

WIKIPEDIA

Comparison of parser generators

This is a list of notable [lexer generators](#) and [parser generators](#) for various language classes.

Contents

[Regular languages](#)

[Deterministic context-free languages](#)

[Parsing expression grammars, deterministic boolean grammars](#)

[General context-free, conjunctive, or boolean languages](#)

[Context-sensitive grammars](#)

[See also](#)

[Notes](#)

[References](#)

[External links](#)

Regular languages

[Regular languages](#) are a category of languages (sometimes termed [Chomsky Type 3](#)) which can be matched by a state machine (more specifically, by a [deterministic finite automaton](#) or a [nondeterministic finite automaton](#)) constructed from a [regular expression](#). In particular, a regular language can match constructs like "A follows B", "Either A or B", "A, followed by zero or more instances of B", but cannot match constructs which require consistency between non-adjacent elements, such as "some instances of A followed by the same number of instances of B", and also cannot express the concept of recursive "nesting" ("every A is eventually followed by a matching B"). A classic example of a problem which a regular grammar cannot handle is the question of whether a given string contains correctly-nested parentheses. (This is typically handled by a Chomsky Type 2 grammar, also termed a [context-free grammar](#).)

Name	Lexer algorithm	Output languages	Grammar, code	Development platform	License
Alex	DFA	Haskell	Mixed	All	Free, BSD
AnnoFlex	DFA	Java	Mixed	Java virtual machine	Free, BSD
Astir	DFA table driven, with branching	C++	Only grammar (actioned)	All	Free, MIT
AustenX	DFA	Java	Separate	All	Free, BSD
C# Flex	DFA	C#	Mixed	.NET CLR	Free, GNU GPL
C# Lex	DFA	C#	Mixed	.NET CLR	?
Carburetta	DFA	C, C++	Mixed	All	Free, Apache 2.0
CookCC	DFA	Java	Mixed	Java virtual machine	Free, Apache 2.0
DFA (inc. w. LRSTAR)	DFA compressed matrix	C, C++	Separate	Visual Studio	BSD
Dolphin	DFA	C++	Separate	All	Proprietary
Flex	DFA table driven	C, C++	Mixed	All	Free, BSD
gelex	DFA	Eiffel	Mixed	Eiffel	Free, MIT
golex	DFA	Go	Mixed	Go	Free, BSD-style
gplex	DFA	C#	Mixed	.NET CLR	Free, BSD-like
JFlex	DFA	Java	Mixed	Java virtual machine	Free, BSD
JLex	DFA	Java	Mixed	Java virtual machine	Free, BSD-like
lex	DFA	C	Mixed	POSIX	Partial, proprietary, CDDL
lexertl	DFA	C++	?	All	Free, GNU LGPL
Quex	DFA direct code	C, C++	Mixed	All	Free, GNU LGPL
Ragel	DFA	Go, C, C++, assembly	Mixed	All	Free, GNU GPL, MIT ^[1] [2]
RE/flex	DFA direct code, DFA table driven, and NFA regex libraries	C++	Mixed	All	Free, BSD
re2c	DFA direct code	C, Go	Mixed	All	Free, public domain

Deterministic context-free languages

Context-free languages are a category of languages (sometimes termed Chomsky Type 2) which can be matched by a sequence of replacement rules, each of which essentially maps each non-terminal element to a sequence of terminal elements and/or other nonterminal elements. Grammars of this type can match anything that can be matched by a regular grammar, and furthermore, can handle the concept of recursive "nesting" ("every A is eventually followed by a matching B"), such as the question of whether a given string contains correctly-nested parentheses. The rules of Context-free grammars are purely local, however, and therefore cannot handle questions that require non-local analysis such as "Does a declaration exist for every variable that is used in a function?". To do so technically would require a more sophisticated grammar, like a Chomsky Type 1 grammar, also termed a context-sensitive grammar. However, parser generators for context-free grammars often support the ability for user-written code to introduce limited amounts of context-sensitivity. (For example, upon encountering a variable declaration, user-written code could save the name and type of the variable into an external data structure, so that these could be checked against later variable references detected by the parser.)

The deterministic context-free languages are a proper subset of the context-free languages which can be efficiently parsed by deterministic pushdown automata.

