NONDETERMINISM

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

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

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

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

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

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

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.

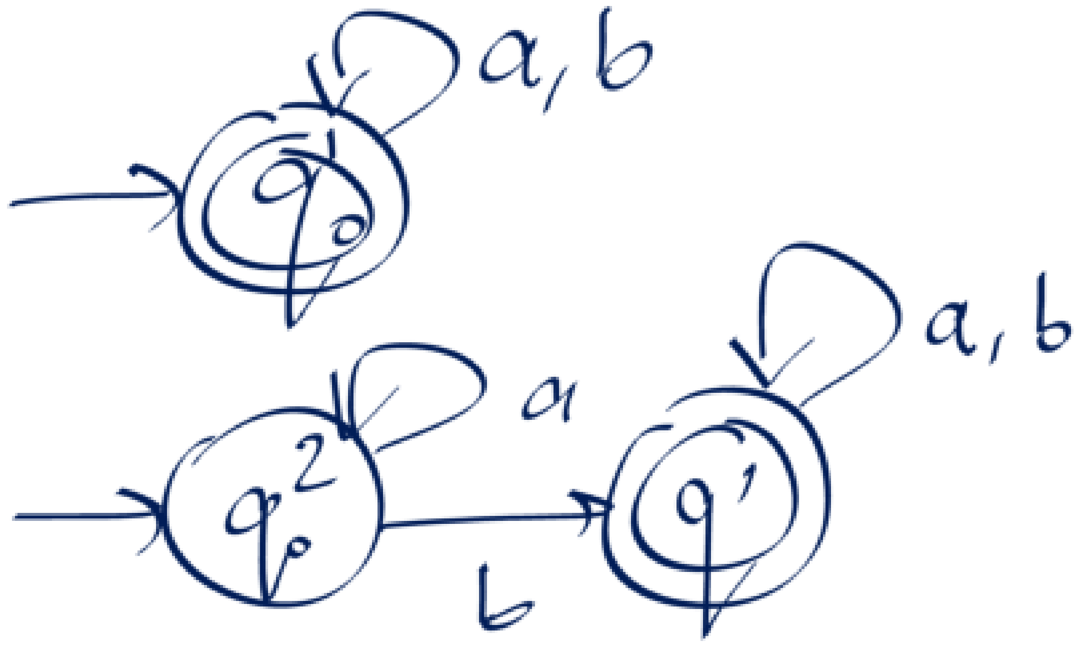
What strings does M accept ?

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

$M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1)$

$M_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2)$

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



q_0 same $\qquad Q$ 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 $L(M) = L(M_1) \cup L(M_2)$.

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

$$A^* = \{ \omega_1 \omega_2 \dots \omega_n \mid n \geq 0, \text{ each } \omega_i \in A \}$$ 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?

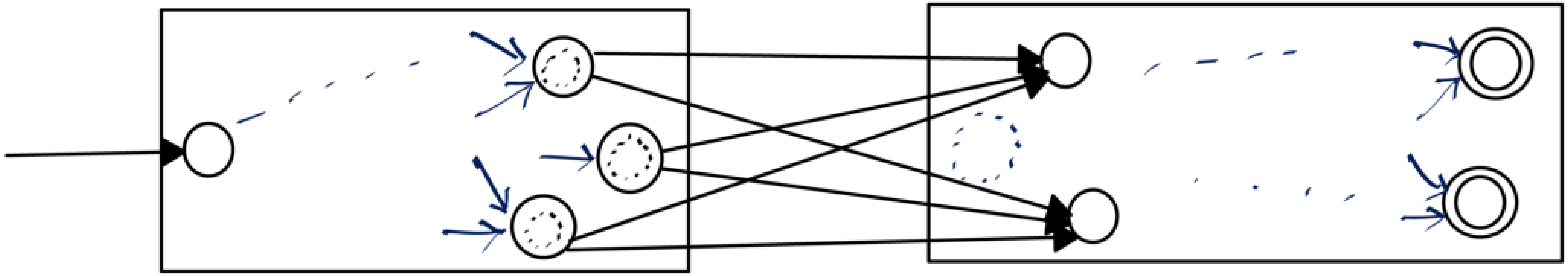
Reg is closed under union, intersection, and complementation

Concatenation:

If A and B are regular (s.t. $A = \mathcal{L}(M_1)$ and $B = \mathcal{L}(M_2)$), is $A \circ B = \{ xy \mid x \in A, y \in B \}$ regular?

