Part IA Formal Languages and Automata Notes

1 Languages

- An alphabet is a finite set, Σ, made up of symbols
- A string of length n is an ordered n-tuple of elements of Σ
- The empty string is ε
- Σ^* is the set of all strings of Σ of a finite length
- If $\Sigma = \varnothing$ then $\Sigma^* = \{\varepsilon\}$
- Concatenation: Strings u and v can be joined to give uv
- Given an alphabet Σ, we can call any subset of Σ^* a formal language over the alphabet Σ
- Axioms are specified by giving an element of U

$$\frac{}{a}$$

- Rules giving a finite subset of U, $\{h_1, \ldots, h_n\}$ (the hypothesis) and an element c (the conclusion)

$$\frac{h_1, \ldots, h_n}{c}$$

- A deviation that a particular element $u \in U$ is in the subset is
  - A finite rooted tree with vertices labelled by elements of U such that:
    - The root is u
    - Each vertex is the conclusion of a rule whose hypothesis are children of the vertex
    - Each leaf is an axiom
- Given a relation R, its transitive closure R^+ is the smallest relation that contains R and is transitive

$$\forall x, y, z \in X.(x, y) \in R^* \wedge (y, z) \in R^* \implies (x, z) \in R^*$$

- R^+ can be defined as follows

$$\text{Axioms: } \frac{}{(x,y)} \text{ for all } (x, y) \in R); \quad \text{Rules: } \frac{(x,y) \quad (y,z)}{(x,z)} \text{ (for all } x, y, z \in X)$$

- Given a relation R, its reflexive-transitive closure R^* is the smallest relation which contains R and is transitive and reflexive

$$\forall x \in X.(x, x) \in R^*$$

- R^* can be defined as follows

$$\text{Axioms: } \frac{}{(x,y)} \text{ for all } (x, y) \in R), \quad \frac{}{(x,x)} \text{ (for all } x \in X); \quad \text{Rules: } \frac{(x,y) \quad (y,z)}{(x,z)} \text{ (for all } x, y, z \in X)$$

2 Regular Expressions

- Regular Expressions over a given alphabet Σ
- Let Σ' be the set $\{\varepsilon, \varnothing, |, *, (,)\}$
- The language of regular expressions is defined as follows, $U = \Sigma \cup \Sigma'$
- Or
  - Axioms

$$\frac{}{a} \qquad \frac{}{\varepsilon} \qquad \frac{}{\varnothing}$$

  - Rules

$$\frac{r}{(r)} \qquad \frac{r \quad s}{r|s} \qquad \frac{r \quad s}{rs} \qquad \frac{r}{r*}$$

  where $a \in \Sigma$ and $r, s \in U$

- Order of precedence

$$\text{Star} > \text{Concat} > \text{Union}$$

- Concat and Union are left associative
- No strings match the regular expression $\varnothing$

3 Finite Automata

3.1 Non-deterministic Finite Automata

- Defined by a 5-tuple

$$M = (Q, \Sigma, \Delta, s, F)$$

  - Q: Finite set of states
  - Σ: Finite set of input symbols
  - Δ: Subset of $Q \times \Sigma \times Q$, the transition relation
  - s: Start state, $s \in Q$
  - F: Finite set of accepting states, $F \subseteq Q$

3.2 Deterministic Finite Automata

- All state-symbol pains must have exactly one mapping in Δ
- Δ is the total function δ
- Defined by the 5-tuple

$$M = (Q, \Sigma, \delta, s, F)$$

3.3 Non-deterministic Finite Automata with ε-Transitions

- Defiend by the 6-tuple

$$M = (Q, \Sigma, \Delta, s, F, T)$$

- As above with

$$T \subseteq Q \times Q, \text{ the } \varepsilon\text{-transition relation}$$

4 Subset Construction

- Given a NFA$^\varepsilon$, M, a DFA, PM can be generated
- Start state of PM is a set containing M's start state and all states accessible by ε-transitions from the start state
- Accepting states of PM would be any subset of M's states containing an accepting state
- Alphabet is the same
- PM's δ for all subsets of M's states given an input symbol find all states accessible from the states in the subset
- An empty set is a stock state (a state in which all outward transitions are to itself)
- For each NFA$^\varepsilon$ $M = (Q, \Sigma, \Delta, s, F, T)$ there is a DFA, $PM = (\mathcal{P}(Q), \Sigma, \delta, s', F')$ that accepts exactly the same strings

Consider a string $a_1 a_2 \ldots a_n \in L(M)$, is accepted by the NFA$^\varepsilon$ M

$$\begin{tikzcd}
s \arrow[r, Rightarrow, "a_1"] \arrow[d, phantom, "\Cap"] & q_1 \arrow[r, Rightarrow, "a_2"] \arrow[d, phantom, "\Cap"] & \ldots \arrow[r, Rightarrow, "a_n"] & q_n \in F \arrow[d, phantom, "\Cap"] \\
s' \arrow[r, Rightarrow, "a_1"] & s_1 \arrow[r, Rightarrow, "a_2"] & \ldots \arrow[r, Rightarrow, "a_n"] & s_n \in F'
\end{tikzcd}$$

so $a_1 a_2 \ldots a_n \in L(PM)$, the same applies the other way, so

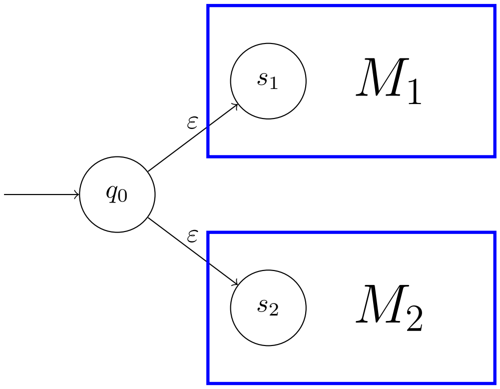
$$L(PM) \subseteq L(M) \wedge L(M) \subseteq L(PM) \implies L(PM) = L(M)$$

