

Re: New symbolic/numeric/dynamic/intuitive programming language

Source: <http://sci.tech-archive.net/Archive/sci.math/2008-02/msg04177.html>

- *From:* Michael Press <rubrum@xxxxxxxxxxxx>
 - *Date:* Mon, 25 Feb 2008 21:38:52 -0800
-

In article <rem-2008feb24-002@xxxxxxxxxx>, rem642b@xxxxxxxxxx (Robert Maas, see <http://tinyurl.com/uh3t>) wrote:

COBOL actually had at least one good idea: PICture clauses.

...

(Asking about BNF used likewise in any modern programming language.)

From: Michael Press <rub...@xxxxxxxxxxxx>
Are you saying that lex and yacc fail this test?

Neither of those is a general-purpose programming language.

Furthermore lex uses regular expressions, not BNF.

I can't tell from 'man yacc' what format it uses for the syntax spec. Google turns up <<http://en.wikipedia.org/wiki/Yacc>> which says it uses something similar to BNF, but the resultant (generated) parser requires a lexical analyzer, such as lex. So I guess you meant the combination together? Still that's only a way to generate C sourcecode for a parser for syntax, which would then have to be somehow integrated into a program, which would presumably have to be written in C. That C program would then have to do all the work of building the abstract parse tree in RAM to represent the syntax that had been parsed. Even so, I see no obvious way to generate formatted output, i.e. reverse the parsing task, using the same BNF definition of the syntax.

Yes, lex, yacc, and C work together to compile a program that can parse for syntax. The yacc source files specify the grammar. No, the reverse is not part of the process.

—

Michael Press

Re: New symbolic/numeric/dynamic/intuitive programming language