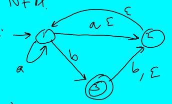
<u>Theorem</u>: Every NFA has an equivalent DFA.

<u>Cor</u>: A language is regular iff it is recognized by some NFA.

First, efficiently simulating an NFA.

<u>Ex</u>:




sample input: $w = abbaaab$

| | possible states |
|---|---|
| initial | 1,2 |
| a | 1,2 |
| b | 3,1,2 |
| b | 1,2,3 |
| a | 1,2 |
| a | 1,2 |
| a | 1,2 |
| b | 1,2,3 |

<u>Proof of the theorem</u>

<u>Def</u>: Let $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ be an NFA, and let $S$ be a set of states ($S \subseteq Q$). $S$ is <u>ε-closed</u> if there does not exist any ε-transition from a state $q \in S$ to a state $r \notin S$.

Given any $S \subseteq Q$, the <u>ε-closure</u> of $S$ ($\underline{\varepsilon\text{-cl}(S)}$) is the minimum ε-closed superset of $S$:

$$\varepsilon\text{-cl}(S) = \bigcap_{\substack{T \supseteq S \\ T \text{ is } \varepsilon\text{-closed}}} T$$

ε-cl(S) is a subset of <u>every</u> ε-closed superset of S.

Algo:

ε-cl(S):

  $C := S$

  <u>while</u> there exist $q \in C$ and $r \notin C$ such that $r \in \delta(q, \varepsilon)$,

    $C := C \cup \{r\}$

  <u>end while</u>

  <u>return C.</u>

Given A as above, define

$$D := \langle 2^Q, \Sigma, \Delta, Q_0, \mathcal{F} \rangle$$

where

$$Q_0 := \varepsilon\text{-cl}(\{q_0\})$$

$$\mathcal{F} := \{ S \subseteq Q : S \cap F \neq \emptyset \}$$

and for all $S \subseteq Q$, $a \in \Sigma$

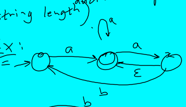
$$\Delta(S, a) := \varepsilon\text{-cl}\left( \bigcup_{q \in S} \delta(q, a) \right)$$

A is equiv to D

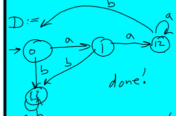
(Proof of $\boxed{\text{correctness}}$ omitted, but uses induction on input string length)

<u>Ex</u>:

A =




D :=




done!

~~Regular Expressions~~ (<u>Regexes</u>)

A regex is a shorthand to denote a language.

Fix an alphabet $\Sigma$.

**Def**: A regex over $\Sigma$ is an expression built from the following primitives and operators:

Atomic regexes:

$\emptyset$

$a$ (any $a \in \Sigma$)

Nonatomic: If $s$ & $t$ are regexes over $\Sigma$, then so are

$s \cup t$ (union)

$st$ (concatenation)

$s^*$ (star operator, Kleene closure, Kleene star)

Semantics (meaning).

Each regex over $\Sigma$ denotes a unique language over $\Sigma$ according to the following recursive rules.

Atomic

$\emptyset$ denotes $\emptyset = \{\}$

for $a \in \Sigma$,

$a$ denotes $\{a\}$

Nonatomic: Let $s$ denote $L$

Let $t$ denote $M$

then

$s \cup t$ denotes $L \cup M$

$st$ denotes $LM$

$:= \{xy : x \in L \ \& \ y \in M\}$

$s^*$ denotes

$\{\varepsilon\} \cup L \cup LL \cup LLL \cup \cdots \cup L^n \cup \cdots$

Ex:

$a \cup ab = \{a\} \cup \{a\}\{b\}$

$= \{a\} \cup \{ab\}$

$= \{a, ab\}$

$a^* = \{\varepsilon, a, aa, aaa, \ldots, a^n, \ldots\}$

$\emptyset^* = \{\varepsilon\} \cup \emptyset \cup \emptyset\emptyset \cup \cdots$

$= \{\varepsilon\}$

Convention: Let $\varepsilon$ denote $\emptyset^*$

(then $\varepsilon = \{\varepsilon\}$)

$(a \cup b)^* = \varepsilon \cup (a \cup b) \cup (a \cup b)(a \cup b) \cdots$

$= \{a, b\}^*$

$a^* \cup b^* \neq (a \cup b)^*$

Facts:

$\cup$ is commutative & associative

concat is associative

Syntactic sugar: ($r$ is a regex)

$\varepsilon := \emptyset^*$

$r^+ := rr^* = r^*r$

($r^*$ — 0 or more $r$'s)

($r^+$ — 1 " " ")

$r? := r \cup \varepsilon$

(0 or 1 occurrences of $r$)

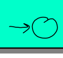
"optional" $r$

---

**Prop**: For any regex $r$ over $\Sigma$, $r$ is a regular (denotes) language.

**Proof**: For any regex $r$, construct an equivalent NFA as follows.

Atomic:

$\emptyset \rightsquigarrow$ 

Nonatomic: Let $s, t$ be regexs over $\Sigma$.

Assume we've built equivalent NFAs for $s$, $t$,

$r = s \cup t$: