anführen: Sie können nachzählen und werden feststellen, daß in diesem Kapitel das Bild der Simpson-Familie dreimal auftaucht. Dieses Bild braucht, das hatte ich erwähnt, 450 KB Speicherplatz. Außerdem tritt das Wappenbild auf, das 160 KB in Anspruch nimmt. Wenn Sie das Programm DVIPS auf die DVI-Datei für dieses Kapitel bis zu dieser Textstelle loslassen, so wird eine PostScript-Datei mit Ausmaßen von etwa 1500 KB erzeugt. Wenn Sie das auf Ihrem heimischen PC machen und die Datei anschließend auf Diskette zu einem PostScript-Drucker transportieren müssen, können Sie da schon in Probleme kommen. Allerdings muß man sagen, daß die vom Programm DVIPS auf diese Weise erzeugten PostScript-Dateien sehr gut packbar sind. Wenn Sie einen Standardpacker wie zum Beispiel LHarc (der sich mit im DVIPS-Paket des HRZ befindet) darauf loslassen, schrumpft Ihnen die PostScript-Datei für dieses Kapitel auf weniger als 500 KB zusammen. Wenn die Datei allerdings auch gepackt nicht mehr auf eine Diskette paßt (das ist mir durchaus schon passiert), dann müßten Sie sie entweder in mehrere kleinere Dateien aufteilen (sie übersetzen die DVI-Datei dann eben seitenweise mit dem Programm DVIPS) oder aber die PostScript-Datei direkt auf dem an den PostScript-Drucker angeschlossenen Rechner erzeugen. Sie sollten dann allerdings auch etwas Geduld mitbringen: Für den Ausdruck der Seite, auf der das Wappenbild ist, hat unser PostScript-Drucker etwa 5 Minuten gebraucht. Speziell der Ausdruck vektorisierter Bilder nimmt recht viel Zeit in Anspruch. Da Sie aber in der Regel ohnehin den zentralen PostScript-Drucker am HRZ-Zentralrechner Cyber 960 benutzen werden, spielt das ja wahrscheinlich keine so große Rolle.

Es gibt allerdings eine Abhilfe, die unbedingt noch erwähnt werden sollte. Die PostScript-Dateien werden ja nur so unglaublich riesig durch die eingebundene Bildinformation. Wenn Sie nun einmal gesehen haben, wie ihr Bild auf dem Papier aussieht (oder Sie das zunächst einmal überhaupt nicht interessiert) und eigentlich nur noch am Text oder am Layout herumfeilen möchten, so würde bei den notwendigen Probeausdrucken eine weiße, freigelassene Fläche von exakt den Abmessungen des Bildes das Bild selber voll und ganz ersetzen können. Außerdem könnte man sich dann die DVI-Datei ja auch wieder mit dem emTEX-Preview anschauen[49] und bräuchte nicht auf GhostScript oder einen zeit- und papierraubenden Ausdruck ausweichen. Diese Möglichkeit gibt es durchaus. Der Style `PSFIG` bietet da zwei weitere zusätzliche Kommandos, nämlich

`\psfull` und `\psdraft`.

Wenn Sie irgendwo im Text das Kommando `\psdraft` geben (zum Beispiel ganz am Anfang des Textes, direkt hinter `\begin{document}`), so wird, wenn Sie dann im weiteren ein Bild einbinden wollen, das hier passieren:

```
\psdraft
\centerline{\psfig{figure=d:/anl/portrait.eps,height=4cm}}
```

[49] Vorausgesetzt natürlich, es sind auch sonst keine PostScript-Effekte oder -Schriften enthalten!

d:/anl/portrait.eps

Kein Bild! Nur eine **umrahmte (!)** weiße Fläche von genau den Ausmaßen des Bildes, in der außerdem noch angegeben wird, welches Bild hier stehen sollte. Da das betreffende Bild 1100 KB Speicherplatz benötigt, wird dadurch die PostScript-Datei für dieses Kapitel um 1100 KB kürzer. Es genügt, das Kommando `\psdraft` einmal zu geben. Es gilt dann für alle nachfolgenden Bilder. Wie aber, wenn Sie anstelle der freien Flächen wieder die Bilder haben möchten? In dem Falle genügt das Kommando `\psfull`.

Die Kommandos `\psdraft` und `\psfull` wirken also wie Schalter, die die Einbindung der Bilder an- oder abschalten. Übrigens muß das Bild auch dann vorhanden sein, wenn Sie nur im `\psdraft`-Modus arbeiten, denn die BoundingBox-Angabe aus der PostScript-Bilddatei wird in jedem Falle benötigt. Also, zum Abschluß noch eine Demonstration der Wirkung des Kommandos `\psfull`, um das eben mit `\psdraft` vermiedene Bild noch zu zeigen:

```
\psfull
\centerline{\psfig{figure=d:/anl/portrait.eps}}
```

Dieses Bild wurde einmal, um zu zeigen, daß das auch geht und wie gut eingescannte Fotos aussehen können, wenn man alles richtig macht, in seiner Originalgröße dargestellt. Es wurden keinerlei Größenangaben gemacht.

10 DVIPS-Extras

Wozu dieses Kapitel? Hier will ich über einige Möglichkeiten des Programms DVIPS berichten, die sich anderswo schlecht unterbringen lassen. Zunächst will ich da auf die *Special-Effects* zu sprechen kommen:

10.1 PostScript-Effekte

Mit DVIPS ist es im Prinzip möglich, innerhalb einer TEX-Datei beliebige PostScript-Kommandos zu geben. Dies geht über den Befehl

`\special{ps:` <*beliebige PostScript-Kommandos*>`}`

Wenn Sie sich nun mit PostScript auskennen, können Sie mit diesem Kommando lustige Dinge veranstalten. Leider müssen Sie da schon sehr genau wissen, was Sie tun und sollten die PostScript-Sequenzen immer in ein Save/Restore-Pair einschließen. Wenn Sie jetzt nicht wissen, was ich meine, wäre ein Beispiel für Sie auch nicht weiter hilfreich und wenn Sie genau wissen, was ich damit meine, brauchen Sie kein Beispiel. Also lasse ich es weg und verweise für näheres auf die originale DVIPS-Anleitung. Sie können natürlich auch die beiden Style-Dateien `ROTATE.STY` und `OUTLINE.STY`, die einige PostScript-Effekte durch dieses `\special`-Kommando in LATEX leicht zugänglich machen, als lehrreiche Beispiele nehmen.

