

Die T_EXnische Komödie

dante
Deutschsprachige
Anwendervereinigung T_EX e.V.

31. Jahrgang Heft 1/2019 Februar 2019

1/2019

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 (z. B. 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 1.09.2 (TeX Live 2019/dev) erstellt. Als Standardschriften 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: Schleunungdruck GmbH
Eltertstraße 27, 97828 Markttheidenfeld

Redaktion: Herbert Voß (verantwortlicher Redakteur)

Mitarbeit: Lukas C. Bossert Rudolf Herrmann Eberhard Lisse
Rolf Niepraschk Günter Partosch Christine Römer
Volker RW Schaa Martin Sievers

Redaktionsschluss für Heft 2/2019: 15. April 2019

ISSN 1434-5897

Die \TeX nische Komödie 1/2019

Editorial

Liebe Leserinnen und liebe Leser,

diese Ausgabe von »Die \TeX nische Komödie« erscheint überpünktlich, da die Einladungsfrist zur nächsten Mitgliederversammlung von DANTE e.V., die an der Hochschule Darmstadt stattfindet, eingehalten werden muss. Der Umfang dieser Ausgabe ist etwas geringer, sodass ich »gebetsmühlenartig« wieder darauf hinweisen möchte, dass wir an Beiträgen jeglicher Art rund um das Thema \TeX interessiert sind.

Seit langer Zeit mal wieder finden Sie einen Beitrag zu Con \TeX t in der DTK; Henning Hraban Ramm hat einen Vortrag, den Tako Hoekwater auf der letzten Con \TeX t-Konferenz gehalten hat, übersetzt. Alle darin enthaltenen Beispiele finden Sie als eigenständige Dokumente unter <http://www.dante.de/DTK/Software.html>. Unabhängig von dem verwendeten \TeX -Programm ist die vollständige Integration einer Schrift im Format OpenType ebenso aufwändig wie beim alten Format Type 1. Con \TeX t und \LaTeX gehen hier unterschiedliche Wege, die bei Con \TeX t als `\typescript` und bei \LaTeX als Schriftpaket bezeichnet werden.

Christoph Grüninger beschreibt in seinem Artikel, wie man größere Dokumente (Projekte) mit CMake organisieren kann. Sein Beitrag unterliegt einer erweiterten Nutzung, was durch eine entsprechende Lizenzangabe auf der Titelseite vermerkt ist. Von Thomas Ratajczak finden Sie einen Tagungsbericht zur letzten Herbsttagung in Chemnitz.

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,

was macht man, wenn man ein wenig unter Zeitdruck steht und nicht so recht weiß, was man schreiben soll? Man schaut sich beispielsweise an, was man im vorherigen Jahr geschrieben hat. Dabei stellt man dann aber leider u. a. auch fest, welche Erwartungen und Planungen sich im Laufe des Jahres nicht so erfüllt und eingestellt haben. Insofern sollte mich der Blick zurück evtl. vorsichtiger für die Prognosen des aktuellen Jahres machen.

Ich hoffe, Sie sind alle gut ins neue Jahr gestartet. Kurz vor Weihnachten konnten wir endlich den Veranstaltungsort für die Frühjahrstagung 2019 verkünden. Wir werden Gast im Fachbereich Mathematik und Naturwissenschaften an der Hochschule Darmstadt sein. Ganz herzlichen Dank schon jetzt an Torsten-Karl Stempel, der dies möglich macht.

Die offizielle Einladung finden Sie auf S. 6. Die Planungen laufen auf Hochtouren. Um es allen an der Organisation Beteiligten etwas einfacher zu machen, bitte ich Sie, sich möglichst frühzeitig auf der Tagungsseite <http://www.dante.de/events/dante2019.html> zu informieren und anzumelden.

Besondere Aufmerksamkeit möchte ich auch dieses Mal auf die Einreichung von Vorträgen und Tutorien richten. Wir brauchen wie immer Ihre Unterstützung. Mitglieder des Vorstands sollten nach Möglichkeit immer nur ergänzend tätig werden müssen. Alles weitere zum »Call for Papers« finden sich auf den Tagungsseiten sowie in dieser DTK auf S. 7.

Wir feiern in diesem Jahr 30 Jahre DANTE e.V. Auch wenn eingefleischte Anhänger der Bits und Bytes für gewöhnlich erst bei der Zahl 32 (= 2^5) die Korken knallen lassen, so lohnt sich sicherlich auch bei diesem runden Jubiläum in Darmstadt ein kurzes Innehalten. Zudem werde ich den Ehrenpreisträger 2019 bekanntgeben, der aus den eingegangenen Vorschlägen der Mitglieder Mitte Januar vom Vorstand bestimmt wurde. Es ist schön zu sehen, dass es jedes Jahr aufs Neue einige wahrhaft geeignete Kandidaten gibt.

Beim Blick nach vorne wird immer wieder auch der Einfluss des demografischen Wandels auf unseren Verein sichtbar. Wir haben das große Glück, nach wie vor auch

junge Menschen zu erreichen, u. a. dadurch, dass \TeX einen typografischen Bereich abdeckt, der – zumindest oberflächlich betrachtet – zeitlos ist. Viele von uns wissen aber, dass es mittlerweile viele Publikationswege am »klassisch« gedruckten Text vorbei gibt. Und auch die technischen Standards für gedrucktes Material verändern sich, wie ich an dieser Stelle schon einige Male schrieb.

Um innovativ zu bleiben und mindestens eine gleichwertige Konkurrenz zu den kommerziellen Werkzeugen darzustellen, braucht es einerseits zwar Kontinuität, aber gleichzeitig auch immer wieder neue Impulse, thematisch wie personell. Das gilt sowohl bei der (Weiter-)Entwicklung von \TeX selbst samt Paketen (siehe z. B. Lua \TeX , L^A \TeX 3, bibl_{at}ex, fontspec) als auch bei der Bereitstellung notwendiger Strukturen hinter den Kulissen. Gerade hier kommen auf unseren Verein in den nächsten Jahren einige Veränderungen zu. Ich wünsche mir sehr, dass wir es auch diesmal schaffen, neue Personen für die verschiedenen Aufgaben zu gewinnen und langfristig begeistern zu können.

Zum Schluss noch zu den weniger schönen Dingen des Lebens. Auch im abgelaufenen Jahr mussten wir uns wieder von einigen Mitgliedern verabschieden. Im einzelnen waren dies (nach Mitgliedsnummer sortiert):

- Frank Holzwarth (0017)
- Winfried Neugebauer (0049)
- Uwe Kloss (2037)
- Rolf Ketzer (2739)
- Gerd Froitzheim (3264)
- Jürgen Plate (3652)
- Hans-Peter Benckendorff (4477)
- Harald Scheffler (5756)
- Heinz Bürger (6114)
- Joachim Heidemeier (6144)
- Max Haas (6385)

Allen Trauernden gilt unsere aufrichtige Anteilnahme. Wenn Sie einen Nachruf auf eine der Personen schreiben wollen, lassen Sie es uns bitte wissen.

So lagen auch 2018 schöne und traurige Momente nah beieinander. Was uns das neue Jahr alles bringt, werden wir natürlich erst am Ende vollständig wissen. Bis dahin wünsche ich uns allen ein gesundes Jahr mit vielen schönen Momenten, egal ob mit oder ohne \TeX .

Zunächst aber viel Vergnügen bei der weiteren Lektüre.

Herzlichst Ihr/Euer
Martin Sievers

Einladung zur Frühjahrstagung 2019 und 60. Mitgliederversammlung von DANTE e.V. in Darmstadt

Martin Sievers, Torsten-Karl Stempel

Liebe Mitglieder von DANTE e.V.,

wir laden Sie herzlich zur Frühjahrstagung 2019 vom 27. bis 29. März 2019 in Darmstadt ein.

Der Zeitplan sieht wie folgt aus:

Dienstag, 26. März,	ab 19 Uhr: Vorabendtreff
Mittwoch, 27. März,	9 bis 17 Uhr: Tutorien und Vorträge
Donnerstag, 28. März,	ab 9 Uhr: 60. Mitgliederversammlung anschließend bis 17 Uhr: Vorträge
	ab 19 Uhr: Tagungssessen
Freitag, 29. März,	9 bis 17 Uhr: Vorträge
	ab 19 Uhr: Abendtreff
Samstag, 30. März,	ab 10 Uhr: Touristikprogramm (geplant)

Die Mitgliederversammlung beginnt am Donnerstag, den 28. März 2019, um 9 Uhr:

Hochschule Darmstadt,
Fachbereich Mathematik und Naturwissenschaften
Schöfferstr. 3, Haus C10
64295 Darmstadt

Die Tagesordnung lautet:

1. Begrüßung und Tagesordnung
2. Bericht des Vorstands
3. Finanzbericht
4. Bericht der Rechnungsprüfer
5. Entlastung des Vorstands
6. Wahl von Rechnungsprüfern
7. Verschiedenes

Ihre Stimmunterlagen erhalten Sie direkt vor Ort; um vorherige Anmeldung wird gebeten. Eine Übertragung des Stimmrechts ist im Rahmen des § 13 (4) der Vereins-satzung möglich. Wie üblich sind auch Nichtmitglieder als Gäste herzlich willkommen. Unter <https://www.dante.de/events/dante2019.html> finden Sie die Tagungs-seite mit allen weiteren Informationen rund um die Veranstaltung. Neuigkeiten

zur Tagung gibt es auch über unseren Twitteraccount @dante_ev. Für alle Nachrichten zur Veranstaltung sollte dabei das Hashtag #DANTE2019 verwendet werden.

Wegen der besseren Planbarkeit wird um eine frühzeitige Anmeldung gebeten. Das geschieht am besten online. Falls Sie ein Tutorium oder einen Vortrag anbieten wollen, beachten Sie bitte die allgemeinen Hinweise unter <https://www.dante.de/events/dante2019/cfp.html>.

Für Fragen, Wünsche und Anregungen nutzen Sie bitte obige E-Mail-Adresse oder wenden Sie sich an

DANTE e.V.
Stichwort: DANTE 2019
Postfach 10 18 40
69008 Heidelberg

Mit freundlichen Grüßen
Martin Sievers (Vorsitzender DANTE e.V.)
Torsten-Karl Stempel (Hochschule Darmstadt)

Frühjahrstagung 2019 – »Call for Papers«

Martin Sievers, Torsten-Karl Stempel

Liebe Mitglieder von DANTE e.V. ,

die Frühjahrstagung 2019 findet vom 27. bis 29. März 2019 an der Hochschule Darmstadt statt. Am Mittwoch sind Tutorien geplant, der Donnerstag ist für Vorträge und die 60. Mitgliederversammlung vorgesehen und der Freitag für Vorträge bzw. Tutorien.

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

- Erfahrungsberichte zum Einsatz von \TeX bzw. Open-Source-Software im Allgemeinen,
- Nutzung von \TeX durch Schüler/Studenten für den Satz von Facharbeiten bzw. anderer Abschlussarbeiten, Präsentationen etc.,
- Vorstellung spezieller Erweiterungen für den Einsatz in Beruf und Ausbildung,
- Einführungen in \TeX und die zugehörigen Makropakete ($\LaTeX 2_{\epsilon}$ / $\LaTeX 3$, Con \TeX t, ...),
- Lösungen mit den neuen Engines $X_{\text{Y}}\TeX$ und Lua \TeX ,

- Beispiele aus der Praxis (beispielsweise Realisierung besonderer Anforderungen), eigene Klassen und Pakete,
- Einbinden von Schriften, Grafiken etc.,
- 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,
- Barrierefreie PDF-Dokumente,
- ...

Für ein gutes Vortragsprogramm sind wir auf Eure/Ihre aktive Unterstützung angewiesen. Fühlen Sie sich angesprochen? Dann senden Sie bitte bis zum 28. Februar 2019 eine E-Mail mit folgenden Angaben an dante2019@dante.de:

- Name der Referentin/des Referenten,
- Titel des Vortrags bzw. Tutoriums,
- Typ der Präsentation: Vortrag oder Tutorium,
- Zeitbedarf,
- kurze Zusammenfassung (ca. 0,5 bis 1,5 Seiten),
- evtl. benötigte Hilfsmittel (jenseits von Beamer und PDF-Viewer),
- evtl. Wünsche bzgl. der Vortragstage/-zeiten.

Tutorien dauern im Allgemeinen 60 bis 90 Minuten (längere Tutorien sind ausdrücklich möglich); für Vorträge beträgt die übliche Dauer 30 Minuten plus 10 Minuten für die anschließende Diskussion. Wir freuen uns über alle Einreichungen; es darf auch gerne Ihr erster »Auftritt« bei einer Tagung von DANTE e.V. sein.

Unter <https://www.dante.de/events/dante2019.html> finden Sie die Tagungsseite mit allen weiteren Informationen rund um die Veranstaltung. Neuigkeiten zur Tagung gibt es auch über unseren Twitteraccount [@dante_ev](https://twitter.com/dante_ev), für alle Nachrichten zur Veranstaltung sollte dabei das Hashtag #DANTE2019 verwendet werden.

