Closure of Regular Langs Under Union, Intersection, Complementation, and Projection
Exposition by William Gasarch

1 Introduction

We give the constructions that show sketch the proof that all if L_1 and L_2 are regular and $L_1 \cap L_2$, $L_1 \cup L_2$, $\overline{L}$, and $proj(L)$ (which we will define) are regular.

Def 1.1 A DFA is a tuple $(Q, \Sigma, \delta, s, F)$ where $\delta : Q \times \Sigma \rightarrow Q$.

We define running a DFA M on a string x in the obvious way. If the DFA ends in a state in F then x is accepted. Otherwise its rejected.

2 The Construction for Intersection

Theorem 2.1 *If L_1 and L_2 are regular then $L_1 \cap L_2$ is regular.*

Proof:
Let $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ be the DFA for L_1. Let $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ be the DFA for L_2.
We define the DFA for $L_1 \cap L_2$. Let $M = (Q_1 \times Q_2, \Sigma, \delta, (s_1, s_2), F_1 \times F_2)$ where δ is defines by, for $(q_1, q_2) \in Q_1 \times Q_2$ and $\sigma \in \Sigma$,

$$\delta((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)).$$

The intuition is that the DFA M runs M_1 and M_2 at the same time. If both end up in $F_1 \times F_2$ then both M_1 and M_2 accepted. ∎

3 The Construction for Union

Theorem 3.1 *If L_1 and L_2 are regular then $L_1 \cup L_2$ is regular.*

Proof:
Let $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ be the DFA for L_1. Let $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ be the DFA for L_2.
We define the DFA for $L_1 \cup L_2$. Let $M = (Q_1 \times Q_2, \Sigma, \delta, (s_1, s_2), F_1 \times Q_2 \cup Q_1 \times F_2)$ where δ is defines by, for $(q_1, q_2) \in Q_1 \times Q_2$ and $\sigma \in \Sigma$,

$$\delta((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)).$$

The intuition is that the DFA M runs M_1 and M_2 at the same time. If M_1 ends up in F_1 then we accept (independent of what M_2 does), and if M_2 ends up in F_2 then we accept (independent of what M_1 does). ∎

4 The Construction for Complementation

Theorem 4.1 *If L is regular then $\overline{L}$ is regular.*

Proof:
Let $M = (Q, \Sigma, \delta, s, F)$ be the DFA for L.
We define the DFA for $\overline{L}$. Let $M' = (Q, \Sigma, \delta, s, Q - F)$ (recall that $Q - F = \{q \mid q \in Q \wedge q \notin F\}$.
The intuition is that the DFA M' runs M but does the opposite when it comes to accepting.
▮

5 The Construction for Complimentation

To Compliment a DFA you say
My DFA, you lovely you look.

6 The Construction for Nondeterminism

Recall the definition of an NDFA:

Def 6.1 An NDFA is a tuple $(Q, \Sigma, \Delta, s, F)$ where $\Delta : Q \times (\Sigma \cup e) \rightarrow 2^Q$. (Recall that 2^Q is the powerset of Q.

We DO NOT define running an NDFA M on a string x. Instead we say that an NDFA accepts x if SOME way of running the machine ends up in a state in F.

Theorem 6.2 *If L is accepted by an NDFA then there exists a DFA such that accepts L.*

Proof: Let $M = (Q, \Sigma, \Delta, s, F)$ be the NDFA for L.
We define the DFA for L. Let $M' = (2^Q, \Sigma, \delta, s, \mathcal{F})$ where for $A \in 2^Q$ and $\sigma \in \Sigma$,

$$\delta(A, \sigma) = \bigcup_{q \in A} \Delta(e^a q e^b, \sigma)$$

(The e^a and e^b are strings of the empty string.)

$$\mathcal{F} = \{A \mid A \cap F \neq \emptyset\}$$

The intuition is that the DFA M' runs ALL possibilities for M. If SOME possibility ends up accepting, then accept. ▮

7 Closure under Projection

Notation 7.1 Let$\Sigma = \{0,1\}^n$. Note that each element of Σ is itself a string of n bits. If $x \in \Sigma^*$ then $proj(x)$ is what you get by taking each symbol in x and chopping off the last bit. So if $x \in (\{0,1\}^n)^*$ then $proj(x) \in (\{0,1\}^{n-1})^*$. If $L \subseteq (\{0,1\}^n)^*$ then

$$proj(L) = \{proj(x) \mid x \in L\}.$$

Theorem 7.2 *If L is regular than $proj(L)$ is regular.*

Proof: Let $M = (Q, (\{0,1\}^n), \delta, s, F)$ be the DFA for L.

We define an NDFA for L. Let $M' = (Q, \{0,1\}^{n-1}, \Delta, s, F)$. For $q \in Q$ and $\sigma \in \{0,1\}^{n-1}$

$$\Delta(q, \sigma) = \{\delta(q, \sigma 0), \delta(q, \sigma 1)\}.$$

∎