Es gibt also einige PostScript-Effekte, die Sie ohne weiteres Wissen über PostScript benutzen können. Einige sind in einer speziellen Style-Datei namens `ROTATE.STY` enthalten. Es handelt sich dabei konkret um den Befehl `\rotate`:

`\rotate[l/r/u/f]{`<*irgendwas*>`}`

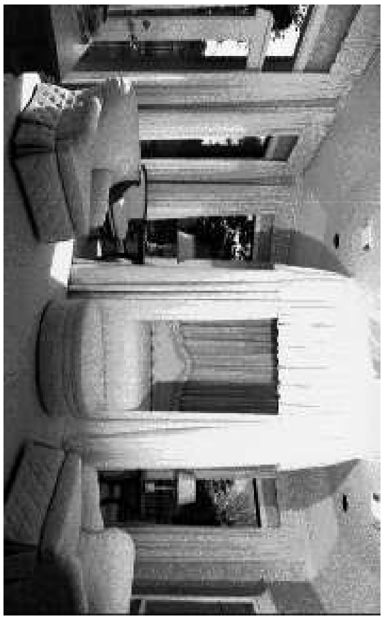
Wie der Name dieses Befehls schon sagt, dient er dazu, etwas zu drehen, und zwar eine TEX-Box. Der erste Parameter des Befehls (`l`, `r`, `u` oder `f`) besagt, wie etwas gedreht wird: `l` bedeutet linksrum, `r` rechts herum, `u` stellt die Sache auf den Kopf und `f` bewirkt eine spiegelverkehrte Darstellung. Am besten sieht man es wohl am Beispiel. Die Kommandos

```
\centerline{\rotate[l]{\hbox{Huch?}}}
\centerline{\rotate[r]{\hbox{Huch?}}}
\centerline{\rotate[u]{\hbox{Huch?}}}
\centerline{\rotate[f]{\hbox{Huch?}}}
```

bewirken das hier:

Huch?
Huch?
Huch?
Huch?

Damit dürfte die Wirkung nun schon mal klar sein. Beachten Sie bitte, daß Sie als zu drehendes Argument nur eine Box übergeben dürfen, sonst können da ganz komische Dinge passieren. Aber wie man aus Texten oder anderen Objekten Boxen macht, das schauen Sie am besten im TeXbook oder im LaTeX-Buch nach. Stellt sich die Frage, wozu es gut ist. Zum einen natürlich kann man damit solche unsinnigen Spielereien machen. Aber es gibt durchaus sinnvolle Anwendungen. Nehmen wir an, wir hätten dieses Bild hier:



Oh! Da hat wohl jemand beim Einscannen des Fotos nicht aufgepaßt. Was nun? Natürlich kann man das Bild, wenn man es noch als GIF- oder TIF- oder PCX- oder Sonstwas-Rasterbild vorliegen hat, mit GraphicWorkshop in die richtige Lage drehen. Was aber, wenn man es nur noch als EPI-Rasterdatei besitzt? In dem Falle macht man

```
\centerline{\rotate[l]{\psfig{figure=master.epi,height=12.0cm}}}
```

Wenn Sie nachmessen, werden Sie feststellen, daß das Bild jetzt 12 cm **breit** (nicht hoch) ist, was aber eigentlich klar sein sollte. Das war aber eine Platzverschwendung, dieses Beispiel hätte man ebenso gut nur mit Worten präsentieren können, aber dann glauben Sie mir womöglich nicht, daß so etwas Tolles so einfach zu machen ist. Es hindert Sie natürlich auch niemand daran, Ihre Abbildungen absichtlich spiegelverkehrt darzustellen...

Es ist auch nicht sonderlich schwer, den `\rotate`-Befehl so zu erweitern, daß um beliebige Winkel gedreht werden kann. Allerdings wird es dann problematisch, auszurechnen, welchen Platz das gedrehte Objekt denn in Anspruch nimmt (man braucht dazu, vielleicht erinnern Sie sich da an die Schulmathematik, trigonometrische Funktionen), und da mir das bis jetzt noch nicht so recht gelungen ist, habe ich es auch nicht eingebaut.

Mit PostScript-Schriften kann man noch einige Dinge machen, die weit über das, was im entsprechenden Kapitel gesagt wurde, hinausgehen. Eine Andeutung der Dinge, die möglich sind, befindet sich als PostScript-Effekt im Style `OUTLINE.STY`. Wenn Sie diesen in Ihre Dokumentenstiloptionen einbauen, haben Sie den Befehl `\outline{`<*irgendwas*>`}` zur Verfügung. Der funktioniert aber **nur mit einer PostScript-Schrift!** Schalten wir also auf eine PostScript-Schrift: `\family{palatino}\selectfont`Und um auch deutlich zu sehen, was der Effekt bewirkt, machen wir es fett und etwas größer: `\series{b}\selectfont\Large` **Das ist schon besser...** `\outline{Was gibt das?}`Was gibt das?

Der Effekt ist also, daß der mit `\outline` geschriebene Text nur noch als Kontur gezeichnet wird. Die Schrift wird nicht mehr ausgefüllt. Das sieht man natürlich nur gut bei fetten, großen Schriften. `\family{palatino}\selectfont`Bei einer normalen Textschrift sieht man es kaum.

