Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends*

September 13, 2021

Contents

Klassen **4**
aufgabe.cls . . . 5
basis.cls . . . 7
examen-scans.cls . . . 8
examen.cls . . . 10
haupt.cls . . . 13
theorie.cls . . . 15

Pakete **16**
abmessung.sty . . . 17
aufgaben-einbinden.sty . . . 18
aufgaben-metadaten.sty . . . 19
automaten.sty . . . 20
Endlicher Automat . . . 20
Kellerautomat . . . 22
Turingmaschine . . . 23
basis.sty . . . 26
IFs . . . 26
baum.sty . . . 32
Binärbaum . . . 33
AVL-Baum . . . 34
B-Baum . . . 35
checkbox.sty . . . 37
chomsky-normalform.sty . . . 38
Faulenzer . . . 38
TeX-Markup-Grundgerüst . . . 38
Konkretes TeX-Markup-Beispiel . . . 38
cpm.sty . . . 41
Faulenzer . . . 41
TeX-Markup-Beispiel: Graph . . . 41
TeX-Markup-Beispiel: Ergebnistabelle . . . 41
TeX-Markup-Beispiel: Nebenrechnungstabelle „Frühester Zeitpunkt“ . . . 41
TeX-Markup-Beispiel: Nebenrechnungstabelle „Spätester Zeitpunkt“ . . . 42
Hilfsmakros für Zeitpunkt-Tabelle . . . 43
cyk-algorithmus.sty . . . 45
Faulenzer . . . 45
TeX-Markup-Beispiel . . . 45
entwurfsmuster.sty . . . 46
Namensschema der Entwurfsmuster-Makros: . . . 46
Reihenfolge . . . 46

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

Abstrakte Fabrik (Abstract Factory) . . . 46
Adapter . . . 48
Beobachter (Observer) . . . 49
Dekorierer (Decorator) . . . 51
Einfache Fabrik (Simple Factory) . . . 52
Einzelstück (Singleton) . . . 53
Erbauer (Builder) . . . 54
Fabrikmethode (Factory Method) . . . 55
Kompositum (Composite) . . . 57
Modell-Präsentation-Steuerung (Model-View-Controller) . . . 57
Stellvertreter (Proxy) . . . 58
Zustand (State) . . . 59
er.sty . . . 61
Tex-Markup-Beispiel: Komplettes Diagramm . . . 61
Tex-Markup-Beispiel: EER Enhanced Entity-Relation-Modell nach Elmasri/Navante . . . 62
Faulenzer . . . 62
formale-sprachen.sty . . . 64
formatierung.sty . . . 67
Schriftarten / Typographie . . . 67
Farben . . . 67
Überschriften . . . 67
Listen . . . 67
Kasten . . . 67
Header . . . 67
Zeilenabstände . . . 67
gantt.sty . . . 69
grafik.sty . . . 70
graph.sty . . . 71
hanoi.sty . . . 73
index.sty . . . 74
komplexitaetstheorie.sty . . . 75
Faulenzer . . . 75
kontrollflussgraph.sty . . . 77
Faulenzer . . . 77
TeX-Markup-Beispiel . . . 77
TikZ: pin . . . 77
Umgebungen . . . 78
Makros . . . 79
kopfzeile.sty . . . 80
literatur-dummy.sty . . . 81
literatur.sty . . . 82
makros.sty . . . 83
Umgebungen, die Inhalte aus- und einblenden können . . . 83
master-theorem.sty . . . 88
Faulenzer . . . 88
mathe.sty . . . 92
meta.sty . . . 93
Einfache Makros (Low level) . . . 93
Zusammengesetzte Makros (High level) . . . 94
minimierung.sty . . . 96
normalformen.sty . . . 99
Faulenzer . . . 99
o-notation.sty . . . 102
Faulenzer . . . 102
TeX-Markup-Beispiel: Funktionsgraphen mit pgfplots . . . 102
petri.sty . . . 103
Faulenzer . . . 103

potenzmengen-konstruktion.sty . . . 105
pseudo.sty . . . 107
pumping-lemma.sty . . . 108
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
TeX-Markup-Beispiel . . . 120
synthese-algorithmus.sty . . . 121
Faulenzer . . . 121
TeX-Markup Grundgerüst . . . 121
TeX-Markup Linksreduktion . . . 121
TeX-Markup Rechtreduktion . . . 121
TeX-Markup Relationen formen . . . 122
tabelle.sty . . . 125
tex-dokumentation.sty . . . 126
typographie.sty . . . 127
uml.sty . . . 128
vollstaendige-induktion.sty . . . 130
Faulenzer . . . 130
wasserfall.sty . . . 132
Latex-Markup-Beispiel . . . 132
wpkalkuel.sty . . . 133
Faulenzer . . . 133

Index **134**

Klassen

aufgabe.cls

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

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

Formatierung muss vor literatur sein, sonst Option clash

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

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

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

In Aufgaben wollen wir das Additum sehen.

```
\ADDITUMtrue
```

```
\ExplSyntaxOn
```

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

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

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

`\bAufgabenMetadaten`

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

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

  {
    \noindent
    \bfseries
```

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

  \medskip

  \noindent
  {\footnotesize\stichwoerter_formatiert:}

  \horizontale_linie:

  \setze_kopfzeile_oben_rechts:n { \gib_aufgaben_pfad_lang_sicher: }

  \bigskip

  \keine_einrueckung:
}
\setze_kopf_fusszeilen:nn {} {}
\AddToHook { begindocument }
{
  \repariere_kopfzeile_breite:
}
\AddToHook { enddocument }
{
  \vfill
  {
    \bLogoTextProjekt
    \bigskip

    \bLogoTextCCLizenz
    \bigskip

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

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

basis.cls

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

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

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

```
\RequirePackage{l3keys2e}
```

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

Lade Pakete die von allen Unterklassen benötigt werden.

```
\bLadePakete{
  kopfzeile
}
```

```
\ExplSyntaxOn
```

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

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

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

```
\ExplSyntaxOff
```

examen-scans.cls

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

Formatierung muss vor literatur sein, sonst Option clash

```
\bLadePakete{formatierung}
```

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

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

\bPruefungsNummer

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

\bPruefungsTitel

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

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

\bTrennSeite

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

\bTitelSeite

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

```
  \clearpage
}
```

\bBindePdfEin

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

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

    \vspace{4cm}

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

    \vspace{4cm}

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

\ExplSyntaxOff

```

examen.cls

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

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

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

Formatierung für die Überschriften setzen.

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

```
\bLadeAllePakete
```

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

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

```
\ExplSyntaxOn
```

`\bSetzeThemaNr`

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

`\bSetzeTeilaufgabeNr`

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

`\bBindeAufgabeEin`

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

`\bAufgabenMetadaten` Das Metadaten-Makro überschreiben

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

```
  \subsubsection{\_gib_aufgaben_titel:}
}

\cs_new:Npn \titel_seite:
{
  \titel_seite:nn
  {
    {
      \bfseries\Huge

      \g_einzelpruefungs_nr_tl
      \par

      \g_jahreszeit_tl
      \par

      \g_jahr_tl
      \par
    }
  }
  {
    \g_examen_fach_tl
    \par

    \vspace{0.5cm}

    Aufgabenstellungen~mit~Lösungsvorschlägen
    \par
  }

}

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

\setcounter{tocdepth}{4}
\RequirePackage[titles]{tocloft}
\AddToHook { begindocument }
{
  \titel_seite:

  \clearpage

  \inhalts_verzeichnis:

  \vfill

  \bLogoTextProjekt
  \bigskip

  \bLogoTextCCLizenz
  \bigskip

  \clearpage
}

\ExplSyntaxOff
```

303

haupt.cls

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

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

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

```
\bLadeAllePakete
```

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

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

```
\ExplSyntaxOn
```

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

Breiterer rechter Rand für die Indexes

```
\geometry{
  right = 3cm,
  marginparwidth = 2.8cm,
}
```

```
\AddToHook { begindocument }
{
  \repariere_kopfzeile_breite:
  \titel_seite:nn
  {
    {
      \Huge
      Die~komplette~Sammlung
    }

  }
  {
    Alle~Aufgaben
  }

  \tableofcontents
  \clearpage

  \pagestyle{fancy}
}
```

```
\AddToHook { enddocument }
{
  \printindex
}
```

`\bAufgabenMetadaten` Das Metadaten-Makro überschreiben

```
\def\bAufgabenMetadaten #1
{
  \bMetaSetze{#1}
  \subsection{\gib_aufgaben_pfad_lang_sicher:}
}
```

```
\ExplSyntaxOff

```

theorie.cls

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

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

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

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

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

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

Pakete

abmessung.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-baum}[2021/09/04 Einstellung der
Seitenabmessung mit Hilfe des geometry-Pakets.]

\RequirePackage{geometry}

\geometry{
  a4paper,
  margin=2cm,
  includeheadfoot,
  % showframe,
  % showcrop,
  % verbose=true,
}

```

aufgaben-einbinden.sty

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

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

```
\RequirePackage{standalone}
```

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

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

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

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

`\bExamensAufgabeTTA`

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

`\bExamensAufgabeTA`

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

`\bExamensAufgabeA`

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

aufgaben-metadaten.sty

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

```
\ExplSyntaxOn
```

`\bMetaSetze` Setze die Metadaten einer Aufgabe.

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

  \tl_clear:N \g_teilaufgabe_nr_tl

  \keys_set:nn { aufgabenmetadaten } {
    #1
  }

  \_setze_relativen_pfad:
}
```

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

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

```
\bAufgabenMetadaten{
  Titel = {Aufgabe 2},
  Thematik = {Petri-Netz},
  Stichwoerter = {Feld (Array), Implementierung in Java}
  ZitatSchluessel = sosy:pu:4,
  ZitatBeschreibung = {Seite 11},
  BearbeitungsStand = OCR,
  Korrektheit = absolut korrekt,
  RelativerPfad = Staatsexamen/46116/2016/03/Thema-2/Teilaufgabe-1/Aufgabe-2.tex,
  EinzelpruefungsNr = 46116,
  Jahr = 2016,
  Monat = 03,
  ThemaNr = 2,
  TeilaufgabeNr = 1,
  AufgabeNr = 2,
}
```

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

  \_gib_examen_titel: {}

  \section{\_gib_aufgaben_titel:}
}
```

```
\ExplSyntaxOff
```

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

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

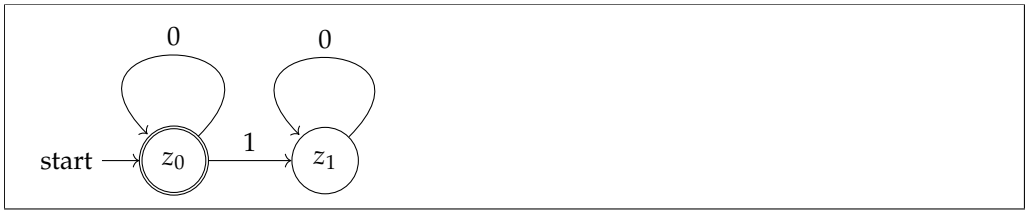
automaten.sty

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

Endlicher Automat

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

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

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




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

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

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

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

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

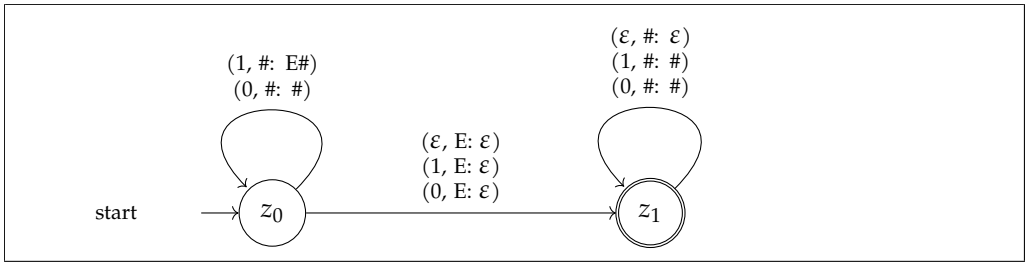
```
    start .code:n = {\tl_set:Nn \l_start_tl {##1}},
    dea .value_forbidden:n = true,
    dea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{DEA}}}},
    nea .value_forbidden:n = true,
    nea .code:n = {\tl_set:Nn \l_typ_tl {\sb{\text{NEA}}}},
  }

  \keys_set:nn { automat } { #2 }

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

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

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

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

Kellerautomat

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

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

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

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



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

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

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

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

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

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

  \keys_set:nn { kellerautomat } { #2 }

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

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

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

(a, #: A)
(b, #: ε)

