

Yacc

Yacc (**Yet Another Compiler-Compiler**) is a computer program for the Unix operating system developed by Stephen C. Johnson. It is a Look Ahead Left-to-Right (LALR) parser generator, generating a LALR parser (the part of a compiler that tries to make syntactic sense of the source code) based on an formal grammar, written in a notation similar to Backus–Naur Form (BNF).^[1] Yacc is supplied as a standard utility on BSD and AT&T Unix.^[2] GNU-based Linux distributions include Bison, a forward-compatible Yacc replacement.^[3]

Yacc

Original author(s)	<u>Stephen C. Johnson</u>
Repository	<u>www.tuhs.org/cgi-bin/utree.pl?file=V6%2Fusr%2Fsource%2Fyacc</u> (<u>https://www.tuhs.org/cgi-bin/utree.pl?file=V6/usr/source/yacc</u>)
Operating system	<u>Unix</u> , <u>Unix-like</u> , <u>Plan 9</u> , <u>Inferno</u>
Platform	<u>Cross-platform</u>
Type	<u>Command</u>

Contents

History

Description

Impact

See also

References

External links

History

In the early 1970s, Stephen C. Johnson, a computer scientist at Bell Labs / AT&T, developed Yacc because he wanted to insert an exclusive or operator into a B language compiler (developed using McIlroy's TMG compiler-compiler^[4]), but it turned out to be a hard task. As a result, he was directed by Bell Labs colleague Al Aho to Donald Knuth's work on LR parsing, which served as the basis for Yacc.^[5] Yacc was influenced by^[6] and received its name in reference to TMG compiler-compiler.^[7]

Yacc was originally written in the B programming language, but was soon rewritten in C.^[4] It appeared as part of Version 3 Unix,^[8] and a full description of Yacc was published in 1975.^[6]

Johnson used Yacc to create the Portable C Compiler.^[8] Bjarne Stroustrup, on the other hand, attempted to use Yacc for his initial work on C++, but "was defeated by C's syntax" (for example, having both prefix and postfix declaration operators).^[9]

In a 2008 interview, Johnson reflected that "the contribution Yacc made to the spread of Unix and C is what I'm proudest of".^[10]

Description

The input to Yacc is a grammar with snippets of C code (called "actions") attached to its rules. Its output is a shift-reduce parser in C that executes the C snippets associated with each rule as soon as the rule is recognized. Typical actions involve the construction of parse trees. Using an example from Johnson, if the call `node(label, left, right)` constructs a binary parse tree node with the specified `label` and children, then the rule

```
expr : expr '+' expr { $$ = node('+', $1, $3); }
```

recognizes summation expressions and constructs nodes for them. The special identifiers `$$`, `$1` and `$3` refer to items on the parser's stack.^[6]

Yacc produces only a parser (phrase analyzer); for full syntactic analysis this requires an external lexical analyzer to perform the first tokenization stage (word analysis), which is then followed by the parsing stage proper.^[6] Lexical analyzer generators, such as Lex or Flex, are widely available. The IEEE POSIX P1003.2 standard defines the functionality and requirements for both Lex and Yacc.^[11]

Some versions of AT&T Yacc have become open source. For example, source code is available with the standard distributions of Plan 9.^[12]

Impact

Yacc and similar programs (largely reimplementations) have been very popular. Yacc itself used to be available as the default parser generator on most Unix systems, though it has since been supplanted by more recent, largely compatible, programs such as Berkeley Yacc, GNU Bison, MKS Yacc, and Abraxas PCYACC. An updated version of the original AT&T version is included as part of Sun's OpenSolaris project. Each offers slight improvements and additional features over the original Yacc, but the concept and basic syntax have remained the same.^[13]

Among the languages that were first implemented with Yacc are AWK, eqn and Pic.^[14] Yacc was also used on Unix to implement the Portable C Compiler, as well as parsers for such programming languages as FORTRAN 77, Ratfor, APL, bc, m4, etc.^{[8][15]}

Yacc has also been rewritten for other languages, including OCaml,^[16] Ratfor, ML, Ada, Pascal, Java, Python, Ruby, Go,^[17] Common Lisp^[18] and Erlang.^[19]

- Berkeley Yacc: The Berkeley implementation of Yacc quickly became more popular than AT&T Yacc itself because of its performance and lack of reuse restrictions.^[20]
- LALR parser: The underlying parsing algorithm in Yacc-generated parsers.
- Bison: The GNU version of Yacc.
- Lex (and Flex lexical analyser), a token parser commonly used in conjunction with Yacc (and Bison).
- BNF is a metasyntax used to express context-free grammars: that is a formal way to describe context-free languages.
- PLY (Python Lex-Yacc) is an alternative implementation of Lex and Yacc in Python.