Ein weiterer PostScript-Effekt ermöglicht es, das gesamte Dokument im Querformat auszugeben. Wenn Sie ganz an den Anfang Ihrer TEX-Datei das Kommando

```
\special{landscape}
```

setzen, wird das gesamte Dokument im DIN-A4-Querformat ausgegeben. Allerdings müssen Sie dann auch TEX die neuen Seitenabmessungen mitteilen, denn das Blatt ist jetzt nicht mehr 21 cm breit und 29.7 cm hoch, sondern gerade umgekehrt 21 cm hoch und 29.7 cm breit. Wenn das Dokument nicht auch von TEX entsprechend dieser Abmessungen formatiert wurde, kommt dabei im wesentlichen Unfug heraus. Leider gilt dieser Befehl für das gesamte Dokument und nicht nur für einige Seiten. Man kann es also nicht mehr abstellen. Aber das kann ja durchaus Sinn machen.

10.2 emTEX-Specials

Wenn Sie viel mit emTEX gearbeitet haben, werden Sie vielleicht wissen, daß die emTEX-DVI-Treiber einige `\special`-Befehle verstehen, so zum Beispiel Befehle zum Zeichnen von Linien mit beliebiger Steigung. Diese `\special`-Befehle von emTEX werden vom DVIPS-Programm ebenfalls verstanden und auch umgesetzt! Sie können also mit TEXCAD Bilder zeichnen unter Verwendung der emTEX-Specials (Dateien `EMLINES.STY` und `EMLINES2.STY`) und diese anschließend auch mit DVIPS verarbeiten.

11 GhostScript — Ein PostScript-Interpreter

11.1 Noch'n Vorwort und eine Installationsanleitung

Also, zunächst soll einmal wieder klargestellt werden, worum es hier geht. Zum DVIPS-Paket gehört auch das Archiv `GHOSTSCR.LZH`. In diesem Archiv ist GhostScript enthalten, und ich habe ja bereits mehrfach darauf verwiesen. Installiert wird GhostScript (wie auch alle anderen Teile des Paketes) indem Sie die Archivdatei `GHOSTSCR.LZH` auf die Festplattenpartition kopieren, auf der die ganze Sache dann dauerhaft installiert werden soll und dann das Kommando

```
lharc x ghostscr
```

geben (natürlich müssen Sie das Programm LHarc (also die Datei `lharc.exe`, die sich auf der ersten Diskette des DVIPS-Paketes befindet) dann auch „greifbar" haben). Die zu GhostScript gehörenden Dateien landen im Verzeichnis `GHOST` und einigen Unterverzeichnissen davon.

Ebenfalls gehören zu GhostScript eine ganze Menge Schriften. Diese GhostScript-Schriften (es sind schlechte Imitationen der Original-PostScript-Schriften, aber wenn Sie den originalen Adobe-Schriften zu ähnlich wären, würden sie auch Geld kosten) finden Sie auf einer der Disketten des DVIPS-Paketes im Archiv `GSFONTS.LZH`. Auch hier wird wieder ebenso einfach installiert wie bei GhostScript selber, die Schriften landen im Verzeichnis `\GHOST\FONTS`. Beachten Sie bitte, daß die Schriften in ausgepackter Form ca. 2400 KB Plattenplatz belegen. Eventuell benötigen Sie die Schriften aber auch gar nicht. Lesen Sie deshalb vor der Installation der Schriften erst einmal weiter.

Und was ist jetzt GhostScript? Das Programm DVIPS erzeugt als Ausgabe (in der Regel, wenn Sie die Ausgabe nicht explizit umlenken) eine PostScript-Datei. Um zu sehen, was in dieser PostScript-Datei nun eigentlich drinsteckt, müssen Sie sie zu einem PostScript-Drucker schicken, der Sie auf einem Blatt Papier ausdruckt (es sei denn, Sie sind in der Lage, aus den dem Leitungsrauschen nicht unähnlichen Zeichenfolgen des DVIPS-Outputs selber vor Ihrem geistigen Auge eine Druckseite aufzubauen. Das jedoch halte ich für schlicht unmöglich). Das ist, wenn Sie eigentlich nur mal eben sehen möchten, wo ein bestimmtes Bild gelandet ist, ob eine Überschrift richtig umgebrochen wurde o. ä., eine ziemliche Papierverschwendung, und wenn Sie daheim arbeiten und dort gar keinen PostScript-Drucker haben, sogar ein echtes Problem. Hier bietet nun GhostScript Abhilfe. Mit GhostScript können Sie eine PostScript-Datei am Bildschirm darstellen und auf einem Nadeldrucker ausgeben. Dabei gibt es allerdings einige kleine Einschränkungen:

- Die Bildschirmdarstellung funktioniert nur, wenn Sie mindestens eine EGA-Karte haben. Eine Hercules-Karte reicht nicht aus und ich werde auch auf Wunsch keine GhostScript-Version für eine Hercules-Karte erstellen (aber ich kann Ihnen die von mir modifizierten C-Sourcen für GhostScript überlassen, wenn Sie es selber tun wollen).
- Die Bildschirmdarstellung auf EGA-Karte konnte ich nicht besonders ausführlich testen (hatte keine EGA-Karte zur Verfügung...) und würde sie deshalb nicht als perfekt bezeichnen (der Rechner könnte schon mal abstürzen...). Perfekt (damit meine ich: ohne dauernde Abstürze) aber funktioniert die Darstellung mit einer beliebigen VGA-Karte.

- Sie brauchen schon an die 512 KB freies RAM, um GhostScript laufen zu lassen. Das sollte auch wirklich frei sein, also raus mit allen überflüssigen TSR-Programmen und ab mit dem DOS-Kern ins Upper-Memory. Zu empfehlen ist ebenfalls etwas freier EMS-Speicher, zwingend erforderlich aber sind nur die 512 KB freies DOS-RAM.

- Unter einem 80286-Prozessor ist gar nichts zu machen. Die Standard-Version von GhostScript ist für einen 80386-Prozessor, eine 80286er-Version habe ich aber auch noch dabei gelegt. Eine Coprozessor-Version können Sie auf Wunsch auch bekommen (bringt aber nicht so sehr viel...).

