

Die T_EXnische Komödie

DANTE
Deutschsprachige
Anwendervereinigung T_EX e.V.

12. Jahrgang Heft 2/2000 Juli 2000

2/2000

Impressum

„Die T_EXnische Komödie“ ist die Mitgliedszeitschrift von DANTE e.V. Der Bezugspreis ist im Mitgliedsbeitrag enthalten. Namentlich gekennzeichnete Beiträge geben die Meinung der Schreibenden wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nur im nicht-kommerziellen Rahmen gestattet. Verwendungen in größerem Umfang bitte zur Information bei DANTE e.V. melden.

Beiträge sollten in Standard-L^AT_EX-Quellcode unter Verwendung der Dokumentenklasse `dtk` erstellt und an untenstehende Anschrift geschickt werden (entweder per E-Mail oder auf Diskette). Sind spezielle Makros, L^AT_EX-Pakete oder Schriften dafür nötig, so müssen auch diese mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden.

Diese Ausgabe wurde mit Hilfe folgender Programme fertiggestellt: **TeX**, Version 3.14159 (Web2c 7.3), **LaTeX2e** <1998/12/01>, **xdvick** 18f und **windvi** (für Bildschirmdarstellung) und **dvips(k)** 5.78 (für Korrektur und Belichtung). Die Schriften zur Belichtung wurden mit dem METAFONT-Modus `linoone` (1270 dpi) berechnet.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2300

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

E-Mail: dante@dante.de
dtk-redaktion@dante.de (Redaktion)

Druck: Konrad Tritsch Print und digitale Medien GmbH
Haugerring 5, 97070 Würzburg

Redaktion: Gerd Neugebauer (verantwortlicher Redakteur)
Jan Braun Luzia Dietsche Rudolf Herrmann
Uwe Münch Rolf Niepraschk Günter Partosch
Bernd Raichle Volker RW Schaa Karin Schwind
Peter Willadt

Redaktionsschluß für Heft 3/2000: 4. August 2000

ISSN 1434-5897

Editorial

Liebe Leserinnen und Leser,

es grenzt manchmal schon an ein Wunder. Am Anfang stehen einige Artikel, die entweder schon halbwegs fertig sind oder deren Fertigstellung versprochen wurde, und am Ende ist wieder eine Ausgabe der Vereinszeitschrift voll. So war es auch dieses Mal – auch wenn wir ein bisschen geschwindelt haben und einige Beiträge von Redakteuren stammen.

Auf jeden Fall sind in dieser Ausgabe wie immer einige lesenswerte Dinge über \LaTeX zu finden. Im vereinsinternen Teil gibt es einen Bericht über eine Aktion zur Öffentlichkeitsarbeit von DANTE e.V.. NTS geht zum ersten Mal mit einer Verteilung an die Öffentlichkeit und sucht dafür Alpha-Tester. Schließlich wurde auch ein Veranstalter für die Herbsttagung gefunden. Die Einladung hierzu ist in dieser Ausgabe enthalten.

\TeX nisches gibt es auch zu vermelden. Ein Erfahrungsbericht von \TeX aus der Praxis ist enthalten. Eine Anwendung von PStricks wird beschrieben. Es gibt etwas über Tabellen zu lesen und schließlich wird gezeigt, wie man eine Liste von Definitionen anlegen kann.

Zu Schriften gibt es zwei Beiträge. Zum einen wird die immer wieder auftauchende Forderung nach Arial als Schrift in \LaTeX erfüllt. Zum anderen wird die Verwendung der Original-Helvetica-Schrift in Ghostscript ermöglicht. In beiden Fällen wird gezeigt, dass man mit ein bisschen Wissen kommerzielle Schriften frei bekommen kann. Die meisten guten Schriften jenseits von METAFONT haben ihren Preis.

Als Übersetzung haben wir einen Artikel zur Zukunft von \TeX . Er hat auch heute nur wenig von seiner Aktualität verloren.

Ich wünsche allen Lesern bei der Lektüre viel Vergnügen und hoffe wieder einmal auf ein Wunder für die nächste Ausgabe.

Ihr Gerd Neugebauer

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

wie Sie vielleicht bereits aus den E-Mail-Listen entnehmen konnten, hat sich ein Veranstalter für die Herbsttagung 2000 gefunden. Die Tagung und die 23. Mitgliederversammlung von DANTE e.V. wird vom Universitätsrechenzentrum der Fernuniversität in Hagen ausgerichtet und findet am 6. und 7. Oktober 2000 statt. Für Freitag, den 6. Oktober, sind dabei, wie üblich, kostenlose Tutorien vorgesehen. Die Mitgliederversammlung von DANTE e.V. wird am Vormittag des 7. Oktober und somit erneut an einem Wochenende stattfinden. Die Einladung zur Mitgliederversammlung und zur Tagung finden Sie in dieser Ausgabe auf Seite 8.

Erfreulich ist, dass sich auch für die Frühjahrstagung 2001 ein Veranstalter gefunden hat. So können wir mitteilen, dass sie an der Fachhochschule Rosenheim stattfindet und von Prof. Dr. Georg Lachenmayr organisiert wird.

Wie Sie dem Protokoll der letzten Mitgliederversammlung in Clausthal-Zellerfeld entnehmen konnten, wurde dort eine Anpassung der Mitgliedsbeiträge anlässlich der Umstellung auf den Euro beschlossen. Unsere neue Satzung erfordert, dass solche Beschlüsse bei der Einladung zur Mitgliederversammlung in der Tagesordnung angekündigt werden. Da diese Beschlussabsicht in der Einladung zur letzten Mitgliederversammlung fehlte, ist der Beschluss somit nicht gültig. Wir werden ihn auf der Versammlung am 7. Oktober in Hagen wiederholen müssen.

Mit dieser Ausgabe erhalten Sie die neueste Ausgabe der CD-ROM „ \TeX Live 5“, die das Team um Sebastian Raetz in unermüthlicher Arbeit zusammengestellt hat. Sie wurde in diesem Jahr zusammen mit Lehmanns Fachbuchhandlung GmbH in Hamburg produziert. Unser Dank gilt Luzia Dietsche und Klaus Höppner, die das Cover und CD-Label gestalteten und auch für den

CD-Master verantwortlich zeichnen. Mit der Ausgabe 3/2000 der Mitgliederzeitschrift „Die T_EXnische Komödie“ planen wir im September die Verteilung der CD-ROMs mit dem CTAN-Abzug.

Als letzte Nachricht möchten wir noch auf die Ankündigung des NTS-Teams hinweisen, dass die Java-Portierung von T_EX das Alpha-Teststadium erreicht hat und Alpha-Tester gesucht werden. Weitergehende Hinweise hierzu finden Sie in dieser Ausgabe ab Seite 9.

Mit T_EXnischen Grüßen

Thomas Koch Volker RW Schaa
(Präsident) (Vizepräsident)

DANTE e.V. bei der MNU

Luzia Dietsche

In der Woche vor Ostern fand in Stuttgart die Tagung MNU 2000 des Vereins zur Förderung des mathematischen und naturwissenschaftlichen Unterrichts statt, bei der einige Freiwillige mit einem Stand Werbung für T_EX und DANTE e.V. machten.

„Was meint Ihr, sollte DANTE e.V. nicht auch bei einer Tagung von Lehrern präsent sein und Werbung für L^AT_EX machen?“ Diese an sich harmlose Frage, bei einem der Stammtische in Stuttgart gestellt, sollte zu einiger Arbeit und einem interessanten Erlebnis führen. Ganz spontan beantwortete ich sie nämlich mit einem „Ja, natürlich!“ und fand mich dann ziemlich schnell als eine derjenigen wieder, die einen Stand bei der diesjährigen Tagung des deutschen Vereins zur Förderung des mathematischen und naturwissenschaftlichen Unterrichts (kurz MNU, <http://www.mnu.de/>) vom 16. bis 20. April organisierten und betreuten.

Die Vorarbeit bestand für mich darin, Beispieldokumente zusammenzusuchen, Werbematerial für DANTE e.V. zu organisieren und Bücherspenden zu akquirieren. Meine Anfrage bei verschiedenen Bekannten und per Kommunikationslisten erbrachte Aufgabensammlungen, Diplomarbeiten, Dokumenta-

tionen und Poster, die sich sehr gut für die Ausstattung des Standes verwenden ließen. Von DANTE e.V. bekamen wir Ansichtsexemplare der Mitgliederzeitung und einige Exemplare der diversen CD-ROMs, sowie natürlich Informationsblatt und Antragsformular für Interessierte. Zu einer Bücherspende war dankenswerterweise der Verlag dpunkt aus Heidelberg bereit, von dem unter anderem je ein Exemplar seiner L^AT_EX-Bücher vorlag. Die restlichen Bücher stammten aus meinem privaten Fundus.

Jan Theofel, der die Eingangsfrage gestellt hatte, kümmerte sich um den Stand selbst und überzeugte weitere Freiwillige, an den drei Tagen der Konferenz dabei zu sein. Er ist *sehr* überzeugend, wenn er sich etwas in den Kopf gesetzt hat. Auch die Hardware besorgte er, die aus zwei Rechnern und einem Drucker bestand, so dass wir T_EX unter Linux und Windows vorführen und Beispiele ausdrucken konnten.

So gerüstet fanden sich am Montagmorgen Claudia Walter, Carsten Geckeler, Jan Theofel und ich am Tagungsort ein, um den Ansturm der Lehrer zu erwarten. Naja, ein Ansturm war es zwar nicht, aber dafür ein stetes Interesse, was T_EX denn sei, wer DANTE e.V. ist und wofür wir wohl werben würden. Diesen regen Fluss an Menschen hatten wir auch der Tatsache zu verdanken, dass man zur Cafeteria und zum Klett-Verlag bei uns vorbei musste. Beste Voraussetzungen also, die wir nach Kräften zu nutzen versuchten.

Die meisten, die bei uns stehen blieben, wussten wenig bis gar nichts über T_EX und zeigten sich sehr beeindruckt von dem, was wir zeigen konnten. Im allgemeinen war der Rechner in Aktion, auf dem Windows, fpT_EX und WinEdt installiert waren. Gerade diese Oberfläche mit der Menüleiste für mathematische Sonderzeichen war für eine Vorführung hervorragend geeignet. Von Vorteil war außerdem, dass man WinEdt mit beliebigen Vorlagen bestücken kann, die das Erstellen typischer Eingabetexte deutlich vereinfachen.

Wenn auch Windows und WinEdt für alle von gleichem Interesse waren, so kamen doch noch viele weitere Wünsche. Seien es spezielle Sonderzeichen, das Erzeugen von pdf-Dateien, die weitere Konfiguration von WinEdt (Belegung einzelner Tasten mit speziellen Befehlen) oder Makros und Styles für Klassenarbeiten und Hausaufgaben. Zum Glück konnten wir alle Wünsche erfüllen und den einen oder anderen Lehrer ausgesprochen glücklich machen („Schon alleine, dass ich dieses Makro von Ihnen bekomme habe, hat den Besuch der MNU für mich lohnend gemacht.“), nicht zuletzt auch dank Bernd Raichle, der abends zu unserer Verstärkung herbeieilte.

Neben den Neueinsteigern waren auch einige da, die $\text{T}_{\text{E}}\text{X}$ von früher kannten und sich gerne überzeugen ließen, erneut damit anzufangen, hat sich doch in den letzten Jahren einiges getan, so dass das Arbeiten leichter geworden ist. Sogar ein ehemaliges Mitglied von DANTE e.V. tauchte bei dem Stand auf, das ziemlich verblüfft war, uns zu sehen. Für mich ein Zeichen, dass der Verein zu wenig auf solche Werbung setzt.

Mit Freude habe ich registriert, dass sich verschiedene Vertreter von Verlagen bei uns einfanden und sich ausführlich erklären ließen, was $\text{T}_{\text{E}}\text{X}$ ist, wie es funktioniert und wie es sich eventuell bei ihrer Buchproduktion einsetzen lässt. Ich hoffe, dass es nicht nur bei der Begeisterung während der Vorführung bleibt, sondern sich daraus eine Zusammenarbeit mit DANTE e.V. entwickelt.

Insgesamt hatte ich den Eindruck, dass etliche Lehrer gerne auf $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ umsteigen würden, wenn sie sich nicht so alleine fühlen würden. Sehr häufig war zu hören, dass $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ja schon sehr schön und praktisch wäre, aber der Austausch der Dateien mit Kollegen eben dann nicht mehr möglich sei. Außerdem könne man niemanden bei Problemen fragen, sondern müsse alles selbst erarbeiten und dazu fehle oft die Zeit. Hieraus ziehe ich den Schluss, dass mehr Werbung an Schulen nötig ist, und zwar sowohl, was die Existenz von DANTE e.V. allgemein als auch die vom Verein geleistete Unterstützung betrifft.

Sehr wünschenswert ist aus meiner Sicht ein Arbeitskreis (oder wie auch immer man das nennen mag) aus Lehrern, Schülern und $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Benutzern, die bei Problemen weiter helfen können. Bei einer solchen Zusammenarbeit könnte beispielsweise eine $\text{T}_{\text{E}}\text{X}$ -Distribution entstehen, die speziell auf die Bedürfnisse im Schulbereich zugeschnitten ist. Auch spezielle Makros und Klassen wären von Interesse, die das Arbeiten erleichtern und damit der Verbreitung in diesem Umfeld dienlich wären. Und da ein solcher Wunsch nur dann eine Chance auf Verwirklichung hat, wenn ein Impuls da ist, rege ich einen Arbeitskreis *Schulen* an und fordere alle Interessierten auf, sich bei mir oder DANTE e.V. hierfür zu melden.

Einladung zur Herbsttagung und 23. Mitgliederversammlung von DANTE e.V.

Thomas Koch, Thomas Feuerstack

Hiermit laden wir Sie herzlich zur Herbsttagung und 23. Mitgliederversammlung von DANTE e.V. ein. Die Tagung wird vom Universitätsrechenzentrum der Fernuniversität in Hagen ausgerichtet und findet am 6. und 7. Oktober 2000 an der

Fernuniversität in Hagen
Allgemeines Verfügungszentrum I (AVZ I)
Feithstr. 140
58097 Hagen

statt. Für Tagung und Versammlung ist folgender Zeitplan vorgesehen:

Donnerstag, 5. Oktober:	Vorabendtreff
Freitag, 6. Oktober:	Tutorien; abends Tagungssessen
Samstag, 7. Oktober:	Mitgliederversammlung von DANTE e.V. 10:00 Uhr im Senatssaal

Die Tagesordnung der Mitgliederversammlung lautet:

1. Begrüßung; Vorstellung des Vorstands
2. Bericht des Vorstands
3. Mitgliedsbeiträge ab 2001; Umstellung auf Euro
4. Verschiedenes

Wie üblich sind auch Nichtmitglieder als Gäste willkommen.

Weitere Informationen finden Sie unter <http://www.dante.de/dante/mv/mv23>. Bitte melden Sie sich dort mit dem Formular zu den Tutorien und der Mitgliederversammlung an. Die Adresse für eine schriftliche Anmeldung und weitere Fragen lautet:

Fernuniversität in Hagen
Universitätsrechenzentrum
Stichwort MV23
58084 Hagen
E-Mail: mv23@dante.de

NTS Enters Alpha Test

NTS Team

Dear Colleagues,

You may be aware that for some years a development programme called “NTS” (for “New Typesetting System”) has been carried out, sponsored initially entirely by DANTE e.V. and more recently by private individuals and other user groups including GUTenberg.

The project is basically a re-engineering, from scratch, of the \TeX typesetting system using an object-oriented approach and implemented using Sun’s Java 2. The goals of the project are (1) to produce an exact re-implementation of \TeX using modern programming theory and methods, and (2) to accomplish this in such a way that the resulting system is easily extensible to accommodate alternative typesetting algorithms and paradigms.

After two years of hard work by our dedicated programmer Karel Skoupy, NTS has reached the stage at which we would like to solicit informed comment and criticism from members of NTS-L and other interested parties. In particular, although not yet quite complete, NTS is now entering alpha test.

Major omissions from current release:

- current-language support;
- hyphenation;
- alignments;
- file searching.

If you receive this message as a member of a \TeX user group, and not as a member of NTS-L, I strongly recommend that before volunteering to participate in the alpha-test programme, you first subscribe to the NTS-L list and read copies of the recent correspondence: this will clarify (for example) the preferred directory structure and the minimal level of Java which is required for successful compilation.

To subscribe to NTS-L, send a one-line message to:

`listserv@urz.uni-heidelberg.de`

containing (in the body, not the subject)

Subscribe NTS-L <*given name*> <*surname*>

If you are interested in becoming an alpha tester, and are willing to comply with the few conditions outlined below, please reply to this message via e-mail or via the link on the NTS web page (<http://nts.tug.org/>); I will then allocate you a username and password, and you will be granted access to the current NTS sources and support files.

Conditions

Access to the NTS alpha-test code is restricted to named individuals who have read and accepted these terms and conditions. Please indicate by return of e-mail whether you are willing to abide by these, in which case a username and password for your individual use will be allocated.

- (a) Access to the NTS alpha-test source code is granted solely for the purposes of evaluation; no warranty, either expressed or implied, is given concerning the reliability of this code, and use of this code is entirely at the risk of the alpha tester. In particular, the NTS group will accept no liability for any damage caused to any system on which the NTS code is compiled or run, no matter whether such damage is caused by the NTS code itself or by user error.
- (b) Access to the NTS alpha-test source code is granted on a strict non-disclosure basis. Re-distribution or publication of the code, in whole or in part, is explicitly prohibited.
- (c) Access to the NTS alpha-test source code is granted in order to obtain user feedback on its design, performance, robustness and implementation. It is a condition of access that a report on one or more of these elements be submitted electronically to `NTS-L@urz.uni-heidelberg.de`

Such a report should normally be made within six weeks of down-loading the code, and in all circumstances within three months.

I hope to hear from some of you soon!

Philip Taylor,
 Technical Director, NTS Project,
 for and on behalf of the NTS Team.

