**E-NFAs**

**Story So Far** We saw that adding the feature of non-determinism does not add to DFA's power — in terms of language recognition ability.

Now, we add another feature called **ε-move**

![Diagram](image)

**Means** transition without looking at or consuming any input symbol.

**E-NFA** same as an NFA, except allowed ε-move.

**Example**

![Diagram](image)

To get **N** for \( L = \{ w \mid w \text{ ends in 01 or 10} \} \)

**Combine**
WHY $\epsilon$-MOVES?

- USEFUL DESCRIPTIVE TOOL (FOR SPECIFICATION)
- GOOD FOR COMPOSING NFAs OR COMBINING THEM
- CAN STILL CONVNET TO DFAS & IMPLEMENT.

FORMALLY

$\epsilon$-NFA $N = (Q, \Sigma, \delta, q_0, F)$

WITH $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow 2^Q$

THUS CAN SAY $\delta(q_1, \epsilon) = \{q_2, q_3, q_4\}$

$\epsilon$-CLOSURE

$\forall q \in Q$, $\text{ECLOSE}(q)$ IS SET OF ALL STATES REACHABLE FROM $q$ USING ANY SEQUENCE OF $\epsilon$-MOVES, INCLUDING THE EMPTY SEQUENCE

THUS $q \in \text{ECLOSE}(q)$.

EXAMPLE

START $\rightarrow q_0 \rightarrow q_1 \rightarrow q_2$ $\epsilon$ $\rightarrow q_3$

OBserve

- $\text{ECLOSE}(q_0) = \{q_0, q_1, q_3\}$
- $\text{ECLOSE}(q_1) = \{q_1, q_3\}$
DEFINING \( \hat{\delta} \) : \( \forall q \in Q, \forall x \in \Sigma^* \), \( \forall a \in \Sigma \)

BASIS \( \hat{\delta}(q, \varepsilon) = \text{eClose}(q) \)

INDUCTION \( \hat{\delta}(q, xa) = \text{eClose} \bigg[ \bigcup_{p_i \in \hat{\delta}(q, x)} \hat{\delta}(p_i, a) \bigg] \)

\[ = \bigcup_{p_i \in \hat{\delta}(q, x)} \text{eClose} \bigg[ \hat{\delta}(p_i, a) \bigg] \]

INDUCTION EXPLAINED

1) LET \( \hat{\delta}(q, x) = \{ p_1, p_2, \ldots, p_k \} \)

2) LET \( \bigcup_{p_i \in \hat{\delta}(q, x)} \hat{\delta}(p_i, a) = \{ q_{i_1}, \ldots, q_{i_m} \} \)

3) THEN \( \hat{\delta}(q, xa) = \text{eClose} \bigg[ \{ q_{i_1}, \ldots, q_{i_m} \} \bigg] \)

\[ = \bigcup_{q_{i_j} \in \text{eClose}(q_i)} \text{eClose}(q_{i_j}) \]

THAT IS \( \hat{\delta}(q, \omega) \) IS SET OF ALL STATES REACHABLE FROM \( q \) ALONG PATHS WHOSE LABELS ON ARCS YIELD \( \omega \) (IGNORING ALL \( \varepsilon \)'S ON WAY)

NOTE

- \( q \in \hat{\delta}(q, \varepsilon) \)
- \( \delta(q, a) \neq \hat{\delta}(q, a) \) [UNLIKE NFA/ DFA]

INFACT \( \hat{\delta}(q, a) = \text{eClose} \bigg[ \delta(\text{eClose}(q), a) \bigg] \)

PRECEDING EXAMPLE

- \( \hat{\delta}(q_0, \varepsilon) = \{ q_0, q_1, q_3 \} \)
- \( \delta(q_0, \varepsilon) = \{ q_1 \} \)
- \( \hat{\delta}(q_1, \varepsilon) = \{ q_1, q_2 \} \)
**Language of e-NFA N**

\[ L(N) = \{ \omega \in \Sigma^* \mid \hat{\delta}(q_0, \omega) \cup F \neq \emptyset \} \]

**Theorem**

For all e-NFA N, there exists an NFA M such that \( L(M) = L(N) \).

**Idea**

Given e-NFA \( N = (Q, \Sigma, \delta, q_0, F) \), remove e-moves to construct an NFA \( M = (Q, \Sigma, \delta_M, q_0, F) \) adding new transitions to capture the effect of e-moves being removed.

