Zusätzlich zu diesen PostScript-Textschriften gibt es noch die PostScript-Symbolschrift Symbol mit den Namen `sym` in normal und `syms` in slanted[22]. Außerdem ist noch eine PostScript-Schrift mit speziellen Symbolen, die es in TeX eigentlich nicht gibt, verfügbar. Sie heißt ZapfDingbats und ist in TeX unter dem Namen `zpfd` ansprechbar. Die Verwendung dieser Schriften gestaltet sich etwas problematischer, da Sie die Zeichen zwar als Buchstaben (bzw. als ASCII-Zeichen) ansprechen können, aber keine Buchstaben (bzw. ASCII-Zeichen) dabei herauskommen. Die Zeichen müssen erstmal umcodiert werden: Welcher Buchstabe liefert welches Zeichen? erst umcodiert werden müssen. Sie können ja mal die ZapfDingbats-Schrift testen, indem Sie sie mit `\newfont{\zpfd}{zpfd}` laden und dann einige Zeichen mal ausprobieren. Vielleicht haben Sie ja Verwendung für das eine oder andere Zeichen, und definieren es sich als Makro. Das kann dann so aussehen:

Eingabe:`{\newfont{\zpfd}{zpfd}\zpfd abcdefghijklmnopqrstuvwxyzABCDEFGHIJK}`
Ergebnis: ❁❂❃❄❅❆❇❈❉❊❋●❍■❏❐❑❒▲▼◆❖◗❘❙❚✡✢✣✤✥✦✧★✩✪✫

Eine vollständige Liste der Symbole aus den Symbol- und ZapfDingbats-Zeichensätzen finden Sie in so ziemlich jedem PostScript-Schrifttest. Einige solche PostScript-Schrifttests finden Sie bei GhostScript. Zur Verwendung der Symbolschrift (die Zeichen enthält, die für den mathematischen Schriftsatz in TeX nutzbar sind) gibt es die Makrodatei `PSMATH.STY`, doch dazu komme ich noch. Natürlich können Sie auch die Symbolschriften auf ähnliche Weise erforschen wie oben.

Doch zurück zum Schriftenkatalog: Wie Sie eventuell sehen, sind die Schriften mit System benannt[23]. Die Original-PostScript-Schriftnamen sind leider viel zu lang, als daß man sie als DOS-Dateinamen hätte verwenden können[24]. Also mußte gekürzt werden, aber das mit System. Über die oben aufgeführte Ausstattung an Schriften verfügt fast **jeder** PostScript-fähige Drucker **mindestens**. Ich will im weiteren auch nur diese Minimalmenge erklären (die aber m. E. schon ein sehr reichhaltiges Angebot an verschiedenen Schriften darstellt). Wenn Sie mehr PostScript-Schriften zur Verfügung haben: Herzlichen Glückwunsch! Sehen Sie zu, wie Sie diese in TeX einbinden können (oder fragen Sie mich... oder lesen Sie das nächste Kapitel...).

Sie sehen vielleicht, daß einige Namensanfänge immer wiederkehren. Im obigen Katalog sind letzten Endes nur acht verschiedene Schriften enthalten, aber diese in ziemlich vielen verschiedenen Ausfertigungen.

Im einzelnen: Zunächst gibt es da die AvantGarde-Familie. Das sind die Schriften, die mit „avant“ beginnen. Diese Schrift gibt es in fünf Ausfertigungen: Als Normalschrift (avant), als Fettschrift (avantd, das 'd' steht für 'demibold', das ist in etwa gleichbedeutend mit 'bold', nur etwas dünner. Wir fassen es als 'bold' auf), als Italics-Schrift (avanti, das 'i' steht für 'italics'), als SmallCaps-Schrift (Kapitälchen) (avantcsc, das 'csc' steht für 'Computed Small-

[22] Die PostScript-Symbolschrift enthält in erster Linie griechische Buchstaben, die auch für den mathematischen Formelsatz unter TeX verwendbar sind.

[23] Allerdings mit eigenem System, das sich nicht mit einem Benennungssystem verträgt, das von T. Rokicki vorgeschlagen wurde. Da sich aber auch das noch nicht endgültig durchgesetzt hat, behalten wir unser System erst einmal bei. Wenn Sie möchten, können Sie ja alle Schriften nach dem System von Rokicki umbenennen; Sie finden seine Systematik in der Original-DVIPS-Anleitung beschrieben.

[24] Unter anderen Betriebssystemen, die Dateinamen mit normalen Wortlängen ermöglichen (alles aus dem UNIX-Bereich etwa) ist die Abkürzerei allerdings auch nicht ganz sinnlos, denn sie erspart unter Umständen lästige Tipparbeit; deshalb wird also auch dort meistens nach System abgekürzt.

Caps') und als Fett-Italics-Schrift (avantdi, das 'di' steht für 'demibold-italics', was in etwa gleichbedeutend mit 'bold-italics' ist, wie Sie sich sicherlich schon gedacht haben)[25].

Das gleiche Schema kann man für die Schriftfamilie Bookman erkennen. Da gibt es die Normalschrift (bookl, l=light=dünne Strichstärke, aber es ist trotzdem keine extraschmale Schrift, sondern die normale), eine Fettschrift (bookd, d=demibold=etwas dickere Strichstärke (fett)), eine Italics-Schrift (bookli, li=light-italics), eine Fett-Italics-Schrift (bookdi, di=demibold-italics) und eine SmallCaps-Schrift (booklcsc).

Dann kommt die Schriftfamilie NewCenturySchoolbook (abgekürzt cent). Hier werden die fetten Schriften gekennzeichnet durch ein 'b' für 'bold'[26]. Diese Schriftfamilie wird in unserem System abgekürzt durch 'cent', was nicht ganz so offensichtlich ist. Anderswo kürzt man es mit 'ncs' ab. Hier gibt es Normalschrift (cent), Fettschrift (centb), Fett-Italics-Schrift (centbi), SmallCaps-Schrift (centcsc) und Italics-Schrift (centi).

Nächste Familie: ZapfChancery. Diese Familie wird mit 'chanc' abgekürzt und es steht nur die Normalschrift zur Verfügung (das ist, voll ausgeschrieben, ZapfChancery-MediumItalic).

Über die Schriften der Courier-Familie (cour) verliere ich dann auch kein überflüssiges Wort, Sie sehen ja oben, was es da gibt.

Die nächste Familie ist die Helvetica-Familie. Sie sehen: Das ist eine serifenlose Schrift. Von dieser Familie gibt es zwei Unterfamilien: Die normalen Helvetica-Schriften (helv, helvb, helvi, helvcsc, helvbi) und dann die etwas geschrumpften Helvetica-Schriften. Diese werden als Helvetica-Narrow oder Helvetica-Condensed bezeichnet. Davon haben wir nur Stücker vier: helvc (die normale Helvetica-Condensed-Schrift), helvcb (eine fette Helvetica-Condensed-Schrift), helvcbi (eine fette Italics-Helvetica-Condensed-Schrift) und helvci. Es gibt hier keine SmallCaps-Schrift, die gibt es nur in der etwas breiteren Version. Ich habe es mit einer SmallCaps-Schrift für Helvetica-Condensed versucht, aber das Ergebnis sah nur scheußlich aus. Also erspare ich Ihnen das.[27]

Dann kommen die Schriften der Palatino-Familie, abgekürzt mit 'pal'. Die lassen sich neigen, strecken und stauchen ohne dabei häßlich zu werden, daß es eine Freude ist. Darum gibt es in dieser Familie ein paar Schriften mehr. Zunächst die üblichen Variationen: pal, palb, palbi, pali, palcsc. Die dürften mittlerweile klar sein. Dann gibt es noch eine Palatino-Condensed-Normalschrift (palc), eine leicht verbreiterte Palatino-Normalschrift (Palatino-Extended, abgekürzt 'palx'), eine Palatino-Slanted (palsl), eine Palatino-UnslantedItalics (palu) und eine Palatino-UnslantedBoldItalics (palbu).

