Deterministic Finite Automata (DFAs)

Lecture 3
Tuesday, January 22, 2019
Part I

DFA Introduction
DFAs also called Finite State Machines (FSMs)

- The “simplest” model for computers?
- State machines that are common in practice.
  - Vending machines
  - Elevators
  - Digital watches
  - Simple network protocols
- Programs with fixed memory
A simple program

Program to check if a given input string $w$ has odd length

```
int $n = 0$
While input is not finished
    read next character $c$
    $n ← n + 1$
endWhile
If ($n$ is odd) output YES
Else output NO
```

```
bit $x = 0$
While input is not finished
    read next character $c$
    $x ← \text{flip}(x)$
endWhile
If ($x = 1$) output YES
Else output NO
```
A simple program

Program to check if a given input string \( w \) has odd length

```
int \( n = 0 \)
While input is not finished
    read next character \( c \)
    \( n \leftarrow n + 1 \)
endWhile
If (\( n \) is odd) output YES
Else output NO
```

```
bit \( x = 0 \)
While input is not finished
    read next character \( c \)
    \( x \leftarrow \text{flip}(x) \)
endWhile
If (\( x = 1 \)) output YES
Else output NO
```
Another view

- Machine has input written on a *read-only* tape
- Start in specified start state
- Start at left, scan symbol, change state and move right
- Circled states are *accepting*
- Machine *accepts* input string if it is in an accepting state after scanning the last symbol.
Directed graph with nodes representing states and edge/arcs representing transitions labeled by symbols in $\Sigma$

For each state (vertex) $q$ and symbol $a \in \Sigma$ there is exactly one outgoing edge labeled by $a$

Initial/start state has a pointer (or labeled as $s$, $q_0$ or “start”)

Some states with double circles labeled as accepting/final states
**Graphical Representation**

A deterministic finite automaton (DFA) is defined as: $M = (Q, \mathcal{A}, s, A)$ where
- $Q$ is a finite set whose elements are called states,
- $\mathcal{A}$ is a finite set called the input alphabet,
- $Q \times \mathcal{A} \rightarrow Q$ is the transition function,
- $s \in Q$ is the start state,
- $A \subseteq Q$ is the set of accepting/final states.

**Problem 4.** Prove that for any state $p$, and string $w \in \mathcal{A}^*$, there is a unique state $q$ such that $p \xrightarrow{w} M q$.

**Notation.**
- $\xrightarrow{w} M$ denotes the transition from state $p$ to state $q$ by reading string $w$.

**Definition 6.** Consider a DFA $M = (Q, \mathcal{A}, s, A)$.
- $M$ accepts string $w \in \mathcal{A}^*$ if $M(s, w) \in A$.
- The language accepted/recognized by a DFA $M$ is $L(M) = \{ w \in \mathcal{A}^* | M \text{ accepts } w \}$.
- A set $L \subseteq \mathcal{A}^*$ is said to be accepted/recognized by $M$ if $L = L(M)$.

**Problem 5.**
1. Which of the following is true?
   - $B \xrightarrow{!} M A$
   - $A \xrightarrow{01} M D$
   - $D \xrightarrow{111} M C$
   - $A \xrightarrow{101} M B$
2. What is the following?
   - $\xrightarrow{2} M (A, 1011) =$
   - $\xrightarrow{2} M (B, 010) =$
   - $\xrightarrow{2} M (C, 100) =$

**Figure 1: DFA $M$ for problem 5**

- Where does $001$ lead? $10010$?
- Which strings end up in accepting state?
- Can you prove it?
- Every string $w$ has a unique walk that it follows from a given state $q$ by reading one letter of $w$ from left to right.
Definition 4. A deterministic finite automaton (DFA) is 
\[ M = (Q, \mathcal{I}, \delta, s, A) \]
where
- \( Q \) is a finite set whose elements are called states,
- \( \mathcal{I} \) is a finite set called the input alphabet,
- \( \delta : Q \times \mathcal{I} \rightarrow Q \) is the transition function,
- \( s \in Q \) is the start state,
- \( A \subseteq Q \) is the set of accepting/final states.

Definition 5. For a DFA \( M = (Q, \mathcal{I}, \delta, s, A) \), string \( w = w_1 w_2 \ldots w_k \), where for each \( i \), \( w_i \in \mathcal{I} \), and states \( p, q \in Q \), \( w \rightarrow M q \) if there is a sequence of states \( r_0, r_1, \ldots, r_k \) such that
(a) \( r_0 = p \), (b) for each \( i \), \( (r_i, w_{i+1}) = r_{i+1} \), and (c) \( r_k = q \).

Problem 4. Prove that for any state \( p \), and string \( w \in \mathcal{I}^* \), there is a unique state \( q \) such that \( p \rightarrow M q \).

Notation. \( \rightarrow^*_M (p, w) = q \) where \( p \rightarrow M q \).

Definition 6. Consider a DFA \( M = (Q, \mathcal{I}, \delta, s, A) \).
- \( M \) accepts string \( w \in \mathcal{I}^* \) if \( M \) accepts \( w \) as defined above.
- The language accepted/recognized by a DFA \( M \) is \( L(M) = \{ w \in \mathcal{I}^* \mid M \text{ accepts } w \} \).
- A set \( L \subseteq \mathcal{I}^* \) is said to be accepted/recognized by \( M \) if \( L = L(M) \).

Problem 5. 1. Which of the following is true?
- \( B \rightarrow M B \)
- \( A \rightarrow M D \)
- \( D \rightarrow M C \)
- \( A \rightarrow M B \)

2. What is the following?
- \( \rightarrow^*_M (A, 1011) \)
- \( \rightarrow^*_M (B, 010) \)
- \( \rightarrow^*_M (C, 100) \)

3. What is \( L(M) \)?
4. What is the language recognized if we change the initial state to \( B \)?
5. What is the language recognized if we change the set of final states to be \( \{B\} \) (with initial state \( A \))?
Graphical Representation

- Where does 001 lead? 10010?
- Which strings end up in accepting state?
- Can you prove it?
- Every string $w$ has a unique walk that it follows from a given state $q$ by reading one letter of $w$ from left to right.
A deterministic finite automaton (DFA) is 
\[ M = (Q, \mathcal{A}, \delta, s, A) \]
where
- \( Q \) is a finite set whose elements are called states,
- \( \mathcal{A} \) is a finite set called the input alphabet,
- \( \delta : Q \times \mathcal{A} \rightarrow Q \) is the transition function,
- \( s \in Q \) is the start state,
- \( A \subseteq Q \) is the set of accepting/final states.

Definition 5. For a DFA \( M = (Q, \mathcal{A}, \delta, s, A) \), string \( w = w_1w_2 \cdots w_k \), where for each \( i \), \( w_i \in \mathcal{A} \), and states \( p, q \in Q \), \( w \in M \) if there is a sequence of states \( r_0, r_1, \ldots, r_k \) such that
  1. \( r_0 = p \),
  2. for each \( i \), \((r_i, w_{i+1}) = r_{i+1} \),
  3. \( r_k = q \).

