

# LaTeX

## Vortrag im Rahmen der Projektgruppe

Janet Fiedler   Sven Schrinner  
PG583: Algorithmen zur Entdeckung  
krebsauslösender Genvarianten

Fakultät für Informatik, Technische Universität Dortmund

March 28, 2014

- 1 Einleitung
- 2 Aufbau Installation
- 3 Dokumentaufbau
- 4 Strukturierung
- 5 Literaturverzeichnis
- 6 Mehrere Dateien
- 7 Mathematische Formeln
- 8 BibT<sub>E</sub>X
- 9 Abbildungen, Skizzen, Tabellen, Algorithmen
- 10 Tipps für wissenschaftliches Schreiben

- Tex: Textsatzsystem in einer Programmiersprache
- 1986 fertiggestellt von Donald E. Knuth für sein Buch The Art of Computer Programming



Figure : Donald E. Knuth

- Problem: schwierig zu benutzen



- Worin unterscheidet sich LaTeX gegenüber anderen Editoren?
  - Trennung von Inhalt und Layout
  - Portabilität und Sicherheit
  - Umfangreiche Formelsätze
- Wofür eignet sich LaTeX?
  - Wissenschaftliche Arbeiten
  - Bücher
  - Präsentationen

## ■ Wie funktioniert LaTeX?

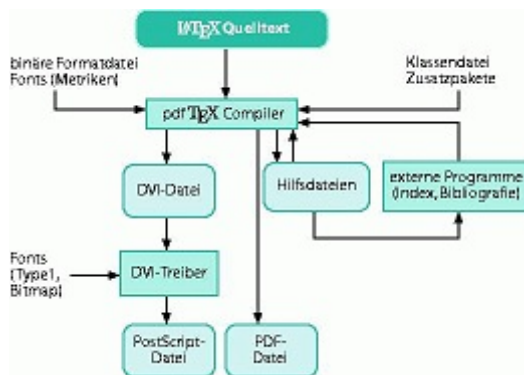


Figure : TeX System

- Welche Software wird für LaTeX benötigt?
  - LaTeX-Distribution
    - Kern des Systems
    - Generiert aus einem Eingabetext eine Ausgabe
    - Enthält Zusatzpakete
    - Für Windows: MikTeX, XemTex, proTeXt und weitere
    - Für Mac: MacTeX, XeTeX, OzTeX und weitere
    - Für Linux: TeXLive, teTeX, VTeX und weitere

- Welche Software wird für LaTeX benötigt?
  - Editor
    - Eingabe des Quelltextes
    - Für Windows: TeXnic Center
    - Für Linux: Kile, gedit
    - Für Mac: TeXShop, Textmate
    - Für Eclipse: TeXlipse
    - Plattformunabhängig: TeXworks, TeXmaker



- Wie ist ein LaTeX-Dokument aufgebaut?
  - 1 Angabe zur Dokumentenklasse
    - Typografische Vorgaben
  - 2 Präambel
    - Wesentliche Eigenschaften des Dokumentes
  - 3 Textkörper

## ■ Dokumentenklassen

article	für Artikel in wissenschaftlichen Zeitschriften (Journals), Präsentationen, Kurzberichte, Programmdokumentation, Einladungen
scrartcl	article im Koma Script
proc	eine Klasse für Berichte, welche auf der article Klasse basieren
minimal	ist so klein wie es nur geht; setzt nur die Größe einer Seite und eine Basisschrift, wird hauptsächlich zum Fehler suchen und bereinigen verwendet
report	für längere Berichte, welche mehrere Kapitel umfassen, kurze Bücher, Diplomarbeiten, ...

## ■ Dokumentenklassen

scrreprt	report im Koma-Script
book	für Bücher
scrbook	book im Koma-Script
letter	zum Schreiben von Briefen
beamer	zum Erstellen von Präsentationen

## ■ Präambel

- Laden von zusätzlichen Pakten
- Erstellung von Tabellen
- Einbindung von Bildern
- Darstellung von mathematischen Formeln

- Besonderheiten zu den Standardklassen
  - richten sich nach US-amerikanischen typografischen Konventionen
  - keine Umlaute und regionale Sonderzeichen
  - für deutsche Dokumente: Koma Script Klassen benutzen
  - erfordert vier zusätzliche Pakete in der Präambel

## ■ Grundlegende Befehle

<code>\</code>	Deklaration eines Befehles
<code>{}</code>	Argumente
<code>[]</code>	Optionale Argumente
<code>%</code>	Kommentar
<code>\begin</code>	Definition einer Umgebung
<code>\end</code>	

- Erstes Beispiel

## ■ Titelseite

- `\author` gibt den Autor an
- `\title` setzt den Titel
- `\date` gibt das Datum an
- `\subject` setzt die Typisierung
- `\publishers` gibt die Herausgeber an
- `\subtitle` setzt den Untertitel
- `\and` trennt Personen im Dokument
- `\maketitle` Befehl für das setzen des Titels im Dokument



- Zweites Beispiel

- Schriftart

## Schriftfamilie

<i>Textmakro</i>	<i>als Schalter</i>	<i>Beispiel</i>
<code>\textrm{Roman}</code>	<code>\rmfamily</code>	Roman
<code>\textsf{Sans Serif}</code>	<code>\sffamily</code>	Sans Serif
<code>\texttt{Typewriter}</code>	<code>\ttfamily</code>	Typewriter

## Schriftserie

<i>Textmakro</i>	<i>als Schalter</i>	<i>Beispiel</i>
<code>\textmd{medium}</code>	<code>\mdseries</code>	medium
<code>\textbf{bold}</code>	<code>\bfseries</code>	<b>bold</b>

## Schriftform

<i>Textmakro</i>	<i>als Schalter</i>	<i>Beispiel</i>
<code>\textup{upright}</code>	<code>\upshape</code>	upright
<code>\textit{italic}</code>	<code>\itshape</code>	<i>italic</i>
<code>\textsl{slanted}</code>	<code>\slshape</code>	<i>slanted</i>
<code>\textsc{Small Capitals}</code>	<code>\scshape</code>	SMALL CAPITALS

