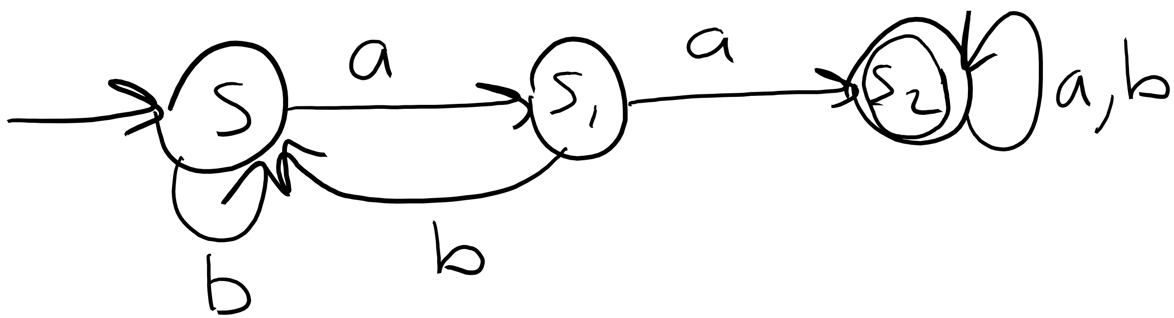
Modeling Computation

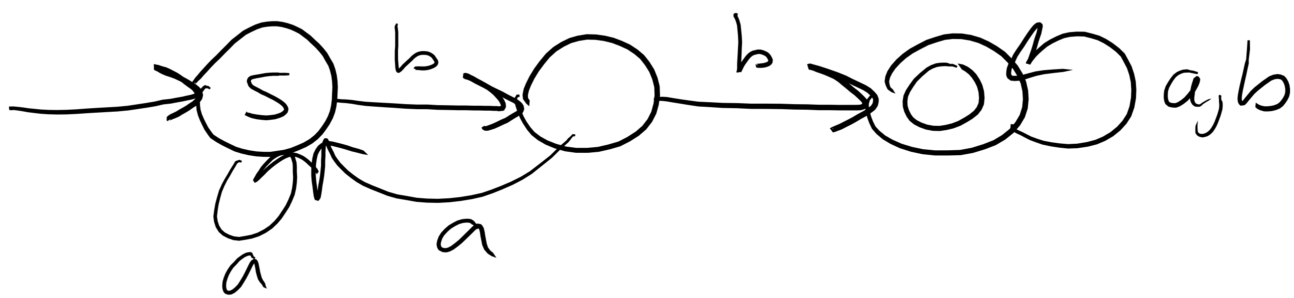
Introduction to Formal Languages and Automata

NFAs

DFA that accepts all strings in $\{a, b\}^*$ that contain aa (2 consecutive as).

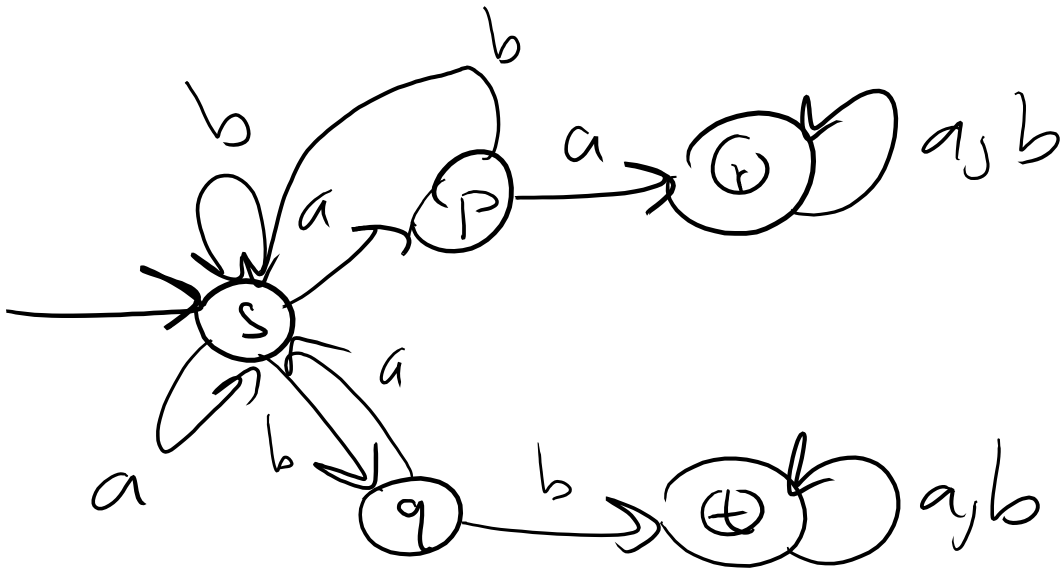


DFA that accepts all strings in $\{a,b\}^*$ that contain bb (2 consecutive bs).



