

Anleitung zur Übertragung von T_EX-Quellcode zu einer herkömmlichen Textverarbeitung und umgekehrt

Gregor Barth

v1.16 vom 25. Februar 2019

1 Vorwort

Diese Studie soll Anwendern nützlich sein, die regelmäßig einen T_EX-Quellcode in ein »herkömmliches Textdokument« (wie Word oder LibreOffice Writer) zu transferieren haben, sei es aus Gründen der Manuskript-Weitergabe oder aus anderem Anlass. Allerdings kann es auch vorkommen, dass man den Text aus einer herkömmlichen Textverarbeitung in einen T_EX-Quellcode umwandeln möchte, um ihn anschließend z. B. mit L^AT_EX zu kompilieren; in diesem Fall gibt Abschnitt 3.1 weitere Hinweise.

Hier soll noch kurz erwähnt sein, dass es verschiedene Programme¹ und Makros gibt, die für genau diese Aufgabe gedacht sind, d. h. die im Quelltext enthaltenen Befehlsstrukturen direkt übersetzen (z. B. `\emph{}`-Anweisung wird kursiviert). Meiner Erfahrung nach funktioniert das je nach Komplexität des ursprünglichen Quellcodes nur eingeschränkt und erfordert in jedem Fall eine manuelle Nacharbeit.

Alternativ kann man einen Parser wie Pandoc verwenden, der zumindest theoretisch die Konvertierung eines Word-Dokuments zu L^AT_EX und umgekehrt ermöglicht (siehe Abschnitt 4). Prinzipiell funktioniert das gut, allerdings habe ich diesen Weg

¹Der Editor TeXMaker bietet im Menüpunkt Werkzeuge|Via TeX4ht exportieren die Möglichkeit, den Quellcode in ein OpenDokument-Format zu überführen. Aus diesem Format ist die Konvertierung z. B. ins Word-eigene Format problemlos.

noch nicht mit wirklich komplexen Dokumenten (wissenschaftliche Veröffentlichung etc.) ausprobiert.

Weiterhin enthält die freie Textverarbeitung Abiword einen Export-Filter zu \LaTeX . Abiword kann dabei das Word- und OpenDocument-Format öffnen. Für LibreOffice gibt es ein Modul, mit dem sich Writer-Text in \TeX -Quellcode überführen lässt; in der Ubuntu-Paketverwaltung heißt es `libreoffice-writer2latex`.

2 Textverarbeitung zu \TeX

Bei der Übertragung des Textes geht jede ursprünglich getroffene Strukturierung, Formatierung und Verlinkung verloren, denn es handelt sich bei \TeX -Quellcode um reinen Text, der erst mit Befehlen ausgestattet werden muss. In der anderen Richtung (\TeX -Quellcode soll in eine herkömmliche Textverarbeitung kopiert werden, siehe Abschnitt 3) wird der Text voller Befehle sein, die nach bestimmten Zeichen abgesucht werden müssen.

Als Ursprungstext sollte immer der in der originalen Textverarbeitung enthaltene Textinhalt kopiert werden² und nicht etwa derjenige aus einem fertig erzeugten PDF – diese enthalten nämlich ggf. bereits integrierte Wortumbrüche am Zeilenende, sodass die Bindestriche ebenfalls kopiert werden (und dann aus dem Quellcode wieder mühsam entfernt werden müssen). Achtung außerdem beim Kopieren ganzer PDF-Inhalte, da auch hier die Seitenzahlen übertragen werden und anschließend mitten im Quelltext stehen!

Die nachfolgende Auflistung gibt die Reihenfolge vor, wonach der hereinkopierte Text auf eine \TeX -Quellcode-Syntax vorbereitet wird, damit er auch kompiliert werden kann.

• Struktur

1. Einfügen der Dokument-Präambel sowie der notwendigen Umgebung `\begin{document}` und `\end{document}`.
2. Strukturierung des Dokuments, d. h. Festlegung der Gliederungseinheiten (z. B. `section`, `subsection` usw.). Hierbei ist darauf zu achten, dass

²Und auch hier müssen vorher ggf. die automatischen Wortumbrüche deaktiviert werden!

die evtl. vorhandene ursprüngliche Gliederungsnummerierung entfernt werden muss. Aus 2.3 `Methods` wird `\subsection{Methods}`.