- Schriftgröße

$\text{\LaTeX}$	Beispiel
<code>\tiny</code>	normal Beispiel normal
<code>\scriptsize</code>	normal Beispiel normal
<code>\footnotesize</code>	normal Beispiel normal
<code>\small</code>	normal Beispiel normal
<code>\normalsize</code>	normal Beispiel normal
<code>\large</code>	normal Beispiel normal
<code>\Large</code>	normal Beispiel normal
<code>\huge</code>	normal <b>Beispiel</b> normal
<code>\Huge</code>	normal <b>Beispiel</b> normal

```
\begin{itemize}  
\item Eintrag 1  
\item Eintrag 2  
\end{itemize}
```

```
\begin{enumerate}  
\item Nummer 1  
\item Nummer 2  
\end{enumerate}
```

```
\begin{description}  
\item[Punkt] ist ein Punkt  
\item[Liste] hat viele Punkte  
\end{description}
```

- Drittes Beispiel

```
\section{Kapitel 1}  
\subsection{Unterkapitel 1}  
\frame{Folie 1 des Unterkapitels 1}  
\frame{Folie 2 des Unterkapitels 1}  
  
\section{Kapitel 2}  
\subsection{Unterkapitel 1}  
\frame{Folie 1 des Unterkapitels 1}  
\subsection{Unterkapitel 2}  
\frame{Folie 1 des Unterkapitels 2}
```



- Schachtelung beliebig
- Inhaltsverzeichnis lässt sich mit `\tableofcontents` erstellen
- Gliederungsebene lässt sich mit `\setcounter{tocdepth}{Zahl}` festlegen
  - Schachtelung beliebig
  - -1 für `\part`
  - 0 für `\chapter`
  - 1 für `\section`
  - 2 für `\subsection`
  - 3 für `\subsubsection`
  - 4 für `\paragraph`
  - 5 für `\subparagraph`

## ■ Viertes Beispiel

Was ist BibT<sub>E</sub>X?

- Eigenständiges Programm neben L<sup>A</sup>T<sub>E</sub>X
- Erzeugung von Literaturverzeichnissen

Was ist BibT<sub>E</sub>X?

- Eigenständiges Programm neben L<sup>A</sup>T<sub>E</sub>X
- Erzeugung von Literaturverzeichnissen

Warum BibT<sub>E</sub>X verwenden?

- In L<sup>A</sup>T<sub>E</sub>X-Distributionen standardmäßig enthalten
- Trennung zwischen Inhalt und Form
- Modifikationen leicht durchzuführen durch Stilvorlagen
- Fast alle Onlinequellen stellen passende BibT<sub>E</sub>X-Einträge zur Verfügung

- Literaturdatenbanken sind .bib-Dateien

- Literaturdatenbanken sind .bib-Dateien
- .bib-Dateien bestehen aus Einträgen

- Literaturdatenbanken sind .bib-Dateien
- .bib-Dateien bestehen aus Einträgen

```
@<typ>{<bezeichner>,  
    <param1> = {<wert1>}  
    ...}
```

```
@article{Wandelt13,  
  author      = {Sebastian Wandelt and  
                 Johannes Starlinger and  
                 Marc Bux and  
                 Ulf Leser},  
  title       = {RCSI: Scalable similarity search in  
                 thousand(s) of genomes},  
  journal     = {PVLDB},  
  volume      = {6},  
  number      = {13},  
  year        = {2013},  
  pages       = {1534-1545},  
  ee          = {http://www.vldb.org/pvldb/vol6/p1534-  
                 wandelt.pdf},  
  bibsource   = {DBLP, http://dblp.uni-trier.de}  
}
```



Referenzart	notwendige Felder	optionale Felder
article	author, title, journal, year	volume, number, pages, month, note
book	author oder editor, title, publisher, year	volume oder number, series, address, edition, month, note, isbn
booklet	title	author, howpublished, address, month, year, note
conference	author, title, booktitle, year	editor, volume oder number, series, pages, address, month, organization, publisher, note
inbook	author oder editor, title, booktitle, chapter und/oder pages, publisher, year	volume oder number, series, type, address, edition, month, note
incollection	author, title, booktitle, publisher, year	editor, volume oder number, series, type, chapter, pages, address, edition, month, note
inproceedings	author, title, booktitle, year	editor, volume oder number, series, pages, address, month, organization, publisher, note
manual	title	author, organization, address, edition, month, year, note
mastersthesis	author, title, school, year	type, address, month, note
misc	-	author, title, howpublished, month, year, note
phdthesis	author, title, school, year	type, address, month, note
proceedings	title, year	editor, volume oder number, series, address, month, organization, publisher, note
techreport	author, title, institution, year	type, number, address, month, note
unpublished	author, title, note	month, year

Figure : Referenzarten

## ■ Fünftes Beispiel

Einbinden eigentlich ganz einfach über ...

```
\bibliography{dateiname}
```

Einbinden eigentlich ganz einfach über ...

```
\bibliography{dateiname}
```

... aber das Kompilieren wird aufwändiger:

1. `pdflatex doc.tex` - Sucht Literaturverweise
2. `bibtex doc.tex` - Erstellt Literaturdaten
3. `pdflatex doc.tex` - Erstellt Literaturverzeichnis
4. `pdflatex doc.tex` - Löst Literaturverweise auf

(Für wiederholte Einträge reicht einfaches Nachkompilieren aus)

Zitate über den `\cite`-Befehl:

Ein Verfahren zum Read-Mapping mit Varianten wird  
in `\cite{Wandelt13}` vorgestellt.

Zitate über den `\cite`-Befehl:

Ein Verfahren zum Read-Mapping mit Varianten wird  
in `\cite{Wandelt13}` vorgestellt.

Wird zu:

Ein Verfahren zum Read-Mapping mit Varianten wird  
in [1] vorgestellt.

Literaturverzeichnis:

[1] Sebastian Wandelt, Johannes Starlinger, Marc Bux, and Ulf  
Leser.

Rcsi: Scalable similarity search in thousand(s) of genomes.  
*PVLDB*, 6(13):1534–1545, 2013.

Für längere Zitate eignen sich die quote- und quotation-Umgebungen

```
\begin{quotation}Hier steht ein längeres Zitat, das  
aus mehreren Zeilen besteht. In dieser Umgebung wird  
die erste Zeile jedes Absatzes eingerückt.  
\cite{Wandelt13}\end{quotation}
```

Dieser Text steht vor dem Zitat:

*Hier steht ein längeres Zitat, das aus mehreren  
Zeilen besteht. In dieser Umgebung wird die erste Zeile  
jedes Absatzes eingerückt.[1]*

Dieser Text steht nach dem Zitat:

Für einzeilige Zitate besser quote verwenden.

