Nice Symbols v0.1.0.0

Andrew Butterfield
Trinity College Dublin

December 20, 2016

```
{-# LANGUAGE CPP #-}
module NiceSymbols where
import Data.Char

versionNS = "0.1.0.0"
```

1 Introduction

We define some nice symbols using unicode, for non-Windows usage, along with dull ASCII equivalents for Windows users.

We assume UTF-8 throughout.

Given a LaTeX symbol macro like `\xyz` we define a Haskell variable `_xyz` to be a string that gives either a Unicode/UTF-8 glyph for that symbol, or an approximation using "ASCII-art".

2 Platform Independent Code

2.1 Follow each character by ...

```
follow "" _ = ""
follow (c:cs) a = c:a:follow cs a
```

2.2 Alphabet conversions

How to convert ASCII ‘a’ to ‘z’ into different fontstyles, in UTF-8 (See `http://qaz.wtf/u/convert.cgi?text=az`).

Style	Code for ’A’	Code for ‘a’
ASCII	65	97
Math Sans Bold	120276	120302

```
styleShift code_A code_a c
 | isUpper c  =  chr (ord c + upperShift)
 | isLower c  =  chr (ord c + lowerShift)
 | otherwise  =  c
 where
   upperShift = code_A - ord 'A'
   lowerShift = code_a - ord 'a'

mathBold     = map $ styleShift 119808 119834
mathSansBold = map $ styleShift 120276 120302
flags        = map $ styleShift 127462 127462
test = map $ styleShift 119886 119886
```

3 Weight Conversions

3.1 Weight Conversion for Unix/OS X

```
#ifndef mingw32_HOST_OS

eSGR n = "\ESC["++show n++"m"

resetSGR = eSGR 0
boldSGR  = eSGR 1
ovlSGR   = eSGR 9

bold str = boldSGR ++ str ++ resetSGR
overline c = ovlSGR ++ c:resetSGR
#endif
```

3.2 Weight "Conversion" for Windows

```
#ifdef mingw32_HOST_OS

bold str = '*':str++"*"
overline str = '^':str++"^"
#endif
```

4 Nice Symbols for OS X/Unix

```
#ifndef mingw32_HOST_OS

_ll = "\171"
_gg = "\187"

_alpha = "\945"
_pi = "\x03C0"
_epsilon = "\x03F5"
_tau = "\x03C4"
_Sigma = "\x2211"

_top = "\x22A4"
_bot = "\x22A5"
_sqcap = "\8851"
_sqcup = "\8852"
_sqsubseteq = "\8849"

_true = bold "true"
_false = bold "false"
_lnot = "\172"
_land = "\8743"
_lor = "\8744"
_implies = "\8658"
_equiv = "\8801"

_emptyset = "\8709"
_cup = "\8746"
_cap = "\8745"
_setminus = "\8726"
_in = "\8712"
_subseteq = "\8838"

_parallel = "\8214"
_Cap = "\8914"
```

```
_overline str = "\ESC[9m"++follow str '\x35e'++"\ESC[0m"

_supChar '2' = '\178'
_supChar '3' = '\179'
_supChar c
  | isDigit c = chr (ord c - ord '0' + 8304)
  | isSpace c = c
  | otherwise = '\175'

_supStr s = map _supChar s
_supNum n = _supStr $ show n
#endif
```

5 “Nice” Symbols for Windows

```
#ifdef mingw32_HOST_OS

_ll = "<<"
_gg = ">>"

_alpha = "alf"
_pi = "pi"
_epsilon = "eps"
_tau = "tau"
_Sigma = "Sigma"

_top = "T"
_bot = "_|_"
_sqcap = "|~|"
_sqcup = "|_|"
_sqsubseteq = "|="

_true = "true"  -- bold true
_false = "false" -- bold false
_lnot = "~"
_land = "/\\"
_lor = "\\/"
_implies = "==>"
_equiv = "=="

_emptyset = "{}"
_cup = "U"
_cap = "I"
_setminus = "\\"
_in = "in"
_subseteq = "subset"

_parallel = "||"
_Cap = "II"

_overline str = "ovl("++str++")"

_supStr = ('^':)
_supNum n = _supStr $ show n

#endif
```

6 Mainline

Basically a catalog of our nice symbols that is easy to display in GHCi

```
nice
 = [ ("_ll", _ll)
   , ("_gg", _gg)
   , ("_pi", _pi)
   , ("_epsilon", _epsilon)
   , ("_tau", _tau)
   , ("_Sigma", _Sigma)
   , ("_top", _top)
   , ("_bot", _bot)
   , ("_sqcap", _sqcap)
   , ("_sqcup", _sqcup)
   , ("_true", _true)
   , ("_false", _false)
   , ("_lnot", _lnot)
   , ("_land", _land)
   , ("_lor", _lor)
   , ("_implies", _implies)
   , ("_equiv", _equiv)
   , ("_emptyset", _emptyset)
   , ("_cup", _cup)
   , ("_cap", _cap)
   , ("_setminus", _setminus)
   , ("_in", _in)
   , ("_subseteq", _subseteq)
   , ("_parallel", _parallel)
   , ("_Cap", _Cap)
   , ("_overline(p)", _overline "p")
   , ("_supNum(42)", _supNum 42)
   ]

niceRender w (_nm, nm)
 = _nm ++ (replicate (w-length _nm) ' ') ++ "  " ++ nm
```

Use `main` in GHCi to see the available strings and functions.

```
main
 = do putStrLn ("Nice Symbols v"++versionNS++" listing:")
      putStrLn $ unlines $ map (niceRender maxw) nice
 where maxw = maximum $ map (length . fst) nice
```