3. Kennzeichnung der Absätze: Nun werden die einzelnen Absätze deklariert. In einer Textverarbeitung wie `Word` oder `Writer` sind neue Absätze mit einem einfachen `Enter` erreicht. Beim Kopieren in den Quellcode liegen zwei nachfolgende Absatz-Blöcke demzufolge in nachstehenden Zeilennummern. In `TeX` werden einzelne Absätze allerdings durch *mindestens eine* leere Zeile gekennzeichnet. Also müssen die einzelnen Absätze im Quellcode von Hand mit einer zusätzlichen Leerzeile voneinander separiert werden.
4. Eventuell im Originaltext vorhandene Listen-Umgebungen oder Aufzählungen in der `TeX`-Syntax neu setzen. Beim Kopieren entsteht evtl. lediglich eine mit Original-Nummerierung versehene Auflistung, die gelöscht und z. B. durch eine `enumerate`- oder `itemize`-Umgebung ersetzt werden muss.
5. Setzen fester Abstandsmarker, d. h. *geschützter Leerzeichen* und *halber Leerzeichen* (beide werden am Zeilenende nicht getrennt umgebrochen). Geschützte Leerzeichen³ kommen z. B. bei Angaben vom Titel oder einer Referenz zum Einsatz und werden mit einer Tilde markiert: `Prof. ~Schmidt`, siehe `Bild~\ref{Karte}`. Halbe Leerzeichen setzt man dagegen zwischen Abkürzungen und Einheiten sowie anderen Gelegenheiten: `z. \, B., 3\, m`. Einer herkömmlichen Textverarbeitung fehlt die Möglichkeit, ein halbes Leerzeichen zu setzen; beim Kopieren wird natürlich ein ganzes Leerzeichen an den betreffenden Stellen übertragen und muss manuell in den Quellcode eingefügt werden. Was die gängigen Abkürzungen angeht, kann man das sehr schnell mit einer Suchen & Ersetzen-Funktion im Editor erledigen.
6. Abgesetzte Formeln ggf. neu in eine Formel-Umgebung einschließen, z. B. in die `equation`-Umgebung.

- **Formatierung**

³Natürlich kann man auch in einer herkömmlichen Textverarbeitung ein geschütztes Leerzeichen an die betreffende Stelle einsetzen, aber dies wird beim Umkopieren in den Quellcode nicht übertragen.

1. Ersetzung von Auszeichnungen (die als unformatierter Text natürlich nicht mit in den \TeX -Quellcode übertragen worden sind!), d. h. Kursivierungen, Fettschrift und anderes. Hierfür muss der ursprüngliche Text (am besten im Textverarbeitungsprogramm) nach bestimmten Formattierungen abgesucht werden. Das betreffende Wort (oder Passage) wird dann mit der Suchmaske im Quellcode angesteuert und dort der \TeX -Befehl für z. B. eine Kursivierung gesetzt. Dieser Arbeitsschritt kann bei umfangreichen Texten durchaus der intensivste von allen sein.
2. Ersetzen von Anführungszeichen. Wie die Tabelle in Abschnitt 2.1 zeigt, ist es nicht zwingend notwendig, die Anführungszeichen in eine \TeX -Syntax zu überführen⁴. Jedoch bietet sich die Nutzung des `enquote`-Pakets an, womit u. a. vermieden wird, dass am Anfang oder Ende einer Anführung das entsprechende Anführungszeichen vergessen wird. Diese Anwendung für die Rücktransformation des Quellcodes für die Verwendung in einer herkömmlichen Textverarbeitung wieder rückgängig zu machen, bedeutet ggf. mehr Arbeit.
3. Ersetzung von Hoch- und Tiefstellungen: Üblicherweise werden hoch- und tiefgestellte Ziffern/Buchstaben in der \TeX -Syntax gesondert deklariert, nämlich mit `\textsuperscript{}` (für Hochstellungen) und für Tiefstellungen: `\textsubscript{}`. Die in Klammern eingefasste Glyphe wird dann hoch- oder tiefgestellt. In einer herkömmlichen Textverarbeitung würde man die Hoch- oder Tiefstellung direkt eingeben, z. B. km^2 . Das funktioniert aber nur für die über die Tastatur zugänglichen Symbole. Für Ziffern beispielsweise kann man eine hochgestellte 2 und 3 direkt eingeben: m^2 , km^3 . Im Prinzip muss man diese Hoch- und Tiefstellungen nicht gesondert in die \TeX -Syntax übertragen, da sie beim Kompilieren (eines `utf8`-konformen Codes) direkt gesetzt werden können. Aus km^3 wird beim Kompilieren also wieder km^3 . Wie man jedoch beim Vergleich mit der Eingabe durch die \TeX -Syntax erkennt (siehe Abschnitt 2.1, 1. Tabellenspalte), kann beim Kompilieren mit \TeX (in Abhängigkeit der benutzten Schriftart) eine speziell für die Hoch- oder Tiefstellung gezeichnete Gly-

