Lecture 2 Summary

This lecture starts where Lecture 1 leaves off with DFA, **Deterministic Finite Automata**. First half of the lecture focuses on proving that the union and intersection of two regular languages are regular.

This book defines **regular language** as any language that is recognized by some finite automaton.

Explanation of *Union* of two regular languages

Union of $A, B = \{x \mid x \in A \text{ or } x \in B\}$

Theorem: If A and B are regular then $A \cup B$ is regular.

M_1 recognizes A

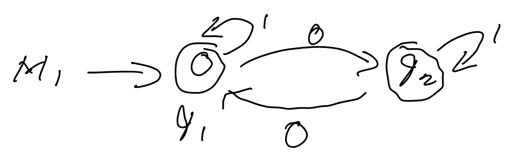
M_2 recognizes B

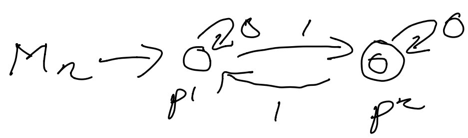
Want M_3 that recognizes $A \cup B$

M_3 Should Simulate M_1 and M_2 running in Parallel.

A = Set of binary string with an even # of 0's

B = Set of binary string with an odd number of 1's





M_3

$$Q = \{(q_1, p_1), (q_1, p_2), (q_2, p_1), (q_2, p_2)\}$$

Start state: (q_1, p_1)

Σ for M_3 is same as A, B

→ Next Page

Transition function

$\delta_3 (Q_3 \times \Sigma) \to Q_3$

$\delta_3 ((q_1, p_1), 0) \to (q_2, p_1)$
$\quad\quad\quad\quad\quad\quad 1 \to (q_1, p_2)$

$\delta_3 ((q_2, p_1), 0) \to (q_1, p_2)$
$\quad\quad\quad\quad\quad\quad 1 \to (q_2, p_1)$

Final state $= \{(q_1, p_1), (q_1, p_2), (q_2, p_2)\}$

Ex. str = 01010011 $\in A$
not in B
$\in A \cup B$

Start state: (q_1, p_1)

$\delta_3 ((q_1, p_1), 0) \to (q_2, p_1)$

$\delta_3 ((q_2, p_1), 1) \to (q_2, p_2)$

$\delta_3 ((q_2, p_2), 0) \to (q_1, p_2)$

$\delta_3 ((q_1, p_2), 1) \to (q_1, p_1)$

$\delta_3 ((q_1, p_1), 0) \to (q_2, p_1)$

$\delta_3 ((q_2, p_1), 0) \to (q_1, p_1)$

$\delta_3 ((q_1, p_1), 1) \to (q_1, p_2)$

$\delta_3 ((q_1, p_2), 1) \to (q_1, p_1)$

This was also assigned as a homework problem in the lecture.

(q_1, p_1) is an accept state of M_3

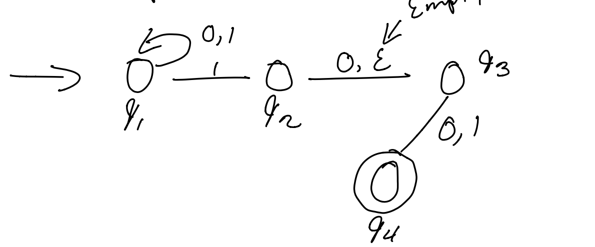
$\therefore$ M_3 recognizes the string 01010011, which is in A but not in B

Transition function takes in the current (and/or start) state of the machine and the alphabet, then it determines the next state the machine will go. **Example of transition function is used to complete the homework problem assigned by the professor in the lecture. The assigned problem's solution is included in the previous page.**

The second half of the lecture focuses on NFA, **Nondeterministic Finite Automata**. A proper definition of NFA was not given in this lecture, instead it was compared to "Many Worlds Interpretation" from Quantum Theory/ Mechanics due to their similarities.

> Many Worlds Interpretation states that for every decision we make, there exists a reality where we did not make that decision. Nondeterministic Finite Automata (NFA) works similarly. A handwritten example is given below.

Example of NFA



Start state: q_1

Alphabet (Σ): 0, 1, ε ← empty string

Final state: q_4

When the machine start at q_1, the machine could either stay in q_1 or go to q_2. Since this is a NFA, the machine would go to both. A new machine would be spawned, and one machine would would stay in q_1 and the other would proceed to next state in the machine.

Furthermore, we are introduced to new alphabet, epsilon (ε), this represents an empty string. When the machine is in q_1 and encounters a 1 in the languages, the machine can go to q_2 or q_3, because empty string can take the machine to the next state without looking at the next character in the language.