DFA (Deterministic Finite Autonoma) : $M = (Q, \Sigma, \delta, q_0, A)$, where

- Q : finite set of states of M
- Σ : finite set of alphabet of M (elements are characters)
- δ : transition function $Q \times \Sigma \to Q$, $\exists$ transition from q to q' on input a if $\delta(q, a) = q'$ (domain finite)
- $q_0 \in Q$: is the start state
- $A \subseteq Q$: set of final sets

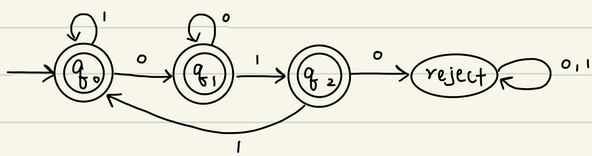
process single character

- $\hat{\delta} : Q \times \Sigma^* \to Q$, $\hat{\delta}(q, x)$ tells where M ends up after processing x

(domain can be infinite)

$\hat{\delta}$: extended transition function defined inductively by $\hat{\delta}(q, \varepsilon) = q$ and $\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x)\, a)$

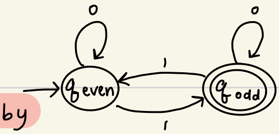
Exercise : Build a DFA M that recognizes the language
$L = \{x \mid x$ does not contain 010 as a substring$\}$



- M accepts x if $\hat{\delta}(q_0, x) = A$
- $L(M) := \{x \in \Sigma^* \mid \hat{\delta}(q_0, x) = A\}$ $L(M)$ is language of M

- languge is a subset of Σ^*

- If $\exists M$ s.t $L(M) = L$, then M "recognizes" L, L is DFA-recognizable proof: $L = \{ x \in \Sigma^* \mid x$ has odd number of 1s$\}$, M is given by



- WTS $L(M) = L$
  - i.e $x \in L(M)$ iff $x \in L$
  - i.e $\forall x \in \Sigma^*$, if $\hat{\delta}(q_{even}, x) = q_{odd}$, then x has odd # of 1s — let $P(x)$ be
- WTS $P(\varepsilon)$ and $P(xa)$ assuming $P(x)$
  - $P(\varepsilon)$: WTS if $\hat{\delta}(q_{even}, \varepsilon) = q_{odd}$ then ε has odd # 1s
    - vacuously true because $\hat{\delta}(q_{even}, \varepsilon) \neq q_{odd}$
  - $P(xa)$: WTS if $\hat{\delta}(q_{even}, xa) = q_{odd}$, then xa has odd # 1s
    - assume $\hat{\delta}(q_{even}, xa) = q_{odd}$ and $P(x)$
      - i.e $\delta(\hat{\delta}(q_{even}, x)a) = q_{odd}$
      - so $\begin{cases} a = 0 \text{ and } \hat{\delta}(q_{even}, x) = q_{odd} \\ a = 1 \text{ and } \hat{\delta}(q_{even}, x) = q_{even} \end{cases}$
      - STUCK!

* strengthened $P(x)$ $\begin{cases} ① \text{ If } \hat{\delta}(q_0, x) = q_{even} \text{ then } x \text{ has an even \# 1s} \\ ② \text{ If } \hat{\delta}(q_0, x) = q_{odd} \text{ then } x \text{ has an odd \# 1s} \end{cases}$

- $P(\varepsilon)$: WTS ① and ②
  - ①, ② holds because ε has even # 1s
- $P(xa)$: WTS ① and ② assuming $P(x)$
  - WTS ① $\begin{cases} \text{if } \hat{\delta}(q_0, x) = q_{even}, \text{ let } a = 0 \\ \text{if } \hat{\delta}(q_0, x) = q_{odd}, \text{ let } a = 1 \end{cases}$
  - WTS ② : $\hat{\delta}(q_{even}, xa) = q_{odd}$, so $\delta(\hat{\delta}(q_{even}, x)a) = q_{odd}$
    - If $\hat{\delta}(q_0, x) = q_{even}$ then $a = 1$, x has even #1s by $P(x)$

      so $xa = x1$ has odd #1s
    - If $\hat{\delta}(q_0, x) = q_{odd}$ then $a = 0$, x has odd #1s by $P(x)$

      so $xa = x0$ has odd # 1s
- If $x \in L(M)$, then $\hat{\delta}(q_{even}, x) = q_{odd}$ so by above x has odd # 1s
- If $x \notin L(M)$, then $\hat{\delta}(q_{even}, x) \neq q_{odd}$ so by above x does not have odd # 1s

proof: if L_1 and L_2 are DFA-recognizable, then so is $L_1 \cup L_2$

- choose arb L_1 and L_2, $\exists$ M_1 and M_2 s.t $L(M_1) = L_1$ and $L(M_2) = L_2$

$$\begin{cases} M_1 = (Q_1, \Sigma, \delta_1, q_{01}, A_1) \\ M_2 = (Q_2, \Sigma, \delta_2, q_{02}, A_2) \\ M = (Q, \Sigma, \delta, q_0, A) \end{cases}$$

