

Die T_EXnische Komödie

dante
Deutschsprachige
Anwendervereinigung T_EX e.V.

29. Jahrgang Heft 3/2017 August 2017

3/2017

Impressum

»Die \TeX nische Komödie« ist die Mitgliedszeitschrift von DANTE e.V. Der Bezugspreis ist im Mitgliedsbeitrag enthalten. Namentlich gekennzeichnete Beiträge geben die Meinung der Autoren wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nicht gestattet. Alle Rechte zur weiteren Verwendung außerhalb von DANTE e.V. liegen bei den jeweiligen Autoren.

Beiträge sollten in Standard- \LaTeX -Quellcode unter Verwendung der Dokumentenklasse `dtk` erstellt und per E-Mail oder Datenträger (CD/DVD) an untenstehende Adresse der Redaktion geschickt werden. Sind spezielle Makros, \LaTeX -Pakete oder Schriften notwendig, so müssen auch diese komplett mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden. Weitere Informationen für Autoren findet man auf der Projektseite <http://projekte.dante.de/DTK/AutorInfo> von DANTE e.V.

Diese Ausgabe wurde mit LuaTeX , Version 1.05.0 (TeX Live 2017) erstellt. Als Standard-Schriften kamen Libertinus Serif, Libertinus Sans, Anonymous Pro und Libertinus Math zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2400

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung \TeX e.V.
Postfach 10 18 40
69008 Heidelberg

E-Mail: info@dante.de (DANTE e.V.)
dtkred@dante.de (Redaktion)

Druck: Konrad Triltsch Print und digitale Medien GmbH
Johannes-Gutenberg-Str. 1–3, 97199 Ochsenfurt-Hohestadt

Redaktion: Herbert Voß (verantwortlicher Redakteur)

Mitarbeit: Lukas C. Bossert Gert Ingold Eberhard Lisse
Rolf Niepraschk Christine Römer

Redaktionsschluss für Heft 4/2017: 15. Oktober 2017

ISSN 1434-5897

Die \TeX nische Komödie 3/2017

Editorial

Liebe Leserinnen und liebe Leser,

die meisten Beispiele dieser Ausgabe finden Sie als komplette und lauffähige Dokumente auf <http://www.dante.de/DTK/Software.html>. Das Paket `dtk-extern` erlaubt innerhalb eines Artikels, vollständige Beispiele zu definieren. Diesen kann optional unter anderem das zu verwendene Programm und die auszugebene(n) Seite(n) mitgegeben werden. Die Ausgabe des Quellcodes kann bezüglich der Präambel gesteuert werden; im Allgemeinen wird man im Text nur die wesentlichen Teile einer Präambel angeben wollen. Dies kann mit den Zeilen

```
%StartVisibleCode  
...  
%StopVisibleCode
```

gesteuert werden. Im Text dieser Ausgabe finden Sie den sichtbaren Präambelteil mit einem etwas stärkeren Grau hinterlegt. Der Bereich des Textkörpers folgt danach, wobei für \LaTeX -Beispiele die beiden Zeilen `\begin{document}` und `\end{document}` nicht mit ausgegeben werden.

»Die \TeX nische Komödie« enthält in dieser Ausgabe wieder einige nützliche Hinweise zur Anwendung von KOMA-Script-Dokumentenklassen. Weiterhin finden Sie einen Überblick über die Möglichkeiten zur Erstellung von speziellen Baumstrukturen, sowie einen Erfahrungsbericht über die Vorgehensweise beim Erstellen eines Buchmanuskripts.

Ich wünsche Ihnen wie immer viel Spaß beim Lesen und verbleibe

mit \TeX nischen Grüßen

Ihr Herbert Voß

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

in den vergangenen Tagen habe ich zu Hause ein wenig umräumen müssen. Dabei habe ich u. a. viele \TeX -Bücher sowie Ausgaben des TUGboat und der DTK in Händen gehalten. Es ist immer wieder beeindruckend, wie viele zehntausende Seiten über die annähernd 40 Jahre hinweg von Menschen auf der ganzen Welt zum Thema \TeX und Textsatz geschrieben, veröffentlicht und gelesen wurden.

Vieles davon ist natürlich nicht mehr ganz »up-to-date«, doch so manches zeitlose Schätzchen verbirgt sich sicherlich in den Papierstapeln. Vermutlich geht es vielen von Ihnen wie mir. Man würde gerne ein wenig schmökern und probieren, doch lässt es der Alltag oftmals zeitlich nicht zu.

Wie schön, dass es dennoch so viele \TeX -Nutzer gibt, die in Buchform oder mittlerweile auch sehr stark im Internet ihr Wissen aufbereiten und anderen Nutzern zugänglich machen. Auch nach vielen Jahren regelmäßiger Arbeit mit \TeX finde ich immer wieder nützliche Tipps und Tricks, fundierte Erklärungen und konkrete Hilfe.

Eine weitere Form der Wissensvermittlung sind traditionell die verschiedenen \TeX -Tagungen. Dazu gehören neben unseren eigenen Veranstaltungen natürlich auch die internationalen Tagungen. Ich möchte daher an dieser Stelle zum einen auf das bevorstehende 11. Internationale Con \TeX T-Nutzertreffen hinweisen, das vom 11. bis 17. September im hessischen Butzbach stattfinden wird. DANTE e.V. sponsort die Tagung als Ganzes sowie mit je einem individuellen Zuschuss von 50 EUR zum Tagungsbeitrag. Mehr Informationen finden Sie unter <http://meeting.contextgarden.net/>. Zum anderen weise ich gerne auch auf die TUG 2018 hin, die vom 20. bis 22. Juli 2018 in Rio de Janeiro ausgerichtet wird (<http://tug.org/tug2018/>). Auch wenn naturgemäß aufgrund der Entfernung nur wenige Mitglieder zu dieser ersten offiziellen TUG-Veranstaltung in Lateinamerika reisen werden, möchte ich darauf hinweisen, dass für Fragen rund um Anreise, Unterbringung etc. mit Paulo Cereda ein direkter Ansprechpartner zur Verfügung steht (paulocereda@gmail.com), der sich sehr auf Gäste aus Deutschland, Österreich und der Schweiz freut.

Wenn Sie diese Zeilen lesen, dann haben einige unserer Mitglieder ihr \TeX -Zelt auf dem SHA2017-Camp gerade wieder abgebaut. Wir wagten uns hiermit auf Neuland für unseren Verein. Neben Workshops zu konkreten Themen boten wir eine allgemeine Anlaufstelle für alle \TeX -Interessierten. Ich bin sehr gespannt, wie die Resonanz sein wird und was die Teilnehmer berichten werden.

Gelegenheit zum Austausch darüber besteht z. B. auf der Herbsttagung in Mönchengladbach am 23. September, zu der ich Sie nochmals ganz herzlich einlade. Sie können sich unter <https://www.dante.de/events/Herbst2017/anmeldung.html> registrieren und dort auch weitere Informationen rund um die Tagung finden. Gerne können Sie auch einen Vortrag einreichen. Den entsprechenden Aufruf finden Sie auch noch einmal in dieser Ausgabe.

Um die Hürden für den Nachwuchs zu senken, hat der Vorstand beschlossen, Schülern und Studenten die Reisekosten nach und von Mönchengladbach zu erstatten. Voraussetzung ist lediglich eine vorherige Tagungsanmeldung (bis zum 15. September) sowie ein Nachweis über die angefallenen Kosten (bis zur Höhe einer Bahnfahrt 2. Klasse). Geben Sie dies bitte gerne an mögliche Interessenten weiter.

Auch in diesem Jahr gibt es wieder eine »frische« Auflage der \TeX -Collection-DVD, die dieser Ausgabe der DTK beiliegt. Wir hatten erneut die Koordination inne und haben in Kooperation mit der TUG sowie vielen europäischen Nutzergruppen insgesamt 5000 Exemplare herstellen lassen. Mein Dank gilt den vielen engagierten Helfern – vom CTAN-Team, das das Tagesgeschäft erledigt und damit die Grundlage für die Distributionen legt, über die Entwickler der verschiedenen Installationsprogramme, die Autoren der Dokumentationen, die Tester bis zu all den vielen Anderen, die ihren Teil zum Gelingen beigetragen haben. Nicht zuletzt danke ich den vielen Paketautoren, die Jahr für Jahr ihre Erweiterungen aktualisieren und verbessern, sowie neue Ideen in Pakete gießen. Über 5000 kostenfreie \TeX -Erweiterungen von mehr als 2400 Entwicklern stehen mittlerweile über <https://www.ctan.org> bereit.

\TeX ist so erfolgreich, dass dies die letzte DVD-Veröffentlichung sein wird. Das klingt paradox, ist aber letztlich nur konsequent: Die vielen Erweiterungen und Zusatzprogramme passen trotz ausgeklügelter Skripte nicht mehr auf eine Double-Layer-DVD. Wir wollen daher vom Vorstand aus die Gelegenheit nutzen, mit Hilfe einer Umfrage gegen Ende des Jahres zu eruieren, ob es überhaupt sinnvoll ist, ein Nachfolgemedium zu suchen oder zukünftig ausschließlich auf die etablierte Möglichkeit der Onlineinstallation zu verweisen. Auch hier warte ich gespannt auf die Ergebnisse.

In diesem Sinne wünsche ich Ihnen viel Vergnügen bei der weiteren Lektüre.

Herzlichst Ihr/Euer
Martin Sievers

Einladung zur Herbsttagung 2017 und 57. Mitgliederversammlung von DANTE e.V.

Ulrike Fischer, Martin Sievers

Liebe Mitglieder von DANTE e.V.,

hiermit laden wir Sie ganz herzlich zur Herbsttagung 2017 und 57. Mitgliederversammlung am 23. September 2017 in Mönchengladbach ein.

Die genaue Anschrift lautet:

Volkshochschule Mönchengladbach
Lüpertzender Straße 85
41061 Mönchengladbach

Für die Tagung und Mitgliederversammlung, ist folgender Zeitplan vorgesehen:

Freitag, 22. September, ab 19:00 Uhr: Vorabendtreff
Samstag, 23. September, 9:00 Uhr: Mitgliederversammlung
ab 11:00 Uhr bis etwa 17:30 Uhr: Tagungsprogramm mit Vorträgen
ab 19:00 Uhr: Abendtreff
Sonntag, 24. September, ab 10:00 Uhr: Touristikprogramm

Die Tagesordnung der Mitgliederversammlung lautet:

1. Begrüßung und Tagesordnung
2. Bericht des Vorstands
3. Verschiedenes

Ihre Stimmunterlagen erhalten Sie direkt vor Ort. Eine Übertragung des Stimmrechts ist im Rahmen des § 13 (4) der Vereinssatzung möglich. Wie üblich sind auch Nichtmitglieder als Gäste herzlich willkommen.

Detaillierte Informationen zur Tagung finden Sie ab sofort unter <http://www.dante.de/events/Herbst2017.html>. Diese Seite wird im Laufe der nächsten Wochen fortlaufend aktualisiert.

Die Teilnahme an der Tagung ist kostenfrei, es wird jedoch um Anmeldung bis zum 10. September 2017 gebeten. Bitte nutzen Sie dazu das Formular auf der Tagungsseite.

Fragen, Wünsche und Anregungen richten Sie bitte per E-Mail an mv57@dante.de.

Beiträge gesucht (»Call for Papers«)

Wir möchten als Organisatoren für die Herbsttagung ein spannendes und vielfältiges Vortragsprogramm anbieten können. Dazu sind wir allerdings auf Eure/Ihre aktive Unterstützung angewiesen. Mögliche Themen für Einreichungen können sein:

- Einführungen in \TeX und die zugehörigen Makropakete ($\LaTeX 2_{\epsilon}/\LaTeX 3$, \ConTeXt , ...),
- Lösungen mit den neuen Programmen $X_{\text{Y}}\TeX$ und $\text{Lua}\TeX$,
- \TeX -Systeme,
- Beispiele aus der Praxis (beispielsweise Realisierung besonderer Anforderungen); eigene Klassen und Pakete,
- Typografie und ihre Umsetzung in \TeX und Co.,
- Zusammenspiel von \TeX mit anderen Dateiformaten (z. B. XML) und anderen (Open-Source-)Werkzeugen,
- die Entwicklung von \TeX und Co. in den vergangenen Jahrzehnten,
- Mustererzeugung und Zeichnen mit TikZ oder PSTricks ,
- Barrierefreie PDF-Dokumente,
- ...

Fühlen Sie sich angesprochen? Dann senden Sie bitte eine E-Mail mit folgenden Angaben an mv57@dante.de:

- Name der Referentin/des Referenten,
- Titel und Art (Vortrag oder Tutorium) des Beitrags,
- Zeitbedarf (Tutorien dauern im Allgemeinen 60 bis 90 Minuten (längere Tutorien sind möglich); für Vorträge beträgt die übliche Dauer 30 Minuten plus 10 Minuten für die anschließende Diskussion.),
- kurze Zusammenfassung (ca. 0,5 bis 1,5 Seiten),
- evtl. benötigte Hilfsmittel (jenseits von Beamer und PDF-Viewer),
- evtl. Wünsche bzgl. der Vortragszeit.

Wir freuen uns über alle Einreichungen, es darf auch gerne Ihr erster »Auftritt« bei einer Tagung von DANTE e.V. sein.

Mit freundlichen Grüßen

Ulrike Fischer (lokale Organisatorin)
Martin Sievers (Vorsitzender DANTE e.V.)

Bretter, die die Welt bedeuten

Definition eines neuen Gliederungsbefehls mit KOMA-Script

Elke Schubert

In diesem Beitrag soll an einem Beispiel gezeigt werden, wie mit dem KOMA-Script-Befehl `\DeclareNewSectionCommand` ab der Version 3.21 ein neuer Gliederungsbefehl erstellt werden kann und bereits vorhandene geändert werden können. Außerdem wird ein Ausblick auf geplante Änderungen in der Version 3.24 gegeben.

Seit der Version 3.15 von KOMA-Script ist es möglich, mit den vier Befehlen `\DeclareNewSectionCommand`, `\DeclareSectionCommand`, `\ProvideSectionCommand` und `\RedeclareSectionCommand` neue Gliederungsbefehle zu erzeugen oder vorhandene anzupassen. Bis zur Version 3.19 war das allerdings auf Überschriften im Stil von Abschnittsüberschriften beschränkt.