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

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

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

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

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

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

Turingmaschine

```
\RequirePackage{amssymb}
```

\bTuringLeerzeichen

□

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

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

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

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

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

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

  \keys_set:nn { kellerautomat } { #2 }

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

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

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

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

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

\bTuringKante

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

Makro-Faulenzer: \let\t=\liTuringKante

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

\bTuringUeberfuehrung

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

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

basis.sty

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

```
\ExplSyntaxOn
```

IFs

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

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

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

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

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

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

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

`\bLadePakete`

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

`\bLadeAllePakete`

```
\def\bLadeAllePakete{
  \bLadePakete{
    aufgaben-einbinden,
    automaten,
    checkbox,
    chomsky-normalform,
    cpm,
    cyk-algorithmus,
    entwurfsmuster,
    er,
    formale-sprachen,
    gantt,
    grafik,
    graph,
    hanoi,
    kontrollflussgraph,
    komplexitaetstheorie,
    makros,
    master-theorem,
    mathe,
    minimierung,
    normalformen,
    petri,
    potenzmengen-konstruktion,
    pumping-lemma,
    pseudo,
    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,
  referenz,
  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,
  Referenz            .tl_gset:N = \g_referenz_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,
  Ueberprueft         .tl_gset:N = \g_ueberprueft_tl,
  %
  RelativerPfad       .tl_gset:N = \g_relativer_pfad_tl,
  IdentischeAufgabe   .tl_gset:N = \g_identische_aufgabe_tl,
  %
  EinzelpruefungsNr   .tl_gset:N = \g_einzelpruefungs_nr_tl,
  ExamenFach          .tl_gset:N = \g_examen_fach_tl,
  Jahr                .tl_gset:N = \g_jahr_tl,
  Monat               .tl_gset:N = \g_monat_tl,
  Jahreszeit          .tl_gset:N = \g_jahreszeit_tl,
  ThemaNr             .tl_gset:N = \g_thema_nr_tl,
  TeilaufgabeNr       .tl_gset:N = \g_teilaufgabe_nr_tl,
  AufgabeNr           .tl_gset:N = \g_aufgabe_nr_tl,
}
\cs_gset:Npn \_setze_relativen_pfad: {
  \tl_if_empty:NTF \g_relativer_pfad_tl
  {
    \bool_if:nTF
    {
      ! \tl_if_empty_p:N \g_einzelpruefungs_nr_tl &&
      ! \tl_if_empty_p:N \g_jahr_tl &&
      ! \tl_if_empty_p:N \g_monat_tl
    }
    {
      \tl_gset:Nn \g_relativer_pfad_tl {
        Staatsexamen /
        \g_einzelpruefungs_nr_tl /
        \g_jahr_tl /
        \g_monat_tl /
        \tl_if_empty:NTF \g_thema_nr_tl {} { Thema - \g_thema_nr_tl / }
        \tl_if_empty:NTF \g_teilaufgabe_nr_tl {} { Teilaufgabe - \g_teilaufgabe_nr_tl / }
        \tl_if_empty:NTF \g_aufgabe_nr_tl {} { Aufgabe - \g_aufgabe_nr_tl .tex }
      }
    }
    {}
  }
  {}
}
\cs_set:Nn \_trenner: {
  \, / \,
}
\cs_gset:Npn \_gib_jahreszeit_durch_monat: #1 {
  % \str_case funktioniert nicht mit den Tokenlist variablen.
  \tl_case:Nn { #1 }
  {
    { 3 } { Frühjahr }
    { 03 } { Frühjahr }
    { 9 } { Herbst }
    { 09 } { Herbst }
  }
}
```

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

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

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

„Einzelprüfungsnummer / Jahr / Jahreszeit“ mit Trennzeichen

```
\cs_gset:Npn \_gib_einzelpruefung_trenner: {
  \g_einzelpruefungs_nr_tl

  \_trenner:

  \g_jahr_tl

  \_trenner:

  \_gib_jahreszeit_durch_monat: \g_monat_tl
}
```

Thema 1 / Teilaufgabe 2 / Aufgabe 3

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

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

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

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

```
    ! \tl_if_empty_p:N \g_monat_tl &&
    ! \tl_if_empty_p:N \g_aufgabe_nr_tl
  }
  {
    {
      \footnotesize
      \par
      \noindent
      Staatsexamen ~
      \g_einzelpruefungs_nr_tl \_trenner:
      \g_jahr_tl \_trenner:

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

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

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

  \strut
  \vspace{1cm}

```

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

  \vfill

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

  \vfill

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

    \vspace{4cm}

    {
      \bfseries
      \Large
      \bMetaBschlangaulSammlung
    }
    \par

    \medskip

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

  \vspace{2cm}

  \strut

  \clearpage
}
```

```
\cs_new:Npn \keine_einrueckung: {
  \par
  \@afterindentfalse
  \@afterheading
}
```

```
\msg_new:nnn { aufgabenmetadaten } { kein-titel }
  { Der~Schlüssel~ist~zwingend~notwendig }
```

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

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

biblatex not working with lualatex and babel

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

baum.sty

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

```
\RequirePackage{tikz}
```

für b binaer baum

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

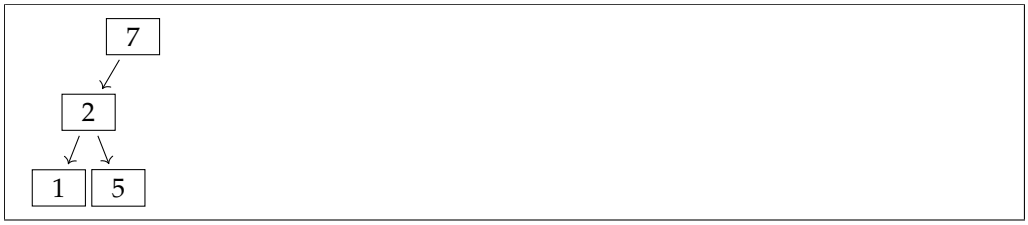
Für b baum

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

Binärbaum

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

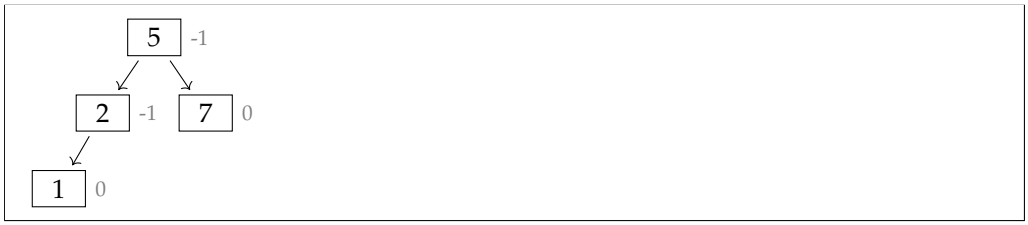
```
\begin{tikzpicture}[b binaer baum]
\Tree
[.7
  [.2
    [.1 ]
    [.5 ]
  ]
  \edge[blank]; \node[blank]{};
]
\end{tikzpicture}
```




```
\tikzset{
  b 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,
    }
  },
}
```

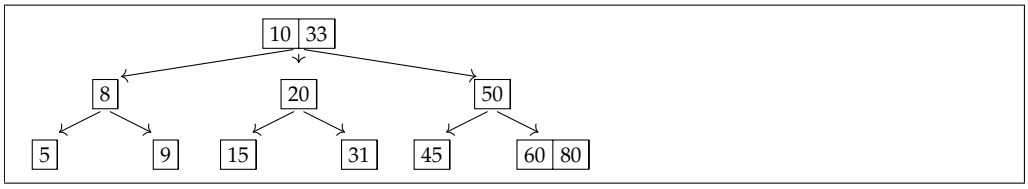
AVL-Baum

```
\begin{tikzpicture}[b 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,
  b 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{
  b bbaum knoten/.style={
    rectangle split parts=10,
    rectangle split,
    rectangle split horizontal,
    rectangle split ignore empty parts,
    draw,
    fill=white
  },
  b bbaum/.style={
    every node/.style={
      b 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,
    },
  }
}
```

`bBaum` `\begin{bBaum}{beschriftung}\end{bBaum}`: Zum Einbetten eines Baums.

```
\NewDocumentEnvironment { bBaum }{ m +b }
{
  {
    \small
    \noindent
    \textit{#1}:
  }
  \begin{center}
  #2
  \medskip
  \end{center}
} {}
```

checkbox.sty

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

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

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

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

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

chomsky-normalform.sty

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

```
\ExplSyntaxOn
```

```
\bLadePakete{typographie}
```

Faulenzer

```
\let\erklaerung=\bChomskyErklaerung
\let\schritt=\bChomskyUeberschrift
\let\schrittE=\bChomskyUeberErklaerung
```

TeX-Markup-Grundgerüst

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

Konkretes TeX-Markup-Beispiel

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

\bNichtsZuTun

\item \schrittE{2}

\begin{bProduktionsRegeln}
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{bProduktionsRegeln}

\item \schrittE{3}

\begin{bProduktionsRegeln}
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{bProduktionsRegeln}

\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{bProduktionsRegeln}
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{bProduktionsRegeln}
\end{enumerate}
```

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

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

\bChomskyErklaerung **Makro-Faulenzer:** `\let\erklaerung=\liChomskyErklaerung`
Hoffmann Seite 180

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

```
  }
}
\def\bChomskyErklaerung#1{
  {
    \itshape
    \footnotesize
    \bParagraphMitLinien{\li@chomsky@erklaerung@texte{#1}}
  }
}
```

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

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

\ExplSyntaxOff
```

cpm.sty

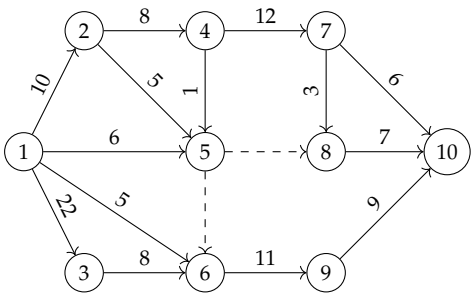
```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-cpm}[2020/09/03]
\RequirePackage{tikz}
```

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

Faulenzer

```
\let\f=\footnotesize
\let\FZ=\bCpmFruehI
\let\SZ=\bCpmSpaetI
\let\v=\bCpmVon
\let\vz=\bCpmVonZu
\let\z=\bCpmZu
```

TeX-Markup-Beispiel: Graph




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

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

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

TeX-Markup-Beispiel: Ergebnistabelle

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

TeX-Markup-Beispiel: Nebenrechnungstabelle „Frühester Zeitpunkt“

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

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

TeX-Markup-Beispiel: Nebenrechnungstabelle „Spätester Zeitpunkt“

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

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

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

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

  \keys_set:nn { cpmEreignis } { #1 }

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

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

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

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

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

  \keys_set:nn { cpmVorgang } { #1 }

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

Hilfsmakros für Zeitpunkt-Tabelle

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

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

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

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

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

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

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

```
\ExplSyntaxOn
```

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

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

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

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

```
  \fi
}
```

\bCpmFruehErklaerung

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

```
\def\bCpmFruehErklaerung{
  \bParagraphMitLinien{
    Wir~führen~eine~Vorwärtsterminierung~durch~
    und~addieren~die~Dauern.~

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

    \textbf{Erläuterungen:}~

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

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

\bCpmSpaetErklaerung

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

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

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

    \textbf{Erläuterungen:}~

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

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

\ExplSyntaxOff
```

cyk-algorithmus.sty

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

Faulenzer

```
\let\l=\bKurzeTabellenLinie
```

TeX-Markup-Beispiel

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

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

\bKurzeTabellenLinie

Makro-Faulenzer: \let\l=\liKurzeTabellenLinie

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

\bWortInSprache

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

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

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

\bWortNichtInSprache

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

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

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

entwurfsmuster.sty

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

Namensschema der Entwurfsmuster-Makros:

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

Reihenfolge

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

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

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

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

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

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

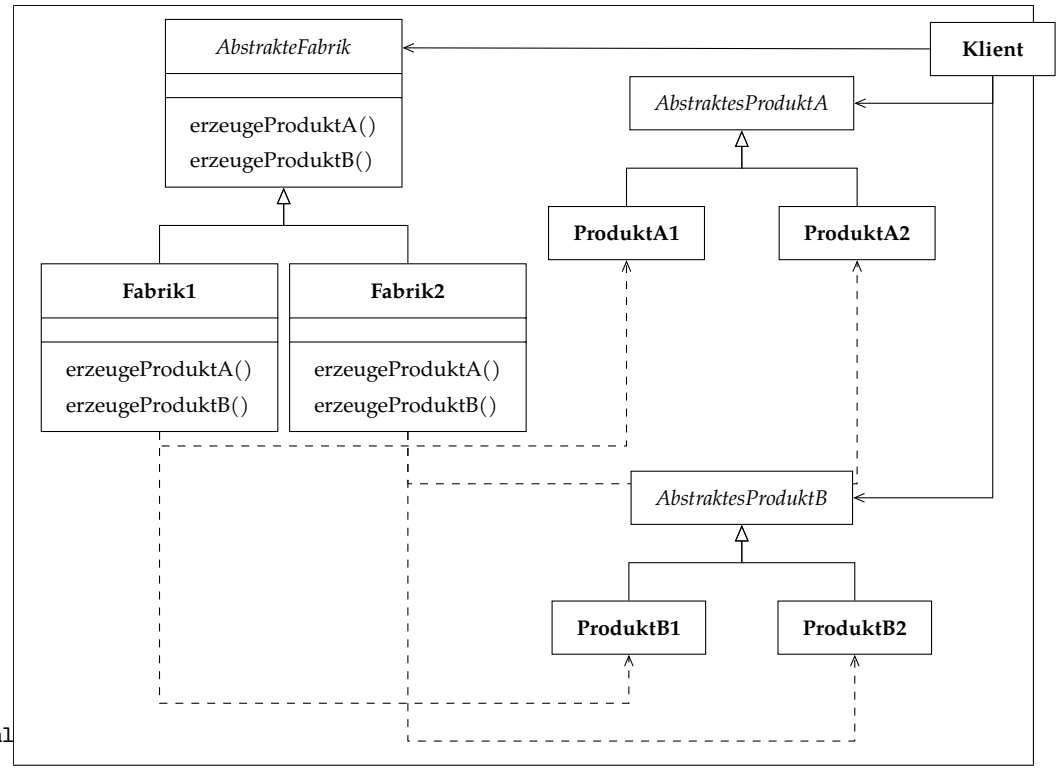
`\li@EntwurfsCodeAllgemein` Allgemeine Code-Beispiele zu den UML-Diagrammen und Stellvertretern

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

Abstrakte Fabrik (Abstract Factory)

`sAbstrakteFabrikBeschreibung`

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



\bEntwurfsAbstrakteFabrikUml

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

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

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

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

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

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

```

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

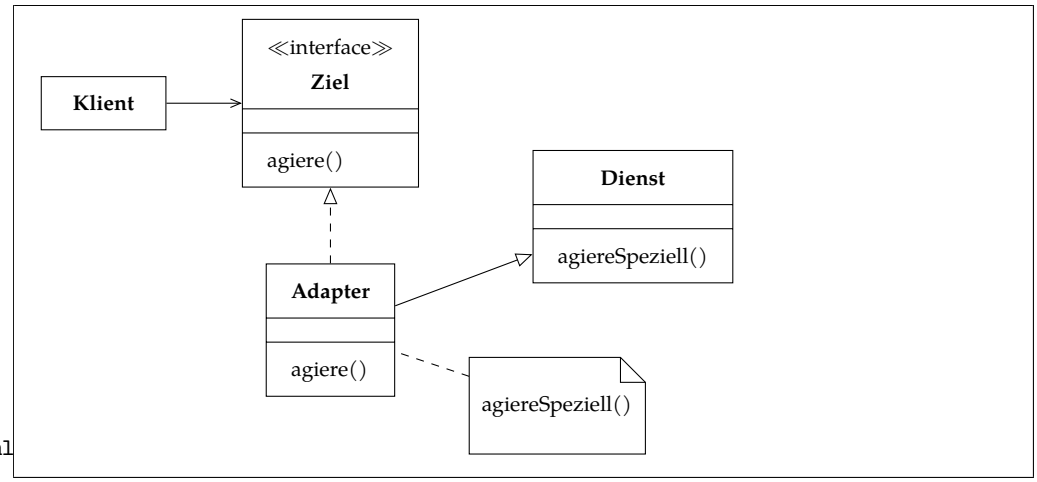
bEntwurfsAbstrakteFabrikCode

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

\bEntwurfsAbstrakteFabrik

```
\def\bEntwurfsAbstrakteFabrik{
  \bEntwurfsAbstrakteFabrikBeschreibung

  \bEntwurfsAbstrakteFabrikUml

  \bEntwurfsAbstrakteFabrikCode
}
```

Adapter



\bEntwurfsAdapterUml

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

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

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

\bEntwurfsAdapterAkteure

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

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

Dienst (Adaptee) Der Dienst bietet wiederzuverwendende Dienstleistungen mit

fest definierter Schnittstelle an.

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

```
\def\bEntwurfsAdapterAkteure{
  \begin{description}

    \item[Ziel (Target)]

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

    \item[Klient (Client)]

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

    \item[Dienst (Adaptee)]

    Der Dienst bietet wiederzuverwendende Dienstleistungen mit fest
    definierter Schnittstelle an.

    \item[Adapter]

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

  \end{description}
}
```

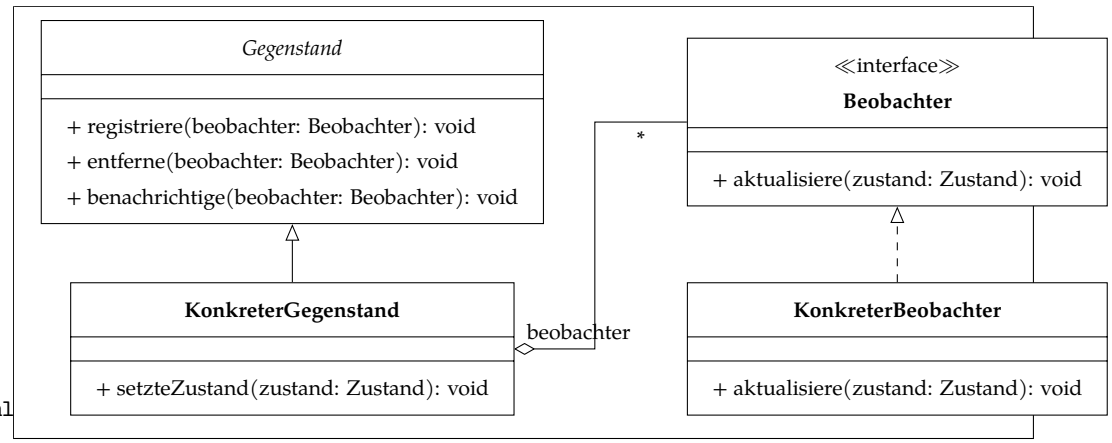
\bEntwurfsAdapterCode

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

\bEntwurfsAdapter

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

Beobachter (Observer)



\bEntwurfsBeobachterUml

```
\def\bEntwurfsBeobachterUml{
```

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

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

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

\bEntwurfsBeobachterAkteure

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

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

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

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

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

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

    \item[Beobachter (Observer)]

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

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

```

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

    \item[Konkrete Beobachter (ConcreteObserver)]

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

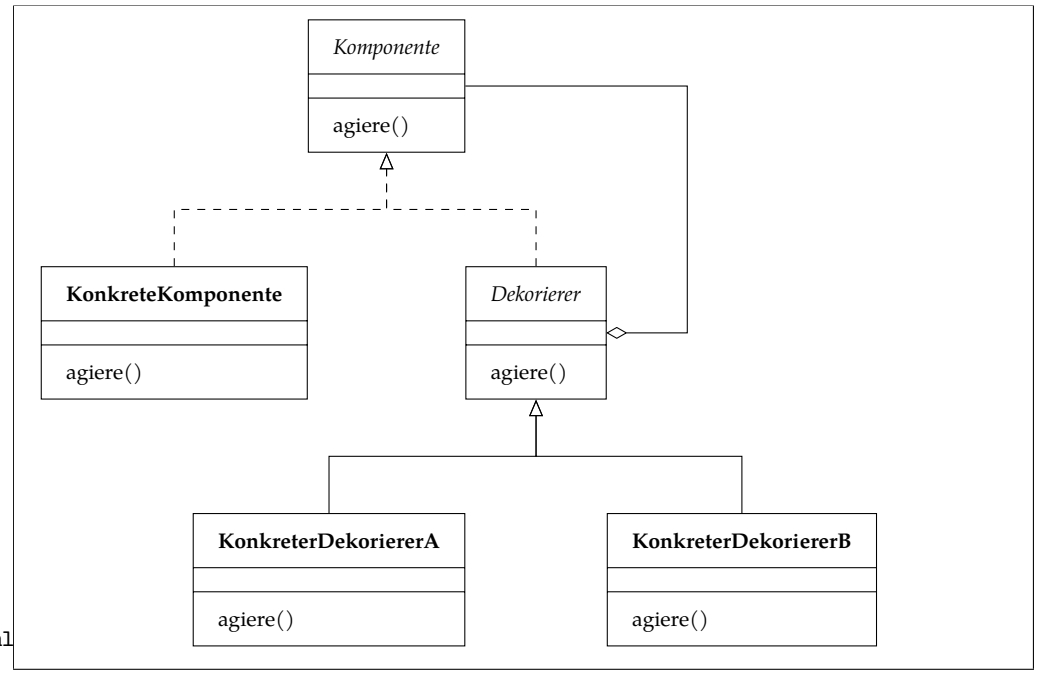
\bEntwurfsBeobachterCode

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

\bEntwurfsBeobachter

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

Dekorierer (Decorator)



\bEntwurfsDekoriererUml

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

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

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

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

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

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

\bEntwurfsDekoriererCode

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

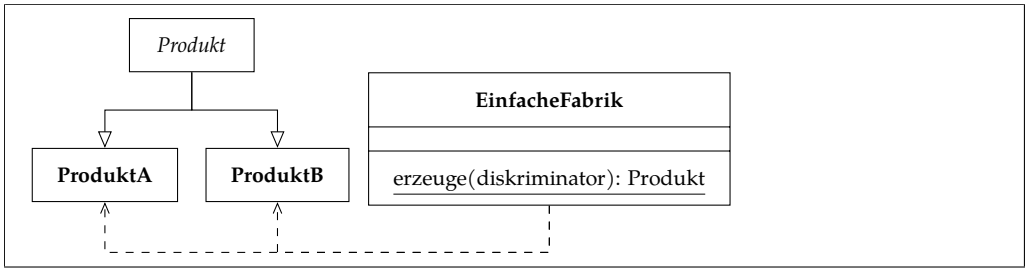
\bEntwurfsDekorierer

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

Einfache Fabrik (Simple Factory)

\bEntwurfsEinfacheFabrikUml

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




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

ntwurfsEinfacheFabrikAkteure

Quelle: frei überstetzt aus GoF

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

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

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

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

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

    \item[Produkt]

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

    \item[KonkretesProdukt]

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

\bEntwurfsEinfacheFabrik

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

Einzelstück (Singleton)

urfsEinzelstueckBeschreibung

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

```

\bEntwurfsEinzelstueckUml

Quelle nach der deutschen Wikipedia

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

bEntwurfsEinzelstueckAkteure

Quelle: frei überstetzt aus GoF

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

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

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

\bEntwurfsEinzelstueckCode

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

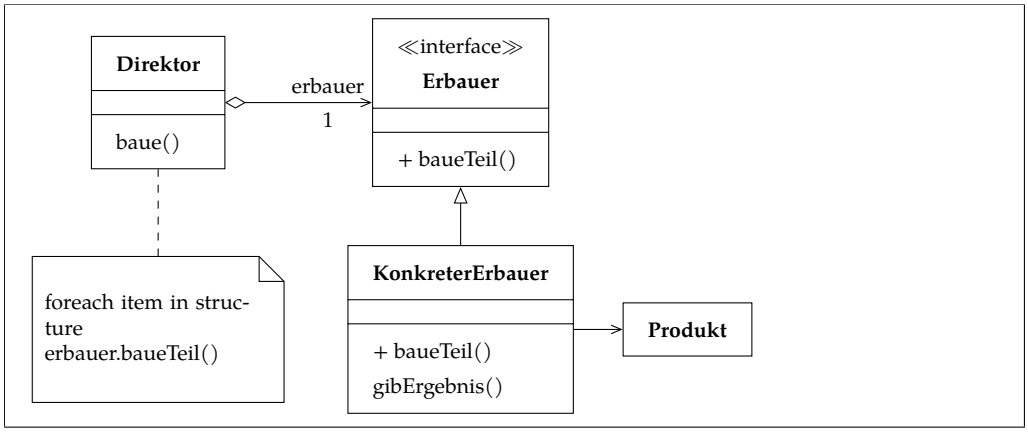
\bEntwurfsEinzelstueck

```
\def\bEntwurfsEinzelstueck{
  \bEntwurfsEinzelstueckBeschreibung

  \bEntwurfsEinzelstueckUml

  \bEntwurfsEinzelstueckAkteure

  \bEntwurfsEinzelstueckCode
}
```

Erbauer (Builder)

\bEntwurfsErbauerUml

Quelle nach der deutschen Wikipedia




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

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

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

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

\bEntwurfsErbauerAkteure

Quelle: deutsche Wikipedia

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

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

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

Produkt Das Produkt repräsentiert das zu konstruierende komplexe Objekt.

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

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

    \item[KonkreterErbauer]

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

    \item[Direktor]

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

    \item[Produkt]

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

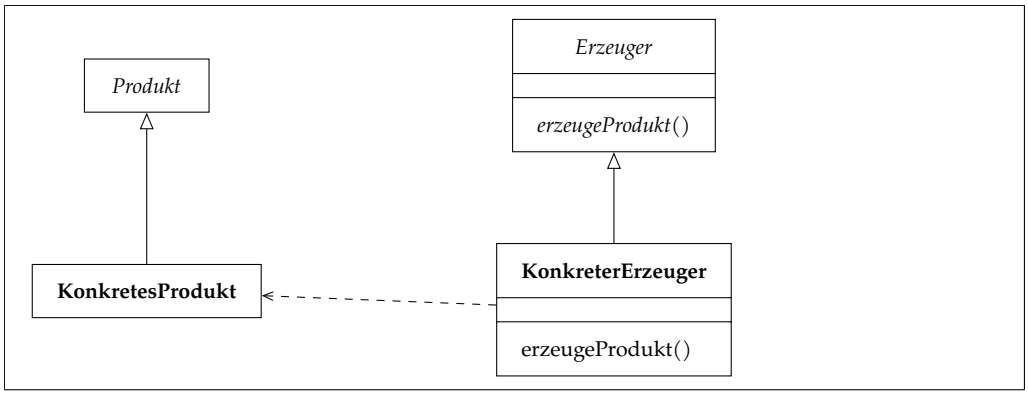
\bEntwurfsErbauer

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

Fabrikmethode (Factory Method)

\bEntwurfsFabrikmethodeUml

Quelle nach der deutschen Wikipedia




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

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

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

`EntwurfsFabrikmethodeAkteure` Quelle: deutsche Wikipedia

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

KonkretesProdukt KonkretesProdukt implementiert die Produkt-Schnittstelle.