Problem 4. Prove that for any state \( p \), and string \( w \in \mathcal{A}^* \), there is a unique state \( q \) such that \( p \xrightarrow{w} M q \).

Notation. \( \xrightarrow{w} M q \) where \( p \xrightarrow{w} M q \).

Definition 6. Consider a DFA \( M = (Q, \mathcal{A}, \delta, s, A) \).
- \( M \) accepts string \( w \in \mathcal{A}^* \) if \( \xrightarrow{w} M s \in A \).
- The language accepted/recognized by a DFA \( M \) is \( L(M) = \{ w \in \mathcal{A}^* \mid M \text{ accepts } w \} \).
- A set \( L \subseteq \mathcal{A}^* \) is said to be accepted/recognized by \( M \) if \( L = L(M) \).

Problem 5.

1. Which of the following is true?
   - \( B \xrightarrow{A} M B \)
   - \( A \xrightarrow{01} M D \)
   - \( D \xrightarrow{111} M C \)
   - \( A \xrightarrow{101} M 2B \)

2. What is the following?
   - \( \xrightarrow{} M 2(\text{A}, 1011) = \)
   - \( \xrightarrow{} M 2(\text{B}, 010) = \)
   - \( \xrightarrow{} M 2(\text{C}, 100) = \)

3. What is \( L(M) \)?

4. What is the language recognized if we change the initial state to \( B \)?

5. What is the language recognized if we change the set of final states to be \( \{B\} \) (with initial state \( A \))?
**Definition**

A **DFA** \( M \) accepts a string \( w \) iff the unique walk starting at the start state and spelling out \( w \) ends in an accepting state.

**Definition**

The language accepted (or recognized) by a **DFA** \( M \) is denoted by \( L(M) \) and defined as: \( L(M) = \{ w \mid M \text{ accepts } w \} \).
Definition 4. A deterministic finite automaton (DFA) is 
\[ M = (Q, \mathcal{A}, \delta, s, A) \]
where
- \( Q \) is a finite set whose elements are called states,
- \( \mathcal{A} \) is a finite set called the input alphabet,
- \( \delta : Q \times \mathcal{A} \rightarrow Q \) is the transition function,
- \( s \in Q \) is the start state,
- \( A \subseteq Q \) is the set of accepting/final states.

Definition 5. For a DFA \( M = (Q, \mathcal{A}, \delta, s, A) \), string \( w = w_1w_2 \cdots w_k \), where for each \( i \), \( w_i \in \mathcal{A} \), and states \( p, q \in Q \), \( M \) accepts \( w \) if there is a sequence of states \( r_0, r_1, \ldots, r_k \) such that
- \( r_0 = p \),
- for each \( i \), \( (r_i, w_{i+1}) = r_{i+1} \), and
- \( r_k = q \).

Problem 4. Prove that for any state \( p \), and string \( w \in \mathcal{A}^* \), there is a unique state \( q \) such that \( p \xrightarrow{w} M q \).

Notation. \( \xrightarrow{w} M \) \[ (p, w) = q \] where \( p \xrightarrow{w} M q \).

Definition 6. Consider a DFA \( M = (Q, \mathcal{A}, \delta, s, A) \).
- \( M \) accepts the string \( w \) iff the unique walk starting at the start state and spelling out \( w \) ends in an accepting state.
- The language accepted (or recognized) by a DFA \( M \) is denoted by \( L(M) \) and defined as:
  \[ L(M) = \{ w \mid M \text{ accepts } w \} \].
“$M$ accepts language $L$” does not mean simply that that $M$ accepts each string in $L$.

It means that $M$ accepts each string in $L$ and no others. Equivalently $M$ accepts each string in $L$ and does not accept/rejects strings in $\Sigma^* \setminus L$.

$M$ “recognizes” $L$ is a better term but “accepts” is widely accepted (and recognized) (joke attributed to Lenny Pitt)
“$M$ accepts language $L$” does not mean simply that that $M$ accepts each string in $L$.

It means that $M$ accepts each string in $L$ and no others. Equivalently $M$ accepts each string in $L$ and does not accept/rejects strings in $\Sigma^* \setminus L$.

$M$ “recognizes” $L$ is a better term but “accepts” is widely accepted (and recognized) (joke attributed to Lenny Pitt)
A deterministic finite automata (DFA) \( M = (Q, \Sigma, \delta, s, A) \) is a five tuple where

- \( Q \) is a finite set whose elements are called states,
- \( \Sigma \) is a finite set called the input alphabet,
- \( \delta : Q \times \Sigma \rightarrow Q \) is the transition function,
- \( s \in Q \) is the start state,
- \( A \subseteq Q \) is the set of accepting/final states.

Common alternate notation: \( q_0 \) for start state, \( F \) for final states.
Formal Tuple Notation

Definition

A deterministic finite automata (DFA) $M = (Q, \Sigma, \delta, s, A)$ is a five tuple where

- $Q$ is a finite set whose elements are called states,
- $\Sigma$ is a finite set called the input alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function,
- $s \in Q$ is the start state,
- $A \subseteq Q$ is the set of accepting/final states.

Common alternate notation: $q_0$ for start state, $F$ for final states.
A deterministic finite automata (DFA) \( M = (Q, \Sigma, \delta, s, A) \) is a five tuple where

- \( Q \) is a finite set whose elements are called states,
- \( \Sigma \) is a finite set called the input alphabet,
- \( \delta : Q \times \Sigma \rightarrow Q \) is the transition function,
- \( s \in Q \) is the start state,
- \( A \subseteq Q \) is the set of accepting/final states.

Common alternate notation: \( q_0 \) for start state, \( F \) for final states.
A deterministic finite automata (DFA) $M = (Q, \Sigma, \delta, s, A)$ is a five tuple where

- $Q$ is a finite set whose elements are called states,
- $\Sigma$ is a finite set called the input alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function,
- $s \in Q$ is the start state,
- $A \subseteq Q$ is the set of accepting/final states.

Common alternate notation: $q_0$ for start state, $F$ for final states.
Definition

A deterministic finite automata (DFA) \( M = (Q, \Sigma, \delta, s, A) \) is a five tuple where

- \( Q \) is a finite set whose elements are called states,
- \( \Sigma \) is a finite set called the input alphabet,
- \( \delta : Q \times \Sigma \to Q \) is the transition function,
- \( s \in Q \) is the start state,
- \( A \subseteq Q \) is the set of accepting/final states.

Common alternate notation: \( q_0 \) for start state, \( F \) for final states.
Formal Tuple Notation

Definition

A deterministic finite automata (DFA) \( M = (Q, \Sigma, \delta, s, A) \) is a five tuple where

- \( Q \) is a finite set whose elements are called states,
- \( \Sigma \) is a finite set called the input alphabet,
- \( \delta : Q \times \Sigma \rightarrow Q \) is the transition function,
- \( s \in Q \) is the start state,
- \( A \subseteq Q \) is the set of accepting/final states.