⁴Vorraussetzung ist, dass es sich um Unicode-konformen Quelltext handelt.

phe zum Einsatz kommen, die sich von der Glyphen durch direkte Eingabe in Größe und Strichstärke unterscheidet. In diesem Fall bleibt dem Benutzer überlassen, ob ihm die Übersetzung in die \TeX -Syntax die eigentlich typografisch korrekte Darstellung mit \TeX die Arbeit wert ist. — Hinsichtlich Tiefstellungen wird es jedoch unvermeidlich sein, die \TeX -Syntax anzuwenden, beispielsweise um die Ziffern in chemischen Formeln darzustellen: CO_2 , H_2O etc. In diesem Fall bieten sich spezielle Pakete für die schnelle Eingabe chemischer Formeln an.

4. Alle Dezimaltrenner (Kommata) in Formel-Umgebungen müssen in eine geschweifte Klammer eingefasst werden, weil sie sonst (wie ein Komma im Text) falsch spationiert werden. Also statt $\$3,5\$$ heißt es: $\$3\{, \}5\$$
5. Einfügen der Tabellen-Struktur. Tabelleninhalte müssen von Hand in die \TeX -Tabellenstruktur überführt werden. Das ist manchmal gar nicht so einfach, weil man sowohl wissen muss, welches Tabellen-Paket geeignet ist (für automatische Spaltenbreiten, Zeilenumbrüche, mehrseitige Tabellen, Ausrichtung des Inhalts am Dezimaltrenner usw.), als auch, wie man die Spalten- und Zeilenvorgaben nachträglich anpasst, dass die Tabelle einerseits gut aussieht und andererseits möglichst kompakt ist und wenig Raum einnimmt. Hier ist vielleicht folgender Trick eine Hilfe: Kann der ursprüngliche Tabelleninhalt in eine Tabellenkalkulation (Excel, Calc) übertragen werden, lässt sich dieses Tabellendokument mit der Software Gnumeric (einer weiteren Tabellenverarbeitung) öffnen. Gnumeric kennt dabei einen Export-Filter zu \TeX -Tabellen, sodass man sich schon einmal die mühsame Dateneingabe (Spaltentrenner mit $\&$, Zeilenende mit \backslash) spart. Die exportierte Textdatei kann direkt in den Quellcode übernommen, und muss nur noch nachträglich angepasst werden.

• Verlinkung

1. Entfernung evtl. vorhandener automatischer Listen, z. B. eines mit dem Textblock in den Quellcode hineinkopierten Inhaltsverzeichnisses. Der Block wird ersetzt mit $\backslash\text{tableofcontents}$. Gleiches gilt für ein evtl. vorhandenes Abbildungs- und Tabellen-Verzeichnis.

