Typesetting Hebrew with LaTeX

Sivan Toledo

School of Computer Science
Tel-Aviv University
Tel-Aviv 69978, Israel
http://www.tau.ac.il/~stoledo

Abstract

This article explains how to use LaTeX to typeset text in the Hebrew script. We use a simple document that contains Hebrew, Greek, and English to illustrate the process and to introduce the main issue involved in Hebrew typesetting: bidirectionality. We then discuss various means for editing input files, including LyX, a free visual LaTeX editor that supports Hebrew, bidirectional editors such as Microsoft Notepad and vim, and left-to-right text editors. Next, we discuss font issues, including availability, glyph repertoires, and font choices for multilingual and math documents. The last issue that the paper addresses involves mechanisms for placing diacritical points in Hebrew.

1. Introduction

The Hebrew script is mainly used to write the Hebrew language, but it is also used to write Yiddish and Ladino. This article focuses on typesetting Hebrew using LaTeX, and more specifically, using LaTeX and Babel. For additional information on the Hebrew script. Let's start with an easy example to appreciate some of the issues that are involved. The input file for our sample document is shown in Figure XXX and the output is shown in Figure 4.

This input file includes three languages: English, Greek, and Hebrew. The first paragraph is a Hebrew one, with a few English words embedded in the first sentence. The second paragraph is an English one, with embedded Hebrew. The third is Greek with Hebrew and the last is Hebrew with Greek. Each new language starts with Babel's `\selectlanguage` command.

The first peculiar thing that you may notice is that the ordering of Hebrew letters and words in the input and output is different. This is a result of the fact that Hebrew is printed from right to left, whereas I typeset the input file with all lines flowing from left to right. For example, the first two letters in this

```
\documentclass[american,greek,hebrew]{article}
\usepackage{babel}
\begin{document}
\sf
\selectlanguage{hebrew}
זוהי פסקה עברית
\L{with some English phrases}.
שנת {\beginL 2001\endL}.

\selectlanguage{american}
This is an English paragraph
\R{עם כמה משפטים בעברית}.
The year 2001.

\selectlanguage{greek}
Aut'o e'inai ellhnik'o ke'imeno kai
\R{עם כמה משפטים בעברית}.

\selectlanguage{hebrew}
זוהי פסקה עברית
\L{{\begin{otherlanguage*}{greek}%
Aut'o e'inai ellhnik'o ke'imeno kai
% \end{otherlanguage*}}}.
שנת {\beginL 2001\endL}.

\end{document}
```

Figure 3: A sample Hebrew-Greek-English LaTeX input file.

זוהי פסקה עברית with some English phrases. שנת 2001.
This is an English paragraph עם כמה משפטים בעברית. The year 2001.
Αυτό είναι ελληνικό κείμενο και עם כמה משפטים בעברית.
זוהי פסקה עברית Αυτό είναι ελληνικό κείμενο και. שנת 2001.

Figure 4: The output that is produced from the input file shown in Figure 3.

document are Zayin followed by a Vav. The Zayin comes before the Vav in the input file (to its left), so it is printed to its right in the output.

The Babel command that switches the text to Hebrew tells LaTeX to print Hebrew paragraphs from right to left. To embed a left-to-right language in a Hebrew paragraph, we used the `\L` command, whose argument is the text to be embedded. This command switches directions and also switches the Babel language in use and the font encoding. Numbers are printed in Hebrew from left to right. To typeset numbers, we introduce a new group and surround the number with `\beginL` and `\endL` commands, as in `{\beginL 2001\endL}`. The commands `\beginL` and `\endL` are eTeX primitives and they tell TeX to start and end a left-to-right typesetting. Unlike the `\L` macro, these primitives do not change the Babel language and the font encoding. (The origin of these primitives dates back to an earlier bidirectional TeX system.)

To switch to Hebrew inside a left-to-right paragraph, we use the `\R` macro.

The need to switch directions even in documents that contains only Hebrew and numbers mean that Hebrew, like Arabic, Persian and several other languages, is *bidirectional*. The bidirectionality requires the use of the four primitives `\beginL`, `\endL`, `\beginR`, and `\endR`. These primitives are not part of TeX, but they are part of eTeX, eLaTeX, pdfeTeX, and pdfeLaTeX, which are included in standard TeX distributions (I use the teTeX distributions that come with RedHat Linux without any modifications).

The last paragraph shows another fine point of the Babel bidirectional macros. The `\L` and `\R` macros switch the language to specific left-to-right and right-to-left languages, so to switch to another language, we use the standard Babel environment `otherlanguage*`.

Note that we use a 7-bit encoding for the Greek text. The input file is encoded in the iso8859-8 encoding (equivalent in this case to Microsoft's code page 1255; cp1255 includes vowel points, which iso8859-8 does not). Like most 8-bit encoding, iso8859-8 uses the range 0–127 to encode ASCII and the range 128–255 to encode Hebrew. It does not encode Greek, so we use a Greek encoding that assigns Greek letters to the 0–127 ASCII range. Using iso8859-8 to encode the Hebrew text and iso8859-7 to encode the Greek in a single 8-bit input file is not supported by editors. Omega should allow us in the future to use Unicode, a 16-bit encoding, to create multilingual documents, but currently it is not as robust for bidirectional texts as eTeX.

Two issues are worth pointing out about the Babel's Hebrew support before we move on. First, it support correct typesetting in all the standard LaTeX environment, which means that in lists the item labels appear on the right, and so on. Second, it is not bundled with Babel's distribution, so you have to get these support files separately.