Die letzte Familie der obigen Liste ist die Times-Familie. Da gibt es Normalschrift (times), dann die üblichen timesb, timesi, timesbi, timescsc. Darüber hinaus noch (ähnlich wie bei Palatino) ein Times-Condensed (timesc), ein Times-Extended (timesx) und ein Times-Slanted

[25] Eigentlich sind die mit *Italics* bezeichneten AvantGarde-Schriften gar keine echten Italics-Schriften, sondern Oblique- Schriften. Aber das sind typographische Feinheiten.

[26] Warum das einmal 'demibold' (='semibold', was ein zulässiges Schriftgewicht für NFSS wäre, siehe Tabelle 2) und einmal 'bold' heißt, wenn wir es doch beides so behandeln, als ob es 'bold' wäre? Das NFSS bietet zwar die Möglichkeit, zwischen verschiedenen Strichdicken bei einer Schrift zu differenzieren, aber wenn man nur zwei unterschiedliche Strichstärken zur Verfügung hat, entfällt der Sinn dieser feinen Differenzierung. Dann ist das eine eben fett und das andere normal, ungeachtet der typographischen Feinheiten, die in den Strichstärken (Schriftgewichten) steckt.

[27] Auch bei den Helvetica-Schriften gibt es wieder keine Italics-Schriften im typographischen Sinne, sondern Oblique-Schriften, die wir aber (der Einfachheit halber) als Italics-Schriften bezeichnen.

(timessl).

Wenn Sie das Kapitel über das NFSS aufmerksam gelesen haben, können Sie sich wahrscheinlich schon denken, wie man auf einfache Weise mit PostScript-Schriften arbeiten kann: Man braucht ja nur die Schriftfamilie umzuschalten! Vorausgesetzt, es wurden vorher die nötigen Definitionen durchgeführt. Doch dazu komme ich noch.

7.3 Noch mehr PostScript-Schriften als der Standard

7.3.1 Einbindung zusätzlicher PostScript-Schriften

Über diese Standardschriften hinaus wurden dem Paket noch die für die Verarbeitung einiger weiterer verbreiteter PostScript-Schriftfamilien nötigen Dateien mitgegeben. Die Schriften selber stehen am HRZ zur Zeit noch nicht zur Verfügung. Wenn Sie jedoch Zugang zu ihnen haben, können Sie sie mit den zusätzlichen Dateien dann auch für TEX nutzbar machen. Dies sind die Schriften der beiden Stone-Familien (stin=StoneInformal und ston=StoneSerif), die Optima-Familie (opt) und die ADOBE-Garamond-Familie (gar). Welche zusätzlichen Schriften es gibt, und wie Sie heißen, das können Sie aus dem Inhalt des Verzeichnisses `\TEXFONTS\VF` ersehen. Hier sind alle dem Paket beigefügten TEX-PostScript-Textschriften mit Ihrem TEX-Namen (das ist natürlich der Dateiname) verzeichnet. Bei den ADOBEGaramond-Schriften ist zu beachten, daß dort das Kürzel 'o' für 'Oblique' steht (wenn Sie die letzten Fußnoten gelesen haben, wissen Sie, was das heißt).

Um die zusätzlichen Schriften einzubinden, benötigen Sie zunächst einmal die Header- oder Definitionsdateien. Diese zeichnen sich durch die Extension `.PFB` aus. Diese Header-Dateien kopieren Sie in das Verzeichnis `\EMTEX\PS`. Dann müssen Sie die Datei `PSFONTS.MAP` im Verzeichnis `\EMTEX\PS` ändern. Suchen Sie die Zeile, in der die Schrift, die Sie einbinden wollen, aufgeführt ist. Am Ende dieser Zeile hinter dem Zeichen '<' steht der Name der Header-Datei. Tragen Sie dort ein, wie Sie Ihre Header-Datei abgekürzt haben. In der Datei `\EMTEX\PS\PSFONTS.MAP` sind nämlich noch alle Header-Dateien mit vollem Namen eingetragen.

Falls Sie eine Schrift einbinden möchten, für die Sie noch keine entsprechende `.VF`- und `.TFM`-Datei haben, müssen Sie diese auf jeden Fall noch mit dem Programm AFM2TFM generieren. Genaueres siehe in der Original-DVIPS-Anleitung. Außerdem muß die neue Schrift noch in der Datei `PSFONTS.MAP` eingetragen werden. In diesem Punkt übrigens finden sich die wesentlichen Abweichungen zwischen TrueType und PostScript. Alles, was DVIPS bietet, können Sie ebensogut auf einem TrueType-Drucker ausgeben, nur wenn Sie mit zusätzlichen Schriften arbeiten, gibt es Probleme, da die Schriftencodierung unter TrueType eine andere ist.

7.3.2 PostScript-Schriften selber machen

Wenn Sie Ahnung von PostScript haben, dann können Sie auch PostScript-Schriften selber erstellen. Da gibt es grundsätzlich zwei Möglichkeiten. Zum einen können Sie selber eine Header-Datei schreiben und damit einen ganz neuen Schrifttyp designen... (wenn Sie wissen, wie man so etwas codiert, und das war bis vor einiger Zeit ein sehr gut gehütetes Geheimnis). Aber glauben Sie mir: Es hat einen einsichtigen Grund, warum die Anschaffung zusätzlicher

Schriften auf dem Software-Markt nicht ganz billig ist!

Zum anderen aber können Sie auch von einer existierenden Schrift ausgehen und diese modifizieren. Sie können z. B. recht einfach eine Schrift ZapfChancery-MediumUnslantedItalics auf der Basis der Standard-Schrift ZapfChancery-MediumItalics herstellen. Dazu denken Sie sich einfach eine sinnvolle Abkürzung für die Schrift aus, sagen wir 'ZpfCMU' und tragen nun in die Datei `PSFONTS.MAP` eine Zeile folgenden Inhalts ein:

```
ZpfCMU ZapfChancery-MediumUn "/ZapfChancery-MediumItalic -0.1804 SlantFont"
```

Dabei ist `ZpfCMU` der abgekürzte Schriftname (Ihre Willkür!), `ZapfChancery-MediumUn` der volle PostScript-Schriftname (den Sie auch nach Willkür setzen können) und der Text in den Anführungszeichen ist die PostScript-Kommandosequenz, mit der die Schrift modifiziert werden muß. Das Muster ist leicht zu verstehen, und Sie finden in der Datei `PSFONTS.MAP` ausreichend Beispiele dafür.

Das war einfach, aber jetzt kommt die wirkliche Arbeit: Sie benötigen nun die `.AFM`-Datei der Schrift, von der Sie ausgegangen sind. Diese Dateien können Sie am HRZ für die PostScript-Standardschriften bekommen. Auf diese müssen Sie nun das Programm AFM2TFM, das auch im Paket ist, loslassen und eine `.TFM`-Datei und eine `.VF`-Datei erzeugen. Dabei müssen Sie zunächst eine `.TFM`-Datei für die reine (nicht umcodierte), modifizierte PostScript-Schrift erzeugen. Dann müssen Sie sich für die Schrift einen TeX-Namen ausdenken (im Beispiel am besten 'chancu') und eine `.TFM`-Datei für die nach TeX-Schema umcodierte Schrift generieren, aus der Sie dann auch die `.VF`-Datei erzeugen können. Etwas genauer steht die Bedienung des Programms AFM2TFM in der Original-DVIPS-Anleitung beschrieben. Aber wenn Sie sich in dieser Richtung an die Arbeit machen und bereit sind, die Früchte Ihrer Arbeit mit anderen Benutzern von DVIPS zu teilen, bin ich gerne bereit, Ihnen jede Hilfestellung zu geben. Nur ist das Ausprobieren so lästig, bis man durch Modifikation einer existierenden Schrift etwas wirklich Schönes generiert hat.