Formatierung des Literaturverzeichnisses mit Befehl:

```
\bibliographystyle{<Stil>}
```



Formatierung des Literaturverzeichnisses mit Befehl:

```
\bibliographystyle{<Stil>}
```

Mitgelieferte Stilvorlagen:

- plain** Sortiert nach Erstautor, Nummern als Referenz
- alpha** Sortiert nach Erstautor, Nummern als Referenz, Vornamen nur als Initialen
- abbrv** Sortiert nach Erstautor, Autor-Jahr-Kombination als Referenz
- unsrt** Sortiert nach Erstreferenz, Nummern als Referenz

...

Weitere Stilvorlagen über das Paket natbib verfügbar:

**plainnat** Sortiert nach Erstautor, Nummern als Referenz

**abbrvnat** Sortiert nach Erstautor, Autor-Jahr-Kombination als Referenz

**unsrtnat** Sortiert nach Erstreferenz, Nummern als Referenz

Weitere Stilvorlagen über das Paket natbib verfügbar:

**plainnat** Sortiert nach Erstautor, Nummern als Referenz

**abbrvnat** Sortiert nach Erstautor, Autor-Jahr-Kombination als Referenz

**unsrtnat** Sortiert nach Erstreferenz, Nummern als Referenz

natbib stellt weitere Funktionalität zur Verfügung:

```
Die Autoren \citet{Wandelt13} haben ...
```

Die Autoren Wandelt et. al. (2013) haben ...

Weitere Stilvorlagen über das Paket natbib verfügbar:

**plainnat** Sortiert nach Erstautor, Nummern als Referenz

**abbrvnat** Sortiert nach Erstautor, Autor-Jahr-Kombination als Referenz

**unsrtnat** Sortiert nach Erstreferenz, Nummern als Referenz

natbib stellt weitere Funktionalität zur Verfügung:

```
Die Autoren \citet{Wandelt13} haben ...
```

Die Autoren Wandelt et. al. (2013) haben ...

```
In \citep[Kapitel 2]{Wandelt13} wurde gezeigt, dass ...
```

In (Wandelt et. al. 2013, Kapitel 2) wurde gezeigt, dass ...

- Ein TeX-Dokument lässt sich in mehrere TeXDokumente aufteilen
- Vorteile dabei:
  - Trennung von Präambel und Textkörper
  - Weniger Mergekonflikte bei Verwendung von Versionsverwaltungen
  - Bessere Übersicht
- Befehle:
  - `\include{Dateipfad}`
  - `\input{Dateipfad}`

- Sechstes Beispiel

Mathematische Formeln werden mit  $\text{\$}$ -Zeichen eingeleitet.

Mathematische Formeln werden mit  $\text{\$}$ -Zeichen eingeleitet.  
Entweder als Inline-Formel ...

Somit ist die Gleichung  $\text{\$}a^2+b^2=c^2\text{\$}$  erfüllt.

Somit ist die Gleichung  $a^2 + b^2 = c^2$  erfüllt.



Mathematische Formeln werden mit  $\$$ -Zeichen eingeleitet.  
Entweder als Inline-Formel ...

Somit ist die Gleichung  $a^2 + b^2 = c^2$  erfüllt.

Somit ist die Gleichung  $a^2 + b^2 = c^2$  erfüllt.

... oder in einer eigenen Zeile mit  $\backslash[$  und  $\backslash]$ :

Es gilt die Formel  $\backslash[E = mc^2\backslash]$  wobei  $c$  die Lichtgeschwindigkeit ist.

Es gilt die Formel

$$E = mc^2$$

wobei  $c$  die Lichtgeschwindigkeit ist.

# Mathematische Formeln - Mathematische Symbole

L<sup>A</sup>T<sub>E</sub>X bietet bereits viele Befehle für mathematische Zeichen und Symbole (siehe z.B. Wikibooks):

Griechische  
Buchstaben:

<code>\alpha</code>	$\alpha$
<code>\beta</code>	$\beta$
<code>\gamma</code>	$\gamma$
<code>\delta</code>	$\delta$
<code>\epsilon</code>	$\epsilon$
...	...

Operatoren:

<code>\rightarrow</code>	$\rightarrow$
<code>\leq</code>	$\leq$
<code>\neq</code>	$\neq$
<code>\subset</code>	$\subset$
<code>\in</code>	$\in$
<code>\not\in</code>	$\notin$
<code>\cap</code>	$\cap$
<code>\cup</code>	$\cup$
<code>\sum</code>	$\sum$

Spezielle Symbole:

<code>\infty</code>	$\infty$
<code>\emptyset</code>	$\emptyset$
<code>\forall</code>	$\forall$
<code>\exists</code>	$\exists$

Brüche mit `\frac{<Zähler>{<Nenner>}`

```
$\frac{x}{y}$
```

$$\frac{x}{y}$$

Brüche mit `\frac{<Zähler>{<Nenner>}`

```
$\frac{x}{y}$
```

$$\frac{x}{y}$$

Super- und Subskripte: `Text_<Subscript>^<Superscript>`

```
$(x_1+x_2)^n = \sum_{i=1}^m y_i$
```

$$(x_1 + x_2)^n = \sum_{i=1}^m y_i$$

- Häufig bei wissenschaftlichem Schreiben benötigt:  
Definitionen, Lemmata, Theoreme, Beweise, ...

- Häufig bei wissenschaftlichem Schreiben benötigt:  
Definitionen, Lemmata, Theoreme, Beweise, ...
- Viele Pakete bieten entsprechende Umgebungen an

- Häufig bei wissenschaftlichem Schreiben benötigt:  
Definitionen, Lemmata, Theoreme, Beweise, ...
- Viele Pakete bieten entsprechende Umgebungen an
- Hier: `ntheorem`, da Unterstützung für deutsche Sprache

- Häufig bei wissenschaftlichem Schreiben benötigt: Definitionen, Lemmata, Theoreme, Beweise, ...
- Viele Pakete bieten entsprechende Umgebungen an
- Hier: `ntheorem`, da Unterstützung für deutsche Sprache

Benötigte Pakete:

```
\usepackage{amsmath}  
\usepackage[amsmath,standard,thmmarks]{ntheorem}
```



- Häufig bei wissenschaftlichem Schreiben benötigt: Definitionen, Lemmata, Theoreme, Beweise, ...
- Viele Pakete bieten entsprechende Umgebungen an
- Hier: `ntheorem`, da Unterstützung für deutsche Sprache

