acolor.sty — Colors for ArabTEX

Karel Mokrý Otakar Smrž

July 31, 2014

Once upon a time, we needed to colorize the diacritics typeset by ArabTEX.[1] We wrote the acolor.sty package to achieve that, and as time went by, we have refined it further.[2] The acolor.sty package depends on xcolor.sty, and you should have ArabTEX installed. Then put acolor.sty where TEX can find it, and in your document, type `\usepackage{arabtex} \usepackage{acolor}`.

The acolor.sty package provides the `\acolor{`*group*`}{`*color*`}` command. You can use it 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`:

```
\def\acoloron{\def\a@color##1##2{{\color{acolor4##1}##2}}}
\def\noacolor{\def\a@color##1##2{##2}}
\def\acolorlet#1#2{\acolor{#1}{acolor4#2}}
\def\acolor#1#2{%
    \ifthenelse{\equal{#1}{everything}}{\acolor{letters}   {#2}%
                                        \acolor{omissibles}{#2}%
                                        \acolor{tatwil}    {#2}%
                                        \acolor{others}    {#2}}{%
    \ifthenelse{\equal{#1}{letters}}   {\acolor{isolated}  {#2}%
                                        \acolor{initial}   {#2}%
                                        \acolor{medial}    {#2}%
                                        \acolor{final}     {#2}}{%
    \ifthenelse{\equal{#1}{isolated}}  {\acolor{0}         {#2}}{%
    \ifthenelse{\equal{#1}{initial}}   {\acolor{3}         {#2}}{%
    \ifthenelse{\equal{#1}{medial}}    {\acolor{2}         {#2}}{%
    \ifthenelse{\equal{#1}{final}}     {\acolor{1}         {#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}}}}}}}}}}}}\ignorespaces}
```

[1] www.ctan.org/pkg/arabtex/ and ftp.informatik.uni-stuttgart.de/pub/arabtex/

[2] github.com/otakar-smrz/encode-arabic/ and www.ucw.cz/~karel/

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.

\acolor{everything}{purple}	كِتَابُ آلْأَيَّامِ لِطٰهٰ حُسَيْنٍ عَنْهُ وَ عَائِلَتِهِ آنَذَاكَ
\acolor{diacritics}{red}	كِتَابُ آلْأَيَّامِ لِطٰهٰ حُسَيْنٍ عَنْهُ وَ عَائِلَتِهِ آنَذَاكَ
\acolor{omissibles}{white}	كتاب الايام لطه حسين عنه و عائلته انذاك
\acolor{hamza}{red!30!yellow}	كِتَابُ آلْأَيَّامِ لِطٰهٰ حُسَيْنٍ عَنْهُ وَ عَائِلَتِهِ آنَذَاكَ
\acolor{vowelmarks}{blue}	كِتَابُ آلْأَيَّامِ لِطٰهٰ حُسَيْنٍ عَنْهُ وَ عَائِلَتِهِ آنَذَاكَ
\acolor{emptymarks}{red!90!blue}	كِتَابُ آلْأَيَّامِ لِطٰهٰ حُسَيْنٍ عَنْهُ وَ عَائِلَتِهِ آنَذَاكَ
\acolor{isolated}{gray}	كِتَابُ آلْأَيَّامِ لِطٰهٰ حُسَيْنٍ عَنْهُ وَ عَائِلَتِهِ آنَذَاكَ

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

Question: When the acolor.sty package is included and the \spreadtrue option is in effect, undesirable artifacts that appear to be misplaced *taṭwīls* are output. When calling the command \acolor{everything}{*color*}, punctuation and numbers are not colored.

George Kamel

Answer: Thanks to this report, we have fixed the issue with *taṭwīls* and introduced new colors tatwil and others into everything.

```
\spreadtrue
\acolor{tatwil}{yellow}
\acolor{others}{blue}
```

الأمم المتحدة: إسقاط رحلة MH17 في أوكرانيا
«جريمة حرب»

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 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 \nospace would suppress any intervening space between the adjacent glyphs, which is useful with various ArabTEX's commands in general.

Let us assume we want to colorize in red the stem (letters and vowels) of the word اِستَطَاعَتِ *istaṭāʿat-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 `ista|.tA‘|at-i`, the stem being `|.tA‘|`. Given the above, the parts will be `{ista-}{-.tA}{‘a-}{-t-i}`. Note that there is no junction between `{-.tA-}{-‘a-}`.

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

```
\<ista- \nospace{\acolor{everything}{red} -.tA}
        \nospace{\acolor{letters}{red}     ‘a-}\nospace -t-i>
```

اِستَطَاعَتِ

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]{%
\<#2 \nospace{\acolor{diacritics}{#1} #3}
     \nospace{\acolor{everything}{#1} #4}
     \nospace{\acolor{letters}   {#1} #5}\nospace #6>}

\colorstem{ista-}{}{-.tA}{‘a-}{-t-i}
%
\acolor{fatha}{blue}\acolor{kasra}{red}
\colorstem[orange]{ista-}{}{-.tA}{‘a-}{-t-i}
%
\acolorlet{everything}{fatha}\acolor{kasra}{magenta}
\colorstem[black]{tasta-}{}{-.tI-}{-‘u}{}
%
\acolorlet{everything}{kasra}\acolor{kasra}{black}
\colorstem[orange]{ista-}{}{-.tI-}{-‘i-}{-y}
```

اِستَطِيعِي تَستَطِيعُ اِستَطَاعَتِ اِستَطَاعَتِ

أوتاكار اِسمرژ