- Die Ausgabe auf Nadeldrucker funktioniert derzeit nur für EPSON-LQ-kompatible und (mit kleinen Macken, bei denen ich die Hoffnung auf Beseitigung inzwischen aufgegeben habe) NEC-Pinwriter.

- Bei der Verwendung von PostScript-Schriften stoßen Sie nach einigen wenigen Schriften bereits an Speichergrenzen, außerdem sind die Ergebnisse alles andere als schön. Darauf komme ich aber noch.

Wenn Sie jetzt bereits das Interesse an GhostScript verloren haben, kann ich es Ihnen kaum verdenken. Aber wenn es auch nicht optimal und so toll ist, wie man sich wünschen würde, es ist zumindest kostenlos (und dank vieler fehlender Features leicht portierbar).

11.2 Die Dateien von GhostScript

Wenn Sie GhostScript wie angegeben installiert haben, dann befinden sich folgende neue Dateien in Ihren Verzeichnissen:

- Im Verzeichnis `\GHOST`:
  - `GS.EXE`: Das eigentliche GhostScript-Programm für einen 80386-Prozessor. Wenn Sie einen 80286-Prozessor haben, können Sie diese Datei löschen.
  - `GS2.EXE`: Das eigentliche GhostScript-Programm für einen 80286-Prozessor. Wenn Sie einen 80386- oder 80486-Prozessor haben, können Sie diese Datei löschen.
  - `GS_INIT.PS`, `GS_STATD.PS`, `GS_FONTS.PS`, `SYMBOL_E.PS`, `GCONFIG.PS`, `FONTMAP`, `QUIT.PS`, `EMPTY.PS`: Initialisierungs- und Systemdateien für GhostScript. Weder löschen noch umbenennen!
  - `PFBTOPS.BAT` und `BFDTOPS.BAT`: DOS-Batch-Dateien, die Sie benötigen, wenn Sie mit GhostScript ADOBE-Type 1-PostScript-Schriften in GhostScript-Schriften konvertieren wollen oder sonstige diverse Konvertierungsaktionen mit GhostScript durchführen wollen.
  - `GSND.BAT` und `GSNDT.BAT`: DOS-Batch-Dateien zum Aufruf von GhostScript im No-Display-Modus: Es wird nichts auf dem Bildschirm angezeigt (z. B. für Ausdrucke).
  - `UGLYR.GSF`: GhostScript-Schrift, die als Ersatz für alle nicht vorhandenen Schriften benutzt wird. Häßlich, aber zweckmäßig. Keinesfalls löschen oder umbenennen, ohne diese Schriftdatei können Sie mit GhostScript nichts mehr anfangen.

- Ins Verzeichnis `\GHOST\PSGOODIE` kommen eine ganze Menge von PostScript-Anwendungs- und Demonstrationsdateien. Ich will hier einmal alle Dateien kurz dokumentieren. Die meisten Demonstrationsdateien erzeugen wirklich eine wunderschöne Ausgabe und es sieht ja auch so beeindruckend aus, aber nicht jeder Benutzer muß unsere zentralen PostScript-Drucker damit belasten. Wenn Sie mal ein Bild ausdrucken wollen ist das ja in Ordnung. Aber wenn jeder Benutzer alle Demo-Dateien ausdruckt, so wissen unsere armen PostScript-Drucker vor Arbeit nicht mehr ein noch aus. Die Demo-Datei `SNOWFLAK.PS` hält zum Beispiel einen unserer zentralen PostScript-Drucker vom Typ QMS-PS 810 über eine halbe Stunde lang von sinnvoller Arbeit ab. Bewundern Sie die Demo-Dateien am Bildschirm und unterlassen Sie möglichst den spaßeshalber oder nur aus Neugierde vorgenommenen Ausdruck dieser Dateien am HRZ. Also, zu den eigentlichen Dateien:

  – PostScript-Demos: Bis Sie Ihre eigenen Bilder im PostScript-Format vorliegen haben, können Sie ja einige dieser Demo-Bilder zum Herumspielen mit DVIPS verwenden. Beachten Sie bitte, daß es sich bei diesen Beispielen nicht um Dateien im EPI-Format handelt! In der Regel sollte die Einbindung mit DVIPS und dem Style PSFIG trotzdem keine Probleme bereiten (ich habe es aber nicht für alle ausprobiert!). Es handelt sich um vollwertige PostScript-Dateien, die Sie einfach so zum PostScript-Drucker schicken können[50].

    ∗ `ALPHABET.PS`: Gibt jeden PostScript-Zeichensatz komplett aus.
    ∗ `BULB.PS`: Malt eine Glühbirne. Wenig Effektvoll.
    ∗ `CAL2.PS`: PostScript-Datei zur Generierung schöner Wandkalender. Begrenzt nützlich. Datei vor dem Ausprobieren anschauen!
    ∗ `CALENDAR.PS`: Zur Generierung schlechter Wandkalender.
    ∗ `CHESS.PS`: Demonstration der PostScript-Schach-Schrift `CHEQ.PS`.
    ∗ `COLORCIR.PS`: Demonstration der Möglichkeiten von Farb-PostScript-Ausgabegeräten. Mit GhostScript und VGA-Karte toll, auf einem Schwarz-Weiß-PostScript-Drucker sieht man, wie Farbpaletten auf Graustufenpaletten umgesetzt werden.
    ∗ `ESCHER.PS`: Gibt ein M. C. Escher-Muster aus. Verwendet Farb-Befehle, sieht aber auch auf einem Schwarz-Weiß-PostScript-Drucker ganz gut aus. Dauert auf einem QMS-PS 810 ca. 25 Minuten.
    ∗ `FONTDEMO.PS`: Testet alle vorhandenen GhostScript-Schriften und gibt sie aus. Witzlos.
    ∗ `GOLFER.PS`: Zeichnet mit Vektorgrafik (unter Verwendung von Flächenvektoren (ausgefüllte Kreise u. ä.)) einen Golfer. Mal ausprobieren.
    ∗ `LIFETREE.PS`: Irgendeine Demo. Vielleicht versteht jemand das, was da rauskommt? Ich nicht (irgendwelche Tarot-Schemata?).
    ∗ `LINES.PS`: GhostScript Test- und Demodatei. Sinnvoll für GhostScript-Portierer, sonst witzlos.
    ∗ `LISTFNTS.PS`: Kann man sich ja wohl denken. Witzlos.
    ∗ `PALETTE.PS`: Gibt die komplette Palette eines Farb-PostScript-Ausgabegerätes aus. Nützlich, wenn man einen Farb-PostScript-Drucker hat.

