Writing Advice

Table of Contents

- Size
- Contents
  - Chapter Structure Example
- Language
- Advice on Theses Written in English
- Use a Spell Checker
- Emphases
  - Foreign Language Passages
  - Symbols, Physical Units, etc.
  - Programs, Function Names, etc.
- References
- Acronyms
- Figures and Listings
- Cross References
- General Syntax
- Style
- Appendix

Size

Folgendes sind grobe Richtwerte für den Umfang der Arbeit, ohne Titelseite, Inhalts-/Abbildungsverzeichnis, leere Seiten, Bibliographie, Anhänge, etc.

- Studien-/Bachelorarbeit: 30-50 Seiten.
- Diplom-/Masterarbeit: 70-120 Seiten.

Contents

- *Ziel der Arbeit* wirklich auf den Punkt bringen (ca. 1/2 bis 3/4 Seite).
- Neue Begriffe (die einem Informatiker nicht allgemein bekannt sind) an geeigneter Stelle einführen, bevor man sie verwendet (z.B. Definitionen).
- Jedes Kapitel sollte eine Einleitung haben.
- Überleitungen von jedem Teil in den anderen, eine sinnvolle Reihenfolge ist notwendig und erleichtert dies.
- Schwammige Begriffe und Füllwörter vermeiden bzw. weglassen.
  - Füllsätze ebenfalls vermeiden.
- Einheitliche/konsistente Begriffe verwenden, bei neu eingeführten Begriffen auf Synonyme verzichten.
- Keine Umgangssprache verwenden, möglichst auf Fachbegriffe zurückgreifen.
- Von Strukturierungselementen wie \itemize, \enumerate, \description sinnvoll Gebrauch machen.
- Grafiken zur Veranschaulichung nutzen (z.B. Aktivitätsdiagramme, Klassendiagramme, Tabellen).
  - Diese können auch für evtl. Präsentationen wiederverwendet werden.
- Beim Erklären von Zusammenhängen zunächst abstrahieren und erst später konkret werden.

Chapter Structure Example

1. **Abstract**: Beschreibt den Inhalt der Arbeit abstrakt und kompakt, soll einem Leser eine Entscheidungshilfe dabei sein, ob die Arbeit für ihn lesenswert ist oder nicht.
2. **Einleitung**: In das Thema einführen, Ziele der Arbeit beschreiben, Ausblick/Kapitelübersicht geben.
3. **Verwandte Arbeiten**: Arbeiten mit ähnlichen/angrenzenden Themen in Bezug zur Arbeit setzen.
4. **Verwendete Technologien**: Optional, Techniken und Technologien soweit für das Verständnis nötig beschreiben.
5. **Ideen**: Vom Ist-Zustand ausgehend die Probleme erörtern, Lösungsansätze auch mit Verweis auf verwandte Arbeiten diskutieren und den eigenen Lösungsweg stichhaltig begründen, hier zunächst die nötige Abstraktion wahren und erst im Verlauf konkreter werden, Implementierungsdetails sind hier irrelevant.
6. **Implementierung**: Mit Verweis auf den eigenen Lösungsweg, soll hier die Implementierung zunächst mit der Struktur beginnend (Begründung!) und an sinnvollen Stellen auch mit konkretem Anwendungscode/Beispielen beschrieben werden.
7. **Evaluation**: Vergleich der Implementierung oder des Lösungsweges mit anderen Ansätzen, Experimente, Auswertung, Bewertung.
8. **Zusammenfassung, Fazit, Ausblick**: Fasst die Probleme und vor allem die aufgezeigten Lösungen/Ergebnisse zusammen und beschreibt als Ausblick auch Erweiterungsmöglichkeiten, Schließt mit einem knappen und für die Arbeit aussagekräftigen Fazit.

Language

Sie können die Arbeit auf Deutsch oder Englisch verfassen. Falls Sie die Arbeit nicht in Ihrer Muttersprache schreiben, wäre es hilfreich, wenn ein muttersprachlicher (oder ähnlich qualifizierter) Korrekturleser Ihre Arbeit auf sprachliche Richtigkeit korrigieren könnte, bevor Sie die Arbeit Ihrem Betreuer geben.

