

# SEMINAR MMI - VORLAGE FÜR SCHRIFTLICHE AUSARBEITUNGEN

Christian Schlette

schlette@mmi.rwth-aachen.de

---

## **Kurzfassung**

Dieses Dokument dient als Anleitung und als Vorlage für die schriftlichen Ausarbeitungen für Seminare am *Institut für Mensch-Maschine-Interaktion (MMI)*. Einsteiger in  $\LaTeX$  sollten parallel mit einem Ausdruck dieser Vorlage und dem Quellcode arbeiten, um so leicht entsprechende Befehle für ein gewünschtes Layout zu finden. Helfen dieses Dokument sowie angegebene Literatur nicht weiter, stehen natürlich die Betreuer für Fragen bereit.

*Stichworte:*  $\LaTeX$ , mmiSeminar-Style, Vorlage

---

# 1 Einführung

---

$\LaTeX$  basiert auf  $\TeX$ <sup>1</sup>, einem Textsatzsystem, das von Donald E. Knuth [Knu84] entwickelt und veröffentlicht wurde, um wissenschaftlichen Texten ein einheitliches und gutes Layout zu geben. Aufgrund seiner komplexen Syntax ist die direkte Verwendung von  $\TeX$  jedoch nicht leicht, deshalb hat Leslie Lamport [Lam94] das Makro-Paket  $\LaTeX$  entwickelt. Dieses definiert Makros, die es dem Benutzer erlauben, komplexere und häufig benötigte Folgen von  $\TeX$ -Kommandos zu verwenden. Die neuste und vereinheitlichte Version ist  $\LaTeX 2_{\epsilon}$  das wir hier verwenden.

## 1.1 Warum $\LaTeX$

$\LaTeX$  (wegen besserer Lesbarkeit auch einfach LaTeX) ist eine Makro-Sprache und muss ähnlich einer Programmiersprache erlernt werden. Entsprechend muss  $\TeX$ -Code erst kompiliert oder „*getext*“ werden, bevor das tatsächliche Layout sichtbar wird. Doch die Vorteile der Verwendung von LaTeX für professionelle wissenschaftliche Texte treten besonders bei längeren (etwa 100 Seiten) Texten mit vielen Bildern und Formeln zutage. LaTeX setzt Grundregeln um, an denen sich auch professionelle Layouter orientieren, um einen Text gut lesbar und optisch ansprechend zu gestalten. Zusätzlich wird durch die festen Befehle (z. B. für Kapitelanfänge) ein hohes Maß an Konsistenz erreicht. Arbeiten im technischen Bereich werden daher häufig in  $\TeX$  ausgeführt. Zum Beispiel verlangen viele Institute verlangen, dass Studien- und Diplomarbeiten in LaTeX geschrieben sind. Fast alle internationalen Zeitschriften und Konferenzen erwarten  $\TeX$ -Dokumente, für die sie entsprechende eigene *Styles* entwickelt haben.

## 1.2 Vom Quelltext zum PostScript

### Dateien rund um $\LaTeX$

Der Quellcode für jedes LaTeX-Dokument steht in einer *.tex*-Datei. Mit dem Befehl `input` können weitere Dateien eingebunden werden. Dies hat den gleichen Effekt, als ob der Quellcode an Stelle des `input`-Befehls stünde<sup>2</sup>. Befehle in LaTeX beginnen übrigens immer mit einem Backslash `\`.

In wissenschaftlichen Arbeiten werden sehr häufig Ergebnisse anderer Forscher zitiert. Im technischen Bereich geschieht dies durch eine Abkürzung oder Indexnummer, die im *Literaturverzeichnis* wiederzufinden ist. Dort stehen dann genaue Angaben zu der jeweiligen Veröffentlichung. In LaTeX gibt es ein einheitliches Format in dem die Daten einer Veröffentlichung, wie z. B. Autor, Titel, Jahr, usw. gespeichert werden. Mittels verschiedener Bibliography-Styles kann das Layout des Literaturverzeichnisses leicht verändert werden. Die Angaben zu den zitierten Arbeiten stehen in der Bibliography (*.bib*-Datei), die dann von BibTeX ausgewertet wird.