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

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

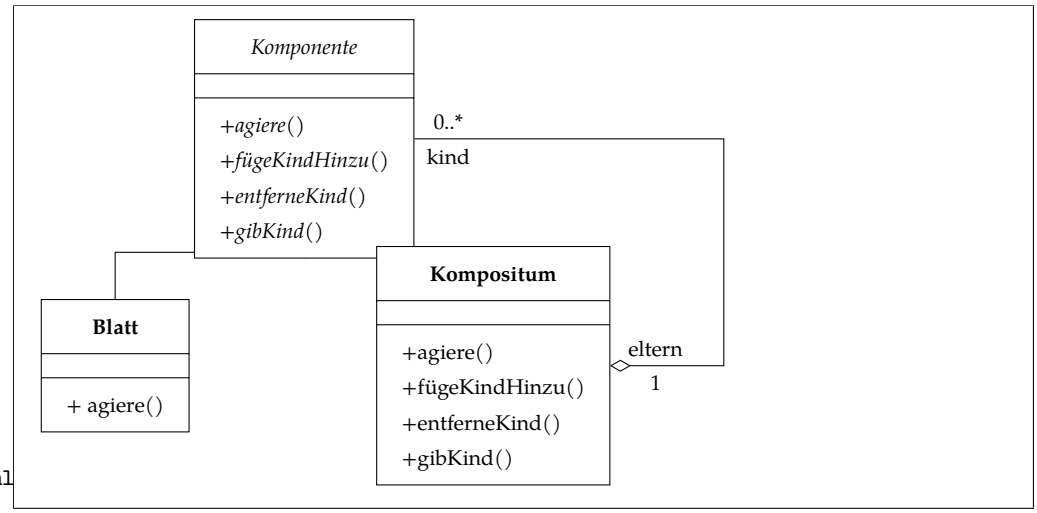
```
\def\bEntwurfsFabrikmethodeAkteure{
  \begin{description}
    \item[Produkt]

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

    \item[KonkretesProdukt]

    KonkretesProdukt implementiert die Produkt-Schnittstelle.

    \item[Erzeuger]

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

    \item[KonkreterErzeuger]

```

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

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

\bEntwurfsFabrikmethode

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

Kompositum (Composite)



\bEntwurfsKompositumUml

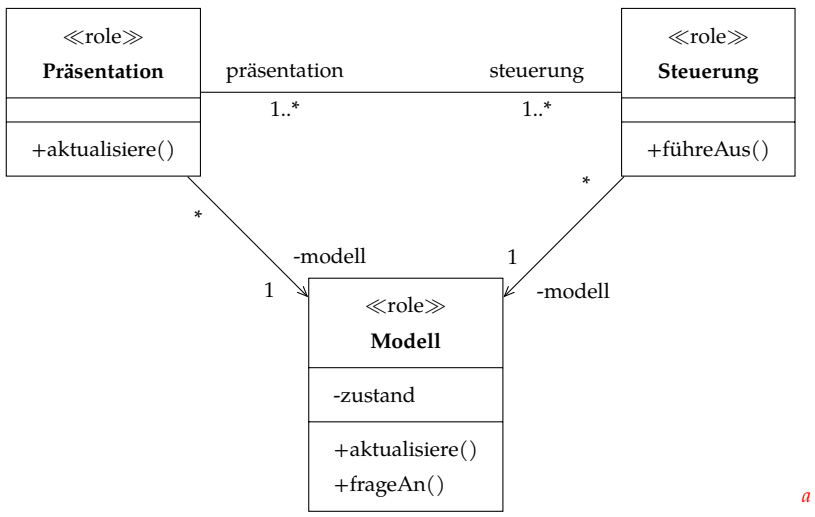
```
\def\bEntwurfsKompositumUml{
  \begin{tikzpicture}
    \umlclass[x=2.5,y=3,type=abstract]{Komponente}{}{
      \textit{+agiere()}\\
      \textit{+fügeKindHinzu()}\\
      \textit{+entferneKind()}\\
      \textit{+gibKind()}
    }
    \umlclass[x=0]{Blatt}{}{+ agiere()}
    \umlclass[x=5]{Kompositum}{}{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

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

\bEntwurfsFabrikmethode

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

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

odellPraesentationSteuerungUml



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

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

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

ModellPraesentationSteuerung

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

Stellvertreter (Proxy)

\bEntwurfsStellvertreterUml

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

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

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

\bEntwurfsStellvertreterCode

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

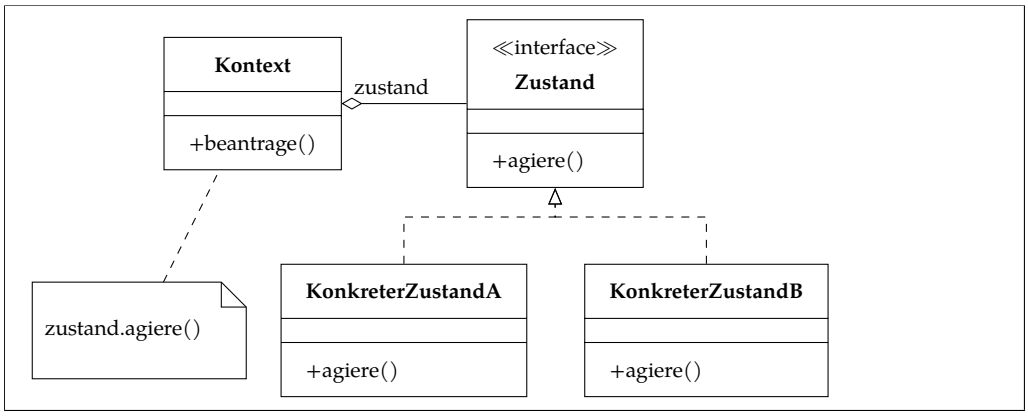
\bEntwurfsStellvertreter

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

Zustand (State)

\bEntwurfsZustandUml

Quelle: Englische Wikipedia, so ähnlich wie in GoF




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

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

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

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

\bEntwurfsZustandAkteure

Quelle: Deutsche Wikipedia

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

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

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

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

    definiert die clientseitige Schnittstelle und verwaltet die separaten
```

```
    Zustandsklassen.

    \item[State (Zustand)]

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

    \item[KontreterZustand (ConcreteState)]

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

\bEntwurfsZustandCode

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

\bEntwurfsZustand

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

er.sty

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

Tex-Markup-Beispiel: Komplettes Diagramm

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Faulenzer

```
\let\a=\bErMpAttribute
\let\d=\bErDatenbankName
\let\e=\bErMpEntity
\let\r=\bErMpRelationship
```

```
\ExplSyntaxOn
```

`\bErEntity`

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

`\bErRelationship`

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

`\bErAttribute`

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

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

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

□

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

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

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

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

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

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

```
\ExplSyntaxOff
```

formale-sprachen.sty

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

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

```
\RequirePackage{hyperref}
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`\bZustandsBuchstabe`

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

`\bZustandsBuchstabeGross`

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

\bZustandsmengeNr

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

\bZustandsmengeNrGross

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

\bZustandsname

\bZustandsname{1}: z_1

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

\bZustandsnameGross

\bZustandsnameGross{1}: Z_1

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

\bAbleitung

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

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

bProduktionsRegeln

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

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

\bProduktionen

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

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

\bZustandsnameTiefgestellt

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

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

```
\ExplSyntaxOn
```

\bAusdruck

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

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

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

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

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

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

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

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

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

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

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

  \keys_set:nn { grammatik } { #2 }

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

formatierung.sty

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

Schriftarten / Typographie

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

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

Farben

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

Überschriften

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

Listen

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

Kasten

```
\RequirePackage{mdframed}
```

`liKasten`

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

Header

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

Zeilenabstände

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

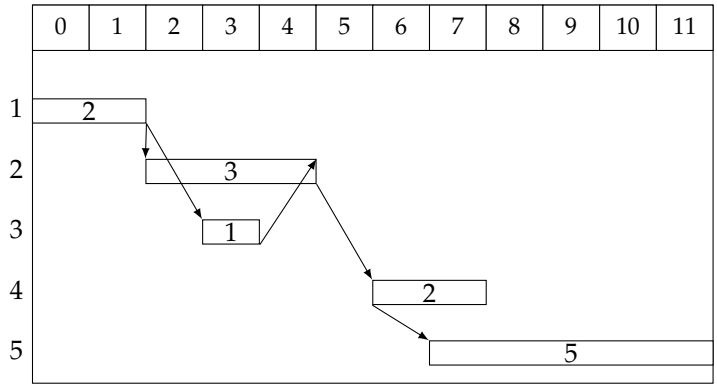
```
\RequirePackage{setspace}
```

1886

gantt.sty

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

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

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

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




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

grafik.sty

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

```
\ExplSyntaxOn
```

```
\RequirePackage{tikz}
```

```
\RequirePackage{graphicx}
```

\bGrafikLogoPfad

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

\bGrafikCCLizenz

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

\bGrafikLogo

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

```
\ExplSyntaxOff
```

graph.sty

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

Für die die Adjazenzliste (xrightarrow)

```
\RequirePackage{amsmath}
```

Für Adjazenz-Matrix

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

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

```
\RequirePackage{blkarray}
```

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

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

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

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

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

```
\begin{bGraphenFormat}
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{bGraphenFormat}
```

```
\NewDocumentEnvironment { bGraphenFormat }{ +b } {} {}
```

hanoi.sty

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

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

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

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

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

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

index.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-index}[2021/09/12 Den Index anzeigen
lassen]
```

```
\ExplSyntaxOn
```

```
\RequirePackage{makeidx}
```

Anzeigen des Indexes auf der rechten Seite

```
\RequirePackage{showidx}
```

Überschreiben des Makros aus showidx um kleinere Schrift zu bekommen

```
\def\@showidx#1
{%
\insert
\indexbox
  {
    \tiny
    \hsize\marginparwidth
    \hangindent\marginparsep \parindent\z@
    \everypar{}\let\par\@@par \parfillskip\@flushglue
    \lineskip\normallineskip
    \baselineskip .8\normalbaselineskip\sloppy
    \raggedright \leavevmode
    \vrule \@height .7\normalbaselineskip \@width \z@\relax
    #1\relax
    \vrule \@height \z@ \@depth .3\normalbaselineskip \@width \z@
  }
}
```

```
\makeindex
```

```
\ExplSyntaxOff
```

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=\bProblemName
\let\r=\bPolynomiellReduzierbar
\let\b=\bProblemBeschreibung
```

```
\bLadePakete{mathe}
```

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

```
\RequirePackage{mdframed}
```

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

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

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

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

`\bProblemBeschreibung` Zu setzen von Problem-Beschreibungen:

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

CLIQUE

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

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

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

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

    \medskip

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

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

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

\bProblemVertexCover

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

\bProblemVertexCover

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

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

\bProblemSubsetSum Kein Karp-21-Problem

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

\bProblemSubsetSum Kein Karp-21-Problem

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

```

kontrollflussgraph.sty

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

Faulenzer

```
\let\b=\bBedingung
\let\c=\bKontrollCode
\let\f=\bBedingungFalsch
\let\k=\bKontrollTextzeileKnoten
\let\p=\bKontrollKnotenPfad
\let\w=\bBedingungWahr
```

TeX-Markup-Beispiel

```
\begin{bKontrollflussgraph}[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{bKontrollflussgraph}
```

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

bKontrollflussgraph

```
\NewDocumentEnvironment { bKontrollflussgraph } { O{} } {
```

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

Makros

\bAnweisung

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

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

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

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

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

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

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

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

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

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

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

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

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

kopfzeile.sty

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

```
\ExplSyntaxOn
```

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

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

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

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

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

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

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

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

```
\ExplSyntaxOff
```

literatur-dummy.sty

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

\literatur

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

\footcite

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

```

literatur.sty

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

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

`\literatur`

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

makros.sty

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

```
\RequirePackage{hyperref}
```

```
\RequirePackage{graphicx}
```

Für die Umgebung `bQuellen` benötigt.

```
\RequirePackage{paralist}
```

```
\ExplSyntaxOn
```

`\inhaltsverzeichnis`

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

`\bEmph` `\bEmph` (`\marginpar` and `\emph`)

```
\def\bEmph#1
{
  \emph{#1}
  \marginpar{
    \tiny#1
  }
}
```

`\SLASH`

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

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

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

  \par
  \noindent
  \textbf{#1}

  \medskip

  \keine_einrueckung:
}
```

`bProjektSprache` `\begin{bProjektSprache}{NameProjektSprache} \end{bProjektSprache}`: 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 { bProjektSprache }{ 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{ bAdditum } { o }
{
  \ifADDITUM
  \else
    % Alles in eine Box verschieben und die dann ignorieren.
    \setbox 0 \vbox
    \bgroup
  \fi

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

  \ifADDITUM
  \else
    \egroup
  \fi
}
```

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

```
\NewDocumentEnvironment{ bAntwort } { O{standard} }
{
  \ifANTWORT
  \else
    \setbox 0 \vbox
    \bgroup
  \fi

  \str_case:nn {#1} {
    {standard} {
      \def\beschriftung{}
      \mdfsetup{backgroundcolor=gray!1}
    }
    {richtig} {
      \def\beschriftung{richtig}
      \mdfsetup{backgroundcolor=green!5,linecolor=green}
    }
    {falsch} {
      \def\beschriftung{falsch}
      \mdfsetup{backgroundcolor=red!3,linecolor=red}
    }
    {muster} {
      \def\beschriftung{Musterlösung}
      \mdfsetup{backgroundcolor=green!12,linecolor=green}
    }
  }
  \ifx\beschriftung\empty\else
    \noindent
    \textbf{\beschriftung{}:}
  \fi
  \begin{mdframed}[
    frametitle={\scriptsize\mdseries Lösungsvorschlag},
    innertopmargin=6pt,
    frametitleaboveskip=-10pt,
    frametitlealignment=\raggedleft
  ]
```

```
}
{
  \end{mdframed}
  \ifANTWORT
  \else
    \egroup
  \fi
}
```

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

```
\NewDocumentEnvironment{ bAdditum } { o }
{
  \ifADDITUM
  \else
    \setbox 0 \vbox
    \bgroup
  \fi

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

  \ifADDITUM
  \else
    \egroup
  \fi
}
```

bExkurs

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

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

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

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

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

Weiterführende Literatur:

- Quelle 1
- Quelle 2

```
\cs_new:Npn \listen_punkt:n #1
{
  \item #1
}
\NewDocumentEnvironment { bQuellen }{ +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:n}
    \end{compactitem}
  \end{mdframed}

  \keine_einrueckung:
} {}
```

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

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

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

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

\bLinie Eine horizontale Linie

```
\def\bLinie{
  \par
  \noindent
  \rule
  {
    \textwidth
  }
  {
    0.4pt
  }
}
```

\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=\bO
\let\o=\bOmega
\let\T=\bT
\let\t=\bTheta
```

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

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

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

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

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

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

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

\bRundeKlammer

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

\bTheta

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
\ExplSyntaxOff
```

\bMasterVariablen

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

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

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

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

\bMasterFaelle

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

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

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

  \hfill falls \bBedingungZwei

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

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

\bMasterVariablenDeklaration

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

    \bRekursionsGleichung

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

    #1

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

```

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

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

    $#3$

    \item[Ergibt folgende Rekursionsgleichung:] \strut

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

\bMasterFallRechnung

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

  #1

  \item[2. Fall:] \bBedingungZwei:

  #2

  \item[3. Fall:] \bBedingungDrei:

  #3
  \end{description}
}
```

\bMasterExkurs

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

  \noindent
  Dann gilt:

  \bMasterFaelle
  \end{bExkurs}
}
```

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

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

mathe.sty

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

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

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

```

meta.sty

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

```
\ExplSyntaxOn
```

```
\bLadePakete{grafik}
```

Einfache Makros (Low level)

\bMetaBschlangaulSammlung

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

\bMetaHermineFriends

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

\bMetaUeberDasProjekt

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

\bMetaCCLink

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

\bMetaEmail

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

\bMetaEmailLink

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

\bMetaHilfMit

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

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

  Das~ist~ein~Community-Projekt.~

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

\bMetaHilfMit

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

Zusammengesetzte Makros (High level)

