CS 301

Lecture 11 – Review

Exam topics

Broadly speaking: Everything about regular languages

- Alphabets, strings, languages
- DFAs, both the mathematical definition as a 5-tuple and as a diagram
- NFAs
- Regular expressions
- Conversions between DFAs, NFAs, and regular expressions
- Nonregular languages
- Closure properties of regular and nonregular languages
- Pumping lemma for regular languages

Types of exam questions

The questions from the exam fall into these types (the exam doesn't include every type of question)

- True/false questions with explanation
- Constructions
  - Construct a DFA/NFA/regular expression for a regular language
  - Convert an NFA to a DFA
  - Convert a DFA/NFA to a regular expression
  - Convert a regular expression to an NFA
- Prove that regular languages are closed under an operation
  - Perform a construction: Given a DFA/NFA/regex for some languages, build a new DFA/NFA/regex for the result of the operation (e.g., how we proved that regular languages are closed under PREFIX)
  - Write the operation in terms of other operations under which regular languages are closed (e.g., SUFFIX or intersection)
- Prove that a language isn't regular
  - Assume it is regular and apply the pumping lemma for regular languages and arrive at a contradiction
  - Assume it is regular and apply closure properties of regular languages to arrive at a contradiction

Some useful notation: δ^*

For a DFA $M = (Q, \Sigma, \delta, q_0, F)$, the transition function takes a state and a symbol and returns a state

$$\delta : Q \times \Sigma \to Q$$

We can extend this notation to a function that takes a state and a string and returns a state

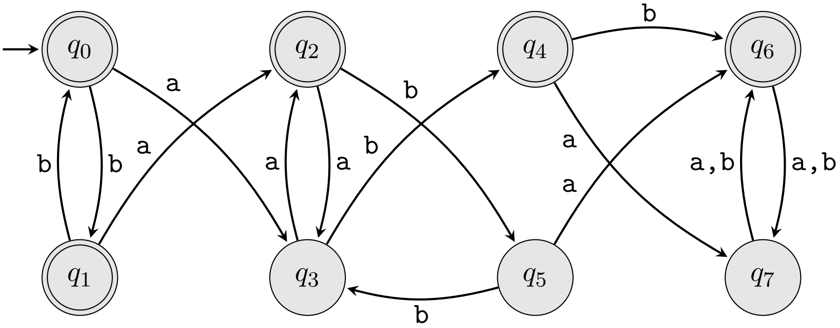
$$\delta^* : Q \times \Sigma^* \to Q$$

We can define δ^* recursively by

$$\begin{aligned} \delta^*(q, \varepsilon) &= q && \text{for all } q \in Q \\ \delta^*(q, tx) &= \delta^*\big(\delta(q, t),\ x\big) && \text{for } q \in Q,\ t \in \Sigma, \text{ and } x \in \Sigma^* \end{aligned}$$

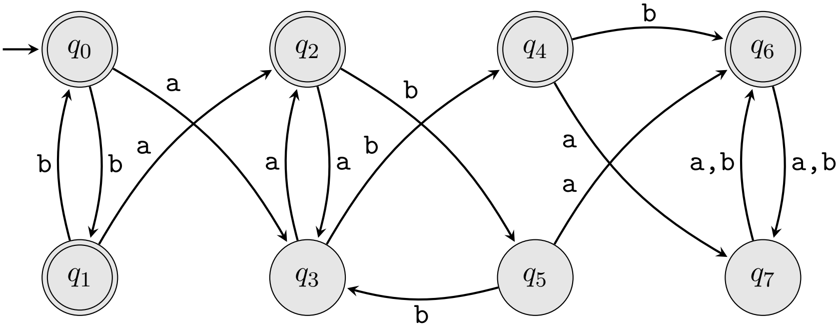
$\delta^*(q, w) = r$ means that starting in state q and moving from state to state according to δ on the symbols of w, the DFA ends in state r