2. Fußnoten: Die als Fußnote gedachten Textpassagen müssen zwingend in eine `\footnote{}`-Klammer eingeschlossen werden. Dabei sagt die Erfahrung, dass die ursprünglichen Fußnoten aus dem Original-Dokument beim Kopieren gar nicht erfasst und in den \TeX -Quellcode übertragen werden! Stattdessen wird nur die eventuelle Fußnoten-Zahl selbst übertragen und steht dann lose mitten im Quellcode: `... Satzende3` (die 3 entspricht der Nummer der ursprünglichen Fußnoten-Zählung, die eigentliche Fußnote fehlt in Quellcode). Der Fußnoten-Inhalt muss also von Hand aus dem Original-Dokument herauskopiert und übertragen werden; für die verwaiste Nummer wird der `\footnote{}`-Befehl gebraucht.
3. Bilder/Tabellen und deren Verlinkung einrichten
 - a) Für jedes Bild bzw. Tabelle ggf. eine Gleitumgebung erstellen, und für diese einen eindeutigen Schlüssel festlegen.
 - b) Im Text dann alle unverlinkten Referenzen (`»siehe Abb. 3«`) korrekt verweisen zu `siehe Abb.~\ref{Schlüssel}`.
4. Ersetzen von Literaturzitaten
 - a) Anlegen einer `bib`-Literaturdatenbank und Festlegung eines eindeutigen Schlüssels für jeden Eintrag.
 - b) Im Quellcode alle Zitate mit dem `bib`-Schlüssel ersetzen; aus `siehe Schmidt (2009)` wird `siehe \textcite{Schmidt2009}`. Auch dieser Arbeitsschritt ist mitunter sehr arbeitsintensiv und lässt sich nur bedingt durch Suchen & Ersetzen automatisieren.

- **\TeX -eigen**

1. Ersetzen von Sonderzeichen. Hierbei soll die in Abschnitt 2.1 dargestellte Tabelle einen Hinweis geben, welche Sonderzeichen zwingend mit ihrem Pendant der \TeX -Notation getauscht werden müssen und welche nicht. Beispielsweise können Anführungszeichen, Auslassungspunkte und Gedankenstriche notfalls als eigene Glyphen verbleiben⁵; aber *geschützte* Steuerzeichen wie die Raute (#), das Prozent-Zeichen (%), das Dollar-Zeichen

⁵Abhängig von der verwendeten Schriftart müssen diese Symbole natürlich im Glyphensatz enthalten sein!

($\$$) und andere müssen zwingend mit der jeweiligen $\text{T}_{\text{E}}\text{X}$ -Schreibweise ausgetauscht werden. Schon allein deshalb, weil es sonst beim Kompilieren Fehlermeldungen hageln würde!

2. Handelt es sich um einen Text über $\text{T}_{\text{E}}\text{X}$ und Abwandlungen (z. B. $\text{E}_{\text{T}}\text{X}$), können die Befehle aus dem `metalogo`-Paket benutzt werden, um statt `LaTeX` zu schreiben: `\LaTeX`. Damit wird das korrekte Logo eingefügt.
3. In deutschen Texten muss an Wortfugen ggf. die automatisch gesetzte Ligatur wieder aufgelöst werden. Das heißt, beim Kompilieren wird automatisch eine `fi`-Ligatur an der Wortfuge zwischen `Schilf`-`insel` gesetzt. Nach offizieller Regelung ist das falsch, die automatische Ligatur muss also unterbunden werden: `Schilf\insel`
4. Wer $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$ zusammen mit der `Libertine`-Schriftart verwendet, muss auf Folgendes achten: Es wird automatisch eine spezielle `tz`-Ligatur gesetzt, die an bestimmten Stellen falsch ist und von Hand wieder aufgelöst werden muss (siehe vorheriger Tipp). Hierfür habe ich eine (sicherlich unvollständige) *Black*- und *Whitelist* erzeugt (Abschnitt 2.2), d. h. eine Liste, mit der man seinen Quellcode durchsuchen und an entsprechenden Stellen die `tz`-Ligatur unterdrücken kann.

2.1 Vergleich direkte Glyphen-Eingabe vs. $\text{T}_{\text{E}}\text{X}$ -Code

Die folgende Tabelle zeigt eine vergleichende Darstellung häufig gebrauchter Glyphen nach der $\text{T}_{\text{E}}\text{X}$ -Schreibweise (obere Zeile) und der direkten Eingabe der betreffenden Glyphe im Quellcode (untere Zeile). Abweichungen in Rot markiert; in diesem Falle sind die rot gefärbten Varianten typografisch falsch.

Dabei bedeuten: 1 Hochstellung, 2 Gedankenstrich, 3–5 Anführungszeichen, 6 Auslassungspunkte, 7 Multiplikationszeichen, 8 Grad Celsius, 9 Minus-Zeichen, 10 Koordinaten, 11 Paragraf, 12 Kreuz.