Ab der Version 3.20 kann mit diesen Befehlen auch der Stil der zugehörigen Einträge im Inhaltsverzeichnis angepasst werden. Bei der Definition neuer Gliederungsüberschriften ist das teilweise sogar nötig. Seit der Version 3.21 werden bei Änderungen an der Hierarchie der Gliederungsbefehle auch die vom Paket `hyperref` erzeugten Lesezeichen korrekt gesetzt.

Ab der KOMA-Script-Version 3.24 wird es Voreinstellungen für etliche der bisher nötigen Optionen geben, auch wenn deren explizites Setzen weiterhin empfohlen wird. Die betroffenen Optionen sind im Folgenden mit Fußnoten markiert und die laut [2] geplanten Änderungen am Ende des Artikels beschrieben.

Einstellungen für zusätzliche neue Gliederungsbefehle

```
\DeclareNewSectionCommand[<Einstellungen>]{<Name der Ebene>}
```

Die anderen drei Befehle haben eine analoge Form. Es wird also in jedem Fall der Name der zu definierenden oder zu ändernden Ebene erwartet. Die Einstellungen

für die Gliederungsbefehle werden im scheinbar optionalen Argument vorgenommen. »Scheinbar« deshalb, weil es eine ganze Reihe von Optionen gibt, denen ein Wert zugewiesen sein muss. Welche Eigenschaften das sind, hängt auch vom Stil der Überschrift (Option `style`), der verwendeten Klasse und dem Stil der Inhaltsverzeichniseinträge (Option `tocstyle`) ab.

Unabhängig von `style` notwendige Einstellungen

Für die Erzeugung einer neuen Gliederungsebene stets nötige Optionen:

- `style`
- `level`
- `beforeskip`¹
- `afterskip`¹
- `font`²

Abhängig vom Wert der Option `tocstyle`, also dem für die Gliederungsebene gültigen Stil der Einträge im Inhaltsverzeichnis, können zu den obigen Eigenschaften noch die beiden folgenden hinzukommen:

- `tocindent`
- `tocnumwidth`

Von `style` abhängige notwendige Einstellungen

Außerdem gibt es Optionen, deren Existenz vom Stil der Überschrift abhängt. Wird eine neue Überschrift mit dem entsprechenden Wert für `style` definiert, müssen sie ebenfalls angegeben werden.

Für `style=section` zusätzlich nötige Option:

- `indent`²

Für `style=chapter` und `style=part` zusätzlich nötige Optionen:

- `prefixfont`²
- `innerskip`² (nur `scrreprt` und `scrbook`)
- `pagestyle`² (nur `scrreprt` und `scrbook`)

Weitere Optionen

Es gibt noch einige Optionen, die angegeben werden können, aber nicht müssen:

¹ Ab der Version 3.24 existieren für diese Option für `style=chapter` und `style=part` Voreinstellungen.

² Ab der Version 3.24 existiert für diese Option eine Voreinstellung.

- counterwithin
- expandtopt
- toplevel
- tocstyle (beeinflusst, welche Einstellungen mit Präfix »toc« notwendig sind)
- abhängig von tocstyle: weitere Optionen mit dem Präfix »toc«

Wird für eine neue Gliederungsüberschrift toplevel nicht gesetzt, dann erhält sie den bei der Definition gültigen Wert von level. Und wird sie für einen neuen Gliederungsbefehl tocstyle weggelassen, dann wird der Stil default für die Einträge ins Inhaltsverzeichnis verwendet, der bei den KOMA-Script-Klassen auf dem tocline-Stil basiert. Eine ausführliche Erklärung aller Optionen sowie deren Voreinstellungen für die einzelnen von den KOMA-Script-Klassen schon definierten Gliederungsbefehle sind in [1] zu finden.

Änderungen an Einstellungen

Bei Änderungen an schon vorhandenen Überschriften brauchen immer nur diejenigen Optionen gesetzt werden, deren Wert angepasst werden soll oder die wegen eines geänderten Wertes für style nun zusätzlich erforderlich sind. Wurde zwischenzeitlich ein anderes Paket zum Formatieren des Inhaltsverzeichnisses geladen, dann kann mit tocstyle= oder tocstyle={} verhindert werden, dass dessen Definition des Inhaltsverzeichniseintrags von KOMA-Script wieder zurück geändert wird.

Auswirkungen der Option tocstyle

Die meisten Stile für die Eintragungen ins Inhaltsverzeichnis benötigen die Vorgabe des Einzuges und der zu reservierenden Nummernbreite im Inhaltsverzeichnis. Dazu gehören auch tocline und der davon abgeleitete Stil default. Deshalb wurden die beiden Eigenschaften tocindent und tocnumwidth oben bei den notwendigen Einstellungen für Gliederungsebenen mit angegeben.

Allerdings gibt es für tocline und alle darauf basierenden Stile noch eine Ausnahme: Wird ein Gliederungsbefehl mit dem Präfix »sub« definiert und existiert bereits ein Befehl ohne dieses Präfix mit Angaben für die beiden Optionen, dann werden aus diesen automatisch passende Werte für die neue Überschrift ermittelt. In diesem speziellen Fall kann also die Angabe von tocindent und tocnumwidth entfallen.

Darüber hinaus kennen tocline und dessen Klone etliche weitere Eigenschaften, die mit dem Präfix »toc« versehen im optionalen Argument von zum Beispiel `\DeclareNewSectionCommand` verwendet werden können, um den Stil der Inhaltsverzeichniseinträge an eigene Wünsche anzupassen. Wählt man den Stil targetocline,

dann kann keine Nummernbreite für das Inhaltsverzeichnis angegeben werden. Und für den Stil `gobble`, der den Eintrag im Prinzip »verschluckt«, dürfen weder `tocindent` noch `tocnumwidth` gesetzt werden.

Mit `tocstyle=` oder `tocstyle={}` wird ein schon vorhandener Befehl für den Inhaltsverzeichniseintrag nicht geändert. Ist dieser Befehl dagegen noch nicht definiert, dann wird wiederum der Stil `default` benutzt. Das ist unter anderem dann relevant, wenn ein anderes Paket zum Formatieren des Inhaltsverzeichnisses geladen wurde, das den Befehl umdefiniert hat. Eine Beschreibung aller Eintragungsstile und Optionen findet sich im Abschnitt »Konfiguration von Verzeichniseinträgen« im Kapitel zu `tocbasic` in [1]. Dabei ist zu beachten, dass die Schlüssel dort ohne das Präfix »`toc`« stehen.

Beispiel: Zusätzliche Gliederungsebene

Angenommen, es wird eine zusätzliche Gliederungsebene `\subsubsection` zwischen `\subsection` und `\paragraph` für die KOMA-Script-Klasse `scrbook` gewünscht, die noch nummeriert und ins Inhaltsverzeichnis eingetragen werden soll. Dann muss einerseits diese neue Überschrift definiert werden und andererseits müssen `\paragraph` und `\subparagraph` angepasst werden. Außerdem muss für die Nummerierung und den Eintrag ins Inhaltsverzeichnis gesorgt werden.

Definition von `\subsubsection`

Die neue Überschrift bekommt den Stil `section` und nimmt in der Hierarchie der Gliederungsebenen den Platz ein, den vorher `\paragraph` inne hatte, also `level=4`. Die Abstände vor und nach der neuen Überschrift sowie die Angaben für `font` und dem wegen `style=section` nötigen `indent` werden hier analog zu denen für `\subsection` gewählt, die in [1] zu finden sind.

Wegen des Präfixes »`sub`« und den vorhandenen Angaben für Einzug und Nummernbreite im Inhaltsverzeichnis für `\subsection` kann auf Einstellungen für `tocindent` und `tocnumwidth` verzichtet werden. Es werden also die folgenden notwendigen Angaben für `\subsubsection` gemacht:

- `style=section`
- `level=4`
- `beforekip=-3.25ex plus -1ex minus -.2ex`
- `afterskip=1.5ex plus .2ex`
- `font=\normalsize`
- `indent=0pt`

Darüber hinaus soll aber noch der Zähler von `\subsubsection` abhängen. Es kommt also noch

- `counterwithin=subsubsection`

hinzu.

Damit ergibt sich als Code für den neuen Gliederungsbefehl:

```
\DeclareNewSectionCommand[
  style=section,
  level=4,
  beforeskip=-3.25ex plus -1ex minus -.2ex,
  afterskip=1.5ex plus .2ex,
  font=\normalsize,
  indent=0pt,
  counterwithin=subsubsection
]{subsubsection}
```

Anpassen von `\paragraph` und `\subparagraph`

Da `\subsubsection` in der Gliederungshierarchie oberhalb von `\paragraph` stehen soll, müssen noch die Werte der Optionen `level` und `toclevel` für `\paragraph` und `\subparagraph` erhöht werden. Zu beachten ist dabei, dass bei einer Änderung von `level` die Option `toclevel` nicht automatisch mit angepasst wird. Es müssen hier also wirklich beide Eigenschaften explizit neu angegeben werden:

- `level=5` für `\paragraph`, `level=6` für `\subparagraph`
- `toclevel=5` für `\paragraph`, `toclevel=6` für `\subparagraph`

Falls eventuell doch noch Einträge ins Inhaltsverzeichnis für die beiden Befehle gewünscht werden, sind außerdem neue Werte für `tocindent` und `tocnumwidth` nötig:

- `tocindent=13em` für `\paragraph`, `tocindent=15em` für `\subparagraph`
- `tocnumwidth=5.9em` für `\paragraph`, `tocnumwidth=6.8em` für `\subparagraph`

Für den Fall, dass diese Überschriftenebenen vielleicht doch nummeriert werden, soll der Zähler für `\paragraph` jetzt natürlich von `\subsubsection` abhängen:

- `counterwithin=subsubsection` für `\paragraph`

Damit ergibt sich als Code für die Änderungen an `\paragraph` und `\subparagraph`:

```
\RedeclareSectionCommand[
  level=5,
  toclevel=5,
  tocindent=13em,
  tocnumwidth=5.9em,
```

```

    counterwithin=subsubsection
  ]{paragraph}
\RedeclareSectionCommand[
  level=6,
  toplevel=6,
  tocindent=15em,
  tocnumwidth=6.8em
]{subparagraph}

```

Anpassen der Zähler `secnumdepth` und `tocdepth`

Welche Gliederungsbefehle nummeriert werden, wird über den Zähler `secnumdepth` bestimmt. Analog gibt der Zähler `tocnumdepth` vor, bis zu welcher Ebene Einträge ins Inhaltsverzeichnis erfolgen. In der Voreinstellung haben beide den Wert 2, so dass die Nummerierung und die Einträge nur bis zur Ebene der Unterabschnitte, also bis `\subsubsection`, erfolgt.

Da die Überschriften bis einschließlich der neu definierten nummeriert und ins Inhaltsverzeichnis eingetragen werden sollen, müssen die beiden Zähler entsprechend neu eingestellt werden:

```

\setcounter{secnumdepth}{\subsubsectionnumdepth}
\setcounter{tocdepth}{\subsubsectionocdepth}

```

Vollständiges Beispiel

```

\documentclass[chapterprefix=on,parskip=half-,DIV=12,fontsize=13pt]{scrbook}
\DeclareNewSectionCommand[
  style=section,
  level=4,
  beforeskip=-3.25ex plus -1ex minus -.2ex,
  afterskip=1.5ex plus .2ex,
  font=\normalsize,
  indent=0pt,
  counterwithin=subsubsection
]{subsubsection}
\RedeclareSectionCommand[
  level=5,
  toplevel=5,
  tocindent=13em,
  tocnumwidth=5.9em,
  counterwithin=subsubsection

```

```

]{}paragraph}
\RedeclareSectionCommand[
  level=6,
  toplevel=6,
  tocindent=15em,
  tocnumwidth=6.8em
]{}subparagraph}
\setcounter{secnumdepth}{\subsubsubsectionnumdepth}
\setcounter{tocdepth}{\subsubsubsectiontocdepth}

```

Schubert-1

```

\tableofcontents
\chapter{Einführung}
\section{Ein Abschnitt}
\subsection{Ein Unterabschnitt}
\subsubsection{Ein Unter-Unterabschnitt}
\subsubsubsection{Ein Unter-Unter-Unterabschnitt}
\paragraph{Der normale Paragraph}
\blindtext
\subparagraph{Der normale Unterparagraph}
\blindtext
\blindeddocument

```

Inhaltsverzeichnis

1	Einführung	3
1.1	Ein Abschnitt	3
1.1.1	Ein Unterabschnitt	3
1.1.1.1	Ein Unter-Unterabschnitt	3
1.1.1.1.1	Ein Unter-Unter-Unterabschnitt	3
2	Oberschrift auf Ebene 0 (chapter)	5
2.1	Überschrift auf Ebene 1 (section)	5
2.1.1	Überschrift auf Ebene 2 (subsection)	5
2.1.1.1	Überschrift auf Ebene 3 (subsubsection)	6
2.2	Listen	6
2.2.1	Beispiel einer Liste (enumerate)	6
2.2.1.1	Beispiel einer Liste (4'terize)	7
2.2.2	Beispiel einer Liste (enumerate)	7
2.2.2.1	Beispiel einer Liste (4'terize)	8
2.2.3	Beispiel einer Liste (description)	8
2.2.3.1	Beispiel einer Liste (4'terize)	8

1

Kapitel 1

Einführung

1.1 Ein Abschnitt

1.1.1 Ein Unterabschnitt

1.1.1.1 Ein Unter-Unterabschnitt

Der normale Paragraph Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huaadest gebuara“? Kjñt – mitsüchten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Der normale Unterparagraph Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huaadest gebuara“? Kjñt – mitsüchten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar

3

Auswirkungen der Änderungen in der Version 3.24

Wie bereits erwähnt, erhalten einige Optionen, die bisher bei der Neudefinition eines Gliederungsbefehls unbedingt nötig waren, ab der Version 3.24 Voreinstellungen. Wird auf diese zurückgegriffen, erhält man einen entsprechenden Hinweis, vorerst wohl als Warnung, in späteren Versionen vielleicht auch nur als Information in der Logdatei. Die ab der Version 3.24 neuen Voreinstellungen für `style=section` sind:

- `font=\normalsize`
- `indent=0pt`

Für die Stile `chapter` und `part` werden ab dieser Version beim Fehlen der folgenden Einstellungen diejenigen der Befehle `\chapter` beziehungsweise `\part` übernommen:

- `beforeskip`
- `afterskip`
- `font`
- `prefixfont`
- `innerskip` (nur `scrreprt` und `scrbook`)
- `pagestyle` (nur `scrreprt` und `scrbook`)

Wird diese Möglichkeit genutzt, dann wirken sich spätere Änderungen der Optionen für `\chapter` beziehungsweise `\part` auch auf den neuen Befehl aus. Damit sind die folgenden Einstellungen bei der Neudefinition einer Gliederungsüberschrift auch weiterhin notwendig:

- `style`
- `level`
- `beforeskip` (wenn `style=section`)
- `afterskip` (wenn `style=section`)

sowie abhängig von `tocstyle` gegebenenfalls noch:

- `tocindent`
- `tocnumwidth`

Im obigen Beispiel könnte man damit bei der Definition des neuen Gliederungsbefehls `\subsubsection` die Einstellungen für `font` und `indent` weglassen, müsste dafür aber mit zwei Warnungen leben.

```
\DeclareNewSectionCommand[
  style=section,
  level=4,
  beforeskip=-3.25ex plus -1ex minus -.2ex,
  afterskip=1.5ex plus .2ex,
  counterwithin=subsubsection
]{subsubsection}
```

Literatur und Software

- [1] Markus Kohm: KOMA-Script ein wandelbares L^AT_EX 2_ε-Paket, 2017, CTAN: /macros/latex/contrib/koma-script/doc/scrguide.pdf (besucht am 06.04.2017).
- [2] – Welches ist die aktuelle KOMA-Script-Release und wo finde ich sie?, <https://komascript.de/release> (besucht am 29.04.2017).

