BU CS 332 – Theory of Computation

Lecture 3:

- Equivalence of NFAs and DFAs
- Closure under regular operations

Reading:

Sipser Ch 1.1-1.2

Ran Canetti

September 10, 2020

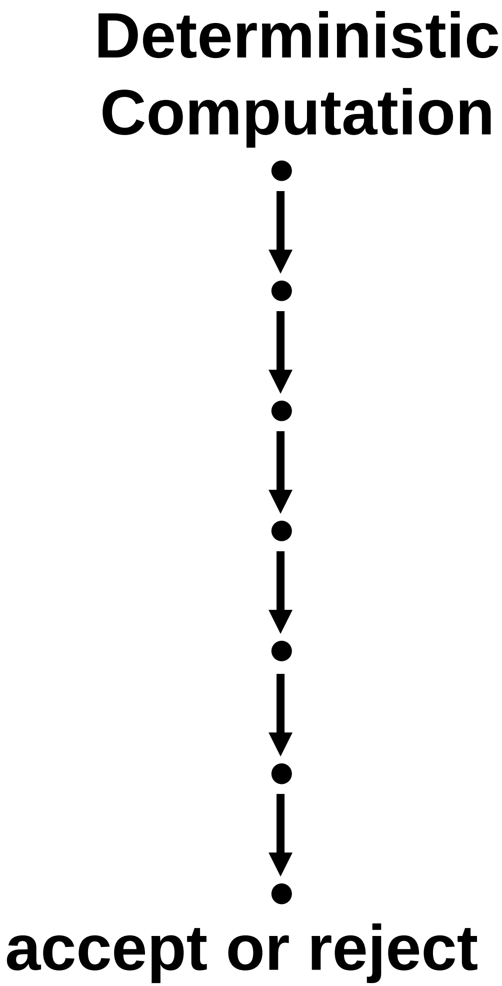
Formal Definition of a NFA

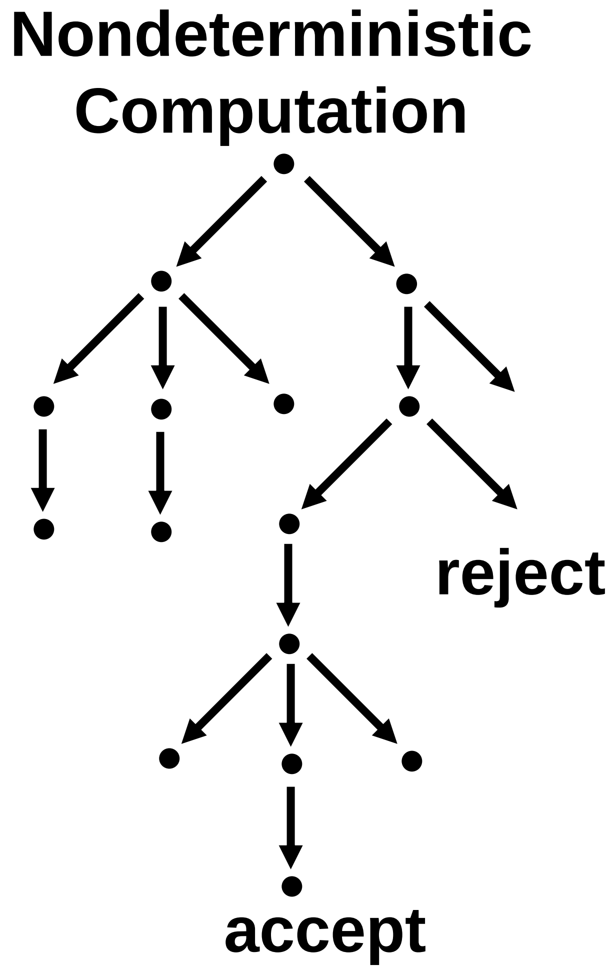
An NFA is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$

- Q is the set of states
- Σ is the alphabet
- $\delta : Q \times \Sigma_\varepsilon \rightarrow P(Q)$ is the transition function
- $q_0 \in Q$ is the start state
- $F \subseteq Q$ is the set of accept states

M ***accepts*** a string w if **there exists** a path from q_0 to an accept state that can be followed by reading w.

DFAs vs. NFAs





Ways to think about nondeterminism

- **(restricted) parallel computation**
- **tree of possible computations**
- **guessing and verifying the "right" choice**

Are NFAs more powerful than DFAs?

