# Algorithms & Models of Computation CS/ECE 374, Fall 2020

**3.4** Product Construction

**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

**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

**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

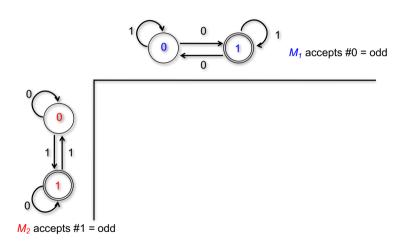
- Simulate  $M_1$  on w
- Simulate  $M_2$  on w
- If both accept than  $w \in L(M_1) \cap L(M_2)$ . If at least one accepts then  $w \in L(M_1) \cup L(M_2)$ .
- Catch: We want a single DFA *M* that can only read *w* once.
- Solution: Simulate  $M_1$  and  $M_2$  in parallel by keeping track of states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

**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  $L(M_1) \cap L(M_2)$ ?

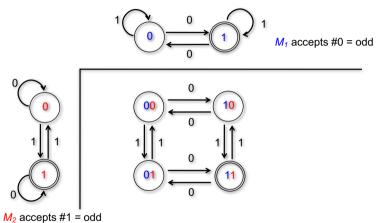
Idea from programming: on input string w

- 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

# Example



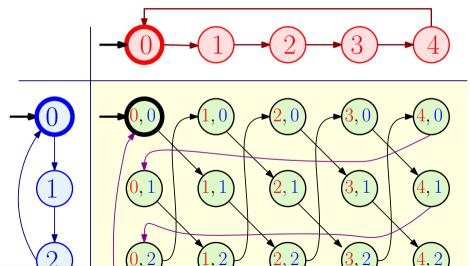
# Example



**Cross-product machine** 

# Example II

Accept all binary strings of length divisible by 3 and  $5\,$ 



Har-Peled (UIUC) CS374 47 Fall 2020

$$extbf{ extit{M}}_1 = ( extbf{ extit{Q}}_1, \Sigma, \delta_1, extbf{ extit{s}}_1, extbf{ extit{A}}_1) ext{ and } extbf{ extit{M}}_2 = ( extbf{ extit{Q}}_1, \Sigma, \delta_2, extbf{ extit{s}}_2, extbf{ extit{A}}_2)$$

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

$$ullet Q = Q_1 imes Q_2 = \{(q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2\}$$

- $s = (s_1, s_2)$
- ullet  $\delta: oldsymbol{Q} imes oldsymbol{\Sigma} o oldsymbol{Q}$  where

$$\delta((\boldsymbol{q}_1,\boldsymbol{q}_2),\boldsymbol{a})=(\delta_1(\boldsymbol{q}_1,\boldsymbol{a}),\delta_2(\boldsymbol{q}_2,\boldsymbol{a}))$$

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

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

$$extbf{ extit{M}}_1 = ( extbf{ extit{Q}}_1, \Sigma, \delta_1, extbf{ extit{s}}_1, extbf{ extit{A}}_1) ext{ and } extbf{ extit{M}}_2 = ( extbf{ extit{Q}}_1, \Sigma, \delta_2, extbf{ extit{s}}_2, extbf{ extit{A}}_2)$$

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

$$\bullet \ \ \textit{\textbf{Q}} = \textit{\textbf{Q}}_1 \times \textit{\textbf{Q}}_2 = \{(\textit{\textbf{q}}_1, \textit{\textbf{q}}_2) \mid \textit{\textbf{q}}_1 \in \textit{\textbf{Q}}_1, \textit{\textbf{q}}_2 \in \textit{\textbf{Q}}_2\}$$

- $s = (s_1, s_2)$
- ullet  $\delta: oldsymbol{Q} imes oldsymbol{\Sigma} 
  ightarrow oldsymbol{Q}$  where

$$\boldsymbol{\delta}((\boldsymbol{q}_1,\boldsymbol{q}_2),\boldsymbol{a}) = (\delta_1(\boldsymbol{q}_1,\boldsymbol{a}),\delta_2(\boldsymbol{q}_2,\boldsymbol{a}))$$

