat (environment) 1848
\liHanoi 1855
\liHaskellCode 3293
\liHaskellDatei 3294
\liInduktionAnfang . 3557
\liInduktionErklaerung 3556
\liInduktionMarkierung 3555
\liInduktionSchritt 3573
\liInduktionVoraussetzung 3565
liJavaAngabe (environment) 3234
\liJavaCode 3237
\liJavaDatei ... 1110, 3255
\liJavaExamen 3275
\liJavaTestDatei ... 3265
liKasten (environment) 1776
\liKellerAutomat 453
\liKellerKante 491
\liKellerUebergang 486, 492
\liKontrollCode 2032
liKontrollflussgraph (environment) . 2020
\liKontrollKnotenPfad 2034
\liKontrollTextzeileKnoten 2033, 2038
\liKurzeTabellenLinie 1092
\liLadeAllePakete 203, 585
\liLadePakete 5, 123, 189, 316, 337, 405, 408, 581, 586, 918, 979, 1648, 1892, 2480, 2581, 2661, 2829, 3198, 3319
\liLatexCode 3247
\liLeereZelle 2619
liLernkartei (environment) 2266
\liLinksReduktion .. 2686
\liLinksReduktionInline 2695, 2703
\liLogoTextCCLizenz 96, 304, 2563
\liLogoTextProjekt 93, 301, 2539
\liMasterExkurs 2448
\liMasterFaelle 2387, 2455
\liMasterFallRechnung 2433
\liMasterVariablen 2364, 2450
\liMasterVariablenDeklaration 2410
\liMasterWolframLink 2458
\liMenge . 422, 423, 425, 464, 465, 466, 470, 522, 523, 524, 528, 1649, 1698, 1737, 1738
\liMengeOhneMathe 1649, 1652, 1654
\liMetaBschlangaulSammlung 2058, 2481, 2548
\liMetaCCLink .. 2493, 2572
\liMetaEmail 2064, 2503, 2508, 2510
\liMetaEmailLink 2506, 2522
\liMetaHermineFriends . 252, 2063, 2484, 2552
\liMetaHilfMit 102, 2513, 2524
\liMetaQuelltext 104, 2524
\liMetaSetze 41, 233, 382, 394
\liMetaUeberDasProjekt 2487, 2559
\liMinimierungErklaerung 2641
\liMinispracheDatei 3290
\linespread 3093
\liNichtsZuTun 3463
\liO 2327, 2355
\liOmega 2317, 2361
\liOmegaOhneMathe 2317, 2322, 2324
\liONotationO 2749
\liOOhneMathe 2327, 2332, 2334
\liParagraphMitLinien 967, 1052, 1070, 2643, 3397, 3464, 3561, 3569, 3577
\liPetriErreichKnotenDrei 2823
\liPetriErreichTransition 2820
\liPetriSetzeSchluessel 2765
\liPetriTransitionsName 2812, 2824
\liPetriTransitionsNameOhneMathe 2812, 2815, 2817
\liPetriTransPfeile 2824
\liPolynomiellReduzierbar 1912
\liPotenzmenge 1658, 1662, 2835
\liPotenzmengeOhneMathe 1659, 1660, 1661
\liPrimaer 3084
\liProblemBeschreibung 1896
\liProblemClique ... 1919
\liProblemName 1895, 1902, 1914, 1916, 1929, 1940, 1941, 1949, 1950
\liProblemSat 1948
\liProblemSubsetSum 1939, 1948
\liProblemVertexCover 1919, 1927
\liProduktionen 1697, 1739
liProduktionsRegeln (environment) . 1687
liProjektSprache (environment) 2151
\liPruefungsNummer .. 126
\liPruefungsTitel ... 129
\liPseudoUeberschrift 2128, 2208, 2211, 2629, 2639, 3558, 3566, 3574
\liPumpingKontextfrei 2881
\liPumpingRegulaer . 2860
liQuellen (environment) 2246
\liRechtsReduktionInline 2703
\liRekursionsGleichung 2351, 2414
\liRelation 2733
liRelationenSchemaFormat (environment) . 3106
\liRelationMenge ... 3100
\liRichtig 911
liRmodell (environment) 3086