**Formally**

In M, all is same as in N, except \( \delta_M \)

- Defining \( \delta_M \) + \( q \in Q \), \( \forall a \in \Sigma \)

\[ \delta_M(q, a) = \hat{\delta}(q_0, a) = \text{eClose}\left[ \delta(\text{eClose}(q), a) \right] \]

Observe \( \delta_M(q, \epsilon) \) is now undefined.

**Applying to preceding example**
 QUESTION: DO YOU SEE A "BUG" IN CONSTRUCTION?

 Observe in original ε-NFA N, we had ε ∈ L(N).

 Why? Because \( \delta(q_0, \varepsilon) = \varepsilon \text{close}(q_0) \)

 contained the final state \( q_3 \)

 But in new NFA M, must make \( q_0 \) a final state

to capture this possibility.

 In general, if ε ∈ L(N) or ε \text{close}(q_0) \cap F \neq \emptyset

 then add \( q_0 \) to \( F \).

 Done.

 Remark: Combining above conversion of an ε-NFA into an NFA, with the subset conversion of an NFA into a DFA, we can convert any ε-NFA directly into DFA.

 In fact, textbook gives this direct conversion into a DFA. — You just need to know the above.
REGULAR EXPRESSIONS

ALGEBRAIC NOTATION FOR DEFINING LANGUAGES — TAKES A DECLARATIVE VIEW UNLIKE COMPUTATIONAL VIEW IN MACHIN

ALGEBRA? FIX AN ALPHABET \( \Sigma \)

OPERANDS \( \varepsilon, \varnothing, \) AND ANY \( a \in \Sigma \)

OPERATORS \( + \) (UNION)
- \( \cdot \) (CONCATENATION)
- \( * \) (CLOSURE)

NOTATION FOR R.E. \( R \) CONSTRUCTED FROM ABOVE, \( L(R) \) WILL DENOTE THE LANGUAGE \( R \) DEFINE

BASIC REGULAR EXPRESSIONS

\[ R \]

\[ L(R) \]

\( \varepsilon \)

\( \{ \varepsilon \} \)

\( \varnothing \)

\( \{ \} \)

\( \forall a \in \Sigma \)

\( \{ a \} \)

UNION \( R, S \) ARE R.E. \( \Rightarrow R+S \) IS R.E.
WHERE \( L(R+S) = L(R) \cup L(S) \).

CONCATENATION \( R, S \) ARE R.E. \( \Rightarrow RS \) IS R.E.
WHERE \( L(R.S) = L(R) \cdot L(S) \)

RECALL FOR LANGUAGES \( L_1, L_2 \)

\[ L_1.L_2 = \{ w=xy \mid x \in L_1, y \in L_2 \} \]
EXAMPLES

\[ R = \text{e} + 1 \implies L(R) = \{ \text{e}, 1 \} \]
\[ S = \text{e} + 0 + 1 \implies L(S) = \{ \text{e}, 0, 1 \} \]
\[ T = RS \implies L(T) = \{ \text{e}, 0, 1, 10, 11 \} \]

LANGUAGE POWERS

\[ L^0 = \{ \text{e} \} \]
\[ L^1 = L \]
\[ L^2 = L \cdot L \]
\[ L^3 = L \cdot L \cdot L \]
\[ \vdots \]

\[ L^k = \{ w \mid w = x_1 x_2 x_3 \ldots x_k, \text{and } \forall i, x_i \in \text{GL} \} \]

CLOSURE

\[ L^* = L^0 \cup L^1 \cup L^2 \cup \ldots \]

CLOSURE

\[ R \text{ is R.E. } \implies R^* \text{ is R.E.} \]

WHERE

\[ L(R^*) = L(R)^* \]
\[ = L(\text{e}) \cup L(R) \cup L(R^2) \cup \ldots \]
\[ = L(\text{e}) \cup L(R) \cup L(R)^2 \cup \ldots \]

EXAMPLE

\[
\begin{cases}
R = 0 + 1 & \implies L(R) = \{ 0, 1 \} \\
S = R^* & \implies L(S) = \{ \text{ALL BINARY STRINGS} \} \\
R = 00 & \implies L(R^*) = \{ \text{ALL EVEN-LENGTH STRINGS OF 0'S} \}
\end{cases}
\]

POSITIVE CLOSURE

\[
\begin{cases}
L^+ = L^1 \cup L^2 \cup L^3 \cup \ldots \\
L(R^+) = L(R) \cup L(R)^2 \cup \ldots
\end{cases}
\]

