Regular Languages

Recall: DFA $M = (Q, \Sigma, \delta, q_0, F)$

$F \subseteq Q$ set of accepting states

alphabet

set of states

$q_0 \in Q$ initial state

$\delta : Q \times \Sigma \to Q$ transition function

Today: Operations on regular languages & corresponding automata

A non-empty string $a_0 \cdots a_{n-1}$, where $a_i \in \Sigma$ for each $0 \le i \le n-1$ is said to be accepted by a DFA $M = (Q, \Sigma, \delta, q_0, F)$ iff there exist $s_0, \dots, s_n \in Q$ s.t. $s_0 = q_0$, and for each $0 \le i \le n$, $\delta(s_i, a_i) = s_{i+1}$, and $s_n \in F$.

Otherwise, it is said to be rejected by M.

(The empty string is accepted if $q_0 \in F$, rejected otherwise)

Given a DFA M, the language of M, denoted $\mathcal{L}(M)$, is the language recognized by M, i.e., the set of all strings accepted by M.

A language recognized by a DFA is said to be regular.

We will represent the class of regular languages by Reg

Saw a couple of languages in Reg

More examples:

$\mathcal{L}_{emp} = \phi$ $\qquad\qquad \mathcal{L}_{triv} = \{\varepsilon\}$

$\mathcal{L}_{7a}$ = The set of all strings over $\Sigma = \{a\}$ of length divisible by 7

$\mathcal{L}_{ab}$ = The set of all strings over $\Sigma = \{a, b\}$ ending in abb.

A language is just a set of strings.
So one can perform set operations like union, intersection etc.
Which of these operations, if any, is Reg closed under?

Complementation:

We saw that if $\mathcal{L}$ is regular, so is $\overline{\mathcal{L}}$.

If $M = (Q, \Sigma, \delta, q_0, F)$ recognizes $\mathcal{L}$,

$M' = (Q, \Sigma, \delta, q_0, Q \setminus F)$ recognizes $\overline{\mathcal{L}}$.

Exercise: Prove this claim!

So Reg is closed under complementation.

Intersection:

If A and B are regular (s.t. $A = \mathcal{L}(M_1)$ and $B = \mathcal{L}(M_2)$),

is $A \cap B$ regular?

Suppose $A \cap B$ is indeed regular, recognized by a DFA M.

What strings does M accept?

Key idea: Run M_1 and M_2 simultaneously on the input word
If both accept, then accept, otherwise reject

Do a cross-product of the machines

$M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1)$ $\qquad$ $M_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2)$

$M = (Q, \Sigma, \delta, q_0, F)$

$Q = Q_1 \times Q_2 = \{(q, q') \mid q \in Q_1, q' \in Q_2\}$

$q_0 = (q_0^1, q_0^2)$ $\qquad$ $F = \{(q, q') \mid q \in F_1, q' \in F_2\}$

$$\delta((q, q'), a) = (\delta_1(q, a), \delta_2(q', a))$$

Exercise: Prove that $\mathcal{L}(M) = \mathcal{L}(M_1) \cap L(M_2)$.

Union :

If A and B are regular (s.t. $A = \mathcal{L}(M_1)$ and $B = \mathcal{L}(M_2)$),

is $A \cup B$ regular?

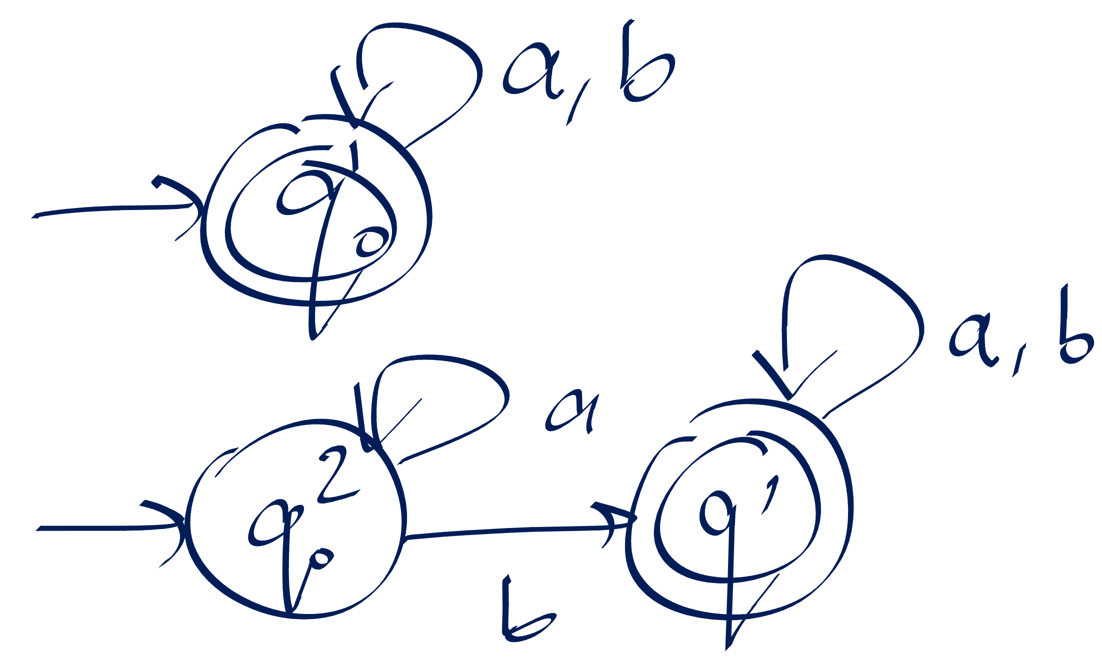
Suppose $A \cup B$ is indeed regular, recognized by a DFA M.