$Q := Q_1 \times Q_2$

$q_0 := (q_{01}, q_{02})$

$A := \{(q_1, q_2) \mid q_1 \in A_1 \text{ or } q_2 \in A_2\}$

$\delta((q_1, q_2), a) := (\delta_1(q_1, a), \delta_2(q_2, a))$

- WTS $L(M) = L_1(M_1) \cup L_2(M_2)$

subclaim

- Let $p(x)$: $\forall x,\ \hat{\delta}(q_0, x) = (\hat{\delta}_1(q_{01}, x), \hat{\delta}_2(q_{02}, x))$
- WTS $p(\varepsilon)$ and $p(xa)$ assuming $p(x)$
  - $p(\varepsilon)$: WTS $\hat{\delta}(q_0, \varepsilon) = (\hat{\delta}_1(q_{01}, \varepsilon), \hat{\delta}_2(q_{02}, \varepsilon))$
    - plug in $\hat{\delta}$, $\hat{\delta}_1$, and $\hat{\delta}_2$ shows that $q_0 = (q_{01}, q_{02})$
  - $p(xa)$: assume $p(x)$

$$\begin{aligned} \hat{\delta}(q_0, xa) &= \delta(\hat{\delta}(q_0, x), a) && \text{by defn of } \hat{\delta} \\ &= \delta((\hat{\delta}_1(q_{01}, x), \hat{\delta}_2(q_{02}, x)), a) && \text{by } P(x) \\ &= (\delta_1(\hat{\delta}(q_{01}, x), a), \delta_2(\hat{\delta}_2(q_{02}, x), a)) && \text{by defn of } \delta \text{ above} \\ &= (\hat{\delta}_1(q_{01}, xa), \hat{\delta}_2(q_{02}, xa)) \end{aligned}$$

- i.e M accepts x iff either M_1 or M_2 accepts x
  - M accepts x means $\hat{\delta}(q_0, x) = A$
  - so $(\hat{\delta}(q_{01}, x), \hat{\delta}_2(q_{02}, x)) = \{(q_1, q_2) \mid q_1 \in A_1 \text{ or } q_2 \in A_2\}$
  - so $\hat{\delta}(q_{01}, x) = A_1$ or $\hat{\delta}_2(q_{02}, x) = A_2$

proof : ∃ unrecognizable language

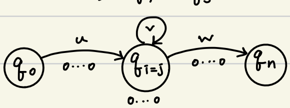
- language is set of sets of strings
  - Σ^* is infinite, so $|2^{\Sigma^*}|$ is uncountable
- String can be ... set of states, alphabet, transition function, etc
  - fewer DFA than string, so DFA is countable

pumping lemma

proof : If L is DFA-recognizable language, then $\exists$ some n s.t $\forall x \in L$ with $\text{len}(x) \geq n$, $\exists$ strings u, v, and w s.t

- $x = uvw$
- $\text{len}(uv) \leq n$
- $\text{len}(v) \geq 0$
- $\forall k \geq 0$, $uv^k w \in L$

rewritten proof : $L = \{0^n 1^n \mid n \in \mathbb{N}\}$ is not recognizable

- For the sake of contradiction, L were recognizable
  - i.e $\exists M$ with $L = L(M)$
- clearly $x \in L$, so M must accept x
  - while processing first n characters, M pass thru n+1 states $q_0, q_1 \cdots q_n$
  - since M takes n states, two of them are same
    create a loop ; $q_i = q_j$ for some $i < j \leq n$
  - 

    consist only 0s
  - uvvw is accepted, so $uvvw \in L(M)$
  - v consists of one or more 0s, so uvvw has more 0s than 1s
  - so $uvvw \notin L$
- This contradicts assumption $L(M) = L$

If it's possible to accept string x, it accepts x

NFA (Non-deterministic Finite Automation) : $N = (Q, \Sigma, \delta, q_0, A)$, where

- Q : finite set of states of M
- Σ : finite set of alphabet of M (elements are characters)
- δ : transition function $Q \times \Sigma \to 2^Q$
- $q_0 \in Q$: is the start state
- $A \subseteq Q$: set of final sets
- $\hat{\delta}$: extended transition function tells us where N could reach on input x

$\hat{\delta}(q, \varepsilon) = \hat{\varepsilon}(\{q\})$, $\hat{\delta}(q, xa) = \bigcup_{q' \in \hat{\delta}(q,x)} \delta(q', a)$