[50] Mit EPI-Dateien geht das nicht so einfach. EPI-Dateien müßten vorher erst irgendwie eingebunden werden.

  * `PLTFILE.PS`: Kleine Demonstration zur Umsetzung von HPGL in PostScript. Gut als Demo-Bild zum Ausprobieren geeignet.
  * `PRFONT.PS`: Und noch ein Schriftkatalog.
  * `ROACHES.PS`: Zeichnet ein Blatt voller Käfer. Die verwendete Technik (die PostScript-Programmierung) dieser Demo ist sehr lehrreich.
  * `SNOWFLAK.PS`: Zeichnet ein effektvolles Farbbild unter exzessiver Verwendung des Zufallszahlengenerators (jedes Bild sieht völlig anders aus). Verlockend, es mal auf einem Drucker statt auf dem Bildschirm auszuprobieren. Doch Vorsicht, braucht auf einem QMS-PS 810 länger als eine halbe Stunde!
  * `SPIRO.PS`: Ein Beispiel aus dem PostScript-Lehrbuch von Adobe, das zeigt, wie man effektvoll eine eigene PostScript-Schrift gestalten kann.
  * `TESTQUAL.PS`: Ein Qualitätstest für Farb-PostScript-Ausgabegeräte.
  * `EXTRA.PS`: PostScript-Demodatei, die das Kapitel über DVIPS-Extras dieser Anleitung enthält. Wird im weiteren zur Erläuterung der GhostScript-Bedienung verwendet.

- PostScript-Anwendungen: Hier gibt es vor allem einige Programme um Adobe-Type-1-Schriften in GhostScript-Schriften umzuwandeln. Falls Sie das wirklich mal tun wollen, verweise ich auf die Lektüre der Kommentare in diesen Dateien und auf die GhostScript-Bedienungsanleitung. Ich werde die hierzu gehörenden Dateien im folgenden einfach mit „Schriftkonversionsdatei“ kennzeichnen und nicht weiter darauf eingehen.
  * `ATAPE.PS`: PostScript-Programm zur Erstellung von Audiocassetten-Labels. Dokumentation als Kommentar in der Datei.
  * `VTAPE.PS`: Das gleiche für Videocassetten.
  * `BARCODE.PS`: Eine Schriftdefinition, um Barcodes (OCR-Strichcodes) auszudrucken.
  * `BB.PS`: Ein PostScript-Programm zur Ermittlung von fehlenden BoundingBox-Kommentaren bei EPI- oder PostScript-Dateien. Wird noch genauer erklärt.
  * `BDFTOPS.PS`: Schriftkonversionsdatei
  * `BUSCARDS.PS`: Einige PostScript-Makros mit Anwendungsbeipiel, die das Erstellen von Geschäftskarten in PostScript einfach machen sollen.
  * `MYCARD.PS`: Modifikation von `BUSCARDS.PS`.
  * `CHEQ.PS`: PostScript-Schriftdefinition zum Zeichnen von Schachdiagrammen (falls Sie so etwas tun möchten, empfehle ich dringend die Chess-Schriften für TeX, die mit dem zugehörigen Makro-Paket wesentlich besser für diesen Zweck geeignet sind; im HRZ erhältlich).
  * `DECRYPT.PS`: Keine Ahnung, was das ist.
  * `FCUTILS.PS`: Schriftkonversionsdatei
  * `PCHARSTR.PS`: Schriftkonversionsdatei
  * `PFBTOPS.PS`: Schriftkonversionsdatei
  * `PS2IMAGE.PS`: Konvertiert eine beliebige PostScript-Datei in ein PostScript-Bild im Pixelformat. Dokumentation als Kommentar in der Datei.
  * `PSTOPPM.PS`: Schriftkonversionsdatei

- PostScript-Tests und Debug-Hilfen: Wenn Sie selber PostScript-Programme schreiben, sind die hier aufgeführten Debug-Hilfen eventuell von Nutzen. Außerdem gibt es noch zwei sehr nützliche PostScript-Testdateien. Damit Sie die nicht ausprobieren, liegt das Ergebnis eines Testlaufes mit einem QMS-PS 810 als Anlage bei.
  - `COORD.PS`: Testet die Seitenausnutzung eines PostScript-Ausgabegerätes aus.
  - `DOCTOR.PS`: Führt einen globalen Test eines PostScript-Ausgabegerätes durch. Ergebnis eines solchen Testlaufs mit einem QMS-PS 810 liegt als Anlage bei.
  - `PPATH.PS`: Debug-Hilfe, Dokumentation in der Datei
  - `TRACEOP.PS`: Debug-Hilfe, Dokumentation in der Datei
  - `UNPROT.PS`: Debug-Hilfe, Dokumentation in der Datei

- Im Verzeichnis `GHOST\FONTS` finden Sie alle GhostScript-Schriften. Wenn Sie GhostScript nicht auf Ihrem Plattenlaufwerk `D:` installiert haben, müssen Sie hier leider eine kleine Änderung vornehmen: Kopieren Sie alle Schriften (erkennbar an der Endung `.GSF`) in das Verzeichnis `GHOST` (oder wo sich halt das eigentliche Programm GhostScript befindet) und löschen Sie das Verzeichnis `GHOST\FONTS` mitsamt Inhalt. Die Erklärung dafür folgt dann gleich.

