Myhill-Nerode Theorem: The language A is regular iff $\sim_A$ has a finite number of equivalence classes. Furthermore, this number of equivalence classes is equal to the number of states in the minimum-state DFA that accepts A.

Proof Idea: to perform the job of a DFA for A you must remember neither more nor less than which equivalence class wrt $\sim_A$ the string that you have read so far is in.

Closure Theorem for Regular Sets: Let $A, B \subseteq \Sigma^\star$ be regular languages and let $h : \Sigma^\star \longrightarrow \Gamma^\star$ and $g : \Gamma^\star \longrightarrow \Sigma^\star$ be homomorphisms. Then the following languages are regular:

1. $A \cup B$
2. $A \cap B$
3. AB
4. $\bar{A} = (\Sigma^\star - A)$
5. $h(A)$
6. $g^{-1}(A)$

Definition: A **context-free grammar** (CFG) is a 4-tuple $G = (V, \Sigma, R, S)$,

- V = variables = nonterminals,
- Σ = terminals,
- R = rules = productions, $R \subseteq V \times (V \cup \Sigma)^\star$,
- $S \in V$,
- V, Σ, R are all finite.

$$G_1 = (\{S\}, \{a, b\}, R_1, S)$$
$$R_1 = \{\langle S, aSb \rangle, \langle S, \epsilon \rangle\} \quad = \quad \{S \rightarrow aSb | \epsilon\}$$

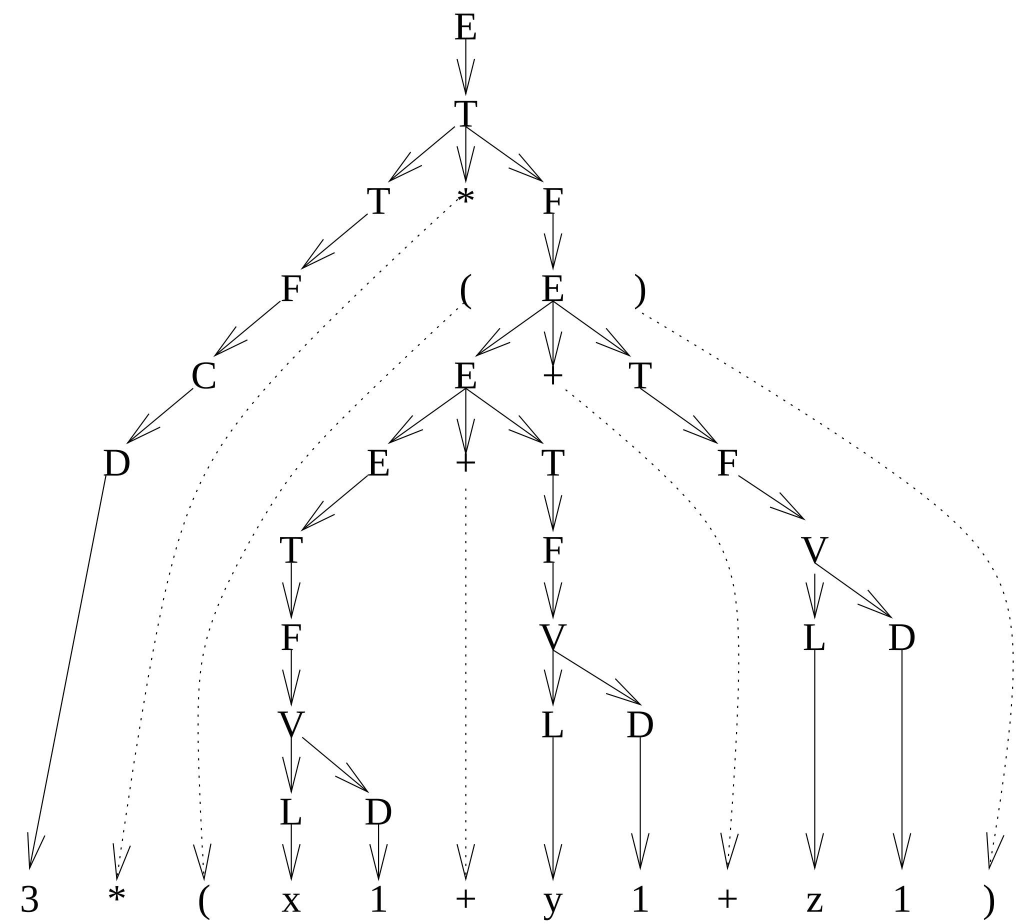
$$S \rightarrow \epsilon$$
$$S \rightarrow aSb \rightarrow ab$$
$$S \rightarrow aSb \rightarrow aaSbb \rightarrow aabb$$
$$S \rightarrow aSb \rightarrow aaSbb \rightarrow aaaSbbb \rightarrow aaabbb$$

