Chapter Three: Closure Properties for Regular Languages

Closure Properties

- Once we have defined languages formally, we can consider combinations and modifications of those languages:
  - unions, intersections, complements, and so on.
- Such combinations and modifications raise important questions.
  - For example, is the intersection of two regular languages also regular—capable of being recognized directly by some DFA?

Outline

- 3.1 Closed Under Complement
- 3.2 Closed Under Intersection
- 3.3 Closed Under Union
- 3.4 DFA Proofs Using Induction

Language Complement

- For any language *L* over an alphabet Σ, the *complement* of *L* is

$$\overline{L} = \left\{ x \in \Sigma^* \mid x \notin L \right\}$$

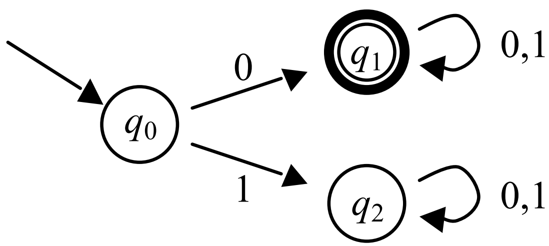
- Example:

$L = \left\{ 0x \mid x \in \{0,1\}^* \right\}$ = strings that start with 0

$\overline{L} = \left\{ 1x \mid x \in \{0,1\}^* \right\} \cup \{\varepsilon\}$ = strings that *don't* start with 0

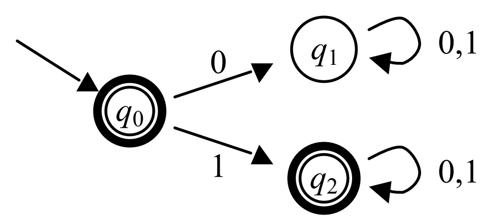
- Given a DFA for any language, it is easy to construct a DFA for its complement

Example



$$L = \left\{ 0x \mid x \in \{0,1\}^* \right\}$$

Reverse Accepting and Non-Accepting States!



$$\overline{L} = \left\{ 1x \mid x \in \{0,1\}^* \right\} \cup \{\varepsilon\}$$

Complementing a DFA

- All we did was to make the accepting states be non-accepting, and make the non-accepting states be accepting
- In terms of the 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$, all we did was to replace F with Q-F
- Using this construction, we have a proof that the complement of any regular language is another regular language

Theorem 3.1

The complement of any regular language is a regular language.

- Let L be any regular language
- By definition there must be some DFA $M = (Q, \Sigma, \delta, q_0, F)$ with $L(M) = L$
- Define a new DFA $M' = (Q, \Sigma, \delta, q_0, Q\text{-}F)$
- This has the same transition function δ as M, but for any string $x \in \Sigma^*$ it accepts x if and only if M rejects x
- Thus $L(M')$ is the complement of L
- Because there is a DFA for it, we conclude that the complement of L is regular

Closure Properties

- A shorter way of saying that theorem: the regular languages are *closed under complement*
- The complement operation cannot take us out of the class of regular languages
- Closure properties are useful shortcuts: they let you conclude a language is regular without actually constructing a DFA for it

Proofs using the Complement

- Show that the following language is regular:

$$L = \left\{ x \in \{a,b\}^* \mid x \text{ does not contain the string } abb \right\}$$

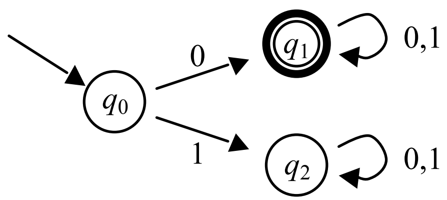
Outline

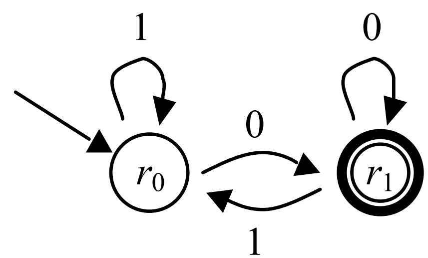
- 3.1 Closed Under Complement
- 3.2 Closed Under Intersection
- 3.3 Closed Under Union
- 3.4 DFA Proofs Using Induction