Software sollte generell auf Englisch geschrieben werden - d.h., Kommentare, Variablen-/Klassen-/Paketnamen etc. sollten englisch sein.

Advice on Theses Written in English

- Lesen und beherzigen Sie Strunk & White (siehe unten).
- Lesen und beherzigen Sie Strunk & White (siehe unten).
- Lesen und beherzigen Sie Strunk & White (siehe unten). Außerdem:
- Sätze sollten nicht mit „But“ oder „And“ anfangen. Stattdessen: „However, ...“ bzw. „Furthermore, ...“.

  Schlecht:

  > *„But this is of exponential complexity. And there is no known alternative.“*

  Gut:

  > *„However, this is of exponential complexity. Furthermore, there is no known alternative.“*

- Die Verwendung des Wortes „like“ im Sinne von „such as“ ist umgangssprachlich und sollte vermieden werden. In noch stärkerem Maße gilt dies für die Bedeutung „as if“.

  Schlecht:

  > *„I, like you, like fast algorithms, like Tarjan's algorithm, ... He uses a sorting algorithm of quadratic complexity, like there is no alternative.“*

  Gut:

  > *„I, like you, like fast algorithms, such as Tarjan's algorithm, ... He uses a sorting algorithm of quadratic complexity, as if there were no alternative.“*

- Mit „which“ wird eine Ergänzung gekennzeichnet, welche in Kommata eingeschlossen wird und deren Streichung nicht sinnentstellend wirkt. Mit „that“ wird eine nähere Identifizierung gekennzeichnet, welche nicht gestrichen werden kann. Und: vor „that“ steht kein Komma!

  Schlecht:

  > *„I told him, that the quicksort algorithm that is very fast is used by the program, which I installed yesterday.“*

  Gut:

  > *„I told him that the quicksort algorithm, which is very fast, is used by the program that I installed yesterday.“*

- Im allgemeinen werden die Wörter „section“, „figure“ etc. klein geschrieben, falls sie nicht am Satzanfang stehen. Falls jedoch eine spezifische Einheit gemeint und durch eine Nummerierung kenntlich gemacht ist, werden diese Wörter groß geschrieben.

  Schlecht:

  > *„The following Section illustrates .... In section 2.3, we describe .... This Chapter includes too many Sections.“*

  Gut:

  > *„The following section illustrates .... In Section 2.3, we describe .... This chapter includes too many sections.“*

Use a Spell Checker

Lassen Sie von Zeit zu Zeit eine Rechtschreibprüfung über ihren Text laufen; dies ist auch für .tex-Dateien möglich. Insbesondere sollte dies geschehen bevor Sie Ihre Arbeit jemand anderem zum Lesen/Korrigieren/Bewerten geben. Ein möglicher Spellchecker ist aspell.

Texlipse unterstützt auch on-the-fly spell checking.

In Emacs (oder Xemacs - generell wird hier nicht zwischen diesen Editoren unterschieden) wird die Rechtschreibprüfung eines Buffers, welcher eine .tex-Datei enthalten kann, mit dem Befehl ispell-buffer gestartet. Der zu verwendende Spellchecker wird durch die Variable ispell-program-name definiert (Default: aspell). Das zu verwendende Wörterbuch wird durch die Variable ispell-dictionary definiert (z.B. deutsch8 oder american), welche durch den Befehl ispell-change-dictionary gesetzt werden kann.

Emacs kann das zu verwendende Wörterbuch auch automatisch für eine .tex-Datei festlegen, wenn die Datei die Variable ispell-dictionary über einen *Local Variables*-Kommentar definiert, welcher vom Emacs beim Einlesen einer Datei ausgewertet wird. Ein Beispiel für einen solchen Kommentar:

```
% Local Variables:
% mode: latex
% compile-command: "make cases2005.pdfv"
% ispell-dictionary: "american"
% End:
```

- Gründliches manuelles Lesen bleibt einem aufgrund Grammatik oder Zeichensetzungsfehlern nicht erspart.
- Möglichst auch einen zweiten, unabhängigen Korrekturleser heranziehen.
- Bei englischer Sprache ist möglichst ein *native speaker* als Korrekturleser heranzuziehen.