$$\mathcal{L}(G_1) = \{w \in \{a, b\}^\star \mid S \underset{G_1}{\overset{\star}{\rightarrow}} w\}$$
$$= \{a^n b^n \mid n \in \mathbf{N}\}$$
$$\mathcal{L}(G) = \{w \in \Sigma^\star \mid S \underset{G}{\overset{\star}{\rightarrow}} w\}$$

$$G_2 = (\{E, T, F, V, L, D, C\}, \{(,), +, \star, x, y, z, 0, 1, \ldots, 9\}, R_2, E)$$

$$R_2 = \begin{array}{lll} E \rightarrow E + T | T & \quad & L \rightarrow x|y|z \\ T \rightarrow T \star F | F & & D \rightarrow 0|1|2|\cdots|9 \\ F \rightarrow (E)|V|C & & C \rightarrow D|CD \\ V \rightarrow LD & & \end{array}$$

Parse Tree:



Pumping Lemma for Regular Sets: Let $N = (Q, \Sigma, \Delta, q_0, F)$ be an NFA. Let $n = |Q|$. Let $w \in \mathcal{L}(N)$ s.t. $|w| \geq n$. Then $\exists x, y, z \in \Sigma^\star$ s.t. the following all hold:

- $xyz = w$
- $|xy| \leq n$
- $|y| > 0$, and
- $\forall k \geq 0 (xy^k z \in \mathcal{L}(N))$

Proof: Let $w \in \mathcal{L}(N)$; $|w| \geq n$, and consider the first $n+1$ states in an accepting path of N on w:

$$\begin{array}{cccccccccccccc} w & = & & w_1 & & w_2 & & w_3 & & \cdots & & w_n & & u \\ & & q_0 & & q_1 & & q_2 & & q_3 & \cdots & q_{n-1} & & q_n & \end{array}$$

By the "Pigeon-Hole principle", $\exists i < j (q_i = q_j)$

$$\begin{array}{cccccccc} w & = & & \overbrace{w_1 \ldots w_i}^{x} & & \overbrace{w_{i+1} \ldots w_j}^{y} & & \overbrace{w_{j+1} \ldots w_n u}^{z} & \\ & & q_0 & & q_i & & q_i & & q_f \end{array}$$

$q_i \in \Delta^\star(q_i, y)$. Thus, $xy^k z \in \mathcal{L}(N)$ for $k = 0, 1, 2, \ldots$. $\square$

Prop: $E = \{a^r b^r \mid r \in \mathbf{N}\}$ is not regular.

Proof: Suppose that E were regular, accepted by a DFA with n states. Let $w = a^n b^n$. By pumping lemma, $w = a^n b^n = xyz$ where

- $|xy| \le n$
- $|y| > 0$, and
- $\forall k \in \mathbf{N}(xy^k z \in E)$

Since $0 < |xy| \le n$, $\quad y = a^i, 0 < i \le n$.

Thus $xy^0 z = a^{n-i} b^n \in E$.

But, $a^{n-i} b^n \notin E$.

$\Rightarrow\Leftarrow$ [“$\Rightarrow\Leftarrow$” is our symbol for contradiction.]