Name	Parsing algorithm	Input grammar notation	Output languages	Grammar, code	Lexer	Development platform	IDE	License
AGL	<u>GLR</u>	<u>EBNF</u>	Java, JavaScript, (other Kotlin targets)	Separate	none	Any Kotlin target platform	No	Free, <u>Apache 2.0</u>
<u>ANTLR4</u>	Adaptive <u>LL(*)</u> ^[3]	<u>EBNF</u>	<u>C#</u> , Java, Python, JavaScript, <u>C++</u> , Swift, Go, <u>PHP</u>	Separate	generated	<u>Java virtual machine</u>	Yes	Free, <u>BSD</u>
<u>ANTLR3</u>	<u>LL(*)</u>	<u>EBNF</u>	ActionScript, Ada95, C, <u>C++</u> , <u>C#</u> , Java, JavaScript, Objective-C, Perl, Python, Ruby	Mixed	generated	<u>Java virtual machine</u>	Yes	Free, <u>BSD</u>
APG	<u>Recursive descent</u> , <u>backtracking</u>	<u>ABNF</u>	C, C++, JavaScript, Java	Separate	none	All	No	Free, <u>GNU GPL</u>
Astir	<u>LL(k)</u> , <u>LL(finite)</u> , <u>LL(*)</u>	Astir hierarchical grammar	<u>C++</u>	Only grammar (actioned)	generated	All	<u>Visual Studio Code</u>	Free, <u>MIT</u>
AXE	<u>Recursive descent</u>	AXE/C++	<u>C++17</u> , <u>C++11</u>	Mixed	none	Any with C++17 or C++11 standard compiler	No	Free, <u>Boost</u>
Beaver	<u>LALR(1)</u>	<u>EBNF</u>	Java	Mixed	external	<u>Java virtual machine</u>	No	Free, <u>BSD</u>
Belr	<u>Recursive descent</u>	<u>ABNF</u>	<u>C++17</u> , <u>C++11</u>	Separate	included	<u>POSIX</u>	No	Partial, <u>GNU GPL</u> , <u>proprietary</u>
<u>Bison</u>	<u>LALR(1)</u> , <u>LR(1)</u> , <u>IELR(1)</u> , <u>GLR</u>	<u>Yacc</u>	C, C++, Java	Mixed	external	All	No	Free, <u>GNU GPL</u> with exception
Bison++ ^[note 1]	<u>LALR(1)</u>	?	<u>C++</u>	Mixed	external	<u>POSIX</u>	No	Free, <u>GNU GPL</u>
Bisonc++	<u>LALR(1)</u>	?	<u>C++</u>	Mixed	external	<u>POSIX</u>	No	Free, <u>GNU GPL</u>
<u>BtYacc</u>	<u>Backtracking Bottom-up</u>	?	<u>C++</u>	Mixed	external	All	No	Free, <u>public domain</u>
<u>byacc</u>	<u>LALR(1)</u>	<u>Yacc</u>	C	Mixed	external	All	No	Free, <u>public domain</u>
BYACC/J	<u>LALR(1)</u>	<u>Yacc</u>	C, Java	Mixed	external	All	No	Free, <u>public domain</u>
Carburetta	<u>LALR(1)</u>	BNF Dialect	C, C++	Mixed	external or generated	All	No	Free, <u>Apache 2.0</u>
CL-Yacc	<u>LALR(1)</u>	<u>Lisp</u>	<u>Common Lisp</u>	Mixed	external	All	No	Free, <u>MIT</u>
<u>Coco/R</u>	<u>LL(1)</u>	<u>EBNF</u>	C, C++, C#, F#, Java, Ada, Object Pascal, Delphi, Modula-2, Oberon, Ruby,	Mixed	generated	<u>Java virtual machine</u> , <u>.NET Framework</u> , <u>Windows</u> , <u>POSIX</u> (depends on	No	Free, <u>GNU GPL</u>

			Swift, Unicon, Visual Basic .NET			output language)		
CookCC	<u>LALR(1)</u>	Java annotations	<u>Java</u>	Mixed	generated	<u>Java virtual machine</u>	No	Free, <u>Apache 2.0</u>
CppCC	<u>LL(k)</u>	?	<u>C++</u>	Mixed	generated	<u>POSIX</u>	No	Free, <u>GNU GPL</u>
CSP	<u>LR(1)</u>	?	<u>C++</u>	Separate	generated	<u>POSIX</u>	No	Free, <u>Apache 2.0</u>
CUP	<u>LALR(1)</u>	?	<u>Java</u>	Mixed	external	<u>Java virtual machine</u>	No	Free, <u>BSD-like</u>
Dragon	<u>LR(1), LALR(1)</u>	?	<u>C++, Java</u>	Separate	generated	All	No	Free, <u>GNU GPL</u>
eli	<u>LALR(1)</u>	?	<u>C</u>	Mixed	generated	<u>POSIX</u>	No	Free, <u>GNU GPL, GNU LGPL</u>
Essence	<u>LR(?)</u>	?	<u>Scheme 48</u>	Mixed	external	All	No	Free, <u>BSD</u>
eyapp	<u>LALR(1)</u>	?	<u>Perl</u>	Mixed	external or generated	All	No	Free, <u>Artistic</u>
Frown	<u>LALR(k)</u>	?	<u>Haskell 98</u>	Mixed	external	All	No	Free, <u>GNU GPL</u>
geyacc	<u>LALR(1)</u>	?	<u>Eiffel</u>	Mixed	external	All	No	Free, <u>MIT</u>
<u>GOLD</u>	<u>LALR(1)</u>	BNF	x86 assembly language, <u>ANSI C, C#, D, Java, Pascal, Object Pascal, Python, Visual Basic 6, Visual Basic .NET, Visual C++</u>	Separate	generated	<u>Windows</u>	Yes	Free, <u>zlib modified</u>
GPPG	<u>LALR(1)</u>	<u>Yacc</u>	<u>C#</u>	Separate	external	<u>Windows</u>	Yes	Free, <u>BSD</u>
<u>Grammatica</u>	<u>LL(k)</u>	BNF dialect	<u>C#, Java</u>	Separate	generated	<u>Java virtual machine</u>	No	Free, <u>BSD</u>
HiLexed	<u>LL(*)</u>	EBNF or Java	<u>Java</u>	Separate	internal	<u>Java virtual machine</u>	No	Free, <u>GNU LGPL</u>
Hime Parser Generator	<u>LALR(1), GLR</u>	BNF dialect	<u>C#, Java, Rust</u>	Separate	generated	<u>.NET Framework, Java virtual machine</u>	No	Free, <u>GNU LGPL</u>
Hyacc	<u>LR(1), LALR(1), LR(0)</u>	<u>Yacc</u>	<u>C</u>	Mixed	external	All	No	Free, <u>GNU GPL</u>
iyacc	<u>LALR(1)</u>	<u>Yacc</u>	<u>Icon</u>	Mixed	external	All	No	Free, <u>GNU LGPL</u>
jacc	<u>LALR(1)</u>	?	<u>Java</u>	Mixed	external	<u>Java virtual machine</u>	No	Free, <u>BSD</u>
<u>JavaCC</u>	<u>LL(k)</u>	<u>EBNF</u>	<u>Java, C++, JavaScript (via GWT compiler)^[4]</u>	Mixed	generated	<u>Java virtual machine</u>	Yes	Free, <u>BSD</u>
jay	<u>LALR(1)</u>	<u>Yacc</u>	<u>C#, Java</u>	Mixed	none	<u>Java virtual machine</u>	No	Free, <u>BSD</u>
<u>JFLAP</u>	<u>LL(1), LALR(1)</u>	?	<u>Java</u>	?	?	<u>Java virtual machine</u>	Yes	?