EASY FACTS

\[ R^+ = R. R^* = R^* R \]
\[ R^* = \text{e} + R^+ \]
\[ L(0^*) = L(0) = \{ 0 \} \]
\[ L(1^*) = L(1) = \{ 1 \} \]
**Example**

\[
L(a^* b^*) = L(a^*) \cdot L(b^*) \\
= \{ \varepsilon, a, b, aa, ab, bb, aaa, aab, \ldots \} \\
= \{ \text{all } w \text{ where } a's \text{ precede } b's \}
\]

**Observe**

\[(a+b)^* \neq a^* b^* \]

**But** \[(a+b)^* = (a^* b^*)^*\]

---

**Precedence of Operators**

- **Highest**: \(*\)
- **Lowest**: \(+\)

**Thus**

\[ R + S^* T^* = R + (S \cdot (T^*)) \]

---

**Associativity**

\[(R \cdot S)^* T = R \cdot (S \cdot T) = R \cdot S \cdot T \]

\[(R + S)^* T = R + (S + T) = R + S + T \]

---

**Distributivity**

\[(R + S)^* T = R \cdot S + R \cdot T \]

---

**Defn** Languages generated by R.E. are called **Regular Languages**.

---

**Relation to Machines**

\[ \text{R.E.} \xrightarrow{?} \text{E-NFA} \xrightarrow{\text{Seen}} \text{NFA} \xrightarrow{?} \text{DFA} \xrightarrow{\text{Seen}} \]
**Theorem** \( \text{R.E.} \rightarrow \text{6-NFA} \)

For any regular language \( L(R) \) defined by R.E. \( R \),

there exists an 6-NFA \( N \) such that

\( L(N) = L(R) \).

**Proof** (Induction on \# operators in R.E.)

**Example**

\begin{align*}
R &= (a+b)^* c \quad \# R = 3 \\
S &= (a+b)^* \quad \# S = 2 \\
T &= c \quad \# T = 0
\end{align*}

**Observe** \( \# R = 1 + \# S + \# T \).

**Good 6-NFA has only 1 final state**

Show by induction on \# \( R \)

For all R.E. \( R \), \( \exists \) good 6-NFA \( N_R \) with

\( L(N_R) = L(R) \).

**Remark** We can prove stronger statement (see book)

For good \( \Theta \)-NFA with no arcs into \( q_0 \) and no arcs out of final state — but not below, although

the proof is the same.

**Basis** \( \# R = 0 \)

Clearly \( R = \epsilon, \phi \), or some \( a \in \Sigma \)

\[
\begin{align*}
N_\epsilon & \quad \rightarrow q_0 \quad \xrightarrow{\epsilon} \quad q_0 \\
N_\phi & \quad \rightarrow q_0 \\
N_a & \quad \rightarrow q_0 \quad \xrightarrow{a} \quad q_0
\end{align*}
\]
**INDUCTION**

Assume I.H. for \( \# R \leq R \), prove for \( \# R = k+1 \)

Let \( \# R = k+1 \)

\[ \Rightarrow 3 \text{ CASES} \begin{cases} R = S+T \\ R = ST \\ R = S^* \end{cases} \]

With \( \# S, \# T \leq R \).

So by I.H. we have good \( \epsilon \)-NFA's \( N_S \) and \( N_T \)

**CASE I** \( [R = S+T] \)

---

Note: \( q_0^S, q_0^T \) no longer final

Verify \( L(N_R) = L(N_S) \cup L(N_T) \)

\[ = L(S) \cup L(T) \]

\[ = L(S+T) \]

\[ = L(R). \]
CASE II  \[ R = \text{ST} \]

CASE III  \[ R = \text{S*} \]

EXAMPLE  \[ R = a^* + b \cdot c \]

CAN SIMPLIFY
**Theorem:** \([\text{DFA}s \rightarrow \text{R.E.}]

Let \(M\) be a DFA. Then \(\exists \text{ R.E. } R \text{ such that } L(R) = L(M)\).

**Proof:**

Given \(M = (Q, \Sigma, \delta, q_0, F)\)

Assume

\[\begin{align*}
&\bullet Q = \{q_1, q_2, \ldots, q_n\} \\
&\bullet q_0 \text{ is } q_1
\end{align*}\]

**Remark:** See also the alternate proof in the book.

**Definition:**

\[L_{ij} = \left\{ \omega \mid \delta(q_i, \omega) = q_j \right\}
\]

= \{strings taking \(M\) from \(q_i\) to \(q_j\)\}

Note \(L_{ij} = L(M_{ij})\), where \(M_{ij} = (Q, \Sigma, \delta, q_i, \{q_j\})\)

**Suppose** we could get R.E. \(R_{ij}\)'s for \(L_{ij}\)’s.

Then can compute \(R = \sum_{q_j \in F} R_{ij}\).