Sie könnten nun natürlich fragen, warum ich denn obiges Beispiel nicht auch in der Praxis ausgeführt und dem Paket beigelegt habe. Ich habe es einmal ausprobiert, das Ergebnis sah verheerend aus. Falls irgendjemand einen vernünftigen Weg findet, die ZapfChancery-Schrift zu modifizieren, (ohne daß es reine Effekthascherei ist), so ist das sicherlich von einigem Wert.

7.4 Verwendung von PostScript-Schriften in TeX-Dokumenten

7.4.1 Vorteile und Nachteile

Vorteile bei der Benutzung von PostScript-Schriften mit der Hilfe von DVIPS und TeX fallen mir natürlich reichlich ein (ich muß ja meine Arbeit irgendwie rechtfertigen):

- Es spart Platz. Für die PostScript-Schriften brauchen Sie nur die `.TFM`- und die `.VF`-Dateien, die ja nicht viel Platz auf Ihrer Platte wegnehmen. Sie brauchen keinerlei `.pk`-Dateien! Sie benötigen kein METAFONT, um die Schriften in beliebigen Größen zu generieren!
- Damit wären wir auch gleich beim zweiten Vorteil: Sie können die PostScript-Schriften mit TeX in beliebigen Größen nutzen. Die Berechnung der Schriften in den entsprechen-

den Größen besorgt das PostScript-Ausgabegerät (und zwar erheblich schneller, als das mit METAFONT möglich wäre)!

- Und nicht zuletzt: Die Abwechslung gegenüber dem starren TEX-Schriftenschema, das bewirkt, daß man jedes TEX-Dokument sofort als solches erkennt, ist doch auch was wert! Ohne weiteren Aufwand an Speicher oder Rechenzeit eröffnet sich ein riesiges, neues Schriftenspektrum.

Nachteile fallen mir natürlich keine ein, bis auf diese unwesentlichen Kleinigkeiten:

- Sie können sich Ihr TEX-Dokument nur noch verzerrt oder gar nicht mehr mit dem emTEX-Preview ansehen, da der natürlich keine PostScript-Schriften kennt, der braucht ja `.pk`-Dateien, um etwas darstellen zu können. Aber da gibt es Abhilfe: Es gibt Programme, mit denen man aus PostScript-Schriften `.pk`-Dateien generieren kann, aber die sind kommerziell (kosten Geld) und außerdem wollte man doch gerade die `.pk`-Dateien einsparen... Eine andere Möglichkeit ist ein Display-PostScript-Interpreter, der PostScript-Dateien auf dem Bildschirm darstellen kann. Ein solcher ist zum Beispiel GhostScript, was ja auch zum DVIPS-Paket gehört. Sie benutzen als Preview dann einfach nicht mehr den emTEX-Preview, sondern GhostScript. Damit wäre dieser kleine Nachteil schon mal entfallen. Eine ganz andere Möglichkeit der Abhilfe ist es, das TEX-Dokument mit den normalen TEX-Schriften erstmal zu entwickeln und zu schreiben und dann später alles auf PostScript-Schriften umzustellen. Wie das einfach und komfortabel geht, erfahren Sie im übernächsten Abschnitt.

- Die Hardware-Anforderungen sind bei Verwendung von PostScript-Schriften etwas höher gegenüber der ausschließlichen Verwendung von TEX-Schriften. Siehe dazu auch im Kapitel *Erfahrungen mit DVIPS*.

- Sie benötigen zum Ausdruck Ihrer Dateien natürlich einen PostScript-fähigen Drucker oder einen PostScript-Interpreter[28] für Nadeldrucker und nicht PostScript-fähige Laserdrucker. Aber auch das ist ja im Prinzip nur eine erhöhte Hardware-Anforderung (wenn auch eine recht teure, doch es stehen Ihnen entsprechende Geräte am HRZ zur Verfügung).

Ich hoffe, Sie können es nachvollziehen, wenn ich behaupte, daß die Vorteile deutlich überwiegen, denn die Nachteile sind ja gar keine.

7.4.2 Direkte, manuelle Verwendung von PostScript-Schriften

Zunächst zur Verwendung von PostScript-Schriften ohne NFSS. Ich will die Verwendung von PostScript-Schriften hier nur an LATEX-Befehlen erläutern. Wenn Sie mit PlainTEX arbeiten, werden Sie wohl wissen, wie man diese Befehle dafür umzusetzen hat. Umgekehrt ist das erfahrungsgemäß nicht immer der Fall.

Sie können die PostScript-Schriften so verwenden, wie sie sich auch jeden anderen Schrifttyp unter LATEX zugänglich machen können; das entscheidende Kommando dafür ist

[28] Den Sie natürlich in Form des Programmes GhostScript haben!

`\newfont{\`*<Schriftkommando>*`}{`*<Schriftname>*`}`.

Die TEX-Namen der PostScript-Schriften waren ja im PostScript-Schriftkatalog angegeben. Wenn Sie also die Schrift Palatino-BoldItalics in Ihrem Dokument verwenden möchten, dann wissen Sie: Der TEX-Name dieser PostScript-Schrift ist „palbi“. Sie können also dieses LATEX-Kommando geben:

```
\newfont{\palbi}{palbi}
```

Danach können Sie mit dem Kommando `\palbi` ***auf die Schrift Palatino-BoldItalics in der normalen Textgröße umschalten. Gesehen?*** Sie können die Schrift natürlich auch anders nennen, z. B. „fettschief“. Das sähe dann so aus:

```
\newfont{\fettschief}{palbi}
```

Jetzt können Sie mit `\fettschief` ***auf die Schrift Palatino-BoldItalics in der normalen Textgröße umschalten. Auch gesehen?*** Nun kann es auch sein, daß Sie die Schrift in etwas größer haben möchten. So lädt man mit

```
\newfont{\fettschiefgr}{palbi scaled \magstep1}
```

die Schrift in der ersten Vergrößerungsstufe. Wenn Sie jetzt das Kommando `\fettschiefgr` ***geben, dann sieht das etwa so aus. Sie merken: Die Schrift ist jetzt etwas größer als die normale Textschrift.*** Sie wissen sicherlich: In LATEX gibt es standardmäßig sechs verschiedene Vergrößerungsstufen: `\magstep0`=normale Textgröße, und dann die Vergrößerungen `\magstephalf` und `\magstep1` bis `\magstep5`[29]. Wie groß die Schrift, die Sie mit `\magstep`... nun anwählen wirklich wird, hängt davon ab, welche Standardschriftgröße Sie für Ihr Dokument gewählt haben (was sie also bei `\documentstyle[]{}` eingestellt haben). Die festen Vergrößerungsstufen von LATEX mit `\magstep` sind **relative** Vergrößerungsangaben. Sie können aber die Vergrößerung auch **absolut** angeben. Und wie gesagt: Bei Verwendung von PostScript-Schriften sind Ihrer Fantasie keine Grenzen gesetzt! Sie können Schriftgrößen anfordern, die bei normalen TEX-Schriften nirgendwo installiert sind. So können Sie z. B. mit einer 38-Punkt-Schrift schreiben:

```
\newfont{\monstroes}{palbu at 38 pt}
```

Mit dem Befehl `\monstroes` **passiert das hier. Schnell wieder abschalten**.
Sie können sich vielleicht denken, wie etwa eine 100-Punkt-Schrift aussieht... Wie gesagt, es gibt keinerlei Begrenzungen außer denen des guten Geschmacks. Natürlich können Sie auch mit absoluten Angaben beliebig verkleinern. Probieren Sie doch einmal eine 0.5-Punkt-Schrift oder eine 10 cm-Schrift aus!

---

[29] Leider gibt es in LATEX nicht auch solche standardmäßigen Verkleinerungsstufen. Verkleinerungen müssen Sie absolut als Punktgrößen angeben. Natürlich gibt es Verkleinerungen als LATEX-Kommandos: `\small`, `\footnotesize` usw.

