Vergleich zwischen TEX und herkömmlichen Textverarbeitungen

Ein sachlicher Vergleich

Gregor Barth

v2.13 vom 21. Februar 2022

Inhaltsverzeichnis

1 Vorwort und Begrifflichkeiten **2**

2 Konzeptbedingte Unterschiede in der Arbeitsweise **3**

3 Anschaffungskosten **6**

4 Technisches **6**
4.1 Speicherplatz . . . 6
4.2 Prozessor, Arbeitsspeicher, Grafik . . . 7
4.3 Betriebssystem . . . 8
4.4 Dateiformat . . . 8
4.5 Systemschriftarten . . . 9
4.6 Konfigurationsdateien . . . 10
4.7 Die Sache mit den Bugs . . . 10

5 Layout und Typografisches **10**
5.1 Ligaturen und OpenType-Features . . . 12
5.1.1 Ligaturen . . . 12
5.1.2 OpenType-Features . . . 13
5.2 Kapitälchen und Unterstreichungen . . . 14
5.3 Rechtschreibprüfung . . . 14
5.4 Formelsatz und Sonderzeichen . . . 14
5.5 Zurichtung (Kerning) . . . 15
5.6 Glyphen-Stauchung und variable Wortabstände . . . 16

5.7 Registerhaltigkeit 16
5.8 Umfang der Dokumente 17
5.9 Verlinkte Abbildungen 18

6 Was TEX kann, und Textverarbeitungen nicht 18

7 Was Textverarbeitungen können, und TEX nicht 22

8 Was soll ich nun nehmen? 24
8.1 Nimm TEX 24
8.2 Nimm eine herkömmliche Textverarbeitung 26

1 Vorwort und Begrifflichkeiten

Verglichen werden sollen die Ergebnisse eines TEX-Systems mit denen einer herkömmlichen Textverarbeitung. Im Folgenden wird von TEX gesprochen, wenn LATEX, XELATEX, LuaTEX und andere Varianten gemeint sind (im weiteren Sinne auch `groff`). Ein TEX-System besteht üblicherweise aus einer TEX-Distribution (Pakete-Sammlung, z. B. TeXLive) und einem beliebigen Text-Editor (WYSIWYM-Editor), idealerweise einem speziellen TEX-Editor (Anzeige der Dokumentstruktur, Syntax-Hervorhebung, Hilfe bei Verweisen etc.). Wo der Unterschied von Bedeutung ist, wird darauf hingewiesen.

Mit dem Ausdruck »herkömmliche Textverarbeitung« ist Software gemeint, die allgemein Bestandteil jeder Büro-Software-Suite ist: LibreOffice Writer, Microsoft Word, Calligra Words, Abiword, Lotus Words, Pages, Corel WordPerfect etc. Es handelt sich hierbei durchweg um WYSIWYG-Editoren. Wo der Unterschied von Bedeutung ist, wird darauf hingewiesen.

Es werden in diesen Vergleich keine DTP-Programme wie Adobe InDesign, Quark Xpress oder Scribus hinzugezogen. Dem Autor ist bekannt, dass mit diesen Programmen komplexer Textsatz (insbesondere mit hohem Bilder-Anteil, Zeitschriften) umgesetzt werden kann, zu dem herkömmliche Textverarbeitungen und auch TEX teilweise nur bedingt fähig sind. Ebenso unbeachtet bleiben in diesem Vergleich zahlreiche Programme für Spezialanwendungen, beispielsweise Editoren für Drehbuchautoren.

Mit diesem Vergleich wird beabsichtigt, die verbreitete Voreingenommenheit gegenüber der WYSIWYM-Arbeitsweise zu lösen, auch wenn die Erstellung kurzer Briefe bis hin zu umfangreichen Dokumenten mit einem GUI-Textverarbeitungsprogramm

einfacher und schneller erscheint. Der häufigste Grund für die Ablehnung von WYSIWYM ist die Arbeit mit Quellcode, jenem mystischen Äther, der nur aus kryptischen Befehlen zu bestehen scheint. Es soll gezeigt werden, dass nicht nur die Arbeit mit dem Quellcode mehr Kontrolle bietet, sondern dass gerade bei umfangreich strukturierten Dokumenten (wissenschaftliche Abschlussarbeiten etc.) und bei Aspekten der Qualitätskontrolle Vorzüge bestehen, die ein gewöhnliches Textverarbeitungsprogramm nicht bieten kann.

2 Konzeptbedingte Unterschiede in der Arbeitsweise

Eine Textverarbeitung wie MS Word oder LO Writer ist ein sog. WYSIWYG-Editor. Das steht für *what you see is what you get* und heißt: Was ich über die Tastatur eingebe, erscheint am Bildschirm mit derjenigen Formatierung, wie sie später gedruckt oder veröffentlicht werden soll. Auf dem Weg zum Drucker oder zum PDF kann allerdings eine Abwandlung der gedachten Formatierung (Layout) nicht ausgeschlossen werden.

WYSIWYG-Editoren bieten normalerweise eine üppig ausgebaute GUI (Programmoberfläche) mit allen erforderlichen Buttons für die Strukturierung oder Formatierung des Textdokuments, darunter Schalter für die Farbwahl, Schriftgröße, Zeilenabstand und Zeichenformatierung (Kursivierung, Fette, Unterstreichung etc.). Diese scheinbare Bequemlichkeit ist eine der Ursachen, wie Nutzer unbewusst zu einer schlechten Typografie kommen: Ihnen wird professioneller Textsatz über Buttons zugetraut, ohne zu wissen, was dahinter steht oder wann eine solche Formatierung wirklich angewendet werden sollte. Nutzer stellen oft Seitenformate nach Augenmaß ein, oder strukturieren ihren Text, indem sie Kapitelnummern von Hand vergeben.

Selbstverständlich entscheidet letztlich das Auge darüber, ob ein Text gut gesetzt ist. Doch es gibt gewisse Regeln, die eingehalten werden sollten, um den Vorgaben guter Lesbarkeit zu entsprechen:

- harmonische Seitenränder (z. B. nach dem Goldenem Schnitt oder ganzzahligen Seitenverhältnissen) und eine dazu passende Schriftgröße, sodass nicht zu viele Wörter pro Zeile gesetzt werden,
- zur Schriftgröße passender Durchschuss,

- eine für Mengentext angenehm lesbare Schriftart und ggf. dazu passende Schrift für Überschriften, Bildunterschriften usw.,
- Anordnung von Bildern immer in Rückwirkung mit dem umgebenden Text (anstatt manuelle Anordnung nach Gutdünken),
- konsequente Anwendung von Absatzvorlagen etc.

Der große Nachteil der WYSIWYG-Arbeitsweise ist, dass der Nutzer permanent, d. h. noch während des Schaffensprozesses, in Layout und Formatierung seines Textes eingreift, anstatt sich, wie es sein sollte, auf den bloßen Inhalt zu konzentrieren.

Dieser Arbeitsweise stehen WYSIWYM-Editoren gegenüber. WYSIWYM steht für *what you see is what you **mean*** und bedeutet schlichtweg: Ich habe die volle Kontrolle über meinen Text und nur was ich an Formatierungsregeln vorgebe, das wird auch angewendet und umgesetzt. Ein im WYSIWYM-Editor geschriebener Text (»Quellcode«) wird dann an den TEX-Compiler zum Setzen übergeben, der daraus das Endergebnis (meist PDF) erzeugt.