Therefore E is not regular. $\square$

CFL Pumping Lemma: Let A be a CFL. Then there is a constant n, depending only on A such that if $z \in A$ and $|z| \geq n$, then there exist strings u, v, w, x, y such that,

- $z = uvwxy$, and
- $|vx| \geq 1$, and
- $|vwx| \leq n$, and
- for all $k \in \mathbf{N}$, $uv^kwx^ky \in A$

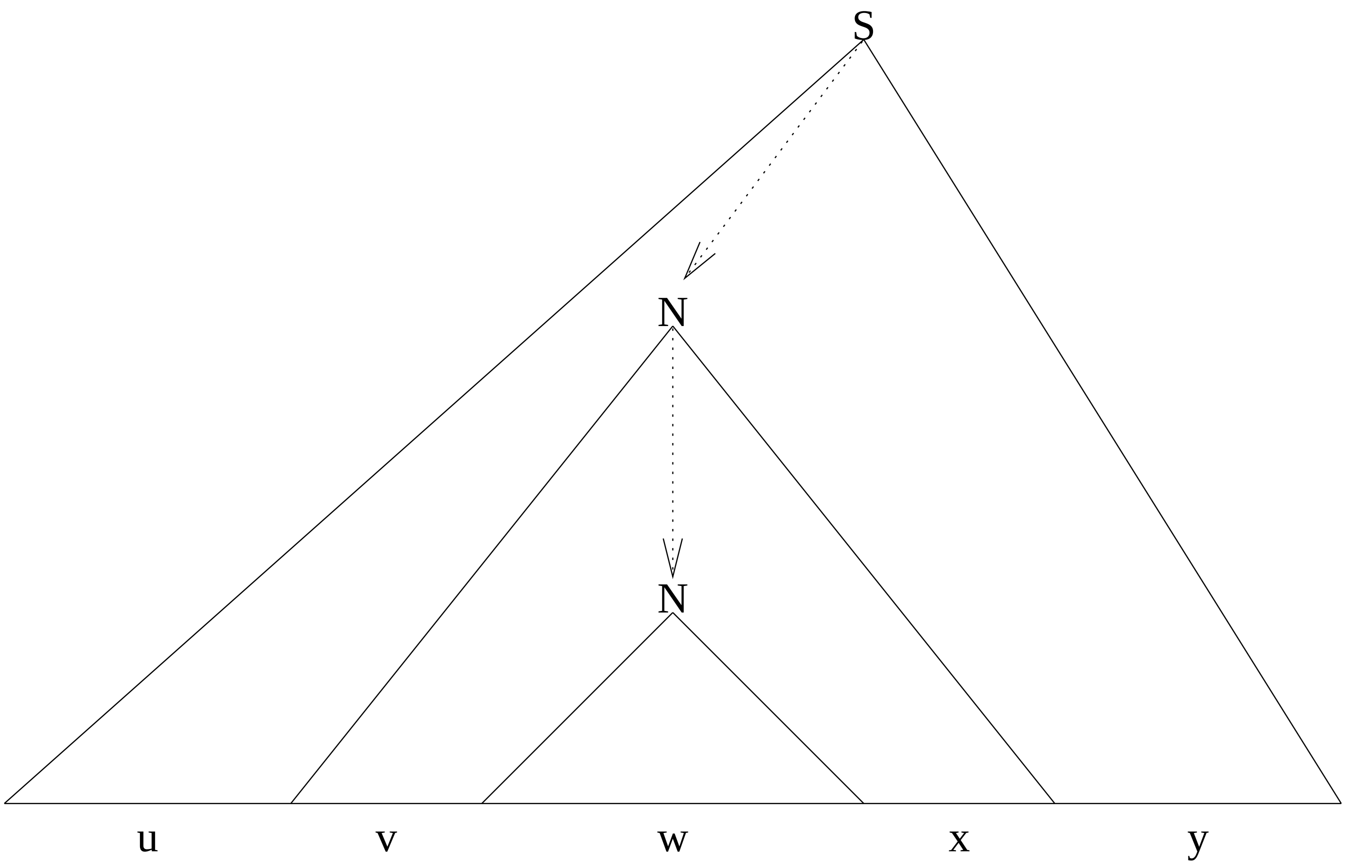
proof:
Let $G = (V, \Sigma, R, S)$ be a CFG with $\mathcal{L}(G) = A$.

Claim: Let $n = d^{|V|+2}$ where d is the maximum of the length of any right-hand-side of a rule of G. Then if T is any subtree of a parse tree of G with at least n leaves, then the height of T is at least $|V| + 2$.