- There exist languages which require strictly more states to recognize with DFA than with NFA.

- Are there languages that can be recognized by an NFA and still cannot be recognized by *any* DFA (with any # of states)?

Are NFAs more powerful than DFAs?

- There exist languages which require strictly more states to recognize with DFA than with NFA.

- Are there languages that can be recognized by an NFA and still cannot be recognized by *any* DFA (with any # of states)?

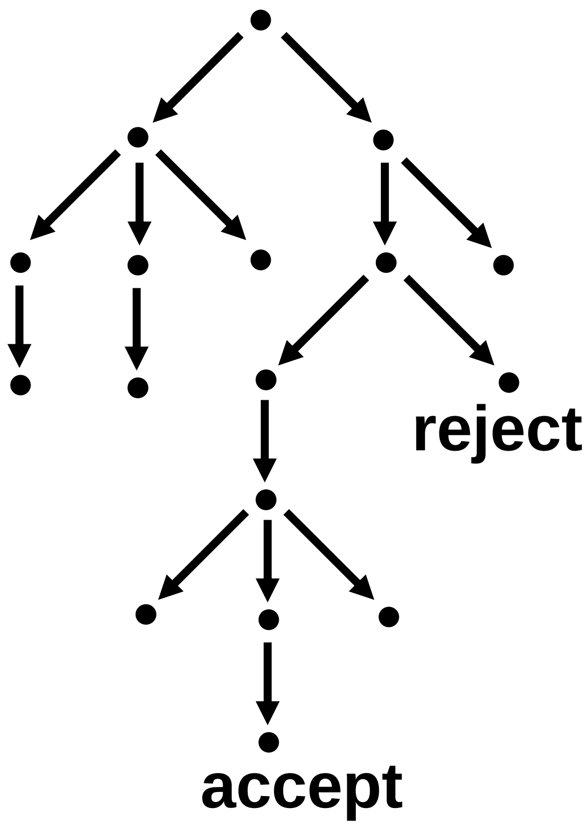
Theorem: For every NFA N, there is a DFA M such that $L(M) = L(N)$

Corollary: A language is regular if and only if it is recognized by an NFA

Equivalence of NFAs and DFAs (Proof)

Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA

Goal: Construct DFA $M = (Q', \Sigma, \delta', q_0', F')$ recognizing $L(N)$



Intuition: Run all threads of N in parallel, maintaining the set of states where all threads are.

Equivalence of NFAs and DFAs (Proof)

Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA

<u>Goal:</u> Construct DFA $M = (Q', \Sigma, \delta', q_0', F')$ recognizing $L(N)$

Intuition: Run all threads of N in parallel, maintaining the set of states where all threads are.

More precisely:

$Q' = P(Q)$

$\delta'(q', \sigma) = q''$, where

$q'' = \{q \in Q \mid q$ is reachable from a state $r \in q'$ by either reading σ, or an ϵ move$\}$

Or in other words: $q'' = \cup_{r \in q'} \{E(q) \mid \delta(r, \sigma) = q\}$

$q'_0 = \{q_0\}$, $F' = \{r \subseteq Q \mid r$ contains a state $q \in F\}$

$\{E(q_0)\}$

$E(q) = \{r \mid r$ is reachable from

Proving the Construction Works

q via 0 or more ε moves

Claim: For every string w, running M on w leads to state

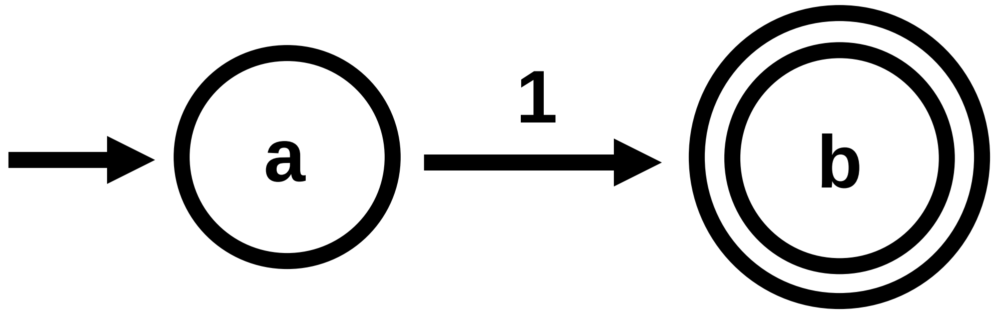
$$\{q \in Q \mid \text{There exists a computation of } N \text{ on input } w \text{ ending at } q\}$$

Proof idea: **"**By construction"

More formally, by induction on $|w|$.

NFA -> DFA Example

NFA:



DFA: states: $\{\emptyset\}$, $\{a\}$, $\{b\}$, $\{a,b\}$

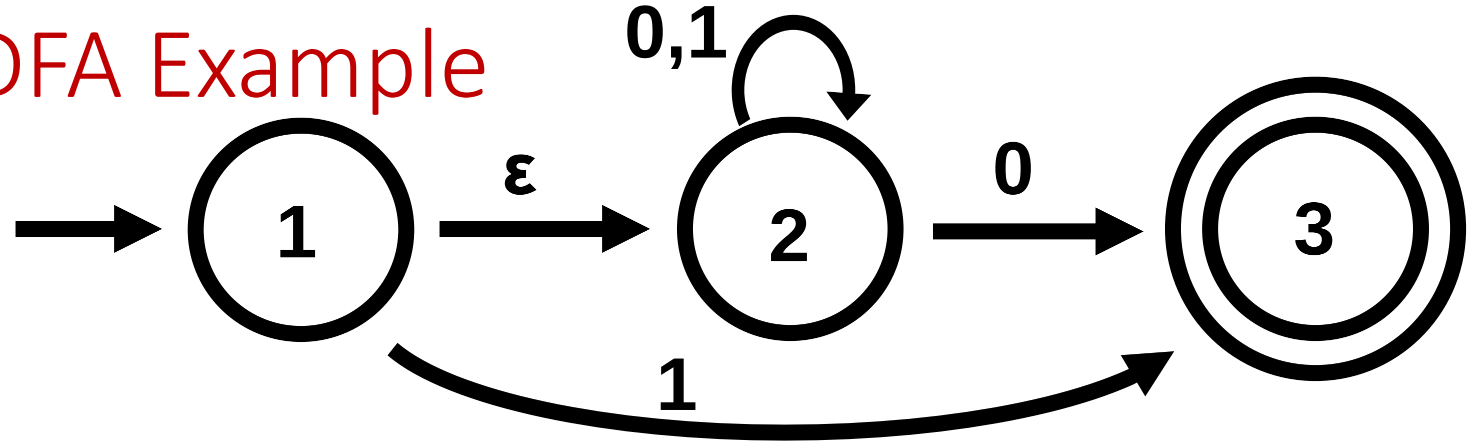
δ'	0	1
$\emptyset$	$\emptyset$	$\emptyset$
a	$\emptyset$	b
b	$\emptyset$	$\emptyset$
a,b	$\emptyset$	$\emptyset$

$q'_0 = \{a\}$

$F' = \{b\}, \{a,b\}$

NFA -> DFA Example

NFA:



DFA:

δ'	0	1
$\emptyset$	$\emptyset$	$\emptyset$
{1}	2	2,3
{2}	2,3	2
{3}	$\emptyset$	$\emptyset$
{1,2}	2,3	2,3
{2,3}	2,3	2,3
{1,3}	2,3	2,3
{1,2,3}	2,3	2,3

Can we make the blowup in # states smaller?

Subset construction converts an n state NFA into a 2^n-state DFA

Could there be a construction that always produces, say, an n^2-state DFA?

Theorem: For every $n \geq 1$, there is a language L_n such that

1. There is an $(n+1)$-state NFA recognizing L_n.
2. There is no DFA recognizing L_n with fewer than 2^n states.

Conclusion: For finite automata, nondeterminism provides an exponential savings over determinism (in the worst case).

What does class of regular languages look like?

- We saw that it's a pretty robust set of languages…
(L(NFAs) = L(DFAs)

- What about closure with respect to natural operations?

- Regular Operations

An Analogy (can ignore...)