Benötigte Pakete:

```
\usepackage{amsmath}  
\usepackage[amsmath,standard,thmmarks]{ntheorem}
```

Unterstützte Umgebungen:

- |              |               |
|--------------|---------------|
| ■ Definition | ■ Beweis      |
| ■ Lemma      | ■ Beispiel    |
| ■ Satz       | ■ Korollar    |
| ■ Theorem    | ■ Proposition |

tu technische universität  
dortmund

```
\begin{Theorem}
  Dies ist ein Beispieltheorem.
\end{Theorem}
\begin{Beweis}
  Dies ist ein Beispielbeweis.
\end{Beweis}
```

**Theorem 1** *Dies ist ein Beispieltheorem. Dies ist ein Beispieltheorem. Dies ist ein Beispieltheorem. Dies ist ein Beispieltheorem. Dies ist ein Beispieltheorem. Dies ist ein Beispieltheorem.*  $\square$

**BEWEIS** Dies ist ein Beispielbeweis. Dies ist ein Beispielbeweis. Dies ist ein Beispielbeweis. Dies ist ein Beispielbeweis. Dies ist ein Beispielbeweis.  $\blacksquare$

Wir wollen Definitionen, Theorem, o.ä. in unserer Arbeit referenzieren

**Wir wollen** Definitionen, Theorem, o.ä. in unserer Arbeit referenzieren

**Problem** Nummerierungen können sich ändern → Referenzen müssten korrigiert werden

**Wir wollen** Definitionen, Theorem, o.ä. in unserer Arbeit referenzieren

**Problem** Nummerierungen können sich ändern → Referenzen müssten korrigiert werden

**Lösung** Referenzierung über Labels!

**Wir wollen** Definitionen, Theorem, o.ä. in unserer Arbeit referenzieren

**Problem** Nummerierungen können sich ändern → Referenzen müssten korrigiert werden

**Lösung** Referenzierung über Labels!

```
\begin{Theorem} \label{theorem_kompliziert}  
    Komplizierte Formel  
\end{Theorem}
```

**Wir wollen** Definitionen, Theorem, o.ä. in unserer Arbeit referenzieren

**Problem** Nummerierungen können sich ändern → Referenzen müssten korrigiert werden

**Lösung** Referenzierung über Labels!

```
\begin{Theorem} \label{theorem_kompliziert}  
    Komplizierte Formel  
\end{Theorem}
```

Theorem nun referenzierbar:

```
Wie bereits in Theorem~\ref{theorem_kompliziert}  
gezeigt, ...
```

- Referenzen erst beim zweiten Kompilieren aufgelöst



- Referenzen erst beim zweiten Kompilieren aufgelöst
- $\text{\LaTeX}$  ersetzt `\ref`-Befehl nur durch Nummer

- Referenzen erst beim zweiten Kompilieren aufgelöst
- $\text{\LaTeX}$  ersetzt `\ref`-Befehl nur durch Nummer
- Auch Abbildungen und Algorithmen und (Unter-)Abschnitte referenzierbar

- Referenzen erst beim zweiten Kompilieren aufgelöst
- $\text{\LaTeX}$  ersetzt `\ref`-Befehl nur durch Nummer
- Auch Abbildungen und Algorithmen und (Unter-)Abschnitte referenzierbar

**Tipp**

- 1 Namensschema für Referenzen
- 2 Nicht mit Labels sparen!

- Parameter „`{l|c|r}`“ gibt Spaltenformatierung an:
  - `l/c/r` für eine linksbündige/zentrierte/rechtsbündige Spalte
  - „`|`“ trennt zwei Spalten durch vertikale Linie
- `&` und `\\`: Trennung und Abschluss der Zeilenelemente

```
\begin{tabular}{l|c|r}  
    Feld 1 & Feld 2 & Feld 3 \\  
    Feld 4 & Feld 5 & Feld 6 \\  
\end{tabular}
```

- Parameter „`{l|c|r}`“ gibt Spaltenformatierung an:
  - `l/c/r` für eine linksbündige/zentrierte/rechtsbündige Spalte
  - „`|`“ trennt zwei Spalten durch vertikale Linie
- `&` und `\\`: Trennung und Abschluss der Zeilenelemente

```
\begin{tabular}{l|c|r}  
    Feld 1 & Feld 2 & Feld 3 \\  
    Feld 4 & Feld 5 & Feld 6 \\  
\end{tabular}
```

**Wichtig** Für jede Spalte muss eine Formatierung vorliegen

- Parameter „`{l|c|r}`“ gibt Spaltenformatierung an:
  - `l/c/r` für eine linksbündige/zentrierte/rechtsbündige Spalte
  - „`|`“ trennt zwei Spalten durch vertikale Linie
- `&` und `\\`: Trennung und Abschluss der Zeilenelemente

```
\begin{tabular}{l|c|r}  
    Feld 1 & Feld 2 & Feld 3 \\  
    Feld 4 & Feld 5 & Feld 6 \\  
\end{tabular}
```

**Wichtig** Für jede Spalte muss eine Formatierung vorliegen

**Aber** Zeilen müssen nicht komplett gefüllt werden

- Parameter „`{l|c|r}`“ gibt Spaltenformatierung an:
  - `l/c/r` für eine linksbündige/zentrierte/rechtsbündige Spalte
  - „`|`“ trennt zwei Spalten durch vertikale Linie
- `&` und `\\`: Trennung und Abschluss der Zeilenelemente

```
\begin{tabular}{l|c|r}  
    Feld 1 & Feld 2 & Feld 3 \\  
    Feld 4 & Feld 5 & Feld 6 \\  
\end{tabular}
```

**Wichtig** Für jede Spalte muss eine Formatierung vorliegen

**Aber** Zeilen müssen nicht komplett gefüllt werden

**Tipp** `\hline` fügt vertikale Linie zwischen Zeilen ein

Tabelle als eigenständiges Objekt:

```
\begin{table}[<param>]
  \begin{tabular}
    ...
  \end{tabular}
  \caption{Beschreibung der Tabelle}
\end{table}
```

Positionierung über optionale Parameter

- h - „here“
- t / b - oben/unten auf der Seite
- ...