Bretter, die die Welt bedeuten

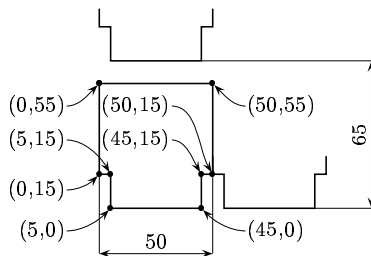
Von der Wiege bis zur Bahre: Formulare

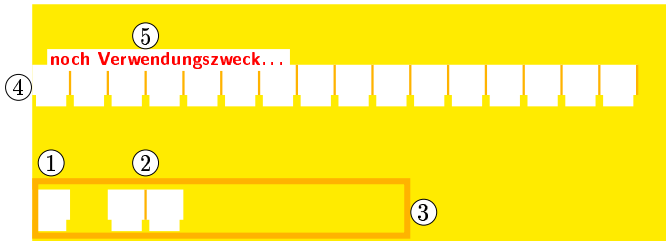
Rolf Niepraschk

Im folgenden Artikel wird anhand des Überweisungsformulars gezeigt, wie hochwertige Grafiken mit Hilfe des Makropakets `PSTricks` erzeugt werden können.

In einem Artikel über das `LATEX`-Paket `overpic` [3] hatte ich als Demonstrationsobjekt eine Grafik verwendet, auf der das bekannte Überweisungsformular dargestellt ist. Ich wurde nun gelegentlich gefragt: „Wie macht man sowas?“ oder „Welche Auflösung hatte dein Scanner?“ Im Folgenden soll einiges dazu gezeigt werden. Um es gleich vorweg zu nehmen: Einen Scanner hatte ich nicht verwendet. Ich entschied mich statt dessen für `PSTricks`-Makros [5], da mit ihnen alle benötigten geometrischen Formen leicht zu erzeugen sind und die gewünschte Ausgabe als EPS-Datei keine Probleme bereitet. Im Gegensatz zu gescannten Vorlagen kann das Ergebnis ohne Qualitätsverlust beliebig skaliert werden. Wenn möglich sollte man immer Grafiken bevorzugen, die unabhängig von der Auflösung des Ausgabegerätes sind.

Im ersten Schritt ermittelte ich die Abmessungen eines vorhandenen Formulars. Für die zu beschriftenden Felder findet man beispielsweise die folgenden Maße bzw. Koordinaten:





Die Zahlen sind Vielfache von 0,1 mm, was später eine ausreichend genaue Positionierung gewährleistet. Nachfolgend sollen stichpunktartig einige Details beleuchtet werden. Die Nummerierung der einzelnen Schritte (①, ② usw.) bezieht sich auf die Abbildung. Alle PSTricks-Anweisungen befinden sich innerhalb einer `pspicture`-Umgebung, die die Gesamtgröße von 106 mm×150 mm zu adressieren gestattet, wobei der Nullpunkt zur Vereinfachung in die linke untere Ecke des ersten Beschriftungsfeldes (Feld ①) gelegt wurde. Um nicht zu komplizierte Gebilde zeichnen zu müssen, wurde gleich zu Anfang ein Großteil der Grafik mit

```
\psframe*[linecolor=orangeL,linestyle=none](0,-15)(1350,665)
```

in hellem Orangeton eingefärbt. Die Sternform bedeutet hier: Fülle das gesamte geometrische Objekt mit der Farbe „`linecolor`“; zeichne also nicht nur einen Rahmen.

- ① Ein erstes Beschriftungsfeld wird an der Stelle der Koordinate 0,0 gezeichnet (der orangefarbene Untergrund wird mit weißer Farbe übermalt):

```
\rput[B1](0,0){\UU}
```

`\UU` ist ein eigenes folgendermaßen definiertes Makro:

```
\newcommand*{\UU}{\pspolygon*[linecolor=white]
(0,55)(50,55)(50,15)(45,15)(45,0)(5,0)(5,15)(0,15)}
```

- ② Zwei weitere Beschriftungsfelder mit einer senkrechten Trennlinie in dunklem Orange (Makro `\VV`) folgen:

```
\rput[B1](100,0){\UU}
\rput[B1](150,0){\UU}
\rput[B1](100,0){\VV}
\newcommand*{\VV}{\psline[linecolor=orangeD,linewidth=3,
linestyle=solid](50,15)(50,55)}
```

- ③ Ein Rahmen ebenfalls in dunklem Orange mit 0,8 mm Linienbreite wird um die Felder gezeichnet:

```
\psframe[linecolor=orangeD,fillstyle=none,linewidth=8]
(0,-10)(500,70)
```

- ④ An dieser Stelle findet das PSTricks-Makro `\multips` zur Erleichterung der Zeichenarbeit Verwendung:

```
\multips(0,165)(50,0){16}{\UU}
\multips(0,165)(50,0){16}{\VV}
```

Der erste Parameter ist die Startkoordinate, der zweite legt den Zuwachs in X- und Y-Richtung fest. Dann folgen die Anzahl der Wiederholungen und zuletzt das, was an diesen Positionen gezeichnet werden soll: zuerst alle weißen Felder, danach alle Trennlinien.

- ⑤ Die gerade gezeichneten Beschriftungsfelder werden oberhalb durch einen erklärenden Text in fetter roter Schrift ergänzt:

```
\rput[B1](20,220){\psframebox*[framesep=1pt,linecolor=white]
{\SmallRB noch Verwendungszweck\dots}}
```

Aus Platzgründen konnte hier beispielhaft nur ein Teil der verwendeten Anweisungen besprochen werden. Wer Lust hat, den gesamten Quelltext des Überweisungsformulars anzusehen, findet ihn unter der folgenden Internet-Adresse

<http://www.tug.org/applications/PSTricks/Various/ueberw.tex>

Die vorgestellten PSTricks-Makros sind nur ein kleiner Teil der vorhandenen Möglichkeiten. Einen guten Einblick in die Leistungsfähigkeit dieses Paketes geben die Beispiele aus [2], die auch im Quelltext verfügbar sind [4]. Weitere wertvolle Informationen gibt es auf Denis Girous Web-Seite zu PSTricks [1].

Literatur

- [1] Denis Girou: *PSTricks (by Timothy van Zandt)*; <http://www.tug.org/applications/PSTricks/>; 1999.
- [2] Michel Goossens, Sebastian Rahtz und Frank Mittelbach: *The L^AT_EX graphics companion*; Kap. 4, S. 95–166; Addison-Wesley Longman; Reading, Massachusetts; 1997.

- [3] Rolf Niepraschk: *Überschreiben erlaubt – das overpic-Paket; Die \TeX -nische Komödie*; 1/99, S. 27–29; 1999.
- [4] Sebastian Rahtz: *Test file for the PSTricks 97 distribution*; CTAN: `tex-archive/graphics/pstricks/doc/test-pst.tex`; Sept. 1998.
- [5] Timothy van Zandt und Denis Girou: *PSTricks*; CTAN: `tex-archive/graphics/pstricks/`; 1993, 1999.

Serienbriefe mit \LaTeX – Ein Erfahrungsbericht

Joachim Grotherr

Serienbriefe in größeren Auflagen bis zu 10 000 Stück bringen gängige Office-Pakete schnell an ihre Leistungsgrenze. Der Einsatz von \LaTeX führt zu einer erheblichen Steigerung der Druckgeschwindigkeit und eröffnet weitere professionelle Möglichkeiten.

Ich bin seit einigen Monaten im EDV-Bereich eines Dienstleisters im Lotto- und Lotteriebereich tätig. Hier werden fast täglich personalisierte Kundenbriefe in Auflagen zwischen 10 und 10 000 Stück hergestellt. Dabei handelt es sich um Rechnungen, Mahnungen, Gewinnmitteilungen usw.

Viele unsere Briefe sind „stark“ personalisiert, soll heißen, es wird nicht nur der Kundename, seine Adresse, die Kundennummer und eine geschlechtsspezifische Anrede in den Brief aufgenommen, er wird auch aus verschiedenen Textblöcken zusammengesetzt. So entsteht ein sehr individuelles Schreiben, das beispielsweise auf die vom Kunden gewünschte Zahlungsform (Zahlkarte oder Bankeinzugsgenehmigung), auf seine Gewinnsituation („Herzlichen Glückwunsch zu Ihrem Gewinn, Frau ...“), und sogar auf die Jahreszeit („Frohe Ostern/Pfingsten/Weihnachten wünschen wir ...“) Bezug nimmt.

Solche Serienbriefe in Microsoft Word herzustellen, ist durchaus bequem. Bei größeren Auflagen allerdings, bei uns etwa ab 1000 Stück, kann man beobachten, wie Word die Systemressourcen eines normalen PCs an sich reißt und das Drucksystem stark ausbremst. Es ist deprimierend, einem Drucker, der eigentlich 30 Seiten pro Minute schaffen könnte, bei vier Seiten pro Minute zuzusehen!

Da kommerzielle Druckbeschleunigungs-Software viel Geld kostet und Versuche mit anderer Textverarbeitungs-Software aus verschiedenen Gründen scheiterten, lag ein Versuch mit \LaTeX nahe. Wir gingen dabei so vor, wie es von Ulrike Fischer in [1] beschrieben wurde.

Als Datenquelle dient uns eine einfache Textdatei, deren Einträge durch Trennzeichen separiert sind. Mit Hilfe eines Perl-Scriptes werden aus der Datenquelle die von Frau Fischer so benannten Dateien `listen.sty` und `daten.txt` hergestellt. Dabei werden gleichzeitig auch alle Sonderzeichen aus der Quelle herausgefiltert, die in \LaTeX Probleme machen könnten. Hierbei erwiesen sich die regulären Ausdrücke von Perl als enorm praktisch. Die Generierung der beiden \LaTeX -Dateien aus einer Quelldatei mit etwa 7000 Kundensätzen dauert etwa eine Minute (Perl für Win32 von ActiveState mit NT 4.0 SP 5 auf einem AMD K6 III, 400 MHz, 128 MB RAM). Im weiteren Verlauf haben wir dann das gesamte \LaTeX -Dokument kompiliert. Die nach etwa 2500 Seiten aufgetretene Fehlermeldung „!TeX capacity exceeded, sorry [...]“ weist meiner Meinung nach tatsächlich auf einen zu kleinen Pufferspeicher hin, denn nach einer Aufteilung der Datenquelle in Portionen zu je 2000 Kundensätzen lief die Herstellung der `dvi`-Datei fehlerfrei.

Für die 2000 Serienbrief-Seiten benötigt \LaTeX (MiK \TeX 1.2 auf der gleichen Maschine wie oben) etwa zwei bis drei Minuten. Der Gebrauch von `dvips` zum Erzeugen der PostScript-Datei dauert ungefähr genauso lange. Nur das Schreiben der PostScript-Datei in die Drucker-Warteschlange, also in eine temporäre Datei auf dem Server, dauert etwa 15 Minuten, denn die `ps`-Datei wird bei uns für 2000 Briefe etwa 750 MB (!) groß. Dazu ist noch zu sagen, dass wir eine Novell-Druckumgebung auf einem 10 MBit-Ethernet benutzen.

Der relativ große Aufwand, insbesondere das Schreiben des Perl-Scriptes und der \LaTeX -Datei, die die eigentliche Formatierung des Briefes enthält, wurde belohnt durch unsere Drucker, die nun mit maximaler Geschwindigkeit arbeiten. Das bedeutet in unserem Fall bei großen Auflagen eine Versiebenfachung der Druckgeschwindigkeit!

Durch die Verwendung von \LaTeX und Perl haben wir außerdem weitere Optionen: Es ist nun möglich, neben den PostScript-Dateien beispielsweise PCL- oder PDF-Formate herzustellen und ihre Gebrauchsfähigkeit zu erkunden. Aber noch schöner finde ich die sich dadurch ergebende Plattformunabhängigkeit, so dass wir jetzt gute Betriebssysteme wie Linux in unsere Druckumgebung einbinden können.

Literatur

- [1] Ulrike Fischer: *Eine Schnittstelle zwischen Datenbanken und L^AT_EX*; *Die T_EXnische Komödie*; 4/98, S. 28–33; 1998.

Arial installieren

Peter Willadt

Layout-Vorschriften erzwingen gelegentlich die Verwendung einer nicht in einer Standard-T_EX-Distribution enthaltenen Schrift. Im Folgenden wird am Beispiel *Arial* beschrieben, wie eine solche Schrift installiert wird, wenn die benötigten **tfm**- und **vf**-Dateien bereits vorliegen.

Einstieg

Um mit T_EX eine Schrift benutzen zu können, die nicht im METAFONT-Format vorliegt, werden mehrere Dateien benötigt:

Eine **tfm**-Datei, damit T_EX die Abmessungen der Buchstaben kennt, eine **vf**-Datei, mit deren Hilfe der Druckertreiber die passenden Glyphen zu den von T_EX gesetzten Zeichen herausuchen kann, die eigentliche Schriftdatei (bei PostScript Type1 mit der Namensweiterung **pfb** oder **pfa**) und dann noch für die Benutzung mit L^AT_EX eine **fd**-Datei und eine **sty**-Datei.

Die T_EX-spezifischen Dateien können mit **fontinst**¹ selbst erstellt werden. Für einige gängige PostScript-Schriften liegen zum Glück auf CTAN schon fertige Pakete vor, so dass das Thema **fontinst** hier nicht weiter verfolgt wird.

Der Weg zur **pfb**-Datei führt normalerweise über einen Händler, da die meisten PostScript-Schriften nur gegen Barschaft erhältlich sind – zu Preisen ab 40 DM pro Schriftschnitt. Die Verwendung einer PostScript-Schrift kann

¹Das **fontinst**-Paket befindet sich auf der CTAN-CD-ROM 1 in dem Verzeichnis `/fonts/utilities`.

schnell ins Geld gehen, da typischerweise mehrere Schriftschnitte benötigt werden.

Für die folgende Installations-Prozedur gehe ich von einem TDS-System aus, bei dem `dvips` zum Weiterverarbeiten der TEX -Ausgabe verwendet wird. Windows-Benutzer sollten statt eines `/` einen `\` tippen, Linux-Benutzer statt `copy` und `move` die Befehle `cp` und `mv` eingeben.

Arial

Arial kommt aus dem Hause Monotype. Sie weist große Ähnlichkeiten mit der Helvetica (digitalisiert von Adobe) auf. Die Abmessungen der Zeichen sind absolut identisch, Unterschiede in der Form der Zeichen findet man mit gutem Auge beispielsweise beim `R` oder `ß`.

Helvetica und Arial sind große Schrift-Familien. Neben der gewöhnlichen Schriftweite in aufrechter und geneigter Ausführung gibt es noch in verschiedenem Maß gestauchte oder gedehnte Schnitte, ebenso Strichstärken von ultra leicht bis hin zum besonders dunklen, für Überschriften im Bild-Zeitungs-Stil gedachten Schnitt. Die vier mit Windows gelieferten Schriftschnitte sind als absolutes Minimum zu betrachten. Für aufwändigere Typographie sollten weitere Schnitte eingesetzt werden.

Der schnellste Weg zur „Arial“ besteht darin, Helvetica zu verwenden und darauf zu vertrauen, dass es niemand merkt. Helvetica ist in jedem PostScript-Drucker fest installiert. In einer typischen TEX -Installation ist die von der Firma URW gestiftete und im GhostScript-Paket enthaltene NimbusSans, ein nahezu perfekter Helvetica-Klon, eingebunden. Mit einem einfachen

```
\usepackage{helvet}
```

ist damit schon alles erledigt.

Da die Installation der echten Arial analog zu der vieler anderer PostScript-Schriften verläuft, möchte ich sie im Folgenden dennoch Schritt für Schritt beschreiben.

Installation in ein TDS-System

Die T_EX Directory Structure (TDS)² stellt eine hierarchisch gegliederte Struktur für die zu einem T_EX-System gehörenden Dateien dar. Nur so lässt sich bei den Tausenden von Dateien, die zu einer typischen T_EX-Installation gehören, noch der Überblick bewahren. An der Spitze der TDS-Hierarchie steht das Verzeichnis `texmf`. In vielen Fällen ist es sinnvoll, mehrere `texmf`-Bäume vorzuhalten, beispielsweise einen mit der T_EX-Distribution gelieferten oder den der CD-ROM T_EX-Live und zusätzlich einen, in den selbst installierte und selbst geschriebene Dateien kommen. Ein Update der T_EX-Distribution wird so sehr erleichtert.

Für die Installation einer Schrift werden Einträge im Verzeichnis `fonts` und im Verzeichnis `tex/latex` vorgenommen. Zudem sollte `dvips` noch etwas über die zusätzlichen Schriften erfahren. Die Konfigurationsdateien dieses Programmes liegen oftmals im Verzeichnis unter `dvips/config`. Haben Sie `texmf`-Bäume, installieren Sie die Konfigurationsdateien am besten in Ihren privaten Baum. Wo im Folgenden `/texmf` steht, sollten Sie sie also jeweils den Pfad zu Ihrem eigenen `texmf`-Verzeichnis verwenden.

Auf der CTAN-CD-ROM 1 liegt im Verzeichnis `fonts/psfonts/monotype` das Paket `arial.zip`. Zusätzlich benötigen Sie noch die `pfb`-Dateien. Erfreulicherweise hat Adobe die vier Standard-Schnitte der Arial in seinen Acrobat Reader 4.0 mit aufgenommen, so dass diese mitbenutzt werden können. Der Schritt zum Händler ist also nur erforderlich, wenn Sie die *ganze* Arial-Familie benötigen.

Beim Kopieren des Arial-Paketes auf die Festplatte und Auspacken mittels des Befehles `unzip arial` ergibt ein Verzeichnis `arial` mit den Unterverzeichnissen `dvips`, `tex`, `tfm` und `vf`. Der nächste Schritt besteht darin, die Dateien in diesen Unterverzeichnissen an die richtigen Stellen der `texmf`-Hierarchie zu bringen.

