

Making Foils Using Foil \TeX

James L. Hafner
IBM Research Division
Almaden Research Center, K53/802
650 Harry Road
San Jose, CA 95120-6099
408-927-1892
bitnet: hafner@almaden
Internet: hafner@almaden.ibm.com

Revised last Aug. 21, 1992
Formatted on January 19, 1995

Foil \TeX is a \LaTeX -like system for typesetting foils. Its features include simplicity of use, compatibility with \LaTeX , large sans serif font as default, extra macros to start foils with bold headings and special mechanisms to control the footer and header. There are also facilities incorporated into Foil \TeX , when used with compatible drivers, for one-pass multi-color printing. This document is the user guide for Foil \TeX and describes its basic features and components.

There are restrictions on the use of Foil \TeX . Please refer to Section 7 for more information.

\TeX is a trademark of the American Mathematical Society. \METAFONT is a trademark of the Addison Wesley Publishing Company. \POSTSCRIPT is a trademark of Adobe Systems, Incorporated. \UNIX is a trademark of AT&T Bell Laboratories. Phaser is a trademark and Tektronix is a registered trademark of Tektronix, Incorporated.

Copyright ©1992 by International Business Machines Corporation.

Contents

1	The Foil\TeX Package	3
2	Getting Started: The \backslashdocumentstyle Command	4
3	The Basic Features	5
3.1	The \backslash maketitle Command	5
3.2	The New \backslash foilhead Macro	6
3.3	The New \backslash MyLogo And \backslash Restriction Macros	6
3.3.1	Toggling The Logo	7
3.4	The Other Three Corners Of The Page.	7
3.5	New Theorem And Proof Environments.	7
3.6	Mathematics In Bold Typeface.	8
3.7	List Environments	9
3.8	This Is <i>Not</i> \LaTeX	9
3.9	AMS Fonts	9
3.10	POSTSCRIPT Fonts	10
3.11	Raggedright	10
3.12	Differences With \LaTeX	10
3.13	Differences With $\SL\TeX$	10
3.14	Future Versions	11
4	Fonts and Their Sizes	11
5	Making Color Foils	12
5.1	The Style File <code>colordvi.sty</code> And Output Drivers	13
5.2	Printing In Black/White, With Or Without <code>blackdvi.sty</code>	14
5.3	The Color Macros: User's Viewpoint	14
5.4	User Definable Colors	15
5.5	Setting The Background Color	15
5.6	Protecting Regions From Spurious Colors	16
6	Installing Foil\TeX	17
6.1	Installing Foil \TeX On VM/CMS	17
6.2	Installing Foil \TeX On UNIX	18
6.3	Installing Foil \TeX On DOS Systems	20
6.4	Problems With Installation	22
6.5	Installation Warnings	22
7	Usage Restrictions	22
8	Acknowledgements, Requests And Help	23
A	Sample Foils	25

The system Foil \TeX for making foils (slides, transparencies, etc.) with \TeX essentially consists of a format file (analogous to `lplain.fmt` for \LaTeX or `splain.fmt` for $\Sl\TeX$) and some style files. It is much simpler to use than $\Sl\TeX$, and should be very easy for typical \LaTeX users to master. This document tells you the special features it has, the extra macros that have been added and some instructions on how to install it on your system. Other than some differences with font availability, it should work under any implementation of \LaTeX . It is assumed that you are already familiar with \LaTeX . If not, you should get the book *\LaTeX : A Document Preparation System* by Leslie Lamport, published by Addison-Wesley and get familiar.

The information in this document expands and updates the more general description about Foil \TeX found in the article, “Foil \TeX , A \LaTeX -like System for Typesetting Foils” which appeared in the proceedings of the 1992 Annual meeting of the \TeX User’s Group (in TUGBoat).

1 The Foil \TeX Package

The Foil \TeX package consists of the files listed in Table 1 on top of the basic implementation of \LaTeX .

Table 1: The files in the Foil \TeX package

<code>fltfonts.tex</code>	<code>README.flt</code>	<code>psfonts.sty</code>
<code>fltplain.tex</code>	<code>foildoc.tex</code>	<code>avantgarde.sty</code>
	<code>sampfoil.tex</code>	<code>bookman.sty</code>
<code>foils.sty</code>	<code>foilfont.tex</code>	<code>chancery.sty</code>
<code>foil17.sty</code>		<code>helvetica.sty</code>
<code>foil20.sty</code>	<code>colordvi.[tex,sty]</code>	<code>ncs.sty</code>
<code>foil25.sty</code>	<code>blackdvi.[tex,sty]</code>	<code>palatino.sty</code>
<code>foil30.sty</code>		<code>times.sty</code>
	<code>amssymb.sty</code>	
<code>foiltex</code>	<code>FOILTEX EXEC</code>	<code>foiltex.bat</code>
<code>foiltex.man</code>	<code>FOILTEX HELPCMS</code>	

The first two files in column one of Table 1 are the heart of Foil \TeX . The first defines the basic set of macros that are used (and includes a request to input `latex.tex`¹) and the second defines all the fonts used by Foil \TeX . Font selection is done via the standard (old) \LaTeX scheme, not the new font selection scheme (NFSS) of Frank Mittelbach and Rainer Schöpf (see Section 3.14). The `initex` program compiles the first two files, `fltplain.tex` and `fltfonts.tex`, to produce the format file `fltplain.fmt`. (See section 6.)

The next group of files in the first column are the style files that are used with the Foil \TeX format. The first, `foils.sty`, is the basic style file used for all foils. The other `.sty` files are used to change default font sizes. See Sections 2 and 4 for more information about these files. There are

¹Throughout we will be using the UNIX file naming convention unless specifically referring to the other operating systems.

no .doc files because the .sty files are relatively well documented and most macros are just slight modifications of standard L^AT_EX macros.

The first group of files in column two of Table 1 are documentation and related files. The first file, README.flt, gives general instructions about installation of FoilT_EX. All of that information is repeated and enhanced in Section 6 but it is provided again for easy reference. The rest of the files in this group are the source for the document you are reading together with a fairly detailed sample foils document. If you don't like reading documentation (like this) you can probably do pretty well with FoilT_EX simply by browsing through the sample file and looking at its output. To see its output (after installation), simply type

```
foiltex sampfoil
```

This creates the .dvi and .aux files. You will need to rerun this step to get the cross-referencing right. You can then preview or print this as you would any other .dvi file. The file foilfont.tex is a FoilT_EX file which loads and prints a sample of every font that FoilT_EX might use, including all the load-on-demand fonts as well as the preloaded fonts. FoilT_EX can use a number of fonts at sizes that are not in standard distributions of T_EX/L^AT_EX so this file can be used to test your system's font availability. Because it loads well over 100 fonts, it may not run successfully under small installations of T_EX and it may need to be done in small pieces. If you find that you are missing some fonts, you should consult your local T_EX support or T_EXpert to acquire them or have them generated for your output devices with METAFONT.

When FoilT_EX was conceived, a suggestion was made to add one-pass color printing capability. This is more related to drivers, but we developed a device independent (but driver dependent) scheme for doing this. The necessary files are included in this package and are listed as the second group of files in the second column of Table 1. The files color_{dvi}._[tex,sty] and black_{dvi}._[tex,sty] contain device-independent macros for using color in FoilT_EX (or any other T_EX). The .tex and .sty files are identical and are duplicates of those provided in the latest release of Tomas Rokicki's dvi2pdf with dvips driver. More on this feature and on printer driver requirements can be found in Section 5.

The large group of files in column three of Table 1 are modifications of L^AT_EX styles for substituting POSTSCRIPT fonts for the standard Computer Modern fonts. More on this can be found in Section 3.10.

The lone file amssymb1.sty in the middle column is a style file that loads font and macro information for the $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX fonts msam, msbm and eufm. More on this can be found in Section 3.9.

Finally the last two rows of Table 1 are system dependent files for running FoilT_EX. These include scripts, execs or batch files for invoking T_EX with the fltplain.fmt format as well as some simple on-line help information. We have included these files for UNIX, VM/CMS and DOS systems only because that was all we could test.

2 Getting Started: The \documentstyle Command

To create a FoilT_EX document, you edit a file very much like a L^AT_EX file. Instead of the standard L^AT_EX options specified in the \documentstyle command, you should use

```
\documentstyle[opts]{foils}
```

Here, the *opts* list can include any standard macro packages that you normally use (and that don't corrupt any macros defined by `foils.sty`).

By default, `foils.sty` loads `foil20.sty` and sets up the normal size fonts at 20pt. Analogous to L^AT_EX's 11pt and 12pt style options, FoilT_EX has 25pt, 30pt and 17pt options. For example, to make normal size at 25pt the command

```
\documentstyle[25pt,opts]{foils}
```

will do the trick. Contrary to L^AT_EX, the default 20pt *is* an acceptable option, though it is redundant.

Once you have created your FoilT_EX file, to run FoilT_EX on it, simply type at the command line or system prompt

```
foiltex filename
```

or

```
virtex &fltplain filename
```

where `virtex` is the name of your compiled T_EX program. Both of these commands create a `.dvi` file which can be printed or previewed in your usual way.

3 The Basic Features

This version of FoilT_EX has a number of new (and hopefully useful) built-in features. The first is that the basic fonts are in large size, approximately 20pt, (so you do not need to do size changing to get large type). The default font is also sans serif as this (in the opinion of many) looks better on foils than serif fonts like roman. We have implemented L^AT_EX's font and font size changing commands relative to this default. More information about fonts and size changing can be found in Section 4.

In spite of the fact that the basic font is sans serif, the numerals and other symbols from the roman font when used in math mode are still in the roman font. Thus mathematics will look exactly the same as in L^AT_EX (only larger) but numerals in text will appear in sans serif.

In addition, almost all L^AT_EX macros are available including automatic referencing and citation, footnotes, and itemize (which will probably be very popular for foils). The user is not expected to have to do anything to control font types or size changing, except as might be expected in a typical L^AT_EX document.

The next subsections describe a number of additional macros and features that have been defined to make foilmaking easier. In the appendix is a small sample foil document in FoilT_EX source and final output form to demonstrate the simplicity and the beauty (we hope you agree) of the output. In the final few subsections of this section we mention a few of the differences between FoilT_EX and L^AT_EX/SL^AT_EX and some plans for the future.

3.1 The `\maketitle` Command

The use of FoilT_EX's `\maketitle` command is the same as for L^AT_EX when using the `titlepage.sty` style option. That is, it reads the contents of `\title{}`, `\author{}`, etc., and produces a titlepage,

actually a title foil. The title itself appears centered and down a small space from the top, in a `\Large` bold sans serif font. The author’s name with address and date appear under the title, centered and in the `\normalsize` font. If desired, this can be followed by a (necessarily short) abstract with the word “Abstract” appearing in bold and centered above the text of the abstract. See the appendix for a sample. The footer of the title page will contain some special text (see Section 3.3 for more details).

3.2 The New `\foilhead` Macro

The first new macro is called `\foilhead`. Its use is described by

```
\foilhead[length]{text}
```

This macro starts a new page and puts *text* in `\large` bold type at the top center of the new page. After the header, a vertical space of approximately 1.0 inch is added providing an automatic cushion between the header and the body of the foil. You can adjust this space either up or down by putting in the optional argument a TeX *length*. For example, if you want the body of your foil to sit closer to the header, you could use the command

```
\foilhead[-.5in]{This is the Header}
```

This macro should be used to start any new foil, especially if a new heading is needed. If you try to put too much text on a single foil, FoilTeX will do its own page break. This could cause some odd vertical spacing since there is a fair amount of stretchability in vertical glue, particularly in list environments. This can easily be fixed simply by forcing a page break with an empty `\foilhead{}` command.

3.3 The New `\MyLogo` And `\Restriction` Macros

Another new pair of macros, `\MyLogo` and `\Restriction`, each of which takes a single argument, are used to control the contents of part of the footline. By design, the footline consists of the contents of `\MyLogo` followed by the contents of `\Restriction` all left justified, with the page number right justified². On the main foils, the default font size is `\tiny`. The contents of these macros can be an empty box as well. By default, `\Restriction` is empty and `\MyLogo` is the phrase “– Typeset by FoilTeX –”.

The declarations for these macros would normally be placed in the preamble to the document, i.e., before the `\begin{document}` command. However, these macros can be declared or redeclared at any place in the document. They (and all the other commands that control the footer and header) are sensitive to FoilTeX’s output routine, which is essentially unchanged from L^AT_EX’s. Consequently, care must be taken in their placement to be sure they act on the correct pages. In the preamble or immediately after the `\foilhead` command are best. In addition, there are macro switches that can be used to easily turn on or off the logo, without having to do any redeclarations. See Section 3.3.1 for more information.

`\MyLogo` is really intended for something idiosyncratic to the speaker or his organization. For example, it is easy to use a package like `psfig` to include some graphic as the logo on every page:

²For the title foil, there is no page number; `\MyLogo` and `\Restriction` are centered and appear in `\footnotesize` font.

```
\MyLogo{\psfig{figure=arclogo.eps,height=1in}}
```

puts a one inch tall version of the IBM Almaden Research Center logo in the lower left corner of all the author's foils. `\Restriction` was included in case you want to have each foil identified for a particular audience. For example, at IBM, we have the option of displaying the IBM logo and words like "Confidential" or "Internal Use Only". The defaults are set in `foils.sty`.

3.3.1 Toggling The Logo

Users of an early IBM version of FoilTeX requested an easier mechanism (than undefining/redefining `\MyLogo`) for inhibiting a logo from appearing on selected foils or all foils. We implemented this feature with two switches. These macros are `\LogoOn` and `\LogoOff` and they do exactly what their names imply. If `\LogoOff` appears before the footer is processed by the output routine no logo will appear (as if `\MyLogo{}` were declared). This stays in effect until `\LogoOn` is encountered, at which point the contents of `\MyLogo` are restored.

So, for example, if you do not want the logo to appear at all, you can put the `\LogoOff` command *before* the `\begin{document}` command. If you want the logo only on the title page, then you can put this command *after* the first occurrence of `\foilhead`. You can then turn the logo back on by putting the `\LogoOn` command in a convenient place.

3.4 The Other Three Corners Of The Page.

Since the macros `\Restriction` and `\MyLogo` control the bottom left corner of the page, there are other macros for putting text in the other three corners. These are, not surprisingly,

```
\rightfooter{text}
\lefthead{text}
\righthead{text}
```

They each take one argument, the text you want to place in the associated corner of the page. These can also be redeclared within the document with the appropriate attention paid to the output routine. See Section 3.3.

By default the headers are empty and the lower right footer is just the page number:

```
\righthead{}
\lefthead{}
\rightfooter{\quad\sff\thepage}}
```

except on the title page where they are all suppressed. You can easily suppress page numbering by declaring `\rightfooter{}`. Unless controlled by a font size changing command, text in these regions appear in a `\tiny` font. These defaults are set in `foils.sty`

We did not add macros for centering text in the header or footer because we felt this simply add unnecessary clutter to the foils.

3.5 New Theorem And Proof Environments.

There are a number of (both starred and unstarred) `\newtheorem` environments built in. These are for `Theorem`, `Lemma`, `Corollary`, `Proposition` and `Definition`. Note the uppercased first

letter (to avoid possible collisions with user-defined environments of this type). Each must begin and end with `\begin{}` and `\end{}` commands as usual. Their text begins with a bold sans serif label like **Theorem** and the content of each is typeset in *slanted sans serif*. The unstarred forms are sequentially numbered and support automatic referencing. The starred forms suppress the numbering and referencing.

All these environments also support an optional argument that can be used for the inventor, common name of the theorem, etc.. Thus

```
\begin{Theorem*}[Gauss] Quadratic reciprocity is true!
\end{Theorem*}
```

will produce (in large type)

Theorem. [Gauss] *Quadratic reciprocity is true!*

The unstarred form will be numbered.

To implement this, we added code to L^AT_EX's `\newtheorem` macro which defines *both* the starred and unstarred forms of these environments at the same time. In this way, users could easily add their own versions of similar environments. For example,

```
\newtheorem{Axiom}{Axiom}
```

would define two environments `Axiom` and `Axiom*` that behaved just like `Theorem` and `Theorem*`. In all other respects, e.g., numbering convention, `\newtheorem` behaves just as in L^AT_EX.

Finally, there is a `Proof` environment which opens with the word **Proof** and ends with a \square . The contents are printed in the normal font.

3.6 Mathematics In Bold Typeface.

FoⁱL^AT_EX uses a modified form of L^AT_EX's font definitions for bold typefaced mathematics. In particular, a `\bf` command in math mode will switch to a bold sans serif font (probably not desirable in mathematics since the rest of mathematics is in serifed fonts). In FoⁱL^AT_EX, L^AT_EX's `\boldmath` command has been modified also. Here, characters from the roman font are emboldened by switching to the bold roman font (`cmbx` family), not the bold math symbol font as in L^AT_EX.

To make using bold mathematics easier some new macros have been defined. The first is

```
\bm{formula}
```

This takes its argument (within mathematics mode) and replaces it with its emboldened version. Unfortunately, it acts a little funny on characters like summation signs and in super- or subscripts partly because it reverts to T_EX's text style (style *T*) first (by enclosing the subformula in an `\mbox{$ $}`). Consequently, this command should be used primarily on individual characters or small parts of formulas. (A more natural syntax for this command ought to be `{\bm formula}` but the author could not find a way to implement the same effects in this way.)

The second method for getting bold mathematics is a pair of environments

```
\begin{boldequation}
formula
\end{boldequation}
```

```

\begin{boldequation*}
formula
\end{boldequation*}

```

They both set *formula* in bold (except for super- and subscripts). The unstarred form has automatic referencing and is numbered; the starred form inhibits the numbering and referencing.

The limitation on the super- and subscripts not appearing in bold face is strictly to limit the number of fonts loaded by Foil \TeX . In the definition of `\boldmath`, the `\scriptfont` and `\scriptscriptfont` styles are not redefined for any of the math font families. It was felt this bold math feature would have limited use and so it is not fully supported. If there is sufficient demand, it could easily be extended.

3.7 List Environments

The vertical spacing of items in list environments is controlled by exactly the same mechanisms as in \LaTeX . We have set the defaults, however, so that at the highest level there is a fair amount of vertical space, but at lower levels this shrinks to nothing. This seemed to produce the best and most pleasing results, at least to the author's personal taste.

3.8 This Is *Not* \LaTeX

At the heart of Foil \TeX is a format file. Consequently, there is usually a system dependent exec (or script or batch program) which calls the main \TeX program with the necessary Foil \TeX format file, `fltplain.fmt`. Testing showed that users (especially hard-core \LaTeX users) tended to run \LaTeX instead out of habit. As a result a special feature was implemented in which, if \LaTeX is called on a Foil \TeX file, the user is prompted with a warning and given a choice of continuing with some unpredictable consequences or aborting.

3.9 AMS Fonts

Included with Foil \TeX is the style file `amssymb1.sty` which can be used with either Foil \TeX or \LaTeX (with the *old* font selection scheme). This style loads the $\mathcal{A}\mathcal{M}\mathcal{S}$ - \TeX symbol fonts from the `msam`, `msbm` and `eufm` fonts at all the necessary magnifications (for the appropriate format). It *adds* to the existing font data for these formats in such a way that the standard size changing mechanisms work correctly on these symbol fonts.

To use this file, you must have both `amssym.tex` and `amssym.def` from the AMSFonts package, version 2.0 or greater (as well as the actual fonts and `.tfm` files).

There are three things to note. First, this file generally needs to be the last style option listed in the `\documentstyle` command which affects font selection. Second, this file is *not* compatible with the new font selection scheme in \LaTeX ³. Finally, this file will load (in Foil \TeX) some of these fonts at the equivalent of `\magstep6`, 7 and 8. Most installations of \TeX will not have these fonts at these sizes. Consequently, you may have to run METAFONT (or have your system administrator do it) to be able to print/preview some Foil \TeX files using this style option. Alternately, you could

³It is the author's (untested) understanding that the file `amsfonts.sty`, a part of $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX , will work in \LaTeX -NFSS.

modify this file and downsize the very large fonts to available sizes with the predictable effects. There are comments in the file itself which explain how to do this.

3.10 POSTSCRIPT Fonts

As mentioned in Section 1, we have included with Foil \TeX a family of style files which substitute POSTSCRIPT fonts for many of the Computer Modern fonts in text (but not in mathematics). These files are revised versions of files already in circulation for use in \LaTeX (again with the old font selection scheme). We have *added* to these files extra code to make them compatible with Foil \TeX as well as \LaTeX . Their behavior in \LaTeX should be just as before.

