Regular Languages and Grammars: Equivalence, Non-Regularity, and Pumping Lemma

Proofs and Constructions

Khushraj

November 14, 2025

Overview

- Equivalence between Regular Grammars and Regular Languages
  - Construction: DFA to Regular Grammar
  - Proof of Correctness
  - Construction: Regular Grammar to NFA
  - Proof of Correctness
- Proof that $L = \{0^n 1^n \mid n \geq 0\}$ is Not Regular
- Statement of the Pumping Lemma for Regular Languages
- General Proof of the Pumping Lemma

Definition: Regular Grammars

A **right-linear grammar** has productions of the form:

- $A \to aB$
- $A \to a$
- $A \to \epsilon$

where A, B are non-terminals, a is a terminal, and ϵ is the empty string.
Left-linear grammars are symmetric ($A \to Ba$ or $A \to a$).
Regular grammars are either right- or left-linear.
We prove equivalence by bidirectional constructions.

Construction: From DFA to Regular Grammar

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$ accepting $L(M)$, construct right-linear grammar $G = (V, \Sigma, P, S)$:

- Non-terminals $V = Q$
- Start symbol $S = q_0$
- Productions P:
  - For each $\delta(q, a) = q'$, add $q \to aq'$
  - For each $f \in F$, add $f \to \epsilon$

Intuition: Derivations mimic paths in the DFA.

Proof of Correctness: DFA to Grammar

Subsets

- $L(G) \subseteq L(M)$: Any derivation starts at q_0, follows transitions, ends in $f \in F$ via ϵ. Traces an accepting path.
- $L(M) \subseteq L(G)$: For accepting path on $w = a_1 \dots a_n$ from $q_0 = q^{(0)}$ to $q^{(n)} \in F$:

$$S \Rightarrow q^{(0)} \rightarrow a_1 q^{(1)} \Rightarrow \cdots \Rightarrow a_1 \dots a_n q^{(n)} \Rightarrow a_1 \dots a_n$$

Thus, $L(G) = L(M)$.

Construction: From Regular Grammar to NFA

Given right-linear $G = (V, \Sigma, P, S)$ generating $L(G)$, construct NFA $M = (Q, \Sigma, \delta, q_0, F)$:

- States $Q = V \cup \{q_f\}$ (new final state)
- Start $q_0 = S$, Final $F = \{q_f\}$
- Transitions δ:
  - For $A \to aB$, $\delta(A, a)$ includes B
  - For $A \to a$, $\delta(A, a)$ includes q_f
  - For $A \to \epsilon$, $\delta(A, \epsilon)$ includes q_f

Intuition: States are non-terminals; transitions follow productions. (Left-linear: Symmetric or convert to right-linear.)

Proof of Correctness: Grammar to NFA

Subsets

- $L(M) \subseteq L(G)$: Accepting path from S to q_f on w corresponds to leftmost derivation.
- $L(G) \subseteq L(M)$: Leftmost derivation $S \Rightarrow^* \alpha A \beta \Rightarrow \alpha a \gamma \beta$ (via $A \to aC$) extends path from S to C on αa. Terminals/ϵ reach q_f.

Thus, $L(G) = L(M)$. Regular grammars generate exactly regular languages.

Proof: $L = \{0^n 1^n \mid n \geq 0\}$ is Not Regular

Assume L is regular. Let p be the pumping length. Choose $w = 0^p 1^p \in L$, $|w| = 2p \geq p$.
By Pumping Lemma: $w = xyz$ with $|xy| \leq p$, $|y| > 0$, $xy^k z \in L$ for all $k \geq 0$.
Since $|xy| \leq p$, x, y in 0^p prefix: $y = 0^m$ $(1 \leq m \leq p)$.
For $k = 2$: $xy^2 z = 0^{p+m} 1^p \notin L$ (unequal exponents).
Contradiction. Thus, L is not regular.

Statement of the Pumping Lemma for Regular Languages

Pumping Lemma: Let L be regular with pumping length $p \geq 1$. For every $w \in L$, $|w| \geq p$, there exist x, y, z s.t. $w = xyz$:

1. $|xy| \leq p$
2. $|y| > 0$
3. $xy^k z \in L$ for all $k \geq 0$

(Contrapositive used for non-regularity proofs.)

General Proof: Pumping Lemma for Regular Languages

Let L regular, accepted by DFA $M = (Q, \Sigma, \delta, q_0, F)$, $|Q| = n \geq 1$. Set $p = n$.
For $w \in L$, $|w| \geq p$: $w = a_1 \dots a_m$ $(m \geq n)$. Accepting run:
$q_0 \xrightarrow{a_1} q_1 \to \cdots \to q_m \in F$, $q_i = \delta(q_0, a_1 \dots a_i)$.
Pigeonhole: Among $q_0, \dots, q_n$ ($n+1$ states), $q_i = q_j$ for $0 \leq i < j \leq n$ $(j - i \geq 1)$.

General Proof: Pumping Lemma (cont.)

Set:

- $x = a_1 \dots a_i$ (to q_i)
- $y = a_{i+1} \dots a_j$ (loop, $|y| = j - i > 0$)
- $z = a_{j+1} \dots a_m$ (from $q_j = q_i$ to $q_m \in F$)

Then $|xy| = j \leq n = p$, $|y| > 0$.
For $k \geq 0$: $xy^k z$ path: prefix to q_i, k loops on y (back to q_i), suffix to $q_m \in F$.
Thus, $xy^k z$ accepted, $\in L$.
Works for any such w. (NFAs: Convert to DFA.)