In algebra, we try to identify operations which are common to many different mathematical structures

Example: The integers $\mathbb{Z} = \{\ldots -2, -1, 0, 1, 2, \ldots\}$ are **closed** under

- Addition: $x + y$
- Multiplication: $x \times y$
- Negation: $-x$
- ...but NOT Division: $x \,/\, y$

We'd like to investigate similar closure properties of the class of regular languages

Regular operations on languages

Let $A, B \subseteq \Sigma^*$ be languages. Define

Union: $A \cup B =$ $\{x \mid x \in A \vee x \in B\}$

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

$A^n = \{x_1 \ldots x_n \mid \forall i, x_i \in A\}$

Star: $A^* =$ $\bigcup_{n \in \mathbb{N}} A^n$

Other operations

Let $A, B \subseteq \Sigma^*$ be languages. Define

Complement: $\bar{A} = \{x \mid x \notin A\}$

Intersection: $A \cap B = \{x \mid x \in A \wedge x \in B\}$

Reverse: $A^R = \{x_1 \dots x_n \mid \forall i \; x_i \in \Sigma, \wedge \; x_n \dots x_1 \in A\}$

Closure properties of the regular languages

Theorem: The class of regular languages is closed under all three regular operations (union, concatenation, star), as well as under complement, intersection, and reverse.

i.e., if A and B are regular, applying any of these operations yields a regular language

Proving Closure Properties

Complement

Complement: $\bar{A} = \{ w \mid w \notin A \}$

Theorem: If A is regular, then $\bar{A}$ is also regular

Proof idea:

- all accepting states become non-accepting
- all non-accepting states become accepting

$F' = Q - F$

Union

Union: $A \cup B = \{ w \mid w \in A \text{ or } w \in B\}$

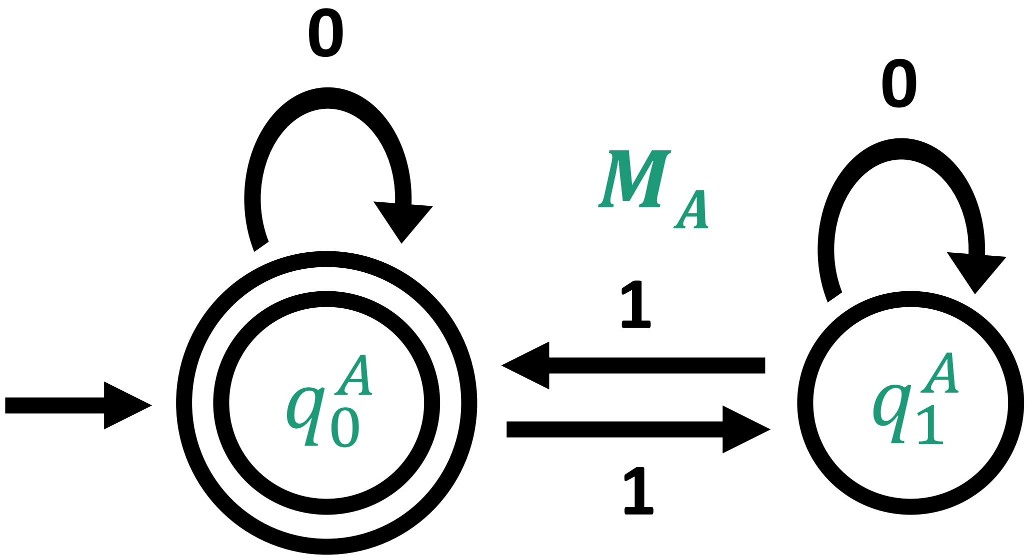
Theorem: If A and B are regular, then so is $A \cup B$

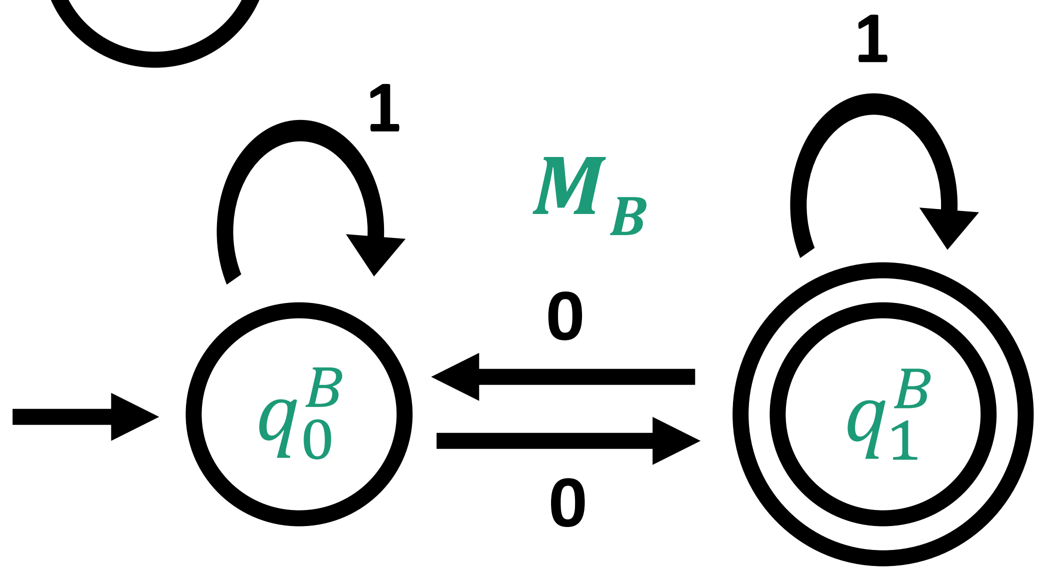
Proof:

Let $M_A = (Q_A, \Sigma, \delta_A, q_0^A, F_A)$ be a DFA recognizing A and
$M_B = (Q_B, \Sigma, \delta_B, q_0^B, F_B)$ be a DFA recognizing B

Goal: Construct a DFA $M = (Q, \Sigma, \delta, q_0, F)$
that recognizes $A \cup B$

Example





$M = ?$

Intersection

Intersection: $A \cap B = \{ w \mid w \in A \text{ and } w \in B\}$

Theorem: If A and B are regular, then so is $A \cap B$

Proof:

Let $M_A = (Q_A, \Sigma, \delta_A, q_0^A, F_A)$ be a DFA recognizing A and
$M_B = (Q_B, \Sigma, \delta_B, q_0^B, F_B)$ be a DFA recognizing B

Goal: Construct a DFA $M = (Q, \Sigma, \delta, q_0, F)$
that recognizes $A \cap B$

Intersection

Intersection: $A \cap B = \{ w \mid w \in A \text{ and } w \in B\}$

Theorem: If A and B are regular, then so is $A \cap B$

Another Proof:

$A \cap B = \overline{\bar{A} \cup \bar{B}}$

Operations on languages

Let $A, B \subseteq \Sigma^*$ be languages. Define

Regular Operations

- Union: $A \cup B$
- Concatenation: $A \circ B = \{ab \mid a \in A, b \in B\}$
- Star: $A^* = \{a_1 a_2 \ldots a_n \mid n \geq 0 \text{ and } a_i \in A\}$

Complement: $\bar{A}$

Intersection: $A \cap B$

Reverse: $A^R = \{a_1 a_2 \ldots a_n \mid a_n \ldots a_1 \in A\}$

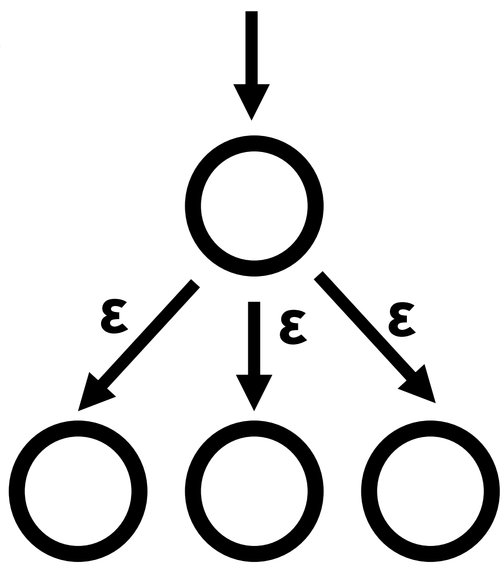
Theorem: The class of regular languages is closed under all six of these operations

Closure under Reverse

Theorem. The reverse of a regular language is also regular

Proof: Let L be a regular language and M be a DFA recognizing it. Construct an NFA M' recognizing L^R:

- Define M' as M with the arrows reversed.
- Make the start state of M be the accept state in M'.
- Make a new start state that goes to all accept states of M by ε-transitions.