$\hat{\varepsilon} : 2^Q \to 2^Q$

- if S is a set of states, $\hat{\varepsilon}(S)$ is the set of states reachable from S with any # of ε-transitions "ε-closure of S"

- N accepts x if $\exists$ an accept state in $\hat{\delta}(q_0, x)$ i.e. $\hat{\delta}(q_0, x) \cap A \neq \emptyset$
- $L(N) := \{ x \in \Sigma^* \mid \hat{\delta}(q_0, x) = A \}$ L(N) is set of strings N accepts

convert NFA to DFA

proof : $\forall$ NFA N, $\exists$ DFA M with $L(M) = L(N)$

- choose an arb N each state of M corresponds to a set of states of N

subclaim

- $\forall x \in \Sigma^*$, if $\hat{\delta}_M(q_{0M}, x) = S$, then $\hat{\delta}_N(q_{0N}, x) = S$
  - i.e. $\hat{\delta}_M(q_{0M}, x) = \hat{\delta}_N(q_{0N}, x)$
  - prove by structural induction

regular expression: a way to define a language

- $r \in RE ::= \emptyset \mid \varepsilon \mid a \mid r_1 r_2 \mid (r_1 + r_2) \mid r^*$
  - $\emptyset$ matches no strings. $L(\emptyset) = \emptyset$
  - ε matches only the empty string. $L(\varepsilon) = \{\varepsilon\}$
  - a matches the string "a". $L(a) = \{a\}$
  - $r_1 r_2$ matches any string that can be broken into x and y with x matching r_1 and y matching r_2. $L(r_1 r_2) = \{xy \mid x \in L(r_1),\ y \in L(r_2)\}$
  - $r_1 + r_2$ matches any string that matches either r_1 or r_2. $L(r_1 + r_2) = L(r_1) \cup L(r_2)$
  - Kleene Star
  - r^* matches the concatenation of any number of strings, each of which match r. $L(r) = \{x_1 x_2 x_3 \ldots \mid x_i \in L(r)\}$

- x matches r if $x \in L(r)$. L(r) is the language of r

Exercise:

- $L = \{x \mid x$ has an odd # 1s$\}$ $\quad 0^* 1 0^*$
- $L = \{x \mid x$ has an even # 1s$\}$ $\quad (0^* 1 0^* 1 0^*)^*$

Kleene's theorem : L is regular if there is a regular expression $r \in RE$ with $L = L'(r)$

- L is DFA-recognizable
- L is NFA-recognizable
- L is regular

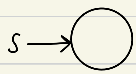
are all same

convert RE to NFA

proof : **$\forall r \in RE$, $\exists$ an NFA N with $L(N) = L(r)$**

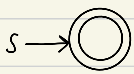
- let $p(r)$: $\exists N$ with $L(N) = L(r)$

$r = \emptyset$



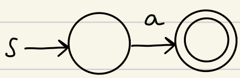
$p(\emptyset)$: $L(N) = L(\emptyset) = \{\emptyset\}$

$r = \varepsilon$



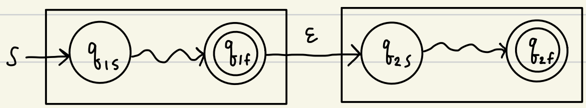
$p(\varepsilon)$: $L(N) = L(\varepsilon) = \{\varepsilon\}$

$r = a$



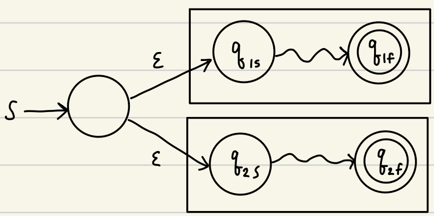
$p(a)$: $L(N) = L(a) = \{a\}$

$r = r_1 r_2$



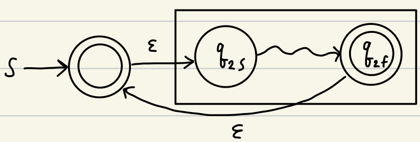
$p(r_1 r_2)$ assuming $p(r_1)$ and $p(r_2)$

$r = r_1 | r_2$



$p(r_1 | r_2)$ assuming $p(r_1)$ and $p(r_2)$

$r = r_1{}^*$

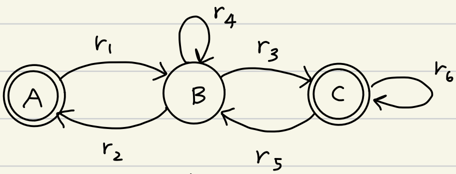


$p(r_1{}^*)$ assuming $p(r_1)$

convert NFA to RE

proof : ∀ NFA N, ∃ a RE r with L(N) = L(r)

- generalized NFA has RE on the transitions
- remove non-final, non-start states and replace them with RE



↓