As an example, if you would like to replace the sans serif font of Foil \TeX , `cmss`, with Helvetica simply add `helvetica` to your document style options list:

```
\documentstyle[helvetica]{foils}
```

This also has other effects like replacing `cmr` with Times-Roman and `cmtt` with Courier so that most text fonts will be in some POSTSCRIPT font.

It should be remembered that these versions of these files assume that your \LaTeX uses the old font selection scheme. They are probably not compatible with the new scheme.

3.11 Raggedright

Some users feel that `\raggedright` is preferable for foils. It was decided not to make this the default (as this is not the author's opinion), but to leave this to the user's discretion. To get this effect, simply put `\raggedright` in the preamble to your document.

3.12 Differences With \LaTeX

One simple difference is that the \LaTeX command `\em` switches from any unslanted font to *slanted sans serif* and from any slanted font to unslanted sans serif, not to *text italics* and roman, respectively.

Unlike \TeX/\LaTeX , numerals in Foil \TeX look different when they are in ordinary text from when they are in math-mode. This means that 12345 in text will print as 12345 and $\$12345\$$ prints as 12345.

Hyphenation has been eliminated from Foil \TeX . It was felt that this improves readability. Because of this, Foil \TeX might have problems fitting things nicely on a line. Overfull and underfull `\hboxes` might occur more often than in \LaTeX but the tolerances are set to reduce their frequency. Since the fonts are so large, Foil \TeX can be more tolerant of white space without being unaesthetic. If they do occur with no obvious fix, a discretionary hyphen strategically placed or a localized `\raggedright` or an `\hfil\break` to force a line break can be used to resolve the problem.

The following features of \LaTeX have been disabled in Foil \TeX because they seemed unnecessary: lists of figures, indexing, glossary. They can easily be added if there is sufficient demand. Some features of \LaTeX , like table of contents, are *not* disabled in Foil \TeX , but they will probably not get much (or any) use.

3.13 Differences With $\SL\TeX$

There are many differences between $\SL\TeX$ and Foil \TeX . The most glaring feature not supported in Foil \TeX is invisible fonts for overlays. Also, as indicated in Table 2, `\rm` and `\sf` do what you

expect, that is switch to roman and sans serif, respectively. In `SLTEX`, they *both* yield the sans serif font (so true roman fonts are disabled in `SLTEX`).

3.14 Future Versions

A possible new feature might be an automatically-generated “Summary of the Talk”, akin to a table of contents, where the user could tag some of the `\foilhead` macros and have them collected in a special foil following the title foil.

There are plans for two major additions/revisions to `FoilTEX`. These are support for invisible fonts for use with overlays and a conversion to the new font selection scheme. There is no time-table for these upgrades, as they will depend in large part on user demand.

4 Fonts and Their Sizes

As noted earlier, the default font at `\normalsize` is a **sans serif** font at size 20pt, unless one of the `[17pt]`, `[25pt]`, or `[30pt]` options have been declared in the `\documentstyle` command. Table 2 shows the control sequences for other accessible text fonts and the name of the font in a sample of its type. These control sequences give the font at the current size. Font size changing commands for each of the normal point size options are described by Table 3. Note that `\bf` and `\sl` yield sans serif fonts, not the usual variations on roman.

Table 2: Available fonts and their names.

command	font names
<code>\sf</code>	Sans Serif
<code>\it</code>	<i>Text Italic</i>
<code>\sl</code>	<i>Slanted Sans Serif</i>
<code>\bf</code>	Bold Sans Serif
<code>\tt</code>	Typewriter
<code>\rm</code>	Roman
<code>\sc</code>	SMALL CAPS

Mathematics is also automatically displayed at normal size unless magnified by a size changing declaration. Table 4 describes the font point sizes for `TEX`'s mathematics styles at each of the normal point size options. `FoilTEX` loads or knows about enough fonts, particularly symbol fonts, that there should never be a discrepancy between the size of text and mathematics at any of the different sizes (unlike `LATEX` where some fonts at `xxvpt` are actually only 20pt fonts).

Since many of `FoilTEX`'s fonts are not in the standard distribution, and so not available on most systems, the installer will probably have to run `METAFONT` to generate the necessary files. The file `foilfont.tex` requires a sample of every preloaded or load-on-demand font and so can be used to test an installation's font availability. (Some drivers, like Tomas Rokicki's `dvi2pdf` with `dvips` program, generate all the missing fonts just by trying to process this file.)

The `LATEX` circle and line fonts have been preloaded at `magstep4` so that small `LATEX` pictures should scale naturally to a foil.

Table 3: Type sizes for Foil \TeX size-changing commands for the different document style options.

size/doc-opt	20pt (default)	17pt	25pt	30pt
<code>\tiny</code>	12pt	12pt	12pt	14pt
<code>\scriptsize</code>	12pt	12pt	14pt	17pt
<code>\footnotesize</code>	14pt	12pt	17pt	20pt
<code>\small</code>	17pt	14pt	20pt	25pt
<code>\normalsize</code>	20pt	17pt	25pt	30pt
<code>\large</code>	25pt	20pt	30pt	36pt
<code>\Large</code>	30pt	25pt	36pt	43pt
<code>\LARGE</code>	36pt	30pt	43pt	43pt
<code>\huge</code>	43pt	36pt	43pt	43pt
<code>\Huge</code>	43pt	43pt	43pt	43pt

Table 4: Mathematics type styles and their point sizes at `\normalsize` for the different document style options.

style/doc-opt	20pt (default)	17pt	25pt	30pt
D, D', T, T'	20pt	17pt	25pt	30pt
S, S'	14pt	12pt	17pt	20pt
SS, SS'	12pt	12pt	14pt	17pt

5 Making Color Foils

This feature is still in the development stage and is *very* device-driver dependent. This last problem is regrettable because it severely limits portability, but this cannot be helped at the moment because \TeX was not designed with color in mind. This scheme was originally developed by the author in consultation with Tomas Rokicki (of Stanford University, at the time of this writing). It was implemented in some versions of Rokicki's `dvi2pdf with dvips` program and tested extensively within I \TeX . Rokicki himself later added a number of additional enhancements, both to `dvi2pdf with dvips` and to the code. We have incorporated into Foil \TeX itself code which takes special advantage of this color scheme. We gratefully acknowledge Rokicki's generosity in letting us include in Foil \TeX his versions of the macro files. See Section 6.5 for additional comments.

The color scheme itself was developed to be device independent (i.e., not just for POSTSCRIPT output devices). At the macro level, the scheme of course uses \TeX 's `\special` command, but does not use any syntax that is dependent on the physical device or output data stream. In this way, it is hoped that more drivers can take advantage of the same set of macros for color printing or display. The only drivers we are aware of that fully support the macros we describe here are Tomas Rokicki's `dvi2pdf with dvips` program (version 5.48) and the `TeX viewer` under NeXT. In the next few sections we will discuss this implementation of color.