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

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

$$extbf{ extit{M}}_1 = ( extbf{ extit{Q}}_1, \Sigma, oldsymbol{\delta}_1, oldsymbol{s}_1, oldsymbol{\delta}_1, oldsymbol{A}_1) ext{ and } extbf{ extit{M}}_2 = ( extbf{ extit{Q}}_1, \Sigma, oldsymbol{\delta}_2, oldsymbol{s}_2, oldsymbol{\delta}_2, oldsymbol{s}_2, oldsymbol{\delta}_2)$$

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

- $ullet \ m{Q} = m{Q}_1 imes m{Q}_2 = \{(m{q}_1, m{q}_2) \mid m{q}_1 \in m{Q}_1, m{q}_2 \in m{Q}_2\}$
- $s = (s_1, s_2)$
- ullet  $\delta: oldsymbol{Q} imes oldsymbol{\Sigma} o oldsymbol{Q}$  where

$$\boldsymbol{\delta}((\boldsymbol{q}_1,\boldsymbol{q}_2),\boldsymbol{a}) = (\delta_1(\boldsymbol{q}_1,\boldsymbol{a}),\delta_2(\boldsymbol{q}_2,\boldsymbol{a}))$$

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

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

$$extbf{ extit{M}}_1 = ( extbf{ extit{Q}}_1, \Sigma, oldsymbol{\delta}_1, oldsymbol{s}_1, oldsymbol{\delta}_1, oldsymbol{A}_1) ext{ and } extbf{ extit{M}}_2 = ( extbf{ extit{Q}}_1, \Sigma, oldsymbol{\delta}_2, oldsymbol{s}_2, oldsymbol{\delta}_2, oldsymbol{s}_2, oldsymbol{\delta}_2)$$

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

- $ullet \ m{Q} = m{Q}_1 imes m{Q}_2 = \{(m{q}_1, m{q}_2) \mid m{q}_1 \in m{Q}_1, m{q}_2 \in m{Q}_2\}$
- $s = (s_1, s_2)$
- ullet  $\delta: {m Q} imes {m \Sigma} o {m Q}$  where

$$\delta((\boldsymbol{q}_1,\boldsymbol{q}_2),a)=(\delta_1(\boldsymbol{q}_1,a),\delta_2(\boldsymbol{q}_2,a))$$

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

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

$$extbf{ extit{M}}_1 = ( extbf{ extit{Q}}_1, \Sigma, oldsymbol{\delta}_1, oldsymbol{s}_1, oldsymbol{\delta}_1, oldsymbol{A}_1) ext{ and } extbf{ extit{M}}_2 = ( extbf{ extit{Q}}_1, \Sigma, oldsymbol{\delta}_2, oldsymbol{s}_2, oldsymbol{\delta}_2, oldsymbol{s}_2, oldsymbol{\delta}_2)$$

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

- $ullet \ m{Q} = m{Q}_1 imes m{Q}_2 = \{(m{q}_1, m{q}_2) \mid m{q}_1 \in m{Q}_1, m{q}_2 \in m{Q}_2\}$
- $s = (s_1, s_2)$
- ullet  $\delta: Q \times \Sigma \to Q$  where

$$\delta((\boldsymbol{q}_1,\boldsymbol{q}_2),a)=(\delta_1(\boldsymbol{q}_1,a),\delta_2(\boldsymbol{q}_2,a))$$

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

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

$$extbf{ extit{M}}_1 = ( extbf{ extit{Q}}_1, \Sigma, oldsymbol{\delta}_1, oldsymbol{s}_1, oldsymbol{A}_1) ext{ and } extbf{ extit{M}}_2 = ( extbf{ extit{Q}}_1, \Sigma, oldsymbol{\delta}_2, oldsymbol{s}_2, oldsymbol{s}_2, oldsymbol{A}_2)$$