In einem WYSIWYM-Editor (= Texteditor beliebiger Wahl) findet man nur im Ausnahmefall Buttons für die Textformatierung. Schriftgröße und -art werden üblicherweise in den Optionen des Editors eingestellt. Der gesamte eingegebene Text erscheint in einer gleichen Schriftgröße und -art (meist eine nüchterne Monospace-Schrift), lediglich durch Syntax-Hervorhebungen unterlegt[1].

Da es im Quellcode sonst keine besondere Hervorhebung für die Dokumentstruktur oder Textformatierungen gibt, kann sich der Autor ganz auf den Inhalt des Dokuments konzentrieren. Er überlässt das Formatieren und Strukturieren, das Setzen und Layouten dem TEX-System.

Der TEX-Compiler setzt den Quellcode nicht einfach nur in Wörter und Absätze um, sondern wendet eine Reihe interner Satzregeln an (siehe unten), um den Text in möglichst gut lesbarer und ästhetischer Weise zusammenzustellen.

Wenn der Anwender in einer herkömmlichen Textverarbeitung permanent das Layout mit den Augen prüft und verändert, erhöht sich auch die Fehlerchance, dass Absatzvorlagen nicht konsequent eingehalten werden: Wie auch unter TEX bietet eine

[1] Richtigerweise muss man ergänzen, dass es auch TEX-Editoren wie LyX gibt, bei denen zwar Quellcode bearbeitet, aber ein teilformatiertes Ergebnis angezeigt wird. Diese Arbeitsweise kommt TEX-Einsteigern entgegen.

herkömmliche Textverarbeitung wie LO Writer die Möglichkeit, den Text mithilfe von sog. Absatzvorlagen zu formatieren. So werden alle Überschriften derselben Ebene und auch der gesamte Brottext gleichartig (Schriftgröße, Schriftart, Farbe etc.) formatiert und wahlweise durchgängig nummeriert. Problematisch wird es, wenn Absatzvorlagen ungenutzt bleiben oder das Fehlen ihrer Anwendung nicht bemerkt wird. Was sich im Fließtext vielleicht noch als unproblematisch erweist, könnte zu einer fehlerhaften Nummerierung (auch im Inhaltsverzeichnis) der Überschriften führen. Möglich ist außerdem, dass verwaiste, also leere Absatzvorlagen zwischen den Absätzen stecken, z. B. eine Überschrift ohne Inhalt, sodass im Inhaltsverzeichnis eine leere Zeile generiert wird. Konzeptbedingt hat sich der Anwender in einer herkömmlichen Textverarbeitung um all diese Dinge selbst zu kümmern[2].

Im Unterschied dazu sind derartige Fehler bei TEX ausgeschlossen, oder zumindest nur mutwillig herbeizuführen: Allein durch die Arbeitsweise mit einer Befehlssyntax wird spätestens beim Kompilieren deutlich, wo Fehler gemacht worden sind (nämlich durch Ausgabe eines Protokolls, das auf fehlerhafte oder unbekannte Befehlssyntax hinweist). Dementsprechend wird die Erstellung des Dokuments erst dann fortgesetzt, wenn alle Fehler behoben sind. Unvollständig angewendete Absatzvorlagen gibt es daher nicht; auch bei der Nummerierung von Überschriften, Fußnoten, Abbildungen usf. wird niemals etwas durcheinander kommen (können).

Wenn sowohl in TEX als auch in einer herkömmlichen Textverarbeitung Absatzvorlagen konsequent angewendet worden sind, ist es problemlos möglich, Formatierungsänderungen vorzunehmen: Alle Überschriften der 2. Ebene doch lieber in 16 pt, kursiv und in Grün? Kein Problem. – Das wäre nicht möglich, wenn der Anwender sich sein Layout selbst bauen würde, indem er beispielsweise manuell seine Überschriften nummeriert und direkt beim Eingeben derselben eine Schriftgröße und -art festlegt.

Zur Vollständigkeit sey auf Hybride wie *Word Perfect* verwiesen, das sich zunächst durchaus als herkömmliche Textverarbeitung bezeichnen lässt. Mithilfe eines Features namens *reveal codes* können allerdings neben dem WYSIWYG-Text sämtliche Formatierungscodes übersichtlich angezeigt, verstanden und manipuliert werden, die zum gegenwärtig sichtbaren Layout geführt haben.

[2] Kleine Erweiterungen wie der `Pepito Cleaner` (als Erweiterung für LibreOffice) checken abschließend das Dokument auf solche Unstimmigkeiten und geben eine Fehlerliste aus, die abgearbeitet werden kann.

3 Anschaffungskosten

Unter den herkömmlichen Textverarbeitungen gibt es kommerzielle und freie. Unter den kommerziellen ist MS Word sicherlich die bekannteste. Natürlich zahlt man auch für jedes Upgrade und ggf. auch für jeden Arbeitsplatz. Außerdem gibt es Unterschiede im Umfang, je nach Anwender (Studenten-Lizenz, Professionell). Neuerdings gibt es von MS Office auch ein Abo-Modell, was wiederum von einer bestehenden Internetverbindung abhängig macht (sofern man die angebundenen Cloud-Dienste nutzen möchte).

Freie (quelloffene) Textverarbeitungen (LibreOffice, Calligra-Suite, Abiword) und auch TEX sind dagegen frei von Kosten zu beziehen, selbstverständlich auch alle Updates. Ebenso gibt es keine Einschränkungen hinsichtlich Anzahl der Arbeitsplätze oder ähnliches. Unter den WYSIWYM-Editoren (Textexitoren, TEX-Editoren) gibt es auch solche mit freier und kommerzieller Lizenz.

4 Technisches

In dieser Kategorie wird angenommen, dass das Textverarbeitungssystem auf dem heimischen Computer betrieben werden soll. Für beide Lager (herkömmliche Textverarbeitungsprogramme und TEX) gibt es nämlich auch reine Online-Anwendungen, bei denen der Nutzer allein über den Browser auf seine Dokumente zugreift, und eine lokale Installation inklusive aller eventuellen Hardware-Fragen entfällt.

4.1 Speicherplatz

Diese Kategorie ist bei heutigen Festplattengrößen eigentlich kein Thema mehr. Dennoch eine kurze Ausführung:

MS Office belegt in seiner aktuellen Version nicht unter 1 GB Speicherplatz; andere Textverarbeitungen wie LibreOffice sind je nach Installationsumfang genügsamer. Einfache Textverarbeitungsprogramme wie Abiword belegen gerade einmal wenige Megabyte.

Den meisten Büro-Suiten ist gemein, dass sie sich modular installieren lassen, d. h. beim Installieren kann man die Komponenten der jeweiligen Office-Suite auswählen.

Auch von einer aktuellen TEX-Distribution wie TeXLive wird wenigstens ein Gigabyte Speicherplatz belegt, je nachdem, wie umfangreich sie installiert wird.

Hier unterscheidet sich eine TEX-Distribution von einem herkömmlichen Office-Programm, denn sie lässt sich wesentlich modularer installieren; genau genommen kann man sich ein System mit nur denjenigen Komponenten zusammenstellen, die man wirklich benötigt. TEX lässt sich also (bei Kenntnis der erforderlichen Pakete) besser an die Bedürfnisse des Nutzers anpassen, und belegt damit nicht unnötig Speicherplatz (insbesondere auf älteren Systemen!).

Bei TEX kommt weiterhin ein Texteditor hinzu. Da es sich bei TEX-Quellcode um reine Textdateien handelt, ist die Wahl des Texteditors unerheblich. Der Nutzer kann sich seine Arbeitsumgebung also frei wählen.