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

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

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

\bLogoTextProjekt

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

    \bMetaHermineFriends
    \par

    \medskip

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

\bLogoTextCCLizenz

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

\ExplSyntaxOff

```

minimierung.sty

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

```
\bLadePakete{typographie}
```

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

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

\bFussnoten

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

\bFussnote

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

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

\bFussnoteEinsText

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

\bFussnoteZweiText

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

\bFussnoteDreiText

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

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

\bFussnoteVierText

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

\bFussnoten

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

```
\def\bFussnoten{
  \bigskip

  \noindent
  \bFussnoteEinsText

  \noindent
  \bFussnoteZweiText

  \noindent
  \bFussnoteDreiText

  \noindent
  \bFussnoteVierText
}
```

\bLeereZelle

\bLeereZelle: ∅

Makro-Faulenzer: \let\l=\liLeereZelle

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

\bZustandsPaarVariablenName

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

\bZustandsPaar

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

liUebergangsTabelle

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

bUeberschriftDreiecksTabelle

\bUeberschriftDreiecksTabelle:

Minimierungstabelle (Table filling)

```
\ExplSyntaxOn
```

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

\bMinimierungErklaerung

Makro-Faulenzer: `\let\erklaerung=\liMinimierungErklaerung`

— Der Minimierungs-Algorithmus (auch Table-Filling-Algorithmus genannt) trägt in seinem Verlauf eine Markierung in alle diejenigen Zellen der Tabelle ein, die zueinander nicht äquivalente Zustände bezeichnen. Die Markierung „x_n“ in einer Tabellenzelle (i, j) bedeutet dabei, dass das Zustandspaar (i, j) in der k-ten Iteration des Algorithmus markiert wurde und die Zustände i und j somit zueinander $(k-1)$-äquivalent, aber nicht k-äquivalent und somit insbesondere nicht äquivalent sind. Bleibt eine Zelle bis zum Ende unmarkiert, sind die entsprechenden Zustände zueinander äquivalent. ——

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

normalformen.sty

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

Mathe für textit tex etc, typographie für geschweifte Klammer

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

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

Faulenzer

```
\let\ah=\bAttributHuelle
\let\ahL=\bLinksReduktion
\let\ahl=\bLinksReduktionInline
\let\ahr=\bRechtsReduktionInline
\let\fa=\bFunktionaleAbhaengigkeit
\let\FA=\bFunktionaleAbhaengigkeiten
\let\m=\bAttributMenge
\let\r=\bRelation
\let\u=\underline
```

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

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

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

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

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

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

`liAHuelle`

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

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

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

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

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

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

\bLinksReduktionInline

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

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

\bLinksReduktionInline

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

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

\bFunktionaleAbhaengigkeit

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

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

\bFunktionaleAbhaengigkeiten

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

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

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

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

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

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

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

o-notation.sty

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

Faulenzer

```
\let\O=\bONotationO
```

TeX-Markup-Beispiel: Funktionsgraphen mit pgfplots

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

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

Für echte Teilmenge \subsetneq: $\subsetneq$

```
\RequirePackage{amssymb}
```

\bRundeKlammer

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

\bONotationO

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

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

petri.sty

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

Faulenzer

```
\let\t=\bPetriTransitionsName
\let\tp=\bPetriTransPfeile
\let\k=\bPetriErreichKnotenDrei
```

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

Für die Darstellungsmatrix

```
\RequirePackage{blkarray}
```

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

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

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

\bPetriSetzeSchluessel

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

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

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

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

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

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

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

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

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

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

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

potenzmengen-konstruktion.sty

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

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

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

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

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

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

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

\bZustandsMengenSammlungNr

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

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

pseudo.sty

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

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

Algorithmus 1: Minimaler Spannbaum nach Kruskal

Data: $G = (V, E, w)$: ein zusammenhängender, ungerichteter, kantengewichteter Graph kruskal(G)
$E' \leftarrow \emptyset$;
$L \leftarrow E$;
Sortiere die Kanten in L aufsteigend nach ihrem Kantengewicht.;
while $L \neq \emptyset$ **do**
  wähle eine Kante $e \in L$ mit kleinstem Kantengewicht;
  entferne die Kante e aus L;
  if *der Graph* $(V, E' \cup \{e\})$ *keinen Kreis enthält* **then**
    $E' \leftarrow E' \cup \{e\}$;
  end
end
Result: $M = (V, E')$ ist ein minimaler Spannbaum von G.

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

pumping-lemma.sty

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

\bPumpingRegulaer

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

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

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

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

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

\bPumpingKontextfrei

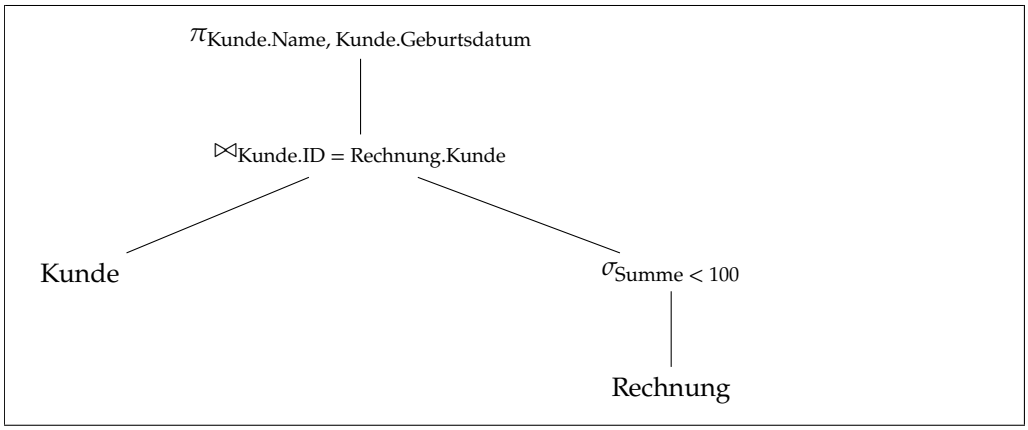
```
\def\bPumpingKontextfrei{%
  Es sei $L$ eine kontextfreie Sprache. Dann gibt es eine Zahl $j$, sodass
  sich alle Wörter $\omega \in L$ mit $|\omega| \geq j$ zerlegen lassen in
  $\omega = uvwxy$, sodass die folgenden Eigenschaften erfüllt sind:

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

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

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

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=\bAttribut
\let\f=\bFremd
\let\p=\bPrimaer
\let\r=\bRelationMenge
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

sortieren.sty

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

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

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

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

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

`\bSortierPfeil`

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

`\bSortierPfeilUnten`

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

`\bSortierMarkierung`

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

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

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

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

\def\myNodes{}

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

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

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

spalten.sty

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

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

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

sql.sty

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

Latex-Markup-Beispiel

```
\begin{bAdditum}[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{bAdditum}
```

```
\bLadePakete{syntax}
```

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

struktogramm.sty

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

\RequirePackage{struktex}

```

syntax.sty

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

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

Faulenzer

```
\let\j=\bJavaCode
\let\s=\bSqlCode
```

```
\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,
}
```

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

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

`\bJavaCode` Im Zeilenfluss einen kurzen Java-Code-Ausschnitt setzen. Es werden automatische Zeilenumbrüche gemacht.

Makro-Faulenzer: `\let\j=\liJavaCode`

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

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

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

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

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

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

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

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

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

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

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

\bAssemblerCode

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

\bAssemblerDatei

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

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

\bMinispracheDatei

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

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

\bHaskellCode

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

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

\bHaskellDatei

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

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

\bSqlCode

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

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

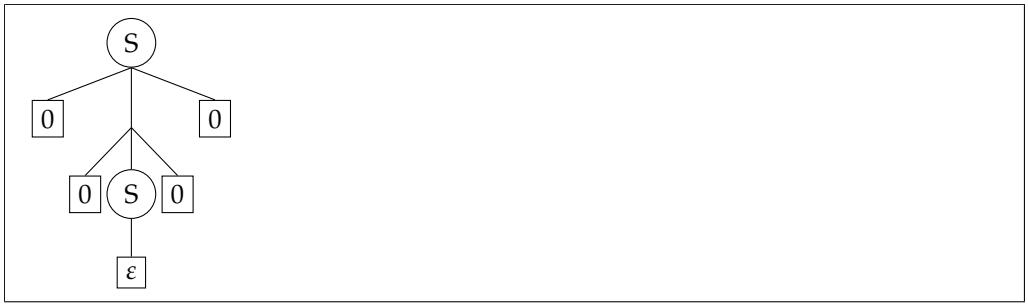
```
\ExplSyntaxOff
```

syntaxbaum.sty

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

TeX-Markup-Beispiel

```
\begin{tikzpicture}[b syntaxbaum,level distance=1cm]
\Tree [.S 0 [ 0 [.S $\varepsilon$ ] 0 ] 0 ]
\end{tikzpicture}
```




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

```
\tikzset{b syntaxbaum/.style={
    every internal node/.style={
      draw,circle
    },
    every leaf node/.style={
      draw,rectangle
    },
  }
}
```

synthese-algorithmus.sty

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

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

```
\ExplSyntaxOn
```

Faulenzer

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

TeX-Markup Grundgerüst

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

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

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

TeX-Markup Linksreduktion

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

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

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

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

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

TeX-Markup Rechtreduktion

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

\bPseudoUeberschrift{F}

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

```
\bPseudoUeberschrift{A}

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

TeX-Markup Relationen formen

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

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

(a) **Kanonische Überdeckung**

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

(i) **Linksreduktion**

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

(ii) **Rechtsreduktion**

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

(iii) **Löschen leerer Klauseln**

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

(iv) **Vereinigung**

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

(b) **Relationsschemata formen**

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

(c) **Schlüssel hinzufügen**

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

(d) **Entfernung überflüssiger Teilschemata**

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

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

```
\def\bSyntheseUeberschrift#1
{
  {
    \bfseries
    \rmfamily
    \str_case:nn {#1} {
      {1} {Kanonische~Überdeckung}
      {1-1} {Linksreduktion}
      {1-2} {Rechtsreduktion}
```

```
      {1-3} {Löschen~leerer~Klauseln}
      {1-4} {Vereinigung}
      {2} {Relationsschemata~formen}
      {3} {Schlüssel~hinzufügen}
      {4} {Entfernung~überflüssiger~Teilschemata}
    }
  }
}
```

\bSyntheseErklaerung **Makro-Faulenzer:** `\let\erklaerung=\liSyntheseErklaerung`

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

```
    }
  }
}
\def\bSyntheseErklaerung#1
{
  {
    \itshape
    \footnotesize
    \bParagraphMitLinien{\li@synthese@erklaerung@texte{#1}}
  }
}
```

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

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

\ExplSyntaxOff
```

tabelle.sty

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

```

tex-dokumentation.sty

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

```
\ExplSyntaxOn
```

```
\RequirePackage{mdframed}
```

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

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

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

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

\bFaulenzer

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

```
\ExplSyntaxOff
```

typographie.sty

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

```
\ExplSyntaxOn
```

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

```
\RequirePackage{fontawesome}
```

`\bErledigt` `\bErledigt`: ☑

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

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

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

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

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

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

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

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

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

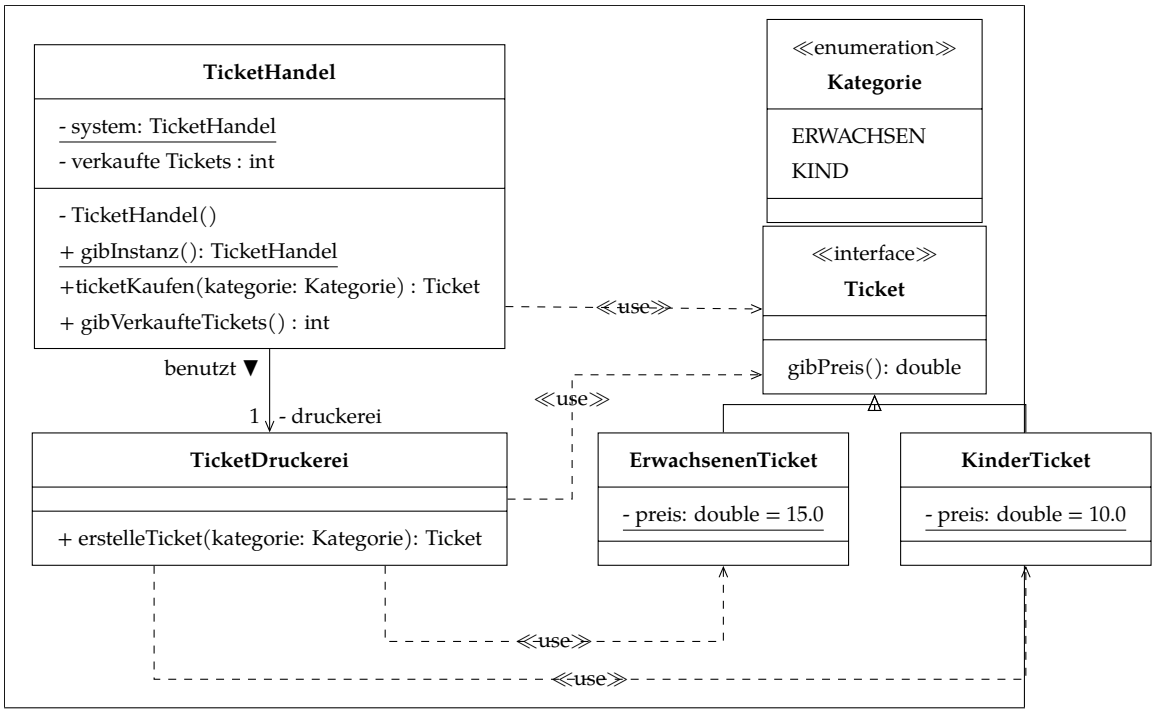
```
\ExplSyntaxOff
```

uml.sty

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

Not compatible with wasysym

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



\bUmlLeserichtung

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

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

  \def\@liPos{above}
```

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

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

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

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

vollstaendige-induktion.sty

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

Faulenzer