Für Fragen, Wünsche und Anregungen nutzen Sie bitte obige E-Mail-Adresse oder schreiben Sie an:

DANTE e.V.
Stichwort: DANTE 2019
Postfach 10 18 40
69008 Heidelberg

Mit freundlichen Grüßen

Martin Sievers (DANTE e.V.), Torsten-Karl Stempel (Hochschule Darmstadt)

Tagungsbericht Herbsttagung 2018

Thomas Ratajczak

Einleitung

Im Mai entdeckte ich beim Stöbern auf der DANTE-Homepage, dass die Herbsttagung 2018 in Chemnitz stattfinden sollte. Obwohl ich durchaus schon des Öfteren in Sachsen unterwegs war, hatte es mich noch nie nach Chemnitz verschlagen. Es stand zwar auch nicht unbedingt auf meiner To-do-Liste, aber so bot sich nun die Gelegenheit und nach kurzer Rücksprache mit Vhe (meiner Frau) meldete ich uns an und buchte auch sogleich eine Ferienwohnung über *Airbnb*. Ein kurzer Check der Bahnverbindungen ergab dann, dass dies für uns keine annehmbare Alternative bot und wir uns auf eine Autoreise einstellten. Die Wochen und Monate bis zur Tagung zogen ins Land und auf einmal, drei Wochen vor der Tagung, war Chemnitz in aller Munde. Dem Geschehen konnte man durch die exorbitante mediale Berichterstattung nicht entgehen und die Ausschreitungen beziehungsweise nachfolgenden Demonstrationen ließen uns ernsthaft zweifeln, ob wir die Reise nach Chemnitz überhaupt antreten sollten. Das Geschehen ebte zwar nicht wirklich ab, aber nach reiflicher Überlegung beschlossen wir dann, trotzdem, wie geplant, an der Tagung teilzunehmen.

Vorabendtreff

Vhe trat die Reise ohnehin schon mit einem komischen Gefühl im Magen an, nachdem sie selbst von Bekannten aus den USA auf dieses »Chemnitz« angesprochen worden war. Der erste Eindruck, als wir in Chemnitz eintrafen, war dann auch dadurch geprägt, dass wir auf dem Weg zu unserer Unterkunft bereits an Wasserwerfern vorbeifuhren und das große Polizeiaufgebot für die abendliche Demo schon am Nachmittag deutlich zu vernehmen war. Unsere Wohnung lag ungefähr gleich weit vom Tagungsort entfernt wie das Tagungshotel, so dass wir Freitag und Samstag alles bequem zu Fuß erreichen konnten. Der Weg zum Vorabendtreff im *Turmbrauhaus* führte dann auch gleich die Straße der Nationen hinunter in die Innenstadt und kreuzte die Brückenstraße, in der sich das bekannteste Wahrzeichen der Stadt – das Karl-Marx-Monument – befindet. Trotz umfassender Erfahrung mit Großereignissen habe ich selten ein derart großes Polizeiaufgebot gesehen, welches nach meiner Wahrnehmung den Umfang an Demonstranten an diesem Abend zu übertreffen schien. Obwohl es voll war, blieb es ruhig; wir setzten unseren Weg zum Abendtreff fort und trafen dann auch bald auf bekannte Gesichter. In den abendlichen Gesprächen ergab sich, dass es selbst für eine Herbsttagung ein eher

überschaubarer Teilnehmerkreis sein würde. Allerdings hatten erfreulicherweise von diesem Kreis schon viele das Angebot des Vorabendtreffs wahrgenommen. An den Nachbartischen konnte man dann noch eine erstaunlich große chinesische Reisegruppe beobachten, die wohl wegen des Vordenkers des Kommunismus nach Chemnitz gekommen war und nun die Vorzüge eines Brauhauses genoss.

Tagung

Am nächsten Morgen machte ich mich zunächst allein auf den Weg zum Tagungsort, da Vhe ausschlafen und erst nach der Mitgliederversammlung dazu stossen wollte. Mit dem Hinweis per Whatsapp-Foto auf ein diesmal erstmals gedrucktes Namenskärtchen für sie, konnte ich sie dann aber doch schon früher zur Tagung locken. Der Kreis für die MV war, wie erwartet, überschaubar (18 ausgegebene Stimmkarten). Hinzu kamen noch wenige Nichtmitglieder, so dass es insgesamt wohl weniger als 30 Teilnehmer waren. Martin Sievers begrüßte alle, die gekommen waren, und stellte die anwesenden Vorstandsmitglieder vor. Es folgten im Wesentlichen die für eine Herbsttagung üblichen Themen der MV wie Rück- und Ausblick sowie Projekte. Für etwas Diskussion sorgte dann die formal nötige Satzungsänderung, aber hier ging es mehr um das *Wording* als um kontroverse Inhalte. Nachdem auch hierfür eine Lösung gefunden wurde, konnte die MV nach knapp zwei Stunden beendet werden.

Es ging weiter mit den Vorträgen des Tagungsprogramms, welches nur durch jeweils Mittags- (wir speisten vorzüglich in einem syrischen Restaurant!) und Kaffeepause unterbrochen wurde. Thomas H. Meyer gab eine interessante Einführung in *TUSTEP* – das »Tuebinger System von Textverarbeitungsprogrammen« – und ging dabei auch auf die Historie ein. Marei Peischl stellte KOMA-Script mit Schwerpunkt auf Layoutänderungen vor und Martin Sievers ging auf die Vorzüge der Nutzung von BibL^AT_EX mit Biber ein. Thomas und Martin stellten dann zusammen mit Uwe Ziegenhagen noch den aktuellen Stand der Einführung *L^AT_EX für Geisteswissenschaftler* vor, bevor Herbert Voß das Programm mit einem Vortrag zu typischen Fehlern und deren (zweifelhaften typografischen) Folgen beendete. Martin Sievers schloss die Tagung und bedankte sich bei Mario Hausteine und seinen fleißigen Helfern und Helferinnen für die rundum gelungene Organisation.

Abendtreff

Der Abendtreff fand im *Mama Joes* statt, das sich ebenfalls in der Innenstadt befindet. Im Vergleich zum Vortag hatte sich das Stadtbild deutlich zum positiven verändert und auf dem Weg zum Lokal blieb, durch Abwesenheit von Polizei und



Demonstranten, jetzt auch die Gelegenheit ein Foto mit dem *Nischel* im Hintergrund zu machen. Bei mexikanischem Essen und Cocktails (Martin Sievers orderte standesgemäß den *El Presidente*) konnte dann der Tag nochmal in gemütlicher Runde Revue passiert werden lassen. Auch hierzu fand sich wieder etliche Teilnehmer ein.

Touristisches Programm

Nachdem wir am Sonntag morgen unsere Unterkunft geräumt hatten, ging es mit dem Auto zum touristischen Beiprogramm. Eine Führung durch das Sächsische Industriemuseum stand an, der wir ein amerikanisches Frühstück im nahegelegenen Diner vorschalteten. Nach der aufschlussreichen Führung blieb noch genug Zeit, auf eigene Faust die restliche Ausstellung zu erkunden, bevor wir uns dann auf den Rückweg nach Köln machten.

Ohne die Tagung wären wir vermutlich nie nach Chemnitz gekommen. Den Besuch haben wir genossen und können die Teilnahme an einer DANTE-Tagung nur empfehlen.

Bretter, die die Welt bedeuten

LaTeX-Compiler in CMake-Projekten verwenden

Christoph Grüninger

Bei größeren C++-Projekten setzt man gerne das Buildsystem CMake ein, um Compiler und Linker flexibel und in unterschiedlichen Umgebungen verwenden zu können. Wenn eine Dokumentation im LaTeX-Code existiert, soll diese meist ebenfalls mithilfe von CMake gebaut werden. Dafür bieten die beiden Projekte UseLATEX.cmake und UseLatexMk alles Notwendige. Der Artikel beschreibt wie die Projekte zu verwenden sind.

LaTeX eignet sich hervorragend für begleitende Dokumente zu Software in Form von Handbüchern, Dokumentationen oder Tutorials. Die oft genannten Stärken, wie Textsatz in hoher Qualität, leicht zu erstellende mathematische Formeln und unterschiedliche Zielformate, sprechen auch hier für dessen Einsatz. Spezifische Vorteile in einem solchen Szenario sind der automatisierbare Bau der Dokumente, die geringen Einstiegshürden in LaTeX für Programmierer im Vergleich zu anderen potentiellen Nutzern und dass die LaTeX-Quellen keine Binärdateien sind – damit sind die Dateien kleiner und eignen sich wie der Quelltext der Software selbst für den Einsatz einer Versionsverwaltung wie Subversion oder Git.

Für im Quelltext verfügbare Software muss ein flexibler Build-Prozess bereitgestellt werden, damit die Software auf den verschiedensten Installationen übersetzt werden kann. Dies muss gelingen trotz optionaler Abhängigkeiten, unter Wahrung der Kompatibilität zu mehreren Versionen von Werkzeugen und Bibliotheken, die wiederum unter nicht einheitlichen Pfaden abgelegt sind. Da die im C++-Umfeld beliebten Make-Dateien dies schon lange nicht mehr leisten können, gibt es Buildsysteme, die Make- oder vergleichbare Dateien nach einem Konfigurationslauf erzeugen. In den letzten Jahren erfreut sich das Buildsystem CMake¹ bei größeren C++-Projekten wachsender Beliebtheit.

Wird dabei die gesamte Konfiguration und Übersetzung der Software mit CMake gesteuert, so kommt der Wunsch auf, den Bau der begleitenden LaTeX-Dokumente

¹ www.cmake.org

aus CMake heraus zu veranlassen. Übliche C++-Werkzeuge, wie Compiler und Linker, werden mit den passenden Parametern aufgerufen und erzeugen im Idealfall keine Textausgabe. Nur Warnungen oder Fehler werden ausgegeben, die dann vom Buildsystem gespeichert oder dem Nutzer angezeigt werden. Im Gegensatz dazu muss man L^AT_EX-Dokumente in mehreren Läufen übersetzen und dazwischen andere Programme aufrufen, damit Referenzen, Inhaltsverzeichnisse, Index, Nomenklatur etc. erstellt werden. Dabei ist nicht offensichtlich, wie eine Warnung zu beheben ist; abhängig von den im Dokument verwendeten Klassen müssen andere Programme wie BibT_EX oder Biber aufgerufen werden. Kleine Änderungen am Dokument können zwischen einem L^AT_EX-Lauf und bis zu vier L^AT_EX-Läufen nach sich ziehen. Eine eigene Routine zu entwickeln, welche die Belange des eigenen Projekts abdeckt, tritt schnell eine Lawine von Korrekturen los, denn man schafft eine robuste und portable Lösung kaum mit dem ersten Wurf. CMake selbst unterstützt nur bei der Suche nach den Programmen `latex`, `pdflatex`, `makeindex`, `bibtex` und so weiter, nicht aber bei deren Aufrufen, Parametern und der Reihenfolge.

Im Folgenden werden zwei Projekte vorgestellt, mit denen sich viele L^AT_EX-Dokumente, komfortabel durch CMake gesteuert, bauen lassen.

UseLATEX.cmake

Bei einer Suche zum Thema stößt man schnell auf UseLATEX.cmake². 2005 im Wiki von CMake zum ersten Mal veröffentlicht, hat es mit breiter Unterstützung einer großen Nutzerbasis immer mehr Funktionen bekommen, und es wurde von Fehlern bereinigt. Es kann Grafiken automatisch konvertieren, L^AT_EX bei Bedarf mehrfach aufrufen und unterstützt L^AT_EX und BibT_EX inklusive einer Reihe von Alternativen. UseLATEX.cmake setzt die gesamte Logik zum Aufrufen der benötigten Programme in CMake um.

Um es zu nutzen, muss nur die Datei UseLATEX.cmake innerhalb des CMake-Projekts abgelegt und eingebunden werden. Dann kann es mit `add_latex_document` und dem Dateinamen des L^AT_EX-Dokuments als Argument aufgerufen werden. Zusätzlich können optional nach `IMAGES` Bilder und nach `BIBFILES` Bibliographie-Dateien angegeben werden. Es können Angaben zum Namen des Build-Target gemacht werden und ob es automatisch gebaut werden soll oder explizit aufgerufen werden muss. So erzeugt der CMake-Code

```
add_latex_document(
  dokument.tex
  BIBFILES referenzen.bib
  IMAGES bild0.png bild1.jpg
```

² gitlab.kitware.com/kmorel/UseLATEX

```
USE_INDEX
EXCLUDE_FROM_ALL)
```

ein Build-Target dokument. Das Dokument wird nicht bei `make` oder `make all` gebaut. Mit `make dokument` wird aus `dokument.tex` zusammen mit den beiden Bildern und der Bibliographie-Datei eine PDF erzeugt.

Um den Umfang der Möglichkeiten zu umreißen, ist die CMake-Funktion mit allen möglichen Argumenten anzugeben. Die Befehle sind weitgehend selbsterklärend. Die Funktion ist in der CMake-üblichen Darstellung angegeben, wobei Wörter in spitzen Klammern für Namen, insbesondere Dateinamen, stehen und eckige Klammern optionale Teile kennzeichnen:

```
add_latex_document(<tex_file>
    [BIBFILES <bib_files>]
    [INPUTS <input_tex_files>]
    [IMAGE_DIRS] <image_directories>
    [IMAGES] <image_files>
    [CONFIGURE] <tex_files>
    [DEPENDS] <tex_files>
    [MULTIBIB_NEWCITES] <suffix_list>
    [USE_BIBLATEX]
    [USE_INDEX]
    [INDEX_NAMES <index_names>]
    [USE_GLOSSARY] [USE_NOMENCL]
    [FORCE_PDF] [FORCE_DVI] [FORCE_HTML]
    [TARGET_NAME] <name>
    [EXCLUDE_FROM_ALL]
    [EXCLUDE_FROM_DEFAULTS])
```

Weitere Optionen und Funktionen, sowie weitere bereitgestellte CMake-Hilfsfunktionen sind in der umfangreichen Dokumentation beschrieben.

UseLatexMk

Das Programm `latexmk`³ ist ein Perl-Skript und automatisiert den Bau von L^AT_EX-Dokumenten, indem es die Ausgaben der beteiligten Programme parst und die passenden Aufrufe in der benötigten Anzahl veranlasst. Entwickelt seit 1998 erfährt es noch immer regelmäßig Verbesserungen und Korrekturen.

