Closure Properties of Regular Languages
Closure properties

• In general, if S is a set, and foo() is a binary operator defined on elements of S, then we say S is closed under foo iff for any pair of elements s, t in S, foo(s,t) is also in S

• Similarly, if bar() is a unary operator defined on S, then we say S is closed under bar iff for all s in S, bar(s) in S
Regular languages closed under $foo$.
Regular languages not closed under $bar$. 

$foo(L_1, L_2) \rightarrow L_1 \leftarrow bar(L_1, L_2)$
Theorem

The set of regular languages is closed under:

- union
- concatenation
- Kleene *
- complement
- intersection
- all boolean operations
- homomorphism
- inverse homomorphism
- many other operations too

Why do we even care?
- capabilities/limitations
- design

Not this year
Proof: union, concatenation, Kleene *

If $L_1$ and $L_2$ are regular, then by definition,

- $L_1 \cup L_2$ is regular
- $L_1L_2$ is regular
- $(L_1)^*$ is regular
Proof via regular expressions

If $L_1$ and $L_2$ are regular, then there are regular expressions $r_1$ and $r_2$ such that

$$L(r_1) = L_1 \text{ and } L(r_2) = L_2$$

and so

$r_1 + r_2$ denotes $L_1 \cup L_2$, hence the union is regular

$r_1 r_2$ denotes $L_1 L_2$, so the concatenation is regular

$r_1^*$ denotes $L_1^*$ so the Kleene star of $L_1$ is regular
Proof: complement

“JUST SWAP THE ACCEPT AND REJECT STATES !”

Let $L$ over alphabet $\Sigma$ be regular.

We show $\Sigma^* - L$ is regular.

Since $L$ regular, it is recognized by some

DFA $M = (Q, \Sigma, \delta, q_0, F)$.

Let $M' = (Q, \Sigma, \delta, q_0, Q-F)$ be the DFA obtained by swapping accept and nonaccept states.

$M'$ accepts $w$ iff $q_0 \xrightarrow{w} p$ iff in $M$ $q_0 \xrightarrow{w} p$

Thus $M'$ recognizes $\Sigma^* - L$

(what if we started with an NFA for $L$ instead?)
Proof: intersection

1. Closed under union; closed under complement, thus closed under intersection via DeMorgan’s law. \( R \cap S = \overline{R} \cup \overline{S} \)
2. “Cross-product” construction we saw earlier
3. Complement of regular expression?

Proof: other Boolean operations

1. Union/complement/intersection are universal
2. “Cross-product” construction we saw earlier
Proof: homomorphism

A homomorphism for finite alphabets $\Sigma$ and $\Delta$ is a function $h: \Sigma \rightarrow \Delta^*$ that is extended to inputs in $\Sigma^*$ via concatenation: $h(au) = h(a)h(u)$.

So, for symbols $a_1,a_2,...a_n$ with $a_i$ in $\Sigma$,

$h(a_1a_2...a_n) = h(a_1)h(a_2)h(a_3).... h(a_n)$

For $L$ a language, $h(L) = \{h(w): w \text{ in } L\}$
Examples. Let $L = \{0^n1^n : n \geq 0\}$.

- Let $h(0) = ab$, and $h(1) = ba$
  Then $h(0011) = ababbaba$
  and $h(L) = \{(ab)^n(ba)^n : n \geq 0\}$
- Let $h(0) = h(1) = a$, then $h(0011) = aaaa$
  and $h(L) = \{a^n a^n : n \geq 0\} = \{a^{2n} : n \geq 0\}$
  = strings of a’s of even length
- Let $h(0) = a$, $h(1) = \varepsilon$, so $h(0011) = aa$
  and $h(L) = a^*$
If $L$ is regular and $h$ is a homomorphism, then $h(L)$ is regular

- Let $L$ be regular over $\Sigma$, and $h: \Sigma \rightarrow \Delta^*$.  
- Let $r$ be a regular expression for $L$.  
- Define $h(r) = \text{replace each } a \text{ in } \Sigma \text{ with } h(a) \text{ to get a regular expression for } h(L)$

**EXAMPLE:** $\Sigma = \{0,1\}$, $\Delta = \{a,b,c\}$,  
$h(0) = abac$, $h(1) = cb$  

$L = 0^*1^* + 1(0+1)0^*$  
$h(L) = (abac)^*(cb)^* + cb(abac+cb)(abac)^*$
**Inverse homomorphism**

- Let \( h : \Sigma \rightarrow \Delta^* \) be a homomorphism.
- Let \( L \) be a subset of \( \Delta^* \).
- Define

  \[
  h^{-1}(L) = \{ w \in \Sigma^* : h(w) \in L \} = \text{strings whose homomorphs are in } L
  \]
Example

• $h: \{a,b,c\} \rightarrow \{0,1\}$
• $h(a) = 0$, $h(b) = 1$, $h(c) = 01$
• $L = 0^*1^*$
• $h^{-1}(0011) = \{aabb, acb\}$
• $h^{-1}(011) = \{abb, cb\}$
• $h^{-1}(L) = a^*(c + \varepsilon)b^*$
If $L$ is regular w/ alphabet $\Delta$, and $h: \Sigma \to \Delta^*$ is a homomorphism, then $h^{-1}(L)$ is regular

• Given $M$ for $L$, construct $M^{-1}$ accepting $h^{-1}(L)$
• $Q$, $q_0$, $F$ the same.
• $\delta^{-1}$ : while reading $w$, run $M$ on $h(w)$
\[ \delta \text{ : \ while reading } w, \quad M^{-1} \text{ runs } M \text{ on } h(w) \]

\[
\begin{align*}
h(0) &= aab \\
h(1) &= ba
\end{align*}
\]

- \( M \) accepts \( baaab \)
- \( h^{-1}(baaab) = 10 \)
- So \( 10 \) should be accepted by \( M^{-1} \)
- By construction, \( M^{-1}(10) = M(baaab) \) and so accepts.
Fun with closure properties

Can use to show languages regular:

IF: Each $L_i$ is regular  
THEN: $L$ must be regular
Example

\[ \text{Majority}(L_1, L_2, L_3) = \{w \mid w \text{ in at least 2 of the } L_i\} \]

Then \textit{Majority} is regular if \( L_i \) are, because

\[ \text{Majority}(L_1, L_2, L_3) = (L_1 \cap L_2) \cup (L_1 \cap L_3) \cup (L_1 \cap L_3) \]
Fun with closure properties

Can use to show languages NOT regular:

Apply closure properties

THEN: \( L_? \) must be non-regular
Example

• Later we’ll learn that \( \{0^n1^n: n \geq 0\} \) is not regular
• Use it to show \( L = \{w: \#0(w) = \#1(w)\} \) isn’t regular
• We know \( L(0^*1^*) \) is regular (since it has a reg. exp.)
• \( L \cap 0^*1^* = \)
  \[ \{0^n1^n: n \geq 0\} \]
• If \( L \) were regular, and regular languages are closed under intersection, then \( \{0^n1^n: n \geq 0\} \) would be regular, but it is not. So \( L \) couldn’t be.