```
\let\m=\bInduktionMarkierung
\let\e=\bInduktionErklaerung
```

```
\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

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

```
\ExplSyntaxOn
```

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

```
\def\bInduktionMarkierung#1
{
```

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

\bInduktionErklaerung

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

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

\bInduktionAnfang

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

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

\bInduktionVoraussetzung

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

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

\bInduktionSchritt

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

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

```
\ExplSyntaxOff
```

wasserfall.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-wasserfall}[2020/06/10 Zu Setzen des
Diagramms zum Wasserfallmodell]
```

Latex-Markup-Beispiel

```
\begin{tikzpicture}[b wasserfall]
\node {Systemanforderung}; % A-1
\node {Softwareanforderung};
\node {Analyse};
\node {Entwurf};
\node {Implementierung};
\node {Test};
\node {Betrieb};

\foreach \i [count=\j] in {2,...,7}
{
  \draw[->, thick] (A-\i) -| (A-\j);
  \draw[->, thick] (A-\j) -| (A-\i);
}
\end{tikzpicture}
```

```
\RequirePackage{tikz}
\usetikzlibrary{chains,positioning,shadows}
\tikzset{
  b 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
    },
  }
}
```

wpkalkuel.sty

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{bschlangaul-wpkalkuel}[2020/06/13 Hilfsmakros zum
Setzen des WP-Kalküls]
```

Faulenzer

```
\let\wp=\bWpKalkuel
\let\equivalent=\bWpEquivalent
\let\erklaerung=\bWpErklaerung
```

```
\RequirePackage{amsmath}
```

```
\ExplSyntaxOn
```

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

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

```
\cs_new:Npn \mathe_umgebung:n #1
{
  \medskip

  \hspace{1em}#1

  \medskip
}
```

\bWpPseudoMatheUmgebung

```
\def\bWpPseudoMatheUmgebung#1
{
  \mathe_umgebung:n{$#1$}
}
```

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

```
\def\bWpEquivalent#1
{
  \mathe_umgebung:n {
    $\equiv$
    \hspace{1em}
    $#1$
  }
}
```