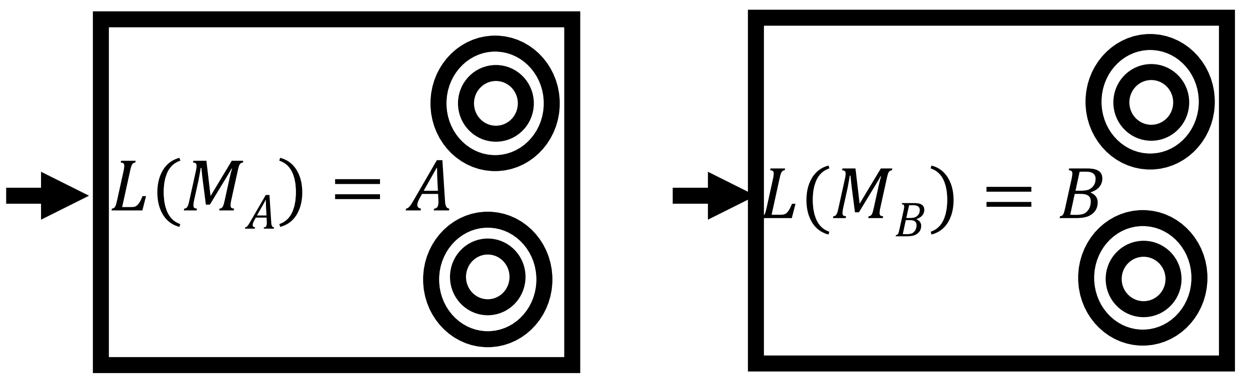
Closure under Concatenation

Concatenation: $A \circ B \ = \ \{ \, ab \mid a \in A$ and $b \in B \, \}$

Theorem. If A and B are regular, $A \circ B$ is also regular.

Proof: Given DFAs M_A and M_B, construct NFA by

- Connecting all accept states in M_A to the start state in M_B.
- Make all states in M_A non-accepting.



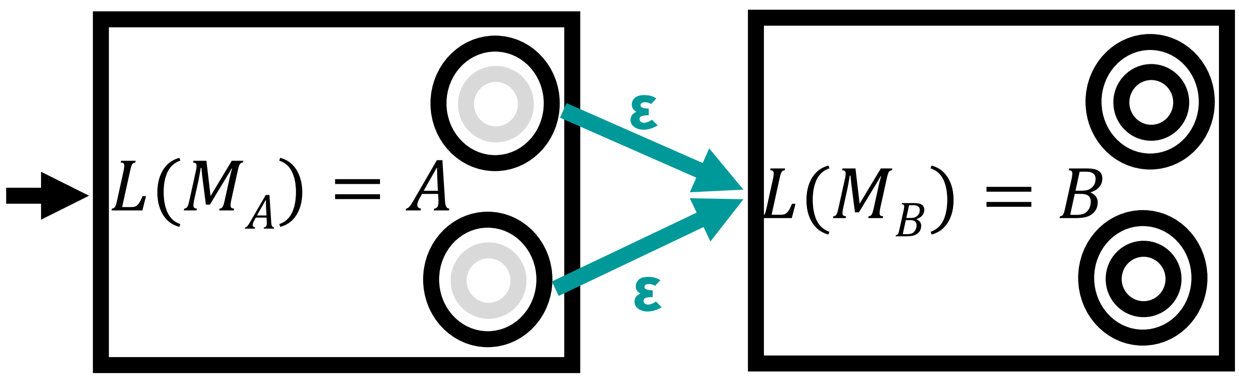
Closure under Concatenation

Concatenation: $A \circ B = \{ ab \mid a \in A \text{ and } b \in B \}$

Theorem. If A and B are regular, $A \circ B$ is also regular.

