CSE331	Automata and Computability

Lecture 6.3: DFA Cross Product Operation

Presenter: Warida Rashid *Scribe: Warida Rashid*

The cross product is the process of constructing a DFA that simulates the steps of two different DFAs in parallel.

Let the two DFAs be M1 and M2 accepting regular languages L1 and L2

1. $A_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1)$
2. $A_2 = (Q_2, \Sigma, \delta_1, q_0^2, F_2)$

From these two DFAs, we can construct a DFA, **$A = (Q, \Sigma, \delta, q_0, F)$** that recognizes

1. $A_1 \cup A_2$
2. $A_1 \cap A_2$
3. $A_1 - A_2$

For the DFA, **$A = (Q, \Sigma, \delta, q_0, F)$:**

Set of states

Q is the set of states. It contains pairs of states from A_1 and A_2

$$Q = Q_1 X Q_2 = \{(q_1, q_2) \mid q_1 \in Q_1 \text{ and } q_2 \in Q_2\}$$

Start State

$q_0 = (q_0^1, q_0^2)$

Transition Function

$\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$ where q_1 and q_2 are states of A_1 and A_2 respectively and (q_1, q_2) represents the state of the new DFA, A that is derived from A_1 and A_2

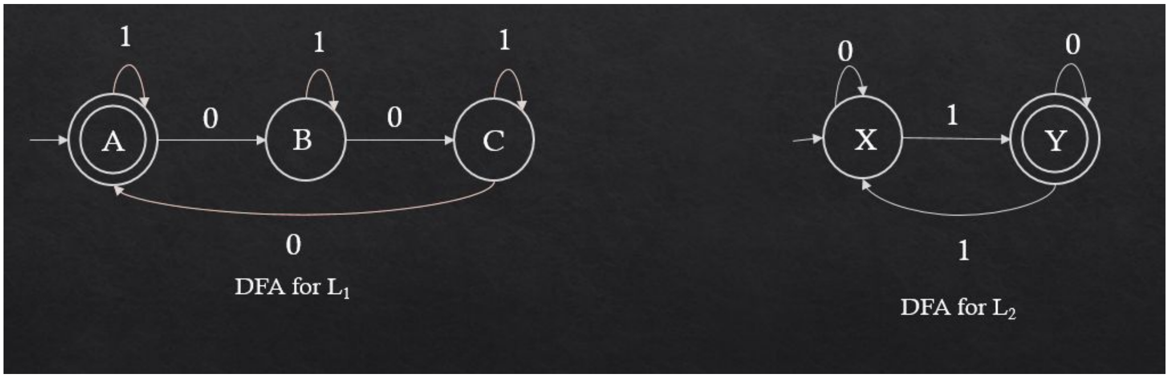
Set of Final States

1. For $A = A_1 \cup A_2$, $F = \{(q_1, q_2) \mid q_1 \in F_1$ **OR** $q_2 \in F_2\}$, it accepts when either A_1 or A_2 accepts
2. For $A = A_1 \cap A_2$, $F = \{(q_1, q_2) \mid q_1 \in F_1$ **AND** $q_2 \in F_2\}$, it accepts when both A_1 and A_2 accepts
3. or $A = A_1 - A_2$, $F = \{(q_1, q_2) \mid q_1 \in F_1$ **AND** $q_2 \notin F_2\}$ it accepts when A_1 accepts and A_2 rejects

Example

Let's consider the DFAs for the following languages:

1. L1 = {w | w ∈ (0,1)* and n0(w) is divisible by 3} i.e. the number of 0 is divisible by 3
2. L2 = {w | w ∈ (0,1)* and n1(w) is odd} i.e. the number of 1 is odd



Here, the **set of states** for the first DFA, **Q_1 = {A, B, C}** the second DFA, **Q_2 = {X, Y}**

Start state of the first DFA is **A** and the second DFA is **X**

Set of final state of the first DFA is **F_1 = {A}** and the second DFA is **F_2 = {Y}**

Therefore for the new DFA:

The set of states is, **Q = Q_1 X Q_2 = {(A, X), (A, Y), (B, X), (B, Y), (C, X), (C, Y) }**

For simplicity, let's name the states as AX, AY, BX, BY, CX, CY. So, **Q = {AX, AY, BX, BY, CX, CY}**

The start state is the state that corresponds to **(A, X)**, which is **AX**

The set of final states:

1. If **L = L1 ∪ L2** = {w | w ∈ (0,1)* and the $n_0(w)$ is divisible by 3 **OR** $n_1(w)$ is odd}

   F = {AX, AY, BY, CY}

2. If **L = L1 ∩ L2** = {w | w ∈ (0,1)* and the $n_0(w)$ is divisible by 3 **AND** $n_1(w)$ is odd},

   F = {AY}

3. **L = L1 - L2** = {w | w ∈ (0,1)* and the $n_0(w)$ is divisible by 3 **AND** $n_1(w)$ is **NOT** odd},

   F = {AX}

The Transitions:

We can figure out the transitions for the new DFA from the transitions of the two DFAs it simulates. It is easier to see from the transition table.

