

# Die T<sub>E</sub>Xnische Komödie

---

DANTE  
Deutschsprachige  
Anwendervereinigung T<sub>E</sub>X e.V.

13. Jahrgang Heft 2/2001 Juli 2001

2/2001

# Impressum

---

„Die T<sub>E</sub>Xnische 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<sup>A</sup>T<sub>E</sub>X-Quellcode unter Verwendung der Dokumentenklasse `dtk` erstellt und an untenstehende Anschrift geschickt werden (entweder per E-Mail oder auf Diskette). Sind spezielle Makros, L<sup>A</sup>T<sub>E</sub>X-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 erstellt: **e-TeX**, Version 3.14159-2.1 (Web2c 7.3.2x), **LaTeX2e** (2000/06/01), **windvi** 0.67 für die Bildschirmdarstellung und **dvips(k)** 5.86d für Korrektur und Belichtung. Die Schriften zur Belichtung wurden mit dem METAFONT-Modus **linoone** (1270 dpi) berechnet.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2700

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung T<sub>E</sub>X e.V.  
Postfach 10 18 40  
69008 Heidelberg

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

Druck: Konrad Triltsch Print und digitale Medien GmbH  
Johannes-Gutenberg-Straße 1–3, 97199 Ochsenfurt-Hohe Stadt

Redaktion: Gerd Neugebauer (verantwortlicher Redakteur)

Luzia Dietsche	Rolf Niepraschk	Volker RW Schaa
Rudolf Herrmann	Günter Partosch	Karin Schwind
Uwe Münch	Bernd Raichle	Peter Willadt

Redaktionsschluss für Heft 3/2001: 3. August 2001

ISSN 1434-5897

*Die T<sub>E</sub>Xnische Komödie 2/2001*

# Editorial

---

Liebe Leserinnen und Leser,

„Die T<sub>E</sub>Xnische Komödie“ lebt! Waren meine zahlreichen Aufrufe doch endlich erfolgreich? Gibt es einfach Zeiten, zu denen nichts zu berichten ist und gehen diese wieder vorüber? Auf jeden Fall sind für diese Ausgabe der Vereinszeitschrift „Die T<sub>E</sub>Xnische Komödie“ rechtzeitig genügend Beiträge bei der Redaktion eingegangen. Und es sind sogar recht interessante Beiträge.

Eine Übersicht über verschiedene Pakete zum Setzen chemischer Gleichungen und Strukturformeln kann allen helfen, die in dieser Richtung aktiv werden wollen.

Briefe waren schon häufiger ein Thema in der Vereinszeitschrift. Nun bekommen wir gezeigt, wie man mit KOMA-Script personalisierte Briefköpfe erstellen kann.

T<sub>E</sub>X hatte ursprünglich als Ausgabeformat nur dvi bereitgestellt. Inzwischen hat PDF eine gewisse Bedeutung erlangt. In dieser Ausgabe gibt es eine Einführung, die zeigt, wie man einen Schritt in diese Richtung mit Hilfe von T<sub>E</sub>X vorankommen kann.

Eine für mich etwas exotische Anwendung von T<sub>E</sub>X haben wir auch. Schon länger gibt es ArabT<sub>E</sub>X, um arabische und hebräische Texte zu setzen. In einem Beitrag wird gezeigt, wie man in diesem Kontext mit dem Mathematiksatz zurecht kommt.

Abgerundet wird diese Ausgabe durch eine Rezension, die wohl eher als Verriß zu bezeichnen ist. Schließlich muss nicht jedes Buch ein voller Erfolg sein.

Ich wünsche unseren Lesern eine lehrreiche und unterhaltsame Lektüre dieser Ausgabe und bin schon gespannt darauf, welche Beiträge für die nächsten Ausgaben bei uns eintreffen.

Mit T<sub>E</sub>Xnischen Grüßen

Ihr Gerd Neugebauer

# Hinter der Bühne

---

Vereinsinternes

## Grußwort

Liebe Mitglieder,

Sie halten nun die zweite Ausgabe von „Die  $\text{T}_{\text{E}}\text{X}$ nische Komödie“ in den Händen und wundern sich vielleicht etwas. Zwar ist diese Ausgabe etwas umfangreicher geraten als üblich, aber die schon fast obligatorische  $\text{T}_{\text{E}}\text{X}$ Live-CD-ROM fehlt. Die Arbeiten an  $\text{T}_{\text{E}}\text{X}$ Live sind leider noch nicht abgeschlossen, sodass keine Möglichkeit besteht, diese CD-ROM zum jetzigen Zeitpunkt zu pressen. Die Gründe liegen hauptsächlich in einer vollständigen Überarbeitung der Skripte, die eine wesentlich bessere benutzerbezogene Installation ermöglichen werden. Es wird jetzt einfacher sein, eine auf die persönlichen Bedürfnisse zugeschnittenes System zu konfigurieren, ohne einen großen Ballast an nicht benötigten Dateien mitzuschleppen. So wurde die Auswahl der Pakete verfeinert, die spezifische Erweiterungen für Landessprachen, Fonts, Chemie, Musik, u. ä. bieten. Die Tests befinden sich in der Abschlussphase, und wir hoffen, dass die  $\text{T}_{\text{E}}\text{X}$ Live 6 der nächsten Ausgabe beiliegen wird.

Im Anschluss an das Grußwort finden Sie die Einladung zur 25. Mitgliederversammlung in Kerkrade/Niederlande. Wie wir schon mitgeteilt hatten, wird diese MV im Anschluss an die Euro $\text{T}_{\text{E}}\text{X}$ -Tagung stattfinden. Mittlerweile steht das Programm der Tagung und ein wichtiger Bestandteil ist: “Hear the latest news about  $\text{T}_{\text{E}}\text{X}$  developments (and meet the people involved)”. Es wird das Nachfolgeprojekt zu  $\mathcal{N}\mathcal{T}\mathcal{S}$  vorgestellt,  $\Omega$  präsentiert sich mit den ersten Ergebnissen der Reimplementierung/Überarbeitung (durchgeführt von einer 25 Personen umfassenden Gruppe im Rahmen eines Universitätsprojektes) und es gibt Neuigkeiten vom  $\text{T}_{\text{E}}\text{X}$ Lib-Projekt (eine  $\text{T}_{\text{E}}\text{X}$ -Reimplementierung in Library-Form). Fonts und Font-Tools, insbesondere für PostScript Type 1, spielen eine wichtige Rolle, und XML mit seinen Konversions-Tools nimmt breiten Raum ein. Wir denken, dass für jeden, der an der Weiterentwicklung und Benutzung von  $\text{T}_{\text{E}}\text{X}$  interessiert ist, interessante Vorträge dabei sein werden. Hinzukommt, dass es die Möglichkeit eines Wiedersehens mit Freunden

aus der T<sub>E</sub>X-Welt gibt, genauso wie Sie die „Macher“ und stillen Helfer hinter der Bühne kennenlernen können. Und das ganze in einer ungezwungenen familiären Umgebung.

Wir würden uns freuen, möglichst viele Mitglieder sowohl auf der EuroT<sub>E</sub>X-Tagung wie auch auf der Mitgliederversammlung begrüßen zu können,  
mit freundlichen Grüßen,

Thomas Koch    Volker RW Schaa  
(Präsident)    (Vizepräsident)

## Einladung zur 25. Mitgliederversammlung von DAN- TE e.V.

Thomas Koch

Hiermit laden wir Sie herzlich zur 25. Mitgliederversammlung von DANTE e.V. ein. Wie bereits im Grußwort der Ausgabe 1/2001 der Mitgliederzeitschrift „Die T<sub>E</sub>Xnische Komödie“ angekündigt, findet die Mitgliederversammlung am 28. September 2001 im Anschluss an die Tagung EuroT<sub>E</sub>X 2001 im

Congresscentrum Rolduc  
Heyendahllaan 82  
6464 EP Kerkrade  
Niederlande

statt.

Die Tagesordnung der Mitgliederversammlung am 28. September 2001 um 10.00 Uhr lautet:

1. Begrüßung; Vorstellung des Vorstands; Tagesordnung
2. Bericht des Vorstands
3. Verschiedenes

Wie üblich sind auch Nichtmitglieder als Gäste der Mitgliederversammlung willkommen.

Sie erreichen Kerkrade mit dem Auto über die Autobahn A 4 Köln–Aachen–Heerlen bis zum *Knooppunt Bocholtz* direkt hinter der Grenze, dann über die N 281 bis zur Abfahrt Kerkrade. Mit der Bahn fahren Sie zum Bahnhof Herzogenrath (erreichbar mit dem Regionalexpress oder der Regionalbahn ab Aachen bzw. mit dem Regionalexpress ab Düsseldorf). Ab dort sind es etwa 10–15 Minuten Fußweg zum *Congresscentrum Rolduc*.

Bitte melden Sie sich zur Mitgliederversammlung an – auch, wenn Sie sich bereits zur EuroT<sub>E</sub>X angemeldet haben. Sie erleichtern uns damit die Planung und verkürzen die Zeit für die Ausgabe der Stimmunterlagen. Das Anmeldeformular und weitere Informationen finden Sie unter <http://www.dante.de/dante/events/mv25>. Bei Fragen können Sie sich per E-Mail an die Adresse [mv25@dante.de](mailto:mv25@dante.de) wenden.

Wir würden uns, ebenso wie die Organisatoren der EuroT<sub>E</sub>X, freuen, Sie neben der Mitgliederversammlung auch auf der EuroT<sub>E</sub>X vom 23.–27. September 2001 begrüßen zu dürfen. Nähere Informationen und das Anmeldeformular zur Tagung finden Sie unter <http://www.ntg.nl/eurotex/>.

Falls Sie sowohl an der EuroT<sub>E</sub>X als auch an der Mitgliederversammlung teilnehmen wollen, buchen Sie bitte bei der Anmeldung zur EuroT<sub>E</sub>X eine zusätzliche Übernachtung vom Donnerstag, den 27. September, auf Freitag, den 28. September. Wenn Sie sich bereits zur EuroT<sub>E</sub>X angemeldet haben oder Sie nur an der Mitgliederversammlung teilnehmen und bereits am 27. September anreisen wollen, können Sie eine Übernachtung auch mit dem Anmeldeformular zur Mitgliederversammlung zum Preis von 40 EUR buchen und per Überweisung auf das Konto von DANTE e.V. bezahlen.

Zeitgleich zur Mitgliederversammlung von DANTE e.V. wird auch die Mitgliederversammlung der niederländischen T<sub>E</sub>X-Gruppe NTG stattfinden. Wir streben an, am Nachmittag des 28. September ein gemeinsames Treffen mit der NTG durchzuführen.

# Bretter, die die Welt bedeuten

---

## L<sup>A</sup>T<sub>E</sub>X und Chemie – ein schönes Paar?

Michael Gohdes

Dieser Artikel beschreibt Möglichkeiten, Texte chemischen Inhalts mit L<sup>A</sup>T<sub>E</sub>X zu setzen. Es wird besonders auf die Frage eingegangen, wie chemische Gleichungen und Strukturformeln in den Text eingebettet werden können. Sowohl verschiedene L<sup>A</sup>T<sub>E</sub>X-Pakete als auch ein externes Programm werden vorgestellt.

### Einleitung

Nachdem ich in den Heften der Vereinszeitschrift ab Ausgabe 2/1994 geblättert und kein Wort zum Thema „L<sup>A</sup>T<sub>E</sub>X und Chemie“ gefunden habe, ist es – denke ich – an der Zeit, dazu etwas zu sagen.

Ich hatte mich im Laufe meines Chemiestudiums bereits nach kurzem Kontakt mit L<sup>A</sup>T<sub>E</sub>X dazu entschlossen, meine Promotion mit diesem vielversprechenden Textsatzprogramm zu schreiben. Die Nervereien mit WinWord während meiner Diplomarbeit waren Grund genug, der damals sehr instabilen Software (Version 2) die Nase zu zeigen. Durch meine Frau, die ihre Arbeiten im Ingenieurstudium mit L<sup>A</sup>T<sub>E</sub>X schreiben musste, bekam ich Lust, damit zu experimentieren. Sehr schnell ergaben sich zwei Probleme:

- Chemische Gleichungen und
- Chemische Strukturformeln

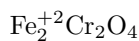
Im folgenden Beitrag werde ich zu beiden Themen etwas sagen, ohne auf einer Vollständigkeit bestehen zu wollen. Noch ein Wort zu meiner Hardware: Ich hatte mir damals ein Notebook zugelegt (PC 486-SX 33, 8 MB, 320 MByte HD), sodass ich im Labor und zu Hause an meiner Arbeit schreiben konnte. Für einen Text von fast 200 Seiten wäre diese Ausstattung unter

Windows etwas überfordert gewesen, L<sup>A</sup>T<sub>E</sub>X (eine emT<sub>E</sub>X-Installation unter DOS) hat nicht einmal geschwitzt.

Nebenbei: Dieser Rechner ist weiterhin mein treuer L<sup>A</sup>T<sub>E</sub>X-Diener, auch eine Erweiterung auf viele PostScript-Fonts wurde klaglos hingenommen.

## Chemische Gleichungen

Die Probleme mit chemischen Gleichungen werden durch folgendes Beispiel schnell auf den Punkt gebracht:



Eingabetext:

```
\underline{\mathrm{Fe}_2^{+2}\text{Cr}_2\text{O}_4}
```

Zur Verdeutlichung wurde die Formel unterstrichen. Die Indizes liegen unterschiedlich tief, abhängig davon, ob zusätzlich ein Exponent an demselben Atom vorkommt. Aber auch die Indizes am Cr und O haben in meinen Augen eine kleine Schwäche: Sie stehen zu hoch. Eine gute Lösung bietet H. Kopka in seiner L<sup>A</sup>T<sub>E</sub>X-Einführung [?] an. Die Ursache liegt in der Behandlung der T<sub>E</sub>X-Befehle `\fontdimen`. Die Befehle für  $n = 16$  und  $n = 17$  bestimmen den Betrag von Tiefstellungen. Eine einheitliche Neudefinition kann hier Abhilfe schaffen. Dies ist genau das Verfahren, welches ich benutzt habe. Ein einfaches Makro für die Schriftgröße 10 pt könnte nach H. Kopka [?] mit einer Anpassung für L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> von B. Raichle [?] folgendermaßen aussehen:

```
\makeatletter
\DeclareRobustCommand{\chem}[1]{%
  {\(\m@th
    \edef\resetfontdimens{\noexpand\)}%
  \fontdimen16\textfont2=\the\fontdimen16\textfont2
  \fontdimen17\textfont2=\the\fontdimen17\textfont2\relax}%
  \fontdimen16\textfont2=2.7pt \fontdimen17\textfont2=2.7pt
  \mathrm{#1}%
  \resetfontdimens}}
\makeatother
```

Analog können so für unterschiedliche Schriftgrößen Makros mit anderen Werten für die Tiefstellung der Indizes geschrieben werden, etwa mit 2,8 pt Tiefstellung für 11 pt-Texte und 2,9 pt für 12 pt-Texte.



Tabelle 1: Vergleich der vorgestellten Formatierungen von chemischen Gleichungen

A. Klassisch:	$\mathrm{Fe}_2^{+2}\mathrm{Cr}_2\mathrm{O}_4$	ergibt	$\mathrm{Fe}_2^{+2}\mathrm{Cr}_2\mathrm{O}_4$
B. <code>\chem</code> -Makro:	<code>\chem{Fe_2^{+2}Cr_2O_4}</code>	ergibt	$\mathrm{Fe}_2^{+2}\mathrm{Cr}_2\mathrm{O}_4$
C. <code>chemsym</code> :	<code>\Fe_2^{+2}\Cr_2\O_4</code>	ergibt	$\mathrm{Fe}_2^{+2}\mathrm{Cr}_2\mathrm{O}_4$

Im Laufe der Recherche zu diesem Thema bin ich in den unerforschlichen Weiten des CTAN auf das Paket `chemsym` [?] gestoßen. Es ist von M. Dahlgren geschrieben, liegt in der Version 2.0 vom Mai 1998 vor und definiert 116 Befehle, u. a. alle chemischen Elemente und verschiedene Substituentengruppen der organischen Chemie. Alle Elemente werden als L<sup>A</sup>T<sub>E</sub>X-Makro (beispielsweise `\Hg` für Hg), Indizes einfach mit `_` und Exponenten mit `^` eingegeben, indem diese Zeichen als aktiv definiert werden. Die Abstände innerhalb von Gleichungen sind etwas größer als im einfachen mathematischen Modus.

Auch in Formulierungen von Isotopen wie beispielsweise  $^{14}_7\mathrm{N}$  oder  $^{238}_{92}\mathrm{U}$  wird die Fähigkeit verlangt, Index und Exponent übereinander zu setzen. Mit `chemsym` ergibt sich mit der Eingabe von `{^{238}_{92}\U}` folgendes Aussehen:  $^{238}_{92}\mathrm{U}$  – beide Zahlen stehen linksbündig. Aber auch beim `\chem`-Makro muss zu einem kleinen Trick gegriffen werden. Die Eingabe `\chem{^{238}_{92}\U}` erzeugt nämlich ebenfalls  $^{238}_{92}\mathrm{U}$ , da sich die Hoch- und Tiefstellung an einem imaginären Zeichen *vor* dem Uran-Symbol orientiert und folgerichtig beide Zahlen linksbündig setzt. Abhilfe folgt mit dem T<sub>E</sub>X-Befehl `\phantom{Zeichen}`, der ein unsichtbares *Zeichen* mit seinem natürlichen Raumbedarf setzt. Es muss also heißen: `\chem{^{238}_{\phantom{2}92}\U}`, um zum richtigen Ergebnis  $^{238}_{92}\mathrm{U}$  zu kommen.

Als Fazit bleibt das kleine Makro von H. Kopka und B. Raichle weiterhin das Optimum für den chemischen Formelsatz. Die Tabelle 1 gibt einen Überblick über alle drei vorgestellten Möglichkeiten.