Die meisten modernen Texteditoren verstehen die Syntax von TEX-Quellcode. Einige spezielle TEX-Editoren bieten auch gleich alles an, was man für die Bearbeitung von TEX-Dokumenten braucht, inklusive Anzeige der Dokumentstruktur bis hin zu Buttons für das Kompilieren. Andere Editoren (z. B. `Atom` oder `Sublime`) lassen sich mit Paketen nachrüsten, um das zu ermöglichen. Manche TEX-Editoren generieren synchron zum Quellcode eine Vorschau des Dokuments (was aber dem Prinzip von WYSIWYM entgegendrängt). Für einen Texteditor fallen jedenfalls selten mehr als 50 MB zusätzlicher Speicherplatz an.

4.2 Prozessor, Arbeitsspeicher, Grafik

Jede herkömmliche Textverarbeitung wie LO Writer oder MS Word hat den Nachteil, dass sie eine grafische Oberfläche (GUI) darstellt, und nicht, wie TEX, wahlweise auch im reinen Textmodus, z. B. im Terminal-Fenster, betrieben werden kann. Dementsprechend benötigen Erstgenannte einen modernen Prozessor, nicht zu wenig RAM und manchmal auch etwas Unterstützung von der Grafikkarte, wenn alles flüssig laufen soll. Besonders MS Office scheint von Version zu Version optisch aufgeblasener zu werden (Transparenz, Schatten, Animationen in der GUI). Je mehr dieser optischen Spielereien verbaut sind, desto mehr werden Systemressourcen verbraucht und desto träger wird die Eingabe oder Verarbeitung von Text sein.

Da, konzeptbedingt (Abschnitt 5), Wortumbrüche und Layout bei herkömmlichen Textverarbeitungen während der Eingabe berechnet werden (»Live-Text«) und nicht

erst, wie bei TEX, während des Kompilierens, wird ebenfalls vermehrt Prozessorleistung beansprucht als bei TEX.

TEX-Quellcode lässt sich mit jedem beliebigen Computer öffnen und bearbeiten, und auf alten wie auf neuen Computern zu einem fertigen Dokument kompilieren (auch wenn es länger dauert). Eine bestimmte Abhängigkeit zu aktuellen Betriebssystemen besteht daher nicht.

4.3 Betriebssystem

Bekannt ist, dass die Abwärtskompatibilität bei MS Word keine Priorität zu haben scheint, weder was das Dateiformat betrifft (Abschnitt 4.4), noch die Anwendung selbst. Ein modernes MS Office auf einem Windows 98 zu betreiben, ist unmöglich. Außerdem ist MS Office nicht für GNU/Linux verfügbar (kann aber emuliert werden). LibreOffice ist da einsichtiger und lässt sich auf nahezu jedem System mit grafischer Oberfläche installieren, ob nun Windows (ab XP), klassischem Desktop-Linux, BSD, Mac, Android, Solaris oder sonst was. Auch einige andere Textverarbeitungsprogramme wie Abiword sind plattformübergreifend verfügbar.

TEX kann ebenfalls auf den meisten Betriebssystemen ausgeführt werden (Windows, Apple, GNU-Linux, OS/2, MS-DOS, OpenVMS, BeOS, Amiga u. a.). Selbst wenn eine TEX-Distribution für ein bestimmtes Betriebssystem nicht verfügbar ist – der Quellcode lässt sich in jedem Fall bearbeiten.

4.4 Dateiformat

Das von MS Word verwendete Format `.doc` (und sein XML-Luftballon `.docx`) gelten weithin als Standard-Dateiformat zum Austausch von Textdokumenten. Nur sind sie kein Standard und sollten niemals einer sein. Bekannt ist, dass Word-Formate gar nicht gut auf Abwärtskompatibilität zu sprechen sind; dass sich unter Word 95 erstellte Dateien mit modernen Word-Versionen mitunter gar nicht mehr öffnen lassen. Layout-Veränderungen sind selbst zwischen nachfolgenden Word-Versionen möglich. Das Word-Format ist kein Format für Dokumente, die auch in ferner Zukunft noch lesbar sein sollen.

Ein Standard ist dagegen das OpenDocument-Format. Bezüglich der Textdokumente heißt es `.odt`. Es handelt sich um ein XML-basiertes und mit ZIP komprimiertes,

offen dokumentiertes Format, das von einem internationalen Gremium (und nicht einem einzelnen Konzern) betreut wird. Es gilt mittlerweile als (berechtigter) Standard für die An- und Ablage von Textdateien. Im Gegensatz zum `.docx`-Format belegt es sogar geringfügig weniger Speicherplatz, da im `.docx` (das ebenfalls auf XML basiert) die XML-Syntax zwar technisch korrekt geschrieben, aber jedes einzelne Wort mit XML-Tags umklammert wird, anstatt, wie bei `.odt`, nur die Absätze. Das bläht natürlich die Dateigröße auf.

Das Dateiformat von TEX, also eine reine Textdatei mit der Endung `.tex`, ist das in diesem Vergleich vermutlich einfachste Dateiformat. Es kann mit jedem beliebigen Texteditor geöffnet und bearbeitet werden. Das gilt freilich auch für XML, das in `.odt` und `.docx` eingebettet ist!

Es ist zu bedenken, dass reine Textformate mitunter die ältesten Austauschformate für digitale Texte sind, d. h. ein in den 80er Jahren erstellter `.tex`-Code lässt sich (mit wenigen Anpassungen, ggf. einer korrigierten Zeichencodierung) auch heute noch lesen, und er wird auch noch in Jahrzehnten lesbar sein.

Hinzu kommen zwei weitere Vorzüge reiner Textdateien:

- Da in `.tex` nur der reine Inhalt gespeichert wird, und nicht etwa Software-spezifische Informationen, bleibt die Größe der Datei, selbst unkomprimiert, sehr klein. Bei Word-Dateien werden dagegen ggf. auch Vorgängerversionen des gleichen Dokuments gespeichert, Hardware-spezifische Informationen im Header u. a.

- Bei reinen Textdateien kann der Editor gelegentlich abstürzen, ohne dass gleich der Inhalt des gesamten Dokuments beschädigt werden muss. Sollte etwas bei komplexen Formaten wie `.odt` oder `.docx` beschädigt werden, kann die Datei ggf. nicht mehr eingelesen werden.

4.5 Systemschriftarten

Prinzipiell können sowohl herkömmliche Textverarbeitungen als auch TEX die auf dem System installierten Schriftarten verwenden. TEX selbst bringt Dutzende weitere Schriftarten mit, die über entsprechende Pakete eingebunden werden. Externe Schriftarten lassen sich mithilfe von XƎTEX oder LuaTEX verwenden.

4.6 Konfigurationsdateien

Die das TEX-Dokument betreffenden Einstellungen werden im Quellcode selbst hinterlegt; die Konfiguration des Texteditors wird separat in einer einfachen Text-Datei (config-Datei) gespeichert. Große Textverarbeitungen wie MS Word oder LO Writer verteilen ihre Programm-internen Einstellungen etwas unübersichtlich auf der Festplatte.

Soll für eine Anzahl von Dokumenten dieselbe Konfiguration zur Verfügung stehen, kann die TEX-Präambel in eine eigene Datei ausgelagert und in allen Dokumenten verlinkt werden. So modifiziert man sie nur noch an einer Stelle.

4.7 Die Sache mit den Bugs

Komplexe Software wie LibreOffice oder MS Office enthalten selbstverständlich immer wieder Bugs, nicht umsonst gibt es Updates. Und ferner: Mit jeder neuen Version gibt es weitere Bugs und weitere Updates. Dieses ewige Hin und Her kennt jede Software. TEX dagegen hat den Ruf, dass Bugs nur selten gefunden werden. Natürlich besteht eine TEX-Distribution aus vielen Paketen, die, selbstverständlich, auch hin und wieder von Bugs befreit oder mit neuen Funktionen ausgestattet werden. Sie entwickeln sich weiter oder veralten und sind dann nicht mehr mit anderen, aktuellen Paketen problemlos nutzbar. Wieder andere Pakete werden seit Jahren nicht mehr weiterentwickelt, ganz einfach, weil sie ausgereift und keine Bugs bekannt sind. Anwender, die einer Flut von Warnungen und Fehlern nach dem Kompilieren ihrer Dokumente gegenüberstehen, neigen gern dazu, auf Programmierfehler zu schließen. Üblicherweise handelt es sich aber um unachtsam gesetzten Code, d. h. Syntax-Fehler.

Der Bug-Anfälligkeit würde ich jedoch keine besondere Wertung in diesem Vergleich beimessen. Hier geht es um Funktionalität und Endergebnis, egal wie oft das Programm zwischendurch abstürzt.

5 Layout und Typografisches

Wie in Abschnitt 2 bereits angedeutet, kann man mit TEX ohne viel Eingreifen einen Text erzeugen, der harmonisch aussieht und gut lesbar ist. Das hat nicht nur damit zu tun, dass TEX die meisten Layout-Fragen selbst übernimmt (Satzspiegel-Größe etc.),

sondern auch der Umbruch-Algorithmus anders funktioniert als bei einer Textverarbeitung wie MS Word oder LO Writer: Hier werden die Wörter nach Sprachvorgaben umgebrochen, die Zusammenstellung der Wörter und ihr Umbruch erfolgt meist auf Zeilenbasis, d. h. sobald ein Wort nicht mehr in die Zeile passt, wird es umgebrochen oder auf die nächste Zeile verwiesen. Beim resultierenden Blocksatz können dann sehr unterschiedliche Ergebnisse herauskommen: Lücken reißende Wortabstände oder wenigstens ein unausgeglichener Grauwert der Seite sind keine Seltenheit.

Bei TEX geschieht der Wort-Umbruch auf Absatzbasis (was auch nicht weiter verwundert, denn schließlich schreibt man einen Absatz erst zu Ende, ehe man ihn setzt): Während des Kompilierens wird der Absatz »analysiert« und der Absatz so gesetzt (und umgebrochen), dass alle Wortabstände gleichmäßig (über den gesamten Absatz) verteilt sind. Daraus resultiert ein einheitlicher Grauwert und der Effekt, dass Testpersonen denselben Text schöner und harmonischer finden, wenn er mit TEX gesetzt worden ist.

Wer ein paar optische Vergleiche zwischen Word und LATEX vorzieht, kann sich beispielsweise hier und hier umsehen. Dort gibt es Beispiele für Ligaturen, Kerning, Kapitälchen und mehr, jeweils im Vergleich zu Word.

Durch Hinzuladen des `microtype`-Pakets kann man durch Aktivierung weiterer typografischer Feinheiten den Grauwert noch erhöhen. So ist es möglich, den sog. optischen Randausgleich zu nutzen, bei dem Zeichen mit viel Fleisch (V, W) oder Bindestriche, sofern sie am Zeilenende stehen, ein wenig nach außerhalb vom Satzblocks verschoben werden. Damit werden »lückige Stellen« aufgefüllt. Das genannte Paket kann außerdem Wortzwischenräume und Kegelbreiten der Buchstaben leicht variieren, sodass der Textblock noch ausgeglichener wirkt. Typisch sind solche Möglichkeiten für DTP-Programme, etwa das sog. `hz-Programm`, das heute in Adobe InDesign integriert ist.

Auch in Textverarbeitungen wie LibreOffice sind solche typografischen Optimierungen möglich, teilweise abhängig von der Schriftart (beispielsweise die Graphite-Version von `Libertine`, die den optischen Randausgleich unterstützt).

Ein bedeutender Unterschied findet sich, wenn man die Stabilität des Layouts betrachtet: Ein mit TEX-Quellcode geschriebenes Dokument wird, unabhängig vom Betriebssystem, der Hardware oder dem Drucker, zu jeder Zeit genau gleich aussehen. Das lässt sich von herkömmlicher Textverarbeitung nicht zwingend behaupten:

Selbst das offene, gut dokumentierte OpenDocument-Format kann ein unterschiedliches Ergebnis liefern, je nachdem ob man dieselbe Datei mit LibreOffice, MS Office oder Calligra öffnet. Prinzipiell gelten OpenDocument-Dateien als »Layout-stabiler« im Vergleich zu Word-Dateien. Bei MS Word kommt dazu, dass das Layout des Dokuments vom installierten Druckertreiber abhängig sein kann, d. h. das gedruckte Dokument kann unterschiedlich aussehen, je nachdem, auf welchem Drucker es gedruckt wurde. Durch Kompatibilitätsprobleme kann es beim Öffnen mit anderen Textverarbeitung zum Verschieben von Bildern und Bildunterschriften oder einem anderen Wortumbruch kommen.

5.1 Ligaturen und OpenType-Features

5.1.1 Ligaturen

Durch Ligaturen an entsprechender Stelle erhöht sich die Lesbarkeit eines Textes. Ligaturen werden üblicherweise automatisch gesetzt, sowohl von TEX als auch von herkömmlichen Textverarbeitungsprogrammen.

Für den Ligaturen-Satz gibt es Regeln, die es zu prüfen gilt. Ob eine Ligatur gesetzt wird (werden darf), hängt von folgenden Aspekten ab:

- Ligaturen müssen in der verwendeten Schriftart enthalten sein. Normal ausgebaute Antiqua und Serifenlose enthalten meist nur die Standard-Ligaturen[3] fi und fl.
- In serifenlosen Schriften stehen die Buchstaben üblicherweise so unabhängig, dass eine fehlende Ligatur nicht negativ ins Gewicht fällt.
- Die Dokumentsprache entscheidet, welche und wie viele Ligaturen gesetzt werden dürfen: In englischer Sprache wird grundsätzlich jede verfügbare Ligatur gesetzt, im Deutschen wird an Wortfugen nach Regelung darauf verzichtet. Da eine manuell aufgelöste Ligatur *nicht selten* eine auffällige Lücke mitten ins Wort reißt, muss man abwägen, inwiefern man dieser Regelung folgen möchte.

[3] Je nach Schriftgruppe meint man mit »Standardligaturen« etwas anderes: In Antiqua-Schriften sind dies fi und fl, bei Gebrochenen Schriften kommen noch viele weitere dazu. In einer Frakturschrift gelten beispielsweise ch, ck und tz als sog. Zwangsligaturen und müssen immer gesetzt werden.

Vergleiche: auf fällig vs. auffällig und Schilf insel vs. Schilfinsel!

Mit einem lückenhaften Textblock hat man guter Typografie nicht unbedingt zugearbeitet.

Soll eine Ligatur nicht gesetzt werden, muss sie manuell unterdrückt werden. Hierfür gibt es unter TEX diverse Kommandos (das bekannteste ist `Kauf\/haus`) oder man kann Ligaturensatz dokumentweit abschalten. Es gibt auch Pakete, die Ligaturen überall dort unterbrechen, wo sie nach Regelung nicht angewendet werden darf (an Wortfugen). Bei einer herkömmlichen Textverarbeitung kommt es auf die verwendete Schriftart an: Manche sind so programmiert, dass sie automatisch Ligaturen ausgeben, bei anderen sind Ligaturen generell deaktiviert und müssen alle von Hand eingefügt werden.