Tiefstellungen, wie sie beispielsweise bei chemischen Formeln unumgänglich sind (H_2O) lassen sich kaum durch direkte Glypheneingabe im Quellcode bewirken. Auch die Hochstellungen sind zumeist auf die auf der Tastatur erreichbaren Ziffern 2 und 3 beschränkt.

1	2	3	4	5	6	7	8	9	10	11	12
km ²	-	»T«	„W“	“F”	...	×	°C	-	4° 12' 45" N	§	†
km ²	-	»T«	„W“	“F”	...	×	°C	-	4° 12' 45" N	§	†

2.2 Black- und Whitelist zur Unterdrückung der tz-Ligatur

2.2.1 Blacklist (tz-Ligatur muss aufgelöst werden!)

Fragment	Beispiel	Fragment	Beispiel
achtz ...	achtzehn, achtzig	nachtz ...	Nachtzeug
aufrechtz ...	aufrechtzuerhalten	notz ...	Notzeiten
bettz ...	Bettzeug	quadratz ...	Quadratzenmeter
dienstz ...	Dienstzeit	rechtz ...	rechtzeitig
entz ...	entzünden, entzückt, Entzug	restz ...	Restzeit
fahrtz ...	Fahrtzeit	schriftz ...	Schriftzug
festz ...	festziehen, festzulegen	selbstz ...	Selbstzweifel
fortz ...	fortzusetzen	stadtz ...	Stadtzentrum
hautz ...	Hautzug	wertz ...	Wertzuordnung
luftz ...	Luftzug	wetz ...	wetzumachen
mitz ...	mitzuteilen, mitzureden	zeitz ...	Zeitzone

2.2.2 Whitelist (tz-Ligatur kann erhalten bleiben)

ätz	götz	kitz	metz	platz	sätz	schütz	stutz
batz	hatz	klotz	mütz	plätz	schätz	schwatz	stütz
blitz	hetz	klötz	netz	plötz	schlitz	schwätz	tratz
brutz	hitz	kratz	nutz	protz	schmitz	schwitz	trotz
dutz	jetz	latz	nütz	putz	schmutz	setz	wetz
fetz	katz	letz	patz	ritz	schnitz	sitz	witz
glotz	ketz	litz	pfütz	satz	schutz	spitz	

3 T_EX-Quellcode zu Textverarbeitung

Die Vorgehensweise ist weitgehend umgekehrt zu der in Abschnitt 2 Beschriebenen. Der Quellcode wird in den Zwischenspeicher kopiert und dann im Textverarbeitungsprogramm (Word, Writer) mittels des Einfügen-Kommandos für *unformatierten Text* wieder eingefügt. Kopiert man stattdessen *formatierten Text* aus der Zwischenablage, kann es vorkommen, dass z. B. die Farben der Darstellungsweise des Editors⁶ mit übertragen und anschließend gesamtheitlich gelöscht werden müssen.

Im Folgenden wird die Präambel gelöscht sowie alle T_EX-eigenen Befehle. Dabei hilft die nachfolgende Korrekturliste.

3.1 Korrekturliste für Bereinigung des T_EX-Codes

Im Grunde ist es einfach, einen ursprünglichen T_EX-Quellcode von allen Befehlsangaben zu »bereinigen«. Denn jeder T_EX-Befehl beginnt mit dem sog. *Backslash* (zu Deutsch: *Rückstrich*): \

Dieser ermöglicht den Zugriff auf die meisten T_EX-eigenen Befehle, die man nur noch löschen bzw. ersetzen muss. Darüber hinaus suche man auch nach dem Dollarzeichen (\$), das freistehende oder Inline-Formeln auszeichnet, sowie nach der Tilde (~), die einem geschützten Leerzeichen entspricht. Es wird, je nach Fähigkeiten der verwendeten Textverarbeitung, ebenso mit einem geschützten Leerzeichen ersetzt. Außerdem ist ggf. noch nach der Kombination " - zu suchen, die einen möglichen Silbentrenner im Wort kennzeichnet. Letztendlich kann es auch nicht schaden, nach vereinzelt Akkoladen zu suchen.

