Equivalence of DFA & NFA

Thm 2.9 **Every NFA has an equivalent DFA.**

Proof: Let $N = (Q, \Sigma, \delta, q_0, F)$ be the NFA recognizing A. Construct a DFA $M = (Q', \Sigma, \delta', q_0', F)$ recognizing A. (Let
$E(R) = \{q \in Q | q \ can \ be \ reached \ from \ R \ by \ traveling \ along \ 0 \ or \ more \ \epsilon \ arrows\}$)

Define M as follows:

1. $Q' = P(Q)$
2. For $R \in Q'$ and $a \in \Sigma$, let

$$\begin{aligned} \delta'(R, a) &= \{q \in Q | q \in E(\delta(r, a)) \ for \ some \ r \in R\} \\ &= \cup_{r \in R} E(\delta(r, a)) \end{aligned}$$

3. $q_0' = E(\{q_0\})$
4. $F' = \{R \in Q' | R \cap F \neq \phi\}$

Cor 2.10 **A language is regular $\Leftrightarrow$ Some NFA recognizes it.**

(*Cor 2.10* can be used to prove *Thm 2.5* :**The class of regular languages is closed under union.**)

Thm 2.11 **The class of regular languages is closed under concatenation.**

Proof: Assume that L_1 and L_2 are two regular languages, and NFA N_1 and N_2 recognizes them respectively. Construct a new NFA to accept $L_1 L_2$ by using multiple $\overset{\epsilon}{\to}$ to connect the accept state(s) of N_1 and the start state of N_2. The constructed NFA N recognizes $L_1 L_2$.

Thm 2.12 **The class of regular languages is closed under Kleene Star.**

Proof: Assume that L is a regular language recognized by N. Construct a new NFA to accept L^* by adding a new start state before N's start state, and use multiple $\overset{\epsilon}{\to}$ to connect the accept state(s) of N and the added start state. The constructed NFA N' recognizes L^*.

Thm 2.13 **The class of regular languages is closed under complement.**

Proof: Let DFA $M = (Q, \Sigma, \delta, q_0, Qaccept)$ be a DFA recognizing A. Construct
$M' = (Q, \Sigma, \delta, q_0, Q'_{accept})$, where

$$Q'_{accept} = Q \setminus Q_{accept}$$

It is easy to verify $L(Q') = \bar{A}$.

Regular Expressions

eg.

Odd numbers	$(0 \cup 1 \cup \ldots \cup 9)^*(1 \cup 3 \cup 5 \cup 7 \cup 9)$
Strings that start with 0 or 1, then append zero or more 0s	$(0 \cup 1)0^*$
Σ^*	$(0 \cup 1)^*$
Strings that start with a 0 or end with a 1	$(0\Sigma^*) \cup (\Sigma^*1)$

Def 2.14 **R is a regular expression if R is:**

1. a for some $a \in \Sigma$
2. ϵ
3. ϕ
4. $(R_1 \cup R_2)$, where R_1, R_2 are regular expressions
5. (R_1R_2), where R_1, R_2 are regular expressions
6. (R^*), where R is a regular expression

---

eg.

1. $0^*10^* = \{w \in \{0,1\}^* | w \ contains \ a \ single \ 1\}$
2. $\Sigma^*1\Sigma^* = \{w | w \ contains \ at \ least \ one \ 1\}$
3. $\Sigma^*001\Sigma^* = \{w | w \ has \ a \ substring \ 001\}$
4. $1^*(01^+)^* = \{w | every \ 0 \ in \ w \ is \ followed \ by \ at \ least \ one \ 1\}$
5. $(\Sigma\Sigma)^* = \{w | the \ length \ of \ w \ is \ even\}$
6. $01 \cup 10 = \{01, 10\}$
7. $(0 \cup \epsilon)1^* = 01^* \cup 1^*$
8. $(0 \cup \epsilon)(1 \cup \epsilon) = \{\epsilon, 0, 1, 01\}$
9. $1^*\phi = \phi$
10. $\phi^* = \{\epsilon\}$

Thm 2.15 **A language is regular $\Leftrightarrow$ some regular expression describes it.**

Lem 2.16 **If a language is described by a regular expression, it's regular.**

Proof:

1. $R = a, a \in \Sigma, L(R) = \{a\}$

$$\rightarrow q_{start} \overset{a}{\rightarrow} q_{accept}$$

2. $R = \epsilon$

$$\rightarrow q_{accept}$$

3. $R = \phi$

$\rightarrow q_{start}$

4. $R = R_1 \cup R_2$ (*Thm 2.5*)
5. $R = R_1R_2$ (*Thm 2.6*)
6. $R = R_1^*$ (*Thm 2.11*)