7.4.3 Bequeme, automatische Verwendung von PostScript-Schriften

Nun ist es recht mühsam, alle PostScript-Schriften mit `\newfont{}{}` einzeln einzubinden. Außerdem möchten Sie vielleicht Ihren gesamten Text in PostScript-Schriften setzen und wünschen dann, daß die Überschriften und Fußnoten automatisch in der entsprechenden Größe in den entsprechenden PostScript-Schriften gesetzt werden. Wenn Sie den normalen Text mit Palatino schreiben, dann sollen die Überschriften natürlich in Palatino-Bold in der entsprechenden Größe gesetzt werden! Ebenso möchten Sie dann mit `\it` nicht auf die normalen TEX-Italics umschalten (das sähe in einer Palatino-Umgebung ganz grausig aus!), sondern es soll natürlich auf Palatino-Italics umgeschaltet werden. Und genau das alles geht sehr einfach, wie Sie sich als (nunmehr) NFSS-Kenner sicher schon gedacht haben.

Aber nur mit LATEX und NFSS! Wenn Sie absolut nicht mit NFSS arbeiten möchten, gibt es auch eine alte Version des DVIPS-Paketes, die das ohne NFSS ermöglicht. Wenden Sie sich in diesem Falle an mich. Zu Beginn eines jeden LATEX-Dokuments müssen Sie den Dokumentenstil mit dem Kommando `\documentstyle` festlegen. Das sieht dann z. B. so aus:

```
\documentstyle[11pt,german]{article}
```

Damit wählen Sie als Standardtextgröße eine 11-Punkt-Schrift, schalten auf deutsche Sonderkonventionen („“ usw.) und schalten letztendlich als Dokumentstil, der Seitenaufteilung u. ä. bestimmt, auf „`article`“. Wenn Sie diese Zeile so modifizieren:

```
\documentstyle[11pt,german,palatino]{article}
```

dann haben Sie es schon geschafft: Alle Textschriften in allen Größen werden durch Palatino-Schriften ersetzt, mit `\bf` schalten Sie auf Palatino-Bold, mit `\it` auf Palatino-Italic, mit `\sl` auf Palatino-Slanted, mit `\sc` auf Palatino-SmallCaps usw. Mit den Befehlen `\large`, `\tiny`, ... schalten Sie auch zwischen unterschiedlichen Palatino-Schriftgrößen um.

Was steckt dahinter? Die Dokumentenstiloption `palatino` macht nichts weiter, als zunächst einmal **sämtliche** PostScript-Schriften für das NFSS zu definieren[30]. Danach wird einfach die Familie auf `palatino` umgeschaltet. Das gleiche gibt es auch noch für die Times-Schriften, hier geben Sie als Dokumentenoption einfach '`times`'. Natürlich geht es auch noch weiter, für Helvetica-Schriften, für Avantgarde-Schriften usw., siehe in den Tabellen 5 – 10. Man gibt also einfach eine dieser Stiloptionen ein und hat dann sämtliche PostScript-Schriftfamilien zur Verfügung. Sie können also (nachdem Sie z. B. `PALATINO.STY` eingebunden haben) auch problemlos das Kommando `\family{avantgarde}\selectfont` geben und schreiben dann halt in Avantgarde weiter, wobei natürlich `\bf` **nun auch auf fette Avangarde umschaltet**. Was im einzelnen worauf umschaltet sehen Sie in Tabelle 4 – 10. In Tabelle 4 ist dargestellt, welche zusätzlichen, vereinfachten Umschaltkommandos es nach Einbindung einer der Dokumentenstiloptionen gibt[31]. Diese Kommandos, die ja nur Abkürzungen für die

[30] Das geschieht durch Aufruf der Style-Datei `PSFONTS.STY`, dort stehen alle notwendigen NFSS-Definitionen drin.

[31] **Warnung bezüglich des `\bf`-Befehles!** Normalerweise (Standard-LATEX) schaltet der Befehl `\bf` auf `\shape{bx}`. Für die PostScript-Schriften wurde er aber auf die Wirkung `\shape{b}` umgestellt, da die PostScript-Schriften in unserem Paket noch nicht als BoldExtended-Schriften zur Verfügung stehen (man kann sie sich natürlich ganz leicht nachdefinieren). Wenn Sie nun noch mit der CMR-Familie schreiben und auf `\bf` schalten, sieht das etwas anders aus als sonst. Sie müssen (wenn die Verwendung von CMB anstelle von CMBX nicht erwünscht ist) das Kommando `\bf` dann eventuell so zurückdefinieren:
`\renewcommand{\bfdefault}{bx}`

etwas längeren NFSS-Kommandos darstellen, sind Vorschläge, mit denen ich persönlich gut auskomme. Wenn Sie Ihnen nicht gefallen, können Sie sie problemlos umdefinieren. Die Definitionen befinden sich am Ende der Datei `PSFONTS.STY`. In Tabelle 4 – 10 sehen Sie, wie Sie nach Einbindung eines der Styles über Schriftumschaltung die einzelnen zur Verfügung stehenden PostScript-Schriften erreichen können. Dabei wird immer davon ausgegangen, daß Sie sich bei der normalen Textschrift befinden. Natürlich können Sie jederzeit die Familie wechseln (und damit die Tabelle). In der ersten Spalte finden Sie den Namen der Schrift (gemäß dem Schriftenkatalog am Anfang dieses Kapitels), in der zweiten Spalte die Einordnung dieser Schrift (Zugehörigkeit zu einer Schriftfamilie, `series`-Kennzeichnung der Schrift, und `shape` der Schrift) und in der dritten Spalte, über welches Schriftumschaltungskommando die Schrift zu erreichen ist, ausgehend von der normalen Textschrift.

Kommando	Wirkung	Zweck
`\norm`		Alle Attribute zurücksetzen und auf Normalschrift (wie standardmäßig definiert) schalten
`\rm`	`\family{`*Standardfamilie*`}`	Auf die eingestellte Standardfamilie zurückschalten (**Achtung:** Nur die Familie wird geändert, es werden keine Attribute zurückgesetzt!)
`\bf`	`\series{b}`	Auf fette Schrift umschalten
`\it`	`\shape{it}`	Auf Kursiv bzw. (wenn Kursiv nicht vorhanden) Slanted umschalten
`\sl`	`\shape{sl}`	Auf Slanted bzw. (wenn Slanted nicht vorhanden) Kursiv umschalten
`\sc`	`\shape{sc}`	Auf SmallCaps-Schrift umschalten
`\sf`	`\family{helvetica}`	Auf eingestellte SansSerif-Schrift (in der Regel Helvetica) umschalten (Achtung: Ausnahmen!)
`\tt`	`\family{courier}`	Auf eine äquidistante Typewriter-Schrift (in der Regel Courier) umschalten
`\uit`	`\shape{u}\series{m}`	Auf *normale* UnslantedItalics schalten (soweit vorhanden!)
`\ex`	`\shape{n}\series{x}`	Auf *normale* Extended-Schrift schalten (soweit vorhanden!)
`\buit`	`\shape{u}\series{b}`	Auf *fette* UnslantedItalics schalten (soweit vorhanden!)
`\bit`	`\shape{it}\series{b}`	Auf *fette* Italics schalten
`\cond`	`\shape{n}\series{c}`	Auf *normale* Condensed-Schrift schalten (soweit vorhanden!)

Tabelle 4: Schriftumschaltkommandos nach Einbindung eines PostScript-Schriftstyles

Nicht als Tabelle aufgeführt ist die Wirkung von `CHANCERY.STY`. Hier werden zwar auch alle PostScript-Schriften für NFSS definiert und dann die Familie auf `chancery` umgeschaltet, jedoch gibt es die ZapfChancery-Schrift nur in einer Ausfertigung, d. h. sämtliche Schriftum-

In dem Falle erhalten Sie dann bei Verwendung von `\bf` mit einer PostScript-Schrift zwar eine Warnung (`Font not defined in this series` oder sinngemäß), es wird aber auf das nächstliegende zu den gewünschten BoldExtended geschaltet, sprich also: auf Boldface. Die Umdefinition von `\bf` wurde im wesentlichen vorgenommen, um die Anzahl der Warnungen zu verringern.