Zur Umsetzung verschiedener Layouts im Text wird ein ähnliches Vorgehen wie für das Literaturverzeichnis angewandt. Zunächst gibt es die *Dokumentenklasse* (Befehl `documentclass`

---

<sup>1</sup>sprich: Tech

<sup>2</sup>siehe auch Datei `frame.tex`

in `frame.tex`), die die grundsätzliche Formatierung festlegt. Dann erweitern die Styles (in `.sty`-Dateien) die Dokumentenklassen und bieten weitere Makros an, die die Arbeit mit LaTeX vereinfachen sollen. Ein Beispiel ist `iso8859-1.sty`: Ohne diesen Style kann man Umlaute nicht direkt in den Text schreiben. Der `mmiSeminar-Style` dagegen dient der einheitlichen Formatierung und einfachen Erstellung der schriftlichen Ausarbeitungen zur späteren Zusammenführung in einem Seminarband.

## Von LaTeX zu PostScript

Der erste Schritt um aus dem geschriebenen Quellcode eine formatierte PostScript-Version zu erstellen ist das so genannte „*Texen*“ — das Ausführen des `latex` Befehls. Dies geht entweder über die Konsole mit Angabe des Dateinamens der Master-Datei (hier: `frame`), oder per Menüpunkt im verwendeten Editor. In diesem Schritt werden einige Hilfsdateien mit gleichem Namen wie die Master-Datei erstellt. Der eigentliche Text steht in der `.dvi`-Datei (`dvi` steht für `device independent`). Unter Linux und unter Windows existieren Programme zur Anzeige dieser Dateien.

Je nachdem, welche Änderungen im Quellcode vorgenommen wurden, muss `latex` bis zu dreimal ausgeführt werden. Um alle Verweise zu finden, erstellt das Programm zunächst Tabellen, die im nächsten Schritt verwendet werden. Beim ersten `Texen` wird auch eine Liste der Literaturverweise (`cite`) erstellt. Wurden die `.bib`-Datei verändert muss auch der Befehl `bibtex` Master-Datei ausgeführt werden. Hierauf folgen in der Regel zwei Aufrufe von `latex`, bis alle Referenzen korrekt sind. Komfortable LaTeX-Entwicklungsumgebungen unter Linux und unter Windows führen die notwendigen Schritte auf Knopfdruck automatisch aus.

Aus der nun korrekten `.dvi`-Datei wird eine PostScript-Datei (`.ps`) mit dem Befehl `dvips -t a4 Master-File` erzeugt. Der Parameter `-t` gibt die Papiergröße A4 an. Die Endung des Master-Files ist wieder wegzulassen. Das Ergebnis kann unter Linux mit `gv` oder unter Windows mit `ghostview` betrachtet werden.

Für das Seminar ist die Befehlsfolge, die bei korrektem Quellcode immer zu einem PostScript führen sollte:

```
latex frame
bibtex frame
latex frame
latex frame
```

## 1.3 Tools

Die für Linux und Windows zur Verfügung stehenden Hilfsmittel unterscheiden sich, obwohl viele der Linux-üblichen Programme ebenso für Windows erhältlich sind. LaTeX kommt aus der Unix-Welt und daher ist hier etwas leichter zu verwenden, insbesondere sind die benötigten Programme bereits installiert. Aber auch unter Windows sind alle Anwendungen, die zum Arbeiten mit LaTeX gebraucht werden, kostenlos verfügbar. Die jeweiligen Werkzeuge sind in den nächsten Abschnitten kurz beschrieben.

## Unter Linux

Die Programmpakete LaTeX, Emacs und dvips müssen installiert werden. Emacs bietet für praktisch alle unter Linux üblichen Dateiformate Syntax-Highlighting und besondere Makros, die das Programmieren vereinfachen. Dies trifft auch für .tex und .bib-Dateien zu. Die Kombination `Ctrl-C Ctrl-E` erzeugt z. B. beliebige LaTeX-Umgebungen. Für BiBTeX erscheint ein Menüpunkt, mit dem sich neue Einträge für das Literaturverzeichnis generieren lassen. PostScript-Dateien können mit den kostenlos in der Distribution enthaltenen Programmen ghostview und gv geöffnet werden. Grafiken können unter Linux mit xfig erstellt werden.