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

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

  \par
  \noindent
  {
```

```
    \scriptsize
    #1
  }
  \par

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

\bWpErklaerungVerzweigung

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

\ExplSyntaxOff
```

Index

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

Symbols

\# 486
\, . . 737, 807, 862, 1158, 1176, 1808, 1810, 2419, 2420, 2421, 2872, 3209, 3216, 3477
\@@par 1998
\@Skip@Erklaerung@Reset 3632, 3635, 3646
\@afterheading 922
\@afterindentfalse . . 921
\@depth 2004
\@flushglue 1998
\@height 2002, 2004
\@liDirLeft 3507, 3512, 3524
\@liDirRight 3508, 3510, 3511, 3512, 3513, 3524
\@liDistance 3518, 3519, 3523
\@liPos . . . 3515, 3516, 3523
\@showidx 1990
\@width 2002, 2004
\\ . . . 155, 157, 181, 185, 189, 1188, 1216, 1217, 1220, 1221, 1224, 1225, 1320, 1321, 1322, 1429, 1463, 1465, 1495, 1504, 1549, 1591, 1592, 1593, 1598, 1599, 1600, 1620, 2255, 2767, 2825, 2828
\{ 584, 1745, 1755, 1767, 1768, 1773, 1807, 2065, 2812, 3070, 3477, 3650
\} 584, 1745, 1755, 1767, 1768, 1775, 1811, 2066, 2812, 3070, 3481, 3650
_ . 47, 61, 101, 242, 409, 417, 422, 424, 682, 712, 736, 739, 749, 772, 775, 779, 781, 783, 785, 788, 800, 801, 802, 805, 807, 821, 822, 828, 831, 834, 844, 851, 853, 856
\␣ 2824, 2834

A

\addbibresource 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227
\ADDITUMfalse 600
\ADDITUMtrue 16, 600
\AddToHook 81, 85, 284, 328, 348
\advance 3119
\AfterEndEnvironment 3195
\Alph 1868
\alph 1868, 1869
\alpha . 3341, 3343, 3344, 3347, 3349, 3350, 3351, 3352, 3353, 3357, 3363, 3364, 3369, 3370, 3371, 3374, 3382, 3383, 3384
\ANTWORTfalse 604
\ANTWORTtrue 604
\arabic 1868
\arraystretch 2762
\AtBeginDocument 178

B

\bAbleitung 1782
bAdditum (environment) 2313
\bAlphabet 1767
bAntwort (environment) 2269
\bAnweisung 2150
\baselineskip 2000
\bAssemblerCode 3264
\bAssemblerDatei . . . 3268
\bAttribut 3073
\bAttributHuelle 2804, 3344, 3349
\bAttributHuelleOhneMathe 2804, 2807, 2809, 2823, 2833, 2841
\bAttributMenge 2812, 2824, 2827, 2834, 2835, 2849, 2851
\bAufgabe 388
\bAufgabenMetadaten 38, 240, 352, 419
\bAufgabenTitel 427
\bAusdruck 1804
\bAutomat 438
\bAutomatenKante 470
\bBandAlphabet 1768
bBaum (environment) . . 990

\bBedingung 2151
\bBedingungDrei 2484, 2527, 2567
\bBedingungEins 2478, 2516, 2559
\bBedingungFalsch .. 2153
\bBedingungWahr 2152
\bBedingungZwei 2481, 2522, 2563
\bBindeAufgabeEin ... 223
\bBindePdfEin 177
\bChomskyErklaerung 1027, 1068
\bChomskyUeberErklaerung 1066
\bChomskyUeberschrift 1015, 1067
\bCpmEreignis 1076
\bCpmFruehErklaerung 1147
\bCpmFruehI ... 1140, 1160
\bCpmSpaetErklaerung 1165
\bCpmSpaetI ... 1133, 1178
\bCpmVon 1116
\bCpmVonOhneMathe 1116, 1119, 1121
\bCpmVonZu 1108
\bCpmVonZuOhneMathe 1108, 1111, 1113
\bCpmVorgang 1093
\bCpmZu 1124
\bCpmZuOhneMathe 1124, 1127, 1129
\BeforeBeginEnvironment 3194
\begin . 95, 153, 170, 874, 883, 891, 997, 1214, 1268, 1283, 1318, 1342, 1389, 1421, 1436, 1461, 1471, 1491, 1511, 1543, 1560, 1589, 1614, 1635, 1659, 1674, 1788, 1873, 1965, 2019, 2028, 2035, 2143, 2241, 2299, 2321, 2342, 2373, 2378, 2489, 2493, 2512, 2535, 2558, 2573, 2664, 2665, 2669, 2692, 2705, 2765, 2766, 2816, 2861, 3001, 3021, 3194, 3224, 3414, 3481
\begingroup 2242, 2814, 3115
\bEmph 2248
\bEntwurfs 1630
\bEntwurfsAbstrakteFabrik 1260
\bEntwurfsAbstrakteFabrikBeschreibung 1208, 1261
\bEntwurfsAbstrakteFabrikCode 1255, 1265
\bEntwurfsAbstrakteFabrikUml 1213, 1263
\bEntwurfsAdapter .. 1312
\bEntwurfsAdapterAkteure 1282, 1314
\bEntwurfsAdapterCode 1306, 1315
\bEntwurfsAdapterUml 1267, 1313
\bEntwurfsBeobachter 1383
\bEntwurfsBeobachterAkteure 1341, 1385
\bEntwurfsBeobachterCode 1375, 1386
\bEntwurfsBeobachterUml 1317, 1384
\bEntwurfsDekorierer 1415
\bEntwurfsDekoriererAkteure 1417
\bEntwurfsDekoriererCode 1407, 1418
\bEntwurfsDekoriererUml 1388, 1416
\bEntwurfsEinfacheFabrik 1451
\bEntwurfsEinfacheFabrikAkteure 1435, 1453
\bEntwurfsEinfacheFabrikUml 1420, 1452
\bEntwurfsEinzelstueck 1481
\bEntwurfsEinzelstueckAkteure 1470, 1486
\bEntwurfsEinzelstueckBeschreibung 1455, 1482
\bEntwurfsEinzelstueckCode 1478, 1488
\bEntwurfsEinzelstueckUml 1460, 1484
\bEntwurfsErbauer .. 1538
\bEntwurfsErbauerAkteure 1510, 1540
\bEntwurfsErbauerUml 1490, 1539
\bEntwurfsFabrikmethode 1584, 1609
\bEntwurfsFabrikmethodeAkteure 1559, 1586
\bEntwurfsFabrikmethodeUml 1542, 1585
\bEntwurfsKompositum 1609
\bEntwurfsKompositumAkteure 1611
\bEntwurfsKompositumUml 1588, 1610
\bEntwurfsModellPraesentationsmuster 1630
\bEntwurfsModellPraesentationsmusterAkteure 1632
\bEntwurfsModellPraesentationsmusterUml 1613, 1631
\bEntwurfsStellvertreter 1654
\bEntwurfsStellvertreterCode 1648, 1656
\bEntwurfsStellvertreterUml 1634, 1655
\bEntwurfsZustand .. 1695
\bEntwurfsZustandAkteure 1673, 1697
\bEntwurfsZustandCode 1691, 1698
\bEntwurfsZustandUml 1658, 1696
\bEpsilon 1753
\bErAttribute 1711, 1725, 1727
\bErDatenbankName .. 1730
\bErEntity 1709, 1713, 1715
\bErledigt 3459
\bErMpAttribute 1724
\bErMpEntity 1712
\bErMpRelationship . 1718
\bErRelationship 1710, 1719, 1721
\beschriftung 2279, 2283, 2287, 2291, 2295, 2297
\beta 3341, 3344, 3348, 3349, 3350, 3353, 3363, 3364, 3365, 3370, 3371
\bExamensAufgabe 391
\bExamensAufgabeA ... 400
\bExamensAufgabeTA .. 397
\bExamensAufgabeTTA . 394
\bExamens (environment) 2338
\bFalsch 1008
\bFaulenzer 3438
\bFlaci 1814
\bFremd 3053
\bfseries . 54, 144, 209, 249, 875, 898, 1017, 1858, 1860, 2682, 3317
\bFunktionaleAbhaengigkeit 2843, 2846, 2854
\bFunktionaleAbhaengigkeiten 2857
\bFussnote 2717, 2719
\bFussnoteDreiText 2731, 2749
\bFussnoteEinsText 2723, 2743
\bFussnoteLink 2397
\bFussnoten 2739
\bFussnoteUrl .. 1628, 2385
\bFussnoteVierText 2735, 2752
\bFussnoteZweiText 2727, 2746
\bGleichheitsklammern 1786, 2858, 3472
\bGrafikCGUnd 1905, 2702
\bGrafikLogo 892, 1910, 2678
\bGrafikLogoPfad 1902, 1907, 1912

\bGrammatik 1825
bGraphenFormat (environment) 1944
\bgroup 2274, 2318
\bHanoi 1951
\bHaskellCode 3276
\bHaskellDatei 3280
\Bigl 3477
\Bigr 3481
\bigskip 76, 90, 93, 295, 298, 840, 1190, 1195, 2257, 2740, 3449
\bInduktionAnfang .. 3546
\bInduktionErklaerung 3542
\bInduktionMarkierung 3538
\bInduktionSchritt . 3562
\bInduktionVoraussetzung 3554
bJavaAngabe (environment) 3203
\bJavaCode 3207
\bJavaDatei ... 1206, 3230
\bJavaExamen 3252
\bJavaTestDatei 3241
\bKellerAutomat 479
\bKellerKante 517
\bKellerUebergang 512, 518
\bKontrollCode 2154
bKontrollflussgraph (environment) . 2142
\bKontrollKnotenPfad 2156
\bKontrollTextzeileKnoten 2155, 2160
\bKurzeTabellenLinie 1188
\bLadeAllePakete 212, 320, 610
\bLadePakete ... 5, 113, 120, 132, 363, 431, 434, 606, 611, 1014, 1075, 1744, 2014, 2604, 2716, 2796, 2964, 3166, 3308, 3532
\bLatexCode 3218
\bLeereZelle 2754
\bLinie 2408
\bLinksReduktion ... 2821
\bLinksReduktionInline 2830, 2838
\bLogoTextCCLizenz 92, 297, 2698
\bLogoTextProjekt 89, 294, 2674
\bMakroFaulenzer ... 3419
\bMasterExkurs 2572
\bMasterFaelle . 2511, 2579
\bMasterFallRechnung 2557
\bMasterVariablen 2488, 2574
\bMasterVariablenDeklaration 2534
\bMasterWolframLink 2582
\bMenge .. 448, 449, 451, 490, 491, 492, 496, 548, 549, 550, 554, 1745, 1794, 1833, 1834
\bMengeOhneMathe 1745, 1748, 1750
\bMetaBschlangaulSammlung . 900, 2180, 2605, 2683
\bMetaCCLink ... 2620, 2707
\bMetaEmail 2186, 2632, 2640, 2642
\bMetaEmailLink 2636, 2655
\bMetaHermineFriends . . 908, 2185, 2609, 2687
\bMetaHilfMit 98, 2645, 2657
\bMetaQuelltext . 100, 2657
\bMetaSetze 40, 241, 354, 408, 420
\bMetaUeberDasProjekt 2613, 2694
\bMinimierungErklaerung 2776
\bMinispracheDatei . 3272
\bNichtsZuTun 3460
\bO 2451, 2479
\bOmega 2441, 2485
\bOmegaOhneMathe 2441, 2446, 2448
\bONotationO 2884
\bOOhneMathe 2451, 2456, 2458
\bool 715, 808
\bowtie 3040, 3043, 3044, 3045
\Box 536
\boxtimes 1007
\bParagraphMitLinien 1063, 1148, 1166, 2778, 3393, 3461, 3550, 3558, 3566
\bPetriErreichKnotenDrei 2958
\bPetriErreichTransition 2955
\bPetriSetzeSchluessel 2900
\bPetriTransitionsName 2947, 2959
\bPetriTransitionsNameOhneMathe 2947, 2950, 2952
\bPetriTransPfeile . 2959
\bPolynomiellReduzierbar 2034
\bPotenzmenge 1754, 1758, 2970
\bPotenzmengeOhneMathe 1755, 1756, 1757
\bPrimaer 3052
\bProblemBeschreibung 2018
\bProblemClique 2041
\bProblemName 2017, 2024, 2036, 2038, 2051, 2062, 2063, 2071, 2072
\bProblemSat 2070
\bProblemSubsetSum 2061, 2070
\bProblemVertexCover 2041, 2049
\bProduktionen . 1793, 1835
bProduktionsRegeln (environment) 1783
bProjektSprache (environment) 2267
\bPruefungsNummer ... 135
\bPruefungsTitel 138
\bPseudoUeberschrift 2256, 2324, 2327, 2764, 2774, 3547, 3555, 3563
\bPumpingKontextfrei 3016
\bPumpingRegulaer .. 2995
bQuellen (environment) 2364
\bRechtsReduktionInline 2838
\bRekursionsGleichung 2475, 2538
\bRelation 2868
\bRelationMenge 3068
\bRichtig 1007
\bRundeKlammer 2428, 2432, 2442, 2452, 2466, 2881, 2885
\bSetzeTeilaufgabeNr . 219
\bSetzeThemaNr 215
\bSortierMarkierung 3093
\bSortierPfeil 3087
\bSortierPfeilUnten 3090
\bSpaltenUmbruch ... 3162
\bSqlCode 3284
\bStrich 2016
\bSyntheseErklaerung 3331, 3399
\bSyntheseUeberErklaerung 3396
\bSyntheseUeberschrift 3314, 3398
\bT .. 2461, 2476, 2490, 2554
\bTeilen 2801
\bTheta 2431, 2482, 2514, 2520, 2525
\bThetaOhneMathe 2431, 2436, 2438
\bTitelSeite 166, 180
\bTOhneMathe 2461, 2470, 2472
\bTrennSeite 149
\bTuringKante 580
\bTuringLeerzeichen 536, 544
\bTuringMaschine 537
\bTuringUeberfuehrung 583
\bTuringUebergaenge 575, 581
\bTuringUebergangZelle 570
\bUeberfuehrungsFunktion 1759

\bUeberfuehrungsFunktionOhneMathe 1759, 1762, 1764
\bUeberschriftDreiecksTabelle 2772
\bUmlLeserichtung .. 3505
\bVertauschen 3081
\bWortInSprache 1189
\bWortNichtInSprache 1194
\bWpEquivalent 3624
\bWpErklaerung 3632
\bWpErklaerungVerzweigung 3648
\bWpKalkuel 3601
\bWpKalkuelOhneMathe 3601, 3607, 3609, 3650, 3652, 3654
\bWpPseudoMatheUmgebung 3620
\bZustandsBuchstabe 1769, 1778, 1780, 1798, 1800
\bZustandsBuchstabeGross 1770, 1779, 1781
\bZustandsmenge 1757
\bZustandsmengeNr 1771, 2979
\bZustandsmengeNrGross 1779
\bZustandsMengenSammlung 2966
\bZustandsMengenSammlungNr 2975
\bZustandsmengeOhneMathe 1757
\bZustandsname 1780
\bZustandsnameGross 1781, 2967, 2976
\bZustandsnameTiefgestellt 1796
\bZustandsPaar 2756
\bZustandsPaarVariablenName 2755, 2758, 2759

C

\c 1967, 1968
\cdot 2464, 2520, 2531
\centerline ... 2024, 2702
\cftbeforesecskip ... 275
\cftbeforesubsecskip 276, 277
\cftbeforesubsubsecskip 278
\cftsubsecafterpnum . 277
\chapter 1858, 1859
\char 2255
\clearpage 150, 164, 167, 175, 210, 288, 300, 344, 917, 2243
\cline 1188
\clist 608, 655, 656, 679, 683, 3127
\columnbreak 3162
\contentsname 279
\csname 1952, 1955
\cup 1768, 2846, 3350, 3364, 3371

D

\DeclareMathSymbol 2597, 2598
\definecolor 1856
\DefineVerbatimEnvironment 3168
\delta 442, 484, 542, 584, 1759
\dh 2421, 3352
\directlua 435, 514, 572, 577, 1740, 1754, 1774, 1782, 1789, 1794, 2797, 2802, 2855, 2862, 2869, 3082, 3181, 3233, 3238, 3239, 3244, 3249, 3250, 3255, 3261, 3262
\dots 1048, 1052, 2065, 3008, 3363, 3364
\DOWNarrow 3511
\draw 1972, 1975, 1978, 2956, 3088, 3091

E

\edef 1963
\edge 652
\egroup 2310, 2335
\else .. 1112, 1120, 1128, 1136, 1143, 1749, 1763, 1799, 2272, 2295, 2309, 2316, 2334, 2360, 2437, 2447, 2457, 2471, 2808, 2845, 2890, 2951, 3145, 3147, 3608
\emph .. 1209, 1456, 1711, 2046, 2075, 2077, 2250
\empty 2295, 2844
\emptyset 2754, 3358, 3379, 3460
\end 102, 161, 172, 879, 887, 911, 1000, 1253, 1279, 1304, 1339, 1373, 1405, 1433, 1449, 1468, 1476, 1507, 1536, 1557, 1582, 1607, 1627, 1646, 1671, 1689, 1790, 1875, 1981, 2031, 2032, 2039, 2148, 2246, 2307, 2331, 2358, 2380, 2381, 2491, 2507, 2532, 2555, 2570, 2580, 2667, 2671, 2672, 2695, 2708, 2769, 2770, 2818, 2863, 3011, 3031, 3195, 3228, 3417, 3481
\endcsname 1952, 1955
\endgroup . 2245, 2819, 3120
\enspace 3465, 3467
environments:
bAdditum 2313
bAntwort 2269
bBaum 990
bExkurs 2338
bGraphenFormat .. 1944
bJavaAngabe 3203
bKontrollflussgraph 2142
bProduktionsRegeln 1783
bProjektSprache . 2267
bQuellen 2364
liAHuelle 2813
liEinbettung 2268
liKasten 1872
liRelationenSchemaFormat 3074
liRmodell 3054
liUebergangsTabelle 2762
\equiv 3627, 3651
\erzeuge@tiefgestellt 1754, 1755, 1759
\everypar 1998
\EXKURSfalse 602
\EXKURStrue 602
\expandafter ... 1952, 3118
\ExplSyntaxOff 105, 124, 192, 302, 357, 426, 469, 511, 516, 569, 574, 579, 1070, 1092, 1107, 1183, 1735, 1813, 1848, 1915, 2008, 2163, 2202, 2422, 2487, 2711, 2790, 2984, 3067, 3128, 3288, 3401, 3451, 3484, 3571, 3656
\ExplSyntaxOn 17, 116, 134, 214, 322, 407, 438, 479, 512, 537, 570, 575, 599, 1013, 1076, 1093, 1132, 1708, 1803, 1825, 1899, 1987, 2156, 2168, 2239, 2426, 2603, 2772, 2877, 2965, 3055, 3125, 3180, 3313, 3410, 3457, 3537, 3600

F

\faCheckSquareO 3459
\faCircleThin 1727
\faGg 1721
\fancyfoot 1879, 1880, 1881, 2184, 2185, 2186, 2187
\fancyhead . 1878, 2179, 2180, 2181, 2182, 2194
\faSquareO 1715
\fi 1114, 1122, 1130, 1138, 1145, 1751, 1765, 1801, 2275, 2298, 2311, 2319, 2336, 2361, 2439, 2449, 2459, 2473, 2810, 2847, 2892, 2953, 3145, 3146, 3149, 3151, 3610
\filcenter 209
\footcite 1211, 1280, 1302, 1349, 1372, 1404, 1457, 1508, 1535, 1581, 2044, 2047, 2054, 2059, 2064, 2068, 2074, 2079, 2207, 2508, 2509, 2777, 3014
\footnote 2387, 2399