## Chemische Strukturformeln

Im zweiten Teil dieses Artikels möchte ich auf die Möglichkeiten eingehen, chemische Strukturformeln in T<sub>E</sub>Xnischen Texten einzusetzen. Es ist bereits sehr viel über die Kombination von (L<sup>A</sup>)T<sub>E</sub>X und Graphik geschrieben worden.

Abbildung 1: Mit X<sup>M</sup>T<sub>E</sub>X erstellte Reaktionsgleichung

Abbildung 2: Mit streeT<sub>E</sub>X erzeugte Reaktionsgleichung

Ein rein textbasiertes Satzprogramm stellt hohe Anforderungen zur Erstellung dieser sehr speziellen Graphiken. Es besteht ein Spannungsfeld zwischen der Variabilität bezüglich der unterschiedlichsten Molekülstrukturen einerseits und einer wünschenswerten Anwenderfreundlichkeit andererseits.

Hier soll es um praktikable Ansätze gehen, die es ermöglichen, schnell zu hochwertigen chemischen Strukturformeln zu gelangen, ohne tagelang mit Geodreieck und trigonometrischem Restwissen dramatisch geformte Polygone zu berechnen.

Ich möchte hier vier sehr unterschiedliche Ansätze vorstellen, chemische Strukturformeln zu erzeugen, und zwar

- X<sup>M</sup>T<sub>E</sub>X von S. Fujita [?],
- streeT<sub>E</sub>X von I. Stokov [?],
- PPCHT<sub>E</sub>X von J. Hagen und T. Otten [?] und
- PLT, ein DOS-Shareware-Programm von H. J. Reich [?]

Die Möglichkeiten der verschiedenen L<sup>A</sup>T<sub>E</sub>X-Pakete werden am Beispiel der Reaktion von Abietinsäure unter Wärmeeinwirkung demonstriert (siehe Abbildungen 1 bis 3). Zusätzlich wird als Vergleich der Ausdruck des DOS-Programms PLT nach Überführung in eine .eps-Datei angeführt (Abbildung 4). Diese Reaktion wurde recht willkürlich aus einem Chemielehrbuch ausgewählt. Sie ist nicht trivial, aber auch nicht allzu kompliziert – die Befehle können auf diese Weise noch nachvollzogen werden. Im Folgenden werden die zugrunde liegenden Befehle kommentiert.

Abbildung 3: Mit PPCHT<sub>E</sub>X erzeugte Reaktionsgleichung

Abbildung 4: Mit PLT erzeugte Reaktionsgleichung

X<sup>Y</sup>MT<sub>E</sub>X

Das X<sup>Y</sup>MT<sub>E</sub>X-Paket liegt in der Version 1.00 vor und wird offensichtlich seit 1993 nicht mehr weiterentwickelt, was ich sehr schade finde. Es setzt auf der `picture`-Umgebung auf und definiert sehr viele verschiedene Makros zur Erstellung von Strukturen. Es können

- Aliphatische Verbindungen,
- Carbocyclische Verbindungen,
- Bi- und Multicyclische Verbindungen,
- Heterocyclische Verbindungen,
- Pyranose- und Furanoseringe und
- verschiedene Bindungstypen

gezeichnet werden.

Die Abbildung 1 wurde mit folgenden Makros erzeugt:

```
\hphenanthrenev[am]{2==CH(CH$_3$)$_2$;8SB==H$_3$C;8SA==COOH;%
  {12}FB==\lmoiety{H$_3$C};{11}FA==H}
\quad\raisebox{15mm}{\$\longrightarrow$}
\phenanthrenev{2==CH(CH$_3$)$_2$;8==CH$_3$}
```

Die Makros `\phenanthrenev` und `\hphenanthrenev` erzeugen jeweils das Phenanthren-Grundgerüst, einmal bereits mit sämtlichen (aromatischen) Bindungen, einmal ohne. Diese Gerüste werden um weitere Bindungen und Substituenten ergänzt. Beispielsweise erzeugt `2==CH(CH$_3$)$_2$` die Isopropylgruppe an Position 2 (rechts oben). Die beiden zusätzlichen Bindungen in den Ringen des linken Moleküls werden durch die Option `[am]` bewirkt, jeder C-C-Bindung ist ein Buchstabe von a bis p eindeutig zugeordnet.

Im Unterschied zum erwünschten Ergebnis in Abbildung 4 fallen drei Dinge auf. Es ist in X<sup>Y</sup>MT<sub>E</sub>X anscheinend unmöglich, eine Bindung in den „Innenraum“ eines Ringes ragen zu lassen, sodass sich hier die H<sub>3</sub>C-Gruppe mit dem H-Atom überlagert. Weiterhin sind die Bindungen zu den beiden

H<sub>3</sub>C-Gruppen zwar fett, aber nicht keilförmig; gestrichelte Bindungen sind ebenfalls nicht möglich.

Die Makroschreibweise von X<sup>M</sup>T<sub>E</sub>X ist eingängig und schnell erlernbar. Viele Molekülstrukturen sind vorgegeben, sollten aber weiter ergänzt werden. Es ist daher umso mehr zu wünschen, daß der Autor des Pakets die Weiterentwicklung nicht einstellt – noch liegt die Software im CTAN unter „supported“.

### streeT<sub>E</sub>X

Einen ganz anderen Ansatz verfolgt I. Stokrov mit seinem Paket streeT<sub>E</sub>X aus dem Jahr 1996, welches ebenfalls auf der picture-Umgebung basiert. Er bemängelt an Paketen wie X<sup>M</sup>T<sub>E</sub>X, dass sie unflexibel und komplizierte Strukturen nur sehr umständlich zu konstruieren seien. Seine Lösung setzt einige Ebenen tiefer an, indem die Moleküle vom Benutzer selbst aufgebaut werden. Es gibt nur einen einzigen Befehl (`\stree{}`), mit dem alle Varianten gezeichnet werden.

Zur Erzeugung der Abbildung 2 wurden folgende Zeilen eingegeben:

```
\stree{
  12 2 4 6 8 [^*7{H$_3$C}] [:5{COOH}] 10 % Ring links
  4 2 % zum mittleren Ring
  4 2 <12 10 [:6H] 8 [^*12{H$_3$C}] % mittlerer Ring
  2 % zum rechten Ring
  12 2 4 [2{CH(CH$_3$)$$_2$}] >6 8 % Ring rechts
}
$\longrightarrow$\quad
\stree{
  12 >2 4 >6 8 [6{CH$_3$}] >10 % Ring links
  4 2 % zum mittleren Ring
  4 <2 12 <10 8 % mittlerer Ring
  2 % zum rechten Ring
  12 >2 4 [2{CH(CH$_3$)$$_2$}] >6 8 % Ring rechts
}
```

Die Zahlen entsprechen den Richtungen auf dem Zifferblatt einer Uhr. Es wird auf diese Weise ein Linienzug definiert, der um Substituenten und weitere Bindungen ergänzt wird. Beide Strukturen wurden an der linken unteren Ecke begonnen und nach oben weitergezeichnet (12 2 4...), wobei die Kommentierung rechts die Zuordnung erleichtert. Eine eckige Klammer wird um

zusätzliche Elemente geschrieben, ohne den Bezugspunkt weiterwandern zu lassen, ein < oder > wird vor innen- bzw. außenliegenden Doppelbindungen eingefügt.

Es ergibt sich so eine große Flexibilität; die Strukturen entsprechen exakt denen aus dem Lehrbuch. Das Paket definiert weitere Raumrichtungen; eigene Bindungslängen und -orientierungen können leicht selbst erstellt werden. Die Syntax ist sehr leicht erlernbar; die fertigen Makros sind übersichtlich und leicht nachträglich zu editieren.

## PPCH<sub>T</sub>E<sub>X</sub>

Das Paket PPCH<sub>T</sub>E<sub>X</sub> wurde ursprünglich für CON<sub>T</sub>E<sub>X</sub>T entwickelt, das mir leider bisher unbekannt geblieben ist. Es ist auch unter T<sub>E</sub>X und L<sup>A</sup>T<sub>E</sub>X lauffähig und wurde ebenfalls für diesen Artikel getestet. In diesem Fall liegt das P<sub>C</sub>T<sub>E</sub>X-Paket zugrunde, der Name ergibt sich aus der Abkürzung für „Plain P<sub>C</sub>T<sub>E</sub>X CON<sub>T</sub>E<sub>X</sub>T Chemie“. Es gibt auch in diesem Fall nur einen zentralen Zeichenbefehl (`\chemie[]`), der mit `\startchemie` eingeleitet und mit `\stopchemie` beendet wird. Ähnlich wie in X<sup>Y</sup>M<sub>T</sub>E<sub>X</sub> existieren vordefinierte Strukturen, die mit Leben gefüllt werden müssen, z. B.

- Ringstrukturen mit 3-, 4-, 5-, 6- und 8-Ringen
- Haworth-Projektionen
- Newman-Projektionen
- Chair-Konformation für Cyclohexan

Die Struktur in Abbildung 3 wurde mit folgendem Makro erstellt:

```
\stellechemieein[format=klein,groesse=gross,breite=passend]
\mbox{
  \startchemie
  \chemie[SIX,B,-RB1,+RB3,-RD3,-RZ1,+RZ3,-RZ3,% Ring links
  MOV1,B,EB1,-RD6,-RZ6,%           % mittlerer Ring
  MOV6,B,EB1,R1,RZ1]%           % Ring rechts
  [H_3C,H_3C,COOH,H,CH(CH_3)_2]   % Substituenten
  \stopchemie}
\mbox{
  $\longrightarrow$\quad
  \startchemie
```

```

\chemie[SIX,B,EB135,R3,RZ3,%           % Ring links
MOV1,B,EB26,%                         % mittlerer Ring
MOV6,B,EB15,R1,RZ1]%                 % Ring rechts
[CH_3,CH(CH_3)_2]                   % Substituenten
\stopchemie}

```

Die Grundstruktur ist hier der carbocyclische Sechsring, der zweimal verschoben werden muß (MOV1 und MOV6), um zum Phenanthren-Molekül zu gelangen. Die Bindungen innerhalb der Ringe werden mit B, spezielle Bindungsrichtungen und -arten mit Befehlen wie -RB1 oder +RZ3 erzeugt, die dem Handbuch entnommen werden müssen und nicht direkt eingängig sind. Beim Durchblättern des Handbuches fällt auf, daß PPCHT<sub>E</sub>X weitaus mehr Möglichkeiten bietet als X<sup>f</sup>M<sub>T</sub>E<sub>X</sub>, diese aber nur schwer nachvollziehbar sind. Vielleicht liegt es an meiner mangelnden Erfahrung mit CONTEX<sub>T</sub>, daß ich mich schwer tat mit PPCHT<sub>E</sub>X, aber der Aufwand zur Erstellung der Grafik war ca. 10-mal größer, als mit X<sup>f</sup>M<sub>T</sub>E<sub>X</sub> oder erst recht mit stree<sub>T</sub>E<sub>X</sub>. Auch war es mir nicht möglich, die wie bei X<sup>f</sup>M<sub>T</sub>E<sub>X</sub> überlappenden Substituenten zu separieren oder die Bindung der mittleren H<sub>3</sub>C-Gruppe im linken Molekül auf das Kohlenstoffatom weisen zu lassen; die vorliegende Form ist jedenfalls chemisch falsch. Da hier das P<sub>I</sub>CT<sub>E</sub>X-Paket zuvor geladen werden muß, sind einige Voreinstellungen nötig, um nicht die Meldung „TeX-capacity exceeded!“ zu erhalten. So ist eine Erhöhung der Größe von parameter stack size auf 1000 sehr sinnvoll (unter em<sub>T</sub>E<sub>X</sub>: /ma:1000 als Kommandozeilen-Option). Die Rechenzeit ist im Vergleich lang, auf schnelleren Rechnern als meinem Oldtimer aber wohl vernachlässigbar.

## PLT

Das DOS-Programm PLT, mit dem die Abbildung 4 erstellt wurde, nimmt hier natürlich eine Sonderstellung ein. Ich habe es in diesen Reigen mit aufgenommen, da ich es in meiner Promotionsarbeit als Molekülzeichenprogramm eingesetzt habe. Ich habe unter Windows 3.11 gearbeitet und ließ es zusammen mit der T<sub>E</sub>XShell von J. Schlegelmilch [?] parallel laufen, um Zeit zu sparen. Die erzeugten Plots wurden als .eps-Datei exportiert und so in den Text eingebettet. Mit Hilfe dieses Programms sind natürlich viele Tricks möglich, um Moleküle zu zeichnen. Es liegt am Benutzer, selbstgesetzte Standards bezüglich Bindungslänge, Schriftgröße, Schriftart, Strichstärke u. ä. einzuhalten. Verschiedene Einstellungen können leicht abgespeichert werden, um hier unterschiedlichen Anforderungen gerecht zu werden. Abbildung 5 zeigt einen Screenshot aus dem Programm.

Abbildung 5: Screenshot des Programms PLT

Abbildung 6: Mit PLT erzeugte Graphik im \*.eps-Format

Als bevorzugte Schriftart wurde Helvetica eingesetzt, da Sans-Serif-Zeichen (in meinen Augen) die Übersichtlichkeit von Strukturformeln erhöhen und sich besser vom Fließtext abheben. Um hier eine größere Vergleichbarkeit zu gewährleisten, wurde als Schrift die Times eingestellt. Der Einsatz von Sans-Serif-Schriften ist bei den oben vorgestellten Makro-Paketen teilweise mit Schwierigkeiten verbunden, da z. B. PPCHT<sub>E</sub>X standardmäßig im mathematischen Modus arbeitet und die Neudefinition der Schriftart Auswirkungen auf alle mathematischen Formeln hätte. Auch das \chem-Makro funktioniert nicht mehr, die Vorteile können leider in den vorgestellten Paketen nicht genutzt werden.

## Fazit

Es gibt für den Chemiker einige Dinge zu bedenken, bevor Texte mit L<sup>A</sup>T<sub>E</sub>X gesetzt werden. Besonders chemische Formeln und Strukturen können Kopferbrechen bereiten. Ich habe in diesem Artikel meine Erfahrungen aus dem Schreiben meiner Promotionsarbeit zusammengefasst und hoffe, dass der eine oder andere Tipp hilfreich war.

Drei Pakete habe ich nicht erwähnt. Zwei davon sind nur unter L<sup>A</sup>T<sub>E</sub>X 2.09 lauffähig, ich wurde beim Test mit Fehlermeldungen im Kompatibilitätsmodus von L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> bombardiert, nämlich chemtex von R. Haas und K. C. O’Kane [?] und chemstruct von M. Ramek [?]. Das ochem-Paket von I. Klöckl [?] läuft, soweit ich sehen kann, nur unter Linux, was ich gerade erst einrichte und deshalb leider noch nicht testen kann.

Zusammenfassend möchte ich sagen, daß ich vielleicht eine andere Wahl getroffen hätte, wenn ich die Pakete früher kennengelernt hätte. In meiner Arbeit kam es besonders darauf an, gute Strukturformeln für Mono- und Polysaccharide zu gestalten, wie sie in Abbildung 6 zu sehen sind. Da ich damals nur das hierfür nicht sehr gut geeignete X<sup>M</sup>T<sub>E</sub>X kannte, war die Qual der Wahl nicht groß. Wenn ich mir allerdings stree<sub>T</sub>E<sub>X</sub> genauer ansehe, können hiermit auch derartige Graphiken erstellt werden, wie das Beispiel in

Abbildung 7: Mit `streeTeX` erzeugte Zuckerstruktur

Abbildung 7 zeigt. Ich glaube, die Chancen für `streeTeX` hätten nach einer genaueren Untersuchung nicht schlecht gestanden.

Wie schon zu Beginn betont, erhebe ich keinerlei Anspruch auf Vollständigkeit und bin für weitere Diskussionen dankbar. Ich schreibe zwar keinerlei chemische Texte mehr, habe aber weiterhin viel Spaß beim Lösen `TeX`nischer Probleme.

## Briefköpfe mit KOMA-Script

### Markus Kohm

Wer eben mal rasch einen Brief schreiben will, stört sich häufig daran, dass er erst eine große Präambel zusammenbasteln oder seine Standardpräambel aus einem alten Brief kopieren muss. Soll dann je nach Brief eine Angabe mehr oder weniger im Briefkopf gesetzt werden, muss man zudem jedesmal von neuem überlegen, wie dies oder jenes am besten erreicht wird. Der folgende Artikel soll zeigen, dass es auch anders geht: mit einem Paket, das einmal alle Varianten implementiert und durch Optionen auswählbar macht.

### Bevor ich loslege

Bevor ich loslege, lassen Sie mich etwas vorausschicken. Im Laufe dieses Artikels entsteht ein voll funktionsfähiges Paket. Dabei wird der Eindruck erzeugt, ein Paket könne real in dieser Form entstehen. Das ist nur die halbe Wahrheit. Normalerweise entstehen die meisten Pakete dadurch, dass Feature für Feature implementiert wird. Oftmals bauen Features aufeinander auf oder bestehen aus unterschiedlichen Teilen. Betrachtet man beispielsweise `\section` als ein Feature, so besteht es aus der Definition des Gliederungsbefehls, des Inhaltsverzeichniseintrags, der Darstellung der Abschnittsnummer und weiteren Teilen. Diese Teile sind in der Klassen- oder Paketdatei meist in unterschiedlichen Abschnitten verstreut, eben einem Abschnitt für Gliederungsbefehle, einem für das Inhaltsverzeichnis und einem für Zähler. Bei der



Implementierung muss daher immer wieder zwischen solchen Abschnitten hin und her gesprungen werden.

Bei der hier vorgestellten Implementierung wird ein anderer Weg beschritten. Zunächst werden die Features geplant und dann für jeden einzelnen logischen Abschnitt die notwendigen Implementierungen vorgenommen. Es wird also nicht in Reihenfolge der Features, sondern in Reihenfolge der logischen Unterteilung des Pakets gearbeitet. Auf diesem Weg wird erreicht, dass die Teile genau in der Reihenfolge, in der sie im Text auftauchen, in eine Paketdatei geschrieben werden können.