Most of the font files and support files that are discussed in this article are available in a single archive file from my web page (`www.tau.ac.il/stoledo`, click on Tools). The only files that are not included are commercial fonts. The archive also contains examples in LyX format, which you should be able to load into LyX and use as starting points. I have also included the LaTeX files that LyX produces, in case you do not wish to use LyX but edit the LaTeX files directly. The archive includes a readme file with detailed instructions.

2. Preparing the Input: LyX versus Text Editors

LyX (`www.lyx.org`) is probably the best tool for preparing Hebrew LaTeX input files. LyX is a free what-you-see-is-what-you-mean editor; that is, it displays the document in a similar fashion to the final LaTeX output, but without paying attention to line breaks, hyphenation, and so on. It also allows you to distinguish between font families (serif, sans serif, and typewriter), styles, and weights, but the specific fonts that you see on the screen may be different than the ones TeX will use in the final output. Most of the time this is useful, since you can use the best screen fonts that you have in LyX, while using another set of fonts for the TeX output. Figure 5 shows a typical LyX window. I now use LyX almost exclusively for all my document preparation (I sometimes modify slightly the LaTeX output to conform to the standards of a specific journal, but only for the final version of a document).

LyX is so useful for Hebrew because it displays bidirectional text correctly, as Figure 6 shows. Entering Hebrew text from left to right using a text editor without special support for Hebrew is possible, but difficult. Using LyX is much easier.

The Hebrew (and Arabic) support for LyX was implemented by Dekel Tzur, a Computer-Science PhD student in Tel-Aviv University. The Hebrew support is included in the standard LyX distributions, so if you have LyX, you have a bidirectional editor for entering and editing Hebrew. To use LyX to edit Hebrew documents, you need to:

— Enable right-to-left typesetting in your `lyxrc` file.
— Set the screen fonts to fonts that can display Hebrew, either in the Options menu or in your `lyxrc` file.
— Declare the document language to be Hebrew (in the Document dialog, under the Layout menu) if the primary language of the document is Hebrew.
— Declare two keyboards, one for Latin languages and one for Hebrew, in the `lyxrc` file.

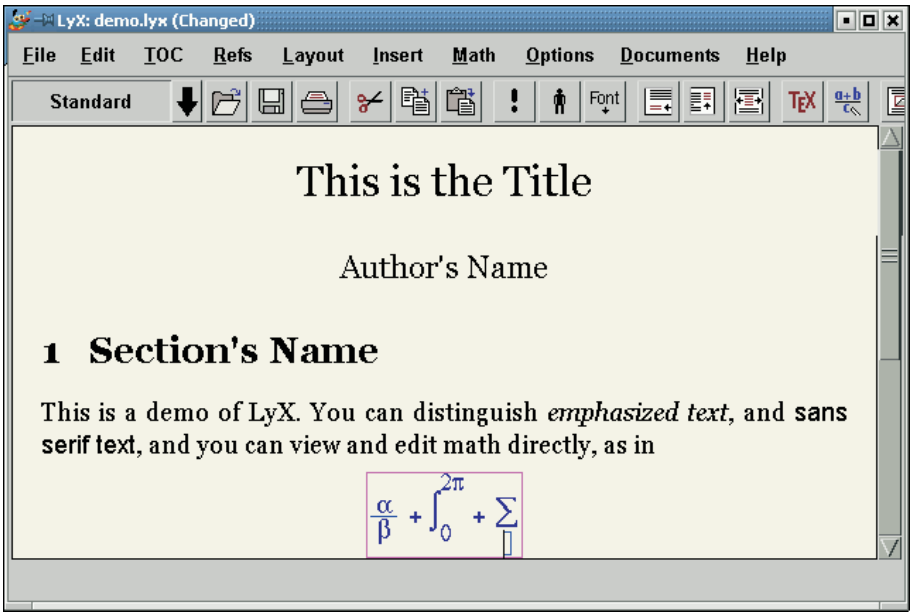

Figure 5: LyX, an editor for LaTeX documents, displaying a sample English document.

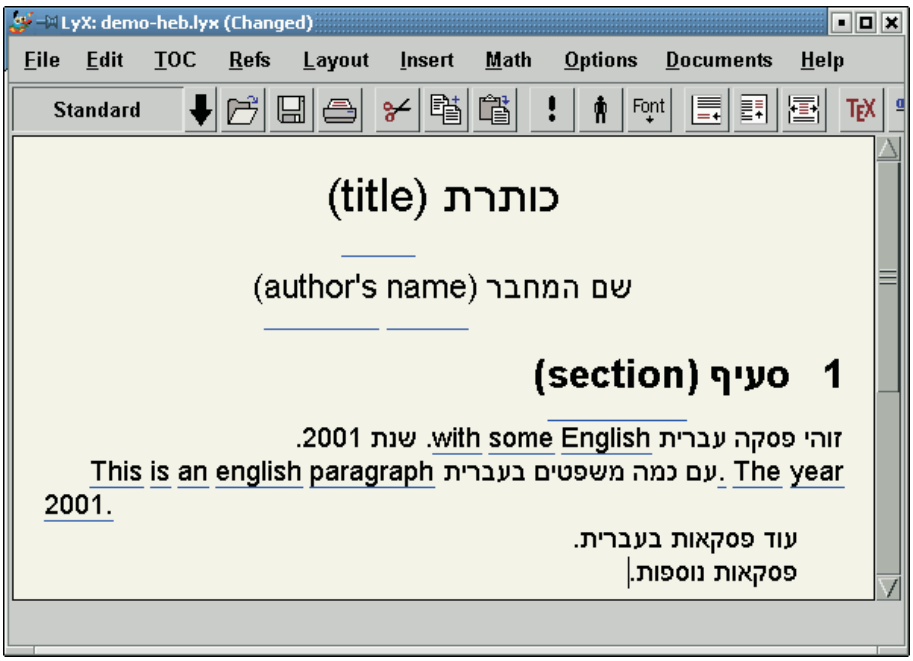

Figure 6: LyX displaying a Hebrew sample document.

