

Re: Greenspunning ML (revisited)

Source: <http://coding.derkeiler.com/Archive/Lisp/comp.lang.lisp/2008-04/msg01568.html>

- *From:* Jon Harrop <jon@xxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 28 Apr 2008 19:23:00 +0100
-

danb wrote:

So this library is called `cl-match`, and it uses the same `s-expression` interface to `ML-style` matching as does `fare-matcher`, but it has a couple extra features. It allows multiple occurrences of variable names, with implied `EQL` comparisons, and allows branch-specific guards to be embedded in individual branches where alternative patterns are allowed (`OR` patterns). The expanded code is optimized somewhat by testing all single-branch patterns first.

Can you implement proper decision-tree-based optimizations by augmenting Lisp with some kind of closed algebraic data types equivalent to ML's variant types?

That is where the enormous performance advantages lie...

There's some documentation here:

<http://www.prairienet.org/~dsb/clmatch.htm>

The main page has links to a user's manual, a page on the internals, and a download directory. Please feel free to comment, criticize, point out bugs, etc..

ML's pattern matching was designed to make it easy to convey statically-checked constraints. I assume your Lisp implementation makes no attempt to provide any static assurances at all. In which case, you may find it more productive to Greenspun Mathematica's pattern matching instead.

Also, you might like to reimplement this old benchmark using your new library:

<http://www.lambdassociates.org/studies/study10.htm>

HTH.

Re: Greenspunning ML (revisited)

—
Dr Jon D Harrop, Flying Frog Consultancy
<http://www.ffconsultancy.com/products/?u>