Outline

FSAs, formally

Formal Languages

Regular Expressions

Notes
Building an automaton with formalities...

A FSA is a quintuple: $(Q, \Sigma, q_0, F, \delta)$
Example: Multiple of 3 machine
Extended Transition Function

\[ \delta^* : Q \times \Sigma^* \rightarrow Q \]

\[ \delta^*(q, s) = \begin{cases} 
q & \text{if } s = \varepsilon \\
\delta(\delta^*(q, s'), x) & \text{if } s' \in \Sigma^*, x \in \Sigma, s = s'x
\end{cases} \]
Extended Transition Function - example
Example: Even machine
Devise a machine that accepts strings over \( \{a, b\} \) with an even number of \( a \)'s

FSA to accept \( L = \{s \in \{a, b\}^*: s \text{ contains an even } \# \text{ of } a \text{’s}\} \)
Example: Even machine

Does $L(A) = L$? Proof by induction using state invariant
Example: Even machine
Does $L(A) = L$? Proof by induction using state invariant
Example: Even machine

Does $L(A) = L$? Proof by induction using state invariant
More odd/even: intersection

$L$ is the language of binary strings with an even number of $a$s, and at least one $b$

Devise a machine for $L$
More odd/even: union

$L$ is the language of binary strings with an even number of $a$s, or at least one $b$

Devise a machine that accepts $L$,
Back to languages
More language operations

Rev($L$): $= \{ s^R : s \in L \}$

concatenation: $LL'$ or $L \cdot L'$ = \{ $rt$ | $r \in L$, $t \in L'$ \}.

Special cases $L\{\varepsilon\} = L = \{\varepsilon\}L$, $L\{\} = \{} = \{\}L$. 
More language operations

exponentiation: $L^k$ is concatenation of $L$ $k$ times. Special case, $L^0 = \{\epsilon\}$, including $L = \{\}$ (!)

Kleene star: $L^* = L^0 \cup L^1 \cup L^2 \cup \ldots$. 
Another way to define languages
In addition to the language accepted by DFSA $L(M)$ and set description $L = \{\ldots\}$.

Definition: The regular expressions (regexps or REs) over alphabet $\Sigma$ is the smallest set such that:

1. $\{\}, \epsilon$ are REs
2. $a$ for every $a \in \Sigma$ are REs over $\Sigma$
3. if $R$ and $S$ are REs over $\Sigma$, then so are:
   - $R + S$ (union) — lowest precedence operator
   - $RS$ (concatenation) — middle precedence operator
   - $R^*$ (star) — highest precedence
Regular Expression to Language

- $L(\emptyset) = \emptyset$ (the empty language — no strings!)
- $L(\epsilon) = \{\epsilon\}$ (the language consisting of just the empty string)
- $L(x) = \{x\}$ (the language consisting of the one-symbol string)

- $L(S + T) = L(S) \cup L(T)$
- $L(ST) = L(S)L(T)$
- $L(T^*) = L(T)^*$
RE Examples

- \( L(a + b) = \{a, b\} \)

- \( L(ab) = \{ab\} \)

- \( L((a + b)a) = \{aa, ba\} = L(aa + ba) \)

- \( L(a^*) = \{\epsilon, a, aa, aaaa, \ldots\} \)

- \( L(aa^*) = \)

- \( L((ab)^*) = \)

- \( L(a^*b^*) = \)
RE Examples

- $L((a + b)^*) =$

- $L(a^* + b^*) =$

- $L((a + b)(a + b)^*) =$

- All strings of $a$’s and $b$’s that have the same first and last symbol?

- All strings of $a$’s and $b$’s that contain at least 1 $a$?