Zuerst wird ein Verzeichnis für die L^AT_EX-Dateien angelegt. Dann werden die zugehörigen Dateien in dieses Verzeichnis verschoben:

```
mkdir /texmf/tex/latex/monotype
move arial/tex/*.* /texmf/tex/latex/monotype
```

Jetzt gilt es, Verzeichnisse für die anderen Dateien anzulegen und diese zu kopieren.

² Dokumentiert auf der CTAN-CD-ROM 1 im Verzeichnis `/tds`

```

mkdir /texmf/fonts/tfm/monotype
mkdir /texmf/fonts/tfm/monotype/arial
copy arial/tfm/*.*/texmf/fonts/tfm/monotype/arial
mkdir /texmf/fonts/vf/monotype
mkdir /texmf/fonts/vf/monotype/arial
copy arial/tfm/*.*/texmf/fonts/vf/monotype/arial
mkdir /texmf/fonts/type1/monotype
mkdir /texmf/fonts/type1/monotype/arial
copy arial/dvips/ma1.map /texmf/dvips/config

```

Nun müssen die `pf`-Dateien aus den Tiefen der Acrobat-Installation an die passende Stelle gebracht werden. Bei dieser Gelegenheit sollten sie gleich umbenannt werden. In der $\text{T}_{\text{E}}\text{X}$ -Welt wird Karl Berrys *font naming scheme* verwendet, das alle Schriften auf achtstellige Namen abbildet. Die erste Stelle steht für den Schrift-Hersteller, die zweite und dritte geben den Namen der Schriftfamilie an, die weiteren Stellen dienen zur Unterscheidung der Schrift-Varianten. Adobe benennt die mit dem Acrobat Reader ausgelieferten Schriften vor allem mit Unterstrichen (`_`), eine Datei namens `AdobeFnt.lst` nennt die wirklichen Namen. Die kanonische Zuordnung von PostScript-Schriften zum *font naming scheme* finden Sie auf der CTAN-CD-ROM 2 im Verzeichnis `info/fontname` in einer nach dem jeweiligen Hersteller benannten Datei, beispielsweise `adobe.map` oder `monotype.map`.

Die zum Acrobat gehörigen Schriften liegen unter Windows typischerweise im Verzeichnis `/programme/adobe/acrobat 4.0/resource/font`. Die gewöhnliche Arial hört auf den Namen `_a_.....pfb`, fette Schnitte tragen ein `b` im Namen, kursive ein `i`, die fette schräge Arial heißt also `_abi_....pfb`.

<i>Schrift</i>	<i>Name Adobe</i>	<i>Name Berry</i>
Arial MT	<code>_a_.....pfb</code>	<code>ma1r8a.pfb</code>
Arial MT Italic	<code>_ai_.....pfb</code>	<code>ma1ri8a.pfb</code>
Arial MT Bold	<code>_ab_.....pfb</code>	<code>ma1b8a.pfb</code>
Arial MT Bold Italic	<code>_abi_....pfb</code>	<code>ma1bi8a.pfb</code>

Wenn Ihr Betriebssystem Links unterstützt, reicht es aus, einen Link von der einen Datei auf die andere zu legen. Im anderen Fall, muss kopiert werden, was am Beispiel der gewöhnlichen Arial so aussieht:

```

copy /programme/adobe/acrobat 4.0/resource/font/_a_.....pfb
  /texmf/fonts/type1/monotype/arial/ma1r8a.pfb

```

Bitte beachten Sie, dass der gesamte Eintrag in *einer* Zeile eingetippt werden muss. `dvips` sollte noch etwas über Arial erfahren. Dazu wird die Datei `texmf/dvips/config/config.ps` mit einem Editor geöffnet und am Ende eine Zeile des Inhalts `p+ ma1.map` angehängt. Die beim Auspacken der `zip`-Datei erstellten Verzeichnisse können gelöscht werden.

Jetzt bleibt nur noch, die Dateinamens-Datenbank der $\text{T}_{\text{E}}\text{X}$ -Installation aufzufrischen, indem beispielsweise `makelsr` eingetippt wird. Anschließend sollte die Installation getestet werden. Eine passende $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ -Datei könnte wie folgt aussehen:

```
\documentclass{article}
\usepackage{arial}
\begin{document}
\sffamily Das ist Arial.
\end{document}
```

Damit haben Sie die Sans-Serif-Schrift Ihres Dokumentes durch Arial ersetzt. Möchten Sie das ganze Dokument in Sans-Serif setzen, sollte in die Präambel noch

```
\renewcommand{\familydefault}{\sfdefault}
\linespread{1.05}
```

eingefügt werden. Zugleich wird der Zeilenabstand etwas aufgeweitet, was für Arial empfehlenswert ist.

Was bleibt?

Wir haben die Sans-Serif-Textschrift durch Arial ersetzt. Mathematikschriften, Serif-Textschrift, Schreibmaschinenschrift und Symbolschriften haben wir nicht angefasst, mangels kostenloser Verfügbarkeit haben wir die anderen Laufweiten der Arial nicht eingebunden. Auch die Verwendung anderer Ausgabeprogramme als `dvips` haben wir vernachlässigt.

Times und Courier mögen als begleitende Textschriften durchgehen, weil sie vielerorts verwendet werden, so dass der Leser sich daran gewöhnt hat – auch wenn diese Schriften nicht gut zueinander passen. Ebenso hat sich der Leser daran gewöhnt, dass beispielsweise für Überschriften, Fußnoten und ähnliches die Textschrift hoch oder herunter skaliert wird, anstatt einen besser angemessenen Schnitt zu verwenden.

Die Firma Micropress³ bietet eine zu Helvetica passende Mathematikschrift zum Kauf an.

Mein besonderer Dank gilt „Krümel“ für die Frage, die den Anstoß zu diesem Artikel gab, und Walter Schmidt für seine Hinweise und Anmerkungen.

In Reih und Glied

Gerd Neugebauer

In \LaTeX erhält man sehr viel Unterstützung, wenn es um das Setzen von Tabellen geht. Trotz der Möglichkeiten, die \LaTeX selbst schon mitbringt, gibt es einige Wünsche, die erst durch Zusatzpakete einfach zu realisieren sind. Es ist nicht immer ratsam, alle Register zu ziehen. Auch hier bewahrheitet es sich, dass es auf einen gezielten und überlegten Einsatz ankommt, wenn man ein ansprechendes und professionell aussehendes Ergebnis erzielen will.

Einleitung

Wenn man einen essenziellen Unterschied zwischen Desktop-Publishing-Programmen und dem Textsatzsystem \LaTeX herausstellen will, dann gilt normalerweise die Regel, dass es mit \LaTeX relativ einfach ist, ein typographisch ansprechendes Ergebnis zu erzielen, während es relativ viel Aufwand erfordert, ein wirklich schlechtes Ergebnis zu erzielen. Bei Desktop-Publishing-Programmen ist es normalerweise genau umgekehrt.

Diese Eigenschaft von \LaTeX kommt unter anderem daher, dass der Anwender nicht mit einer Vielzahl von Optionen alleine gelassen wird, sondern dass es Vorgaben gibt, die schon ein ansprechendes Erscheinungsbild ergeben. Die Änderungen dieser Vorgaben sind auch möglich, jedoch meistens nicht so einfach, wie das Herumklicken in irgendwelchen Menüs bei DTP-Programmen.

³Demo-Version als Bitmap-Schrift mit 300 dpi auf der CTAN-CD-ROM 1 im Verzeichnis `/fonts/micropress/hvmath`, Demos und Online-Kaufmöglichkeit unter <http://www.micropress-inc.com/samples/hfonts.htm>.

Tabelle 1: Schriftgrade

Grad	Größe		
	pt	mm	in
Nonpareille	6	2,1088	0,083022
Petit	8	2,8117	0,110696
Borgis	9	3,1631	0,124533
Korpus (Garamond)	10	3,5146	0,138370
Cicero	12	4,2175	0,166044
Mittel	14	4,9204	0,193718
Tertia	16	5,6234	0,221392
Text	20	7,0292	0,276740
Zweicicero	24	8,4350	0,332088
Doppelmittel	28	9,8409	0,387436
Dreicicero	36	12,6526	0,498132

Es gibt jedoch auch in \LaTeX Bereiche, wo diese Regel nicht gilt. Ein solcher Bereich ist der Tabellensatz. \LaTeX stellt alles bereit, um beliebige rechteckige Anordnungen von Information auf der Seite zu organisieren. Zusätzlich können diese Tabellen fast beliebig mit Linien „verunziert“ werden. Einige Pakete helfen in dieser Richtung sogar noch nach.

Andererseits kann man die ansprechendsten Ergebnisse erzielen, indem man solche Stilmittel nur sparsam, dafür aber gezielt einsetzt. Anhand eines Beispiels werden ich mich im Folgenden diesem Ideal annähern. Auf dem Weg dahin werde ich einige Möglichkeiten aufzeigen und bewerten. Ich hoffe, dass am Ende klar wird, dass auch in diesem Bereich weniger mehr sein kann.

Terminologie

Ich fange mit einer kleinen Beispieltabelle an, deren Inhalt auch in den weiteren Varianten der gleiche bleiben wird. Dargestellt sind die Namen von Schriftgraden, die [5, 2] entnommen wurden, sowie ihre Größen in Punkt (pt), Millimeter (mm) und Inch (in). Bei der Umrechnung wurden die von \TeX verwendeten Umrechnungsfaktoren zugrundegelegt (siehe [3]).

Das erste Beispiel ist in der Tabelle 1 zu sehen. Bevor ich mich den typographischen und ästhetischen Aspekten zuwende, sollen daran einige Begriffe erläutert werden. Eine Tabelle kann prinzipiell in drei Bereiche aufgegliedert werden. Der Tabellenkopf geht über die gesamte Tabellenbreite und enthält die dort stehenden Erläuterungen zur Funktion der Spalten. Der Kopf kann dabei auch aus mehreren Zeilen bestehen.

Die Vorspalte ist die linke Spalte unterhalb des Kopfes. Hierin stehen die Erläuterungen zur Funktion der Tabellenzeilen. Der Rest wird als Fuß bezeichnet, in dem die Werte in Zellen oder Feldern enthalten sind. Diese Bezeichnungen sowie viele weitere nützliche Überlegungen zu Tabellen sind in [2] zu finden.

Nachdem ich nun einige Bezeichnungen vermittelt habe, kann ich mich den \TeX nischen Aspekten zuwenden. Zuerst fällt auf, dass die Tabelle ziemlich viele Linien enthält. Zusätzlich werden auch noch Doppellinien und ein Doppelrahmen benutzt. Gesetzt wurde die Tabelle mit der `tabular`-Umgebung. Da es damit nicht möglich ist, den äußeren Doppelrahmen ungebrochen zu realisieren, wurde die gesamte Tabelle noch einmal in eine `\fbox` eingeschlossen. Den gleichen Effekt hätte ich auch mit dem Paket `hhline` von David Carlisle erreichen können. Dieses, wie alle im Folgenden erwähnten Pakete sind auf den CTAN-Servern in dem Verzeichnis `tex-archive/macros/latex/required/tools` zu finden. Da solche überflüssigen Linien aber gerade vermieden werden sollen, will ich nicht näher darauf eingehen.

Ein Kommentar ist noch zu der höheren, zweizeiligen Tabellenzeile nötig. Da die Vorspalte in dem Format `p{20mm}` gesetzt wurde, werden die folgenden Zeilen an der ersten Zeile der umbrochenen Zelle ausgerichtet. Leider bietet \LaTeX hier keine einfache Möglichkeit, diese Ausrichtung zu ändern. Erst einige Pakete können hier Abhilfe schaffen, wie wir später noch sehen werden.

Einige positive Eigenschaften an dieser Tabelle sollen allerdings nicht unerwähnt bleiben. Die Doppellinien innerhalb der Tabelle stehen wenigstens an „logischen“ Stellen, nämlich an den Trennstellen zwischen Kopf, Vorspalte und Fuß. Die Maßangaben erscheinen richtigerweise im Kopf und nicht im Fuß.

Weniger . . .

Nachdem ich gezeigt habe, dass die Tabelle mit Linien überladen ist, werden in der nächsten Fassung die Doppellinien durch einfache Linien ersetzt und

Tabelle 2: Schriftgrade

Grad	Größe		
	pt	mm	in
Nonpareille	6	2,1088	0,083022
Petit	8	2,8117	0,110696
Borgis	9	3,1631	0,124533
Korpus (Garamond)	10	3,5146	0,138370
Cicero	12	4,2175	0,166044
Mittel	14	4,9204	0,193718
Tertia	16	5,6234	0,221392
Text	20	7,0292	0,276740
Zweicicero	24	8,4350	0,332088
Doppelmittel	28	9,8409	0,387436
Dreicicero	36	12,6526	0,498132

die horizontalen Linien im Fuß und in der Vorspalte ganz fallen gelassen. Das Ergebnis ist in der Tabelle 2 zu sehen.

Weitere Änderungen betreffen die Abstände. Der Tabellenkopf war in der Tabelle 1 viel zu eng gesetzt. Dieses Problem ist jetzt gelöst worden. Hierzu diente eine unsichtbare Box, die über und unter den Text hinausragt. Diese wurde mit einem Befehl der folgenden Art realisiert:

```
\rule[-2ex]{0pt}{5.5ex}
```

Dieser Befehl setzt ein gefülltes Rechteck der Höhe 5,5ex und der Breite 0pt, sodass dieses Rechteck eigentlich unsichtbar ist. Es hat aber doch den Effekt, die Höhe und die Tiefe der umgebenden Box zu beeinflussen. Mit dem optionalen Argument wird das Rechteck um 2ex nach unten verschoben.

Ein anderes Problem, das ich in der Tabelle 1 noch hatte, war die vertikale Ausrichtung der doppelt hohen Zeile. Um diesem Problem zu Leibe zu rücken, benutze ich das Paket `array`, das von Frank Mittelbach stammt. Dieses Paket erweitert die Möglichkeiten der Umgebungen `tabular`, `tabular*` und `array`. Zu den Möglichkeiten, die von dem `array`-Paket geboten werden,

gehört unter anderem eine größere Zahl von Absatzformaten. Diese kann ich jetzt einsetzen.

In der Vorspalte soll es im Prinzip möglich sein, mehrere Zeilen beschreibenden Textes unterzubringen. Im normalen \LaTeX nach L^AT_EX nach Lamport [3] steht dafür das Format `p` zur Verfügung. Der Nachteil ist allerdings, dass der Text darin linksbündig gesetzt wird. Will man das ändern, so müssen man in jeder Zelle eingreifen, wie das bereits in der Tabelle 1 geschehen ist. Mit dem Paket `array` lässt sich das aber vereinfachen. Ich nehme statt des Formats `p` die folgende Konstruktion:

```
>\raggedright}m{6em}
```

Diese Konstruktion enthält zwei Teile. Das Format `m` sorgt dafür, dass die Zelle in einer `\parbox` der angegebenen Breite gesetzt wird. Dieses Format bewirkt allerdings im Gegensatz zum Format `p`, dass die Zelle vertikal zentriert erscheint, womit die folgenden Zellen dieser Zeile besser positioniert werden. Das Konstrukt `>\raggedright}` sorgt nun noch dafür, dass der Zelleninhalt linksbündig gesetzt wird, wobei der rechte Rand nicht ausgeglichen wird, also flattert, indem das Makro `\raggedright` vor dem Inhalt jeder Zelle eingefügt wird. Als Nachteil habe ich immer noch in Kauf zu nehmen, dass jetzt die Breite der Spalte fest vorgegeben ist und nicht von \LaTeX anhand des Inhalts bestimmt wird, auch wenn sie sich jetzt durch die Verwendung der Maßeinheit „em“ der Schriftgröße anpasst.

Doch an dieser Stelle kann man mit einem kleinen Trick nachhelfen, der zwar nicht die ganze, aber doch einen Teil der Arbeit abnimmt. Man kann einfach \TeX veranlassen, einen Text zu vermessen. Das geht beispielsweise durch das Abspeichern der Breite des Textes in einem Längenregister.

```
\newlength{\Breite}
\settowidth{\Breite}{Doppelmittel}
```

Nun kann man anstelle der festen Länge `20mm` das Register `\Breite` eintragen. Mit dieser Konstruktion muss man nicht mehr die numerische Zellenbreite raten, sondern „nur“ noch den breitesten Zelleninhalt.

Einige der schlimmsten Probleme mit der Tabelle 1 habe ich damit korrigiert. In anderer Hinsicht lässt die Tabelle allerdings noch zu wünschen übrig. Dies werde ich in folgenden Abschnitten noch zu verbessern haben.

Tabelle 3: Schriftgrade

<i>Grad</i>	<i>Größe</i>		
	<i>pt</i>	<i>mm</i>	<i>in</i>
Nonpareille	6	2,1088	0,083022
Petit	8	2,8117	0,110696
Borgis	9	3,1631	0,124533
Korpus (Garamond)	10	3,5146	0,138370
Cicero	12	4,2175	0,166044
Mittel	14	4,9204	0,193718
Tertia	16	5,6234	0,221392
Text	20	7,0292	0,276740
Zweicicero	24	8,4350	0,332088
Doppelmittel	28	9,8409	0,387436
Dreicicero	36	12,6526	0,498132

... ist mehr

Ein erstrebenswertes Ziel in Tabellen ist es, die Spalten im Fuß gleich breit zu machen. Dies ist insbesondere dann wichtig, wenn eine senkrechte Lineatur verwendet wird, weil in diesem Fall der Leser sehr leicht unterschiedliche Breiten als Störung der Harmonie empfindet. Da ich aber auf dem Weg bin, die senkrechten Linien zu beseitigen, wird diese Empfehlung für unsere Tabelle keine entscheidende Rolle spielen. Oftmals besteht beim Tabellensatz auch der Zwang, eine gewisse Tabellenbreite nicht zu überschreiten, der solch einen verschwenderischen Umgang mit den Spaltenbreiten verbietet.