Benötigt zusätzliches Paket

- z.B. wrapfig

Benötigt zusätzliches Paket

- z.B. wrapfig

Abbildungen sind eigenständige Objekte

```
\usepackage{wrapfig}
...
\begin{figure}[<param>]
    beliebiger Inhalt ...
\end{figure}
```

Benötigt zusätzliches Paket

- z.B. wrapfig

Abbildungen sind eigenständige Objekte

```
\usepackage{wrapfig}
...
\begin{figure}[<param>]
    beliebiger Inhalt ...
\end{figure}
```

- Positionierung und Bildunterschrift wie bei Tabellen
- Kann beliebigen Inhalt (auch Text) enthalten

- Können innerhalb einer figure-Umgebung mit `includegraphics` eingefügt werden
- Benötigt das Paket `graphicx`

```
\usepackage{graphicx}
...
\begin{figure}[<param>]
  \centering
  \includegraphics[scale = .3]
                    {bilder/beispiel_1}
  \includegraphics[height=50px,width=.50\textwidth]
                    {bilder/beispiel_2}
  \caption{Bildunterschrift}
\end{figure}
```

Was ist PGF?

Was ist PGF?

- „Portable Graphics Format“

Was ist PGF?

- „Portable Graphics Format“
- Makropaket zur Erstellung von Vektorgrafiken

Was ist PGF?

- „Portable Graphics Format“
- Makropaket zur Erstellung von Vektorgrafiken

Was ist TikZ?



Was ist PGF?

- „Portable Graphics Format“
- Makropaket zur Erstellung von Vektorgrafiken

Was ist TikZ?

- „**TikZ** ist **kein Zeichenprogramm!**“

Was ist PGF?

- „Portable Graphics Format“
- Makropaket zur Erstellung von Vektorgrafiken

Was ist TikZ?

- „**TikZ** ist **kein Zeichenprogramm!**“
- Benutzerfreundliche Schnittstelle für PGF

Was ist PGF?

- „Portable Graphics Format“
- Makropaket zur Erstellung von Vektorgrafiken

Was ist TikZ?

- „**TikZ** ist **kein Zeichenprogramm!**“
- Benutzerfreundliche Schnittstelle für PGF
- Interpretation durch  $\text{\LaTeX}$

Was ist PGF?

- „Portable Graphics Format“
- Makropaket zur Erstellung von Vektorgrafiken

Was ist TikZ?

- „**TikZ** ist **kein Zeichenprogramm!**“
- Benutzerfreundliche Schnittstelle für PGF
- Interpretation durch  $\text{\LaTeX}$

Warum PGF/TikZ?

Was ist PGF?

- „Portable Graphics Format“
- Makropaket zur Erstellung von Vektorgrafiken

Was ist TikZ?

- „**TikZ** ist **kein Zeichenprogramm!**“
- Benutzerfreundliche Schnittstelle für PGF
- Interpretation durch  $\text{\LaTeX}$

Warum PGF/TikZ?

- Einfache Möglichkeit, um Grafiken zu erstellen

## Was ist PGF?

- „Portable Graphics Format“
- Makropaket zur Erstellung von Vektorgrafiken

## Was ist TikZ?

- „**TikZ** ist **kein Zeichenprogramm!**“
- Benutzerfreundliche Schnittstelle für PGF
- Interpretation durch  $\text{\LaTeX}$

## Warum PGF/TikZ?

- Einfache Möglichkeit, um Grafiken zu erstellen
  - Beispiele: Graphen, Bäume, Automaten, Skizzen, etc.

## Was ist PGF?

- „Portable Graphics Format“
- Makropaket zur Erstellung von Vektorgrafiken

## Was ist TikZ?

- „**TikZ** ist **kein Zeichenprogramm!**“
- Benutzerfreundliche Schnittstelle für PGF
- Interpretation durch  $\text{\LaTeX}$

## Warum PGF/TikZ?

- Einfache Möglichkeit, um Grafiken zu erstellen
  - Beispiele: Graphen, Bäume, Automaten, Skizzen, etc.
- Umfangreiche Dokumentation

Paket einbinden:

```
\usepackage{tikz}
```



Paket einbinden:

```
\usepackage{tikz}
```

Bei Bedarf Zusatzbibliotheken einbinden:

```
\usetikzlibrary{automata, arrows, shapes, etc.}
```

# Installation und Verwendung von TikZ

Paket einbinden:

```
\usepackage{tikz}
```

Bei Bedarf Zusatzbibliotheken einbinden:

```
\usetikzlibrary{automata, arrows, shapes, etc.}
```

Zwei Verwendungsarten:

**Inline** Präfix `\tikz` vor jeden Befehl

**Umgebung** Mit `tikzpicture`-Umgebung:

# Installation und Verwendung von TikZ

Paket einbinden:

```
\usepackage{tikz}
```

Bei Bedarf Zusatzbibliotheken einbinden:

```
\usetikzlibrary{automata, arrows, shapes, etc.}
```

Zwei Verwendungsarten:

**Inline** Präfix `\tikz` vor jeden Befehl

**Umgebung** Mit `tikzpicture`-Umgebung:

```
\begin{tikzpicture}[<opt. Param.>]  
  ...  
\end{tikzpicture}
```

Grundlegender Zeichenbefehl: `\path`

```
\path [<opt>] <Zeichenaktionen>;
```

Grundlegender Zeichenbefehl: `\path`

```
\path[<opt>] <Zeichenaktionen>;
```

Beispiel:

```
\path[draw] (0,0) -- (2,0) -- (2,1);
```

Zeichnet eine Linie vom Punkt  $(0,0)$  über  $(2,0)$  zum Punkt  $(2,1)$ .

Grundlegender Zeichenbefehl: `\path`

```
\path[<opt>] <Zeichenaktionen>;
```

Beispiel:

```
\path[draw] (0,0) -- (2,0) -- (2,1);
```

Zeichnet eine Linie vom Punkt (0,0) über (2,0) zum Punkt (2,1).

<code>draw</code>	Zeichenart (hier: Zeichnen), weitere Optionen: <code>fill</code>
<code>(x,y)</code>	Absolute Koordinaten
<code>--</code>	Was gezeichnet wird (hier: Linie)

Grundlegender Zeichenbefehl: `\path`

```
\path[<opt>] <Zeichenaktionen>;
```

Beispiel:

```
\path[draw] (0,0) -- (2,0) -- (2,1);
```

Zeichnet eine Linie vom Punkt  $(0,0)$  über  $(2,0)$  zum Punkt  $(2,1)$ .