Common alternate notation: \( q_0 \) for start state, \( F \) for final states.
A deterministic finite automata (DFA) \( M = (Q, \Sigma, \delta, s, A) \) is a five tuple where

- \( Q \) is a finite set whose elements are called states,
- \( \Sigma \) is a finite set called the input alphabet,
- \( \delta : Q \times \Sigma \rightarrow Q \) is the transition function,
- \( s \in Q \) is the start state,
- \( A \subseteq Q \) is the set of accepting/final states.

Common alternate notation: \( q_0 \) for start state, \( F \) for final states.
A deterministic finite automata (DFA) $M = (Q, \Sigma, \delta, s, A)$ is a five tuple where

- $Q$ is a finite set whose elements are called states,
- $\Sigma$ is a finite set called the input alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function,
- $s \in Q$ is the start state,
- $A \subseteq Q$ is the set of accepting/final states.

Common alternate notation: $q_0$ for start state, $F$ for final states.
DFA Notation

\[ M = \left( \begin{array}{c} Q \text{ set of all states} \\ \Sigma \text{ alphabet} \\ \delta \text{ transition func} \\ s \text{ start state} \\ A \text{ set of all accept states} \end{array} \right) \]
Example

- **Q** = \{q_0, q_1, q_1, q_3\}
- **Σ** = \{0, 1\}
- **δ**
- **s** = q_0
- **A** = \{q_0\}
A deterministic finite automaton (DFA) is \( M = (Q, \Sigma, \delta, s, A) \) where
- \( Q \) is a finite set whose elements are called states,
- \( \Sigma \) is a finite set called the input alphabet,
- \( \delta : Q \times \Sigma \to Q \) is the transition function,
- \( s \in Q \) is the start state,
- \( A \subseteq Q \) is the set of accepting/final states.

### Definition 5
For a DFA \( M = (Q, \Sigma, \delta, s, A) \), string \( w = w_1w_2\cdots w_k \), where for each \( i \), \( w_i \in \Sigma \), and states \( p, q \in Q \), we say \( p \xrightarrow{w} M q \) if there is a sequence of states \( r_0, r_1, \ldots, r_k \) such that:
1. \( r_0 = p \),
2. for each \( i \), \( (r_i, w_{i+1}) = r_{i+1} \), and
3. \( r_k = q \).

### Problem 4
Prove that for any state \( p \), and string \( w \in \Sigma^* \), there is a unique state \( q \) such that \( p \xrightarrow{w} M q \).

### Notation
\( \xrightarrow{w} M \) \( (p, w) = q \) where \( p \xrightarrow{w} M q \).

### Definition 6
Consider a DFA \( M = (Q, \Sigma, \delta, s, A) \).
- \( M \) accepts string \( w \in \Sigma^* \) if \( M \) accepts \( \vdash w \).
- The language accepted/recognized by a DFA \( M \) is \( L(M) = \{ w \in \Sigma^* | M \text{ accepts } w \} \).
- A set \( L \subseteq \Sigma^* \) is said to be accepted/recognized by \( M \) if \( L = L(M) \).

### Problem 5
1. Which of the following is true?
   - \( B \xrightarrow{!} M B \)
   - \( A \xrightarrow{01} M D \)
   - \( D \xrightarrow{111} M C \)
   - \( A \xrightarrow{101} M 2B \)
2. What is the following?
   - \( \xrightarrow{2} M (A, 1011) = \)
   - \( \xrightarrow{2} M (B, 010) = \)
   - \( \xrightarrow{2} M (C, 100) = \)

---

Example

![DFA Diagram](image)

- \( Q = \{ q_0, q_1, q_3 \} \)
- \( \Sigma = \{ 0, 1 \} \)
- \( \delta \)
- \( s = q_0 \)
- \( A = \{ q_0 \} \)
Example

A deterministic finite automaton (DFA) is $M = (Q, \Sigma, \delta, s, A)$ where
- $Q$ is a finite set whose elements are called states,
- $\Sigma$ is a finite set called the input alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function,
- $s \in Q$ is the start state,
- $A \subseteq Q$ is the set of accepting/final states.

Problem 4. Prove that for any state $p$ and string $w \in \Sigma^*$, there is a unique state $q$ such that $p \xrightarrow{w} M q$.

Notation. $\xrightarrow{w} M q$ where $p \xrightarrow{w} M q$.

Definition 6. Consider a DFA $M = (Q, \Sigma, \delta, s, A)$.
- $M$ accepts string $w \in \Sigma^*$ if $M(s, w) \in A$.
- The language accepted/recognized by a DFA $M$ is $L(M) = \{ w \in \Sigma^* | M \text{ accepts } w \}$.
- A set $L \subseteq \Sigma^*$ is said to be accepted/recognized by $M$ if $L = L(M)$.

Problem 5.
1. Which of the following is true?
   - $B \xrightarrow{!} M B$
   - $A \xrightarrow{01} M D$
   - $D \xrightarrow{111} M C$
   - $A \xrightarrow{101} M 2 B$

2. What is the following?
   - $\xrightarrow{2} M(A, 1011)$
   - $\xrightarrow{2} M(B, 010)$
   - $\xrightarrow{2} M(C, 100)$

Figure 1: DFA $M$ for problem 5

- $Q = \{ q_0, q_1, q_1, q_3 \}$
- $\Sigma = \{ 0, 1 \}$
- $\delta$
- $s = q_0$
- $A = \{ q_0 \}$
Example

Definition 4. A deterministic finite automaton (DFA) is $M = (Q, \Sigma, \delta, s, A)$ where

- $Q$ is a finite set whose elements are called states,
- $\Sigma$ is a finite set called the input alphabet,
- $\delta : Q \times \Sigma \to Q$ is the transition function,
- $s \in Q$ is the start state,
- $A \subseteq Q$ is the set of accepting/final states.

Definition 5. For a DFA $M = (Q, \Sigma, \delta, s, A)$, string $w = w_1w_2 \cdots w_k$, where for each $i$, $w_i \in \Sigma$, and states $p, q \in Q$, $w$ is a word if there is a sequence of states $r_0, r_1, \ldots, r_k$ such that

- $r_0 = p$,
- for each $i$, $(r_i, w_{i+1}) = r_{i+1}$,
- and $r_k = q$.

Problem 4. Prove that for any state $p$, and string $w \in \Sigma^*$, there is a unique state $q$ such that $p \xrightarrow{w} M q$.

Notation. $\xrightarrow{w} M$ is defined as $\exists q \in Q : p \xrightarrow{w} M q$.

Definition 6. Consider a DFA $M = (Q, \Sigma, \delta, s, A)$.

