

Ragel State Machine Compiler

[Originalartikel](#)

[Backup](#)

Ragel compiles executable finite state machines from regular languages. Ragel targets C, C++ and ASM. Ragel state machines can not only recognize byte sequences as regular expression machines do, but can also **execute code** at arbitrary points in the recognition of a regular language. Code embedding is done using inline operators that do not disrupt the regular language syntax.

The core language consists of standard regular expression operators (such as **union**, **concatenation** and **Kleene star**) and action embedding operators. The user's regular expressions are compiled to a deterministic state machine and the embedded actions are associated with the transitions of the machine. Understanding the formal relationship between regular expressions and deterministic finite automata is key to using Ragel effectively.

Ragel also provides operators that let you control any non-determinism that you create, construct scanners, and build state machines using a statechart model. It is also possible to influence the execution of a state machine from inside an embedded action by jumping or calling to other parts of the machine, or reprocessing input.

Ragel provides a very flexible interface to the host language that attempts to place minimal restrictions on how the generated code is integrated into the application. The generated code has no dependencies.

Ragel code looks like:

```
action dgt { printf(„DGT: %c\n“, fc); } action dec { printf(„DEC: .\n“); } action
exp { printf(„EXP: %c\n“, fc); } action exp_sign { printf(„SGN: %c\n“, fc); } action number {
/*NUMBER*/ } number = (
```

```
[0-9]+ $dgt ( '.' @dec [0-9]+ $dgt )?
( [eE] ( [+\\-] $exp_sign )? [0-9]+ $exp )?
```

) %number; main := (number '\n');

.. and it compiles to:

```
if ( ++p == pe )
    goto out0;
if ( 48 &lt;= (*p) && (*p) &lt;= 57 )
    goto tr0;
goto st_err;
```

tr0:

```
{ printf("DGT: %c\n", (*p)); }
```

st1:

```
if ( ++p == pe )
    goto out1;
switch ( (*p) ) {
    case 10: goto tr5;
    case 46: goto tr7;
    case 69: goto st4;
    case 101: goto st4;
}
```

```
if ( 48 &lt;= (*p) &amp;&amp; (*p) &lt;= 57 )
    goto tr0;
goto st_err;
```

… and it visualizes as:

https://www.colm.net/images/number_lg.png

What kind of task is Ragel good for?

- Writing robust protocol implementations.
- Parsing data formats.
- Lexical analysis of programming languages.
- Validating user input.

Features

- Construct finite state machines using:
 - regular language operators
 - state chart operators
 - a scanner operator
 - some mix of the above
- Embed actions into machines in arbitrary places.
- Control non-determinism using guarded operators.
- Minimize state machines using Hopcroft's algorithm.
- Visualize output with <http://www.graphviz.org/> Graphviz.
- Use byte, double byte or word-sized alphabets.
- Generate C, C++ or ASM (GNU, x86_64, System V ABI) code with no dependencies.
- Choose from table or control flow driven state machines.

Download

Stable

March 24, 2017

<https://www.colm.net/files/ragel/ragel-6.10.tar.gz> (ragel-6.10.tar.gz) (<https://www.colm.net/files/ragel/ragel-6.10.tar.gz.asc> sig) (<https://www.colm.net/files/thurston.asc> key)

<https://www.colm.net/files/ragel/ragel-guide-6.10.pdf> ragel-guide-6.10.pdf

Development

May 11, 2017

<https://www.colm.net/files/ragel/ragel-7.0.0.10.tar.gz> ragel-7.0.0.10.tar.gz

Discussion and GIT Repos

The <https://www.colm.net/cgi-bin/mailman/listinfo/ragel> ragel mailing list is back on the colm.net domain.

```
git clone git://git.colm.net/ragel.git
```

License

Beginning with the next development release (> 7.0.0.9) Ragel is licensed under an MIT style license. Ragel 6 remains under GPL v2. Please see the file COPYING in the source.

Note: Part of the Ragel output is copied from Ragel source, covered by the MIT (or GPL v2) license. As an exception, you may use the parts of Ragel output copied from Ragel source without restriction. The remainder of Ragel output is derived from the input and inherits the copyright and license of the input file. Use of Ragel makes absolutely no requirement about the license of generated code.

From:
<https://schnipsl.qgelm.de/> - Qgelm

Permanent link:
<https://schnipsl.qgelm.de/doku.php?id=wallabag:ragel-state-machine-compiler>

Last update: **2021/12/06 15:24**

