Language Operations
Language Operations

- Operations that can be used to construct languages from other languages
- Since languages are sets, we can use set operations:
  - Union,
  - Intersection
  - Complement
  - Set difference
- Additional operations that strictly deal with strings
  - Concatenation
  - Star
- Example
  - $A = \{\text{good, bad}\}$
  - $B = \{\text{boy, girl}\}$
  - $A \cup B = \{\text{good, bad, boy, girl}\}$,
  - $A \circ B = \{\text{goodboy, goodgirl, badboy, badgirl}\}$, and
  - $A^* = \{\varepsilon, \text{good, bad, goodgood, goodbad, badgood, badbad, goodgoodgood, goodgoodbad, goodbadgood, goodbadbad, ...}\}$. 
Closure of Regular Languages (FA-recognizable)

- The set of FA-recognizable languages is closed under all six string operations.
  - If we start with regular languages and apply the operations, a new regular language is created.
  - May not work with previous finite automata but will for some finite automata

- Theorem 1:
  - FA-recognizable languages are closed under complement

- Proof:
  - Start with a language $L_1$ over alphabet $\Sigma$, recognized by some FA, $M_1$
  - Produce another FA, $M_2$, with $L(M_2) = \Sigma^* - L(M_1)$.
    - Just interchange accepting and non-accepting states
  - The new language is recognized by a finite automata and is considered FA-recognizable
Complement of Example 1

- Theorem 1: FA-recognizable languages are closed under complement
- Proof: Interchange accepting and non-accepting states

Example: FA for \{ w \mid w \text{ does not contain 111} \}

- Start with FA for \{ w \mid w \text{ contains 111} \}:

- Only accepted strings with 111 substring
- Convert to complement language
Complement of Example 1

- Example: FA for \( \{ w \mid w \text{ does not contain } 111 \} \)
  - Interchange accepting and non-accepting states

- States a, b, and c become accept states
- State d becomes a non-accept state
- Only way to reach d is to have a string with a 111 substring.
  - New FA only recognizes strings that do not have a 111 substring
Closure under Intersection

- **Theorem 2**: FA-Recognizable languages are closed under intersection

- **Proof**
  - Start with FAs $M_1$ and $M_2$ for the same alphabet $\Sigma$
  - Get another FA, $M_3$, with $L(M_3) = L(M_1) \cap L(M_2)$

- **Reasoning**
  - Run $M_1$ and $M_2$ “in parallel” on the same input
    - If both reach accepting states, accept

- **Example**
  - $L(M_1)$: Contains substring 01
  - $L(M_2)$: Contains an odd number of ones
  - $L(M_3)$: Contains 01 and has an odd number of 1s
**Closure under Intersection**

- Run both “in parallel”
  - Only accept string if both accept

- Symbols combine to become new states
  - $\Sigma_1 = \{a,b,c\}$
  - $\Sigma_2 = \{d,e\}$
  - $\Sigma_3 = \{ad,ae,bd,be,cd,ce\}$

**Example:**

- $M_1$: Substring 01
- $M_2$: Odd number of 1s
- $M_3$:
Closure under Intersection

- New Formal Definition
  - $M_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, F_1)$
  - $M_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, F_2)$

Define $M_3 = (Q_3, \Sigma_3, \delta_3, q_{03}, F_3)$, where
  - $Q_3 = Q_1 \times Q_2$
    - Cartesian Product, $\{ (q_1, q_2) | q_1 \in Q_1 \text{ and } q_2 \in Q_2 \}$
  - $\Sigma_3 = \{0,1\}$
  - $\delta_3((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$
  - $q_{03} = (q_{01}, q_{02})$
  - $F_3 = F_1 \times F_2 = \{ (q_1, q_2) | q_1 \in F_1 \text{ and } q_2 \in F_2 \}$
Theorem 3: FA-Recognizable languages are closed under union

Proof

Similar to intersection
Start with FAs $M_1$ and $M_2$ for the same alphabet $\Sigma$
Get another FA, $M_3$, with $L(M_3) = L(M_1) \cup L(M_2)$

Reasoning
Run $M_1$ and $M_2$ “in parallel” on the same input
If either reach accepting states, accept

Example
$L(M_1)$: Contains substring 01
$L(M_2)$: Contains an odd number of ones
$L(M_3)$: Contains 01 or has an odd number of 1s
Closure under Union

- Run both “in parallel”
- Symbols combine to become new states
  - $\Sigma_1 = \{a,b,c\}$
  - $\Sigma_2 = \{d,e\}$
  - $\Sigma_3 = \{ad,ae,bd,be,cd,ce\}$
- New states = accept if ordered pair contains old accepting state

**Example:**

$M_1$: Substring 01

$M_2$: Odd number of 1s

$M_3$: 1
Closure under Union

- **New Formal Definition**
  - $M_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, F_1)$
  - $M_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, F_2)$

- **Define $M_3 = (Q_3, \Sigma_3, \delta_3, q_{03}, F_3)$, where**
  - $Q_3 = Q_1 \times Q_2$
    - Cartesian Product, $\{ (q_1, q_2) : q_1 \in Q_1 \text{ and } q_2 \in Q_2 \}$
  - $\Sigma_3 = \{0,1\}$
  - $\delta_3((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$
  - $q_{03} = (q_{01}, q_{02})$
  - $F_3 = \{ (q_1, q_2) : q_1 \in F_1 \text{ or } q_2 \in F_2 \}$
Closure under Set Difference

- **Theorem 4**
  - FA-Recognizable languages are closed under set difference

- **Proof**
  - Similar proof to those for union and intersection
    - Accept if $L_1$ accepts and $L_2$ does not

- **Alternatively**
  - Since $L_1 - L_2$ is the same as $L_1 \cap (L_2)^c$, just apply Theorems 1 and 2


## Closure under Concatenation

- **Theorem 5**: FA-Recognizable Languages are Closed under concatenation

- **Proof**
  - Start with FAs $M_1$ and $M_2$ for the same alphabet $\Sigma$
  - Get another FA, $M_3$, with
    - $L(M_3) = L(M_1) \circ L(M_2) = \{ x_1 x_2 \mid x_1 \in L(M_1) \text{ and } x_2 \in L(M_2) \}$

- **Reasoning**
  - Attach accepting states of $M_1$ somehow to the start state of $M_2$
  - Don't know when string is done with $M_1$ portion of $M_3$
    - Careful as string may go through accepting states of $M_1$ several times
Closure under Concatenation

- Example
  - $\Sigma = \{0,1\}$, $L_1 = \Sigma^*$, $L_2 = \{0\}\{0\}^*$ (just zeros, at least one)
  - $L_1L_2 = \text{Strings that end with a block of at least one } 0$

- $M_1$: 

- $M_2$: 

- How to combine?
  - Need to “guess” when to shift to $M_2$
  - Leads to our next model, Nondeterministic Finite Automata
  - FAs that can guess

- Closure under star operation is an extension of this.