- Schließlich entsteht noch das Verzeichnis `\GHOST\DOC`, in das die Original-Dokumentationsdateien von GhostScript kommen. Dazu sage ich weiter nichts.

11.3 Bedienung und Aufruf von GhostScript

Ich gehe im weiteren mal davon aus, daß Sie Ihre GhostScript-Programmdatei `GS.EXE` genannt haben (sprich: Sie haben im Bedarfsfalle die Datei `GS2.EXE` umbenannt). Als PostScript-Beispieldatei will ich im weiteren `EXTRA.PS` verwenden. Angenommen, Sie wollen das tun, was man im allgemeinen mit GhostScript wohl tun will, nämlich sich eine PostScript-Datei (in diesem Falle `EXTRA.PS`) am Bildschirm ansehen. Dann gibt es zunächst einmal ein Problem: GhostScript benötigt seine ganzen Initialisierungsdateien und eventuell die Schriften. Wie soll es die finden? Die notwendigen Pfade dafür müssen bei der Kompilation des Programmes eingestellt werden. Sie wurden von mir auf `D:\GHOST` (bzw. für die GhostScript-Schriften auf `D:\GHOST\FONTS`) festgelegt. Falls Sie also GhostScript auf Ihrem Laufwerk `D:` installiert haben, kein Problem. Ansonsten müssen Sie sich auf andere Weise behelfen, indem Sie nämlich das GhostScript-Verzeichnis entweder in den `APPEND`-Pfad mit aufnehmen oder eben dieses Verzeichnis als aktuelles Verzeichnis einstellen und GhostScript von dort aus aufrufen. Im letzteren Falle bietet es sich an, die darzustellende PostScript-Datei halt auch in dieses Verzeichnis zu kopieren. Davon will ich mal ausgehen.

GhostScript ist ein interaktiver PostScript-Interpreter. Zunächst einmal rufen Sie auf

```
gs extra.ps
```

und schon... beginnt GhostScript nach einigen Eingangsmeldungen die PostScript-Datei auf den Monitor zu zeichnen. Leider werden Protokollausgaben und Fehlermeldungen mitten ins Bild geschrieben. Am Ende der Seite steht dann womöglich mitten in Ihrem Bild die Meldung

```
>>showpage, press <return> to continue<<
```

Wenn Sie das nicht haben wollen, läßt es sich abstellen, allerdings arbeiten Sie dann blind, das heißt, Sie müssen alle notwendigen Eingaben ohne Eingabeaufforderung machen. Dazu kommen wir aber gleich noch. Zunächst weiter zu GhostScript: Es wird stur vorwärts gearbeitet: Eine Seite nach der anderen. Es gibt bei GhostScript keine Möglichkeit, Seiten zu überspringen. Die Datei `EXTRA.PS` besteht aus drei Seiten, und Sie haben nun die erste gesehen. Drücken Sie auf `<RETURN>`, wie angegeben. Es wird die zweite Seite aufgebaut, welche zwei Bilder enthält. Nachdem Sie dann noch einmal auf `<RETURN>` gedrückt haben, sehen Sie die dritte Seite und nach der dritten Seite kommen Sie in den interaktiven Modus von GhostScript. Diesen können Sie verlassen mit der Eingabe `quit` (und dann `<RETURN>`). Das war zwar ganz nett, aber sicher sind Ihnen folgende Mängel aufgefallen:

- Wie unterdrücke ich die Ausgaben, die die schöne PostScript-Seite verunstalten? Das ist einfach, dazu gleich.
- Wie kann ich die Darstellung ein wenig vergrößern, so daß man auch ohne Lupe den Text lesen kann? Das ist schon schwieriger.

Zunächst dazu, wie man die Ausgaben umlenkt. Sie rufen GhostScript auf mit

```
gs extra.ps > prot
```

Als alter DOS-Kenner haben Sie sicherlich sofort erkannt, daß mit „`>` <*Dateiname*>“ nur die normale DOS-Ausgabenumlenkung benutzt wird. Die Ausgaben von GhostScript landen jetzt nicht mehr auf dem Bildschirm, sondern in der Datei `prot`. Dort können Sie sie nach dem GhostScript-Lauf dann ansehen. Nachteil bei diesem Verfahren: Sie bekommen nicht mehr gesagt

```
>>showpage, press <return> to continue<<
```

wenn die Seite zu Ende ist und nach der letzten Seite stehen Sie vor einem herausfordernd weißen Bildschirm. Obwohl keine Eingabeaufforderung zu sehen ist, befinden Sie sich doch im interaktiven Modus und können jetzt `quit` eingeben. **Übrigens:** Falls Sie an irgendeiner Stelle auf Probleme stoßen und sich im „Blindmodus“ befinden, hilft in der Regel die bewährte Kombination `<CNTL-C>`[51]. Falls auch das nicht hilft (dauert manchmal eine Minute, bis GhostScript mal wieder die Tastatur abfragt) und Sie unsicher sind, was los ist: Reset! Danach können Sie dann immer noch nachschauen, was in der Protokolldatei steht, in die Sie die Ausgaben umgelenkt haben (im obigen Beispiel hieße sie `prot`).

Nachdem das nun hoffentlich klar ist zum größeren Problem: Wie kann man die Ausgabe vergrößern? Dazu nur kurz einige Andeutungen: Rufen Sie GhostScript mit dem Kommando `gs` ohne Parameterangabe auf. Sie befinden sich dann im interaktiven Modus. Geben Sie nun das Kommando

```
2 2 scale
```

[51] Falls Sie nach dem Verlassen von GhostScript auf diesem oder einem anderen Wege noch im Grafikmodus sind (erkennbar daran, daß eventuell etwas Müll auf dem Bildschirm ist, das Scrolling sehr langsam geht und die Zeichen ein wenig seltsam aussehen), können Sie in der Regel mit dem DOS-Kommando `mode co80` wieder in den normalen Textmodus zurückschalten.