<u>JetPAG</u>	<u>LL(k)</u>	<u>?</u>	<u>C++</u>	Mixed	generated	All	No	Free, <u>GNU GPL</u>
<u>JS/CC</u>	<u>LALR(1)</u>	<u>EBNF</u>	<u>JavaScript</u> , <u>JScript</u> , <u>ECMAScript</u>	Mixed	internal	All	Yes	Free, <u>BSD</u>
<u>KDevelop-PG-Qt</u>	<u>LL(1)</u> , <u>backtracking</u> , <u>shunting-yard</u>	<u>?</u>	<u>C++</u>	Mixed	generated or external	All, <u>KDE</u>	No	Free, <u>GNU LGPL</u>
<u>Kelbt</u>	<u>Backtracking LALR(1)</u>	<u>?</u>	<u>C++</u>	Mixed	generated	<u>POSIX</u>	No	Free, <u>GNU GPL</u>
<u>kmyacc</u>	<u>LALR(1)</u>	<u>?</u>	<u>C</u> , <u>Java</u> , <u>Perl</u> , <u>JavaScript</u>	Mixed	external	All	No	Free, <u>GNU GPL</u>
<u>Lapg</u>	<u>LALR(1)</u>	<u>?</u>	<u>C</u> , <u>C++</u> , <u>C#</u> , <u>Java</u> , <u>JavaScript</u>	Mixed	generated	<u>Java virtual machine</u>	No	Free, <u>GNU GPL</u>
<u>Lemon</u>	<u>LALR(1)</u>	<u>?</u>	<u>C</u>	Mixed	external	All	No	Free, public domain
<u>Lime</u>	<u>LALR(1)</u>	<u>?</u>	<u>PHP</u>	Mixed	external	All	No	Free, <u>GNU GPL</u>
<u>LISA</u>	<u>LR(?)</u> , <u>LL(?)</u> , <u>LALR(?)</u> , <u>SLR(?)</u>	<u>?</u>	<u>Java</u>	Mixed	generated	<u>Java virtual machine</u>	Yes	Free, public domain
<u>LLgen</u>	<u>LL(1)</u>	<u>?</u>	<u>C</u>	Mixed	external	<u>POSIX</u>	No	Free, <u>BSD</u>
<u>LLnextgen</u>	<u>LL(1)</u>	<u>?</u>	<u>C</u>	Mixed	external	All	No	Free, <u>GNU GPL</u>
<u>LLLPG</u>	<u>LL(k) + syntactic and semantic predicates</u>	<u>ANTLR-like</u>	<u>C#</u>	Mixed	generated (?)	<u>.NET Framework</u> , <u>Mono</u>	<u>Visual Studio</u>	Free, <u>GNU LGPL</u>
<u>LPG</u>	<u>Backtracking LALR(k)</u>	<u>?</u>	<u>Java</u>	Mixed	generated	<u>Java virtual machine</u>	No	Free, <u>EPL</u>
<u>LRSTAR</u>	<u>LALR(1)</u> , <u>LALR(k)</u>	<u>YACC</u> , <u>ANTLR</u> , <u>EBNF</u>	<u>C++</u>	separated	generated	Windows	<u>Visual Studio</u>	<u>BSD</u>
<u>Menhir</u>	<u>LR(1)</u>	<u>?</u>	<u>OCaml</u>	Mixed	generated	All	No	Free, <u>QPL</u>
<u>ML-Yacc</u>	<u>LALR(1)</u>	<u>?</u>	<u>ML</u>	Mixed	external	All	No	<u>?</u>
<u>Monkey</u>	<u>LR(1)</u>	<u>?</u>	<u>Java</u>	Separate	generated	<u>Java virtual machine</u>	No	Free, <u>GNU GPL</u>
<u>Msta</u>	<u>LALR(k)</u> , <u>LR(k)</u>	<u>YACC</u> , <u>EBNF</u>	<u>C</u> , <u>C++</u>	Mixed	external or generated	<u>POSIX</u> , <u>Cygwin</u>	No	Free, <u>GNU GPL</u>
<u>MTP (More Than Parsing)</u>	<u>LL(1)</u>	<u>?</u>	<u>Java</u>	Separate	generated	<u>Java virtual machine</u>	No	Free, <u>GNU GPL</u>
<u>MyParser</u>	<u>LL(*)</u>	<u>Markdown</u>	<u>C++11</u>	Separate	internal	Any with standard C++11 compiler	No	Free, <u>MIT</u>
<u>NLT</u>	<u>GLR</u>	<u>C#/BNF-like</u>	<u>C#</u>	Mixed	mixed	<u>.NET Framework</u>	No	Free, <u>MIT</u>
<u>ocamlyacc</u>	<u>LALR(1)</u>	<u>?</u>	<u>OCaml</u>	Mixed	external	All	No	Free, <u>QPL</u>
<u>olex</u>	<u>LL(1)</u>	<u>?</u>	<u>C++</u>	Mixed	generated	All	No	Free, <u>GNU GPL</u>
<u>Parsec</u>	<u>LL</u> , <u>backtracking</u>	<u>Haskell</u>	<u>Haskell</u>	Mixed	none	All	No	Free, <u>BSD</u>
								Free, <u>GNU</u>

