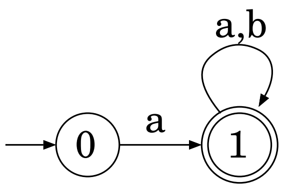
04-0: **Non-Determinism**

- A Deterministic Finite Automata's transition function has exactly one transition for each state/symbol pair
- A *Non*-Deterministic Finite Automata can have 0, 1 or more transitions for a single state/symbol pair
- Example: $L = \{w \in \{a, b\} : w \text{ starts with } a\}$
  - Regular expression?

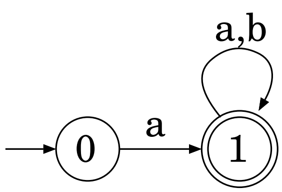
04-1: **NFA Example**

- Example: $L = \{w \in \{a, b\} : w \text{ starts with } a\}$
  - a(a+b)*



04-2: **NFA Example**

- Example: $L = \{w \in \{a, b\} : w \text{ starts with } a\}$
  - a(a+b)*



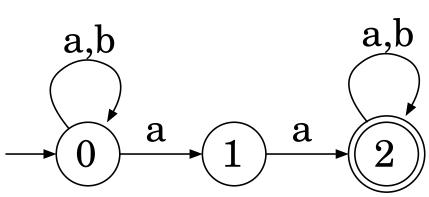
- What happens if a 'b' is seen in state q_0?
  - The machine "crashes", and does not accept the string

04-3: **NFA Example**

- Example: $L = \{w \in \{a, b\} : w \text{ contains the substring } aa\}$
  - Regular Expression?

04-4: **NFA Example**

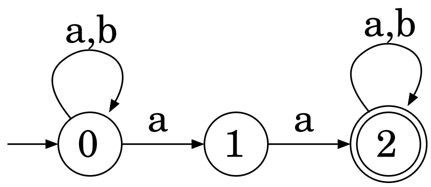
- Example: $L = \{w \in \{a, b\} : w \text{ contains the substring } aa\}$
  - (a+b)*aa(a+b)*



- What happens if a a is seen in state q_0?

04-5: **NFA Example**

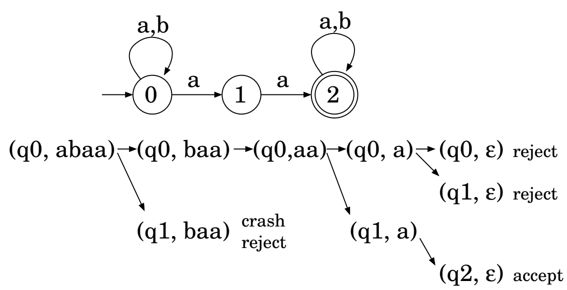
- Example: $L = \{w \in \{a, b\} : w$ contains the substring $aa\}$
  - (a+b)*aa(a+b)*



- What happens if a a is seen in state q_0?
  - Stay in state q_0, or go on to state q_1
  - Multiple Computational Paths (board example)

04-6: **NFA Example**

- Example: $L = \{w \in \{a, b\} : w$ contains the substring $aa\}$



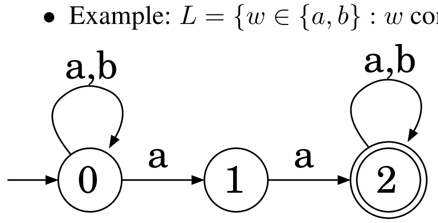
- Does this machine accept $abaa$?

04-7: **NFA Acceptance**

- If there is any computational path that accepts a string, then the machine accepts the string
- Two ways to think about NFAs:
  - Magic "Oracle", which always picks the correct path to take
  - Try all possible paths

04-8: **NFA Example**

- Example: $L = \{w \in \{a, b\} : w$ contains the substring $aa\}$



- If a string contains aa, will there be a computational path that accepts it?
- If a string does not contain aa, will there be a computational path that accepts it?

