Outline

FSAs, continued

FSAs, formally

Notes
odd/even

$L$ is the language of binary strings \{a, b\} with an even length.
Devise a machine for $L$. 
Example: Even machine

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

$L$ is the language of binary strings with an even number of $a$

Devise a machine for $L$
Example: Even machine

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

$L$ is the language of binary strings with an even number of $a$s, but even length.

Devise a machine for $L$. 

Example: Even machine

Does \( L(A) = L \)? Proof by induction using state invariant
Example: Multiple of 3 machine
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 $L$, 
Building an automaton with formalities...

A FSA is a quintuple: \((Q, \Sigma, \delta, q_0, F)\)

- \(Q\) is set of states,
- \(\Sigma\) is finite, non-empty alphabet,
- \(\delta: Q \times \Sigma \mapsto 2^Q\) or \(\delta \subseteq Q \times \Sigma \times Q\) is transition function or transition relation,
- \(q_0\) is start state, and
- \(F\) is set of accepting states
Extended Transition Function

\[ \delta^* : Q \times \Sigma^* \rightarrow 2^Q \]

\[ \delta^*(q, s) = \begin{cases} 
\{q\} & \text{if } s = \varepsilon \\
\delta(q', x) & \text{if } s' \in \Sigma^*, x \in \Sigma, s = s'x \\
& \text{and } q' \in \delta^*(q, s') 
\end{cases} \]
Extended Transition Relation

\[ \delta^* \subseteq Q \times \Sigma^* \times Q \]

\[ \delta^*(q, s) = \begin{cases} 
\{q\} & \text{if } s = \varepsilon \\
\delta(q', x) & \text{if } s' \in \Sigma^*, x \in \Sigma, s = s'x \\
& \text{and } q' \in \delta^*(q, s')
\end{cases} \]
Extended Transition Function - example
Example of NFA:

Give an NFA that recognizes the language $L = \{ \varepsilon, (ab + aba)^i, ab, ababa, abaaba, \ldots \}$
Configurations and Computation Steps of NFAs
Do DFA and NFA recognize the same languages? (i.e. are equivalent)