Parse::Yapp	LALR(1)	?	Perl	Mixed	external	All	No	GPL
Parser Objects	LL(k)	?	Java	Mixed	?	Java virtual machine	No	Free, zlib
PCCTS	LL	?	C, C++	?	?	All	No	?
PLY	LALR(1)	BNF	Python	Mixed	generated	All	No	Free, MIT
PlyPlus	LALR(1)	EBNF	Python	Separate	generated	All	No	Free, MIT
PRECC	LL(k)	?	C	Separate	generated	DOS, POSIX	No	Free, GNU GPL
racc ^[5]	LALR(1)	BNF-like , yacc-like ^[6]	Ruby	Mixed	?	Windows, Linux, macOS, FreeBSD, NetBSD	No	LGPL
QLALR	LALR(1)	?	C++	Mixed	external	All	No	Free, GNU GPL
SableCC	LALR(1)	?	C, C++, C#, Java, OCaml, Python	Separate	generated	Java virtual machine	No	Free, GNU LGPL
SLK ^[7]	LL(k) LR(k) LALR(k)	EBNF	C, C++, C#, Java, JavaScript	Separate	external	All	No	SLK ^[8]
SLY ^[9]	LALR(1)	BNF	Python	Mixed	generated	All	No	Free, MIT
SP (Simple Parser)	Recursive descent	Python	Python	Separate	generated	All	No	Free, GNU LGPL
Spirit	Recursive descent	?	C++	Mixed	internal	All	No	Free, Boost
Styx	LALR(1)	?	C, C++	Separate	generated	All	No	Free, GNU LGPL
Sweet Parser	LALR(1)	?	C++	Separate	generated	Windows	No	Free, zlib
Tap	LL(1)	?	C++	Mixed	generated	All	No	Free, GNU GPL
TextTransformer	LL(k)	?	C++	Mixed	generated	Windows	Yes	Proprietary
TinyPG	LL(1)	?	C#, Visual Basic	?	?	Windows	Yes	Partial, CPOL 1.0
Toy Parser Generator	Recursive descent	?	Python	Mixed	generated	All	No	Free, GNU LGPL
TP Yacc	LALR(1)	?	Turbo Pascal	Mixed	external	All	Yes	Free, GNU GPL
Tree-Sitter ^[10]	LR(1) , GLR	JavaScript DSL , JSON	C , bindings (Rust , WebAssembly , JavaScript , Python , many other)	Separate	generated + external	All	Neovim , Visual Studio Code	Free, MIT
Tunnel Grammar Studio	Recursive descent , backtracking	ABNF	C++	Separate	generated	Windows	Yes	Proprietary
UltraGram	LALR(1) , LR(1) , GLR	BNF	C++, Java, C#, Visual Basic .NET	Separate	external	Windows	Yes	Free, public domain
UniCC	LALR(1)	EBNF	C, C++, Python, JavaScript, JSON, XML	Mixed	generated	POSIX	No	Free, BSD
UrchinCC	LL(1)	?	Java	?	generated	Java virtual machine	No	?

Whale	<u>LR(?)</u> , some conjunctive stuff, see <u>Whale Calf</u>	?	<u>C++</u>	Mixed	external	All	No	<u>Proprietary</u>
wisent	<u>LALR(1)</u>	?	<u>C++, Java</u>	Mixed	external	All	No	<u>Free, GNU GPL</u>
<u>Yacc AT&T/Sun</u>	<u>LALR(1)</u>	<u>Yacc</u>	<u>C</u>	Mixed	external	<u>POSIX</u>	No	<u>Free, CPL & CDDL</u>
Yacc++	<u>LR(1)</u> , <u>LALR(1)</u>	<u>Yacc</u>	<u>C++, C#</u>	Mixed	generated or external	All	No	<u>Proprietary</u>
Yapps	<u>LL(1)</u>	?	<u>Python</u>	Mixed	generated	All	No	<u>Free, MIT</u>
yecc	<u>LALR(1)</u>	?	<u>Erlang</u>	Separate	generated	All	No	<u>Free, Apache 2.0</u>
<u>Visual BNF</u>	<u>LR(1)</u> , <u>LALR(1)</u>	?	<u>C#</u>	Separate	generated	<u>.NET Framework</u>	Yes	<u>Proprietary</u>
YooParse	<u>LR(1)</u> , <u>LALR(1)</u>	?	<u>C++</u>	Mixed	external	All	No	<u>Free, MIT</u>
Parse ^[11]	<u>LR(1)</u>	<u>BNF in C++ types</u>	?	?	none	<u>C++11 standard compiler</u>	No	<u>Free, MIT</u>
GGLL	<u>LL(1)</u>	Graph	<u>Java</u>	Mixed	generated	<u>Windows</u>	Yes	<u>Free, MIT</u>
Product	Parsing algorithm	Input grammar notation	Output languages	Grammar, code	Lexer	Development platform	IDE	License

Parsing expression grammars, deterministic boolean grammars

This table compares parser generators with parsing expression grammars, deterministic boolean grammars.