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

- $ullet \; m{Q} = m{Q}_1 imes m{Q}_2 = \{(m{q}_1, m{q}_2) \; | \; m{q}_1 \in m{Q}_1, m{q}_2 \in m{Q}_2\}$
- $s = (s_1, s_2)$
- $\delta: Q \times \Sigma \rightarrow Q$  where

$$\delta((\mathbf{q}_1,\mathbf{q}_2),\mathbf{a})=(\delta_1(\mathbf{q}_1,\mathbf{a}),\delta_2(\mathbf{q}_2,\mathbf{a}))$$

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

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

$$extbf{ extit{M}}_1 = ( extbf{ extit{Q}}_1, \Sigma, \delta_1, extbf{ extit{s}}_1, extbf{ extit{A}}_1) ext{ and } extbf{ extit{M}}_2 = ( extbf{ extit{Q}}_1, \Sigma, \delta_2, extbf{ extit{s}}_2, extbf{ extit{A}}_2)$$

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

- $ullet \ m{Q} = m{Q}_1 imes m{Q}_2 = \{(m{q}_1, m{q}_2) \mid m{q}_1 \in m{Q}_1, m{q}_2 \in m{Q}_2\}$
- $s = (s_1, s_2)$
- $\delta: Q \times \Sigma \rightarrow Q$  where

$$\delta((\boldsymbol{q}_1, \boldsymbol{q}_2), \boldsymbol{a}) = (\delta_1(\boldsymbol{q}_1, \boldsymbol{a}), \delta_2(\boldsymbol{q}_2, \boldsymbol{a}))$$

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

### **Theorem**

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

$$extbf{ extit{M}}_1 = ( extbf{ extit{Q}}_1, \Sigma, oldsymbol{\delta}_1, oldsymbol{s}_1, oldsymbol{A}_1) ext{ and } extbf{ extit{M}}_2 = ( extbf{ extit{Q}}_1, \Sigma, oldsymbol{\delta}_2, oldsymbol{s}_2, oldsymbol{s}_2, oldsymbol{A}_2)$$

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

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

$$\delta((\boldsymbol{q}_1, \boldsymbol{q}_2), \boldsymbol{a}) = (\delta_1(\boldsymbol{q}_1, \boldsymbol{a}), \delta_2(\boldsymbol{q}_2, \boldsymbol{a}))$$

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

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

$$extbf{ extit{M}}_1 = ( extbf{ extit{Q}}_1, \Sigma, \delta_1, extbf{ extit{s}}_1, extbf{ extit{A}}_1) ext{ and } extbf{ extit{M}}_2 = ( extbf{ extit{Q}}_1, \Sigma, \delta_2, extbf{ extit{s}}_2, extbf{ extit{A}}_2)$$

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

- $ullet \ m{Q} = m{Q}_1 imes m{Q}_2 = \{(m{q}_1, m{q}_2) \mid m{q}_1 \in m{Q}_1, m{q}_2 \in m{Q}_2\}$
- $s = (s_1, s_2)$
- $\delta: Q \times \Sigma \rightarrow Q$  where

$$\delta((\boldsymbol{q}_1, \boldsymbol{q}_2), \boldsymbol{a}) = (\delta_1(\boldsymbol{q}_1, \boldsymbol{a}), \delta_2(\boldsymbol{q}_2, \boldsymbol{a}))$$

$$\bullet \ \ \pmb{A} = \pmb{A}_1 \times \pmb{A}_2 = \{(\pmb{q}_1, \pmb{q}_2) \mid \pmb{q}_1 \in \pmb{A}_1, \pmb{q}_2 \in \pmb{A}_2\}$$