Same as the previous

DFA that accepts all strings in $\{a, b\}^*$ that contain aa or bb.



Not a DFA!

It is a Nondeterministic finite Automaton.

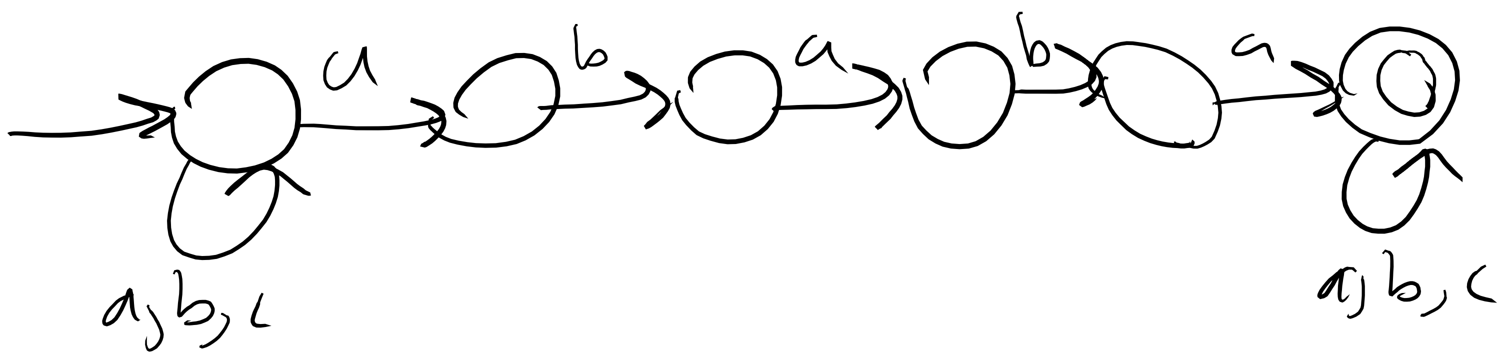
Nondeterministic Finite Automata (NFA)

- $M = (Q, \Sigma, \Delta, s, F)$
- Q is a finite set of states
- Σ is a finite input alphabet
- $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$ is the transition relation.

  ε is the empty string

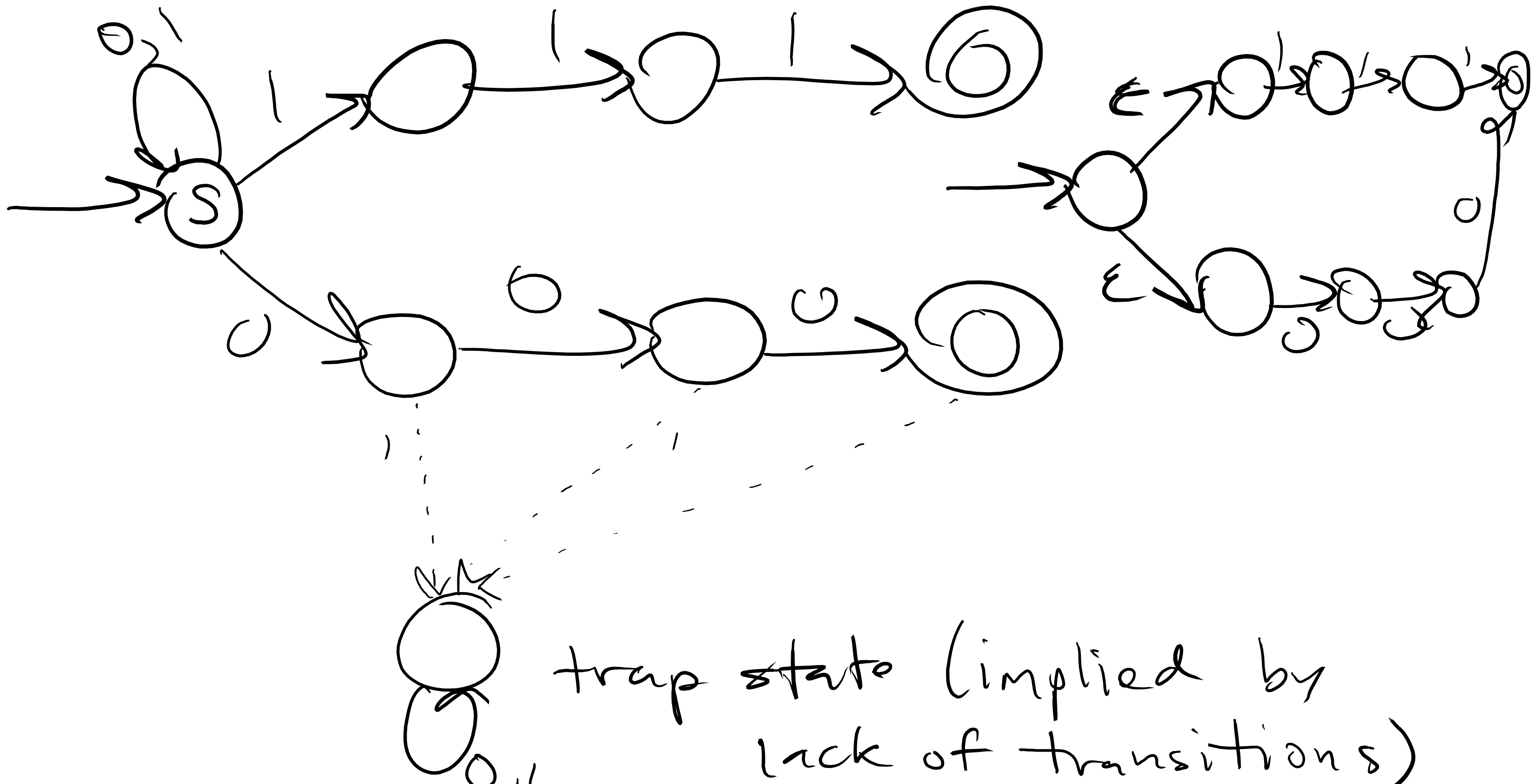
  Δ is a relation, not a function
- $s \in Q$ is the initial state
- $F \subseteq Q$ is the set of favorable states

