Nondeterministic Finite Automata (NFA)

Lecture 4

Binghui Peng

Announcement

- Homework 1 came out, due in March 1st; check gradescope
- TA office hour location: AVW4160.

Formal Definition of NFAs

An **NFA** is defined as a 5-tuple $(Q, \Sigma, \Delta, s, F)$ where:

1. Q is a finite set of **states**.
2. Σ is a finite **alphabet**.
3. $\Delta : Q \times (\Sigma \cup \{e\}) \to 2^Q$ is the **transition function**.
4. $s \in Q$ is the **start state**.
5. $F \subseteq Q$ is the set of **final states**.

Acceptance: M accepts $x \in \Sigma^*$ if there is some sequence of transitions leading from s to a state in F.

Equivalence of NFAs and DFAs

Theorem: If L is accepted by an NFA, then L is also accepted by a DFA.

Proof Step 1: Remove e-transitions Given an NFA with e-transitions, we define an equivalent NFA Δ_1 with no e-transitions:

$$\Delta_1(q, \sigma) = \bigcup_{0 \leq i, j \leq n} \Delta(q, e^i \sigma e^j)$$

where $\Delta(q, e^i \sigma e^j)$ represents reaching a set of states by taking i e-steps, then σ, then j e-steps.

Proof Step 2: Subset Construction

Given an NFA $(Q, \Sigma, \Delta, s, F)$ with no e-transitions, we construct a DFA $(2^Q, \Sigma, \delta, \{s\}, F')$:

- **States:** Every state in the DFA corresponds to a **subset** of states in the NFA.
- **Transition Function:**

$$\delta(A, \sigma) = \bigcup_{q \in A} \Delta(q, \sigma)$$

- **Final States:**

$$F' = \{A \subseteq Q : A \cap F \neq \emptyset\}$$

Result: The DFA tracks all possible states the NFA could be in simultaneously. If the NFA has n states, the DFA has at most 2^n states.

Break

Closure property of regular languages (via NFA)

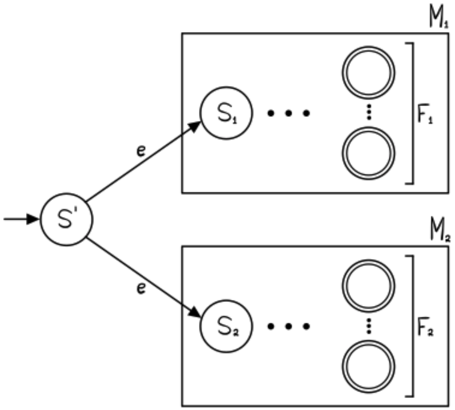
Definition: A language L is **regular** if there exists a DFA M such that $L = L(M)$.

Since DFAs and NFAs are equivalent:
Definition: A language L is **regular** if there exists an NFA M such that $L = L(M)$.

We will prove closure properties of regular languages using NFAs and keep track of the number of states required for each construction.

Closure Under Union: Intuition

If L_1 and L_2 are regular, we construct an NFA for $L_1 \cup L_2$ by creating a new start state that branches to the start states of M_1 and M_2 using e-transitions.



Closure Under Union: Formal Construction

Let L_1 have NFA $(Q_1, \Sigma, \Delta_1, s_1, F_1)$ with $|Q_1| = n_1$. Let L_2 have NFA $(Q_2, \Sigma, \Delta_2, s_2, F_2)$ with $|Q_2| = n_2$.

$L_1 \cup L_2$ is recognized by:

$$(\{s'\} \cup Q_1 \cup Q_2, \Sigma, \Delta', s', F_1 \cup F_2)$$

where:

- $\Delta'(q, \sigma) = \Delta_i(q, \sigma)$ for $q \in Q_i$.
- $\Delta'(s', e) = \{s_1, s_2\}$.

States: The resulting NFA has $n_1 + n_2 + 1$ states.

Closure Under Intersection

If L_1 and L_2 are regular, we can construct an NFA for $L_1 \cap L_2$ using the product construction, similar to DFAs.

Construction: Let $M_1 = (Q_1, \Sigma, \Delta_1, s_1, F_1)$ and $M_2 = (Q_2, \Sigma, \Delta_2, s_2, F_2)$. The NFA $M = (Q_1 \times Q_2, \Sigma, \Delta, (s_1, s_2), F_1 \times F_2)$ recognizes $L_1 \cap L_2$, where:

$$\Delta((q_1, q_2), \sigma) = \{(p_1, p_2) : p_1 \in \Delta_1(q_1, \sigma) \wedge p_2 \in \Delta_2(q_2, \sigma)\}$$

States: The resulting machine has $n_1 n_2$ states.

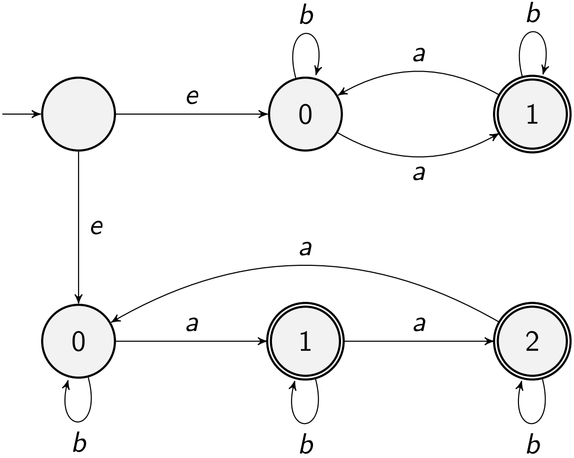
Regular Languages Closed Under Complementation

How do you complement a regular language?