Trotzdem habe ich diese Forderung in der nächsten Verfeinerung – der Tabelle 3 – für die letzten beiden Spalten realisiert. Wie kann man nun diesen Effekt erreichen? Eine Möglichkeit besteht darin, beispielsweise die Köpfe der Spalten in Boxen gleicher Breite einzuschließen. In diesem Fall muss man aber deren Breite wissen. Besser ist es da schon, wenn L^AT_EX uns die Ermittlung dieser Breite abnimmt. Dazu verwende ich das Paket `tabularx` von David Carlisle. Dieses Paket stellt die Umgebung `tabularx` bereit, die ähnlich wie die Umgebung `tabular*` wirkt, nämlich die Tabelle auf eine vorgegebene Breite einzurichten. Allerdings wird dazu nicht wie in `tabular*` der Spalten-

zwischenraum gedehnt, sondern die Breite von gewissen Spalten angepasst. Diese dehnbaren Spalten werden mit dem Format `X` markiert. Der verbliebene Raum nach dem Setzen der übrigen Spalten und der Zwischenräume wird dann gleichmäßig unter den mit dem Format `X` gesetzten Spalten aufgeteilt.

Bisher habe ich das Format `r` verwendet, um die Zahlen untereinander auszurichten. In unserem Beispiel haben die Zahlen einer Spalte die gleiche Anzahl von Nachkommastellen. Deshalb ist dieses Verfahren vollkommen ausreichend. Wenn man allerdings in einer Spalte Zahlen mit stark schwankender Anzahl von Nachkommastellen zu setzen hat, dann funktioniert dieser Ansatz nicht mehr. Die manuelle Lösung wäre, die fehlenden Nachkommastellen mit Nullen aufzufüllen. Diese kann man noch mit dem `TEX`-Makro `\phantom` unsichtbar machen – `\phantom` setzt sein Argument in eine Box und gibt diese Box ohne den Inhalt zurück.

Mit diesem Verfahren wäre es allerdings notwendig, für jede Zelle diesen Aufwand zu betreiben. Es gibt aber das Paket `dcolumn` von David Carlisle, das neue Format `D` dafür bereitstellt. Damit kann man angeben, wie viele Dezimalstellen vorgegeben sind und an welchem Zeichen diese ausgerichtet werden müssen. Der Rest sollte dann automatisch passieren.

Feinheiten

Die Tabelle in dem vorangegangenen Abschnitt ist eigentlich schon ganz ansehnlich geworden. Es gibt allerdings einige Überlegungen, wie man diese Tabelle noch ein bisschen verbessern könnte. Dazu stellt Simon Fear in [1] sein Paket `booktabs` vor. Bei dieser Gelegenheit wird auch dargelegt, dass es angeraten ist, weitere Feinheiten beim Tabellensatz zu berücksichtigen.

Einen Teil davon haben ich schon in den vorangegangenen Verfeinerungsstufen eingebaut – nämlich die Abstandskorrekturen. Die Kopflinie, das ist die Linie über dem Kopf, hat etwas zusätzlichen Abstand vom Kopfmateriale. Das Gleiche gilt für die Halslinie, die Linie zwischen Kopf und Fuß, und die Fußlinie, die Linie unter dem Fuß. Dies habe ich durch unsichtbare `\rule`-Konstruktionen erreicht. In dem Paket `booktabs` gibt es für diese Linien jeweils spezielle Makros: `\toprule`, `\midrule` und `\bottomrule`. Diese Makros sorgen für den benötigten zusätzlichen Abstand, wobei die Größe des jeweiligen Abstandes durch Optionen angegeben werden kann. Allerdings sind die Vorgaben meistens recht passend, sodass man an dieser Stelle nicht mehr eingreifen muss.

Tabelle 4: Schriftgrade

<i>Grad</i>	<i>Größe</i>		
	<i>pt</i>	<i>mm</i>	<i>in</i>
Nonpareille	6	2,1088	0,083022
Petit	8	2,8117	0,110696
Borgis	9	3,1631	0,124533
Korpus (Garamond)	10	3,5146	0,138370
Cicero	12	4,2175	0,166044
Mittel	14	4,9204	0,193718
Tertia	16	5,6234	0,221392
Text	20	7,0292	0,276740
Zweicicero	24	8,4350	0,332088
Doppelmittel	28	9,8409	0,387436
Dreicicero	36	12,6526	0,498132

Mit den Makros `\toprule`, `\midrule` und `\bottomrule` wird nicht nur der Abstand verändert, sondern auch die Liniestärke angepasst. Dieser Effekt ist im direkten Vergleich von Tabelle 3 und Tabelle 4 zu sehen, wobei letztere mit Hilfe des Paketes `booktabs` gesetzt wurde. Nebenbei haben ich durch die Verwendung dieser Makros unsere Eingabe von der visuellen Hacker-Lösung in Richtung auf ein „logisches Markup“, wie es der Philosophie von \LaTeX entspricht, verbessert.

Noch weniger?

Mit der bisher erarbeiteten Variante für den Tabellensatz habe ich schon ein Layout erreicht, das professionellen Ansprüchen gerecht werden kann. In diesem Abschnitt habe ich die Lineatur noch weiter reduziert. Hier will ich zeigen, dass es ausreichen kann, die Tabelle mit Hilfe geeigneter Schrift-Auswahl und passend gewählten Abständen zu strukturieren.

Das Ergebnis dieser radikalen Abmagerungskur ist in der Abbildung 5 zu sehen. Hier werden überhaupt keine Linien mehr verwendet. Stattdessen ist der Kopf durch ein einfaches `\emph` ausgezeichnet und durch zusätzlichen Abstand vom Fuß getrennt.

Tabelle 5: Schriftgrade

<i>Grad</i>	<i>Größe</i>		
	<i>pt</i>	<i>mm</i>	<i>in</i>
Nonpareille	6	2,1088	0,083022
Petit	8	2,8117	0,110696
Borgis	9	3,1631	0,124533
Korpus (Garamond)	10	3,5146	0,138370
Cicero	12	4,2175	0,166044
Mittel	14	4,9204	0,193718
Tertia	16	5,6234	0,221392
Text	20	7,0292	0,276740
Zweicicero	24	8,4350	0,332088
Doppelmittel	28	9,8409	0,387436
Dreicicero	36	12,6526	0,498132

Dadurch erreiche ich, dass die Tabelle noch weiter zurücktritt. Die Tabelle ist weniger auffällig als die vorhergehenden Varianten, bei denen die Lineatur das Auge des Lesers immer mehr oder weniger stark auf die Tabelle gezogen hat.

Ob diese extreme Unauffälligkeit der Tabelle gewünscht ist, muss für jedes Dokument einzeln entschieden werden. Wichtig ist aber zu betonen, dass diese Entscheidung für alle Tabellen eines Dokuments einheitlich getroffen werden muss. Unter diesem Aspekt erscheint diese Lösung dann vielleicht zu minimalistisch.

Strukturierung von Tabellen

Bisher habe ich in dem Beispiel eine recht einfache Tabelle gesehen. Es kann aber sein, dass eine Tabelle eine innere Struktur aufweist, die es hervorzuheben gilt. In unserer Tabelle ist dies in den drei Spalten des Fußes der Fall, die eine gemeinsame Überschrift haben. Ein anderes Beispiel bekommt man, wenn man mehrere Zeilen jeweils unter einem Oberbegriff zusammenfasst – eigentlich hat man dann eine dreidimensionale Tabelle, die man aber auf dem ebenen Papier schlecht auf diese Weise darstellen können.

In dieser Situation kommt man leicht in Versuchung, doch einmal vertikale Linien zur Abgrenzung einzusetzen. Dem sollte man aber nicht nachgeben. In den meisten Fällen kann man die Strukturierung einfach durch einen zusätzlichen kleinen Abstand kenntlich machen. Falls das nicht hilft, ist auf jeden Fall noch einmal die logische Struktur der Tabelle zu überdenken.

Eine weitere Möglichkeit der Strukturierung einer Tabelle ist das Hinterlegen von Spalten oder Zeilen mit Farbe oder einem Graumuster. Dabei sollte man sich sehr genau überlegen, für welches Ausgabegerät das fertige Dokument konzipiert wird. Beispielsweise fallen Grauraster auf Laserdruckern niedriger Auflösung (300/600 dpi) zu grob aus. Das heißt, dass die Druckerpunkte, aus denen der Graueindruck entsteht, so groß sind, dass die Rezeption der darüber liegenden Schrift gestört wird.

Ab einer mittleren Auflösung des Ausgabegeräts (1 200 dpi) kann man die Strukturierung der Tabelle mit solchen hinterlegten Zeilen oder Spalten hervorheben. Allerdings ist dann zu beachten, dass die Hervorhebung in gleichmäßigen Abständen vorgenommen wird. Hierbei kommt das gleiche Argument zum Tragen, das auch den gleichen Spaltenbreiten den Vorzug gibt, nämlich die dadurch erreichte Harmonie.

Schließlich ist noch zu bedenken, dass der schönste Graueindruck verloren gehen kann, wenn das Dokument am Ende mit einem Kopierer vervielfältigt wird. Dort zeigen sich manchmal recht unschöne Ergebnisse. Schon aus diesem Grund ist es ratsam, sehr sparsam mit solchen Kunstgriffen umzugehen.

Schriftauswahl in Tabellen

Bisher wurde von der äußeren Gestaltung von Tabellen gesprochen. Nun will ich mich noch einmal den Schriftarten zuwenden. Die erste Wahl für die Schrift im Fuß der Tabelle sollte die Grundschrift des Dokumentes sein. Dabei sollte auch die gleiche Schriftgröße wie im restlichen Dokument gewählt werden. Natürlich ist das ein Ideal, das man nicht immer realisieren kann. Insbesondere wenn die Tabelle eine gewisse Größe überschreitet, ist das nicht mehr machbar.

In solchen Situationen gibt es mehrere Auswege. Man kann auf einen kleineren Schriftgrad ausweichen, man kann eine andere Schrift nehmen oder versuchen, durch Drehen der Tabelle oder durch den Einsatz von mehrseitigen Tabellen die Informationsflut unterzubringen. Zu den letzten beiden Möglich-

Tabelle 6: Schriftgrade

<i>Grad</i>	<i>Größe</i>		
	<i>pt</i>	<i>mm</i>	<i>in</i>
Nonpareille	6	2,1088	0,083022
Petit	8	2,8117	0,110696
Borgis	9	3,1631	0,124533
Korpus (Garamond)	10	3,5146	0,138370
Cicero	12	4,2175	0,166044
Mittel	14	4,9204	0,193718
Tertia	16	5,6234	0,221392
Text	20	7,0292	0,276740
Zweicicero	24	8,4350	0,332088
Doppelmittel	28	9,8409	0,387436
Dreicicero	36	12,6526	0,498132

keiten werden ich später noch kommen. An dieser Stelle soll erst einmal über die Variationen der Schrift nachgedacht werden.

Eine Stelle, die in Tabellen Probleme bereiten kann, ist der Tabellenkopf. Hier ist es meistens nötig, zur Erklärung mehr Material unterzubringen als in den darunter stehenden Zellen. Deshalb ist es hier angeraten, einen kleineren Schriftgrad zu wählen. Dabei sollte allerdings nur um eine Vergrößerungsstufe, nur in Ausnahmefällen auch zwei, zurückgegangen werden. Das heißt, dass bei einer Grundschrift in `\normalsize` ein Kopf in `\small` angebracht ist.

Eine weitere Möglichkeit, um Tabellen komprimierter setzen zu können, ist der Einsatz anderer Schriften. Diese müssen aber auf jeden Fall zu der Grundschrift des Dokuments passen. Für viele Schriften gibt es sogenannte „magere“ Schnitte. Diese fallen nicht so breit aus und sind daher geeignet, in Tabellen eingesetzt zu werden.

Eine andere Möglichkeit ist der Einsatz von serifenlosen Schriften für Tabellen. Dies wurde beispielsweise in der Tabelle 6 getan. Zu solchen Schriften gibt es manchmal schmalere laufende Varianten – „light“ oder „condensed“. Diese erlauben es, die Information auf kleinerem Raum unterzubringen. Aber auch hier muss unbedingt darauf geachtet werden, dass diese Schriften zu der gewählten Grundschrift passen.

Die Computer-Modern-Schriften von T_EX enthalten von Hause aus keine mageren Schrift-Schnitte. Aber bei anderen Schriften, insbesondere im PostScript-Bereich, sind diese durchaus üblich. Als Beispiele seien hier die serifenlosen Schriften Helvetica, Frutiger, Univers und Gill Sans genannt, deren „light“- oder „condensed“-Schnitte im Tabellensatz Anwendung finden können.

Wie auch immer die Auswahl für die Schrift einer Tabelle ausfällt, so gibt es doch ein übergeordnetes Gebot, das unbedingt bedacht sein will. Dies ist die Konsistenz innerhalb eines Dokumentes. Das bedeutet in diesem Fall, dass die Festlegung der Schriften, der Schriftgrade und der Lineatur immer für *alle* Tabellen in einem Dokument zu gelten hat. Es muss unbedingt vermieden werden, beispielsweise eine Tabelle in der Grundschrift – wie Computer Modern Roman – zu setzen, während die nächste Tabelle eine andere Schrift – wie zum Beispiel eine serifenlose Schrift – verwendet.

Einige Regeln zum Inhalt

Den Einsatz von Abkürzungen sollte man vermeiden. Im laufenden Text stören Abkürzungen den Lesefluss. Dieses Argument trifft auf Tabellen aber nicht zu. Deshalb kann es zulässig und sinnvoll sein, in Tabellen Abkürzungen einzusetzen. Insbesondere der Tabellenkopf neigt leicht dazu, die Tabelle breiter als gewünscht werden zu lassen. Hier können Abkürzungen helfen, dieses Problems Herr zu werden. Man kann sich auch überlegen, ob es nicht besser ist, die Tabelle um 90° zu drehen. Damit ist aber nicht der Einsatz eines Paketes zum Drehen von Text gemeint, sondern die logische Drehung, die aus dem Kopf die Vorspalte werden lässt und umgekehrt.

Drei andere kleine Regeln habe ich in [1] gefunden. Mir selbst wäre es nicht in den Sinn gekommen, dies zu erwähnen, weil sie so selbstverständlich erscheinen. Demnach sind Einheiten immer im Tabellenkopf (oder der Vorspalte) unterzubringen und niemals im Fuß. Die monotone Wiederholung von Information im Fuß verbraucht Platz und lenkt vom Wesentlichen ab.

In einer Tabelle sollte man niemals Wiederholungszeichen benutzen. Wenn eine Zelle den gleichen Wert wie die darüber liegende Zelle enthält, so kann man dies durch Gruppierung kenntlich machen. In diesem Fall wird durch die Strukturierung einfach die entsprechende Zelle leer gelassen. Die einzige verbleibende Möglichkeit ist die Wiederholung des Wertes.

Schließlich ist noch eine Unart zu rügen, die wohl aus dem Einsatz von Programmiersprachen inspiriert wurde. In vielen Programmiersprachen ist es nämlich möglich, bei einer Zahl die führende Null vor dem Komma oder Dezimalpunkt wegzulassen. Dieser Unart sollte man in Tabellen – und auch generell in gut gesetzten Dokumenten – nicht frönen. Das heißt, dass beispielsweise bei 0,375 die führende 0 nicht weggelassen werden darf.

Zusammenfassung

Als Zusammenfassung will ich hier noch einmal die in diesem Beitrag erarbeiteten Regeln und Vorschläge stichwortartig zusammenfassen, wobei auch einige Betrachtungen nachgereicht werden, die bisher vielleicht zu kurz gekommen sind.

- Mehrfachlinien sind in Tabellen auf jeden Fall zu vermeiden. Damit entfallen auch alle Probleme, die \LaTeX damit haben könnte.
- Vertikale Lineatur ist auf jeden Fall zu vermeiden.
- Die scheinbare Notwendigkeit einer senkrechten Linie ist ein Indiz, dass die Tabelle schlecht strukturiert ist oder in zwei Tabellen geteilt werden sollte.
- Die Kopflinie und die Fußlinie sollten, falls vorhanden, dicker als die Halslinie und die weiteren Linien gesetzt werden.
- Die Vorspalte wird linksbündig gesetzt.
- Meistens ist es angebracht, die Kopfzellen zentriert zu setzen.
- Horizontale Lineatur im Fuß sollte vermieden werden.
- Der Kopf kann durch eine Schriftumschaltung ausgezeichnet werden. Dabei ist `\emph` meistens ausreichend. Halbfette und fette Schriftschnitte sollten, wie auch im laufenden Text, vermieden werden.
- Werden in einer Tabelle Fußnoten benötigt, so ist der entsprechende Fußnotentext an das Ende der Tabelle und nicht am Ende der Seite zu setzen. Dies kann man erreichen, indem man die Tabelle in eine `minipage` einschließt.
- Tabellen werden in \LaTeX als Floats mit `table` gesetzt. Diese erhält eine Beschriftung in Form einer `\caption`. Die Tabelle muss im Text verankert

werden – das heißt im Text muss ein Bezug auf die Tabellennummer, deren symbolischer Name mit `\label` definiert wird, vorhanden sein.

- Die Tabellen sollten nicht im laufenden Text untergebracht werden, sondern auf eine eigene Seite oder an den Rand der Seite – vorzugsweise den oberen Rand – geschoben werden.

Mit diesen Regeln sollte es nicht allzu schwer sein, ansprechende Tabellen zu setzen und Fallen zu vermeiden, die \LaTeX auf diesem Feld allzu leichtfertig vor uns öffnet. Auf der anderen Seite sind Regeln auch dazu da, gebrochen zu werden. Deshalb kann und sollte man sich darüber hinwegsetzen, wenn dies aus guten Gründen erforderlich sein sollte. Dies bedeutet aber, dass man sich darüber Gedanken machen muss und nicht einfach aus Lust an den \TeX nischen Möglichkeiten alle Optionen in einem Text auf einmal ausprobiert. Einige weitere Anregungen kann man [4] entnehmen, falls die hier vorgestellten Pfade verlassen werden sollen.

Literatur

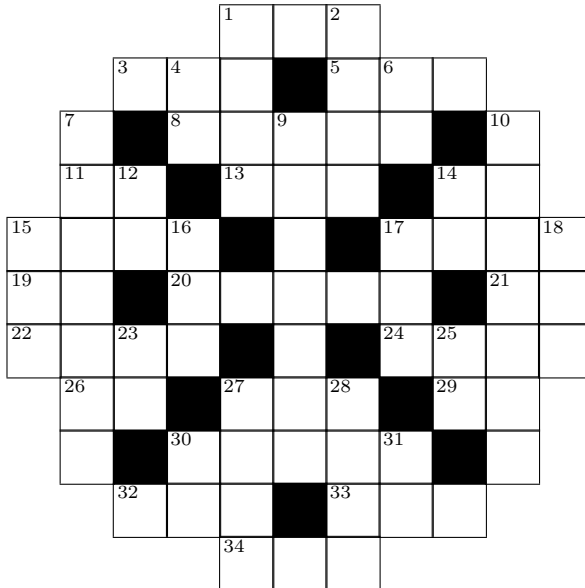
- [1] Simon Fear: *Publication quality tables in \LaTeX* ; Teil des `booktabs`-Paketes; Nov. 1995; CTAN: `tex-archive/macros/latex/supported/booktabs/booktabs.dtx`.
- [2] Jürgen Gulbins und Christine Kahrmann: *Mut zur Typographie – Ein Kurs für DTP und Textverarbeitung*; Edition Page; Springer-Verlag; Berlin, Heidelberg, New York; 1993.
- [3] Leslie Lamport: *\LaTeX A Document Preparation System*; Addison-Wesley Publishing Company; Reading, Mass.; 2. Aufl.; 1994.
- [4] Axel Reichert: *Satz von Tabellen*; `ftp://dante.ctan.org/tex-archive/info/german/tabsatz/`; Febr. 1999.
- [5] Jan Tschichold: *Erfreuliche Drucksachen durch gute Typographie – Eine Fibel für jedermann*; Maro-Verlag; Augsburg; 1988.

TeX-Rätsel

Groß- und Kleinbuchstaben werden nicht unterschieden. Sonderzeichen wie Ziffern oder \ werden nicht eingetragen.

Waagrecht: 1 mathematische Funktion (ggT) 3 ψ 5 mathematische Funktion 8 geneigte Schriftfamilie (PlainTeX) 11 METAFONT-Schriftfamilie (veraltet) 13 Logarithmus 14 Å 15 Uhrzeit 17 Abstandsangabe (PlainTeX) 19 Höhe einer Box 20 BibTeX-Feld 21 handschriftliche METAFONT-Schrift 22 Maßeinheit 24 BibTeX-Jahresfeld 26 Alternative 27 Großer Begrenzer 29 L^AT_EX 2_ε-Paket für externe Referenzen 30 \mathcal{J} (*A_MS*) 32 \ggg (*A_MS*) 33 Name of the game 34 L^AT_EX-Paket zur Unterstützung der Versionskontrolle

Senkrecht: 1 Typograph (Eric), auch Schriftfamilie 2 † 4 ß 6 fontabhängige TeX-Maßeinheit 7 BibTeX-Feld (Aufgabe) 9 Schriftattribut (PlainTeX) 10 L^AT_EX-Paket 12 metrische Maßeinheit 14 æ 15 L^AT_EX 2_ε-Paket 16 η 17 BibTeX-Feld (Schlüssel) 18 BibTeX-Stil für das Journal „Nucleic Acid Research“ 23 Ende der Spalten (PlainTeX) 25 fontabhängige TeX-Maßeinheit 27 Große rechte Klammer 28 \leftarrow (PlainTeX) 30 \gg 31 \leq



Was Sie schon immer über T_EX wissen wollten. . .

Anlegen einer Liste der Definitionen

Bernd Raichle

Folgende Anfrage wurde vor längerer Zeit in der Newsgroup `de.comp.tex` – jetzt `de.comp.text.tex` – gestellt:

Ich würde gerne in L^AT_EX ein eigenes Verzeichnis ähnlich dem Inhaltsverzeichnis anlegen. In diesem Fall wollte ich ein Verzeichnis der Definitionen in meinem Text anlegen lassen. Irgendwelche Vorschläge, wie ich das erreichen könnte?

Auch hier gilt, wie so oft, dass man auf viele Fragen Antworten und Hinweise auf Makropakete im Buch „Der L^AT_EX-Begleiter“ [1, Kap. 2.4] nachlesen kann. Dort findet man zu obiger Frage einen Abschnitt, der aber für Anfänger etwas zu kurz sein dürfte, da nur einzelne Definitionen gezeigt werden. Für diejenigen, die lieber etwas Komplettes austesten möchten, habe ich hier ein kleines Beispiel aus meiner unerschöpflichen Fundgrube `/dev/null` ausgegraben. Es legt für eine Umgebung namens `definition`, die mit `\newtheorem` definiert wird, ein eigenes Verzeichnis an. Im Folgenden sollten Sie nur die mit einer Zeilennummer versehenen Makroteile in eine `sty`-Datei ablegen oder mit `\makeatletter... \makeatother` geklammert in die Dokumentenpräambel schreiben. Die hier gezeigten Makros können auch mit dem Paket `theorem` verwendet werden.

Wer gleich eine fertige Lösung haben will, sollte besser das Paket `ntheorem`¹ verwenden, das diese Funktionalität in allgemeiner Form für Theoreme bereitstellt. Hier will ich nur die prinzipielle Vorgehensweise demonstrieren.

¹ CTAN: `macros/latex/contrib/supported/ntheorem`

In \LaTeX werden Einträge in die standardmäßig vorgesehenen Verzeichnisse, wie Inhalts- oder Abbildungsverzeichnis, in Hilfsdateien abgelegt, die durch die Anweisungen `\tableofcontents` oder `\listoffigures` gelesen und formatiert werden. \LaTeX besitzt mit dem internen Makro `\starttoc` schon ein Hilfsmittel, mit dem eine solche Hilfsdatei erzeugt und, falls bereits durch einen vorherigen Formatierlauf vorhanden, eingelesen werden kann. Die Datei-Endung gibt man dabei durch das erste Argument von `\starttoc` an. Für die Liste der Definitionen habe ich die Endung `.lod` gewählt. Das Makro `\starttoc` sorgt außer, daß es die Datei erzeugt und einliest, noch dafür, dass diese Datei geöffnet und alle `lod`-Einträge in diese Datei geschrieben werden. Will man mehrere eigene Verzeichnisse anlegen, sollte man bedenken, dass deren Gesamtzahl begrenzt ist. Normalerweise sind noch ca. 10 bis 12 möglich. Für den Dokumentautor wird der Aufruf von `\starttoc` in der Anweisung `\listofdefs` „versteckt“, die wie `\listoffigures` verwendet wird.

```

1 \newcommand{\listofdefs}{%
2   \section*{\listdefname
3     \@mkboth{\uppercase{\listdefname}}%
4       {\uppercase{\listdefname}}}%
5   \@starttoc{lod}}
6 \newcommand{\listdefname}{Definitionen}
```

Jeder Eintrag für eine Definitionen wird nicht in diese neu erzeugte Hilfsdatei, sondern zuerst in die `aux`-Datei geschrieben. Am Ende des Dokuments wird die `aux`-Datei gelesen und die einzelnen Einträge in die verschiedenen Verzeichnisse geschrieben, unter anderem auch die Einträge für die durch `\listofdefs` initiierte `lod`-Datei. In dieser Datei steht jeder Eintrag in der Form

```

\contentsline{definition}%
           {\numberline{Nummer}\ignorespaces Begriff}%
           {Seitennummer}
```

Beim Lesen der `lod`-Datei expandiert das Makro `\contentsline` einen solchen Eintrag zum Makro `\l@definition`, dessen Name aus dem ersten Argument des `\contentsline`-Eintrags gebildet wird. `\l@definition` ist bisher noch nicht definiert, so dass die folgende Definition notwendig ist.

```

7 \newcommand{\l@definition}{\@dottedtocline{1}{1.5em}{2.3em}}
```

Nachdem man somit die `lod`-Datei erzeugen und deren Inhalt formatieren kann, sollte sie nun mit Einträgen gefüllt werden. Normalerweise definiert man mit

```
\newtheorem{definition}{Definition}[section]
```

eine Umgebung namens `definition`, die hier innerhalb einer `section` durchnummeriert wird. Die Umgebung `definition` kann man danach mit

```
\begin{definition}[Name der Definition]
  Definitionstext
\end{definition}
```

in seinem Dokument verwenden. Da wir jedoch so den Namen der Definition für die gewünschte Liste der Definitionen nicht erhalten, muss die Umgebung `definition` indirekt definiert werden.

Zuerst wird eine Hilfs Umgebung namens `definitionIntern` definiert, die nicht vom Autor verwendet werden sollte.

```
8 \newtheorem{definitionIntern}{Definition}[section]
```

Anschließend definiert man die Umgebung `definition`, die die Umgebung `definitionIntern` verwendet. Zuerst wird abgeprüft, ob der Autor ein optionales Argument mit dem Namen der Definition angegeben hat. Wenn ein optionales Argument vorhanden ist, wird dieses mit Hilfe des Makros `\definitionMitOptArg` in die `lod`-Datei geschrieben.

```
9 \newenvironment{definition}{%
10   \@ifnextchar [%   optionales Argument?
11     {\definitionMitOptArg}%
12     {\begin{definitionIntern}}}%
13 }{%
14   \end{definitionIntern}%
15 }
```

Übrig bleibt somit nur noch das Makro `\definitionMitOptArg`, das das optionale Argument liest und mit dem \LaTeX -Makro `\addcontentsline` in die Hilfsdatei schreibt, deren Namensendung durch das erste Argument angegeben wird. Das zweite Argument gibt den Eintragstyp, in unserem Falle eine `definition`, und das dritte Argument den Inhalt des Eintrages an. In diesem sollte die Nummer der Definition mit Hilfe des \LaTeX -Makros `\numberline` gesetzt werden.

```

16 \newcommand{\definitionMitOptArg}{
17 \def\definitionMitOptArg[#1]{%
18   \begin{definitionIntern}%
19     [\addcontentsline{lod}{definition}%
20       {\protect\numberline{\thedefinitionIntern}%
21         \ignorespaces #1}%
22   #1]}

```

Die Anweisung `\addcontentsline` habe ich dabei an den Anfang des optionalen Arguments der Hilfs Umgebung `definitionIntern` eingefügt. Dadurch ist gewährleistet, dass der Eintrag mit der korrekten Seitennummer versehen wird. Wenn die Anweisung `\addcontentsline` durch entsprechende Makros einfach nach dem optionalen Argument wie in

```

\begin{definition}[Name der Definition]%
  \addcontentsline{lod}{definition}{ ... }%   Definitionstext
\end{definition}

```

gesetzt würde, könnte es passieren, dass zwischen dem Definitionskopf und dem nachfolgenden Definitionstext ein Seitenumbruch stattfindet. \LaTeX verhindert diesen Umbruch und genau dies nutzen wir im Makro `\definitionMitOptArg` aus.

Die gezeigten Makros könnte man noch verallgemeinert in einer Deklaration „verstecken“, die beispielsweise den Namen `\newlistedtheorem` hätte und analog wie `\newtheorem` verwendet werden könnte. Jedoch sprengt dies den Rahmen dieser Kolumne.

Literatur

- [1] Michel Goossens, Frank Mittelbach und Alexander Samarin: *Der \LaTeX -Begleiter*; Addison-Wesley Verlag, Bonn, 1994.

T_EX-Beiprogramm

T_EX im Jahr 2003: Vorschläge und Thesen zur Zukunft von T_EX

NTG T_EX future working group¹

Vorbemerkung des Übersetzers

Der nachstehende Artikel erschien bereits vor gut zwei Jahren im TUGBoat, der Zeitschrift der US-amerikanischen und weltweiten T_EX-Benutzergruppe TUG. Er wird hier in Übersetzung nachgedruckt, da er in vieler Hinsicht noch immer aktuell ist und sich viele Leser Gedanken um die Zukunft von T_EX machen.

Der Artikel deckt ein weites Spektrum ab; allgemeinverständliche Anliegen sind ebenso enthalten wie technische, die nur für die Entwickler von Makropaketen sinnvoll sind, konsensfähige Argumente genauso wie solche, die sehr kontrovers diskutiert werden können. Einige Ideen aus diesem Artikel sind in die Entwicklung des Makropaketes ConT_EXt und die von ϵ -T_EX eingeflossen. Hans Hagen, einer der Autoren des Artikels, ist inzwischen im NTS-Team tätig, so dass wir hoffentlich auch bei NTS einige Anregungen wiederfinden werden.

Einleitung

1996 wurde innerhalb der niederländischen T_EX-Benutzergruppe (NTG) intensiv über die Zukunft von T_EX diskutiert. Diese Diskussion wurde durch

¹ Dieser Artikel erschien in *TUGboat* Band 19 (1998), Heft 3, Seite 323ff und wurde von Peter Willadt übersetzt.

einige Veröffentlichungen, die Hans Hagen und Taco Hoekwater in der Diskussions-Liste TEX-NL vorgenommen hatten, ausgelöst und breitete sich in kurzer Zeit in einem weiten Kreis von Teilnehmern aus.

Schließlich kam es im Dezember 1997 zu einem Treffen zwischen einigen der engagiertesten Teilnehmern. Dieser Artikel fasst die auf längere Zeit angelegten Ziele und Forderungen dieser Gruppe zusammen; die kurzzeitig umsetzbaren Forderungen wurden an das ε -T_EX-Team weitergereicht.

Unsere Ansicht vom Stand der Dinge

Zur Zeit haben T_EX-Benutzer Zugang zu mindestens drei Projekten, die an einer Erweiterung von T_EX arbeiten: Omega, pdfT_EX and ε -T_EX.

Die ersten beiden sind in gewisser Weise Nischenprodukte: Wenn Sie weder in nicht-lateinischen Sprachen setzen müssen noch eine pdf-Ausgabe benötigen, bringen Ihnen diese Programme keinen nennenswerten Nutzen. Das dritte Projekt, ε -T_EX, verfolgt einen allgemeineren Ansatz und könnte daher für jeden T_EX-Anwender von Interesse sein.

ε -T_EX ist gut durchdacht und das Programm selbst stabil und nahezu fehlerfrei, doch wird es kaum im Alltag angewendet. Dafür gibt es einfache Gründe: Autoren von Makropaketen werden ε -T_EX-Kommandos nicht benutzen, wenn sie nicht sicher sein können, dass ε -T_EX tatsächlich generell verfügbar ist. Andererseits kann sich ε -T_EX nicht weiterentwickeln, wenn es keine Autoren von Makros gibt, die ε -T_EX einsetzen möchten. Hierdurch ergibt sich eine Henne-Ei-Situation, die zur folgenden These führt:

1. ε -T_EX ist eine gute Idee, aber es bietet zu wenig Unterschiede, um die Leute zum Umsteigen zu bewegen.

Ein weiteres gewichtiges Problem entsteht, wenn Benutzer sowohl die Funktionalität von ε -T_EX *oder* pdfT_EX *oder* Omega *und* zusätzlich die einer der beiden anderen Erweiterungen benötigen: In *einem* Dokument ist das nicht möglich. Alle drei verfügen über eigene, spezifische Sprach-Erweiterungen, die sich in anderen Programmen nur schwerlich nachbilden lassen. Für dieses Problem gibt es zur Zeit keine Lösung, was uns zu folgender These führt:

2. Die Entwicklungen von Omega, pdfT_EX und ε -T_EX sollten baldmöglichst zusammenfließen.

Und dann gibt es noch ein viertes Projekt, das momentan in der Entwicklung ist: Das New Typesetting System (NTS).

Die NTS-Gruppe will NTS für einige Jahre garantiert vollständig kompatibel zu T_EX halten, um die Akzeptanz für NTS zu erhöhen. Wir halten dies für falsch, denn einige der grundlegenden Dinge, die NTS bewältigen sollte, um dem „New“ in seinem Namen gerecht zu werden, können nur erreicht werden, wenn die Kompatibilität zu T_EX geopfert wird. Mechanismen wie registerhaltigen Satz und bessere Kontrolle über Gleitobjekte werden vermutlich vollkommen neue Algorithmen erfordern, woraus sich eine völlig andere Implementierung ergibt. Selbstverständlich kann beides parallel entwickelt werden; der Versuch, etwas Neues zu entwickeln und gleichzeitig absolute Kompatibilität zu wahren, erschwert die Entwicklung allerdings unnötigerweise. Letztlich sollte, wer T_EX benutzen möchte, eben T_EX (und nicht NTS) benutzen.

Daher bitten wir die NTS-Gruppe, ihre Entscheidung, NTS mindestens fünf Jahre lang vollkommen kompatibel zu T_EX zu halten, zu überdenken.

3. NTS bringt nichts, wenn es kompatibel zu T_EX82 bleibt.

Unsere zweite Anmerkung zu NTS hängt mit der beabsichtigten Modularität des Systems zusammen, die durch die Benutzung von Java betont wird.

4. NTS ist zugleich ein Schritt nach vorn *und* ein Rückschritt.

Einfache Erweiterbarkeit wird einer der großen Vorteile von NTS sein. Das Verfahren ist in vieler Hinsicht ähnlich zu dem der L^AT_EX-Pakete, allerdings weit tiefgreifender. Da NTS in Java geschrieben ist, kann man es leicht mit selbst geschriebenen Klassen erweitern. Wir vermuten, dass es eine einfach verwendbare Schnittstelle für Erweiterungen zu NTS geben wird; falls nicht, wird jemand die Quellen so hacken, dass es diese Schnittstelle gibt.

Aller Voraussicht nach wird genau das zu den Problemen führen, die L^AT_EX zur Zeit hat:

- Benutzer überblicken die verfügbaren Pakete nicht, daher stellen sie laufend Fragen wie *Wie kann ich ... mit L^AT_EX machen?*
- Des weiteren sehen wir die Portabilität der Quelltexte (der `.tex`- oder `.nts`-Eingabedateien) ernstlich gefährdet. Wir erwarten jetzt schon Meldungen der folgenden Art:

```
Error: this .nts style-file requires module x.y
which has not been installed on your system.
```

Das NTS-Team sollte strenge Regeln für diese Erweiterungen aufstellen, sonst stellt sich eine Situation wie im Bereich der `\special`-Befehle ein. Eine zentrale Verwaltung und eine verantwortliche Instanz für die Erweiterungsmodul werden erforderlich sein, um solche Probleme zu verhindern. Es erscheint sinnvoll, daraus eine Vollzeitstelle, beispielsweise unter Kontrolle der T_EX Users Group, zu machen.

5. Wir brauchen Zeit zum Experimentieren und dürfen nicht in die Jedes-Jahr-eine-neue-Version-Falle laufen.

Interessanterweise weisen alle gegenwärtigen Entwicklungen darauf hin, dass nur Erfahrung zur Entwicklung von sinnvollen Funktionen führt. Es ist anzunehmen, dass sich dies auch für NTS abzeichnen wird.

Hieraus ergibt sich für die nächste Generation von T_EX-Programmen, dass die Benutzer sich gedulden sollten, bis die Entwickler von Erweiterungen und Makropaketen den neuen Funktionen trauen und eine Aufwärts-Kompatibilität gewährleisten können. Ebenso folgt daraus, dass es einige Jahre dauern wird, bis ϵ -T_EX und NTS als Nachfolger akzeptiert sein werden.

Wir sollten nicht vergessen, dass Knuth sein erstes T_EX komplett neu schrieb!

Zusammenstellung von T_EX-Distributionen

Während der letzten fünf Jahre wurde die Installation von T_EX deutlich einfacher. Die wichtigsten Gründe sehen wir in Folgendem:

- CD-ROMs sind inzwischen weit verbreitet. Sie können problemlos ein vollständiges T_EX-System enthalten. Die altmodischen Diskettenstapel waren viel zu verwirrend, und Bänder sind nur etwas für Profis.
- Festplatten sind so preisgünstig geworden, dass es nicht mehr ungewöhnlich ist, ein vollständiges T_EX-System auf der Platte installiert zu haben.
- Es gibt inzwischen Installationsprogramme, die den Benutzern die unangenehmen Konfigurations- und Installationstätigkeiten weitgehend abnehmen.

Dennoch bleiben etliche Probleme, die sich letztlich nicht vermeiden lassen, weil sie mit der Arbeitsweise von T_EX zusammenhängen.

- Ein typisches T_EX-System besteht aus einer beeindruckenden Zahl von Dateien (mehr als 31 415). Niemand kann abschätzen, welche Teile notwendig und welche entbehrlich sind. Mit anderen Worten: jedes System ist zu umfangreich.
- Auf CTAN gibt es „alles“, aber immer nur die jeweils neueste Version. Jedoch werden gelegentlich ältere Versionen benötigt, um ältere Dokumente zu setzen. Alte CTAN-Abzüge auf CD-ROM helfen bei der Suche nach älteren Versionen, sind aber kein Ersatz für eine professionelle Versionskontrolle.
- Wartung ist nur für Profis machbar. Alle anderen sind besser beraten, wenn sie ein System komplett neu aufspielen, obwohl auch dies zweifelsohne Probleme nach sich zieht. Der Haken an „Plug&Play“-Systemen ist eben, dass die Benutzer keine Vorstellung mehr von den inneren Abläufen ihres Systemes haben. Ist dies eine gute oder schlechte Sache?
- So etwas wie ein einfaches Aufrüsten gibt es nicht. Es ist sehr schwierig bis unmöglich, einfach einige zusätzliche Dateien aufzunehmen und sie nahtlos in ein bestehendes System zu integrieren.
- Die Erstinstallation lässt sich automatisieren, aber eine nachträgliche Änderung der Konfiguration ist sehr schwierig. Ein typisches T_EX-System enthält Dutzende Konfigurationsdateien in ebenso vielen Stilrichtungen. In der Regel sind sie über die ganze Festplatte verteilt und nur absolute Experten kommen damit klar.

Daraus ergeben sich eine Reihe von Folgerungen:

6. Die Zahl der Dateien in einer typischen T_EX-Installation sollte um den Faktor 100 reduziert werden.

Wir können dieses Ziel erreichen, indem wir die Art, wie ein Programm seine Ressourcen findet, ändern. Eine zentrale Datenbank sollte alle Fragen nach Ressourcen beantworten. Diese Datenbank sollte alle Ressourcen physikalisch enthalten. Selbstverständlich sollte sie auch in der Lage sein, aufzulisten, was verfügbar ist, und das in jeder gewünschten Ausführlichkeit. Die Datenbank sollte eventuell eine Verbindung zu CTAN (das dann auch eine Datenbank wäre) herstellen und lokal nicht erhältliche Ressourcen dort abrufen.

Dadurch wäre es möglich, minimale lokale Installationen einzurichten, die per Internet nach Bedarf anwachsen können.

7. Die Konfiguration eines T_EX-Systems sollte zentralisiert und automatisiert erfolgen.

Wenn wir den vorigen Punkt umsetzen können, bietet auch dieser kaum Schwierigkeiten. Programme sollten in formeller Weise angeben, welche Konfigurations-Umgebung sie benötigen. Diese könnte dann durch Mentis oder automatisch durch Überprüfung der Installation – das heißt, Befragung der Datenbank – erzeugt werden.

8. Installation und Wartung sollten weit weniger Fachwissen erfordern.

Die Datenbank könnte CTAN gelegentlich nach Updates abfragen. Der Administrator erhielte eine kurze Beschreibung der verfügbaren Updates zusammen mit Links zur verfügbaren Dokumentation. Er könnte dann entscheiden, welche Updates installiert werden. Dies könnte auch vollautomatisch über Nacht ablaufen, falls ein immer topaktuelles System gewünscht wird. Programme sollten automatisch aufgefordert werden, sich selbsttätig neu zu konfigurieren, wenn erforderlich.

Dieses Setup sollte sich auch der endlosen Probleme um nicht portable `dvi`-Dateien annehmen. Wir sollten alle nach Möglichkeit die selben Ressourcen verwenden und, falls wir das nicht tun, hinsichtlich möglicher Probleme gewarnt werden. Falls wir uns dazu entschließen sollten, `dvi`-Dateien zu erzeugen, die gar keine virtuellen Schriften enthalten (was bedeutet, dass T_EX selbst statt des `dvi`-Treibers die `vf`-Dateien lesen sollte), wäre eine gewichtige Fehlerquelle beseitigt.

9. CTAN sollte einen vollständigen Index mit kompletten Beschreibungen und Querverweisen zwischen allem, was irgendwie miteinander zu tun hat, besitzen.

Wenn wir die Systeme wie oben beschrieben miteinander verzahnen wollen, ist dies zwingend. Veröffentlichungen auf CTAN müssen sorgfältiger geprüft werden: Beschreibungen, Spezifikationen, Versionsnummern, Beziehungen zu anderen Paketen, Abhängigkeit von anderen Ressourcen usw. müssen angegeben werden. Was sich nicht an diese Konventionen hält, sollte ausgelagert („nicht unterstützt“) und nach einer gewissen Zeit gelöscht werden.

Wir sehen, dass wir damit einen kulturellen Schock in der T_EX-Welt auslösen, doch wir halten dies für unabdingbar, um T_EX auch im nächsten Jahrtausend am Leben zu halten.

10. Anarchie hat T_EX groß gemacht; Anarchie wird T_EX erledigen.

Das müssen wir verhindern!

Wunschliste für Online-Dokumente

Mit dem zunehmenden Wachstum des Internet entstand eine ganz neue Art von Dokumenten: Dokumente, die vorrangig oder ausschließlich zum Betrachten am Bildschirm gedacht sind. Die verwendeten Formate unterscheiden sich, aber es ist leicht zu sehen, dass einige Punkte von gemeinsamem Interesse sind: Kurze Download-Zeiten, Unterstützung für Hyperlinks und einfache Benutzung sind von größter Wichtigkeit für diese Formate.

11. T_EX ist bereits in seiner jetzigen Form gut geeignet, diesen Bedürfnissen nachzukommen, aber einige Erweiterungen sind notwendig, um sicherzustellen, dass T_EX eine Führungsrolle in diesem Bereich erreicht/behält.

Fünfzehn Jahre lang konnte T_EX nur `dvi`-Ausgabe erzeugen. Die Beschränkungen sowohl von T_EX als auch des `dvi`-Formates betreffen in erster Linie Grafikerunterstützung und farbigen Satz, doch Farbdrucker waren selten und die fehlende Grafikerunterstützung konnte umgangen werden.

Wiewohl ursprünglich angenommen wurde, T_EX sei in der Lage, alles selbst zu behandeln, haben uns diese 15 Jahre gezeigt, dass dem nicht so ist: Viele Anwendungen, wie das Einfügen von Grafiken oder die Verwendung von Farbe, beruhen ganz wesentlich auf Nachbearbeitung auf der Stufe der `dvi`-Treiber. Für Online-Veröffentlichungen ist eine solche Vorgehensweise weder wünschenswert noch machbar. Online-Formate sind zwingend geräte-unabhängig – andernfalls müssten verschiedene Versionen des selben Dokumentes veröffentlicht werden.

An sich sind sowohl pdfT_EX als auch die Alternative, per `dvips` und Adobe Acrobat¹ in der Lage, `pdf` zu erzeugen, wobei wir im letzteren Fall noch Postprozessoren benötigen, aber wir können davon ausgehen, dass die verschiedenen Verfahren sich auseinander entwickeln, und zudem empfinden wir die Verwendung von externen Programmen, um interne Probleme zu lösen, als eine unnötige Komplizierung des Systems.

¹ oder seit geraumer Zeit auch `dvipdfm` (A.d.Ü.)

12. Online-Veröffentlichungen brauchen eingebaute Unterstützung.

Selbstverständlich können die meisten Erweiterungen wie zum Beispiel Hyperlinks auch mit `dvi`-Ausgabe und `\special` erreicht werden. Die Benutzung könnte jedoch wesentlich robuster sein. Hier mag pdfT_EX als Beispiel dienen, das bereits eingebaute Unterstützung für Hyperlinks enthält, wodurch die Notwendigkeit entfällt, weitere Programme laufen zu lassen. Entsprechendes gilt für die Wiederverwendung von Objekten, für Formulare, Skripten (JavaScript) und die Einbindung von Grafiken.

Doch Systeme wie pdfT_EX schaffen auch neue Probleme. Betrachten wir zum Beispiel die Grafik-Einbindung: Reines T_EX muss sich nur um die Abmessungen der Box kümmern; Generatoren für Online-Dokumente müssen die Grafikdatei selbst einbinden.

Hyperlinks stellen eine weitere konzeptuelle Erweiterung dar. Wiewohl man mit raffinierten Makros brauchbare Ergebnisse erzielen kann, reiben sich doch alle auf dem jetzigen T_EX basierenden Lösungen an den Wünschen der Autoren oder an dem Zeilen- und Absatzumbruch-Mechanismus von T_EX.

13. T_EX-Objekte sollten einfach wiederverwendbar sein.

Das Konzept der Wiederverwendung von Objekten findet sich leider nicht in `dvi` – mit `\special`-Anweisungen – wieder. Vermutlich hat es dort keinen Einzug gehalten, weil es vor allem für Online-Dokumente von Bedeutung ist; für Ausgabe auf Papier spielt es kaum eine Rolle.

Aus Sicht des Anwenders mag die Wiederverwendung von Objekten sehr einfach aussehen (etwa wie die Wiederverwendung von Boxes mittels `\copy`); aus der Sicht dessen, der sie implementiert, sind Definitionen von Objekten nur eine weitere Art von miteinander möglicherweise in Konflikt tretendem *whatsit*. Wieso in Konflikt treten? Weil T_EX kein Verfahren kennt, das alle Überschneidungen und Konflikte vermeidet. Sicherlich können wir eine Box verwenden und ebenso auch Dinge an verschiedenen Stellen in einem Dokument tun, ohne dass Schaden angerichtet wird, doch das Problem ist keinesfalls optimal gelöst.

Wenn Wiederverwendung zum Beispiel zur wiederholten Einbindung von Abbildungen verwendet wird, ist sie unproblematisch zu handhaben, sei es mit dem ursprünglichem T_EX (reines `dvi`, mittels Gilberts `dviview`), mit dem traditionellen `dvi-dvips`-Acrobat-Pfad oder auch mit Thànhs pdfT_EX. Doch pdf-Formulare benötigen mehr Unterstützung.

Hier ist ein sinnvolles Beispiel, das Objekte benötigt: in pdf kann ein Kästchen zum Ankreuzen mit unterschiedlichem Aussehen definiert werden (an, aus, mit Maus darüber...). Technisch sollte das (in pdfT_EX) etwa wie folgt umgesetzt werden:

```
\setbox0=\hbox{\star $} \pdfform0
  \edef\on {\the\pdflastform}
\setbox0=\hbox{\bullet$} \pdfform0
  \edef\off {\the\pdflastform}
\setbox0=\hbox{\times $} \pdfform0
  \edef\down{\the\pdflastform}
```

Wenn wir das Ankreuzfeld definieren, können wir dann auf `\on`, `\off` und `\down` zurückgreifen, wie im folgenden Code:

```
\pdfannot{ ... /On \on\space0 R ...}
```

Zur Zeit schreibt pdfT_EX Formularelemente nur in die Ausgabedatei, wenn auf diese Objekte auch tatsächlich zugegriffen wird. Dieses Verhalten ist durchaus sinnvoll, da wir auch in der Lage sein möchten, Dinge zu versuchen, ohne mit redundanten Objekten gestraft zu werden – stellen wir uns etwa ein Makro vor, das drei verschiedene Dinge ausprobiert und das beste Ergebnis verwendet.

Zurück zu den drei Objekten: Sie werden nicht in der Ausgabedatei landen, wenn wir uns in der Feld-Definition wie oben beschrieben auf sie beziehen, da Felddefinitionen wie `\special`-Anweisungen behandelt werden – pdfT_EX reicht die Informationen einfach weiter.

Somit erhalten wir eine ungültige Referenz: Das Objekt wird benötigt, landet aber nicht in der Datei. Was lernen wir daraus?

14. T_EX benötigt ein vollwertiges Objekt-Modell.

Eines mit unmittelbar wie auch verzögert wirksamen Definitionen, die unabhängig von den internen Listen sind, die T_EX erstellt und das sowohl Vorwärts- als auch Rückwärtsverweise ermöglicht.

Eine andere typische Eigenschaft von Online-Dokumenten ist die weit umfangreichere Verwendung von Farben und Schmuckelementen als bei Papier-Dokumenten im traditionellen Schriftsatz. Über deren Nutzen lässt sich trefflich streiten, doch gibt es durchaus etliche sinnvolle Einsatzmöglichkeiten im Alltag für solche Elemente.

Bei der T_EX Users Group-Tagung 1997 gab es verschiedene Vorführungen zum Thema Grafik. Die damit zusammenhängenden Diskussionen brachten unter anderem eine BOF-Sitzung über Grafik-Primitive hervor. Direkte Einbindung von METAPOST-Ausgabe (in pdfT_EX) hatte bereits bewiesen, dass eine relativ kleine Untermenge der PostScript-Sprache für ausgefeilte Grafiken verwendet werden kann; daher beschränkte sich die Diskussion auf diese Befehle.

Diese Grafikbefehle in T_EX sind nicht dazu gedacht, per Hand gezeichnete Grafiken zu erstellen, für die man ansonsten Programme wie Corel Draw, Adobe Illustrator oder Macromedia Freehand verwendet. Sie sollen vielmehr verwendet werden, um Statistiken zu visualisieren, Funktionen zu zeichnen und mehr oder minder mathematische Formen darzustellen, mit denen sich bestimmte Layouts verdeutlichen lassen. In diesen Grafiken spielt Text eine große Rolle, daher sollte dieser Text vorzugsweise von T_EX gesetzt werden. Hieraus ergibt sich, dass die Einbindung einer externen Datei sich nicht eignet, und unsere Schlussfolgerung lautet:

15. T_EX braucht ein zuverlässiges System für Inline-Grafik und Farbe.

Als wichtigstes Ergebnis der BOF-Sitzung auf der T_EX Users Group-Tagung 1997 wurde vereinbart, wie vorgegangen werden soll: Ein Satz von erweiterten Funktionen soll definiert werden, der die direkte Einbindung von METAPOST-Grafiken ermöglicht. Man war sich darüber einig, dass sich mittels dieser Befehle auch die Bedürfnisse der meisten in T_EX geschriebenen Grafik-Makropakete erfüllen lassen.

Während des „Future of T_EX“-Treffens der NTG war eines der wichtigsten Ziele der Teilnehmer, die exakte Spezifikation dieser Grafik-Befehle (die momentan als `\special`-Befehle implementiert sind) festzulegen. Daher mussten wir die zu verwendende Syntax formal definieren, und das brachte uns mitten in die `\special`-Problematik.

Unsere endgültigen Ergebnisse zu diesem Thema stellen wir an anderer Stelle vor², doch Gilbert hat inzwischen einiges Grundlegende umgesetzt. Unten finden Sie seine Erläuterungen zu den `\specials`, die bereits in `dviview` umgesetzt sind. Dieser Text ist hier abgedruckt, weil er zeigt, dass bereits wenige einfache Kommandos ausreichen, um Inline-Grafiken fast vollständig zu unterstützen.

² *TUGboat*, Jahrgang 18 (1997), Heft 3, Seite 330 ff.

Um beispielsweise METAPOST-Grafiken in T_EX einzubinden, benötigen Sie folgende Dinge:

- Ein Makro, um METAPOSTs PostScript-Ausgabe zu interpretieren. Hans Hagen schrieb einen Satz von Makros für pdfT_EX, der `\pdfliteral`-Kommandos erzeugt. Diese Makros lassen sich leicht so anpassen, dass sie einen anderen Standard, der die `\special`-Syntax benutzt, unterstützen.
- Es wird eine einfache Untermenge der PostScript-Kommandos benötigt. METAPOST verwendet nur wenige PostScript-Kommandos, um seine Figuren zu zeichnen.

Um den vorgeschlagenen Standard für Inline-Grafiken zu testen, benötigten wir ein Anzeigeprogramm, in das diese Unterstützung einfach zu integrieren war. `dviview` entstand gerade zu dieser Zeit, daher bot es sich an, dieses Programm als Umgebung für Tests und zur Entwicklung einzusetzen.

Fast alle Primitiven – mit Ausnahme einiger weniger wie Clipping – sind einfach umzusetzen. Die Syntax wird vielleicht nochmals geändert werden, wenn die neue `\special`-Syntax standardisiert ist. Es ist sehr einfach, diese `\specials` in PostScript umzuwandeln (indem beispielsweise `dvips` umgeschrieben wird), da die Kommandos kaum eine Übersetzung benötigen.

`\specials` und Befehle für Inline-Grafiken in `dviview`:

```

\special{dv:startgraphic}
\special{dv:stopgraphic}
\special{dv:moveto x y}
\special{dv:lineto x y}
\special{dv:curveto x1 y1 x2 y2 x3 y3}
\special{dv:stroke}
\special{dv:setlinejoin j}
\special{dv:setlinecap c}
\special{dv:setdash offset values}
\special{dv:setlinewidth w}
\special{dv:setmiterlimit m}
\special{dv:rotate r}
\special{dv:translate x y}
\special{dv:concat x1 y1 x2 y2 x3 y3}
\special{dv:newpath}
\special{dv:closepath}

```

```
\special{dv:clip}  
\special{dv:fill}  
\special{dv:gsave}  
\special{dv:grestore}
```

Es ist leicht zu sehen, dass die Zahl der Kommandos zur Unterstützung der METAPOST-Ausgabe recht klein ist.

Einige Erläuterungen:

dv:startgraphic

beginnt eine Grafik. Es speichert die momentane Position und den Kontext des dvi-Interpreters. Die momentane Position wird als Ursprung (0,0) verwendet. Wie in PostScript weist positives x, y nach rechts und oben.

dv:stopgraphic

beendet eine Grafik und stellt den Kontext wieder her.

dv:moveto x y

verlegt die aktuelle Position nach (x, y) .

dv:lineto x y

zeichnet eine Linie nach (x, y) . Die Linie wird nicht sofort gezeichnet, lediglich die Koordinaten werden vermerkt. Das Zeichnen selbst wird mittels **stroke** vorgenommen.

dv:curveto x1 y1 x2 y2 x3 y3

zeichnet eine Bezier-Kurve vom aktuellen Punkt nach (x_3, y_3) . Die Kontrollpunkte sind durch (x_1, y_1) und (x_2, y_2) gegeben.

dv:stroke

führt die Zeichnung mit dem aktuellen Stift, der aktuellen Farbe und Breite aus.

dv:setlinejoin j

gibt an, wie Linien verbunden werden. j darf 0, 1 oder 2 sein.

dv:setlinecap c

Gibt an, wie Linien-Enden aussehen. j darf 0, 1 oder 2 sein.

dv:setdash offset vals

setzt den Stil des Stiftes. **vals** gibt an, wie lange der Stift „an“ und „aus“

ist. `offset` kann verwendet werden, um den Anfang des `vals`-Musters zu verschieben.

`dv:setlinewidth w`

legt die Dicke des Stiftes fest.

`dv:setmiterlimit m`

setzt die Größe der Dehnungsfuge *miterlimit*.

`dv:rotate r`

ändert die aktuelle Transformationsmatrix, so dass alles folgende um `r` Grad gedreht dargestellt wird.

`dv:translate x y`

ändert die aktuelle Transformationsmatrix, so dass alles folgende um (x, y) verschoben dargestellt wird.

`dv:concat x1 y1 x2 y2 x3 y3`

multipliziert die aktuelle Transformationsmatrix mit den angegebenen Werten.

`dv:newpath`

verwirft alle momentan aktuellen Pfade und beginnt einen neuen Pfad.

`dv:closepath`

schließt den aktuellen Pfad. Anschließend kann `fill` angewendet werden, um den geschlossenen Pfad auszufüllen.

`dv:clip`

verwendet den aktuellen Pfad als Clipping-Pfad. Alle im folgenden ausgeführten Füllungen und Zeichnungen werden durch diesen Pfad begrenzt. Der Clipping-Pfad kann aus einem oder mehreren geschlossenen Pfaden bestehen.

`dv:fill`

füllt den aktuellen Pfad mit der aktuellen Farbe.

`dv:gsave`

sichert den Grafikstatus.

`dv:grestore`

stellt den gesicherten Grafik-Status wieder her.

`dv:setrgbcolor r g b`

setzt die aktuelle Farbe. `r`, `g` und `b` liegen zwischen 0 und 1.

`dv:setcmykcolor c m y k`

setzt die aktuelle Farbe. `c`, `m`, `y` und `k` liegen zwischen 0 und 1.

`dv:setgray g`

setzt die Graustufe. Werte zwischen 0 (schwarz) und 1 (weiß) sind zulässig.

Es wäre ein leichtes, diesen Befehlssatz zu erweitern und wesentlich mehr PostScript-Befehle aufzunehmen, doch dies ist nicht beabsichtigt. Wir möchten betonen, dass komplexe Grafiken, die den ganzen Satz von PostScript-Anweisungen benötigen, integriert werden sollten, indem `eps`-Dateien eingebunden werden und ein PostScript-Interpreter die Arbeit übernimmt.

Wunschliste zur Spracherweiterung

Entfernung der Beschränkungen in Bezug auf Schriften

Die dem `tfm`-Format eigenen Beschränkungen für Schriften sollten beseitigt werden. Ein recht einfacher Weg hierzu wäre, T_EX direkt `.pl`- und `.vpl`-Dateien anstatt `tfm`-Dateien lesen zu lassen; es ist aber ebenso möglich, ein neues Format wie Omegas `OFM`-Dateien zu übernehmen oder womöglich eine ganz neue Spezifikation zu schaffen.

Eine Übersicht der Einschränkungen im gegenwärtigen T_EX zeigt Grenzen an fast allen Stellen: die Anzahl der Zeichen in einer `tfm`-Datei, die Anzahl unterschiedlicher `width/height/depth/italics-correction`-Werte, Anzahl der Ligaturen und Unterschneidungspaare, Parameter für den Mathematiksatz und so weiter. Kaum eine dieser Beschränkungen wird noch benötigt; die meisten entstanden aus Knuths Wunsch, mit möglichst wenig Speicher auszukommen.

Insbesondere bei der gegenwärtigen Implementierung des Mathematik-Modus werden einige sehr ungewöhnliche Anforderungen an die verwendeten Schriften gestellt (einige Zeichen landen an sehr ungewöhnlichen Stellen innerhalb des Koordinatensystems, beispielsweise liegen Integrale und Delimiter vollständig unterhalb der Grundlinie, und die Höhe des `\sqrt`-Zeichens wird verwendet, um über die Breite des horizontalen Armes zu entscheiden). Dies sollte beseitigt werden, damit es einfacher wird, nicht mit METAFONT erstellte Schriften im Mathematikmodus zu verwenden und um die Schaffung neuer Mathematik-Schriften zu unterstützen. Die momentane Lage macht es unmöglich, Nicht-T_EX-Schriften für Mathematik, wie beispielsweise aus *Mathematica*, ohne *eine Menge* Tricks mit `vf`-Dateien zu verwenden.

All das lässt sich im Quelltext von T_EX recht einfach beheben, allerdings ist das zum gegenwärtigen Zeitpunkt nicht sinnvoll, weil wir nach wie vor auf das `tfm`-Format angewiesen sind.

16. Die Art, wie `tfm`- und `vf`-Formate definiert und implementiert sind, ist die wesentliche Ursache des derzeitigen Schrift-Chaos.

Wenn ein neues Format geschaffen werden sollte, wäre die Erweiterbarkeit der PL-Datei-Syntax ein gewichtiger Vorteil; es könnten auch neue Möglichkeiten eingebracht werden, ohne die Abwärts-Kompatibilität einzubüßen. Wiewohl es für die Verwendung binärer Dateien keine Gründe hinsichtlich Plattenplatz oder Verarbeitungsgeschwindigkeit mehr gibt, haben binäre Dateien doch den Vorteil, dass sie nicht einfach editierbar sind (was verhindert, dass sie vom Benutzer versehentlich beschädigt werden).

17. Wir brauchen symbolische Bezeichner für Schriftzeichen.

Anstelle der Namen der Lettern verwendet T_EX zur Zeit die Position im Zeichensatz. Dieses sogenannte *Encoding* ist altmodisch und dient vor allem der Geschwindigkeit. Die Verbindung zwischen Name und Letter selbst sollte jedoch eine T_EX-interne Tätigkeit sein.

Die per Namen angegebenen benutzten Lettern aus den Schriften sollten aus der ausgegebenen `dvi`-Datei hergeleitet werden können, um Programmen zur Nachbearbeitung eine nochmalige Umkodierung zu ersparen. Um dieses Ziel zu erreichen, wird T_EX vermutlich ein internes Benennungsschema für Lettern benötigen, das von der Kodierung der jeweiligen Schrift unabhängig ist. Das ε -T_EX-Team arbeitet bereits in dieser Richtung. Es erscheint uns nicht sehr wahrscheinlich, dass Unicode das Problem löst, doch könnte eine Lösung, die auf dem vordefinierten Satz von Unicode-Namen aufsetzt (diese Richtung wird von Omega eingeschlagen), sich als sinnvoll erweisen.

18. Ligaturen und Unterschneidungsinformationen sollten unabhängig von den Abmessungen der Zeichen sein.

Gegenwärtige Schriftdefinitionen können zwar Ligaturen enthalten, doch wir würden die Ligaturtabelle gerne von T_EX aus, sozusagen intern, ändern können. Diese Anforderung wurde bereits an das ε -T_EX-Team weitergeleitet, aber wir brauchen eine allgemeinere Lösung als die Primitiven `\noligs` und `\nolig char`, die für ε -T_EX vorgeschlagen wurden. Entsprechendes gilt für die Unterschneidungstabellen.

Der Mechanismus, mit dem ein Benutzer Schriften in T_EXs Speicher lädt, ist viel zu unflexibel. Es sollte möglich sein, Kodierungen, Unterscheidungen und Ligaturtabellen unabhängig von den Abmessungen der Lettern zu laden. Das Ligaturproblem beruht übrigens auf zwei sehr unterschiedlichen Schwierigkeiten.

Der einfache Fall liegt vor, wenn *verbatim*-Ausgabe in einer anderen als Schreibmaschinen-Schrift gewünscht wird. So etwas wird oftmals in Lehrbüchern über Programmiersprachen benötigt.

Der schwierige Fall kommt zum Tragen, weil Ligaturen von der Sprache abhängen, nicht von der Schriftart. Die spanischen Ausrufezeichen werden beispielsweise nie außerhalb von Spanien verwendet, und doch sind alle Schriften damit versehen. Im Idealfall sollte jede Sprache ihre Ligaturtabelle haben, die genauso Teil der Sprachspezifikation ist wie beispielsweise die Trennmuster.

19. METAFONT wird überholt, auch wenn T_EX noch zeitgemäß bleibt.

Es wird eine neue Version von METAFONT benötigt, die akzeptable Outline-Schriften anstelle des jetzt erzeugten Bitmap-Formates erzeugt; die Verwendung von Nicht-METAFONT-Schriften (PostScript, TrueType) sollte erleichtert werden. Wie oben erwähnt, sollte T_EX selbst sich um die Virtualität der Schriften kümmern. Das muss *nicht* die Verwendung von *vf*-Dateien nach sich ziehen. Es gibt einige andere Möglichkeiten, die nicht so mächtig sein mögen wie *vf*, aber dafür erheblich weniger verwirrend sind: Die einzigen verbreiteten Anwendungen von virtuellen Schriften sind das Umkodieren von Lettern und das Erstellen von zusammengesetzten Lettern.

Benutzerschnittstelle

T_EX zeigt zur Zeit eine eigenartige Dualität: überwiegend wird es zur Stapelverarbeitung eingesetzt, jedoch gibt es einige Situationen, in denen Benutzereingriffe benötigt werden.

Einerseits gilt: wenn T_EX als Stapelverarbeitungs-Programm überleben sollte (sei es als selbstständiges Satzprogramm oder als Nachbearbeiter beispielsweise für SGML-Verarbeitungssysteme), benötigt es Erweiterungen, damit es 100%ig sicher ohne Benutzereingriffe laufen kann. Dinge wie der Umbruch mathematischer Formeln und die Platzierung von Abbildungen können T_EX zur Zeit nicht immer überlassen werden.

Am anderen Ende der Skala steht eine in Echtzeit lauffähige grafische Benutzerschnittstelle, um interaktive Benutzer zufrieden zu stellen (hier könnte eventuell eine teilweise Umsetzung ausreichen, wie beispielsweise GUI-basierte Editoren für Gleichungen und Tabellen). Dieses Ziel ist nur zu erreichen, wenn den GUI-basierten Programmen ein narrensicheres T_EX-Eingabeformat zur Verfügung steht, auf das sie aufbauen können.

Wir sehen zwei mögliche Wege:

- Indem eine große Anzahl Makros in Programm-Code umgewandelt wird, gibt es wesentlich weniger Unterschiede zwischen Makroformaten; allerdings bleiben Probleme im Zusammenhang mit der Umdefinition von T_EX-Primitiven bestehen.
- Könnte eine bereits in Token umgewandelte Eingabe in einem vorkompilierten Format akzeptiert werden, ließen sich diese Probleme eventuell umgehen. Die dahinterstehende Idee ist, dass ein externes Programm, das T_EX-Code erzeugt, absolut sicher sein sollte, dass `\par` tatsächlich auch `\par` bedeutet.

Es gibt allerdings auch einige Eigenheiten in der T_EX-Sprache selbst, die einer Überarbeitung bedürfen: Schlüsselwörter und -zeichen, die manchmal optional und manchmal zwingend sind, wie beispielsweise das Gleichheitszeichen; Argumente in Klammern im Gegensatz zu durch Leerzeichen begrenzten und so fort.

20. Unter allen Umständen bedarf die T_EX-Sprache einer gründlichen Säuberung.

Die Syntax sollte definitiv gesäubert werden. Jeder, der jemals versucht hat, ein nicht-triviales Makro zu schreiben, weiß, dass auch, wenn der Ansatz an sich richtig ist, das Makro oftmals doch nicht funktioniert, weil ein „dummer“ Fehler mit `\expandafter` oder zu vielen/zu wenigen Leerzeichen gemacht wurde. Lösungen, die Markup in der Art von SGML oder Lisp verwenden, wären der gegenwärtigen Lage vorzuziehen. Die jetzige Syntax rechtfertigt oft die folgende Aussage:

21. T_EXs Makrosprache unterstützt das Schreiben von Müll.

Wir können mit Gewissheit sagen, dass viele Quellen im Hinblick auf ihre Formatierung schrecklich aussehen, wie etwa die Quellen der Stildatei, mit der dieser Artikel gesetzt wurde. (Oder sehen Sie sich den Quelltext des

T_EXbook an: Die Ausgabe ist schön, die Eingabe einfach nur hässlich.) In den Händen des einfachen Benutzers wird jedoch aus schlechter Eingabe immer auch schlechte Ausgabe.

22. Wir könnten aus besseren Programmier-Primitiven Nutzen ziehen.

Letztlich zeigt die Erfahrung, dass Formatdateien keineswegs einfach und klein sind, wie Knuth ursprünglich erwartete. Stattdessen sind Formatdateien umfangreiche Programme mit komplexen Wechselwirkungen zwischen einzelnen Komponenten. T_EXs Makrosprache war niemals dazu gedacht, dies zu unterstützen, und bietet daher nur wenig Unterstützung für Programmierer. Zu den fehlenden Dingen gehören Datenstrukturen wie Listen und Schlangen, begrenzte Namensräume, Kontrollstrukturen (wie *Case* und *While*), Signale und verlässliche `\if`-Tests.

Rezensionen

Robert Bringhurst: The Elements of Typographic Style

Peter Willadt

Es gibt viele Bücher über Typographie: Von striktem Konservatismus geprägte wie die des älteren Tschichold, Platitüden, die nicht nur die Inhalte, sondern auch gleich das Layout von Vorgängern abkupfern, Verfechtungen von typographischen Mode-Erscheinungen. Spätestens nach dem zehnten Buch beschlich mich das Gefühl, dass mich auf diesem Sektor nichts mehr angenehm überraschen könne.

Um so mehr war ich von Robert Bringhursts Buch angetan. Sein Schwerpunkt liegt auf der Mikrotypographie, sein besonderes Interesse darin, Originalen gerecht zu werden und behutsam vorzugehen. Das Buch beginnt mit einem Appell, die Typographie dem jeweiligen Text anzupassen: *Typography exists to honor content*. Dem schließen sich Beispiele für dezenten Satz und Regeln zur Mikrotypographie an. Der Autor regt an, auch die Geschichte der Schriften zu beherzigen: Eine Schrift, die 200 Jahre ohne fette Kursive auskam, sollte vielleicht immer noch ohne fette Kursive auskommen.

Alleine die ersten vier Kapitel, die sich mit dieser Thematik befassen, rechtfertigen die Anschaffung des Buches. Das fünfte Kapitel ist den Trenn- und Bindestrichen, Klammern und weiteren Symbolen gewidmet. Hier wird nicht nur der Unterschied zwischen Trennzeichen, Bindestrich und von-bis-Strich gezeigt (den jeder Leser des „ \TeX Book“ ohnehin kennt), Bringhurst weist beispielsweise auch darauf hin, dass der Schrägstrich in *und/oder* in traditioneller Typographie ein anderer als der in $\frac{3}{4}$ ist. Der Autor führt einige Beispiele an, wo der ursprüngliche Charakter einer Schrift im Rahmen der Digitalisierung dem Zeitgeschmack angepasst wurde, indem beispielsweise der ursprüngliche schräge Trennstrich einem geraden weichen musste oder wo – wie bei der Garamond – die ursprünglich durchgehend Haarlinienstärke aufweisenden Klammern () durch Klammern wechselnder Strichstärke ersetzt

wurden. Er empfiehlt, solche Verfälschungen zu beheben, sei es, indem man sich passende Zeichen aus einer anderen Schrift „ausleiht“ oder indem man gegebenenfalls selbst zum Font-Editor greift.

Kapitel 6 nimmt sich der Schriftmischung an. Bringhurst plädiert hier nicht nur für harmonisch zusammen passende Mischungen, sondern vor allem auch dafür, historische Gegebenheiten zu berücksichtigen und nicht etwa eine Renaissance-Antiqua mit einer Rokoko-Kursiven zu mischen. Auch die Mischung von lateinischen und nicht-lateinischen Schriften findet hier eine ausreichende Würdigung. Das darauf folgende Kapitel gibt einen kurzen Abriss über die Geschichte der römischen Druckschrift. Im achten Kapitel geht es dann um die Einteilung der Seite, das Verhältnis zwischen Satzspiegel und Rändern. Bringhurst stimmt hier in vielerlei Hinsicht mit den Ansichten überein, die Jan Tschichold in den *willkürfreien Maßverhältnissen* dargelegt hat.

Nach einem Kapitel, das sich mit neuen Entwicklungen im Satzbereich befasst – unter anderem werden der HZ-Algorithmus, Multiple-Master-Schriften und GX-Fonts vorgestellt – aber leider deutlich macht, dass der Autor eben Typograph ist und das mir deswegen oft zu vage bleibt, folgt das mit fast 80 Seiten umfangreichste Kapitel, in dem Bringhurst etliche Schriften vorstellt. Er berücksichtigt hierbei nicht nur in digitalisierter Form kommerziell erhältliche Schriften, sondern geht auch auf einige Schriften ein, die den Sprung in das digitale Zeitalter nicht geschafft haben. Bei digitalen Standards wie der Garamond finden sich wertvolle Hinweise auf die Unterschiede zwischen den vielen, von verschiedenen Schriftanbietern auf den Markt gebrachten Versionen. Ebenso erfreulich ist, dass Bringhurst auch Fraktur, Schreibschriften, Griechisch und ähnliches berücksichtigt. Mehrere Anhänge und eine ausführliche Bibliographie beschließen den Band, den ich jedem ernsthaft an klassischer Typographie interessierten Leser empfehlen möchte.

Robert Bringhurst
The Elements of Typographic Style
2. Auflage 1996
Hartley & Marks
ISBN 0-88179-132-6
US-\$ 24,95 (Paperback)

Von fremden Bühnen

Original oder Fälschung?

Walter Schmidt

In Verbindung mit dem freien PostScript-Interpreter Ghostscript wird eine fehlerhafte Version der Schrift Helvetica verteilt. Der Aufsatz beschreibt, wie man diese durch die originale Adobe-Helvetica ersetzt, und wie letztere, völlig legal, kostenlos beschafft werden kann.

PostScript-Basisschriften

Bestimmte Schriften dürfen in PostScript-Dateien verwendet werden, ohne dass man sie mit dem Dokument mitliefern oder in die PostScript-Datei einbetten muss. Sie werden vom PostScript-Interpreter selbst zur Verfügung gestellt. Bei den üblichen Level-II-Interpretern sind das 35 Fonts des Herstellers Adobe, unter anderem auch die folgenden Schnitte aus der Schriftfamilie Helvetica: Helvetica, Helvetica-Oblique, Helvetica-Bold und Helvetica-BoldOblique. Das gilt auch für den PostScript-Interpreter Ghostscript, der PostScript-Dateien auf dem Bildschirm wiedergeben kann oder auf einem Drucker, der keinen eigenen PostScript-Interpreter besitzt.

Das Problem

Tatsächlich enthält die Ghostscript-Verteilung aber nicht die Adobe-Schriften, sondern Nachahmungen des Herstellers URW. Sie werden gern als gleichwertig mit Adobes Originalen bezeichnet, und es sind tatsächlich auch fast keine Unterschiede sichtbar. Leider nur „fast“, denn URWs Schrift Nimbus Sans, die die Helvetica ersetzen soll, ist mit letzterer nicht wirklich identisch, siehe Bild 1: In der URW Nimbus Sans ist die Letter ß zu schmal, so dass vor diesem Buchstaben eine Lücke im Wort entsteht. Die zugehörige afm-Datei nennt

grüßen groß Buße

grüßen groß Buße

Abbildung 1: Adobe Helvetica (oben) im Vergleich zu URW Nimbus (unten)

aber für die Letter ß die gleiche Zeichenbreite wie in der Adobe-Helvetica, so dass man hier von einem Fehler sprechen kann.

Die Lösung

Adobes Originalschriften beschaffen

Das Problem verschwindet, wenn man sich Adobes Originalschriften beschafft und Ghostscript so konfiguriert, dass er sie anstelle der URW-Fonts benutzt. Diese Fonts sind in diversen Software-Produkten von Adobe enthalten: Wer beispielsweise ATM für Windows oder den Acrobat Reader, Version 3.x, installiert hat, sollte keine Probleme haben, die folgenden Dateien auf der Festplatte zu lokalisieren:

```
HVBO_----.PFB  
HVB_-----.PFB  
HVO_-----.PFB  
HV_-----.PFB
```

Je nach Betriebssystem sind die Dateinamen eventuell klein statt groß geschrieben, und auf Unix-Systemen ist .PFB vermutlich durch .pfa zu ersetzen.

Finden sich die Dateien aber nicht, dann gibt es trotzdem einen Weg, sie kostenlos und völlig legal zu beschaffen, indem man sich zunächst einen Acrobat Reader, Version 3.x, besorgt. Aktuell ist zwar inzwischen die Version 4.05, aber sie enthält die betreffenden Schriften *nicht* mehr! Zum Glück verteilt Adobe auch die Version 3.x immer noch, siehe <http://www.adobe.com/products/acrobat/alternate.html>. Eventuell findet man Version 3.x des Acrobat Reader auch auf einer älteren CD-ROM, und selbst die veraltete Version 2.x genügt für unseren Zweck. Um an die Font-Dateien heranzukommen, muss der Reader entweder installiert werden, oder man extrahiert die .pfa-Dateien aus einer für Unix bestimmten Verteilung im tar-Format.

Es empfiehlt sich übrigens, auch die restlichen Type-1-Fonts aus der Acrobat-Verteilung herauszuziehen und zu sichern, denn Version 3.x des Acrobat Reader und damit die kostenlose Quelle dieser Schriften, wird sicherlich nicht „ewig“ verfügbar bleiben. Zu beachten ist, dass diese Fonts *nicht* weitergegeben werden dürfen; nur wer sich den kompletten Acrobat Reader beschafft hat, darf sie auch verwenden!

Ghostscript konfigurieren

Die oben genannten vier Dateien kopiert (oder linkt) man in das Verzeichnis, wo sich die mit Ghostscript verteilten Fonts befinden. Beispielsweise ist das unter Linux normalerweise das Verzeichnis `/usr/share/ghostscript/fonts/`. Weiterhin ist die *Fontmap*-Datei von Ghostscript entsprechend anzupassen. Mit dem aktuellen Ghostscript 6.x heißt sie `Fontmap.GS`, in früheren Versionen einfach `Fontmap`. In dieser Datei sind die Zeilen

```
/Helvetica                /NimbusSanL-Regu      ;
/Helvetica-Oblique        /NimbusSanL-ReguItal  ;
/Helvetica-Bold           /NimbusSanL-Bold      ;
/Helvetica-BoldOblique    /NimbusSanL-BoldItal  ;
```

zu ersetzen durch:

```
/Helvetica                (HV_____.PFB)  ;
/Helvetica-Oblique        (HVO____.PFB)  ;
/Helvetica-Bold           (HVB____.PFB)  ;
/Helvetica-BoldOblique    (HVBO____.PFB) ;
```

(Man beachte gegebenenfalls Klein- statt Großschreibung der Dateinamen und `.pfa` statt `.pfb`.)

Spielplan

Termine

- 12.8.–18.8.2000 TUG'2000 – “ $\text{T}_{\text{E}}\text{X}$ enters a new millenium”
21st annual meeting of the $\text{T}_{\text{E}}\text{X}$ User Group
Wadham College, Oxford, UK
Kontakt: $\text{T}_{\text{E}}\text{X}$ Users Group
- 11.9.–12.9.2000 PODDP'00 – 5th International Workshop on Principles
of Digital Document Processing
TU München, München
Der Workshop findet im Zusammenhang mit der DDEP
2000 direkt vor dieser statt.
- 13.9.–15.9.2000 DDEP 2000 – Digital Documents and Electronic Pu-
blishing 2000
TU München, München
Kontakt: Anne Brüggemann-Klein
- 6.10.–7.10.2000 Herbsttagung und 23. Mitgliederversammlung
von DANTE e.V.
Fernuniversität in Hagen
Kontakt: Thomas Feuerstack
- Frühjahr 2001 DANTE 2001 und 24. Mitgliederversammlung
von DANTE e.V.
Fachhochschule Rosenheim
Kontakt: Prof. Dr. Georg Lachenmayr

Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßig Treffen von T_EX-Anwendern statt, die für Jeden offen sind. Im WWW gibt es aktuelle Informationen unter <http://www.dante.de/dante/Stammtische.html>.

Berlin

Rolf Niepraschk
Tel.: 0 30/3 48 13 16
niepraschk@ptb.de
*Gaststätte „Bärenschenke“
Friedrichstr. 124
Zweiter Donnerstag im Monat, 19.00 Uhr*

Bremen

Martin Schröder
Tel.: 04 21/2 23 94 25
martin@oneiros.de
*Universität Bremen, Hoyers Unikum
Erster Donnerstag im Monat, 18.30 Uhr*

Chemnitz

Hans Friedrich Steffani
hans.steffani@e-technik.tu-chemnitz.de
*Bahnhofstr. 8, 1. OG, Raum G104
Zweiter Mittwoch im Monat, 18.00 Uhr*

Dortmund

Stephan Lehmke
Stephan.Lehmke@cs.uni-dortmund.de
*Cafe Durchblick
Universität Dortmund, Campus Nord
Zweiter Mittwoch im Monat, 20.00 Uhr*

Dresden

Hilmar Preuße
hille42@gmx.de
*Cafe B'liebig
Liebigstr. 24
Letzter Mittwoch im Monat, 19.00 Uhr*

Erlangen

Walter Schmidt, Peter Seitz
walter.schmidt@arcormail.de,
p.seitz@koehler-seitz.de
*Gaststätte „Gambrinus“
Vierzigmannstr. 7
Dritter Dienstag im Monat, 20.00 Uhr*

Freiburg

Heiko Oberdiek
Tel.: 07 61/4 34 05

oberdiek@ruf.uni-freiburg.de
*Gaststätte „Aquila“
Sautierstr. 19
Dritter Donnerstag im Monat, 19.30 Uhr*

Hamburg

Volker Hüttenrauch
volker_huettenrauch@hh.maus.de
*Vereinsheim der Hamburger
Microcomputer-Hochschulgruppe
Grindelallee 143 (Hinterhof)
Letzter Donnerstag im Monat, 18.00 Uhr*

Hannover

Mark Heisterkamp
Regionales Rechenzentrum
Tel.: 05 11/7 62 51 34
heisterkamp@rrzn.uni-hannover.de
*Seminarraum RRZN
Schloßwender Str. 5
Zweiter Mittwoch von geraden Monaten,
18.30 Uhr*

Heidelberg

Luzia Dietsche
Tel.: 06 21/54 45 27
luzia.dietsche@urz.uni-heidelberg.de
*China-Restaurant „Palast“, Lessingstr. 36
Letzter Mittwoch im Monat, 20.00 Uhr*

Karlsruhe

Klaus Braune
Tel.: 07 21/6 08 40 31
braune@rz.uni-karlsruhe.de
*Universität Karlsruhe, Rechenzentrum
Zirkel 2, 3. OG Raum 316
Erster Donnerstag im Monat, 19.30 Uhr*

München

Michael Niedermair
m.g.n@gmx.de
*Gastwirtschaft „Rhaetenhaus“
Luisenstr. 27
Erster Dienstag im Monat, 19.00 Uhr*

Münster

Johannes Reese
reesej@uni-muenster.de
Im Blauen Haus
Kreuzstr. 16/17
Erster Montag im Monat, 20.00 Uhr

Oldenburg

Konrad Blum
konrad@blum.physik.uni-oldenburg.de
Bistro „DaCapo“
Friedhofsweg 64
Zweiter Montag im Monat, 20.00 Uhr

Stuttgart

Marcus Schweizer
Tel.: 07 11/6 85 44 44
schweiz@theochem.uni-stuttgart.de
Gaststätte „Alte Mira“

Büchsenstr. 24
Zweiter Dienstag im Monat, 19.30 Uhr

Wiesbaden

Christian Kayssner
Tel.: 06 11/4 81 17
„Andreas Klause“
Elsässer Platz 3
Erster Montag im Monat, 20.00 Uhr

Wuppertal

Andreas Schrell
Tel.: 02 02/50 63 81
schrell@wupperonline.de
Croatia „Haus Johannisberg“
Südstr. 10, an der Schwimmoper
Wuppertal-Elberfeld
Zweiter Donnerstag im Monat, 19.30 Uhr

Adressen

DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 10 18 40
69008 Heidelberg

Tel.: 0 62 21/2 97 66 (Mo, Mi–Fr, 10⁰⁰–12⁰⁰ Uhr)
Fax: 0 62 21/16 79 06
E-Mail: dante@dante.de

Konten: Volksbank Rhein-Neckar eG
BLZ 670 900 00, Kontonummer 2 310 007
Postbank Karlsruhe Auslandsüberweisungen
BLZ 660 100 75, Kontonummer 213 400 757

Beiträge:	ermäßigte Mitgliedschaft	60,- DM
	Privatmitgliedschaft	80,- DM
	Institutionen des öffentlichen Rechts und Forschungseinrichtungen	120,- DM
	Firmen, die T _E X anwenden	300,- DM
	Firmen, die Produkte in Verbindung mit T _E X anbieten	500,- DM

Präsidium

Präsident:	Thomas Koch	president@dante.de
Vizepräsident:	Volker RW Schaa	vice-president@dante.de
Schatzmeister:	Horst Szillat	treasurer@dante.de
Schriftführer:	Günter Partosch	secretary@dante.de
Beisitzer:	Klaus Höppner	adviser@dante.de

Server

ftp: [ftp.dante.de](ftp://ftp.dante.de) [134.93.8.251]
E-Mail: ftpmail@dante.de
WWW: <http://www.dante.de/>

Autoren/Organisatoren

- Anne Brüggemann-Klein** [63] **Rolf Niepraschk** [11]
 Technische Universität München
 Arcisstraße 21
 80333 München
 Anne.Brueggemann-Klein@informatik.tu-muenchen.de
 Persiusstr. 12
 10245 Berlin
 niepraschk@ptb.de
- Luzia Dietsche** [5] **NTG T_EX future working group** [40]
 Universitätsrechenzentrum
 Im Neuenheimer Feld 293
 69120 Heidelberg
 luzia.dietsche@urz.uni-heidelberg.de
 Postbus 394,
 1740 AJ Schagen,
 Niederlande.
<http://www.ntg.nl>
 ntg-toekomsttex@ntg.nl
- Thomas Feuerstack** [7] **Bernd Raichle** [36]
 Fernuniversität in Hagen
 Universitätsrechenzentrum
 58084 Hagen
 Stettener Str. 73
 73732 Esslingen
 german@dante.de
- Joachim Grotherr** [14] **Volker RW Schaa** [4]
 Jackpot Spielservice GmbH
 Tratziger Straße 21
 22043 Hamburg
 achim@jackpotlotto.de
 siehe Seite 66
- Thomas Koch** [4, 7] **Walter Schmidt** [60]
 siehe Seite 66
 Schornbaumstraße 2
 91052 Erlangen
 walter.schmidt@arcormail.de
- Gerd Neugebauer** [3, 21] **Philip Taylor** [9]
 Mainzer Str. 8
 56321 Rhens
 gerd.neugebauer@sdm.de
 P.Taylor@alpha1.rhnc.ac.uk
- Peter Willadt** [16, 58]
 Heinrich-Wieland-Allee 5
 75177 Pforzheim
 willadt@t-online.de

Die T_EXnische Komödie

12. Jahrgang Heft 2/2000 Juli 2000

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 5 DANTE e.V. bei der MNU
- 8 Einladung zur Herbsttagung und 23. Mitgliederversammlung von DANTE e.V.
- 9 NTS Enters Alpha Test

Bretter, die die Welt bedeuten

- 11 Von der Wiege bis zur Bahre: Formulare
- 14 Serienbriefe mit L^AT_EX – Ein Erfahrungsbericht
- 16 Arial installieren
- 21 In Reih und Glied
- 35 T_EX-Rätsel

Was Sie schon immer über T_EX wissen wollten

- 36 Anlegen einer Liste der Definitionen

T_EX-Beiprogramm

- 40 T_EX im Jahr 2003: Vorschläge und Thesen zur Zukunft von T_EX

Rezensionen

- 58 Robert Bringhurst: The Elements of Typographic Style

Von fremden Bühnen

- 60 Original oder Fälschung?

Spielplan

- 63 Termine
- 64 Stammtische

Adressen

- 67 Autoren/Organisatoren