

**NAME**

yacc — yet another compiler-compiler

**SYNOPSIS**

yacc [ -vd ] grammar

**DESCRIPTION**

*Yacc* converts a context-free grammar into a set of tables for a simple automaton which executes an LR parsing algorithm. The grammar may be ambiguous; specified precedence rules are used to break ambiguities.

The output file, *y.tab.c*, must be compiled by the C compiler to produce a program *yyparse*. This program must be loaded with the lexical analyzer program, *yylex*, as well as *main* and *yyerror*, an error handling routine. These routines must be supplied by the user; *lex(1)* is useful for creating lexical analyzers usable by *yacc*.

If the *-v* flag is given, the file *y.output* is prepared, which contains a description of the parsing tables and a report on conflicts generated by ambiguities in the grammar.

If the *-d* flag is used, the file *y.tab.h* is generated with the *#define* statements that associate the *yacc*-assigned 'token codes' with the user-declared 'token names'. This allows source files other than *y.tab.c* to access the token codes.

**FILES**

<i>y.output</i>	
<i>y.tab.c</i>	
<i>y.tab.h</i>	defines for token names
<i>yacc.tmp</i> , <i>yacc.acts</i>	temporary files
<i>/usr/lib/yaccpar</i>	parser prototype for C programs

**SEE ALSO**

*lex(1)*

*LR Parsing* by A. V. Aho and S. C. Johnson, Computing Surveys, June, 1974.

*YACC — Yet Another Compiler Compiler* by S. C. Johnson.

**DIAGNOSTICS**

The number of reduce-reduce and shift-reduce conflicts is reported on the standard output; a more detailed report is found in the *y.output* file. Similarly, if some rules are not reachable from the start symbol, this is also reported.

**BUGS**

Because file names are fixed, at most one *yacc* process can be active in a given directory at a time.