Der Ansatz von `UseLatexMk`⁴ besteht darin, das Rad nicht mit CMake neu zu erfinden, sondern `latexmk` damit zu betrauen. `UseLatexMk` ist noch jung und hat

³personal.psu.edu/jcc8/latexmk/

⁴github.com/dokempf/UseLatexMk

keine große Nutzerbasis, aber es greift auf die Erfahrungen von latexmk zurück und ist eine dünne CMake-Hülle um den latexmk-Aufruf. Es kommt dadurch mit deutlich weniger⁵ eigenem Code aus, der Fehler enthalten kann oder Anpassungen nötig macht.

Ein L^AT_EX-Dokument wird ebenfalls mit `add_latex_document` angelegt. Weitere L^AT_EX-Dateien, Bilder, Bibliographie-Dateien oder ähnliches müssen nicht angegeben werden, da latexmk diese selbst ermittelt. Obiges Beispiel würde mit UseLatexMk umgesetzt werden als

```
add_latex_document(
  SOURCE dokument.tex
  EXCLUDE_FROM_ALL)
```

und das Build-Target `dokument_tex` erzeugen.

Die CMake-Funktion von UseLatexMk mit allen möglichen Argumenten lautet:

```
add_latex_document(SOURCE texsource
  [TARGET target]
  [EXCLUDE_FROM_ALL]
  [REQUIRED]
  [FATHER_TARGET father1 [father2 ...]]
  [RCFILE rcfile1 [rcfile2 ...]]
  [INSTALL destination]
  [BUILD_ON_INSTALL])
```

Obwohl weniger Argumente aufgeführt werden, bietet UseLatexMk einen vergleichbaren Funktionsumfang wie UseLATEX.cmake.

Mit RC-Dateien kann detailliert in den Bau der Dokumente eingegriffen werden; es kann die Standard-RC-Datei von UseLatexMk verwendet werden. Zusätzlich muss man mit `FindLatexMk.cmake` eine Datei angeben, mit der latexmk gesucht und CMake bekannt gemacht wird. Dies ist bei CMake eine Routineaufgabe.

Verwendung und Einordnung

Für beide Ansätze muss nur eine bzw. drei CMake-Dateien ins eigene Projekt kopiert und die entsprechende Datei eingebunden werden. Beide Projekte stehen gleichermaßen unter einer BSD-Lizenz (3-clause) und sollten in den meisten Fällen unproblematisch verwendet werden können. Beide eignen sich, um in Projekten, die CMake verwenden, L^AT_EX-Quellen zu übersetzen. Der konzeptionelle Unterschied ist lediglich, ob die Logik für die richtigen Aufrufe der L^AT_EX-Programme mit CMake

⁵ UseLatexMk kommt mit etwa 15% der Quelltextzeilen aus, 252 statt 1777.

oder außerhalb von CMake umgesetzt wird. UseLatexMk erkennt Abhängigkeiten und ist deshalb für Dokumente mit vielen Bildern und sonstigen zusätzlichen Dateien komfortabler zu verwenden, da die Dateien nicht explizit angegeben werden müssen. Die Liste von vorhandenen Bildern und anderen Dateien kann man sich aber auch mit wenigen Zeilen CMake-Code erstellen lassen.

Trotz der guten Unterstützung durch UseLATEX.cmake und UseLatexMk verhält sich \LaTeX in der CMake-Welt anders als die üblichen, im Idealfall schweigsamen Programmierwerkzeuge: Es erzeugt seitenweise Ausgaben aus mehrfachen Aufrufen, gespickt mit Warnungen, bis man das gewünschte Ergebnis erhält.

Generell scheint es einen Bedarf an einer \LaTeX -Build-Automatisierung wie `latexmk` zu geben. Wäre es nicht schön, wenn man ohne weitere Hilfsmittel nur `pdflatex` mit dem Pfad zum Masterdokument aufrufen müsste und der Rest könnte automatisch erledigt werden? Das würde nicht nur das Leben von Entwicklern erleichtern, die \LaTeX -Dokumente in CMake-Projekten erstellen wollen.

Wie installiere ich eine Schrift für ConTeXt?¹

Taco Hoekwater

Eine neue Schriftfamilie für ConTeXt zu installieren, ist nicht wirklich schwierig, kann für unerfahrene Anwender allerdings etwas abschreckend sein. Dieser Artikel gibt ein Beispiel anhand der frei verfügbaren Schriftfamilie »IBM Plex«.

Installation

Die eigentliche Installation der Schriftdateien ist der einfachste Schritt. Wenn Sie die Schrift ausschließlich mit ConTeXt verwenden wollen, kopieren Sie die Dateien im Format OpenType oder TrueType einfach ins Verzeichnis `fonts/<Typ>` unterhalb von `texmf-local` sowohl für ConTeXt-Minimals als auch für `TEXLive/MiKTEX`. Dann aktualisieren Sie die ConTeXt-Datenbank der vorhandenen Dateien:

```
> mtxrun --generate
```

Wenn Sie die Schrift auch mit Ihrem Betriebssystem verwenden möchten, installieren Sie die Schriftdateien wie jede andere Schrift auch. Den vorherigen Schritt können Sie dann weglassen, weil die Systemschriften unabhängig von einer vorhanden ConTeXt-Datenbank gefunden werden.

¹ Zuerst veröffentlicht im ConTeXt-Group-Journal 2018. Übersetzt von Henning Hraban Ramm.

Ich finde es jedoch praktisch, alle meine ConT_EXt-Schriften irgendwo in der Distribution zu haben, denn dann finde ich sie leichter in Backups usw. Daher neige ich dazu, beides zu tun. Wie dem auch sei, als Nächstes müssen Sie ConT_EXts Schriftendatenbank aktualisieren:

```
> mtxrun --script fonts --reload
```

Das kann eine Weile dauern, je nachdem, wie viele Schriften und Verzeichnisse Sie haben, aber es sollte ohne Fehler durchlaufen. Wenn Sie an dieser Stelle Fehlermeldungen bekommen, sollten Sie auf der ConT_EXt-Mailingliste um Rat fragen, denn dann ist etwas »Merkwürdiges« passiert, was ich hier nicht voraussehen kann. An diesem Punkt ist es sinnvoll, sicherzustellen, dass alles in Ordnung ist. Lassen Sie `mtxrun` eine Liste der Schriften anzeigen und prüfen Sie, ob Ihre neuen Schriften enthalten sind. Sie könnten

```
> mtxrun --script fonts --list --all
```

aufrufen, aber die Ausgabe kann sehr lang werden. Wenn Sie den Namen Ihrer Schriftfamilie kennen, sollten Sie die Ausgabe auf diejenigen Schriften beschränken, die diesen Namen enthalten.

In meinem Beispiel kann ich sicher sein, dass alle richtigen Schrifteinträge den String `Plex` enthalten, und ich bin mir ziemlich sicher, dass nicht allzuvielen anderen Schriften darauf passen. Also:

```
> mtxrun --script fonts --list --all --pattern=Plex
```

Vergessen Sie `--all` nicht, sonst bekommen Sie nur ein einzelnes Ergebnis statt aller passenden Einträge.

In Tabelle 1 auf der nächsten Seite sehen Sie die Ausgabe des Befehls (in einer winzigen Schreibmaschinenschrift, weil sie sonst nicht auf die Seite passt). Stellen Sie sicher, dass alle Schriftdateien, die Sie installiert haben, irgendwo in der Ausgabe erscheinen. Wenn nicht, stimmt vielleicht etwas nicht mit den Schriftdateien oder ihren Zugriffsrechten. Wenn Sie nicht selbst dahinter kommen, sollten Sie wiederum auf der ConT_EXt-Mailingliste nachfragen.

Die Ausgabe von `mtxrun` verstehen

Die Ausgabe von `mtxrun` besteht aus folgenden Spalten:

identifizier Diese interne Kennung kann in ConT_EXt mit dem Präfix `spec:` verwendet werden. Manchmal gibt es mehrere Zeilen für eine Schriftdatei, wobei sich nur diese Kennung unterscheidet, da ConT_EXt versucht, die ziemlich beliebigen Schriftnamen in irgendeine Ordnung zu bringen. Meine Erfahrungen mit der `spec:-`

Tab. 1: Teilausgabe von `mtxrun --script fonts --list --all --pattern=Plex`

resolver	tree	analyzing	'home:texmf'						
identifier	familyname	fontname	filename	subfont	instances				
ibmplexmono	ibmplexmono	ibmplexmono	IBM Plex Mono-Regular.otf						
ibmplexmonobold	ibmplexmono	ibmplexmonobold	IBM Plex Mono-Bold.otf						
ibmplexmonobolditalic	ibmplexmono	ibmplexmonobolditalic	IBM Plex Mono-BoldItalic.otf						
ibmplexmonoextralight	ibmplexmono	ibmplexmonoextralight	IBM Plex Mono-ExtraLight.otf						
ibmplexmonoextralightitalic	ibmplexmono	ibmplexmonoextralightitalic	IBM Plex Mono-ExtraLightItalic.otf						
ibmplexmonoitalic	ibmplexmono	ibmplexmonoitalic	IBM Plex Mono-Italic.otf						
ibmplexmonolight	ibmplexmono	ibmplexmonolight	IBM Plex Mono-Light.otf						
ibmplexmonolightitalic	ibmplexmono	ibmplexmonolightitalic	IBM Plex Mono-LightItalic.otf						
ibmplexmonomedium	ibmplexmono	ibmplexmonomedium	IBM Plex Mono-Medium.otf						
ibmplexmonomediumitalic	ibmplexmono	ibmplexmonomediumitalic	IBM Plex Mono-MediumItalic.otf						
ibmplexmononormal	ibmplexmono	ibmplexmonoitalic	IBM Plex Mono-Italic.otf						
ibmplexmonoregular	ibmplexmono	ibmplexmonoextralight	IBM Plex Mono-ExtraLight.otf						
ibmplexmonosembold	ibmplexmono	ibmplexmonosembold	IBM Plex Mono-SemiBold.otf						
ibmplexmonosembolditalic	ibmplexmono	ibmplexmonosembolditalic	IBM Plex Mono-SemiBoldItalic.otf						
ibmplexmonotext	ibmplexmono	ibmplexmonotext	IBM Plex Mono-Text.otf						
ibmplexmonotextitalic	ibmplexmono	ibmplexmonotextitalic	IBM Plex Mono-TextItalic.otf						
ibmplexmonothin	ibmplexmono	ibmplexmonothin	IBM Plex Mono-Thin.otf						
ibmplexmonothinitalic	ibmplexmono	ibmplexmonothinitalic	IBM Plex Mono-ThinItalic.otf						
ibmplexsans	ibmplexsans	ibmplexsans	IBM Plex Sans-Regular.otf						
ibmplexsansbold	ibmplexsans	ibmplexsansbold	IBM Plex Sans-Bold.otf						
ibmplexsansbolditalic	ibmplexsans	ibmplexsansbolditalic	IBM Plex Sans-BoldItalic.otf						
ibmplexsanscond	ibmplexsanscondensed	ibmplexsanscond	IBM Plex SansCondensed-Regular.otf						
ibmplexsanscondbold	ibmplexsanscondensed	ibmplexsanscondbold	IBM Plex SansCondensed-Bold.otf						
ibmplexsanscondbolditalic	ibmplexsanscondensed	ibmplexsanscondbolditalic	IBM Plex SansCondensed-BoldItalic.otf						
ibmplexsanscondensed	ibmplexsanscondensed	ibmplexsanscond	IBM Plex SansCondensed-Regular.otf						
ibmplexsanscondensedbold	ibmplexsanscondensed	ibmplexsanscondbold	IBM Plex SansCondensed-Bold.otf						
ibmplexsanscondensedbolditalic	ibmplexsanscondensed	ibmplexsanscondbolditalic	IBM Plex SansCondensed-BoldItalic.otf						
ibmplexsanscondensedextralight	ibmplexsanscondensed	ibmplexsanscondextralight	IBM Plex SansCondensed-ExtraLight.otf						
ibmplexsanscondensedextralightitalic	ibmplexsanscondensed	ibmplexsanscondextralightitalic	IBM Plex SansCondensed-ExtraLightItalic.otf						
ibmplexsanscondenseditalic	ibmplexsanscondensed	ibmplexsansconditalic	IBM Plex SansCondensed-Italic.otf						
ibmplexsanscondensedlight	ibmplexsanscondensed	ibmplexsanscondlight	IBM Plex SansCondensed-Light.otf						
ibmplexsanscondensedlightitalic	ibmplexsanscondensed	ibmplexsanscondlightitalic	IBM Plex SansCondensed-LightItalic.otf						
ibmplexsanscondensedmedium	ibmplexsanscondensed	ibmplexsanscondmedium	IBM Plex SansCondensed-Medium.otf						
ibmplexsanscondensedmediumitalic	ibmplexsanscondensed	ibmplexsanscondmediumitalic	IBM Plex SansCondensed-MediumItalic.otf						
ibmplexsanscondensednormal	ibmplexsanscondensed	ibmplexsansconditalic	IBM Plex SansCondensed-Italic.otf						
ibmplexsanscondensedregular	ibmplexsanscondensed	ibmplexsanscondextralight	IBM Plex SansCondensed-ExtraLight.otf						
ibmplexsanscondensedsemibold	ibmplexsanscondensed	ibmplexsanscondsemibold	IBM Plex SansCondensed-SemiBold.otf						

[...]