5 Kleene's Theorem

- A language is regular iff it is equal to $L(M)$ for some deterministic finite automaton M
- For any regular expression, the set of strings matched is a regular language

Induction on the depth of the syntax tree
Base Case: $\{a\}$, $\{\varepsilon\}$, and $\varnothing$ are regular
Induction:

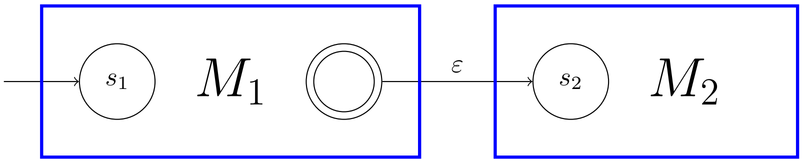
1. For $r_1|r_2$



For $u \in L(M_1)$, $s_1 \overset{u}{\Longrightarrow} q_1$ since $q_0 \overset{\varepsilon}{\Longrightarrow} s$, it is accepted, the same can be said for M_2

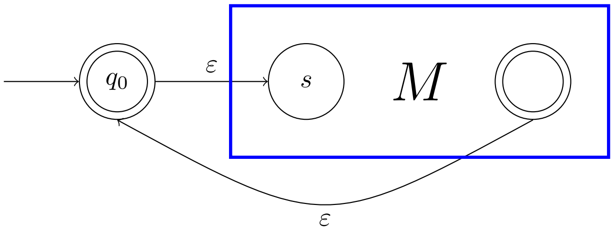
Can't accept anything else because can only go to the start state of M_1 or M_2

2. For $r_1 r_2$



   Accepting states those in M_2

3. For $r*$



   Accepts ε because start state accepting, anything accepted by M must be made up of components accepted by M

- Given a NFA can a regular expression be found

Induction on number of states
Base Case: $S = \varepsilon$
Induction: S has $n+1$ elements
Pick some $q_0 \in S$ and define S^- to be $S \setminus \{q_0\}$
Get from $qtpq'$ avoiding q_0 or via q_0 visiting q_0 an arbitrary number of times

$$r^S_{q,q_1} = r^{S^-}_{q,q'} \Big| r^{S^-}_{q,q_0} \left[r^{S^-}_{q_0,q_0} \right]^* r^{S^-}_{q_0,q'}$$

- Two DFAs are equivelent
  - iff, $L(r) \subseteq L(s) \land L(s) \subseteq L(r)$
  - iff, $L((\sim r)\ \&\ s) = \varnothing = L((\sim s)\ \&\ r)$

6 The Pumping Lemma

- The Pumping Lemma property is necessary for a language to be regular, but not sufficient
- Stated as follows:

For every regular language, L, there is a number $l \leq 1$ satisfying the pumping lemma property:
All $w \in L$ with $|w| \geq l$ can be expressed as a concatenation of three strings, $w = u_1 v u_2$, where $u_1 v u_2$ satisfy

$$|v| \geq 1$$
$$|u_1 v| \leq l$$
$$\forall n \geq 0\ u_1 v^n u_2 \in P$$

Write as

$$
\begin{array}{l}
\forall L \subseteq \Sigma^* \\
L \text{ regular} \implies \exists l \geq 1 \text{ s.t.} \\
\quad \forall w \in L. |w| \geq l \\
\quad \exists u_1, v, u_2 \in \Sigma^* \text{ with} \\
\qquad |u_1 v| \leq l \\
\qquad |v| \geq 1 \\
\qquad \forall n \in \mathbb{N}_0 . u_1 v^n u_2 \in L
\end{array}
$$

6.1 Proof of the Pumping Lemma

Suppose $L = L(M)$ for a DFA $M = (Q, \Sigma, \Delta, s, F)$

Take $l = \#Q$, if $n \geq l$ then

$$s = \underbrace{q_0 \overset{a_1}{\to} q_1 \overset{a_2}{\to} q_2 \dots \overset{a_l}{\to} q_l}_{l+1 \text{ states}} \dots \overset{a_n}{\to} q_n \in F$$

q_0 to q_l cannot all be distinct states, so $q_i = q_j$, for some $0 \leq i \leq j \leq l$ so the above looks like

$$\begin{tikzcd}
s = q_0 \arrow[r, "u_1"] & q_i \arrow[rr, bend left, "v"] & = & q_j \arrow[r, "u_2"] & q_n \in F
\end{tikzcd}$$

where $u_1 \triangleq a_1 \dots a_i$, $v \triangleq a_{i+1} \dots a_j$, and $u_2 \triangleq a_{j+1} \dots a_n$