Because \(L(R) = \bigcup_{q_j \in F} L(R_{ij}) = \bigcup_{q_j \in F} L_{ij}\)

\[= \left\{ \omega \mid \delta(q_i, \omega) \in F \right\} = L(M)\]

**Defn:** For \(0 \leq r \leq n\),

\[L_{ij}^r = \left\{ \omega \mid \text{M goes from } q_i \text{ to } q_j \text{ on input } \omega \text{ using only } q_1, \ldots, q_r \text{ on way} \right\}\]
EXAMPLE

\[ \text{DEF } \text{EF } \in \mathcal{L}_{28} \]

\[ \text{ABC } \notin \mathcal{L}_{28} \quad \text{BUT} \quad \text{ABC } \in \mathcal{L}_{28} \]

\[ \text{ALSO} \quad \mathcal{L}_{24} = \{A, D\}, \quad \mathcal{L}_{25} = \{AEF, DEF\}, \quad \mathcal{L}_{28} = \{ABC, AEF, DBC, DEF\} \]

DEFINE \[ R^k_{ij} \] FOR \[ L^k_{ij} \]

FACT \[ R^0_{ij} = R^n_{ij}, \quad L^0_{ij} = L^n_{ij} \]

NOW FOCUS ON COMPUTING \[ R^k_{ij} \] — BY INDUCTION ON \[ k \]

2-STEP PROCESS

BASIS (CONSTRUCT \[ R^0_{ij} \]) FOR ALL \( i,j \)

INDUCTION (GIVEN ALL \[ R^k_{ij} \], COMPUTE ALL \[ R^{k+1}_{ij} \]

BASIS \( [R^0_{ij}] \)

OBSERVE \( \omega \in L^0_{ij} \) ONLY IF DIRECT TRANSITION

SUPPOSE

\[ \text{THEN} \quad R^0_{ij} = a_1 + a_2 + \ldots + a_k \]

SPECIAL CASES

A. NO DIRECT TRANSITION \[ \Rightarrow R^0_{ij} = \emptyset \]

B. \( i = j \) \[ \Rightarrow R^0_{ii} = \epsilon + a_1 + a_2 + \ldots + a_k \]
INDUCTION \([r \geq 1]\)  

Given \(R_{ij}^r\) values, compute \(R_{ij}^{r+1}\) values.

Clearly, we have \(R_{ij}^{r+1}\) but \(w \notin L_{ij}^r\) only if in going \(q_{i_j} \rightarrow q_{j_i}\), we use \(q_{r+1}\) at least once.

In general, \(w \in L_{ij}^{r+1} \subseteq L_{ij}^r\) is such that \(w = x_1 \ldots x_t\) with \(q_{r+1}\) hiding all states in \(\{q_{1}, \ldots, q_{r+2}\}\).

Clearly then, \(x_1 \in L_{i,j}^{r+1}\), \(x_2, \ldots, x_{t-1} \in L_{r+1,t+1}\), \(x_t \in L_{r+1,t+1}\).

Thus, \(R_{ij}^{r+1} = R_{ij}^r + R_{ij}^r \cdot (R_{r+1,t+1}) \cdot R_{r+1,t+1}\).

Example:

\[
\begin{array}{c}
\quad \rightarrow q_{i_1} \quad 1 \quad q_{i_2} \quad 0,1 \\
\end{array}
\]

<table>
<thead>
<tr>
<th>(k)</th>
<th>(R_{11}^k)</th>
<th>(R_{12}^k)</th>
<th>(R_{21}^k)</th>
<th>(R_{22}^k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(\varepsilon + 0)</td>
<td>1</td>
<td>(\emptyset)</td>
<td>(\varepsilon + 0 + 1)</td>
</tr>
<tr>
<td>1</td>
<td>((\varepsilon + 0)* + (\varepsilon + 0))</td>
<td>(O^* 1)</td>
<td>(\emptyset)</td>
<td>(\varepsilon + 0 + 1)</td>
</tr>
<tr>
<td>2</td>
<td>(O^* 1 + O^* 1)</td>
<td>((\varepsilon + 0 + 1)*)</td>
<td>(\emptyset)</td>
<td>(\varepsilon + 0 + 1)</td>
</tr>
</tbody>
</table>

\(L(M) = L(R_{12}^2)\), \(R = R_{12}^2\) \(= 0^* 1 (0+1)^*\).

All strings with at least one!