Emphases

Einzelne Textpassagen können *ausgezeichnet* werden. Grundsätzlich sollten Auszeichnungen in folgenden Gegebenheiten eingesetzt werden:

- Einführung/Definition neuer Begriffe - wie zum Beispiel „auszeichnen“ (und damit auch „Auszeichnung“) in obigem Absatz.
- Zitate.
- Teilüberschriften, evt. in den Fließtext integriert.
- Titel von Büchern, Zeitschriften *etc*.
- Beispiele, wenn diese - wie hier - in Form von (wenn auch fiktiven) Zitaten erscheinen.
- Hervorhebung von Kernpunkten.
- Fremdsprachige Textfragmente (siehe unten).
- Symbole, physikalische Größen usw. (siehe unten).
- Programme, Funktionsname usw. (siehe unten).

Man sollte sich beim Gebrauch von Auszeichnungen bewusst sein, dass diese ein besonderer Hinweis für den Leser sind und ihn zu besonderer Aufmerksamkeit veranlassen sollen. Sie bewirken stets eine - wenn auch minimale - Unterbrechung des Leseflusses. Auszeichnungen sollten systematisch und im Zweifelsfall eher sparsam eingesetzt werden. Zu *viele*Auszeichnungen *verhindern* das *flüssige Lesen* (und können auch etwas *akade misch* wirken). Es ist dann *schwierig* für den Leser zu *erkennen*, was *denn nun wirklich wichtig ist - insbesondere bei geschachtelten Auszeichnungen, möglicherweise mehrfach* ***überlagert*** !!!!

Bei Auszeichnungen sollte der *Scope* eingehalten werden. Wenn z.B. Begriffe und sowie deren Abkürzungen eingeführt werden, sollten diese einzeln ausgezeichnet werden - Klammerungen selbst sollten nicht ausgezeichnet werden.

Schlecht:

> *„Die Message Sequence Charts (MSCs) sind ein Formalismus ...“*

Gut:

> *„Die Message Sequence Charts (*MSCs*) sind ein Formalismus ...“*

In LaTeX werden Auszeichnungen mit dem Befehl \emph vorgenommen, welche dann normalerweise *kursiv* erscheinen. *In einem kursiv gesetzten Kontext - wie in den Beispielen hier - sind Auszeichnungen jedoch* nicht-kursiv*, um diese vom Kontext abzuheben.*

Foreign Language Passages

Zunächst ein Beispiel für ein englischsprachiges Dokument:

Schlecht:

> *„As Schmidt et al. have documented, this agreement today is a de-facto standard for layout etc.“*

Gut:

> *„As Schmidt* et al*. have documented, this agreement today is a* de-facto *standard for layout* etc*.“*

Bei deutschsprachigen Dokumenten gilt: es sind grundsätzlich Begriffe ausgenommen, welche im Duden stehen (wie zum Beispiel „Layout“). Weiterhin wird bei Auszeichnungen die Groß-/Kleinschreibung der Fremdsprache übernommen.

Schlecht:

> *„Wie Schmidt et al. dokumentiert haben, ist dieses Agreement heute de-facto ein Standard in layout etc.“*

Gut:

> *„Wie Schmidt et al. dokumentiert haben, ist diese Vereinbarung heute de-facto ein Standard in Layout etc.“*