Example



$\delta^*(q_3, \texttt{abaa}) = q_7$

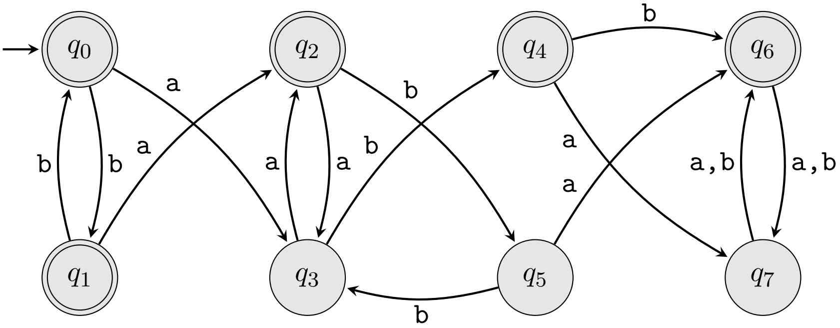
Example



$$\delta^*(q_3, \texttt{abaa}) = q_7$$

$$\delta^*(q_4, \varepsilon) =$$

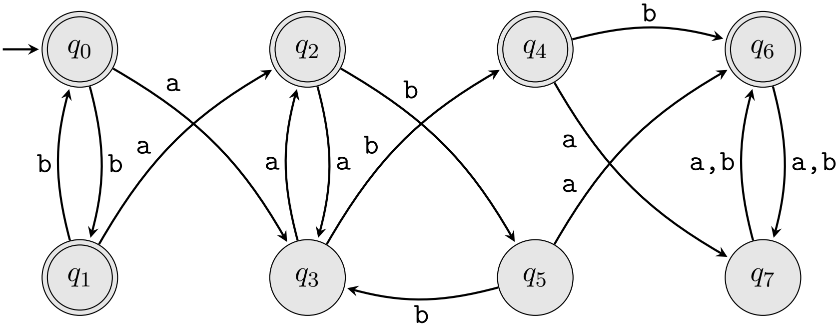
Example



$$\delta^*(q_3, \texttt{abaa}) = q_7$$

$$\delta^*(q_4, \varepsilon) = q_4$$

Example

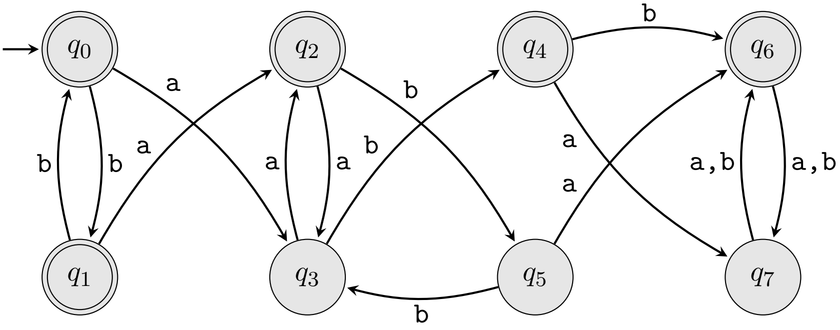


$$\delta^*(q_3, \texttt{abaa}) = q_7$$

$$\delta^*(q_4, \varepsilon) = q_4$$

$$\delta^*(q_0, \texttt{ba}) =$$

Example



$$\delta^*(q_3, \texttt{abaa}) = q_7$$

$$\delta^*(q_4, \varepsilon) = q_4$$

$$\delta^*(q_0, \texttt{ba}) = q_2$$

Utility of δ^*

Remember what it means for a DFA $M = (Q, \Sigma, \delta, q_0, F)$ to accept a string $w = w_1 w_2 \cdots w_n$:
There exist states $r_0, r_1, \ldots, r_n$ such that

1. $r_0 = q_0$
2. $r_i = \delta(r_{i-1}, w_i)$ for all $0 < i \leq n$
3. $r_n \in F$

Equivalently: M accepts w if $\delta^*(q_0, w) \in F$

Useful fact: If $x, y \in \Sigma^*$, then

$$\delta^*(q, xy) = \delta^*\big(\delta^*(q, x),\ y\big)$$

Use of δ^* in a proof

Recall that given a language A and a string u, both over alphabet Σ, we defined the left quotient of A by u as

$$u^{-1}A = \{x \mid x \in \Sigma^* \text{ and } ux \in A\}$$

Theorem

If A is a regular language and u is a string, then $u^{-1}A$ is regular.

Use of δ^* in a proof

Recall that given a language A and a string u, both over alphabet Σ, we defined the left quotient of A by u as

$$u^{-1}A = \{x \mid x \in \Sigma^* \text{ and } ux \in A\}$$

Theorem

If A is a regular language and u is a string, then $u^{-1}A$ is regular.

Proof.

Let $M = (Q, \Sigma, q_0, F)$ be a DFA that recognizes a language A.
Construct $M' = (Q, \Sigma, q_0', F)$ where $q_0' = \delta^*(q_0, u)$.

Use of δ^* in a proof

Recall that given a language A and a string u, both over alphabet Σ, we defined the left quotient of A by u as

$$u^{-1}A = \{x \mid x \in \Sigma^* \text{ and } ux \in A\}$$

Theorem

If A is a regular language and u is a string, then $u^{-1}A$ is regular.

Proof.

Let $M = (Q, \Sigma, q_0, F)$ be a DFA that recognizes a language A.
Construct $M' = (Q, \Sigma, q_0', F)$ where $q_0' = \delta^*(q_0, u)$.

M' accepts a string x if and only if $\delta^*(q_0', x) \in F$. But

$$\delta^*(q_0', x) = \delta^*\big(\delta^*(q_0, u),\, x\big) = \delta^*(q_0, ux)$$

Thus M' accepts x iff $\delta^*(q_0, ux) \in F$ iff M accepts ux.

Therefore $L(M') = u^{-1}A$ so $u^{-1}A$ is regular. □

Non sequitur