Schrift	Definiert als	Erreichbar mit	Schrift	Definiert als	Erreichbar mit
times	times,m,n	`\norm`	timesb	times,b,n	`\bf`
timesbi	times,b,it	`\bit`	timesc	times,c,n	`\cond`
timescsc	times,m,sc	`\sc`	timesi	times,m,it	`\it`
timessl	times,m,sl	`\sl`	timesx	times,x,n	`\ex`
helv	helvetica,m,n	`\sf`	cour	courier,m,n	`\tt`

Tabelle 5: Schriftschema nach Einbindung von `TIMES.STY`

Schrift	Definiert als	Erreichbar mit	Schrift	Definiert als	Erreichbar mit
helv	helvetica,m,n	`\norm`	helvb	helvetica,b,n	`\bf`
helvbi	helvetica,b,it	`\bit`	helvbi	helvetica,b,sl	`\bf\sl`
helvi	helvetica,m,sl	`\sl`	helvi	helvetica,m,it	`\it`
helvcsc	helvetica,m,sc	`\sc`			
helvc	helvetica,c,n	`\cond`	helvcb	helvetica,bc,n	`\cond\bf`
helvcbi	helvetica,bc,it	`\cond\bf\it`	helvcbi	helvetica,bc,sl	`\cond\bf\sl`
helvci	helvetica,c,it	`\cond\it`	helvci	helvetica,c,sl	`\cond\sl`
avant	avantgarde,m,n	`\sf`	cour	courier,m,n	`\tt`

Tabelle 6: Schriftschema nach Einbindung von `HELVETIC.STY`

Schrift	Definiert als	Erreichbar mit	Schrift	Definiert als	Erreichbar mit
bookl	bookman,m,n	`\norm`	bookd	bookman,b,n	`\bf`
bookdi	bookman,b,it	`\bit`	bookdi	bookman,b,sl	`\bf\sl`
bookli	bookman,m,it	`\it`	bookli	bookman,m,sl	`\sl`
booklcsc	bookman,m,sc	`\sc`			
helv	helvetica,m,n	`\sf`	cour	courier,m,n	`\tt`

Tabelle 7: Schriftschema nach Einbindung von `BOOKMAN.STY`

Schrift	Definiert als	Erreichbar mit	Schrift	Definiert als	Erreichbar mit
cent	ncs,m,n	`\norm`	centb	ncs,b,n	`\bf`
centbi	ncs,b,it	`\bit`	centbi	ncs,b,sl	`\bf\sl`
centi	ncs,m,it	`\it`	centi	ncs,m,sl	`\sl`
centcsc	ncs,m,sc	`\sc`			
helv	helvetica,m,n	`\sf`	cour	courier,m,n	`\tt`

Tabelle 8: Schriftschema nach Einbindung von `NCS.STY` (`NCS` = NewCenturySchoolbook)

Schrift	Definiert als	Erreichbar mit	Schrift	Definiert als	Erreichbar mit
pal	palatino,m,n	`\norm`	palb	palatino,b,n	`\bf`
palbi	palatino,b,it	`\bit`	palbu	palatino,b,u	`\buit`
palc	palatino,c,n	`\cond`	palcsc	palatino,m,sc	`\sc`
pali	palatino,m,it	`\it`	palsl	palatino,m,sl	`\sl`
palu	palatino,m,u	`\uit`	palx	palatino,x,n	`\ex`
helv	helvetica,m,n	`\sf`	cour	courier,m,n	`\tt`

Tabelle 9: Schriftschema nach Einbindung von `PALATINO.STY`

Schrift	Definiert als	Erreichbar mit	Schrift	Definiert als	Erreichbar mit
avant	avantgarde,m,n	`\norm`	avantd	avantgarde,b,n	`\bf`
avantdi	avantgarde,b,it	`\bit`	avantdi	avantgarde,b,sl	`\bf\sl`
avanti	avantgarde,m,it	`\it`	avanti	avantgarde,m,sl	`\sl`
avantcsc	avantgarde,m,sc	`\sc`			
helv	helvetica,m,n	`\sf`	cour	courier,m,n	`\tt`

Tabelle 10: Schriftschema nach Einbindung von `AVANTGAR.STY`

schaltkommandos (`\bf`, `\it` usw.) führen zu Warnungen.

Das hört sich alles sehr einfach an, ist es auch. Es gibt nur leichte Probleme: Die PostScript-Schriften brauchen beim Übersetzen der TEX-Dateien zusätzlichen Speicherplatz, der von emTEX explizit angefordert werden muß. NFSS ist leider (bei Verwendung mehrerer Schriftfamilien) ein extremer Speicherfresser. In der Regel werden Sie keine Probleme haben, und können weiter zum Übersetzen einer deutschen LATEX-Datei das emTEX-Kommando `GLATEX` verwenden. Wenn es jedoch Probleme gibt, und die Übersetzung mit dieser Fehlermeldung abgebrochen wird:

```
TFM-file not loaded, not enough room left!
```

(das ist eine Beschwerde von TEX über mangelnden Speicher), dann versuchen Sie es einmal mit dem Kommando `PSLATEX`, bzw. schauen Sie sich zunächst einmal die Datei `PSLATEX.BAT` im Verzeichnis `\EMTEX` an.

Wenn das immer noch nicht reicht und emTEX für die Übersetzung noch mehr Speicherplatz haben möchte, dann müssen Sie leider mit BigTEX arbeiten (also das Kommando `GBLATEX` in einer Standard-emTEX-Installation verwenden). Das geht dann deutlich langsamer, aber es geht wenigstens. Überhaupt keine Speicherplatz- und Geschwindigkeitsprobleme haben Sie natürlich bei Verwendung von emTEX386; in dem Falle vergessen Sie einfach, daß es solche Probleme überhaupt eventuell geben könnte.

7.4.4 Mathematischer Formelsatz mit PostScript-Schriften

Hier gibt es, wie Sie sich sicher schon gedacht haben, Probleme. Es ist zwar (siehe NFSS-Kapitel) ohne weiteres mit NFSS möglich, die mathematischen Sonder- und sonstigen Zeichen auf eine andere Schrift umzustellen, vorausgesetzt, man hat eine solche. Die Symbolschriften von PostScript enthalten zwar eine Menge Zeichen, die es auch in den Mathematik-Zeichensätzen von TEX gibt, aber eben nicht alle. Es gibt natürlich auch einen kompletten Satz von mathematischen Sonderschriften als PostScript-Schriften (die Lucida-Schriften), und für die gibt es auch entsprechende Styles, um sie mit NFSS unter TEX nutzbar zu machen[32], aber es ist mit Ihnen, wie mit vielen PostScript-Schriften: Man will dafür Geld haben. Sie stehen auch nicht auf den zentralen PostScript-Druckern am HRZ zur Verfügung.

Aber man kann ja aus dem vorhandenen das Beste machen: Zum DVIPS-Paket gehört ein Style namens `PSMATH.STY`, der zumindest die in den PostScript-Symbolschriften vorhandenen

[32] Diese Styles gehören zwar nicht zum DVIPS-Paket, aber Sie können sie am HRZ bekommen.

Zeichen als mathematische Sonderzeichen im TEX-Formelsatz einsetzt. Nicht vorhandene Zeichen werden dann aus dem Reservoir der originalen, mathematischen Zeichensätze von TEX „aufgefüllt“.