Proof: Given DFAs M_A and M_B, construct NFA by

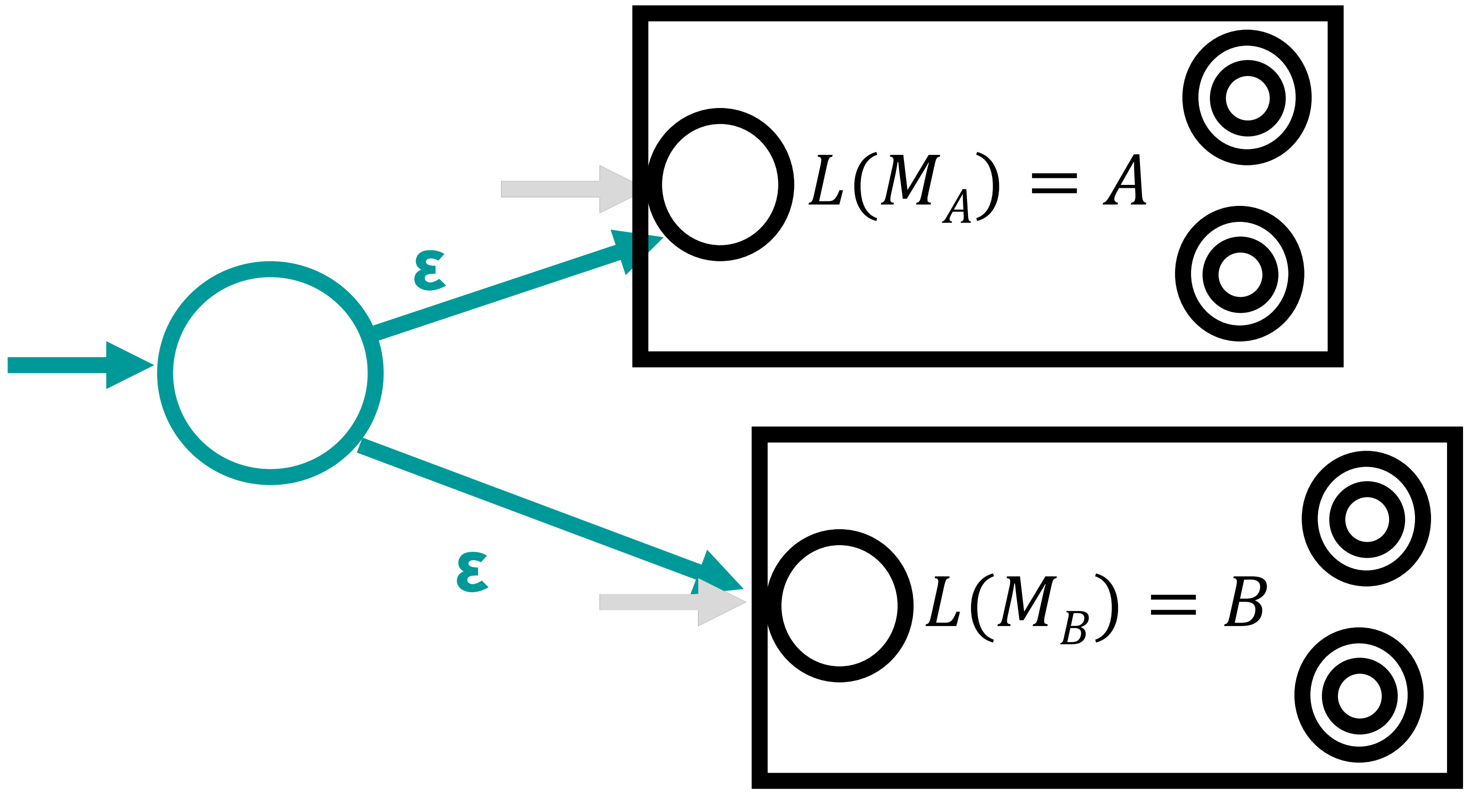
- Connecting all accept states in M_A to the start state in M_B.
- Make all states in M_A non-accepting.



Question: What happens if (instead) we unify the starting states of B with the ~~accepting~~ accepting states of A?

A Mystery Construction

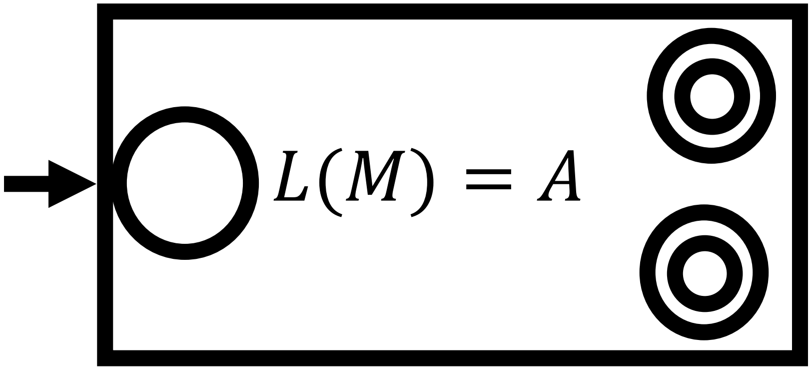
Given DFAs M_A recognizing A and M_B recognizing B, what does the following NFA recognize?



Closure under Star

Star: $A^* = \{ a_1 a_2 \ldots a_n \,|\, n \geq 0$ and $a_i \in A\}$

Theorem. If A is regular, A^* is also regular.



Closure under Star

Star: $A^* = \{ a_1 a_2 ... a_n | n \geq 0 \text{ and } a_i \in A\}$

Theorem. If A is regular, A^* is also regular.