Kennung sind nicht so gut; sie ist aber sinnvoll im Zusammenhang mit Modulen wie `selectfont`.

familyname Dies ist eine bereinigte Version des Schriftfamilien-Eintrags in den einzelnen Schriftdateien. Auch dieser Name ist hauptsächlich für `selectfont` sinnvoll, und Sie können ihn einfach ignorieren.

fontname Verwenden Sie diese Schriftbezeichnung im Zusammenhang mit dem Präfix `name:`. Dies ist meine bevorzugte Methode, Schriften anzusprechen. Leider funktioniert sie nicht immer; dann ist die nächste Spalte hilfreich.

filename Diese Bezeichnung wird in ConT_EXt mit dem Präfix `file:` verwendet. Das ist besonders dann sinnvoll, wenn Sie verschiedene Versionen von Schriften mit identischen Namen in verschiedenen Schriftdateien haben. Ich greife darauf zurück, wenn ich Schriften sowohl in der ConT_EXt-Distribution als auch im Betriebssystem installiert habe, um sicher zu gehen, dass ich die richtige Version bekomme, wenn eine davon sich bei einem Update ändert.

subfont Bei TrueType-Collections (`.ttc`-Dateien) gibt diese Spalte den Index innerhalb der Collection an. In ConT_EXt sprechen Sie solche Fonts an, indem Sie (`<index>`) an den Dateinamen anhängen, wenn Sie das Präfix `file:` verwenden.

instances Bei variablen Fonts gibt diese Spalte die bekannten Instanzen an.

Die Bezeichnung in der ersten Spalte können Sie auch als Argument der diversen Sub-Optionen von `--list` angeben.

Schriften testen

An dieser Stelle werde ich gerne etwas paranoid. Ich empfehle, eine kleine Testdatei zu erstellen, etwa so:

```
\startTEXpage
\definedfont[name:ibmplexmono] \input knuth
\stopTEXpage
```

Überprüfen Sie die erzeugte Seite optisch, ob sie dem entspricht, was Sie erwarten, und achten Sie auch auf die Zeile `loaded fonts` in der Ausgabe. Die sollte etwa so aussehen:

```
mkiv lua stats > loaded fonts: 3 files: ibmplexmono-regular.otf, latinmodern-math.
↳otf, lmroman12-regular.otf
```

Zur Zeit scheint ConT_EXt immer ein paar Schriften zu laden, die nicht wirklich benutzt werden: Es braucht eine Matheschrift, um irgendwelche Interna zu initialisieren, und da noch keine Matheschrift definiert ist, lädt es `latinmodern-math.otf` als Grundeinstellung. Aus ähnlichen Gründen wird `lmroman12-regular.otf` geladen, aber uns interessiert natürlich `ibmplexmono-regular.otf`.

Zur Sicherheit können Sie alle neu installierten Fonts einzeln testen, aber normalerweise sind alle Datenbankeinträge in Ordnung, wenn einer stimmt.

Zeichenvorrat

Wahrscheinlich haben Sie bei der Auswahl einer neuen Schrift darauf geachtet, dass die Zeichen enthalten sind, die Sie für Ihre Texte brauchen. Aber wahrscheinlich haben Sie das auf der Website des Anbieters getan, und es ist sinnvoll, es zur Vorsicht unter ConT_EXt zu bestätigen. Für einen schnellen Test der lateinischen Zeichen reicht `\showfont`, aber besser sehen Sie sich die ganze Liste an:

```
\usemodule[fnt-10]
\starttext
\ShowCompleteFont{name:ibmplexmono}{20pt}{1}
\stoptext
```

Das erzeugt eine PDF-Datei, in der jedes einzelne Zeichen mit Unicode-Nummer, Nummer in der Schrift, Name des Zeichens (falls vorhanden), erwartetem Namen laut Adobes Glyph-Liste (falls bekannt) und ggf. ConT_EXt-Makro in einer Tabelle aufgelistet ist. Wiederum könnten Sie jeden einzelnen Schriftschnitt testen, normalerweise reicht aber einer. Unterschiede im Zeichenvorrat zwischen Schriftfamilien innerhalb einer Sippe sind normal, innerhalb einer Familie sollten jedoch alle Schnitte den gleichen Umfang aufweisen.

Bei unserem Beispiel enthält `ibmplexsans` kyrillische und griechische Zeichen, aber keine hebräischen (`ibmplexsans` ist die einzige Familie mit Griechisch), und `ibmplexsanshebrew` enthält Hebräisch, aber kein Kyrillisch.

Font-Features

Da wir es mit OpenType-Schriften zu tun haben, gibt es wahrscheinlich Features, die man ein- und ausschalten kann. Sie können wieder `mtxrun` nach den Features einer Schrift fragen:

```
mtxrun --script fonts --list --info ibmplexsans
```

Ein Auszug aus der Ausgabe:

```
mtx-fonts      | gpos features:
mtx-fonts      |
mtx-fonts      | feature  script  languages
mtx-fonts      |
mtx-fonts      | kern     cyrl    dflt
mtx-fonts      |          dflt    dflt
```

mtx-fonts		grek	df1t
mtx-fonts		latn	df1t
mtx-fonts	mark	cyrl	df1t
mtx-fonts		df1t	df1t
mtx-fonts		grek	df1t
mtx-fonts		latn	df1t
mtx-fonts			
mtx-fonts	gsub	features:	
mtx-fonts			
mtx-fonts	feature	script	languages
mtx-fonts			
mtx-fonts	aalt	cyrl	df1t
mtx-fonts		df1t	df1t
mtx-fonts		grek	df1t
mtx-fonts		latn	df1t

Das sagt uns, dass es zwei Positionierungs-Features (gpos) gibt: Kerning (kern) und Akzentplatzierung (mark). Das ist ziemlich normal und beides wird von ConT_EXt standardmäßig eingeschaltet, daher braucht man nicht darauf zu achten.

Wir sehen auch, dass diese Schrift vier Schriftsysteme (Scripts) definiert: kyrillisch (cyrl), default (df1t), griechisch (grek) und lateinisch (latn). Theoretisch könnte es weitere Scripts geben, auf welche diese Features nicht zutreffen, das ist in diesem Fall allerdings höchst unwahrscheinlich. Dass languages den Wert df1t hat, bedeutet einfach, dass dieses Feature angewendet wird, wenn die Sprache auf »default« gesetzt ist. Bei dieser Schrift ist das für alle Features der Fall, einschließlich der Zeichenersetzung (gsub), aber es ist nicht unüblich, dass gsub-Features nur auf bestimmte Kombinationen von Schriftsystem und Sprache angewendet werden. Die hebräischen Schriften in diesem Beispiel definieren ein zusätzliches Feature loc1 für die Sprache iwr im Script hebr.

Ich habe diesen Befehl für eine Reihe verschiedener Schriften aufgerufen, und es sieht so aus, dass alle ungefähr die gleiche Auswahl von Features aufweisen, unabhängig von der Familie. Ein paar Unterschiede gibt es jedoch innerhalb der IBM-Plex-Sippe: Nur die serifenlosen Schriften enthalten griechische Zeichen, daher definieren sie als einzige das Schriftsystem grek. Nur die hebräischen Schriften definieren das Script hebr. Und die diktengleichen (monospaced) Schriften kennen weder kern noch liga.

<i>Feature</i>	<i>Bedeutung</i>
mark	Mark positioning (Akzentplatzierung)
aalt	Access all alternates
ccmp	Glyph (de)composition (Zeichenzusammensetzung)

<i>Feature</i>	<i>Bedeutung</i>
dnom	Denominators (Nenner)
frac	Fractions (Brüche)
liga	Standard ligatures
numr	Numerators (Zähler)
ordn	Ordinals
salt	Stylistic alternates
sinf	Scientific inferiors (Indexzahlen)
ss01	Stylistic set 1
ss02	Stylistic set 2
ss03	Stylistic set 3
ss04	Stylistic set 4
ss05	Stylistic set 5
sup	Superscript
zero	Slashed zero

Das verrät Ihnen jedoch nicht, welche Zeichen von welchem Feature betroffen sind, und ich kenne keinen einfachen Weg, das herauszufinden. Man sollte annehmen, dass der Anbieter Ihnen das mitteilen kann. Oder Sie müssen experimentieren...

Erstellen einer typescript-Vereinbarung

Nach all dieser Vorarbeit wird es Zeit, eine erste Version von einem typescript zu erzeugen, welches ConT_EXt sagt, wie es mit den Schriften umgehen soll. Normalerweise arbeite ich in einer einzelnen Datei, mit einem typescript in der Präambel (Setup-Bereich), gefolgt von ein oder mehreren `\usetypescript`-Zeilen, während der Hauptteil des Dokuments nur Wiederholungen von

```
\showbodyfont[<typescriptname>,12pt]
```

enthält. Ein vollständiges Beispiel folgt später. Zuerst müssen wir uns mit etwas anderem beschäftigen. Normalerweise setzen Sie ein typescript so auf, dass jedes eine der Schriftfamilien aufnimmt, und weisen sie den Klassen Sans, Serif oder Mono zu. Bei einer traditionellen Schriftfamilie aus vier Schnitten ist das einfach. Aber hier zeigen sich ConT_EXts Wurzeln im alten T_EX: Während drei Stile (aufrecht, kursiv und schräg) unterstützt werden, sind nur zwei Stärken (normal und halbfett) vorgesehen. Moderne Schriftfamilien haben dagegen typischerweise nur zwei Stile, dafür aber bis zu neun Stärken! ConT_EXt unterstützt außerdem einen separaten Schriftschnitt für Kapitälchen, der heutzutage kaum noch benutzt wird, da man Kapitälchen als OpenType-Feature anspricht. Der einfachste Weg, damit umzugehen

(zumindest solange bis ConT_EXt irgendwann in der Zukunft erweitert wird) ist, die Schriftfamilien in so viele Vier-Schnitt-Familien aufzuteilen wie nötig, und für jede ein eigenes typescript zu schreiben.

Sehen wir uns die Familie `ibmplexmono` an. Sie hat acht Stärken (`thin`, `extralight`, `light`, `normal`, `text`, `medium`, `semibold` und `bold`), jede mit einem aufrechten und kursiven Schnitt. Die Namen der Stärken sind nicht immer einfach zu interpretieren, so dass man manchmal mehrere Versuche braucht, um passende Schnitte für »normal« und »halbfett« zu finden, in diesem Fall sind die Kombinationen ziemlich klar. Wir brauchen vier verschiedene typescript, und das Ergebnis sieht so aus:

```

\starttypescript [mono] [ibmplex-thin]
  \definefontsynonym [Mono]           [name:ibmplexmonothin]
  \definefontsynonym [MonoItalic]     [name:ibmplexmonothinitalic]
  \definefontsynonym [MonoBold]       [name:ibmplexmonotext]
  \definefontsynonym [MonoBoldItalic] [name:ibmplexmonotextitalic]
\stoptypescript

\starttypescript [mono] [ibmplex-extralight]
  \definefontsynonym [Mono]           [name:ibmplexmonoextralight]
  \definefontsynonym [MonoItalic]     [name:ibmplexmonoextralightitalic]
  \definefontsynonym [MonoBold]       [name:ibmplexmonomedium]
  \definefontsynonym [MonoBoldItalic] [name:ibmplexmonomediumitalic]
\stoptypescript

\starttypescript [mono] [ibmplex-light]
  \definefontsynonym [Mono]           [name:ibmplexmonolight]
  \definefontsynonym [MonoItalic]     [name:ibmplexmonolightitalic]
  \definefontsynonym [MonoBold]       [name:ibmplexmonosemibold]
  \definefontsynonym [MonoBoldItalic] [name:ibmplexmonosemibolditalic]
\stoptypescript

\starttypescript [mono] [ibmplex]
  \definefontsynonym [Mono]           [name:ibmplexmono]
  \definefontsynonym [MonoItalic]     [name:ibmplexmonoitalic]
  \definefontsynonym [MonoBold]       [name:ibmplexmonobold]
  \definefontsynonym [MonoBoldItalic] [name:ibmplexmonobolditalic]
\stoptypescript

\starttypescript [ibmplex-thin,ibmplex-extralight,ibmplex-light,ibmplex]
  \definetypeface [\typescriptone] [tt] [mono] [\typescriptone] [default]
\stoptypescript

\usetypescript[ibmplex-thin,ibmplex-extralight,ibmplex-light,ibmplex]

```



```
\showbodyfont[ibmplex-thin,11pt]
\showbodyfont[ibmplex-extralight,11pt]
\showbodyfont[ibmplex-light,11pt]
\showbodyfont[ibmplex,11pt]
```

[ibmplex-thin] [ibmplex-thin,11pt] \mr : Ag													
	\tf	\sc	\sl	\it	\bf	\bs	\bi	\tfx	\tfx	\tfa	\tfb	\tfc	\tfd
\rm	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\ss	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\tt	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag

[ibmplex-extralight] [ibmplex-extralight,11pt] \mr : Ag													
	\tf	\sc	\sl	\it	\bf	\bs	\bi	\tfx	\tfx	\tfa	\tfb	\tfc	\tfd
\rm	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\ss	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\tt	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag

[ibmplex-light] [ibmplex-light,11pt] \mr : Ag													
	\tf	\sc	\sl	\it	\bf	\bs	\bi	\tfx	\tfx	\tfa	\tfb	\tfc	\tfd
\rm	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\ss	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\tt	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag

[ibmplex] [ibmplex,11pt] \mr : Ag													
	\tf	\sc	\sl	\it	\bf	\bs	\bi	\tfx	\tfx	\tfa	\tfb	\tfc	\tfd
\rm	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\ss	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\tt	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag

Die zwei Parameter [mono] [ibmplex] usw. in den einzelnen Blöcken werden vom dritten und vierten Parameter von `\definetypeface` verwendet. Für eine Schrift-

familie mit Serifen würden Sie `[serif]` als ersten Parameter verwenden, für eine serifenlose `[sans]`.

Der zweite Parameter bleibt Ihnen überlassen, sinnvollerweise verwenden Sie den gleichen Namen später für die eigentlichen typescripts, um Verwirrung zu vermeiden und um Vereinfachungen wie `\typescriptone` zu verwenden.