See also

- Compiler-compiler
- hoc (programming language)

References

1. "The A-Z of Programming Languages: YACC" (<http://www.techworld.com.au/article/252319/a-z-programming-languages-yacc/>). Computerworld. Retrieved 30 November 2012.
2. Levine, John (1992). *Lex & yacc* (<https://archive.org/details/lexyacc00levi>). Sebastopol, CA: O'Reilly & Associates. p. xx. ISBN 1-56592-000-7.
3. Levine, John (2009). *Flex & bison*. Sebastopol, Calif: O'Reilly Media. p. xv. ISBN 978-0-596-15597-1.
4. Ritchie, Dennis M. (April 1993). *The Development of the C Language* (<http://heim.ifi.uio.no/inf2270/programmer/historien-om-C.pdf>) (PDF). Association for Computing Machinery, Inc.
5. Morris, Richard (1 October 2009). "Stephen Curtis Johnson: Geek of the Week" (<https://www.red-gate.com/simple-talk/opinion/geek-of-the-week/stephen-curtis-johnson-geek-of-the-week/>). Red Gate Software. Retrieved 19 January 2018.
6. Johnson, Stephen C. (1975). *Yacc: Yet Another Compiler-Compiler* (<http://dinosaur.compilertools.net/yacc/>) (Technical report). Murray Hill, New Jersey: AT&T Bell Laboratories. 32. Retrieved 31 January 2020.
7. "Early Translator Writing Systems" (<http://www.chilton-computing.org.uk/acl/applications/cc/p007.htm>). Atlas Computer Laboratory.
8. McIlroy, M. D. (1987). *A Research Unix reader: annotated excerpts from the Programmer's Manual, 1971–1986* (<http://www.cs.dartmouth.edu/~doug/reader.pdf>) (PDF) (Technical report). CSTR. Bell Labs. 139.
9. Stroustrup, Bjarne. "A History of C++: 1979–1991" (<http://www.stroustrup.com/hopl2.pdf>) (PDF).
10. Hamilton, Naomi (10 July 2008). "Yacc, Unix, and Advice from Bell Labs Alumni Stephen Johnson" (<https://www.computerworld.com/article/2534739/app-development/yacc-unix--and-a-dvice-from-bell-labs-alumni-stephen-johnson.html>). Computerworld. Retrieved 19 January 2018.
11. *lex* (<https://www.opengroup.org/onlinepubs/9699919799/utilities/lex.html>) – Commands & Utilities Reference, The Single UNIX Specification, Issue 7 from The Open Group, *yacc* (<https://www.opengroup.org/onlinepubs/9699919799/utilities/yacc.html>) – Commands & Utilities Reference, The Single UNIX Specification, Issue 7 from The Open Group.
12. "plan9: UC Berkeley release of Plan 9 under the GPLv2" (<https://github.com/brho/plan9/blob/89d43d2262ad43eb4b26c2a8d6a27cfeddb33828/sys/src/cmd/yacc.c>). 26 December 2017. Retrieved 2 January 2018.
13. Bison Manual: History (https://www.gnu.org/software/bison/manual/html_node/History.html#History)
14. "UNIX Special: Profs Kernighan & Brailsford" (https://www.youtube.com/watch?v=vT_J6xc-Az0). Computerphile. September 30, 2015.
15. Kernighan, Brian W.; Pike, Rob (1984). *The Unix Programming Environment*. Prentice Hall. ISBN 0-13-937681-X.
16. "OCaml User's Manual: Chapter 12 Lexer and parser generators (ocamllex, ocaml yacc)" (<http://caml.inria.fr/pub/docs/manual-ocaml-4.00/manual026.html>). Retrieved 25 Nov 2013.
17. "Yacc.go: A version of Yacc for the Go Programming Language" (<https://godoc.org/golang.org/x/tools/cmd/goyacc>). Retrieved 15 July 2017.
18. "CL-Yacc: A Common Lisp version of Yacc" (<http://www.pps.univ-paris-diderot.fr/~jch/software/c-l-yacc/>).
19. "yecc: An Erlang implementation of Yacc" (<http://erlang.org/doc/man/yecc.html>).
20. John Levine (August 2009), *flex & bison*, O'Reilly Media

External links

- `yacc` (<https://www.opengroup.org/onlinepubs/9699919799/utilities/yacc.html>) – Commands & Utilities Reference, The Single UNIX Specification, Issue 7 from The Open Group
 - `yacc(1)` (http://man.cat-v.org/plan_9/1/yacc) – Plan 9 Programmer's Manual, Volume 1
 - `yacc(1)` (<http://man.cat-v.org/inferno/1/yacc>) – Inferno General commands Manual
-

Retrieved from "<https://en.wikipedia.org/w/index.php?title=Yacc&oldid=981942820>"

This page was last edited on 5 October 2020, at 08:56 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.