Damit vergrößern Sie die Ausgaben von GhostScript in horizontaler wie auch in vertikaler Richtung um den Faktor zwei. Nun müssen Sie nur noch die darzustellende Datei laden, das geht mit dem GhostScript-Kommando

```
(extra.ps) run
```

Jetzt dauert es eine ganze Weile und Sie werden mit Erschrecken feststellen, daß die Ausgabe zwar lesbar ist (aber immer noch furchtbar aussieht), doch es wird nur noch die untere Hälfte der Seite dargestellt. Das ist eigentlich klar, denn die Ausgabe ist doppelt so groß, also paßt auch nur noch halb so viel auf den Bildschirm. Wenn Sie die obere Hälfte sehen wollen, müssen Sie wohl die Ausgabe noch verschieben. Das geht mit dem Kommando `translate`. Fangen Sie also noch einmal von vorne an. `gs` aufrufen[52], und nun wird die Ausgabe um eine halbe DIN-A4-Seite nach unten geschoben (wir sahen nur die untere Hälfte, also muß die Ausgabe nach unten geschoben werden, um auch die obere Hälfte ins feststehende Fenster zu bekommen). Dazu muß man erstmal wissen, wieviel das in PostScript-Einheiten ist. Diese Rechnerei mit den PostScript-Einheiten hatte ich bereits im Unterkapitel über die Verschiebung der Ausgaben von DVIPS vorgestellt. Eine Einheit = $\frac{1}{72}$ Inch. Eine DIN-A4-Seite ist 29.7 cm hoch, daher muß etwa um 500 Einheiten nach unten geschoben werden. Also geben wir zuerst das Kommando

```
translate 0 -500
```

Natürlich können Sie mit dem ersten Parameter auch noch nach links und rechts verschieben. Der Einfluß der Vorzeichen auf die Verschiebungsrichtung ist identisch mit dem, was im Unterkapitel über die Verschiebung der Ausgaben von DVIPS gesagt wurde. Nun geben Sie das Kommando

```
2 2 scale
```

womit wieder die Ausgabe in horizontaler wie in vertikaler Richtung um Faktor 2 gestreckt wird. Und dann wieder

```
(extra.ps) run
```

Schon sehen Sie die obere Hälfte. Sicherlich wissen Sie inzwischen auch, wie Sie die Ausgabe auf ein Drittel der ursprünglichen Größe reduzieren oder die Ausgabe dreimal so groß machen.

Ein Problem werden Sie allerdings sehen, sobald Sie die erste Seite lange genug angeschaut haben und RETURN drücken, um die zweite Seite zu sehen. Alle mühevoll gemachten Einstellungen zur Vergrößerung und Verschiebung sind weg! Da läßt sich leider auch gar nichts dran machen. Unter anderem auch aus diesem Grund bin ich der Ansicht, daß GhostScript im wesentlichen eigentlich nur dazu geeignet ist, Bilder und Layout zu überprüfen sowie eventuell PostScript-Dateien (für Probeausdrucke!) auf einem Nadeldrucker auszugeben oder um selbstgeschriebene PostScript-Programme zu testen.

Zu einem weiteren Problemkreis: Die GhostScript-Schriften. Wenn Sie in Ihrem Dokument PostScript-Schriften verwenden, so ist es ja Sache des Ausgabegerätes, diese Schriften zu berechnen. GhostScript verwendet dazu seine Schriften, die im Verzeichnis `\GHOST\FONTS` sein sollten (bzw. im GhostScript-Verzeichnis (wo auch die Datei `GS.EXE` ist), wenn die Installation nicht auf Laufwerk `D:` erfolgte). Diese Schriften sind nicht geeignet, um damit korrespon-

[52] Natürlich können Sie auch `gs >` <*Dateiname*> aufrufen, dann arbeiten Sie eben blind, haben dafür aber auch keinen Dreck im Bild.

denzfähige Ausdrucke zu erzeugen. Sie ähneln den originalen PostScript-Schriften, mehr aber auch nicht. Insbesondere werden Sie mit den PostScript-Ausgaben von DVIPS bei Verwendung von PostScript-Schriften, für die als Ersatz die GhostScript-Schriften herhalten müssen, Flatterränder rechts bekommen. Wegen dieser Probleme und wegen des enormen Speicherplatzbedarfs der GhostScript-Schriften empfehle ich im allgemeinen, auf sie zu verzichten. Jede nicht vorhandene GhostScript-Schrift wird durch die Schrift `UGLYR.GSF` ersetzt, die sich im GhostScript-Verzeichnis und noch einmal im Verzeichnis `\GHOST\FONTS` befinden sollte. Damit können Sie mit GhostScript bei Verwendung von PostScript-Schriften immer noch das Layout und die Bilder kontrollieren und auch noch lesbare Ausdrucke auf einem Nadeldrucker erzeugen.

Dazu sollte ich jetzt endlich kommen: Wie macht man Ausdrucke auf einem Drucker? Einfach durch Angabe eines zusätzlichen Parameters beim Aufruf von GhostScript. Rufen Sie GhostScript so auf:

`gs -sDEVICE=epson` <*Dateiname*>

Dann wird die angegebene Datei auf einem am Standard-Druckerport `LPT1:` angeschlossenen EPSON-LQ- oder NEC-Pinwriter-kompatiblen Drucker ausgegeben. Die Protokollausgaben landen weiterhin auf dem Bildschirm; in diesem Falle ist ein Umlenken in eine Protokolldatei nicht mehr sinnvoll. Weitere Parameter und interaktiven Modus (vergrößern und verschieben) können Sie wie sonst auch handhaben.

Falls Sie intensiver in die Benutzung von GhostScript einsteigen möchten, sei die Datei `USE.DOC` im Verzeichnis `\GHOST\DOC` zur Lektüre empfohlen. Insbesondere sollten Sie sich diese Datei durchlesen, wenn Sie vorhaben, mit GhostScript selbstgeschriebene PostScript-Programme zu testen. GhostScript hat einige Debug-Options, die in diesem Falle sehr nützlich sind.

