Syntax

Regular Expressions

- Logic
- Symbols (1): $0, 1$
- Concatenation $\cdot$.
- Implication $\Rightarrow$

Terminals

- Table of US-ASCII:
- A **word** is recursively defined.
  - Basis (5):
    - 0 is a word.
    - 1 is a word.
  - Recursion (6): let w be a word.
    - $w \cdot 0$ is a word.
    - $w \cdot 1$ is a word.
- LAYOUT is any subset of $(0, 1)*$. Its elements are **white space characters.**

Atoms

- Strings are words delimited with either single quotes `'` or double quotes `"`. More precisely (), if w is a word without quotes, then:
  - ‘w’ is a string.
  - “w” is a stirng.

  w is the **contents** of the string.
- Escaped characters ():
  - Every instance of ' in w is replaced as ‘.
  - Every instance of " in w is replaced as “.
  - Every instance of \ in w is replaced by \.
- Identifiers are strings without white space.
- Numbers are a subset of strings with an injective function $q : \text{NUMBER} \to Q$.
  - Q is set of strings

    $$p/q$$

  where p, q are in scientific notation.

Grammar

- LL
  - Unambiguous
- Welkin Grammar: //SPDX-FileCopyrightText: 2023 Oscar Bender-Stone <oscarbenderstone@gmail.com>
  //SPDX-License-Identifier: CC-BY-4.0

  ```
  terms      = term*
  term       = connections | alias | graph | series | member
  connections = vertex (connector vertex)+
  connector   = "-" vertex? "-" -> edge
              | "-" vertex? "->" -> left_arrow
              | "<-" vertex? "-" -> right_arrow
  ```

```
alias       = vertex ":=" vertex
vertex      = graph | member _
graph       = (member | "_".) "{" terms "}"
series      = term "," (term ",")* term ","?
member      = "."+.(ident | string | "#" number).element*
            | unit.element*
element     = ".".(ident | string) | "#".number
unit        = ident | string | number
// TODO: fix ident! Must exclude whitespace!
ident       = CHAR*
string      = STRING
number      = NUMBER
```