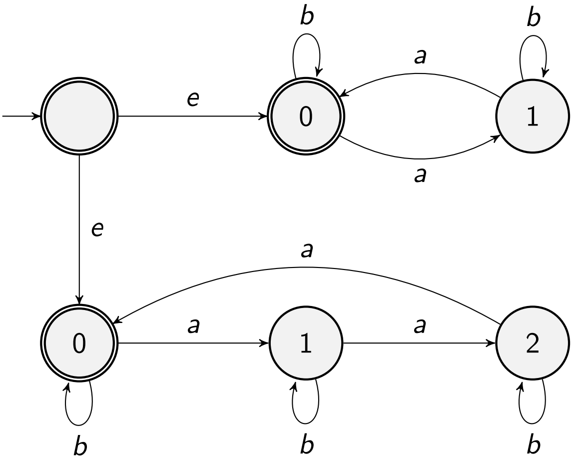
Caution: Swapping the final and non-final states **does not work** for an NFA.

Counterexample for NFA Complementation

$L = \{a^n : n \not\equiv 0 \pmod 6\}$



Final and Non-final States Swapped



Observation: This machine does not recognize $\overline{L}$.

Complement via DFA Conversion

To prove closure under complementation:

1. Start with an n-state NFA for L.
2. Convert it to an equivalent DFA with up to 2^n states.
3. Complement the DFA by swapping final and non-final states.
4. This gives a 2^n-state DFA, which is also a 2^n-state NFA for $\overline{L}$.

Efficiency: There are languages where an n-state NFA's complement requires an NFA with $\Omega(2^n)$ states.

Example: Exponential Blowup in Complementation

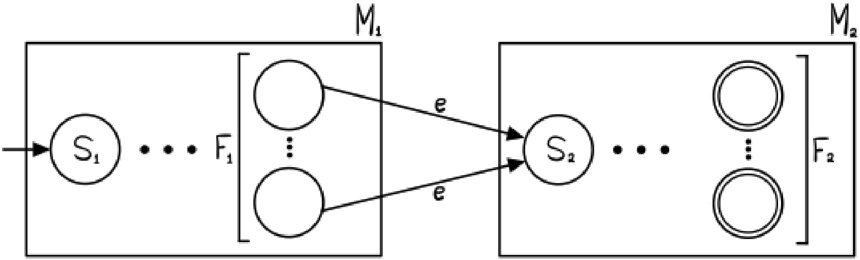
Let M_n be the product of the first n primes $p_1, p_2, \ldots, p_n$.

$$L_n = \{a^i \colon i \not\equiv 0 \pmod{M_n}\}$$

- There is an NFA for L_n of size $O(\sum p_i) = O(\frac{n^2}{\log n})$.
- Any NFA for $\overline{L_n} = \{a^i \colon i \equiv 0 \pmod{M_n}\}$ requires size $\Omega(p_1 p_2 \cdots p_n) = \Omega(e^{n \log n})$.

Closure Under Concatenation: Intuition

To recognize $L_1 \cdot L_2$, we add e-transitions from every final state of M_1 to the start state of M_2.



Closure Under Concatenation: Formal Construction

Let L_1 have NFA $(Q_1, \Sigma, \Delta_1, s_1, F_1)$ and L_2 have NFA $(Q_2, \Sigma, \Delta_2, s_2, F_2)$.

$L_1 \cdot L_2$ is recognized by:

$$(Q_1 \cup Q_2, \Sigma, \Delta', s_1, F_2)$$

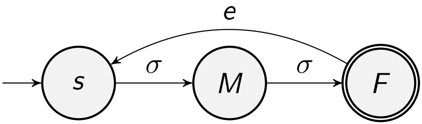
where:

$$\Delta' = \Delta_1 \cup \Delta_2 \cup \{\Delta(f, e) = s_2 : f \in F_1\}$$

States: The resulting NFA has $n_1 + n_2$ states.

Closure Under Kleene Star: First Attempt

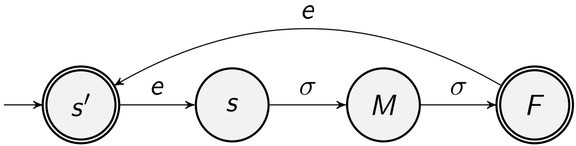
Idea: Add an e-transition from every final state of M back to the start state s.



Problem: This does not automatically accept the empty string e, which is required for L^*.

Closure Under Kleene Star: Correct Construction

To correctly recognize L^*, we add a new start state s' which is also an accepting state.



States: The resulting NFA has $n + 1$ states.

Summary of Closure Properties

The following table summarizes the state complexity for DFA and NFA constructions. (L_i has n_i states).

Closure Property	**DFA**	**NFA**
$L_1 \cup L_2$	$n_1 n_2$	$n_1 + n_2 + 1$
$L_1 \cap L_2$	$n_1 n_2$	$n_1 n_2$
$L_1 \cdot L_2$	Complex	$n_1 + n_2$
$\overline{L}$	n	$\sim 2^n$
L^*	Complex	$n + 1$