- $M$ accepts string $w \in \Sigma^*$ if $M$ enters an accepting state.
- The language accepted/recognized by a DFA $M$ is $L(M) = \{ w \in \Sigma^* : M \text{ accepts } w \}$.
- A set $L \subseteq \Sigma^*$ is said to be accepted/recognized by $M$ if $L = L(M)$.

Problem 5.

1. Which of the following is true?
   - $B \not\xrightarrow{!} M B$
   - $A \not\xrightarrow{01} M D$
   - $D \not\xrightarrow{111} M C$
   - $A \not\xrightarrow{101} M 2 B$

2. What is the following?
   - $\xrightarrow{2} M (A, 1011) = q_2$
   - $\xrightarrow{2} M (B, 010) = q_1$
   - $\xrightarrow{2} M (C, 100) = q_2$

3. What is $L(M)$?

4. What is the language recognized if we change the initial state to $B$?

5. What is the language recognized if we change the set of final states to be $\{B\}$ (with initial state $A$)?

Diagram: DFA $M$ for problem 5:

- $Q = \{q_0, q_1, q_1, q_3\}$
- $\Sigma = \{0, 1\}$
- $\delta$
- $s = q_0$
- $A = \{q_0\}$
Definition 4. A deterministic finite automaton (DFA) is $M = (Q, \Sigma, \delta, s, A)$ where
- $Q$ is a finite set whose elements are called states,
- $\Sigma$ is a finite set called the input alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function,
- $s \in Q$ is the start state,
- $A \subseteq Q$ is the set of accepting/final states.

Definition 5. For a DFA $M = (Q, \Sigma, \delta, s, A)$, string $w = w_1w_2\cdots w_k$, where for each $i$, $w_i \in \Sigma$, and states $p, q \in Q$, $p \xrightarrow{w} M q$ if there is a sequence of states $r_0, r_1, \ldots, r_k$ such that
- $r_0 = p$,
- for each $i$, $(r_i, w_{i+1}) = r_{i+1}$,
- $r_k = q$.

Problem 4. Prove that for any state $p$, and string $w \in \Sigma^*$, there is a unique state $q$ such that $p \xrightarrow{w} M q$.

Notation. $\xrightarrow{w} M q$ where $p \xrightarrow{w} M q$.

Definition 6. Consider a DFA $M = (Q, \Sigma, \delta, s, A)$.
- $M$ accepts string $w \in \Sigma^*$ if $w \xrightarrow{} M q$ for some $q \in A$.
- The language accepted/recognized by a DFA $M$ is $L(M) = \{ w \in \Sigma^* | M \text{ accepts } w \}$.
- A set $L \subseteq \Sigma^*$ is said to be accepted/recognized by $M$ if $L = L(M)$.

Problem 5.
1. Which of the following is true?
   - $B \xrightarrow{} M B$
   - $A \xrightarrow{01} M D$
   - $D \xrightarrow{111} M C$
   - $A \xrightarrow{101} M 2$
2. What is the following?
   - $\xrightarrow{} M (A, 1011) = q_0$
   - $\xrightarrow{} M (B, 010) = q_1$
   - $\xrightarrow{} M (C, 100) = q_2$

Figure 1: DFA $M$ for problem 5

3. What is $L(M)$?
4. What is the language recognized if we change the initial state to $B$?
5. What is the language recognized if we change the set of final states to be $\{B\}$ (with initial state $A$)?

Example

- $Q = \{q_0, q_1, q_1, q_3\}$
- $\Sigma = \{0, 1\}$
- $\delta$
- $s = q_0$
- $A = \{q_0\}$
A deterministic finite automaton (DFA) is $M = (Q, \Sigma, \delta, s, A)$ where
- $Q$ is a finite set whose elements are called states,
- $\Sigma$ is a finite set called the input alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function,
- $s \in Q$ is the start state,
- $A \subseteq Q$ is the set of accepting/final states.

The language accepted/recognized by a DFA $M$ is $L(M) = \{ w \in \Sigma^* \mid M \text{ accepts } w \}$.

Problem 4: Prove that for any state $p$, and string $w \in \Sigma^*$, there exists a unique state $q$ such that $p \xrightarrow{w} M q$.

Notation: $\xrightarrow{\mathcal{M}} (p, w) = q$ where $p \xrightarrow{w} M q$.

Problem 5: 1. Which of the following is true?
   - $B \xrightarrow{\mathcal{M}} B$
   - $A \xrightarrow{01} M D$
   - $D \xrightarrow{111} M C$
   - $A \xrightarrow{101} M 2 B$

   2. What is the following?
   - $\xrightarrow{\mathcal{M}} (A, 1011)$
   - $\xrightarrow{\mathcal{M}} (B, 010)$
   - $\xrightarrow{\mathcal{M}} (C, 100)$

   3. What is $L(M)$?

   4. What is the language recognized if we change the initial state to $B$?

   5. What is the language recognized if we change the set of final states to be $\{B\}$ (with initial state $A$)?

Example

- $Q = \{q_0, q_1, q_3\}$
- $\Sigma = \{0, 1\}$
- $\delta$
- $s = q_0$
- $A = \{q_0\}$

\[\text{Figure 1: DFA } M \text{ for problem 5}\]
**Definition 4.** A deterministic finite automaton (DFA) is $M = (Q, \Sigma, \delta, s, A)$ where
- $Q$ is a finite set whose elements are called states,
- $\Sigma$ is a finite set called the input alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function,
- $s \in Q$ is the start state,
- $A \subseteq Q$ is the set of accepting/final states.

**Definition 5.** For a DFA $M = (Q, \Sigma, \delta, s, A)$, string $w = w_1w_2\cdots w_k$, where for each $i$, $w_i \in \Sigma$, and states $p, q \in Q$, we say $p \xrightarrow{w} M q$ if there is a sequence of states $r_0, r_1, \ldots, r_k$ such that
- $r_0 = p$,
- for each $i$, $(r_i, w_{i+1}) = r_{i+1}$,
- $r_k = q$.

**Problem 4.** Prove that for any state $p$, and string $w \in \Sigma^*$, there is a unique state $q$ such that $p \xrightarrow{w} M q$.

**Notation.** $\xrightarrow{M} (p, w) = q$ where $p \xrightarrow{w} M q$.

**Definition 6.** Consider a DFA $M = (Q, \Sigma, \delta, s, A)$.
- $M$ accepts string $w \in \Sigma^*$ if $M(\varepsilon, w) \in A$.
- The language accepted/recognized by a DFA $M$ is $L(M) = \{ w \in \Sigma^* | M(\varepsilon, w) \in A \}$.
- A set $L \subseteq \Sigma^*$ is said to be accepted/recognized by $M$ if $L = L(M)$.