Eine spezielle Problematik bei deutschsprachigen Dokumenten in der Informatik ist, dass hier viele Begriffe aus dem Englischen kommen, und als Teil des deutschen Informatik-Fachvokabulars angesehen werden können, auch wenn diese noch nicht im Duden erscheinen. Diese Begriffe sollten dann nicht ausgezeichnet werden, damit die Lesbarkeit des Dokuments nicht durch übermäßige Auszeichnungen leidet. Im Zweifelsfall gilt: Wenn es für einen englischen Begriff keinen äquivalenten deutschen Begriff gibt, oder der deutsche Begriff in der Praxis seltener eingesetzt wird als der englische Begriff, sollte der englische Begriff als Teil des deutschen Fachvokabulars angesehen werden und auf eine Auszeichnung verzichtet werden. Dies betrifft auch Begriffe, welche nicht wirklich Fachbegriffe sind (wie z.B. „Feature“). Ebenso sollten Eigennamen von Formalismen, Programmiersprachen, Werkzeugen usw. nicht ausgezeichnet werden. In jedem Fall sollten Sie sich konsistent für oder gegen die Auszeichnung eines Begriffs im gesamten Dokument entscheiden; einzige Ausnahme ist die einmalige Auszeichnung eines Begriffes bei einer Begriffsdefinition. Zusammenfassend gilt: es ist oft im Ermessen des Autors, ob eine Auszeichnung verwendet werden soll oder nicht; im Zweifelsfall gilt das Gebot der Sparsamkeit, insbesondere, falls Sie einen einzelnen Begriff sehr häufig verwenden.

Schlecht:

> *„Die Synthese von Statecharts nach Java .... Der Kellerspeicher vom Übersetzer ist hierfür standardmäßig zu klein. Diese software hat viele features und noch mehr bugs. [Aber:] Die Mental Map des Betrachters ....“*

Gut:

> *„Die Synthese von Statecharts nach Java .... Der Stack des Compilers ist hierfür standardmäßig zu klein. Diese Software hat viele Features und noch mehr Bugs. [Aber:] Die mental map des Betrachters ....“*

Nach diesem Plädoyer für die sparsame Verwendung von Auszeichnungen bei originär englischen Ausdrücken noch ein Plädoyer für sparsame Verwendung englischer Ausdrücke, wenn es hierfür gängige und präzise Äquivalente im Deutschen gibt:

Schlecht:

> *„Das Agreement war, die Action Items und die Schedule des nächsten Meetings in die Minutes zu pasten, und sich zum Matchen der Deadlines zu committen.“*

Gut:

> *„Die Vereinbarung war, die Aktionspunkte und die Zeitplanung der nächsten Besprechung in das Protokoll zu übernehmen, und sich zur Einhaltung der Termine zu verpflichten.“*

In der Informatik ist dies oft eine schwierige Balance zwischen zu viel und zu wenig deutsch, welche letztlich eine Sache des persönlichen Geschmacks ist. So scheint „Übersetzer“ ebenso akzeptabel wie „Compiler“, „Betriebssystem“ mehr verbreitet als „Operating System“, jedoch wirkt „Kellerspeicher“ eher verschroben im Vergleich zu „Stack“.

Symbols, Physical Units, etc.

Neben den bereits erwähnten Fällen, in welchen Textbestandteile ausgezeichnet werden sollten, sollten auch Mathematische Symbole, Mengenbezeichner, physikalische Größen etc. ausgezeichnet werden. In LaTeX kann man hierfür, wie bereits erwähnt, den Befehl \emph{...} verwenden - oder alternativ den Begriff mit „$...$“ auszeichnen. Letzterer Mechanismus ist jedoch mit etwas Vorsicht zu verwenden, falls die Begriffe mehr als ein Zeichen lang sind, da dieser Mechanismus den Begriff als mathematische Formel interpretiert und spezielle Richtlinien für dessen Layout verwendet. Insbesondere wird der Buchstabe f als eigenständiger Funktionsname interpretiert und ein entsprechender Zwischenraum produziert, was generell nicht erwünscht ist.

Schlecht:

> *„Sei $Koffer$ die Menge der Koffer ...“*

Gut:

> *„Sei \emph{Koffer} die Menge der Koffer ...“*

Programs, Function Names, etc.

Namen von Programmen sowie von Funktionsnamen, Variablennamen etc. sollten ebenfalls ausgezeichnet werden. Jedoch sollte man hierfür eine andere Auszeichnung verwenden, nämlich diejenige, welche auch in Programmlistings verwendet wird. Typischerweise ist dies typewriter (erzeugt mit \texttt{...}) oder SansSerif (erzeugt mit \textsf{...}). Ein Beispiel hierfür sind die in diesem Text erwähnten LaTeX-Befehle.