04-9: **NFA Definition**

- Difference between a DFA and an NFA
  - DFA has exactly only transition for each state/symbol pair
    - $\delta : (K \times \Sigma) \mapsto K$
  - NFA has 0, 1 or more transitions for each state/symbol pair

04-10: **NFA Definition**

- Difference between a DFA and an NFA
  - DFA has exactly only transition for each state/symbol pair
    - Transition function: $\delta : (K \times \Sigma) \mapsto K$
  - NFA has 0, 1 or more transitions for each state/symbol pair
    - Transition relation: $\Delta \subseteq ((K \times \Sigma) \times K)$

04-11: **NFA Definition**

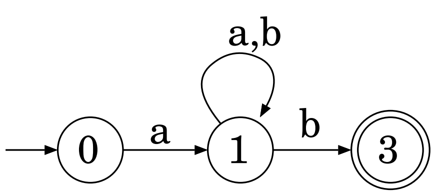
- A NFA is a 5-tuple $M = (K, \Sigma, \Delta, s, F)$
  - K Set of states
  - Σ Alphabet
  - $\Delta : (K \times \Sigma) \times K$ is a Transition relation
  - $s \in K$ Initial state
  - $F \subseteq K$ Final states

04-12: **Fun with NFA**
Create an NFA for:

- All strings over {a, b} that start with a and end with b

04-13: **Fun with NFA**
Create an NFA for:

- All strings over {a, b} that start with a and end with b



(example compuational paths for ababb, abba, bbab)

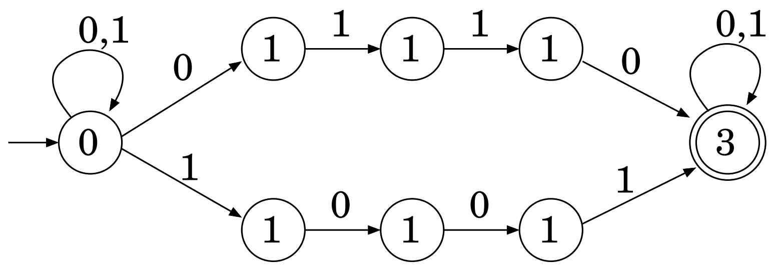
04-14: **Fun with NFA**
Create an NFA for:

- All strings over {0, 1} that contian 0110 or 1001

04-15: **Fun with NFA**

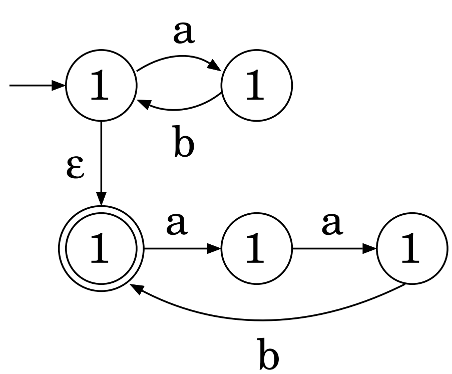
Create an NFA for:

- All strings over $\{0, 1\}$ that contian 0110 or 1001



04-16: ϵ**-Transitions**

- ϵ transition consumes no input
  - NFA (with ϵ transitions) for (ab)*(aab)*



04-17: ϵ**-Transitions**

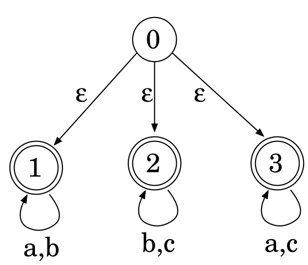
Create an NFA (with ϵ-transitions) for:

- All strings over $\{a, b, c\}$ that are missing at least one letter. For example, aabba, cbbc, ccacc $\in L$, while abbc $\notin L$

04-18: ϵ**-Transitions**

Create an NFA (with ϵ-transitions) for:

- All strings over $\{a, b, c\}$ that are missing at least one letter. For example, aabba, cbbc, ccacc $\in L$, while abbc $\notin L$



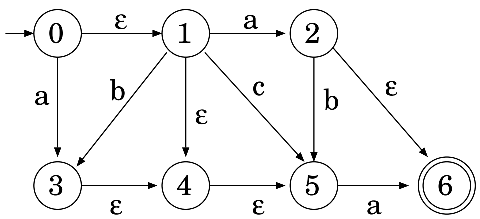
abcb, bbab, abbab, abc

04-19: **Yet More Formalism**

- ϵ-closure
  - ϵ-closure(q) = set of all states that can be reached following zero or more ϵ-transitions from state q.

04-20: ϵ**-closure**

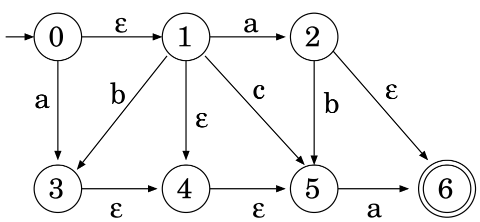
- ϵ-closure examples



(quick review: What is L[M]?)