Example: NFA that accepts all strings in $\{a, b, c\}^*$ that contain $ababa$.



Example: NFA that accepts all strings in $\{0,1\}^*$ that end with 111 or 000.

$L = \{ w \mid w \text{ ends with } 111 \text{ or } 000 \}$



Language Recognition by FSMs

- The language recognized or accepted by the machine M, denoted $L(M)$, is the set of all string that are accepted by M.

- Two automata are equivalent if they accept the same language.

Compare NFAs to DFAs

- Which is more computationally powerful?

they have the same power. they are equivalent.

- Which is easier to design?

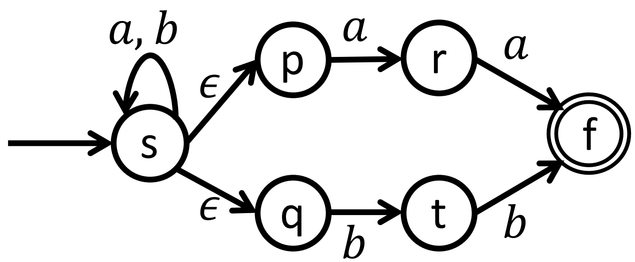
NFA

Theorem: For each NFA $A = (Q_N, \Sigma, \Delta, s_N, F_N)$, there exists a DFA $B = (Q_D, \Sigma, \delta, s_D F_D)$ equivalent to A.

Proof: by construction (sketch)

- Each state in B corresponds to a **set** of states in A
  - Subset construction problem
- For $T \subseteq Q_N$, $\epsilon - close(T)$ is the set of states reachable from a state in T using ϵ-jumps
- $s_D = \epsilon - close(s_N)$
- $Q_D = \{S \subseteq Q_N \mid S = \epsilon - close(S)\}$, ϵ-closed subsets of Q_N
  - Do lazy evaluation to find
- $F_D = \{S \mid (S \in Q_D) \wedge (S \cap F_N \neq \emptyset)\}$, states that contain at least 1 favorable state of A
- $\delta(s, a)$ for $a \in \Sigma$ and $s \in Q_D$ is computed as
  - Let s $= \{p_1, \ldots, p_k\}$
  - Let $\bigcup_{i=1}^{k} \Delta(p_i, a) = \{r_1, \ldots, r_m\}$
  - $\delta(s, a) = \epsilon - close(\{r_1, \ldots, r_m\})$

Example: Convert NFA to DFA



$S_D = \epsilon - close(s) = \{s, p, q\}$

	S	a	b
→ A	s,p,q	s,r,p,q	s,p,q,t
B	spqr	s,p,q,r,f	spqt
C	spqt	s,p,q,r	s,p,q,t,f
D	spqrf	s,p,q,r,f	s,p,q,t
E	spqtf	s,p,q,r	s,p,q,t,f