Ein weiterer Punkt gilt zu beachten: Manche Editoren wie K_il_e ermöglichen, den Quellcode während der Eingabe mittels »statischen Zeilenumbruchs« in kürzere Fragmente zu zerlegen, sodass – je nach Einstellung – beispielsweise während des Tippens eine neue Zeile begonnen wird, nachdem 80 Zeichen eingegeben wurden. Das Ganze hat Vor- und Nachteile (dazu lasse ich mich an anderer Stelle aus), ist aber für das Kompilieren des T_EX-Quellcodes unerheblich. Wird der Quellcode jedoch in eine herkömmliche Textverarbeitung übertragen, enthält er natürlich auch noch

⁶Manche Editoren wie TeXMaker färben die Kommando-Syntax (Befehle) Rot ein, Gliederungsstrukturen (section, subsection usw.) in Blau.

diese statischen Zeilenumbrüche, die (recht mühselig) rückgängig gemacht werden müssen. Hierbei hilft, die Absatz-Zeichen einzublenden.

4 Nutzung von Pandoc

Wie eingangs erwähnt, kann man sich das manuelle Nacharbeiten der im Grunde nur von Anwendung zu Anwendung kopierten Texte sparen, wenn man sich auf einen Parser wie Pandoc verlässt, der ganz automatisch einen Text von \LaTeX .doc/.docx (oder .odt) überführen kann und dabei alle internen Formatierungs- und Strukturierungsanweisungen weitgehend richtig übersetzt.

Ich hatte kürzliche das Vergnügen, aufgrund von Zeitmangel auf Pandoc zurückzugreifen, und möchte im Folgenden von meinen Erfahrungen berichten.

Übersetzt wurde ein mehrere hundert Seiten langes Dokument im .docx-Format, das sowohl eine Strukturierung in Kapitel- und Unterkapitel enthielt, als auch Bilder mit Unterschrift sowie Fußnoten. Tabellen und Formeln waren nicht enthalten, dazu kann ich also nichts sagen.

Die Umwandlung erfolgte auf der Kommandozeile (im selbigen Ergebnis wie die Eingabe-Datei) sehr einfach mittels:

```
1 pandoc -f docx -t latex -o Datei_Ergebnis.tex Datei_Original.docx
```

Dabei stehen:

- f docx für das Format des Original-Dokuments.
 - t latex für das Format des Ausgabe-Dokuments.
 - o Datei_Ergebnis.tex für das Output-Ergebnis.
- Datei_Original.docx kommt zuletzt und steht für die Eingabe-Datei.

Einige Sekunden später war das Dokument fertig. Schauen wir mal ...

4.1 Positive Beobachtungen

- Alle Absätze werden korrekt getrennt, d. h. durch eine Leerzeile separiert. Das ist schon mal viel wert, v. a. bei so umfangreichen Dokumenten.

- Gedankenstriche werden korrekt zu - - aufgelöst.
- Überschriften 1. und 2. Ebene heißen korrekt `\section{ }` und `\subsection{ }`.

4.2 Problemstellen

- Alle Bild-Umgebungen sind weg und es wurde keine einzige der Bildunterschriften übertragen, d. h. manuelles Rauskopieren aus dem Originaltext und Neuanlegen einer kompletten figure-Umgebung wird notwendig. Vom Bild bleibt nur übrig:
`\includegraphics[width=3.13in,height=2.34in]{media/image1.jpeg}`, also die ursprünglichen Abmessungen und eine Art Link, der aber nicht wirklich erstellt wurde, geschweige denn dass irgendeines der eingebetteten Bilder vom Original dort hinterlegt wurden. Natürlich ist auch die Verlinkung im Text futsch, dort steht nur noch z. B. Abbildung 1.
- Die im Word vorprogrammierte Absatzvorlage für Zitate wurde komplett aufgelöst und ist im Quelltext vom normalen Fließtext nicht sonderlich abgehoben oder gekennzeichnet. Muss also manuell nachgeprüft werden und z. B. in eine `\quote{ }`-Umgebung eingeschlossen werden.
- Beim Kompilieren werden Fehlermeldungen ausgegeben, und zwar für jene Stellen, an denen im Original-Text besondere Feldbefehle (Weiches Trennzeichen, Geschützter Bindestrich usw.) verwendet wurden. Diese benutzt man, um zu steuern, wann die Textverarbeitung die Worttrennung zu unterlassen hat oder an welchen Stellen sie trennen darf. Diese Fehlstellen müssen im Quelltext natürlich ausgetauscht werden. Sie sind dort unsichtbar, erscheinen also im Falle der Weichen Trennzeichen als Leerzeichen. Einfach die Position löschen und ein neues Leerzeichen eintippen.
- Nur ein Problem, wenn man im Original-Text unsauber formatiert hat: Kursive Wortauszeichnungen werden von Pandoc zwar korrekt mit `\emph{ }`-Anweisungen übersetzt, allerdings ggf. auch die vorangehenden oder folgenden Satzzeichen! Zur Erklärung: Normalerweise schließt man in \LaTeX nur das auszuzeichnende Wort in eine `\emph{ }`-Klammer ein und nicht noch das folgende