Ob man Deutsch oder Englisch schreibt, welche Schriftart man verwendet, welche Textart man setzt (Antiqua, Serifenlose, Fraktur) und welcher typografischen Schule man folgt – Ligaturen bedeuten für den Setzer stets erhöhte Aufmerksamkeit und ggf. Nachbearbeitung des Textes.

5.1.2 OpenType-Features

Viele modern ausgebaute Schriften enthalten sog. OpenType-Features wie alternative Buchstabenformen, kontextsensitive Ersetzungen oder sogar weitere Ziffernsätze (Versalziffern vs. Mediävalziffern, Tabellenziffern vs. Proportionalziffern).

Unter TEX müssen solche Spielereien erst separat aktiviert werden. Das geht z. B. mit XƎTEX und LuaTEX. Bei MS Word findet sich ab Version 2013 eine entsprechende Funktionserweiterung im Zeichen-Dialog. Unter LibreOffice können derartige Features aktiviert werden (Anleitung), indem man die entsprechenden OpenType-Kürzel an die Schriftart setzt. Außerdem gibt es für LO Writer eine Erweiterung namens `Typography toolbar`, welche eine Werkzeugleiste einblendet, über die sämtliche OpenType-Funktionalitäten einer Schriftart verfügbar werden[4].

Wie bei den Ligaturen gilt: Die Verwendung dieser Möglichkeiten muss abgewogen erfolgen und hängt ab von Textinhalt, Schriftart usf.

[4] Diese Erweiterung arbeitet nur mit der Graphite-Version einer Schriftart zusammen, z. B. Libertine!

5.2 Kapitälchen und Unterstreichungen

Je nach Schriftart sind im Glyphensatz echte Kapitälchen enthalten. Kapitälchen werden häufig und typografisch falsch durch Verkleinerung von Versalbuchstaben generiert: herkömmliche Textverarbeitungen stellen diese Möglichkeit durch die Kapitälchen-Formatierung bereit, ohne darauf hinzuweisen, dass die verwendete Schriftart vielleicht gar keine Kapitälchen enthält. Werden Kapitälchen von TEX angefordert, die in der Schriftart nicht enthalten sind, wird dies beim Kompilieren mit einer Warnung protokolliert.

Was Unterstreichungen angeht, werden diese bei herkömmlichen Textverarbeitungen wie LO Writer durch die Unterlängen der Buchstaben gezogen. Obwohl Unterstreichungen für sich allein typografisch fragwürdig sind, ist die Durchschneidung der Unterlängen typografisch doppelt falsch. Je nach Paket werden Unterstreichungen bei TEX dagegen unterhalb der Unterlängen der Buchstaben gesetzt.

5.3 Rechtschreibprüfung

Die »Rechtschreibprüfung« (Orthografie) entspricht sowohl bei herkömmlichen Textverarbeitungen als auch bei Text-Editoren einfach einem Abgleich der Wörter mit einer *Whitelist*. Ist das betreffende Wort nicht enthalten, wird es rot unterkringelt oder kann hinzugefügt werden. Der Umfang dieser Whitelist ist wenig von der Textverarbeitung abhängig; im Internet können derartige Wörterbücher (*dictionaries*, Datei-Endung `.dic`) gedownloaded und in den Editor oder die Textverarbeitung einbezogen werden.

Grammatikalische Unstimmigkeiten werden in einem Text-Editor üblicherweise nicht geprüft. Mithilfe eines Plugins (z. B. `LanguageTool`) können sowohl in MS Word und auch LibreOffice die Wortstellung und -auswahl auf grammatikalische Richtigkeit geprüft werden. Auf eventuelle Beanstandungen ist allerdings kritisch einzugehen.

5.4 Formelsatz und Sonderzeichen

TEX ist bekannt für seinen exzellenten Formelsatz beliebig umfangreicher und komplexer Formeln; spezielle, auf unterschiedliche Formelebenen abgestimmte Glyphengrößen helfen beim Setzen einer insgesamt harmonisch aussehenden Formel-Struktur.

MS Word und LO Writer haben zwar auch eigenständige Formelsatz-Module, die aber v. a. bei großen und komplexen Formeln schnell an die Grenzen der ästhetischen Darstellung stoßen. Meistens werden für Hoch- und Tiefstellungen aller Art einfach die Glyphen der derzeitigen Schriftart skaliert, ohne dass sie auf gute Lesbarkeit in kleinen Größen optimiert sind; außerdem werden Zeichen oftmals enger zusammengepresst als sie sollten. Da ich normalerweise keine komplexeren Formeln setze (außer durch Bruchstrich getrennte Terme), kann ich die Leistungsfähigkeit nur unzureichend beurteilen. Viele Anwender kritisieren auch eine auf amerikanische Maßstäbe getrimmte Formeldarstellung, die nicht immer allen Ansprüchen genügt. Nicht unerwähnt soll sein, dass hochkomplexe Formeln im TEX-Quellcode zu einem Gewirr aus Buchstaben und Zahlen ausarten können, in denen schnell die Übersicht verloren geht. Spezielle Formelmodule wie das von LibreOffice zeigen dagegen direkt das Endergebnis (oder wahlweise den dahinterliegenden Code, der dann TEX ähnelt), sodass man stets die Übersicht behält und auf Veränderungen leichter einwirken kann.

Was die Symbolvielfalt angeht, sind TEX und herkömmliche Textverarbeitungen vergleichbar. Beide greifen zurück auf einen Fundus von speziellen Symbol-Schriftarten (oder, im Falle von TEX, auf entsprechende Symbol-Pakete), sodass sich jedes nur erdenkliche Zeichen darstellen lassen kann.

TEX ist darüber hinaus in der Lage, mithilfe von Paketen musikalische Noten oder chemische Strukturformeln (mit optimalen Abständen) zu setzen; etwas, das ich von keiner herkömmlichen Textverarbeitung kenne.

5.5 Zurichtung (Kerning)

Viele professionelle Schriftarten sind mit Tausenden Kerning-Paaren ausgestattet, die den Textfluss bestimmter Buchstaben-Paare optimieren (z. B. die Unterschneidung der Buchstabenpaare T-a und V-o). In den meisten Textverarbeitungsprogrammen (LO, MS Office älter als 2010) sind nur ein Teil dieser Kerning-Tabellen ansprechbar. Mit der Nutzung von XETEX oder LuaTEX kann dagegen der gesamte Umfang der OpenType-Schrift, darunter die vollständigen Kerning-Tabellen, ausgereizt werden. Eigene Tests (Kerning-Vergleiche) zeigen, dass der Satz derselben Paare zuweilen besser in TEX aussieht als in LO Writer oder MS Word.

Selbstverständlich sind die Möglichkeiten der Zurichtung abhängig von der Anzahl programmierter Kerning-Paare in der Schriftart. Eine schlecht zugerichtete Schriftart wird auch mit TEX schlecht zugerichtet aussehen.

5.6 Glyphen-Stauchung und variable Wortabstände

TEX erreicht seinen überzeugenden Textsatz teilweise durch Anwendung mikrotypografischer Feinheiten, etwa der (fast nicht sichtbaren) Stauchung von Glyphen oder der Variation von Wortabständen, die in jeder Zeile geringfügig variieren kann.

Diese Features können beispielsweise durch Hinzuladen des `microtype`-Pakets (in seinen Standardeinstellungen) aktiviert werden. `microtype` gleicht den Text damit so aus, dass der Grauwert einer Seite erhöht wird. Abhängig von der verwendeten Schriftart und der Sprache (englischer Text mit kurzen Wörtern vs. deutscher Text mit langen, schlechter umbrechbaren Buchstabenschlangen) kann die Buchstabenstauchung mehr oder weniger auffällig sein und sollte dann deaktiviert werden (Option `font expansion`). Ich persönlich hielt das noch nie für notwendig und lade das Paket stets mit Standardoptionen in jedem Dokument.

Um in einer herkömmlichen Textverarbeitung ein ausgeglichenes Satzbild zu erzeugen, bleibt oft nur das manuelle Verteilen von sog. weichen Trennstellen, die zu den Standard-Trennstellen nach Wörterbuch dazukommen. Eine umständliche Angelegenheit bei langen Texten.

5.7 Registerhaltigkeit

Die sog. Registerhaltigkeit ist ein in pdfETEX fehlendes und häufig nachgefragtes Thema. Darunter versteht man die identische Zeilenhöhe auf gegenüberliegenden Seiten eines Blattes Papier. Das heißt, wenn man ein Blatt gegen das Licht hält, sollten die Grundlinien der Zeilen genau deckungsgleich sein.

Nun ist es so, dass TEX ein harmonisches Satzbild (gleichmäßige Verteilung der Textblöcke, Überschriften, Bildelemente etc.) über die gelegentlich freizügige Dehnung der strikten Registerhaltigkeit erreicht. Texte, die mit pdfETEX gesetzt worden sind, können, müssen aber nicht registerhaltig sein.

Dieser Umstand kann mehr oder weniger auffällig sein. Für mich ist dieses »Manko« so unbedeutend, dass ich nie versucht habe oder versuchen werde, meine Texte strikt

registerhaltig zu setzen. Wer dennoch diesen Anspruch erhebt, der kann Pakete wie `grid` oder `gridset` verwenden, oder gleich seinen Quellcode mit ConTEXt setzen.

Für alle, die nicht mit TEX arbeiten und sich dennoch der Registerhaltigkeit sicher sein wollen, müssen zu einer der großen DTP-Programme wie InDesign oder QuarkXPress greifen.

5.8 Umfang der Dokumente

Beispiele für umfangreiche Dokumente gibt es viele: Handbücher, Abschlussarbeiten, Bücher, Zeitschriften mit separaten Beiträgen etc.

Nicht selten ist es sinnvoll, das Dokument in einzelne Kapitel aufzutrennen. Bei Verwendung einer herkömmlichen Textverarbeitung geschieht dies häufig durch Anlage einzelner Textdokumente. Das Endergebnis wird dann in einem gemeinsamen PDF zusammengeführt.

Handelt es sich aber um Abschnitte, die durch Querverweise, gemeinsame Literatur- und Indexlisten oder ein Inhaltsverzeichnis aufeinander aufbauen und voneinander abhängen, muss sich der Anwender um deren Konsistenz und Korrektheit selbst bemühen. Gleiches gilt für eine evtl. durchgängige Nummerierung von Bildunterschriften, Seitenzahlen, Überschriften-Nummern usw.

Mit LibreOffice kann man dieser Problematik mit dem wenig bekannten »Globaldokument« (Dateiendung `.odm`) entgegnen, das als Header für beliebig viele Unterdateien fungiert, die z. B. als einzelne `.odt`-Dateien (Kapitel) in einem Ordner liegen. Jede der Dateien lässt sich einzeln bearbeiten, aber im Globaldokument sind alle zusammen sichtbar, mit einheitlicher Nummerierung, Seitenzahl, gemeinsamen Inhaltsverzeichnis usw.

Mit TEX können auf dieselbe Weise Dokumente untereinander verkettet werden (Kommando *input*): Einzeln angelegte `.tex`-Dateien (= Kapitel) werden über eine globale `.tex`-Datei vereint. Beim Kompilieren werden die einzelnen Dateien zusammengefügt, Seitenzahlen, Abbildungsnummerierungen etc. korrekt gezählt und gesetzt. Die TEX-internen Zähler und Automatismen arbeiten dabei so effizient, dass man im Endergebnis sicher sein kann, dass alle Querverweise, Literaturzitate und sonstiges, das dynamische Felder gebraucht, fehlerfrei sind. Und bei Unstimmigkeiten würde man während des Kompilierens mit einer ausgegebenen Warnung darauf hingewiesen.

Diese Möglichkeit ist einer der Hauptgründe, weshalb ich die Arbeit mit einem TEX-System bevorzuge.

5.9 Verlinkte Abbildungen

Einfacherweise werden bei illustrierten Textdokumenten die Bilder in die Datei eingebettet. Dadurch können die Dateien rasch eine Dateigröße annehmen, bei denen der Programmabsturz zu erwarten ist. Zumindest dürfte das Arbeiten mit dem Dokument stets träge reagieren. Hinzu kommt die Dauer für das Öffnen und Speichern der Datei. Sinnvoller ist hier das Verlinken der Bilder.

In TEX-Quellcode können, konzeptbedingt, Bilder nur verlinkt werden, denn es handelt sich um reine Textdateien, die keine Bilder aufnehmen können.

In beiden Fällen ist die Verlinkung eines Bildes sinnvoll, denn sie bewahrt die volle Kontrolle über die Bild-Dateien: Obwohl auch innerhalb der Textverarbeitungen wie MS Word oder LO Writer eine nachträgliche Korrektur (Farbstich, Kontrast, Zuschnitt etc.) möglich ist, sollten sie extern verwaltet und bearbeitet werden. Da sie verlinkt sind, werden sie im (schlankeren) Dokument aktualisiert, sobald sie verändert wurden.

6 Was TEX kann, und Textverarbeitungen nicht

- Möglichkeit, auch mit transparenten und übereinander angeordneten Schriftblöcken zu arbeiten. Ich gebe zu, dass mir keine sinnvolle Anwendung dafür einfällt.

- Manuelle Trennstellen (weiche Trenner) für falsch umgebrochene Wörter lassen sich wesentlich einfacher setzen. Alternativ kann eine dokumentweit gültige Wortliste für häufig gebrauchte, sonst fehlerhaft umgebrochene Wörter angelegt werden.

- Effizienter und verlässlicher Automatismus zum Erzeugen von Inhaltsverzeichnissen, Literaturverzeichnissen, bibliografischen Zitaten, Nummerierung von Bildunterschriften, Indizes, Kopfzeilen etc. Derartigen Automatismen in einer herkömmlichen Textverarbeitung zu vertrauen, setzt eine einwandfreie

Anwendung von Absatzvorlagen voraus. Ebenso verlässlich ist die Änderung von Absatzvorlagen mit TEX.

Hier wird der Vorteil des zusätzlichen Arbeitsschrittes (»Kompilieren« des Quellcodes) offenbart, denn dabei erfolgt die Prüfung, ob die eingegebene Syntax fehlerfrei ist und Sinn ergibt.