One other comment: these color macros are not necessarily limited to Foil \TeX but can run under any other \TeX . However there are subtleties about how footers, headers, and other special

regions of the text will handle the color changes. For very successful use, some stylizing macros need to be modified with implied color. We have not tested this explicitly but foresee no special difficulties (provided the driver operates compatibly). The relevant macros in Foil \TeX already have these features built in. For example, the footer and header macros wrap everything in $\backslash\text{Black}$ so colors in the text that cross a page boundary will not affect these regions. See Section 5.6 for more information.

5.1 The Style File `colordvi.sty` And Output Drivers

As we see it, the “best” way to use color in Foil \TeX (or other \TeX) files is with the `colordvi.sty` file. These macros can be included in Foil \TeX , for example, by simply adding `colordvi` to the $\backslash\text{documentstyle}$ command:

```
 $\backslash\text{documentstyle}[colordvi]\{\text{foils}\}$ 
```

(In \TeX s that don’t have document styles, the appropriate $\backslash\text{input}$ command will work as well since there is a `.tex` version of this file.) This file defines all the color macros using \TeX ’s $\backslash\text{special}$ command. The internal syntax has forms like

```
 $\backslash\text{special}\{\text{color push Red}\}$ 
```

Nested Red text.

```
 $\backslash\text{special}\{\text{color pop}\}$ 
```

```
 $\backslash\text{special}\{\text{color Blue}\}$ 
```

Default or global color now Blue.

depending on whether this is a nested color or global color change (see Section 5.3). Consequently, a compatible driver must be able to recognize the $\backslash\text{special}$ keyword `color` and process something to the output file that signals the color change, tracking the nesting level, etc. It is also important that a driver be able to track the color state across page boundaries or any other boundary where the output state can change. A driver should ideally also produce output where each page has self-contained color state information, so that pages can be printed in different orders, or by selected pages.

An additional macro in `colordvi.sty` can be used to set the background color. For this macro, a driver needs to recognize the $\backslash\text{special}$ keyword `background` and must be able to set the specified background color on the *current* page and remember that color until changed explicitly.

Furthermore, the actual color parameters need to be set in some device dependent way, say with a special prologue file that defines the color `Red` in terms the output device can understand, and in such a way that the parameters are tuned to the particular device. (Each output mechanism uses different color renditions which makes it very difficult to set a universal standard.)

For Rokicki’s `dvi2pdf with dvips`, we have done all of the above. There is a color prolog file which `dvi2pdf with dvips` includes in its header list whenever it encounters the keywords `color` or `background`. The particular one we wrote has the color parameters tuned to the Tektronix PHASER printer. We added code to `dvi2pdf with dvips` to track the color history and states during the prescan. In this way, it can initialize color state on each page of the output file during the final scan. (As mentioned before our original code for `dvi2pdf with dvips` and our original set of macros were greatly improved by Tomas Rokicki. We are grateful for his help and for including these features in his driver, version 5.48 and later.)

Detailed information about use of color in `dvi2pdf` with `dvips` can be found in the user manual for program. In the next few sections, we will describe the simplest aspects of the use of color, paying particular attention to its use in `FoilTeX`.

Finally, we remark that we have used the names `color` and `black` suffixed by `dvi` so as not to conflict with Leslie Lamport's `color.sty` which has become somewhat wide-spread. We chose the suffix `dvi` because it reflects the device independent nature of the macros.

5.2 Printing In Black/White, With Or Without `blackdvi.sty`

A `FoilTeX` (or other `TeX`) document written with color macros can be printed in black and white in two ways. If the device is a black and white version of a color device (e.g., `display` or `POSTSCRIPT` printer) then it should print in corresponding grey-levels. This is useful since in this way one can get a rough idea of where the colors are changing without using expensive color printing devices. The second option is to replace the call to input `colordvi` with `blackdvi`. This “black” style file turns all the color macros into no-ops, and so will produce normal black/white printing without the user having to ferret out the color commands. Also, most device drivers will simply ignore the color commands and so print in normal black and white.

5.3 The Color Macros: User's Viewpoint

There are two kinds of color macros, ones for local color changes to, say, a few words or even one character and one for global color changes. All the color names use a mixed case scheme. There are 68 predefined colors, with names taken primarily from the Crayola 64 crayon box, and one pair of macros for the user to set his own color pattern. More on this extra feature later. There is one other special macro which we will describe later in Section 5.6. Users can browse the file `colordvi.sty` for a list of the predefined color names.

A local color command is in the form

```
\ColorName{this will print in color}
```

As this example shows, this type of command takes one argument which is the text that is to print in the selected color. This can be used for nested color changes since it should restore the original color state when it completes. For example, suppose a user was writing in green and wanted to switch temporarily to red, then blue, back to red and restore green. Here is one way to do this:

```
This text is green but here we are
\Red{switching to red,
\Blue{nesting blue} recovering the
red} and back to original green.
```

In principle the nesting level is unlimited, but it is not advisable to nest too deep lest one loose track of the default color or exceed the driver's capacity.

The global color command has the form

```
\textColorName
```

This macro takes no arguments and immediately changes the default color from that point on to the specified color. This of course can be overridden globally by another such command or locally by local color commands. For example, expanding on the example above, we might have

```

\textGreen
This text is green but here we are
\Red{switching to red,
\Blue{nesting blue,} recovering the
red} and back to original green.
\textCyan
The text from here on will be cyan
unless \Yellow{locally changed
to yellow}. Now we are back to cyan.

```

The color commands will even work in math mode and across math mode boundaries. This means that a color state going into math mode will force the mathematics to be set in that color as well. More importantly however, in alignment environments like `tabular` and `eqnarray`, local color commands *cannot* extend beyond the alignment characters.

Because local color commands respect only some environment and delimiter changes besides their own, care must be taken in setting their scope. It is best not to have them stretch too far.

5.4 User Definable Colors

There are two ways for the user to specify colors not pre-defined. For local changes, there is the command `\Color` which takes two arguments. The first argument is a quadruple of numbers between zero and one and specifies the intensity of cyan, magenta, yellow and black (CMYK) in that order. The second argument is the text that should appear in the given color. For example, if a user wants the words “this color is pretty” to appear in a color which is 50% cyan, 85% magenta, 40% yellow and 20% black, they would use the command

```
\Color{.5 .85 .4 .2}{this color is pretty}
```

For global color changes, there is a command `\textColor` which takes one argument, the CMYK quadruple of relative color intensities. For example, to make the default color to be as above, then the command

```

\textColor{.5 .85 .4 .2}
The text from now on will be this pretty color.

```

will suffice.

If the intended output device does not treat color in CMYK terms, then the device *driver* should convert these values to the device dependent parameters, e.g., RGB.