Das Ergebnis sieht nicht sehr überzeugend aus, denn die Mischung zweier Zeichensätze vollkommen unterschiedlichen Charakters (und sei es auch nur im Formelsatz) liefert generell selten einen überzeugenden optischen Eindruck und bietet (speziell in diesem Fall) keinen auch nur annähernd gleichwertigen Ersatz zur Verwendung eines einzelnen, einheitlichen Symbolzeichensatzes, also entweder des kompletten, gewohnten TEX-Symbolzeichensatzes oder des Satzes der Lucida-Schriften. Ich rate daher von der Verwendung des Styles `PSMATH.STY` ab. Wenn Sie allerdings in Ihren Formeln z. B. nur die griechischen Buchstaben und Sonderzeichen verwenden, die auch im PostScript-Symbolzeichensatz vorhanden sind (die finden Sie in jeder Schriftdemo dieses Zeichensatzes), was bei einer begrenzten Menge an Mathematik im Text ohne weiteres möglich ist, kann die Verwendung von `PSMATH.STY` doch eher angeraten werden.

8 Erfahrungen mit DVIPS

Hier soll es vor allem um Hardware-Anforderungen gehen. DVIPS würde im Prinzip auf einem einfachen PC mit 640 KB laufen. Aber richtig gut läuft es erst auf einem AT mit mindestens 1 MB. Die `.EXE`-Dateien sind auch nur für einen 80286-Prozessor (und alles, was noch größer ist...) generiert, worauf ich aber schon hinwies. Da leider bei DVIPS alle überhaupt im Dokument verwendeten Schriften gleichzeitig im Speicher gehalten werden müssen, stellt der verfügbare Speicherplatz eine Begrenzung für die gleichzeitig verwendbaren Schriften dar (das ist bei den emTEX-DVI-Treibern ja auch nicht anders). DVIPS nimmt sich radikal alles, was es an EMS-Speicher kriegen kann. Wenn Sie also Erweiterungsspeicher für DVIPS nutzbar machen möchten, so müssen Sie den Speicher als EMS-Speicher nach LIM 4.0 konfigurieren[33]. Sie sollten Ihrer Festplatte bei Verwendung von DVIPS übrigens mindestens 200 KB Cache-Speicher spendieren, ansonsten wird es nahezu unerträglich langsam. Der Cache-Speicher bringt eine Geschwindigkeitssteigerung um Faktor 3 bis 5. Der Speicher für den Cache geht dann natürlich von dem für Schriften zur Verfügung stehenden Speicher ab. Im Prinzip gilt: Eine Konfiguration, mit der Sie emTEX vernünftig betreiben können (12 MHz AT, 1 MB) ist auch für DVIPS ausreichend. Ich hatte noch keine Probleme mit zu wenig Speicher für PostScript-Schriften (außer bei diesem Dokument, in dem ja alle überhaupt existierenden PostScript-Schriften gleichzeitig verwendet wurden; das ließ sich erst mit 4 MB RAM in einem Durchlauf nach PostScript übersetzen).

Ein Tip, wenn Sie Probleme mit zu wenig Speicher für zu viele Schriften haben: In der Regel werden Sie nicht alle Schriften Ihres Dokumentes auf der gleichen Seite verwenden. Nutzen Sie also die Optionen '`p`' und '`l`' beim Aufruf von DVIPS und übersetzen Sie Ihr Dokument in einzelnen Seiten (oder kleineren Teilen) nach PostScript. Denn während eines Übersetzungslaufes werden natürlich nur die Schriften im Speicher gehalten, die auch gebraucht werden! Dann haben Sie hinterher zwar mehrere PostScript-Dateien, aber das hat ja auch den Vor-

[33] In der ausgelieferten Version sollte DVIPS auch Zusatzspeicher erkennen und verwenden, wenn er mit dem Windows-XMS-Treiber konfiguriert wurde, aber manchmal klappt das nicht (ich weiß auch nicht wieso!). In dem Falle sollten Sie dann doch auf die Verwendung eines EMS-Treibers umsteigen, wenn Sie in Speicherplatznöte kommen. Ein EMS-Treiber gehört zum Lieferumfang des DVIPS-Paketes.

teil, daß die einzelnen PostScript-Dateien kleiner sind und eventuell leichter transportabel (Diskettengröße).

Für die Ausgaben von DVIPS benötigen Sie, wie schon mehrfach erwähnt, einen PostScript-Drucker. Die am HRZ installierten Geräte vom Typ QMS-PS 810 machten bislang bei normalen Anforderungen noch keine Probleme, auf PostScript aufgerüstete HP-Laserjet II ebensowenig. Für extreme Anforderungen (viele Bilder oder viele Schriften) kann es sein, daß eine Seite mal 20 Minuten braucht, bis sie rauskommt oder daß auch mal eine Seite die vorhandenen Möglichkeiten (Speicher u.ä.) sprengt. Bislang haben wir aber mit etwas Geduld und Herumprobieren noch jede PostScript-Datei ausdrucken können, sogar die, auf der diese Anleitung beruht. Wenn Sie einen kommerziellen PostScript-Interpreter benutzen, kann ich aus eigener Erfahrung berichten, daß es mit „Freedom of Press“ in der Version 2.0 keine Probleme gibt. Bei Verwendung von GoScript werden Sie bei älteren Versionen Probleme bekommen, da diese nur die Version 1.0 der Sprache PostScript verstehen. DVIPS erzeugt aber PostScript-Ausgaben in der PostScript-Version 2.0[34] (nach ADOBE). Auf GhostScript werde ich ja noch ausführlich eingehen.

Noch eine kleine Besonderheit von PostScript-Schriften: Manchmal passen leider die Design-Größen von verschiedenen Schriften nicht zueinander. Dieser Absatz ist z. B. in einer 11-Punkt-Times-Normalschrift gesetzt. Und das hier ist eine 11-Punkt-Helvetica-Normalschrift. Sie sehen (vielleicht): Die Schriften passen in der gleichen Punktgröße in ihrer Erscheinung nicht ganz zueinander[35]. Diese leichten „Inkompatibilitäten“ können Sie nur durch Ausprobieren herausfinden. Aber ganz so schlimm ist das auch nicht. Es gibt halt Schriften, die im Grunddesign größer, und welche, die kleiner sind. Sie werden es ja merken. Die Helvetica-Schrift läßt sich aber ganz einfach in Ihrer Größe (ja, dies ist die in der Größe angepaßte Helvetica-Schrift!) an die 11-Punkt-Times anpassen, wenn man sie etwas kleiner wählt.

---

[34] Das bedeutet aber nicht, daß etwa PostScript-Bilder, die dem Sprachstandard höherer Versionen entsprechen, nicht auch eingebunden werden können. So etwas ist problemlos möglich, vorausgesetzt, Ihr Drucker versteht PostScript, Sprachversion 3.0. Es bedeutet lediglich, daß sich DVIPS auf den Sprachumfang der PostScript-Version 2.0 beschränkt.

[35] Obwohl diese Ansicht fast schon dem rein subjektiven Geschmack zuzurechnen ist.

9 Das Einbinden von Bildern

Hier will ich nun den Weg vom Bild in irgendeiner Form zum Bild im TEX-Dokument erläutern. Es gibt der Wege viele, und ich will mich bemühen, möglichst alle wenigstens zu erwähnen, werde aber nur wenige detailliert beschreiben. Als Grundregel für das Einbinden von Bildern mit dem Programm DVIPS müssen Sie sich nur merken, daß das Bild auf irgendeine Weise in PostScript-Form gebracht werden muß. Wie das geht, will ich im weiteren erklären.