- Die Eingabe von halben Leerzeichen, Geviertstrichen und anderen typografischen Kleinigkeiten erfordert nicht den Wechsel in die Glyphentabelle (und die Suche nach dem Zeichen), sondern ist direkt im Quellcode möglich, und zwar mit allen auf der Tastatur sichtbaren Zeichen (Standard-ASCII-Zeichen).
- Programmierbarkeit nach eigenen Bedürfnissen, z. B.:
  - Einrichtung einer Dokumentklasse, speziell für Geschäftsbriefe: Festlegung der Position von Brief-Inhalt, Anhangsliste, Absender-Feld, Firmen-Logo, Falz- und Lochermarken. Einiges davon geht über die vertrauten Dokument-Vorlagen hinaus, die man von MS Word und LO Writer kennt.
  - Definition von Umgebungen und wiederkehrenden Wortbausteinen.
- Kontrollgewinn/Transparenz des Dokuments: Durch das reine Textformat sehe ich stets, warum ein bestimmter Text diese oder jene Formatierung angenommen hat. Eine herkömmliche Textverarbeitung zeigt mir in der Tat nur das Ergebnis und verbirgt im Unsichtbaren, im Hintergrund, alle Informationen, die zu diesem Ergebnis geführt haben (insbesondere bei proprietären Dateiformaten). Es ist also möglich, dass in dem Dokument Informationen stehen, die ich am Bildschirm gar nicht sehen kann. TEX-Quellcode zeigt mir dagegen immer an, was wirklich in der Datei enthalten ist: Ich weiß genau, mit welchem Zeichen die Datei beginnt, und mit welchem sie endet. Das gibt mir auch die Kontrolle, dass die Datei bei Weitergabe nur das enthält, was sie enthalten soll.

  Richtigerweise muss man hinzufügen, dass XML-basierte Dokumentformate selbstverständlich die Möglichkeit bieten, die zugrundeliegende `.xml`-Datei mit einem Texteditor zu öffnen und einzusehen. Auch hier wäre theoretisch jede Strukturierung und Formatierung, jede enthaltene Information sichtbar.

- Vermeidung typografischer Fehler, z. B.:
  - Im Quellcode doppelt (oder beliebig mehrfach!) eingegebene Leerzeichen werden nach dem Kompilieren auf ein einziges reduziert. Im fertigen PDF stößt man also nie auf zwei Wörter, die aus Versehen durch zwei statt einem Leerzeichen spationiert worden sind (außer man forciert das mittels Befehlen). In LibreOffice kann man in den Autokorrektur-Optionen auch einstellen, dass doppelt eingegebene Leerzeichen ignoriert werden. Der bereits erwähnte `Pepito Cleaner` (als Erweiterung für LibreOffice) kann das Dokument am Ende nach doppelten Leerzeichen durchsuchen, oder man tut das mit der Suchen/Ersetzen-Funktion der Textverarbeitung.
  - Ein durch TEX erzeugter Text kann nur diejenigen Schriftarten enthalten, die in der Präambel des Dokuments festgelegt wurden. Dass also das fertige PDF zwei sehr ähnlich aussehende Schriften[5] nebeneinander enthält, ohne dass es weiter auffällt, ist nicht möglich. Dieses Szenario geschieht gelegentlich beim unsachgemäßen Kopieren von (vorformatierten) Text aus der Zwischenablage ins Dokument. Würde solcher Text aus der Zwischenablage in den TEX-Quellcode übertragen, wäre er gleichwertig mit dem bereits bestehenden Text. Eine Schriftart wird erst beim Kompilieren festgesetzt.
  - Die Nutzung des `csquotes`-Pakets vorausgesetzt, werden Anführungszeichen nicht mehr manuell eingegeben, sondern das Wort wird mit einem Befehl eingeklammert. Nur am Dokumentanfang wird definiert, welche Form von Anführungszeichen verwendet werden sollen (deutsch/englisch, einfach/doppelt etc.). Beim Kompilieren werden diese nach Vorgabe gesetzt, und können ebenso einfach, dokumentweit!, ausgetauscht werden. Durch die Einklammerung ist außerdem ausgeschlossen, dass man aus Versehen nur das eine Anführungszeichen setzt und das zweite vergisst.
  - Literaturverwaltung zusammen mit BibTEX bzw. BibLATEX: Literatur in einer Datenbank, eindeutige Schlüssel werden im Quellcode zitiert; verwaiste Zitate im Mengentext oder überflüssige Einträge im Literaturverzeichnis sind gar nicht möglich! Korrekte Sortierung der Einträge und

[5] Beispielsweise zwei Garamonds, die sich alle recht ähnlich sehen.

einheitliche Formatierung werden durch Automatismen und einen ausgewählten, eindeutigen Zitierstil übernommen.

- TEX »stürzt« so gut wie niemals ab, egal wie groß das Dokument wird oder wie alt das System ist.
- Mit TEX erzeugter Text sieht immer identisch aus, egal auf welchem System man ihn kompiliert.
- Der Editor zum Bearbeiten des TEX-Codes ist nach Vorlieben, Hardware und Gewöhnung frei wählbar. Dazu zählt auch, dass ich mir ein (augenschonendes) dunkles Farbthema einrichten kann, während ich bei herkömmlichen Textverarbeitungen auf vorinstallierte Designs angewiesen bin. Sie lassen bestenfalls eine Auswahl von Farbschattierungen der Buttons oder den Austausch der Symbole zu. Gleichwohl bleibt mir das weiße, grelle Blatt als Eingabefeld erhalten[6].
- Die verwendeten Kommandos zur Formatierung eines Textes sind seit Jahrzehnten weitgehend die gleichen. Daran werden auch neuere Versionen eines Editors nichts ändern. Hin und wieder bringt eine Paket-Aktualisierung auch einige neue Befehle mit sich und markiert andere als obsolet. Das ist aber in der Dokumentation ausführlich beschrieben und kann mit Alternativen entsprechend ausgeglichen werden.
- Man kann auch ungewöhnliche Strukturen setzen, z. B. Musiknoten, Sternenkarten oder Schachbrett-Konstellationen.

  Für die Dokumentation von Quellcode (z. B. Code-Schnipsel im Rahmen eines Programm-Handbuchs) gibt es eigene Pakete, die Quellcode abgesetzt, mit einer Monospace-Schrift, Syntax-Hervorhebung und Zeilennummern darstellen. Natürlich kann man auch mit MS Word und LO Writer eine Absatzvorlage einrichten, bei der der Inhalt als Monospace mit Zeilennummern erscheint. Jedoch habe ich noch nie gesehen, dass damit auch die aus Texteditoren bekannte automatische Einrückung oder Syntax-Hervorhebung möglich wäre.

[6] Korrekt ist, auf spezielle Texteditoren für Vielschreiber hinzuweisen. Diese zeigen bedarfsweise eine Vollbild-Ansicht ganz ohne Buttons und wahlweise mit augenschonender Farbe von Hintergrund und Schrift.

- TEX-Code kann kommentiert werden, d. h. das Kommentar-Zeichen für diese Sprache (das Prozent-Zeichen) leitet Text ein, der nur im Editor sichtbar ist, nicht aber im kompilierten Ergebnis erscheint. Das eröffnet die Möglichkeit für die Hinterlegung von Bemerkungen (beliebiger Länge), wie bestimmte Befehle funktionieren oder gedacht sind; oder man »versioniert« seinen Text, indem man veraltete Passagen auskommentiert und so unsichtbar erhält. Oder man nutzt den Kommentar zum Ein-/Ausschalten alternativer Präambel-Befehle für Testzwecke. In einer herkömmlichen Textverarbeitung kann man natürlich auch Kommentare anbringen, die dann als schwebende Farbboxen am Seitenrand erscheinen (sie erhalten sogar automatisch Erstellungszeit und Namen des Erzeugers). Nachteilig bei diesen Boxen ist, dass sie nur begrenzt Text aufnehmen können.

