CS304: Automata and Formal Languages

Lec 11

Pumping Lemma and Closure and Decision Properties of Regular Languages

Rachit Nimavat

August 26, 2025

Outline

NFA vs DFA

NFA

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

$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \mapsto 2^Q$

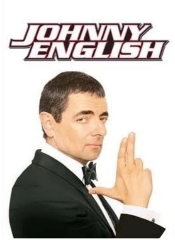
Multiple Transitions. Possibly multiple outgoing arrows for same symbol
Zero Transitions. Possibly no outgoing arrow for a symbol (that path "dies" has license to kill)
ε-**Transitions.** Can change state without consuming an input symbol

DFA

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

$\delta : Q \times \Sigma \mapsto Q$

No such powers!



Theorem 1.1

A language is recognized by an NFA if and only if it is recognized by a DFA.

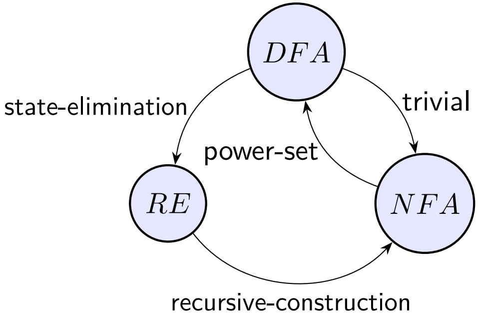
Regular Language

Defn: A language L is called a regular language if and only if there exists a regular expression R that describes it: $L = L(R)$

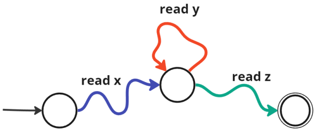
Theorem 1.2 (Kleene's Theorem)

All these definitions are equivalent:

- $\exists$ *a DFA* D *such that* $L = L(D)$
- $\exists$ *an NFA* A *such that* $L = L(A)$
- $\exists$ *an RE* R *such that* $L = L(R)$



Pumping Lemma



Lemma 2.1

For each regular language L, there is a constant p (the pumping length) such that any string $s \in L$ with $|s| \geq p$ can be divided into three parts, $s = xyz$, such that

i. $|y| > 0$ *("looped-string" y is not empty)*

ii. $|xy| \leq p$ *(the loop starts within the first p characters)*

iii. $\forall i \geq 0$, *the string* $xy^iz \in L$ *(we can pump the loop zero, one, or many times, and the resulting string must still be accepted)*

How to use Pumping Lemma?

Goal: Prove that a language L is irregular.
Assume that L is regular.

Pumping Lemma: Player-1

Us: Player-2

1. Provides 'pumping length' p
2. Cleverly choose $s \in L$ such that $|s| \geq p$
3. Provides a partition $s = xyz$ such that $|y| > 0$ and $|xy| < p$
4. WIN by choosing $i \geq 0$ such that $xy^iz \notin L$

If $xy^iz \notin L$, then we have a contradiction. Thus, our assumption that L is regular must be false. Therefore, L is irregular.

Note: For this to work, we must have a winning strategy for each p and for each partition $s = xyz$ (that satisfies $|y| > 0$ and $|xy| < p$).

Example

Prove that $L = \{0^n 1^n \mid n \geq 0\}$ is not regular

Assume for contradiction that L is regular

From pumping lemma, there is a constant p with "looping property"

Let's pick a string $s = 0^p 1^p$. Notice that $s \in L$

Consider its decomposition $s = xyz$ given by the pumping lemma

$|xy| \leq p \implies y = 0^{|y|}$ y consists of all zeroes

Choose $i = 2$. From pumping lemma, $s_2 = xy^2 z = xy \cdot y \cdot z = 0^{p+|y|} 1^p \in L$

Contradiction! since $|y| > 0$)

Example II

Prove that $L = \{a^n \mid n$ is a perfect square$\}$ is not regular

Assume for contradiction that L is regular

From pumping lemma, there is a constant p with "looping property"

Let's pick a string $s = a^{p^2}$. Notice that $s \in L$

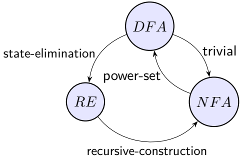
Consider its decomposition $s = xyz$ given by the pumping lemma

Pumping lemma guarantees $1 < y \leq p$

Choose $i = 2$. From pumping lemma, $s_2 = xy^2z$ has length $|s_2| = |s| + |y| \in [p^2 + 1, p^2 + p]$

Contradiction! since $p^2 < p^2 + 1 \leq p^2 + p < (p+1)^2$ and hence the length of s_2 is not a perfect square.

Closure Properties



Closure Properties: A set is closed under an operation if applying that operation to its elements results in elements that are also in the set

- Regular languages are closed under union Regular expressions!
- Regular languages are closed under complementation DFAs!
- Regular languages are closed under intersection De Morgan's Laws!
- Difference of regular languages is regular $A \setminus B = A \cap \overline{B}$
- Reversal of regular languages is regular Reverse DFA and add a new start state with ε-transitions to old accept states
- Kleene-star of regular languages is regular Regular expressions!
- Concatenation of regular languages is regular Regular expressions!
- Substitution of characters by strings (**Homomorphism**) in regular languages is regular Regular expressions!
- Inverse Homomorphism is also regular, but don't worry about that. See Chap 4.2 of ALC for details.

Example III

Prove that $K = \left\{w \in \{0,1\}^* \mid w \text{ has an equal number of 0's and 1's}\right\}$ is not regular.

Assume K is regular

Consider a known regular language $K' = 0^*1^*$

Then, by closure properties, $K \cap K' = \{0^n1^n \mid n \geq 0\}$ is also regular

Contradiction!

Example IV

Prove that $L = \{ab^nc^n + a^k(b+c)^* \mid k \neq 1$ and $n \geq 0\}$ over $\Sigma = \{a, b, c\}$ is irregular

Assume for contradiction that L is regular

From pumping lemma, there is a constant p with "looping property"

Let's pick a string $s \in L$ with $|s| \geq p$

- if s starts with a, then a valid decomposition is $x = \varepsilon$, $y = a$, $z = s[1:]$. Notice that for all $i \geq 0$, $s_i = xy^iz \in L$.
- if s does not start with a, then $s \in (b+c)^*$ and can be trivially pumped

No Contradiction!

Is L a regular language then? **NO!**

- Assume L is regular
- Know: $\{a^k(b+c)^* \mid k \neq 1\}$ is regular
- Closure Properties $\implies$ b^nc^n is regular. **Contradiction!**

Decision Properties

Three major questions:

1. Is the given regular language L empty?
   Assume we are given a DFA for L. If it has no accept states, then it is empty. Is this a sufficient condition? Is this a necessary condition? Can an empty language's DFA have accept states?
   L is empty iff the DFA has no **reachable** accept states.
2. Is the given regular language L finite?
   Assume we are given a NFA for L. L is infinite iff NFA has a 'loop' on a path to accept state. You'll learn an algorithm for this task in the Data Structures and Algorithms course.
3. Is the given regular language L equal to another regular language K?
   Check whether $(L \backslash K) \cup (K \backslash L)$ is empty or not.

HW: Can you determrine if a given NFA accepts some string? Lookup graph reachability

HW: Can you determine if a given NFA accepts all strings?

HW: Prove that the condition for equality is sufficient and necessary.