Non-context-free language

Lin Chen

Email: Lin.Chen@ttu.edu

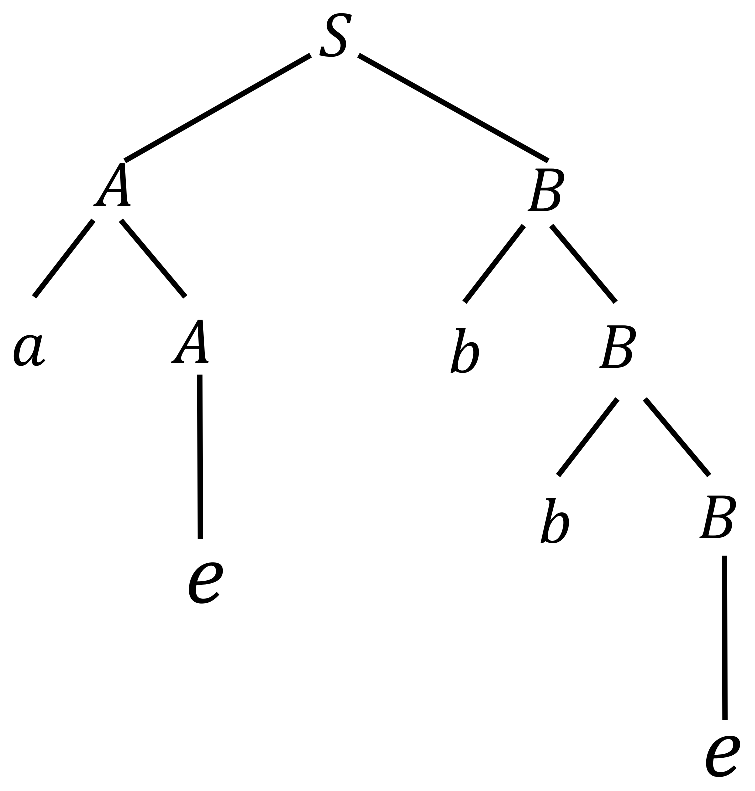
TEXAS TECH
UNIVERSITY

Pumping theorem

- A parse tree is a graphical representation of a derivation
  - Example:

$$S \rightarrow AB$$
$$A \rightarrow aA|e$$
$$B \rightarrow bB|e$$

$$\begin{aligned} S &\Rightarrow AB \\ &\Rightarrow aAB \\ &\Rightarrow aAbB \\ &\Rightarrow abB \\ &\Rightarrow abbB \\ &\Rightarrow abb \end{aligned}$$

