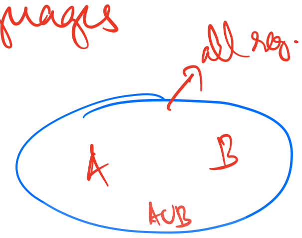
<u>Theorem 1.25</u>: The class of regular languages is closed under the union operation.



In other words, if A_1 and A_2 are regular, it means that $A_1 \cup A_2$ is regular.

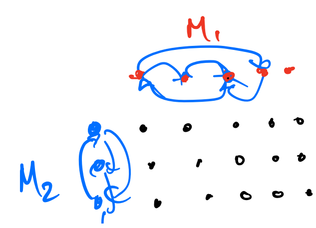
Suppose A_1 and A_2 are regular, this means that there exist DFA's M_1, M_2 such that $L(M_1) = A_1$ and $L(M_2) = A_2$. Can we build a DFA M such that $L(M) = A_1 \cup A_2$?

<u>Idea 1</u>: Combine M_1 and M_2 such that first M_1 runs on the input, followed by M_2.

Does not work. We cannot go back and read the string again. 01001

<u>Moral</u>: We need to keep track of the workings of M_1 and M_2 at the same time. 01011

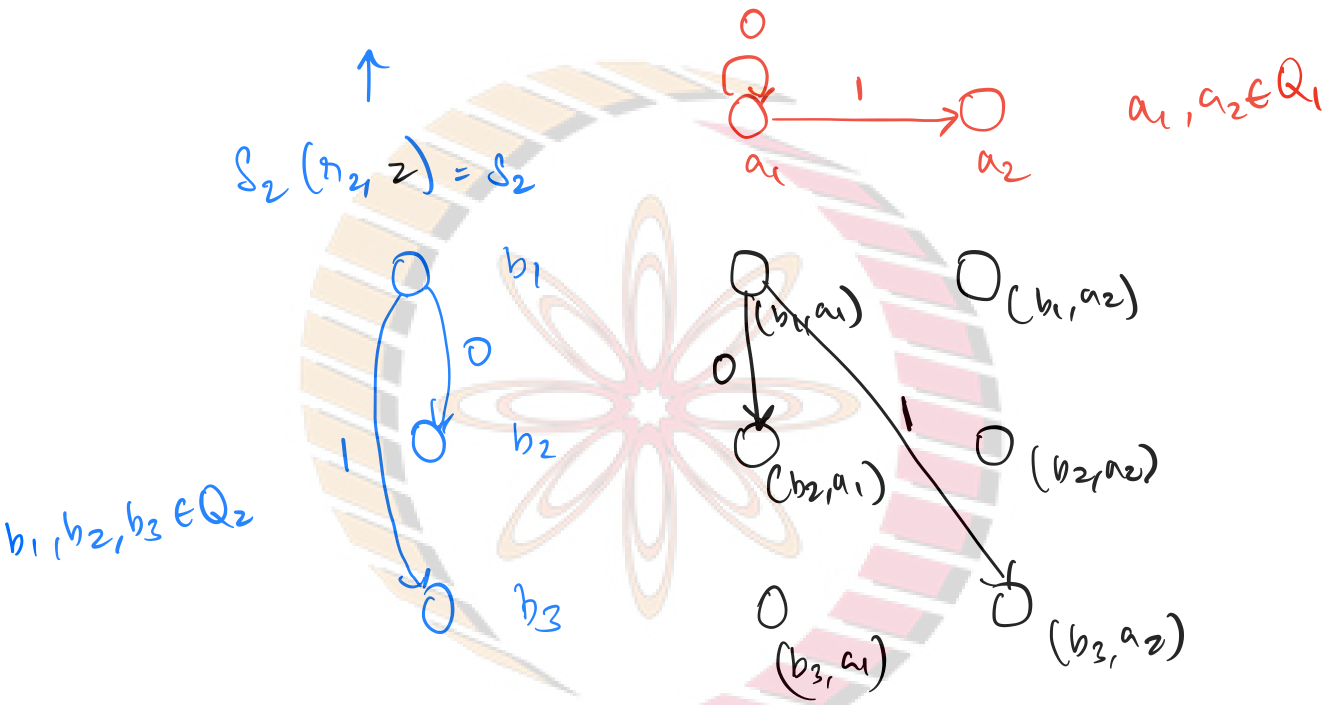
How can we accomplish this?



Idea: Cartesian Product. DFA whose states are of the form (r, s).

$$\delta((r_1, r_2), z) = (s_1, s_2) \leftarrow \delta_1(r_1, z) = s_1$$

$$\delta_2(r_2, z) = s_2$$



Proof: We have A_1, A_2 regular languages. Let M_1, M_2 be DFA's such that $L(M_1) = A_1$ and $L(M_2) = A_2$. Assume $A_1, A_2 \subseteq \Sigma^*$.

$$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$$

$$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$$

Goal: Need to build $M = (Q, \Sigma, \delta, q_0, F)$

such that $L(M) = A_1 \cup A_2 = L(M_1) \cup L(M_2)$

(1) $Q = \{ (r_1, r_2) \mid r_1 \in Q_1 \text{ and } r_2 \in Q_2 \}$

$= Q_1 \times Q_2$

(2) Σ is the same alphabet of A_1 and A_2.

If A_1 and A_2 are not over the same alphabet, let $\Sigma = \Sigma_1 \cup \Sigma_2$.

(3) δ needs to keep track of δ_1 and δ_2.

$$\delta : (Q \times \Sigma) \to Q$$

$$\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$$

(4) Starting State : $q_0 = (q_1, q_2)$

Starting M_2.

Starting of M_1

(5) Accepting States : We accept w if it ends at (r_1, r_2) if either $r_1 \in F_1$ or $r_2 \in F_2$.

$$F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ or } r_2 \in F_2\}$$

$$= (F_1 \times Q_2) \cup (Q_1 \times F_2)$$

Q: Why not $F_1 \times F_2 = \{(r_1, r_2) \mid r_1 \in F_1 \text{ AND } r_2 \in F_2\}$?

Correctness of the proof is straightforward.

Let $w \in A_1 \cup A_2 \implies w \in A_1$ or $w \in A_2$.

Let $w \in A_1$. Then there is a sequence of states $s_0, s_1, s_2, \ldots s_n$ such that

(1) $s_0 = q_1$

(2) $\delta(s_{i-1}, w_i) = s_i$

(3) $s_n \in F_1$

Condition for M_1 to accept w.

When M reads w, it will end in (s_n, r)

When M reads w, then M ends in $\underbrace{F_1 \times Q_2}_{\subseteq F}$. Similarly for $\underbrace{w \in A_2}_{w \in L(M_2)}$ as well.

Q: Is it enough to show that for all $w \in A_1 \cup A_2$, w is accepted by M?

Does this mean that $L(M) = A_1 \cup A_2$?

No. This only means that $A_1 \cup A_2 \subseteq L(M)$.

We also need to show that $L(M) \subseteq A_1 \cup A_2$.

Theorem: Regular languages are closed under intersection.

Q: How can we show this?

TRY.

(1) Modify the above construction.

(2) We have seen that reg. languages are closed under union and complement. Can we combine this?

Theorem 1.26: Regular languages are closed under concatenation operation.

If A_1 and A_2 are regular, then $A_1 \circ A_2$ is regular.

0110 001 11

Can we try to simulate M_1 and M_2 on two pieces of the input? We need to decide on which split to choose. The input w could be split anywhere. In DFA's we have only one chance and cannot go back and try other splits. This leads to the introduction of nondeterminism.

Next : Nondeterministic Finite Automata (NFA).