Proof that the subset construction works

We start with a NFA

$$N = (Q, \Sigma, \delta_N, q_N, F_N)$$

we create a DFA

$$M = (\mathcal{P}(Q), \Sigma, \delta_D, q_D, F_D)$$

where: q_D = original start state q_N, also anything reachable by ε in NFA.

so $q_D = \delta_N^*(q_N, \varepsilon)$

F_D = set of subsets containing some element of F_N

$$= \{ S \in \mathcal{P}(Q) \mid S \cap F_N \neq \emptyset \}$$

$$\delta_D(R, a) = \bigcup_{q \in R} \delta_N^*(q, a)$$

Goal is to show $L(N) = L(M)$

remember

$$L(N) = \{ x \in \Sigma^* \mid \delta_N^*(q_N, x) \cap F_N \neq \emptyset \}$$

$$L(M) = \{ x \in \Sigma^* \mid \delta_D^*(q_D, x) \in F_D \}$$

$$= \{ x \in \Sigma^* \mid \delta_D^*(\delta_N^*(q_N, \varepsilon), x) \in F_D \}$$

$$= \{ x \in \Sigma^* \mid \delta_N^*(q_N, x) \in F_D \}$$

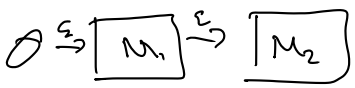
$$= \{ x \in \Sigma^* \mid \delta_N^*(q_N, x) \cap F_N \neq \emptyset \} = L(N)$$

DFA & NFA have the same computational power!

They both can compute regular languages.

Thm Regular langs are closed under:

- complements
- unions
- intersections } product construction
- difference (HW)
- concatenation



Regular Expressions

A regex is a systematic way of accepting / rejecting strings.

on $\Sigma = \{a, b\}$,

a regex looks like: $((ab)^* + b^*ab)a = r$,

this makes a language $L(r)$.

here, $aba \in L(r)$

$bb \notin L(r)$

How it works: given some alphabet Σ,

a regex consists of

letters, $+$, $*$, $($, $)$, ε, $\emptyset$

Letters represent themselves,

for $r = a$, then $L(r) = \{a\}$

$\emptyset$ gives the empty set $L(\emptyset) = \emptyset$

ε matches ε $L(\varepsilon) = \{\varepsilon\}$

Letters, $\emptyset$, ε are the 3 "atomic" regexs.

3 operations

$+$ (in programing, it's $|$) means "OR"

$L(a+b) = \{a, b\}$ $\quad \{a,b\}^*$

in general, $L(r_1 + r_2) = L(r_1) \cup L(r_2)$

$*$: matches the preceding thing 0 or more times

$L(a^*) = \{a^n \mid n \in \mathbb{N}\}$

$(a+b)^*$ matches any string made of a's & b's

aaa, aba, bb, ε, etc.

$a^* + b^*$ matches any a^n OR any b^n

generally, $L(r^*) = (L(r))^*$

concatenation

ab matches $\{ab\}$

$L(aab + ba) = \{aab, ba\}$

$L(a^*b) = \{a^n b \mid n \in \mathbb{N}\}$

$L(b^*a^*) = \{b^n a^m \mid n, m \in \mathbb{N}\}$

$L((ab)^*) = \{(ab)^n \mid n \in \mathbb{N}\}$

$L((a+b^2)^*)$ all strings made of a mixture of a's and bb's

$((ab)^* + b^*ab)a$

matches any string like $(ab)^n a$

or $b^n aba$