# MetaRules

MetaRules is a grammar specifically used for defining other symbols and grammars.

## Tokens

|  |  |
| --- | --- |
| Token | Lexeme |
| number | `0`…`9` |
| char | `A`…`Z` | `a`…`z` | `\_` |
| escape\_char | `\` (`n` | `r` | `t` | `v` | `f` | `a` | `b` | `\` | "'" | `"` ) |
| start | `start` |
| string | `"` { char | escape\_char } `"` |
| var | char { char | number } |

## Grammar

Assume all symbols have the option of being space separate.

|  |  |
| --- | --- |
| Symbol | LEXEME |
| grouping | `(` rvalue `)` |
| option | `[` rvalue `]` |
| repetition0 (0 to more) | `{` rvalue `}` |
| repetition1 (1 to more) | `{` rvalue `}+` |
| wrap | grouping | option | repetition0 | repetition1 |
| regex | `regex` `(` string `)` |
| internal | string | var | regex | wrap |
| alternation | internal | ( alternation `|` internal ) |
| rvalue | alternation | ( rvalue alternation ) |
| assignment | var `=` rvalue `;` |
| program | assignment { assignment } |

## States

Forward edges of each state denote the transition to the next possible state.

../Downloads/staticlexer.png

## Ambiguities

Notice that concatenation is also represented as spaces. This can be resolved by evaluating the left and right symbols each set of spaces (spaces, tabs, and other standard delimiters).

../../../Downloads/staticlexer%20(3).png

## Resolving Ambiguities

Let the edge in state graphs represent either spaces or concatenation, resolve the state by:

../../../Downloads/staticlexer%20(4).png

## LR(1) Grammar

|  |  |
| --- | --- |
| Symbol | Lexeme |
| Start | Program |
| Program | Stmt |
| Program | S Stmt |
| Stmt | Var Assign Rvalue End |
| Rvalue | Mid |
| Rvalue | Rvalue Concat Mid |
| Alterop | Mid Alter |
| Mid | Base |
| Mid | Alterop Rvalue |
| Base | String |
| Base | Var |
| Base | Regex Lparen String Rparen |
| Base | Group |
| Base | Option |
| Base | Rep0 |
| Base | Rep1 |
| Group | Lparen Rvalue Rparen |
| Option | Lsb Rvalue Rsb |
| Rep0 | Lcb Rvalue Rcb |
| Rep1 | Lcb Rvalue Rcbmore |