

Abstracts

Deutsche Kurzfassungen der TUGboat-Artikel [German Abstracts of TUGboat Articles]

Luzia Dietsche

E-TeX: Richtlinien für zukünftige Erweiterungen von TeX (F. Mittelbach, S. 337)

Mit der Ankündigung von TeX 3.0 erkannte Don Knuth das Bedürfnis der (immer mehr anwachsenden) TeX-Gemeinde nach einem verbesserten System an. Gleichzeitig machte er aber auch deutlich, daß er an keinerlei weiteren Entwicklungen, die Änderungen im TeXbuch mit sich bringen würden, beteiligt sein wird.

TeX war ursprünglich dazu entwickelt worden, die eigenen Veröffentlichungen des Autors zu setzen. Inzwischen verwenden es Tausende von Benutzern. Nach 10 Jahren Erfahrung ist es jetzt an der Zeit, sich zu überlegen, ob TeX 3.0 eine adäquate Antwort auf die Anforderungen von Setzern in den 90er Jahren sein kann oder nicht.

Eine Ausgabe, die mit TeX erzeugt wurde, hat einen höheren Standard als mit den meisten anderen Textsatzsystemen erzeugte Ausgaben. Deshalb werden wir uns in diesem Bericht auf den Qualitätsstandard, den Buchdrucker für "handsatzte" Dokumente aufgestellt haben, konzentrieren und fragen, inwieweit dieser Standard von TeX verwirklicht wird. Die Grenzen von TeX's Algorithmen werden analysiert, fehlende Besonderheiten sowie neue Konzepte werden umrissen.

Senkrecht Setzen mit TeX (H. Hamano, S. 346)

Als technischer Verlag hatte ASCII Corporation schon lange den Eindruck, daß auf dem japanischen Markt ein tatsächlich japanisches Textsatzsystem nötig wäre. In einem ersten Schritt entwickelten wir vor drei Jahren eine japanische Version von TeX, mit der es möglich war, *Kanji* zu verwenden.

Dieser Bericht stellt nun unseren zweiten Schritt vor, in dem wir ein erheblich komplizierteres TeX-System entwickelten — das Hinzufügen einer Funktion, mit der man senkrecht Setzen kann.

Ein strukturiertes System zum Erstellen von Dokumenten *AutoLayouter* (Y. Miyabe, H. Ohta, K. Tsuga, S. 353)

Wir haben ein strukturiertes System zum Erstellen von Dokumenten entwickelt (*AutoLayouter*), das aus einem einfach aufgebauten Editor und einem Formtierer besteht, der auf japanischem L^AT_EX basiert.

Dazu haben wir aber nicht nur eine bessere Benutzeroberfläche entwickelt, sondern auch eine einfache Dokumentenstruktur eingeführt. Ein Dokument, das mit *AutoLayouter* hergestellt wurde, ist eine Liste, in der jeder Knoten einer logischen Komponente des Dokuments entspricht. Diese Verwendung einer einfachen Struktur ermöglichte es, den Editor einfach in der Anwendung zu gestalten und dennoch mächtig genug zu lassen, um sowohl die Struktur des Dokuments als auch seines Inhalts, der zusätzliche feinere Untergliederungen enthalten kann, zu editieren.

Seit der Verwendung dieser einfachen Struktur mußten wir verschiedene Makros hinzufügen, um die Unterschiede zwischen unserer Struktur und den *begin-end* Umgebungen von L^AT_EX auszugleichen. Zusätzlich zum Editor haben wir einen Treiber entwickelt, der *dvi*-Files in *Kanji* PostScript Files umwandelt.

TeXnisch werdend: Einblicke in Techniken, mit denen man TeX-Makros schreiben kann (A. Hendrickson, S. 359)

Die meisten von uns verstehen die Grundformen von TeX-Makros; allein das bloße Verstehen ist oft ungenügend, wenn wir bestimmte Probleme zu lösen haben. Wir benötigen dann zusätzliche Einblicke, um Methoden entwickeln zu können, mit denen man Informationen weitergeben, Text mit verändertem Catcode verschieben, Leerzeile erhalten kann und vieles mehr. Schreibt man ein großes Makropaket, sind eine Reihe neuer Aspekte zu beachten: Wie kann man es vermeiden, über die Einschränkungen der diversen Implementationen zu stolpern, wie sie z.B. die Zwänge von *hash size*, *string size* und andere Größen darstellen? Wie macht man eine komfortable Benutzeroberfläche? Wie macht man die Eingabe so knapp wie möglich?