## Unter Windows

Alle Routinen, die für die Verwendung von LaTeX nötig sind, werden unter Windows im MikTeX-Paket geliefert (siehe <http://www.mmi.rwth-aachen.de/Lehre/Seminare/Download>). Zur Eingabe des Quellcodes kann prinzipiell jeder Editor verwendet werden, mit dem ASCII-Dateien geschrieben werden können. Empfehlenswert ist aber die Verwendung spezieller Entwicklungsumgebungen wie zum Beispiel TexnicCenter (siehe Downloads), die die beschriebenen Abläufe automatisieren. Viewer, wie Yap (zur Anzeige von dvi-Formaten) oder GhostView (zur Anzeige von PostScript-Dateien) werden getrennt installiert, stehen dann allerdings in die Entwicklungsumgebungen integriert zur Verfügung.

## Grafiken in LaTeX-Dokumenten

Grundsätzlich können in LaTeX nur *Encapsulated PostScript* Dateien (.eps) eingebunden werden. Diese sollten so erstellt werden, dass sie möglichst wenig Speicherbedarf haben. Da die Seminarbände nicht farbig gedruckt werden, können die Grafiken also als Grauwertbilder gespeichert werden. Weiterhin ist es vorteilhaft, die Bilder direkt auf die benötigte Größe zu bringen. Maximale Breite ist die Textbreite von 13,7 cm. Die maximale sinnvolle Auflösung für Bilder ist 150 dpi. Höhere Auflösungen gehen beim Kopiervorgang verloren. Diese Richtlinien gelten natürlich nur für die Umwandlung von Pixel-basierten Grafiken. Die Größe von Vektorgrafiken ist unabhängig von Skalierung und Auflösung.

Für die Erstellung von Vektorgrafiken eignen sich xfig, Visio, PowerPoint und CorelDraw. Grundsätzlich darf *nie* eine Vorschau mitgespeichert werden. Für das Bearbeiten von Pixel-basierten Grafiken eignen sich besonders Adobe PhotoShop und gimp. Im eps-Format enthaltene Bitmaps können mit JPEG komprimiert werden (Option beim Speichern), was für kleine Dateien sorgt.

Das Einscannen von Bildern sollte soweit wie möglich vermieden werden. Zum einen ist die Qualität meist nicht gut und die Dateien sehr groß. Zum anderen ist das Verwenden von Teilen aus Büchern selten gestattet. Optimal ist eine Nachahmung der Abbildung mit Verweis auf das Original: nach `\cite{OriginalAutor}`. Sollte dies nicht gehen, kann aus Artikeln das Bild oft mit Grafikprogrammen extrahiert werden. Bei Scans sollte darauf geachtet werden, dass das Rauschen möglichst klein ist, d. h. ein weisser Hintergrund sollte keine schwarzen Pünktchen und durchscheinende Seiten enthalten.

## 1.4 Organisation der Dateien

Um diesen Text texen zu können sind eine Reihe von Dateien nötig. Ihre Aufgaben sind im Folgenden kurz beschrieben:

- `frame.tex`  
Dient als Ersatz für die Master-Datei des gesamten Seminarbandes. Der Befehl `\input{MyText}` fügt dieses Dokument (`MyText.tex`) ein. „MyText“ ist durch den Namen der eigenen Datei zu ersetzen. Weiter unter wird mit dem Befehl `\bibliography{MyBib}` festgelegt, dass die Literaturhinweise in der Datei „MyBib.bib“ stehen. Diese sollte bei den Seminararbeiten den gleichen Namen wie der Text haben.
- `MyText.tex`  
Der Quelltext dieser Anleitung. Neben dieser Funktion dient die Datei auch als Vorlage für die Seminararbeiten. Mit Hilfe der PostScript Version und des Quellcodes können die Befehle um ein bestimmtes Layout zu erhalten leicht gefunden werden. Die eigene Arbeit soll den Namen `nachname.tex` haben (alles klein).
- `MyBib.bib`  
Enthält die Literaturangaben zu diesem Text. Die Datei enthält weiterhin Vorlagen für andere Typen von Literaturangaben für die Nicht-Emacs-User. Das eigene Literaturverzeichnis soll den Namen `nachname.bib` haben (alles klein).
- `xxx.tex xxx.sty xxx.bst`  
In diesen Dateien werden Styles und Makros definiert, die Erweiterungen oder Veränderungen von LaTeX darstellen. Sie dürfen auf keinen Fall verändert werden.