\liRundeKlammer 2304, 2308, 2318, 2328, 2342, 2746, 2750
\liSetzeExamenTeilaufgabeNr 211
\liSetzeExamenThemaNr 207
\liSortierMarkierung 3125
\liSortierPfeil 3119
\liSortierPfeilUnten 3122
\liSpaltenUmbruch . . 3194
\liSqlCode 3298
\listen@punkt . . 2246, 2258
\liStrich 1894
\liSyntheseErklaerung 3337, 3402
\liSyntheseUeberErklaerung 3400
\liSyntheseUeberschrift 3321, 3401
\liT . . 2337, 2352, 2366, 2430
\liTeilen 2666
\literatur 2084, 2108
\liTheta 2307, 2358, 2390, 2396, 2401
\liThetaOhneMathe 2307, 2312, 2314
\liTitelSeite 157, 171
\liTOhneMathe 2337, 2346, 2348
\liTrennSeite 140
\liTuringKante 554
\liTuringLeerzeichen 510, 518
\liTuringMaschine . . . 511
\liTuringUeberfuehrung 557
\liTuringUebergaenge 549, 555
\liTuringUebergangZelle 544
\liTypoUeberGROSS . . 3497
\liTypoUeberGross 3491, 3497
\liTypoUeberschrift 3485, 3494, 3500
\liUeberfuehrungsFunktion 1663
\liUeberfuehrungsFunktionOhneMathe 1663, 1666, 1668
liUebergangsTabelle (environment) . 2627
\liUeberschriftDreiecksTabelle 2637
\liUmlLeserichtung . 3525
\liVertauschen 3113
\liWortInSprache . . . 1093
\liWortNichtInSprache 1098
\liWpEquivalent 3628
\liWpErklaerung 3631
\liWpErklaerungVerzweigung 3646
\liWpKalkuel 3608
\liWpKalkuelOhneMathe 3608, 3613, 3615, 3647, 3649, 3651
\liZustandsBuchstabe 1673, 1682, 1684, 1702, 1704
\liZustandsBuchstabeGross 1674, 1683, 1685
\liZustandsmenge . . . 1661
\liZustandsmengeNr 1675, 2844
\liZustandsmengeNrGross 1683
\liZustandsMengenSammlung 2831
\liZustandsMengenSammlungNr 2840
\liZustandsmengeOhneMathe 1661
\liZustandsname 1684
\liZustandsnameGross 1685, 2832, 2841
\liZustandsnameTiefgestellt 1700
\liZustandsPaar 2621
\liZustandsPaarVariablenName 2620, 2623, 2624
\llap 3073
\LoadClass 4, 114, 120, 121, 188, 315, 336
\log 2355, 2358, 2361, 2390, 2396
\logo 2528, 2541, 2565
\loop 2906
\lor 3650
\ltimes 2466

M

\makeatletter 3145
\makeatother 3153
\marginpar 1618, 1624, 1630, 2126
\mathbb . . . 1950, 2893, 3570
\mathbin . . 3075, 3076, 3077
\mathcal 2328, 2750, 3375, 3380, 3382, 3383, 3384
\Mathe 3625
\MatheEnv . 3618, 3626, 3629
\mathord 2473, 2474
\mdfsetup 2164, 2168, 2172, 2176
\mdseries 2184
\medskip 49, 68, 1904, 2135, 2145, 2147, 2256, 2284, 2555, 3091, 3097, 3472, 3476, 3619, 3623
\memph 2126
\mintinline 3240, 3247, 3286, 3293, 3298
\mkern 3075, 3076, 3077
\mlq 2471, 2473
\mrq 2471, 2474
\msg 840
\myList 3166, 3167, 3168, 3171
\myNodes 3155, 3170, 3176, 3180, 3182