Zunächst aber ein paar kurze Vorabbemerkungen über andere Möglichkeiten, Bilder ohne das Programm DVIPS einzubinden (solche gibt es natürlich). Bei den meisten Verfahren, Bilder (speziell Rasterbilder[36]) in TEX-Dokumente einzubinden, wird der Weg über sogenannte `\special`-Befehle von DVI-Treibern gewählt. Das geht in der Regel sehr elegant, und auch die emTEX-DVI-Treiber, die Sie ja zweifellos bisher verwendet haben, können das. Dazu muß das Bild nur als Rasterbild im passenden Rasterbildformat (es gibt da halt sehr viele, siehe dazu auch in der Anleitung für das Programm ALCHEMY, auf das ich später noch zu sprechen komme) vorliegen. So einfach und elegant diese Lösung auch scheinen mag, sie hat ihre Tücken. Stellen Sie sich vor, Sie müssen plötzlich ohne emTEX (oder ohne den speziellen DVI-Treiber, den Sie bisher verwendeten und der seine eigenen `\special`-Befehle versteht) auskommen, weil die lokale TEX-Installation, an der Sie gerade arbeiten, nicht auf PCs sondern auf einem Großrechner untergebracht ist. Die typische Eigenart der `\special`-Befehle ist es, nur von einem speziellen DVI-Treiber (oder nur von einer speziellen DVI-Treiber-Familie) verstanden zu werden, wie das auch bei den emTEX-Treibern der Fall ist. Also, der Weg Bilder über `\special`-Befehle einzubinden, ist (wenn es denn auf Portabilität der TEX-Dokumente ankommt) nicht das Gelbe vom Ei. Deshalb wurden Programme entwickelt, Rasterbilder in Schriftdateien (die `.pk`-Dateien von TEX) zu konvertieren. Solche Programmpakete sind zum Beispiel BM2FONT von Friedhelm Sowa und **rumgraph** von H.-W. Kisker. Allerdings besteht auch hier wieder die Einschränkung, daß beide Konvertierungsprogramme nur unter MS-DOS und OS/2 laufen, also auch wieder nur für den PC-Bereich gedacht sind. Aber immerhin: Wer diese Lösung zur Bildeinbindung wählt und seine Bilder einmal in `.pk`-Dateien konvertiert hat, kommt damit dann bei jeder TEX-Installation durch. Beide Programmpakete sind am HRZ erhältlich. Wenn ich noch meinen „persönlichen Senf" dazu abgeben darf: Dieser Weg ist zwar sicher (`.pk`-Dateien können mit jeder TEX-Installation verarbeitet werden und funktionieren auf jedem Ausgabegerät, sowohl daheim am Nadeldrucker als auch in der Druckerei am Laserbelichter), aber sehr umständlich und in der Durchführung ziemlich aufwendig. Auch wenn das Bild als Vektorbild vorliegt, gibt es einen TEX-konformen Weg zur Einbindung: Es gibt Programme (auch diese gibt es kostenlos am HRZ), die ein Vektorbild in eine METAFONT-

[36] Eine kleine Computergrafik-Fußnote: Es gibt grundsätzlich zwei verschiedene Möglichkeiten, Bildinformationen für den Computer „eßbar" zu machen: Zum einen rastert man das Bild in viele Punkte und gibt dann für jeden Punkt an, welche Farbe er haben soll. Dies ist die *Pixel-* oder *Rasterdarstellung* eines Bildes. Dieses Bildformat ist technisch am einfachsten zu handhaben, denn die Geräte, auf denen Bilder dargestellt werden sollen, sind ja auch punktrasterorientiert. Allerdings kann man sich leicht vorstellen, daß dieses Bildformat für Bilder, die ausschließlich Linien enthalten, nicht so sinnvoll ist. Hier wählt man besser die *Vektordarstellung* eines Bildes, in der nur angegeben wird, wo welche Linien in welcher Farbe zu ziehen sind (mal ganz grob erklärt). Das erspart einem dann, alle weißen Punkte auch noch mit anzugeben (auch wieder ganz grob gesagt, das Vektorformat hat noch andere Vorzüge und auch Nachteile). Mit PostScript kann man grundsätzlich beide Arten der Bilddarstellung verwenden. Man sollte von dieser Freiheit der Darstellungsart in PostScript dann auch Gebrauch machen, doch dazu später.

Datei wandeln. Aus dieser `.MF`-Datei muß man dann nur noch mit METAFONT eine `.pk`-Datei zu machen, und schon kann man das Bild einbinden. Auch in diesem Falle gilt wieder meine soeben zum Ausdruck gebrachte persönliche Meinung.

Einen sehr einfachen Weg zur Bildeinbindung gibt es ja auch in LaTeX: Hier können Sie mit LaTeX-Befehlen Bilder malen, im sogenannten `picture`-Environment. Das will ich hier aber nicht erläutern, denn es steht bestens beschrieben im LaTeX-Buch. Sehr viel weiter in dieser Richtung geht das Makropaket PicTeX, das auch noch das Schraffieren von Flächen und (dank noch einiger weiterer Befehle) die Anfertigung komplizierterer Zeichnungen ermöglicht. Es gibt (bei emTeX mitgeliefert) sogar ein mausgesteuertes Malprogramm (TeXCAD) zum Zeichnen von Bildern mit diesen Hilfsmitteln. Wenn man allerdings selber etwas kompliziertere Dinge zeichnen will, so empfehle ich eher ein richtiges Zeichen- oder Malprogramm (z. B. CorelDraw!), dazu gleich noch mehr. Für einfache Zeichnungen mit wenigen Linien ist die LaTeX-Bildumgebung mit den Erweiterungen, die sich bei Verwendung von emTeX oder PostScript oder PicTeX ergeben (siehe dazu auch im nächsten Unterkapitel) der richtige Weg (wenn das Bild erst noch gezeichnet werden muß. Wenn es natürlich schon in irgendeiner Form vorliegt, sollte man es auch in der Form verarbeiten).

Der von mir propagierte Weg der Bildeinbindung ist aber die Verwendung des `PSFIG`-Styles mit dem Programm DVIPS, und diesen Weg werde ich auch noch lang und breit erläutern. Auch dieses Verfahren geht zurück auf die Verwendung von `\special`-Befehlen, die ich ja gerade eben noch in allen ihren Nachteilen dargestellt habe. Warum soll es hier nun anders sein? Das hat im wesentlichen zwei Gründe: Zunächst einmal ist das Programm DVIPS sehr verbreitet. Im UNIX-Bereich wird man kaum eine TeX-Installation finden ohne DVIPS. Außerdem ist DVIPS als C-Source vorhanden und somit überall dorthin portierbar, wo es einen C-Compiler gibt (und den gibt es mittlerweile auch auf Großrechnern). Des weiteren ist DVIPS kein DVI-Treiber, der in der Lage ist, TeX-Dateien auf dem Drucker X aus der XY-Serie der Firma XYZ auszugeben, sondern DVIPS konvertiert DVI-Dateien nach PostScript, und PostScript kann prinzipiell auf jedem Drucker ausgegeben werden: Für den heimischen Nadeldrucker gibt es da spezielle Programme (siehe auch im Kapitel über GhostScript) und auch der Laserbelichter in der Druckerei versteht PostScript. Außerdem gibt es mittlerweile fast überall, wo Computer stehen, mit denen Textverarbeitung gemacht wird, auch normale Laserdrucker, die PostScript verstehen. Wenn man also seine Bilder mit dem hier vorgeschlagenen Verfahren einbindet, bewegt man sich auf zukunftssicherem Boden. Sollte es eines Tages mit einer neuen TeX-Version einen Standard zur Grafikeinbindung geben, so kann man mit Sicherheit davon ausgehen, daß dieser Standard den PostScript-Weg zur Bildeinbindung nicht außer Acht läßt. Also (wieder der eigene Senf): Der Weg der Bildeinbindung über die PostScript-Schiene ist einfach und sehr komfortabel (wie die meisten Verfahren zur Bildeinbindung, die über `\special`-Befehle laufen), aber zugleich in dem oben schon erklärten Sinne sicher, weil portabel.

Doch nun endlich zur Sache: Wie kommt das Bild in den Text? Dazu muß man zunächst das Bild in ein DVIPS-gerechtes Format bringen, nämlich ins PostScript-Format.