<code>draw</code>	Zeichenart (hier: Zeichnen), weitere Optionen: <code>fill</code>
<code>(x,y)</code>	Absolute Koordinaten
<code>--</code>	Was gezeichnet wird (hier: Linie)

Ergebnis: 

```
\path[draw] (0,0) rectangle (2,1);
```





```
\path[draw] (0,0) rectangle (2,1);
```



```
\path[fill] (0,0) circle (1);
```



```
\path[draw] (0,0) rectangle (2,1);
```



```
\path[fill] (0,0) circle (1);
```



```
\path[draw=blue,fill=red] (0,0) -- (2,0) -- (2,1);
```



- Knoten sind Objekte auf der Zeichenfläche

- Knoten sind Objekte auf der Zeichenfläche
- Beistzen u.a. eine Form, einen Inhalt und ein Label

- Knoten sind Objekte auf der Zeichenfläche
- Beistzen u.a. eine Form, einen Inhalt und ein Label
- Können über Bezeichner referenziert werden

- Knoten sind Objekte auf der Zeichenfläche
- Beistzen u.a. eine Form, einen Inhalt und ein Label
- Können über Bezeichner referenziert werden

Syntax:

```
\node [<opt. Param>] (<name>) at (x,y) {<Inhalt>};
```

- Knoten sind Objekte auf der Zeichenfläche
- Beistzen u.a. eine Form, einen Inhalt und ein Label
- Können über Bezeichner referenziert werden

Syntax:

```
\node[<opt. Param>] (<name>) at (x,y) {<Inhalt>;}
```

Beispiel:

```
\node (v1) at (0,0) {$v_1$};  
\node[circle,draw] (v2) at (2,0) {$v_2$};  
\node[rectangle,draw] (v3) at (4,0) {$v_3$};  
\node[rectangle,draw, right of=v3] (v4) {$v_4$};
```

- Knoten sind Objekte auf der Zeichenfläche
- Beistzen u.a. eine Form, einen Inhalt und ein Label
- Können über Bezeichner referenziert werden

Syntax:

```
\node[<opt. Param>] (<name>) at (x,y) {<Inhalt>;}
```

Beispiel:

```
\node (v1) at (0,0) {$v_1$};  
\node[circle,draw] (v2) at (2,0) {$v_2$};  
\node[rectangle,draw] (v3) at (4,0) {$v_3$};  
\node[rectangle,draw, right of=v3] (v4) {$v_4$};
```

**Beachte** Der Knoteninhalt ist obligatorisch (notfalls Klammern leer lassen)!



Syntax:

```
\path (<Start>) edge[<params>] node {<Label>} (<Ziel>);
```

Syntax:

```
\path (<Start>) edge[<params>] node {<Label>} (<Ziel>);
```

Beispiel:

```
\begin{tikzpicture}[auto]{Beispiel}  
  \node (v1) at (0,0) {$v_1$};  
  \node[circle,draw] (v2) at (2,0) {$v_2$};  
  \node[rectangle,draw] (v3) at (4,0) {$v_3$};  
  \node[rectangle,draw, right of=v3] (v4) {$v_4$};  
  \path (v2) edge[->,red,thick] node {1} (v3);  
  \path (v2) edge[->,blue,bend left=45] node {2} (v4);  
\end{tikzpicture}
```

Syntax:

```
\path (<Start>) edge[<params>] node {<Label>} (<Ziel>);
```

Beispiel:

```
\begin{tikzpicture}[auto]{Beispiel}  
  \node (v1) at (0,0) {$v_1$};  
  \node[circle,draw] (v2) at (2,0) {$v_2$};  
  \node[rectangle,draw] (v3) at (4,0) {$v_3$};  
  \node[rectangle,draw, right of=v3] (v4) {$v_4$};  
  \path (v2) edge[->,red,thick] node {1} (v3);  
  \path (v2) edge[->,blue,bend left=45] node {2} (v4);  
\end{tikzpicture}
```

- Der Teil node {<Label>} ist optional
- Parameter auto verbessert die Anordnung von Labels

Vorteile:

Vorteile:

- Aufteilung zwischen Beschreibung und Bilderstellung

Vorteile:

- Aufteilung zwischen Beschreibung und Bilderstellung
- Vektorgrafiken, Elemente einheitlich

Vorteile:

- Aufteilung zwischen Beschreibung und Bilderstellung
- Vektorgrafiken, Elemente einheitlich
- Leicht nachträglich modifizierbar

## Vorteile:

- Aufteilung zwischen Beschreibung und Bilderstellung
- Vektorgrafiken, Elemente einheitlich
- Leicht nachträglich modifizierbar
- Viele weitere Möglichkeiten (Plots, Diagramme, etc.)



Vorteile:

- Aufteilung zwischen Beschreibung und Bilderstellung
- Vektorgrafiken, Elemente einheitlich
- Leicht nachträglich modifizierbar
- Viele weitere Möglichkeiten (Plots, Diagramme, etc.)

Nachteile:

## Vorteile:

- Aufteilung zwischen Beschreibung und Bilderstellung
- Vektorgrafiken, Elemente einheitlich
- Leicht nachträglich modifizierbar
- Viele weitere Möglichkeiten (Plots, Diagramme, etc.)

## Nachteile:

- Vielfalt für Einsteiger schwierig

## Vorteile:

- Aufteilung zwischen Beschreibung und Bilderstellung
- Vektorgrafiken, Elemente einheitlich
- Leicht nachträglich modifizierbar
- Viele weitere Möglichkeiten (Plots, Diagramme, etc.)

## Nachteile:

- Vielfalt für Einsteiger schwierig
- Gleiches Ergebnis auf viele Weisen erreichbar

Pseudocode lässt sich direkt in  $\text{\LaTeX}$  erstellen:

---

**Algorithmus** Maximumsberechnung

---

**Require:** Integer-Array  $a$

**Ensure:** Maximum aus  $a$

```
1:  $max = a[0]$ 
2: for  $i = 1$  to  $a.length - 1$  do
3:   if  $a[i] > max$  then
4:      $max = a[i]$ 
return  $max$ 
```

---

Pseudocode lässt sich direkt in  $\LaTeX$  erstellen:

---

## **Algorithmus** Maximumsberechnung

---

**Require:** Integer-Array  $a$

**Ensure:** Maximum aus  $a$

```
1:  $max = a[0]$ 
2: for  $i = 1$  to  $a.length - 1$  do
3:   if  $a[i] > max$  then
4:      $max = a[i]$ 
   return  $max$ 
```

---

Zwei Pakete benötigt:

```
\usepackage{algorithm}
\usepackage{algpseudocode}
```

- Beinhaltet den eigentlichen Psuedocode

- Beinhaltet den eigentlichen Psuedocode
- Enthält vordefinierte Befehle für Anweisungen
  - `\State`, `\If`, `\For`, `\While`, etc.