Weiterhin ist sehr wichtig, dass alle Bilder als `.eps` gespeichert sind und sich im Unterverzeichnis `bilder` relativ zu eurem Hauptfile befinden. Wenn also euer `nachname.tex` in `home/username/seminar` steht, gehören die Bilder in `home/username/seminar/bilder`. Um Verwechslungen vorzubeugen, müssen alle Abbildungen nach folgenden Schema benannt werden: `INITIALENDateiname.eps`. Hier wäre das z.B. `CSbeispiel1.eps`.

## 2 Sehr kurze Einführung in L<sup>A</sup>T<sub>E</sub>X \_\_\_\_\_

Für komplexere Aufgaben sind weitere Nachschlagewerke unverzichtbar. Eine recht umfassende Beschreibung ist kostenlos erhältlich: „*The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X2 $\epsilon$* “ [OPHS00], eine Einführung in Latex. Für schwierige Formeln ist das  $\mathcal{A}\mathcal{M}\mathcal{S}$ -L<sup>A</sup>T<sub>E</sub>X-Package sehr zu empfehlen. Das „ *$\mathcal{A}\mathcal{M}\mathcal{S}$ -L<sup>A</sup>T<sub>E</sub>X User's Guide* [Ame96] beschreibt diese häufig verwendete Erweiterung zu LaTeX. Die beiden Dokumente sind über die Seminar-Seite (<http://www.mmi.rwth-aachen.de/Lehre/Seminare/Download>) verlinkt. Weiterhin gibt es einige durchaus empfehlenswerte Bücher: „*L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*“, das Reference Manual [Lam94] und „*The L<sup>A</sup>T<sub>E</sub>X Companion*“ [GMS94], Erklärungen zu verschiedenen

Zusatzpackages u. a.  $\LaTeX$ . Ausserdem gibt es noch die drei Latex Bücher von Kopka, von denen nur das erste „*LaTeX Einführung*“ [Kop00] für den einfachen Anwender von LaTeX nötig ist.

## 2.1 Überschriften und Referenzen

Wie in diesem Beispieldokument zu sehen ist, gibt es für Artikel drei Hierarchieebenen der Überschriften. Sie werden mit den Befehlen `\section{...}`, `\subsection{...}` und `\subsubsection{...}` gesetzt. Dabei ist `...` durch die jeweilig Überschrift zu ersetzen. Innerhalb jedes Blocks kann durch den Befehl `label` eine Marke für spätere Referenzierung gesetzt werden. In allen solchen Marken sollten wiederum (wie bei Bilddateien) am Anfang stehen. Für diesen Abschnitt ist im Quellcode der Befehl `\label{sec:Cslatex}` zu finden. Mit dem Befehl `\ref{sec:Cslatex}` wird die Abschnittsnummer dieses Abschnitts ausgegeben: 2. Für jede Überschriftenebene, sowie Formeln, Aufzählungspunkte, Definitionen usw. gibt es eigene Zähler, die entsprechen mit `label` gesetzt und mit `ref` wieder abgerufen werden können (s. Quelltext). Es zählt jeweils die unterste Stufe. Daher ist es sinnvoll, den `label` Befehl direkt nach einem neuen Block einleitenden Befehl zu verwenden.

Absätze werden in LaTeX durch eine oder mehrere Leerzeilen erreicht. Alle überflüssigen Leerzeichen und -zeilen werden ignoriert. Befehle wie `\\ (newline)`, `\clearpage` oder `\newpage` sind nur in Sonderfällen zu verwenden und in dieser Arbeit gar nicht.