### **Theorem**

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

$$extbf{ extit{M}}_1 = ( extbf{ extit{Q}}_1, \Sigma, oldsymbol{\delta}_1, oldsymbol{s}_1, oldsymbol{\delta}_1, oldsymbol{\delta}_1, oldsymbol{\delta}_1, oldsymbol{\delta}_2, oldsymbol{s}_2, oldsymbol{\delta}_2, oldsymb$$

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

- $ullet \ m{Q} = m{Q}_1 imes m{Q}_2 = \{(m{q}_1, m{q}_2) \mid m{q}_1 \in m{Q}_1, m{q}_2 \in m{Q}_2\}$
- $s = (s_1, s_2)$
- ullet  $\delta: oldsymbol{Q} imes oldsymbol{\Sigma} o oldsymbol{Q}$  where

$$\delta((\boldsymbol{q}_1,\boldsymbol{q}_2),\boldsymbol{a})=(\delta_1(\boldsymbol{q}_1,\boldsymbol{a}),\delta_2(\boldsymbol{q}_2,\boldsymbol{a}))$$

ullet  $m{A} = m{A}_1 imes m{A}_2 = \{(m{q}_1, m{q}_2) \mid m{q}_1 \in m{A}_1, m{q}_2 \in m{A}_2\}$ 

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

# 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

$$extbf{ extit{M}}_1 = ( extbf{ extit{Q}}_1, \Sigma, \delta_1, extbf{ extit{s}}_1, extbf{ extit{A}}_1) ext{ and } extbf{ extit{M}}_2 = ( extbf{ extit{Q}}_1, \Sigma, \delta_2, extbf{ extit{s}}_2, extbf{ extit{A}}_2)$$

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

- $ullet \ m{Q} = m{Q}_1 imes m{Q}_2 = \{(m{q}_1, m{q}_2) \mid m{q}_1 \in m{Q}_1, m{q}_2 \in m{Q}_2\}$
- $s = (s_1, s_2)$
- $\delta: Q \times \Sigma \rightarrow Q$  where

$$\delta((\boldsymbol{q}_1,\boldsymbol{q}_2),\boldsymbol{a})=(\delta_1(\boldsymbol{q}_1,\boldsymbol{a}),\delta_2(\boldsymbol{q}_2,\boldsymbol{a}))$$

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

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

## Product construction for union

$$extbf{ extit{M}}_1 = ( extbf{ extit{Q}}_1, \Sigma, oldsymbol{\delta}_1, oldsymbol{s}_1, oldsymbol{\delta}_1, oldsymbol{\delta}_1, oldsymbol{\delta}_1, oldsymbol{\delta}_2, oldsymb$$

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

- $ullet \ m{Q} = m{Q}_1 imes m{Q}_2 = \{(m{q}_1, m{q}_2) \mid m{q}_1 \in m{Q}_1, m{q}_2 \in m{Q}_2\}$
- $s = (s_1, s_2)$
- $\delta: Q \times \Sigma \rightarrow Q$  where

$$\delta((\boldsymbol{q}_1, \boldsymbol{q}_2), \boldsymbol{a}) = (\delta_1(\boldsymbol{q}_1, \boldsymbol{a}), \delta_2(\boldsymbol{q}_2, \boldsymbol{a}))$$

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

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

# Set Difference

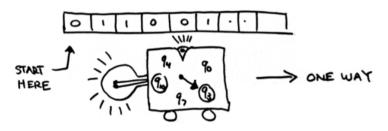
### **Theorem**

 $M_1$ ,  $M_2$  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

# Things to know: 2-way DFA



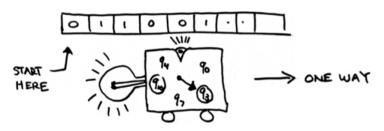
**Question:** Why are DFAs required to only move right?

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
- ullet 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 NFAs

Har-Peled (UIUC) CS374 52 Fall 2020 52 / 58

# Things to know: 2-way DFA



**Question:** Why are DFAs 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 NFAs

Har-Peled (UIUC) CS374 52 Fall 2020 52 / 58

# THE END

...

(for now)