Wenn Sie GhostScript mit dem Parameter `-?` aufrufen, erhalten Sie eine Kurzanleitung (quasi als Gedächtnisstütze).

11.4 Herausfinden von BoundingBoxes

Ich hatte bereits mehrfach darauf hingewiesen: Es gibt bei GhostScript ein PostScript-Programm, mit dem es möglich ist, die BoundingBox eines PostScript-Dokuments herauszufinden, wenn das PostScript-Dokument keinen BoundingBox-Kommentar hat (der ist ja für das Einbinden von PostScript-Bildern mit TeX zwingend erforderlich). Hierzu dient das PostScript-Programm `BB.PS` im Verzeichnis `\GHOST\PSGOODIES`. Sie rufen GhostScript wie gewohnt auf und lassen diese PostScript-Datei ausführen:

```
gs bb.ps
```

Dann dauert es etwas, und Sie befinden sich im interaktiven Modus (sprich: es erwartet Sie eine Eingabeaufforderung). Nun geben Sie die Datei an, von der Sie die BoundingBox herausfinden möchten; in unserem Falle ist das `BULB.PS` (diese Datei befindet sich im Verzeichnis `\GHOST\PSGOODIES` und enthält in der Tat keinen BoundingBox-Kommentar, deshalb habe ich sie als Beispiel ausgewählt). Also bei der GhostScript-Eingabeaufforderung eingeben:

```
(bulb.ps) run
```

Nun wird die PostScript-Datei `BULB.PS` ganz normal ausgeführt, es entsteht auf dem Bildschirm das mittlerweile gewohnte Bild. Aber: Nachdem die Seite fertig aufgebaut ist wird sie umrahmt und unter die Seite werden die Abmessungen der BoundingBox geschrieben. In diesem Falle steht dort:

```
%%BoundingBox: 180 273 420 545
```

Wenn Sie nun dieses Ergebnis, also obige Zeile, genau so in die zweite Zeile der Datei einfügen, haben Sie eine PostScript-Datei mit gültigem BoundingBox-Kommentar, und das wars auch schon.

Es ist übrigens auch möglich, **beide** Dateien als Parameter anzugeben, so daß sie direkt hintereinander ausgeführt werden, konkret:

```
gs bb.ps bulb.ps
```

Das geht immer, Sie können also auf diese Weise beliebig viele Dateien hintereinander ausführen lassen.

12 Unterstützung durch das HRZ und Schlußwort

Zuständiger Mitarbeiter für TEX und Anwendungen (wie DVIPS) am HRZ ist GÜNTHER PARTOSCH (HRZ, Heinrich-Buff-Ring 44, Raum 135, Tel. 0641/702-2170). Seine Email-Anschrift (in der X.400-Notation):
`c=de; admd=dbp; prmd=uni-giessen; ou=hrz; s=partosch; g=guenter`,
und in der Domain-Schreibweise:
`PARTOSCH@HRZ.UNI-GIESSEN.DBP.DE` (Benutzernummer `U129` am HRZ-Zentralrechner); der Verfasser dieses Textes (STEFAN HOFSTETTER, Institut für theoretische Physik, Heinrich-Buff-Ring 16, Raum 322, Tel. 0641/702-2815) hat die Email-Anschrift (in der X.400-Notation):
`c=de; admd=dbp; prmd=uni-giessen; ou=hrz; s=hofstetter; g=stefan`,
und in der Domain-Schreibweise:
`HOFSTETTER@HRZ.UNI-GIESSEN.DBP.DE` (Benutzernummer `U1079` am HRZ-Zentralrechner). Unterstützung bei Problemen mit DVIPS kann durch das HRZ an Benutzer jederzeit gewährleistet werden. Wenden Sie sich aber vor allem bei Problemen mit Ausdrucken an den PostScript-Druckern des HRZ zunächst an die HRZ-Beratung und die Operateure.

Sie können am HRZ nicht nur dieses Paket bekommen, sondern auch emTEX und die TEX-Schriften für verschiedene Auflösungen (360 dpi, 180 dpi sowie 300 dpi)[53]. Sie können diese Schriftpakete in der Regel nur als Fontlibraries bekommen. Falls Sie die Schriften als einzelne `.pk`-Dateien haben möchten (warum auch immer), wenden Sie sich direkt an die zuständigen Mitarbeiter. Wenn eine neue Version von DVIPS erscheint oder sich wesentliche Neuerungen ergeben sollten, wird dies in den HRZ-News auf dem zentralen Großrechner oder in der Informationszeitschrift des HRZ (LOGIN) bekanntgegeben.

Ich hoffe, Sie sind nach der Lektüre dieses Textes davon überzeugt, daß die Einbindung von PostScript und vor allem die Einbindung von PostScript-Schriften und -Bildern durch DVIPS eine ideale Ergänzung von TEX (bzw. emTEX) darstellen und nutzen diese Möglichkeiten. Wenn Sie die C-Sourcen des DVIPS-Programmes haben möchten, so läßt sich das machen. Wenn Sie DVIPS weiterentwickeln möchten oder den Umfang dieses Paketes erweitern wollen (Modifikation existierender PostScript-Schriften oder so etwas), so wird Ihnen jede Unterstützung gewährleistet, wenn Sie die Früchte Ihrer Arbeit mit anderen Benutzern des HRZ teilen. Diese Dokumentation ist sicher nicht vollständig; das soll sie ja auch gar nicht sein. Es handelt sich um eine Zusammenstellung des notwendigsten (gewissermaßen „the best of“) aus mehreren einzelnen, wesentlich umfangreicheren Anleitungen und einiges selbst erarbeiteten Erfahrungswissens. Für Details oder mehr Information sei in jedem Falle auf diese Einzelanleitungen (meist Englisch und nicht sehr verständlich) verwiesen, die sich auch (meist als ausdruckfertige PostScript-Dateien) im DVIPS-Paket befinden.

Viel Spaß beim Schreiben
schönerer Dokumente mit PostScript!

[53] Alles auf dem anonymen FTP-Server.