Proof: Exercise: this is an easy induction. $\square$

Let $z \in A$, $|z| \geq n$.
The parse tree for z has height at least $|V| + 2$.

Thus, some path repeats a nonterminal, N. Choose such a pair of repeating nonterminals that is lowest in the parse tree. The rest of the proof is in the following diagram.



$$z \;=\; uvwxy; \qquad \forall k \in \mathbf{N}(uv^k wx^k y \in A) \qquad \square$$

Prop: $P_2 = \{a^n b^m a^n b^m \mid n, m \in \mathbf{N}\}$ is not a CFL.

Proof: Suppose P_2 were a CFL and let n be the constant of the pumping lemma.
Let $z = a^n b^n a^n b^n$. By pumping lemma, $z = uvwxy$, and

1. $|vx| \geq 1$,
2. $|vwx| \leq n$, and
3. for all $k \in \mathbf{N}$, $uv^k wx^k y \in P_2$

Since $|vwx| \leq n$, $vwx \in a^\star b^\star$ or $vwx \in b^\star a^\star$.

Suppose that vx contains at least one a. Then, uv^2wx^2y is not in P_2, because it has more a's in one group than the other.

Suppose that vx contains at least one b. Then, uv^2wx^2y is not in P_2, because it has more b's in one group than the other.

Thus, uv^2wx^2y is not in P_2.

$\Rightarrow\Leftarrow$

Thus P_2 is not a CFL. $\square$

Definition: A **pushdown automaton** (PDA) is a 7-tuple, $P = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0, F)$

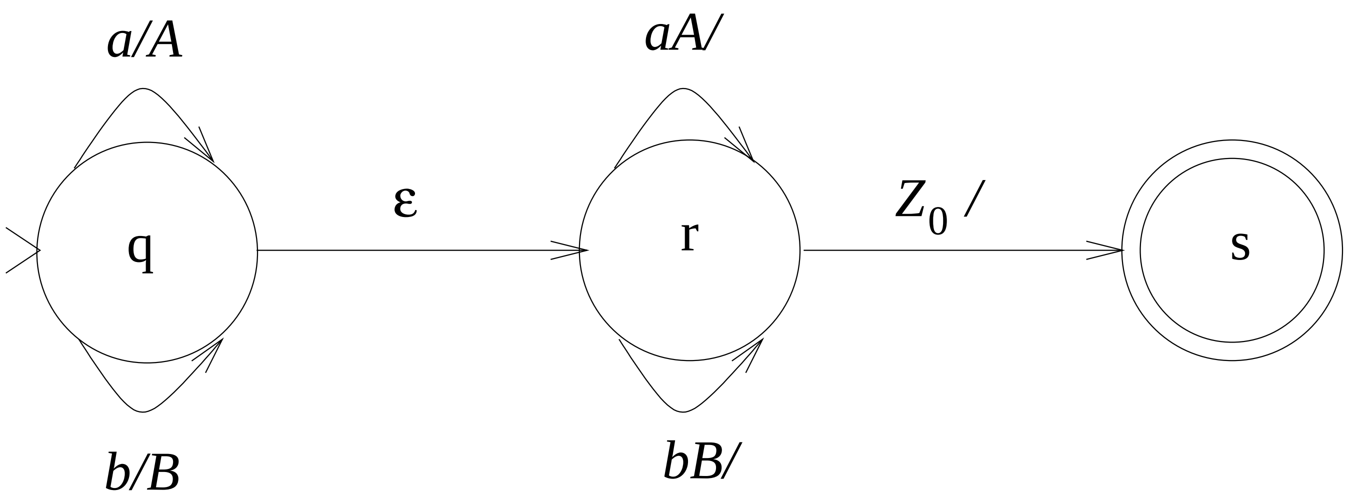
- $Q =$ finite set of states,
- $\Sigma =$ input alphabet,
- $\Gamma =$ stack alphabet,
- $\Delta \subseteq (Q \times \Sigma^\star \times \Gamma^\star) \times (Q \times \Gamma^\star)$ finite set of transitions,
- $q_0 \in Q$ start state,
- $Z_0 \in \Gamma$ initial stack symbol,
- $F \subseteq Q$ final states.