Suppose $A \circ B$ is regular.

What does a DFA recognizing $A \circ B$ look like?



Consider the following languages over $\Sigma = \{a, b\}$

L_A : all strings containing at least one a (M_1)

L_B : all strings containing at least one b (M_2)

What do M_1, M_2, and M look like?

The machine needs to "know" when a "relevant" substring ends, and check membership in the appropriate language accordingly.

How can it know such a thing? Magic!

Suppose the machine could correctly guess when the substring x ends and y begins, s.t. $x \in A$ and $y \in B$.

Then we add the transitions between the "appropriate" states, and done!

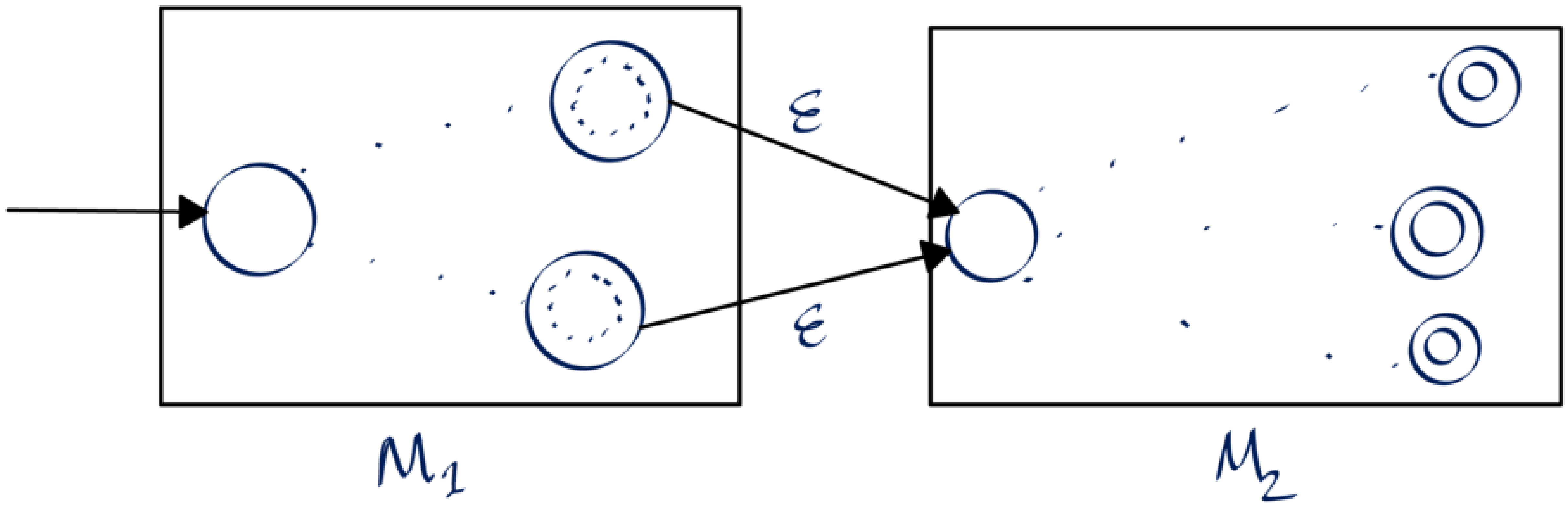
The question is: what labels do these transitions take on?

Must not affect the behaviour of M_1 and M_2,

but still allow this "magically correct" guess!

We move, therefore, to an extended model of computation,

a nondeterministic finite-state automaton (NFA).


ε
ε
M1
M2

DFA	NFA
$M = (Q, \Sigma, \delta, q_0, F)$	$M = (Q, \Sigma \cup \{\varepsilon\}, \Delta, Q_0, F)$
Q: finite set of states	"
Σ: alphabet	$\Sigma \cup \{\varepsilon\} = \Sigma_\varepsilon$
δ: transition function $\delta : Q \times \Sigma \to Q$	Δ: transition relation $\Delta \subseteq Q \times \Sigma_\varepsilon \times Q$
q_0: initial state $\in Q$	Q_0: Set of initial states $\subseteq Q$
F: set of accepting states $\subseteq Q$	"
M accepts a word w iff the run of M on w terminates in a final state from F.	M accepts a word w iff M has at least one run on w which terminates in a state $\in F$.