Wenn Sie diese Datei übersetzen, sollte manches schon gut aussehen. Natürlich sind alle Einträge für die Zeilen `\rm` und `\ss` falsch, aber das war zu erwarten; im Moment interessieren wir uns nur für `\tt`.

In jeder dieser Zeilen sollten `\tt\tf`, `\tt\it`, `\tt\bf`, `\tt\bi` und alle Einträge für `\tt\tfXXX` die gerade installierten Schriften verwenden.

Die Einträge für die schrägen (slanted) Schnitte und die Kapitälchen sind aber noch falsch. Hier sollten wir das T_EX-Erbe in der Schriftbehandlung von ConT_EXt kompensieren. Falls jemand (oder ein anderes Makro) die Schalter `\sl`, `\bs` oder `\sc` verwendet, sollten wir die falsche Schriftart besser abfangen.

Ich löse das normalerweise, in dem ich drei Zeilen zu jedem `\definefontsynonym`-Block hinzufüge:

```
\definefontsynonym [MonoSlanted] [MonoItalic]
\definefontsynonym [MonoBoldSlanted] [MonoBoldItalic]
\definefontsynonym [MonoCaps] [Mono]
```

Das macht den Schalter `\sc` allerdings gegenstandslos. Wenn die Schrift ein Kapitälchen-Feature hätte, ließe sich dieses folgendermaßen einrichten:

```
\definefontsynonym [MonoCaps] [name:ibmplexmono] [features=smallcaps]
```

Diese Schriftfamilie enthält jedoch keine Kapitälchen, also ist das nutzlos. Es ist aber immer noch besser, einen IBM-Plex-Schnitt ohne Kapitälchen in der richtigen Stärke zu sehen als normale Latin-Modern-Kapitälchen.

Als nächstes setzen wir die `\definefontsynonym`-Blöcke für die anderen Familien auf. Das ist sehr ähnlich wie bei der Mono-Familie, außer dass es hier sinnvoll ist, an jede `\definefontsynonym`-Zeile `[features=default]` anzuhängen. Und natürlich müssen die Blöcke mit `[sans]` bzw. `[serif]` anfangen. Da alle drei Familien Schnitte in acht Stärken mitbringen, können wir die Namen der typescripts einfach wiederverwenden, zumindest für `ibmplexserif` und `ibmplexsans`.

Nach all dieses Schriftdefinitionen können wir die Liste der »typefaces« weiter ausfüllen. Zumindest können wir die Einträge für `\rm` und `\ss` vervollständigen:

```
\starttypescript [ibmplex-thin,ibmplex-extralight,ibmplex-light,ibmplex]
\definetypescript [\typescriptone] [rm] [serif] [\typescriptone] [default]
\definetypescript [\typescriptone] [ss] [sans] [\typescriptone] [default]
```

```
\definetypface [\typescriptone] [tt] [mono] [\typescriptone] [default]
\stotypescript
```

Wir werden nicht alle Definitionen hier abdrucken. Sie finden die endgültige Version im ConT_EXt-Wiki (<https://wiki.contextgarden.net/>).

Eigentlich hätten wir gerne auch für Mathematik ein `\definetypface`, doch hier etwas Passendes zu finden, ist nicht so einfach.

Für `ibmplexsanscondensed` und `ibmplexsanshebrew` sollten Sie neue `typescript`-Namen erfinden, wenn Sie sie als eigene `typescripts` verwenden wollen. Nachdem es nur eine schmale Serifenlose gibt, können Sie darauf auch verzichten und die schmalen Schnitte nur zu besonderen Gelegenheiten verwenden, wo die Größe nicht variabel sein muss, wie in Überschriften. Für solche Anwendungen wäre es sinnvoll, ein paar globale `\fontsynonyms` in die `typescript`-Datei aufzunehmen:

```
\definefontsynonym [SansCondensed] [name:ibmplexsanscond] [features=default]
```

Das ist nicht zwingend notwendig, erlaubt aber eine sauberere Verwendung von `\definedfont`. Die hebräischen Schriften richten wir in diesem Fall besser nur als Fallbacks ein.

Fallbacks

Gehen wir einen Schritt zurück und kümmern uns um die hebräischen Schriften. Wir würden diese gerne verwenden, wenn wir Hebräisch setzen, ohne die Möglichkeit zu verlieren, Kyrillisch zu benutzen. Sinnvollerweise definieren wir die hebräischen Schriften daher als Fallback (Rückgriff, Auswechlösung) für die normalen Serifenlosen.

Im Unicode reicht der hebräische Block grob von `uni0590` bis `uni05FF`. Nicht alle Codepunkte sind gefüllt, und von den gefüllten unterstützt die IBM Plex Hebrew nicht alle. Aber besser als nichts. Wir müssen mehrere Fallbacks definieren, hier ist ein Beispiel:

```
\definefontfallback
  [SansHebrewFallback]
  [SansHebrew]
  [0x0590-0x05ff]
  [check=yes, force=no]
```

Wenn dies im dritten Parameter einer `\definefontsynonym`-Zeile aufgerufen wird, weiß ConT_EXt, dass es das Zeichen aus der Ersatzschrift `SansHebrew` nehmen soll, wenn in der aktuellen Schrift ein Zeichen im Unicode-Bereich zwischen hexadezimal `0590` und `05ff` fehlt.

Ich habe hier `force=no` verwendet, weil ich vermute, dass der hebräische Block irgendwann in die normalen Schriften übertragen wird. Dann wäre diese Einstellung zukunftssicher. Mit `force=yes` würden im Font vorhandene Zeichen in diesem Bereich dauerhaft ignoriert. Wir brauchen zwei dieser Definitionen, die obige und eine weitere für `SansBoldHebrew`. Ich habe sie global definiert, damit können wir sie für die verschiedenen Stärken wiederverwenden. Die kursiven Stile lassen wir auf die aufrechten zurückfallen, weil es keine kursiven Schnitte von »Sans Hebrew« gibt.

Wir müssen das Feature `hebrew` statt `default` für die hebräischen Schriften verwenden, es ist in ConT_EXt vordefiniert.

Zurück zu den vier `typescript`-Definitionen für `sans`. Wir müssen die `SansHebrewXXXX`-Schriften ergänzen und die `\definefontsynonym`-Zeilen anpassen. Das Endergebnis eines dieser Blöcke sieht dann so aus:

```
\starttypescript [sans] [ibmplex]
  \definefontsynonym
    [SansHebrew]
    [name:ibmplexsanshebrew] [features=hebrew]
  \definefontsynonym
    [SansHebrewBold]
    [name:ibmplexsanshebrewbold] [features=hebrew]
  \definefontsynonym
    [Sans]
    [name:ibmplexsans]
    [features=default, fallbacks=SansHebrewFallback]
  \definefontsynonym
    [SansItalic]
    [name:ibmplexsansitalic]
    [features=default, fallbacks=SansHebrewFallback]
  \definefontsynonym
    [SansBold]
    [name:ibmplexsansbold]
    [features=default, fallbacks=SansHebrewBoldFallback]
  \definefontsynonym
    [SansBoldItalic]
    [name:ibmplexsansbolditalic]
    [features=default, fallbacks=SansHebrewBoldFallback]
  \definefontsynonym [SansSlanted] [SansItalic]
  \definefontsynonym [SansBoldSlanted] [SansBoldItalic]
  \definefontsynonym [SansCaps] [Sans]
\stoptypescript
```

Jetzt können wir Hebräisch ebenso wie Kyrillisch schreiben und brauchen zur Schriftumschaltung nur ein einfaches `\ss`. Vergessen Sie nicht, die Schreibrichtung

zu ändern! Wenn wir auch die anderen Blöcke ausgefüllt haben, sind wir mit der typescript-Datei fertig!

Stilistische Alternativen und andere Features

Was ist mit den anderen Features wie `ss01` und `zero`?

Letzteres lässt sich einfach testen, denn nach der Definition im OpenType-Standard ersetzt es einfach nur die Null durch eine gestrichene Variante:

```
\usetypescriptfile[plex]
\setupbodyfont[plex,ss,10pt]
```

```
\usebodyfont[plex]
\definedfont[Sans]{There is a 0 chance of \adfff{zero} 0 changes.}
```

There is a 0 chance of 0 changes.

Hoekwater-2

In diesem Fall war es am einfachsten, den Befehl `\adfff` zu verwenden, aber das funktioniert nur mit ein paar einfachen Features. Ziemlich oft brauchen besondere Features den Lua-basierten OpenType-Prozessor, der mit `mode=node` in `\definefontfeature` aktiviert wird.

Die Features, die mit stilistischen Alternativen zu tun haben, sind schwerer zu erforschen. Hier habe ich mir den Font mit einem Schriftditor angesehen. Dabei habe ich herausgefunden, dass die ersten beiden verwendet werden können, um einen »kursiven« Stil einiger Zeichen einzuschalten.

`ss01` tut das für die verschiedenen »a«- und »alpha«-basierten Zeichen, und `ss02` für die »g«-basierten Zeichen.

Feature `ss03` ist ein Alias für `zero`, `ss04` erzeugt eine Null mit einem Punkt in der Mitte, und `ss05` ändert das Aussehen des β -Zeichens.

```
\usetypescriptfile[plex]
\setupbodyfont[plex,ss,10pt]
```

```
\usebodyfont[plex]
\definefontfeature
  [mynormal]
  [mode=node,ss01=yes,ss02=yes,ss04=yes,ss05=yes]
\definedfont[Sans*default]{There is a 0 chance of changes  $\beta$ .}
\definedfont[Sans*mynormal]{There is a 0 chance of changes  $\beta$ .}
```

Hoekwater-3

There is a 0 chance of changes ß.

There is a 0 chance of changes ſ.

Das Feature `frac` sollte einfache Brüche im laufenden Text erzeugen. Um zu sehen, was dabei herauskommt, machen wir am besten einen kleinen Test:

```
\usetypescriptfile[plex]
```

```
\setupbodyfont[plex,ss,10pt]
```

```
\usebodyfont[plex]
```

```
\definefontfeature
```

```
[mynormal]
```

```
[mode=node,frac=yes,ss04=yes]
```

```
\definedfont[Sans*default]{1/2 1/3 2/3 1/4 2/4 3/4 5/9 20/42}
```

```
\definedfont[Sans*mynormal]{1/2 1/3 2/3 1/4 2/4 3/4 5/9 20/42}
```

1/2 1/3 2/3 1/4 2/4 3/4 5/9 20/42

½ ⅓ ⅔ ¼ ⅜ ⅝ ⅞ ²⁰⁄₄₂

Hoekwater-4

In dieser Schrift funktioniert das sehr gut, weil alle numerischen Zeichen vor dem Schrägstrich durch die hochgestellte und danach durch die tiefgestellte Variante ersetzt werden. Manche anderen Schriften implementieren `frac`, indem sie die Eingabe in Ligaturen umwandeln, dann kommt es stark darauf an, wie viele Bruch-Ligaturen in der Schrift enthalten sind.

Je nach Schrift hängen manche Features von anderen ab. Die Dokumentation zur Schrift sollte das erklären, aber oft ist die einzige Möglichkeit, eine Testdatei zu schreiben, in der man alle Features ein- und ausschaltet und nach Unterschieden in der Ausgabe sucht.

Anmerkung

Dieser Artikel kam zustande, weil Pavneet Arora mich um Hilfe beim Aufsetzen der IBM Plex bat. Beim ConT_EXt Meeting 2018 in Sibřina war der allgemeine Konsens, dass uns die IBM Plex lieber war als die Alwyn New, mit der wir bisher das ConT_EXt-Group-Journal gesetzt haben. Die drei großen Vorteile der IBM Plex sind bessere Lesbarkeit, größerer Zeichenvorrat und freie Verfügbarkeit.

Die ConT_EXt-Group hat Alwyn New nicht aufgegeben. Wir werden sie weiterhin für unsere offizielle Kommunikation verwenden. Aber das Journal wird ab sofort mit der IBM Plex Sans und Mono gesetzt.

Rekursive Aufrufe am Beispiel von Abkürzungen

Herbert Voß

Die Rekursion ist eine häufig in Programmen genutzte Strategie zur effektiven Lösung von Problemen. Da \TeX eine vollwertige Programmiersprache ist (Turing-vollständig), lassen sich auch hier rekursive Aufrufe erstellen. Dies soll an einem Beispiel zur Ausgabe von Abkürzungen gezeigt werden.

Problemstellung

Die Typografie sieht für Abkürzungen vor, dass die einzelnen Teile durch $\,$ (\thinspace) miteinander verknüpft werden. Die häufig benutzte Abkürzung »z. B.« sollte daher wie folgt gesetzt werden: z. $\,$ B. Sowohl »z.B.« (ohne Zwischenraum) als auch »z. B.« (normaler Zwischenraum) sind typografisch unbefriedigend. Zur Vereinfachung des Problems soll die Abstandsetzung automatisch erfolgen. Eine Eingabe von beispielsweise $\backslash\text{AK}\{zB\}$ soll die korrekte Ausgabe z. $\,$ B. liefern. Ebenso soll für $\backslash\text{AK}\{aa0\}$ die Ausgabe a. $\,$ a. $\,$ 0. (am angegebenen Ort) erfolgen.¹ Ein weiteres Beispiel ist $\backslash\text{AK}\{\text{DrPhil}\}$, welches zu der Ausgabe Dr. $\,$ Phil. führen soll. Daraus folgt, dass zwei Dinge zu unterscheiden sind: Ein Punkt mit anschließendem Zwischenraum nach *jedem* Zeichen oder nur nach einem Großbuchstaben, wenn kein Kleinbuchstabe folgt.