Die hier vorgeführte Implementierung eines Briefkopfpaketes geht von der Briefklasse des KOMA-Script-Paketes, `scrlettr.cls`, aus (siehe [?]). Der einzige Grund dafür ist, dass ich mich mit dieser Briefklasse am besten auskenne. Es ist jedoch sicher möglich, das Paket für andere Klassen anzupassen oder sogar so zu erweitern, dass es mit unterschiedlichen Briefklassen zusammenarbeitet. Auf die Möglichkeiten von `scrlettr.cls` wird nicht im Detail eingegangen. Sie finden diese in der Anleitung zu KOMA-Script, dem `scrguide` (siehe ebenfalls [?]).

## Das Paket

Das Paket soll später so einiges können. Angedeutet sind bereits ein paar Punkte, aber mit Andeutungen kann man kein Paket planen. Stattdessen ist zusammenzustellen und festzuhalten, was das Paket am Ende wirklich können soll. Wir formulieren deshalb die *Anforderungen* an das neue Paket:

1. Privatbriefe und Geschäftsbriefe werden getrennt voneinander behandelt.
2. Der Absender kann auf der ersten Seite optional oben links, in der Mitte oder oben rechts gesetzt werden.
3. Optional kann der Absender durch eine Linie entweder unter dem Namen oder unter der kompletten Angabe hervorgehoben werden.
4. Telefonnummer, Faxnummer, E-Mail-Adresse und URL der eigenen Homepage können im Briefkopf optional gesetzt werden.
5. Die Bankverbindung kann optional im Fuß der ersten Seite oder neben dem Adressfeld gesetzt werden.
6. Im Briefkopf kann optional ein Logo eingebunden werden.

7. Optional können so genannte Falt- oder Falzmarken gesetzt werden.
8. Das Adressfeld kann optional eingeblendet werden.
9. Es kann optional mit Rücksendeadresse für Fensterbriefumschläge gearbeitet werden.
10. Alle Absenderangaben werden einfach per Option aus einer Absenderdatei gelesen.
11. Die Grundeinstellungen des Pakets können aus einer Konfigurationsdatei gelesen werden.
12. Alle Absenderangaben werden mit einfachen Anweisungen definiert, sei es im Brief oder in der Absenderdatei.

Nachdem wir nun wissen, was das Paket leisten soll, beginnen wir im nächsten Abschnitt mit der Implementierung.

## Am Anfang jedes Pakets

Am Anfang jedes Pakets sollten einige Kommentarzeilen zu finden sein, die folgende Informationen beinhalten:

- Name des Pakets und der Paketdatei
- Name und Erreichbarkeit des Paketautors
- Erscheinungszustand des Pakets
- Rechtlicher Status des Pakets
- Zweck des Pakets

Im Beispiel sieht das dann so aus:

```
% briefkopf.sty
% Copyright (c) 1998-2001 Markus Kohm
% E-Mail: Markus.Kohm@gmx.de
%
% This is only a unsupported beta version!
%
% This file may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either
% version 1.2 of this license or (at your option) any later
```

```

% version.
% The latest version of this license is in
% http://www.latex-project.org/lppl.txt
% and version 1.2 or later is part of all distributions of
% LaTeX version 1999/12/01 or later.
%
% This package implements letter-heads to be used with letter
% class ‘scrletter’ of KOMA-Script bundle.

```

Kann das Paket nur mit einem bestimmten T<sub>E</sub>X-Format oder nur ab einer bestimmten Version verwendet werden, so ist dies nun formal festzulegen. Darüber hinaus wird an dieser Stelle L<sup>A</sup>T<sub>E</sub>X der Name, das Versionsdatum und eine weitere Identifikation formal mitgeteilt. Ein Versionsdatum ist dabei immer in der Form *YYYY/MM/DD* anzugeben. Dabei steht *YYYY* für das Jahr einschließlich Jahrhundert, *MM* für den zweistelligen numerischen Monat und *DD* für den zweistelligen Tag im Monat.

```

\NeedsTeXFormat{LaTeX2e}[1998/06/01]
\ProvidesPackage{briefkopf}
  [1999/09/12 v0.5beta letter-heads with scrletter]

```

Damit sind die Vorarbeiten abgeschlossen und im nächsten Abschnitt widmen wir uns den Optionen.

## Die Optionen des Pakets

Die Optionen des Pakets sind die Schalter, mit denen die Eigenschaften eines Pakets global und zentral beeinflusst werden können. Beim Lesen der Anforderungen zu diesem Paket ist sicher aufgefallen, dass sehr häufig das Wort „*optional*“ zu finden ist. All diese Punkte können wir als Optionen implementieren und werden dies nun auch Schritt für Schritt tun.

### Anforderung 1: Privat- und Geschäftsbriefe

Wir werden zwei Optionen definieren, mit denen entweder das Layout für Geschäftsbriefe oder das Layout für Privatbriefe aktiviert wird. Um später in der Layout-Implementierung zwischen den beiden unterscheiden zu können, wird ein Schalter benötigt. L<sup>A</sup>T<sub>E</sub>X stellt eigens für die Definition solcher Schalter die Anweisung `\newif` zur Verfügung, dem als Argument der Name des Verzweigungsbefehls übergeben wird:

```
\newif\if@privat
```

L<sup>A</sup>T<sub>E</sub>X definiert dabei nicht nur den Verzweigungsbefehl `\if@privat`, der in der Form „`\if@privat dann führe dies aus \else sonst führe das aus \fi`“ verwendet werden kann. Es werden außerdem die beiden Schaltbefehle `\@privattrue` und `\@privatfalse` definiert, mit denen der Schalter ein- und ausgeschaltet werden kann.

Bei der Definition der beiden Klassenoptionen, `privat` für Privatbriefe und `business` für Geschäftsbriefe, wird von dieser Tatsache Gebrauch gemacht:

```
\DeclareOption{privat}{\@privattrue}
\DeclareOption{business}{\@privatfalse}
```

Damit besteht nun die Möglichkeit, mit `\usepackage[privat]{briefkopf}` das Layout für Privatbriefe oder mit `\usepackage[business]{briefkopf}` das Layout für Geschäftsbriefe zu wählen. Da nach `\newif` der Schalter zunächst im Zustand *aus* ist, sind Geschäftsbriefe voreingestellt.

## Anforderung 2: Absenderposition

Die Wahl der Absenderposition ist nicht über einen einfachen Schalter möglich, da wir drei verschiedene Positionen vorgesehen haben, ein solcher Schalter aber nur zwei Zustände haben kann. L<sup>A</sup>T<sub>E</sub>X bietet neben *if-then-else-fi*-Konstrukten aber auch ein *ifcase-or-else-fi*-Konstrukt. Dabei kann aufgrund eines numerischen Werts zwischen maximal 65536 Alternativen ausgewählt werden. Dabei steht nach `\ifcase` der Auswahlwert gefolgt von der Alternative, falls dieser Auswahlwert gleich 0 ist. Nach jedem `\or` steht die Alternative für den nächst höheren Auswahlwert. Schließlich kann hinter `\else` eine Alternative folgen, die ausgeführt wird, wenn der Auswahlwert irgend einen anderen Wert besitzt. Deutlicher wird dies weiter unten bei der Implementierung der Alternativen (Abschnitt „Die Layoutdefinition“, Seite 28).

Als Auswahlwert werden häufig Zähler verwendet. Dies ist jedoch nicht zwingend. Es muss sich lediglich um eine positive, ganzzahlige, numerische Angabe handeln. Ein Makro kann ebenfalls eine solche Angabe beinhalten und kostet zudem keines der wenigen Zählerregister, von denen T<sub>E</sub>X nur 256 besitzt. Daher verwenden wir ein Makro. Diesem geben wir vorab eine Voreinstellung:

```
\newcommand*{\@alignfrom}{0}
```

Da das Makro `\@alignfrom` keinen Parameter erwartet, könnte hier an Stelle der Sternvariante `\newcommand*` auch `\newcommand` verwendet werden. Ich habe mir jedoch angewöhnt, immer dann die Sternvariante zu verwenden, wenn ich nicht bewusst ein Makro *long* definieren will. Zum Unterschied der beiden Varianten siehe [?].

Wir legen fest, dass der Wert 0 für links, 1 für mittig und 2 für rechts stehen soll. Damit ergibt sich für die drei Auswahloptionen zur Definition der Absenderposition:

```
\DeclareOption{leftfrom}{\renewcommand*\@alignfrom{0}}
\DeclareOption{centerfrom}{\renewcommand*\@alignfrom{1}}
\DeclareOption{rightfrom}{\renewcommand*\@alignfrom{2}}
```

Für einen Geschäftsbrief mit zentriertem Absender müssten wir also nur noch `\usepackage[centerfrom]{briefkopf}` schreiben.

### Anforderung 3: Linie im Absender

Des Öfteren findet man in Briefköpfen eine Linie im oder unter dem Absender. Diese Linie dient dazu, den Absender stärker hervorzuheben und ist außerdem mehr oder weniger netter Zierrat. Selten befindet sich die Linie unter dem gesamten Absender. Häufiger findet man die Variante, bei der diese Linie unter dem Absendernamen steht. Zusammen mit dem Verzicht auf die Linie haben wir also wieder drei Alternativen. Wie bei der Position des Absenders verwenden wir ein Makro mit einem numerischen Wert für die Auswahl. 0 steht dabei für den Verzicht auf die Linie, 1 für eine Linie direkt unter dem Absendernamen und 2 für eine Linie unter dem Absender. In der Grundeinstellung soll auf die Linie verzichtet werden:

```
\newcommand*\@rulefrom{0}
\DeclareOption{nofromrule}{\renewcommand*\@rulefrom{0}}
\DeclareOption{fromnamerule}{\renewcommand*\@rulefrom{1}}
\DeclareOption{fromrule}{\renewcommand*\@rulefrom{2}}
```

### Anforderung 4: Optionale Absenderangaben

Verschiedene Angaben will man in manchen Briefen haben, in anderen nicht. So sieht es in einem Brief an Tante Trude aus Buxtehude vielleicht nett aus, wenn man eine URL angibt, etwas anfangen kann sie damit jedoch nicht. E-Mail hat die Tante aber auch, und es schadet nicht, wenn man ihr auf dem

Wege auch die E-Mail-Adresse mitteilt. Bei einem Brief an die Firma „SPAM everywhere“ will man die E-Mail-Adresse aber vielleicht lieber verschweigen, ebenso wie die Telefonnummer bei einem Brief an „TUUB – Telefonmarketing unverlangt und unerwünscht aber beharrlich“.

Da es bei den entsprechenden Optionen nur zwei Möglichkeiten gibt, nämlich entweder die Angabe ausgeben oder unterdrücken, kann wieder auf Schalter mit `\newif` zurückgegriffen werden. Die Optionen werden einfach und konsistent *Option* und *noOption* genannt:

```
\newif\if@phone
\DeclareOption{phone}{\@phonetrue}
\DeclareOption{nophone}{\@phonefalse}
\newif\if@fax
\DeclareOption{fax}{\@faxtrue}
\DeclareOption{nofax}{\@faxfalse}
\newif\if@email
\DeclareOption{email}{\@emailtrue}
\DeclareOption{noemail}{\@emailfalse}
\newif\if@www
\DeclareOption{www}{\@wwwtrue}
\DeclareOption{nowww}{\@wwwfalse}
\DeclareOption{url}{\@wwwtrue}
\DeclareOption{nourl}{\@wwwfalse}
```

Eine Besonderheit ist bei den Optionen `www` und `url` beziehungsweise `nowww` und `nourl` zu finden. Diese haben paarweise die gleiche Bedeutung. Dies wurde so definiert, weil ich mir nie merken kann, ob der Name der Option nun `www` oder `url` ist.

#### Anforderung 5: Bankverbindung

Bei der Bankverbindung wird wieder ein Drei-Wege-Schalter benötigt. Die Bankverbindung soll entweder weggelassen (Wert 0), im Seitenfuß (Wert 1) oder neben der Adresse (Wert 2) gesetzt werden. Voreingestellt wird der Verzicht auf die Bankverbindung.

```
\newcommand*{\@bankat}{0}
\DeclareOption{nobank}{\renewcommand*{\@bankat}{0}}
\DeclareOption{bankatfoot}{\renewcommand*{\@bankat}{1}}
\DeclareOption{bankatlocation}{\renewcommand*{\@bankat}{2}}
```

## Anforderung 6: Logo im Briefkopf

Ob das Logo gesetzt wird oder nicht, stellt wieder nur eine Auswahl zwischen zwei Möglichkeiten dar:

```
\newif\if@logo
\DeclareOption{logo}{\@logotrue}
\DeclareOption{nologo}{\@logofalse}
```

## Anforderung 7: Faltmarken

Bei den Falt- oder Falzmarken ergibt sich eine Besonderheit. Die KOMA-Script-Klasse `scrlettr.cls` bietet eine solche Auswahl bereits an. Einschaltet werden die Marken dort mit `\foldmarkson`, während sie mit `\foldmarksoff` unterdrückt werden. Wir wollen die Namen der Optionen aber auch in dem Fall konsistent zu den oben definierten Namen der optionalen Angaben wählen. Innerhalb der Optionen werden aber die KOMA-Script-Makros verwendet:

```
\DeclareOption{foldmarks}{\foldmarkson}
\DeclareOption{nofoldmarks}{\foldmarksoff}
```

## Anforderung 8: Adressfeld

Genau wie bei den Faltmarken existiert auch für das Ein- und Ausschalten des Adressfeldes bei KOMA-Script bereits jeweils ein Makro, auf das hier zurückgegriffen wird.

```
\DeclareOption{addrfield}{\addrfieldon}
\DeclareOption{noaddrfield}{\addrfieldoff}
```

## Anforderung 9: Rücksendeadresse

Für die Rücksendeadresse gibt es bei KOMA-Script ebenfalls einen Mechanismus. Dieser funktioniert jedoch nicht über einen Schalter, sondern über den Inhalt eines Makros, das genau die Rücksendeadresse angibt. Dies hat für die Definition der Option zunächst keine Bedeutung. Hier kann ein einfacher `\newif`-Schalter verwendet werden.

```
\newif\if@backaddr
\DeclareOption{backaddr}{\@backaddrtrue}
```

Als Problem ergibt sich dann allerdings, dass KOMA-Script für das Setzen der Rücksendeadresse bereits die Anweisung `\backaddress` bietet, die als einziges Argument eben die Rücksendeadresse erwartet. Genau dieses Argument wird dann in einem internen Makro gespeichert. Hat das interne Makro einen Inhalt, so wird die Rücksendeadresse gesetzt, ansonsten wird darauf verzichtet. In dem Briefkopf-Paket sollen aber alle Absenderangaben, also auch die Rücksendeadresse, über eine per Option ladbare Absenderdatei eingestellt werden können. Gleichzeitig soll die Entscheidung, ob die Rücksendeadresse im Brief erscheint, per Option getroffen werden können. Natürlich könnte man in der Absenderdatei eine Abfrage auf den Schalter einbauen. Das ist aber nicht sehr komfortabel. Deshalb sorgen wir bei Auswahl der Option `nobackaddr` dafür, dass die Definition einer Rücksendeadresse mit Hilfe von `\backaddress` wirkungslos bleibt:

```
\DeclareOption{nobackaddr}{\@backaddrfalse
\renewcommand{\backaddress}[1]{}}
```

Bei der Umdefinierung muss die Anweisung weiterhin ein Argument erwarten, das dann eben ignoriert wird.

#### Anforderung 10: Absenderdatei

Die Option der Absenderdatei weicht von allen bisherigen Optionen ab. Dazu spezifizieren wir die Absenderdateien zunächst etwas genauer. Eine Absenderdatei soll nichts weiter als eine  $\LaTeX$ -Datei sein, in der über einfache Makros Absenderangaben gesetzt werden. So soll beispielsweise mit der KOMA-Script-Anweisung `\backaddress` die Rücksendeadresse für den Absender gesetzt werden. Jeder unterschiedliche Absender verfügt also über seine eigene Absenderdatei. Bei allen bisherigen Optionen sorgte ein fester Optionennamen für die Ausführung einer festen Aktion. Bei dieser Option soll an Hand des Optionennamens eine Datei eingelesen werden. Es wird also eine Option ausgeführt, deren Namen zum Zeitpunkt der Implementierung noch unbekannt ist. Solche Optionen beherrscht  $\LaTeX$  mit Hilfe der Stern-Variante von `\DeclareOption`. Diese wird immer dann verwendet, wenn eine unbekannte Option explizit an ein Paket übergeben wird. Für globale Optionen, das sind alle Klassenoptionen, wird sie jedoch nicht ausgeführt. Den Namen der Option findet man dann im Makro `\CurrentOption`.

Nun wollen wir natürlich nicht, dass einfach stur die Datei „`\CurrentOption`“ gelesen wird. Wir hängen dem Namen noch die Extension „.from“ an und prüfen, ob die Datei überhaupt existiert:



```
\DeclareOption*{\IfFileExists{\CurrentOption.from}}
```

Existiert die Datei, so wird sie nicht unmittelbar eingelesen, sondern ein internes Makro wird zur Speicherung ihres Namens definiert. Dies aber auch nur, falls das Makro nicht bereits existiert.

```
{\ifundefined{@fromfile}{%
  \edef\@fromfile{\CurrentOption.from}}
```

An dieser Stelle muss auf `\edef` statt auf `\def` oder `\newcommand` zurückgegriffen werden. Anderenfalls würde `\@fromfile` später zu genau der Definition expandieren, die `\CurrentOption` dann hätte, zuzüglich der Extension „.from“. `\edef` expandiert hingegen bereits bei der Definition, schreibt also den aktuell gültigen Wert von `\CurrentOption` nach `\@fromfile`.

Ist `\@fromfile` bereits definiert, wurde also bereits eine Absenderdatei ausgewählt, wird ein Fehler ausgegeben:

```
{\PackageError{briefkopf}{Double use of from-file-option}
  {You've used two from-file-options at
   \protect\usepackage,
   \MessageBreak but only one is allowed!}}
```

Ebenso wird ein Fehler gemeldet, wenn keine entsprechende Absenderdatei existiert. Dann handelte es sich schlicht um eine unbekannte Option:

```
{\PackageError{briefkopf}
  {Unknown option \CurrentOption}
  {You've used an unknown option.\MessageBreak
   Read the documentation for further
   information.}}
```

Es stellt sich möglicherweise die Frage, warum die Absenderdatei nicht bereits hier gelesen wird. Der Grund ist einfach und sollte sofort einleuchten: Innerhalb der Absenderdatei sollen mit einfachen Anweisungen in Form von Makros Absenderangaben gesetzt werden können. Diese Makros sind im Optionenabschnitt des Pakets aber noch gar nicht definiert, sondern werden erst später festgelegt.

## Anforderung 11: Grundeinstellungen

Bisher wurden also die Optionen definiert. Dabei wurde jeder Option bereits eine Grundeinstellung mitgegeben beziehungsweise die Grundeinstellung der

KOMA-Script-Klasse übernommen. Wir haben aber bei den Anforderungen festgelegt, dass Grundeinstellungen auch aus einer Konfigurationsdatei gelesen werden sollen. Diese Datei soll „briefkopf.cfg“ heißen. In dieser Datei wird mindestens eine Angabe `\ExecuteOptions{...}` erwartet. Im Argument könnten dann die Standardoptionen aus den obigen Optionen ausgewählt werden. Eine Absenderdatei sollte hier jedoch nicht gewählt werden.

Die Datei „briefkopf.cfg“ wird ebenfalls nur eingelesen, wenn sie auch existiert. Existiert sie nicht, werden als Standard keine Optionen ausgeführt, sondern die Standardeinstellungen des Pakets beibehalten.

```
\InputIfFileExists{briefkopf.cfg}
  {\typeout{Using briefkopf.cfg}}
  {\ExecuteOptions{}}
```

### Ausführen der Optionen

Zum Abschluss des Optionenabschnitts werden die Optionen, die als Standard aus der Konfigurationsdatei gelesen wurden, und die Optionen, die der Anwender per `\usepackage[...]{briefkopf}` festlegt, tatsächlich ausgeführt:

```
\ProcessOptions\relax
```

Das `\relax` am Ende ist vom L<sup>A</sup>T<sub>E</sub>X3-Team in [?] explizit dokumentiert. Paketautoren sollten deshalb darauf achten, es nicht zu vergessen.

### Das Setzen der Absenderangaben

Das Setzen der Absenderangaben ist weitgehend mit Makros möglich, die bereits von KOMA-Script zur Verfügung gestellt werden. Soll eine andere Briefklasse verwendet werden, müssen hier gegebenenfalls weitere Definitionen ergänzt werden.

#### Anforderung 12: Absenderangaben

In `scrletter.cls` fehlen Makros zum Setzen des Logos, der Faxnummer, der E-Mail-Adresse, der Homepage und der Bankverbindung. Für andere Felder, beispielsweise der Telefonnummer, sind Makros vorhanden und können im `scrguide` nachgelesen werden. Genau wie bei KOMA-Script definieren wir für das Setzen der Angaben einfache Anweisungen, die als Argument die jeweilige

Angabe erwarten. Diese wird dann in anderen Makros mit dem Präfix „from“ gespeichert.

```
\newcommand*{\logo}[1]{\def\fromlogo{#1}}
\newcommand*{\fax}[1]{\def\fromfax{#1}}
\newcommand*{\email}[1]{\def\fromemail{#1}}
\newcommand*{\www}[1]{\def\fromwww{#1}}
\newcommand*{\bank}[1]{\def\frombank{#1}}
```

Zur geschickten Vordefinition der `\from`-Makros kommen wir später.

## Bezeichnernamen

Bezeichnernamen wie „Betreff“ sind in KOMA-Script in Makros gespeichert, die in Abhängigkeit von der Sprache andere Werte annehmen können. Für einige Begriffe fehlen solche Variablen jedoch. Um den Rahmen des Artikels nicht zu sprengen, wird hier auf die Definition dieser fehlenden Bezeichner für mehrere Sprachen einschließlich der kniffligen Auswahl der richtigen Sprachdefinition verzichtet. Wie man so etwas machen kann, ist der dokumentierten Briefklasse `scrletter.dtx` aus dem KOMA-Script-Paket zu entnehmen. Das Briefkopf-Paket wird lediglich auf die Mehrsprachigkeit vorbereitet. Hierzu werden die benötigten Bezeichner in genau einer Sprache – Deutsch – definiert:

```
\providecommand*{\phonename}{Telefon}
\providecommand*{\faxname}{Fax}
\providecommand*{\emailname}{EMail}
\providecommand*{\wwwname}{WWW}
\providecommand*{\bankname}{Bankverbindung}
```

Da hier `\providecommand` an Stelle von `\newcommand` verwendet wird, werden eventuell bereits vorhandene Definitionen nicht verändert. Man kann also entsprechende Werte beispielsweise in der Konfigurationsdatei auch anders vorbelegen oder dort sogar die komplette Logik für Mehrsprachigkeit unterbringen.

## Variabilität in der Schriftauswahl

Variabilität in der Schriftauswahl wird oftmals gewünscht und ist bei KOMA-Script an vielen Stellen zu finden. Für eine Briefkopfdati liegt es nahe, die Schriftart für den Absendernamen, die restliche Absenderadresse und den

Seitenfuß einstellen zu können. Dies geschieht analog zu den Schriftvariablen in KOMA-Script mit Hilfe von Makros. Auch hier wird für die Definition `\providecommand` verwendet, um andere Voreinstellungen in der Konfigurationsdatei zu ermöglichen.

```
\providecommand*{\fnamefont}{\normalfont\scshape}
\providecommand*{\faddrfont}{\normalfont}
\providecommand*{\footfont}{\normalfont\sffamily}
```

Die Definition von `\fnamefont` zu SMALL CAPITALS geschieht hier weniger als Empfehlung, denn zur Demonstration.

## Die Layoutdefinition

Die Layoutdefinition findet für Privatbriefe und Geschäftsbriefe getrennt statt. Die Auswahl erfolgt über den bei den Optionen definierten Schalter.

```
\if@privat
```

Dieser führt uns zuerst in den Bereich für private Korrespondenz.

KOMA-Script stellt für die Definition des Briefkopfs die Anweisung `\firsthead` zur Verfügung, die als Argument den kompletten Briefkopf erwartet. Natürlich könnten alle Schalter innerhalb einer einzigen universellen Definition abgearbeitet werden. Die Makrodefinition wird jedoch übersichtlicher, kürzer und damit effizienter, wenn zumindest die Entscheidung über die Position des Absenders vorgezogen wird. Wir definieren also in Abhängigkeit von der gewünschten Position andere Briefköpfe.

```
\ifcase\@alignfrom
```

Wir befinden uns nun in dem Bereich für `\@alignfrom=0`. Entsprechend unserer Festlegung wird daher als Erstes der linksbündige Absender definiert. Wir verwenden dafür eine *parbox*, in der der Text linksbündig angeordnet wird. Diese Box soll über die gesamte Breite gehen dürfen:

```
\firsthead{\parbox[c]{\textwidth}{\flushleft
```

Zuerst wird der Name gesetzt. Dabei soll, wie bereits erwähnt, eine eigene Schriftart für den Namen verwendet werden können. Um die Änderung der Schriftart lokal zu halten, wird mit „{“ eine eigene Gruppe geöffnet.

```
{\fnamefont\fromname%
```

Hat `\@rulefrom` den numerischen Wert 1, so soll direkt auf den Absender eine Linie folgen. Die Länge dieser Linie kann vom Anwender bestimmt oder automatisch gewählt werden. Wählt der Anwender sie selbst, muss er sie in eine Länge oder ein Makro mit dem Namen `\fromrulewidth` schreiben. Existiert ein solcher Name nicht, so wird die Länge automatisch bestimmt. Bei der automatischen Längenbestimmung wird dafür gesorgt, dass die Linie nicht bis in das Logo ragt, das eventuell rechtsbündig neben dem Absender steht:

```
{\ifnum\@rulefrom=1\[-.5\baselineskip]
  \ifundefined{fromrulewidth}{%
    \if@logo
      \settowidth{\@tempdima}{-\@fromlogo}
      \addtolength{\@tempdima}{\textwidth}
      \edef\fromrulewidth{\the\@tempdima}
    \else
      \let\fromrulewidth\textwidth
    \fi}{%
  \rule{\fromrulewidth}{.4pt}\fi}\}
```

In obiger Definition werden zwei Tricks verwendet. Da  $\text{\LaTeX}$  an Längen ebenso wie an Zählern sehr knapp ist, wird kein weiteres entsprechendes Register belegt. Stattdessen wird für die Berechnung die temporär verwendbare interne Länge `\@tempdima` des  $\text{\LaTeX}$ -Kernels verwendet und `\fromrulewidth` mit Hilfe von `\edef` zu dem expandierten Wert dieses Registers definiert. Wird kein Wert berechnet, erhält `\fromrulewidth` mit `\let` einfach die gleiche Bedeutung wie `\textwidth`, dem Register für die Breite des Textkörpers.

Als Nächstes wird die Absenderadresse einschließlich aller optionalen Felder gesetzt. Dabei werden optionale Felder nur gesetzt, wenn der entsprechende Schalter *true* ist. Die Adresse wird ebenfalls innerhalb einer Gruppe in einer eigenen Schriftart gesetzt.

```
{\faddrfont\fromaddress%
  \if@phone\\\phonename:~\telephonenumber\fi%
  \if@fax\\\faxname:~\fromfax\fi%
  \if@email\\\emailname:~\fromemail\fi%
  \if@www\\\wwwname:~\fromwww\fi%
```

Wie weiter oben erwähnt, kann auch an dieser Stelle eine Linie gesetzt werden. Wir gehen davon aus, dass die Linie automatisch nicht in das Logo

ragt, da es in diesem Fall über der Linie zu stehen hat. Die entsprechende Berechnung kann also entfallen.

```

\ifnum\@rulefrom=2\[-.5\baselineskip]
\@ifundefined{fromrulewidth}{%
\let\fromrulewidth\textwidth}{%
\rule{\fromrulewidth}{.4pt}\fi\}%
}%

```

Die „parbox“ ist damit ebenfalls beendet.

Soll das Logo gesetzt werden, wird es links neben dem rechten Rand ausgegeben. Da wir uns noch rechts neben der parbox befinden, von der die gesamte Textbreite eingenommen wird, sind wir bereits am rechten Rand und können das Logo einfach mit `\llap` nach links ausgeben:

```

\if@logo
\llap{\fromlogo}
\fi}

```

Nebenbei bemerkt verändert `\llap` nicht die horizontale Position. Die äquivalente Anweisung, um eine Box ohne horizontale Positionsänderung nach rechts auszugeben, lautet übrigens `\rlap`.

Damit ist die Definition des linksbündigen Absenders abgeschlossen. Es geht weiter mit dem zentrierten Absender. Dieser hat als Besonderheit, dass das Logo nicht in den Briefkopf, sondern in den Bereich rechts neben dem Empfänger gesetzt wird. Dies ist bei KOMA-Script sehr einfach mit Hilfe der `\location`-Anweisung möglich.

```

\or
\firsthead{\parbox[c]{\textwidth}{\centering
{\fnamefont\fromname%
{\ifnum\@rulefrom=1\[-.5\baselineskip]
\@ifundefined{fromrulewidth}{%
\let\fromrulewidth\textwidth}{%
\rule{\fromrulewidth}{.4pt}\fi\}%
{\faddrfont\fromaddress%
\if@phone\\\phonename:~\telephonenumber\fi%
\if@fax\\\faxname:~\faxnum\fi%
\if@email\\\emailname:~\fromemail\fi%
\if@www\\\wwwname:~\fromwww\fi%

```

```

\ifnum\@rulefrom=2\[-.5\baselineskip]
\rule{\textwidth}{.4pt}\fi\}%
}}
\if@logo\location{\fromlogo\par\vfill\null}\fi

```

Der rechtsbündige Absender ist eher unüblich, kann aber bei Bedarf beim linksbündigen abgeschaut werden. An dieser Stelle lassen wir ihn weg und geben stattdessen einen Fehler aus.

```

\or
\PackageError{briefkopf}
{‘rightfrom’ not provided yet}{Define it!}
\fi

```

In der Layoutdefinition fehlt nun nur noch die Ausgabe der Bankverbindung. Diese kann optional an zwei Orten erfolgen oder unterbleiben. Der einfachste Fall ist, wenn die Ausgabe unterbleibt:

```

\ifcase\@bankat
% Nothing to do

```

Für die Bankverbindung im Fuß bedienen wir uns des Makros `\firstfoot`, das ebenfalls von KOMA-Script zur Verfügung gestellt wird:

```

\or
\firstfoot{\parbox[b]{\textwidth}{%
\centering\footfont\fontsize{8}{9}\selectfont
\@ifundefined{footrulewidth}{%
\let\footrulewidth\textwidth}{%
\null\rule{\footrulewidth}{.4pt}}\relax
\def\{\ \textperiodcentered\ \ignorespaces}
\bankname: \frombank}}

```

Die Definition ist etwas trickreich. Wir gehen davon aus, dass als Trennzeichen zwischen den einzelnen Feldern der Bankverbindung – wie in Tabellen oder bei linksbündigem Satz üblich – „\“ verwendet wird. Normalerweise führt dies zu einem harten Zeilenumbruch. Für die Bankverbindung im Fuß soll aber stattdessen ein kleiner, zentrierter Punkt verwendet werden.

Wie aus obiger Definition hervorgeht, wird der Fuß mit einer Linie vom Textbereich abgetrennt. Will man diese unterdrücken, schreibt man beispielsweise in die Konfigurationsdatei `\newcommand{\footrulewidth}{0pt}`.

Für den letzten Fall, Bankverbindung rechts neben dem Empfänger, verwenden wir wieder die KOMA-Script-Anweisung `\location`. Dabei muss berücksichtigt werden, dass an dieser Stelle möglicherweise auch noch das Logo stehen soll.

```
\or
  \location{\if@logo\ifnum\@alignfrom=1\fromlogo\par
    \vspace{2\baselineskip}\fi\fi\flushleft
    {\hspace*{-1em}\bankname:\\\frombank}\vfill\null}
\fi
```

Damit ist die Layoutdefinition für Privatbriefe vollständig. Da es sich bei dem Paket um eine frühe Betaversion handelt, fehlt die Definition für Geschäftsbriefe noch. Bei Geschäftsbriefen wird deshalb auf die Standarddefinition der zugrundeliegenden Briefklasse zurückgegriffen. Damit dies nicht überraschend geschieht, wird zusätzlich eine entsprechende Warnung ausgegeben:

```
\else
  \PackageWarning{briefkopf}
  {There is no layout definition for\MessageBreak
  business letters.\MessageBreak
  We use class layout instead.\MessageBreak
  If you want something else try\MessageBreak
  option ‘‘private’’ or define your\MessageBreak
  own layout.}
\fi
%
```

Das muss bis auf weiteres für Geschäftsbriefe genügen.

## Höchste Zeit für die Absenderdatei

Höchste Zeit für die Absenderdatei, wird nun so mancher denken. Richtig. Irgendwann sollte sie tatsächlich gelesen werden, wenn denn eine vorgegeben wurde. Wir erinnern uns, dass nur in diesem Fall das Makro `\@fromfile` definiert ist. Das machen wir uns jetzt zu Nutze:

```
\@ifundefined{@fromfile}{-}{\input\@fromfile}
```



## Fehler kann man vermeiden

Fehler kann man vermeiden oder dafür sorgen, dass sie nicht ins Chaos führen, sondern der Anwender entsprechend informiert wird. Häufigster Fehler dürfte sein, dass per Option geforderte Felder schlicht nicht definiert sind. Durch geschickte Vorbelegung aller Felder können diese Fehler komfortabel an den Anwender gemeldet werden. Dies geschieht einfach, indem man eine einheitliche Fehlermeldung definiert und die Felder mit dieser Fehlermeldung vorbelegt. In unserem Fall ist es schöner, wenn die Fehlermeldung über einige Parameter außerdem an das jeweilige Feld angepasst wird:

```
\newcommand*{\brfkipf@opt@err}[3]{%
  \PackageError{briefkopf}
    {Missing \expandafter\string\csname #2\endcsname}
    {You've used option ‘‘#1’’ at \protect\usepackage\space
      without defining a #3.}}
```

Diese Fehlermeldung wird nun mehrmals verwendet, aber jeweils nur, wenn die entsprechenden Felder auch angefordert werden:

```
\if@backaddr
  \providecommand{\frombackaddress}{
    \brfkipf@opt@err{backaddress}{backaddress}{backaddress}}
\fi
\if@logo
  \providecommand{\fromlogo}{\brfkipf@opt@err{logo}{logo}%
    {logo}}
\fi
\if@phone
  \providecommand{\telephonenumber}{
    \brfkipf@opt@err{phone}{telephone}{phone-number}}
\fi
\if@fax
  \providecommand{\fromfax}{
    \brfkipf@opt@err{fax}{fax}{fax-number}}
\fi
\if@email
  \providecommand{\fromemail}{
    \brfkipf@opt@err{email}{email}{email-address}}
\fi
\if@www
```

```

\providecommand{\fromwww}{\brfkrpf@opt@err{www}{www}%
                                {WWW-URL}}
\fi
\ifnum\@bankat >0\relax
  \providecommand{\frombank}{
    \brfkrpf@opt@err{bankatfoot}' or '{bankatlocation}{bank}
    {bank}}
\fi

```

## Am Ende des Pakets

Am Ende des Pakets wird L<sup>A</sup>T<sub>E</sub>X und dem Leser deutlich mitgeteilt, dass weiter nichts einzulesen ist:

```

\endinput
% end of file ‘briefkopf.sty’

```

Viele ältere Pakete verzichten auf diese beiden Zeilen, da L<sup>A</sup>T<sub>E</sub>X normalerweise kein `\endinput` benötigt. Leider kann dann aber niemand erkennen, ob die Datei hier wirklich enden soll oder beispielsweise aufgrund eines Übertragungsfehlers unvollständig ist. Ich empfehle daher dringend, an dieser Stelle nicht unnötig sparsam zu sein.

## Was nun noch fehlt

Was nun noch fehlt, ist ein Beispiel für eine Absenderdatei und ein Beispielbrief, um fleißig testen zu können.

Eine Absenderdatei könnte beispielsweise so aussehen:

```

\ProvidesFile{dante.from}
  [2001/05/10 Absenderdatei f"ur briefkopf.sty]

\newcommand\ev{e.V\kern-.18em\@ifnextchar.{}{.}}
\newcommand\dante{DANTE \ev}
\newcommand\Dante{DANTE, Deutschsprachige
  Anwendervereinigung \TeX~\ev}

\name{\Dante}
\address{Postfach 10\,18\,40\
  69008 Heidelberg}

```

```

\backaddress{\dante,
  Postfach 10\,18\,40,
  69008 Heidelberg}
\place{Heidelberg,}
\logo{\textsf{
  {\fontsize{20.74}{24.88}\selectfont%
    \@ifundefined{scalebox}{dante}%
      {\scalebox{2}[2]{dante}}}%
  {\small\, \eV}}}}
\bank{Volksbank Rhein-Neckar\
  Bankleitzahl: 670\,900\,00\
  Konto-Nummer: 2\,310\,007}
\telephone{(0\,62\,21)~2\,97\,66}
\fax{(0\,62\,21)~16\,79\,06}
\email{dante@dante.de}

\endinput

```

Dies führt zusammen mit folgender Beispieldatei zu dem in der Abbildung auf Seite 36 gezeigten Ergebnis.

```

% brfkpfd.tex
\documentclass{scrletter}
\usepackage[dante,fromnamerule,privat,logo]{briefkopf}
\usepackage{graphics}
\usepackage{ngerman}

\begin{document}
\begin{letter}
  {Karlo Kater\
    Katzengasse 17\
    12345 Entenhausen}
\opening{Lieber Karlo,}
ich habe keine Ahnung, was ich dir schreiben soll.
\closing{Dein Donald}
\end{letter}
\end{document}

```

Abbildung 1: Ein kleiner Beispielbrief unter Verwendung des Briefkopfpakets

Um die Auswirkungen der einzelnen Optionen zu testen, sollten Sie in dem Beispiel beliebig Optionen hinzufügen oder löschen und dann neue  $\LaTeX$ -Durchläufe durchführen.

## Zum Schluss

Zum Schluss bleibt für mich die Hoffnung, dass dieser Artikel in mehrerer Hinsicht nützlich war. Zum einen könnte er die Lust auf das Erstellen eigener Pakete geweckt haben. Zum anderen könnte er gezeigt haben, dass es gar nicht so schwer ist, eigene Briefköpfe zu definieren. Als Drittes sollte das entstandene Paket direkt nutzbar sein. Nach dem abschließenden Hinweis, dass man natürlich ganz analog zu `briefkopf.sty` ein Paket für Faxe schreiben oder eine entsprechende Option in das Paket einbauen könnte, wünsche ich allen Lesern frohes Schaffen.

# Einführung in pdf $\TeX$

Thomas Feuerstack

Der folgende Artikel richtet sich an Interessierte und Neueinsteiger, die mit Hilfe von pdf $\TeX$  möglichst schnell PDF-Dokumente erstellen wollen, ohne sich vorab intensiv mit Hintergrundwissen auseinandersetzen zu müssen. Darüberhinaus sollen beim Erzeugen von PDF-Dokumenten häufig auftretende Probleme erörtert sowie Lösungswege aufgezeigt werden – natürlich ohne Anspruch auf Vollständigkeit.

## Vorab

Sieht man sich in den  $\TeX$ -spezifischen News-Gruppen oder Diskussionslisten um, so wird man schnell feststellen, dass sich ein großer Teil der gestellten Fragen um die Verknüpfung von  $\TeX$  mit dem Portable Document Format

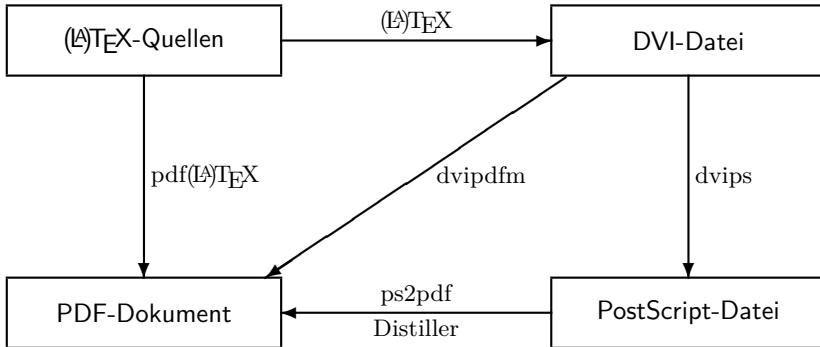


Abbildung 1: Wenn schon nicht alle, so führen doch zumindest mehrere Wege nach PDF.

(kurz: PDF) von Adobe dreht. Insbesondere Neueinsteiger scheinen in schöner Regelmäßigkeit an den gleichen Ecken und Kanten hängen zu bleiben – häufig ist das erste auftretende Problem, wie bzw. ob ein  $\text{\TeX}$ -Dokument nach PDF konvertiert werden kann. Wird diese Hürde erfolgreich genommen, lauern in aller Regel Schwierigkeiten bei der Einbindung von Grafiken und/oder der Schriftdarstellung.

Um es kurz zu machen: Die meisten dieser „Ecken und Kanten“ sind mit einfachen Mitteln locker zu meistern – sofern man denn weiß wie’s geht. Für diejenigen, die es (noch) nicht wissen, wurde dieser Artikel verfasst.

## Von $\text{\TeX}$ / $\text{\LaTeX}$ nach PDF

Bekanntermaßen führen alle Wege nach Rom, nichts ist unmöglich, und jeder ist seines Glückes Schmied. Es sollte daher nicht verwundern, dass es nicht nur einen, sondern – ein bisschen Kreativität vorausgesetzt – mindestens ein halbes Dutzend Möglichkeiten gibt, das eigene  $\text{\TeX}$ -File in Richtung PDF zu trimmen – als Übersicht mag Abbildung 1 dienen.

Prinzipiell können hier grob die beiden folgenden Richtungen unterschieden werden:

*Der klassische Weg:* Das Dokument wird wie gewohnt ge $\text{\TeX}$ t und mit Hilfe eines speziellen Treibers (beispielsweise `dvipdfm`) oder über den Umweg

Abbildung 2: Die Wirkung des `hyperref`-Pakets.

PostScript (`dvips` und im Anschluss `ps2pdf` oder Distiller) in PDF umgewandelt.

*Der elegante Weg:* pdfT<sub>E</sub>X erzeugt aus Ihrem Dokument das gewünschte PDF-Format – und das ohne weitere Umwege.

Da die zweite Alternative im Normalfall nicht nur schneller zu beschreiben, sondern in der Tat auch schneller umzusetzen ist, lassen wir daher nun Taten folgen.

## Zwölf Zeichen die Ihr Dokument „portabel“ machen

Fügen Sie die Anweisung `\pdfoutput=1` in die Präambel Ihres Dokuments ein, und Sie erhalten anstelle der gewohnten DVI-Datei die Ausgabe in PDF. Sollten Sie weder eine PDF- noch eine DVI-Datei vorfinden, dafür jedoch die Fehlermeldung „`Undefined control sequence...`“ am Bildschirm, so haben Sie (ebenfalls wie gewohnt) `latex` anstelle von `pdflatex` aufgerufen, dies sollten Sie ab sofort ändern.

Da wir schon mal dabei sind: Lassen Sie mich kurz auf den Unterschied zwischen pdfT<sub>E</sub>X und pdfL<sup>A</sup>T<sub>E</sub>X eingehen. In Analogie zum „klassischen“ T<sub>E</sub>X-System, bei dem auf der Anwendungsseite zwischen (Plain-)T<sub>E</sub>X und L<sup>A</sup>T<sub>E</sub>X unterschieden wird, existieren innerhalb von pdfT<sub>E</sub>X die Kommandos `pdftex` und `pdflatex`. Wenn in diesem Artikel jedoch fast ausschließlich von pdfT<sub>E</sub>X die Rede ist, so liegt das im Wesentlichen daran, dass heutzutage generell meistens L<sup>A</sup>T<sub>E</sub>X gemeint ist, wenn über T<sub>E</sub>X gesprochen wird.

Durch das Einfügen von `\pdfoutput=1` erzielen Sie zwar erst einmal den gewünschten Effekt. Den kompletten Satz spezifischer pdfT<sub>E</sub>X-Sequenzen finden Sie in [?]. Mit ein klein wenig Mehraufwand (in Form von 29 Zeichen) können Sie die Wirkung Ihrer PDF-Ausgabe jedoch unmittelbar steigern.

`\usepackage[pdftex]{hyperref}` als letztes Paket in die Präambel Ihrer T<sub>E</sub>X-Datei eingebunden bewirkt nicht nur die PDF-Ausgabe, die folgenden Funktionen gibts außerdem gratis dazu:

- Sämtliche Verzeichniseinträge, wie Inhaltsverzeichnis, Abbildungsverzeichnis, usw., werden automatisch mit dem Text verlinkt. Zur Verdeutlichung erhalten die anklickbaren Textteile einen farblich abgesetzten Rahmen.
- Auf Basis der Struktur des Inhaltsverzeichnisses wird eine Liste von Bookmark-Einträgen erzeugt, über die – ähnlich wie beim Inhaltsverzeichnis selbst – schnell auf die jeweiligen Textteile zugegriffen werden kann.
- Querverweise innerhalb des Textes wie Fußnoten, Indices, etc. werden mit dem Text verlinkt.

Laden Sie nun Ihr frisch erzeugtes PDF-Dokument in den Acrobat-Reader; es könnte bereits ein Aussehen wie in Abbildung 2 besitzen.

Natürlich kann `hyperref` noch viel mehr...

Über das gezielte Setzen `hyperref`-spezifischer Optionen kann das Aussehen Ihres PDF-Dokuments beinahe bis ins letzte Detail beeinflusst werden. Die Optionen werden, wie beim Einbinden von L<sup>A</sup>T<sub>E</sub>X-Paketen üblich, vor dem eigentlichen Paket-Namen in eckige Klammern gesetzt. Dabei unterscheiden wir zwischen einfachen Parametern, wie im oberen Beispiel der Parameter `pdftex`, und Parametern mit Wertzuweisung, beispielsweise `pdfstartview=Fit`. Eine typische Einbindung von `hyperref` könnte daher wie folgt aussehen.

```
\documentclass{article}
...
\usepackage[pdftex,a5paper,%
            pdftitle={Die Ducks - Psychogramm einer Sippe},%
            pdfauthor={Grobian Gans},%
            colorlinks=true,%
            linkcolor=blue%
            ]{hyperref}
...
\begin{document}
...
```

Der komplette Umfang aller verfügbaren Optionen befindet sich in der Dokumentation von Sebastian Rahtz [?], mit teilweise ausführlicherer Beschreibung auch in [?]. Die in den Manuals beschriebenen Möglichkeiten scheinen

dabei schier unendlich zu sein, auf der anderen Seite jedoch häufig genauso unverständlich. Aus diesem Grund werden wir einige häufiger gebrauchte Optionen näher betrachten.

## Generelle Optionen

Der „*Backend Driver*“, in unserem Beispiel `pdftex`, teilt dem `hyperref`-Paket mit, auf welchem Wege Sie Ihr PDF-Dokument erstellen wollen. Sollten Sie trotz dieses Artikels einen anderen Weg zur PDF-Erstellung wählen (siehe Abbildung 1), müssen Sie diese Option entsprechend ändern.

Da sich auch andere L<sup>A</sup>T<sub>E</sub>X-Pakete auf einen Backend Driver beziehen, beispielsweise das zu einem späteren Zeitpunkt noch benötigte Paket `graphicx`, können Sie ihn auch direkt in der `\documentclass`-Anweisung unterbringen – er gilt dann für alle Pakete, die ihn benötigen und muss nicht jedesmal explizit in der `\usepackage`-Anweisung aufgeführt werden.

Die *Papiergröße* ist standardmäßig auf `a4paper` festgelegt, kann jedoch durch `a5paper`, `b5paper`, `letterpaper`, `legalpaper` und `executivepaper` ersetzt werden.

Eine Sonderrolle spielen die beiden Optionen `draft` und `debug`. Während `draft` alle `hyperref`-Funktionen abschaltet, erweitert `debug` Ihr Log-File um zusätzliche Diagnose-Infos.

## Bildschirmanzeige und Dokument-Informationen

Normalerweise wird Ihr Dokument wie in Abbildung 2 durch den Acrobat-Reader dargestellt. Dies können Sie mit Hilfe der folgenden Optionen ändern:

`pdfpagemode` legt fest, wie der Acrobat-Reader geöffnet wird. Mögliche Zuweisungen werden im Folgenden beschrieben. `UseOutlines`, der Default, zeigt die Bookmarks im linken Fenster und den Text im rechten. `UseThumbs`, zeigt die einzelnen Seiten als Thumbnails im linken Fenster und den Text im rechten. `None` und `FullScreen`, zeigt das Dokument ohne Menüleisten, Bookmarks, Thumbnails oder generell linkem Fenster; es wird nur der Text angezeigt.

Der *Zoomfaktor* des Dokuments kann darüberhinaus wie folgt beeinflusst werden:



`pdfstartview` legt die Größe der zuerst angezeigten Seite fest. Normalerweise ist dies Seite 1; das jedoch kann wiederum über `pdfstartpage` geändert werden. Mögliche Werte sind `Fit`, für die komplette Seite, `FitH`, passt die Seitenbreite in den Fensterausschnitt ein, sowie `FitB`, was den Seiteninhalt in das Fenster packt.

`pdfview` regelt die Anzeigegröße einer Seite, die über einen Link aufgerufen wurde. Die Zuweisungswerte sind mit denen von `pdfstartview` identisch; als Defaultwert wird die Ansichtsgröße der vorherigen Seite übernommen.

*Zusätzliche Informationen* zum Dokument, die im Acrobat-Reader über das Menü `Datei – Dokumentinfo – Allgemein` abgerufen werden können, beinhalten im Normalfall den Dokumententitel, den Namen des Autors, sowie Schlüsselwörter, die gegebenenfalls für eine Suchmaschine von Belang sind. Diese Werte können Sie über die Optionen `pdftitle`, `pdfauthor`, `pdfkeywords`, ... besetzen – unsere Optionsliste wird nun langsam aber sicher umfangreicher.

```
\documentclass{article}
...
\usepackage[pdftex,a5paper,%
    pdftitle={Die Ducks - Psychogramm einer Sippe},%
    pdfauthor={Grobian Gans},%
    pdfkeywords={Enten Gaense Wasservogel},%
    pdfpagemode=FullScreen,%
    pdfstartview=FitB%
]{hyperref}
...
\begin{document}
...
```

## Beeinflussung der Bookmarks

Wie Ihnen vielleicht in Abbildung 2 aufgefallen ist, erzeugt das `hyperref`-Paket aus Ihren Abschnittsüberschriften Lesezeichen, auf englisch `Bookmarks` genannt. Der Acrobat Reader stellt `Bookmarks` in einem eigenen Fensterteil dar (sie sind damit ständig sichtbar), und durch Mausklick auf ein `Bookmark` können Sie direkt an die gewünschte Stelle innerhalb des Dokuments springen.

Die voreingestellte Erzeugung der `Bookmarks` kann über `bookmarks=false` abgestellt werden, was wiederum bedeutet, wie Sie im letzten Kapitel gelernt haben, dass die Option `pdfpagemode` den Wert `None` erhält.

Im Normalfall werden nur die Bookmark-Einträge der oberen Hierarchie dargestellt, rangtiefere Bookmarks müssen manuell „ausgeklappt“ werden. Dies lässt sich durch die Option `bookmarksopen=true` ändern, die wiederum den kompletten Bookmark-Baum entfaltet.

Sollten die Bookmarks dadurch zu unübersichtlich werden, so lässt sich mit Hilfe von `bookmarksopenlevel` in aller Regel ein vernünftiger Kompromiss finden.

## Farben

Farben treten, wie schnell zu erkennen ist, mit den durch `hyperref` erzeugten Links auf. Je nach Art des Verweises erhält der zugehörige Link eine andere Farbe – so wird beispielsweise Rot für interne Links benutzt (wie beim Inhaltsverzeichnis), Cyan für Links in das World Wide Web oder auf E-Mail-Adressen, Grün für Links in die Literaturliste und so weiter und so fort.

Mit Hilfe der Optionen `citebordercolor` (Literatur), `linkbordercolor` („normale“ Links), `urlbordercolor` (World Wide Web), etc. lassen sich die Default-Farbwerte überschreiben. Der gewünschte Farbwert wird in RGB-Notation über drei numerische Werte angegeben, die jeweils zwischen 0 und 1 liegen müssen. Beispiel: Durch `linkbordercolor={0 0 1}` wird das Standard-Rot der „normalen“ Links in Blau geändert.

Breite und Stil der Linkbox können durch die Option `pdfborder` beeinflusst werden, die auch ein numerisches Dreiertupel erwartet. Mit der Option `pdfborder={0 0 5}` würde die standardmäßige Strichstärke von einem Punkt auf fünf Punkte erhöht – mit dem Ergebnis, dass von dem umgebenen Text natürlich kaum noch etwas zu erkennen ist.

Besonders interessante Effekte können durch Hinzufügen eines weiteren optionalen Parameters zum Dreiertupel erzielt werden. `pdfborder={0 0 1 [3]}` erzeugt einen Kasten mit einer Strichstärke von einem Punkt, wobei die Umrandung jedoch unterbrochen ist – je höher der im optionalen Parameter gewählte Wert, desto länger werden die einzelnen Linienstücke.

`colorlinks=true` sorgt schließlich dafür, dass die Boxen ganz entfallen (zur Selbstkontrolle: `pdfborder={0 0 0}`) – dafür werden jetzt die zu verlinkenden Texte eingefärbt. Dabei bleibt das für die Boxen geltende Farbschema erhalten, die Farben können durch die Optionen `linkcolor`, `citecolor`, `urlcolor`, etc. geändert werden.

Angenehm: Bei Farbänderungen, die sich auf den Text auswirken, können die Farbnamen des Paketes `color` benutzt werden, oder im Klartext: Eine Anweisung wie `linkcolor=blue` funktioniert ohne Probleme.

Nicht so schön: Beim Ausdruck werden die eingefärbten Textteile selbstverständlich auch farbig gedruckt. Das heißt, Inhaltsverzeichnisse erscheinen in Rot, bzw. auf einem Schwarzweißdrucker in Grau.

### Dokumentübergreifende Voreinstellungen

Voreinstellungen, die für mehrere Dokumente einheitlich sein sollen, müssen nicht jedesmal in der `\usepackage`-Anweisung auftauchen, sie können stattdessen in einer systemweit bekannten Datei `hyperref.cfg` abgelegt werden.

Sollen beispielsweise für all unsere Dokumente `colorlinks` verwendet und die „normale“ Linkfarbe auf blau gesetzt werden, so lässt sich das wie folgt realisieren.

```
%% hyperref.cfg: Allgemein geltende Voreinstellungen
\hypersetup{colorlinks=true,%
            linkcolor=blue,%
            pdfproducer={Federvieh-Interessensverband},%
            pdfstartview=FitB}
```

Die Optionsliste des `hyperref`-Pakets kann nun auf die Angaben reduziert werden, die ausschließlich für das Dokument gelten.

```
\documentclass{article}
...
\usepackage[pdftex,a5paper,%
            pdftitle={Die Ducks - Psychogramm einer Sippe},%
            pdfauthor={Grobian Gans},%
            pdfkeywords={Enten Gaense Wasservoegel}%
            ]{hyperref}
...
\begin{document}
...
```

Optionen, die in der `\usepackage`-Anweisung gesetzt werden, überschreiben gleichnamige Wertzuweisungen aus der Datei `hyperref.cfg`.

## Eigene Links

Obwohl das `hyperref`-Paket für Sie bereits standardmäßig so gut wie alles verlinkt, was sich innerhalb eines T<sub>E</sub>X-Dokuments verknüpfen lässt, kann es tatsächlich vorkommen, dass Sie darüberhinaus noch eigene Verweise setzen möchten. Prinzipiell wird dabei zwischen Verweisen auf URLs (also auf WWW- oder E-Mail-Adressen) und dokumentinternen Links unterschieden.

*Links auf URLs* können sehr einfach durch den Befehl `\href` realisiert werden.

```
Ein neuer Morgen bricht in der Stadt
\href{http://www.entenhausen.com}{Entenhausen}
an, dem Wohnort der reichsten
\href{mailto:dagobert.duck@entenhausen.com}{Ente}
der Welt.
```

Im oberen Beispiel werden die Wörter **Entenhausen** und **Ente** je nach `colorlinks`-Einstellung farblich hervorgehoben. Durch Klick auf **Entenhausen** erreichen wir (wahrscheinlich) die Homepage der Stadt; über das Wort **Ente** hingegen können Sie Dagobert Duck eine E-Mail zukommen lassen – naturgemäß sind Bettelbriefe zwecklos!

Für *Querverweise innerhalb des Dokuments* benötigen Sie die beiden Anweisungen `\hypertarget` und `\hyperlink`. Dabei definieren Sie über `\hypertarget` ein Sprungziel (auch Anker genannt), welches Sie dann mit `\hyperlink` anvisieren können.

```
Zur Sicherung seiner finanziellen Interessen wohnt
Dagobert Duck seit Jahrzehnten in einem
\hypertarget{tresor}{Geldspeicher}.
```

```
...
"‘Los geht’s", riefen die Panzerknacker, "‘Heute ist der alte
\hyperlink{tresor}{Geizkragen}
dran!"’.
```

In diesem Beispiel wird das Wort **Geizkragen** anklickbar gemacht – durch Betätigung der Maustaste landen Sie direkt im **Geldspeicher**. Die Verbindung zwischen Link und Sprungziel wird dabei durch den frei wählbaren Namen `tresor` hergestellt.

## Einbindung von Grafiken

Vertraut man dem Volksmund, so sagt ein Bild bekanntlich mehr als tausend Worte – was wiederum manche Autoren zu der Annahme verleitet, die Qualität Ihrer Arbeit würde prozentual mit der Anzahl der darin enthaltenen Grafiken steigen.

Die komplizierteste Grafik, die pdfTeX-User in ihr Dokument einbinden wollen, ist in aller Regel die Erste – aber der Reihe nach. Sofern Sie in Ihre „normalen“ L<sup>A</sup>T<sub>E</sub>X-Dokumente bereits Grafiken eingebunden haben, sollten Sie dort im Idealfall auf folgende Konstrukte stoßen:

```
\documentclass{article}
```

```
...
```

```
\usepackage{graphicx}
```

```
...
```

```
\begin{document}
```

```
...
```

Wenn wir schon Beispiele aus der Comic-Welt benutzen, dann bietet sich auch die Einbindung von Bildern an.

```
\begin{figure}
```

```
\includegraphics{entenhausen.eps}
```

```
\end{figure}
```

```
...
```

Nachdem Sie Ihr Dokument durch Einfügen des `hyperref`-Pakets auf PDF-Kurs getrimmt haben, werden Sie mit höchst anzunehmender Wahrscheinlichkeit bereits beim ersten `\includegraphics`-Befehl [?] auf einen Fehler laufen. pdfTeX unterstützt direkt nämlich nur die Einbindung der Grafikformate JPEG, TIFF, PDF und PNG – jedoch kein EPS!

Ich habe an dieser Stelle einen neuen Absatz begonnen, damit Sie etwas mehr Zeit haben, diesen Schock zu verdauen. Da gibt man sich jahrelang alle erdenkliche Mühe, die benötigten Grafiken irgendwie nach Encapsulated PostScript zu bringen und dann diese Schlappe.

Um Ihre Geduld nicht überzustrapazieren, will ich mich an dieser Stelle nicht unnötig mit Erklärungen nach dem „Warum“ aufhalten – merken Sie sich einfach, dass sich pdfTeX mit PostScript-Konstrukten aller Art, dazu gehört ganz nebenbei auch das beliebte Paket `pstricks`, sehr schwer tut – und das ist eher freundlich formuliert.

Denken Sie besser positiv, und halten Sie sich zum Abschluss dieses Kapitels folgendes vor Augen:

- Sie müssen in Zukunft kaum noch Zeit mit der Konvertierung nach EPS zubringen, da die meisten Grafikprogramme mindestens eines der oben angegebenen Formate direkt unterstützen.
- Für den Fall, dass von bereits eingebundenen EPS-Bildern keine Kopien in einem der oben angegebenen Formate mehr vorliegen, hilft Ihnen in den meisten Fällen der Konverter `epstopdf` weiter, der eigentlich bei jeder T<sub>E</sub>X-Installation von Haus aus dabei sein sollte.

Bei Dokumenten, die sowohl mit T<sub>E</sub>X als auch mit pdfT<sub>E</sub>X bearbeitet werden sollen, hilft der folgende Trick. Erzeugen Sie die Graphik sowohl in einem Format für pdfT<sub>E</sub>X (beispielsweise `entenhausen.jpg`) als auch im EPS-Format (`entenhausen.eps`), und binden Sie die Graphik ohne explizite Dateikennung, also durch `\includegraphics{entenhausen}` ein. Egal ob Sie nun `latex` oder `pdflatex` starten, es wird immer das jeweils „richtige“ Bild eingefügt.

## Schriften und Schriftbilder

Endlich! Sie haben das `hyperref`-Paket eingebunden und nach Ihren Wünschen konfiguriert, alle Grafiken wurden fein säuberlich für pdfT<sub>E</sub>X konvertiert, und Ihr Dokument hat beim T<sub>E</sub>Xen keine Fehler mehr erzeugt – mit einem Wort: Sie sind nahe am göttlichen Ziel.

Häufig werden Sie dann doch noch auf der Zielgeraden abgefangen, zumindest enthüllt der Acrobat-Reader Grausiges: Das Schriftbild sieht aus, als wenn anstelle der üblichen T<sub>E</sub>X-Schrift der Font `Horror` benutzt worden wäre.

Sofern Sie nicht wirklich `Horror` als Grundschriftart verwendet haben (und daher mit dem Schriftbild eigentlich ganz zufrieden sind), liegt’s wahrscheinlich an Folgendem:

T<sub>E</sub>X verwendet im Normalfall sogenannte Punkt- oder Bitmap-Schriften (beispielsweise `cmr10.pk`), deren Bildschirmdarstellung dem Acrobat einige Schwierigkeiten bereitet. pdfT<sub>E</sub>X umgeht das automatisch, indem es anstelle der Bitmaps die Vektor- oder PostScript-Varianten der Computer Modern Schriften benutzt; Sie können dies gut gegen Ende des pdfT<sub>E</sub>X-Laufs an Meldungsfragmenten wie `...<cmr10.pfb><cmbx12.pfb>...` erkennen.

Um nun auf die angenagte Bildschirmdarstellung Ihres Dokuments zurückzukommen, so ist die Ursache im Normalfall in Ihrer Dokumenten-Präambel, genauer in der Anweisung

```
\usepackage[T1]{fontenc}
```

zu finden, mit der Sie eigentlich unter anderem Ihre Silbentrennung zu verbessern suchten – die dazu jedoch anstelle der Computer Modern die EC-Schriften einbindet, welche wiederum standardmäßig nicht als PostScript-Varianten in Ihrer TeX-Installation vorhanden sind. Ein sicheres Indiz für diese These liegt vor, wenn sich pdfTeX von Ihrem Dokument mit Meldungen wie `<...ecrm1000.pk>` verabschiedet.

Vor das Problem gestellt, sich nun zwischen einer verbesserten Silbentrennung oder der schöneren Bildschirmdarstellung entscheiden zu müssen, kann die Antwort natürlich nur „Beides!“ lauten. Die folgenden Möglichkeiten stehen zur Auswahl:

- Sie benutzen anstelle der gewohnten TeX-Schrift eine PostScript-Standardschrift (Times, Helvetica), beispielsweise durch Verwendung des Paketes `times`.
- Sofern Sie (wie ich) an Ihrem TeX-typischen Layout hängen, investieren Sie \$150,- (zzgl. Versand) und bestellen bei Y&Y die European-Modern-Schriften.
- Sofern Ihnen das (wie einer in diesem Artikel häufig erwähnten Comic-Figur) zu teuer erscheint, binden Sie das (kostenlose) Paket `ae` in Ihr Dokument ein. Durch `ae` wird das modernere Font-Encoding benutzt, die dargestellten Zeichen werden jedoch den Computer-Modern-PostScript-Fonts entnommen.

## Weitere PostScript-Schriften

Ist die Verwendung der Standard-PostScript-Fonts noch ein ziemlich problemloses Unterfangen, so kann die Einbindung weiterer, exotischer Schriften ein ziemlich arbeitsintensives Vergnügen werden – insbesondere, da viele Firmen und Institutionen, dank Corporate-Design, auf Frutiger, Futura, Garamond, etc. bestehen. Die Einbindung solcher, größtenteils kommerziell vertriebener, Type-1-Fonts ist sicher nicht trivial, aber da unter TeX bekanntlich kaum etwas unmöglich ist, können Sie sich ungefähr an folgendem Leitfaden orientieren.

- Sie haben eine PostScript-Schrift erworben und sollten daher auf dem Datenträger Dateien mit der Endung `.pfb`, `.afm`, und `.pfm` finden.
- Für die Verwendung mit pdfTeX benötigen Sie jetzt noch die zugehörigen Dateien mit den Endungen `.vf`, `.tfm`, `.map`, `.fd` und günstigerweise ein Style-File, um das Ganze als Paket in Ihr Dokument einbinden zu können. Diese Dateien sind im Regelfall *nicht* auf dem gekauften Datenträger vorhanden! Mit ein wenig Glück können Sie sie jedoch auf dem nächstgelegenen CTAN-Server im Verzeichnis `fonts/psfonts` finden.
- Werfen Sie einen Blick in den Verzeichnisbaum (TDS) Ihrer TeX-Installation, und versuchen Sie herauszufinden, wo andere Font-Pakete die Dateien mit den oben angegebenen Endungen untergebracht haben. Spielen Sie die Dateien der neuen Schriftart an den entsprechenden Stellen ein.
- Suchen Sie die Konfigurationsdatei `pdftex.cfg` und erweitern Sie sie um den Namen Ihrer `.map`-Datei. Für ein fiktives Map-File `neuer_font.map` würde der Eintrag `map +neuer_font.map` lauten.

Sofern Sie alles richtig gemacht haben, sollte die neue Schrift nun unter pdfTeX nutzbar sein. Sollten Sie darüberhinaus noch daran interessiert sein zu erfahren, was Sie da eigentlich gemacht haben beziehungsweise wofür die ganzen Dateien mit den seltsamen Endungen gut sind, habe ich abschließend noch einen Tipp für Sie. Besuchen Sie die nächste DANTE-Tagung und fragen Sie Walter Schmidt, oder besser noch: Hören Sie sich einen Vortrag von Walter Schmidt an.

## Fazit

Für mich bleibt an dieser Stelle eigentlich nur noch das Fazit, mit dem dieser überlange Artikel nun auch enden soll. Sofern Ihnen der Einstieg in pdfTeX noch immer sehr kompliziert erscheint: Geben Sie sich einen Ruck und fangen Sie endlich an.

Bei der Erstellung neuer Dokumente ist die direkte Verwendung von pdfTeX in aller Regel problemlos, da es außer der Einbindung des `hyperref`-Pakets erstmal keine zusätzlich benötigten Aktionen braucht. Anstelle der üblichen Previewer oder Druckertreiber kann der Acrobat-Reader ohne Schwierigkeiten für beide Aufgaben herangezogen werden.



Etwas komplizierter kann es dagegen sein, bestehende  $\text{T}_{\text{E}}\text{X}$ -Files für  $\text{pdfT}_{\text{E}}\text{X}$  umzustricken. Insbesondere für den Fall, dass viele (EPS-)Grafiken konvertiert werden müssen, ist es häufig effizienter, den traditionellen Weg zu wählen und Tools wie  $\text{pstopdf}$  zur PDF-Erzeugung zu nutzen.

## A package for typesetting Arabic mathematical formulas

Azzeddine Lazrek

$\text{T}_{\text{E}}\text{X}$  and its extension  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  have been adapted to handle passages of Arabic script. The package  $\text{ArabT}_{\text{E}}\text{X}$  extends the capabilities of  $\text{T}_{\text{E}}\text{X}$  and  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  to produce documents containing passages in Arabic. A package that extends  $\text{ArabT}_{\text{E}}\text{X}$  to typeset Arabic mathematical formulas, with specific symbols and running from right to left, is presented in this paper.

### Overview

Although  $\text{T}_{\text{E}}\text{X}$ , the powerful typesetting system by Donald E. Knuth [3], has been designed according to the conventions of English and Western languages, it is also able to handle passages of Arabic script [4]. Actually, systems such as  $\text{ArabT}_{\text{E}}\text{X}$ , by Klaus Lagally [5], allow generating documents with passages in Arabic or some other language using the Arabic script. However, many Arabic mathematical texts present mathematical formulas as they are in English or in French texts, and a large part of Arabic handbooks display mathematical formulas using original symbols in a writing running from right to left. Our system is designed for generating Arabic texts including such mathematical writing. The system consists of a small set of  $\text{T}_{\text{E}}\text{X}$  macro packages, two additional extension and symbol fonts, and  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  font definition files. It will run with Plain  $\text{T}_{\text{E}}\text{X}$ , and has been adapted for  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  [6]. Some problems are still to be solved. In particular, some compatibility questions are under investigation. This paper has been typeset with this package. Arabic mathematical document processing is discussed in [2], [7] and [1]. In the following, syntactic commands used in this system are presented. This paper is organized as follows: in section 2, we show how to load our package.

In section 3, we present some options. In section 4, commands that can be used to typeset Arabic mathematical formulas are listed. Some examples are given. There are still some open questions that are outside the scope of this paper.

Throughout the paper, we will talk about “Arabic mathematical” texts, documents, formulas . . . Of course, mathematics is unique, it has nothing to see with any national specificity. When we use this term, it refers only to the presentation of mathematics in the Arabic language.

## Preamble

In order to use Arabic mathematical formulas with  $\text{T}_{\text{E}}\text{X}$  or  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , the user may insert, in the preamble of his document, the command: `\input arabmath` when using  $\text{T}_{\text{E}}\text{X}$  or `\usepackage{arabmath}` when  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  is the processing system in use. Obviously, the package  $\text{ArabT}_{\text{E}}\text{X}$  has to be loaded first.

## Options

The commands defined in this system are prefixed with the initials of Arabic Mathematical “am”. This distinguishes the newly introduced commands from the basic commands of  $\text{T}_{\text{E}}\text{X}$ ,  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , or other packages. The rest of the command names derive from their corresponding  $\text{T}_{\text{E}}\text{X}$  commands. The following options are offered:

`\amarabmath` to begin an environment where Arabic mathematical formulas are generated (e.g.  $\bar{\alpha}$ ). This is the default.

`\amlatinmath` to begin an environment where mathematical formulas are in their Latin presentation (e.g.  $\sqrt{7}$ ).

`\amwarabnum` for generating the standard western Arabic numerals:  $\{9, 8, 7, 6, 5, 4, 3, 2, 1, 0\}$ . These numerals are used in North Africa. This is the default.

`\amoldstylenum` for generating the old style western Arabic numerals:  $\{9, 8, 7, 6, 5, 4, 3, 2, 1, 0\}$ .

`\amearabnum` for generating the eastern Arabic numerals:  $\{٩, ٨, ٧, ٦, ٥, ٤, ٣, ٢, ١, ٠\}$ . These numerals are used in Middle East.

These options may be specified in the preamble of the input file when they should effect the whole document. Or if they are specified at the beginning of an environment in mathematical mode, then the option effects only the part until the end of this environment.

## Commands

Generally, in Arabic writing, the order of the letters in a word as well as the order of the words in a sentence, runs from right to left. However, *numbers* are an important exception: The digits in a number run from left to right! Therefore, it is important that the order of the tokens to be printed can be inverted whenever necessary.

Our macro package adds new commands to the basic set of T<sub>E</sub>X commands. All these commands work in mathematical mode only, either in display style or in text style.

The commands are listed below. The resulting text will appear in the frame following the command.

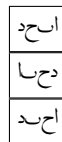
### Direction inversion

The command `\amrl{expr}` inverts the handwriting direction in order to generate an Arabic mathematical expression *expr* spreading out from right to left. Notice that the command `\amrl` does not change the direction of text portions given in additional braces:

`$$\amrl{abjd}$$`

`$$\amrl{{abjd}}$$`

`$$\amrl{a{bj}d}$$`



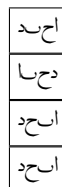
If the command `\amrl` is used in the argument of `\amrl` it and its argument has to be enclosed in braces.

`$$\amrl{a\amrl{bj}d}$$`

`$$\amrl{\amrl{abjd}}$$`

`$$\amrl{a{\amrl{bj}}d}$$`

`$$\amrl{{\amrl{abjd}}}$$`



In general, all commands with arguments that are used in an `\amrl` command, should be written within braces.

*Examples:*

```
\amrl{\hat b} , b{\amsp{17}} , {\amsqrt{s+3}}$
```

In order to invert the writing direction of a specific argument, use an additional `\amrl` for this argument. Because this is often annoying we propose a set of new commands where a necessary inversion is already included.

*Examples:*

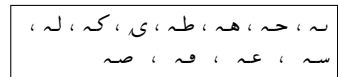
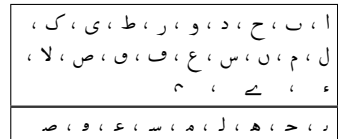
```
\amrl{\overleftarrow{\amrl{abj}}} , {\amoverleftarrow{abj}}$
```

The example shows that the user can now choose between two possibilities of input: Either use the traditional  $\TeX$  command and enclose its argument with an additional `\amrl` or use the new `\am...` commands.

## Alphabet

The letters used in the composition of Arabic mathematical symbols are of three shapes:

1. `\amrl{a, b, j, d, w, r, t, y, k, l, m, n, s, g, f, q, c, X, A, Y, N}$`
2. `\amrl{B, J, H, L, M, S, G, F, C}$`
3. `\amrl{\amb, \amj, \amh, \amt, \amy, \amk, \aml, \ams, \amg, \amf, \amc}$`

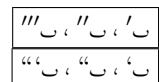


This coding scheme differs slightly from the basic one used in  $\text{Arab}\TeX$  to generate the Arabic writing. The used transliteration is not optimal. Since the package is intended for an Arab user, it would be better to get a direct coding scheme from the keyboard.

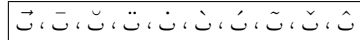
## Accents

The system offers the following accents:

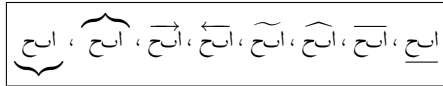
```
\amrl{b{'}}, b{''}, b{'''}}$  
\amrl{b' , b'' , b'''}$
```



`\amrl{\hat b}` , `\check b` , `\tilde b` , `\acute b` ,  
`\grave b` , `\dot b` , `\ddot b` , `\breve b` ,  
`\bar b` , `\vec b`}}



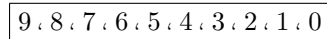
`\amrl{\amunderline{abj}}` , `\amoverline{abj}}` ,  
`\amwidehat{abj}}` , `\amwidetilde{abj}}` ,  
`\amoverleftarrow{abj}}` , `\amoverrightarrow{abj}}` ,  
`\amoverbrace{abj}}` , `\amunderbrace{abj}}`}}



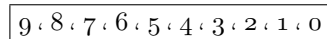
### Numbers

Numbers have to be enclosed with braces. Only if a number consists of a single digit, the braces can be omitted. The sign of the number has to be put outside the braces.

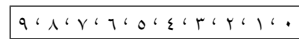
`\amrl{0,1,2,3,4,5,6,7,8,9}`



`\amoldstylenum{\amrl{0,1,2,3,4,5,6,7,8,9}}`



`\amearabnum{\amrl{0,1,2,3,4,5,6,7,8,9}}`



For `\amoldstylenum` and `\amearabnum` the expression should be given within braces.

Numbers can be given with or without a fractional part, separated either by a decimal comma or a decimal dot. The numbers can be prefixed by an optional sign. The first example shows the formatting according to the North Africa typographic conventions, the second according to those used in Middle East.

Examples:

`\amrl{7 , {5} , +{92} , -8 , {107} , {123.45} , {3{\amrlatinmath ,}14}}$`

3,14 : 12345 : 107 : 8- : 92+ : 5 : 7

`\amearabnum{\amrl{7 , {5} , +{92} , -8 , {107} , {123.45} , {3{\amrlatinmath ,}14}}}$`

٣,١٤ : ١٢٣٤٥ : ١٠٧ : ٨- : ٩٢+ : ٥ : ٧

## Punctuation

Arabic punctuation is available, too:

`\amrl{,;:;!}?$`

!,:;.

## Delimiters

Delimiters are obtained as follows:

`\amrl{( , | , [ , || , \{ \cdots \} , || , ] , | , )}$`

( : | : [ : || : { : \cdots : } : || : ] : | : )

`\amrl{\amrlbrace , \amrbrace}$`

{ }

## Symbols

The basic symbols:

`\amrl{+ , - , * , \times , \ = \ , \cdots}$`

⋯ : + : = : × : \* : - : +

`\amrl{/ , \% , \cdots}$`

⋯ : % : /

`\amrl{< , > ; \amin , \amni , \cdots}$`

⋯ : ∈ : ∃ : < : >

`\amrl{\amangle ; \amseqm , \amleqm ; \amleqv}$`

تعبر تعبر  
↔ : = : ≐ : ∇

## Superscript and subscript

The commands `\amsp{expr}` or `^expr` format *expr* as an exponent. *expr* should be given within braces unless it consists of a single token. The command `^` does not change the direction of *expr* and therefore can be used only for a single token.

Examples:

$\$ \backslash \text{amrl}\{\text{b}\{\backslash \text{amsp}\{\{17\}\}\}+5 ;$   
 $\text{b}\{\}\{\backslash \text{amsb}\{2}\}+5*\text{s}\{\{\}^{\text{b}}\}\}$$

$$\boxed{\text{س} * 5 + 2 \text{؛} 5 + 17 \text{س}}$$

The commands  $\backslash \text{amsb}\{expr\}$  or  $\_expr$  format  $expr$  as an index.  $expr$  should be within braces unless it consists of a single token. The command  $\_$  does not change the direction of  $expr$  and therefore can be used only for a single token.

Examples:

$\$ \backslash \text{amrl}\{\text{b}\{\backslash \text{amsb}\{\{17\}\}\}+5 ;$   
 $\text{b}\{\}\{\backslash \text{amsb}\{2}\}+5*\text{s}\{\{\}_b\}\}$$

$$\boxed{\text{س} * 5 + 2 \text{س} ؛ 5 + 17 \text{س}}$$

The empty braces  $\{\}$  are necessary to get the exponent or the index closer to the basic symbol.

## Fraction

The command  $\backslash \text{amfrac}\{expr1\}\{expr2\}$  gives a fraction with  $expr1$  as numerator and  $expr2$  as denominator.

Examples:

$\$ \backslash \text{amrl}\{\{\backslash \text{amfrac}\{1\}\{2\}\} ;$   
 $\{\backslash \text{amfrac}\{2*s\}\{b\}\} \}$$

$$\boxed{\frac{\text{س} * 2}{\text{س}} ؛ \frac{1}{2}}$$

## Functions

### Common functions

There are symbols for usual abbreviations representing elementary functions in use in mathematics. These function names based on typographical conventions are predefined:

Logarithm  $\backslash \text{amlg}$  (لو), exponent  $\backslash \text{amexp}$  (قهم), sine  $\backslash \text{amsin}$  (جا), Cosine  $\backslash \text{amcos}$  (جتا), Tangent  $\backslash \text{amtan}$  (ظا), Cotangent  $\backslash \text{amcot}$  (ظتا), Secant  $\backslash \text{amsec}$  (قا), Cosecant  $\backslash \text{amcsc}$  (قتا), Arc sine  $\backslash \text{amarcsin}$  (زجا), Arc cosine  $\backslash \text{amarccos}$  (زجتا), Arc tangent  $\backslash \text{amarctan}$  (زظا), Arc cotangent  $\backslash \text{amarccot}$  (زظتا), Arc secant  $\backslash \text{amarcsec}$  (زقا), Arc cosecant  $\backslash \text{amarccsc}$  (زقتا), Hyperbolic

sine `\amsinh` (جان), Hyperbolic cosine `\amcosh` (جتان), Hyperbolic tangent `\amtanh` (ظان), Hyperbolic cotangent `\amcoth` (ظتان), Hyperbolic secant `\amsech` (قان), Hyperbolic cosecant `\amcsch` (قتان), Arc hyperbolic sine `\amarcsinh` (زجان), Arc hyperbolic cosine `\amarccosh` (زجتان), Arc hyperbolic tangent `\amarctanh` (زظان), Arc hyperbolic cotangent `\amarccoath` (زظتان), Arc hyperbolic secant `\amarcsech` (زقان), Arc hyperbolic cosecant `\amarccsch` (زقتان).

### Limit

The command `\amlim{expr1\amtoexpr2}` gives the limit when *expr1* tends to *expr2*.

*Examples:*

```
\amrl{\amlim{s\amto 0}s^2} ;
\amrl{\amlim{s\amto +\infty} s^2}
```

$$\lim_{s \rightarrow 0} s^2 ; \lim_{s \rightarrow \infty} s^2$$

### Function defined with cases

The command `\amcases{array}` generates a function defined with different cases as shown in the following example.

*Examples:*

```
\amrl{d(s)=\amcases{
\amrl{-4s\&\hbox{<'i_dA kAn>}}\ s<0}\cr
\amrl{4s \&\hbox{<'i_dA kAn>}}\ s>0}\cr
\amrl{-2 \&\hbox{<.gyr _dlk>}}\cr}}}
```

$$d(s) = \left. \begin{array}{l} -4s \text{ إذا كان } s < 0 \\ 4s \text{ إذا كان } s > 0 \\ -2 \text{ غير ذلك} \end{array} \right\}$$

### New function

The command `\amnewfunc{fname}` defines a new function named *fname*.

*Examples:*

```
\amrl{\amnewfunc{.s.gr}}(s) =
\amcos(s\amsp 2)+\amsin(s)-6}
```

$$\text{صغر}(s) = \text{جتا}(s)^2 + \text{جا}(s) - 6$$



Square Root, Integral, Sum, Product

The command `\amsqrt{expr}` gives the square root of *expr*. The command `\amroot{expr1}\of{expr2}` gives the square *expr1* of *expr2*.

Examples:

`\amrl{\amsqrt{s{\}\ampsp 2}} ;`  
`{\amroot{3}\of{b*9}} }`

The command `\amlint\amsb{expr1}\ampsp{expr2}` gives the integral from *expr1* to *expr2* using the symbol  $\int$ .

Examples:

`\amrl{\amlint_1^t} s{\}^b\,A\,s}`

The command `\amsint\amsb{expr1}\ampsp{expr2}` gives the integral from *expr1* to *expr2* using the symbol  $\int$ .

Examples:

`\amrl{\amsint\limits_1^t} s{\}^b\,A\,s}`

The command `\amlsum{expr1}{expr2}` gives the sum from *expr1* to *expr2* with the conventional Arabic symbol  $\sum$ .

Examples:

`\amrl{\amlsum{b=1}{c}}s{\}^b ;`  
`{\amlsum{b=a-1}{c}}s{\}^b}`

The command `\amssum\amsb{expr1}\ampsp{expr2}` gives the sum from *expr1* to *expr2* with the inverted symbol  $\sum$ .

Examples:

`\amrl{\amssum}{\amsb{b=1}^c}\ s{\}^b ;`  
`{\amssum\limits\amsb{b=a-1}^c} s{\}^b}`

The same can be done with the symbol for product: `\amlprod{expr1}{expr2}` gives the product from *expr1* to *expr2* with the symbol  $\prod$ ,

Examples:

```
\amrl{\amprod{b=1}{c}s^b} ;
\amrl{b=a-1}{c}s^b}
```

$$\prod_{1=c}^b c ; \prod_{1=c}^b c$$

The command `\amsprod\amsb{expr1}\amsp{expr2}` gives the product from *expr1* to *expr2* with symbol  $\prod$ .

Examples:

```
\amrl{\amsprod}{\amsb{b=1}^c} s^b ;
\amsprod\limits\amsb{b=a-1}^c} s^b}
```

$$\prod_{1=c}^b c ; \prod_{1=c}^b c$$

Matrix

The command `\ammatrix{array}` generates a matrix.

Examples:

```
\amrl{m={\left|\ammatrix{
\amrl{\left|\ammatrix{
\amrl{1 & 2c}\cr \amrl{4a & 5}\cr
\right|}& 2c & 3 & 5}\cr
\amrl{4 & 9 & 4a & 5}\cr
\amrl{6 & g & 4a & 5}\cr}
\right|}}}
```

$$\left| \begin{array}{cc|cc} 5 & 3 & 2 & 1 \\ 5 & 4 & 9 & 4 \\ 5 & 4 & & 6 \end{array} \right| = 2$$

System of equations

The command `\amsystem{system of equations}` generates a system of equations with a big closing brace in the right, as in Latin notation.

Examples:

```
\amsystem{\amrl{2s - 4c + 6 &= 8}\cr
\amrl{5s - 7c + 9 &= 11}\cr \amrl{5s - 7c + 9t &= 11}\cr}
```

$$\left. \begin{array}{l} 8 = 6 + 4 - 2 \\ 11 = 9 + 7 - 5 \\ 11 = 9 + 7 - 5 \end{array} \right\}$$

Equation numbering

Different ways for numbering equations are offered:

$$\begin{aligned}
 & \text{\$}\amrl{2ms{\}^2} - 3s + 5 = 0\amleqno{3a}\text{\$}\$ \\
 & \qquad\qquad\qquad 0 = 5 + 3 - 2س2 \qquad\qquad\qquad (3)
 \end{aligned}$$

$$\begin{aligned}
 \text{or } & \text{\$}\amrl{2ms{\}^2} - 3s + 5 = 0\amreqno{3b}\text{\$}\$ \\
 (\_3) & \qquad\qquad\qquad 0 = 5 + 3 - 2س2
 \end{aligned}$$

### Size variation

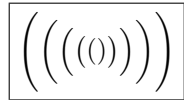
Delimiters as well as super- and subscripts vary in their size according to nesting.

Examples:

```

\amrl{\Big}{\bigg}{\Big}{\big}{(
){\big}{\Big}}{\bigg}}{\Big)}\text{\$}

```



```

\amrl{\amb{\ampsp{\amb{
\ampsp{\amb{\ampsp{\amb}}}
}}}}\text{\$}
\amrl{b{\amsb{b{\amsb{b{\amsb{b}}}
}}}}}\text{\$}

```

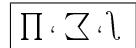


The size of the operator symbols vary in text and display math mode:

```

\amrl{\amsint , \amssum , \amsprod}\text{\$}

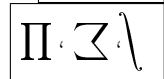
```



```

\amrl{\amsint , \amssum , \amsprod}\text{\$}\text{\$}

```



### Miscellaneous

The command `\mbox{string}` or `\hbox{string}` introduces the string *string* in a formula. `\amtime` gives the current time, `\amday` the current day, `\ammonth` the current month, and `\amyear` the current year.

To get a list of the western arabic month names, the command `\amlwm` can be used:

يناير، فبراير، مارس، أبريل، ماي، يونيه،  
 يوليو، غشت، شتمبر، أكتوبر، نونبر، دجنبر

`\amlem` lists the eastern Arabic months names:

كانون الثاني، شباط، آذار، نيسان، أيار، حزيران،  
تموز، آب، أيلول، تشرين الأول، تشرين الثاني، كانون الأول

`\amwtoday` gives the actual date consisting of the day, the western Arabic month name and the year. To give the eastern Arabic month names, `\amtoday` should be used, and `\annumtoday` shows a numerical month.

*Examples:*

`\amwtoday`

28 يناير 2007

`$$\amearabnum{\amtoday}$$`

٢٨ كانون الثاني ٢٠٠٧

`\annumtoday`

2007/1/28

`$$\amearabnum{\annumtoday}$$`

٢٠٠٧/١/٢٨

## Conclusion

This package could be adapted to different needs and situations. Styles can be designed according to the typographic context. The present transliteration is still hard to use. Many open questions arise about this issue. The user should be aware about the use of `\amr1` command and braces. The automatic management of spaces among terms doesn't work perfectly until now. All commands can be renamed. For instance, the command `\amsqrt` can be changed into `\jdr` in order to get a name closer to the Arabic pronunciation of the symbol. An adaptation of the system to L<sup>A</sup>T<sub>E</sub>X commands for Arabic mathematical documents is under development.

## References

- [1] عز الدين لزرق وخالد سامي، نظام مبتكر لمعالجة النصوص العربية  
ضيق العربية، المؤتمر السابع لتعريب العلوم، تعريب العلوم في منظومة  
التنمية القومية، مصر ٢٠٠١

(Azzeddine Lazrek and Khalid Sami, Towards a system for Arabic mathematical document processing, 7th congress of sciences Arabization, *Sciences Arabization in development system*, Egypt 2001).

- [2] عز الدين لزرق وخالد سامي، أفاق مع الملحمة النصوص الرياضي  
التي باللغة العربية، مؤتمر المعلوماتية والصناعة البرمجية ودورها المستقبلي،  
اتحاد مجالس البحث العلمي العربية وكلية الحدباء الجامعة، العراق ٢٠٠٠

(Azzeddine Lazrek and Khalid Sami, Arabic mathematical document processing perspectives, Congress *Computer science and software development and its future impact*, Federation of Arab scientific research councils and University AlHadba' Faculty, Iraq, 2000).

- [3] Donald E. Knuth: *The T<sub>E</sub>Xbook*; Addison-Wesley; 1996.
- [4] Donald E. Knuth und Pierre MacKay: *Mixing right-to-left texts with left-to-right texts*; *TUGboat*; 8(1), S. 14–25; 1987.
- [5] Klaus Lagally: *ArabT<sub>E</sub>X – Typesetting Arabic with Vowels and Ligatures*; *EuroT<sub>E</sub>X'92*; 1992.
- [6] Leslie Lamport: *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*; Addison-Wesley; 2. Aufl.; 1994.
- [7] Azzeddine Lazrek: *Aspects de la problématique de la confection d'une fonte pour les mathématiques arabes*; *Cahiers GUTenberg*; 39–40; 2001.

# Rezensionen

---

## „Handbook of Typography for the Mathematical Sciences“ von Steven G. Krantz

Armin Geisse

Mit seinem neuesten Buch möchte Steven G. Krantz Hilfestellung für Autoren in spe geben, die sich mit dem Gedanken tragen, ein mathematisches Werk zu veröffentlichen. In insgesamt sieben Kapiteln gibt Krantz eine Einführung in  $\LaTeX$ , er erläutert den Produktionsprozess eines Buches und streift die Grundlagen der Veröffentlichung im WWW. Trotz des aktuellen Erscheinungsdatums beschreibt das Buch in großen Teilen den Softwarestand vor 4–5 Jahren und wurde nur sehr unvollständig aktualisiert. Es leidet unter vielen Wiederholungen und einer schlecht strukturierten Aneinanderreihung der einzelnen Kapitel.

Steven G. Krantz hat ein neues Buch veröffentlicht, das den Leser „in die Lage versetzt, mit Verlegern, Lektoren, Schriftsetzern und Grafik-Designern sowie  $\TeX$  Consultants informiert zu verhandeln und bessere Bücher zu veröffentlichen“. So weit die vollmundige Werbung, passend zum vielversprechenden Titel, der mich letztendlich dazu verführt hat, das Buch ohne vorherige „Bersichtigung“ im Internet zu bestellen. Krantz schränkt diese Versprechungen jedoch schon im Vorwort ein, wo er vorsichtshalber darauf hinweist, dass es kein Buch wäre, um  $\TeX$  zu lernen. Es solle vielmehr dazu motivieren, aus anderen Quellen mehr über  $\TeX$  zu erfahren. Dies hindert ihn jedoch nicht daran, sich in fast allen Kapiteln mit  $\LaTeX$  zu befassen.

Im ersten Kapitel „Basic Principles“ erläutert der Autor im Schnelldurchlauf die Entstehung eines Buches, gibt erste stilistische Tipps und erwähnt  $\LaTeX$  als *Dialekt* von  $\TeX$ . Eine Bezeichnung, die etwas später auch noch für  $\mathcal{Z}\TeX$  und  $\mathcal{Z}\LaTeX$  Verwendung findet. Auffällig ist schon auf den ersten Seiten die Ausführlichkeit, mit der er (im Jahre 2001) auf  $\LaTeX$  2.09 eingeht. Selbst die Beispieldatei in Anhang VIII ist sowohl für  $\LaTeX$  2<sub>ε</sub> als auch für Version 2.09

abgedruckt. Das deutet schon am Anfang daraufhin, dass hier ein betagteres Manuskript notdürftig aktualisiert und veröffentlicht wurde. Ein Eindruck, der sich im Laufe der weiteren Lektüre leider bestätigte.

Kapitel 2 „Typesetting Mathematics“ beschreibt die Notation der mathematischen Symbole sowie die unterschiedlichen Darstellungen von Gleichungen. Hier wird auch lobend die Portabilität von  $\text{T}_{\text{E}}\text{X}$  zwischen den verschiedenen Betriebssystemen erwähnt. Dies scheint dem Autor so wichtig zu sein, dass er denselben Satz fast wortwörtlich zwei Seiten später noch einmal wiederholt. Diese Wiederholung ist allerdings kein Einzelfall, in Kapitel 6 wird sogar ein mehrzeiliger Abschnitt mehrfach verwendet. So wird Krantz auch nicht müde, gebetsmühlenartig in nahezu jedem Kapitel zu erwähnen, dass sich ein Mathematiker unbedingt mit  $\text{T}_{\text{E}}\text{X}$  befassen sollte. Da stellt sich mir die Frage, ob dieses Buch jemals an einem Stück Korrektur gelesen wurde.

Fast erheiternd wirkt es, wenn der Autor den charakteristischen Unterschied zwischen einer Schreibmaschine und  $\text{T}_{\text{E}}\text{X}$  erklärt: „ $\text{T}_{\text{E}}\text{X}$  ist nicht *monospaced*“! Geradezu ärgerlich ist dagegen der Abschnitt über die Einbindung von Grafiken. So wird hier die Verwendung von skalierbaren Fonts in PDF- und PostScript-Dateien als Ausnahmefall dargestellt. Im weiteren Verlauf rät Krantz dann davon ab, für die Veröffentlichung eigene PostScript-Grafiken herzustellen, da die verschiedenen PostScript-Versionen untereinander nicht kompatibel sind. Die Möglichkeit, vom Verleger auf das korrekte Format hingewiesen zu werden, sieht er nicht. Es folgt dann, auf über zwei Seiten, ein Beispiel zur Einbindung einer PostScript-Grafik (also doch!) mit Hilfe des `epsf.sty`. Das aktuellere (und bessere) `graphics`-Paket wird dagegen auf einer halben Seite abgehandelt – ein weiterer Hinweis für die unausgereifte Aktualisierung des Manuskriptes. Abgeschlossen wird das zweite Kapitel mit der Beschreibung von  $\text{PCT}_{\text{E}}\text{X}$  und  $\text{BCT}_{\text{E}}\text{X}$ , die beide die Einbindung von `.bmp`-Dateien erlauben. Krantz bleibt uns allerdings die Erklärung schuldig, wie sich die Einbindung dieser Grafikformate mit der von ihm mehrfach erwähnten Portabilität verträgt.

Kapitel 3 und 4 befassen sich dann mit den Standardabschnitten eines Buches, wie Inhaltsverzeichnis, Bibliographie und Glossar. Desweiteren geht der Autor noch auf Fußnoten, Theoreme und Definitionen ein. Er stellt auch über mehrere Seiten eine `.bib`-Datei vor, ohne jedoch den Gebrauch von  $\text{BIB}_{\text{E}}\text{X}$  näher zu erläutern. In den nächsten beiden Kapiteln „Copy Editing“ und „Production Process“ verlässt Krantz den Themenbereich  $(\text{L})_{\text{A}}\text{T}_{\text{E}}\text{X}$  und gibt Hinweise zur Kommunikation mit dem Lektor, er erläutert knapp die ver-

schiedenen Stufen der Entwicklung vom Manuskript zur fertigen Druckvorlage und schildert in kurzen Worten den eigentlichen Druckprozess. Hier erfährt der geneigte Leser, dass E-Mail und FAX sehr effektive Kommunikationsmittel sind und dass er seine Dateien am besten auf mehreren 3,5"-Disketten oder auf einer Zip-Disk übersendet. CD-ROMs und die dazugehörigen Brenner existieren für Krantz an dieser Stelle noch nicht, deren erste Erwähnung bleibt dem letzten Kapitel vorbehalten.

Dieses letzte Kapitel, das mit „Publishing on the Web“ überschrieben ist, beginnt mit einer Einführung in HTML, die sich über eine halbe Seite erstreckt, sowie einer kurzen Anweisung, wie man eine Web-Seite im Browser aufruft – welche Zielgruppe unter den Mathematikern auch immer damit angesprochen werden soll. Es folgt die mittlerweile fast unverzichtbare Web-Adressensammlung, bevor Krantz danach auf die Darstellung von Mathematik im WWW eingeht. Hier zeigt sich, dass dieser Abschnitt, ganz im Gegensatz zu den vorherigen, recht aktuell ist. So werden nacheinander `latex2html`, `TeX4ht`, `techexplorer` und sogar MathML erwähnt, ohne jedoch erschöpfend auf die einzelnen Programme einzugehen. Es bleibt der Eindruck, dass hier zum Schluss unbedingt dem derzeit aktuellen Thema „World Wide Web“ Rechnung getragen werden musste, ohne allerdings den logischen Anschluss an die anderen Abschnitte gefunden zu haben. Überhaupt verlässt den Leser nie das Gefühl, dass es sich bei diesem Buch um die Aneinanderreihung von – ursprünglich unabhängigen – Aufsätzen handeln könnte. Wie ließen sich sonst die zahlreichen Wiederholungen und Gedankensprünge erklären?

Und wo bleibt das Positive? Das versteckt sich in den Anhängen am Ende des Buches. Hier findet sich eine Übersicht von Korrekturzeichen, die auch an einem Beispiel vorgeführt werden. Es werden grundlegende  $\TeX$ -Befehle und mathematische Zeichen aufgelistet, gefolgt von einigen recht aktuellen Web-Adressen zum Themenkomplex „ $\TeX$ , PostScript und PDF“, bevor die Anhänge mit einem mehrseitigen, sehr ausführlichem Glossar beendet werden. In diesem findet sich eine bunte Mischung von Begriffen aus der Computer-Welt, dem Textsatz und aus der Welt des Buchdruckes.  $\LaTeX$  wird hier auch nicht mehr als „Dialekt“ von  $\TeX$  sondern (treffender) als „Makro-Paket“ bezeichnet.

Es ist mir bei diesem Buch nie klar geworden, für welche Zielgruppe es eigentlich geschrieben wurde. Warum erläutert Krantz seitenweise  $\LaTeX$ -Befehle in einem Buch, das nur neugierig machen soll? Warum geht er im Jahre 2001 noch derart ausführlich auf  $\LaTeX$  2.09 ein? Warum hat er sein Manuskript



so schlampig aktualisiert und nicht früher veröffentlicht? Warum finden seine Ratschläge zur Buchveröffentlichung keine Anwendung bei seinem eigenen Werk? Die Beantwortung dieser Fragen bleibt wohl sein Geheimnis. Doch, wie heißt es so schön? *Never judge a book by its cover!* – der Titel dieses Buches hat definitiv zuviel versprochen.

Steven G. Krantz

*Handbook of Typography for the Mathematical Sciences*

Chapman & Hall/CRC, Boca Raton, Florida, 2001

192 Seiten

ISBN 1-58488-149-6 (Paperback), Preis \$ 39.95 / £ 26.99

# Spielplan

---

## Termine

- 12.8.–15.8.2001** TUG 2001 – A T<sub>E</sub>X Live Odyssey  
University of Newark/Delaware, USA  
<http://www.tug.org/tug2001>
- September und Oktober 2001** Calligraphic Type Design in the Digital Age: An Exhibition in Honor of the Contributions of Hermann and Gudrun Zapf  
San Francisco Public Library, Skylight Gallery  
San Francisco, USA  
<http://www.zapfest.com>
- 23.9.–27.9.2001** EuroT<sub>E</sub>X 2001 – “T<sub>E</sub>X and Meta: the Good, the Bad, and the Ugly Bits”  
Kerkrade/Niederlande  
<http://www.ntg.nl/eurotex>
- 28.9.2001** 25. Mitgliederversammlung, DANTE e.V.  
Kerkrade/Niederlande  
<http://www.dante.de/dante/events/mv25/>
- 10.–25.10.2001** Frankfurter Buchmesse 2001  
Frankfurt am Main  
<http://www.frankfurter-buchmesse.de/>  
Kontakt: Ausstellungs- und Messe GmbH Frankfurter Buchmesse
- 24.–26.10.2001** CIDE 2001  
4<sup>th</sup> International Conference on The Electronic Document  
Toulouse, Frankreich  
<http://www.irit.fr/CIDE2001/>

## Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßig Treffen von  $\text{T}_{\text{E}}\text{X}$ -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.: 030/3481316  
[niepraschk@ptb.de](mailto:niepraschk@ptb.de)  
Gaststätte „Bärenschenke“  
Friedrichstr. 124  
Zweiter Donnerstag im Monat, 19.00 Uhr

### Bremen

Martin Schröder  
Tel.: 0421/2239425  
[martin@oneiros.de](mailto:martin@oneiros.de)  
Universität Bremen, Hoyers Unikum  
Erster Donnerstag im Monat, 18.30 Uhr

### Chemnitz

Ralf König  
Tel.: 0371/5905475  
[ralf.koenig@s1998.tu-chemnitz.de](mailto:ralf.koenig@s1998.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](mailto: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](mailto:hille42@gmx.de)  
Cafe B'liebig  
Liebigstr. 24  
Letzter Mittwoch im Monat, 19.00 Uhr

### Erlangen

Walter Schmidt, Peter Seitz  
[was@VR-Web.de](mailto:was@VR-Web.de),  
[p.seitz@koehler-seitz.de](mailto:p.seitz@koehler-seitz.de)  
Gaststätte „Gambrinus“  
Vierzigmannstr. 7  
Dritter Dienstag im Monat, 19.00 Uhr

### Freiburg

Heiko Oberdiek  
Tel.: 0761/43405  
[oberdiek@ruf.uni-freiburg.de](mailto:oberdiek@ruf.uni-freiburg.de)  
Gaststätte „Aquila“  
Sautierstr. 19  
Dritter Donnerstag im Monat, 19.30 Uhr

### Hamburg

Volker Hüttenrauch  
[volker\\_huettenrauch@hh.maus.de](mailto: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 für Niedersachsen  
Tel.: 0511/7625134  
[heisterkamp@rrzn.uni-hannover.de](mailto:heisterkamp@rrzn.uni-hannover.de)  
Seminarraum RRZN  
Schloßwender Str. 5  
Zweiter Mittwoch von geraden Monaten,  
18.30 Uhr

### Heidelberg

Luzia Dietsche  
Tel.: 06221/544527  
[luzia.dietsche@urz.uni-heidelberg.de](mailto:luzia.dietsche@urz.uni-heidelberg.de)  
China-Restaurant „Palast“  
Lessingstr. 36  
Letzter Mittwoch im Monat, 20.00 Uhr

### Karlsruhe

Klaus Braune  
Tel.: 0721/6084031  
[braune@rz.uni-karlsruhe.de](mailto:braune@rz.uni-karlsruhe.de)  
Universität Karlsruhe, Rechenzentrum  
Zirkel 2, 3. OG, Raum 316  
Erster Donnerstag im Monat, 19.30 Uhr

**Köln**

Bruno Hopp

b.hopp@lepkes-frings.de

Institut für Kristallographie

Zülpicher Str. 49b

Letzter Mittwoch im Monat, 19.30 Uhr

**München**

Michael Niedermaier

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 Schwimmooper Wuppertal-Elberfeld

Zweiter Donnerstag im Monat, 19.30 Uhr

# Adressen

---

DANTE, Deutschsprachige Anwendervereinigung T<sub>E</sub>X e.V.  
Postfach 10 18 40  
69008 Heidelberg

Tel.: 0 62 21/2 97 66 (Mo, Mi–Fr, 10<sup>00</sup>–12<sup>00</sup> Uhr)  
Fax: 0 62 21/16 79 06  
E-Mail: [dante@dante.de](mailto: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

## Präsidium

Präsident:	Thomas Koch	<a href="mailto:president@dante.de">president@dante.de</a>
Vizepräsident:	Volker RW Schaa	<a href="mailto:vice-president@dante.de">vice-president@dante.de</a>
Schatzmeister:	Horst Szillat	<a href="mailto:treasurer@dante.de">treasurer@dante.de</a>
Schriftführer:	Günter Partosch	<a href="mailto:secretary@dante.de">secretary@dante.de</a>
Beisitzer:	Klaus Höppner	<a href="mailto:adviser@dante.de">adviser@dante.de</a>

## Server

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

# Die T<sub>E</sub>Xnische Komödie

---

13. Jahrgang Heft 2/2001 Juli 2001

## Impressum

## Editorial

## Hinter der Bühne

- 4 Grußwort
- 5 Einladung zur 25. Mitgliederversammlung von DANTE e.V.

## Bretter, die die Welt bedeuten

- 7 L<sup>A</sup>T<sub>E</sub>X und Chemie – ein schönes Paar?
- 16 Briefköpfe mit KOMA-Script
- 36 Einführung in pdfT<sub>E</sub>X
- 49 A package for typesetting Arabic mathematical formulas

## Rezensionen

- 62 „Handbook of Typography for the Mathematical Sciences“ von Steven G. Krantz

## Spielplan

- 66 Termine
- 67 Stammtische

## Adressen