- Regular languages: $\aleph_0$ (countable)
- All languages: $\aleph_1$ (uncountable)

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

$$DFA \rightarrow GNFA \rightarrow Regular\ expression$$

Def 2.17 **A generalized nondeterministic finite automaton (GNFA) is a 5-tuple $(Q, \Sigma, \delta, q_{start}, q_{accept})$, where**

1. Q is a finite set of states
2. Σ is the alphabet
3. $\delta : (Q - \{q_{accept}\}) \times R \rightarrow (Q - \{q_{start}\})$ is the transition function
   - R is the set of all regular expression, q_{accept} has no incoming edge, and q_{start} has no outgoing edge.
4. q_{start} is the start state
5. q_{accept} is the accept state

- **Except for the start state and accept state, there is one arrow from one state to every other state, and also from each state to itself.**

eg. 3-state DFA $\overset{?}{\rightarrow}$ 5-state GNFA $\overset{?}{\rightarrow}$ 4-state GNFA $\rightarrow$ 3-state GNFA $\rightarrow$ 2-state GNFA $\rightarrow$ ($q_{start} \overset{R}{\rightarrow} q_{accept}$)

Proof: Let M be the DFA for language A. Convert M to a GNFA as follows:

1. Add a new start state with ϵ arrow to the old start state
2. Add a new accept state with ϵ arrows from the old accept states
3. Replace multiple edges by one edge using "$\cup$"
4. Add arrows with ϕ between states that had no arrow
   - $Convert(G)$:
   1. Let k be the number of states in G.
   2. If $k = 2$, return $\delta(q_{start}, q_{accept})$
   3. If $k > 2$, choose $q_{rip} \in Q \setminus \{q_{start}, q_{accept}\}$
      - Let G' be the GNFA($Q', \Sigma, \delta', q_{start}, q_{accept}$), where
      1. $Q' = Q - \{q_{rip}\}$
      2. $\forall q_i \in Q' - \{q_{accept}\}, \forall q_j \in Q' - \{q_{start}\}$
      3. Let $\delta'(q_i, q_j) = \delta(q_i, q_j) \cup \delta(q_i, q_{rip})\delta(q_{rip}, q_{rip})^*\delta(q_{rip}, q_j)$

   (Notice q_i can be equal to q_j)
   4. Return $Convert(G')$

Claim: **For any GNFA G, $Convert(G)$ is equivalent to $Convert(G')$.**

Proof:

- $L(G) \subseteq L(G')$

  For any $w \in \Sigma^*$, if G accepts w, then G' accepts w

  $$q_{start}, q_{t_1}, q_{t_2}, \dots, q_{accept}$$

  If none of them is q_{rip}, then G' also accepts w

  If q_{rip} appears (Assume that $q_{t_{i+1}} = q_{rip}$)

  $$q_{start}, q_{t_1}, \dots, q_{t_i}, q_{rip}, q_{t_{i+2}}, \dots$$

  Then we can use $\delta(q_{t_i}, q_{t_{i+1}})\delta(q_{t_{i+1}}, q_{t_{i+2}})$ to connect q_{t_i} and $q_{t_{i+2}}$

  Then, G' also accepts w

- $L(G') \subseteq L(G)$

  ......

Lem 2.17(Pumping Lemma) **If A is a regular language, then $\exists p \in \mathbb{N}$, s.t. for any string s of length at least p, $\exists x, y, z \in \Sigma^*$ s.t. $s = xyz$ and**

1. $xy^iz \in A$ for every $i \geq 0$
2. $|y| > 0$
3. $|xy| \leq p$

Proof:

Let $M = (Q, \Sigma, \delta, q_{start}, F)$ be a DFA recognizing A. Let $P = |Q|, S = s_1s_2 \dots s_n \in \Sigma^n, n \geq p$

Let $r_1, r_2, \dots, r_{n+1} \in Q$ be the sequence of n states. That is

$$r_1 = q_{start}$$
$$r_{i+1} = \delta(r_i, s_i), for\ i = 1, 2, \dots, n$$

By PHP(Pegion-House Principle), there exists at least 2 states that are same. Call the first r_j, the second r_l

Let $x = s_1s_2 \dots s_{j-1}$, $y = s_js_{j+1} \dots s_{l-1}$, $z = s_ls_{l+1} \dots s_n$

(Consider the first occurence of repeated states as such, $l - 1 \leq p$, i.e. $|xy| \leq p$)

Algorithm

What is an algorithm?

An algorithm is a mechanical process to be followed in calculations or other problem-solving operation.

Def 3.1(Turing Machine) **A Turing Machine is a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, where**

1. Q is the set of states
2. Σ is the input alphabet, $\sqcup \notin \Sigma$
3. Γ is the tape alphabet, where $\sqcup \in \Gamma, \triangleright \in \Gamma, \Sigma \subseteq \Gamma$
4. $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\}$ is the transition function
5. q_0 is the start state

6. q_{accept} is the accept state
7. q_{reject} is the reject state