**Problem 5.**
1. Which of the following is true?
   - $B \xrightarrow{\varepsilon} M B$
   - $A \xrightarrow{01} M D$
   - $D \xrightarrow{111} M C$
   - $A \xrightarrow{101} M 2B$
2. What is the following?
   - $\xrightarrow{M} (A, 1011) = q$
   - $\xrightarrow{M} (B, 010) = q$
   - $\xrightarrow{M} (C, 100) = q$

Figure 1: DFA $M$ for problem 5

- $Q = \{q_0, q_1, q_1, q_3\}$
- $\Sigma = \{0, 1\}$
- $\delta$
- $s = q_0$
- $A = \{q_0\}$
Example

- \( Q = \{q_0, q_1, q_1, q_3\} \)
- \( \Sigma = \{0, 1\} \)
- \( \delta \)
- \( s = q_0 \)
- \( A = \{q_0\} \)
Example

A deterministic finite automaton (DFA) is $M = (Q, \Sigma, \delta, s, A)$ where
- $Q$ is a finite set whose elements are called states,
- $\Sigma$ is a finite set called the input alphabet,
- $\delta: Q \times \Sigma \rightarrow Q$ is the transition function,
- $s \in Q$ is the start state,
- $A \subseteq Q$ is the set of accepting/final states.

Problem 4. Prove that for any state $p$, and string $w \in \Sigma^*$, there is a unique state $q$ such that $p \xrightarrow{w} M q$.

Notation. $\xrightarrow{\Sigma} M (p, w) = q$ where $p \xrightarrow{w} M q$.

Definition 6. Consider a DFA $M = (Q, \Sigma, \delta, s, A)$.
- $M$ accepts string $w \in \Sigma^*$ if $M(s, w) \in A$.
- The language accepted/recognized by a DFA $M$ is $L(M) = \{ w \in \Sigma^* | M \text{ accepts } w \}$.
- A set $L \subseteq \Sigma^*$ is said to be accepted/recognized by $M$ if $L = L(M)$.
Extending the transition function to strings

Given DFA $M = (Q, \Sigma, \delta, s, A)$, $\delta(q, a)$ is the state that $M$ goes to from $q$ on reading letter $a$

Useful to have notation to specify the unique state that $M$ will reach from $q$ on reading string $w$

Transition function $\delta^* : Q \times \Sigma^* \rightarrow Q$ defined inductively as follows:

- $\delta^*(q, \epsilon) = q$ if $w = \epsilon$
- $\delta^*(q, ax) = \delta^*(\delta(q, a), x)$ if $w = ax$. 
Extending the transition function to strings

Given \( \text{DFA } M = (Q, \Sigma, \delta, s, A) \), \( \delta(q, a) \) is the state that \( M \) goes to from \( q \) on reading letter \( a \).

Useful to have notation to specify the unique state that \( M \) will reach from \( q \) on reading string \( w \).

Transition function \( \delta^* : Q \times \Sigma^* \rightarrow Q \) defined inductively as follows:

- \( \delta^*(q, \epsilon) = q \) if \( w = \epsilon \)
- \( \delta^*(q, w) = \delta^*(\delta(q, a), x) \) if \( w = ax \).
The language $L(M)$ accepted by a DFA $M = (Q, \Sigma, \delta, s, A)$ is

$$\{ w \in \Sigma^* | \delta^*(s, w) \in A \}.$$
Example

A deterministic finite automaton (DFA) is $M = (Q, \Gamma, \delta, s, A)$ where

• $Q$ is a finite set whose elements are called states,
• $\Gamma$ is a finite set called the input alphabet,
• $\delta : Q \times \Gamma \rightarrow Q$ is the transition function,
• $s \in Q$ is the start state,
• $A \subseteq Q$ is the set of accepting/final states.

Definition 5. For a DFA $M = (Q, \Gamma, \delta, s, A)$, string $w = w_1 w_2 \cdots w_k$, where for each $i$, $w_i \in \Gamma$, and states $p, q \in Q$, we say $p \xrightarrow{w} M q$ if there is a sequence of states $r_0, r_1, \ldots, r_k$ such that (a) $r_0 = p$, (b) for each $i$, $(r_i, w_{i+1}) = r_{i+1}$, and (c) $r_k = q$.

Problem 4. Prove that for any state $p$, and string $w \in \Gamma^*$, there is a unique state $q$ such that $p \xrightarrow{w} M q$.

Notation. $\xrightarrow{w} M q$.

Definition 6. Consider a DFA $M = (Q, \Gamma, \delta, s, A)$.

• $M$ accepts string $w \in \Gamma^*$ if $s \xrightarrow{w} M q$ for some $q \in A$.
• The language accepted/recognized by a DFA $M$ is $L(M) = \{ w \in \Gamma^* | s \xrightarrow{w} M q \text{ for some } q \in A \}$.

Problem 5. 1. Which of the following is true?
   • $B \xrightarrow{0} M B$
   • $A \xrightarrow{01} M D$
   • $D \xrightarrow{111} M C$
   • $A \xrightarrow{101} M 2 B$

2. What is the following?
   • $\xrightarrow{} M 2 (A, 1011) = q_1$
   • $\xrightarrow{} M 2 (B, 010) = q_0$
   • $\xrightarrow{} M 2 (C, 100) = q_4$

3. What is $L(M)$?

4. What is the language recognized if we change the initial state to $B$?

5. What is the language recognized if we change the set of final states to be $\{B\}$ (with initial state $A$)?

What is:

• $\delta^*(q_1, \epsilon)$
• $\delta^*(q_0, 1011)$
• $\delta^*(q_1, 010)$
• $\delta^*(q_4, 10)$
Example continued

A deterministic finite automaton (DFA) is $M = (Q, \mathcal{A}, s, A)$ where
- $Q$ is a finite set whose elements are called states,
- $\mathcal{A}$ is a finite set called the input alphabet,
- $\delta : Q \times \mathcal{A} \to Q$ is the transition function,
- $s \in Q$ is the start state,
- $A \subseteq Q$ is the set of accepting/final states.

Definition 5. For a DFA $M = (Q, \mathcal{A}, s, A)$, string $w = w_1 w_2 \cdots w_k$, where for each $i$, $w_i \in \mathcal{A}$, and states $p, q \in Q$, we say $p \xrightarrow{w} M q$ if there is a sequence of states $r_0, r_1, \ldots, r_k$ such that
   - (a) $r_0 = p$,
   - (b) for each $i$, $(r_i, w_{i+1}) = r_{i+1}$, and
   - (c) $r_k = q$.

Problem 4. Prove that for any state $p$, and string $w \in \mathcal{A}^*$, there is a unique state $q$ such that $p \xrightarrow{w} M q$.

Notation. $\xrightarrow{w} M (p, w) = q$ where $p \xrightarrow{w} M q$.

Definition 6. Consider a DFA $M = (Q, \mathcal{A}, s, A)$.
- $M$ accepts string $w \in \mathcal{A}^*$ if $M$ reaches an accepting state from $s$ upon reading $w$.
- The language accepted/recognized by a DFA $M$ is $L(M) = \{ w \in \mathcal{A}^* | M$ accepts $w \}$.
- A set $L \subseteq \mathcal{A}^*$ is said to be accepted/recognized by $M$ if $L = L(M)$.