5.5 Setting The Background Color

There is an additional macro for setting the background color. It takes a single argument, which can either be one of the predefined color names or a quadruple of CMYK values. For example,

```
\background{SkyBlue}
```

or

```
\background{.1 .2 .3 .1}
```


These should appear somewhere on the page (preferably near the beginning) where the background color is to change. The background should stay this color until explicitly changed by another such command. It should be remembered that the placement of this is sensitive to the output routine.

5.6 Protecting Regions From Spurious Colors

Because color is defined via \TeX 's `\special` command, it cannot be sensitive to the output routine or certain regions of the page like the header and footer. Consequently, these regions need to be protected from spurious color changes (particularly when local colors spread across page breaks). This is true in $\text{Foil}\TeX$ as well as in any other version of \TeX .

The default color should always be black (of course) and this is what happens in $\text{Foil}\TeX$ and what `dvi2pdf` with `dvips` initializes. The header and the footer in $\text{Foil}\TeX$ are forced to always also be black, even if the text global color changes. Consequently, the contents of `\MyLogo`, `\Restriction`, `\rightfooter` (the page number by default), `\leftheader` and `\rightheader` will all appear in black.

Overriding the color selection for any of these regions of the text in $\text{Foil}\TeX$ can be done by using *local* color commands in their declarations. For example, a very sensitive talk requiring the words “Need-To-Know” in red would use the declaration

```
\Restriction{\Red{Need-To-Know}}
```

In other \TeX 's, user's need to be aware of the possibility of certain regions getting unwanted or unpredicted colors. Headers and footers are most worrisome so style designers who want to use color should keep this in mind.

One particular region of text that gets spurious color effects is labels in list environments. Because of the way list items are defined in standard \LaTeX , the following situation can occur.

```
\begin{itemize}
\item This is the default color (black).
\item \Red{This is Red because we nested a red command.}
\end{itemize}
```

In this example, one would expect that the labels (normally bullets) for these items would both be black. Unfortunately that is not the case. The label of the second item will in fact be red!

To give the user a simple mechanism to solve this problem (and other unforeseen effects of this type) one other special macro is automatically defined. This macro is called `\globalColor`. It is actually a *local* color macro and so takes a single argument. But the color effect it produces is always the same as that set by the *last* `\textColor` or `\textColorName` type command. In effect, when a `\textColorName` command is called, `\globalColor` gets a new definition equivalent to the local `\ColorName` macro. For example, when the default is black, `\globalColor=\Black` and when `\textGreen` appears, `\globalColor=\Green`. This special macro can then be used to protect sensitive regions of the text.

To give an example, in \LaTeX files, one might make sure that the header and footers have `\globalColor` wrapping their contents. In this way, they will inherit the current active default/global (unnested) color state. In $\text{Foil}\TeX$ we chose not to do this effect but to wrap headers and footers in black since it was presumed that in these regions on foils, they should retain a constant color always.

Furthermore, to correct the list environment problem described above, the `\@item` command in FoilTeX is redefined from its L^AT_EX version by wrapping the item label by the `\globalColor` macro. In this way, at least in FoilTeX, item labels will appear as one should expect.

6 Installing FoilTeX

Because installations of T_EX/L^AT_EX differ so much from system to system and even within systems (e.g., there are numerous T_EX packages available for DOS systems), these installation instructions are mostly just an outline of the general procedure. It is hoped that your local T_EXperts or your own T_EX installation instructions will be sufficient to fill in any gaps not handled by these instructions.

In the following subsections we outline the installation procedure for VM/CMS, UNIX and DOS. In every case, there are some basic assumptions made about your installation of T_EX/L^AT_EX. These are:

1. You must have T_EX/L^AT_EX installed, including a version of `initex` and a copy of `latex.tex`.
2. You should be running a version of LaTeX after Nov. 89. The primary concern here is that you have access to the fonts `lcircle10` and `lcirclew10`, as opposed to `circle10` and `circlew10`. If not, then you should either get the `.tfm` files for these or edit the relevant lines of `fltfonts.tex` by changing `lcircle` to `circle` in both places. These files should be where all your `.tfm` files reside (usually in the path covered by your `TEXFONTS` environment variable, or its analogue).
3. It is also assumed that if you do not have all the fonts needed (generally this will be the CM fonts and if, desired, AMSFonts at magnifications equivalent to `\magstep6`, 7 and 8) then you will have access to T_EXperts with METAFONT who can generate them for your device. Since devices and font naming conventions differ so much, it was felt that we could not include the actual binaries for the fonts in the basic package.

In general, the installation of FoilTeX then becomes

- generating a `fltplain.fmt` format file by running `initex` (and installing this in the appropriate location for the system);
- installing the various style, macro, script and on-line help files in the appropriate location for the system (see Section 6.5 for a note of special concern).
- testing font availability by running T_EX with this format against `foilfont.tex` and trying to print this;
- generating all the missing fonts;

6.1 Installing FoilTeX On VM/CMS

On this system, you need only do the following. Install the following files on your T_EX disk (or your A-disk, or any other disk on which you have write access and would like FoilTeX to reside):

```

FLTPLAIN TEX
FLTFONTS TEX
FOILS     STY
FOIL17   STY
FOIL20   STY
FOIL25   STY
FOIL30   STY
FOILTEX  EXEC
FOILTEX  HELPCMS
FOILDOC  TEX
SAMPFOIL TEX

```

You should also install `COLORDVI STY` and `BLACKDVI STY` and their twins `COLORDVI TEX` and `BLACKDVI TEX` if you have access to color output devices and compatible drivers. You can also install `FOILFONT TEX` and all the miscellaneous files (appropriate to your system) listed in Table 1 if you wish access to these utilities (but see Section 6.5 before installing any of these files).

The next step is to run `INITEX` to generate the format file you will need. You do this by

1. making sure you have `INITEX` on an accessed disk (this is usually an `exec` and a `module`),
2. making sure you have `LATEX TEX` on a read-accessed disk, and
3. typing at the CMS prompt

```
INITEX FLTPLAIN
```

This should create a file called `FLTPLAIN FMT`. You should copy this file to the same location as your other format files.

You are now setup to run `FoilTEX`. You can test it on `SAMPFOIL TEX` by typing

```
FOILTEX SAMPFOIL
```

This should generate a DVI file (and `AUX` file) just like `LATEX` and you can output this to your favorite printer in the same way you would any DVI file from `TEX/LATEX`.

You can test your font availability by typing

```
FOILTEX FOILFONT
```

and previewing (if available) and printing on your favorite devices.

See Section 6.4 if you have any problems with the installation.

6.2 Installing FoilTEX On UNIX

We are assuming here a few more things about your installation of `TEX`. First, that you have an environment variable `TEX` defined which is the path to your basic `TEX` system and that `virtex` is the executable binary for `TEX` (if not, then the shell script `foiltex` will have to be modified). Next, we assume the following associations which may be environment variables:

- TEXINPUTS – a directory or path where your macros, style files, etc. are located. This is usually either `$TEX/macros` or `$TEX/inputs`.
- TEXFORMATS – a directory or path where your `.fmt` files are located (like `lplain.fmt`), e.g., `$TEX/formats`.
- MANUAL – a directory to put a `.man` file (`foiltex.man` is a flat ascii file, i.e., *not* in nroff/troff form).
- TEXDOCS – a directory where T_EX documentation can be found. This might be the same as TEXINPUTS or it might be `$TEX/doc`.