Language Intersection

- $L_1 \cap L_2 = \{x \mid x \in L_1 \text{ and } x \in L_2\}$
- Example:
  - $L_1 = \{0x \mid x \in \{0,1\}^*\}$ = strings that start with 0
  - $L_2 = \{x0 \mid x \in \{0,1\}^*\}$ = strings that end with 0
  - $L_1 \cap L_2 = \{x \in \{0,1\}^* \mid x \text{ starts and ends with } 0\}$
- Usually we will consider intersections of languages with the same alphabet, but it works either way
- Given two DFAs, it is possible to construct a DFA for the intersection of the two languages

Two DFAs





$L_1 = \{0x \mid x \in \{0,1\}^*\}$

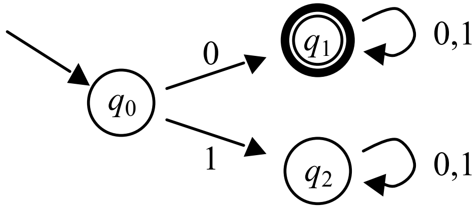
$L_2 = \{x0 \mid x \in \{0,1\}^*\}$

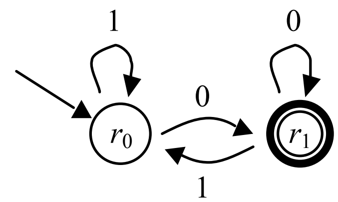
$M_1 = (Q, \Sigma, \delta_1, q_0, F_1)$

$M_2 = (R, \Sigma, \delta_2, r_0, F_2)$

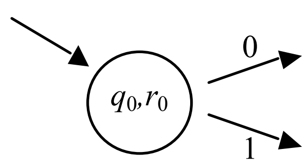
$L_1 = L(M_1)$

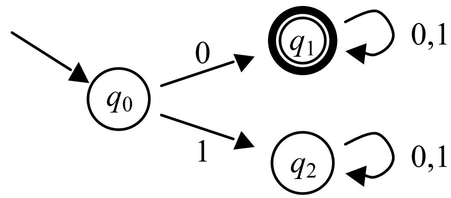
$L_2 = L(M_2)$

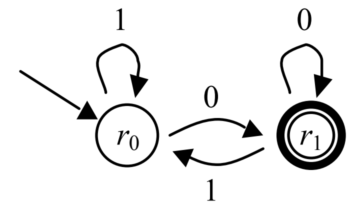




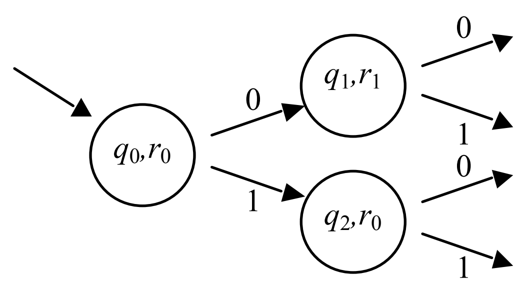
- We'll make a DFA that keeps track of the pair of states (q_i, r_j) the two original DFAs are in
- Initially, they are both in their start states:

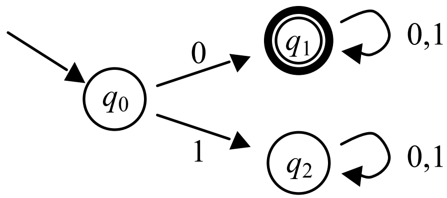


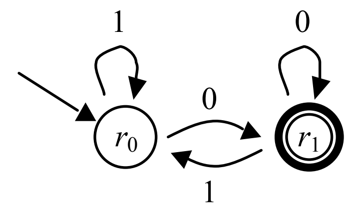




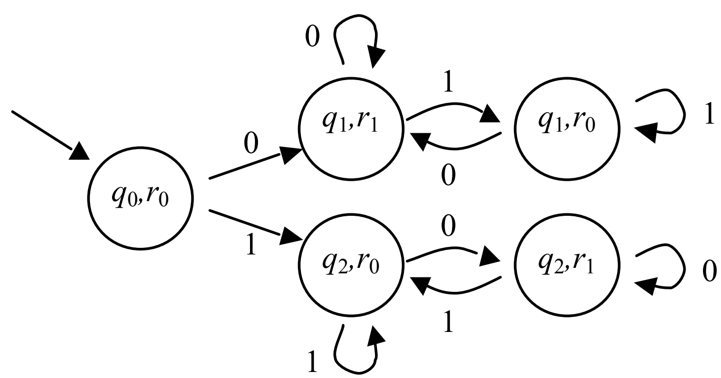
- Working from there, we keep track of the pair of states (q_i, r_j):