## 2.2 Textformatierung

LaTeX bietet einige Möglichkeiten, die Schriftart zu verändern. Diese sollten sparsam eingesetzt werden, um die Lesbarkeit des Textes nicht zu beeinträchtigen. *Wichtige* oder *neue* Begriffe werden betont (engl. *emphasize*) mit dem Befehl `\emph{...}`. Die Schreibmaschinenenschrift `\texttt{...}` wird für Befehle o.ä. verwendet. Bei Befehlen, die LaTeX-Steuerzeichen enthalten empfiehlt sich der Befehl `\verb+...+` (vgl. Kommentar im Quellcode in Abschnitt 1.3).

## 2.3 Aufzählungen

LaTeX stellt im wesentlichen zwei Arten von Aufzählungen zur Verfügung: nummerierte und unnummerierte. Es handelt sich dabei um so genannte Umgebungen (engl. *environments*), die immer von `\begin{env}` und `\end{env}` umschlossen werden. Dabei ist `env` durch den Namen der jeweiligen Umgebung zu ersetzen. Wer Emacs verwendet kann mit `Ctrl-c Ctrl-e` leicht Environments erstellen. Eine spezifische Aufzählungsumgebung für Algorithmen wird in Abschnitt 3.2 erläutert. Es folgen zwei Beispiele zu den beiden Aufzählungsarten.

### Itemize-Umgebung

Dient der nicht nummerierten Aufzählung. Beispiel:

- Item1
- Item2
- Item3

## Enumerate-Umgebung

Dient der nummerierten Aufzählung. Beispiel:

1. Item1
2. Item2
3. Item3
  - a) Item31
  - b) Item32
  - c) Item33

Der `label`-Befehl funktioniert auch innerhalb von Aufzählungen (siehe Quellcode): Punkt 3 ist in mehrere Unterpunkte aufgeteilt.

## 2.4 Formel-Darstellung

LaTeX bietet zahlreiche komfortable Möglichkeiten, um Formeln darzustellen. Da aber die Darstellung sehr komplexer Formeln ermöglicht wird, existieren entsprechend viele Befehle zu ihrer Beschreibung. Zunächst muss man zwischen mathematischen Ausdrücken im Text, wie  $\alpha \in \mathbb{R}$ , und abgesetzten Formeln oder Ausdrücken entscheiden. Es folgen einige Beispiele aus anderen Texten, die unter Zuhilfenahme des Quellcodes den Umgang mit Formeln verdeutlichen sollten. Hier zunächst zwei einfache Formeln.

$$h_i(z) = \frac{1}{\sqrt{2\pi\sigma_i}} \cdot e^{-\left(\frac{z^2}{2\sigma_i^2}\right)} \quad (1)$$

$$o_j = f_{\text{sig}} \left( \sum_{i=1}^N c_{ij} h(|x - w_i|) \right) \quad (2)$$

Mit `\intertext{...}` lässt sich Text innerhalb ausgerichteter Formelblöcke einfügen.

$$\begin{aligned} w_i(t+1) &= w_i(t) - \alpha(t) [x(t) - w_i(t)] \\ w_j(t+1) &= w_j(t) + \alpha(t) [x(t) - w_j(t)] \end{aligned} \quad (3)$$

where  $w_i$  must be of a different class than  $x$  and  $w_j$  of the same class. If the two closest prototypes are of the same class (training of type (2) the following modification scheme is applicable (no "window" constraint):

$$w_k(t+1) = w_k(t) + \epsilon\alpha(t) [x(t) - w_k(t)] \quad (4)$$

$$\begin{aligned} \Sigma &= E\{(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T\} \\ &= \begin{bmatrix} E\{(x_1 - m_1)(x_1 - m_1)\} & \cdots & E\{(x_1 - m_1)(x_n - m_n)\} \\ \vdots & & \vdots \\ E\{(x_n - m_n)(x_1 - m_1)\} & \cdots & E\{(x_n - m_n)(x_n - m_n)\} \end{bmatrix} \\ &= \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & & \vdots \\ c_{n1} & \cdots & c_{nn} \end{bmatrix} \end{aligned} \quad (5)$$

Auch Formeln können natürlich Label besitzen, z.B. Die Modifikation der Gewichte beim LVQ3 Training (4) versteht ohne Erklärung keiner.