With this notation, you should copy the following files of the FoilT_EX package into these places:

Into the appropriate directory in TEXINPUTS put

```
fltplain.tex
fltfonts.tex
foils.sty
foil20.sty
foil17.sty
foil25.sty
foil30.sty
```

Into TEXDOCS put

```
foildoc.tex
sampfoil.tex
```

Into MANUAL put

```
foiltex.man
```

Into `$HOME/bin` (or `/usr/bin` or any other bin you choose) put

```
foiltex (this is a shell script which calls virtex)
```

Make sure that the last file, `foiltex`, has executable privileges (e.g., `chmod a+x foiltex`). Also check that it calls the right executable binary (`virtex`, by default). For C-Shell users, a `rehash` after all of this might be useful too.

You should also install `colordvi.[tex,sty]` and `blackdvi.[tex,sty]` if you have access to color output devices and compatible drivers. You can also install `foilfont.tex` and all the miscellaneous files (appropriate to your system) listed in Table 1 if you wish access to these utilities (but see Section 6.5 before installing any of these files). You might want to put the `README.flt` file someplace more appropriate than TEXINPUTS.

Be sure you have read access (in the TEXINPUTS path) to `latex.tex`. The file `foiltex.man` is a flat ascii file and may need to have its extension changed to match your systems `manpage` structure.

The next step once all the files are available on your system is to invoke `initex` to create the necessary `.fmt` file. One way to do this is to go to the directory TEXFORMATS (do you have write permission to this directory?). Then type

```
initex fltplain
```

This will create a file `fltplain.fmt` in this directory. If this fails, then perhaps your environment variable for TEXINPUTS does not agree with where you put `fltplain.tex` and `fltfonts.tex`.

This completes the installation procedure. From here, you can test your setup by going to a write-accessable directory (someplace you might run $\text{T}_{\text{E}}\text{X}$ from ordinarily) and typing (assuming TEXDOCS is included in your TEXINPUTS search path)

```
foiltex sampfoil
```

This should create the `.dvi` file (and `.aux` file) which you can preview or print in the usual way, provided all the fonts are installed.

To test font availability, type

```
foiltex foilfont
```

and then try previewing or printing the resulting `.dvi` file.

See Section 6.4 if you have any problems with the installation.

6.3 Installing Foil $\text{T}_{\text{E}}\text{X}$ On DOS Systems

Because there are so many different product/packageings of $\text{T}_{\text{E}}\text{X}$ for the DOS world, we can't assume a standard setup. We have tried, however, to make these instructions general enough so that Foil $\text{T}_{\text{E}}\text{X}$ can be installed with little difficulty on different systems. Unfortunately, we have only tested this with Arbortext's $\mu\text{-T}_{\text{E}}\text{X}$ package, Version 3.1a.

We first assume the following associations between our keyword and a directory or path in your installation of $\text{T}_{\text{E}}\text{X}$ (we use lower case file names here to distinguish the file names from the keywords).

- TEX** – a directory below which all of $\text{T}_{\text{E}}\text{X}$ resides. This usually is derived from the name of the $\text{T}_{\text{E}}\text{X}$ package, like `\emtex`, `\pctex` or `\arbortxt`.
- TEXBIN** – a directory where `initex.exe` and `tex.exe` are found, e.g., the `\bin` subdirectory of **TEX**.
- TEXINPUTS** – a directory where your macros, style files, etc., are located, usually the `\inputs` subdirectory of **TEX**.
- TEXFORMATS** – a directory where your `.fmt` files are located (e.g., `lplain.fmt`), usually the `\formats` subdirectory of **TEX**.
- TEXDOCS** – a directory where $\text{T}_{\text{E}}\text{X}$ documentation can be found (this might be the same as **TEXINPUTS**).

With this notation, you should copy the following files of the Foil $\text{T}_{\text{E}}\text{X}$ package into these places:
Into **TEXINPUTS** put

```
fltplain.tex  
fltfonts.tex  
foils.sty  
foil20.sty  
foil17.sty  
foil25.sty  
foil30.sty
```

Into TEXDOCS put

```
foildoc.tex
sampfoil.tex
```

Into TEXBIN (or any other directory in your PATH) put

```
foiltex.bat (see below)
```

You will also need a copy of `latex.tex` available (and readable) on your system. In some cases, the installation of L^AT_EX may erase this file, so you will need to get another copy. A good place to put it is in TEXINPUTS.

You should also install `colordvi.[tex,sty]` and `blackdvi.[tex,sty]` if you have access to color output devices and compatible drivers. You can also install `foilfont.tex` and all the miscellaneous files (appropriate to your system) listed in Table 1 if you wish access to these utilities (but see Section 6.5 before installing any of these files). You might want to put the `README.flt` file someplace more appropriate than TEXINPUTS.

The next step once all the files are available on your system is to invoke `initex` to create the necessary `.fmt` file. (For Arbortext's mu-TeX package, you can also build a `.exe` file.) The following pseudo-bat file is one way to do this. Check the installation documentation for your TeX package for possible alternatives.

```
rem This generates FLTPLAIN.FMT
rem First, get to a clean temporary directory.
cd TEX
md ftdir
cd ftdir
rem Next build the .FMT file.
rem Note, this command does not need to be followed by \dump.
initex fltplain
rem Now put this where it belongs and clean up.
copy fltplain.fmt TEXFORMATS\fltplain.fmt
del fltplain.fmt
del fltplain.log
cd TEX
rd ftdir
rem You can stop here and run FoilTeX with the command
rem TEX &FLTPLAIN <filename>
rem where <filename> is the name of your input file (e.g. SAMPFOIL)
rem OR you can use the FOILTEX.BAT file provided with FoilTeX,
rem if you (modify if necessary and) install that in the
rem appropriate directory.
rem With Arbortext's mu-TeX, you can build a FOILTEX.EXE file
rem which will run a bit faster.
rem
rem Now we build the .EXE file.
preload
rem At the "Template.EXE file" prompt type
TEXBIN\initex.exe
rem At the "Output.EXE file" prompt type
```

```

TEXBIN\foiltex.exe
rem  At the "Preload.FMT file" prompt type
TEXFORMATS\fltplain.fmt
rem  This should do it with FOILTEX.EXE should be installed in
rem  the TEXBIN (you won't need FOILTEX.BAT in this case).

```

Once you have completed this, you can test your setup by going to a directory from which you might ordinarily run \TeX and typing (assuming `TEXDOCS` is in your `TEXINPUTS` search path)

```
foiltex sampfoil
```

This should create the `.DVI` file (and `.AUX` file) which you can preview or print in the usual way, provided all the fonts are there.

To test your font availability, type

```
foiltex foilfont
```

and try previewing and printing.