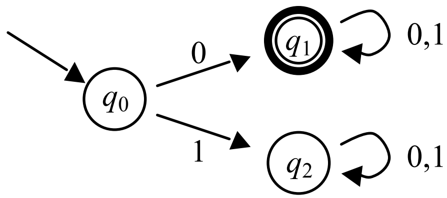


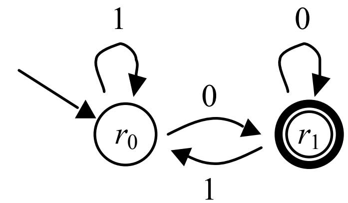




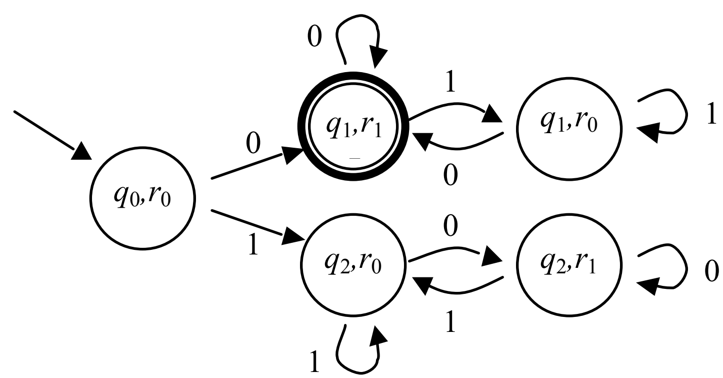
- Eventually state-pairs repeat; then we're almost done:







- For intersection, both original DFAs must accept:



Cartesian Product

- In that construction, the states of the new DFA are pairs of states from the two originals
- That is, the state set of the new DFA is the *Cartesian product* of the two original sets:

$$Q \times R = \{(q,r) \mid q \in Q \text{ and } r \in R\}$$

- The construct we just saw is called the *product construction*

Theorem 3.2

If L_1 and L_2 are any regular languages, $L_1 \cap L_2$ is also a regular language.

- Let L_1 and L_2 be any regular languages
- By definition there must be DFAs for them:
  - $M_1 = (Q, \Sigma, \delta_1, q_0, F_1)$ with $L(M_1) = L_1$
  - $M_2 = (R, \Sigma, \delta_2, r_0, F_2)$ with $L(M_2) = L_2$
- Define a new DFA $M_3 = (Q \times R, \Sigma, \delta, (q_0,r_0), F_1 \times F_2)$
- For δ, define it so that for all $q \in Q$, $r \in R$, and $a \in \Sigma$, we have $\delta((q,r),a) = (\delta_1(q,a), \delta_2(r,a))$
- M_3 accepts if and only if both M_1 and M_2 accept
- So $L(M_3) = L_1 \cap L_2$, so that intersection is regular

Notes

- Formal construction assumed that the alphabets were the same
  - It can easily be modified for differing alphabets
  - The alphabet for the new DFA would be $\Sigma_1 \cap \Sigma_2$
- Formal construction generated all pairs
  - When we did it by hand, we generated only those pairs actually reachable from the start pair
  - Makes no difference for the language accepted
  - The formal construction will just have a bunch of unreachable states in its set of states that have no impact on the language accepted by the machine.
- The new DFA runs both of the constituent DFAs simultaneously and accepts if and only if both DFAs accept.