N

\NeedsTeXFormat 1, 111, 117, 185, 311, 333, 347, 360, 378, 403, 571, 846, 907, 914, 976, 1089, 1104, 1605, 1641, 1754, 1791, 1800, 1821, 1850, 1888, 1960, 2043, 2082, 2088, 2110, 2300, 2463, 2476, 2578, 2657, 2740, 2760, 2826, 2851, 2856, 2900, 3066, 3079, 3108, 3189, 3196, 3203, 3208, 3300, 3315, 3406, 3410, 3456, 3505, 3547, 3584, 3604
\neg 3651
\negthinspace . . 2305, 2747
\newcounter . . . 2920, 2921
\NewDocumentCommand 413, 454, 491, 512, 554, 581, 981, 998, 1037, 1044, 1093, 1098, 1708, 1730, 1809, 1814, 1912, 2035, 2086, 2288, 2292, 2722, 2733, 2820, 3255, 3265, 3275, 3287, 3290, 3294, 3525
\NewDocumentEnvironment . 1687, 1776, 1848, 2020, 2151, 2152, 2153, 2197, 2222, 2247, 2266, 2276, 2628, 2678, 3088, 3106
\newif 575, 577, 579
\newlength 3631
\newminted 3234
\node 994, 2028, 2033, 2946, 2951, 2956, 2962, 2968, 2974, 3125, 3170, 3542
\noexpand 3034, 3035, 3036, 3055, 3170
\noindent . . . 28, 46, 54, 70, 101, 792, 1095, 1100, 2132, 2144, 2148, 2180, 2237, 2239, 2254, 2270, 2272, 2280, 2452, 2607, 2610, 2613, 2616, 3101, 3435, 3448, 3465, 3477, 3637
\nolinkurl 3252
\normalsize . . . 1764, 3241
\notin 1101

\null 3470

O

\o 2749, 2754, 2756
\o@join 3072, 3075, 3076, 3077
\Omega 2318
\omega 2862, 2863, 2883, 2884
\or 2983

P

\pagestyle 238, 1786
\par 23, 27, 30, 49, 66, 79, 241, 245, 249, 251, 252, 259, 261, 266, 270, 274, 791, 812, 971, 1719, 1727, 2131, 2137, 2143, 2238, 2262, 2271, 2550, 2553, 3025, 3047, 3062, 3103, 3401, 3434, 3438, 3450, 3471, 3475, 3483, 3636, 3642
\paragraph 1764
\parindent 3094
\path . . . 445, 492, 555, 1009
\pgfkeys . . . 2776, 3528, 3529, 3530, 3531, 3532, 3535, 3538, 3540
\pgfmath@count 3147, 3149, 3151
\pgfmath@smuggleone 3152
\pgfmathdeclarefunction 3146
\pgfmathint 3147
\pgfmathparse 1862, 3164, 3169, 3172, 3186, 3187
\pgfmathresult 1863, 3147, 3148, 3150, 3152, 3165, 3173, 3186, 3187
\pgfutil@empty 3148
\pgfutil@loop 3149
\pgfutil@repeat 3152
\preceq 1915
\prime 1894
\printbibliography . 2108
\prop . 3440, 3443, 3444, 3446
\ProvidesClass 2, 112, 118, 186, 312, 334
\ProvidesPackage . 348, 361, 379, 404, 572, 847, 908, 915, 977, 1090, 1105, 1606, 1642, 1755, 1792, 1801, 1822, 1851, 1889, 1961, 2044, 2083, 2089, 2111, 2301, 2464, 2477, 2579, 2658, 2741, 2761, 2827, 2852, 2857, 2901, 3067, 3080, 3109, 3190, 3197, 3204, 3209, 3301, 3316, 3407, 3411, 3457, 3506, 3548, 3585, 3605

Q

\QS@list 3027, 3038, 3042, 3049, 3055, 3060, 3063
\QS@select@equal 3002, 3006
\QS@select@greater 3003, 3007
\QS@select@smaller 2998, 3001, 3005
\QS@sort@a 2980, 3013, 3034, 3035
\QS@sort@b 2980, 2981
\QS@sort@c 2984, 2991
\QS@sort@d 2992, 3000
\QS@sort@empty . 2982, 2987
\QS@sort@single 2983, 2988
\QSinitialize 2905, 3017, 3052
\QSIr . . 2988, 2994, 3002, 3022, 3036, 3041, 3044
\QSIrr 3023, 3036, 3037, 3045
\QSLr 2994, 3001, 3012, 3013, 3021, 3034, 3039, 3043
\QSpivotStep 2907, 3017, 3021, 3032
\QSr 2994
\QSRr 3003, 3024, 3035, 3040, 3046, 3055, 3056, 3057
\QSsortStep 2909, 3017, 3033, 3034
\quad 2585