Note: some drivers (like Arbortext's `PREVIEW` and `DVILASER`) have special files (`*.CF`, `*.FNT`, and `*.OPT` in Arbortext's case) which control the fonts and font paths that the driver can access. You may have to modify these files to include all the additional fonts used by Foil \TeX .

See Section 6.4 if you have any problems with the installation.

6.4 Problems With Installation

If you encounter problems in the basic installation procedure, you should first review this document to make sure that the installation you followed is compatible or appropriate to your \TeX installation. In particular take special note of the beginning of this section concerning "Basic Assumptions". You should also make sure that the scripts or execs or batch files are appropriate for your system. Remember, the biggest problem you might encounter is lack of fonts. Next you might consult your local \TeX perts, or the person who installed \TeX on your system. These are also the people you should contact about font availability. If all else fails, you can contact the author and he might be able to help.

6.5 Installation Warnings

Users are warned that Foil \TeX and Rokicki's `dvi2pdf with dvips` both include versions of the same files. The `POSTSCRIPT` in Foil \TeX supersede those of Rokicki's package (unless he also starts distributing our version of these files). On the other hand, the `colordvi.[tex,sty]` and `blackdvi.[tex,sty]` files should be the same in both Foil \TeX and `dvi2pdf with dvips` but may not be because of the difficulty keeping our two packages in synch. Consequently, care must be taken in installing new versions of either Foil \TeX or `dvi2pdf with dvips` lest the wrong version of some file get installed and the old version lost. If you do find some inconsistencies, please contact (at least) the author of Foil \TeX .

7 Usage Restrictions

Experimental Software Disclaimer

As experimental, research software, this program is provided free of charge on an "as is" basis without warranty of any kind, either expressed or implied, including but not limited to implied

warranties of merchantability and fitness for a particular purpose. IBM does not warrant that the functions contained in this program will meet the user's requirements or that the operation of this program will be uninterrupted or error-free. Acceptance and use of this program constitutes the user's understanding that he will have no recourse to IBM for any actual or consequential damages, including, but not limited to, lost profits or savings, arising out of the use or inability to use this program. Even if the user informs IBM of the possibility of such damages, IBM expects the user of this program to accept the risk of any harm arising out of the use of this program, or the user shall not attempt to use this program for any purpose.

User Agreement

BY ACCEPTANCE AND USE OF THIS EXPERIMENTAL PROGRAM THE USER AGREES TO THE FOLLOWING:

- a. This program is provided for the user's personal, non-commercial, experimental use and the user is granted permission to copy this program to the extent reasonably required for such use.
- b. All title, ownership and rights to this program and any copies remain with IBM, irrespective of the ownership of the media on which the program resides.
- c. The user is permitted to create derivative works to this program. However, all copies of the program and its derivative works must contain the IBM copyright notice, the EXPERIMENTAL SOFTWARE DISCLAIMER and this USER AGREEMENT. Furthermore, the user must document and initial within the program all changes he/she makes.
- d. By furnishing this program to the user, IBM does *not* grant either directly or by implication, estoppel, or otherwise any license under any patents, patent applications, trademarks, copyrights or other rights belonging to IBM or to any third party, except as expressly provided herein.
- e. The user understands and agrees that this program and any derivative works are to be used solely for experimental uses and are not to be sold, distributed to a commercial organization, or be commercially exploited in any manner.
- f. IBM requests that the user supply to IBM a copy of any changes, enhancements, or derivative works which the user may create. The user grants to IBM and its subsidiaries an irrevocable, nonexclusive, worldwide and royalty-free license to use, execute, reproduce, display, perform, prepare derivative works based upon, and distribute, (INTERNALLY AND EXTERNALLY) copies of any and all such materials and derivative works thereof, and to sublicense others to do any, some or all of the foregoing, (including supporting documentation).

8 Acknowledgements, Requests And Help

We would like to thank and acknowledge the following IBM people in IBM for their great assistance in helping to put Foil_{TEX} together: Katherine Hitchcock, Myron Flickner, Ekkehard Blanz, Melanie

Fulgham, Peter Haas, Rocky Bernstein and the many users who contributed their constructive comments on the early test versions within IBM.

A special thanks goes to Tomas Rokicki for implementing our color setup in his driver and another to Sheri Gish of IBM for asking the right (or was it wrong?) question that got this project started.

Foil \TeX is intended to be easy to use, useful and to produce beautiful foils. Consequently, the author welcomes any comments or suggestions.

If you have a question that you can't answer by reading *both* this document and the `dvi2pdf` with `dv` manual, or by posting your question to your local \TeX erts or to the usual forums, you can contact the author.

A Sample Foils

Below is source for a short two page sample foil that demonstrates most of the features of Foil_{TeX}, followed by a facsimile of the output from this source.

```
%%%%%%%% First we load the correct style file
\documentstyle{foils}
%%%%%%%% This first section is for a title page; it is typical LaTeX
\title{Rock protocols for binary Quarries}
%
\author{Fred Flintstone\
Rock Quarry Research Center}
\date{\today}
%%%%%%%% This next command controls part of the footline.
%%%%%%%% Note the ‘‘FoilTeX’’ logo will print automatically.
%\MyLogo{-- Typeset by \FoilTeX\ --}
\Restriction{TUG Use Only}
%
\begin{document}
\maketitle
\begin{abstract} This is where an abstract might go.\end{abstract}
%%%%%%%% This next command starts a new foil with header.
\foilhead{Variability of Rock Quality}
%
What can we prove using only marble rocks?
%%%%%%%% Itemize, mathematics, auto-referencing and footnotes are built-in.
\begin{itemize}
\item  $\Omega(t^2)$  rocks needed \cite{rocky}\footnote{What’s that?}.
\item Worst case structure uses
\begin{equation} \label{equation}
O(n+t\sqrt{t})
\end{equation}
\end{itemize}
%%%%%%%% Here is a sample theorem with proof.
\begin{Theorem} Everything you know about rocks is false.
\end{Theorem}
%
\begin{Proof} The proof is obvious from equation (\ref{equation}).
\end{Proof}
%%%%%%%% Bibliographies work even with BibTeX.
\begin{thebibliography}{99}
%
\bibitem{rocky} Rocky and Bullwinkle, Open problems, in {\sl Mr.
Know-it-all’s Rock Encyclopedia}.
%
\end{thebibliography}
\end{document}
```

Rock protocols for binary Quarries

Fred Flintstone
Rock Quarry Research Center

January 19, 1995

Abstract

This is where an abstract might go.

Variability of Rock Quality

What can we prove using only marble rocks?

- $\Omega(t^2)$ rocks needed [1]¹.
- Worst case structure uses

$$O(n + t\sqrt{t}) \tag{1}$$

Theorem 1. *Everything you know about rocks is false.*

Proof. The proof is obvious from equation (1). \square

References

- [1] Rocky and Bullwinkle, Open Problems, in *Mr. Know-it-all's Rock Encyclopedia*.

¹What's that?