7 Was Textverarbeitungen können, und TEX nicht

- Soll das Textdokument (nicht als PDF) weitergegeben werden, ist die Arbeit mit einer herkömmlichen Textverarbeitung einfacher: Über den Speichern-Unter-Dialog kann man eines von vielen gängigen Dateiformaten auswählen. Das `.tex`-Format sollte man dagegen nur an Leute weitergeben, die damit auch arbeiten können oder zumindest zur Nutzung eines Texteditors mit Syntax-Hervorhebung bereit sind. Für die Konvertierung eines TEX-Quellcodes zu einem Dateiformat wie `.docx` oder `.odt` existieren diverse Programme (z. B. der Parser `Pandoc`), denen gemein ist, dass das Ergebnis fast immer einer Nachbearbeitung bedarf. Manchmal ist der Zwischenschritt über HTML besser, das wiederum von MS Word oder LO Writer eingelesen wird. Für den umgekehrten Weg gibt es beispielsweise die LibreOffice-Erweiterung `Writer2LaTeX`, über die ein mit einer herkömmlichen Textverarbeitung erstelltes Dokument für TEX aufbereitet werden kann. Editoren wie `LyX` haben eingebaute Encoder für den Export z. B. ins OpenDocument-Format.

- Herkömmliche Textverarbeitungen legen nur eine Datei pro Dokument an. TEX-Quellcode liegt zwar ebenfalls nur in einer einzigen Datei vor (sofern nicht aufgeteilt), aber beim Kompilieren entstehen daraus zahlreiche Metadateien, die

Dokumentstruktur, Bibliografie etc. aufnehmen. Das kann den Arbeitsordner unordentlich werden lassen. Beim Archivieren gilt es daher alle Arbeitsdateien zu löschen, was einige TEX-Editoren auf Knopfdruck oder automatisch beim Schließen des Editors erledigen. Bei der Arbeit mit dem `LyX-Editor` entfällt dieser Punkt, denn es wird mit einer `.lyx`-Datei gearbeitet, die sich beim Kompilieren nicht »vermehrt«.

- Herkömmliche Textverarbeitungen bringen direkt eine Verschlüsselungsfunktion für ihre Dateien, mit denen die Berechtigung zum Lesen und/oder Schreiben geregelt werden kann. TEX-Quellcode ist zunächst nicht verschlüsselt, sondern muss nachträglich über ein Drittprogramm verschlüsselt werden (z. B. verschlüsseltes ZIP-Archiv).

- Wortzählung ist bei MS Word und LO Writer einfacher; hier genügt ein Klick auf den entsprechenden Button, oder die Anzahl wird direkt in der Statuszeile angezeigt. Die Wort-Zählung von TEX-Quellcode ist nicht so trivial, denn Kommando-Befehle (auch die Präambel) zählen nicht dazu. Verschiedene Editoren bieten eine Wortzählung an, die genau diese Befehlsstruktur »herausrechnet«, allerdings alle mit unterschiedlichem Ergebnis bei demselben Text (siehe hier). Bei Online-Editoren wird die Wortanzahl häufig in einer Statusleiste direkt angezeigt. Einige TEX-Editoren wie `LyX` können ebenfalls auf Knopfdruck die Wortstatistik ausgeben.

- Absatzvorlagen oder individuelle Listen-Umgebungen sind für TEX-Anfänger nicht leicht programmierbar; das geht in einer herkömmlichen Textverarbeitung einfacher.

- In MS Word/LO Writer ist die Anbindung einer Datenbank, z. B. für Serienbriefe, einfacher.

- Das Einfügen von Tabellen, deren Ausrichtung, Sortierung, Gliederung, Formatierung (auch Textrichtung) ist in einer herkömmlichen Textverarbeitung wesentlich einfacher als in TEX. Dort wird eine komplexe Tabelle rasch zum undurchsichtigen Gewirr aus Befehlen und Text, und die Orientierung, in wel-

cher Zelle man gerade etwas verändert, ist kaum nachvollziehbar[7]. Gleichwohl können sich Tabellen aus einer herkömmlichen Textverarbeitung mit solchen aus einem TEX-Quellcode ästhetisch nicht messen. Diverse TEX-Editoren wie LyX oder Kile enthalten Tabellen-Assistenten, die vieles vereinfachen, bei LyX sogar auf Knopfdruck Zeilen/Spalten löschen oder anlegen und Zellen verbinden können.

- Das beliebte Feature »Änderungen nachverfolgen«, mit dem sich das Editieren des Dokuments durch Zweit- und Drittautoren nachverfolgen und dokumentieren lässt, lässt sich in einer herkömmlichen Textverarbeitung bequemer bedienen. Selbstverständlich ist so ein Vorgang in einem TEX-Dokument ebenso möglich (Paket changes)! Der TEX-Editor LyX enthält bereits eine Funktion für die Änderungsnachverfolgung, die sich genauso bequem bedienen lässt wie in einer großen Textverarbeitung.

  Andererseits bieten manche Online-TEX-Editoren Funktionen zur Zusammenarbeit an: Dokumente könnten geteilt, d. h. gleichzeitig daran gearbeitet werden; Kommentarboxen und vollzogene Änderungen lassen sich für andere hinterlegen usf.

- Da man sich am fertigen Seitenlayout orientieren kann, ist das schnelle Korrigieren kleiner Textfehler einfacher. In Quellcode sucht man erfahrungsgemäß länger, um die zu korrigierende Stelle zu finden, insbesondere, wenn man sich an dem dazugehörigen PDF orientiert. Wer den TEX-Editor LyX nutzt, sieht bereits ein teilformatiertes Layout, mit dem sich genauso rasch und übersichtlich arbeiten lässt.

8 Was soll ich nun nehmen?

8.1 Nimm TEX …

- wenn es um typografisch und ästhetisch einwandfreien Textsatz geht,
- wenn du komplexe Formeln setzen musst,

[7] Als Einstieg könnte man die Tabelle mit der Tabellenverarbeitung Gnumeric vorbereiten und dann als TEX-Quellcode exportieren.

- wenn du dir keine Gedanken um die korrekte Nummerierung (Überschriften, Bilder, Seitenzahlen) und Zuordnung von Verweisen (Bibliografie-Zitate!) machen willst. Manch einer wird schon Stunden in den manuellen Abgleich gesteckt haben, ob Textzitate zu den Einträgen der Literaturliste passen!

8.2 Nimm eine herkömmliche Textverarbeitung …

- wenn dir die Arbeit mit Quellcode-Syntax zu abgehoben ist (oder nimm LyX),
- wenn du dir keinen Überblick über die Funktionalität der einzelnen TEX-Pakete verschaffen willst/kannst (oder nimm LyX),
- wenn du sofort optisches Feedback für deine Eingaben brauchst, z. B. weiche Trennstellen (oder nimm LyX),
- wenn du viel mit Schriftarten experimentieren willst,
- wenn du komplexe Tabellen setzen musst, die nicht in einem Tabellendokument abgelegt werden können (oder nimm LyX),
- wenn an deinem Dokument noch intensiv gearbeitet wird (wissenschaftliches Dokument mit mehreren Co-Autoren), und man sich intensiv auf »Änderungen nachverfolgen«, Kommentarboxen und dergleichen verlässt. Obwohl gerade dann die konsequente Anwendung von Absatzformatierungen kompliziert ist[8], wenn ständig Absätze und Bilder eingefügt und verschoben werden, und auch das endgültige Layout noch gar nicht klar ist (oder bestimmt werden darf, siehe Weitergabe des Manuskripts an den Verlag). Oder nimm LyX!

[8] Dokumentinterne Vorgaben wie Absatzvorlagen können beeinträchtigt werden, wenn das Dokument mit verschiedenen Programmversionen bearbeitet wird!