Problem 5.
1. Which of the following is true?
   - $B \xrightarrow{} M B$
   - $A \xrightarrow{} M D$
   - $D \xrightarrow{} M C$
   - $A \xrightarrow{} M 2B$

2. What is the following?
   - $\xrightarrow{} M (A, 1011) = q_0$
   - $\xrightarrow{} M (B, 010) = q_1$
   - $\xrightarrow{} M (C, 100) = q_2$

3. What is $L(M)$?

4. What is the language recognized if we change the initial state to $B$?

5. What is the language recognized if we change the set of final states to be $\{B\}$ (with initial state $A$)?

Figure 1: DFA $M$ for problem 5

- What is $L(M)$ if start state is changed to $q_1$?
- What is $L(M)$ if final/accept states are set to $\{q_2, q_3\}$ instead of $\{q_0\}$?
Advantages of formal specification

- Necessary for proofs
- Necessary to specify abstractly for class of languages

**Exercise:** Prove by induction that for any two strings $u, v$, any state $q$, $\delta^*(q, uv) = \delta^*(\delta^*(q, u), v)$. 
Part II

Constructing DFAs
DFAs: State = Memory

How do we design a DFA $M$ for a given language $L$? That is $L(M) = L$.

- DFA is a like a program that has fixed amount of memory independent of input size.
- The memory of a DFA is encoded in its states
- The state/memory must capture enough information from the input seen so far that it is sufficient for the suffix that is yet to be seen (note that DFA cannot go back)
DFA Construction: Example

Assume $\Sigma = \{0, 1\}$

- $L = \emptyset$, $L = \Sigma^*$, $L = \{\epsilon\}$, $L = \{0\}$.
- $L = \{w \in \{0, 1\}^* \mid |w| \text{ is divisible by 5}\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ ends with 01}\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ contains 001 as substring}\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ contains 001 or 010 as substring}\}$
- $L = \{w \mid w \text{ has a 1 } k \text{ positions from the end}\}$
DFA Construction: Example

Assume $\Sigma = \{0, 1\}$

- $L = \emptyset$, $L = \Sigma^*$, $L = \{\epsilon\}$, $L = \{0\}$.
- $L = \{w \in \{0, 1\}^* \mid |w| \text{ is divisible by 5}\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ ends with 01}\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ contains 001 as substring}\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ contains 001 or 010 as substring}\}$
- $L = \{w \mid w \text{ has a 1 } k \text{ positions from the end}\}$
DFA Construction: Example

Assume $\Sigma = \{0, 1\}$

- $L = \emptyset$, $L = \Sigma^*$, $L = \{\epsilon\}$, $L = \{0\}$.
- $L = \{w \in \{0, 1\}^* \mid |w| \text{ is divisible by } 5\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ ends with } 01\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ contains } 001 \text{ as substring}\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ contains } 001 \text{ or } 010 \text{ as substring}\}$
- $L = \{w \mid w \text{ has a } 1 \text{ } k \text{ positions from the end}\}$
DFA Construction: Example

Assume $\Sigma = \{0, 1\}$

- $L = \emptyset$, $L = \Sigma^*$, $L = \{\epsilon\}$, $L = \{0\}$.
- $L = \{w \in \{0, 1\}^* | |w| \text{ is divisible by 5}\}$
- $L = \{w \in \{0, 1\}^* | w \text{ ends with 01}\}$
- $L = \{w \in \{0, 1\}^* | w \text{ contains 001 as substring}\}$
- $L = \{w \in \{0, 1\}^* | w \text{ contains 001 or 010 as substring}\}$
- $L = \{w \mid w \text{ has a 1 k positions from the end}\}$
DFA Construction: Example

Assume $\Sigma = \{0, 1\}$

- $L = \emptyset$, $L = \Sigma^*$, $L = \{\epsilon\}$, $L = \{0\}$.
- $L = \{w \in \{0, 1\}^* \mid |w| \text{ is divisible by } 5\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ ends with } 01\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ contains } 001 \text{ as substring}\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ contains } 001 \text{ or } 010 \text{ as substring}\}$
- $L = \{w \mid w \text{ has a } 1 \text{ k positions from the end}\}$
DFA Construction: Example

Assume $\Sigma = \{0, 1\}$

- $L = \emptyset$, $L = \Sigma^*$, $L = \{\varepsilon\}$, $L = \{0\}$.
- $L = \{w \in \{0, 1\}^* \mid |w| \text{ is divisible by 5}\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ ends with 01}\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ contains 001 as substring}\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ contains 001 or 010 as substring}\}$
- $L = \{w \mid w \text{ has a 1 } k \text{ positions from the end}\}$
**DFA Construction: Example**

\[ L = \{ \text{Binary numbers congruent to } 0 \mod 5 \} \]

Example: \(1101011 = 107 = 2 \mod 5\), \(1010 = 10 = 0 \mod 5\)

Key observation:

\[ w_0 \mod 5 = a \text{ implies } w_0 \mod 5 = 2a \mod 5 \text{ and } w_1 \mod 5 = (2a + 1) \mod 5 \]
**DFA Construction: Example**

\[ L = \{ \text{Binary numbers congruent to } 0 \mod 5 \} \]

Example: \(1101011 = 107 = 2 \mod 5, 1010 = 10 = 0 \mod 5\)

**Key observation:**

\( w_0 \mod 5 = a \) implies

\( w_0 \mod 5 = 2a \mod 5 \) and \( w_1 \mod 5 = (2a + 1) \mod 5 \)
Part III

Product Construction and Closure Properties
Part IV

Complement
Question: If $M$ is a DFA, is there a DFA $M'$ such that $L(M') = \Sigma^* \setminus L(M)$? That is, are languages recognized by DFAs closed under complement?
Complement

Example...

Just flip the state of the states!
Languages accepted by DFAs are closed under complement.

Proof.

Let $M = (Q, \Sigma, \delta, s, A)$ such that $L = L(M)$.
Let $M' = (Q, \Sigma, \delta, s, Q \setminus A)$. Claim: $L(M') = \overline{L}$. Why?

$\delta^*_M = \delta^*_{M'}$. Thus, for every string $w$, $\delta^*_M(s, w) = \delta^*_{M'}(s, w)$.

$\delta^*_M(s, w) \in A \Rightarrow \delta^*_{M'}(s, w) \not\in Q \setminus A$.

$\delta^*_M(s, w) \not\in A \Rightarrow \delta^*_{M'}(s, w) \in Q \setminus A$. 


Complement

Theorem

Languages accepted by DFAs are closed under complement.

Proof.
Let $M = (Q, \Sigma, \delta, s, A)$ such that $L = L(M)$.
Let $M' = (Q, \Sigma, \delta, s, Q \setminus A)$. Claim: $L(M') = \bar{L}$. Why?