Einige der Techniken, die hier vorgestellt werden, zeigen, wie ein Makro mit variabler Anzahl von Argumenten geschrieben werden kann; wie man Catcodes in Makros ändern kann, indem man ein anderes Makro definiert, dessen Argument absichtlich nie benutzt wird; wie man den Wert von *hash size* gering halten kann, indem man Zähler anstelle

von newifs verwendet; Techniken mit csname und non-outer dynamischen Allokierungen; Techniken, um Tabellen zu setzen; einen Blick auf output routinen. Zu guter Letzt folgen Arbeitsmethoden, die man beim Schreiben neuer Makros verwenden kann.

Wo ist der Umschalter für griechische Buchstaben? (S. A. Fulling, S. 371)

Erfahrene Setzer mathematischer Formeln ermüden bald beim Tippen der Namen von griechischen Buchstaben. Meist schreiben sie dafür einfache Makros wie z.B. \za für \alpha. Hier wird nun eine Standardübereinstimmung zwischen griechischen und lateinischen Buchstaben vorgestellt, in der phonetischer Ähnlichkeit Vorrang vor typographischer gegeben wird. Mit den daraus resultierenden Makros wird es möglich, beliebige griechische Buchstaben durch dreifachen Anschlag zu tippen (oder sogar durch zweifachen, wenn man \z auf eine Funktionstaste legt).

TransFig: Übertragbare Grafiken für T_EX (M. Beck, A. Siegel, S. 373)

Das TransFig Software Paket definiert eine übertragbare Beschreibungssprache für technische Grafiken. Es werden Übersetzungen von dieser Sprache in oft verwendete Grafik-Beschreibungsformate geliefert, die dann in eine Eingabe eingebunden werden können. TransFig beinhaltet außerdem einen ausgesprochen angenehmen Rahmen, um Figuren in L^AT_EX einzubinden. Die Grafik-Sprache, die mit TransFig definiert ist, ermöglicht den Austausch von strukturierten und modifizierbaren Grafiken zwischen verschiedenen Anwendungen. In diesem Bericht beschreiben wir unsere Erfahrungen mit TransFig, um die Notwendigkeit einer standardisierten Grafik-Sprache auf Anwendungsebene zu verdeutlichen und schlagen einige Richtlinien für deren Entwicklung vor.

BAS_IX: Ein in T_EX geschriebener Interpreter (A. M. Greene, S. 381)

Ein komplett in T_EX geschriebener Interpreter für BASIC wurde entwickelt. Er stellt Techniken zum scannen und analysieren zur Verfügung, die oftmals da von Nutzen sind, wo Daten, die keine Formatierungsvorschriften enthalten, mit T_EX formatiert werden sollen. T_EX's Expansions Regeln werden ausgenutzt, um ein Makro bereit zu stellen, das den Rest der Eingabe als Argument einliest und nicht mehr aufhört, ihn zu expandieren.

Ein Anfängerleitfaden für die Benutzung von T_EX in einer Textproduktion: Vom Manuskript zum Bromid (H. Gibson, S. 393)

Die Absicht dieses Berichts ist es, ein Beispiel für eine PC-Anpassung zu geben, mit der qualitativ hochwertige Druckausgabe erzeugt werden kann. Einer kurzen Beschreibung der Anforderungen, die an Veröffentlichungen des Wellcome Institutes gestellt werden, folgt eine ausführliche Darstellung der Hard- und Software-Konfiguration, der Umlenkung der Ausgabe zu PostScript und der Verwendung von elektronischer Kommunikation, mit der auf Laserdruckern kontrollierter Text auf eine Linotronic 300 Lichtsatzmaschine geschickt werden kann. Außerdem wird der Wert der Verwendung von WordPerfect Makros, alternativer Keyboard Layouts und Style Files als eine Oberfläche für die Eingabe durch Sekretärinnen dargestellt.

Probleme mit der T_EX/PostScript/Grafik Oberfläche (R. A. Adams, S. 403)

In diesem Bericht werden die unterschiedlichen Probleme behandelt, die bei der gleichzeitigen Verwendung von T_EX und PostScript zur Produktion von zwei Formelsammlungen auftreten können. Drei solcher Probleme sind besonders wichtig. Das erste ist es, eine vernünftige Kombination von mit PostScript (skaliertem) Text und den mathematischen Formeln zu erhalten, die auch noch bei einer 1270 dpi Ausgabe auf einer Linotronic Lichtsatzmaschine gut aussehen. Das zweite Problem ist, eine brauchbare Methode zu ersinnen, um eine passende (und gut ausgerichtete) 2-Farben-Trennung von Text und Grafik zu erhalten. Das dritte bringt es mit sich, T_EX Marken mit PostScript Grafiken zu verbinden. Die Lösungen zu diesen Problemen sind bestimmt durch die Software, die zu der Zeit vorhanden war, als sie benötigt wurden (vor ca. 1½ Jahren).

T_EX in der Praxis: Kommentare zu einem 4-bändigen, 1400 Seiten starken Werk in T_EX (S. v. Bechtolsheim, S. 409)

Dieser Artikel handelt von einem 4-bändigen, 1400 Seiten starken Werk über T_EX. Zwei Aspekte werden behandelt: Als erstes werde ich zeigen, wie die Produktion eines Dokuments dieser Größe organisiert wurde. Als Zweites zeige ich Erweiterungen in T_EX, die ich dazu für wünschenswert hielt.

Veröffentlichung von Büchern — 1990 und darüber hinaus (M. L. Lafrenz, S. 413)

Die Produktion universitärer Veröffentlichungen und Handbücher mit \TeX ist etwas Selbstverständliches. Trotzdem verstehen, und dadurch akzeptieren, Verlage erst allmählich die Mächtigkeit und Vielfalt dieses Programms. Einige Verlage verwenden bereits \TeX -Files von Autoren, um Kamerafertige Seiten zu produzieren, während andere nur sehr zögernd diese Möglichkeit in Betracht ziehen. Der Widerstand von kommerziellen Anwendern, Makros, Tricks und Techniken auszuplaudern, verhinderte die Akzeptanz von \TeX durch die Verlage.

Wir werden die Schulung und Unterstützung von Autoren und Verlegern sowie die Einwirkung von Offenheit auf den technischen Bereich untersuchen. Nur durch die Aufteilung in verschiedene Gebiete werden wir in der Lage sein, die beste Umgebung zu schaffen, um \TeX in diesem Jahrzehnt zur Blüte zu bringen.

\TeX -Fehler mit einem Präprozessor diagnostizieren (D. Ness, S. 417)

\TeX findet unsere Fehler mit Leichtigkeit. Allerdings teilt es uns diese manchmal auf eine schwer verständliche Weise mit. Normalerweise kommt das dadurch, daß wir \TeX durch die unbegründete Verwendung von Irgendetwas irritieren. Ein einfaches Beispiel: Wenn wir vergessen haben, vor ein Dollarzeichen ein Befehlszeichen zu setzen, gibt uns \TeX im Normalfall eine eigenartige Meldung über den mathematischen Modus zurück. Der vorliegende Bericht handelt von einem Präprozessor, der uns vor potentiellen Fehlerquellen warnt, bevor wir unsere Files von \TeX bearbeiten lassen. Besonders für Neulinge kann dieses Programm von Nutzen sein.

Zusätzliche \TeX Leistung durch die Verwendung von erweiterten EDT Editor-Möglichkeiten (L. Williams, L. Hall, S. 421)

\TeX ist auf vielen Computersystemen installiert, allerdings oft mit wenig oder gar keinen Gedanken an die zeitsparenden Vorteile eines leistungsstarken Editors. Sowohl für Neulinge als auch für erfahrene \TeX er eröffnet das Arbeiten mit den erweiterten EDT Editor-Möglichkeiten den effizienten Gebrauch von \TeX . Da mehrere Bände dessen verschiedene Hilfsmittel beschreiben, wurde es nötig, diese entsetzliche Anhäufung von Material auf ein anwendbares Minimum zu reduzieren. In diesem Artikel werden einzelne EDT-Möglichkeiten vorgestellt, wie Editor-Initialisierungs-Files und andere Komman-

dos, die, wenn sie richtig angewandt werden, die Editor-Möglichkeiten von \TeX noch steigern.

\TeX für \TeX nische Sekretärinnen (C. R. Martin, S. 425)

Viele \TeX Benutzer sind keine Programmierer oder Mathematiker, sondern Fachsekretärinnen, Praktiker eines erlernten Handwerks. Solche Benutzer finden bereits vorhandene \TeX -Eingaben oftmals einschüchternd und geheimnisvoll. \TeX für \TeX nische Sekretärinnen ist gerade für solche Benutzer gedacht. Jede Einheit beschreibt einige Konzepte, wobei diese sofort durch praktische Anwendungen in einem kurzen Dokument vertieft werden. Schüler können sich die Resultate direkt ansehen, was zu schnellerem Fortschritt und größerer Zufriedenheit führt. Der Kurs fördert eine experimentelle Haltung, die sich in der Praxis gut bewährt hat. \TeX für \TeX nische Sekretärinnen scheint ein effizienter Weg zu sein, mit dem Fachkräften beigebracht werden kann, schöne Dokumente zu erstellen.