References

Verwenden Sie Referenzierungen im Ihrer Arbeit so, dass sie kein grammatikalischer Bestandteil des Satzes sind - der Text sollte auch nach Eliminieren aller geklammerten Bestandteile, einschließlich der in eckigen Klammern befindlichen Referenzen, noch grammatikalisch korrekt sein. Üblicherweise erreicht man dies durch die Nennung der Autoren eines Werkes, gefolgt von der Referenz. Im Text sollten Autoren typischerweise ohne Vornamen genannt werden (jedoch sollten Autorenvornamen bei den Referenzen selbst vollständig genannt werden, siehe unten). Bei mehreren Autoren sollten entweder alle genannt werden, oder es sollte die Existenz weiterer Autoren durch „et al." kenntlich gemacht werden.

The citation comes at the end of the sentence, yet it appears *inside* the period (because it is part of the sentence). If the citation is preceded by a quotation from the cited work, then the citation is placed *outside* the quotation marks (because it isn't part of the quotation) [LitWeb]

Schlecht:

> *„Wie in [42] beschrieben, ..."*

Gut:

> *„Wie von Harel* et al*. beschrieben, ... [42]"*

The citation comes at the end of the sentence, yet it appears *inside* the period (because it is part of the sentence). If the citation follows a quotation from the cited work, then the citation is placed *outside* the quotation marks (because it isn't part of the quotation).

Vor Referenzen steht eine Leerstelle, hier sollte aber nicht umgebrochen werden (in Latex: ~).

Schlecht:

> *„Wie von Harel[42] beschrieben, ..."*

Gut:

> *„Wie von Harel [42] beschrieben, ..."*

Die Referenzen selbst sollten vollständig sein - also mehr sein als eine Sammlung von Suchbegriffen für google. Grundsätzlich sollten Angaben so vollständig wie möglich sein. So gehören z.B. zu einer Artikelreferenz nicht nur Autoren und Titel des Artikels, sondern auch bei Jahr/Monat des Erscheinens, bei Journalartikeln auch Name des Journals und gegebenenfalls Nummer und Herausgeber, sowie bei Tagungs-/Workshopartikeln der Name und Ort der Tagung. Vermeiden Sie Abkürzungen. Tagungsnamen sollten sowohl in ausgeschriebener Form als auch mit ihrem Kürzel genannt werden.

Schlecht:

> *„A. Mooij and N. Goga and J. Romijn. Non-local Choice and Beyond: Intricacies of MSC Choice Nodes. FASE'05, pages 273-288, 2005." „A. Mooij and N. Goga and J. Romijn. Non-local Choice and Beyond: Intricacies of MSC Choice Nodes. In Fundamental Approaches to Software Engineering, pages 273-288, April 2005."*

Gut:

> *„Arjan J. Mooij and Nicolae Goga and Judi Romijn. Non-local Choice and Beyond: Intricacies of MSC Choice Nodes. In Fundamental Approaches to Software Engineering (FASE '05), pages 273-288, Edinburgh, Scotland, April 2005."*

Achten Sie auf die **Konsistenz** der Einträge. So sollten z.B. alle Referenzen für ein Journal dieses Journal gleich referenzieren. Autoren sollten so referenziert werden, wie sie in der referenzierten Arbeit erscheinen; so sollten Vornamen nicht abgekürzt werden, wenn sie in der Arbeit selbst ausgeschrieben sind, ebenso sollten angegebene Mittelinitialen übernommen werden. Bei der Verwendung von URLs sollten Sie darauf achten, dass diese nicht über den Satzspiegel hinausragen; das Paket breakurl sorgt für den entsprechenden Zeilenumbruch von URLs.