$$\text{PDA} \quad = \quad \text{NFA} \; + \; \text{stack}$$

$$\mathcal{L}(P) \quad = \quad \{w \in \Sigma^\star \mid (q_0, Z_0) \xrightarrow[P]{w} (q, X),\; q \in F, X \in \Gamma^\star\}$$

$$P_1 \ = \ (\{q, r, s\}, \{a, b\}, \{A, B, Z_0\}, \Delta_1, q, Z_0, \{s\})$$

$$\Delta_1 \ = \ \{\langle(q, a, \epsilon), (q, A)\rangle, \langle(q, b, \epsilon), (q, B)\rangle, \langle(q, \epsilon, \epsilon), (r, \epsilon)\rangle$$
$$\langle(r, a, A), (r, \epsilon)\rangle, \langle(r, b, B), (r, \epsilon)\rangle, \langle(r, \epsilon, Z_0), (s, \epsilon)\rangle\}$$



$$\mathcal{L}(P_1) \ = \ \{ww^R \ | \ w \in \{a, b\}^\star\}$$

Theorem 3.1 *Let $A \subseteq \Sigma^\star$ be any language. Then the following are equivalent:*

1. *$A = \mathcal{L}(G)$, for some CFG G.*
2. *$A = \mathcal{L}(P)$, for some PDA P.*
3. *A is a context-free language.*

Proof: See [HMU] or [LP] or [S]. $\square$

Closure Theorem for Context-Free Languages: Let $A, A_1, A_2 \subseteq \Sigma^\star$ be CFL's, let R be a regular language and let $h : \Sigma^\star \to \Gamma^\star$ and $g : \Gamma^\star \to \Sigma^\star$ be homomorphisms. Then the following languages are CFL's:

1. $A_1 \cup A_2$
2. $A_1 A_2$
3. $A \cap R$
4. $h(A)$
5. $g^{-1}(A)$

Proof Sketch: (1,2): Let $A_i = L(G_i) = (V_i, \Sigma, R_i, S_i)$ where $V_1 \cap V_2 = \emptyset$. Then the grammar for $A_1 \cup A_2$ is $(V_1 \cup V_2 \cup \{S\}, \Sigma, R_1 \cup R_2 \cup \{S \to S_1 | S_2\}, S)$ and the grammar for $A_1 A_2$ is $(V_1 \cup V_2 \cup \{S\}, \Sigma, R_1 \cup R_2 \cup \{S \to S_1 S_2\}, S)$,

(3): Given $A = L(P_1)$ for PDA $P_1 = (Q_1, \Sigma, \Gamma, \Delta_1, s_1, Z_0, F_1)$ and $R = L(N_2)$ for NFA $N_2 = (Q_2, \Sigma, \Delta_2, s_2, F_2)$, we can form the PDA accepting $A \cap R$: $P_1 \times N_2 = (Q_1 \times Q_2, \Sigma, \Gamma, \Delta', \langle s_1, s_2 \rangle, Z_0, F_1 \times F_2)$. $P_1 \times N_2$ is essentially P_1 with a bigger state space so that it also keeps track of the possible states of N_2 as we read the input.

(4): Let G be a CFG for A. Define $H(G)$ to be the grammar where each terminal symbol, σ on the right hand side of any rule of G is replaced by $h(\sigma)$. Then it is fairly easy to see that $\mathcal{L}(H(G)) = h(A)$.

(6): Let P be a PDA accepting A. We want to build a PDA, P', that accepts $g^{-1}(A) = \{w \in \Gamma^\star \mid g(w) \in A\}$. As in the corresponding proof for regular languages, we want P' on reading symbol γ to do exactly what P does on reading the string $g(\gamma)$. To do this, we add enough memory to the finite control, i.e., enough new states, to save a "buffer" that holds the characters in $g(\gamma)$ that have not been read yet. The size of the buffer is one less than the maximum length of $g(\gamma)$ for $\gamma \in \Gamma$. While the buffer is not empty, P' must remove the next character from the buffer and treat it as P would treat the next symbol of its input. When the buffer is empty, P' reads the next character, γ, from its input, has P read the first character of $g(\gamma)$, and places the remaining characters into the buffer. The accept states of P' are the accept states of P with the buffer empty.