

# Re: Program compression

---

*Source:* <http://coding.derkeiler.com/Archive/General/comp.programming/2008-07/msg00233.html>

---

- *From:* [jaycx2.3.calrobert@xxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:jaycx2.3.calrobert@xxxxxxxxxxxxxxxxxxxxxxxxxx) (Robert Maas, <http://tinyurl.com/uh3t>)
  - *Date:* Sat, 26 Jul 2008 01:43:34 -0700
- 

Haskell, SML, OCaml, Mathematica, F# and Scala all allow real problems to be solved much more concisely than with Lisp. Indeed, I think it is difficult to imagine even a single example where Lisp is competitively concise.

What does "solved much more concisely" mean??

From: Jon Harrop <j...@xxxxxxxxxxxxxxxxxxxx>  
The solutions are shorter in modern FPLs.

OK, let's say you have a text string which contains the notation for a nested list starting from the beginning of the string. How many lines of code, in various modern FPLs, would it take to parse that nested-list notation to produce an actual nested list structure, and also report back in another value the position in the string where the parse left off at the end of the nested-list notation? In Common Lisp it's just one line of code: `(multiple-value-setq (ls ei) (read-from-string ts))` where `ts` has the given text string, `ls` gets the resultant list structure, and `ei` gets the resultant end index.

Now let's say you want to pick up where you left off, because after that first nested list notation there's another within the same text string you were given. How many lines of code in your favorite FPLs? In Common Lisp, it's another one line of code: `(multiple-value-setq (ls2 ei2) (read-from-string ts :start ei))`

My point is that different languages are based on different things being important enough to put into the core language, but with add-on libraries it's possible to make a lot of things be essentially the same length in a whole bunch of languages. If you don't mind Unix-style ultra-short names for commonly use utilities, and you do a lot of this kind of stuff, if you really want to make your code as short as possibly, you can use macros to abbreviate

## Re: Program compression

the special operator and function used above, so the two lines of code will easily fit into just one run-on line of code:

```
(mvs (ls ei) (rfs ts)) (mvs (ls2 ei2) (rfs ts :start ei))
```