Schlecht:

> *[1] Alur, Etessami, and Yannakakis. Inference of message sequence charts.* IEEETSE: IEEE Transactions on Software Engineering*, 29, 2003.*

> *[2] Alan W. Biermann and Ramachandran Krishnaswamy. Constructing programs from example computations. IEEE Transactions on Software Engineering, 2(3):141-153, September 1976.*

Gut:

> *[1] Rajeev Alur, Kousha Etessami, and Mihalis Yannakakis. Inference of message sequence charts.* IEEE Transactions on Software Engineering*, 29:304NAK313, 2003.*

> *[2] Alan W. Biermann and Ramachandran Krishnaswamy. Constructing programs from example computations.* IEEE Transactions on Software Engineering*, 2(3):141-153, September 1976.*

Die Bibliographie sollte nach Nachnamen des Erstautors sortiert sein. Bei der (stark empfohlenen) Verwendung von BibTeX sollten Sie für jede Referenz den richtigen, möglichst spezifischen Referenztyp verwenden (z.B. @InProceedings statt @Misc). Die für den jeweiligen Referenztyp vorgesehenen Pflichtfelder geben bereits einen ersten Hinweis für die Mindestinformationen; die weiteren optionalen Eintragsfelder sollten aber auch so weit wie möglich verwendet werden. Falls bibtex bei Erzeugung der Referenzen einen Fehler meldet, z.B. aufgrund eines fehlenden Pflichtfeldes, sollte dieses auf alle Fälle korrigiert werden. Für Ihre Arbeitserleichterung sollten Sie ein konsistentes Schema für die Indizierung verwenden, so dass Sie den Index möglichst aus Autoren & Erscheinungsjahr erschließen können, und Referenzierungen in Ihren Text einfügen können, ohne den Index in der BibTeX-Datei nachsehen zu müssen. Die Verwendung eines bestimmten Schemas ist insbesondere dann erforderlich, falls Sie eine gemeinsam genutzte BibTeX-Datei bearbeiten. So gibt z.B. die am Lehrstuhl genutzte cau-rt.bib-Datei ein Schema vor, welches am Anfang dieser Datei beschrieben ist.

- Literaturangaben möglichst als integrierten Teil des Satzes (nicht außerhalb des Satzes anbringen).
- Literaturangaben keinesfalls hinter das Satzendzeichen (z.B. nach dem Punkt).
- Auch in Abbildungen, die nicht selbst/selbständig erstellt wurden darf die Literaturangabe nicht fehlen (z.B. *(nach Schmidt [3])*).
- Für URLs, Verweise auf Webseiten (z.B. www.eclipse.org) eignen sich am besten Fußnoten.
- Bib-Einträge kontrollieren (Titel, Autoren (durch and getrennt, nicht kommasepariert!), Datum, Ort!).
- Bib-Einträge: Wörter, die groß geschrieben werden sollten in geschweifte Klammern setzen.
- Bib-Einträge auf Englisch oder Deutsch (je nach Arbeit!).
- Literaturangaben nicht als Nomen in den Satz einbetten.
  Merkregel: Der Satz sollte sich auch ohne die Literaturangabe sinnvoll lesen lassen.
- Nur Primärliteratur oder (bewährte) Fachbücher: Z.B. keine Foliensätze/Lectures referenzieren.

Acronyms

- Das Akronympaket ist zu verwenden (z.B. \ac{kieler}).
- Indentierung bei Akronymverzeichnis ist einheitlich und linksbündig zu gestalten.
- In Überschriften (gilt auch für Bildunterschriften) explizit die Langform verwenden (\acl). Achtung: Hier darauf achten, daß das Akronym im Text dennoch eingeführt wird!
- Ausnahmeregelung: Allgemein bekannte Abkürzungen wie GUI. Hier die Langform gar nicht verwenden.
- Akronyme konsistent halten.
- Allgemein übliche Akronyme nicht so üblichen Akronymen vorziehen.
- Akronyme niemals trennen.