Verzeichnisse ohne neue Umgebung

Markus Kohm

Schon seit vielen Jahren bietet das KOMA-Script-Paket `tocbasic` die Möglichkeit, neue Verzeichnisse anzulegen und zu verwalten. Seit Mitte 2006 wurde dies ab Version 3.06 mit `\DeclareNewTOC` sukzessive ausgebaut. Durch eine kleine Änderung bei den KOMA-Script-Klassen ab Version 3.23 kann damit auch die alte Frage nach getrennten Verzeichnissen für den Anhang neu beantwortet werden.

Neues Verzeichnis, neue Umgebung?

Einigen Anwendern ist die Anweisung `\DeclareNewTOC` bereits bekannt, um neue Gleitumgebungen mit eigenem Verzeichnis ähnlich zu `figure` oder `table` und `\listoffigures` oder `\listoftables` zu definieren. In [1] und [2] findet sich dafür das Beispiel einer Gleitumgebung für Merksätze:

```
\DeclareNewTOC[%
  type=remarkbox,%
  types=remarkboxes,%
  float,% Gleitumgebungen sollen definiert werden.
  floatype=4,%
  name=Merksatz,%
  listname={Verzeichnis der Merksätze}%
]{lor}
```

Dabei wird eine Gleitumgebung (Option `float`) `remarkbox` (Option `type`) definiert. In den Titeln wird vor der Nummer der Name »Merksatz« (Option `name`) verwendet. Das zugehörige Verzeichnis erhält – unabhängig von der aktuellen Sprache – die Überschrift »Verzeichnis der Merksätze« (Option `listname`). Das Verzeichnis ist außer unter dem etwas unhandlichen Namen `\listoflor` oder dem Befehl `\listoftoc{lor}` auch als `\listofremarkboxes` (Option `types`) verfügbar. Die

remarkbox-Umgebungen werden reihenfolgenunabhängig von tabular und figure platziert, aber gegebenenfalls reihenfolgentreu mit anderen Umgebungen mit gesetztem Bit 2 (Option floattype).

Die Anweisung bietet viele weitere Optionen. Beispielsweise kann man den Stil von Verzeichniseinträgen weitreichend verändern. Näheres dazu findet sich in [2], Kapitel 15.

Als Endung für die Hilfsdatei des Verzeichnisses wird .lor verwendet. Eine solche Hilfsdatei benötigt grundsätzlich jedes Gleitumgebungsverzeichnis. So schreibt beispielsweise die Anweisung `\caption` in einer figure-Umgebung über den Umweg der aux-Datei eine `\contentsline`-Anweisung mit der Bildunterschrift in die Hilfsdatei mit der Endung .lof. Diese Datei wird dann bei `\listoffigures` eingelesen und erzeugt so die Einträge des Abbildungsverzeichnisses. Ebenso funktioniert es im Beispiel für die remarkbox-Umgebung – nur eben für die Endung .lor und die Verzeichnisanweisung `\listofremarkboxes`.

Dieses Beispiel aus der Anleitung geht von einem Szenario aus, in dem eine neue Gleitumgebung definiert werden soll. Das Verzeichnis ist sozusagen nur eine Beigabe. Dies entspricht auch der klassischen Denkweise von Anwendern. Bei dieser Betrachtung ähnelt `\DeclareNewTOC` Anweisungen wie `\DeclareFloatingEnvironment` aus dem Paket `newfloat` (siehe [5]). `newfloat` ist Bestandteil von `caption` (siehe [6]). Bei einem genaueren Blick fällt aber auf, dass bei `\DeclareFloatingEnvironment` das obligatorische Argument der Name der Umgebung ist und die Endung der Hilfsdatei als Option angegeben wird. Dagegen ist das obligatorische Argument von `\DeclareNewTOC` die Endung der Hilfsdatei und der Name der Umgebung lediglich eine Option. Dies deutet darauf hin, dass der eigentliche Blick bei `\DeclareNewTOC` nicht auf die neue Gleitumgebung, sondern auf die Hilfsdatei und damit auf das Verzeichnis gerichtet ist. Dies wird noch dadurch verstärkt, dass man die Definition der Gleitumgebung durch Option `float` erst anfordern muss. Auch die Definition eines neuen Zählers erfolgt nur, wenn die Option `float` oder `nonfloat` gesetzt ist.

Anwendung für Verzeichnisse im Anhang

Wie im letzten Abschnitt gezeigt, liegt also die Hauptbedeutung von `\DeclareNewTOC` darin, dass man damit neue Verzeichnisse deklarieren kann. Die Definition neuer Umgebungen ist dabei keineswegs zwingend, sondern eher eine häufig erwünschte Zugabe. Wir könnten beispielsweise mit

```
\DeclareNewTOC[%
  listofname={Verzeichnis der Abbildungen im Anhang}
]{alof}
```

ein neues Verzeichnis definieren, das die Hilfsdatei mit der Endung `.alof` verwendet und unter den Befehlen `\listofalofs` oder `\listoftoc{alof}` verfügbar ist. Will man zusätzlich die etwas leichter zu merkende Anweisung `\listofappendixfigures` haben, muss man in der Deklaration lediglich `type=appendixfigure`, als Option ergänzen.

Einträge in das Verzeichnis könnten mit

```
\addcontentsline{alof}{Nummer und Text des Eintrags}
```

vorgenommen werden. Um jedoch Abbildungen im Anhang tatsächlich automatisch in dieses Verzeichnis, statt in das normale Abbildungsverzeichnis eintragen zu lassen, bedarf es etwas mehr. Einem ersten Impuls folgend, könnte man natürlich in der Deklaration die Option `float` ergänzen und dann im Anhang die Umgebung `appendixfigure` anstelle von `figure` verwenden. Es geht aber noch besser.

Die Umgebung `figure` wird in den \LaTeX -Klassen nämlich über die interne Umgebung `@float` des \LaTeX -Kerns definiert. Diese erhält als obligatorisches Argument lediglich den Namen der Gleitumgebung, also im Beispiel `figure`. Alles, was sich von Gleitumgebung zu Gleitumgebung unterscheidet, wird über Makros festgelegt, die von diesem obligatorischen Argument, also dem Namen der Gleitumgebung, abhängen. Für die Endung der Hilfsdatei ist dies bei Abbildungen `\ext@figure`.

Um ab dem Anhang Abbildungen nicht mehr in das normale Abbildungsverzeichnis, sondern in das zuvor neu deklarierte »Verzeichnis der Abbildungen im Anhang« einzutragen, muss daher also nur `\ext@figure` umdefiniert werden:

```
\makeatletter
\renewcommand*{\ext@figure}{alof}
\makeatother
```

Für Tabellen kann in gleicher Weise verfahren werden. Der fertige Code findet sich unter [3].

Das bisher Gezeigte funktioniert übrigens unabhängig von den KOMA-Script-Klassen auch beispielsweise mit den Standardklassen. Voraussetzung ist jedoch, dass alle Pakete wirklich `\ext@figure` beziehungsweise `\ext@table` verwenden, um Einträge in die entsprechenden Verzeichnisse vorzunehmen. Das ist beispielsweise bei `longtable` bis zur Version 4.11 nicht der Fall. Es gibt dazu aber bereits einen Bug-Report, so dass sich das hoffentlich demnächst ändern wird. Anderenfalls ist unter [3] ein entsprechender Patch zu finden.

Anwendung auf das Inhaltsverzeichnis

Außer für Abbildungen und für Tabellen und gegebenenfalls andere Gleitumgebungen wird häufig für die Gliederungsüberschriften im Anhang ein eigenes Verzeichnis gewünscht. Dieses weitere Verzeichnis kann ebenfalls mit `\DeclareNewTOC` deklariert werden:

```
\DeclareNewTOC[%
  owner=\jobname,
  listname={Inhalt des Anhangs},
]{atoc}
```

Hier fällt auf, dass mit `owner` eine neue Option ins Spiel kommt.

Das Paket `tocbasic` erlaubt es, bestimmte Eigenschaften auf alle ihm bekannten Verzeichnisse oder nur auf die mit demselben `owner` anzuwenden. In der Voreinstellung wird für alle Verzeichnisse `owner=float` verwendet. Die KOMA-Script-Klassen verwenden dies beispielsweise, um die Einstellungen der Option `listof` auf alle Verzeichnisse mit eben diesem `owner` anzuwenden. Wir wollen aber natürlich nicht, dass diese für den »Inhalt des Anhangs« gelten. Daher setzen wir hier willkürlich `\jobname` als `owner`.

Aber natürlich genügt es auch hier nicht, ein neues Verzeichnis zu deklarieren. Es muss ebenfalls dafür gesorgt werden, dass die Einträge statt in die `toc`-Datei in die neue `atoc`-Datei geschrieben werden. Hierfür gibt es anders als für die Hilfsdateien der Gleitumgebungen keine allgemeingültige Lösung. Bei KOMA-Script ab der Version 3.23 verwendet aber nicht nur `tocbasic`, sondern auch alle anderen Pakete und die Klassen das Makro `\ext@toc`, um die Endung der Hilfsdatei festzulegen. Um Gliederungsüberschriften fortan nicht mehr in das normale Inhaltsverzeichnis, sondern in den neuen »Inhalt des Anhangs« einzutragen, genügt daher:

```
\makeatletter
\renewcommand*{\ext@toc}{atoc}
\makeatother
```

Auch dies funktioniert natürlich nur solange `\ext@toc` wirklich verwendet wird. Nimmt man dies selbst oder ein Paket beispielsweise Inhaltsverzeichniseinträge an KOMA-Script nicht per

```
\addchaptertocentry{A}{Beispieleintrag}
```

sondern per

```
\addcontentsline{toc}{chapter}{\protect\numberline{A}Beispieleintrag}
```

vor, so ist KOMA-Script machtlos. Um solche Fälle abzufangen, müsste man zusätzlich `\addcontentsline` umdefinieren. Nur über einen solchen tiefgreifenden Eingriff könnte die gezeigte Lösung auch für Klassen und Pakete adaptiert werden, die `\ext@toc` nicht kennen oder beachten.

Schlussbemerkung

Ich habe mich in diesem Artikel bewusst auf die Erklärung der Grundlagen beschränkt. Wer an der Anwendung derselben in einem vollständigen Beispiel interessiert ist, sei ausdrücklich auf [3] verwiesen. Erweiterungen und Verbesserungen der hier umrissenen Lösung veröffentliche ich nach Bedarf ebenfalls dort. Beispielsweise gibt es bereits eine Lösung für ältere KOMA-Script-Versionen oder eine Erweiterung, um Einträge in das neue Inhaltsverzeichnis automatisch für Bookmarks mit `hyperref` zu nutzen. Des Weiteren können Anwender das KOMA-Script-Forum für Anregungen und eigene Vorschläge nutzen.

Wie bereits im ersten Abschnitt erwähnt, bietet das Paket `tocbasic` viele weitere Möglichkeiten. So könnte man beispielsweise über `\setuptoc` sehr einfach die neuen Verzeichnisse auf einer tieferen Gliederungsebene anordnen und so unter einer gemeinsamen Überschrift »Verzeichnisse des Anhangs« zusammenfassen. Oder man könnte den Stil sämtlicher Einträge über `\DeclareTOCStyleEntry` oder für das Inhaltsverzeichnis wahlweise über `\RedeclareSectionCommand` weitreichend ändern.

Darüber hinaus bietet das Paket `tocbasic` viele weitere Dinge. Als meines Wissens einziges Paket erledigt es bereits seit fast zehn Jahren beispielsweise quasi nebenbei die *Anmeldung* weiterer Verzeichnisse bei `babel` und sorgt dafür, dass Kollisionen bei den per `\DeclareNewTOC` angegebenen Dateiendungen erkannt und gemeldet werden.¹

Über die Jahre habe ich für Anhangsverzeichnisse bereits verschiedene Lösungen vorgestellt. All diese Lösungen sind entweder direkt auf [4] zu finden oder dort verlinkt. Jede hat ihre ganz individuellen Vor- und Nachteile. Die hier umrissene Lösung erscheint mir persönlich derzeit als besonders *elegant* und zugleich verblüffend einfach. Ob sie zukünftig durch weitere Änderungen in KOMA-Script weiter vereinfacht oder ausgebaut werden wird oder es sogar gänzlich neue Lösungsansätze geben wird, hängt von Rückmeldungen und konkreten Anforderungen von Anwendern ab.

¹Übrigens könnten andere Paketautoren durchaus alle Vorteile von `tocbasic` mit wenigen Code-Zeilen nutzen. Stattdessen verwende ich `srchack`, um sie nachträglich und daher umständlich einigen wenigen Paketen beizubringen.

Literatur und Software

- [1] Markus Kohm: KOMA-Script, Eine Sammlung von Klassen und Paketen für \LaTeX 2_ε, 5. Aufl., DANTE und Lehmanns Media, Heidelberg und Berlin, 2014, ISBN: 978-3-86541-613-1.
- [2] — KOMA-Script, Ein wandelbares \LaTeX 2_ε-Paket, 2017, CTAN: /macros/latex/contrib/koma-script/doc/scrguide.pdf (besucht am 01. 04. 2017).
- [3] — Mit einem aktuellen KOMA-Script geht alles viel einfacher, 2017, <https://komascript.de/comment/5578#comment-5578> (besucht am 31. 03. 2017).
- [4] Markus Kohm, Freunde: KOMA-Script Documentation Project, <https://komascript.de> (besucht am 17. 04. 2017).
- [5] Axel Sommerfeldt: newfloat -- Define new floating environments, Version 1.1, 2013, CTAN: /macros/latex/contrib/caption (besucht am 17. 04. 2017).
- [6] — caption -- Customising captions in floating environments, 2016, CTAN: /macros/latex/contrib/caption (besucht am 17. 04. 2017).

Strukturbäume für Kategorialgrammatiken

Christine Römer

Problemstellung

Der Terminus Kategorialgrammatik bezeichnet in der Linguistik eine Erklärung von Sätzen, bei der den Komponenten syntaktische Kategorien zugeordnet werden. Sie will nur die logische Form erfassen und stellt in ihrer klassischen Form eine Rekognitionsbeschreibung (Anerkennungsbeschreibung) hinsichtlich der Wohlgeformtheit dar. Die Bedeutungen komplexer Ausdrücke werden nach dem »Frege-Prinzip« aus den Bedeutungen der Teilausdrücke ermittelt, das heißt kompositionell.

Auf der Satzebene werden zwei semantische Grundkategorien – Name und Satz – unterschieden, die sich nach dem polnischen Logiker Ajdukiewicz (1935) durch den Sinn der Wörter ergeben: Namenkategorien (n) bedürfen keiner Ergänzung. Satzkategorien (s) beinhalten im Regelfall Argumente und Funktoren (Prädikate) und haben, wenn sie wohlgeformt sind, einen Wahrheitswert (genauer siehe beispielsweise [4, Kap. 7] oder [3]). Weiterentwicklungen der klassischen Kategorialgrammatik nehmen andere Basiskategorien an, beispielsweise »e« (bezeichnen Entitäten) und »t« (ihnen können Wahrheitswerte zugeordnet werden) [11, Kap. E2 1].

Es ist üblich, die Beschreibungen mittels kategorialgrammatischen syntaktischen Strukturbäumen zu verdeutlichen, wie nachfolgend: Eine zweistellige Funktorkategorie (so ein einstelliges Verb) verbindet sich mit einem nominalen Ausdruck zu einem wohlgeformten Ausdruck (wie *Helga lacht*).

Roemer-1

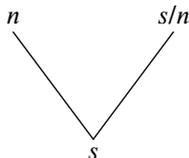


Abb. 1: Einfacher kategorialer Analysebaum

Es handelt sich dabei um »auf dem Kopf stehende« Strukturbäume, wofür es meines Erachtens bisher kein entsprechendes \TeX -Makro gibt. Nachfolgend soll vorgestellt werden, wie ich dies in meinen Arbeiten versucht habe zu lösen. Allen Varianten ist gemeinsam, dass viel händige Anpassung nötig war. Ein neuer Versuch wurde durch das instruktive und gut lesbare Einführungsbuch von Walter Entenmann »METAPOST– Grafik für \TeX und \LaTeX « in der *dante*-Publikationsreihe angeregt, weshalb ich nur darauf etwas genauer eingehen werde. [1]

»Auf dem Kopf stehende Bäume« mit \LaTeX erzeugen

Mit `xytree` [9]

Roemer-2

```
\PassOptionsToPackage{dvips}{xy}% latex|dvips|ps2pdf benutzen!
\usepackage{xytree}

\xytree{%
  \xynode{\shortstack{Leider\\ s/s}} \xyconnect[-](D,U){4,4}
  & & \xynode{\shortstack{rauscht\\ s/n}}
  \xyconnect[-](D,U){3,3} & &
  \xynode{\shortstack{das\\ n/n}} \xyconnect[-](D,U){2,2} & &
  \xynode{\shortstack{blaue\\ n/n}} \xyconnect[-](D,U){1,1}
  & & \xynode{\shortstack{Meer\\ n}} \xyconnect[-](D,U){1,-1} \\
  & & & & \xynode{n} \xyconnect[-](D,U){1,-1} \\
  & & & & \xynode[-1]{n} \\
  & & & & \xynode{s} \xyconnect[-](D,U){1,-1} \\
  & & & \xynode{s $\surd$}%
}
```

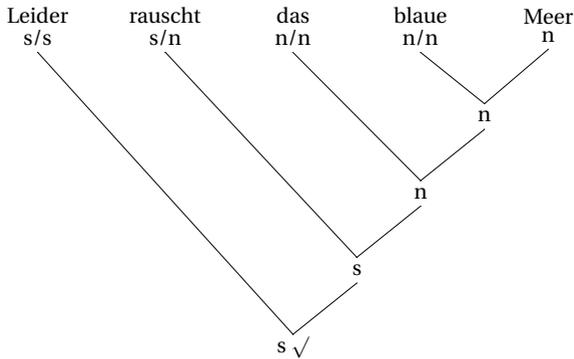


Abb. 2: Kategoriale Analyse mit xytree

Das Paket xytree hat vor allem das Problem, dass die Striche in den Grafiken nicht richtig durchgezogen (»zitterig«) sind. Wenn man sich in den Konstruktionsmechanismus eingearbeitet hat, kann man eigentlich gut damit arbeiten.

Mit der Umgebung psmatrix [8]

```
\usepackage{auto-pst-pdf}%% pdflatex -shell-escape <Datei>
\usepackage{pst-node}
```

```
\begin{psmatrix}[colsep=0.5cm,rowsep=20pt,mcol=c]
Facebook & vermarktet & die & Beiträge & seiner & Nutzer\\[-20pt]
n & s/nn & n/n & n & n/nn & n \\
& & n\psspan{2} & & n/n\psspan{2} & \\
s/n\psspan{2} & & n\psspan{2} & & & \\
& & s & & & \\
\psset{nodesep=3pt}
\ncline{2,1}{4,1} & \ncline{2,2}{4,1}
\ncline{4,1}{5,3} & \ncline{2,3}{3,3}
\ncline{2,5}{3,4} & \ncline{2,6}{3,4}
\ncline{2,4}{3,3} & \ncline{3,3}{4,3}
\ncline{4,3}{3,4} & \ncline{4,3}{5,3}
\end{psmatrix}
```

Roemer-3

Bei der Umgebung psmatrix, welche in pst-node von PSTricks definiert ist, besteht das Problem, dass man »basteln« muss, damit die Linien einander symmetrisch zugeordnet werden. Es können damit jedoch relativ schnell auch komplexere Beispiele erzeugt werden.

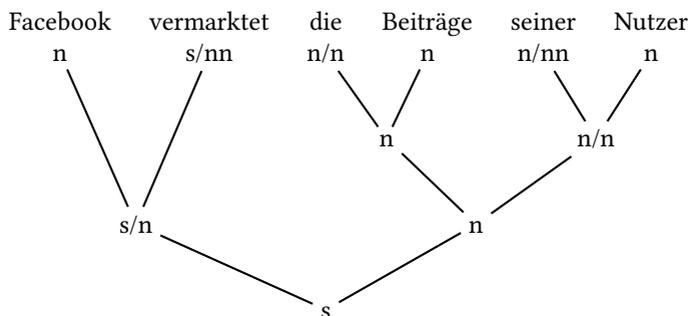
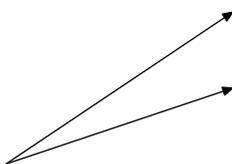


Abb. 3: Kategoriale Analyse mit PSTricks

METAPOST als Alternative

»METAPOST ist eine grafische Programmiersprache und ein Compiler/Interpreter zur Erstellung von quantitativ hochwertigen Grafiken im technisch-wissenschaftlichen Bereich.« [1, S. 1]. Man kann damit eps-Vektorgrafiken erzeugen (mp-Dateien), die gegebenenfalls in andere Formate konvertiert werden können. Die aus dem METAPOST-Quellcode entstandenen mp-Dateien und auch der mp-Code können in tex-Dateien in verschiedener Weise eingebunden und weiterverarbeitet werden. Für die Erzeugung der mp-Dateien und ihre Einbindung gibt es etliche Pakete.

Die folgende einfache mp-Grafik kann über den angegebenen Code mit dem Konvertierungsprogramm `mptopdf` `datei.mp` erstellt werden:



```

%% datei.mp
beginfig(1)
  pair A,B;
  A:=(3cm,2cm); B:=(3cm,1cm);
  drawarrow origin--A;
  drawarrow origin--B;
endfig;
end;

```

Die erzeugte Datei `datei-1.pdf` kann dann auf übliche Weise in eine tex-Datei mit `\includegraphics[]{ }` eingefügt werden.

Mit dem Programm `LuaTeX` erlaubt das Paket `luamplib` die direkte Integration von METAPOST-Code. [10, Kap. 7.2]

Mit metaobj

Mit dem METAPOST-Paket metaobj können unter anderem über Boxen verschiedenartige Bäume erzeugt werden (siehe [6] und [7, Kap. 7.5]). In »The \TeX Graphics Companion« [2] findet man dazu auch das Beispiel 3-5-69 (CTAN:/info/examples/lgc2/).

Es kann nun in die Umgebung

```
\begin{mplibcode}
  METAPOST-Code
\end{mplibcode}
```

eingebunden werden (wie im nachfolgenden Listing) und mit dem Compiler `lualatex -shell-escape` die PDF-Abbildung 4 abgeleitet werden:

```
\usepackage{luamplib}% lualatex <Datei>
```

```
\everymplib{ beginfig(0);} \everyendmplib{ endfig;}
\begin{mplibcode}
input metaobj
newBox.a(btex apples\strut etex);
newBox.b(btex oranges\strut etex);
newBox.c(btex bananas\strut etex);
newBox.d(btex potatoes\strut etex);
newBox.e(btex peas\strut etex);
newBox.f(btex fruits\strut etex);
newBox.v(btex vegetables\strut etex);
newBox.fo(btex food\strut etex);
warningcheck:=0;
newTree.fruits(f)(a,b,c) "Ualign(bot)", "treemode(U)";
newTree.vegetables(v)(d,e) "Ualign(center)", "treemode(U)";
newTree.food(fo)(fruits,vegetables) "hsep(1cm)", "treemode(U)";
scaleObj(food, .7);
food.c=origin;
drawObj(food);
\end{mplibcode}
```

Roemer-4

Im Standardfall werden mit dem Paket metaobj die Äste als Pfeile gesetzt, dies ist jedoch abänderbar, wie auch die Liniendicke, indem man an den Beginn der Grafikdefinition beispielsweise Folgendes einfügt:

```
setCurveDefaultOption("arrows")("-");
setCurveDefaultOption("linewidth")(.6pt);
```

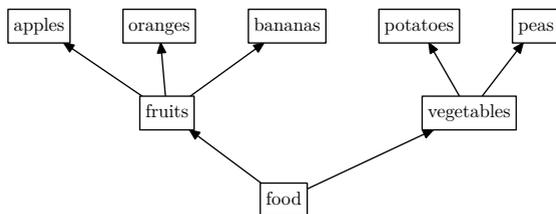


Abb. 4: Strukturbaum mit metaobj

Die Richtung des Baumaufbaus wird mit dem Befehl `treemode(U)` umgekehrt. Die Einrahmungen von Knotennamen lassen sich durch die Angabe von `framed(false)` bei der jeweiligen Box unterdrücken. Mit dem Befehl `\shortstack{... \\\...}` können die Labels in der ersten Zeile als einspaltige Tabelle eingefügt werden. Für die »schöne« Anordnung der Labels sorgen die `strut`-Befehle (siehe Beispiel 5, das von Beispiel 4 abgeleitet ist). Mit dem Paket `luamplib` und der Umgebung `mplibcode` kann METAPOST nativ, wie schon erwähnt, direkt mit `lualatex` verwendet werden. Das Paket `luamplib` kommt ebenfalls bei dem nachfolgend angesprochenen METAPOST-Paket `mptrees` zum Einsatz.

```

\usepackage{luamplib}% lualatex <Datei>
\newcommand\gStrut{\vphantom{Hg}}
\newcommand\sStrut{\vphantom{/}}

```

Roemer-5

```

\everymplib{ beginfig(0);} \everyendmplib{ endfig;}
\begin{mplibcode}
input metaobj
newBox.he(btex \shortstack{Helga\n\sStrut} etex) "framed(false)";
newBox.lie(btex \shortstack{liest\gStrut\s\nn} etex) "framed(false)";
newBox.sa(btex s/n etex) "framed(false)";
newBox.ei(btex \shortstack{ein\gStrut\n/n} etex) "framed(false)";
newBox.bu(btex \shortstack{Buch\gStrut\n\sStrut} etex) "framed(false)";
newBox.sb(btex n\sStrut etex) "framed(false)";
newBox.wurzel(btex s etex) "framed(false)";
%
warningcheck:=0;
setCurveDefaultOption("arrows")("-");
setCurveDefaultOption("linewidth")(.6pt);
newTree.satz(sa)(he,lie)"treemode(U)";
newTree.satz(sb)(ei,bu)"treemode(U)";
newTree.satz(wurzel)(satz,satzb) "treemode(U)";
satz.c=origin;

```

```
drawObj(satz);
\end{mplibcode}
```

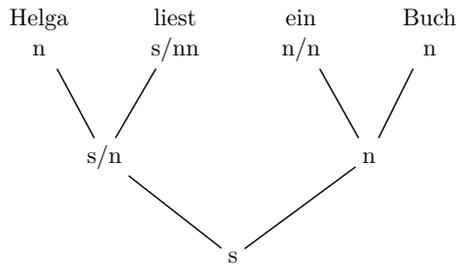


Abb. 5: Kategoriale Analyse mit metaobj

Mit mptrees

Einfache Beispiele, wie die folgenden in Abb. 6 bis 8, kann man mit dem aktuellen METAPOST-Paket `mptrees` leicht erzeugen, das auf dem Paket `latexmp` aufbaut und von ihm mitgeladen wird [5, S. 1]. Man kann die Strukturdarstellungen auch mit $\text{Lua}\text{\LaTeX}$ unter Einbeziehung des Pakets `luamp1ib` erzeugen. Letzteres ist empfehlenswerter, da bei der Kompilierung mit `pdf\text{\LaTeX}` zusätzlich `m\text{\LaTeX}` aufgerufen werden muss und nach fehlerbehafteten Aufrufen neben der Modifizierung der `tex`-Datei die Hilfsdateien vor einem erneuten Kompilieren gelöscht werden müssen. Zum Kompilieren siehe [5, Kap. 4].

Für den angestrebten Zweck ist es sehr günstig, dass es im Paket `mptrees` die Option `dirtree` gibt, die es ermöglicht, Bäume zu drehen. Dabei ändert sich natürlich auch die Reihenfolge der Ebenen: Aus der ersten wird die letzte, etc. Nach dem Befehl `draw tree` kommt, wie im Listing für die Abbildung 6 zu sehen ist, die Ebene in eckigen Klammern, danach in runden Klammern die Höhe und Spreizungsbreite der Äste. Danach folgt die Knotenbeschriftung (Label) und eine mögliche Beschriftung auf dem Ast, die hier leer bleibt, weil sie nicht benötigt wird.

Ein Vorzug des Pakets ist, dass die Symmetrie der Ästespreizung vom Paket hergestellt wird. Eine Schwierigkeit besteht bei `mptrees` jedoch darin, dass die Länge der Äste für jeden beschrifteten Knoten ausprobiert werden muss, um eine diesbezügliche Längensymmetrie herzustellen.

Mit der Option `scaleev:=` kann die Größe der Labels verändert werden. Mit `edgearrow:=true` können die Äste mit einem Endpfeil versehen werden und mit

endedgeshifft kann vertikaler Platz an das Ende der Äste eingefügt werden (zu den weiteren Optionen siehe [5]).

Roemer-6

```
\usepackage{luamplib}%% lualatex <Datei>

\everymplib{beginfig(1);} \everyendmplib{endfig;}
\begin{mplibcode}
input mptrees;
dirtree:=90;
draw startlabel("s");
draw tree[1][1](1cm,2.6cm)("n",,,,,,"s/n",,,,,); %% Höhe Breite
draw tree[2][1](1.3cm,1.2cm)("n/n",,,,,,"n",,,,,);
draw tree[2][2](1.2cm,1.5cm)("s/nn",,,,,,"n",,,,,);
\end{mplibcode}
```

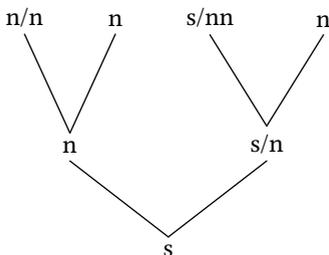


Abb. 6: Kategoriale Analyse mit mptrees

Das analysierte Textbeispiel kann man, wie in der Abb. 7, über den Befehl `draw endlabel` in das Bild einbringen. Man kann auch einzelne Arme verlängern, muss die Längen jedoch durch Ausprobieren anpassen. Dies trifft auch auf die Abstände des Analysebeispiels zu den Knoten zu.

Roemer-7

```
\usepackage{luamplib}%% lualatex <Datei>

\begin{mplibcode}
input mptrees;
dirtree:=90;
abscoord:=true;
endlabellspace:=0.1cm;
beginfig(1)
draw startlabel("n");
draw tree[1][1]((2.4cm,1.5cm),(1cm,-1.5cm))("n/n",,"$$$","n\phantom{/}",,"$$$");
draw tree[2][2](2.4cm,1cm)("n\phantom{/}",,"$$$","n/n",,"$$$");
draw endlabel[2][1]("das");
\end{mplibcode}
```

```
draw endlabel[3][1]("gute");
draw endlabel[3][2]("Buch");
endfig;
\end{mplibcode}
```

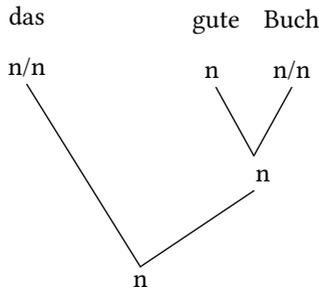


Abb. 7: Kategoriale Analyse mit Beispileinfügung

Beim abschließende Beispiel (Abb. 8) handelt sich um eine Abwandlung des Beispiels 22 aus [5], welches wiederum ein berühmtes Analysebeispiel von N. Chomsky ist. Um die Anpassung der Äste »feiner« vornehmen zu können, wurde vom Paketautor die Längeneinheit $u:=0.4\text{cm}$ als Ausgangspunkt eingeführt.

```
\usepackage{luamplib}%% lualatex <Datei>
```

```
\everymplib{beginfig(1);} \everyendmplib{endfig;}
\begin{mplibcode}
input mptrees;
u:=0.4cm;
dirlabel:=90;
abscoord:=true;
endlabelfspace:=0.2cm;
draw startlabel("s");
draw tree[1][1]((-5.5u,4u),(5.5u,8u))("n",,,,,,"s/n",,,,,);
draw tree[2][1]((-8.5u,12u),(-2.5u,8u))("n/n",,,,,,"n",,,,,);
draw tree[2][2]((3.5u,12u),(7.5u,12u))("s/n",,,,,,"s/n/s/n",,,,,); % 2. Zeile
draw tree[3][2]((-4.5u,12u),(-0.5u,12u))("n/n",,,,,,"n\phantom{/}",,,,,);
draw endlabel[3][1]("Colorless");
draw endlabel[4][1]("green");
draw endlabel[4][2]("ideas");
draw endlabel[3][3]("sleep");
draw endlabel[3][4]("furiously");
\end{mplibcode}
```

Roemer-8

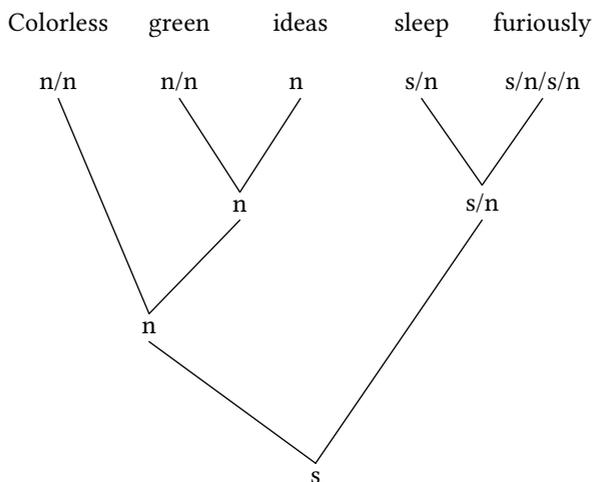


Abb. 8: Komplexer kategorialer Baum mit Analysebeispiel

Fazit

Das im Beitrag besprochene Objekt verdeutlicht, dass es sich für knifflige grafische Aufgaben lohnt, sich in METAPOST einzuarbeiten und es anzuwenden. Dies ist durch die dafür geschaffenen Pakete (wie `luamplib`, `metaobj` und `mptrees`) deutlich einfacher geworden.

Alle Beispiele findet man als eigenständige und lauffähige Programme unter <http://www.dante.de/DTK/Software.html>

Literatur und Software

- [1] Walter Entenmann: METAPOST, Grafik für \TeX und \LaTeX , Lehmanns Media, Berlin und Heidelberg, 2017, <http://www.dante.de/index/Literatur/MP.html>.
- [2] Michel Goossens u. a.: The \LaTeX Graphics Companion, 2. Aufl., Pearson Education, Boston, 2008, http://ptgmedia.pearsoncmg.com/images/9780321508928/.../0321508920_Sample.pdf.
- [3] Klaus von Heusinger: Kategoriale Unifikationsgrammatik, <https://ling.uni-konstanz.de/pages/publ/PDF/ap037.pdf>.
- [4] Horst Lohnstein: Formale Semantik und natürliche Sprache, 2. Aufl., De Gruyter, Berlin und New York, 2011.

- [5] Olivier Péault: Documentation of `mptrees.mp`, Version 17.06, 2017, CTAN : [/graphics/metapost/contrib/macros/mptrees/mptrees/mptrees-doc.pdf](http://graphics.metapost/contrib/macros/mptrees/mptrees/mptrees-doc.pdf).
- [6] Denis Roegel: »METAOBJ: Very high-level objects in METAPOST«, *TUGboat*, 23.1 (73 2002), TUG 2002 Proceedings (Trivandrum), 93–100, <http://www.tug.org/TUGboat/tb23-1/roegel.pdf>.
- [7] — The METAOBJ tutorial and reference manual, 2002, CTAN: [/graphics/metapost/contrib/macros/metaobj/doc/momanual.pdf](http://graphics/metapost/contrib/macros/metaobj/doc/momanual.pdf).
- [8] Christine Römer: Dokumentation Schematische Übersichten mit `psmatrix`, CTAN: [/graphics/pstricks/contrib/pst-node/doc/](http://graphics.pstricks/contrib/pst-node/doc/).
- [9] Koaunghi Un: `xytree.sty` \LaTeX package for drawing syntactic trees, Version 1.5, 2006, CTAN: [/macros/latex/contrib/xytree](http://macros/latex/contrib/xytree).
- [10] Herbert Voß: Einführung in Lua \TeX und Lua \LaTeX , Lehmanns Media, Berlin und Heidelberg, 2013, <http://www.dante.de/index/Literatur/EinfuehrungInLuaTeX.html>.
- [11] Gisela Zifonun, Ludger Hoffmann, Bruno Strecker: Grammatik der deutschen Sprache, De Gruyter, Berlin und New York, 1997.

Ein Workflow für ein Sachbuch

Rainer-Maria Fritsch

Im Folgenden möchte ich *einen* Workflow für ein Sachbuch beschreiben. Dieser bestimmte die Entstehung meines ersten Sachbuchs, wohl wissend, dass es auch andere Wege gibt, so ein Projekt zu bewältigen. Ein Sachbuch zu veröffentlichen, braucht ein gutes Projektmanagement. Manches war vorher schon strukturiert, manches hat sich erst aus Fehlern und iterativen Verbesserungen während des Schreibens ergeben. Das ganze Projekt beruhte auf ehrenamtlicher Arbeit von vielen Beteiligten. An dem Sachbuch waren eine deutsche Herausgeberin, vier Übersetzerinnen vom Russischen ins Deutsche, bis zu drei Lektoren, ein Buchdesigner und kurz vor Druck nochmals vier Korrekturleser beteiligt. Fotos von professionellen Fotografen und viele Bilder aus Familienarchiven mussten gesichtet und für das Buch ausgewählt und bearbeitet werden. Grundlage war eine belarussische Fassung, die für die deutschen Leserinnen und Leser nicht nur übersetzt, sondern auch editiert und teilweise neu zusammengestellt werden musste. Am Ende wurde ein Buch im Format A5 mit rund 520 Seiten, 260 Abbildungen, davon 140 farbig,

in den Druck gegeben. Ich hatte die Ehre – und gelegentlich die leichte Überforderung ;-) – Projektmanager, Korrektor, Lektor, Bildbearbeiter und Setzer zu sein (vgl. auch [10]).

Sich selbst Projektregeln geben

Mit dem Schreiben verständlichen Codes, dem Nachdenken über Namensgebung und dem Kommentieren verhält es sich ein bisschen wie mit Safer Sex: Jeder erkennt den Nutzen und ist dafür, und doch wird es selten praktiziert – insbesondere wenn es wirklich darauf ankommt. [6]

Das Projekt sichern

Bevor ein Projekt startet, sollte bereits festgelegt sein, wo und wie das Projekt abgelegt und die entstehenden Dateien gesichert werden. Die meisten Fehler treten auf, wenn man noch schnell etwas ändern will, obwohl man bereits sechs oder mehr Stunden vor dem Rechner gesessen hat. Oder man bekommt von Projektbeteiligten verschiedene Zuarbeiten, die man falsch zuordnet, oder schlimmer noch, bereits vorhandenen Code oder gleich ganze Dateien unwiederbringlich überschreibt.

Ich benutze als Versionskontrollsystem Git [7] und eigene Backupregeln [9, 5]. Versionskontrollsysteme sind keine Hexerei [11]. Da private Accounts auf `github` kostenpflichtig sind und es auch nur eine Person gab, die alle Dateien bearbeitete und verwaltete, habe ich mich für Git ausschließlich lokal auf meiner Festplatte entschieden. In Projekten mit mehreren Autoren, die alle am Quelltext arbeiten, kann ein privater Git-Account sicherlich nützlich sein. Es gibt verschiedene grafische Tools für Git.¹ Ich benutze auf dem Mac die kostenpflichtige App Tower.² Die App ist intuitiv zu bedienen, und es lassen sich verschiedene Repositorien sehr übersichtlich verwalten.

Die beste Versionskontrolle nützt jedoch nichts, wenn diese nicht ständig benutzt wird. Deshalb sollte man vor jedem Arbeitsbeginn Git aufrufen. So stellt man sicher, dass man das am vorherigen Arbeitsende nicht vergessen hatte.³ Beim nächsten Arbeitsende wiederum Git starten lassen. Das geht einerseits per Kommandozeile, andererseits per grafischem Frontend. In Tower sieht man beim Öffnen des Repository gleich, was alles geändert wurde und noch in einen Commit gesichert werden muss. [8] Mein Lieblingseditor TextMate 2 zeigt mir im Datei-Browser mit

¹ Eine Übersicht findet sich hier: <https://git-scm.com/download/gui/linux>

² <https://www.git-tower.com/mac/>, auch für Windows.

³ Nach den Mühen eines langen Tages schleicht sich schon mal eine Nachlässigkeit ein: »Ach, das mache ich gleich morgen früh.« Das gilt es zu vermeiden.

entsprechenden Icons an, wenn eine \LaTeX -Datei noch nicht in die Versionierung aufgenommen wurde. [4]

Da Git alle Dateien in die Versionierung aufnehmen kann, können auch Fotos und Grafiken in die Versionierung aufgenommen werden. Daher hat es sich als sehr hilfreich erwiesen, vor größeren Arbeitspaketen – wie zum Beispiel der stapelweisen Bearbeitung von Fotos – ebenfalls Git aufzurufen und den aktuellen Arbeitsstand zu sichern. Mir ist einmal beim Export von Fotos ein schlimmer Fehler passiert und viele bereits bearbeitete Bilder wurden überschrieben. Die vorherige Version konnte fix über Git wieder hergestellt und der Export der Fotos, nun mit den richtigen Dateinamen, wiederholt werden. Vor jeder großen Änderung, und dazu gehört auch die Bearbeitung von mehreren Fotos oder Grafiken, sollte man einmal sicherstellen, dass die Versionierung tatsächlich erfolgt ist, in dem man Git auf der Kommandozeile oder per grafischem Frontend vorher noch einmal aufruft – sicher ist sicher.