### 3 Der mmiSeminar-Style ---

Der zunächst von Peter Dörfler für die Seminare am Lehrstuhl für Technische Informatik erstellte Style wird mit dem Wintersemester 06/07 auch für die Seminare des Instituts für Mensch-Maschine-Interaktion eingesetzt. Der Style bietet einige Makros, die das Erstellen von LaTeX Dokumenten vereinfachen. Durch Ändern einiger Verhaltensweisen von LaTeX wird es möglich, den Seminarband als ein TeX-Dokument zu behandeln. Weiterhin sind darin einheitliche Formatierungen und das Layout für den Text definiert.

#### 3.1 Die Titelseite

Die Titelseite lässt sich recht schnell erstellen (siehe Quellcode). Es müssen nur die folgenden Werte (eigentlich Makros) gesetzt werden:

- Der Autor des Textes
- Der Titel
- Die e-mail des Autors
- Zusammenfassung
- Schlüsselworte für den Inhalt

Der Befehl `\makeArticleTitle` erstellt dann die Titelseite.

## 3.2 Listen-Umgebungen im MMI-Style

### Aufzählungen

Zwei Arten von Aufzählungen sind hinzugekommen:

1. `narrowenum`
2. `narrowitems`

Die erste der beiden hat direkt hier Anwendung gefunden. Der andere Aufzählungstyp ist in Abschnitt 3.1 zu sehen. Diese Umgebungen sind für Fälle gedacht, in denen die einzelnen Punkte sehr kurz sind. Die normalen Abstände wirken einfach zu groß, wenn pro Punkt nur zwei, drei Worte stehen.

### Algorithmus-Umgebung

Die `algorithm`-Umgebung dient dazu Algorithmen darzustellen. Sie funktioniert im Prinzip wie andere Listenumgebungen, hat aber ein etwas anderes Layout. Zusätzlich sind alle weiteren Stufen nach der ersten in die `subalgorithm`-Umgebung zu setzen. Üblicherweise wird ein Algorithmus in eine *floating*-Umgebung gesetzt. Das heisst, dass er nicht unbedingt an der Stelle erscheint, an der der Algorithmus im Text steht, sondern eventuell in den nächsten freien Platz geschoben wird.

---

```

1. begin begin begin begin begin begin begin begin begin
   begin begin begin begin begin begin begin begin begin

2.  a) Item21 Item21 Item21 Item21 Item21 Item21 Item21
   b) Item22 Item22 Item22 Item22 Item22 Item22 Item22

3. end

```

---

Abbildung 1: Ein Beispiel-Algorithmus

Die `figure`-Umgebung (siehe Quellcode) ist für *floating* Bilder, mit der Position als Parameter (`here`, `top`, `bottom`). Der Befehl `caption` legt die Bildunterschrift fest. Diese Umgebung gehört zu Standard-Latex.

## 3.3 Bilder und Tabellen

Sowohl für Tabellen als auch für Abbildungen gibt es im `mmiSeminar`-Style *floating*-Umgebungen, die einfacher in der Anwendung sind als die Standardumgebungen.

## Abbildungen

Es gibt fünf verschiedene Arten von Abbildungen für unterschiedliche Zwecke. Sie sind in Tabelle 1 aufgelistet. Die ersten beiden Befehle unterscheiden sich dadurch, dass der erste das Bild

Tabelle 1: Die verschiedenen Abbildungs-Befehle

Befehl	Parameter			
	1	2	3	4
lpsfig	Position	Dateiname	Caption	
lfwpsfig	Position	Dateiname	Caption	
lwpsfig	Position	Breite	Dateiname	Caption
lhpsfig	Position	Höhe	Dateiname	Caption
lpspsfig	Position	Skalierung	Dateiname	Caption

in seiner ursprünglichen Größe darstellt, während die fw-Variante (full width) die ganze Textbreite ausnutzt. Die Breite und Höhe werden entweder als feste Werte, z. B. 6cm, oder relativ zur Textbreite, z. B.  $0.5 \backslash \text{linewidth}$  gleich halbe Zeilbreite, angegeben.