— Bind a key to the LyX command that toggles the language between Hebrew and another language, typically American (that is, US English). This is done in the `lyxrc` file.

Here are the relevant lines from my `lyxrc` file:

```
\rtl true
\screen_font_encoding iso8859-8
\screen_font_roman "-*-times new roman"
\screen_font_sans  "-*-arial"
\screen_font_typewriter "-*-courier new"
\kbmap true
\kbmap_primary null
\kbmap_secondary hebrew
\bind "F12" "language hebrew"
```

The iso8859-8 font encoding does not encode vowel points (see Section 4), so if your document includes vowel points, you should use the microsoft-cp1255 screen-font encoding.

There are several other options for entering and editing Hebrew LaTeX documents. Since I use LyX almost exclusively I will not go into detail regarding these other options, but I do want to mention their availability.

The simplest option is to use a text editor that can use a Hebrew font but that does not support bidirectionality. The input file will appear like the one in Figure 3. It will be hard to read and edit the Hebrew text, since it will be displayed from left to right.

The second option is to use a text editor that allows the user to choose between left-to-right and right-to-left ordering of all the text. Such an editor allows you to ''reflect'' all the lines. That is, you can flip the display so that all lines flow from right to left. In the default mode all the Latin text (also Greek and Cyrillic) and the TeX markup are displayed correctly, but the Hebrew is not. In the reflected right-to-left mode, the Hebrew is displayed correctly but the Latin and markup are not. One editor that supports this feature is `vim`, a free clone of `vi`. As in LyX, this feature is part of the standard source distribution of the software. However, I believe that in some binary distributions the Hebrew support is turned off at compile time, so you may have to build `vim` from the sources to activate this feature. `Vim` also supports another Hebrew-related feature, called push mode. Some people use it to mix Hebrew and English in their correct direction on the screen, but this required a special preprocessor so I will not go into further details.

A final option is to use a Unicode text editor that implements the Unicode Bidirectional algorithm. Figure 7 shows our example input file as displayed by `Notepad`, an accessory text editor that can display and edit Unicode text on Windows 2000 and Windows NT. It is bundled with Microsoft's Windows operating systems. As you can see, the Hebrew text is ordered correctly. However, the editor applies the bidirectional reordering not only to the text, but also to the LaTeX markup. This can be quite confusing, especially when the main language of the document is Hebrew. Microsoft Word 2000 can also edit bidirectional Unicode text files. It seems that in the not-so-distant future both `emacs` and KDE editors will also support Unicode bidirectionality.

3. Hebrew Fonts

This section addresses a major issue in the typesetting of most non-Latin languages, the issue of fonts.

A Little Background

Let us start with a little background on Hebrew text fonts. Tables 4 and 5 display most of the fonts that I discuss below.

Frank-Rühl is the ubiquitous Hebrew text font style. There are many fonts that belong to this style, and all are based on a turn-of-the-20th-century design by Raphael Frank. Some of the fonts are actually called Frank-Rühl and some are not. It is used in most of the newspapers, magazines, and books printed in Israel. It was originally designed in a single weight. In typical newspaper use, a different typeface, Drugulin, is used for emphasis. Drugulin is quite similar in style to Haralambous' Tiqwah.

David and Hadassah are 1950's designs by Ismar David and Henri Friedlaender, respectively. Narkissim is a 1980's design by Zvi Narkiss, a prolific Hebrew type designer. All three are serifed text faces suitable for extended reading. David is the most uneven of the three. David was originally designed in one weight only, but bold weights are widely available now. Hadassah was designed in two weights, normal and bold. Narkiss designed several weights of Narkissim, three of which are shown in Table 4. Narkiss also designed oblique fonts that slant either left or right. The left slant fits the writing direction better (the letters lean forward), the right slant matches Latin italics better. Most of the other oblique Hebrew fonts are mechanically slanted.