\footnotesize 70, 524, 817, 956, 1062, 1732, 2353, 2374, 2693, 2721, 2815, 2832, 2840, 2969, 2978, 3054, 3169, 3201, 3392, 3497, 3524
\footrulewidth . 1884, 2190
\foreach .. 1967, 1970, 1977
\forestFirst ... 3142, 3145
\forestLast ... 3143, 3145
\forestOget ... 3142, 3143
\forestOnes 3155
\forestOv . 3144, 3145, 3148
\forestov .. 3134, 3138, 3139, 3142, 3143, 3144, 3145, 3147, 3148
\forestset 3130, 3153
\forestSortLevel 3132, 3140, 3154, 3155
\frac . 2466, 2499, 2531, 2546
\fullouterjoin 3045

G

\g 19, 21, 32, 34, 42, 56, 58, 136, 139, 155, 157, 185, 189, 216, 220, 224, 228, 229, 230, 231, 232, 234, 235, 237, 251, 254, 257, 262, 411, 655, 656, 679, 683, 689, 690, 691, 692, 693, 694, 696, 697, 698, 700, 701, 703, 704, 705, 706, 707, 708, 709, 710, 713, 717, 718, 719, 722, 724, 725, 726, 727, 728, 729, 773, 777, 781, 784, 785, 787, 788, 790, 791, 795, 797, 810, 811, 812, 813, 821, 822, 824, 830, 831, 833, 834, 836, 837, 849, 852, 857, 859, 863
\Gamma .. 483, 541, 584, 1768
\gappto 3150
\geometry 324, 375
\geq 2072, 2497, 2997, 3002, 3018, 3022
\gib 74, 355, 794

H

\hangindent 1997
\hbox 3040
\headrulewidth . 1883, 2189
\headwidth 2200
\hfill 64, 2516, 2522, 2527, 3468
\hline 2767
\horizontale 25, 72
\href 1821, 2400, 2584, 2623, 2638, 3227
\hsize 1996
\hspace ... 2959, 3616, 3628
\ht 3041
\Huge 159, 185, 249, 334
\huge 209

I

\i 1977, 1978
\ifADDITUM . 600, 2315, 2333
\ifANTWORT . 604, 2271, 2308
\ifEXKURS 602, 2340
\ifmmode 1110, 1118, 1126, 1134, 1141, 1747, 1761, 1797, 2435, 2445, 2455, 2469, 2806, 2888, 2949, 3606
\IfNoValueTF 2322, 2389, 2401
\ifnum 3117, 3133, 3141, 3147
\ifx 2295, 2844, 3145
\in 1036, 1192, 2072, 2479, 2482, 2485, 2514, 2520, 2525, 2997, 3008, 3018, 3028, 3341, 3343, 3349, 3370, 3559
\includegraphics 1906, 1911
\includepdf 177
\indexbox 1993
\inhalts 272, 290
\inhaltsverzeichnis 2240
\input 225, 389, 392, 395, 398, 401, 926
\inputminted 3232, 3243, 3254, 3270, 3274, 3282
\insert 1992
\int 3127
\item .. 1007, 1008, 1285, 1289, 1294, 1299, 1343, 1352, 1357, 1365, 1437, 1442, 1446, 1472, 1512, 1517, 1524, 1532, 1561, 1566, 1570, 1575, 1675, 1680, 1685, 2029, 2030, 2366, 2371, 2494, 2499, 2503, 2513, 2519, 2524, 2536, 2540, 2544, 2548, 2552, 2559, 2563, 2567, 3002, 3005, 3008, 3022, 3025, 3028
\itshape 1061, 3391

J

\j 1967, 1968, 1970, 1971, 1972, 1977, 1978, 1979

K

\k 1977
\keine .. 78, 919, 2265, 2383
\keys . 117, 413, 447, 459, 489, 499, 547, 557, 687, 1080, 1084, 1098, 1103, 1832, 1839
\kopfzeile 2170, 2180, 2181, 2182, 2185, 2186, 2187, 2195

L

\l ... 440, 441, 442, 443, 444, 445, 448, 449, 450, 451, 452, 454, 456, 461, 462, 463, 464, 465, 466, 481, 482, 483, 484, 485, 486, 487, 490, 491, 492, 493, 494, 495, 496, 502, 503, 504, 505, 506, 507, 508, 539, 540, 541, 542, 543, 544, 545, 548, 549, 550, 551, 552, 553, 554, 560, 561, 562, 563, 564, 565, 566, 1078, 1081, 1086, 1087, 1090, 1095, 1096, 1099, 1100, 1105, 1827, 1828, 1829, 1830, 1833, 1834, 1835, 1836, 1842, 1843, 1844, 1845, 2159, 2160, 2161, 2370, 2371, 2372, 2379, 3437, 3440, 3441, 3443
\labelenumi 1869
\labelenumii 1870

\labelitemi 1863
\labelitemii 1864
\labelitemiii 1865
\labelitemiv 1866
\land 3652, 3654
\LARGE 155, 189, 1858
\Large 55, 181, 876, 899
\large 46, 157, 884, 907, 2024, 3104
\leaders 3468
\leavevmode 2001
\left 2429, 2882
\LEFTarrow 3512
\leftarrow 1124
\leftouterjoin 3043
\leftskip . 3635, 3636, 3646
\LehramtInformatikGitBranch 848, 3188
\LehramtInformatikGithubCodeRepo 3187
\LehramtInformatikGithubDomain 845, 3184
\LehramtInformatikGithubRawDomain 3185
\LehramtInformatikGithubTexRepo 846, 3186
\LehramtInformatikRepository 177, 226, 389, 392, 395, 398, 401, 1903, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 3183
\leq 2531, 3005, 3025
\let 1757, 1758, 1998, 2243, 3116, 3154, 3155, 3419, 3434, 3446, 3459
\li@chomsky@erklaerung@texte 1027, 1063
\li@EntwurfsCode 1205, 1256, 1257, 1258, 1307, 1308, 1309, 1310, 1376, 1377, 1378, 1379, 1380, 1381, 1408, 1409, 1410, 1411, 1412, 1413, 1479, 1649, 1650, 1651, 1652, 1692, 1693
\li@EntwurfsCodeAllgemein 1204
\li@fussnote@text 2718, 2724, 2728, 2732, 2736
\li@GithubLink 3222, 3237, 3248, 3260
\li@mget .. 1954, 1958, 1978
\li@minc 1957, 1979
\li@mset 1951, 1959, 1968, 1971
\li@numdiscs 1963, 1972, 1978
\li@Rmodell@Schrift 3054, 3063, 3073
\li@SansFett .. 141, 155, 157, 159, 181, 185, 189
\li@sequence ... 1964, 1977
\li@synthese@erklaerung@texte 3331, 3393
liAHuelle (environment) 2813
liEinbettung (environment) 2268
liKasten (environment) 1872
\lineskip 1999
\linespread 3061
liRelationenSchemaFormat (environment) . 3074
liRmodell (environment) 3054
\listen 2364, 2379
\literatur 2206, 2230
liUebergangsTabelle (environment) . 2762
\llap 3041
\LoadClass 4, 110, 129, 130, 197, 308, 362
\log 2479, 2482, 2485, 2514, 2520
\logo 2662, 2676, 2700
\lor 3653
\ltimes 2590

M

\makeatletter 3113
\makeatother 3121
\makeindex 2007
\marginpar 1714, 1720, 1726, 2251
\marginparsep 1997
\marginparwidth 1996
\mathbb ... 2072, 3028, 3559
\mathbin .. 3043, 3044, 3045
\mathcal 2452, 2885, 3370, 3375, 3377, 3378, 3379
\mathe 3612, 3622, 3626
\mathord 2597, 2598
\mdfsetup 2280, 2284, 2288, 2292
\mdseries 2300
\medskip 48, 67, 904, 999, 2026, 2263, 2377, 2690, 3059, 3065, 3470, 3475, 3614, 3618
\mintinline 3211, 3220, 3266, 3278, 3286
\mkern 3043, 3044, 3045
\mlq 2595, 2597
\mrq 2595, 2598
\msg 924
\myList 3134, 3135, 3136, 3139
\myNodes 3123, 3138, 3144, 3148, 3150

N

\NeedsTeXFormat 1, 107, 126, 194, 304, 359, 371, 384, 404, 429, 597, 930, 1003, 1010, 1072, 1185, 1200, 1701, 1737, 1850, 1887, 1896, 1917, 1946, 1984, 2010, 2082, 2165, 2204, 2210, 2232, 2424, 2587, 2600, 2713, 2792, 2875, 2895, 2961, 2986, 2991, 3034, 3047, 3076, 3157, 3164, 3171, 3176, 3290, 3304, 3403, 3407, 3453, 3486, 3528, 3573, 3596
\neg 3654
\negthinspace .. 2429, 2882
\NewDocumentCommand 439, 480, 517, 538, 580, 606, 1077, 1094, 1133, 1140, 1189, 1194, 1804, 1826, 1905, 1910, 2034, 2157, 2208, 2385, 2397, 2857, 2868, 2955, 3230, 3241, 3252, 3268, 3272, 3280, 3505
\NewDocumentEnvironment 990, 1783, 1872, 1944, 2142, 2267, 2268, 2269, 2313, 2338, 2368, 2763, 2813, 3056, 3074
\newif 600, 602, 604
\newlength 3632
\newminted 3203
\node 1090, 2150, 2155, 3093, 3138, 3523
\noexpand 3138
\noindent 27, 45, 53, 69, 97, 819, 994, 1191, 1196, 2260, 2296, 2354, 2356, 2375, 2410, 2576, 2742, 2745, 2748, 2751, 3069, 3432, 3445, 3463, 3476, 3639
\nolinkurl 3227
\normalbaselineskip 2000, 2002, 2004
\normallineskip 1999
\normalsize ... 1860, 3212
\notin 1197
\null 3468

O

\o 2884, 2889, 2891
\o@join 3040, 3043, 3044, 3045

\Omega 2442
\omega 2997, 2998, 3018, 3019

P

\pagestyle . . 346, 869, 1882
\par 22, 26, 29, 48, 65, 252, 255, 258, 263, 268, 818, 839, 878, 886, 893, 902, 910, 920, 1067, 1815, 1823, 1998, 2259, 2355, 2409, 2685, 2688, 3071, 3398, 3431, 3435, 3447, 3469, 3474, 3482, 3638, 3644
\paragraph 1860
\parfillskip 1998
\parindent 1997, 3062
\path . . . 471, 518, 581, 1105
\pgfkeys . . . 2911, 3509, 3510, 3511, 3512, 3513, 3516, 3519, 3521
\pgfmath@count 3115, 3117, 3119
\pgfmath@smuggleone 3120
\pgfmathdeclarefunction 3114
\pgfmathint 3115
\pgfmathparse 1958, 3132, 3137, 3140, 3154, 3155
\pgfmathresult 1959, 3115, 3116, 3118, 3120, 3133, 3141, 3154, 3155
\pgfutil@empty 3116
\pgfutil@loop 3117
\pgfutil@repeat 3120
\preceq 2037
\prime 2016
\printbibliography . 2230
\printindex 350
\ProcessKeysOptions . 123
\prop . 3437, 3440, 3441, 3443
\ProvidesClass 2, 108, 127, 195, 305, 360
\ProvidesPackage . 372, 385, 405, 430, 598, 931, 1004, 1011, 1073, 1186, 1201, 1702, 1738, 1851, 1888, 1897, 1918, 1947, 1985, 2011, 2083, 2166, 2205, 2211, 2233, 2425, 2588, 2601, 2714, 2793, 2876, 2896, 2962, 2987, 2992, 3035, 3048, 3077, 3158, 3165, 3172, 3177, 3291, 3305, 3404, 3408, 3454, 3487, 3529, 3574, 3597

Q

\quad 2720

R

\raggedleft 2303
\raggedright 2001
\raisebox 2155
\relax 2002, 2003, 2243, 3117, 3119
\renewcommand 277, 279, 1863, 1864, 1865, 1866, 1869, 1870, 1883, 1884, 2189, 2190, 2762
\repariere . . 83, 330, 2198
\RequirePackage 15, 111, 112, 131, 133, 208, 213, 283, 321, 369, 374, 387, 432, 535, 608, 927, 933, 934, 1006, 1074, 1203, 1204, 1704, 1706, 1707, 1743, 1852, 1853, 1855, 1857, 1862, 1871, 1877, 1885, 1889, 1890, 1900, 1901, 1919, 1920, 1921, 1949, 1950, 1988, 1989, 2015, 2084, 2169, 2207, 2212, 2213, 2229, 2236, 2237, 2238, 2427, 2591, 2592, 2878, 2879, 2880, 2897, 2899, 2989, 3036, 3037, 3038, 3051, 3079, 3110, 3161, 3167, 3174, 3179, 3190, 3191, 3293, 3405, 3411, 3458, 3490, 3491, 3492, 3493, 3576, 3599
\right 2429, 2882
\RIGHTarrow . . . 3508, 3513
\Rightarrow . . . 1192, 1197
\rightarrow 584, 1031, 1036, 1044, 1048, 1050, 1051, 1053, 1108, 1116, 2959, 3341, 3348, 3350, 3353, 3358, 3363, 3364, 3369
\rightouterjoin 3044
\rmfamily 145, 1018, 1978, 3318
\Roman 1868
\roman 1868, 1870
\rtimes 2590
\rule 28, 2411, 3041

S

\sb 444, 454, 456, 485, 543, 1043, 1044, 1048, 1051, 1052, 1053, 1135, 1137, 1142, 1144, 1798, 1800, 2479, 2482, 2485, 2514, 2520, 2782, 2947, 2956, 3363, 3364, 3365, 3370, 3374, 3375, 3378, 3379, 3382, 3383, 3384
\scriptscriptstyle 1108, 1116, 1124
\scriptsize 1817, 2095, 2102, 2108, 2173, 2300, 2706, 3544, 3603, 3641
\section 209, 217, 424
\sectionbreak 210
\seq . . . 2159, 2160, 2161, 2370, 2371, 2372, 2379
\setbox . . . 2273, 2317, 3040
\setcounter . 211, 282, 1861
\setganttlinklabel . . 1891, 1892, 1893, 1894
\setlength 275, 276, 278, 2200, 3062, 3635, 3636, 3646
\setmainfont 1854
\setmainlanguage 928
\setminted 3196, 3197
\setminus 2842
\setul 3053
\setze 74, 80, 323, 2177, 2192
\shoveleft 2822
\shoveright 2826
\Sigma 441, 482, 540, 1767, 1768, 1828
\sigma 1041, 1043, 1044
\SLASH 2255
\sloppy 2000
\small 993
\sort 3127
\sortList 3126, 3135
\square 1008
\stichwoerter 18, 70
\str 136, 139, 155, 157, 185, 189, 740, 1019, 1028, 2277, 2971, 2980, 3319, 3333
\string 2824, 2834
\StrSubstitute . 3134, 3136
\strut 151, 163, 168, 174, 871, 915, 2536, 2540, 2544, 2548, 2552, 3162
\subsection 221, 355
\subseteq . 3344, 3377, 3384
\subsubsection . . 242, 3442

T

\tableofcontents 280, 343, 2244
\TeX 2659
\text 454, 456, 559, 2804, 3544, 3603

\textbackslash 3423, 3424, 3426
\textbf 20, 1155, 1173, 1709, 2042, 2051, 2062, 2071, 2261, 2297, 2355, 2376, 2767, 3433
\textcolor 2154, 3210, 3540
\textit 33, 995, 1549, 1591, 1592, 1593, 1594, 2812, 2872
\textsc 2017
\textsf 2376
\textstyle 2499, 2531
\texttt 1732, 2017, 2151, 2152, 2153, 2154, 3421, 3603
\textwidth .. 28, 2200, 2413
\thematik 31, 64
\thepage 1881, 2187
\theparagraph 1860
\thesection 209
\Theta 2432
\thinspace 3603
\thispagestyle 179
\tikz 2155
tikz: b binaer baum 33
tikz: bbaum 35
\tikzchildnode 951
\tikzparentnode 951
\tikzset 473, 520, 586, 936, 962, 1923, 2086, 2937, 3100, 3294, 3578
\tikzumlset 3495
\times 584
\tiny 96, 1715, 1721, 1727, 1995, 2154, 2252, 3225
\titel 244, 246, 286, 331, 867
\titleformat 209, 1858, 1860
\titlespacing 1859
\tl 19, 32, 42, 56, 216, 220, 224, 231, 234, 411, 440, 441, 442, 443, 444, 445, 448, 449, 450, 451, 452, 454, 456, 481, 482, 483, 484, 485, 486, 487, 490, 491, 492, 493, 494, 495, 496, 539, 540, 541, 542, 543, 544, 545, 548, 549, 550, 551, 552, 553, 554, 680, 684, 713, 717, 718, 719, 722, 727, 728, 729, 741, 750, 784, 787, 790, 795, 810, 811, 812, 813, 824, 830, 833, 836, 852, 859, 1078, 1081, 1086, 1087, 1095, 1096, 1099, 1100, 1806, 1827, 1828, 1829, 1830, 1833, 1834, 1835, 1836, 2462
\tmp 2844
\TmpPlaceEight 2919
\TmpPlaceFive 2916
\TmpPlaceFour 2915
\TmpPlaceNine 2920
\TmpPlaceOne 2912
\TmpPlaceSeven 2918
\TmpPlaceSix 2917
\TmpPlaceTen 2921
\TmpPlaceThree 2914
\TmpPlaceTwo 2913
\TmpScale 2932
\TmpTransitionEight 2908, 2929
\TmpTransitionFive 2905, 2926
\TmpTransitionFour 2904, 2925
\TmpTransitionNine 2909, 2930
\TmpTransitionOne 2901, 2922
\TmpTransitionSeven 2907, 2928
\TmpTransitionSix 2906, 2927
\TmpTransitionTen 2910, 2931
\TmpTransitionThree 2903, 2924
\TmpTransitionTwo 2902, 2923
\TmpX 2933
\TmpY 2934
\ttfamily 3054

U

\ul 1710, 3052, 3053
\umlaggreg 1668
\umlassoc 1626
\umlclass 1215, 1219, 1223, 1270, 1271, 1272, 1319, 1324, 1329, 1332, 1390, 1391, 1392, 1397, 1398, 1427, 1462, 1493, 1494, 1497, 1548, 1551, 1590, 1596, 1597, 1615, 1616, 1617, 1638, 1639, 1640, 1660, 1661, 1662, 1663
\umldep 1556
\umlHVHaggreg 1337, 1403, 1606
\umlinherit 1276, 1327, 1501, 1546, 1554
\umlnote .. 1278, 1503, 1670
\umlreal 1274, 1335
\umlsimpleclass 1230, 1231, 1232, 1236, 1238, 1239, 1240, 1269, 1422, 1423, 1424, 1492, 1544, 1545, 1636
\umlstatic 1429, 1463
\umluniaggreg 1499
\umluniassoc 1250, 1275, 1500, 1624, 1625, 1644, 1645
\umlVHuniassoc . 1251, 1252
\umlVHVdep . 1244, 1245, 1247, 1248, 1431, 1432
\umlVHVinherit .. 1227, 1228, 1233, 1234, 1241, 1242, 1400, 1401, 1425, 1426, 1604, 1605, 1642, 1643
\umlVHVreal 1394, 1395, 1665, 1666
\UParrow 3510
\url 853, 2388
\usemintedstyle 3193
\usetikzlibrary .. 433, 935, 1705, 1922, 2085, 2898, 3039, 3080, 3111, 3494, 3577

V

\varepsilon 1020, 1031, 1032, 1753, 2479, 2485, 2517, 2528
\vbox 2273, 2317
\vfill . 87, 152, 162, 169, 173, 292, 881, 889, 3162
\vrule 2002, 2004, 3464, 3468
\vspace 156, 158, 183, 187, 265, 277, 872, 895, 913, 2341, 2359, 3478, 3480

X

\xappto ... 3138, 3144, 3148
\xdef 1952

Z

\z@ 1997, 2002, 2004
\ZB 2420
\zB 2419
\zustandsnamens@liste 1771, 1778, 1779