$\delta^*_M = \delta^*_{M'}$. Thus, for every string $w$, $\delta^*_M(s, w) = \delta^*_{M'}(s, w)$.

$\delta^*_M(s, w) \in A \Rightarrow \delta^*_{M'}(s, w) \notin Q \setminus A$.

$\delta^*_M(s, w) \notin A \Rightarrow \delta^*_{M'}(s, w) \in Q \setminus A$. 

□
Theorem

Languages accepted by DFAs are closed under complement.

Proof.

Let \( M = (Q, \Sigma, \delta, s, A) \) such that \( L = L(M) \).

Let \( M' = (Q, \Sigma, \delta, s, Q \setminus A) \). Claim: \( L(M') = \overline{L} \). Why?

\[ \delta^*_M = \delta^*_{M'} \]. Thus, for every string \( w \), \( \delta^*_M(s, w) = \delta^*_{M'}(s, w) \).

\( \delta^*_M(s, w) \in A \Rightarrow \delta^*_{M'}(s, w) \notin Q \setminus A \).

\( \delta^*_M(s, w) \notin A \Rightarrow \delta^*_{M'}(s, w) \in Q \setminus A \).
Part V

Product Construction
Union and Intersection

**Question:** Are languages accepted by **DFA**s closed under union? That is, given **DFA**s $M_1$ and $M_2$ is there a **DFA** that accepts $L(M_1) \cup L(M_2)$? How about intersection $L(M_1) \cap L(M_2)$?

Idea from programming: on input string $w$

- Simulate $M_1$ on $w$
- Simulate $M_2$ on $w$
- If both accept than $w \in L(M_1) \cap L(M_2)$. If at least one accepts then $w \in L(M_1) \cup L(M_2)$.
- **Catch:** We want a single **DFA** $M$ that can only read $w$ once.
- **Solution:** Simulate $M_1$ and $M_2$ in parallel by keeping track of states of both machines
Question: Are languages accepted by DFAs closed under union? That is, given DFAs $M_1$ and $M_2$ is there a DFA that accepts $L(M_1) \cup L(M_2)$? How about intersection $L(M_1) \cap L(M_2)$?

Idea from programming: on input string $w$

- Simulate $M_1$ on $w$
- Simulate $M_2$ on $w$
- If both accept than $w \in L(M_1) \cap L(M_2)$. If at least one accepts then $w \in L(M_1) \cup L(M_2)$.
- Catch: We want a single DFA $M$ that can only read $w$ once.
- Solution: Simulate $M_1$ and $M_2$ in parallel by keeping track of states of both machines.
Union and Intersection

**Question:** Are languages accepted by **DFA**s closed under union? That is, given **DFA**s $M_1$ and $M_2$ is there a **DFA** that accepts $L(M_1) \cup L(M_2)$? How about intersection $L(M_1) \cap L(M_2)$?

Idea from programming: on input string $w$
- Simulate $M_1$ on $w$
- Simulate $M_2$ on $w$
- If both accept then $w \in L(M_1) \cap L(M_2)$. If at least one accepts then $w \in L(M_1) \cup L(M_2)$.

**Catch:** We want a single **DFA** $M$ that can only read $w$ once.

**Solution:** Simulate $M_1$ and $M_2$ in parallel by keeping track of states of both machines.
**Union and Intersection**

**Question:** Are languages accepted by DFA s closed under union? That is, given DFAs $M_1$ and $M_2$ is there a DFA that accepts $L(M_1) \cup L(M_2)$? How about intersection $L(M_1) \cap L(M_2)$?

Idea from programming: on input string $w$

- Simulate $M_1$ on $w$
- Simulate $M_2$ on $w$
  
- If both accept then $w \in L(M_1) \cap L(M_2)$. If at least one accepts then $w \in L(M_1) \cup L(M_2)$.

  **Catch:** We want a single DFA $M$ that can only read $w$ once.

  **Solution:** Simulate $M_1$ and $M_2$ in parallel by keeping track of states of both machines.
Example

$M_1$ accepts #0 = odd

$M_2$ accepts #1 = odd
Example

$M_1$ accepts #0 = odd

$M_2$ accepts #1 = odd

Cross-product machine
Example II

Accept all binary strings of length divisible by 3 and 5

Assume all edges are labeled by 0, 1.
Product construction for intersection

\[ M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1) \text{ and } M_2 = (Q_1, \Sigma, \delta_2, s_2, A_2) \]

Create \( M = (Q, \Sigma, \delta, s, A) \) where

- \( Q = Q_1 \times Q_2 = \{(q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2\} \)
- \( s = (s_1, s_2) \)
- \( \delta : Q \times \Sigma \to Q \) where
  \[
  \delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))
  \]
- \( A = A_1 \times A_2 = \{(q_1, q_2) \mid q_1 \in A_1, q_2 \in A_2\} \)

**Theorem**

\[ L(M) = L(M_1) \cap L(M_2). \]
Product construction for intersection

\( M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1) \) and \( M_2 = (Q_1, \Sigma, \delta_2, s_2, A_2) \)

Create \( M = (Q, \Sigma, \delta, s, A) \) where

- \( Q = Q_1 \times Q_2 = \{ (q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2 \} \)
- \( s = (s_1, s_2) \)
- \( \delta : Q \times \Sigma \to Q \) where
  \[ \delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a)) \]

- \( A = A_1 \times A_2 = \{ (q_1, q_2) \mid q_1 \in A_1, q_2 \in A_2 \} \)

**Theorem**

\( L(M) = L(M_1) \cap L(M_2) \).
Product construction for intersection

$M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1)$ and $M_2 = (Q_1, \Sigma, \delta_2, s_2, A_2)$

Create $M = (Q, \Sigma, \delta, s, A)$ where

- $Q = Q_1 \times Q_2 = \{(q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2\}$
- $s = (s_1, s_2)$
- $\delta : Q \times \Sigma \to Q$ where

$$\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$$

- $A = A_1 \times A_2 = \{(q_1, q_2) \mid q_1 \in A_1, q_2 \in A_2\}$

**Theorem**

$L(M) = L(M_1) \cap L(M_2)$. 

