Re: Quieter glyphs than parentheses

Source: http://coding.derkeiler.com/Archive/Lisp/comp.lang.lisp/2004−02/0906.html

---

From: Steven M. Haflich (*smh_no_spiced_ham_at_alum.mit.edu*)
Date: 02/08/04

Date: Sun, 08 Feb 2004 00:39:44 GMT

Ray Dillinger wrote:

> *We're getting there. Slowly. Moving a Lisp to Unicode is quite an*
> *undertaking if you intend to do it right, because source code and data*
> *are the same 'language' so you've got an entire toolchain that mostly*
> *has to be built from scratch.*

I must disagree. When Franz converted Allegro CL from an 8−bit character system to one that could be configured (albeit globally) either with 8−bit ASCII or 16−bit Unicode characters, it did not require rewriting the entire tool change. There were issues here and there, but most things just work. There is some nonobvious hair internally supporting Unicode (e.g. handling the correspondence between upper− and lower−case chars while preserving speed of the char/string comparison and predicate functions, but once the compiler understands both formats of chars and strings, most code works automatically.

Some additional work comes in supporting all the external formats needed by non ISO8859 language scripts. While Unicode represents pretty much everything in a nice, flat, 16−bit code (ignoring that Unicode has actually recently overflowed 16 bits −− sigh!) and UTF−8, which is a simple space−saving encoding of Unicode, most of the difficult languages have one or more different variable−length encodings. For example, Japanese has three popular non−Unicode−based encodings, and Lisp applications may ultimately need to deal with each. (For example, my Japanese wife regularly receives email in _four_ different encodings.)

In addition to having Lisp understand all these obscure encodings, there is continual work (and customer support) helping users set up their tools (e.g. Emacs, X, whatever) so that their displays have the proper fonts and their input methods work. But this is a system configuration issue more that a Lisp imlpementation issue. For example, CLIM tries to support the Lisp Machine Hyper and Super modifiess in addition to the standard Control and Meta key modifiers. This worked silently for many years (although was only rarely used by anyone) with the Num Lock key being remapped as the shift for Hyper. But this remapping interfered with a certain standard input method for Russian characters which also assigned some odd shift behavior to the Num Lock key −− so a patch was

needed for CLIM to keep it from acting on the Hyper key modifier.

To return to the original question, the brittleness of input methods and display fonts is one reason I would caution against using characters from some obscure font in order to improve visual typography. Few non−ISO−8859 fonts are standard across many platforms, and someone trying to read this code or papers without all the obscure fonts installed (or with his system improperly configured) will see garbage instead of the desired slender parentheses.