finite automaton: mechanical string recognizer

(NEs are generators)

transitions labelled with chars determine how we move between states
∅ are accepting states
∅ is start state

deterministic finite automaton: exactly one outgoing transition from each state for each char

non-deterministic violates that rule or/and has ε transitions (allow transitions w/ no reading input)

aabbab
machine accepts ambab

for scanning, we want DFAs that accept valid tokens

Pascal scanner (incomplete) (* comments *)

white space

non-

(key or comment ***)

digit

boolean is Comment(String s)
{ if (s.charAt(0) == '(')
   return false;
   if (s.charAt(0) == ')')
   return false;
   int i = 0;
   while (s.charAt(i) != 'w')
real constants

\[
\begin{align*}
&1 & 1.15 & 1.5E+10 \\
&10 & 16.94 & 1.5E10 \\
&3.1415926 & 3.94E-09 & 0.9E5
\end{align*}
\]

\[
\text{\textbf{RE $\rightarrow$ NFA}}
\]

\[
\begin{align*}
\varepsilon : & \rightarrow \circ \\
\alpha : & \rightarrow \circ \rightarrow \circ \\
\text{concatenation $\alpha \beta$ ($\alpha, \beta \text{ are REs}$)}
\end{align*}
\]

1) build NFA for $\alpha$
2) build NFA for $\beta$
3) $\rightarrow \alpha \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \beta$

$\alpha$: NFA for $(ab)^*$
$\beta$: NFA for $(abb)^+$

\[
\text{if (s.charAt(i+1) == ')' return true; else}
\]

\[
\text{(i++)}
\]
alternation $\alpha | \beta$

$ab | ab\beta$

Diagram: A graph with nodes and edges representing the automata for $\alpha$ and $\beta$. The diagram shows transitions between nodes labeled with symbols, indicating the states and transitions in the automaton.