Closure Properties of Regular Languages

Union, Intersection, Difference, Concatenation, Kleene Closure, Reversal, Homomorphism, Inverse Homomorphism

Closure Under Union

- If L and M are regular languages, so is $L \cup M$.
- Proof: Let L and M be the languages of regular expressions R and S, respectively.
- Then R+S is a regular expression whose language is $L \cup M$.

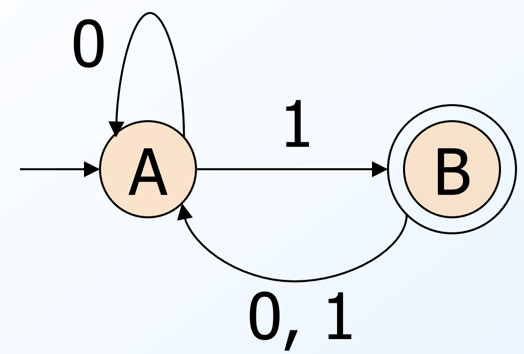
Closure Under Concatenation and Kleene Closure

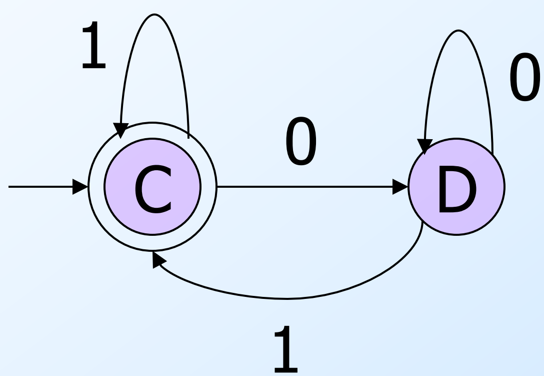
- Same idea:
  - RS is a regular expression whose language is LM.
  - R* is a regular expression whose language is L*.

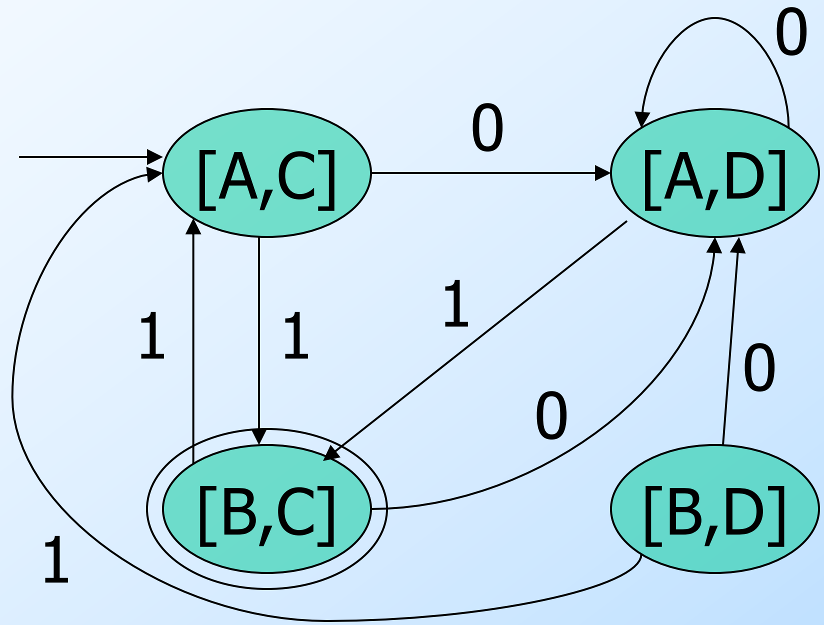
Closure Under Intersection

- If L and M are regular languages, then so is $L \cap M$.
- Proof: Let A and B be DFA's whose languages are L and M, respectively.
- Construct C, the product automaton of A and B.
- Make the final states of C be the pairs consisting of final states of both A and B.

Example: Product DFA for Intersection







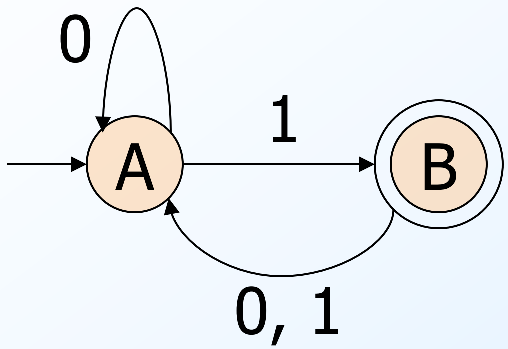
Example: Use of Closure Property

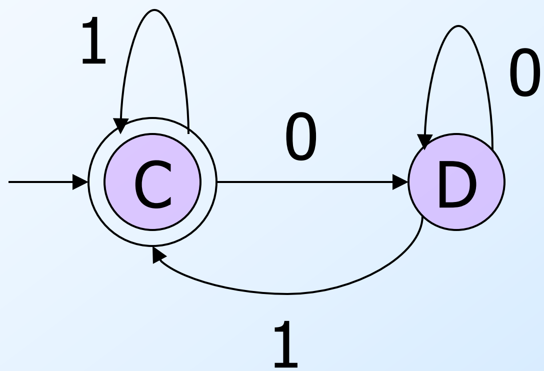
- We proved $L_1 = \{0^n1^n \mid n \geq 0\}$ is not a regular language.
- L_2 = the set of strings with an equal number of 0's and 1's isn't either, but that fact is trickier to prove.
- Regular languages are closed under $\cap$.
- If L_2 were regular, then $L_2 \cap L(\mathbf{0}^*\mathbf{1}^*) = L_1$ would be, but it isn't.