Lösungsansatz

Mit dem Aufruf des Makros AK wird diesem *ein* Parameter übergeben, unabhängig davon, wieviele Zeichen er enthält. Um an die einzelnen Zeichen zu kommen, benötigt man ein Hilfsmakro, dem der Parameter *ohne* Klammern übergeben wird. Um bei der variablen Anzahl der Einzelzeichen festzulegen, welches das letzte ist, muss das Ende *künstlich* gekennzeichnet werden. Hierfür gibt es verschiedene Möglichkeiten:

```
\newcommand*\AK[1]{\@AK#1!!}% Hauptmakro -> Hilfsmakro
\def\@AK#1#2#3{...}
```

Das Makro $\backslash\text{AK}$ nimmt sein Argument, hängt an dieses zwei Ausrufezeichen an und übergibt es an das Hilfsmakro $\backslash\text{@AK}$. Die Ausrufezeichen wurden willkürlich gewählt; es wären auch andere Zeichen oder das häufig verwendete Makro $\backslash\text{@nil}$ möglich. Nur Buchstaben sind als Begrenzer ausgeschlossen, das sie Teil der Abkürzung sein können. Ausgehend davon, dass $\backslash\text{AK}$ mit mindestens einem Buchstaben aufrufen

¹ Wird bei Verweisen zu Bibliografieangaben verwendet.

wird, ist durch die beiden Begrenzer `!!` sichergestellt, dass `\@AK` mindestens drei Zeichen übergeben werden, so dass im Extremfall `#1=<Zeichen>`, `#2=!` und `#3=!` vorliegt oder dass sich beim Aufruf von `\AK{zB}` die Zuordnung `#1=z`, `#2=B` und `#3=!!` ergibt.

Die beiden Fälle `\AK{aa0}` und `\AK{ProfDr}` unterscheiden sich grundlegend, sodass ein zweiter Parameter eingeführt werden muss, der die Unterscheidung »nach jedem Zeichen ein Punkt« und »nach einem Kleinbuchstaben nur ein Punkt, wenn ein Großbuchstabe folgt«. Dazu kann das optionale Argument beim `\TeX`-Makro genutzt werden:

```
\newcommand*\AK[2][[]]{...}% Hauptmakro-> Hilfsmakro
```

Ist das optionale Argument leer, soll zwischen Kleinbuchstaben *kein* Punkt gesetzt werden. Im anderen Fall wird nach *jedem* Buchstaben ein Punkt gesetzt. Da diese Festlegung dem Hilfsmakro mitgeteilt werden muss und dies nur einmal erfolgen soll, wird die Angabe zwischengespeichert. In der Zeichentabelle beginnen die Kleinbuchstaben bei Nummer 97 und enden bei Nummer 122². [1] Die jeweils zugeordnete Zahl lässt sich auch mit `\TeX` ermitteln:

Die Kleinbuchstaben belegen den ASCII-Bereich von 97 bis 122.

```
1 \count0=`a \count1=`z
2 Die Kleinbuchstaben belegen den ASCII"-Bereich
3 von \the\count0\ bis \the\count1.
```

Damit kann man im Makro `\@AK` ermitteln, ob der nächste in der Liste vorhandene Buchstabe ein großer oder kleiner ist. Ist der optionale Parameter `#1` leer, so wird erfolgreich auf Kleinbuchstaben getestet, anderenfalls nicht, denn bei 122 enden die Kleinbuchstaben. Die Unterscheidung Klein/Groß ist dann irrelevant.

```
\if\relax#1\relax
\def\check@Lower{96}\else\def\check@Lower{123}%
\fi
```

Ist `#1` leer, so wird `\relax` mit `\relax` verglichen, was zu einer wahren Aussage führt. Ist dagegen `#1` *nicht* leer, so wird `\relax` mit `#1` verglichen, was unter der Annahme, dass `#1` selbst nicht `\relax` sein kann, zu einer falschen Aussage führt. Es wird dann der `\else`-Zweig bearbeitet.

Die Abfrage

```
\ifnum`#2>\check@Lower\relax% Es folgt ein Kleinbuchstabe
```

² Die Verwendung von Umlauten müsste man gesondert erfassen, da sie außerhalb des ASCII liegen. Der Einfachheit halber soll hier darauf verzichtet werden.

ist dann wahr, wenn das Zeichen #2 ein Kleinbuchstabe ist *und* `\check@Lower` kleiner als 123 ist. Das aktuelle Zeichen, hier #1, wird in jedem Fall zuerst ausgegeben. Folgt dann ein Kleinbuchstabe, wird das Hilfsmakro erneut aufgerufen:

```
\def\@AK#1#2#3{%
  #1%                aktuelles Zeichen ausgeben
  \ifnum`#2>\check@Lower\relax% Es folgt formal ein Kleinbuchstabe, wenn #2>Vorgabe
  \def\next{\@AK#2#3}%      Rekursion mit dem Rest
```

Der nächste Aufruf wird in einem Hilfsmakro `\next` gespeichert. Dies hat bei der formalen Strukturierung immer dann Vorteile, wenn der rekursive Aufruf mit unterschiedlichen Parametern erfolgen muss. Am Ende des Makros `\@AK` wird dann einfach `\next` ausgeführt.

Ist der Parameter #2 gleich einem Ausrufungszeichen, so ist #1 das letzte Zeichen in der Liste und die gesamte Rekursion kann mit `\next{\@}` beendet werden. `\@ nach` einem Punkt gesetzt verhindert für den Fall, dass kein `\frenchspacing [2]` aktiv ist, den zusätzlichen Zwischenraum, der nach einem Punkt als Satzende eingefügt wird.

Ist der Parameter #2 *nicht* gleich dem Ausrufezeichen, so folgt also ein weiterer Buchstabe. Nun wird zuerst der Zwischenraum `\,` eingefügt und der nächste Aufruf mit `\next{\@AK#2#3}` begonnen. Am Ende der gesamten Abfragen wird das `\next` aufgerufen, was entweder zum erneuten Aufruf von `\@AK` führt oder zur Ausgabe von `\@`.

```
\makeatletter
\newcommand\AK[2][ ]{%
  \if\relax#1\relax
    \def\check@Lower{96}%
  \else
    \def\check@Lower{123}%
  \fi
  \@AK#2!!}
\def\@AK#1#2#3{%
  #1%                aktuelles Zeichen ausgeben
  \ifnum`#2>\check@Lower\relax% Es folgt ein Kleinbuchstabe
  \def\next{\@AK#2#3}%      Rekursion mit dem Rest
  \else%              Großbuchstabe
    .\ifx#2! %        der letzte Buchstabe??
    \def\next{\@}%    Ja, dann \@ ausgeben
  \else
    \,%              \, ausgeben
    \def\next{\@AK#2#3}%      Rekursion mit dem Rest
  \fi
\fi
```

<code>\next%</code>	Ende oder Rekursion
<code>}</code>	
<code>\makeatother</code>	

z. B.	<code>1 \AK{zB}\</code>
a. a. O. ist richtig	<code>2 \AK[.]{aaO} ist richtig\</code>
aa. O. ist falsch	<code>3 \AK{aaO} ist falsch\</code>
Dr.	<code>4 \AK{Dr}\</code>
Prof. Dr.	<code>5 \AK{ProfDr}\</code>
i. d. R.	<code>6 %\AK{ProfDr-Ing}\</code>
M. Sc.	<code>7 \AK[.]{idR}\</code>
Dr. Phil.	<code>8 \AK{MSc}\</code>
	<code>9 \AK{DrPhil}</code>

Literatur und Software

- [1] WIKIPEDIA: American Standard Code for Information Interchange, 2008, https://de.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange (besucht am 18. 11. 2018).
- [2] Clemens Niederberger: Was ist French Spacing? Was macht `\frenchspacing?`, 2015, <https://texwelt.de/wissen/fragen/1154/was-ist-french-spacing-was-macht-frenchspacing> (besucht am 18. 11. 2018).

Ergänzung zum Beitrag »KOMA-Script für Paketautoren am Beispiel tocbasic«

Markus Kohm

Im Artikel »KOMA-Script für Paketautoren am Beispiel tocbasic« in Die \TeX -nische Komödie 4/2018 wurde am Beispiel von `nomenc1` gezeigt, welche Vorteile andere Paketautoren aus der Verwendung von `tocbasic` ziehen könnten. Durch Neuigkeiten bei `nomenc1` haben sich exakt für dieses Beispiel Veränderungen ergeben.

`nomenc1` hat einen neuen Maintainer

Das Paket `nomenc1` [2] hat seit Ende 2018 mit Boris Veytsman wieder einen aktiven Maintainer, der auch bereits eine neue Version des Pakets veröffentlicht hat. Damit

ist die in [1] erklärte Problematik der unklaren Maintainerschaft Vergangenheit. Die Pflege des Pakets geschieht nun außerdem öffentlich auf GitHub [3]. Durch die Einträge 2 und 3 im dortigen *issue tracker* konnte auch bereits eine Verbesserung der Kompatibilität mit KOMA-Script erreicht werden. Es wurden von mir außerdem weitere Verbesserungen vorgeschlagen, die sogar über die Fähigkeiten der alten scrhack-Lösung hinausgehen.

Inkompatibilität von scrhack und dem neuen nomencl

Zunächst fiel auf, dass mit nomencl 5.0 von scrhack bis Version 3.26a eine Warnung wegen eines nicht funktionierenden Patches ausgegeben wurde. Das rührt aus einer Änderung in nomencl und ist bei scrhack eine ständige Gefahr. Daher ist es immer vorzuziehen, wenn solche nachträglichen Patches gar nicht notwendig werden, sondern die Paketautoren selbst entsprechende Änderungen in ihre Pakete einbauen. Leider bedeutete die Änderung in nomencl nicht, dass alle Probleme verschwunden sind. Stattdessen funktionierte lediglich die Korrektur derselben durch scrhack nicht mehr. In scrhack wurde daher eine Änderung vorgenommen, um dieses Problem zu beseitigen. Allerdings hat sich der nomencl-Maintainer nach einem entsprechenden Hinweis ebenfalls der Problematik angenommen, so dass der nomencl-Hack in scrhack zukünftig überflüssig sein sollte.

Probleme beim Kolumnentitel

Trotzdem zeigt das Bemühen des nomencl-Maintainers, alte Probleme des Pakets zu beseitigen, dass die Verwendung von tocbasic einiges einfacher machen und die Wiederholung von Fehlern vermeiden helfen könnte.

So behandelt Issue 2 beispielsweise die Problematik von lebenden Kolumnentiteln. Die Version 5.0 von nomencl verwendet im Fall, dass die Überschrift der Nomenklatur mit `\chapter*` erstellt wird, die Anweisung `\markboth{\nomname}{\nomname}`, um passende Kolumnentitel zu erzeugen. Das erscheint auf den ersten Blick korrekt. Verwendet man jedoch beispielsweise den Seitenstil `myheadings` mit einem statischen Kolumnentitel, so wird durch diese Anweisung der statische Kolumnentitel in der Nomenklatur ersetzt. Das ist sicher nicht gewollt und wird von tocbasic bereits besser gemacht.

Als Korrektur des Fehlers wurde eine Änderung von `\markboth` in `\@mkboth` vorgenommen. Damit wurde in der Tat das genannte Problem beseitigt. Es gibt aber ein weiteres: Die Standardklassen verwenden in der Voreinstellung Kolumnentitel in Großbuchstaben. Durch `\@mkboth{\nomname}{\nomname}` setzt nomencl aber einen Kolumnentitel in gemischter Groß-/Kleinschreibung. Auch dies wird von tocbasic

bereits besser gemacht, darüber hinaus bietet das Paket hierfür eine Konfigurationsmöglichkeit.

Desweiteren setzt die Version 5.0 von `nomencl` zwar einen Kolumnentitel für Klassen, die `\chapter` bieten. Verwendet man aber beispielsweise `article` oder `scrartcl`, so unterbleibt die Anpassung eines Kolumnentitels. Das kann `tocbasic` besser.

All diese weiteren Probleme wurden im *issue tracker* von `nomencl` vermerkt. Für einen Teil der Probleme existiert bereits ein Patch. Es besteht die Hoffnung, dass der `nomencl`-Maintainer die restlichen Probleme entweder durch eigenen Code oder durch generelle Verwendung von `tocbasic` beseitigen wird. Mit der bereits erfolgten Änderung und dem besagten Patch besteht für den Anwender in jedem Fall die Möglichkeit, den `tocbasic`-Weg einfach durch Laden von `tocbasic` in der Dokumentpräambel zu wählen. Bei Verwendung einer KOMA-Script-Klasse geschieht dies implizit.

Fazit

Das soll nun nicht heißen, dass `tocbasic` in jedem Fall und für jede Klasse und alle Kombinationen mit Paketen die beste Lösung bietet. Allerdings sind die im vorherigen Abschnitt dokumentierten Probleme nicht neu. Sie traten schon mit anderen Paketen auf und haben auch bei der Entwicklung von `tocbasic` einiges Kopfzerbrechen bereitet. Wenn jeder Paketautor sich dieser Aufgabenstellung erneut annimmt, besteht natürlich die Gefahr, dass die Probleme immer wieder auftreten. Ziel von `tocbasic` ist ja gerade, solche Fehlerpotentiale zu minimieren und Paketautoren eine Schnittstelle zu bieten, bei der solche Probleme gar nicht erst auftauchen. Gibt es doch einmal ein Problem, so kann dieses für alle Pakete, die auf `tocbasic` aufbauen, gemeinsam gelöst werden, nämlich in `tocbasic`.

Literatur und Software

- [1] Markus Kohm: »KOMA-Script für Paketautoren am Beispiel tocbasic«, *Die T_EXnische Komödie*, 30.4 (2018), 50–57.
- [2] Lee Netherton u. a.: `nomencl` – Produce lists of symbols as in nomenclature, 2018, CTAN: /pkg/nomencl (besucht am 29. 1. 2019).
- [3] Boris Veytsman, <https://github.com/borisveytsman/nomencl> (besucht am 29. 1. 2019).