Backups erfolgen bei Speicherung einer Datei auf der lokalen Festplatte automatisch zugleich in der Cloud (auf einem eigenen Server mit einer ähnlichen Funktionalität wie Dropbox), täglich zusammen mit dem Git-Repository auf eine externe Festplatte und zusätzlich auf dem Webspace meines Internet-Service-Providers. Dazu benutze ich ein Tool, das zeitgesteuert das Projektverzeichnis mit den externen Laufwerken synchronisiert.⁴ Insgesamt liegt das ganze Projekt auf vier verschiedenen physikalischen Datei-Systemen. Automatisierte nächtliche Backups der gesamten lokalen Festplatte sind darüberhinaus selbstverständlich (Vgl. auch [6, Kapitel 12 und 21]).

Dateikonventionen

Das ganze Projekt zu erstellen hat etwa 10 Monate gedauert. Es hat sich dabei angeboten, das Projekt nach den Kapiteln zu strukturieren. Im Laufe des Prozesses kann es zu einer Umstellung, Zusammenfassung oder Trennung von Kapitel kommen. Auch dann müssen die folgenden eigenen Regeln konsequent umgesetzt werden. Ein Editor, der projektübergreifendes Suchen und Ersetzen beherrscht (vielleicht sogar mit regulären Ausdrücken [3]), ist dabei von großem Vorteil.

Jedes Kapitel wird in einer eigenen Datei in einem eigenen Verzeichnis gespeichert. Die dazugehörigen Bilder und Grafiken werden in einem Unterverzeichnis `nn_fotos` abgelegt. Hier befinden sich zwei weitere Unterverzeichnisse: `nn_originale` für die Bilder/Grafiken in beliebigen Datei-Formaten und `nn_bearbeitet` für bearbeitete Bilder/Grafiken, diese nur im Format JPG oder PDF. Nur aus diesem Verzeichnis wird das Bildmaterial in das zugehörige \LaTeX -Dokument eingebunden. Die Verzeichnis- und Dateinamen beginnen mit führender zweistelliger Nummer entsprechend der

⁴ Sync Folders Pro <http://www.greenworldsoft.com/sync-folders-pro-help.php>, ca. 8 €.

Kapitelnummer, Kapitelnummern kleiner 10 mit führender Null. Die Nummern 00 und 70–90 sind reserviert für Vor- und Nachspann, Buchumschlag und druckfertiges PDF.

Alle Verzeichnis- und Dateinamen werden ausschließlich klein und ohne Umlaute geschrieben. Zur besseren Lesbarkeit wird der Unterstrich als Trenner in Verzeichnis- und Dateinamen genutzt. Ziel ist es »sprechende« Dateinamen zu haben, um eine gute Übersicht zu behalten. Innerhalb des \LaTeX -Dokuments hat es sich als sinnvoll erwiesen, den einzelnen Gliederungsebenen eindeutige Präfixe bei den Labeln zu vergeben wie beispielsweise:

```
\label{cha:kapitelname}    % Kapitel
\label{sec:abschnitt}     % Abschnitt
\label{sub:unterabschnitt} % Unterabschnitt
```

Das Ende einer Gliederungsebene wird ebenso eindeutig gekennzeichnet:

```
% chapter kapitelname (end)
% section sectionname (end)
% subsection:subsectionname (end)
```

Das erhöht die Übersichtlichkeit in langen Dateien, und Editoren, die Codefaltung beherrschen, können darauf zurückgreifen.

Label für Bilder/Grafiken werden ebenfalls mit einem Präfix für Farbe bzw. schwarz/weiß versehen (`\label{color:fig:}` oder `\label{swfig:}`). Das erleichtert das Zählen der farbigen Abbildung für die Berechnung der maximal zu erwartenden Druckkosten ungemein – soweit der verwendete Editor projektübergreifendes Suchen in allen \LaTeX -Dateien ermöglicht. [4]

Der Verzeichnisbaum

Die Projektstruktur ließe sich auch in einer anderen Verzeichnisstruktur abbilden. Bewährt hat sich jedoch, konsequent bei der Nummerierung der Kapitelverzeichnisse und entsprechend der Unterverzeichnisse für das Bildmaterial zu bleiben. Ergeben sich während der Arbeit am Sachbuch neue Reihenfolgen der Kapitel und/oder der Zuordnung des Bildmaterials, müssen die Dateien, Verzeichnisse und Referenzierung konsequent entsprechend der neuen Zuordnung umbenannt werden. Auch nur eine kurzzeitige Disziplinlosigkeit kann schnell ins Chaos führen.

```

/
├── projektverzeichnis.....Hauptverzeichnis des Projekts
│   ├── jahr_auflage_mein_sachbuch.tex..... Masterdatei
│   ├── titelei.tex..... Titelei
│   └── typospec.tex..... Typografische Besonderheiten
```

├─ trennungen.tex.....	besondere Trennregeln
├─ acronyme.tex.....	Verzeichnis der Akronyme
├─ 00_Vorspann.....	Verzeichnis
│ └─ 00_vorwort.tex.....	Vorwort
│ └─ 00_danksagung.tex.....	Danksagung
├─ 01_kapitel_name_kurzform.....	Verzeichnis für Kapitel 01
│ └─ 01_kapitel_langform.tex.....	Datei für Kapitel 01
│ └─ 01_fotos.....	Bildmaterial zum Kapitel 01
│ │ └─ 01_originale.....	Bildmaterial – Original, beliebige Formate
│ │ └─ 01_bearbeitet.....	Bildmaterial – bearbeitet, JPG oder PDF
├─ 02_kapitel_name_kurzform.....	Verzeichnis für Kapitel 02
│ └─ 02_kapitel_langform.tex.....	Datei für Kapitel 02
│ └─ 02_fotos.....	Bildmaterial zum Kapitel 02
│ │ └─ 02_originale.....	Bildmaterial – Original, beliebige Formate
│ │ └─ 02_bearbeitet.....	Bildmaterial – bearbeitet, JPG oder PDF
├─ nn_kapitel_name_kurzform.....	Verzeichnis für Kapitel nn
│ └─ nn_kapitel_langform.tex.....	Datei für Kapitel nn
│ └─ nn_fotos.....	Bildmaterial zum Kapitel nn
│ │ └─ nn_originale.....	Bildmaterial – Original, beliebige Formate
│ │ └─ nn_bearbeitet.....	Bildmaterial – bearbeitet, JPG oder PDF
├─ 70_nachspann.....	Verzeichnis für Nachspann: Karten, Danksagung
│ └─ 70_fotos.....	Bildmaterial zum Nachspann
│ │ └─ 70_originale.....	Bildmaterial – Original, beliebige Formate
│ │ └─ 70_bearbeitet.....	Bildmaterial – bearbeitet, JPG oder PDF
├─ 80_umschlag.....	Verzeichnis für Buchumschlag
│ └─ 80_fotos.....	Bildmaterial zum Buchumschlag
│ │ └─ 80_originale.....	Bildmaterial – Original, beliebige Formate
│ │ └─ 80_bearbeitet.....	Bildmaterial – bearbeitet, JPG oder PDF
├─ 90_druck_upload.....	Verzeichnis für druckfertige PDF
│ └─ jahr_auflage_mein_sachbuch_JJ_MM_DD.pdf..	mit Jahr, Monat, Tag der Druckfreigabe
│ └─ jahr_auflage_mein_sachbuch_umschlag_JJ_MM_DD.pdf	mit Jahr, Monat, Tag der Druckfreigabe

Die Masterdatei

Der Dateiname der Masterdatei beginnt mit dem Jahr, gefolgt von der Zahl für die Auflage und dem Kurznamen des Sachbuchs (JJJJ_01_mein_sachbuch). Eine weitere, ggf. korrigierte oder erweiterte Auflage sollte gleich eingeplant werden. Die Titel- (titelei.tex), die typografischen Grundeinstellungen wie Schriften, Anführungs-

zeichen für die deutsche sowie für die benutzten Fremdsprachen, die Gestaltung der Bildunterschriften oder eigene Schriftauszeichnungen (`typospec.tex`), Akronyme (`acronym.tex`) und Worttrennungen (`trennungen.tex`) werden in eigene Dateien ausgelagert und per `input`-Befehl der Präambel hinzugefügt.

In die Datei `typospec.tex` wird auch die von Axel Reichert in [2, 1.8 Nr. 4] empfohlene »moderate Anpassung der Parameter, die \TeX bei der Berechnung der Zeilenumbrüche verwendet«, eingetragen. Diese Parameter habe ich eher zufällig gefunden, sie sind aber für den gesamten Umbruch eines Buches von großem Nutzen.

Es hat sich bewährt, von vornherein vor dem `\begin{document}` den Befehl `\includeonly{}` aufzunehmen. Damit können fertige Kapitel von den Übersetzungsläufen ausgeschlossen und nur die Buchteile gesetzt werden, an denen man gerade arbeitet oder mal wieder einen Fehler sucht. Man spart sich also eine Menge Übersetzungszeit. Wenn das Buch schon einmal komplett übersetzt wurde, bleibt anhand der Hilfsdateien die Nummerierung der Seiten im Allgemeinen erhalten. Das kann hilfreich sein, wenn man zu mehreren vor dem Monitor sitzt und anhand der früheren Ausdrucke nach den Stellen sucht, die überarbeitet werden sollen. Dann folgt der Aufbau des eigentlichen Dokuments.

```
%\includeonly{
% }
\begin{document}
\frontmatter
% \listoftodos % siehe Abschnitt Tools
\maketitle
% Vorwort, Danksagung usw.
\include{00_vorspann/vorwort}
\mainmatter
\include{01_kapitel\01_erstes_kapitel}
% ...
\include{nn_kapitel\nn_letztes_kapitel}
\backmatter
\appendix
\include{acronyme}
\printindex
% ggf. weitere Register
\include{70_nachspann\lastpage} % letzte Seite mit Verweisen auf Webadressen o.ä.
% \listofchanges % siehe Abschnitt Tools
\end{document}
```

Tools

Als äußerst hilfreich hat sich ein Mindmapping-Tool erwiesen.⁵ Man gewinnt bei paralleler Pflege eines Mindmaps einen guten Überblick nicht nur über das Buch, sondern auch über das gesamte Projekt.

Vier \LaTeX -Pakete möchte ich hervorheben, die für einen Workflow für ein Sachbuch sehr hilfreich sind:

- draftwatermark
- todonotes
- changes
- snapshot

Gibt man den aktuellen Stand seines Manuskripts Dritten zu lesen, kann es hilfreich sein, auf der Titelseite oder auf jeder einzelnen Seite ein Wasserzeichen anzubringen. Das Paket `draftwatermark` ermöglicht eine individuelle Gestaltung des Wasserzeichens, z. B. auch zur Kennzeichnung des aktuellen Arbeitstandes.

Das Paket `todonotes` bietet gute Möglichkeiten, notwendige Aufgaben auch farblich unterschiedlich zu markieren und fehlende Bilder auszuweisen. Die zu erledigen Aufgaben werden als Marginalien gesetzt. Auf fehlende Bilder kann mit dem Befehl `\missingfigure` prägnant bereits in den späteren Bildmaßen aufmerksam gemacht werden. Das Paket `changes` bietet nicht nur die Möglichkeit Streichungen, Ersetzungen und Ergänzungen im Text selbst, sondern auch als Liste auszugeben. Den verschiedenen Lektoren/Korrektoren können Namenskürzel und jeweils eine eigene Farbe zugewiesen werden.

Ohne die Pakete `todonotes` und `changes` hätte die Zusammenarbeit mit den Projektpartner nicht so gut funktioniert. Aktuelle Arbeitsstände mussten als PDF oder ausgedruckt ausgeliefert werden, da nur ein Projektmitglied mit \LaTeX umgehen konnte. Beide Pakete bieten die Möglichkeit, Listen der Änderungen und der unerledigten Aufgaben mit auszuliefern. Das ersparte viel E-Mail-Verkehr.

Das letzte Paket leitet bereits in den nächsten Abschnitt *Aufräumen* über. Mit `snapshot` wird ein komplettes Abbild aller Pakete und Dateien in die Datei `*.dep` geschrieben, die man im Kopf seiner Präambel auskommentiert einfügen kann. `snapshot` listet und die verwendeten Klassen- und Paketdateien mit der zuletzt verwendeten Version auf, ebenso alle Dateien mit dem vollständigen Verzeichnispfad. Das Paket `snapshot` zeigt einem sofort, wenn Bilder/Grafiken in einem Kapitel in Wirklichkeit aus einem anderen Verzeichnis/Kapitel entsprechend dem oben skizzierten Verzeichnisbaum stammen. Wer also viel später das Sachbuch noch einmal setzen oder den Quelltext an einen Dritten weitergeben möchte, hat alle notwen-

⁵ Zum Beispiel für alle Betriebssysteme XMind: <https://www.xmind.net/de/>

digen Informationen über die vollständige Datei-Struktur und die verwendeten Klassen- und Paketdateien.

Aufräumen

Code wird ohne Rücksicht auf spätere Wartbarkeit (durch einen selbst oder durch andere) geschrieben. Das liegt in erste Linie an mangelnder Erfahrung. Erst wenn man oft genug vor einem selbst erzeugten undurchdringlichen Codeschungel gestanden hat, fällt es etwas leichter, beim Schreiben an den künftigen Leser zu denken. [6]

Ist man endlich mit seinem Projekt fertig und das Buch im Druck, hat man sicherlich eine Pause verdient. Dennoch wartet noch ein wichtiges Arbeitspaket – *aufräumen*. Bevor Sie das Paket snapshot einsetzen, ist erst einmal aufräumen angesagt:

- Wurden die Datei-Konventionen eingehalten?
- Sind bei der Zusammenlegung oder der Auftrennung von Kapiteln die Verzeichnis- und Dateinamen angepasst worden?
- Versteht jeder hinterher noch, warum bestimmte Pakete mit bestimmten optionalen Parametern aufgerufen wurden?
- Wurden die zugehörigen Pakete und Befehle konsequent in die externen Dateien verlagert?
- Oder ist in der Zwischenzeit ein bisschen »Kuddelmuddel« entstanden?