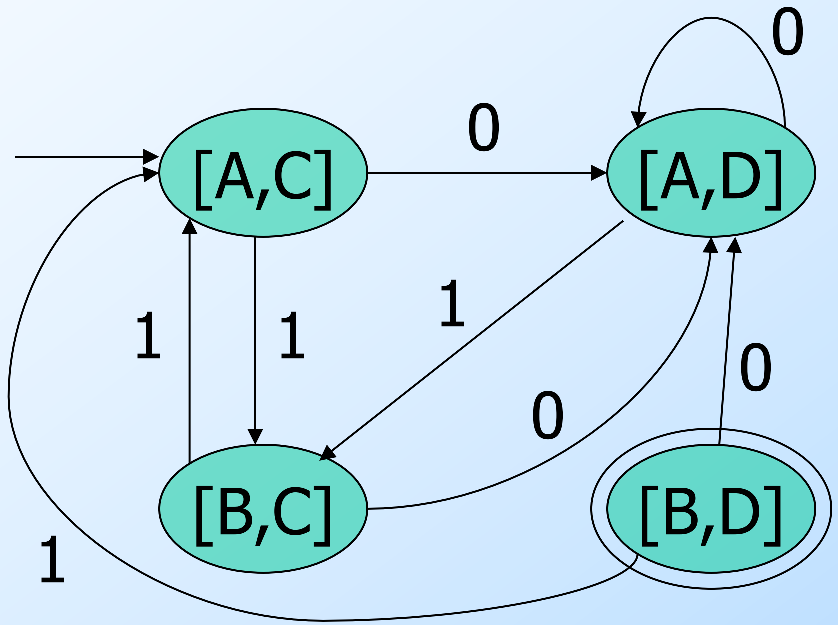
Closure Under Difference

- If L and M are regular languages, then so is $L - M$ = strings in L but not M.
- Proof: Let A and B be DFA's whose languages are L and M, respectively.
- Construct C, the product automaton of A and B.
- Final states of C are the pairs whose A-state is final but whose B-state is not.

Example: Product DFA for Difference







Closure Under Complementation

- The *complement* of a language L (with respect to an alphabet Σ such that Σ^* contains L) is $\Sigma^* - L$.
- Since Σ^* is surely regular, the complement of a regular language is always regular.

Closure Under Reversal

- Recall example of a DFA that accepted the binary strings that, as integers were divisible by 23.
- We said that the language of binary strings whose reversal was divisible by 23 was also regular, but the DFA construction was tricky.
- Here’s the “tricky” construction.

Closure Under Reversal – (2)

- Given language L, L^R is the set of strings whose reversal is in L.
- Example: L = {0, 01, 100};
  L^R = {0, 10, 001}.
- Proof: Let E be a regular expression for L. We show how to reverse E, to provide a regular expression E^R for L^R.

Reversal of a Regular Expression

- Basis: If E is a symbol a, ε, or ∅, then $E^R = E$.
- Induction: If E is
  - F+G, then $E^R = F^R + G^R$.
  - FG, then $E^R = G^R F^R$
  - F*, then $E^R = (F^R)^*$.

Example: Reversal of a RE

- Let E = $\mathbf{01}^* + \mathbf{10}^*$.
- $E^R = (\mathbf{01}^* + \mathbf{10}^*)^R = (\mathbf{01}^*)^R + (\mathbf{10}^*)^R$
- $= (\mathbf{1}^*)^R\mathbf{0}^R + (\mathbf{0}^*)^R\mathbf{1}^R$
- $= (\mathbf{1}^R)^*\mathbf{0} + (\mathbf{0}^R)^*\mathbf{1}$
- $= \mathbf{1}^*\mathbf{0} + \mathbf{0}^*\mathbf{1}$.

Homomorphisms

- A *homomorphism* on an alphabet is a function that gives a string for each symbol in that alphabet.
- Example: $h(0) = ab$; $h(1) = \epsilon$.
- Extend to strings by $h(a_1 \ldots a_n) = h(a_1) \ldots h(a_n)$.
- Example: $h(01010) = ababab$.

Closure Under Homomorphism

- If L is a regular language, and h is a homomorphism on its alphabet, then $h(L) = \{h(w) \mid w \text{ is in } L\}$ is also a regular language.
- Proof: Let E be a regular expression for L.
- Apply h to each symbol in E.
- Language of resulting RE is $h(L)$.

