acolor.sty
Colors for Arab\TeX

Karel Mokrý Otakar Smrž

June 19, 2007

We used to have a problem—how to colorize the diacritics typeset by Arab\TeX. So we decided to improve Arab\TeX's macros a little. This is done via a small style file acolor.sty. We hope you will find it useful.

The acolor.sty package depends on xcolor.sty, and of course, you should have Arab\TeX installed... Usage is quite easy. Put acolor.sty to some place where \TeX can find it, and in your document, type \usepackage{acolor} after having included Arab\TeX.

The acolor.sty package provides the \acolor{group}{color} command which you can use to set colors for individual types of graphemes or their groups. The effect of acolor.sty can be turned on and off with \acoloron and \noacolor:

\begin{verbatim}
\gdef\acoloron{\def\acolor##1##2{\textcolor{acolor4##1}{##2}}}
\gdef\noacolor{\def\acolor##1##2{##2}}
\end{verbatim}

\begin{verbatim}
\gdef\acolorlet#1#2{\acolor{#1}{acolor4#2}}
\gdef\acolor#1#2{\ifthenelse{\equal{#1}{everything}}{\acolor{letters} {#2}\
\acolor{omissibles} {#2}}{\ifthenelse{\equal{#1}{letters}}{\acolor{isolated} {#2}\
\acolor{initial} {#2}\
\acolor{medial} {#2}\
\acolor{final} {#2}}{\ifthenelse{\equal{#1}{omissibles}}{\acolor{diacritics} {#2}\
\acolor{hamzamadda} {#2}}{\ifthenelse{\equal{#1}{diacritics}}{\acolor{vowelmarks} {#2}\
\acolor{emptymarks} {#2}\
\acolor{shadda} {#2}}{\ifthenelse{\equal{#1}{vowelmarks}}{\acolor{fatha} {#2}\
\acolor{kasra} {#2}\
\acolor{damma} {#2}\
\acolor{quran} {#2}\
\acolor{ammad} {#2}\
\acolor{zwarakay} {#2}\
\acolor{bars} {#2}}{\ifthenelse{\equal{#1}{emptymarks}}{\acolor{sukun} {#2}\
\acolor{wasla} {#2}}{\ifthenelse{\equal{#1}{hamzamadda}}{\acolor{hamza} {#2}\
\acolor{madda} {#2}}{\colorlet{acolor4#1}{#2}}}}}}}}
\end{verbatim}

\url{ftp://ftp.informatik.uni-stuttgart.de/pub/arabtex/arabtex.htm}
\url{http://www.ucw.cz/~karel/\textit{with updates on http://sf.net/projects/encode-arabic/}}
The color definitions are the responsibility of the xcolor.sty package, esp. its \colorlet command. This means that color settings are valid within the enclosing \TeX{} group only, which seems a very desirable feature. From the source code above, you can see how groups of graphemes are defined, and what dependencies there are among them.

sample text

\texttt{\acolor{diacritics}{red}}

\texttt{\acolor{omissibles}{white}}

\texttt{\acolor{hamza}{red!30!yellow}}

\texttt{\acolor{vowelmarks}{blue}}

\texttt{\acolor{emptymarks}{red!90!blue}}

\texttt{\acolor{isolated}{gray}}

For assigning the values of ‘acolors’ to themselves, it is advisable to use the \acolorlet{\textit{group}}{\textit{grapheme}} macro instead of the package-internal prefix for color names disclosed with the code above.

Question: I would like to colorize the endings of verb and noun forms without losing the correct bindings where the color switches. Do you see any possibility I could realize that without too complicated effort? Torsten Nieland

Answer: Inspired by this request, we have extended \texttt{acolor.sty} to allow all the commands mentioned above to be used even as part of the input notation. This does save much of the effort. The other key thing is that \texttt{\nospace} would suppress any intervening space between the adjacent glyphs, which is useful with various \TeX{}‘s commands in general.

Let us assume we want to colorize in red the stem (letters and vowels) of the word \texttt{\textit{ista}at-i}, and leave its prefix and suffix in the default style (the original problem is an inverse to this, but rather analogous).

We will typeset the Arabic script in parts, each featuring its settings of \acolors{}. We can enforce the connecting variants of the glyphs via hyphens -. The grouping into the parts has to be done by complete “syllables”. There might be cases when the letter belongs to one group, but its vocalization into another, and shall have the different style. These border cases will have to be singled out into groups of their own.

We would like to divide the word graphically like \texttt{\textit{ista}\texttt{\texttt{.t}}\texttt{A}'\texttt{at}-i}, the stem being \texttt{.tA}'1. Given the above, the parts will be \texttt{\{ista\}}\texttt{\{-\texttt{.tA}\}}\texttt{\{\texttt{a}\}}\texttt{\{-\texttt{t-i}\}}. Note that there is no junction between \texttt{\{-\texttt{.tA}\}}\texttt{\{-\texttt{a}\}}.

We have to style the \{\texttt{a}\} group only in the letter, not in the diacritics, while \{-\texttt{.tA}\} will have both the letters and the vocalization in the style of the stem. The code will then be:

\begin{verbatim}
<ista- \nospace\acolor{everything}{red} \-\texttt{.tA}>
\nospace\acolor{letters}{red} \texttt{\texttt{a}}-\nospace \texttt{-t-i>
\end{verbatim}
Of course, one could write a macro of five fields (prefix, prefix-border, stem, suffix-border, suffix), with an optional parameter for the stem’s color, to avoid the inconvenient noise in the document:

\newcommand\colorstem[6][red]{% 
\textcolor{red}{\hspace{2} \textcolor{#1}{#1} \hspace{3}} 
\textcolor{#1}{\hspace{3}} 
\textcolor{#5}{\hspace{5}} \textcolor{#6}{\hspace{6}}
\colorstem{ista-}{}{-.tA}{'a-}{-t-i} 
\colorstem{ista-}{}{-.tA}{'a-}{-t-i} 
\colorstem{ista-}{}{-.tI-}{-'u}{-y} 
\colorstem{ista-}{}{-.tI-}{-'i}{-y} 

أوترك أحرَر