Tipps und Tricks

Schriften für mehrsprachige Texte

Herbert Voß

Die von Google entwickelte Schrift »Noto« (<https://www.google.com/get/noto/>) gibt es mittlerweile in so vielen Varianten, dass die komprimierte Datei über ein Gigabyte groß ist. Mit einer Vollinstallation von \TeX Live oder $\text{MiK}\TeX$ hat man die für die europäischen Sprachen wesentlichen Schriftdateien im Format OpenType bereits auf seinem Rechner.

Andere Sprachen kann man bei Verwendung von $\text{Xe}\LaTeX$ oder $\text{Lua}\LaTeX$ leicht verfügbar machen, wenn man die entsprechenden Schriftdateien installiert. Das folgende Beispiel verwendet Beispieltex-te, die ohne Rücksicht auf die eigentliche Schreibrichtung ausgegeben werden. Die Definition der Schriften lautet:

```
\setmainfont {Noto Serif}
\newfontfamily\Thai[Language=Thai,Script=Thai]{Noto Serif Thai}
\newfontfamily\Bengali[Language=Bengali,Script=Bengali]{Noto Serif Bengali}
\newfontfamily\Hindi[Language=Hindi,Script=Devanagari]{Noto Serif Devanagari}
\newfontfamily\Nepali[Language=Nepali,Script=Devanagari]{Noto Serif Devanagari}
\newfontfamily\Japanese[Language=Japanese,Script=CJK]{Noto Sans CJK JP}
\newfontfamily\Korean[Language=Korean,Script=CJK]{Noto Sans CJK KR}
\newfontfamily\Mandarin[Language=Chinese Traditional,Script=CJK]{Noto Sans CJK TC
↵}
\newfontfamily\Tibetan[Language=Tibetan,Script=Tibetan]{Noto Sans Tibetan}
\newfontfamily\Arabic[Language=Arabic,Script=Arabic]{Noto Naskh Arabic}
\newfontfamily\ArabicII[Language=Arabic,Script=Arabic]{Noto Kufi Arabic}
\newfontfamily\Urdu[Language=Arabic,RawFeature=+arab.URD %Script=Arab/Urdu
]{Noto Nastaliq Urdu}
\newfontfamily\Hebrew[Language=Hebrew,Script=Hebrew]{Noto Sans Hebrew}
\newfontfamily\Armenian[Script=Armenian]{Noto Armenian}
```

Noto Serif	Some text without any meaning
Thai	ข้อความบางส่วนไม่มีความหมายใด ๆ
Bengali	কোন অর্থ ছাড়া কিছু টেক্সট
Hindi	बना किसी अर्थ के कुछ पाठ
Nepali	कुनै पाठ बना कुनै पाठ

Japanese	意味のないテキスト
Korean	의미없는 일부 텍스트
Mandarin	有些文字沒有任何意義
Tibetan	འགྲོ་བ་མིའི་རིགས་ལྗོངས་ལ་
Kufi Arabic	یٰ نعم یٰ نود صلا ض
Nashk Arabic	یٰ نعم یٰ نود صلا ض
Nastaliq Urdu	نتمھجک ریغے ک ی ن عم ی ہب ی س ک
Hebrew	תועמשמ לכ אללהוהשלכ טסוקט
Armenian	մի տերքստ առանց որևէ իմաստով

Garamond-Math

Herbert Voß

Mit der Schrift Garamond-Math liegt eine neue mathematische Schrift im Format OpenType vor. Diese ist jedoch momentan noch nicht vollständig, denn viele Symbole sind aus anderen Schriften übernommen worden (Libertinus Math und XITS Math). Den jeweils aktuellen Stand kann man unter <https://github.com/YuanshengZhao/Garamond-Math/> erfahren. Die Schrift ist vorrangig für den Einsatz unter Xe_LTeX gedacht; mit anderen T_EX-Engines kann es zu Problemen mit dem Kerning kommen.

Die Anwendung der OpenType-Schrift geschieht in der üblichen Art und Weise:

```
\documentclass{article}
\usepackage[math-style=ISO, bold-style=ISO]{unicode-math}
\setmainfont{EB Garamond}%You should have installed the font
\setmathfont{Garamond-Math.otf}[StylisticSet={7,9}]% Beispiel
\begin{document}
  \[x^3+y^3=z^3\]
\end{document}
```

*ABCDEFGHIJKLMN**OP**QRSTUVWXYZ*

abcdefghijklmnopqrstuvwxy

ABCDEFGHIJKLMNOP**QRSTUVWXYZ**

abcdefghijklmnopqrstuvwxy

ABCDEFGHIJKLMN**OP**QRSTUVWXYZ

abcdefghijklmnopqrstuvwxy

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxy

ABΓΔΕΖΗΘΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ

αβγδεζηθθικκλμνξοπϖρρσςτυφφχψω

ABΓΔΕΖΗΘΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ

αβγδεζηθθικκλμνξοπϖρρσςτυφφχψω

ABΓΔΕΖΗΘΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ

αβγδεζηθθικκλμνξοπϖρρσςτυφφχψω

ABΓΔΕΖΗΘΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ

αβγδεζηθθικκλμνξοπϖρρσςτυφφχψω

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxy

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxy

$$1 + 2 - 3 \times 4 \div 5 \pm 6 \mp 7 + 8 = -a \oplus b \otimes c - \{z\}$$

$$\forall \varepsilon, \exists \delta : x \in A \cup B \subset S \cap T \not\subseteq U$$

$$R_{\nu\kappa\lambda}^{\mu} = \partial_{\kappa} \Gamma_{\lambda\nu}^{\mu} - \partial_{\lambda} \Gamma_{\kappa\nu}^{\mu} + \Gamma_{\kappa\sigma}^{\mu} \Gamma_{\lambda}^{\nu\sigma} - \Gamma_{\lambda\sigma}^{\mu} \Gamma_{\kappa\nu}^{\sigma}$$

$$T_{\alpha_1, \dots, \alpha_k}^{\beta_1, \dots, \beta_l} = T_{i_1, \dots, i_k}^{j_1, \dots, j_l} \frac{\partial x^{i_1}}{\partial x'^{\alpha_1}} \dots \frac{\partial x^{i_k}}{\partial x'^{\alpha_k}} \frac{\partial x'^{\beta_1}}{\partial x^{j_1}} \dots \frac{\partial x'^{\beta_l}}{\partial x^{j_l}}$$

$$\int \sqrt{\frac{1-mu+md/k^2}{2mu/k}} \overbrace{1+2+3+4+5+6+7+8}^{X_p}$$

$$\int_0^1 \frac{\ln(x+1)}{x} dx = \int_0^1 \sum_{i=1}^{\infty} \frac{(-x)^{i-1}}{i} dx = \sum_{i=1}^{\infty} \int_0^1 \frac{(-x)^{i-1}}{i} dx = \sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i^2} = \frac{\pi^2}{12}$$

$$\int_0^{\infty} \int_0^{\infty} \sum_{i=1}^{\infty} \prod_{j=i}^{\infty} \prod_{k=i}^{\infty} \text{ff} \text{ff} \text{ff} \text{ff}$$

$$\left(\left(\left(\left((x) \right) \right) \right) \right) \left[\left[\left[\left[[x] \right] \right] \right] \right] \left\{ \left\{ \left\{ \{x\} \right\} \right\} \right\} \left| \left| \left| |x| \right| \right| \right| \left| \left| \left| \left| \|x\| \right| \right| \right| \right| \right|$$

$$\left\langle \left\langle \left\langle \langle x \rangle \right\rangle \right\rangle \right\rangle \left(\left(\left((x) \right) \right) \right) \left[\left[\left[[x] \right] \right] \right] \left[\left[\left[[x] \right] \right] \right]$$

$$\langle x | + |x \rangle + \langle \alpha | \beta \rangle + |\alpha \rangle \langle \beta | + \left\langle \frac{1}{2} \right| + \left| \frac{1}{2} \right\rangle + \left\langle \frac{1}{2} \right| \frac{1}{2} \rangle + \left| \frac{1}{2} \right\rangle \left\langle \frac{1}{2} \right| + \left\langle \frac{a^2}{b^2} \right| + \left| \frac{e^{x^2}}{e^{y^2}} \right\rangle$$

$$\begin{aligned} \mathcal{F}^{-1}(|j\rangle) &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left(-2\pi i \frac{jk}{2^n}\right) |k\rangle. \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_{n-1}=0}^1 \cdots \sum_{k_0=0}^1 \exp\left(-2\pi i j \sum_{l=0}^{n-1} \frac{2^l k_l}{2^n}\right) |k_{n-1} \cdots k_0\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_{n-1}=0}^1 \cdots \sum_{k_0=0}^1 \bigotimes_{l=1}^n \left[\exp\left(-2\pi i j \frac{k_{n-l}}{2^l}\right) |k_{n-l}\rangle \right] \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^n \left[\sum_{k_{n-l}=0}^1 \exp\left(-2\pi i j \frac{k_{n-l}}{2^l}\right) |k_{n-l}\rangle \right] \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^n \left[|0\rangle_{n-l} + e^{-2\pi i j / 2^l} |1\rangle_{n-l} \right] \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^n \left[|0\rangle_{n-l} + e^{-2\pi i (0 \cdot j_{l-1} \cdots j_0)} |1\rangle_{n-l} \right]. \end{aligned}$$

$$\begin{aligned} S &= \frac{m}{2} \int_0^{t_f} \left[\left(-\omega x_i \sin \omega t + \omega \frac{x_f - x_i \cos \omega t_f}{\sin \omega t_f} \cos \omega t \right)^2 + \sum_{n=1}^{\infty} \left(\frac{a_n n \pi}{t_f} \right)^2 \cos^2 \frac{n \pi t}{t_f} \right] dt \\ &\quad - \frac{m \omega^2}{2} \int_0^{t_f} \left[\left(x_i \cos \omega t + \frac{x_f - x_i \cos \omega t_f}{\sin \omega t_f} \sin \omega t \right)^2 + \sum_{n=1}^{\infty} a_n^2 \sin^2 \frac{n \pi t}{t_f} \right] dt \\ &= \sum_{n=1}^{\infty} \int_0^{t_f} \left[\frac{m}{2} \left(\frac{a_n n \pi}{t_f} \right)^2 \cos^2 \frac{n \pi t}{t_f} - \frac{m \omega^2}{2} a_n^2 \sin^2 \frac{n \pi t}{t_f} \right] dt \\ &\quad + \frac{m \omega^2}{2} \int_0^{t_f} \left[x_i^2 - \left(\frac{x_f - x_i \cos \omega t_f}{\sin \omega t_f} \right)^2 \right] (\sin^2 \omega t - \cos^2 \omega t) dt \\ &\quad - \frac{m \omega^2}{2} \int_0^{t_f} 4x_i \left(\frac{x_f - x_i \cos \omega t_f}{\sin \omega t_f} \right) (\sin \omega t \cos \omega t) dt. \end{aligned}$$

$$\begin{aligned} U(x_f, t_f; x_i, t_i) &= \sqrt{\frac{m \omega}{2\pi i \hbar \sin[\omega(t_f - t_i)]}} \\ &\quad \times \exp \left\{ \frac{i m \omega}{2 \hbar \sin[\omega(t_f - t_i)]} \left[(x_i^2 + x_f^2) \cos[\omega(t_f - t_i)] - 2x_i x_f \right] \right\}. \end{aligned}$$

Von fremden Bühnen

Neue Pakete auf CTAN

Jürgen Fenn

Der Beitrag stellt neue Pakete auf CTAN seit der letzten Ausgabe bis zum Redaktionsschluss in umgekehrter chronologischer Reihenfolge vor. Bloße Updates können auf der moderierten *CTAN-ann*-Mailingliste verfolgt werden.

ltxguidex von *Rebecca Turner* ist eine Erweiterung des Standardpakets `ltxguide`, mit der das Schreiben »schöner \LaTeX -Dokumentationen« einfacher werden soll.
CTAN:macros/latex/contrib/ltxguidex

rgltxdoc von *Richard Grewe* fasst \LaTeX -Code zusammen, den der Autor zur Dokumentation seiner eigenen Pakete verwendet.
CTAN:macros/latex/contrib/rgltxdoc

elegantpaper von *Dongsheng Deng* stellt ein »elegantes« Layout für wirtschaftswissenschaftliche Arbeiten bereit.
CTAN:macros/latex/contrib/elegantpaper

exam-randomizechoices von *Jesse op den Brouw* setzt auf dem Paket `exam` auf und stellt Befehle bereit, um die Antworten (nicht aber die Fragen) bei Multiple-Choice-Aufgaben in einer zufälligen Reihenfolge auszugeben.
CTAN:macros/latex/contrib/exam-randomizechoices

mi-solns von *Donald P. Story* dient zum *Markieren* und Einfügen (englisch: *to insert*) von Lösungen (englisch: *solutions*) aus einer Übung oder aus einem Quiz in ein anderes Dokument. Unterstützt werden die Pakete `exerquiz` und `eqexam`.
CTAN:macros/latex/contrib/mi-solns

latex4musicians von *Guido Gonzato* ist eine ausführliche Anleitung zum Einsatz von \LaTeX in der Musik: Pakete und Programme für den Satz von Musiknoten, von Liedertexten, von Griffbildern von Gitarrenakkorden, für Partituren und für Gitarrentabulaturen werden vorgestellt.
CTAN:info/latex4musicians

latexcolors von *Tobias Plüss* setzt auf dem Paket `xcolor` auf und definiert die Farbwerte, die auf der Website <https://latexcolor.com/> aufgeführt werden.
CTAN:macros/latex/contrib/latexcolors

exercisepoints von *Henning Kerstan* erledigt das Zusammenzählen der Punktwerte, die man in einem Fragebogen oder bei einem Übungsblatt erreichen kann.