04-21: ϵ**-closure**

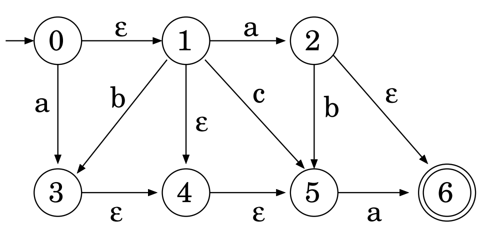
- ϵ-closure examples



L[M] = {a, aa, ba, ca, aba}

04-22: ϵ**-closure**

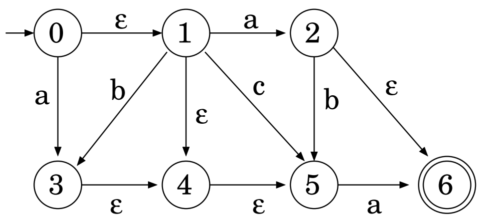
- ϵ-closure examples



- ϵ-closure(q_0) =
- ϵ-closure(q_3) =
- ϵ-closure(q_2) =
- ϵ-closure(q_5) =

04-23: ϵ**-closure**

- ϵ-closure examples



- ϵ-closure(q_0) = $\{q_0, q_1, q_4, q_5\}$
- ϵ-closure(q_3) = $\{q_3, q_4, q_5\}$
- ϵ-closure(q_2) = $\{q_2, q_6\}$
- ϵ-closure(q_5) = $\{q_5\}$

04-24: **ϵ-closure – Sets**

- ϵ-closure
  - ϵ-closure(q) = set of all states that can be reached following zero or more ϵ-transitions from state q.
- Can extend ϵ-closure to a set of states
  - ϵ-closure(S) = $\bigcup\{A : q \in S \wedge \epsilon\text{-closure}(q) = A\}$

04-25: **NFA Definition (revised)**

- A NFA is a 5-tuple $M = (K, \Sigma, \Delta, s, F)$
  - K Set of states
  - Σ Alphabet
  - $\Delta : (K \times (\Sigma \cup \{\epsilon\})) \times K$ is a Transition relation
  - $s \in K$ Initial state
  - $F \subseteq K$ Final states

04-26: **NFA** $\vdash_M$

- Binary relation $\vdash_M$: What machine M yields in one step
  - $\vdash_M \subseteq (K \times \Sigma^*) \times (K \times \Sigma^*)$
  - $\vdash_M = \{((q_1, aw), (q_2, w)) : q_1, q_2 \in K_M, w \in \Sigma_M^*, a \in \Sigma_M \cup \{\epsilon\}, ((q_1, a), q_2) \in \Delta_M\}$
- Binary relation $\vdash_M^*$: Transitive, reflexive closure of $\vdash_M$

04-27: **NFA Languages**

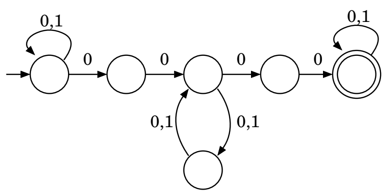
- $L[M]$ – Language defined by NFA M
  - $L[M] = \{w : (s_M, w) \vdash_M^* (f, \epsilon)$ for some $f \in F_M*\}$
- $L_{NFA} = \{L : \exists NFA \quad M, L[M] = L\}$

04-28: **NFA Examples**

- Create an NFA for the language
  - Give an NFA for the language L = All strings over $\{0,1\}$ that contain two pairs of adjacent 0's separated by an even number of symbols. So, 0100110011, 01100101100101, and 01001000 are in the language, but 0100100, 1011001, and 0111011 are not in the language.

04-29: **NFA Examples**

- Create an NFA for the language
  - Give an NFA for the language L = All strings over $\{0,1\}$ that contain two pairs of adjacent 0's separated by an even number of symbols. So, 0100110011, 01100101100101, and 01001000 are in the language, but 0100100, 1011001, and 0111011 are not in the language.



04-30: **NFA Examples**

- Create an NFA for the language
  - L = All strings over $\{a,b\}$ that have an a as one of the last 3 charaters in the string. So, a, baab, bbbab, aabbaaabb $\in L$, but bb, baabbb, bbabbbbb $\notin L$

04-31: **NFA Examples**

- Create an NFA for the language
  - L = All strings over $\{a,b\}$ that have an a as one of the last 3 charaters in the string. So, a, baab, bbbab, aabbaaabb $\in L$, but bb, baabbb, bbabbbbb $\notin L$