Proofs using the Intersection

- Show that the following language is regular:

$$L = \left\{ x \in \{a,b\}^* \mid x \text{ contains both the strings } abb \text{ and } bba \right\}$$

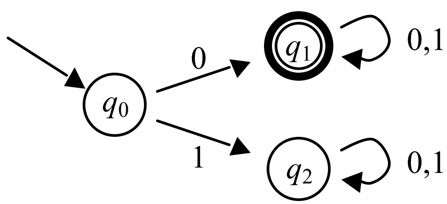
Outline

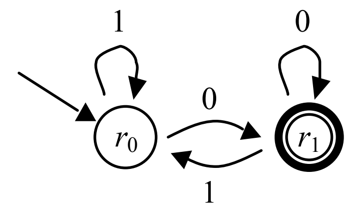
- 3.1 Closed Under Complement
- 3.2 Closed Under Intersection
- 3.3 Closed Under Union
- 3.4 DFA Proofs Using Induction

Language Union

- $L_1 \cup L_2 = \{x \mid x \in L_1 \text{ or } x \in L_2 \text{ (or both)}\}$
- Example:
  - $L_1 = \{0x \mid x \in \{0,1\}^*\}$ = strings that start with 0
  - $L_2 = \{x0 \mid x \in \{0,1\}^*\}$ = strings that end with 0
  - $L_1 \cup L_2 = \{x \in \{0,1\}^* \mid x \text{ starts with 0 or ends with 0 (or both)}\}$
- Usually we will consider unions of languages with the same alphabet, but it works either way

Two DFAs





$L_1 = \{0x \mid x \in \{0,1\}^*\}$

$L_2 = \{x0 \mid x \in \{0,1\}^*\}$

$M_1 = (Q, \Sigma, \delta_1, q_0, F_1)$

$M_2 = (R, \Sigma, \delta_2, r_0, F_2)$

$L_1 = L(M_1)$

$L_2 = L(M_2)$

Theorem 3.3

If L_1 and L_2 are any regular languages, $L_1 \cup L_2$ is also a regular language.

- Proof 1: using DeMorgan's laws
  - Because the regular languages are closed for intersection and complement, we know they must also be closed for union:

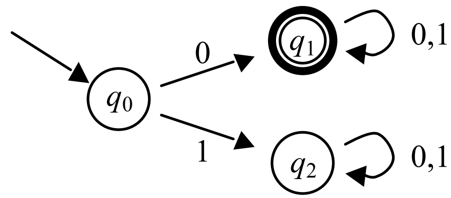
$$L_1 \cup L_2 = \overline{\overline{L_1} \cap \overline{L_2}}$$

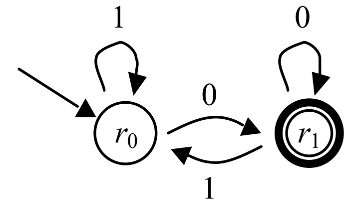
Theorem 3.3

If L_1 and L_2 are any regular languages, $L_1 \cup L_2$ is also a regular language.

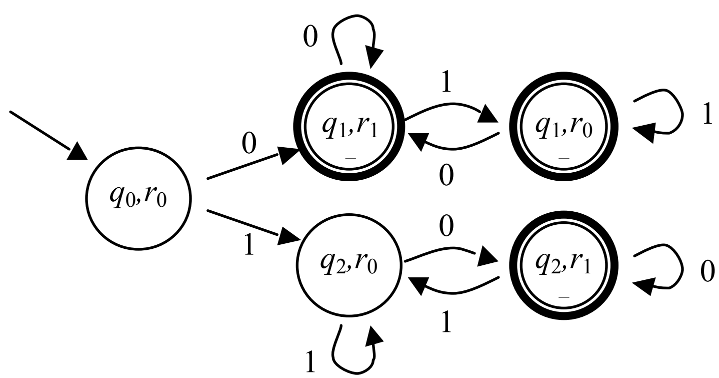
- Proof 2: by product construction
  - Same as for intersection, but with different accepting states
  - Accept where either (or both) of the original DFAs accept
  - Accepting state set is $(F_1 \times R) \cup (Q \times F_2)$
  - Define a new DFA:

    $M_3 = (Q \times R, \Sigma, \delta, (q_0, r_0), (F_1 \times R) \cup (Q \times F_2))$





- For union, at least one original DFA must accept:



Proofs using the Union

- Show that the following language is regular:

$$L = \left\{ x \in \{a,b\}^* \mid x \text{ contains either the string } abb \text{ or } bba \text{ or both} \right\}$$

Assignment

- Assignment #2 – see website