What strings does M accept?

Key idea: Run M_1 and M_2 simultaneously on the input word. If either accepts, then accept, otherwise reject

$M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1)$ $\quad$ $M_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2)$

$M = (Q, \Sigma, \delta, q_0, F)$



q_0 same $\quad$ Q same

δ same

$$F = \{ (q, q') \mid \underset{\substack{\text{and} \\ q' \in Q_2}}{q \in F_1} \text{ or } \underset{\substack{\text{and} \\ q \in Q_1}}{q' \in F_2} \}$$

Exercise: Prove that $L(M) = L(M_1) \cup L(M_2)$.

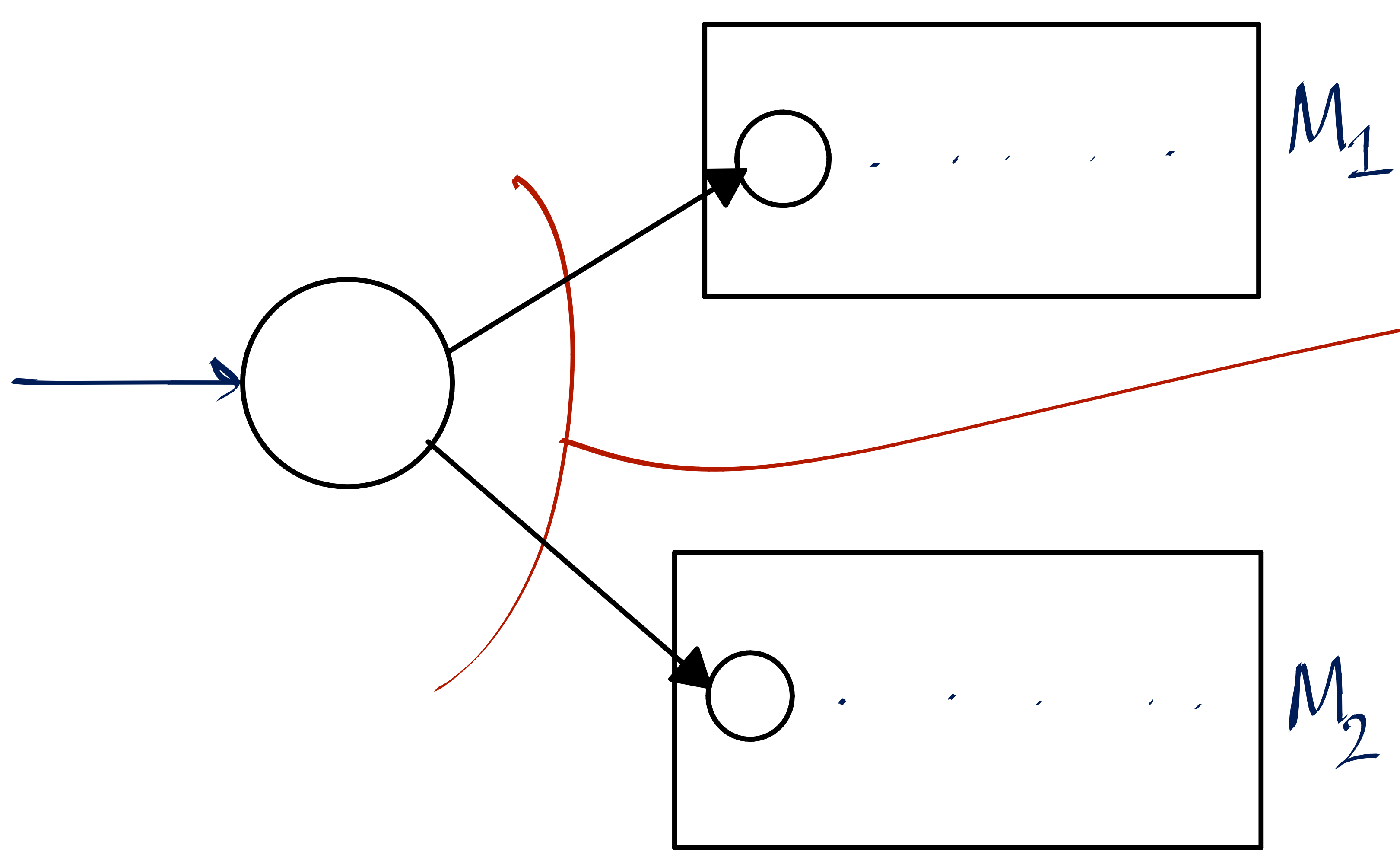
So Reg is closed under union, intersection, complement.

Aside:

What if I wanted a direct automaton construction for union?

M accepts strings accepted by either M_1 or M_2 or both.

If I could do the following, then I get M



But what letters label these transitions?

Now, we can consider two new kinds of ("regular") operations.

Concatenation:

If A and B are regular (s.t. $A = \mathcal{L}(M_1)$ and $B = \mathcal{L}(M_2)$), is $A \circ B$ regular, where

$$A \circ B = \{ xy \mid x \in A,\ y \in B \} \ ?$$

Star: If A is regular (s.t. $A = \mathcal{L}(M)$), is

$$A^* = \{ w_1 w_2 \dots w_n \mid n \geq 0,\ \text{each } w_i \in A \} \text{ regular?}$$

Both these operations require the machine to "know" where a "relevant" substring ends, so as to check membership in the appropriate regular language. How can it know that?