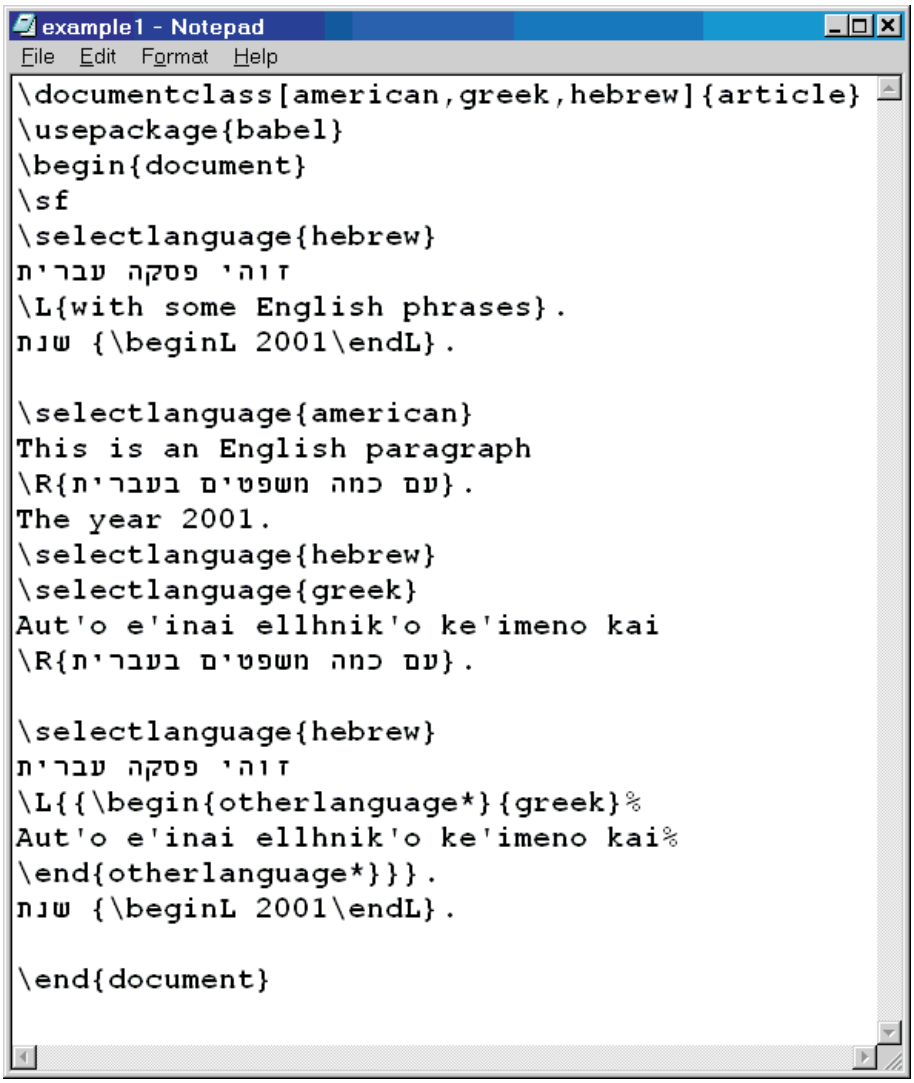

Figure 7: The `Notepad` text editor on Windows 2000 and Windows NT can display bidirectional text.

A true italic style was only designed for David; it is not shown here.

Narkiss designed another excellent serifed family, called Narkiss (a newer variant is called Narkiss Classic), which is not shown here.

Narkiss Tam, also by Zvi Narkiss, is the most even and regular Hebrew sans serif font. It comes in many weights, including a useful black, and in slanted versions. There are several other excellent Hebrew sans serif, but most are not as suitable for extended blocks of text as Narkiss Tam. Arial is not nearly as good, primarily since some of the letters, like Gimel, have somewhat odd shapes. Still, it is ubiquitous on personal computers so it is common to see it in printed documents.

Font Availability

Table 4 shows many of the Hebrew fonts that are used with TEX and LATEX. Table 5 shows longer samples of a few. The fonts fall into several classes:

— Commercial fonts from a large Israeli font distributor, Masterfont (`www.masterfont.co.il`). Masterfont distributes a few more families of text fonts, and hundreds of display fonts. My school purchased licenses to the fonts shown here.

— Metafont fonts, including Tiqwah, a family that Yannis Haralambous developed for typesetting bibles, and several free fonts by anonymous implementors. Most of the latter ones are digitizations of commercial fonts. Some come in only one weight and style, and they are, generally speaking, of lesser quality than the commercial fonts.

— TrueType fonts that Microsoft bundles with its operating systems and also distributes freely on its typography web site, `www.microsoft.com/typography`. Linux users can extract the fonts from the archives using a program called `cabextract`. These fonts are licensed from Monotype (now owned by Agfa). The other font families that Microsoft distributes freely, such as Georgia and Verdana, do not contain glyphs for the Hebrew letters. Microsoft bundles with Windows another sans serif font family that includes Hebrew glyphs, Tahoma. This family was designed primarily for the screen; it does not look particularly good on paper.

I think that it is fair to say that the quality of the commercial fonts is better than that of the Metafonts. This is particularly true for David; its Metafont version is highly irregular. Frank-Rühl 3 is better (the Metafont source is called DeadSea). The other Metafont versions of Frank-R\"uhl are not as good, in my judgment. The Metafont version of Narkiss Tam is also quite good.