Example: Closure under Homomorphism

- Let $h(0) = ab$; $h(1) = \epsilon$.
- Let L be the language of regular expression $\mathbf{01}^* + \mathbf{10}^*$.
- Then h(L) is the language of regular expression $\mathbf{ab}\epsilon^* + \epsilon(\mathbf{ab})^*$.

Note: use parentheses to enforce the proper grouping.

Example – Continued

- $\mathbf{ab}\epsilon^*$ + $\epsilon(\mathbf{ab})^*$ can be simplified.
- $\epsilon^* = \epsilon$, so $\mathbf{ab}\epsilon^* = \mathbf{ab}\epsilon$.
- ϵ is the identity under concatenation.
  - That is, $\epsilon E = E\epsilon = E$ for any RE *E*.
- Thus, $\mathbf{ab}\epsilon + \epsilon(\mathbf{ab})^* = \mathbf{ab} + (\mathbf{ab})^*$.
- Finally, L(**ab**) is contained in L((**ab**)*), so a RE for h(L) is (**ab**)*.

Inverse Homomorphisms

- Let h be a homomorphism and L a language whose alphabet is the output language of h.
- $h^{-1}(L)$ = {w | h(w) is in L}.

Example: Inverse Homomorphism

- Let $h(0) = ab$; $h(1) = \epsilon$.
- Let $L = \{abab, baba\}$.
- $h^{-1}(L)$ = the language with two 0's and any number of 1's = $L(\mathbf{1^*01^*01^*})$.

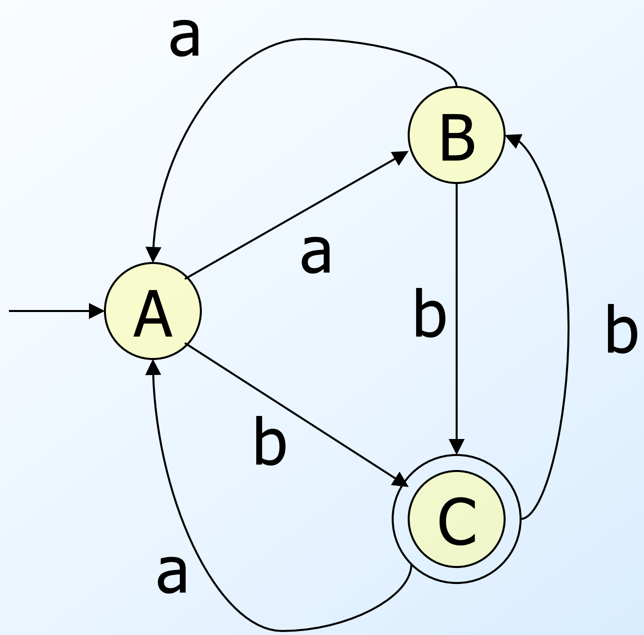
Closure Proof for Inverse Homomorphism

- Start with a DFA A for L.
- Construct a DFA B for $h^{-1}(L)$ with:
  - The same set of states.
  - The same start state.
  - The same final states.
  - Input alphabet = the symbols to which homomorphism h applies.

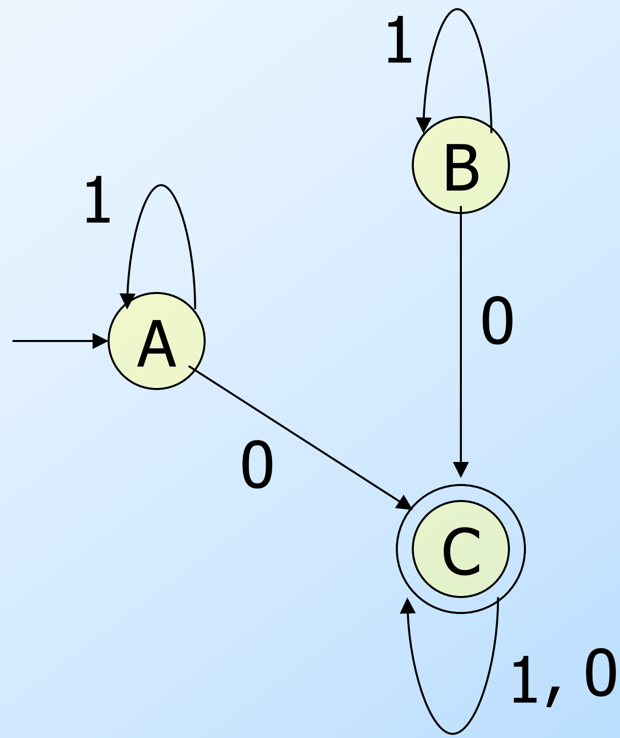
Proof – (2)

- The transitions for B are computed by applying h to an input symbol *a* and seeing where A would go on sequence of input symbols h(a).
- Formally, $\delta_B(q, a) = \delta_A(q, h(a))$.

Example: Inverse Homomorphism Construction



$h(0) = ab$
$h(1) = \epsilon$



Since
$h(1) = \epsilon$

Since
$h(0) = ab$

Proof – Inverse Homomorphism

- An induction on |w| (omitted) shows that $\delta_B(q_0, w) = \delta_A(q_0, h(w))$.
- Thus, B accepts w if and only if A accepts h(w).