- Beinhaltet den eigentlichen Psuedocode
- Enthält vordefinierte Befehle für Anweisungen
  - `\State`, `\If`, `\For`, `\While`, etc.
- If-, For- und While-Anweisungen müssen mit `\End*` beendet werden



- Beinhaltet den eigentlichen Psuedocode
- Enthält vordefinierte Befehle für Anweisungen
  - `\State`, `\If`, `\For`, `\While`, etc.
- If-, For- und While-Anweisungen müssen mit `\End*` beendet werden

```
\begin{algorithmic}
  \State i = 0
  \While{i < a.length}
    \If{a[i] = 0}
      \State a[i] = 0
    \Else
      \State a[i] = -1
    \EndIf
    \State i = i + 1
  \EndWhile
\end{algorithmic}
```

- Bettet den Pseudocode ein (ähnlich wie eine Abbildung)

- Bettet den Pseudocode ein (ähnlich wie eine Abbildung)
- Bietet Optionen für Titel, Bezeichnung, Referenzierung, etc.

- Bettet den Pseudocode ein (ähnlich wie eine Abbildung)
- Bietet Optionen für Titel, Bezeichnung, Referenzierung, etc.

```
\begin{algorithm}  
\floatname{algorithm}{Algorithmus}  
\caption{Berechnung von xyz}  
  \begin{algorithmic}  
  ...  
  \end{algorithmic}  
\end{algorithm}
```

Ein- und Ausgabe:

```
\renewcommand{\algorithmicrequire}{\textbf{Eingabe:}}  
\renewcommand{\algorithmicensure}{\textbf{Ausgabe:}}
```

```
\begin{algorithmic}  
\Require Eingabetext  
\Ensure Ausgabertext  
...  
\end{algorithmic}  
\end{algorithm}
```

Ein- und Ausgabe:

```
\renewcommand{\algorithmicrequire}{\textbf{Eingabe:}}  
\renewcommand{\algorithmicensure}{\textbf{Ausgabe:}}
```

```
\begin{algorithmic}  
\Require Eingabetext  
\Ensure Ausgabertext  
...  
\end{algorithmic}  
\end{algorithm}
```

Zeilennummerierung:

```
\begin{algorithmic}[1]
```

Zum Schluss Ein paar Regeln und Hinweise

Zum Schluss Ein paar Regeln und Hinweise

- Keine Anleitung, wie man wissenschaftliche Texte schreibt



## Zum Schluss Ein paar Regeln und Hinweise

- Keine Anleitung, wie man wissenschaftliche Texte schreibt
- Eher technische Hinweise für Schreiben in der Gruppe

## Zum Schluss Ein paar Regeln und Hinweise

- Keine Anleitung, wie man wissenschaftliche Texte schreibt
- Eher technische Hinweise für Schreiben in der Gruppe

Implizit vorausgesetzt:

## Zum Schluss Ein paar Regeln und Hinweise

- Keine Anleitung, wie man wissenschaftliche Texte schreibt
- Eher technische Hinweise für Schreiben in der Gruppe

Implizit vorausgesetzt:

- ### ■ Angemessene Wortwahl/Sprache

## Zum Schluss Ein paar Regeln und Hinweise

- Keine Anleitung, wie man wissenschaftliche Texte schreibt
- Eher technische Hinweise für Schreiben in der Gruppe

Implizit vorausgesetzt:

- Angemessene Wortwahl/Sprache
- Keine Plagiate!

## Zum Schluss Ein paar Regeln und Hinweise

- Keine Anleitung, wie man wissenschaftliche Texte schreibt
- Eher technische Hinweise für Schreiben in der Gruppe

## Implizit vorausgesetzt:

- Angemessene Wortwahl/Sprache
- Keine Plagiate!
- Kurze und prägnante Ausdrucksweise

## Zum Schluss Ein paar Regeln und Hinweise

- Keine Anleitung, wie man wissenschaftliche Texte schreibt
- Eher technische Hinweise für Schreiben in der Gruppe

## Implizit vorausgesetzt:

- Angemessene Wortwahl/Sprache
- Keine Plagiate!
- Kurze und prägnante Ausdrucksweise
- Stoffauswahl und -gruppierung vor dem Schreiben

## Zum Schluss Ein paar Regeln und Hinweise

- Keine Anleitung, wie man wissenschaftliche Texte schreibt
- Eher technische Hinweise für Schreiben in der Gruppe

## Implizit vorausgesetzt:

- Angemessene Wortwahl/Sprache
- Keine Plagiate!
- Kurze und prägnante Ausdrucksweise
- Stoffauswahl und -gruppierung vor dem Schreiben
- Text nicht in einem Lauf „durschreiben“

## Zum Schluss Ein paar Regeln und Hinweise

- Keine Anleitung, wie man wissenschaftliche Texte schreibt
- Eher technische Hinweise für Schreiben in der Gruppe

## Implizit vorausgesetzt:

- Angemessene Wortwahl/Sprache
- Keine Plagiate!
- Kurze und prägnante Ausdrucksweise
- Stoffauswahl und -gruppierung vor dem Schreiben
- Text nicht in einem Lauf „durschreiben“
- ...



Grundsätzlich Gliederung sollte vor dem Schreiben bekannt sein!

Für Gliederung die Möglichkeiten von  $\text{\LaTeX}$  nutzen:

Grundsätzlich Gliederung sollte vor dem Schreiben bekannt sein!

Für Gliederung die Möglichkeiten von  $\text{\LaTeX}$  nutzen:

- Kapitel (chapter)

Grundsätzlich Gliederung sollte vor dem Schreiben bekannt sein!

Für Gliederung die Möglichkeiten von  $\text{\LaTeX}$  nutzen:

- Kapitel (`chapter`)
- Abschnitt (`section`)

Grundsätzlich Gliederung sollte vor dem Schreiben bekannt sein!

Für Gliederung die Möglichkeiten von  $\text{\LaTeX}$  nutzen:

- Kapitel (`chapter`)
- Abschnitt (`section`)
- Unterabschnitt (`subsection`)

Grundsätzlich Gliederung sollte vor dem Schreiben bekannt sein!

Für Gliederung die Möglichkeiten von  $\text{\LaTeX}$  nutzen:

- Kapitel (`chapter`)
- Abschnitt (`section`)
- Unterabschnitt (`subsection`)
- ...