Name	Parsing algorithm	Output languages	Grammar, code	Development platform	License
AustenX	Packrat (modified)	<u>Java</u>	Separate	All	<u>Free, BSD</u>
Aurochs	Packrat	<u>C</u> , <u>OCaml</u> , <u>Java</u>	Mixed	All	<u>Free, GNU GPL</u>
BNFlite	Recursive descent	<u>C++</u>	Mixed	All	<u>Free, MIT</u>
Canopy	Packrat	<u>Java</u> , <u>JavaScript</u> , <u>Python</u> , <u>Ruby</u>	Separate	All	<u>Free, GNU GPL</u>
CL-peg	Packrat	<u>Common Lisp</u>	Mixed	All	<u>Free, MIT</u>
Drat!	Packrat	<u>D</u>	Mixed	All	<u>Free, GNU GPL</u>
Frisby	Packrat	<u>Haskell</u>	Mixed	All	<u>Free, BSD</u>
<u>grammar::peg</u>	Packrat	<u>Tcl</u>	Mixed	All	<u>Free, BSD</u>
Grako	Packrat + Cut + Left Recursion	<u>Python</u> , <u>C++</u> (beta)	Separate	All	<u>Free, BSD</u>
IronMeta	Packrat	<u>C#</u>	Mixed	<u>Windows</u>	<u>Free, BSD</u>
Laja	2-phase scannerless top-down backtracking + runtime support	<u>Java</u>	Separate	All	<u>Free, GNU GPL</u>
lars::Parser	Packrat (supporting left-recursion and grammar ambiguity)	<u>C++</u>	Identical	All	<u>Free, BSD</u>
LPeg	Parsing machine	<u>Lua</u>	Mixed	All	<u>Free, MIT</u>
lug	Parsing machine	<u>C++17</u>	Mixed	All	<u>Free, MIT</u>
Mouse	Recursive descent	<u>Java</u>	Separate	<u>Java virtual machine</u>	<u>Free, Apache 2.0</u>

Narwhal	Packrat	C	Mixed	POSIX , Windows	Free, BSD
Nearley	Earley	JavaScript	Mixed	All	Free, MIT
Nemerle.Peg	Recursive descent + Pratt	Nemerle	Separate	All	Free, BSD
neotoma	Packrat	Erlang	Separate	All	Free, MIT
NPEG	Recursive descent	C#	Mixed	All	Free, MIT
OMeta	Packrat (modified, partial memoization)	JavaScript , Squeak , Python	Mixed	All	Free, MIT
PackCC	Packrat (modified, left-recursion support)	C	Mixed	All	Free, MIT
Packrat	Packrat	Scheme	Mixed	All	Free, MIT
Pappy	Packrat	Haskell	Mixed	All	Free, BSD
parboiled	Recursive descent	Java , Scala	Mixed	Java virtual machine	Free, Apache 2.0
Lambda PEG	Recursive descent	Java	Mixed	Java virtual machine	Free, Apache 2.0
parsepp	Recursive descent	C++	Mixed	All	Free, public domain
Parsnip	Packrat	C++	Mixed	Windows	Free, GNU GPL
Patterns	Parsing machine	Swift	Identical	All	Free, MIT
peg	Recursive descent	C	Mixed	All	Free, MIT
PEG.js	Packrat (partial memoization)	JavaScript	Mixed	All	Free, MIT
Peggy ^[12]	Packrat (partial memoization)	JavaScript	Mixed	All	Free, MIT
Pegasus	Recursive descent, Packrat (selectively)	C#	Mixed	Windows	Free, MIT
pegc	Recursive descent	C	Mixed	All	Free, public domain
pest	Recursive descent	Rust	Separate	All	Free, MPL
PetitParser	Packrat	Smalltalk , Java , Dart	Mixed	All	Free, MIT
PEGTL	Recursive descent	C++11 , C++17	Mixed	All	Free, Boost
Parser Grammar Engine (PGE)	Hybrid recursive descent / operator precedence ^[13]	Parrot bytecode	Mixed	Parrot virtual machine	Free, Artistic 2.0
PyPy rlib	Packrat	Python	Mixed	All	Free, MIT
Rats!	Packrat	Java	Mixed	Java virtual machine	Free, GNU LGPL
Rekex (https://github.com/zhong-j-yu/rekex)	Recursive descent	Java	Mixed	Java virtual machine	Free, Apache 2.0
Spirit2	Recursive descent	C++	Mixed	All	Free, Boost
Treetop	Recursive descent	Ruby	Mixed	All	Free, MIT
Yard	Recursive descent	C++	Mixed	All	Free, MIT or public domain
Waxeye	Parsing machine	C , Java , JavaScript , Python , Racket , Ruby	Separate	All	Free, MIT