Commerial Fonts Distributed by Masterfont			
Aharoni	תל־אביב 123 !?	**תל־אביב 123 !?**	**תל־אביב 123 !?**
Alexandra	*תל־אביב 123 !?*		
David	תל־אביב 123 !!	*תל־אביב 123 !!*	**תל־אביב 123 !!**
Frank-Ruhl	תל־אביב 123 !?	*תל־אביב 123 !?*	**תל־אביב 123 !?**
Hadassah	תל־אביב 123 !?	*תל־אביב 123 !?*	**תל־אביב 123 !?**
Narkissim	תל־אביב 123 !?	*תל־אביב 123 !?*	**תל־אביב 123 !?**
	תל־אביב 123 !?	תל־אביב 123 !?	
Narkiss Tam	**תל־אביב 123 !?**	*תל־אביב 123 !?*	**תל־אביב 123 !?**
	תל־אביב 123 !?	תל־אביב 123 !?	**תל־אביב 123 !?**
Metafonts by Yannis Haralambous			
Tiqwah	תל–**אביב** 123 !?	תל–**אביב** 123 !?	
Anonymous Metafonts			
Frank-Ruhl 1	תל–אביב **123** !?	*תל–אביב 123 !?*	תל–אביב 123 !?
Frank-Ruhl 2	תל–**אביב 123** !?	*תל–אביב 123 !?*	**תל–אביב** 123 !?
Frank-Ruhl 3	תל-**אביב** 123 !?		
David	תל-אביב 123 !?		
Miryam	תל-אביב 123 !?		
Miryam Wide	תל-אביב 123 !?		
Narkiss Tam	תל–אביב **123 !?**		
Microsoft's Free-Distributed TrueType Fonts			
Arial	תל–אביב 123 !?	*תל–אביב 123 !?*	**תל–אביב 123 !?**
Times New	תל–אביב 123 !?	*תל–אביב 123 !?*	**תל–אביב 123 !?**
Courier New	תל–אביב 123 !?		**תל–אביב 123 !?**

Table 4: Hebrew fonts that can be used with LaTeX. Some of the families shown have bold-italic fonts, which are not shown in the table.

Narkissim	Frank-Rühl 3 (MF)	Frank-Rühl
מיתוג תכוף ומיותר בין חוטים עלול לקרות גם כאשר חוטים רבים מתחרים על מעבדים מעטים. נניח שתוכנית שרצה על מחשב עם שני מעבדים מריצה 100 חוטים שכמעט אינם ממתינים למנעולים ולאירועים. מערכת ההפעלה תמתג בין כל החוטים שאינם ממתינים על מנת לתת לכל אחד חלק פחות או יותר שווה של המעבדים. כאשר מספר גדול של חוטים מתחרים על המעבדים, התקורה של מיתוג בינהם עלולה לצרוך את רוב משבי המחשב. אי לכך, עדיף בדרך כלל ליצור מספר חוטים כזה שמספר החוטים הלא ממתינים יהיה קרוב למספר המעבדים. זה עלול לפגוע במודולריות הטמפורלית אך עשוי לשפר ביצועים.	מיתוג תכוף ומיותר בין חוטים עלול לקרות גם כאשר חוטים רבים מתחרים על מעבדים מעטים. נניח שתוכנית שרצה על מחשב עם שני מעבדים מריצה 100 חוטים שכמעט אינם ממתינים למנעולים ולאירועים. מערכת ההפעלה תמתג בין כל החוטים שאינם ממתינים על מנת לתת לכל אחד חלק פחות או יותר שווה של המעבדים. כאשר מספר גדול של חוטים מתחרים על המעבדים, התקורה של מיתוג בינהם עלולה לצרוך את רוב משבי המחשב. אי לכך, עדיף בדרך כלל ליצור מספר חוטים כזה שמספר החוטים הלא ממתינים יהיה קרוב למספר המעבדים. זה עלול לפגוע במודולריות הטמפורלית אך עשוי לשפר ביצועים.	מיתוג תכוף ומיותר בין חוטים עלול לקרות גם כאשר חוטים רבים מתחרים על מעבדים מעטים. נניח שתוכנית שרצה על מחשב עם שני מעבדים מריצה 100 חוטים שכמעט אינם ממתינים למנעולים ולאירועים. מערכת ההפעלה תמתג בין כל החוטים שאינם ממתינים על מנת לתת לכל אחד חלק פחות או יותר שווה של המעבדים. כאשר מספר גדול של חוטים מתחרים על המעבדים, התקורה של מיתוג בינהם עלולה לצרוך את רוב משבי המחשב. אי לכך, עדיף בדרך כלל ליצור מספר חוטים כזה שמספר החוטים הלא ממתינים יהיה קרוב למספר המעבדים. זה עלול לפגוע במודולריות הטמפורלית אך עשוי לשפר ביצועים.

Hadassah	David (MF)	David
מיתוג תכוף ומיותר בין חוטים עלול לקרות גם כאשר חוטים רבים מתחרים על מעבדים מעטים. נניח שתוכנית שרצה על מחשב עם שני מעבדים מריצה 100 חוטים שכמעט אינם ממתינים למנעולים ולאירועים. מערכת ההפעלה תמתג בין כל החוטים שאינם ממתינים על מנת לתת לכל אחד חלק פחות או יותר שווה של המעבדים. כאשר מספר גדול של חוטים מתחרים על המעבדים, התקורה של מיתוג בינהם עלולה לצרוך את רוב משבי המחשב. אי לכך, עדיף בדרך כלל ליצור מספר חוטים כזה שמספר החוטים הלא ממתינים יהיה קרוב למספר המעבדים. זה עלול לפגוע במודולריות הטמפורלית אך עשוי לשפר ביצועים.	מיתוג תכוף ומיותר בין חוטים עלול לקרות גם כאשר חוטים רבים מתחרים על מעבדים מעטים. נניח שתוכנית שרצה על מחשב עם שני מעבדים מריצה 100 חוטים שכמעט אינם ממתינים למנעולים ולאירועים. מערכת ההפעלה תמתג בין כל החוטים שאינם ממתינים על מנת לתת לכל אחד חלק פחות או יותר שווה של המעבדים. כאשר מספר גדול של חוטים מתחרים על המעבדים, התקורה של מיתוג בינהם עלולה לצרוך את רוב משבי המחשב. אי לכך, עדיף בדרך כלל ליצור מספר חוטים כזה שמספר החוטים הלא ממתינים יהיה קרוב למספר המעבדים. זה עלול לפגוע במודולריות הטמפורלית אך עשוי לשפר ביצועים.	מיתוג תכוף ומיותר בין חוטים עלול לקרות גם כאשר חוטים רבים מתחרים על מעבדים מעטים. נניח שתוכנית שרצה על מחשב עם שני מעבדים מריצה 100 חוטים שכמעט אינם ממתינים למנעולים ולאירועים. מערכת ההפעלה תמתג בין כל החוטים שאינם ממתינים על מנת לתת לכל אחד חלק פחות או יותר שווה של המעבדים. כאשר מספר גדול של חוטים מתחרים על המעבדים, התקורה של מיתוג בינהם עלולה לצרוך את רוב משבי המחשב. אי לכך, עדיף בדרך כלל ליצור מספר חוטים כזה שמספר החוטים הלא ממתינים יהיה קרוב למספר המעבדים. זה עלול לפגוע במודולריות הטמפורלית אך עשוי לשפר ביצועים.

Arial (TTF)	Narkiss Tam (MF)	Narkissm Tam
מיתוג תכוף ומיותר בין חוטים עלול לקרות גם כאשר חוטים רבים מתחרים על מעבדים מעטים. נניח שתוכנית שרצה על מחשב עם שני מעבדים מריצה 100 חוטים שכמעט אינם ממתינים למנעולים ולאירועים. מערכת ההפעלה תמתג בין כל החוטים שאינם ממתינים על מנת לתת לכל אחד חלק פחות או יותר שווה של המעבדים. כאשר מספר גדול של חוטים מתחרים על המעבדים, התקורה של מיתוג בינהם עלולה לצרוך את רוב משבי המחשב. אי לכך, עדיף בדרך כלל ליצור מספר חוטים כזה שמספר החוטים הלא ממתינים יהיה קרוב למספר המעבדים. זה עלול לפגוע במודולריות הטמפורלית אך עשוי לשפר ביצועים.	מיתוג תכוף ומיותר בין חוטים עלול לקרות גם כאשר חוטים רבים מתחרים על מעבדים מעטים. נניח שתוכנית שרצה על מחשב עם שני מעבדים מריצה 100 חוטים שכמעט אינם ממתינים למנעולים ולאירועים. מערכת ההפעלה תמתג בין כל החוטים שאינם ממתינים על מנת לתת לכל אחד חלק פחות או יותר שווה של המעבדים. כאשר מספר גדול של חוטים מתחרים על המעבדים, התקורה של מיתוג בינהם עלולה לצרוך את רוב משבי המחשב. אי לכך, עדיף בדרך כלל ליצור מספר חוטים כזה שמספר החוטים הלא ממתינים יהיה קרוב למספר המעבדים. זה עלול לפגוע במודולריות הטמפורלית אך עשוי לשפר ביצועים.	מיתוג תכוף ומיותר בין חוטים עלול לקרות גם כאשר חוטים רבים מתחרים על מעבדים מעטים. נניח שתוכנית שרצה על מחשב עם שני מעבדים מריצה 100 חוטים שכמעט אינם ממתינים למנעולים ולאירועים. מערכת ההפעלה תמתג בין כל החוטים שאינם ממתינים על מנת לתת לכל אחד חלק פחות או יותר שווה של המעבדים. כאשר מספר גדול של חוטים מתחרים על המעבדים, התקורה של מיתוג בינהם עלולה לצרוך את רוב משבי המחשב. אי לכך, עדיף בדרך כלל ליצור מספר חוטים כזה שמספר החוטים הלא ממתינים יהיה קרוב למספר המעבדים. זה עלול לפגוע במודולריות הטמפורלית אך עשוי לשפר ביצועים.

Table 5: A comparison of several Hebrew text fonts. The table shows a paragraph set in nine different fonts, including commercial fonts, Metafonts (marked MF), and a TrueType font (marked TTF). The text is set in 8.5/9 points, except Arial, which is set in 8/9 points.

There are several additional sources of Hebrew fonts that are worth mentioning. A German foundry, Elsner and Flake, produced an excellent set of Hebrew Text fonts in Macintosh Type1 format. Two families are available from the Linotype library. The other families are available from an Israeli distributor, Panergy (`www.panergy.co.il`).

Fontworld (`www.fontforld.com`) is a foundry that produces many Hebrew text fonts, which they sell them directly to end users. I have not used their fonts so I cannot comment on their quality.

Microsoft bundles a few more Hebrew fonts with the Hebrew-localized versions of its operating systems. The company also bundles a large number of Hebrew fonts with the Hebrew-localized versions of Microsoft Office. The quality of some of these is good, including Aharoni (a single weight), David, Monotype Hadassah, and Monotype Levenim (a version of Miryam). The other text families, in my judgment, are too poor for high-quality text typesetting.

A few more high-quality TrueType fonts used to be distributed with Apple's Hebrew Language Kit; I think that this software is now discontinued but I am not sure. These included New Peninim (a version of Frank-Rühl, with glyph similar or identical to the Hebrew glyphs in Times New Roman), Arial (same as Microsoft's version except that the numerals and punctuation are better proportioned with respect to the Hebrew letters), and Hebrew Corsiva.