Ich gebe zu, das ist fast der schwierigste Teil. Gerade bei einem größeren Projekt lernt man viel Neues zu \LaTeX und den vielen hilfreichen Paketen. Man versucht ein Problem während des Projekts zu lösen, recherchiert im Internet dazu und findet letztendlich eine praktikable Lösung. Dabei benutzt man verschiedene Befehle oder Parameter, ohne ausreichend zu kommentieren, warum man das gerade so gelöst hat. Hilfreich ist es auch – wenigstens nachträglich – einen Link, der zur Lösung des Problems geführt hat, an der betreffenden Stelle im Quelltext in den Kommentar aufzunehmen.

Zum Schluss noch einmal Git aufrufen und alles ins Repository schreiben. Git bietet die Möglichkeit über *Tags* diesen Arbeitsstand z. B. als »Druckfreigabe« besonders zu markieren. [1, Nr. 2.6] Der Zeitdruck gegen Ende eines Projekts ist immer groß und die Entspannung nach Abgabe entsprechend tief. Dennoch:

Die Welt wird sicherlich kein schlechterer Ort dadurch, dass Sie stattdessen die allgemeine Unzulänglichkeit von fast allem anerkennen und sich damit begnügen, Code zu schreiben, den Sie selbst nach sechs Monaten noch verstehen können. [6]

Also aufräumen, ggf. neu strukturieren und alles an Kommentierungen nachholen, was man unter Druck und in Zeitnot noch nicht geschrieben hatte. Und dabei sicherstellen, dass der Output immer noch identisch mit dem abgegebenen Werk ist. Um diesen Arbeitsstand einzufrieren, bietet es sich an, das ganze Projekt auf eine DVD zu brennen.

Literatur und Software

- [1] Scott Chacon, Ben Straub: Pro Git, <https://git-scm.com/book/en/v2/Git-Basics-Tagging> (besucht am 14. 06. 2017).
- [2] Marc Ensenbach, Mark Trettin: Das \LaTeX 2 _{ϵ} -Sündenregister oder Veraltete Befehle, Pakete und andere Fehler, CTAN : /12tabu / german / 12tabu . pdf (besucht am 14. 06. 2017).
- [3] Jeffrey E. F. Friedl: Reguläre Ausdrücke, O'Reilly, Heidelberg, 2007.
- [4] Rainer-Maria Fritsch: »TextMate 2 – \LaTeX ing unter MacOS«, *Die \TeX nische Komödie*, 04 (2016), 50–56.
- [5] Gerald Himmelein, Axel Vahldiek, Lutz Labs: »Backup statt Lösegeld – Daten Trojaner-sicher speichern«, *c't – Magazin für Computertechnik*, 11 (2016), 102.
- [6] Kathrin Passig, Johannes Jander: Weniger schlecht programmieren, O'Reilly, Heidelberg, 2013.
- [7] E. Frank Sandig: Versionsverwaltung mit Git – Kurze Einführung am Beispiel von \LaTeX , http://www.suedraum.de/latex/stammtisch/v-git_ps.pdf (besucht am 02. 06. 2016).
- [8] Joachim Schlosser: Git fürs Versionieren von \LaTeX -Dokumenten, <https://www.schlosser.info/git-latex-versionieren/> (besucht am 01. 06. 2017).
- [9] Ingo T. Storm: »3, 2, 1 – ewig deins! So finden Sie die richtige Backup-Strategie«, *c't – Magazin für Computertechnik*, 13 (2013), 112.
- [10] Herbert Voß: Das eigene Buch: Von der Idee zu »Print on Demand«, Lehmanns Media oder selbst den Vertrieb übernehmen, <http://www.dante.de/events/Herbst2016/Programm/Vortraege/VossGoettingen2016.pdf> (besucht am 14. 06. 2017).
- [11] Uwe Ziegenhagen: »Dokumentenmanagement mit \LaTeX und Subversion«, *Die \TeX nische Komödie*, 03 (2008), 43–47.

Von fremden Bühnen

Im Netz gefunden

In den verschiedenen Mailinglisten, Webforen, Newsgroups u. a. findet man immer wieder hilfreiche Angaben zur Arbeit mit und um das Thema Textsatz mit \TeX , \LaTeX , Con \TeX t usw.

Kleine Berechnungen innerhalb eines \LaTeX -Artikels¹

Ausgangspunkt war der folgende Wunsch:

1. ISO Datum plus 283 Tage → Ergebnis soll ein ISO Datum sein, welches die übliche Schwangerschaftsdauer ist.
2. ISO Datum plus 168 Tage → Ergebnis soll ein ISO Datum sein, welches die 24 Schwangerschaftswochen berechnet.
3. ISO Datum plus 168 Tage → Ergebnis sollte das ISO Datum des nächsten Freitags vor oder hinter dem eigentlichen Ergebnis sein (je nachdem welches näher ist).

```
\usepackage{luacode}%% luatex <Datei>
\pagestyle{empty}
\begin{luacode*}
function gbt(refdat)
  local year, month, day = refdat:match("(%d%d%d%d)-(%d%d)-(%d%d)")
  local gebtermin = os.time({day = day + 283, month = month, year = year})
  local t = os.date("!!%Y-%m-%d", gebtermin)
  return t
end

function hpv(refdat)
  local year, month, day = refdat:match("(%d%d%d%d)-(%d%d)-(%d%d)")
  local w24 = os.time({day = day + 168, month = month, year = year})
```

¹ Nikolaus Breuer am 20.4.2017 in <https://www.listserv.dfn.de/sympa/arc/tex-d-1/2017-04/msg00042.html>

```

local VersEnde = os.date("!*t", w24)
local wdshift = {-2,-3,3,2,1,0,-1}
local sh = wdshift[VersEnde.wday]
local korrrhpv = os.time({day=VersEnde.day + sh, month=VersEnde.month, year=
↵VersEnde.year})
local v = os.date("!*Y-%m-%d", korrrhpv)
return v
end
\end{luacode*}
\newcommand\Geburtstermin[1]{\directlua{tex.sprint(gbt("#1"))}}
\newcommand\Versicherungsende[1]{\directlua{tex.sprint(hpv("#1"))}}

```

```

2017-04-29: Geburtstermin \Geburtstermin{2017-04-29}
           Versicherungsende \Versicherungsende{2017-04-29}\par
2017-04-30: Geburtstermin \Geburtstermin{2017-04-30}
           Versicherungsende \Versicherungsende{2017-04-30}\par
2017-05-01: Geburtstermin \Geburtstermin{2017-05-01}
           Versicherungsende \Versicherungsende{2017-05-01}\par
2017-05-02: Geburtstermin \Geburtstermin{2017-05-02}
           Versicherungsende \Versicherungsende{2017-05-02}\par
2017-05-03: Geburtstermin \Geburtstermin{2017-05-03}
           Versicherungsende \Versicherungsende{2017-05-03}\par
2017-05-04: Geburtstermin \Geburtstermin{2017-05-04}
           Versicherungsende \Versicherungsende{2017-05-04}\par
2017-05-05: Geburtstermin \Geburtstermin{2017-05-05}
           Versicherungsende \Versicherungsende{2017-05-05}

```

```

2017-04-29: Geburtstermin 2018-02-06 Versicherungsende 2017-10-13
2017-04-30: Geburtstermin 2018-02-07 Versicherungsende 2017-10-13
2017-05-01: Geburtstermin 2018-02-08 Versicherungsende 2017-10-13
2017-05-02: Geburtstermin 2018-02-09 Versicherungsende 2017-10-20
2017-05-03: Geburtstermin 2018-02-10 Versicherungsende 2017-10-20
2017-05-04: Geburtstermin 2018-02-11 Versicherungsende 2017-10-20
2017-05-05: Geburtstermin 2018-02-12 Versicherungsende 2017-10-20

```

Lua^{La}TeX ist – wie man oben sieht – sicher sehr empfehlenswert für solche kleinen Rechnereien; die ganze »Datumsrechnerei« bringt Lua beispielweise schon mit. Wer Perl beherrscht, kann sich sicher schnell in Lua einarbeiten, sogar auf dem iPad (ich nutze dort Touch Lua+, kostenlos in dem hier benötigten Umfang, mit Tutorials, Wiki, Referenz, ...). Kleine Unterschiede gibt es natürlich zu Perl, beispielsweise zählt Lua array/list/table-Einträge nicht ab 0 wie Perl, sondern ab 1.

APA-Bibliografiestil²

Die ursprüngliche Fragestellung lautete:

Während beim Befehl `\parencite` zwei Autoren durch `&` getrennt werden, werden sie beim Befehl `\textquote` durch ein »und« getrennt. Ich würde das gerne angleichen, sodass zwei Autoren immer durch ein `&` getrennt werden. Beim Befehl `\textcite` wird bei mehr als zwei Autoren der dritte durch ein Komma und ein `&` getrennt. Ich würde das Komma gern verschwinden lassen. Bei mehr als sechs Autoren wird im Literaturverzeichnis zwischen dem sechsten und dem letzten Autor »...« ausgegeben. Ich würde das gerne ändern, sodass alle Autoren aufgeführt werden.

Für die ersten beiden Probleme muss man das Makro `\finalnamedelim` undefinieren.

```
\renewcommand{\finalnamedelim}{\ifnum\value{liststop}>2 \fi\addspace\&\space}
```

Allerdings setzt der APA-Stil das Makro bei `\printbibliography` zurück. Deshalb braucht man zusätzlich

```
\AtBeginBibliography{%
  \renewcommand{\finalnamedelim}{\ifnum\value{liststop}>2 \fi\addspace\&\space}}
```

Standardmäßig kürzt APA nach sieben Autoren mit »...« ab. Um diese Zahl zu erhöhen, gibt man `biblatex` die Option `apamaxprtauth=99` mit.

```
\usepackage[ngerman]{babel}
\usepackage[babel,german=quotes]{csquotes}
\usepackage[style=apa,language=ngerman,apamaxprtauth=99]{biblatex}
\renewcommand{\finalnamedelim}{\ifnum\value{liststop}>2 \fi\addspace\&\space}
\AtBeginBibliography{%
  \renewcommand{\finalnamedelim}{\ifnum\value{liststop}>2 \fi\addspace\&\space}}
\DeclareLanguageMapping{ngerman}{ngerman-apa}
\DeclareFieldFormat{apacase}{#1}
\addbibresource{\jobname.bib}
```

Netz-2

```
Hier das erste Problem:~\parencite{Test.2}
und~\textquote[{\cite[vgl.][\]{Test.2}}]{Test}.
Und hier das nächste Problem:~\textcite{Test.3}.
Und zuletzt dieses Zitat für das Literaturverzeichnis~\cite{Test.1}
\printbibliography
```

²Henri Menke in <http://texwelt.de/wissen/fragen/18024/fragen-zum-apa-style>

Hier das erste Problem: (Mustermann & Musterfrau, 1999) und „Test“ (vgl. Mustermann & Musterfrau, 1999). Und hier das nächste Problem: Duck, Fritz & Duck (2009). Und zuletzt dieses Zitat für das Literaturverzeichnis Mustermann et al., 2016

Literatur

Duck, D., Fritz, P. & Duck, D. (2009). Testbeispiel, 175–190.

Mustermann, M., Musterfrau, M., Musterma, M., Mustermanu, M., Musterfra, M., Muster, M., Musterfrau, M., Muster, M., Mustermann, K. & Muster, M. (2016). Der Mustermann. *Der Mustermamax*, (44), 20–26.

Mustermann, M. & Musterfrau, M. (1999). Minimalbeispiel. *Rundfunk und Fernsehen*, 21, 512–530.

Neue Pakete auf CTAN

Jürgen Fenn

Der Beitrag stellt neue Pakete auf CTAN seit der letzten Ausgabe vor. Die Liste folgt der umgekehrten chronologischen Reihenfolge. Bloße Updates können auf der moderierten Mailingliste `ctan-ann` verfolgt werden, die auch auf Twitter als `@ctanannounce` verfügbar ist.

poetry von *Donald P. Goodman* unterstützt u. a. verschiedene Zeilennummerierungen und einen Index der ersten Verse beim Setzen von Lyrik.

CTAN: `/macros/latex/contrib/poetry`

spectralsequences von *Hood Chatham* ist ein Paket zur Darstellung von Spektralsequenzen bzw. Spektralfolgen mit Hilfe von PGF/TikZ.

CTAN: `graphics/pgf/contrib/spectralsequences`

biblatex-enc von *Jean-Baptiste Camps* ist der Bib \LaTeX -Stil der *École nationale des chartes* (Paris).

CTAN: `/macros/latex/contrib/biblatex-contrib/biblatex-enc`

latex-mr von *Rohit Dilip Holkar* ist eine Einführung in die Verwendung von \LaTeX mit dem Paket `polyglossia` speziell für Marathi und andere indische Sprachen.

CTAN: `/macros/latex/contrib/latex-mr`

knowledge von *Thomas Colcombet* ist eine Lösung, die die konsistente Auszeichnung von Fachbegriffen in einem Text unterstützt. Die Begriffe können mit Hyperlinks zu der Stelle versehen werden, an der sie zuerst verwendet worden sind. Zudem ist die Übernahme in den Index möglich.

CTAN: `/macros/latex/contrib/knowledge`

dviinfox von *Dag Langmyhr* und *Hironobu Yamashita* ist ein Perl-Skript, das Metadaten über DVI- und XDV-Dateien ausgibt (nicht zu verwechseln mit dem Programm von *Valentino Kyriakides* aus den 1990er Jahren).

CTAN: dviware/dviinfox

alkalami von *Bob Tennent* ist ein Font für die von dem Arabischen abgeleitete Schrift Ajami, die in Nigeria und Niger für die dort gebräuchlichen Dialekte verwendet wird.

CTAN: fonts/alkalami

cje von *Kim Nesbitt* ist eine Klasse für Beiträge zum *Canadian Journal of Economics*. Es gibt Optionen zum Review und für das Preprint.

CTAN: /macros/latex/contrib/cje

pst-vehicle von *Thomas Söll*, *Manuel Luque* und *Jürgen Gilg* erstellt mit Hilfe von PSTricks Zeichnungen von beliebig großen Rädern von Fahrzeugen, die auf mathematisch beschreibbaren Kurven »entlangfahren«.

CTAN: graphics/pstricks/contrib/pst-vehicle

bxjaprnind von *Takayuki Yato* passt den Einzug der ersten Zeile in japanischen Texten an die Regeln der japanischen Typografie an, wenn der Absatz mit einer Klammer beginnt. Das Paket funktioniert derzeit nicht mit XeTeX.