PHP PEG	PEG Parser?	PHP	Mixed	All	Free, BSD
---------	-------------	-----	-------	-----	---------------------------

General context-free, conjunctive, or boolean languages

This table compares parser generator languages with a general [context-free grammar](#), a [conjunctive grammar](#), or a [boolean grammar](#).

Name	Parsing algorithm	Input grammar notation	Output languages	Grammar, code	Lexer	Development platform	IDE	License
ACCENT	Earley	Yacc variant	C	Mixed	external	All	No	Free, GNU GPL
APaGeD	GLR , LALR(1) , LL(k)	?	D	Mixed	generated	All	No	Free, Artistic
Bison	LALR(1) , LR(1) , IELR(1) , GLR	Yacc	C , C++ , Java , XML	Mixed, except XML	external	All	No	Free, GNU GPL
DMS Software Reengineering Toolkit	GLR	?	Parlanse	Mixed	generated	Windows	No	Proprietary
DParser	Scannerless GLR	?	C	Mixed	scannerless	POSIX	No	Free, BSD
Dypgen	Runtime-extensible GLR	?	OCaml	Mixed	generated	All	No	Free, CeCILL-B
E3	Earley	?	OCaml	Mixed	external, or scannerless	All	No	?
Elkhound	GLR	?	C++ , OCaml	Mixed	external	All	No	Free, BSD
GDK	LALR(1) , GLR	?	C , Lex , Haskell , HTML , Java , Object Pascal , Yacc	Mixed	generated	POSIX	No	Free, MIT
Happy	LALR , GLR	?	Haskell	Mixed	external	All	No	Free, BSD
Hime Parser Generator	GLR	?	C# , Java , Rust	Separate	generated	.NET Framework , Java virtual machine	No	Free, GNU LGPL
IronText Library	LALR(1) , GLR	C#	C#	Mixed	generated or external	.NET Framework	No	Free, Apache 2.0
Jison	LALR(1) , LR(0) , SLR(1)	Yacc	JavaScript , C# , PHP	Mixed	generated	All	No	Free, MIT
Syntax	LALR(1) , LR(0) , SLR(1) , CLR(1) , LL(1)	JSON/Yacc	JavaScript , Python , PHP , Ruby , C++ , C# , Rust , Java	Mixed	generated	All	No	Free, MIT
Laja	Scannerless, two phase	Laja	Java	Separate	scannerless	All	No	Free, GNU GPL
ModelCC	Earley	Annotated class model	Java	Generated	generated	All	No	Free, BSD
P3	Earley–combinators	BNF-like	OCaml	Mixed	external, or scannerless	All	No	?
P4	Earley–combinators, infinitary CFGs	BNF-like	OCaml	Mixed	external, or scannerless	All	No	?
Scannerless Boolean Parser	Scannerless GLR (Boolean grammars)	?	Haskell , Java	Separate	scannerless	Java virtual machine	No	Free, BSD