Technicalities

Some preparation is necessary before most of these fonts can be used with TeX. The issues that are involved are not unique to Hebrew, so I will just go over these preparations procedures quickly. The Metafont-format fonts are easiest to use. On many distributions, TeX and `dvips` can generate the `tfm` files and the bitmap font files automatically, so all you need to do is put the Matafont sources in a directory where they can be found. Fonts in Macintosh format must be converted to PC format before they can be used on Unix or Windows. There are several tools on the market that can perform this conversion; I use `refont`, an easy-to-use DOS program. For PostScript fonts, you will need to prepare or obtain an encoding file. Since many fonts (including Microsoft's and Monotype's) use nonstandard glyph names, a single encoding file cannot support all of these fonts. I use three encoding files, one for the fonts that IBM distributes freely for OS/2 and bundles with AIX, its Unix operating system; another for Microsoft's fonts, and a third for Masterfont's commercial fonts. For TrueType fonts, you will need to prepare a `tfm` file and usually also an `pfa` file that `dvips` can use. The `tfm` file can be prepared using `ttf2afm` and `afm2tfm`. If you use pdfeTeX or pdfeLaTeX, then the `tfm` and an encoding file

are all you need. If you use `dvips` then you also need to convert the font to a PostScript format that `dvips` can include in the PostScript file. Several utilities, such as `ttfps`, can perform the conversion. I have found that a Linux graphical program called `gfontview` performs the conversion particularly well (`ttfps` sometimes produces incorrect PostScript fonts from large TrueType fonts).

Glyph Repertoire

Early Hebrew typefaces, including Frank-Rühl and Miryam, were designed with a limited set of glyphs that did not include punctuation marks and numerals. When these glyphs were needed, they were borrowed from Latin fonts. Over time, designers added appropriate numerals and punctuation to old typefaces and started designing new typefaces with a complete set of glyphs. For example, matching numerals were added to Miryam; Hadassah was designed with a complete glyph set, including numerals, punctuation, and vowel points (see Section 4).

The numerals and punctuation in Hebrew typefaces match the letters in weight, size, and overall style. Numerals that do not match diminish the visual quality of the text.

Some digital fonts include numerals and punctuation that do not match the Hebrew letters. Usually the mismatch is a result of an attempt to include a single glyph for each letter in a multilingual font. Arial and Times New Roman, for example, include glyphs for Latin, Greek, Cyrillic, Hebrew, and Arabic. The numerals in these fonts match the Latin, Greek and Cyrillic letters, but they are too tall for the Hebrew. The sample of Arial in Table 5 shows this phenomenon. The digital fonts that IBM distributes for OS/2 and for AIX suffer from the same problem. In most of these cases the foundry has appropriate glyphs for Hebrew numerals, but they do not include them in the final fonts. Such fonts are less suitable for high-quality typesetting than fonts with appropriately designed numerals.

Typeface Choice for Multilingual Documents

There are several issues to bear in mind when choosing fonts for multilingual documents.

Hebrew typefaces are usually heavier than the common Latin typefaces. This is particularly true for Frank-Rühl, but it is also true for Hadassah and Narkissim. David is lighter than these three. There is a light version of Narkissim, but it is not the ''normal'' weight. In light of the widespread use of Frank-Rühl, it is reasonable to say that the Israeli eye is used to darker pages

than the eye of an American, say. Therefore, if the Hebrew and Latin are to achieve a similar color on the page, one should avoid light Latin typefaces. In particular, Computer Modern is usually too light; if you use it, try to match it to David or to Narkissim Light. Another consideration related to weight is the number of different weights in a typeface family. When either the Hebrew or Latin families contains several weights (or continuous weights in a multiple master font), it is easier to match weights than when both families include a single text weight and a single bold weight.

Hebrew has a single case of letters; there are no capitals. Furthermore, there is only a single ascender and a few descenders. To match the Latin typeface in apparent size, the Hebrew letters should typically be slightly taller than the Latin small letters. When designing a document style, match the sizes carefully; do not rely on the design sizes. A Hebrew font that is nominally 10 points may be too small or two large than a 10 point Latin fonts. (The same applies to matching typefaces of a single script, of course). The easiest way to do this is using the scaling operator of LaTeX's new font selection scheme. Another outcome of the single case of Hebrew letters is that Latin typefaces with a large x-height and short Capitals usually match Hebrew typefaces better than typefaces with small x-height and/or tall capitals.

The overall texture of Hebrew text is also quite different than that of Latin, Greek, and Cyrillic. This is partially due to the fact that the serifs in Hebrew are not horizontal but vertical, and partially due to the fact that Hebrew typefaces have much less contrast than Latin ones. Therefore, if you want to match the texture as closely as possible, you should consider not only standard Latin typefaces, but also ''fringe'' ones. You can be more liberal in a document that is primarily Hebrew and that do not contain large blocks of Latin text; you should be more conservative in a primarily Latin document.

My favorite combination is Hadassah with Raleigh. Raleigh is primarily a display typeface, but it is regular enough to be used as a text face. It has little contrast, and it comes in several weights (6 in the Bitstream family that I use). It's capitals are not too tall.

Another issue to consider beyond the choice of individual typefaces is the number of different typeface families in a document. A document that uses many different typefaces looks cluttered. A multilingual document that contains Hebrew almost always uses many typefaces, since there are almost no typefaces with glyphs in Hebrew and another script (Lucida Sans, Arial, and Tahoma being the only exceptions). To keep the design clean and uncluttered, you should minimize the number of different fonts. For example, I tend to use sans serifs only in headings and only in very heavy weights (e.g., Narkiss Tam Black and Helvetica Black) or not at all, to use a fixed-width fonts only for

Latin and only in a single weight, and to avoid italics and obliques altogether, except in primarily Latin bibliographies.