9.1 Der Weg zum PostScript-Bild

9.1.1 Wenn das Bild noch gar nicht vorliegt...

...sondern Sie es erst noch zeichnen müssen, dann gibt es verschiedene Möglichkeiten (wie immer). Wenn es nur ein einfaches Bild ist, dann sollten Sie sich überlegen, ob es sich nicht am schnellsten mit dem TEX-Zeichenprogramm TEXCAD malen läßt. Bekanntermaßen läßt die LATEX-`picture`-Umgebung ja nur Linien mit bestimmten Steigungen zu, was natürlich eine starke Einschränkung darstellt. Hier bietet entweder das Makropaket PicTEX Abhilfe (was hieße, mit Kanonen auf Spatzen zu schießen, aber vielleicht haben Sie auch Gefallen an den übrigen Möglichkeiten von PicTEX, sie können es selbstverständlich am HRZ bekommen[37]) oder die emTEX-DVI-Treiber ermöglichen ja auch Linien mit beliebiger Steigung. Siehe dazu auch in der emTEX-Anleitung, die entsprechenden Dateien sind `EMLINES.STY` und `EMLINES2.STY`. Siehe auch im Kapitel über DVIPS-Extras. Und natürlich: Auch mit Hilfe von PostScript-Befehlen, die mit der Verwendung von DVIPS einsetzbar werden, können Sie Linien beliebiger Steigung zeichnen. Dazu gibt es einen speziellen Style, nämlich `PSPIC.STY`. Lesen Sie sich bitte zuerst den Kommentar in diesem Style-File durch, bevor Sie ihn verwenden. Nachdem Sie diesen Style eingebunden haben (sprich: Sie haben ihn in die Dokumentenstiloptionen aufgenommen), unterliegen die LATEX-Befehle `\line`, `\vector`, `\circle` und `\oval` nicht mehr den üblichen Einschränkungen: Linien und Vektoren können beliebige Steigungen haben und Kreise beliebige Größen.

Falls Sie Bezier-Splines zeichnen wollen, genügt auch die Einbindung von `PSPIC.STY`. Darin sind alle notwendigen Definitionen für die Verwendung von Bezier-Splines enthalten, der Style `BEZIER.STY` ist dann überflüssig. Zu beachten ist bei der Verwendung von `PSPIC.STY` aber unbedingt, daß es als erste Dokumentenstiloption angegeben werden muß, also schreiben Sie an den Anfang der Datei nicht etwa

```
\documentstyle[11pt,german,pspic]{article}
```

sondern immer `PSPIC.STY` als erste Option angeben (wenn Sie es denn benutzen):

```
\documentstyle[pspic,11pt,german]{article}
```

Dann sollte es keine Probleme geben. Außer, daß Sie bei Verwendung von `PSPIC.STY` bereits `\special`-Befehle für DVIPS in Ihrer Datei drinstehen haben und sie somit nicht mehr mit dem emTEX-Preview ansehen können. Aber dazu siehe im Kapitel über GhostScript.

Wenn Ihr Bild jedoch etwas komplizierter ist, so empfiehlt es sich eher, das Bild mit einem richtigen Zeichen- oder Malprogramm zu erstellen, z. B. steht am HRZ das Zeichenprogramm „CorelDraw!“ zur Verfügung. Die meisten Zeichen- und Malprogramme bieten die Möglichkeit, die erstellten Bilder als PostScript-Vektorbild oder PostScript-Rasterbild abzuspeichern. In der Regel empfiehlt sich die Abspeicherung als PostScript-Vektorbild. Falls das von Ihnen verwendete Zeichen- oder Malprogramm seine Erzeugnisse nicht als PostScript-Bild abspeichern kann, müssen Sie eben ein anderes Format (Raster oder Vektor) wählen und dieses dann nach PostScript konvertieren.

[37] Leider können wir Ihnen nicht auch die Anleitung von PicTEX mitgeben, weil auf dieser ein Copyright besteht. Eine sehr gute Anleitung für PicTEX finden Sie aber auch im Buch LATEX-*Erweiterungsmöglichkeiten* von H. Kopka.

Oder aber sie zeichnen Ihr Bild mit der Hand und haben dann den Fall, der jetzt erläutert werden soll.

9.1.2 Wenn das Bild da, aber nicht im Computer ist...

...dann muß es eben irgendwie in den Computer gebracht werden, und da gibt es nur einen einfachen Weg: **Scannen!**. Ein Scanner ist ein Gerät, mit dem man Bilder von einer Vorlage in den Rechner übertragen kann (und noch so einiges mehr, aber darum geht es hier nicht). Am HRZ und auch an der HRZ-Außenstelle in der Universitätsbibliothek stehen Ihnen mehrere Scanner zur Verfügung. Erkundigen Sie sich nach der Bedienung und Verfügbarkeit bei den zuständigen HRZ-Mitarbeitern. Es gibt auch spezielle Workshops zur Einweisung in die Bedienung der Scanner. Darum will ich hier nicht weiter auf Details eingehen.

Die Scanner-Programme am HRZ sind alle in der Lage, das eingescannte Bild als PostScript-Rasterbild abzuspeichern, aber das ist manchmal gar nicht so sinnvoll. Wenn Sie nämlich eine reine Strichzeichnung eingescannt haben, ist es vorteilhafter, diese zunächst zu vektorisieren (dazu im nächsten Kapitel). Nachdem Sie ein Bild eingescannt haben, liegt es also entweder als PostScript-Bild in irgendeiner Form vor, oder...

9.1.3 Das Bild ist in irgendeiner Weise im Computer

Dann haben Sie den Regelfall und müssen dieses Bild ins PostScript-Format bringen. Hierzu bietet sowohl das HRZ als auch das vom HRZ herausgebrachte DVIPS-Paket reichlich Unterstützung. Darum will ich systematisch alle Fälle durchgehen:

Zunächst eine Vorabbemerkung: Das für ein Bild verwendete Vektor- oder Rasterbildformat läßt sich in der Regel an der Endung seines Dateinamens ablesen. Für MS-DOS gibt es da genormte Abkürzungen, die das Erkennen des verwendeten Dateiformates schon am Dateinamen ermöglichen sollen. Unter anderen Betriebssystemen, die unter Umständen längere Dateinamen als acht Zeichen + drei Zeichen Endung zulassen, werden meist auch diese Abkürzungen verwendet, sie variieren manchmal allerdings auch minimal. So wird etwa unter dem Betriebssystem Amiga-DOS aus der MS-DOS-typischen Endung `.TIF` die Endung `.TIFF`.

Zunächst zu dem wohl selteneren Fall, daß Ihr Bild in einem Vektorformat vorliegt. Solches sind zum Beispiel das GEM-Metafile-Format (Endung `.GEM`), das Win-Metafile-Format (Endung `.WMF`), das HPGL-Format (Endung `.PLT`, das ist wohl das verbreitetste) oder das Vektorformat für AutoCAD-Zeichnungen (Endung `.DXF`). Wer soll da noch den Durchblick behalten? Dieser Formatzoo ist jedoch noch gar nichts gegen die Unzahl der Formate, die es für Rasterbilder gibt. Wenn also dieser Fall eintritt, dann muß das Bild aus dem Vektorformat ins PostScript-Vektorformat konvertiert werden und dazu sind mir nur kommerzielle Programme bekannt, die auch am HRZ zur Verfügung stehen (da möchte ich vor allem auf die Vollversion des Programmes ALCHEMY und das Programm HiJaak hinweisen). Erkundigen Sie sich also bei Bedarf.

Das Bild liegt aber aller Wahrscheinlichkeit nach als Rasterbild vor. Da gibt es viele verschiedene Formate (die sich durch die Endungen `.TIF`, `.GIF`, `.PCX`, `.MSP`, `.BMP`, `.TGA`, `.IFF`, `.LBM`, `.PBM`, `.JPG`, `.PIC`, `.HSI`, `.PCL`, um nur einmal einige zu nennen, identifizieren lassen[38])

[38] Diese Liste entspricht exakt den Bildformaten, die Sie mit dem Programm ALCHEMY ins PostScript-