<u>SDF/SGLR</u>	<u>Scannerless GLR</u>	<u>SDF</u>	<u>C, Java</u>	<u>Separate</u>	<u>scannerless</u>	<u>All</u>	<u>Yes</u>	<u>Free, BSD</u>
SmaCC	<u>GLR(1), LALR(1), LR(1)</u>	<u>?</u>	<u>Smalltalk</u>	<u>Mixed</u>	<u>internal</u>	<u>All</u>	<u>Yes</u>	<u>Free, MIT</u>
SPARK	<u>Earley</u>	<u>?</u>	<u>Python</u>	<u>Mixed</u>	<u>external</u>	<u>All</u>	<u>No</u>	<u>Free, MIT</u>
Tom	<u>GLR</u>	<u>?</u>	<u>C</u>	<u>Generated</u>	<u>none</u>	<u>All</u>	<u>No</u>	<u>Free, "No licensing or copyright restrictions"</u>
UltraGram	<u>LALR, LR, GLR</u>	<u>?</u>	<u>C++, C#, Java, Visual Basic .NET</u>	<u>Separate</u>	<u>generated</u>	<u>Windows</u>	<u>Yes</u>	<u>Proprietary</u>
Wormhole	<u>Pruning, LR, GLR, Scannerless GLR</u>	<u>?</u>	<u>C, Python</u>	<u>Mixed</u>	<u>scannerless</u>	<u>Windows</u>	<u>No</u>	<u>Free, MIT</u>
Whale Calf	General tabular, SLL(k), Linear normal form (conjunctive grammars), LR, Binary normal form (Boolean grammars)	<u>?</u>	<u>C++</u>	<u>Separate</u>	<u>external</u>	<u>All</u>	<u>No</u>	<u>Proprietary</u>
yaep	<u>Earley</u>	<u>Yacc-like</u>	<u>C</u>	<u>Mixed</u>	<u>external</u>	<u>All</u>	<u>No</u>	<u>Free, GNU LGPL</u>

Context-sensitive grammars

This table compares parser generators with context-sensitive grammars.

Name	Parsing algorithm	Input grammar notation	Boolean grammar abilities	Development platform	License
LuZc ^{[14][15]}	delta chain	modular	Conjunctive, not complimentary	<u>POSIX</u>	<u>Proprietary</u>
<u>bnf2xml (http://sourceforge.net/p/bnf2xml/)</u>	Recursive descent (is a text filter output is xml)	simple BNF grammar (input matching), output is xml	<u>?</u>	Beta, and not a full EBNF parser	<u>Free, GNU GPL</u>

See also

- Compiler-compiler
- List of program transformation systems

Notes

1. Bison 1.19 fork

References

1. "Ragel State Machine Compiler" (http://www.colm.net/open-source/ragel/).
2. http://www.colm.net/open-source/ragel/
3. "Adaptive LL(*) Parsing: The Power of Dynamic Analysis" (http://wwwantlr.org/papers/allstar-techreport.pdf) (PDF). Terence Parr. Retrieved 2016-04-03.
4. "Building parsers for the web with JavaCC & GWT (Part one)" (http://consoliii.blogspot.co.uk/2014/04/creating-gwt-compatible-parser-using.html). Chris Ainsley. 14 April 2014. Retrieved 2014-05-04.
5. "Racc" (https://i.loveruby.net/en/projects/racc/). *i.loveruby.net*. Retrieved 2021-11-26.
6. "Racc Grammar File Reference" (https://i.loveruby.net/en/projects/racc/doc/grammar.html). *i.loveruby.net*. Retrieved 2021-11-26.
7. "The SLK Parser Generator supports C, C++, Java, JavaScript, and C#, optional backtracking, free" (http://www.slkpg.site/).
8. http://www.slkpg.site/license.txt

9. "SLY (Sly Lex Yacc)" (<https://sly.readthedocs.io/en/latest/sly.html>).
10. "Tree-Sitter - An incremental parsing system for programming tools" (<https://tree-sitter.github.io/>).
11. "Parse - Compile time (LR) type safe parser generator for C++" (<https://github.com/MathiasVP/Parse/>). *GitHub*. 30 December 2021.
12. Maintained fork of PEG.js
13. "Parrot: Grammar Engine" (https://parrot.github.com/html/docs/book/pct/ch04_pge.pod.html). The Parrot Foundation. 2011. "PGE rules provide the full power of recursive descent parsing and operator precedence parsing."
14. "LuZ: A context sensitive parser" (<https://web.archive.org/web/20161017051112/http://qyxz.net.au.net/>). 2016-10-17. Archived from the original (<http://qyxz.net.au.net/>) on 2016-10-17. Retrieved 2018-10-17.
15. "LuZc – A conjunctive context-sensitive parser" (<http://luzc.zohosites.com/>). *luzc.zohosites.com*. Retrieved 2018-10-17.

External links

- [The Catalog of Compiler Construction Tools](http://catalog.compilertools.net/lexparse.html) (<http://catalog.compilertools.net/lexparse.html>)
 - [Open Source Parser Generators in Java](http://java-source.net/open-source/parser-generators) (<http://java-source.net/open-source/parser-generators>)
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Comparison_of_parser_generators&oldid=1079156879"

This page was last edited on 25 March 2022, at 09:16 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License 3.0; 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.