**Grundsätzlich** Gliederung sollte vor dem Schreiben bekannt sein!

Für Gliederung die Möglichkeiten von  $\text{\LaTeX}$  nutzen:

- Kapitel (`chapter`)
- Abschnitt (`section`)
- Unterabschnitt (`subsection`)
- ...

Gliederungseinheiten sollen in sich geschlossen sein

**Grundsätzlich** Gliederung sollte vor dem Schreiben bekannt sein!

Für Gliederung die Möglichkeiten von  $\text{\LaTeX}$  nutzen:

- Kapitel (`chapter`)
- Abschnitt (`section`)
- Unterabschnitt (`subsection`)
- ...

Gliederungseinheiten sollen in sich geschlossen sein

- Neuer Gedanke/neues Problem/neues Thema  
→ neuer Absatz/neuer Abschnitt/neues Kapitel

**Grundsätzlich** Gliederung sollte vor dem Schreiben bekannt sein!

Für Gliederung die Möglichkeiten von  $\text{\LaTeX}$  nutzen:

- Kapitel (`chapter`)
- Abschnitt (`section`)
- Unterabschnitt (`subsection`)
- ...

Gliederungseinheiten sollen in sich geschlossen sein

- Neuer Gedanke/neues Problem/neues Thema  
→ neuer Absatz/neuer Abschnitt/neues Kapitel
- Wiederum unterteilt in Einleitung und Hauptteil



Abschnitte und Kapitel immer mit Text einleiten!

## **Kapitel 3**

### **Abschnitt 3.1**

Text zum Abschnitt



## **Kapitel 3**

In diesem Kapitel geht es um ...

### **Abschnitt 3.1**

Text zum Abschnitt



- Abschnitte nummerieren (automatisch bei Verwendung der  $\LaTeX$ -Befehle)
- Mindestens zwei (oder gar keinen) Unterabschnitt

## Kapitel 3

In diesem Kapitel geht es um ...

### Abschnitt 3.1

Text zum Abschnitt

## Kapitel 4

...



**Klar** Definitionen und Theoreme müssen in logisch nachvollziehbarer Reihenfolge vorkommen

**Klar** Definitionen und Theoreme müssen in logisch nachvollziehbarer Reihenfolge vorkommen

Erleichterung für den Leser

- Wichtige Definitionen in eigenem Abschnitt bündeln

**Klar** Definitionen und Theoreme müssen in logisch nachvollziehbarer Reihenfolge vorkommen

Erleichterung für den Leser

- Wichtige Definitionen in eigenem Abschnitt bündeln
- Neue oder wichtige Begriffe *hervorheben* (aber nicht **übertreiben**)

**Klar** Definitionen und Theoreme müssen in logisch nachvollziehbarer Reihenfolge vorkommen

Erleichterung für den Leser

- Wichtige Definitionen in eigenem Abschnitt bündeln
- Neue oder wichtige Begriffe *hervorheben* (aber nicht **übertreiben**)
- Nebendefinitionen sollten lokal erfolgen

**Klar** Definitionen und Theoreme müssen in logisch nachvollziehbarer Reihenfolge vorkommen

Erleichterung für den Leser

- Wichtige Definitionen in eigenem Abschnitt bündeln
- Neue oder wichtige Begriffe *hervorheben* (aber nicht **übertreiben**)
- Nebendefinitionen sollten lokal erfolgen
- Definitionen sollten nicht wiederholt werden (Leser kann zurückblättern, Zuhörer nicht)

Geschmacksfrage bzw. kontrovers:

- „Wir“-Form
- Passivkonstruktionen



Geschmacksfrage bzw. kontrovers:

- „Wir“-Form
- Passivkonstruktionen

Grundsätzlich unterscheiden während des Schreibens:

- Was stammt von mir? Was von anderen Autoren?
- Auf Konsistenz achten

Geschmacksfrage bzw. kontrovers:

- „Wir“-Form
- Passivkonstruktionen

Grundsätzlich unterscheiden während des Schreibens:

- Was stammt von mir? Was von anderen Autoren?
- Auf Konsistenz achten

Im ersten Schritt bestimmen wir ein perfektes Matching auf dem gegebenen Graphen.



Im ersten Schritt bestimmt der Algorithmus ein perfektes Matching auf dem gegebenen Graphen.



Guter Schreibstil:

- Keine mathematische Formel am Satzanfang

Guter Schreibstil:

- Keine mathematische Formel am Satzanfang
- Niemals zwei Formeln unmittelbar hintereinander

## Guter Schreibstil:

- Keine mathematische Formel am Satzanfang
- Niemals zwei Formeln unmittelbar hintereinander
- Aussagen werden wie Objekte behandelt: Satz muss grammatikalisch korrekt bleiben, wenn Aussage durch Bezeichnet ersetzt wird

## Guter Schreibstil:

- Keine mathematische Formel am Satzanfang
- Niemals zwei Formeln unmittelbar hintereinander
- Aussagen werden wie Objekte behandelt: Satz muss grammatikalisch korrekt bleiben, wenn Aussage durch Bezeichnet ersetzt wird

$A \Rightarrow B$  heißt, dass aus  $A = \top$   $B = \top$  folgt.



## Guter Schreibstil:

- Keine mathematische Formel am Satzanfang
- Niemals zwei Formeln unmittelbar hintereinander
- Aussagen werden wie Objekte behandelt: Satz muss grammatikalisch korrekt bleiben, wenn Aussage durch Bezeichnet ersetzt wird

$A \Rightarrow B$  heißt, dass aus  $A = \top$   $B = \top$  folgt.



$A \Rightarrow B$  heißt, dass aus  $A = \top$  die Aussage  $B = \top$  folgt. (✓)

## Guter Schreibstil:

- Keine mathematische Formel am Satzanfang
- Niemals zwei Formeln unmittelbar hintereinander
- Aussagen werden wie Objekte behandelt: Satz muss grammatikalisch korrekt bleiben, wenn Aussage durch Bezeichnet ersetzt wird

$A \Rightarrow B$  heißt, dass aus  $A = \top$   $B = \top$  folgt.



$A \Rightarrow B$  heißt, dass aus  $A = \top$  die Aussage  $B = \top$  folgt. (✓)

Die Aussage  $A \Rightarrow B$  bedeutet, dass aus  $A = \top$  die Aussage  $B = \top$  folgt.





Vielen Dank für die  
Aufmerksamkeit!