\VTeX Steigerungen zur \TeX -Sprache (M. Vulis, S. 429)

\VTeX erweitert \TeX um die Unterstützung durch skalierbare Fonts und damit um das Erlangen echter Treiber-Unabhängigkeit. \VTeX bringt \TeX in ein kompaktes System (weniger als 10% seiner ursprünglichen Größe), unterstützt eine Druckertreiber Definitionssprache, vervollständigt das \TeX -System mit einer Anzahl von neuen, qualitativ hochwertigen Schriften und implementiert eine Reihe von Font-Effekten (verdichtete, schraffierte, umrißene, mit Schatten versehene Fonts). Die Unterstützung von skalierbaren Fonts erfordert bestimmte Änderungen im \TeX Programm, der Syntax und den Fonts; dieser Artikel beschreibt einige der Änderungen. Da es wahrscheinlich ist, daß weitere Anpassungen für ein skalierbares \TeX folgen werden, hofft der Autor, daß eine Standardisierung darüber gefunden werden kann, bevor Konflikte durch verschiedene Sätze von Definitionen entstehen.

Ein konstruiertes Dürer Alphabet (A. Hoenig, S. 435)

Der Autor benutzte METAFONT, um ein Albrecht-Dürer-Alphabet in Roman Großbuchstaben zu entwickeln. Obwohl Dürer nur an ein Alphabet in Roman-Schrift dachte, wurde der Versuch unternommen, ein METAFONT-Programm zu entwickeln, mit dem verwandte Fonts erzeugt werden können in einem bold, sans serif, typewriter-ähnlichen, slanted und casual Stil. (Das letzte ist ein Stil in Anlehnung an die formlose Roman Buchstabengestaltung, die

von Sumner Stone und Neenie Billawala entwickelt wurde.)

Luzia Dietsche
Rechenzentrum der Universität
Heidelberg
Im Neuenheimer Feld 293
D-6900 Heidelberg 1
X68@DHDURZ1

Status Reports

TeX 3.0 and METAFONT 2.0

Nelson H.F. Beebe

Introduction

During the last few years, use of TeX has spread to Chinese, Japanese, Korean, Coptic, Russian, Thai, Vietnamese, several Indian languages, Persian, Arabic, Hebrew, and all major European languages. These uses made some of the limitations of TeX 2.x more evident. The most serious of these are the seven-bit character set and uni-lingual hyphenation. Consequently, Donald Knuth announced at the 10th Annual Meeting of TUG, held at Stanford University in August, 1989, that a new version of TeX and METAFONT would be produced to address problems of multinational support; those versions were officially released on the ides of March, 1990. Don resisted calls for more sweeping changes, in the interests of (a) restricting the impact of the changes, and (b) getting back to writing *The Art of Computer Programming* books.

At the time of writing this report (late June 1990), the two new programs, and their related TeXware and METAFONTware, including BibTeX, have been successfully installed on numerous UNIX variants, using version 5.0c of the Web2C translator, as well as on the IBM PC, VAX VMS, and IBM VM/CMS. Several commercial vendors announced release of the new versions at the June TUG meeting. We hope that, by the end of 1990, all TeX users the world over will have had the opportunity to upgrade to the new versions.

What's New

The new features of TeX 3.0 and METAFONT 2.0 are described in detail in [3]. It is important to

emphasize that, apart from the obscure exceptions noted in section 12 of that paper, changes should not affect existing TeX and METAFONT files. In particular, *DVI files are unchanged, and old font files can still be used.*

Here is a list of the current support programs and their versions:

bibtex	0.99c	pltotf	3.2
dvitype	3.2	tangle	4.0
gftodvi	3.0	tex	3.0
gftopk	2.2	texinfo	3.0
gftype	3.0	tftopl	3.1
inimf	2.0	vftovp	1.0
initex	3.0	virmf	2.0
mft	2.0	virtex	3.0
pktogf	1.0	vptovf	1.0
pktype	2.2	weave	4.1

Hyphenation can be applied to input words coded using only 8-bit characters. This means that a user with a European keyboard can enter a word like *liberté* (the accented letter is a single character) without losing a hyphenation opportunity. However, if the word is entered using a control sequence, `libert\'e`, as for earlier versions of TeX, hyphenation will not be tried.

Michael Ferguson has graciously made available a change file for TeX 3.0 that translates input accented character sequences to the internal 8-bit form, and at DVI output time, translates them back again. This change makes it possible for words accented with control sequences to be hyphenated, and seems to be a very valuable extension.

The new ligature mechanism allows letters to change, based on their position within a word. This is of particular importance in Arabic, but may be used in other languages: observe that in normal handwriting, letter shapes may also vary according to position, so a handwriting font might usefully incorporate such ligatures.

`\emergencystretch` (section 7 of [3]) allows better control of excessive white space. L^ATeX users in particular should note this new parameter. The L^ATeX `\sloppy` command sets `\tolerance` to its maximum value of 10000, allowing all lines to be loosely spaced. Under TeX 3.0, a smaller `\tolerance` value can be chosen and `\emergencystretch` can be brought into play when TeX deems it necessary, thus eliminating excessive white space. The L^ATeX bibliography environment by default invokes `\sloppy`; try a two-column bibliography first without `\emergencystretch`, then with it and a reduced `\tolerance` to see the difference.