**Lecture Thu 2/21, 2019**

**Course:** CS 396

**Title:** NFA

---

**Top Hat**

Table from prev. lecture:

<table>
<thead>
<tr>
<th>S</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>1</td>
<td>b</td>
</tr>
</tbody>
</table>

\[ \Delta(\{S\}, a) = \{S, b\} \]

Need to know the \( S \) table and:

\[ S = \{S, f\} \text{ with } F = \{f\} \]

\[ \Delta(S, a) = \{S, a\} \cup \Delta(f, a) \]

\[ \Delta(S, b) = \{S, b\} \cup \Delta(f, b) \]

\[ \Delta(f, a) = f \]

\[ \Delta(f, b) = \emptyset \]

\[ \Delta(S, a) = \{S, a\} \)

\[ \Delta(S, b) = \{S, b\} \)

\[ \Delta(f, a) = f \)

\[ \Delta(f, b) = \emptyset \]

Thus we needed only 5 of 12 possible states.

**Next Example:**

\[ N = \begin{array}{c}
 0 \\
 1 \\
 2 \\
 3 \\
\end{array} \]

\[ (0, a) \rightarrow (1, b) \rightarrow (2, a) \rightarrow (3, b) \rightarrow (0, a) \]

\[ L(N) = \{x \in \Sigma^* : \text{the third char from } 1 \leq x \leq 3 \text{ and the right is a } b \} \]

\[ = \{x \in \Sigma^* : 1 \leq x \leq 3 \text{ and } x_{n-2} = b \} \]

Next week: Given any \( K \geq 1 \), put \( N_k = (a \cup b)^* b (a \cup b)^* \).

The NFA \( N_k \) still needs only \( K+1 \) states. Exponential needs \( 2^K \) states!

**Defn.** A "generalized NFA" (GNFA) is \( N = (Q, \Sigma, \delta, s, F) \) again but with \( S \subseteq Q \times \text{Regexp}(\Sigma) \times Q \). "Instructions: look like"

\[ p \xrightarrow[\alpha]{\ } q \]

where \( \alpha \) is a regular expression over \( \Sigma \).

The instruction "processes" a substring \( W \) from \( p \) to \( q \) if \( W \) matches \( \alpha \).

---

Computation paths have the form \((q_0, W_1, q_1, W_2, q_2, \ldots, W_m, q_m)\)

where \( x = W_1 \ldots W_m \) and for all \( j \), \( 1 \leq j \leq m \), there is an instruction \((q_j-1, \alpha, q_j)\) such that \( W_j \) matches \( \alpha \). Then \( x \in L(q_0, q_m) \).
The General Two-State GNFA

$N: \alpha \xrightarrow{s} \beta \xrightarrow{\gamma} f$

$s = \eta$

Alternative Form "Centred" on $f$ not $s$.

$Lsf = (\gamma + \eta \alpha \beta)^*$.

$Lsf = \alpha \beta \cdot Lsf = (\gamma + \eta \alpha \beta)^*$.

Example when it simplifies further:

$Lsf = (aa)^* \cdot \varepsilon \cdot (ba + \emptyset \cdot (aa)^* \cdot \varepsilon)$

$= (aa)^* (ba + \emptyset)^* = (aa)^* (ba)^*$

Adding $Lss$ when $F = \{s, f\}$ does not change this particular language!

$Lss = (aa)^*$

$Lsf = Lss \cup Lsf = (aa)^* \cup (aa)^* (ba)^*$

Idea of the Conversion from GNFA to Regular Exp:

If $q$ is a state not in $F$ and $q \neq s$, then we can eliminate $q$ by bypassing all incoming arcs $(p, \beta, q)$ to all outgoing $(q, \eta, r)$.

Once done for each outgoing $r, r', ..., \text{delete edge } \beta$.

Once all incoming edges to $q$ are deleted, eliminate $q$.

Guts of algorithm is to eliminate all nonace states different from $s$.