CTAN:macros/latex/contrib/exercisepoints

quran-de von *Seiied-Mohammad-Javad Razavian* ist eine deutsche Erweiterung und Anpassung des Pakets `quran`, das vier verschiedene Übersetzungen der Koransuren ins Deutsche bietet.

CTAN:macros/xetex/latex/quran-de

chordbars von *Sebastien Kramm* verwendet `TikZ`, um die Gitarren-, Bass- und Klavier-Akkorde für Jazz- und Popmusik auszugeben.

CTAN:graphics/pgf/contrib/chordbars

memorygraphs von *Camil Staps* zeichnet mithilfe von `TikZ` sogenannte *memory graphs*, mit denen sich die Verwendung des Arbeitsspeichers eines Computers bei der Ausführung eines Programms darstellen lässt.

CTAN:graphics/pgf/contrib/memorygraphs

beamerauxtheme von *Louis Stuart* ist eine Sammlung von »schönen« beamer-Themen, die die Standardthemen abwandeln und ergänzen.

CTAN:macros/latex/contrib/beamer-contrib/themes/beamerauxtheme

CrimsonPro von *Bob Tennent* enthält die gleichnamige Schriftart (gestaltet von *Jacques Le Bailly*) und die dazugehörige \TeX -Unterstützung. Die Schrift ist abgeleitet von der `Crimson Text` von Sebastian Kosch.

CTAN:fonts/crimsonpro

iodhbwm von *Felix Faltin* ist eine inoffizielle Vorlage für wissenschaftliche Arbeiten an der Dualen Hochschule Baden-Württemberg in Mannheim.

CTAN:macros/latex/contrib/iodhbwm

fascicules von *Martin Moritz* ist ein Paket zum Erstellen von Mathematiklehrbüchern für die höheren Schulklassen, zunächst nur in französischer Sprache.

CTAN:macros/latex/contrib/fascicules

multicolrule von *Karl Hagen* stellt Befehle bereit, mit denen man eine dekorative Trennlinie zwischen den Spalten im mehrspaltigen Satz einfügen kann. Es funktioniert sowohl mit den Standardklassen und den mit ihnen kompatiblen Lösungen als auch mit dem Paket `multicol`. Das Paket `multicolrule` beruht auf `expl3` und `xparse`.

CTAN:macros/latex/contrib/multicolrule

lstfiracode von *Ruixi Zhang* ermöglicht es, die Schriftart `Fira` zusammen mit dem Paket `listings` für die Ausgabe von Quelltexten zu verwenden.

CTAN:macros/latex/contrib/lstfiracode

luarandom von *Herbert Voß* ist ein Lua-Programm, das eine Liste mit Zufallszahlen ausgibt. Es funktioniert nur mit Lua \TeX .

CTAN:macros/luatex/latex/luarandom

pkgcheck von *Manfred Lotz* ist ein Programm, das auf der Kommandozeile unter Linux aufgerufen werden kann und das Uploads auf CTAN auf Fehler überprüft.

CTAN:support/pkgcheck

econ-bst von *Shiro Takeda* ist ein Bib \TeX -Stil für wirtschaftswissenschaftliche Veröffentlichungen, der auf natbib aufsetzt und weitgehend konfigurierbar ist.

CTAN:biblio/bibtex/contrib/econ-bst

pst-venn von *Herbert Voß* zeichnet Venn-Diagramme mit drei Kreisen mithilfe von PSTricks.

CTAN:graphics/pstricks/contrib/pst-venn

makecookbook von *Rosalie A. D'Amico* und *Terrence P. Murphy* enthält alles Nötige, um ein Familienkochbuch mit Lua \TeX zu setzen, das man dann bei den gängigen Print-on-Demand-Anbietern drucken lässt. Die Anleitung dient gleichzeitig als eine Einführung in den Buchsatz am Beispiel eines Kochbuchs. Zu jedem Rezept kann man eine kleine Geschichte, praktische Anregungen, Erfahrungen und Tipps vermerken.

CTAN:macros/luatex/latex/makecookbook

xindex von *Herbert Voß* besteht aus einem Lua-Programm und einem ergänzenden \TeX -Paket, mit denen man Indizes von Unicode-kodierten Quelldateien erstellen kann. Die Lösung ist vollständig kompatibel zu makeindex, unterstützt aber die Kodierungen UTF-8, 16, 32 und 64.

CTAN:indexing/xindex

icon-appr von *Donald P. Story* dient dazu, verschiedene Formen von Buttons zum Anklicken in PDF-Dokumenten mit Icons darzustellen.

CTAN:macros/latex/contrib/icon-appr

tblvar von *Antoine Missier* stellt Makros zur Darstellung von *tableaux de variations* einer Funktion zur Verfügung, womit vorwiegend im französischsprachigen Raum der Verlauf der Funktionswerte im Überblick zusammengefasst wird. Pfeile werden mit PSTricks oder mit TikZ erstellt.

CTAN:macros/latex/contrib/tblvar

modeles-factures-belges-assocs von *Robert Sebille* ist eine Sammlung von Vorlagen, mit denen man Rechnungen für gemeinnützige Organisationen nach belgischem Recht setzen kann.

CTAN:macros/latex2e/contrib/modeles-factures-belges-assocs

eqexp1 von *Konstantin Morenko* stellt Makros bereit, um die Komponenten einer Gleichung korrekt auszurichten.

CTAN:macros/latex/contrib/eqexp1

pdftex-quiet von *Jiří Kozlovský* ist ein Bash-Skript, das dafür sorgt, dass nur die wichtigsten Fehlermeldungen von pdftex ausgegeben werden.

CTAN: support/pdftex-quiet

tikzlings von *Susanne Raab* ist eine Sammlung von verschiedenen Tierbildern, die man mit pgf/TikZ zeichnen kann, um sie in einem Dokument zu verwenden. Zur Verfügung stehen Bär, Nasenbär, Koala, Murmeltier, Maulwurf, Maus, Eule, Pinguin und Faultier. Außerdem gibt es einen Schneemann.

CTAN: graphics/pgf/contrib/tikzlings

topiclongtable von *Paolo Brasolin* erweitert das Paket longtable, so dass in einer Tabelle übereinander liegende Zellen miteinander verschmelzen, wenn sie denselben Inhalt haben *und* wenn die Zellen links davon bereits zusammengelegt worden sind. Die Zellen werden nicht zusammengelegt, wenn ein Seitenumbruch unmittelbar vorausgeht.

CTAN: macros/latex/contrib/topiclongtable

zhlineskip von *Ruixi Zhang* hilft, den Zeilenabstand beim Setzen von Texten in Chinesisch, Japanisch und Koreanisch (CJK) einzustellen.

CTAN: language/chinese/zhlineskip

identkey von *Robert White* erleichtert das Setzen dichotomer Bestimmungsschlüssel, wie sie in der Biologie zur Klassifizierung von Arten eingesetzt werden.

CTAN: macros/latex/contrib/identkey

changelog von *Rebecca Turner* stellt eine Umgebung bereit, mit der man ein Änderungsprotokoll (*changelog*) setzen kann. Das Paket wurde angeregt durch <https://keepachangelog.com/>.

CTAN: macros/latex/contrib/changelog

pst-moire von *Jürgen Gilg*, *Manuel Luque* und *Jean-Michel Sarlat* ist ein PSTricks-Paket, mit dem man Moiré-Muster erzeugen kann. Die Anleitung enthält einen zehnjährigen Abschnitt zu den mathematischen Grundlagen für die Erzeugung solcher Dessins.

CTAN: graphics/pstricks/contrib/pst-moire

brandeis-problemset von *Rebecca Turner* ist ein Paket, das Erweiterungen bereitstellt, um die strengen Vorgaben für die Formatierung zu erfüllen, die in den Kursen an der amerikanischen *Brandeis University* üblich sind.

CTAN: macros/latex/contrib/brandeis-problemset

ditaa von *Hiroshi Ukai* ermöglicht es, Diagramme, die als ASCII Art vorliegen, in \LaTeX -Dokumente zu übernehmen. Mehr zu dem dahinter stehenden Java-Programm *ditaa* (*DIagrams Through Ascii Art* von *Stathis Sideris*) unter <http://ditaa.sourceforge.net/>.

CTAN: macros/latex/contrib/ditaa

grabbox von *Jonathan P. Spratte* lädt ein Argument in eine Box und führt dann den darauffolgenden Code aus.

CTAN:macros/latex/contrib/grabbox

pgf-cmykshadings von *David Purton* erweitert das Paket pgf um eine Unterstützung für das CMYK-Farbmodell und für Graustufen, angelehnt an das Paket xcolor. Standardmäßig verwendet pgf nur das RGB-Farbmodell.

CTAN:graphics/pgf/contrib/pgf-cmykshadings

Bücher

Edition dante – Neuerscheinung

Herbert Voß:

Mathematiksatz mit \LaTeX ;

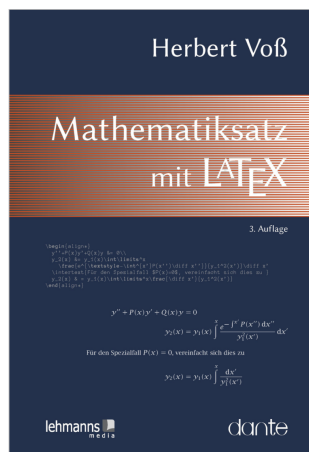
3. Auflage 2018,

DANTE e.V. und Lehmanns Media,

430 Seiten; ISBN 978-3-86541-976-7;

24,95 € (Ladenpreis) bzw. 22,- € für Mitglieder

von DANTE e.V., jeweils versandkostenfrei.



Bestellung

Bitte schicken Sie eine E-Mail an office@dante.de mit Angabe von *Name, Anschrift, Mitgliedsnummer* und *Anzahl der Exemplare*, und überweisen Sie den Betrag auf das Konto von DANTE e.V. oder bezahlen Sie per PayPal. Die Kontonummer finden Sie am Ende dieses Heftes und Informationen zu PayPal auf <http://www.dante.de/index/Intern/Zahlung.html>.

Bitte beachten Sie für Bestellungen bei DANTE e.V. folgende Informationen zum Widerrufsrecht: Käufer können bei Bestellungen per E-Mail, Internet, Brief oder Telefon den Kaufvertrag innerhalb einer Frist von 14 Tagen ab Erhalt der Ware per Brief, Fax oder E-Mail oder durch Rücksendung der Ware widerrufen (siehe Kontaktadresse). Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs oder der Ware. Der Besteller hat in jedem Fall die Rücksendekosten zu tragen. Bei Verschlechterung der Ware, die über die übliche Prüfung der Ware hinausgeht, hat der Besteller gegebenenfalls Wertersatz zu leisten.

Spielplan

2019

16. 3. – 17. 3. **Chemnitzer Linuxtage 2019**
Technische Universität Chemnitz
Reichenhainer Straße 90, 09126 Chemnitz
<https://chemnitzer.linux-tage.de/2019/de/>
27. 3. – 29. 3. **Frühjahrstagung**
und 60. Mitgliederversammlung von DANTE e.V.
Hochschule Darmstadt
Fachbereich Mathematik und Naturwissenschaften
Schöfferstr. 3, Haus C10, 64295 Darmstadt
<http://www.dante.de/events/dante2019.html>
9. 8. – 11. 8. **TUG 2019**
Palo Alto, USA
<http://www.tug.org/tug2019/>
16. 9. – 21. 9. **13th International ConTeXt Meeting**
Bassenge, Belgien
<http://meeting.contextgarden.net/>

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

Stralsund

Heiner Richter,

Heiner.Richter@hochschule-stralsund.de

Hochschule, Haus 21, Raum 326

Zweiter Mittwoch im Monat, 17.00 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., Do. von 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:	Klaus Höppner	
	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>

T_EXnische Fragen

beraterkreis@dante.de

Autoren/Organisatoren

Jürgen Fenn

Friedensallee 174/20
63263 Neu-Isenburg
juergen.fenn@gmx.de

Christoph Grüninger

Sindelfingen
foss@grueninger.de

Taco Hoekwater

taco@elvenkind.com

Markus Kohm

Freiherr-von-Drais-Straße 66
68535 Edingen-Neckarhausen
komascript@gmx.info

Henning Hraban Ramm

hraban@fieee.net

[41] **Thomas Ratajczak** [9]

Weißhausstr. 23
50939 Köln
ratajczak@gmail.com

[13] **Martin Sievers** [4,6,7]

siehe Seite 50

[17] **Torsten-Karl Stempel** [6,7]

Hochschule Darmstadt
torsten-karl.stempel@h-da.de

[34] **Herbert Voß** [3,31,37,38,46]

Wasgenstraße 21
14129 Berlin
herbert@dante.de

Die T_EXnische Komödie

31. Jahrgang Heft 1/2019 Februar 2019

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 6 Einladung zur Frühjahrstagung 2019 in Darmstadt
- 7 Frühjahrstagung 2019 – »Call for Papers«
- 9 Tagungsbericht Herbsttagung 2018

Bretter, die die Welt bedeuten

- 13 \LaTeX -Compiler in CMake-Projekten verwenden
- 17 Wie installiere ich eine Schrift für ConT_EXt?
- 31 Rekursive Aufrufe am Beispiel von Abkürzungen
- 34 Ergänzung zum Beitrag »KOM_A-Script für Paketautoren am Beispiel tocbasic«

Tipps und Tricks

- 37 Schriften für mehrsprachige Texte
- 38 Garamond-Math

Von fremden Bühnen

- 41 Neue Pakete auf CTAN

Bücher

- 46 Edition *dante* – Neuerscheinung

Spielplan

- 47 Termine
- 48 Stammtische

Adressen

- 51 Autoren/Organisatoren