Satzzeichen, z. B. ein Komma⁷. In Word werden beim Markieren mit Mauszeiger jedoch automatisch das nachfolgende Satzzeichen mit kopiert. Kursiviert man dann das Markierte, gilt das eben auch für das nachfolgende Komma. Und ein kursives Komma kann man von einem normal stehenden Komma meist nur noch schwer unterscheiden. Gleiches gilt für nachfolgende Fragezeichen und andere. Mit anderen Worten: Es muss manuell geprüft werden, ob eine von Pandoc generierte `\emph{ }`-Umgebung auch nur das Wort enthält: „`\emph{Hurra! }`“ ist falsch, richtig ist: „`\emph{Hurra}!`“

- Eine andere Sache mit den Anführungszeichen, die den Pandoc-Programmierern offenbar auch noch nicht aufgefallen ist: Beim Übersetzen werden abführende (deutsche) Anführungszeichen (das Gänsefüßchen “) mit `` übersetzt. Da steht also im Quelltext: „Beispiel``. Soweit kein Problem beim Kompilieren – es werden wieder deutsche Gänsefüßchen erzeugt. *Das Problem erscheint erst, wenn den abführenden Anführungszeichen ein Satzzeichen wie ? oder ! vorangeht!* Denn dann wird der Code ganz anders gelesen und aus „Beispiel?`` wird „BeispielꞤ“ – Wie man sieht, ist das inakzeptabel. Man muss also mit einem Suchen & Ersetzen-Kommando das `` gegen `\grqq` oder die direkte Glyphen austauschen.

4.3 Seltsam, aber unproblematisch

- Eckige Klammern werden zusätzlich in geschweifte Klammern eingeschlossen: Aus `[Test]` wird also: `{[Test]}`. Stört aber nicht weiter beim Kompilieren.
- Eine Ellipse („drei Fortsetzungspunkte“ ...) werden mit `\ldots{}` übersetzt. Geht in Ordnung.

⁷An dieser Stelle muss angemerkt werden, dass einige Typografen die Meinung vertreten, dass im Falle der Kursivierung eines Wortes auch zwingend das nachfolgende Satzzeichen kursiviert werden muss, allein schon, um ein Ineinanderfließen des kursiven letzten Buchstabens und dem nachfolgenden, aufrechtstehenden Ausrufezeichen zu verhindern! Das scheint Sinn zu machen, nur dann müsste auch ein darauffolgendes oberes Gänsefüßchen mit kursiviert werden, um ebenfalls nicht in Konflikt zu geraten. Das alles hängt natürlich auch von der Zurichtung der verwendeten Schriftart ab, aber ich empfehle nach wie vor, allein das hervorzuhebende Wort in die Kursivierung einzuschließen. Und sei es nur, um den Auszeichnungsunterschied zwischen anführenden Anführungszeichen und abführenden Anführungszeichen desselben Wortes zu vermeiden!

- Jede `\section{ }`-Überschrift bekommt ein `label` zugewiesen, das der Original-Überschrift entspricht: Aus `Inhalt` wird also `\section{Inhalt}\label{inhalt}`. Wenn man eigentlich nie auf die Überschriften verweist, verkompliziert das natürlich den Code, stört aber auch nicht.
- Hin und wieder enthielt der erzeugte Quellcode mitten im Text folgenden String: `\protect\hypertarget{__Ddelink__ (mehrere Zahlen)}`. Ich habe keine Ahnung, was das sein soll; scheint irgendein Hyperlink-Relikt im Original-Text gewesen zu sein.