R

\raggedright 2187
\raisebox 2033
\relax 2121, 2994, 3037, 3039, 3040, 3041, 3149, 3151
\renewcommand 285, 287, 1767, 1768, 1769, 1770, 1773, 1774, 1787, 1788, 2067, 2068, 2627
\repariere . . 87, 329, 2076
\repeat 2910
\RequirePackage . . . 16, 115, 122, 124, 199, 204, 205, 291, 324, 344, 345, 350, 406, 509, 573, 583, 843, 849, 850, 910, 978, 1107, 1108, 1608, 1610, 1611, 1647, 1756, 1757, 1759, 1761, 1766, 1775, 1781, 1789, 1793, 1794, 1804, 1805, 1823, 1824, 1825, 1853, 1854, 1893, 1962, 2047, 2085, 2090, 2091, 2107, 2114, 2115, 2116, 2303, 2467, 2468, 2743, 2744, 2745, 2762, 2764, 2854, 2914, 2916, 3068, 3069, 3070, 3083, 3111, 3142, 3193, 3199, 3206, 3211, 3222, 3223, 3303, 3408, 3414, 3461, 3509, 3510, 3512, 3513, 3551, 3552, 3553, 3586, 3606
\right 2305, 2747
\RIGHTarrow . . . 3527, 3532
\Rightarrow . . . 1096, 1101
\rightarrow 558, 935, 940, 948, 952, 954, 955, 957, 1012, 1020, 2824, 3346, 3353, 3355, 3358, 3363, 3368, 3369, 3374
\rightouterjoin 3076
\rmfamily 136, 922, 1882, 3324, 3487
\Roman 1772
\roman 1772, 1774
\romannumeral 2992
\rtimes 2466
\rule 29, 3025, 3047, 3062, 3073

S

\sb 418, 428, 430, 459, 517, 947, 948, 952, 955, 956, 957, 1039, 1041, 1046, 1048, 1702, 1704, 2355, 2358, 2361, 2390, 2396, 2647, 2812, 2821, 3368, 3369, 3370, 3375, 3379, 3380, 3383, 3384, 3387, 3388, 3389
\scriptscriptstyle 1012, 1020, 1028
\scriptsize 1721, 1973, 1980, 1986, 2051, 2184, 2571, 3556, 3609, 3639
\section 200, 209, 398
\sectionbreak 201
\seq . . . 2037, 2038, 2039, 2249, 2250, 2251, 2258
\setbox . . . 2157, 2201, 3072
\setcounter . . 202, 290, 1765, 3026, 3048, 3062
\setganttlinklabel . . 1795, 1796, 1797, 1798

\setlength 283, 284, 286, 2078, 3094, 3633, 3634, 3644
\setmainfont 1758
\setmainlanguage 844
\setminted 3228, 3229
\setminus 2707
\setul 3085
\setze 75, 84, 326, 2055, 2070
\shoveleft 2687
\shoveright 2691
\Sigma 415, 456, 514, 1671, 1672, 1732
\sigma 945, 947, 948
\SLASH 2127
\small 2279
\sort 3159
\sortList 3158, 3167
\square 912
\stepcounter 2946, 2951, 2956, 2959, 2961, 2965, 2967, 2971, 2973
\stichwoerter 19, 71
\str 127, 130, 146, 148, 176, 180, 713, 923, 932, 2161, 2836, 2845, 3325, 3338
\string 2689, 2699
\StrSubstitute . 3166, 3168
\strut ... 142, 154, 159, 165, 2412, 2416, 2420, 2424, 2428, 3194
\subsection 213
\subseteq . 3349, 3382, 3389
\subsubsection .. 234, 3445