CTAN: language/japanese/BX/bxjaprnind

rutitlepage von *Mart Lubbers* erzeugt die Titelseite für wissenschaftliche Arbeiten der Radboud-Universität im niederländischen Nijmegen.

CTAN: /macros/latex/contrib/rutitlepage

currency von *Antoine Lejay* beruht auf dem Paket siunitx und erlaubt es, Währungsangaben standardkonform und konsistent zu setzen.

CTAN: /macros/latex/contrib/currency

combofont von *Ulrike Fischer* ist ein experimentelles Paket, das NFSS-Befehle für sogenannte *Combofonts* bereitstellt, die man sich mit luaotfload aus den Glyphen verschiedener Schriften unter LuaTeX zusammensetzen kann.

CTAN: /macros/luatex/latex/combofont

luapackageloader von *Deepak Jois* ermöglicht das Hinzuladen von Lua-Paketen aus package.path und package.cpath, einschließlich Paketen, die mittels LuaRocks installiert worden waren.

CTAN: /macros/luatex/generic/luapackageloader

bxcalc von *Takayuki Yato* ist ein Bundle, das das Paket calc so ergänzt, dass nunmehr einige Operationen, die bisher standardmäßig nicht möglich waren, ausführbar sein sollten. Außerdem können neue Längen definiert werden, mit denen man rechnen kann.

CTAN: /macros/latex/contrib/bxcalc

xsim von *Clemens Niederberger* ist der Nachfolger des Pakets exsheets zum Gestalten von Aufgaben und Lösungen für Arbeitsblätter. xsim funktioniert intern völlig anders als sein nunmehr obsoleter Vorgänger, der zwar weiter gepflegt wird, aber keine neuen Features mehr erhält.

CTAN: /macros/latex/contrib/xsim

pst-poker von *Herbert Voß* beruht auf dem Paket *poker* von *Olaf Encke* und ermöglicht es, Poker-Spielkarten und -Symbole mit Hilfe von *PSTricks* sowohl in Spielstellungen als auch im Fließtext zu »zeichnen«.

CTAN:graphics/pstricks/contrib/pst-poker

ladder von *Aurélien Cadiou* ist ein Paket zum Setzen von einfachen *Ladder diagrams*, eine grafische Programmiersprache, die an Stromlaufpläne angelehnt ist, mit Hilfe von *TikZ*.

CTAN:graphics/pgf/contrib/ladder

scratch von *Christian Tellechea* stellt Makros bereit, mit denen man Algorithmen wie bei der visuellen Programmiersprache *Scratch* darstellen kann.

CTAN:/macros/latex/contrib/scratch

biblatex-oxref von *Alex Ball* stellt zwei *Bib_{La}TeX*-Stile bereit, die sich an den *New Hart's Rules* und dem *Oxford Guide to Style* orientieren, angelehnt an die Standardstile *verbose* und *authority*. Unterstützt werden auch Multimedia- und Social-Media-Inhalte sowie juristische Nachweise.

CTAN:/macros/latex/contrib/biblatex-contrib/biblatex-oxref

bredzenie von *Marcin Woliński* erzeugt einen Blindtext aus einem polnischen Korpus anstelle des üblichen *lorem ipsum*.

CTAN:/macros/latex/contrib/bredzenie

pst-spinner von *Manuel Luque* und *Herbert Voß* dient zum Zeichnen eines sogenannten *Fidget Spinners* mithilfe von *PSTricks*.

CTAN:graphics/pstricks/pst-spinner

mptrees von *Olivier Péault* stellt Makros zum Zeichnen von Entscheidungsbäumen mit *MetaPost* bereit.

CTAN:graphics/metapost/contrib/macros/mptrees

draftfigure von *Lukas C. Bossert* gibt anstelle von Abbildungen einen Rahmen in der Größe des Bildes aus und ergänzt diesen um weitere konfigurierbare Angaben.

CTAN:/macros/latex/contrib/draftfigure

gotoh von *Takuto Asakura* implementiert den *Gotoh*-Algorithmus für die Sequenzanalyse des Erbguts.

CTAN:/macros/latex/contrib/gotoh

babel-azerbaijani von *Javier Bezos López* enthält die Unterstützung für die aserbaidische Sprache für das Paket *babel*.

CTAN:/macros/latex/contrib/babel-contrib/azerbaijani

ifptex von *Takayuki Yato* ist das Gegenstück zu den Paketen *ifxetex*, *ifluatex* usw. Es testet zur Laufzeit, ob ein Dokument mit der *TeX*-Engine *pTeX* oder einer ihrer Abkömmlinge für Texte in japanischer Sprache gesetzt wird.

CTAN:/macros/generic/ifptex

thaienum von *Abhabongse Janthong* ermöglicht es, Zahlwörter und Schriftzeichen in thailändischer Sprache in Listenumgebungen zu verwenden.

CTAN:/macros/latex/contrib/thaienum

modular von *Daniel Thomas Sank* ist eine Verbesserung des Pakets *coseooul*, mit dem man man auch modular aufgebaute Dokumente mit relativen Abschnittsbefehlen zur

Gliederung auszeichnen kann.

CTAN:/macros/latex/contrib/modular

childdoc von *Niklas Beisert* bietet eine Alternative zu dem Befehl `\includeonly`, wenn Teile eines Dokuments separat kompiliert werden sollen. Die Ausgabe trägt den Namen des Teildokuments, nicht des Hauptfiles.

CTAN:/macros/latex/contrib/childdoc

uhhassignment von *Kim Wittenburg* ist eine Klasse, mit der man Hausarbeiten an der Universität Hamburg setzen kann; sie eignet sich aber auch für Hausaufgaben in der Schule.

CTAN:/macros/latex/contrib/uhhassignment

biochemistry-colors von *Engelbert Buxbaum* stellt die Standardfarben bereit, die in der Biochemie üblicherweise zur Darstellung von Säuren, Nukleotiden, Zuckersorten und Atomen verwendet werden.

CTAN:/macros/latex/contrib/biochemistry-colors

typoaid von *Daniele Ratti* berechnet einige Größen, die für typographische Entscheidungen maßgeblich sein können (Alphabetlänge, 1 ex-Höhe und 1 em-Breite), und gibt das Ergebnis aus.

CTAN:/macros/latex/contrib/typoaid

lucida-otf von *Herbert Voß* ist die \LaTeX -Unterstützung für die Schrift Lucida Bright, die die Mitglieder aller \TeX -Benutzervereinigungen von der TUG kostengünstig erhalten können.

CTAN:/macros/latex/contrib/lucida-otf

Spielplan

2017

19. 8. – 20. 8. **FrOSCon**, Free and Open Source Conference
Hochschule Bonn-Rhein-Sieg
Grantham-Allee 20, 53757 Sankt Augustin
<https://www.froscon.de/startseite/>
11. 9. – 17. 9. **11th International ConT_EXt Meeting**
35510 Butzbach – Alt-Maibach
<http://meeting.contextgarden.net/2017/>
23. 9. **Herbsttagung**
und 57. Mitgliederversammlung von DANTE e.V.
Volkshochschule Mönchengladbach
Lüpertzender Straße 85
41061 Mönchengladbach
<http://www.dante.de/events/Herbst2017.html>

2018

4. 4. – 6. 4. **Frühjahrstagung**
und 58. Mitgliederversammlung von DANTE e.V.
Universität Passau
Gebäude ITZ
Innstraße 43, 94032 Passau
- Sept. **Herbsttagung**
und 59. Mitgliederversammlung von DANTE e.V.
Universität Bielefeld

Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßig Treffen von \TeX -Anwendern statt, die für jeden offen sind. Im Web gibt es aktuelle Informationen unter <http://projekte.dante.de/Stammtische/WebHome>.

Aachen

Torsten Bronger,
bronger@physik.rwth-aachen.de
Restaurant Galilei, Marienbongard 24, 52062 Aachen
Erster Donnerstag im Monat, 20.00 Uhr

Berlin

Michael-E. Voges, Tel.: (03362) 50 18 35,
mevoges@t-online.de
Mantee – Café Restaurant, Chausseestraße 131, 10115 Berlin
Zweiter Donnerstag im Monat, 19.00 Uhr

Dresden

Daniel Borchmann, daniel@algebra20.de, <http://tug-dd.dtnet.org>
auf Anfrage

Erlangen

Walter Schmidt, Peter Seitz,
w.a.schmidt@gmx.net
Gaststätte »Deutsches Haus«, Luitpoldstraße 25, 91052 Erlangen
Dritter Dienstag im Monat, 19.00 Uhr

Frankfurt

Harald Vajkonny,
<http://wiki.lug-frankfurt.de/TeXStammtisch>
Restaurant »Zum Jordan«, Westerbachstr. 7, 60489 Frankfurt
Zweimonatlich, Vierter Donnerstag im Monat, 19.30 Uhr

Göttingen

Holger Nobach,
holger.nobach@nambis.de, <http://goetex.nambis.de/>
Restaurant Mazzoni Cucina Italiana,
Hermann-Rein-Straße 2, 37075 Göttingen
Dritter Donnerstag im Monat, 18.00 Uhr

Hamburg I

Lothar Fröhling,
lothar@thefroehlings.de
Letzter Dienstag im Monat an wechselnden Orten, 19.00 Uhr

Hamburg II

Günther Zander,
guenther.zander@lug-balista.de, <http://www.lug-hamburg.de/kalender>
Bürgerhaus in Barmbek, Lorichsstraße 28a, 22307 Hamburg
Zweiter Montag im Monat, 20.00 Uhr

Hannover

Mark Heisterkamp,
heisterkamp@rrzn.uni-hannover.de
Seminarraum RRZN, Schloßwender Straße 5, 30159 Hannover
Zweiter Donnerstag im Monat, 18.30 Uhr

Heidelberg

Martin Wilhelm Leidig, Tel.: 0170 418 33 29,

moss@moss.in-berlin.de

Anmeldeseite zur Mailingliste: <http://tinyurl.com/stammtisch-HD>

Wechselnder Ort

Letzter Freitag im Monat, ab 19.30 Uhr

Köln

Uwe Ziegenhagen

Dingfabrik, Erzbergerplatz 9, 50733 Köln

Zweiter Dienstag im Monat, 19.00 Uhr

München

Uwe Siart,

uwe.siart@tum.de, <http://www.siart.de/typografie/stammtisch.xhtml>

Erste Woche in geradzahligen Monaten an wechselnden Tagen, 20.00 Uhr

Stuttgart

Bernd Raichle,

bernd.raichle@gmx.de

»Trollinger-Stubn«, Rotebühlstr. 50, 70178 Stuttgart

Zweiter Dienstag im Monat, 19.30 Uhr

Trier

Martin Sievers,

ttt@schoenerpublizieren.de

Anmeldeseite zur Mailingliste: <http://lists.schoenerpublizieren.de/cgi-bin/mailman/listinfo/ttt>

Universität Trier

nach Vereinbarung

Wuppertal

Andreas Schrell, Tel.: (02193) 53 10 93,

as@schrell.de

Restaurant Croatia »Haus Johannisberg«, Südstraße 10, 42103 Wuppertal

Zweiter Donnerstag im Monat, 19.30 Uhr

Würzburg

Bastian Hepp,

LaTeX@sning.de

nach Vereinbarung

Adressen

DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.

Postfach 10 18 40

69008 Heidelberg

Tel.: (0 62 21) 2 97 66 (Mo., Mi.–Fr., 10.00–12.00 Uhr)

Fax: (0 62 21) 16 79 06

E-Mail: info@dante.de

Konto: VR Bank Rhein-Neckar eG

IBAN DE67 6709 0000 0002 3100 07

SWIFT-BIC GENODE61MA2

Vorstand

Vorsitzender:	Martin Sievers	president@dante.de
stv. Vorsitzender:	Herbert Voß	vice-president@dante.de
Schatzmeisterin:	Doris Behrendt	treasurer@dante.de
Schriftführer:	Manfred Lotz	secretary@dante.de
Beisitzer:	Harald König	
	Volker RW Schaa	
	Uwe Ziegenhagen	

Ehrenmitglieder

Peter Sandner	22.03.1990	Klaus Thull († 2012)	22.03.1990
Yannis Haralambous	05.09.1991	Barbara Beeton	27.02.1997
Luzia Dietsche	27.02.1997	Donald E. Knuth	27.02.1997
Eberhard Mattes	27.02.1997	Hermann Zapf († 2015)	19.02.1999
Joachim Lammarsch	12.04.2014	Rainer Schöpf	12.04.2014

Webserver und Mailingliste

DANTE: <http://www.dante.de/> (Rainer Schöpf, Joachim Schrod)

CTAN: <http://mirror.ctan.org/> (Gerd Neugebauer)

DANTE-EV: <https://lists.dante.de/mailman/listinfo/dante-ev>

FAQ

DTK: <http://projekte.dante.de/DTK/WebHome>

T_EX: <http://projekte.dante.de/DanteFAQ/WebHome>

Autoren/Organisatoren

Jürgen Fenn Friedensallee 174/20 63263 Neu-Isenburg juergen.fenn@gmx.de	[42]	Christine Römer Iltisweg 39 07749 Jena tex@christine-roemer.de	[21]
Ulrike Fischer Bismarckstr. 91 41061 Mönchengladbach fischer@troubleshooting-tex.de	[5]	Elke Schubert Hagenstraße 5 76297 Stutensee elke.schubert@kabelbw.de	[8]
Rainer-Maria Fritsch Speerweg 67 13465 Berlin mail@rmf.berlin	[31]	Martin Sievers siehe Seite 50	[4,5]
Markus Kohn Freiherr-von-Drais-Straße 66 68535 Edingen-Neckarhausen komascript@gmx.info	[16]	Herbert Voß Wasgenstraße 21 14129 Berlin herbert@dante.de	[3,40]

Die T_EXnische Komödie

29. Jahrgang Heft 3/2017 August 2017

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 6 Einladung zur Herbsttagung 2017
- 7 Beiträge gesucht («Call for Papers«)

Bretter, die die Welt bedeuten

- 8 Definition eines neuen Gliederungsbefehls mit KOMA-Script
- 16 Verzeichnisse ohne neue Umgebung
- 21 Strukturbäume für Kategorialgrammatiken
- 31 *Ein* Workflow für ein Sachbuch

Von fremden Bühnen

- 40 Im Netz gefunden
- 43 Neue Pakete auf CTAN

Spielplan

- 47 Termine
- 48 Stammtische

Adressen

- 51 Autoren/Organisatoren