Chan, Har-Peled, Hassanieh (UIUC) CS374 Spring 2019 32 / 36
Product construction for intersection

\[ M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1) \text{ and } M_2 = (Q_1, \Sigma, \delta_2, s_2, A_2) \]

Create \( M = (Q, \Sigma, \delta, s, A) \) where

- \( Q = Q_1 \times Q_2 = \{(q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2\} \)
- \( s = (s_1, s_2) \)
- \( \delta : Q \times \Sigma \rightarrow Q \) where
  \[
  \delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))
  \]
- \( A = A_1 \times A_2 = \{(q_1, q_2) \mid q_1 \in A_1, q_2 \in A_2\} \)

Theorem

\[ L(M) = L(M_1) \cap L(M_2). \]
Product construction for intersection

\[ M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1) \] and \[ M_2 = (Q_1, \Sigma, \delta_2, s_2, A_2) \]

Create \[ M = (Q, \Sigma, \delta, s, A) \] where

- \[ Q = Q_1 \times Q_2 = \{ (q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2 \} \]
- \[ s = (s_1, s_2) \]
- \[ \delta : Q \times \Sigma \rightarrow Q \] where \[ \delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a)) \]

- \[ A = A_1 \times A_2 = \{ (q_1, q_2) \mid q_1 \in A_1, q_2 \in A_2 \} \]

**Theorem**

\[ L(M) = L(M_1) \cap L(M_2). \]
Product construction for intersection

\[ M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1) \] and \[ M_2 = (Q_1, \Sigma, \delta_2, s_2, A_2) \]

Create \( M = (Q, \Sigma, \delta, s, A) \) where

- \( Q = Q_1 \times Q_2 = \{(q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2\} \)
- \( s = (s_1, s_2) \)
- \( \delta : Q \times \Sigma \rightarrow Q \) where

\[
\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))
\]

- \( A = A_1 \times A_2 = \{(q_1, q_2) \mid q_1 \in A_1, q_2 \in A_2\} \)

Theorem

\[ L(M) = L(M_1) \cap L(M_2). \]
Product construction for intersection

\( M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1) \) and \( M_2 = (Q_1, \Sigma, \delta_2, s_2, A_2) \)

Create \( M = (Q, \Sigma, \delta, s, A) \) where

- \( Q = Q_1 \times Q_2 = \{(q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2\} \)
- \( s = (s_1, s_2) \)
- \( \delta: Q \times \Sigma \rightarrow Q \) where
  \[
  \delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))
  \]
- \( A = A_1 \times A_2 = \{(q_1, q_2) \mid q_1 \in A_1, q_2 \in A_2\} \)

Theorem

\[ L(M) = L(M_1) \cap L(M_2). \]
Product construction for intersection

\[ M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1) \text{ and } M_2 = (Q_1, \Sigma, \delta_2, s_2, A_2) \]

Create \( M = (Q, \Sigma, \delta, s, A) \) where

- \( Q = Q_1 \times Q_2 = \{(q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2\} \)
- \( s = (s_1, s_2) \)
- \( \delta : Q \times \Sigma \to Q \) where

\[ \delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a)) \]

- \( A = A_1 \times A_2 = \{(q_1, q_2) \mid q_1 \in A_1, q_2 \in A_2\} \)

**Theorem**

\[ L(M) = L(M_1) \cap L(M_2). \]
Product construction for intersection

\[ M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1) \] and \[ M_2 = (Q_1, \Sigma, \delta_2, s_2, A_2) \]

Create \[ M = (Q, \Sigma, \delta, s, A) \] where

- \[ Q = Q_1 \times Q_2 = \{(q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2\} \]
- \[ s = (s_1, s_2) \]
- \[ \delta : Q \times \Sigma \rightarrow Q \] where

\[
\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))
\]

- \[ A = A_1 \times A_2 = \{(q_1, q_2) \mid q_1 \in A_1, q_2 \in A_2\} \]

**Theorem**

\[ L(M) = L(M_1) \cap L(M_2). \]
Product construction for intersection

\[ M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1) \text{ and } M_2 = (Q_1, \Sigma, \delta_2, s_2, A_2) \]

Create \( M = (Q, \Sigma, \delta, s, A) \) where

- \( Q = Q_1 \times Q_2 = \{ (q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2 \} \)
- \( s = (s_1, s_2) \)
- \( \delta : Q \times \Sigma \rightarrow Q \)

where

\[ \delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a)) \]

- \( A = A_1 \times A_2 = \{ (q_1, q_2) \mid q_1 \in A_1, q_2 \in A_2 \} \)

**Theorem**

\[ L(M) = L(M_1) \cap L(M_2). \]
Correctness of construction

Lemma

For each string $w$, $\delta^*(s, w) = (\delta^*_1(s_1, w), \delta^*_2(s_2, w))$.

Exercise: Assuming lemma prove the theorem in previous slide.
Proof of lemma by induction on $|w|$. 
Correctness of construction

**Lemma**

For each string $w$, $\delta^*(s, w) = (\delta^*_1(s_1, w), \delta^*_2(s_2, w))$.

**Exercise:** Assuming lemma prove the theorem in previous slide.

Proof of lemma by induction on $|w|$. 
Correctness of construction

**Lemma**

For each string $w$, $\delta^*(s, w) = (\delta_1^*(s_1, w), \delta_2^*(s_2, w))$.

**Exercise**: Assuming lemma prove the theorem in previous slide. Proof of lemma by induction on $|w|$.
Product construction for union

\[ M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1) \text{ and } M_2 = (Q_1, \Sigma, \delta_2, s_2, A_2) \]

Create \( M = (Q, \Sigma, \delta, s, A) \) where

- \( Q = Q_1 \times Q_2 = \{ (q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2 \} \)
- \( s = (s_1, s_2) \)
- \( \delta : Q \times \Sigma \rightarrow Q \) where

\[
\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))
\]

- \( A = \{ (q_1, q_2) \mid q_1 \in A_1 \text{ or } q_2 \in A_2 \} \)

**Theorem**

\[ L(M) = L(M_1) \cup L(M_2). \]
Product construction for union

\[ M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1) \text{ and } M_2 = (Q_1, \Sigma, \delta_2, s_2, A_2) \]

Create \( M = (Q, \Sigma, \delta, s, A) \) where

- \( Q = Q_1 \times Q_2 = \{(q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2\} \)
- \( s = (s_1, s_2) \)
- \( \delta : Q \times \Sigma \to Q \) where
  \[
  \delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))
  \]

- \( A = \{(q_1, q_2) \mid q_1 \in A_1 \text{ or } q_2 \in A_2\} \)

**Theorem**

\[ L(M) = L(M_1) \cup L(M_2). \]
**Theorem**

$M_1, M_2$ DFAs. There is a DFA $M$ such that $L(M) = L(M_1) \setminus L(M_2)$.

**Exercise:** Prove the above using two methods.

- Using a direct product construction
- Using closure under complement and intersection and union
Question: Why are DFA\textsubscript{s} required to only move right? Can we allow DFA to scan back and forth? Caveat: Tape is read-only so only memory is in machine’s state.

- Can define a formal notion of a “2-way” DFA
- Can show that any language recognized by a 2-way DFA can be recognized by a regular (1-way) DFA
- Proof is tricky simulation via NFA\textsubscript{s}
**Question:** Why are **DFA**s required to only move right? Can we allow **DFA** to scan back and forth? **Caveat:** Tape is read-only so only memory is in machine’s state.

- Can define a formal notion of a “2-way” **DFA**
- Can show that any language recognized by a 2-way **DFA** can be recognized by a regular (1-way) **DFA**
- Proof is tricky simulation via **NFA**s