Matching Hebrew and Math Typefaces

Hebrew documents with a significant amount of mathematics are even more difficult to design than other multilingual documents. Two factors contribute to this difficulty: the relatively small variety of easy-to-use math typefaces and the traditional use of italics in mathematical notation. The lack of variety poses a problem since it makes it harder to choose matching Hebrew and math typefaces. The use of italics poses a problem since a Latin or Greek italic letter embedded in a Hebrew text leans backwards, in a direction opposite to the flow of text.

Before continuing, I should say that most Hebrew TEX users use Computer Modern Math. The resulting documents are readable, although I think that they are not as beautiful as they can be.

The only math typeface family that I am aware of that is not slanted is AMS Euler, and that is what I normally use. The fonts, which were designed by Zapf in consultation with Knuth, are excellent and free. I usually use them with Hadassah and Raleigh, but Euler also works well with David and Narkissim.

Copyrights

Many Hebrew fonts are unauthorized digitizations of copyrighted designs. The digitization, distribution, sale, and use of such fonts may or may not be legal. I am not a an expert on the legal issues, but I think that unauthorized digitizations should be avoided from a moral standpoint. Designing and producing text typefaces requires a lot of hard work. The designers and producers should be able to benefit from their work if they choose to. In a language like Hebrew that serves a small population, it is even harder to profit from font production than in Latin languages. Some of the Metafonts, in particular Narkiss Tam, are unauthorized copies. Some of the commercial foundries also sell unauthorized copies. Fontworld, for example, sells Neshika, which appears to be an unauthorized copy of Narkiss Tam. I encourage you to consider this issue when choosing fonts.

4. Vowel Points

The Hebrew script uses diacritical points (Nikud in Hebrew) as vowels or parts of vowels, to distinguish between consonants that are written with the same letter, and for other grammatical distinctions that are not articulated. Due to the historical difficulties and to the difficulties of spelling correctly with vowel points (most Hebrew speakers cannot spell correctly with vowel points), vowel points are not used in general printing. There is an official orthography for Hebrew without vowel points. Vowel points are used in three main settings:

— When disambiguity and correct pronunciation are important, such as in poetry, prayer books, and bibles.

— In children's books.

— In general printing to disambiguate between words what have the same spelling without vowels, or to specify the pronunciation of unfamiliar foreign words and names. In such settings, vowel points are used on very few words in the text.

Because vowel points are used rarely, typesetting systems and fonts that do not support them are quite usable. For example, most display fonts, even commercial ones, do not include vowel points.

Printed bibles include another set of diacritical points called Te'amim. They make text even harder to set. Since they are needed only for typesetting Bibles (and perhaps academic articles concerning the Bible), very few fonts and typesetting systems support them, so I exclude them from further discussion. For a complete discussion, of the issues that are involved and of Tiqwah, a TeX-based system that can typeset te'amim.

Vowel points pose a challenge to typesetting systems since they need to be placed carefully with respect to the letters, but there is not enough space in an 8-bit font for all the combinations (a single letter can carry up to two points). In some cases it may be desirable to increase the spacing between characters when certain vowel points are attached to a character (say a wide vowel point to a narrow character), but in general one may treat the vowel points as zero-width combining characters with good results.

There are several ways to produce Hebrew with vowel points with TeX, but none of them is standardized or well-supported. The simplest way is simply to use a font in which vowel points have zero advance width and in which they print to to right of the insertion point. The commercial Fonts that we have purchased are produced in this way. The placement that this produces is not optimal, since a vowel point is placed in the same way with

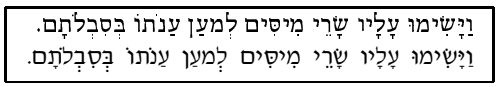

Figure 8: Hebrew with vowel points. The top line is set in Hadassah; the vowel placement mechanism uses Ligatures. The bottom line is set in Omega Hebrew, a digitization of David; the vowel placement uses a special PostScript Type3 font, as explained in my TUGboat article. In both cases, the vowels are well placed.

respect to all the letters, but it is better than nothing. We actually use a better way. It turns out that the Macintosh Hebrew fonts that we purchased come with four sets of vowel points. Each set is designed to fit a specific subset of letters. Additionally, there are precomposed letter-vowel combinations for several hard-to-place combinations. I built special ligature tables to select an appropriate vowel point for each letter, and to use the precomposed glyphs. Figure 8 shows a sentence that I set in Hadassah using this mechanism.

Two other mechanisms for placing vowels were described by Yannis Haralambousand by me. Figure 8 shows an example typeset using my technique.

The reader should note that kerning tables cannot be used to place vowel points. Kerning tables in PostScript, TrueType, and TEX fonts tell the typesetting program to move pairs of letters closer or further apart. If we used this mechanism to place vowels, the kerning between a letter and a vowel point would affect the placement of subsequent letters, not only the placement of the vowel point. This is undesirable.

Another comment worth making is that the OpenType font standard contains a sophisticated mechanism for accurate positioning of diacritical points. OpenType is an extension of TrueType fonts. Microsoft and Adobe have produced several such fonts, although none use the OpenType positioning mechanism to place Hebrew vowel points. Adapting TEX to use the positioning information in OpenType fonts would simplify the setting of complex scripts such as Hebrew, Arabic, and Indic.