Figures and Listings

- Größe: Nicht unter 80% der Seitenbreite, möglichst 100% (Textbreite ausnutzen!)
- Ähnliche Abbildungen sollten auch vergleichbar/gleich groß sein.
- Schriftgröße in Abbildungen sollte ca. der Schriftgröße des Textes entsprechen.
- Wenn Textteile der Abbildung im Text zitiert werden, ist auf ein **konsistentes** Schriftbild zu achten, ggf. \sf verwenden.
- Abbildungen immer mit Nummer und sinnvoller Überschrift versehen.
- Überschrift abstrakt und so kurz wie möglich, so lang wie nötig halten (es sollte ohne Lesen des Textes verständlich sein, was die Abbildung zeigt).
- Diagramme sollten i.d.R. eine Legende haben (Bedeutung von Farben/Formen nicht nur im Text erläutern).
- Abbildungen immer in der selben Reihenfolge im Dokument anordnen, in welcher sie im Text referenziert werden.
- Abbildungen möglichst immer in der Nähe des erklärenden Textes anordnen.
- Jede Abbildung muss auch beschrieben werden.
- Autolayout einsetzen für entsprechende Diagrammtypen (z.B. Klassendiagramm).
- Einheitliche Schriftgröße für alle Listings verwenden.
- Akronyme in Bildunterschriften s.o.
- Listings in Figures vermeiden (dies kann zu speziellen Latex-Problemen führen)
- For listings, use a \sf (sans serifs) or \tt (typewriter/courier) font. For an \sf font, use flexible column spacing (\lstset{columns=flexible} or \lstset {columns=fullflexible})
- And perhaps the most common issue: when referring to statements, variable names, etc. from a listing, use a formatting that is **consistent** with the listing. Eg, if you use \sf in your listings, write "then \textsf{x} is set to 17". (Not: "then x is set to 17" or "then \textsf{x} is set to \textsf{17}".) The same applies to graphical languages. Eg, if a Statechart transition has a transition label "x / y" and is formatted in \sf, then references to x and y should also be formatted set in \sf.

Cross References

- Bezeichnung sollte korrekt sein: z.B. Kapitel 2, Unterkapitel 2.1, Abschnitt 2.3.4, Absatz
  - Nicht: z.B. Kapitel 2.1
- Rückwärtsverweise sind ok, Vorwärtsverweise sind zu vermeiden.
- Zu viele Verweise stören den Lesefluss, wichtige Dinge sollten an geeigneter Stelle auch (evtl. kürzer) wiederholt werden.
- Abbildungen/Tabellen immer konkret benennen.

General Syntax

- Doppelpunkt als Teil des Satzes auffassen; nach *:* sollte kein neuer Absatz beginnen.
- Hervorheben von Wörtern (\emph{}) möglichst selten einsetzen (s.o.).
- Fremdsprachenwörter kursiv hervorheben.
- Klammern vermeiden, sie stören den Lesefluss.
- Fußnoten vermeiden, sie stören den Lesefluss.
- Fußnoten für URLs sind ok.
- Möglichst auf Querseiten verzichten, sie stören den Lesefluss.
- Keine Einzelzeilen oder Überschriften vor oder nach einem Seitenumbruch stehen lassen.
- Es sollten keine Zeilen oder Abbildungen/Tabellen über den Rand hinausragen.
- Auf unnötigen Whitespace verzichten.
- Codefont für Funktionsnamen/Prozeduren/... verwenden und auch hier auf Konsistenz achten.
- LaTeX kennt keine Abkürzungs-Punkte. Z.B. Bei "e.g." oder "et al." ein xspace hinter dem letzten Punkt verwenden, da ansonsten ein zu großer Abstand zum nächsten Wort erzeugt wird (wie beim Satzende).

Style

- Füllwörter, Füllphrasen und Füllsätze vermeiden, z.B.
  - due to the fact that
  - in order to
- Zusammenhänge: "It" oder "This" vermeiden, wenn der Zusammenhang nicht wirklich eindeutig klar ist. In der Regel lieber den Bezug noch einmal explizit zum Ausdruck bringen.
- Passiv in Sätzen nach Möglichkeit vermeiden:
  - Schlecht: „The syntax is explained in Chapter 5“
  - Gut: „Chapter 5 explains the syntax“
- Wir/Ich-Sätze vermeiden. Ausnahme: Es geht um den inhaltlichen Kernbeitrag.

Appendix

- Wesentliche Teile, die im Text wichtig sind, nicht im Anhang verstecken.
- Anleitungen (z.B. *How-To*s) gehören in den Anhang.