RegEx: Home

This website provides some simple tools for experimenting with "classic" regular expressions as described in computer science textbooks and implemented in the early versions of *grep*. These tools are an outgrowth of research exploring automata that implement multiple formal languages simultaneously.

On the Examine page you can enter a regular expression to generate several kinds of data: parse trees, synthetic examples, and the automata state lists. Links from there produce diagrams of either the NFA or DFA for the language.

```
Regular Expression: \d+(\.\d*)?(e\d+)?
Augmented Parse Tree: ((([0-9]+(.[0-9]*)?)(e[0-9]+)?)#)
Examples:
6  8  1  8  7  0  8  1  8  0  9  2  9  8  5  4  3  6  4  4  2  9  4  5  8  3
4.7  1e1  9e9  94e25  64.66  39e83  64.0e9  93.9e04  01.29e0  5.  62.43e22
1e09  26.462  1808.75e6  7003.232e98  5.956  24e948  559  422.2e053
```
submit this example to see full output

The Compare page accepts multiple expressions and shows how their languages overlap or differ. The results page shows synthesized examples and indicates which expressions they match. You can also submit your own examples for testing. Again, there are links to produce automata diagrams.

3 expressions:
A: \d+\.\d*
B: \d+\.\d+
C: \d*\.\d+

A	B	C	example
–	–	✓	.1
✓	–	–	1.
✓	✓	✓	1.9

submit this example to see full output

The Syntax page briefly outlines the forms of regular expressions implemented here.

Home | Examine | Compare | Syntax | About | Contact