T

\tableofcontents 288, 2122
\TeX 2525
\text 428, 430, 533, 2669, 3556, 3609
\textbackslash 3426, 3427, 3429
\textbf . 21, 1059, 1077, 1613, 1920, 1929, 1940, 1949, 2133, 2146, 2181, 2238, 2255, 2271, 2632, 3436
\textcolor 2032, 3239, 3555
\textit 34, 1453, 1495, 1496, 1497, 1498, 2281, 2677, 2737
\textsc 1895
\textsf 2255
\textstyle 2375, 2407
\texttt 1636, 1895, 2029, 2030, 2031, 2032, 3424, 3609
\textwidth 29, 2078
\thematik 32, 65
\thepage 1785, 2065
\theparagraph 1764
\thesection 200
\Theta 2308
\thinspace 3609
\thispagestyle 170
\tikz 2033
tikz: bbaum 34
tikz: li binaer baum 32
\tikzchildnode 867
\tikzparentnode 867
\tikzset 447, 494, 560, 852, 878, 1827, 1964, 2802, 2928, 3132, 3305, 3587
\tikzumlset 3515
\times 558
\tiny ... 100, 1619, 1625, 1631, 2032, 2126, 3250
\titel 236, 293
\titleformat 200, 1762, 1764
\titlespacing 1763
\tl 20, 33, 43, 57, 208, 212, 216, 223, 226, 385, 414, 415, 416, 417, 418, 419, 422, 423, 424, 425, 426, 428, 430, 455, 456, 457, 458, 459, 460, 461, 464, 465, 466, 467, 468, 469, 470, 513, 514, 515, 516, 517, 518, 519, 522, 523, 524, 525, 526, 527, 528, 655, 659, 686, 690, 691, 692, 695, 700, 701, 702, 714, 723, 757, 760, 763, 768, 783, 784, 785, 786, 797, 803, 806, 809, 825, 832, 982, 985, 990, 991, 999, 1000, 1003, 1004, 1710, 1731, 1732, 1733, 1734, 1737, 1738, 1739, 1740, 2338
\tmp 2709
\TmpPlaceEight 2784
\TmpPlaceFive 2781
\TmpPlaceFour 2780
\TmpPlaceNine 2785
\TmpPlaceOne 2777
\TmpPlaceSeven 2783
\TmpPlaceSix 2782
\TmpPlaceTen 2786
\TmpPlaceThree 2779
\TmpPlaceTwo 2778
\TmpScale 2797
\TmpTransitionEight 2773, 2794
\TmpTransitionFive 2770, 2791
\TmpTransitionFour 2769, 2790
\TmpTransitionNine 2774, 2795
\TmpTransitionOne 2766, 2787
\TmpTransitionSeven 2772, 2793
\TmpTransitionSix 2771, 2792
\TmpTransitionTen 2775, 2796
\TmpTransitionThree 2768, 2789
\TmpTransitionTwo 2767, 2788
\TmpX 2798
\TmpY 2799
\ttfamily 3086

U

\ul 1614, 3084, 3085
\umlaggreg 1572
\umlassoc 1530
\umlclass 1119, 1123, 1127, 1174, 1175, 1176, 1223, 1228, 1233, 1236, 1294, 1295, 1296, 1301, 1302, 1331, 1366, 1397, 1398, 1401, 1452, 1455, 1494, 1500, 1501, 1519, 1520, 1521, 1542, 1543, 1544, 1564, 1565, 1566, 1567
\umldep 1460
\umlHVHaggreg 1241, 1307, 1510
\umlinherit 1180, 1231, 1405, 1450, 1458
\umlnote .. 1182, 1407, 1574
\umlreal 1178, 1239
\umlsimpleclass 1134, 1135, 1136, 1140, 1142, 1143, 1144, 1173, 1326, 1327, 1328, 1396, 1448, 1449, 1540
\umlstatic 1333, 1367
\umluniaggreg 1403
\umluniassoc 1154, 1179, 1404, 1528, 1529, 1548, 1549
\umlVHuniassoc . 1155, 1156
\umlVHVdep . 1148, 1149, 1151, 1152, 1335, 1336
\umlVHVinherit .. 1131, 1132, 1137, 1138, 1145, 1146, 1304, 1305, 1329, 1330, 1508, 1509, 1546, 1547
\umlVHVreal 1298, 1299, 1569, 1570
\UParrow 3529
\url 826, 2289
\usemintedstyle 3225

\usetikzlibrary .. 407, 851, 1609, 1826, 1963, 2763, 3071, 3112, 3143, 3514, 3602

V

\value 2908
\varepsilon 924, 935, 936, 1657, 2355, 2361, 2393, 2404
\vbox 2157, 2201
\vfill 91, 143, 153, 160, 164, 247, 254, 278, 299, 3194
\vrule 3466, 3470
\vspace 147, 149, 174, 178, 243, 264, 268, 272, 285, 2224, 2242, 3479, 3481

X

\xappto ... 3170, 3176, 3180
\xdef 1856
\xintApply 2996
\xintApplyUnbraced .. 2995, 3001, 3002, 3003
\xintCSVtoList 3055
\xintFor 2945, 2950, 2955, 2960, 2966, 2972, 3013
\xintifEq 3006
\xintifForLast 2962, 2968, 2974
\xintifGt 3007
\xintifLt 3005
\xintLength 2980
\xintnthelt 2992

Z

\ZB 2296
\zB 2295
\zustandsnamens@liste 1675, 1682, 1683