Als Bilddateien können nur solche im .eps Format verwendet werden. Die Endung gehört dabei *nicht* zum Dateinamen im Befehl — also ohne .eps. Gleichzeitig wird er als Label für dieses Bild verwendet. Die Eindeutigkeit ist für jeden Artikel automatisch gewährleistet. Hier sind also keine Initialen notwendig. Beziehen kann man sich auf Abbildungen einfach mit dem Befehl `fref{Dateiname}`. Die Erstellung von Bildern wurde in Abschnitt 1.3 besprochen.

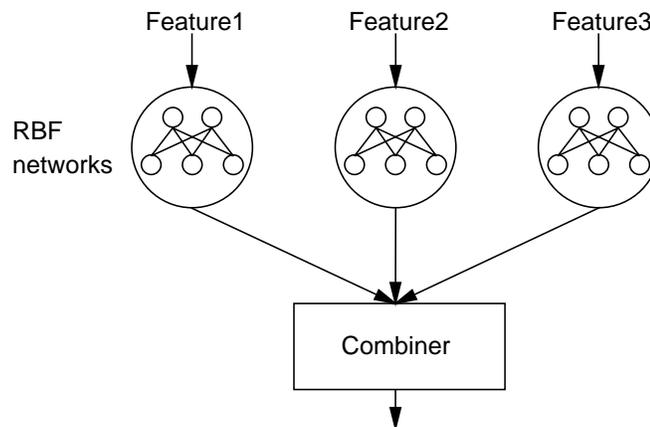


Abbildung 2: Beispielhafte Abbildung

Auch in Abbildungen sollte es natürlich möglich sein, allen Text ohne Probleme zu lesen. Leider sieht man oft das Gegenteil.

## Tabellen

Die Tabellen-Umgebung `vtable` ist so ähnlich aufgebaut wie die `xxxpsfig` Befehle weiter oben. Die Parameter sind in Reihenfolge:

1. Position
2. Label
3. Caption
4. eigentliche Tabelle

Für die Labels gilt das gleich wie schon bei den Abbildungen. Der Referenzierungsbefehl ist `tref`. Die eigentliche Tabelle ist am besten an einem Beispiel, Tabelle 1, erklärt. Direkt nach dem Beginn der `tabular`-Umgebung folgt ein Format-String, der die Ausrichtung des Textes in den einzelnen Spalten festlegt. Die Optionen `l`, `r` und `c` stehen für links, rechts und mittig ausgerichtet. Mit `p{Länge}` kann man eine Spalte fester Breite deklarieren. Der senkrechte Strich erzeugt einen ebensolchen. Vom Layout gefälliger und auch lesbarer ist die Verwendung nur nötiger Linien. Die im Beispiel wäre z. B. auch nicht nötig gewesen. Mit `multiline{spalten}{ausrichtung}{text}` kann man einen Text über mehrere Spalten ziehen und ausrichten wie angegeben. Alles weitere steht in der angegebenen Literatur.

## A Testing the Appendix

---

Der Anhang wird durch den Befehl `appendixOfArticle` eingeleitet. Auf keinen Fall ist der Befehl `appendix` zu verwenden.

Im Anhang stehen solche Dinge, die nicht in den sonstigen Text passen. Lange Beweise und zusätzliche Beispiele oder Abbildungen gehören dazu. Im Allgemeinen sollte ein Anhang möglichst ganz vermieden werden.

## Literaturverzeichnis

---

- [Ame96] American Mathematical Society. *AMS- $\TeX$  Version 1.2 User's Guide*, November 1996.
- [GMS94] Michael Goossens, Frank Mittelbach, and Alexander Samarin. *The  $\TeX$  Companion*. Addison-Wesley, Reading, Massachusetts, USA, 1994. ISBN 0-201-54199-8.
- [Knu84] Donald E. Knuth. *The  $\TeX$ book*, Volume A of *Computers and Typesetting*. Addison-Wesley, 1984.
- [Kop00] Helmut Kopka.  *$\TeX$  Einführung*. Addison-Wesley, Reading, Massachusetts, USA, 3. edition, 2000. ISBN 3-827-31557-3.
- [Lam94] Leslie Lamport.  *$\TeX$ : A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, USA, 2. edition, 1994. ISBN 0-201-52984-1.

[OPHS00] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. *The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>*, Version 3.16, September 2000.