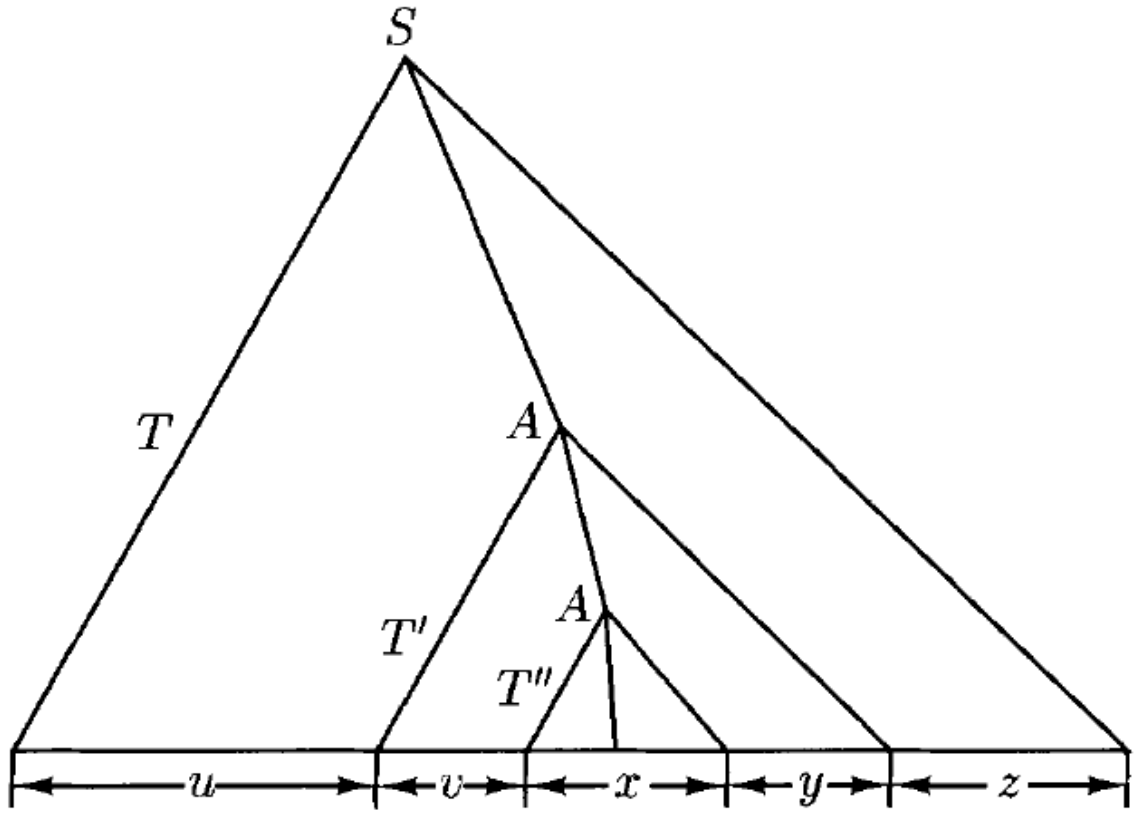


Pumping theorem

- A parse tree is a graphical representation of a derivation
  - The root of a parse tree is the start symbol S
  - A leaf of a parse tree is a terminal
  - The leaves of a parse tree, from left to right, form the string

Pumping theorem

- Consider a parse tree of a long enough string
  - some of the rule is reused

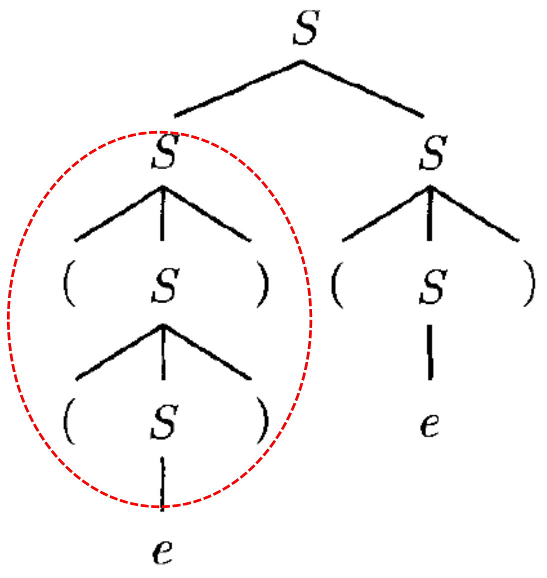


Pumping theorem

- Consider a parse tree of a long enough string
  - some of the rule is reused

$$V = \{S, (,)\},$$
$$\Sigma = \{(,)\},$$
$$R = \{S \to e,$$
$$S \to SS,$$
$$S \to (S)\}.$$

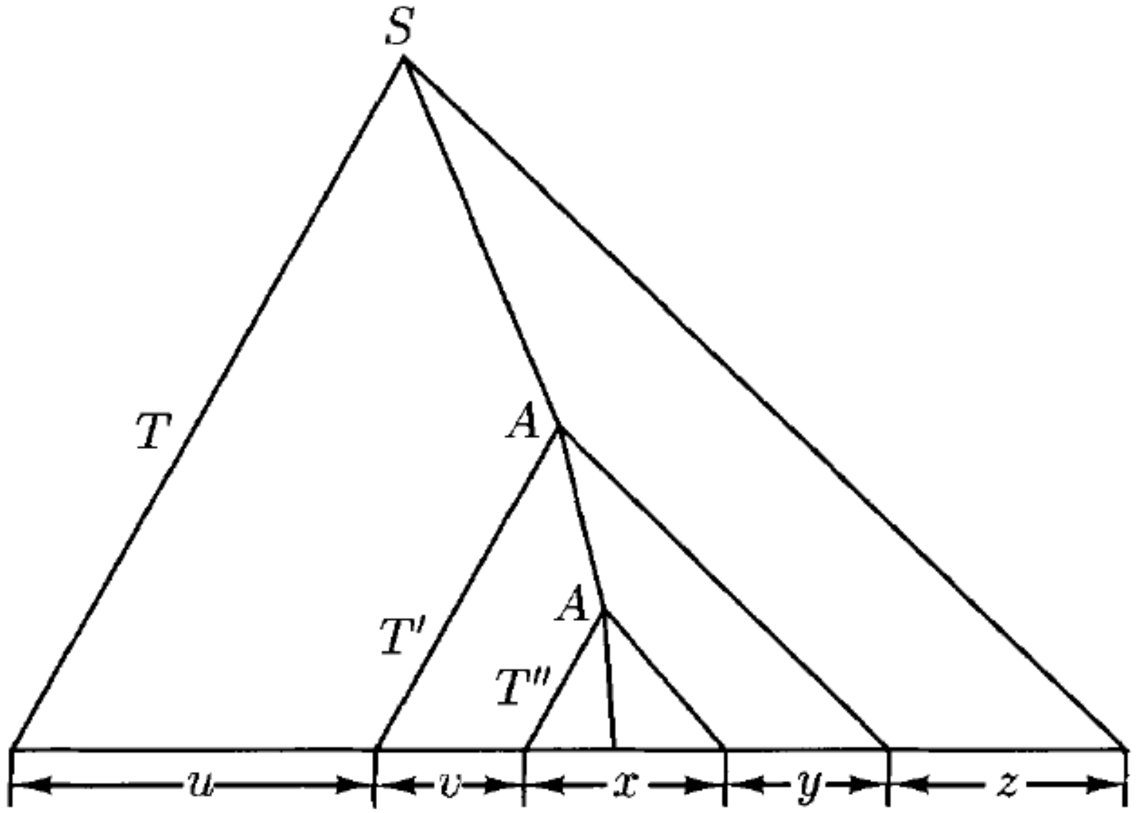
$$D_1 = S \Rightarrow SS \Rightarrow (S)S \Rightarrow ((S))S \Rightarrow (())S \Rightarrow (())(S) \Rightarrow (())()$$



Pumping theorem

- Consider a parse tree of a long enough string
  - some of the rule is reused
  - If $A \to \cdots \to vAy$, then

    $A \to \cdots \to vAy \to \cdots \to vvAyy$



Pumping theorem

Pumping lemma for context-free languages If A is a context-free language, then there is a number p (the pumping length) where, if s is any string in A of length at least p, then s may be divided into five pieces $s = uvxyz$ satisfying the conditions

1. for each $i \geq 0$, $uv^ixy^iz \in A$,
2. $|vy| > 0$, and
3. $|vxy| \leq p$.

Pumping theorem

Use Pumping theorem to show the followings are not context-free:

- a). $\{a^n b^n c^n : n \geq 0\}$
- b). $\{a^p : p \text{ is prime}\}$
- c). $\{a^{n^2} : n \geq 0\}$
- d). $\{a^n b^n a^n b^n : n \geq 0\}$
- d). $\{ww : w \in \{a, b\}^*\}$
- e). $\{a^n b a^{2n} b a^{3n} b : n \geq 0\}$
- f). $\{w_1 \# w_2 : w_1, w_2 \in \{a, b\}^*, w_1 \text{ is a substring of } w_2\}$

$\{a^n b^n c^n : n \geq 0\}$

- a). Suppose on the contrary that $L = \{a^n b^n c^n : n \geq 0\}$ is CFG, then there exists some sufficiently large number N, for any $n \geq N$, we have $a^n b^n c^n = uvxyz$ such that $|vy| > 0$, $|vxy| \leq N$, and $uv^i xy^i z \in L$ for any $i \geq 0$.
- Pick $n = N$ and consider $a^N b^N c^N = uvxyz$. $|vxy| \leq N$, so there are 5 different possibilities.
- i). $vxy = a \cdots a,\ or\ b \cdots b, or\ c \cdots c$, i.e., it only consists one symbol
- ii).$vxy = a \cdots ab \cdots b$ or $vxy = b \cdots bc \cdots c$, i.e., vxy contains both a, b or b, c.

$\{a^n b^n c^n : n \geq 0\}$

- a). Suppose on the contrary that $L = \{a^n b^n c^n : n \geq 0\}$ is CFG, then there exists some sufficiently large number N, for any $n \geq N$, we have $a^n b^n c^n = uvxyz$ such that $|vy| > 0$, $|vxy| \leq N$, and $uv^i xy^i z \in L$ for any $i \geq 0$.
- Pick $n = N$ and consider $a^N b^N c^N = uvxyz$. $|vxy| \leq N$, so there are 5 different possibilities.
- i). $vxy = a \cdots a,\ or\ b \cdots b, or\ c \cdots c$, i.e., it only consists one symbol
- We show the case of $vxy = a \cdots a,$ the other two cases are the same. Since $|vy| > 0$, we know $v^2 xy^2$ contains exactly $|vy|$ more a's than vxy. That is, $uv^2 xy^2 z$ will contain $N + |vy| > N$ copies of a, i.e., $uv^2 xy^2 z = a^{N+|vy|} b^N c^N \notin L,$ contradicting that $uv^i xy^i z \in L$ for any $i \geq 0$.
- ii).$vxy = a \cdots ab \cdots b$ or $vxy = b \cdots bc \cdots c$, i.e., vxy contains both a, b or b, c.
- We show that case of $vxy = a \cdots ab \cdots b$, the other case is the same. Since $|vy| > 0$, we assume $vy = a^\alpha b^\beta$ for some $\alpha, \beta \geq 0$ and $\alpha + \beta > 0$. Now we have $uv^2 xy^2 z$ contains $N + \alpha$ copies of a's, $N + \beta$ copies of b's and N copies of c's, which is not in L, contradicting that $uv^i xy^i z \in L$ for any $i \geq 0$
- (Note that since $|vxy| \leq N$, it is impossible for vxy to contain all a, b, c. Thus we have exhausted all the possibilities.)

Closure property

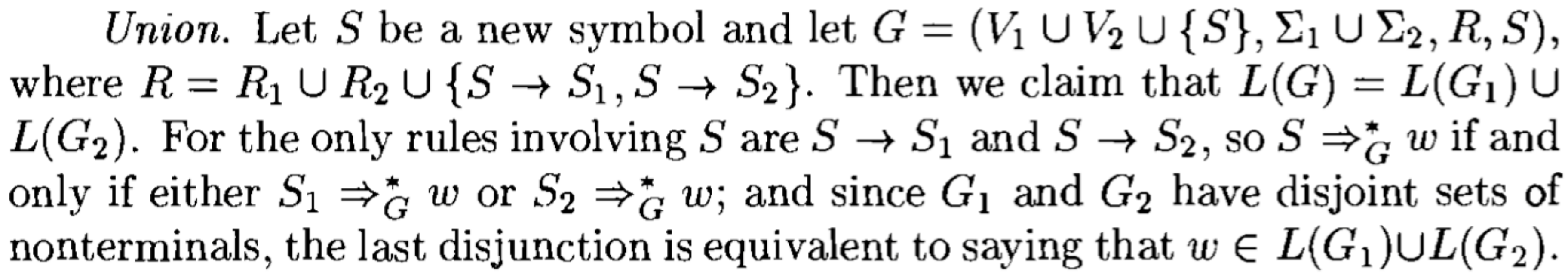
- Regular language is closed under
  - Union
  - Concatenation
  - Kleene star
  - Complementation
  - Intersection

Closure property

- Context-free language is closed under
  - Union
  - Concatenation
  - Kleene star
- Context-free language is not closed under
  - Complementation
  - Intersection

Closure property

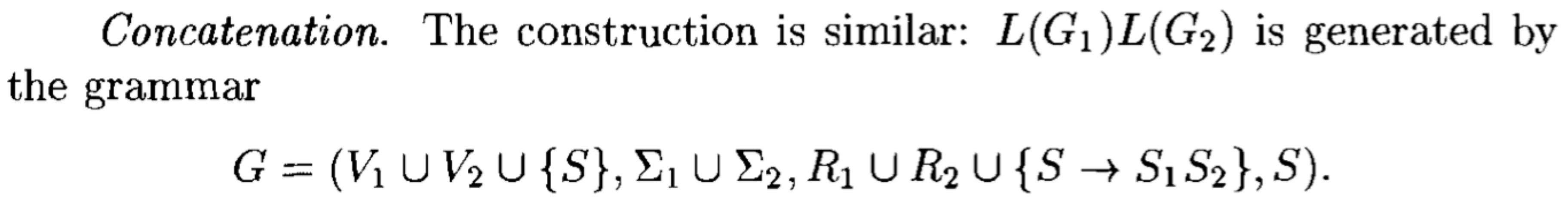
- Context-free language is closed under
  - Union



Union. Let S be a new symbol and let $G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R, S)$, where $R = R_1 \cup R_2 \cup \{S \to S_1, S \to S_2\}$. Then we claim that $L(G) = L(G_1) \cup L(G_2)$. For the only rules involving S are $S \to S_1$ and $S \to S_2$, so $S \Rightarrow_G^* w$ if and only if either $S_1 \Rightarrow_G^* w$ or $S_2 \Rightarrow_G^* w$; and since G_1 and G_2 have disjoint sets of nonterminals, the last disjunction is equivalent to saying that $w \in L(G_1) \cup L(G_2)$.

Closure property

- Context-free language is closed under
  - Concatenation



Concatenation. The construction is similar: $L(G_1)L(G_2)$ is generated by the grammar

$$G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R_1 \cup R_2 \cup \{S \to S_1S_2\}, S).$$

Closure property

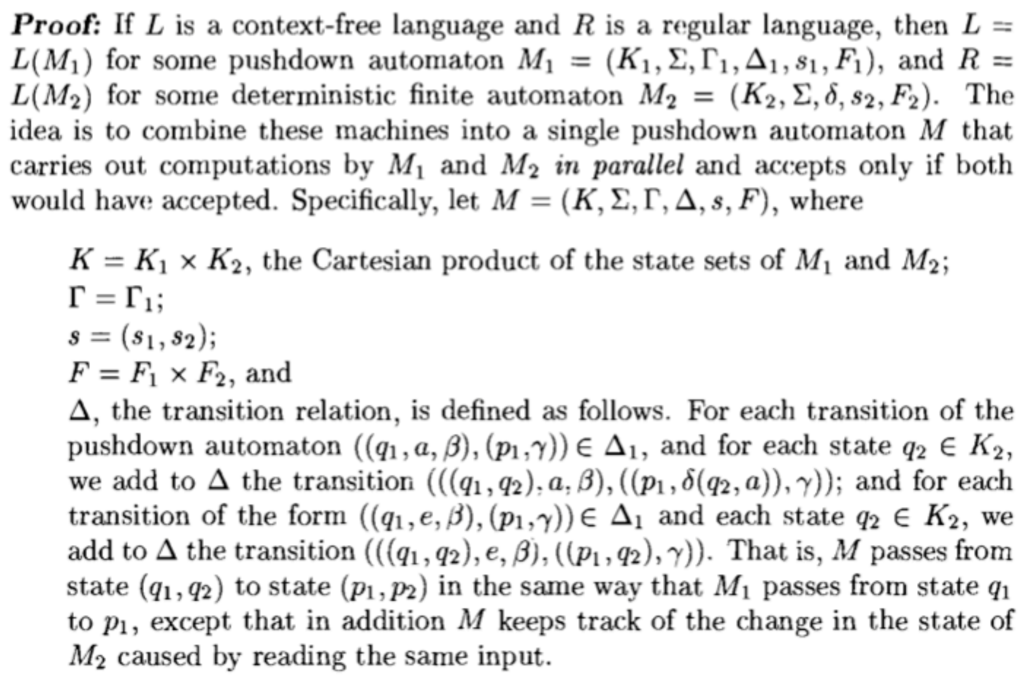
- Context-free language is closed under
  - Kleene star

Kleene Star. $L(G_1)^*$ is generated by

$$G = (V_1 \cup \{S\}, \Sigma_1, R_1 \cup \{S \to e, S \to SS_1\}, S).$$

Closure property

- The intersection of context-free language and regular language is context-free.



Proof: If L is a context-free language and R is a regular language, then $L = L(M_1)$ for some pushdown automaton $M_1 = (K_1, \Sigma, \Gamma_1, \Delta_1, s_1, F_1)$, and $R = L(M_2)$ for some deterministic finite automaton $M_2 = (K_2, \Sigma, \delta, s_2, F_2)$. The idea is to combine these machines into a single pushdown automaton M that carries out computations by M_1 and M_2 *in parallel* and accepts only if both would have accepted. Specifically, let $M = (K, \Sigma, \Gamma, \Delta, s, F)$, where

$K = K_1 \times K_2$, the Cartesian product of the state sets of M_1 and M_2;
$\Gamma = \Gamma_1$;
$s = (s_1, s_2)$;
$F = F_1 \times F_2$, and
Δ, the transition relation, is defined as follows. For each transition of the pushdown automaton $((q_1, a, \beta), (p_1, \gamma)) \in \Delta_1$, and for each state $q_2 \in K_2$, we add to Δ the transition $(((q_1, q_2), a, \beta), ((p_1, \delta(q_2, a)), \gamma))$; and for each transition of the form $((q_1, e, \beta), (p_1, \gamma)) \in \Delta_1$ and each state $q_2 \in K_2$, we add to Δ the transition $(((q_1, q_2), e, \beta), ((p_1, q_2), \gamma))$. That is, M passes from state (q_1, q_2) to state (p_1, p_2) in the same way that M_1 passes from state q_1 to p_1, except that in addition M keeps track of the change in the state of M_2 caused by reading the same input.