	0	1
→ * A	B	A
B	C	B
C	A	C

Image: Transition table for L_1

	0	1
→ X	X	Y
* Y	Y	X

Image: Transition table for L_2

δ (A, 0) = B and δ (X, 0) = X
So, δ (AX, 0) = BX

δ (B, 0) = C and δ (X, 0) = X
So, δ (BX, 0) = CX

δ (C, 0) = A and δ (X, 0) = X
So, δ (CX, 0) = AX

δ (A, 0) = B and δ (Y, 0) = Y
So, δ (AY, 0) = BY

δ (B, 0) = C and δ (Y, 0) = Y
So, δ (BY, 0) = CY

δ (C, 0) = A and δ (Y, 0) = Y
So, δ (CY, 0) = AY

δ (A, 1) = A and δ (X, 1) = Y
So, δ (AX, 1) = AY

δ (B, 1) = B and δ (X, 1) = Y
So, δ (BX, 1) = BY

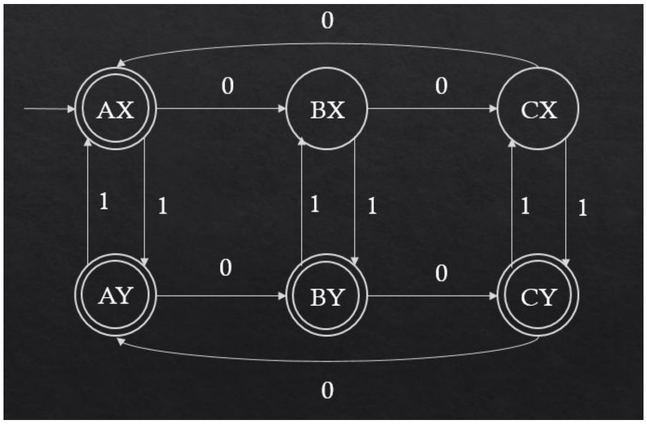
δ (C, 1) = C and δ (X,1) = Y
So, δ (CX, 1) = CY

δ (A, 1) = A and δ (Y, 1) = X
So, δ (AY, 1) = AX

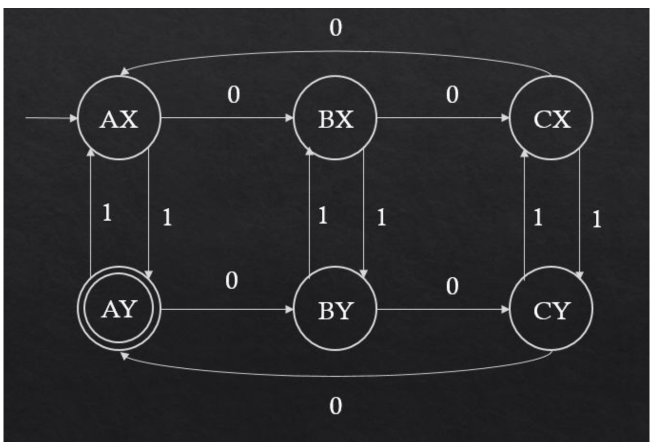
δ (B, 1) = B and δ (Y, 1) = X
So, δ (BY, 1) = BX

δ (C, 1) = C and δ (Y, 1) = X
So, δ (CY, 1) =CX

The DFA for L = L1 ∪ L2 = {w | w ∈ (0,1)* and the $n_0(w)$ is divisible by 3 **OR** $n_1(w)$ is odd}:



The DFA for L = L1 ∩ L2 = {w | w ∈ (0,1)* and the $n_0(w)$ is divisible by 3 **AND** $n_1(w)$ is odd}:



The DFA for L = L1 - L2 = {w | w ∈ (0,1)* and the $n_0(w)$ is divisible by 3 **AND** $n_1(w)$ is **NOT** odd}: