Article

Not peer-reviewed version

A Mathematical Approach to Context-Free Languages

Chac Kwan *

Posted Date: 3 June 2025

doi: 10.20944/preprints202506.0158.v1

Keywords: Theoretical Computer Science; Computability Theory; Context Free Languages; Finite Automata; Pushdown Automata; Pumping Lemma; Chomsky Normal Form; Discrete Mathematics

Article

A Mathematical Approach to Context-Free Languages

Chac B Kwan

ckwan4@yahoo.com

Abstract: Students are getting confused and losing interest in theoretical computer science because most instructors are doing a poor job in teaching the subject matter. Instructors are doing a poor job in teaching because they do not have a well-organized theory to explain the concepts and they are unwilling to spend the time to write up better lecture notes for the class. This paper presents a rigorous mathematical approach to the theory of context-free languages which doesn't currently exist in the literature of theoretical computer science. Basic definitions are developed in mathematical terms and used as the foundation for constructing mathematical proofs for theorems. It provides a model for instructors to write better lecture notes and authors to write better textbooks for educational purpose. It also corrects some critical errors and erroneous arguments that can be found in many textbooks which are widely used for the education of theoretical computer science.
Students can use this paper for supplemental reading.

Keywords: Theoretical Computer Science; Computability Theory; Context Free Languages; Finite Automata; Pushdown Automata; Pumping Lemma; Chomsky Normal Form; Discrete Mathematics

---

2.1. Context-Free Grammars (CFG)

In Chapter 1, we use finite automata and regular expressions to describe regular languages. In this chapter, we introduce the concept of Context-Free Grammar which is a more powerful tool for describing languages.

A Context-Free Grammar is formally defined as follows.

Definition 2.1. A Context-Free Grammar denoted by CFG is a 4-tuple $G = (V, \Sigma, R, S)$, where

(i) V is a finite set of variables;
(ii) Σ is a finite set of terminals such that $V \cap \Sigma = \emptyset$;
(iii) $S \in V$ is the start variable; and
(iv) $R \subset V \times (V \cup \Sigma)^*$ is a finite relation

For any $(A, u) \in R$, we usually write $A \to u$ and call it a *rule*.
Accordingly, the relation R is also called the set of rules for the CFG.
A is sometimes called the head of the rule whereas u is called the body of the rule.

Example 2.2. Let $V = \{< SENTENSE >, < NOUN\ PHRASE >, < VERB\ PHRASE >, < PREP\ PHRASE >,$
$< CMPLX\ NOUN >, < CMPLX\ VERB >, < PREP >, < ARTICLE >, < NOUN >, < VERB >\}$

$\Sigma = \{a, the, boy, girl, flower, touches, likes, sees, with\}$

$S = < SENTENSE >$

Let R consist of the following rules:

$< SENTENSE > \longrightarrow < NOUN\ PHRASE >< VERB\ PHRASE >$

$< NOUN\ PHRASE > \longrightarrow < CMPLX\ NOUN > | < CMPLX\ NOUN >< PREP\ PHRASE >$

$< VERB\ PHRASE > \longrightarrow < CMPLX\ VERB > | < CMPLX\ VERB >< PREP\ PHRASE >$

$< PREP\ PHRASE > \longrightarrow < PREP >< CMPLX\ NOUN >$

$< CMPLX\ NOUN > \longrightarrow < ARTICLE >< NOUN >$

 doi:10.20944/preprints202506.0158.v1

$< CMPLX\ VERB > \longrightarrow < VERB > | < VERB >< NOUN\ PHRASE >$

$< ARTICLE > \longrightarrow a \mid the$

$< NOUN > \longrightarrow boy \mid girl \mid flower$

$< VERB > \longrightarrow touches \mid likes \mid sees$

$< PREP > \longrightarrow with$

$G = (V, \Sigma, R, S)$ is a CFG.

The following are examples of strings in Σ^* that can be derived by G.

(i) $< SENTENSE >$

$\longrightarrow < NOUN\ PHRASE >< VERB\ PHRASE >$

$\longrightarrow < CMPLX\ NOUN >< VERB\ PHRASE >$

$\longrightarrow < ARTICLE >< NOUN >< VERB\ PHRASE >$

$\longrightarrow a < NOUN >< VERB\ PHRASE >$

$\longrightarrow a\ boy < VERB\ PHRASE >$

$\longrightarrow a\ boy < CMPLX\ VERB >$

$\longrightarrow a\ boy < VERB >$

$\longrightarrow a\ boy\ sees$

(ii) $< SENTENSE >$

$\longrightarrow < NOUN\ PHRASE >< VERB\ PHRASE >$

$\longrightarrow < CMPLX\ NOUN >< VERB\ PHRASE >$

$\longrightarrow < ARTICLE >< NOUN >< VERB\ PHRASE >$

$\longrightarrow the\ boy < VERB\ PHRASE >$

$\longrightarrow the\ boy < CMPLX\ VERB >$

$\longrightarrow the\ boy < VERB >< NOUN\ PHRASE >$

$\longrightarrow the\ boy\ sees < CMPLX\ NOUN >$

$\longrightarrow the\ boy\ sees < ARTICLE >< NOUN >$

$\longrightarrow the\ boy\ sees\ a\ flower$

(iii) $< SENTENSE >$

$\longrightarrow < NOUN\ PHRASE >< VERB\ PHRASE >$

$\longrightarrow < CMPLX\ NOUN >< PREP\ PHRASE >< VERB\ PHRASE >$

$\longrightarrow < ARTICLE >< NOUN >< PREP >< CMPLX\ NOUN >< VERB\ PHRASE >$

$\longrightarrow a\ girl\ with < ARTICLE >< NOUN >< VERB\ PHRASE >$

$\longrightarrow a\ girl\ with\ a\ flower < CMPLX\ VERB >$

$\longrightarrow a\ girl\ with\ a\ flower < VERB >< NOUN\ PHRASE >$

$\longrightarrow a\ girl\ with\ a\ flower\ likes < CMPLX\ NOUN >$

$\longrightarrow a\ girl\ with\ a\ flower\ likes < ARTICLE >< NOUN >$

$\longrightarrow a\ girl\ with\ a\ flower\ likes\ the\ boy$

Definition 2.3. Let $G = (V, \Sigma, R, S)$ be a CFG.

For any $u, v \in (V \cup \Sigma)^*$, we say u yields v (or v is derivable from u) in one step (written as $u \overset{1}{\Rightarrow} v$ or simply $u \Longrightarrow v$) if and only if

$\exists\ A \in V, \gamma, \alpha, \beta \in (V \cup \Sigma)^*$ and a rule $A \longrightarrow \gamma$ such that $u = \alpha A \beta$ and $v = \alpha\gamma\beta$.

 doi:10.20944/preprints202506.0158.v1

Note that the process of deriving v from u is basically a replacement of a variable in u by the body of the variable's rule to obtain v.

In addition, we define $u \overset{0}{\Rightarrow} v$ iff $u = v$.

For any integer $n \geq 0$, we say u yields v (or v is derivable from u) in $n+1$ steps (written as $u \overset{n+1}{\Longrightarrow} v$) iff $\exists w \in (V \cup \Sigma)^*$ such that $u \overset{n}{\Rightarrow} w$ and $w \overset{1}{\Rightarrow} v$.

If there are more than one CFG to be considered, (e.g. G and G') and if we need to distinguish between derivations in G from derivations in G', we can write

$u \overset{n,G}{\Longrightarrow} v$ to mean v is derivable from u in n steps by use of rules in G; and

$u \overset{n,G'}{\Longrightarrow} v$ to mean v is derivable from u in n steps by use of rules in G'.

Furthermore, if we need to specify the rule to be applied in each step, we can use

$u \overset{n,G,(R_1,R_2,\cdots R_n)}{\Longrightarrow} v$ to mean v is derivable from u in n steps by use of rules in G with rule R_i to be applied in the i^{th} step; and

$u \overset{n,G',(R_1',R_2',\cdots R_n')}{\Longrightarrow} v$ to mean v is derivable from u in n steps by use of rules in G' with rule R_i' to be applied in the i^{th} step.

Since there can be more than one way of deriving a string, it is sometimes useful to require the derivation to be leftmost. A leftmost derivation is a derivation in which the leftmost variable at every step is replaced by the body of its rule.

Formally, we define leftmost derivation as follows.

For any $u, v \in (V \cup \Sigma)^*$, v is a leftmost derivation of u in one step (written as $u \overset{1,lm}{\Longrightarrow} v$ or simply $u \overset{lm}{\Rightarrow} v$) iff $\exists w \in \Sigma^*$, $w' \in (V \cup \Sigma)^*$, $A \in V, \alpha \in (V \cup \Sigma)^*$ and a rule $A \longrightarrow \alpha$ such that $u = wAw'$ and $v = w\alpha w'$.

For any integer $n \geq 0, u \overset{n,lm}{\Longrightarrow} v$ is defined similarly as $u \overset{n}{\Rightarrow} v$.

Definition 2.4. Let $G = (V, \Sigma, R, S)$ be a CFG; $\overset{*}{\Rightarrow}$ be a subset of $(V \cup \Sigma)^* \times (V \cup \Sigma)^*$.
We define the relation $\overset{*}{\Rightarrow}$ as follows:
$\forall u, v \in (V \cup \Sigma)^*, u \overset{*}{\Rightarrow} v$ if and only if $u \overset{n}{\Rightarrow} v$ for some integer $n \geq 0$.
n is called the length of the derivation of v from u.

Note that whenever there is an n such that $u \overset{n}{\Rightarrow} v$, there is a minimum n' such that $u \overset{n'}{\Rightarrow} v$.

If there are more than one CFG to be considered,

$u \overset{*,G}{\Rightarrow} v$ if and only if $u \overset{n,G}{\Longrightarrow} v$ for some integer $n \geq 0$.

$\overset{*,lm}{\Longrightarrow}$ is defined similarly as $\overset{*}{\Rightarrow}$.

Proposition 2.5. $\overset{*}{\Rightarrow}$ (respectively $\overset{*,lm}{\Longrightarrow}$) is reflexive and transitive.

Proof. Since $u \overset{0}{\Rightarrow} u$ for all $u \in (V \cup \Sigma)^*$, $u \overset{*}{\Rightarrow} u$ for all $u \in (V \cup \Sigma)^*$.

Therefore, $\overset{*}{\Rightarrow}$ is reflexive.

For transitivity, assume $u \overset{*}{\Rightarrow} v$ and $v \overset{*}{\Rightarrow} w$.

There exist integers $m \geq 0$ and $n \geq 0$ such that
$u \overset{m}{\Rightarrow} v$ and $v \overset{n}{\Rightarrow} w$.

 doi:10.20944/preprints202506.0158.v1

There are two cases to examine, $n = 0$ or $n \neq 0$

(i) $n = 0$

$$v \overset{0}{\Rightarrow} w$$

By definition, $v = w$.

Since $u \overset{m}{\Rightarrow} v$, $u \overset{m}{\Rightarrow} w$.

Therefore, $u \overset{*}{\Rightarrow} w$.

(ii) $n \neq 0$

$v \overset{n}{\Rightarrow} w$

$v \overset{n-1}{\Longrightarrow} \alpha_{n-1} \overset{1}{\Rightarrow} w$ for some $\alpha_{n-1} \in (V \cup \Sigma)^*$.

With a backward induction argument, we have

$v \Longrightarrow \alpha_1 \Longrightarrow \alpha_2 \cdots\cdots\cdots \Longrightarrow \alpha_{n-1} \Longrightarrow w$ for some $\alpha_1, \alpha_2, \cdots \alpha_{n-1} \in (V \cup \Sigma)^*$.

We now have

$u \overset{m}{\Rightarrow} v \Longrightarrow \alpha_1 \Longrightarrow \alpha_2 \cdots\cdots\cdots \Longrightarrow \alpha_{n-1} \Longrightarrow w$.

Since $\left(u \overset{m}{\Rightarrow} v \Longrightarrow \alpha_1\right) \Longrightarrow \left(u \overset{m+1}{\Longrightarrow} \alpha_1\right)$,

$(u \overset{m+1}{\Longrightarrow} \alpha_1 \Longrightarrow \alpha_2) \Longrightarrow \left(u \overset{m+2}{\Longrightarrow} \alpha_2\right)$,

$\vdots$

With a forward induction argument, we have

$u \overset{m+n-1}{\Longrightarrow} \alpha_{n-1}$.

Finally, $(u \overset{m+n-1}{\Longrightarrow} \alpha_{n-1} \Longrightarrow w) \Longrightarrow (u \overset{m+n}{\Longrightarrow} w)$.

Therefore, $u \overset{*}{\Rightarrow} w$.

Combining (i) and (ii), $\overset{*}{\Rightarrow}$ is transitive.

With a similar argument, we can establish $\overset{*,lm}{\Longrightarrow}$ is also reflexive and transitive.

Definition 2.6. Let $G = (V, \Sigma, R, S)$ be a CFG.

The language of G is defined as

$L(G) = \{w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w\}$.

Note that if $S \overset{n}{\Rightarrow} w$, $n \geq 1$ because $S \overset{0}{\Rightarrow} w$ implies $S = w$ which is a contradiction.

Definition 2.7. Let $G = (V, \Sigma, R, S)$ be a CFG.

Let Q represent the rule $A \to \alpha$ in R.

$\forall u, v \in (V \cup \Sigma)^*$, we say u yields v (or v is derivable from u) using the rule Q (written as $u \overset{Q}{\Rightarrow} v$) if and only if there exist $w_1, w_2 \in (V \cup \Sigma)^*$ such that $u = w_1 A w_2$ and $v = w_1 \alpha w_2$.

Proposition 2.8. Let $G = (V, \Sigma, R, S)$ be a CFG.

For any $A \in V$, $\alpha \in (V \cup \Sigma)^*$ and $x, y, z \in (V \cup \Sigma)^*$,

(i) $(A \longrightarrow \alpha) \Leftrightarrow (A \Longrightarrow \alpha)$

(ii) If there is no α in $(V \cup \Sigma)^*$ such that $S \longrightarrow \alpha$ is a rule, $L(G) = \emptyset$.

(iii) If $\{\alpha \in (V \cup \Sigma)^* | A \longrightarrow \alpha\} = \emptyset$ and $x \Longrightarrow y$, then A appears in $x \Longrightarrow A$ appears in y.

(iv) Let $S \Longrightarrow u_1 \Longrightarrow u_2 \cdots\cdots\cdots \Longrightarrow u_n \Longrightarrow w$, where $u_i \in (V \cup \Sigma)^*$ for all $i \in \{1,2,3, \cdots n\}$, $w \in \Sigma^*$ and $n \geq 1$.

If $A \in V$ and A appears in u_i for some $i \in \{1,2,3, \cdots n\}$, then $\exists \alpha \in (V \cup \Sigma)^*$ such that $A \longrightarrow \alpha$ is a rule.

Proof.

(i) If $A \longrightarrow \alpha$ is a rule, since ϵ is in $(V \cup \Sigma)^*$, $\epsilon A \epsilon \Longrightarrow \epsilon \alpha \epsilon$.

Therefore, $A \Longrightarrow \alpha$.

Conversely if $A \Longrightarrow \alpha$, $\exists Q \in V, \beta, w_1, w_2 \in (V \cup \Sigma)^*$ such that $A = w_1 Q w_2$ and

 doi:10.20944/preprints202506.0158.v1

$\alpha = w_1\beta w_2$ and $Q \longrightarrow \beta$ is a rule.

Since $A = w_1 Q w_2$, w_1, w_2 must be ϵ and $A = Q$.

Therefore, $\alpha = \beta$ and $Q \longrightarrow \beta$ becomes $A \longrightarrow \alpha$.

(ii) Assume for contradiction that $L(G) \neq \emptyset$.

$\exists w \in L(G)$.

$\exists\, k \in \mathbb{N} \cup \{0\}, w_1, w_2, \cdots\cdots\cdots w_k \in (V \cup \Sigma)^*$ such that $S \Longrightarrow w_1 \Longrightarrow w_2 \cdots\cdots\cdots \Longrightarrow w_k \Longrightarrow w$ or $S \Longrightarrow w$. (Note that $k = 0 \Longrightarrow S = w_k$.)

By (i), $S \longrightarrow w_1$ or $S \longrightarrow w$.

This contradicts the assumption that there is no α in $(V \cup \Sigma)^*$ such that $S \longrightarrow \alpha$ is a rule.

(iii) Since $x \Longrightarrow y$, $\exists B \in V, w_1, w_2, \beta \in (V \cup \Sigma)^*$ such that $x = w_1 B w_2$, $y = w_1 \beta w_2$ and $B \longrightarrow \beta$ is a rule.

Since $A \longrightarrow \alpha$ is not a rule $\forall \alpha \in (V \cup \Sigma)^*, A \neq B$.

Since A appears in x, A appears either in w_1 *or* w_2.

In either case, A appears in y.

(iv) Assume for contradiction $\exists A$ which appears in u_i for some $i \in \{1,2,3 \cdots n\}$ such that $A \longrightarrow \alpha$ is not a rule for all $\alpha \in (V \cup \Sigma)^*$.

By (iii), A appears in u_{i+1}.

By repeated application of (iii), we can conclude that A appears in $u_{i+2}, u_{i+3}, \cdots u_n$ and w, which is a contradiction because w contains no variables.

Example 2.9. Let $G = (\{S\}, \{0,1\}, R, S)$ be a CFG.

Create the rules in R so that $L(G) = \{0^n 1^{2n+1} \mid n \in \mathbb{N}\}$.

The rule is $S \longrightarrow 0S11 \mid 1$ as can be seen from the following applications of the rule.

$S \longrightarrow 0S11$ (1^{st} application of $S \longrightarrow 0S11$)

$\longrightarrow 00S1111$ (2^{nd} application of $S \longrightarrow 0S11$)

$\longrightarrow 000S111111$ (3^{rd} application of $S \longrightarrow 0S11$)

⋮

⋮

$\longrightarrow 0^n S 1^{2n}$ (n^{th} application of $S \longrightarrow 0S11$)

$\longrightarrow 0^n 1^{2n+1}$ (Application of $S \longrightarrow 1$)

Example 2.10. Let $G = (\{S\}, \{0,1\}, R, S)$ be a CFG.

Create the rules in R so that $L(G) = \{0^{2n} 1^{3n} \mid n \in \mathbb{N}\}$.

The rule is $S \longrightarrow 00S111 \mid \epsilon$ as can be seen from the following applications of the rule.

$S \longrightarrow 00S111$ (1^{st} application of $S \longrightarrow 00S111$)

$\longrightarrow 0000S111111$ (2^{nd} application of $S \longrightarrow 00S111$)

$\longrightarrow 000000S111111111$ (3^{rd} application of $S \longrightarrow 00S111$)

⋮

⋮

$\longrightarrow 0^{2n} S 1^{3n}$ (n^{th} application of $S \longrightarrow 00S111$)

$\longrightarrow 0^{2n} \epsilon 1^{3n}$ (Application of $S \longrightarrow \epsilon$)

$\longrightarrow 0^{2n} 1^{3n}$

Example 2.11. Let $G = (\{S\}, \{0,1\}, R, S)$ be a CFG.

Create the rules in R so that $L(G) = \{0^{2n+7} 1^{3n+9} \mid n \in \mathbb{N}\}$.

The rule is $S \longrightarrow 00S111 \mid 0^7 1^9$ as can be seen from the following applications of the rule.

$S \longrightarrow 00S111$ (1^{st} application of $S \longrightarrow 00S111$)

$\longrightarrow 0000S111111$ (2^{nd} application of $S \longrightarrow 00S111$)

$\longrightarrow 000000S111111111$ (3^{rd} application of $S \longrightarrow 00S111$)

 doi:10.20944/preprints202506.0158.v1

⋮

⋮

$\longrightarrow 0^{2n}S1^{3n}$ (n^{th} application of $S \longrightarrow 00S111$)

$\longrightarrow 0^{2n}0^{7}1^{9}1^{3n}$ (Application of $S \longrightarrow 0^{7}1^{9}$)

$\longrightarrow 0^{2n+7}1^{3n+9}$

Definition 2.12. Let $G = (V, \Sigma, R, S)$ be a CFG.

Let $R_1, R_2, R_3, \cdots\cdots R_n$ *and* Q be rules in R where $n \geq 1$.

$(R_1, R_2, R_3, \cdots\cdots R_n)$ and Q are equivalent if $\forall u \in (V \cup \Sigma)^*$, there exists $v \in (V \cup \Sigma)^*$ such that $u \xrightarrow{n,(R_1,R_2,\cdots R_n)} v$ if and only if $u \overset{Q}{\Rightarrow} v$

Proposition 2.13.

(i) $A \overset{A\to\alpha}{\Longrightarrow} \alpha$ if and only if $A \to \alpha$ is a rule.

(ii) If A does not appear in α and A does not appear in β and $A \to \alpha$ is a rule, then

$\alpha A\beta \overset{A\to\gamma}{\Longrightarrow} x \; iff \; x = \alpha\gamma\beta$.

Proof.

(i) If $A \to \alpha$ is a rule, $\epsilon A\epsilon \overset{A\to\alpha}{\Longrightarrow} \epsilon\alpha\epsilon$ and therefore, $A \overset{A\to\alpha}{\Longrightarrow} \alpha$.

Conversely, if $A \overset{A\to\alpha}{\Longrightarrow} \alpha$, by definition $A \to \alpha$ is a rule.

(ii) If $x = \alpha\gamma\beta$, since $A \to \gamma$, by definition $\alpha A\beta \overset{A\to\gamma}{\Longrightarrow} \alpha\gamma\beta$. Therefore, $\alpha A\beta \overset{A\to\gamma}{\Longrightarrow} x$.

Conversely, if $\alpha A\beta \overset{A\to\gamma}{\Longrightarrow} x$, $\exists u_1, u_2 \in (V \cup \Sigma)^*$ such that $\alpha A\beta = u_1 A u_2$ and $x = u_1\gamma u_2$.

Since A does not appear in α and A does not appear in β, there is only one appearance of A in $\alpha A\beta$.

Therefore, there is only one appearance of A in $u_1 A u_2$.

Therefore, $(\alpha A\beta = u_1 A u_2) \Longrightarrow (\alpha = u_1 \;\&\; \beta = u_2)$.

Therefore, $x = \alpha\gamma\beta$.

Proposition 2.14. $(A \longrightarrow \alpha)$ & $(B \longrightarrow \beta)$ are equivalent if and only if $(A = B)$ & $(\alpha = \beta)$.

Proof. If $(A \longrightarrow \alpha)$ & $(B \longrightarrow \beta)$ are equivalent,

$(A \longrightarrow \alpha) \Longrightarrow (A \overset{A\to\alpha}{\Longrightarrow} \alpha)$ (Proposition 2.13)

Since $(A \longrightarrow \alpha)$ & $(B \longrightarrow \beta)$ are equivalent, $A \overset{B\to\beta}{\Longrightarrow} \alpha$.

There exist $w_1, w_2 \in (V \cup \Sigma)^*$ such that $A = w_1 B w_2$ and $\alpha = w_1\beta w_2$.

$A = w_1 B w_2 \Longrightarrow A = B$ since A and B are both variables.

$A = w_1 A w_2 \Longrightarrow w_1 = w_2 = \epsilon$.

Therefore, $\alpha = \beta$.

Conversely, if $(A = B)$ & $(\alpha = \beta)$, $(A \longrightarrow \alpha)$ & $(B \longrightarrow \beta)$ are the same rule and hence they are equivalent.

Proposition 2.15. Let $G = (V, \Sigma, R, S)$ be a CFG.

$\forall A, B \in V$ and $x, y, z \in (V \cup \Sigma)^*$, $(A \longrightarrow xBz)$ & $(B \longrightarrow y)$ are equivalent to $A \longrightarrow xyz$.

 doi:10.20944/preprints202506.0158.v1

Proof. $\forall u \in (V \cup \Sigma)^*$, let

R_1 be $A \longrightarrow xBz$

R_2 be $B \longrightarrow y$

R_3 be $A \longrightarrow xyz$

If $u \overset{2,(R_1,R_2)}{\Longrightarrow} v$, $\exists w_1 \in (V \cup \Sigma)^*$ such that $u \overset{R_1}{\Rightarrow} w_1 \overset{R_2}{\Rightarrow} v$

Since $u \overset{R_1}{\Rightarrow} w_1$, $u = \alpha_1 A \alpha_2$ and $w_1 = \alpha_1 xBz\alpha_2$ for some $\alpha_1, \alpha_2 \in (V \cup \Sigma)^*$.

Since $R_2 = (B \longrightarrow y)$, $\alpha_1 xBz\alpha_2 \overset{R_2}{\Rightarrow} \alpha_1 xyz\alpha_2$.

That is, $w_1 \overset{R_2}{\Rightarrow} \alpha_1 xyz\alpha_2$.

Therefore, $\exists v = \alpha_1 xyz\alpha_2$ such that $u \overset{2,(R_1,R_2)}{\Longrightarrow} v$.

Since $R_3 = (A \longrightarrow xyz)$, $\alpha_1 A \alpha_2 \overset{R_3}{\Rightarrow} \alpha_1 xyz\alpha_2$.

That is, $u \overset{R_3}{\Rightarrow} \alpha_1 xyz\alpha_2$.

Therefore, $u \overset{R_3}{\Rightarrow} v$.

Conversely, if $u \overset{R_3}{\Rightarrow} v$, $u = \alpha_1 A \alpha_2$ and $v = \alpha_1 xyz\alpha_2$ for some $\alpha_1, \alpha_2 \in (V \cup \Sigma)^*$.

Let $w_1 = \alpha_1 xBz\alpha_2$.

Since $R_1 = (A \longrightarrow xBz)$, $\alpha_1 A \alpha_2 \overset{R_1}{\Rightarrow} \alpha_1 xBz\alpha_2$.

Since $R_2 = (B \longrightarrow y)$, $\alpha_1 xBz\alpha_2 \overset{R_2}{\Rightarrow} \alpha_1 xyz\alpha_2$.

Therefore, $\alpha_1 A \alpha_2 \overset{R_1}{\Rightarrow} \alpha_1 xBz\alpha_2 \overset{R_2}{\Rightarrow} \alpha_1 xyz\alpha_2$.

That is, $u \overset{R_1}{\Rightarrow} w_1 \overset{R_2}{\Rightarrow} v$.

That is, $u \overset{2,(R_1,R_2)}{\Longrightarrow} v$

Combining both directions, (R_1, R_2) and R_3 are equivalent.

Proposition 2.16. Let $G = (V, \Sigma, R, S)$ be a CFG.

(a) $\forall A, B \in V$, $x, y, z \in (V \cup \Sigma)^*$, if $A \Longrightarrow xBz$ & $B \Longrightarrow y$ then $A \overset{*}{\Rightarrow} xyz$.

(b) $\forall \alpha, \beta, \gamma, \beta' \in (V \cup \Sigma)^*$, if $\beta \Longrightarrow \beta'$ then $\alpha\beta\gamma \Longrightarrow \alpha\beta'\gamma$.

(c) $\forall \alpha, \beta, \gamma, \beta' \in (V \cup \Sigma)^*$, if $\beta \overset{*}{\Rightarrow} \beta'$ then $\alpha\beta\gamma \overset{*}{\Rightarrow} \alpha\beta'\gamma$.

(d) Let $\alpha_1, \alpha_2, \cdots \alpha_n, \beta_1, \beta_2, \cdots \beta_n, \gamma_1, \gamma_2, \cdots \gamma_n \in (V \cup \Sigma)^*$.

If $\beta_i \overset{*}{\Rightarrow} \gamma_i$ for $i \in \{1,2,3, \cdots n\}$, then $\alpha_1\beta_1\alpha_2\beta_2\alpha_3\beta_3 \cdots \alpha_n\beta_n \overset{*}{\Rightarrow} \alpha_1\gamma_1\alpha_2\gamma_2\alpha_3\gamma_3 \cdots \alpha_n\gamma_n$.

In the special case of $\alpha_1 = \alpha_2, \cdots = \alpha_n = \epsilon$, $\beta_1\beta_2\beta_3 \cdots \beta_n \overset{*}{\Rightarrow} \gamma_1\gamma_2\gamma_3 \cdots \gamma_n$.

Proof.

(a)

By Proposition 2.8 (i), $(B \Longrightarrow y) \Leftrightarrow (B \longrightarrow y)$.

By definition of derivation, $xBz \Longrightarrow xyz$.

Therefore, $A \Longrightarrow xBz \Longrightarrow xyz$.

Therefore, $A \overset{*}{\Rightarrow} xyz$.

(b)

Since $\beta \Longrightarrow \beta'$, $\exists \beta_1, \beta_2 \in (V \cup \Sigma)^*$, $A \in V$, $\eta \in (V \cup \Sigma)^*$ and a rule $A \longrightarrow \eta$ such that $\beta = \beta_1 A \beta_2$, $\beta' = \beta_1 \eta \beta_2$.

Therefore, $\alpha\beta\gamma = \alpha\beta_1 A \beta_2\gamma$ and $\alpha\beta'\gamma = \alpha\beta_1\eta\beta_2\gamma$.

Therefore, $\alpha\beta\gamma \Longrightarrow \alpha\beta'\gamma$.

(c)

Since $\beta \overset{*}{\Rightarrow} \beta'$, $\exists u_1, u_2 \cdots u_n \in (V \cup \Sigma)^*$ where $n \geq 0$ such that

$\beta \Longrightarrow u_1 \Longrightarrow u_2 \Longrightarrow u_3 \cdots u_{n-1} \Longrightarrow u_n \Longrightarrow \beta'$.

$\alpha\beta\gamma \Longrightarrow \alpha u_1\gamma$ $(\beta \Longrightarrow u_1$ & (b))

$\Longrightarrow \alpha u_2\gamma$ $(u_1 \Longrightarrow u_2$ & (b))

$\vdots$

$\Longrightarrow \alpha u_n\gamma$ $(u_{n-1} \Longrightarrow u_n$ & (b))

 doi:10.20944/preprints202506.0158.v1

$\Rightarrow \alpha\beta'\gamma$ $(u_n \Rightarrow \beta'.$ & (b))

Therefore, $\alpha\beta\gamma \overset{*}{\Rightarrow} \alpha\beta'\gamma$.

(d)

$\alpha_1\beta_1\alpha_2\beta_2\alpha_3\beta_3\cdots\alpha_n\beta_n \overset{*}{\Rightarrow} \alpha_1\gamma_1\alpha_2\beta_2\alpha_3\beta_3\cdots\alpha_n\beta_n$ $(\beta_1 \overset{*}{\Rightarrow} \gamma_1$ & (c))

$\overset{*}{\Rightarrow} \alpha_1\gamma_1\alpha_2\gamma_2\alpha_3\beta_3\cdots\alpha_n\beta_n$ $(\beta_2 \overset{*}{\Rightarrow} \gamma_2$ & (c))

$\overset{*}{\Rightarrow} \alpha_1\gamma_1\alpha_2\gamma_2\alpha_3\gamma_3\cdots\alpha_n\beta_n$ $(\beta_3 \overset{*}{\Rightarrow} \gamma_3$ & (c))

$\vdots$

$\overset{*}{\Rightarrow} \alpha_1\gamma_1\alpha_2\gamma_2\alpha_3\gamma_3\cdots\alpha_n\gamma_n$ $(\beta_n \overset{*}{\Rightarrow} \gamma_n$ & (c))

Therefore, $\alpha_1\beta_1\alpha_2\beta_2\alpha_3\beta_3\cdots\alpha_n\beta_n \overset{*}{\Rightarrow} \alpha_1\gamma_1\alpha_2\gamma_2\alpha_3\gamma_3\cdots\alpha_n\gamma_n$.

This completes the proof of Proposition 2.16.

By replacing $\Rightarrow$ with $\overset{lm}{\Rightarrow}$ and $\overset{*}{\Rightarrow}$ with $\overset{*,lm}{\Longrightarrow}$, we have the following proposition.

Proposition 2.17. Let $G = (V, \Sigma, R, S)$ be a CFG.

(a) $\forall A, B \in V,\ x, y, z \in (V \cup \Sigma)^*$, if $A \overset{lm}{\Rightarrow} xBz$ & $B \overset{lm}{\Rightarrow} y$ then $A \overset{*,lm}{\Longrightarrow} xyz$.

(b) $\forall \alpha, \beta, \gamma, \beta' \in (V \cup \Sigma)^*$, if $\beta \overset{lm}{\Rightarrow} \beta'$ then $\alpha\beta\gamma \overset{lm}{\Rightarrow} \alpha\beta'\gamma$.

(c) $\forall \alpha, \beta, \gamma, \beta' \in (V \cup \Sigma)^*$, if $\beta \overset{*,lm}{\Longrightarrow} \beta'$ then $\alpha\beta\gamma \overset{*,lm}{\Longrightarrow} \alpha\beta'\gamma$.

(d) Let $\alpha_1, \alpha_2, \cdots \alpha_n, \beta_1, \beta_2, \cdots \beta_n, \gamma_1, \gamma_2, \cdots \gamma_n \in (V \cup \Sigma)^*$.

If $\beta_i \overset{*,lm}{\Longrightarrow} \gamma_i$ for $i \in \{1,2,3,\cdots n\}$, then $\alpha_1\beta_1\alpha_2\beta_2\alpha_3\beta_3\cdots\alpha_n\beta_n \overset{*,lm}{\Longrightarrow} \alpha_1\gamma_1\alpha_2\gamma_2\alpha_3\gamma_3\cdots\alpha_n\gamma_n$.

In the special case of $\alpha_1 = \alpha_2, \cdots = \alpha_n = \epsilon$, $\beta_1\beta_2\beta_3\cdots\beta_n \overset{*,lm}{\Longrightarrow} \gamma_1\gamma_2\gamma_3\cdots\gamma_n$.

Proposition 2.18. Let $G = (V, \Sigma, R, S)$ be a CFG.

$A, A_1, A_2, \cdots\cdots A_n \in V,\ \alpha_1, \alpha_2, \cdots \alpha_n, \beta_1, \beta_2, \cdots \beta_n \in (V \cup \Sigma)^*$.

If $A \longrightarrow \alpha_1 A_1 \beta_1,\ A_1 \longrightarrow \alpha_2 A_2 \beta_2,\ \cdots A_{n-1} \longrightarrow \alpha_n A_n \beta_n$, then $A \overset{*}{\Rightarrow} \alpha_1 \cdots \alpha_{n-1}\alpha_n A_n \beta_n \beta_{n-1} \cdots \beta_1$.

Proof.

The proof is by induction on n.

$(n = 1)$

If $A \longrightarrow \alpha_1 A_1 \beta_1$, by Proposition 2.8 (i), $A \Longrightarrow \alpha_1 A_1 \beta_1$.

Therefore, $A \overset{*}{\Rightarrow} \alpha_1 A_1 \beta_1$.

$(n = k + 1, k \geq 1)$

Assume $A \longrightarrow \alpha_1 A_1 \beta_1,\ A_1 \longrightarrow \alpha_2 A_2 \beta_2,\ \cdots A_{k-1} \longrightarrow \alpha_k A_k \beta_k,\ A_k \longrightarrow \alpha_{k+1} A_{k+1} \beta_{k+1}$.

By induction hypothesis, $A \overset{*}{\Rightarrow} \alpha_1 \cdots \alpha_{k-1}\alpha_k A_k \beta_k \beta_{k-1} \cdots \beta_1$.

Since $A_k \longrightarrow \alpha_{k+1} A_{k+1} \beta_{k+1}$, by definition of derivation,

$\alpha_1 \cdots \alpha_{k-1}\alpha_k A_k \beta_k \beta_{k-1} \cdots \beta_1 \Longrightarrow \alpha_1 \cdots \alpha_{k-1}\alpha_k \alpha_{k+1} A_{k+1} \beta_{k+1} \beta_k \beta_{k-1} \cdots \beta_1$.

Therefore, $A \overset{*}{\Rightarrow} \alpha_1 \cdots \alpha_{k-1}\alpha_k A_k \beta_k \beta_{k-1} \cdots \beta_1 \Longrightarrow \alpha_1 \cdots \alpha_{k-1}\alpha_k \alpha_{k+1} A_{k+1} \beta_{k+1} \beta_k \beta_{k-1} \cdots \beta_1$.

Therefore, $A \overset{*}{\Rightarrow} \alpha_1 \cdots \alpha_{k-1}\alpha_k \alpha_{k+1} A_{k+1} \beta_{k+1} \beta_k \beta_{k-1} \cdots \beta_1$.

This completes the proof of Proposition 2.18.

Proposition 2.19. Let $G = (V, \Sigma, R, S)$ be a CFG, $B \in V, R_1, R_2, R_3, \cdots\cdots R_n$ be rules in R where $n \geq 0$.

Let $\alpha_1, \alpha_2, \alpha_1', \alpha_2' \in (V \cup \Sigma)^*$.

If $\alpha_1 B \alpha_2 \overset{n,(R_1,R_2,\cdots R_n)}{\Longrightarrow} \alpha_1' B \alpha_2'$ then $\alpha_1 x \alpha_2 \overset{n,(R_1,R_2,\cdots R_n)}{\Longrightarrow} \alpha_1' x \alpha_2'$ $\forall x \in (V \cup \Sigma)^*$ where the two $B's$ in the two strings are the same B (Note that there can be more than one B in the string $\alpha_1 B \alpha_2$) and B is not the head of any rule R_i $(i = 1,2,\cdots\cdots n)$.

(Note that when $n = 0$, the statement becomes

$(\alpha_1 B \alpha_2 = \alpha_1' B \alpha_2') \Longrightarrow (\alpha_1 x \alpha_2 = \alpha_1' x \alpha_2'$ $\forall x \in (V \cup \Sigma)^*)$

 doi:10.20944/preprints202506.0158.v1

Proof. For $n = 0$, $\alpha_1 B \alpha_2 = \alpha_1' B \alpha_2'$

Since the two $B's$ in the two strings are the same B, replacing them with x must yield two equal strings.

Therefore, $\alpha_1 x \alpha_2 = \alpha_1' x \alpha_2'$.

Therefore, the statement is true for $n = 0$.

For $n = 1$, if $\alpha_1 B \alpha_2 \overset{R_1}{\Rightarrow} \alpha_1' B \alpha_2'$,

Let $A \longrightarrow \alpha$ be the rule represented by R_1.

By definition of yielding, $\exists u_1, u_2 \in (V \cup \Sigma)^*$ such that

$\alpha_1 B \alpha_2 = u_1 A u_2$ and $\alpha_1' B \alpha_2' = u_1 \alpha u_2$.

The B that appears in $\alpha_1 B \alpha_2$ must also appear in $u_1 A u_2$.

Since R_1does not originate from this particular B, A and B cannot be the same object in the string $\alpha_1 B \alpha_2$ or $u_1 A u_2$ and hence there are only two cases to examine: B appears in u_1 or B appears in u_2.

(i) If B appears in u_1

Let u_1' be the string obtained by replacing B in u_1 with x.

Since $\alpha_1 B \alpha_2 = u_1 A u_2$, replacing B with x on both sides would yield two equal strings.

That is, $\alpha_1 x \alpha_2 = u_1' A u_2$.

Since $\alpha_1' B \alpha_2' = u_1 \alpha u_2$, replacing B with x on both sides would yield two equal strings.

That is, $\alpha_1' x \alpha_2' = u_1' \alpha u_2$.

However, $u_1' A u_2 \overset{R_1}{\Rightarrow} u_1' \alpha u_2$ since $A \longrightarrow \alpha$ is a rule.

Therefore, $\alpha_1 x \alpha_2 \overset{R_1}{\Rightarrow} \alpha_1' x \alpha_2'$.

Therefore, the statement is true for $n = 1$.

(ii) If B appears in u_2, with a similar argument, we can show that the statement is also true for $n = 1$.

With the results established on $n = 0$ and $n = 1$ and an induction argument, we can conclude that for $n \geq 0$,

$$(\alpha_1 B \alpha_2 \overset{n,(R_1,R_2,\cdots R_n)}{\Longrightarrow} \alpha_1' B \alpha_2') \Longrightarrow (\alpha_1 x \alpha_2 \overset{n,(R_1,R_2,\cdots R_n)}{\Longrightarrow} \alpha_1' x \alpha_2' \; \forall x \in (V \cup \Sigma)^*)$$

Proposition 2.20. If $G = (V, \Sigma, R, S)$ is a CFG and there exist $u_1, u_2, \cdots u_k \in (V \cup \Sigma)^*$, $w \in \Sigma^*$ such that $S \Longrightarrow u_1 \Longrightarrow u_2 \Longrightarrow u_3 \cdots \Longrightarrow u_r \Longrightarrow \cdots \Longrightarrow u_k \Longrightarrow w$, then

The # of variables in $u_r \leq$ the # of steps remaining from u_r to w.

Proof. Let n be the number of steps remaining from u_r to w.

$n = k + 1 - r$

We'll prove this proposition by induction on n.

(For $n = 1$)

$r = k$

Therefore, $u_r = u_k \Longrightarrow w$.

Since $u_k \Longrightarrow w$, $\exists \alpha, \beta, \gamma \in (V \cup \Sigma)^*$ and a rule $A \longrightarrow \gamma$ such that

$u_k = \alpha A \beta$ and $w = \alpha \gamma \beta$.

Since $w \in \Sigma^*$, $\alpha, \beta \in \Sigma^*$.

Therefore, u_k has only one variable.

Therefore, u_r has only one variable.

Therefore, # of variables in $u_r \leq$ the # of steps remaining from u_r to w.

(For induction)

The # of steps remaining from u_{r-1} to w is $n + 1 = k + 2 - r$.

Since $u_{r-1} \Longrightarrow u_r$, $\exists \alpha, \beta, \gamma \in (V \cup \Sigma)^*$ and a rule $A \longrightarrow \gamma$ such that

$u_{r-1} = \alpha A \beta$ and $u_r = \alpha \gamma \beta$.

Let m be the number of variables in u_{r-1}.

 doi:10.20944/preprints202506.0158.v1

# of variables in $u_r = m - 1 +$ (# of variables in γ) $\geq m - 1$.
By induction hypothesis, # of variables in $u_r \leq n = k + 1 - r$.
Therefore, $m - 1 \leq$ # of variables in $u_r \leq n = k + 1 - r$.
Therefore, $m - 1 \leq k + 1 - r$.
Therefore, $m \leq k + 2 - r$.
Therefore, number of variables in $u_{r-1} \leq$ # of steps remaining from u_{r-1} to w.
This completes the proof of Proposition 2.20.

Example 2.21. Let $G = (V, \Sigma, R, S)$ be a CFG and there exist $u_1, u_2, \cdots u_k \in (V \cup \Sigma)^*$, $w \in \Sigma^*$ such that $S \Longrightarrow u_1 \Longrightarrow u_2 \Longrightarrow u_3 \cdots \Longrightarrow u_r \Longrightarrow \cdots \Longrightarrow u_k \Longrightarrow w$. Show that the statement
(# of variables in $u_r =$ the # of steps remaining from u_r to w) is not always true.
(*Hint:* Consider $V = \{S, A, B, C\}, \Sigma = \{a, b, c\}, R = \{S \longrightarrow AB, A \longrightarrow C, C \longrightarrow c, B \longrightarrow b\}$ and
$S \overset{S \to AB}{\Longrightarrow} AB \overset{A \to C}{\Longrightarrow} CB \overset{C \to c}{\Longrightarrow} cB \overset{B \to b}{\Longrightarrow} cb$.)

Definition 2.22. $\forall \alpha, \beta \in (V \cup \Sigma)^*, \alpha$ is a substring of β (written as $\alpha \sqsubset \beta$) if $\exists \alpha', \alpha'' \in (V \cup \Sigma)^*$ such that
$\beta = \alpha' \alpha \alpha''$. α' is called the left complement of α in β, written as $LC(\alpha)$. α'' is called the right complement of α in β, written as $RC(\alpha)$.

Proposition 2.23. For any strings α_1, α_2, u such that $\alpha_1, \alpha_2 \sqsubset u$, if $\alpha_1 \sqsubset \alpha_2$, then
(i) $LC(\alpha_2) \sqsubset LC(\alpha_1)$ & $LC(\alpha_2) \cdot r = LC(\alpha_1)$ for some string r
(ii) $RC(\alpha_2) \sqsubset RC(\alpha_1)$ & $l \cdot RC(\alpha_2) = RC(\alpha_1)$ for some string l.

Proof. $\alpha_1 \sqsubset u \Longrightarrow u = x_1 \alpha_1 y_1$ for some strings x_1, y_1.
$\alpha_2 \sqsubset u \Longrightarrow u = x_2 \alpha_2 y_2$ for some strings x_2, y_2.
$x_1 = LC(\alpha_1)$; $y_1 = RC(\alpha_1)$.
$x_2 = LC(\alpha_2)$; $y_2 = RC(\alpha_2)$.
$\alpha_1 \sqsubset \alpha_2 \Longrightarrow \alpha_2 = r \alpha_1 l$ for some strings r & l.
Therefore, $u = x_2 \alpha_2 y_2 = x_2 r \alpha_1 l y_2$.
Since $u = x_1 \alpha_1 y_1$, $x_1 \alpha_1 y_1 = x_2 r \alpha_1 l y_2$.
Therefore, $x_1 = x_2 r$ and $y_1 = l y_2$.
Therefore, $LC(\alpha_1) = LC(\alpha_2) \cdot r$ and $RC(\alpha_1) = l \cdot RC(\alpha_2)$.
Therefore, $LC(\alpha_2) \sqsubset LC(\alpha_1)$ and $RC(\alpha_2) \sqsubset RC(\alpha_1)$.
This completes the proof of Proposition 2.23.

Definition 2.24. For any strings α_1, α_2, u such that $\alpha_1, \alpha_2 \sqsubset u$, α_1 is said to be on the left of α_2 if there exist strings x, y, z such that $u = x \alpha_1 y \alpha_2 z$.

Proposition 2.25. Let $G = (V, \Sigma, R, S)$ be a CFG.
Let $u_0 \Longrightarrow u_1 \Longrightarrow u_2 \Longrightarrow u_3 \cdots \cdots \Longrightarrow u_n$, where $u_0, u_1, u_2, u_3 \cdots u_n \in (V \cup \Sigma)^*$.
Let $0 \leq i < j \leq n$.
If $\alpha_i \sqsubset u_i$, then $\exists \alpha_{i+1}, \alpha_{i+2}, \cdots \alpha_j$ where $\alpha_{i+1} \sqsubset u_{i+1}, \alpha_{i+2} \sqsubset u_{i+2}, \cdots \alpha_j \sqsubset u_j$ such that
$\alpha_i \overset{\lambda_1}{\Rightarrow} \alpha_{i+1} \overset{\lambda_2}{\Rightarrow} \alpha_{i+2} \overset{\lambda_3}{\Rightarrow} \alpha_{i+3} \cdots \cdots \overset{\lambda_{j-i}}{\Longrightarrow} \alpha_j$ where $\lambda_1, \lambda_2, \cdots \cdots \lambda_{j-i} \in \{0,1\}$.

Hence, $\alpha_i \overset{*}{\Rightarrow} \alpha_j$ in no more than $j - i$ steps.

α_j is called the $(j - i)$-step expansion of α_i within the derivation of $u_0 \overset{*}{\Rightarrow} u_n$ and it is written as $\alpha_j = Expan(\alpha_i, j - i)$.

 doi:10.20944/preprints202506.0158.v1

Proof. Let $k = j - i$.

$1 \leq k \leq n$

This proposition can be proved by induction on k.

$(k = 1)$:

$j = i + 1$

Since $u_i \Longrightarrow u_{i+1}$, $\exists \alpha, \beta, \gamma \in (V \cup \Sigma)^*$ and a rule $A \longrightarrow \gamma$ such that

$u_i = \alpha A \beta$ and $u_{i+1} = \alpha\gamma\beta$.

Since $\alpha_i \sqsubset u_i$, $\exists \alpha', \beta' \in (V \cup \Sigma)^*$ such that $u_i = \alpha'\alpha_i\beta'$.

(i) If $A \sqsubset \alpha_i$

$\exists \alpha'', \beta'' \in (V \cup \Sigma)^*$ such that $\alpha_i = \alpha'' A \beta''$.

$u_i = \alpha'\alpha_i\beta' = \alpha'\alpha'' A \beta''\beta'$.

Also, $u_i = \alpha A \beta$.

Therefore, $\alpha A \beta = \alpha'\alpha'' A \beta''\beta'$.

Therefore, $\alpha = \alpha'\alpha''$ and $\beta = \beta''\beta'$.

Since $u_{i+1} = \alpha\gamma\beta$, $u_{i+1} = \alpha'\alpha''\gamma\beta''\beta'$.

Take $\alpha_{i+1} = \alpha''\gamma\beta''$.

Since $\alpha_i = \alpha'' A \beta''$ and $A \longrightarrow \gamma$ is a rule, $\alpha_i \Longrightarrow \alpha_{i+1}$.

(ii) If A is not a substring of α_i

Since $u_i = \alpha A \beta$ and $\alpha_i \sqsubset u_i$, either $\alpha_i \sqsubset \alpha$ or $\alpha_i \sqsubset \beta$.

Since $u_{i+1} = \alpha\gamma\beta$, $\alpha \sqsubset u_{i+1}$ and $\beta \sqsubset u_{i+1}$.

Therefore $(\alpha_i \sqsubset \alpha$ or $\alpha_i \sqsubset \beta) \Longrightarrow \alpha_i \sqsubset u_{i+1}$.

Take $\alpha_{i+1} = \alpha_i$.

Therefore, $\alpha_i \overset{0}{\Rightarrow} \alpha_{i+1}$.

Combining (i) and (ii), $\alpha_i \overset{\lambda_1}{\Rightarrow} \alpha_{i+1}$ where $\lambda_1 \in \{0,1\}$.

(Induction):

By induction assumption,

$\alpha_i \overset{\lambda_1}{\Rightarrow} \alpha_{i+1} \overset{\lambda_2}{\Rightarrow} \alpha_{i+2} \overset{\lambda_3}{\Rightarrow} \alpha_{i+3} \cdots\cdots \overset{\lambda_{j-i}}{\Longrightarrow} \alpha_j$ where $\lambda_1, \lambda_2, \cdots\cdots \lambda_{j-i} \in \{0,1\}$ and

$\alpha_{i+1} \sqsubset u_{i+1}, \alpha_{i+2} \sqsubset u_{i+2}, \cdots \alpha_j \sqsubset u_j$.

Since $\alpha_j \sqsubset u_j$ and $u_j \Longrightarrow u_{j+1}$, by applying the same argument as in the case of $(k = 1)$, we can find $\alpha_{j+1} \sqsubset u_{j+1}$ such that $\alpha_j \overset{\lambda_{j-i+1}}{\Longrightarrow} \alpha_{j+1}$ where $\lambda_{j-i+1} \in \{0,1\}$.

We now have $\alpha_i \overset{\lambda_1}{\Rightarrow} \alpha_{i+1} \overset{\lambda_2}{\Rightarrow} \alpha_{i+2} \overset{\lambda_3}{\Rightarrow} \alpha_{i+3} \cdots\cdots \overset{\lambda_{j-i}}{\Longrightarrow} \alpha_j \overset{\lambda_{j-i+1}}{\Longrightarrow} \alpha_{j+1}$ where

$\lambda_1, \lambda_2, \cdots\cdots \lambda_{j-i}, \lambda_{j-i+1} \in \{0,1\}$ and $\alpha_{i+1} \sqsubset u_{i+1}, \alpha_{i+2} \sqsubset u_{i+2}, \cdots \alpha_j \sqsubset u_j, \alpha_{j+1} \sqsubset u_{j+1}$.

This completes the proof of Proposition 2.25.

Proposition 2.26. Let $G = (V, \Sigma, R, S)$ be a CFG.

Let $u_0 \Longrightarrow u_1 \Longrightarrow u_2 \Longrightarrow u_3 \cdots\cdots \Longrightarrow u_n$, where $u_0, u_1, u_2, u_3 \cdots u_n \in (V \cup \Sigma)^*$.

Let $0 \leq i < j \leq n$, $\alpha_i \sqsubset u_i$, $\alpha_i' \sqsubset u_i$, $\alpha_j = Expan(\alpha_i, j - i)$ & $\alpha_j' = Expan(\alpha_i', j - i)$.

If α_i is to the left of α_i' within u_i, then α_j is to the left of α_j' within u_j.

Proof. Let $k = j - 1$.

$0 \leq k \leq n$.

We can prove this proposition by induction on k.

$(k = 0)$

$j = i \Longrightarrow \alpha_i = \alpha_j$ & $\alpha_i' = \alpha_j'$.

Therefore, $\alpha_i \overset{0}{\Rightarrow} \alpha_j$ & $\alpha_i' \overset{0}{\Rightarrow} \alpha_j'$.

$\alpha_j = Expan(\alpha_i, 0)$ and $\alpha_j' = Expan(\alpha_i', 0)$.

α_i is to the left of $\alpha_i' \Longrightarrow \alpha_j$ is to the left of α_j'.

The statement is true for $k = 0$.

(Induction)

 doi:10.20944/preprints202506.0158.v1

Induction Hypothesis:

$(\alpha_i$ is to the left of $\alpha_i') \Rightarrow (\alpha_{i+k}$ is to the left of $\alpha_{i+k}')$.

Since α_{i+k} is to the left of α_{i+k}', there exist $x, y, z \in (V \cup \Sigma)^*$ such that

$u_{i+k} = x\alpha_{i+k}y\alpha_{i+k}'z$.

Since $u_{i+k} \Rightarrow u_{i+k+1}$, there exists a rule $A \longrightarrow \gamma$ such that

$u_{i+k} = \alpha A\beta$ & $u_{i+k+1} = \alpha\gamma\beta$.

We now have five situations to examine: $A \sqsubset x, A \sqsubset \alpha_{i+k}, A \sqsubset y, A \sqsubset \alpha_{i+k}', A \sqsubset z$.

(i) $A \sqsubset x$

$RC(A) = l \cdot RC(x)$ (Proposition 2.23)

$\beta = l \cdot \alpha_{i+k}y\alpha_{i+k}'z$ $(RC(A) = \beta,\ RC(x) = \alpha_{i+k}y\alpha_{i+k}'z)$

$u_{i+k+1} = \alpha\gamma\beta = \alpha\gamma l\alpha_{i+k}y\alpha_{i+k}'z$.

Take $\alpha_{i+k+1} = \alpha_{i+k}$ and $\alpha_{i+k+1}' = \alpha_{i+k}'$.

Therefore, $\alpha_{i+k} \overset{0}{\Rightarrow} \alpha_{i+k+1}$ & $\alpha_{i+k}' \overset{0}{\Rightarrow} \alpha_{i+k+1}'$.

In addition, $u_{i+k+1} = \alpha\gamma l\alpha_{i+k+1}y\alpha_{i+k+1}'z$.

Therefore, α_{i+k+1} is to the left of α_{i+k+1}'.

(ii) $A \sqsubset \alpha_{i+k}$

$\exists \alpha', \beta' \in (V \cup \Sigma)^*$ such that

$\alpha_{i+k} = \alpha' A\beta'$

$u_{i+k} = x\alpha_{i+k}y\alpha_{i+k}'z = x\alpha' A\beta' y\alpha_{i+k}'z$

Since $u_{i+k} = \alpha A\beta,\ \alpha = x\alpha'$ & $\beta = \beta' y\alpha_{i+k}'z$.

Therefore, $u_{i+k+1} = \alpha\gamma\beta = x\alpha'\gamma\beta' y\alpha_{i+k}'z$.

Take $\alpha_{i+k+1} = \alpha'\gamma\beta'$ & $\alpha_{i+k+1}' = \alpha_{i+k}'$.

Now, $u_{i+k+1} = x\alpha_{i+k+1}y\alpha_{i+k+1}'z$.

So, α_{i+k+1} is to the left of α_{i+k+1}'.

In addition, $\alpha_{i+k} \Rightarrow \alpha_{i+k+1}$ because $\alpha_{i+k} = \alpha' A\beta'$ & $\alpha_{i+k+1} = \alpha'\gamma\beta'$.

Also, $\alpha_{i+k}' \overset{0}{\Rightarrow} \alpha_{i+k+1}'$ because $\alpha_{i+k+1}' = \alpha_{i+k}'$.

(iii) $A \sqsubset y$

With a similar argument as in (i), we can show that $\exists \alpha_{i+k+1}, \alpha_{i+k+1}'$ in u_{i+k+1} such that $\alpha_{i+k} \overset{\lambda}{\Rightarrow} \alpha_{i+k+1}$ & $\alpha_{i+k}' \overset{\lambda'}{\Rightarrow} \alpha_{i+k+1}'$ where $\lambda, \lambda' \in \{0,1\}$ and

α_{i+k+1} is to the left of α_{i+k+1}'.

(iv) $A \sqsubset \alpha_{i+k}'$

With a similar argument as in (ii), we can show that $\exists \alpha_{i+k+1}, \alpha_{i+k+1}'$ in u_{i+k+1} such that $\alpha_{i+k} \overset{\lambda}{\Rightarrow} \alpha_{i+k+1}$ & $\alpha_{i+k}' \overset{\lambda'}{\Rightarrow} \alpha_{i+k+1}'$ where $\lambda, \lambda' \in \{0,1\}$ and

α_{i+k+1} is to the left of α_{i+k+1}'.

(v) $A \sqsubset z$

With a similar argument as in (i), we can show that $\exists \alpha_{i+k+1}, \alpha_{i+k+1}'$ in u_{i+k+1} such that $\alpha_{i+k} \overset{\lambda}{\Rightarrow} \alpha_{i+k+1}$ & $\alpha_{i+k}' \overset{\lambda'}{\Rightarrow} \alpha_{i+k+1}'$ where $\lambda, \lambda' \in \{0,1\}$ and

α_{i+k+1} is to the left of α_{i+k+1}'.

Combining all (i) to (v) and the induction hypothesis, we now have:

If α_i is to the left of α_i' within u_i, then α_{i+k+1} is to the left of α_{i+k+1}' within u_{i+k+1}.

This completes the proof of Proposition 2.26.

Proposition 2.27. Let $G = (V, \Sigma, R, S)$ be a CFG.

Let $u_0 \Rightarrow u_1 \Rightarrow \cdots \Rightarrow u_i \Rightarrow u_{i+1} \cdots \Rightarrow u_n$, where $u_0, u_1, \cdots u_i, u_{i+1} \cdots u_n \in (V \cup \Sigma)^*$ & $0 \leq i < n$.

Let $\alpha_i, \beta_i \in (V \cup \Sigma)^*$.

If $\alpha_i\beta_i \sqsubset u_i$, then $Expan(\alpha_i\beta_i, 1) = Expan(\alpha_i, 1)Expan(\beta_i, 1)$.

Proof. Since $\alpha_i\beta_i \sqsubset u_i$, $u_i = x\alpha_i\beta_i y$ for some $x, y \in (V \cup \Sigma)^*$.

Since $u_i \Rightarrow u_{i+1}$, $\exists \alpha, \beta, \gamma \in (V \cup \Sigma)^*$ and a rule $A \longrightarrow \gamma$ such that

 doi:10.20944/preprints202506.0158.v1

$u_i = \alpha A\beta$ & $u_{i+1} = \alpha\gamma\beta$.
Since $A \sqsubset u_i$ & $u_i = x\alpha_i\beta_i y$, we have four cases to examine:
$A \sqsubset x$, $A \sqsubset \alpha_i$, $A \sqsubset \beta_i$, $A \sqsubset y$.

(i) $A \sqsubset x$

$x = x'Ay'$ for some $x', y' \in (V \cup \Sigma)^*$.
$u_i = x\alpha_i\beta_i y = x'Ay'\alpha_i\beta_i y$.
Since u_i is also equal to $\alpha A\beta$, $\alpha A\beta = x'Ay'\alpha_i\beta_i y$.
Therefore, $\alpha = x'$ and $\beta = y'\alpha_i\beta_i y$.
Since $u_{i+1} = \alpha\gamma\beta$, $u_{i+1} = x'\gamma y'\alpha_i\beta_i y$.
Now we have $\alpha_i, \beta_i, \alpha_i\beta_i \sqsubset u_i$ & $\alpha_i, \beta_i, \alpha_i\beta_i \sqsubset u_{i+1}$.
In addition, $\alpha_i \overset{0}{\Rightarrow} \alpha_i$, $\beta_i \overset{0}{\Rightarrow} \beta_i$, $\alpha_i\beta_i \overset{0}{\Rightarrow} \alpha_i\beta_i$.
Therefore, $Expan(\alpha_i\beta_i, 1) = \alpha_i\beta_i$, $Expan(\alpha_i, 1) = \alpha_i$ and $Expan(\beta_i, 1) = \beta_i$.
Therefore, $Expan(\alpha_i\beta_i, 1) = Expan(\alpha_i, 1)Expan(\beta_i, 1)$.

(ii) $A \sqsubset \alpha_i$

$\alpha_i = x'Ay'$ for some $x', y' \in (V \cup \Sigma)^*$.
Since $u_i = x\alpha_i\beta_i y$, $u_i = xx'Ay'\beta_i y$.
Since $u_i = \alpha A\beta$, $\alpha A\beta = xx'Ay'\beta_i y$.
Therefore, $\alpha = xx'$ and $\beta = y'\beta_i y$.
Since $u_{i+1} = \alpha\gamma\beta$, $u_{i+1} = xx'\gamma y'\beta_i y$.
Let $\alpha_{i+1} = x'\gamma y'$. Then $u_{i+1} = x\alpha_{i+1}\beta_i y$.
Since $\alpha_i = x'Ay'$ and $A \longrightarrow \gamma$ is a rule, $\alpha_i \Longrightarrow \alpha_{i+1}$.
Since $\alpha_i \sqsubset u_i$ and $\alpha_{i+1} \sqsubset u_{i+1}$, $Expan(\alpha_i, 1) = \alpha_{i+1}$.
Since $\beta_i \sqsubset u_i$ and $\beta_i \sqsubset u_{i+1}$ and $\beta_i \overset{0}{\Rightarrow} \beta_i$, $Expan(\beta_i, 1) = \beta_i$.
Since $\alpha_i = x'Ay'$, $\alpha_i\beta_i = x'Ay'\beta_i$.
$x'Ay'\beta_i \Longrightarrow x'\gamma y'\beta_i$ because $A \longrightarrow \gamma$ is a rule.
Therefore, $\alpha_i\beta_i \Longrightarrow x'\gamma y'\beta_i$.
Therefore, $\alpha_i\beta_i \Longrightarrow \alpha_{i+1}\beta_i$ $\quad (\alpha_{i+1} = x'\gamma y')$
Since $\alpha_i\beta_i \sqsubset u_i$ and $\alpha_{i+1}\beta_i \sqsubset u_{i+1}$, $Expan(\alpha_i\beta_i, 1) = \alpha_{i+1}\beta_i$.
Therefore, $Expan(\alpha_i\beta_i, 1) = Expan(\alpha_i, 1)Expan(\beta_i, 1)$.

(iii) $A \sqsubset \beta_i$

With a similar argument as in (ii), we can show that $Expan(\alpha_i\beta_i, 1) = Expan(\alpha_i, 1)Expan(\beta_i, 1)$.

(iv) $A \sqsubset y$

With a similar argument as in (i), we can show that
$Expan(\alpha_i\beta_i, 1) = Expan(\alpha_i, 1)Expan(\beta_i, 1)$.
This completes the proof of Proposition 2.27.

Proposition 2.28. Let $G = (V, \Sigma, R, S)$ be a CFG.

Let $u_0 \Longrightarrow u_1 \Longrightarrow \cdots \Longrightarrow u_i \cdots \Longrightarrow u_n$, where $u_0, u_1, \cdots u_i, \cdots u_n \in (V \cup \Sigma)^*$ & $0 \le i \le n$.

(i) For $0 \le k \le n - i$, $Expan(\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_m}, k) = Expan(\alpha_{i_1}, k)Expan(\alpha_{i_2}, k)\cdots Expan(\alpha_{i_m}, k)$
where
$\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_m} \sqsubset u_i$.

(ii) If $u_0 = X_1X_2\cdots X_m$ where $X_1, X_2, \cdots X_m \in V \cup \Sigma$ & $u_n = w \in \Sigma^*$, then $\exists w_1, w_2 \cdots w_m \in \Sigma^*$ such that $X_i \overset{*}{\Rightarrow} w_i$ in no more than n steps &
$w = w_1w_2\cdots w_m$.

Proof. Claim.

$\forall\alpha_i, \beta_i \in (V \cup \Sigma)^*$ such that $\alpha_i\beta_i \sqsubset u_i$ and $0 \le k \le n - i$,
$Expan(\alpha_i\beta_i, k) = Expan(\alpha_i, k)Expan(\beta_i, k)$
This Claim can be proved by induction on k.
$(k = 0)$

 doi:10.20944/preprints202506.0158.v1

$Expan(\alpha_i\beta_i, 0) = \alpha_i\beta_i.$
$Expan(\alpha_i, 0) = \alpha_i$ and $Expan(\beta_i, 0) = \beta_i.$
Therefore, $Expan(\alpha_i\beta_i, 0) = Expan(\alpha_i, 0)Expan(\beta_i, 0).$
The statement is true for $k = 0.$
(Induction)
Induction Hypothesis:
$Expan(\alpha_i\beta_i, k) = Expan(\alpha_i, k)Expan(\beta_i, k)$ where
$Expan(\alpha_i\beta_i, k), Expan(\alpha_i, k), Expan(\beta_i, k) \sqsubset u_{i+k}.$
$Expan(\alpha_i\beta_i, k+1) = Expan(Expan(\alpha_i\beta_i, k), 1)$
$= Expan(Expan(\alpha_i, k)Expan(\beta_i, k), 1)$ (Induction Hypothesis)
$= Expan(Expan(\alpha_i, k), 1)Expan(Expan(\beta_i, k), 1)$ (Proposition 2.27)
$= Expan(\alpha_i, k+1)Expan(\beta_i, k+1)$
This completes the proof of Claim.
The proof of (i) is by induction on $m.$
$(m = 1)$
$LHS = Expan(\alpha_{i1}, k).$
$RHS = Expan(\alpha_{i1}, k).$
Therefore, the statement is true for $m = 1.$
(Induction)
Induction Hypothesis: $Expan(\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_m}, k) = Expan(\alpha_{i_1}, k)Expan(\alpha_{i_2}, k)\cdots Expan(\alpha_{i_m}, k).$
$Expan(\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_m}\alpha_{i_{m+1}}, k) = Expan(\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_m}, k)Expan(\alpha_{i_{m+1}}, k)$ (Claim)
$= Expan(\alpha_{i_1}, k)Expan(\alpha_{i_2}, k)\cdots Expan(\alpha_{i_m}, k)Expan(\alpha_{i_{m+1}}, k)$ (Induction Hypothesis)
This completes the proof of (i).
(i)
Set $i = 0$ & $k = n$ for the result in (i).
$\alpha_{0_1} = X_1, \alpha_{0_2} = X_2, \cdots \alpha_{0_m} = X_m.$
$u_0 = X_1X_2\cdots X_m = \alpha_{0_1}\alpha_{0_2}\cdots\alpha_{0_m}$
Therefore, $\alpha_{0_1}\alpha_{0_2}\cdots\alpha_{0_m} \sqsubset u_0.$
By (i), $Expan(\alpha_{0_1}\alpha_{0_2}\cdots\alpha_{0_m}, n) = Expan(\alpha_{0_1}, n)Expan(\alpha_{0_2}, n)\cdots Expan(\alpha_{0_m}, n)$
Therefore, $Expan(X_1X_2\cdots X_m, n) = Expan(X_1, n)Expan(X_2, n)\cdots Expan(X_m, n).$
$Expan(X_1X_2\cdots X_m, n) = Expan(u_0, n) = u_n = w.$
Therefore, $Expan(X_1, n)Expan(X_2, n)\cdots Expan(X_m, n) = w.$
Therefore, $Expan(X_i, n) = w_i$ for some $w_i \in \Sigma^*,\ i \in \{1,2,\cdots n\}.$
Therefore, $w = w_1w_2\cdots w_m$ &
$X_i \overset{*}{\Rightarrow} w_i$ in no more than n steps.

Proposition 2.29. Let $G = (V, \Sigma, R, S)$ be a $CFG,\ \alpha, \beta \in (V \cup \Sigma)^*,\ X \in V$ and $w \in \Sigma^*.$
If $\alpha X\beta \overset{*}{\Rightarrow} w$, then $X \overset{*}{\Rightarrow} w'$ for some $w' \in \Sigma^*.$

Proof. $\exists n \geq 1$ such that $\alpha X\beta \overset{n}{\Rightarrow} w.$
$Expan(\alpha X\beta, n) = w.$
$Expan(\alpha, n)Expan(X, n)Expan(\beta, n) = w$ (Proposition 2.28)
$Expan(X, n) = w'$ for some $w' \in \Sigma^*$
$X \overset{*}{\Rightarrow} w'$ for some $w' \in \Sigma^*$ (Proposition 2.25)
This completes the proof of Proposition 2.29.

Example 2.30. Prove that the non-regular set $A = \{a^n b^n | n \geq 0\}$ is a $CFL.$

Proof. Let $G = (V, \Sigma, R, S)$ be a CFG such that
$V = \{S\},\ \Sigma = \{a, b\},\ R = \{S \longrightarrow aSb, S \longrightarrow \epsilon\}.$

 doi:10.20944/preprints202506.0158.v1

In short form, $S \longrightarrow aSb|\epsilon$.

Claim 1. If $S \overset{n+1}{\Longrightarrow} \alpha \; \forall \, n \geq 0$ where $\alpha \in (V \cup \Sigma)^*$, then

$\exists \gamma \in (V \cup \Sigma)^*$ such that $S \overset{n}{\Rightarrow} \gamma$ and $\gamma \overset{1}{\Rightarrow} \alpha$ and $\gamma = a^n S b^n$.

Claim 1 can be proved by induction on n.

For $n = 0$, if $S \overset{1}{\Rightarrow} \alpha$, by definition, $\exists \gamma \in (V \cup \Sigma)^*$ such that $S \overset{0}{\Rightarrow} \gamma$ and $\gamma \overset{1}{\Rightarrow} \alpha$.

Therefore, $S = \gamma$.

Therefore, $\gamma = a^0 S b^0$.

Therefore, the statement is true for $n = 0$.

Assume the statement is true for $n = k$ for $k \geq 0$.

That is, $\left(S \overset{k+1}{\Longrightarrow} \alpha\right) \Longrightarrow \left(\exists \gamma \in (V \cup \Sigma)^* \; such \; that \; S \overset{k}{\Rightarrow} \gamma \; \& \; \gamma \overset{1}{\Rightarrow} \alpha \; \& \; \gamma = a^k S b^k\right)$ for $k \geq 0$ and $\alpha \in (V \cup \Sigma)^*$.

For $n = k + 1$, assume $S \overset{k+2}{\Longrightarrow} \alpha$.

By definition, $\exists \gamma' \in (V \cup \Sigma)^*$ such that

$S \overset{k+1}{\Longrightarrow} \gamma'$ and $\gamma' \overset{1}{\Rightarrow} \alpha$.

By induction assumption, $\exists \gamma \in (V \cup \Sigma)^*$ such that $S \overset{k}{\Rightarrow} \gamma$ and $\gamma \overset{1}{\Rightarrow} \gamma'$ and $\gamma = a^k S b^k$.

Since there are only two rules in R, namely $S \longrightarrow aSb \; or \; S \longrightarrow \epsilon$.

If we use $S \longrightarrow \epsilon$ on $\gamma \overset{1}{\Rightarrow} \gamma'$, then $a^k S b^k \overset{S \longrightarrow \epsilon}{\Longrightarrow} \gamma'$.

By Proposition 2.13 (ii), $\gamma' = a^k \epsilon b^k = a^k b^k$.

This contradicts the conclusion $\gamma' \overset{1}{\Rightarrow} \alpha$ we derive above because $a^k b^k$ does not contain a variable.

Therefore, we must use rule $S \longrightarrow aSb$.

Therefore, $\gamma \overset{S \longrightarrow aSb}{\Longrightarrow} \gamma'$.

Therefore, $a^k S b^k \overset{S \longrightarrow aSb}{\Longrightarrow} \gamma'$.

Again by Proposition 2.13 (ii), $\gamma' = a^k aSbb^k = a^{k+1} S b^{k+1}$.

This completes the proof of Claim 1.

Claim 2. $S \overset{n+1}{\Longrightarrow} a^n b^n \; \forall n \geq 0$.

For $n = 0$, $S \overset{S \longrightarrow \epsilon}{\Longrightarrow} \epsilon$ by Proposition 2.13 (i).

Therefore, $S \overset{1}{\Rightarrow} a^0 b^0$ and hence the statement is true for $n = 0$.

For $n \geq 1$, by Proposition 2.13 (i) & (ii),

$S \overset{S \longrightarrow aSb}{\Longrightarrow} aSb \overset{S \longrightarrow aSb}{\Longrightarrow} a^2 S b^2 \overset{S \longrightarrow aSb}{\Longrightarrow} a^3 S b^3 \cdots \cdots \overset{S \longrightarrow aSb}{\Longrightarrow} a^n S b^n$.

Therefore, $S \overset{n}{\Rightarrow} a^n S b^n$.

In addition, $a^n S b^n \overset{S \longrightarrow \epsilon}{\Longrightarrow} a^n b^n$ by Proposition 2.13 (ii).

Therefore, $S \overset{n+1}{\Longrightarrow} a^n b^n$.

This completes the proof of Claim 2.

It remains to show that $L(G) = A$.

$u \in A \Longrightarrow u = a^n b^n$

$\Longrightarrow S \overset{n+1}{\Longrightarrow} u$ (by Claim 2)

$\Longrightarrow u \in L(G)$

Conversely, if $u \in L(G)$, $u \in \Sigma^*$ and

$S \overset{n+1}{\Longrightarrow} u$ for some $n \geq 0$.

$\exists \gamma \in (V \cup \Sigma)^*$ such that $S \overset{n}{\Rightarrow} \gamma$ and $\gamma \overset{1}{\Rightarrow} u$ and $\gamma = a^n S b^n$ by Claim 1.

Since there are only two rules in R, either $\gamma \overset{S \longrightarrow aSb}{\Longrightarrow} u$ or $\gamma \overset{S \longrightarrow \epsilon}{\Longrightarrow} u$.

 doi:10.20944/preprints202506.0158.v1

$\left(\gamma \overset{S\to aSb}{\Longrightarrow} u\right)$

$\Rightarrow \left(a^nSb^n \overset{S\to aSb}{\Longrightarrow} u\right)$

$\Rightarrow a^{n+1}Sb^{n+1} = u$ (Proposition 2.13 (ii))

$\Rightarrow$ a contradiction to $u \in \Sigma^*$.

Therefore, we must use $\gamma \overset{S\to\epsilon}{\Longrightarrow} u$.

Therefore, $a^nSb^n \overset{S\to\epsilon}{\Longrightarrow} u$.

$a^n\epsilon b^n = u$ by Proposition 2.13 (ii).

Therefore, $u = a^nb^n$ and hence $u \in A$.

Combining both directions, $L(G) = A$.

Before proceeding to the proof of some important theorems in CFG, we need to review some Tree terminology and Graph Theory. The readers are assumed to have some background in the subject matter and the following are stated without proof.

T1. A tree is a directed acyclic graph (DAG).

T2. Trees are collections of nodes and edges.

T3. If (A, B) is the directed edge from node A to node B, A is called the parent and B is called the child.

T4. A node has at most one parent, drawn above the node and zero or more children, drawn below.

T5. There is one node that has no parent. This node is called the root and appears at the top of the tree. Nodes that have no children are called leaves. Nodes that are not leaves are called interior nodes.

T6. A simple directed path from v_0 to v_n is represented by $(v_0, v_1, v_2, \cdots v_n)$ where (v_i, v_{i+1}) with $i \in \{0,1,2, \cdots n-1\}$ are directed edges joining the nodes, $v_0, v_1, v_2, \cdots v_n$ of the tree and $v_i \neq v_j$ for $i \neq j$. The length of the simple directed path is equal to the number of directed edges connecting the nodes $v_0, v_1, v_2, \cdots v_n$ and is equal to n in this case.

T7. For any two nodes A and B, if there is a simple directed path from A to B, B is a descendant of A and A is the ancestor of B. Since every simple directed path from A to B must pass through a child of A, there is simple directed path from one of $A's$ children to B.

T8. There is a unique simple directed path from the root to any other node.

T9. Let $d(r, l) =$ the length of the path from the root r to a leaf l. The height of the tree is defined as $h = Max\{d(r, l) | r = root; l = a\ leaf\}$. Therefore, the height of a tree is the longest path from the root to a leaf.

T10. The length of the path from the root to a node v is called the level of v.

T11. The simple directed path from an interior node to a leaf is called a branch. The combination of all branches is the largest subtree with the interior node as the root. The length of any branch is no longer than the height of the subtree which in turn is no longer than the height of the parent tree.

T12. The children of a node are ordered from left to right. If node A is to the left of node B, then all the descendants of A are to be to the left of all the descendants of B at the same level.

T13. A subtree is a tree of which the vertices and edges are also the vertices and edges of the parent tree. If a subtree has a leaf, the leaf is also a leaf of the parent tree.

Definition 2.31. For any context-free grammar, $G = (V, \Sigma, R, S)$, a parse tree for G is a tree that satisfies the following conditions:

(i) Each interior node is labeled as a variable in V.

(ii) Each leaf is labeled either as a variable in V, a terminal in Σ or ϵ.

(iii) If an interior node labeled A (a variable) has children $X_1, X_2, X_3, \cdots X_n$ where $X_i \in V \cup \Sigma$ for $i \in \{1,2, \cdots n\}$, then

$A \longrightarrow X_1X_2X_3 \cdots X_n$ is a rule in R.

 doi:10.20944/preprints202506.0158.v1

(iv) If an interior node labeled A (a variable) has ϵ as a child, then ϵ is the only child of A and $A \longrightarrow \epsilon$ is a rule in R.

Note that any subtree of a parse tree is also a parse tree.

Definition 2.32. The yield of a parse tree is the concatenation of all the leaves of the tree from left to right.

Theorem 2.33. Let $G = (V, \Sigma, R, S)$ be a CFG. The following statements are equivalent.

(i) $\exists$ a parse tree with root $A \in V$ and a yield $w \in \Sigma^*$.

(ii) $A \overset{*,lm}{\Longrightarrow} w, w \in \Sigma^*$.

(iii) $A \overset{*}{\Rightarrow} w,\ w \in \Sigma^*$.

Proof. "(i) $\Longrightarrow$(ii)"

This can be proved by an induction on the height of the tree in statement (i).

Let $h\ (\geq 1)$ be the height of the parse tree in statement (i).

"$h = 1$"

The parse tree looks like the following figure.

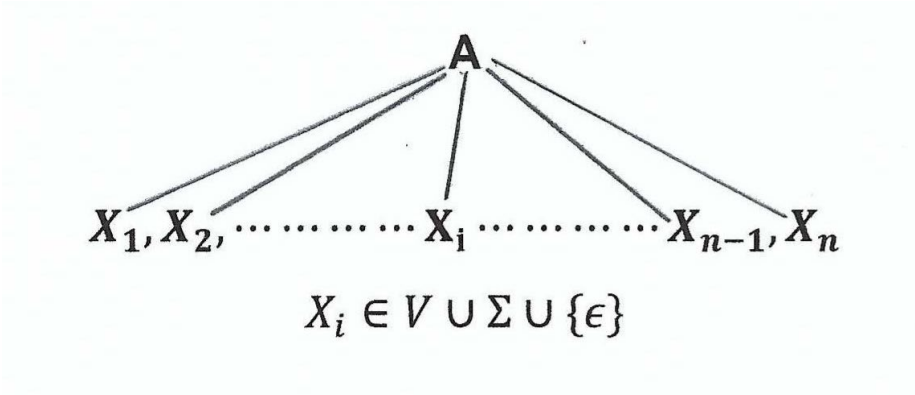


Figure 2.1. Caption.

By definition of parse tree, $A \longrightarrow X_1X_2X_3 \cdots X_n$ is a rule in R.

By Proposition 2.8(i), $A \Longrightarrow X_1X_2X_3 \cdots X_n$.

Therefore, $A \overset{*}{\Rightarrow} X_1X_2X_3 \cdots X_n$.

The yield of this tree is $X_1X_2X_3 \cdots X_n$ which is equal to w by statement (i).

Therefore, $A \overset{*}{\Rightarrow} w$.

Since A is the only variable in the string A, it is therefore also the leftmost variable in the string A.

Therefore, $A \overset{*,lm}{\Longrightarrow} w$.

Hence, the statement "(i) $\Longrightarrow$(ii)" is true for $h = 1$.

"Induction"

Let k be an integer such that $k \geq 1$.

Induction Hypothesis:

The statement "(i) $\Longrightarrow$(ii)" is true for any parse tree with height h if $h \leq k$.

Consider now a parse tree $Pt(A, w, k+1)$ that has root A, yield w and a height of $k+1$.

This parse tree looks like the following figure.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 June 2025 doi:10.20944/preprints202506.0158.v1

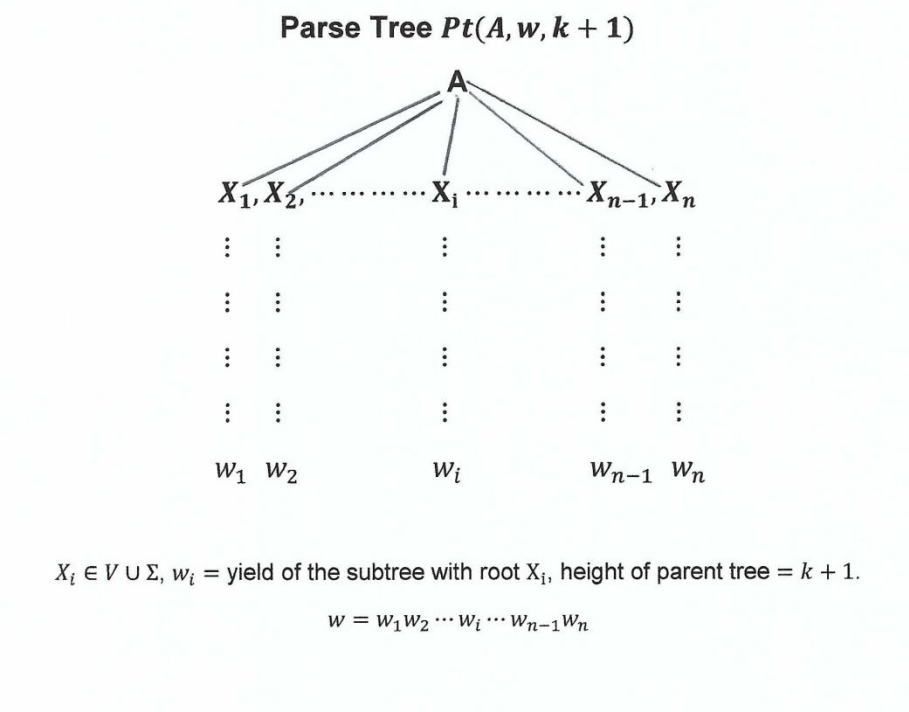


Figure 2.2. Caption.

$\forall i \in \{1,2,\cdots n\},\ X_i \in V \cup \Sigma$.

There are 2 cases to examine.

(a) $X_i \in \Sigma$

$X_i = w_i$ for some $w_i \in \Sigma$.

$X_i \overset{0}{\Rightarrow} w_i$.

$X_i \overset{*}{\Rightarrow} w_i$.

$X_i \overset{*,lm}{\Longrightarrow} w_i$ (X_i is the only variable in the head)

Furthermore, since $X_i \in \Sigma$, $X_i = w_i$ is a leaf.

Therefore, $w_i \sqsubset w$.

(b) $X_i \in V$

By T11 and T13, the combination of all branches of X_i forms a subtree of $Pt(A, w, k+1)$ and every leaf of the subtree is also a leaf of the parent tree.

Let w_i be the yield of X_i.

By definition of yield, every symbol in w_i is a leaf and therefore a symbol in w.

Therefore, $w_i \sqsubset w$.

Since $w \in \Sigma^*$, $w_i \in \Sigma^*$.

Claim: $w = w_1 w_2 \cdots w_n$.

By T12, w_i is to the left of w_j for $i < j$ since X_i is to the left of X_j.

Therefore, $w = x_0 w_1 x_1 w_2 \cdots w_n x_n$ where $x_0, x_1, \cdots x_n \in \Sigma^*$.

Let l be a symbol in w.

l is a leaf in $Pt(A, w, k+1)$ because w is the yield.

By T8, there is a simple directed path from A to l.

By T7, there is a simple directed path from X_i to l for some $i \in \{1,2,\cdots n\}$.

Since l has no children, l must be a leaf descendant of X_i.

Therefore, l is a symbol in w_i because w_i is the yield of the subtree with root X_i.

Therefore, l is a symbol in $w \Longrightarrow l$ is a symbol in w_i for some $i \in \{1,2,\cdots n\}$.

Therefore, $|w| \leq |w_1 w_2 \cdots w_n|$.

Therefore, $|x_0 w_1 x_1 w_2 \cdots w_n x_n| \leq |w_1 w_2 \cdots w_n|$.

 doi:10.20944/preprints202506.0158.v1

This means that $x_0 = x_1 = \cdots = x_n = \epsilon$.
Therefore, $w = w_1 w_2 \cdots w_n$.
Now, back to the subtree with root X_i and yield w_i.
The height of this subtree = the length of the longest branch in the subtree
= the length of a simple directed path in the parent tree
from X_i to a leaf l
= the length of a simple directed path in the parent tree
from A to a leaf l minus 1 (By T7 & X_i is a child of A)
$\leq$ the height of the parent tree minus 1
$= k + 1 - 1$
$= k$
By induction hypothesis, $X_i \overset{*,lm}{\Longrightarrow} w_i$.
Combining (a) & (b), we now have $X_i \overset{*,lm}{\Longrightarrow} w_i$ for all $i \in \{1,2,\cdots n\}$ and $w = w_1 w_2 \cdots w_n$.
For the parent tree $Pt(A, w, k+1)$,
$A \Longrightarrow X_1 X_2 X_3 \cdots X_n$ (Proposition 2.8(i))
$A \overset{lm}{\Rightarrow} X_1 X_2 X_3 \cdots X_n$ (A is the only variable in the head)
Since $X_i \overset{*,lm}{\Longrightarrow} w_i$ and by Proposition 2.17,
$X_1 X_2 X_3 \cdots X_n \overset{*,lm}{\Longrightarrow} w_1 w_2 \cdots w_n$.
Therefore, $A \overset{lm}{\Rightarrow} X_1 X_2 X_3 \cdots X_n \overset{*,lm}{\Longrightarrow} w_1 w_2 \cdots w_n$.
$A \overset{*,lm}{\Longrightarrow} w_1 w_2 \cdots w_n$.
Since $w = w_1 w_2 \cdots w_n$, $A \overset{*,lm}{\Longrightarrow} w$.
The statement "(i) $\Longrightarrow$(ii)" is true for $h = k + 1$.
This completes the proof of "(i) $\Longrightarrow$(ii)".
"(ii) $\Longrightarrow$(iii)"
The proof of this statement is trivial because every leftmost derivation is a derivation.
"(iii) $\Longrightarrow$(i)"
Since $A \overset{*}{\Rightarrow} w$, $\exists n \geq 1$ such that $A \overset{n}{\Rightarrow} w$. (Note that $n \neq 0$ because $A \in V$ and $w \in \Sigma^*$.)
The proof of this statement, "(iii) $\Longrightarrow$(i)", is by induction on n.
($n = 1$)
$\exists w_1, w_2 \cdots w_m \in \Sigma$ such that $w = w_1 w_2 \cdots w_m$ & $A \Longrightarrow w_1 w_2 \cdots w_m$.
By Proposition 2.8(i), $A \longrightarrow w_1 w_2 \cdots w_m$.
The following is a parse tree with root A and yield w.

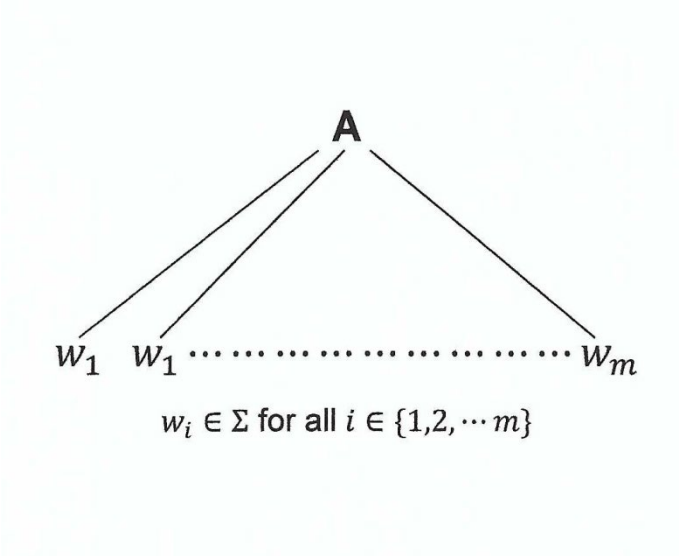


Figure 2.3. Caption.

Therefore, the statement is true for $n = 1$.

 doi:10.20944/preprints202506.0158.v1

(Induction)

Induction Hypothesis:

Let k be an integer such that $k \geq 1$.

For any $n \leq k$, if $A \overset{n}{\Rightarrow} w$, then $\exists$ a parse tree with root A and yield w.

Now, consider $n = k + 1$.

If $A \overset{k+1}{\Longrightarrow} w$,

$\exists u_1, u_2, \cdots u_k \in (V \cup \Sigma)^*$ such that

$A \Longrightarrow u_1 \Longrightarrow u_2 \cdots \cdots \Longrightarrow u_k \Longrightarrow w$.

$\exists X_1, X_2, \cdots X_m \in V \cup \Sigma$ such that $u_1 = X_1 X_2 \cdots X_m$.

Therefore, $X_1 X_2 \cdots X_m \Longrightarrow u_2 \cdots \cdots \Longrightarrow u_k \Longrightarrow w$.

By Proposition 2.28(ii),

$X_i \overset{n_i}{\Rightarrow} w_i$ with $n_i \leq k$ and $w_1 w_2 \cdots w_m = w$.

By induction hypothesis, $\exists$ a parse tree with root X_i and yield w_i which looks like the following figure.

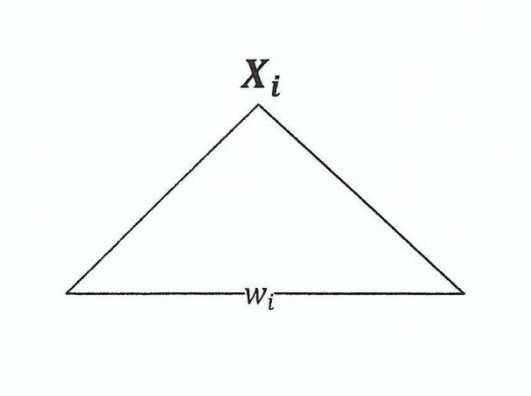


Figure 2.4. Caption.

We now can construct a parse tree, $Pt(A, w, k+1)$ as follows.

(1) Start with a one level parse tree that has root A and yield $X_1 X_2 \cdots X_m$ that looks like the following figure.

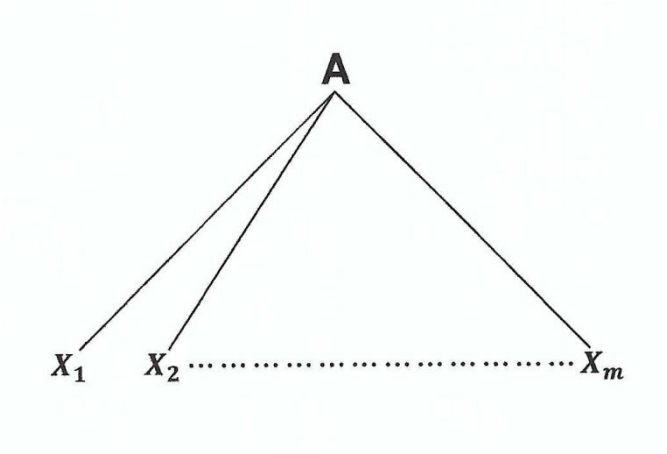


Figure 2.5. Caption.

(2) For each $i \in \{1,2, \cdots m\}$, if $X_i \in \Sigma$, set $X_i = w_i$ for some $w_i \in \Sigma$.

If $X_i \in V$, add the parse tree as shown in Figure 2.4 to the parse tree as shown in Figure 2.5. The resulting tree would look like the following figure.

$Pt(A, w, k+1)$

 doi:10.20944/preprints202506.0158.v1

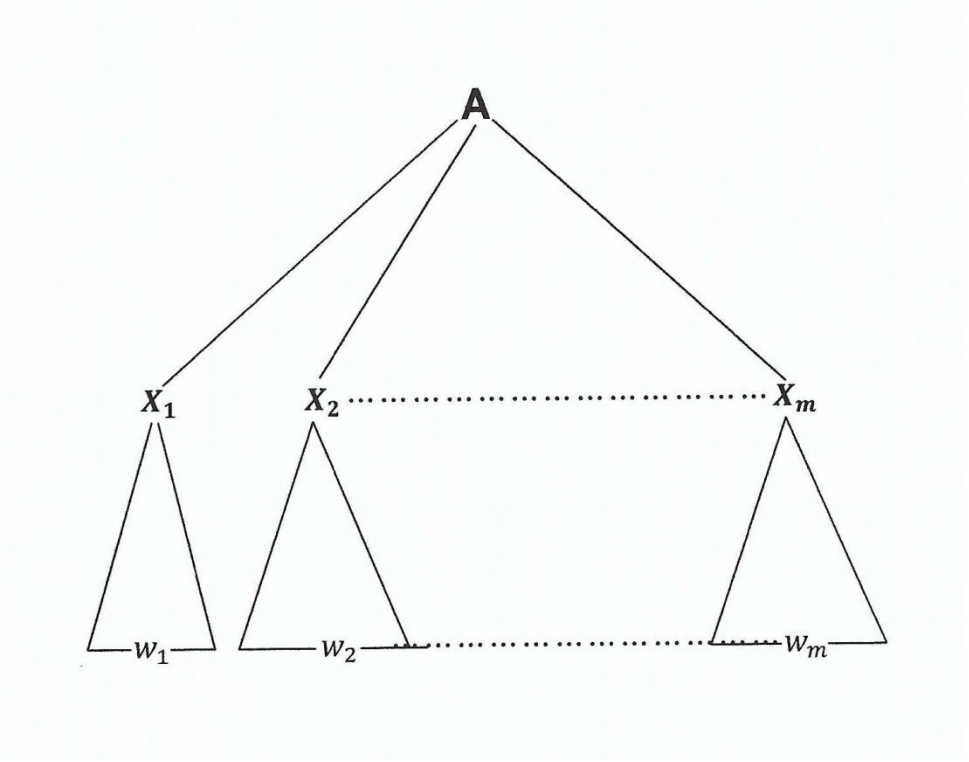


Figure 2.6. Caption.

Clearly, this tree $(Pt(A, w, k+1))$ with root A is a parse tree since the one level tree and all the subtrees with root X_i and yield w_i are parse trees.

In addition, since $w_1 w_2 \cdots w_m = w$, the yield of this parse tree is w.

Therefore, the statement "(iii) $\Rightarrow$(i)" is true for $n = k+1$.

This completes the proof of "(iii) $\Rightarrow$(i)" and also the proof of Theorem 2.33.

2.2. Chomsky Normal Form (*CNF*)

Definition 2.34. Let $G = (V, \Sigma, R, S)$ be a CFG.

G is in Chomsky normal form if very rule of G is of the following form:

$A \longrightarrow BC$ where $A \epsilon V$ and $B, C \epsilon V \backslash \{S\}$

$A \longrightarrow a$ where $a \epsilon \Sigma$

$S \longrightarrow \epsilon$ where $S =$ Start Variable

Lemma 2.35. For every $CFG\ G = (V, \Sigma, R, S)$, there is a $CFG\ G'$ with no ϵ-rule ($A \longrightarrow \epsilon$ where $A \neq S$) or unit rule ($A \longrightarrow B$ where $A, B \in V$) such that $L(G) = L(G')$.

Proof. We can inductively construct a new set of rules, R' using the following procedure:

(i) Copy all the rules in R to R'.

(ii) If $B \neq S, A \longrightarrow \alpha B \beta$ and $B \longrightarrow \epsilon$ are in R', create $A \longrightarrow \alpha\beta$ in R'.

(iii) If $A \longrightarrow B$ and $B \longrightarrow \gamma$ are in R', create $A \longrightarrow \gamma$ in R'.

We can further assume that R' is the smallest one of all the sets that can be thus created because we can always rename the smallest one to R' knowing that the minimum exists.

Let $G' = (V, \Sigma, R', S)$.

It's clear from construction that $R \subset R'$.

Therefore, every derivation in G is a derivation in G' and hence $L(G) \subset L(G')$.

 doi:10.20944/preprints202506.0158.v1

On the other hand, every new rule that is created in G' is equivalent to the two rules that it is created from by Proposition 2.15 and therefore, every derivation in G' can be simulated by either the same rules or equivalent rules in G.

Hence, $L(G') \subset L(G)$.

It remains to show that all the ϵ and unit rules in G' are redundant for the production of any $x \in L(G')$.

Since $L(G') = \{x \in \Sigma^* \mid S \overset{*,G'}{\Longrightarrow} x\}$, knowing that minimum derivations exist, we can assume every derivation of $x \in L(G')$ is the one of minimum length.

Claim 1. Any derivation $S \overset{*,G'}{\Longrightarrow} x$ does not use an ϵ-rule.

Proof of Claim 1. Assume for contradiction that $B \longrightarrow \epsilon$ where $B \neq S$ is used at some point of the derivation.

$S \overset{*,G'}{\Longrightarrow} x$ can be rewritten as

$S \overset{*,G'}{\Longrightarrow} \gamma B \delta \overset{1,G'}{\Longrightarrow} \gamma\delta \overset{*,G'}{\Longrightarrow} x$ where $\gamma, \delta \in (V \cup \Sigma)^*$.

This B must have been generated at an earlier point of the derivation in the form of

$\eta A \theta \overset{1,G'}{\Longrightarrow} \eta\alpha B\beta\theta$ where $\eta, \alpha, \beta, \theta \in (V \cup \Sigma)^*$.

Therefore, $S \overset{*,G'}{\Longrightarrow} x$ can be further rewritten as

$S \overset{m,G'}{\Longrightarrow} \eta A \theta \overset{1,G'}{\Longrightarrow} \eta\alpha B\beta\theta \overset{n,G'}{\Longrightarrow} \gamma B\delta \overset{1,G'}{\Longrightarrow} \gamma\delta \overset{k,G'}{\Longrightarrow} x$ where $k, m, n \geq 0$.

(Note that $\eta\alpha B\beta\theta \overset{n,G'}{\Longrightarrow} \gamma B\delta$ is a derivation in which the rule in each step does not originate from this particular B.)

Since $A \longrightarrow \alpha B\beta$ and $B \longrightarrow \epsilon$ are in R', by construction (ii), $A \longrightarrow \alpha\beta$ is in R'.

Therefore, $\eta A \theta \overset{1,G'}{\Longrightarrow} \eta\alpha\beta\theta$ is a valid production in G'.

Furthermore, since $\eta\alpha B\beta\theta \overset{n,G'}{\Longrightarrow} \gamma B\delta$, by Proposition 2.19, we can substitute ϵ for B to obtain the following valid production in G':

$\eta\alpha\beta\theta \overset{n,G'}{\Longrightarrow} \gamma\delta$.

If we apply these two new productions at the corresponding points of the original derivation of x, we have the following valid derivation:

$S \overset{m,G'}{\Longrightarrow} \eta A \theta \overset{1,G'}{\Longrightarrow} \eta\alpha\beta\theta \overset{n,G'}{\Longrightarrow} \gamma\delta \overset{k,G'}{\Longrightarrow} x$.

We note that this new derivation of x has a length of $k + m + n + 1$ which is shorter than the original one of $k + m + n + 2$.

This contradicts the assumption that the original derivation is of minimum length.

Claim 2. Any derivation $S \overset{*,G'}{\Longrightarrow} x$ does not use a unit rule.

Proof of Claim 2. Assume for contradiction that a unit rule $A \longrightarrow B$ is used at some point of the derivation $S \overset{*,G'}{\Longrightarrow} x$.

We can rewrite this derivation as

$S \overset{*,G'}{\Longrightarrow} \alpha A\beta \overset{1,G'}{\Longrightarrow} \alpha B\beta \overset{*,G'}{\Longrightarrow} x$.

This B must be eventually gotten rid of before reaching the final product of $x \in \Sigma^*$ and the production that we need for getting rid of B is:

$\eta B\theta \overset{1,G'}{\Longrightarrow} \eta\gamma\theta$ where $B \longrightarrow \gamma$ is a rule in G'.

We can now rewrite $S \overset{*,G'}{\Longrightarrow} x$ as

$S \overset{m,G'}{\Longrightarrow} \alpha A\beta \overset{1,G'}{\Longrightarrow} \alpha B\beta \overset{n,G'}{\Longrightarrow} \eta B\theta \overset{1,G'}{\Longrightarrow} \eta\gamma\theta \overset{k,G'}{\Longrightarrow} x$.

Since $A \longrightarrow B$ and $B \longrightarrow \gamma$ are rules in R', $A \longrightarrow \gamma$ is a rule in R' by construction (iii).

$\alpha A\beta \overset{1,G'}{\Longrightarrow} \alpha\gamma\beta$ is a valid production in G'.

 doi:10.20944/preprints202506.0158.v1

Furthermore, since $\alpha B\beta \overset{n,G'}{\Longrightarrow} \eta B\theta$, by Proposition 2.19, we can substitute γ for B to obtain the following valid production:

$\alpha\gamma\beta \overset{n,G'}{\Longrightarrow} \eta\gamma\theta$.

By applying these two new productions at the corresponding points of the derivation of x, we have the following derivation:

$S \overset{m,G'}{\Longrightarrow} \alpha A\beta \overset{1,G'}{\Longrightarrow} \alpha\gamma\beta \overset{n,G'}{\Longrightarrow} \eta\gamma\theta \overset{k,G'}{\Longrightarrow} x$.

This new derivation has a length of $k+m+n+1$ which is shorter than the original one of $k+m+n+2$.

This contradicts the assumption that the original given derivation of x is of minimum length.

Combining Claim 1 and Claim 2, we can conclude Lemma 2.35.

We now examine a method for converting a CFG into one in Chomsky Normal form.

Definition 2.36 (The Method (M)). From every CFG, $G=(V,\Sigma,R,S)$, that doesn't have ϵ-rules or unit rules, we can construct a CFG, $G'=(V',\Sigma,R',S)$ using a method called Method (M) as described in the following steps:

Step 1

For every $a\in\Sigma$, create a variable A_a and a rule $A_a\longrightarrow a$. Note that A_a is a newly and uniquely created variable such that $A_a\notin V$ and $A_a\neq A_b$ for any $a,b\in\Sigma$ such that $a\neq b$.

Step 2

$\forall r\in R$, r can be expressed as $A\longrightarrow u_1u_2\cdots u_k$ where $A\in V$, $u_1,u_2,\cdots u_k\in V\cup\Sigma$ & $k\geq 0$. Create a set of rules (called $P(r)$) and a set of nodes (called $V(r)$) according to the following steps:

(i) For $k=0$

r becomes $A\longrightarrow\epsilon$.

Since R doesn't have any ϵ-rule, except $S\longrightarrow\epsilon$, A must be equal to S and r becomes $S\longrightarrow\epsilon$.

Copy $S\longrightarrow\epsilon$ into $P(r)$.

In this case, $P(r)=\{S\longrightarrow\epsilon\}=\{r\}$ and $V(r)=\emptyset$.

(ii) For $k=1$

r becomes $A\longrightarrow u_1$.

Since R doesn't have any unit rule, $u_1\in\Sigma$.

Copy r into $P(r)$.

In this case, $P(r)=\{A\longrightarrow u_1\}=\{r\}$ and $V(r)=\emptyset$.

(iii) For $k=2$

r becomes $A\longrightarrow u_1u_2$.

If $u_1,u_2\in V$, copy r into $P(r)$. In this case, $P(r)=\{A\longrightarrow u_1u_2\}=\{r\}$ and $V(r)=\emptyset$.

If $u_1\in\Sigma$ & $u_2\in V$, create $A\longrightarrow U_{u_1}u_2$ and add this rule and $U_{u_1}\longrightarrow u_1$ to $P(r)$. Add U_{u_1} to $V(r)$.

(Note that $U_{u_1}\longrightarrow u_1$ was created in Step 1 above).

In this case, $P(r)=\{A\longrightarrow U_{u_1}u_2, U_{u_1}\longrightarrow u_1\}$ and $V(r)=\{U_{u_1}\}$.

If $u_1\in V$ & $u_2\in\Sigma$, create $A\longrightarrow u_1U_{u_2}$ and add this rule and $U_{u_2}\longrightarrow u_2$ to $P(r)$. Add U_{u_2} to $V(r)$.

(Note that $U_{u_2}\longrightarrow u_2$ was created in Step 1 above).

In this case, $P(r)=\{A\longrightarrow u_1U_{u_2}, U_{u_2}\longrightarrow u_2\}$ and $V(r)=\{U_{u_2}\}$.

 doi:10.20944/preprints202506.0158.v1

If both $u_1, u_2 \in \Sigma$, create $A \longrightarrow U_{u_1}U_{u_2}$ and add it along with $U_{u_1} \longrightarrow u_1$, $U_{u_2} \longrightarrow u_2$ to $P(r)$. Add U_{u_1}, U_{u_2} to $V(r)$.

(Note that $U_{u_1} \longrightarrow u_1$ and $U_{u_2} \longrightarrow u_2$ were created in Step 1 above).

In this case, $P(r) = \{A \longrightarrow U_{u_1}U_{u_2}, U_{u_1} \longrightarrow u_1, U_{u_2} \longrightarrow u_2\}$ and $V(r) = \{U_{u_1}, U_{u_2}\}$.

(iv) For $k \geq 3$

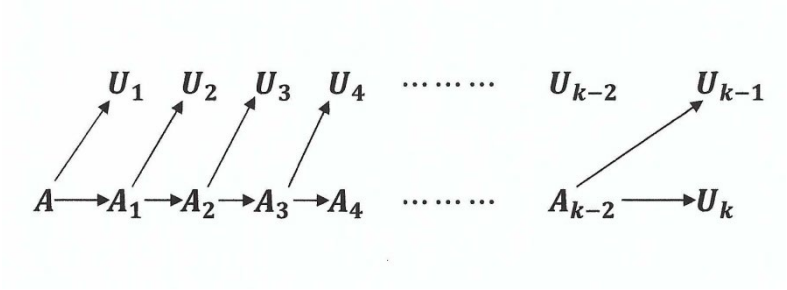


Figure 2.7. Caption.

As depicted by the above figure, create the following rules and add them to $P(r)$.

$A \longrightarrow U_1A_1$
$A_1 \longrightarrow U_2A_2$
$A_2 \longrightarrow U_3A_3$
$\vdots$
$A_i \longrightarrow U_{i+1}A_{i+1}$
$\vdots$
$A_{k-2} \longrightarrow U_{k-1}U_k$

where $A_1, A_2, \cdots A_{k-2}$ are variables newly and uniquely created for each r and therefore, they are not in V.

For any $i \in \{1,2,3, \cdots k\}$, if $u_i \in V$, $U_i = u_i$ and if $u_i \in \Sigma$, set $U_i = U_{u_i}$ and add $U_{u_i} \longrightarrow u_i$ to $P(r)$. Add U_{u_i} to $V(r)$.

(Note that $U_{u_i} \longrightarrow u_i$ for each $u_i \in \Sigma$ were created in Step 1 above).

In this case, $P(r)$ includes all the rules:

$A \longrightarrow U_1A_1$
$A_1 \longrightarrow U_2A_2$
$A_2 \longrightarrow U_3A_3$
$\vdots$
$A_i \longrightarrow U_{i+1}A_{i+1}$
$\vdots$
$A_{k-2} \longrightarrow U_{k-1}U_k$

and the rules $U_{u_i} \longrightarrow u_i$ for any $u_i \in \Sigma$ whereas

$V(r) = \{U_{u_i} | u_i \in \Sigma\} \cup \{A_i | i = 1,2, \cdots k-2\}$

Step 3

Set

$$V' = V \cup \bigcup_{r \in R} V(r)$$

And

$$R' = \bigcup_{r \in R} P(r)$$

We note the following properties of the rules created by Method (M):

N1. All the rules in R' are in Chomsky Normal Form.

N2. For any $r' \in R'$, there exists $r \in R$ such that $r' \in P(r)$. Furthermore, $P(r_1) \neq P(r_2)$ for any $r_1, r_2 \in R$ such that $r_1 \neq r_2$.

N3. For any $r \in R$, r is equivalent to the rules in $P(r)$ by Proposition 2.15.

N4. V and $\cup_{r \in R} V(r)$ are disjoint. That is $V \cap \cup_{r \in R} V(r) = \emptyset$.

 doi:10.20944/preprints202506.0158.v1

N5. For any $r' \in P(r)$, either $Head(r') = Head(r)$ or $Head(r') \notin V$. Or equivalently, $Head(r') \in V \Longrightarrow Head(r') = Head(r)$.

N6. $\forall r' \in P(r)$, if $|Body(r')| = 2$, then r' is unique for $P(r)$. That is, $r' \notin P(r_1)$ for any $r_1 \in R$ such that $r \neq r_1$.

N7. If $k = 0$ or $|Body(r)| = 0$, $P(r) = \{S \longrightarrow \epsilon\} = \{r\}$ and $V(r) = \emptyset$.

N8. If $k = 1$ or $|Body(r)| = 1$, $P(r) = \{A \longrightarrow u_1\} = \{r\}$ and $V(r) = \emptyset$ where $u_1 \in \Sigma$.

We now have the following theorem.

Theorem 2.37. Every context-free language is generated by a CFG in Chomsky normal form (CNF).

Proof. Since every context-free language is generated by a CFG, we need to show that every CFG can be converted to an equivalent CFG in Chomsky normal form.

Also, because of Lemma 2.35, we can start with a CFG that has no ϵ-rule ($A \longrightarrow \epsilon$ where $A \neq S$) or unit rule ($A \longrightarrow B$ where $A, B \in V$).

Let $G = (V, \Sigma, R, S)$ be the CFG that has no ϵ-rule or unit rule except $S \longrightarrow \epsilon$.

Let $G' = (V', \Sigma, R', S)$ be a CFG constructed from G by use of Method (M).

In the following, we shall show $L(G) = L(G')$ by showing $x \in L(G) \Leftrightarrow x \in L(G')$ $\forall x \in \Sigma^*$.

" $\Longrightarrow$ " (If $x \in L(G)$)

$S \overset{*,G}{\Rightarrow} x$.

$\exists r_1, r_2, \cdots r_i, \cdots r_n, r_{n+1} \in R$ and $u_1, u_2, \cdots u_i, \cdots u_n \in (V \cup \Sigma)^*$ such that

$S \overset{r_1,G}{\Longrightarrow} u_1 \overset{r_2,G}{\Longrightarrow} u_2 \cdots \cdots u_{i-1} \overset{r_i,G}{\Longrightarrow} u_i \cdots \cdots \overset{r_n,G}{\Longrightarrow} u_n \overset{r_{n+1},G}{\Longrightarrow} x$.

By N3, for any $i \in \{1,2,\cdots, n+1\}$, r_i is equivalent to a sequence of rules from $P(r_i)$ which is a subset of R'.

Therefore, $S \overset{P(r_1),G'}{\Longrightarrow} u_1 \overset{P(r_2),G'}{\Longrightarrow} u_2 \cdots \cdots u_{i-1} \overset{P(r_i),G'}{\Longrightarrow} u_i \cdots \cdots \overset{P(r_n),G'}{\Longrightarrow} u_n \overset{P(r_{n+1}),G'}{\Longrightarrow} x$.

Note that $u_1, u_2, \cdots u_i, \cdots u_n \in (V' \cup \Sigma)^*$ because $V \subset V'$.

Therefore, $S \overset{*,G'}{\Longrightarrow} x$.

Therefore, $x \in L(G')$.

"$\Longleftarrow$" (If $x \in L(G')$)

$S \overset{*,G'}{\Longrightarrow} x$.

By Theorem 2.33, $\exists$ a parse tree (in G') with root $S \in V'$ and a yield $x \in \Sigma^*$.

Let's call this parse tree (T').

By definition of parse tree, S and its children must be the head and body of a rule in R'.

Let's call this rule r' and hence $Head(r') = S$.

By N1, r' must be in one of the following forms:

- $S \longrightarrow \epsilon$
- $A \longrightarrow a$ where $a \in \Sigma$, $A \in V'$
- $A \longrightarrow U_1 U_2$ where $A \in V'$, $U_1, U_2 \in V' \backslash \{S\}$

If r' is $S \longrightarrow \epsilon$, ϵ is the only child of S.

Since ϵ has no children and x is a descendant of S, this is possible only if $\epsilon = x$.

Furthermore, by construction of (M), $S \longrightarrow \epsilon$ in R' is created from $S \longrightarrow \epsilon$ in R.

Therefore, $S \longrightarrow \epsilon$ is also a rule in R.

Therefore, $S \overset{1,G}{\Rightarrow} \epsilon$. (Proposition 2.8(i))

Therefore, $S \overset{1,G}{\Rightarrow} x$.

Therefore, $S \overset{*,G}{\Rightarrow} x$.

Therefore, $x \in L(G)$

If r' is $A \longrightarrow a$, since $S = Head(r')$, $S = A$.

Therefore, r' is $S \longrightarrow a$ and S has only one child which is a.

Since x is a descendant of S and a has no children, $a = x$.

 doi:10.20944/preprints202506.0158.v1

By construction of (M), $A \longrightarrow a$ in R' is created from $A \longrightarrow a$ in R.

Therefore, $A \longrightarrow a$ is also a rule in R.

Therefore, $S \longrightarrow x$ is a rule in R.

Therefore, $S \overset{1,G}{\Rightarrow} x$. (Proposition 2.8(i))

Therefore, $S \overset{*,G}{\Rightarrow} x$.

Therefore, $x \in L(G)$.

If r' is $A \longrightarrow U_1U_2$ where $A \in V'$, $U_1, U_2 \in V'\backslash\{S\}$

Since $Head(r') = S$ and $S \in V$, $Head(r') \in V$.

Since $Head(r') = A$, $A = S$.

Therefore, r' becomes $S \longrightarrow U_1U_2$ where $U_1, U_2 \in V'\backslash\{S\}$.

By N2, $\exists r \in R$ such that $r' \in P(r)$.

Let r be $A' \longrightarrow u_1u_2 \cdots u_k$ where $A' \in V$, $u_1, u_2, \cdots u_k \in V \cup \Sigma$.

By N5, $Head(r') \in V \Longrightarrow Head(r') = Head(r)$.

Therefore, $S = A'$.

Therefore, r becomes $S \longrightarrow u_1u_2 \cdots u_k$.

We now analyze the different situations for different values of k.

If $k = 0$, r becomes $S \longrightarrow \epsilon$.

By construction of (M), $P(r) = \{S \longrightarrow \epsilon\}$.

Since $r' \in P(r)$, r' is $S \longrightarrow \epsilon$.

This contradicts the underlying assumption that r' is $A \longrightarrow U_1U_2$ where $A \in V'$, $U_1, U_2 \in V'\backslash\{S\}$.

Therefore, k cannot be 0.

If $k = 1$, r becomes $S \longrightarrow u_1$.

Since R doesn't have any unit rule, $u_1 \in \Sigma$.

By construction of (M), $P(r) = \{S \longrightarrow u_1\}$.

Therefore, r' is $S \longrightarrow u_1$ where $u_1 \in \Sigma$.

This contradicts the underlying assumption that r' is $A \longrightarrow U_1U_2$ where $A \in V'$, $U_1, U_2 \in V'\backslash\{S\}$.

Therefore, k cannot be 1.

Therefore, we can exclude the cases of $k \in \{0,1\}$ under the assumption that r' is $A \longrightarrow U_1U_2$ where $A \in V'$, $U_1, U_2 \in V'\backslash\{S\}$.

If $k = 2$, r becomes $S \longrightarrow u_1u_2$.

By construction of (M), $P(r)$ is one of the following:

(i) $P(r) = \{S \longrightarrow u_1u_2\}$ if $u_1, u_2 \in V$

(ii) $P(r) = \{S \longrightarrow U_{u_1}u_2, U_{u_1} \longrightarrow u_1\}$ if $u_1 \in \Sigma$ & $u_2 \in V$

(iii) $P(r) = \{S \longrightarrow u_1U_{u_2}, U_{u_2} \longrightarrow u_2\}$ if $u_1 \in V$ & $u_2 \in \Sigma$

(iv) $P(r) = \{S \longrightarrow U_{u_1}U_{u_2}, U_{u_1} \longrightarrow u_1, U_{u_2} \longrightarrow u_2\}$ if $u_1, u_2 \in \Sigma$

For (i), r' is $S \longrightarrow u_1u_2$.

In this case, r and r' are the same and the sub parse tree in (T') with root S and children u_1, u_2 as shown on the right of the following figure can be replaced by a parse tree in G with the same root and children as shown on the left.

 doi:10.20944/preprints202506.0158.v1

Figure 2.8. Caption.

For (ii), r' is either $S \longrightarrow U_{u_1}u_2$ or $U_{u_1} \longrightarrow u_1$.

However, since $Head(r') = S$ which is in V and $U_{u_1} \notin V$, r' cannot be $U_{u_1} \longrightarrow u_1$.

r' must be $S \longrightarrow U_{u_1}u_2$.

By N3, $S \longrightarrow u_1u_2$ is equivalent to $S \longrightarrow U_{u_1}u_2$ and $U_{u_1} \longrightarrow u_1$.

We have the following equivalent parse trees with the same root and yield.

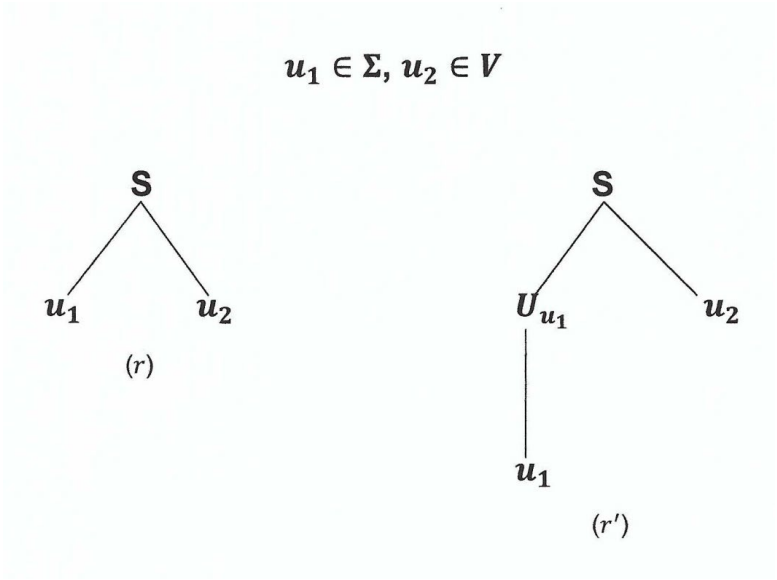


Figure 2.9. Caption.

The one on the left is a parse tree in G whose root and its children are the head and body of a rule in R whereas the one on the right is a sub parse tree of (T').

Therefore we can replace a sub parse tree of (T') with an equivalent parse tree in G whose root and yield are the head and body of a rule in R.

For (iii), by a similar argument, we have the following equivalent parse trees with the same root and yield.

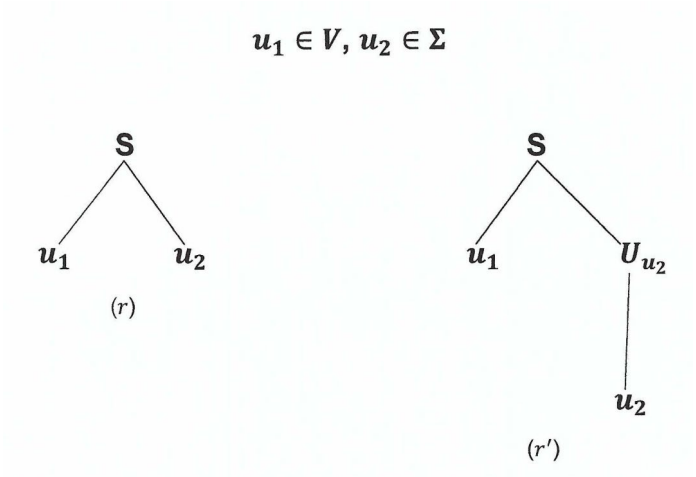


Figure 2.10. Caption.

The one on the left is a parse tree in G whose root and its children are the head and body of a rule in R whereas the one on the right is a sub parse tree of (T').

Therefore we can replace a sub parse tree of (T') with an equivalent parse tree in G whose root and yield are the head and body of a rule in R.

For (iv), by a similar argument, we have the following equivalent parse trees with the same root and yield.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 June 2025 doi:10.20944/preprints202506.0158.v1

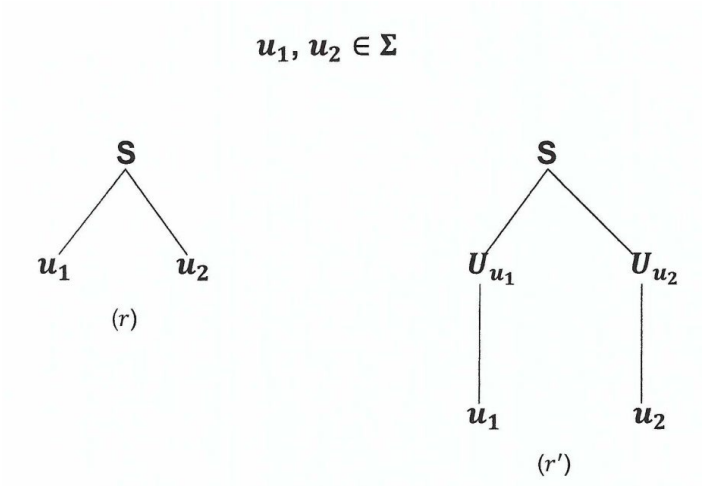


Figure 2.11. Caption.

The one on the left is a parse tree in G whose root and its children are the head and body of a rule in R whereas the one on the right is a sub parse tree of (T').

Therefore we can replace a sub parse tree of (T') with an equivalent parse tree in G whose root and yield are the head and body of a rule in R.

If $k \geq 3$, r is $S \longrightarrow u_1 u_2 \cdots u_k$.

$P(r)$ consists of the following rules:

$S \longrightarrow U_1 A_1$

$A_1 \longrightarrow U_2 A_2$

$A_2 \longrightarrow U_3 A_3$

$\vdots$

$A_i \longrightarrow U_{i+1} A_{i+1}$

$\vdots$

$A_{k-2} \longrightarrow U_{k-1} U_k$

$U_{u_i} \longrightarrow u_i$ if $u_i \in \Sigma \ \forall i \in \{1,2,3,\cdots k\}$

where $U_i = u_i$ if $u_i \in V$ and $U_i = U_{u_i}$ if $u_i \in \Sigma$.

Since $Head(r') = S$, r' is $S \longrightarrow U_1 A_1$.

Since (T') is a parse tree of G', by definition of parse tree,

$U_1 A_1$ are children of S.

$U_2 A_2$ are children of A_1.

$U_3 A_3$ are children of A_2.

$\vdots$

$U_{k-1} U_k$ are children of A_{k-2}.

By N3, r is equivalent to the sequence of rules contained in $P(r)$.

Therefore, we have the following equivalent parse trees with the same root and yield.

 doi:10.20944/preprints202506.0158.v1

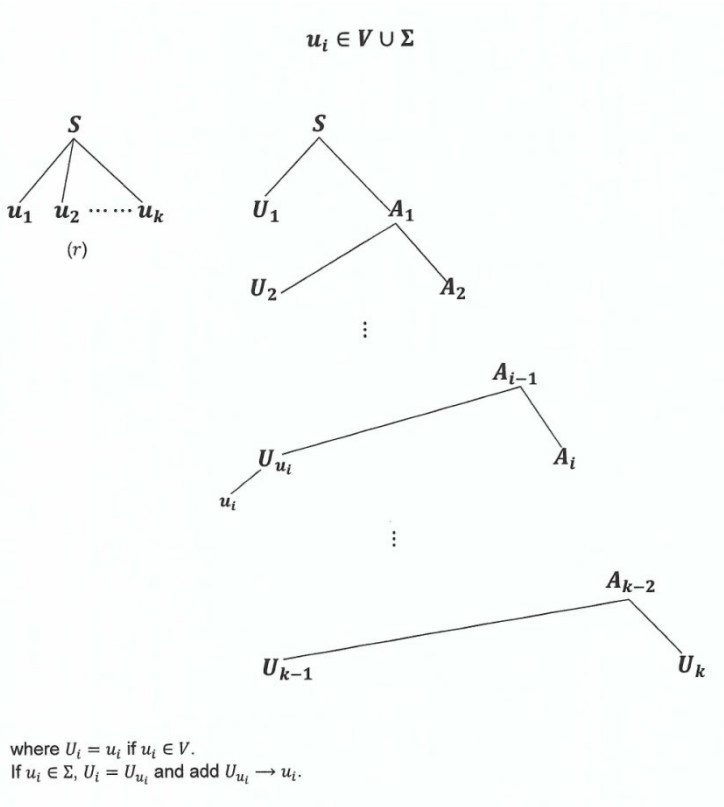


Figure 2.12. Caption.

The one on the left is a parse tree in G whose root and its children are the head and body of a rule in R whereas the one on the right is a sub parse tree of (T').

Therefore we can replace a sub parse tree of (T') with an equivalent parse tree in G whose root and yield are the head and body of a rule in R.

Combining all cases, we conclude that there is a sub parse tree in (T') with root S that can be replaced by an equivalent parse tree in G whose root and yield are the head and body of a rule in R.

We can write this rule in R as $S \longrightarrow u_1 u_2 \cdots u_k$ where $k \geq 0$ and $u_i \in V \cup \Sigma$ for $i \in \{1,2,\cdots k\}$

(a) If all u_i's are terminals

In this case, $u_1 u_2 \cdots u_k = x$, the yield of the parent tree (T').

The reason is that a leaf of a subtree is also a leaf of the parent tree.

Therefore, $u_i \sqsubset x \; \forall i \in \{1,2,\cdots k\}$.

On the other hand, if l is a leaf in x, there is a simple directed path from S to l. This simple directed path must pass through one of the nodes $u_1, u_2 \cdots u_k$ because $u_1 u_2 \cdots u_k$ is the yield of a sub parse tree in (T') which is obtained by branching out from S in all possible directions.

Therefore, l must be one of the nodes $u_1, u_2 \cdots u_k$.

After replacement, we now have a new tree which is a parse tree in G, and furthermore, the root and yield of this tree are respectively S and x.

By Theorem 2.33, $S \overset{*,G}{\Rightarrow} x$.

Therefore, $x \in L(G)$.

(b) If some u_i's are variables

For each u_i that is a variable, we can repeat the above replacement process to replace the sub parse tree (with root u_i) in (T') with a parse tree in G whose root (u_i) and the root's children are the head and body of a rule in R.

Since every time we do a replacement, we get down to a lower level of (T') and since the height of (T') and the number of subtrees of (T') are both finite, this process of replacement must come to a stop after a finite number of operations. When this happens, we have a new tree in which every

 doi:10.20944/preprints202506.0158.v1

interior node and its children are the head and body of a rule in R. This means that the new tree thus created is a parse tree in G.

Furthermore, this replacement process only affects the nodes which are variables. Therefore, the yield of (T'), namely x, is untouched and remains at the bottom after the replacement is complete.

This means that x is also the yield of the newly created tree.

We now have a new tree with root S and yield x and the tree is also a parse tree in G.

By Theorem 2.33, $S \overset{*,G}{\Rightarrow} x$.

Therefore, $x \in L(G)$.

Combining (a) and (b), we complete the proof of Theorem 2.37**.**

On the basis of Theorem 2.37 and the results proved in Lemma 2.35, we can now develop a set of operational rules for the conversion of a CFG to one in CNF.

Let $G = (V, \Sigma, R, S)$ be the CFG to be converted.

Let $G' = (V', \Sigma, R', S_0)$ be the CFG to be created in CNF.

$\boldsymbol{CR-1}$.

Create $S_0 \longrightarrow S$ and add it to R'.

(Note that this creation will ensure that the start variable will not occur on the right hand side of a rule.)

$\boldsymbol{CR-2}$. (Elimination of ϵ-rules)

If $\exists$ a rule $B \longrightarrow \epsilon$ in R, do the following:

(i) For every rule in R in the form $A \longrightarrow u_1Bu_2Bu_3Bu_4 \cdots\cdots u_{n-1}Bu_nBu_{n+1}$

(1) For each single occurrence of B, on the RHS, create a rule with that occurrence deleted and add it to R'.

For example, $A \longrightarrow u_1u_2Bu_3Bu_4 \cdots\cdots u_{n-1}Bu_nBu_{n+1}$

$A \longrightarrow u_1Bu_2u_3Bu_4 \cdots\cdots u_{n-1}Bu_nBu_{n+1}$

$\vdots$

$A \longrightarrow u_1Bu_2Bu_3Bu_4 \cdots\cdots u_{n-1}Bu_nu_{n+1}$

(2) For each group occurrence of 2 B's on the RHS, create a rule with that group occurrence deleted and add it to R'.

For example, $A \longrightarrow u_1u_2u_3Bu_4 \cdots\cdots u_{n-1}Bu_nBu_{n+1}$

$A \longrightarrow u_1u_2Bu_3u_4 \cdots\cdots u_{n-1}Bu_nBu_{n+1}$

$\vdots$

$A \longrightarrow u_1Bu_2Bu_3Bu_4 \cdots\cdots u_{n-1}u_nu_{n+1}$.

$\vdots$

$\vdots$

(n) For each group occurrence of n B's on the RHS, create a rule with that group occurrence deleted and add it to R'.

For example, $A \longrightarrow u_1u_2u_3u_4 \cdots\cdots u_{n-1}u_nu_{n+1}$.

(ii) Repeat (i) until all rules of the form of $B \longrightarrow \epsilon$ are eliminated.

$\boldsymbol{CR-3}$. (Elimination of unit rules)

If $\exists$ rules $A \longrightarrow B$ and $B \longrightarrow u$ in R, do the following:

(i) Create $A \longrightarrow u$ and add it to R'.

(ii) Copy $B \longrightarrow u$ to R' .

(iii) Do not copy $A \longrightarrow B$ to R'.

(iv) Repeat (i) and (ii) until all unit rules of the form $A \longrightarrow B$ are eliminated.

 doi:10.20944/preprints202506.0158.v1

$\boldsymbol{CR-4}$. (Conversion of remaining rules)

For every remaining rule A in R, $A \longrightarrow u_1u_2 \cdots \cdots u_k$ where each $u_i \in V \cup \Sigma$ for $i \in \{1,2, \cdots \cdots k\}$. Create in R' the following sequence of rules and add the corresponding created variables to V':

$A \longrightarrow U_1A_1$

$A_1 \longrightarrow U_2A_2$

$A_2 \longrightarrow U_3A_3$

$\vdots$

$A_{k-2} \longrightarrow U_{k-1}A_{k-1}$

$A_{k-1} \longrightarrow U_k$

where $U_i = u_i$ if $u_i \in V$ and if $u_i \in \Sigma$, add $U_i \longrightarrow u_i$.

Example 2.38. Let $G = (V, \Sigma, R, S)$ be the CFG consisting of the following rules:

$S \longrightarrow ASA|aB$

$A \longrightarrow B|S$

$B \longrightarrow b|\epsilon$

Convert G to $G' = (V', \Sigma, R', S_0)$ in CNF.

Step 1. (Applying $\boldsymbol{CR-1}$.)

$S_0 \longrightarrow S$

$S \longrightarrow ASA|aB$

$A \longrightarrow B|S$

$B \longrightarrow b|\epsilon$

Step 2. (Removing $B \longrightarrow \epsilon$ using $\boldsymbol{CR-2}$)

$S_0 \longrightarrow S$

$S \longrightarrow ASA|aB|a$

$A \longrightarrow B|S|\epsilon$

$B \longrightarrow b$

Step 3 (Removing $A \longrightarrow \epsilon$ using $\boldsymbol{CR-2}$)

$S_0 \longrightarrow S$

$S \longrightarrow ASA|aB|a|SA|AS|S$

$A \longrightarrow B|S$

$B \longrightarrow b$

Step 4 (Removing $S \longrightarrow S$ because of redundancy)

$S_0 \longrightarrow S$

$S \longrightarrow ASA|aB|a|SA|AS$

$A \longrightarrow B|S$

$B \longrightarrow b$

Step 5 (Removing $S_0 \longrightarrow S$ using $\boldsymbol{CR-3}$)

$S_0 \longrightarrow ASA|aB|a|SA|AS$

$S \longrightarrow ASA|aB|a|SA|AS$

$A \longrightarrow B|S$

$B \longrightarrow b$

Step 6 (Removing $A \longrightarrow B$ using $\boldsymbol{CR-3}$)

$S_0 \longrightarrow ASA|aB|a|SA|AS$

$S \longrightarrow ASA|aB|a|SA|AS$

$A \longrightarrow b|S$

$B \longrightarrow b$

Step 7 (Removing $A \longrightarrow S$ using $\boldsymbol{CR-3}$)

$S_0 \longrightarrow ASA|aB|a|SA|AS$

$S \longrightarrow ASA|aB|a|SA|AS$

$A \longrightarrow b|ASA|aB|a|SA|AS$

$B \longrightarrow b$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 June 2025 doi:10.20944/preprints202506.0158.v1

Step 8 (Conversion of remaining rules into CNF)

Since $S_0 \longrightarrow ASA \Leftrightarrow \begin{cases} S_0 \longrightarrow AA_1 \\ A_1 \longrightarrow SA \end{cases}$ and $S_0 \longrightarrow aB \Leftrightarrow \begin{cases} S_0 \longrightarrow UB \\ U \longrightarrow a \end{cases}$,

$S \longrightarrow ASA \Leftrightarrow \begin{cases} S \longrightarrow AA_1 \\ A_1 \longrightarrow SA \end{cases}$ and $S \longrightarrow aB \Leftrightarrow \begin{cases} S \longrightarrow UB \\ U \longrightarrow a \end{cases}$, and

$A \longrightarrow ASA \Leftrightarrow \begin{cases} A \longrightarrow AA_1 \\ A_1 \longrightarrow SA \end{cases}$ and $A \longrightarrow aB \Leftrightarrow \begin{cases} A \longrightarrow UB \\ U \longrightarrow a \end{cases}$,

the rules in R' now become

$S_0 \longrightarrow AA_1|UB|a|SA|AS$

$S \longrightarrow AA_1|UB|a|SA|AS$

$A \longrightarrow b|AA_1|UB|a|SA|AS$

$B \longrightarrow b$

$A_1 \longrightarrow SA$

$U \longrightarrow a$

Example 2.39. Convert $S \longrightarrow aSb|\epsilon$ to CNF where $S \in V$ and $a, b \in \Sigma$ and show that there is more than one way of deriving the string a^2b^2 using rules in CNF.

Conversion of rules.

$S \longrightarrow aSb|\epsilon$

$\Leftrightarrow S \longrightarrow aSb|ab$

$\Leftrightarrow \begin{cases} S \longrightarrow ASB|AB \\ A \longrightarrow a \\ B \longrightarrow b \end{cases}$

$\Leftrightarrow \begin{cases} S \longrightarrow AC|AB; C \longrightarrow SB \\ A \longrightarrow a; B \longrightarrow b \end{cases}$

Derivation of a^2b^2

There is more than one way of deriving the string a^2b^2. Below are a few examples.

(i) $S \overset{S\longrightarrow AC}{\Longrightarrow} AC \overset{C\longrightarrow SB}{\Longrightarrow} ASB \overset{A\longrightarrow a}{\Longrightarrow} aSB \overset{B\longrightarrow b}{\Longrightarrow} aSb \overset{S\longrightarrow AB}{\Longrightarrow} aABb \overset{A\longrightarrow a}{\Longrightarrow} aaBb \overset{B\longrightarrow b}{\Longrightarrow} aabb.$

(ii) $S \overset{S\longrightarrow AC}{\Longrightarrow} AC \overset{C\longrightarrow SB}{\Longrightarrow} ASB \overset{S\longrightarrow AB}{\Longrightarrow} AABB \overset{A\longrightarrow a}{\Longrightarrow} aABB \overset{A\longrightarrow a}{\Longrightarrow} aaBB \overset{B\longrightarrow b}{\Longrightarrow} aabB \overset{B\longrightarrow b}{\Longrightarrow} aabb.$

(iii) $S \overset{S\longrightarrow AC}{\Longrightarrow} AC \overset{C\longrightarrow SB}{\Longrightarrow} ASB \overset{S\longrightarrow AB}{\Longrightarrow} AABB \overset{B\longrightarrow b}{\Longrightarrow} AABb \overset{B\longrightarrow b}{\Longrightarrow} AAbb \overset{A\longrightarrow a}{\Longrightarrow} Aabb \overset{A\longrightarrow a}{\Longrightarrow} aabb.$

2.3. Pushdown Automata (PDA)

Pushdown automata is another kind of nondeterministic computation model similar to nondeterministic finite automata except that they have an extra component called **stack**. The purpose of the stack is to provide additional memory beyond what is available in finite automata.

Pushdown automata are equivalent in power to context-free grammars which will be proved later. In addition to reading symbols from the input alphabet Σ, a PDA also reads and writes symbols on the stack. Writing and reading on the stack must be done at the top. Either symbol from input or stack can be ϵ thereby allowing the machine to move without actually reading or writing. Upon reading a symbol from the input alphabet, the PDA decides to make one of the following moves on the stack before entering the next state:

(i) Replace

Replace the symbol at the top of the stack with another symbol. This move is referred to as the "Replace" move.

(ii) Push

Add a symbol to the top of the stack. This move is referred to as the "Push" move.

(iii) Pop

Erase or remove a symbol from the top of the stack. This move is referred to as the "Pop" move.

 doi:10.20944/preprints202506.0158.v1

(iv) Untouched

Do nothing to change the stack. This move is referred to as "Untouched" move.

A PDA is formally defined as follows.

Definition 2.40. A PDA is a 7-tuple, $M = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ where Q, Σ, Γ & F are finite sets such that

(a) Q is the set of states

(b) Σ is the input alphabet

(c) Γ is the stack alphabet

(d) $\delta: Q \times \Sigma_\epsilon \times \Gamma_\epsilon \longrightarrow \wp(Q \times \Gamma_\epsilon)$ is the transition function

(e) $q_0 \in Q$ is the start state

(f) $\perp \in \Gamma$ is the initial stack symbol signaling an empty stack

(g) $F \subset Q$ is the set of accept states.

M computes as follows.

Let $w = w_1 w_2 \cdots w_m$ where $w_i \in \Sigma_\epsilon$ for $1 \leq i \leq m$.

M accepts w iff $\exists r_0, r_1 \cdots r_m \in Q$ and $s_0, s_1 \cdots s_m \in \Gamma^*$ such that the following conditions are satisfied:

(i) $r_0 = q_0$ and $s_0 = \perp$

(ii) $(r_{i+1}, b_i) \in \delta(r_i, w_{i+1}, a_i))$ for $0 \leq i \leq m-1$ where $a_i, b_i \in \Gamma_\epsilon$ and $s_i = a_i t_i$, $s_{i+1} = b_i t_i$ where $t_i \in \Gamma^*$

(iii) $r_m \in F$

When $m = 0, w = \epsilon$ and only conditions (i) and (iii) are valid which then becomes $r_0 = q_0$ and $s_0 = \perp$ and $r_0 \in F$.

Therefore, we define a PDA to accept ϵ whenever the start state is also an accept state and the stack is signaled to be empty.

If we write $r_i \xrightarrow{w_{i+1}, a_i \to b_i, \delta} r_{i+1}$ for $(r_{i+1}, b_i) \in \delta(r_i, w_{i+1}, a_i)$, conditions (i), (ii) and (iii) can be written as follows:

$$q_0 = r_0 \xrightarrow{w_1, a_0 \to b_0, \delta} r_1 \xrightarrow{w_2, a_1 \to b_1, \delta} r_2 \cdots r_i \xrightarrow{w_{i+1}, a_i \to b_i, \delta} r_{i+1} \cdots r_{m-1} \xrightarrow{w_m, a_{m-1} \to b_{m-1}, \delta} r_m, r_m \in F.$$

When there is only one transition function under consideration, the showing of δ in the computation is usually omitted and the following shorthand is used instead:

$$q_0 = r_0 \xrightarrow{w_1, a_0 \to b_0} r_1 \xrightarrow{w_2, a_1 \to b_1} r_2 \cdots r_i \xrightarrow{w_{i+1}, a_i \to b_i} r_{i+1} \cdots r_{m-1} \xrightarrow{w_m, a_{m-1} \to b_{m-1}} r_m, r_m \in F.$$

For simplicity, we sometimes can use the notation $q_0 \xrightarrow{w, *, \delta} r_m$ to represent a computation of w from q_0 to r_m without showing the intermediate states.

We now can use the transition function to describe the four basic moves of the PDA as mentioned above:

(i) **Replace**

$r \xrightarrow{a, b \to c} r'$ signifies a replacement of b by c at the top of the stack upon reading symbol a from input.

(ii) **Push**

$r \xrightarrow{a, \epsilon \to c} r'$ signifies adding the symbol c to the top of the stack upon reading symbol a from input.

(iii) **Pop**

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 June 2025 doi:10.20944/preprints202506.0158.v1

$r \xrightarrow{a,b\to\epsilon} r'$ signifies removing the symbol b from the top of the stack upon reading symbol a from input.

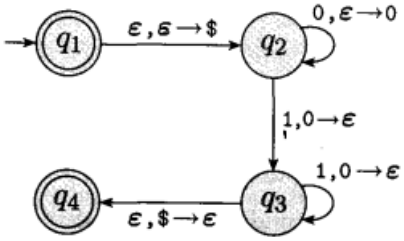
(iv) **Untouched**

$r \xrightarrow{a,\epsilon\to\epsilon} r'$ signifies nothing is done to change the stack upon reading symbol a from input.

We further note that when $a = \epsilon$, $r \xrightarrow{\epsilon,\epsilon\to\epsilon} r'$ signifies a change of state from r to r' with no input read and no change made to the stack.

Example 2.41. Let $M = (Q, \Sigma, \Gamma, \delta, q_1, \perp, F)$ be a PDA where

$Q = \{q_1, q_2, q_3, q_4\}$, $\Sigma = \{0,1\}$, $\Gamma = \{0, \perp, \$\}$, $F = \{q_1, q_4\}$ with the following state diagram:



M recognizes the language $\{0^n1^n | n \geq 0\}$.

If the stack is signaled to be empty at the beginning, M accepts the empty string ($\epsilon = 0^01^0$), because q_1 is both a start and accept state. Furthermore, if the input string is not empty at the start state, the PDA would not read anything from the string except to push $\$$ onto the stack.

M accepts the string 0^31^3 with the following computation:

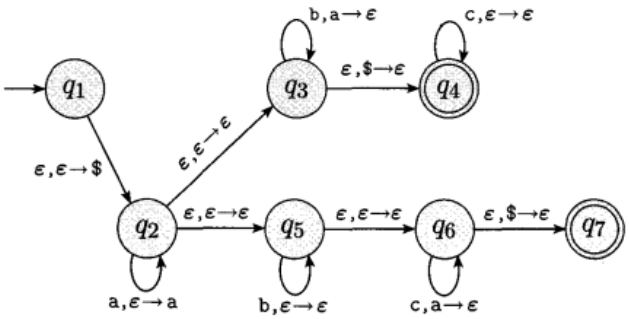
$q_1 \xrightarrow{\epsilon,\epsilon\to\$} q_2 \xrightarrow{0,\epsilon\to0} q_2 \xrightarrow{0,\epsilon\to0} q_2 \xrightarrow{0,\epsilon\to0} q_2 \xrightarrow{1,0\to\epsilon} q_3 \xrightarrow{1,0\to\epsilon} q_3 \xrightarrow{1,0\to\epsilon} q_3 \xrightarrow{\epsilon,\$\to\epsilon} q_4$, $q_4 \in F$.

Note that the above illustration is not a proof that M recognizes the language $\{0^n1^n | n \geq 0\}$. To make such a proof, one must argue that every string of the form 0^n1^n is accepted by M and every string accepted by M is of the form 0^n1^n.

Note also that the steps $q_1 \xrightarrow{\epsilon,\epsilon\to\$} q_2$ and $q_3 \xrightarrow{\epsilon,\$\to\epsilon} q_4$ can be replaced by $q_1 \xrightarrow{\epsilon,\epsilon\to\epsilon} q_2$ and $q_3 \xrightarrow{\epsilon,\epsilon\to\epsilon} q_4$ to transition to another state without making a change to the stack.

Example 2.42. Let $M = (Q, \Sigma, \Gamma, \delta, q_1, \perp, F)$ be a PDA where

$Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$, $\Sigma = \{a, b, c\}$, $\Gamma = \{a, \perp, \$\}$, $F = \{q_4, q_7\}$ with the following state diagram:



M recognizes the language $\{a^ib^jc^k | i, j, k \geq 0 \text{ and } i = j \text{ or } i = k\}$.

M accepts the empty string ($\epsilon = a^0b^0c^0$) with the following computation:

$q_1 \xrightarrow{\epsilon,\epsilon\to\$} q_2 \xrightarrow{\epsilon,\epsilon\to\epsilon} q_3 \xrightarrow{\epsilon,\$\to\epsilon} q_4$, $q_4 \in F$.

 doi:10.20944/preprints202506.0158.v1

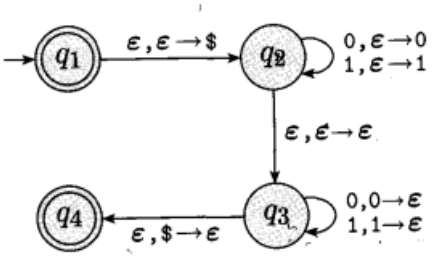
M accepts the string $a^2b^2c^3$ with the following computation:
$q_1 \xrightarrow{\epsilon,\epsilon\to\$} q_2 \xrightarrow{a,\epsilon\to a} q_2 \xrightarrow{a,\epsilon\to a} q_2 \xrightarrow{\epsilon,\epsilon\to\epsilon} q_3 \xrightarrow{b,a\to\epsilon} q_3 \xrightarrow{b,a\to\epsilon} q_3 \xrightarrow{\epsilon,\$\to\epsilon} q_4 \xrightarrow{c,\epsilon\to\epsilon} q_4 \xrightarrow{c,\epsilon\to\epsilon} q_4 \xrightarrow{c,\epsilon\to\epsilon} q_4,\ q_4 \in F.$

M accepts the string $a^2b^3c^2$ with the following computation:
$q_1 \xrightarrow{\epsilon,\epsilon\to\$} q_2 \xrightarrow{a,\epsilon\to a} q_2 \xrightarrow{a,\epsilon\to a} q_2 \xrightarrow{\epsilon,\epsilon\to\epsilon} q_5 \xrightarrow{b,\epsilon\to\epsilon} q_5 \xrightarrow{b,\epsilon\to\epsilon} q_5 \xrightarrow{b,\epsilon\to\epsilon} q_5 \xrightarrow{\epsilon,\epsilon\to\epsilon} q_6 \xrightarrow{c,a\to\epsilon} q_6 \xrightarrow{c,a\to\epsilon} q_6 \xrightarrow{\epsilon,\$\to\epsilon} q_7$, $q_7 \in F$.

Note also that the computations $q_1 \xrightarrow{\epsilon,\epsilon\to\$} q_2$, $q_3 \xrightarrow{\epsilon,\$\to\epsilon} q_4$, and $q_6 \xrightarrow{\epsilon,\$\to\epsilon} q_7$ can be replaced by $q_1 \xrightarrow{\epsilon,\epsilon\to\epsilon} q_2$ and $q_3 \xrightarrow{\epsilon,\epsilon\to\epsilon} q_4$ and $q_6 \xrightarrow{\epsilon,\epsilon\to\epsilon} q_7$ to transition to another state without making a change to the stack.

Example 2.43.

Let $M = (Q, \Sigma, \Gamma, \delta, q_1, \perp, F)$ be a PDA where
$Q = \{q_1, q_2, q_3, q_4\}$, $\Sigma = \{0,1\}$, $\Gamma = \{0,1,\perp,\$\}$, $F = \{q_1, q_4\}$ with the following state diagram:



M recognizes the language $\{ww^R \mid w \in \{0,1\}^*\}$.

If the stack is signaled to be empty at the beginning, M accepts the empty string ($\epsilon = \epsilon\epsilon^R$), because q_1 is both a start and accept state.

M accepts the string 001100 with the following computation:
$q_1 \xrightarrow{\epsilon,\epsilon\to\$} q_2 \xrightarrow{0,\epsilon\to 0} q_2 \xrightarrow{0,\epsilon\to 0} q_2 \xrightarrow{1,\epsilon\to 1} q_2 \xrightarrow{\epsilon,\epsilon\to\epsilon} q_3 \xrightarrow{1,1\to\epsilon} q_3 \xrightarrow{0,0\to\epsilon} q_3 \xrightarrow{0,0\to\epsilon} q_3 \xrightarrow{\epsilon,\$\to\epsilon} q_4,\ q_4 \in F.$

Note also that the steps $q_1 \xrightarrow{\epsilon,\epsilon\to\$} q_2$ and $q_3 \xrightarrow{\epsilon,\$\to\epsilon} q_4$ can be replaced by $q_1 \xrightarrow{\epsilon,\epsilon\to\epsilon} q_2$ and $q_3 \xrightarrow{\epsilon,\epsilon\to\epsilon} q_4$ to transition to another state without making a change to the stack.

Instead of writing symbols one at a time to the stack, we can actually design PDAs which can write a string of symbols to the stack in one step. These PDAs are called extended PDAs. It turns out that the two kinds of PDAs are equivalent in power in that given one, we can construct the other such that the two recognize the same language. The equivalence of these two kinds of PDAs will be proved later.

Definition 2.44. An extended PDA is a 7-tuple, $M_E = (Q, \Sigma, \Gamma, \hat{\delta}, q_0, \perp, F)$ where Q, Σ, Γ & F are finite sets such that

(a) Q is the set of states

(b) Σ is the input alphabet

(c) Γ is the stack alphabet

(d) $\hat{\delta}: Q \times \Sigma_\epsilon \times \Gamma_\epsilon \longrightarrow \wp(Q \times \Gamma^*)$ is the transition function

(e) $q_0 \in Q$ is the start state

(f) $\perp \in \Gamma$ is the initial stack symbol signaling an empty stack

(g) $F \subset Q$ is the set of accept states.

M computes as follows.

Let $w = w_1 w_2 \cdots w_m$ where $w_i \in \Sigma_\epsilon$ for $1 \le i \le m$.

M accepts w iff $\exists r_0, r_1 \cdots r_m \in Q$ and $s_0, s_1 \cdots s_m \in \Gamma^*$ such that the following conditions are satisfied:

(1) $r_0 = q_0$ and $s_0 = \perp$

 doi:10.20944/preprints202506.0158.v1

(2) $(r_{i+1}, b_i) \in \hat{\delta}(r_i, w_{i+1}, a_i))$ for $0 \leq i \leq m-1$ where $a_i \in \Gamma_\epsilon$, $b_i \in \Gamma^*$ and $s_i = a_i t_i$, $s_{i+1} = b_i t_i$ where $t_i \in \Gamma^*$

(3) $r_m \in F$

When $m = 0, w = \epsilon$ and only conditions (i) and (iii) are valid which then becomes $r_0 = q_0$ and $s_0 = \perp$ and $r_0 \in F$.

Therefore, we define the extended PDA to accept ϵ whenever the start state is also an accept state and the stack is signaled to be empty.

If we write $r_i \xrightarrow{w_{i+1}, a_i \to b_i, \hat{\delta}} r_{i+1}$ for $(r_{i+1}, b_i) \in \hat{\delta}(r_i, w_{i+1}, a_i)$, conditions (i), (ii) and (iii) can be written as follows:

$$q_0 = r_0 \xrightarrow{w_1, a_0 \to b_0, \hat{\delta}} r_1 \xrightarrow{w_2, a_1 \to b_1, \hat{\delta}} r_2 \cdots r_i \xrightarrow{w_{i+1}, a_i \to b_i, \hat{\delta}} r_{i+1} \cdots r_{m-1} \xrightarrow{w_m, a_{m-1} \to b_{m-1}, \hat{\delta}} r_m, r_m \in F.$$

When there is only one transition function under consideration, the showing of $\hat{\delta}$ in the computation is usually omitted and the following shorthand is used instead:

$$q_0 = r_0 \xrightarrow{w_1, a_0 \to b_0} r_1 \xrightarrow{w_2, a_1 \to b_1} r_2 \cdots r_i \xrightarrow{w_{i+1}, a_i \to b_i} r_{i+1} \cdots r_{m-1} \xrightarrow{w_m, a_{m-1} \to b_{m-1}} r_m, r_m \in F.$$

For simplicity, we sometimes can use the notation $q_0 \xrightarrow{w, *, \hat{\delta}} r_m$ to represent a computation of w from q_0 to r_m without showing the intermediate states.

Theorem 2.45. For any extended PDA, (M_E), there is a PDA (M), such that $L(M_E) = L(M)$ and vice versa.

Proof. Construction of M from M_E.

Let $M_E = (Q_E, \Sigma, \Gamma, \hat{\delta}, q_0, \perp, F)$ be an extended PDA.

Construct PDA, $M = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ where Q and δ are to be defined as follows.

For every $(q, a, s) \in Q_E \times \Sigma_\epsilon \times \Gamma_\epsilon$, we define $\delta(q, a, s)$ as follows.

(i) If $\hat{\delta}(q, a, s) = \emptyset$, $\delta(q, a, s) = \emptyset$.

(ii) If $\hat{\delta}(q, a, s) \neq \emptyset$, $\exists$ at least one $(r, u) \in \hat{\delta}(q, a, s)$.

Let $\delta_1(q, a, s) = \{(r, \epsilon) | (r, \epsilon) \in \hat{\delta}(q, a, s)\}$.

$\forall (r, u) \in \hat{\delta}(q, a, s)$ where $(r, u) \in Q_E \times \Gamma^*$ and $u \neq \epsilon$, $\exists u_1, u_2 \cdots u_l \in \Gamma$, $l \geq 1$ such that $u = u_1 u_2 \cdots u_l$.

(Note that none of $u_1, u_2 \cdots u_l$ is ϵ.)

Create new states $q_1, q_2, \cdots q_{l-1}$ that satisfy the following conditions:

$q \xrightarrow{a, s \to u_l, \delta} q_1$ (by making $(q_1, u_l) \in \delta(q, a, s)$)

$q_1 \xrightarrow{\epsilon, \epsilon \to u_{l-1}, \delta} q_2$ (by making $\delta(q_1, \epsilon, \epsilon) = \{q_2, u_{l-1}\}$)

$q_2 \xrightarrow{\epsilon, \epsilon \to u_{l-2}, \delta} q_3$ (by making $\delta(q_2, \epsilon, \epsilon) = \{q_3, u_{l-2}\}$)

$\vdots$

$q_{l-1} \xrightarrow{\epsilon, \epsilon \to u_1, \delta} r$ (by making $\delta(q_{l-1}, \epsilon, \epsilon) = \{r, u_1\}$)

Note that the states $q_1, q_2, \cdots q_{l-1}$ thus created are not in Q_E and that

$\delta(q_i, a, s) = \emptyset$ for any other combinations of $(a, s) \neq (\epsilon, \epsilon)$ and $i \in \{1, 2, \cdots l-1\}$.

Note also that there can be more than one set of states $q_1, q_2, \cdots q_{l-1}$ and stack symbols $u_1, u_2 \cdots u_l$ to be created from each combination of (q, a, s) because there can be more than one $(r, u) \in \hat{\delta}(q, a, s)$ based on which the states and the stack symbols are created.

Let $\delta_2(q, a, s) = \bigcup_{(r,u) \in \hat{\delta}(q,a,s)} \{(q_1, u_l)\}$ where $u = u_1 u_2 \cdots u_l$, $l \geq 1$, $u_i \in \Gamma$ and

q_1 is created from (ii) above.

Let $\delta(q, a, s) = \delta_1(q, a, s) \cup \delta_2(q, a, s)$.

For each $(q, a, s, r, u) \in Q_E \times \Sigma_\epsilon \times \Gamma_\epsilon \times Q_E \times \Gamma^*$, where $(r, u) \in \hat{\delta}(q, a, s)$ and $u \neq \epsilon$, define

$$Q(q, a, s, r, u) = \{q_i | 1 \leq i \leq l-1; l \,\&\, q_i \, are \, created \, from \, (ii); (r, u) \in \hat{\delta}(q, a, s); u \neq \epsilon\}$$

(Note that $q_i \in Q(q, a, s, r, u) \Longrightarrow q_i \notin Q_E$.)

Let $P(q, a, s) = \bigcup_{(r,u) \in \hat{\delta}(q,a,s); u \neq \epsilon} Q(q, a, s, r, u)$

 doi:10.20944/preprints202506.0158.v1

Set $Q = Q_E \cup \left(\bigcup_{(q,a,s)\in Q_E\times\Sigma_\epsilon\times\Gamma_\epsilon} P(q,a,s)\right)$.
So, $M = \left(Q_E \cup \left(\bigcup_{(q,a,s)\in Q_E\times\Sigma_\epsilon\times\Gamma_\epsilon} P(q,a,s)\right), \Sigma, \Gamma, \delta, q_0, \perp, F\right)$.
The construction is now complete and it remains to show that $L(M_E) = L(M)$.
Suppose $w \in L(M)$.
$\exists w_1, w_2 \cdots w_n \in \Sigma_\epsilon$ such that $w = w_1, w_2 \cdots w_n$ where $n \geq 1$.
$\exists r_0, r_1 \cdots r_n \in Q$, $a_i, b_i \in \Gamma_\epsilon$ such that
$q_0 = r_0 \xrightarrow{w_1, a_0 \to b_0, \delta} r_1 \xrightarrow{w_2, a_1 \to b_1, \delta} r_2 \cdots r_i \xrightarrow{w_{i+1}, a_i \to b_{i,} \delta} r_{i+1} \cdots r_{n-1} \xrightarrow{w_n, a_{n-1} \to b_{n-1,} \delta} r_n, r_n \in F.$

Claim: $\forall 0 \leq i \leq n-1$, if $r_i \in Q_E$, then
$\exists j$ and $u \in \Gamma^*$ such that $i < j \leq n$, $r_i \xrightarrow{w_{i+1}, a_i \to u, \hat{\delta}} r_j$ and $w_k = \epsilon$ for $i+2 \leq k \leq j$.

Proof of Claim. From $r_i \xrightarrow{w_{i+1}, a_i \to b_{i,} \delta} r_{i+1}$ in the given computation, it follows that $(r_{i+1}, b_{i,}) \in \delta(r_i, w_{i+1}, a_i)$.
By assumption, $r_i \in Q_E$.
By construction (ii), $\delta(r_i, w_{i+1}, a_i) = \delta_1(r_i, w_{i+1}, a_i) \cup \delta_2(r_i, w_{i+1}, a_i)$.
Either $(r_{i+1}, b_{i,}) \in \delta_1(r_i, w_{i+1}, a_i)$ or $(r_{i+1}, b_{i,}) \in \delta_2(r_i, w_{i+1}, a_i)$.
(a) If $(r_{i+1}, b_{i,}) \in \delta_1(r_i, w_{i+1}, a_i)$
Since $\delta_1(r_i, w_{i+1}, a_i) = \{(r, \epsilon) | (r, \epsilon) \in \hat{\delta}(r_i, w_{i+1}, a_i)\}$, $b_{i,} = \epsilon$ and
$(r_{i+1}, \epsilon) \in \hat{\delta}(r_i, w_{i+1}, a_i)$.
Therefore, $r_i \xrightarrow{w_{i+1}, a_i \to \epsilon, \hat{\delta}} r_{i+1}$.
Since $i < i+1 \leq n$ and $\epsilon \in \Gamma^*$, Claim is proved by taking $j = i+1$ and $u = \epsilon$.
(b) If $(r_{i+1}, b_{i,}) \in \delta_2(r_i, w_{i+1}, a_i)$
Since $\delta_2(r_i, w_{i+1}, a_i) = \bigcup_{(r,u)\in\hat{\delta}(r_i, w_{i+1}, a_i)}\{(q_1, u_l)\}$, $(r_{i+1}, b_{i,}) = (q_1, u_l)$ for some $(r, u) \in \hat{\delta}(r_i, w_{i+1}, a_i)$ where $u = u_1 u_2 \cdots u_l$, $l \geq 1$, $u_i \in \Gamma$ and q_1 is created from construction (ii) above.
Therefore $r_{i+1} = q_1$ and $b_{i,} = u_l$.
$r_i \xrightarrow{w_{i+1}, a_i \to b_{i,} \delta} r_{i+1}$ now becomes $r_i \xrightarrow{w_{i+1}, a_i \to u_l, \delta} r_{i+1}$.
Furthermore, from $r_{i+1} \xrightarrow{w_{i+2}, a_{i+1} \to b_{i+1,} \delta} r_{i+2}$ in the given computation, we have
$(r_{i+2}, b_{i+1}) \in \delta(r_{i+1}, w_{i+2}, a_{i+1}) = \delta(q_1, w_{i+2}, a_{i+1})$.
Since $\delta(q_1, a, s) = \emptyset$ for all $(a, s) \neq (\epsilon, \epsilon)$, we must have $w_{i+2} = a_{i+1} = \epsilon$.
Therefore, $(r_{i+2}, b_{i+1}) \in \delta(q_1, \epsilon, \epsilon) = \{(q_2, u_{l-1})\}$.
Therefore, $r_{i+2} = q_2$ and $b_{i+1} = u_{l-1}$.
$r_{i+1} \xrightarrow{w_{i+2}, a_{i+1} \to b_{i+1,} \delta} r_{i+2}$ becomes $r_{i+1} \xrightarrow{\epsilon, \epsilon \to u_{l-1}, \delta} r_{i+2}$.
By repeating the above argument, we can obtain the following computation:
$r_i \xrightarrow{w_{i+1}, a_i \to u_l, \delta} r_{i+1} \xrightarrow{\epsilon, \epsilon \to u_{l-1}, \delta} r_{i+2} \xrightarrow{\epsilon, \epsilon \to u_{l-2}, \delta} r_{i+3} \cdots r_{i+l-1} \xrightarrow{\epsilon, \epsilon \to u_1, \delta} r_{i+l}.$
where $r_{i+1} = q_1$, $r_{i+2} = q_2$, $\cdots$ $r_{i+l-1} = q_{l-1}$, $r_{i+l} = r$ and $w_{i+2} = w_{i+3} \cdots = w_{i+l} = \epsilon$.
Let $j = i + l$.
$r_j = r_{i+1} = r$ and $r \in Q_E \Longrightarrow r_j \in Q_E$.
Also, $w_{i+2} = w_{i+3} \cdots = w_j = \epsilon$.
$(r, u) = (r_j, u)$.
Since $(r, u) \in \hat{\delta}(r_i, w_{i+1}, a_i)$, $(r_j, u) \in \hat{\delta}(r_i, w_{i+1}, a_i)$.
Therefore, $r_i \xrightarrow{w_{i+1}, a_i \to u, \hat{\delta}} r_j$.
$l \geq 1 \Longrightarrow i + l > i \Longrightarrow j > i$.
Assume for contradiction that $j > n$.
$i < n \leq j - 1$.
$i < n \leq i + l - 1$.
Therefore $r_n \in \{r_{i+1}, r_{i+2}, r_{i+3} \cdots r_{i+l-1}\} = \{q_1, q_2, q_3, \cdots q_{l-1}\}$.
This implies $r_n \notin Q_E$, which is a contradiction because $r_n \in F$ and $F \subset Q_E$.
Therefore, $i < j \leq n$.

 doi:10.20944/preprints202506.0158.v1

Claim is also true under condition (b).

Combining (a) and (b), we conclude the proof of Claim.

Since $r_0 = q_0 \in Q_E$, we can apply Claim on r_0 to obtain j_0 such that

$0 < j_0 \leq n;\ w_2 = w_3 = \cdots = w_{j_0} = \epsilon;\ r_0 \xrightarrow{w_1, a_0 \to u_0, \hat{\delta}} r_{j_0}$ with r_{j_0} also in Q_E and $u_0 \in \Gamma^*$.

Since $r_{j_0} \in Q_E$, we can again apply Claim on r_{j_0} to get r_{j_1} such that

$0 < j_0 < j_1 \leq n;\ w_{j_0+2} = w_{j_0+3} \cdots = w_{j_1} = \epsilon;\ r_{j_0} \xrightarrow{w_{j_0+1}, a_{j_0} \to u_{j_0}, \hat{\delta}} r_{j_1}$ with r_{j_1} also in Q_E & $u_{j_0} \in \Gamma^*$.

By repeating this process a number of times, we will obtain $0 < j_0 < j_1 < \cdots < j_{m-1} < j_m \leq n$ such that

$r_0 \xrightarrow{w_1, a_0 \to u_0, \hat{\delta}} r_{j_0} \xrightarrow{w_{j_0+1}, a_{j_0} \to u_{j_0}, \hat{\delta}} r_{j_1} \cdots \cdots r_{j_{m-1}} \xrightarrow{w_{j_{m-1}+1}, a_{j_{m-1}} \to u_{j_{m-1}}, \hat{\delta}} r_{j_m}$, where $u_0, u_{j_0} \cdots u_{j_{m-1}} \in \Gamma^*$.

Since n is finite, this process of creation must stop at some point and at this point, $j_m = n$.

Therefore, M_E accepts $w_1 w_{j_0+1} w_{j_1+1} \cdots w_{j_{m-1}+1}$.

By Claim, we have

$w_2 = w_3 = \cdots = w_{j_0} = \epsilon$

$w_{j_0+2} = w_{j_0+3} \cdots = w_{j_1} = \epsilon$

$\vdots$

$w_{j_{m-1}+2} = w_{j_{m-1}+3} \cdots = w_{j_m} = \epsilon$ where $j_m = n$.

Therefore, $w_1 w_{j_0+1} w_{j_1+1} \cdots w_{j_{m-1}+1} = w_1 w_2 \cdots w_n = w$.

Therefore, M_E accepts $w_1, w_2 \cdots w_n = w$.

Therefore, $w \in L(M_E)$ and hence $L(M) \subset L(M_E)$.

Conversely, assume $w \in L(M_E)$.

$\exists r_0, r_1 \cdots r_n \in Q_E;\ a_i \in \Gamma_\epsilon,\ b_i \in \Gamma^*$ for $0 \leq i \leq n-1;\ w_1, w_2 \cdots w_n \in \Sigma_\epsilon$ such that

$w = w_1 w_2 \cdots w_n$ and

$q_0 = r_0 \xrightarrow{w_1, a_0 \to b_0, \hat{\delta}} r_1 \xrightarrow{w_2, a_1 \to b_1, \hat{\delta}} r_2 \cdots r_i \xrightarrow{w_{i+1}, a_i \to b_i, \hat{\delta}} r_{i+1} \cdots r_{n-1} \xrightarrow{w_n, a_{n-1} \to b_{n-1}, \hat{\delta}} r_n, r_n \in F.$

Since $Q_E \subset Q$, $r_0, r_1 \cdots r_n \in Q$.

For all $0 \leq i \leq n-1$, $(r_{i+1}, b_i) \in \hat{\delta}(r_i, w_{i+1}, a_i) \Longrightarrow \hat{\delta}(r_i, w_{i+1}, a_i) \neq \emptyset$.

[If $b_i = \epsilon$]

$(r_{i+1}, \epsilon) \in \hat{\delta}(r_i, w_{i+1}, a_i)$

By construction (ii), $\delta_1(r_i, w_{i+1}, a_i) = \{(r, \epsilon) | (r, \epsilon) \in \hat{\delta}(r_i, w_{i+1}, a_i)\}$.

Therefore, $(r_{i+1}, \epsilon) \in \delta_1(r_i, w_{i+1}, a_i)$.

Also by construction (ii), $\delta(r_i, w_{i+1}, a_i) = \delta_1(r_i, w_{i+1}, a_i) \cup \delta_2(r_i, w_{i+1}, a_i)$.

Therefore, $(r_{i+1}, \epsilon) \in \delta(r_i, w_{i+1}, a_i)$.

Therefore, $r_i \xrightarrow{w_{i+1}, a_i \to \epsilon, \delta} r_{i+1}$.

Therefore, $r_i \xrightarrow{w_{i+1}, *, \delta} r_{i+1}$.

[If $b_i \neq \epsilon$]

$\exists b_i(1), b_i(2) \cdots b_i(l) \in \Gamma,\ l \geq 1$ such that $b_i = b_i(1) b_i(2) \cdots b_i(l)$.

By construction (ii), $\exists q_1, q_2, \cdots q_{l-1} \in Q$ such that

$r_i \xrightarrow{w_{i+1}, a_i \to b_i(l), \delta} q_1$

$q_1 \xrightarrow{\epsilon, \epsilon \to b_i(l-1), \delta} q_2$

$q_2 \xrightarrow{\epsilon, \epsilon \to b_i(l-2), \delta} q_3$

$\vdots$

$q_{l-1} \xrightarrow{\epsilon, \epsilon \to b_i(1), \delta} r_{i+1}$.

Therefore, $r_i \xrightarrow{w_{i+1}, *, \delta} r_{i+1}$.

Combining both cases of $[b_i = \epsilon]$ and $[b_i \neq \epsilon]$, we have

$r_i \xrightarrow{w_{i+1}, *, \delta} r_{i+1}$ for all $0 \leq i \leq n-1$.

Therefore,

$q_0 = r_0 \xrightarrow{w_1, *, \delta} r_1 \xrightarrow{w_2, *, \delta} r_2 \cdots r_i \xrightarrow{w_{i+1}, * \delta} r_{i+1} \cdots r_{n-1} \xrightarrow{w_n, * \delta} r_n, r_n \in F.$

Therefore, M accepts $w_1 w_2 \cdots w_n = w$.

$w \in L(M)$.

 doi:10.20944/preprints202506.0158.v1

$L(M_E) \subset L(M)$.

This completes the proof of $L(M_E) = L(M)$ for the construction of M from M_E.

(A) Construction of M_E from M.

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ be a PDA.

Construct $M_E = (Q, \Sigma, \Gamma, \hat{\delta}, q_0, \perp, F)$ where

$\hat{\delta}: Q \times \Sigma_\epsilon \times \Gamma_\epsilon \longrightarrow \wp(Q \times \Gamma^*)$ such that

$\forall (q, a, s) \in Q \times \Sigma_\epsilon \times \Gamma_\epsilon, \hat{\delta}(q, a, s) = \delta(q, a, s)$.

(Note that this is possible because $\Gamma_\epsilon \subset \Gamma^*$.)

It remains to show that $L(M_E) = L(M)$.

Let $w = w_1 w_2 \cdots w_n$ where $w_i \in \Sigma_\epsilon$ for $1 \leq i \leq n$ & $n \geq 1$.

Suppose $w \in L(M)$.

$\exists r_0, r_1 \cdots r_n \in Q,\ a_i \in \Gamma_\epsilon,\ b_i \in \Gamma_\epsilon$ for $0 \leq i \leq n-1$ such that

$q_0 = r_0 \xrightarrow{w_1, a_0 \to b_0, \delta} r_1 \xrightarrow{w_2, a_1 \to b_1, \delta} r_2 \cdots r_i \xrightarrow{w_{i+1}, a_i \to b_i, \delta} r_{i+1} \cdots r_{n-1} \xrightarrow{w_n, a_{n-1} \to b_{n-1}, \delta} r_n, r_n \in F.$

For $0 \leq i \leq n-1$,

since $r_i \xrightarrow{w_{i+1}, a_i \to b_i, \delta} r_{i+1},\ (r_{i+1}, b_i) \in \delta(r_i, w_{i+1}, a_i)$.

since $\hat{\delta}(r_i, w_{i+1}, a_i) = \delta(r_i, w_{i+1}, a_i),\ (r_{i+1}, b_i) \in \hat{\delta}(r_i, w_{i+1}, a_i)$.

Therefore, $r_i \xrightarrow{w_{i+1}, a_i \to b_i, \hat{\delta}} r_{i+1}$ for $0 \leq i \leq n-1$.

Therefore,

$q_0 = r_0 \xrightarrow{w_1, a_0 \to b_0, \hat{\delta}} r_1 \xrightarrow{w_2, a_1 \to b_1, \hat{\delta}} r_2 \cdots r_i \xrightarrow{w_{i+1}, a_i \to b_i, \hat{\delta}} r_{i+1} \cdots r_{n-1} \xrightarrow{w_n, a_{n-1} \to b_{n-1}, \hat{\delta}} r_n, r_n \in F.$

M_E accepts w.

$w \in L(M_E)$.

$L(M) \subset L(M_E)$.

Conversely, suppose $w \in L(M_E)$.

$q_0 = r_0 \xrightarrow{w_1, a_0 \to b_0, \hat{\delta}} r_1 \xrightarrow{w_2, a_1 \to b_1, \hat{\delta}} r_2 \cdots r_i \xrightarrow{w_{i+1}, a_i \to b_i, \hat{\delta}} r_{i+1} \cdots r_{n-1} \xrightarrow{w_n, a_{n-1} \to b_{n-1}, \hat{\delta}} r_n, r_n \in F,$

where $a_i \in \Gamma_\epsilon,\ b_i \in \Gamma^*$ for $0 \leq i \leq n-1$.

For $0 \leq i \leq n-1,\ r_i \xrightarrow{w_{i+1}, a_i \to b_i, \hat{\delta}} r_{i+1} \Longrightarrow (r_{i+1}, b_i) \in \hat{\delta}(r_i, w_{i+1}, a_i)$.

Since $\hat{\delta}(r_i, w_{i+1}, a_i) = \delta(r_i, w_{i+1}, a_i),\ (r_{i+1}, b_i) \in \delta(r_i, w_{i+1}, a_i)$.

Therefore, $r_i \xrightarrow{w_{i+1}, a_i \to b_i, \delta} r_{i+1}$ for $0 \leq i \leq n-1$.

Therefore,

$q_0 = r_0 \xrightarrow{w_1, a_0 \to b_0, \delta} r_1 \xrightarrow{w_2, a_1 \to b_1, \delta} r_2 \cdots r_i \xrightarrow{w_{i+1}, a_i \to b_i, \delta} r_{i+1} \cdots r_{n-1} \xrightarrow{w_n, a_{n-1} \to b_{n-1}, \delta} r_n, r_n \in F.$

Therefore M accepts w and hence $w \in L(M)$.

Therefore, $L(M_E) \subset L(M)$.

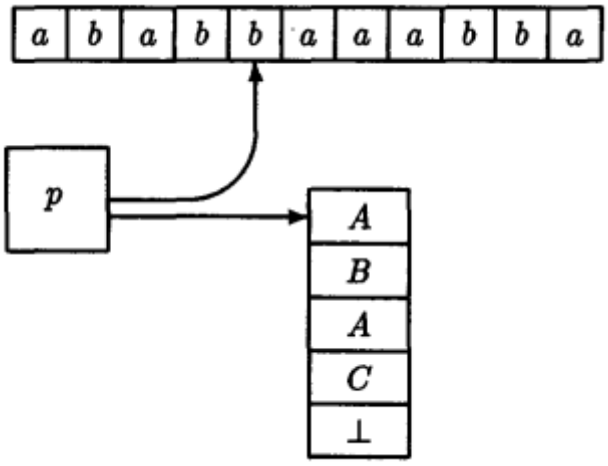
This completes the proof of $L(M_E) = L(M)$ for the construction of M_E from M.

Combining (A) and (B), we conclude the proof of Theorem 2.45.

Now that we have proved the equivalence of PDA and extended PDA, we shall no longer distinguish between M and M_E or between δ and $\hat{\delta}$. From here on, we shall be using extended PDA exclusively because it is a much more convenient tool for solving problems. We shall be using M and δ for all PDAs with the understanding that the PDAs that we are dealing with can write a string to the stack in one single step.

Definition 2.46 (Configurations of a *PDA*). A configuration of a $PDA, M = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ is an element of $Q \times \Sigma^* \times \Gamma^*$ describing the current state, the portion of the input still unread and the current stack contents at some point of a computation. For example, the configuration $(p, baaabba, ABAC\perp)$ describes the situation as shown in the following diagram.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 June 2025 doi:10.20944/preprints202506.0158.v1



Note that the portion of the input to the left of the input head, namely $abab$, has been read and cannot affect the computation hereon.

The start configuration on input w is defined as $(q_0, w, \perp)$. That is, the PDA always starts in its start state q_0, with the input head pointing to the leftmost input symbol and the stack containing only the start stack symbol $\perp$.

The next-configuration relation (denoted by $\overset{1,M}{\longrightarrow}$ or simply $\overset{M}{\rightarrow}$) describes how the PDA moves from one configuration to another in one step. It is formally defined as follows.

Definition 2.47. Let $M = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ be a PDA.

$\forall p, q \in Q, a \in \Sigma_\epsilon, A \in \Gamma_\epsilon, y \in \Sigma^*, \beta \in \Gamma^*, \gamma \in \Gamma^*,$

$\left(p \xrightarrow{a, A \to \gamma, \delta} q\right) \overset{def}{\Leftrightarrow} \left((p, ay, A\beta) \overset{1,M}{\longrightarrow} (q, y, \gamma\beta)\right).$

For any configurations C, D of M,

$\left(C \overset{0,M}{\longrightarrow} D\right) \overset{def}{\Leftrightarrow} (C = D).$

$\left(C \overset{n+1,M}{\longrightarrow} D\right) \overset{def}{\Leftrightarrow} \left(\exists E\ C \overset{n,M}{\longrightarrow} E\ \&\ E \overset{1,M}{\longrightarrow} D\right).$

$\left(C \overset{*,M}{\longrightarrow} D\right) \overset{def}{\Leftrightarrow} \left(\exists n \geq 0\ C \overset{n,M}{\longrightarrow} D\right).$

Proposition 2.48. Let $M = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ be a PDA.

For any $w = w_1 w_2 \cdots w_m$ where $w_1, w_2 \cdots w_m \in \Sigma_\epsilon$, M accepts w iff $(q_0, w, \perp) \overset{*,M}{\longrightarrow} (q, \epsilon, \gamma)$ for some $q \in F$ and $\gamma \in \Gamma^*$.

Proof. By Definition 2.44, M accepts $w = w_1 w_2 \cdots w_m$ iff $\exists r_0, r_1 \cdots r_m \in Q$, $a_i \in \Gamma_\epsilon$, $b_i \in \Gamma^*$ for $0 \leq i \leq m-1$ such that

$$q_0 = r_0 \xrightarrow{w_1, a_0 \to b_0} r_1 \xrightarrow{w_2, a_1 \to b_1} r_2 \cdots r_i \xrightarrow{w_{i+1}, a_i \to b_i} r_{i+1} \cdots r_{m-1} \xrightarrow{w_m, a_{m-1} \to b_{m-1}} r_m, r_m \in F.$$

By Definition 2.47, each one-step transitional movement is equivalent to one step of configuration movement.

For all $0 \leq i \leq m-1$, $\exists$ configurations C_i and C_{i+1} such that

$\left(r_i \xrightarrow{w_{i+1}, a_i \to b_i} r_{i+1}\right) \Leftrightarrow \left(C_i \overset{1,M}{\longrightarrow} C_{i+1}\right)$

The above transitional computation is equivalent to

$C_0 \overset{1,M}{\longrightarrow} C_1 \overset{1,M}{\longrightarrow} C_2 \cdots C_i \overset{1,M}{\longrightarrow} C_{i+1} \cdots C_{m-1} \overset{1,M}{\longrightarrow} C_m,\ r_m \in F.$

That is, $C_0 \overset{n,M}{\longrightarrow} C_m,\ r_m \in F.$

That is, $C_0 \overset{*,M}{\longrightarrow} C_m,\ r_m \in F.$

Since $C_0 = (q_0, w, \perp)$ and $C_m = (r_m, \epsilon, \gamma)$ where $\gamma \in \Gamma^*$ is the final stack content, we have

$(q_0, w, \perp) \overset{*,M}{\longrightarrow} (r_m, \epsilon, \gamma),\ r_m \in F.$

 doi:10.20944/preprints202506.0158.v1

Therefore, $(q_0, w, \perp) \xrightarrow{*,M} (q, \epsilon, \gamma)$ for some $q \in F$ and $\gamma \in \Gamma^*$.

This completes the proof of Proposition 2.48.

Proposition 2.49. Let $M = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ and $M' = (Q', \Sigma, \Gamma', \delta', q_0', \perp', F')$ be two PDAs such that

$Q \subset Q'$, $\Gamma \subset \Gamma'$ and

$\delta'(q, a, A) = \delta(q, a, A) \forall (q, a, A) \in Q \times \Sigma_\epsilon \times \Gamma_\epsilon$.

$\forall p, q \in Q$, $u, v \in \Sigma^*$ and $\alpha, \beta \in \Gamma^*$, the following statements hold:

(a) $\left[(p, u, \alpha) \xrightarrow{1,M} (q, v, \beta)\right] \Leftrightarrow \left[(p, u, \alpha) \xrightarrow{1,M'} (q, v, \beta)\right]$

(b) $\left[(p, u, \alpha) \xrightarrow{n,M} (q, v, \beta)\right] \Leftrightarrow \left[(p, u, \alpha) \xrightarrow{n,M'} (q, v, \beta)\right]$ for any $n \geq 0$

Proof.

(a)

$(p, u, \alpha) \xrightarrow{1,M} (q, v, \beta) \Leftrightarrow p \xrightarrow{a, A \to \gamma, \delta} q$ where $u = av$, $a \in \Sigma_\epsilon$, $\alpha = A\eta$, $\beta = \gamma\eta$, $A \in \Gamma_\epsilon$, $\gamma, \eta \in \Gamma^*$.

Since $Q \subset Q'$, $p, q \in Q \Rightarrow p, q \in Q'$.

Since $\Gamma \subset \Gamma'$, $A \in \Gamma_\epsilon \Rightarrow A \in \Gamma_\epsilon'$.

Since $\Gamma \subset \Gamma'$, $\Gamma^* \subset (\Gamma')^*$ and hence $\gamma \in \Gamma^* \Rightarrow \gamma \in (\Gamma')^*$.

Since $\delta'(p, a, A) = \delta(p, a, A)$ for all $(p, a, A) \in Q \times \Sigma_\epsilon \times \Gamma_\epsilon$, we have

$p \xrightarrow{a, A \to \gamma, \delta} q \Rightarrow p \xrightarrow{a, A \to \gamma, \delta'} q$

$\Rightarrow (p, av, A\eta,) \xrightarrow{1,M'} (q, v, \gamma\eta)$

$\Rightarrow (p, u, \alpha) \xrightarrow{1,M'} (q, v, \beta)$

Conversely,

$(p, u, \alpha) \xrightarrow{1,M'} (q, v, \beta)$ where $p, q \in Q$, $u, v \in \Sigma^*$ and $\alpha, \beta \in \Gamma^*$

$\Leftrightarrow p \xrightarrow{a, A \to \gamma, \delta'} q$ where $p, q \in Q$, $u, v \in \Sigma^*$, $\alpha, \beta \in \Gamma^*$, $a \in \Sigma_\epsilon$, $A \in \Gamma_\epsilon'$, $\gamma \in (\Gamma')^*$, $\alpha = A\eta$, $\beta = \gamma\eta$.

Since $\alpha \in \Gamma^*$ & $\alpha = A\eta$, $A \in \Gamma_\epsilon$ and $\eta \in \Gamma^*$.

Since $\beta \in \Gamma^*$ & $\beta = \gamma\eta$, $\gamma \in \Gamma^*$ and $\eta \in \Gamma^*$.

Therefore $(p, a, A) \in Q \times \Sigma_\epsilon \times \Gamma_\epsilon$ and hence $\delta'(p, a, A) = \delta(p, a, A)$.

Therefore,

$p \xrightarrow{a, A \to \gamma, \delta'} q \Rightarrow p \xrightarrow{a, A \to \gamma, \delta} q \Rightarrow (p, u, \alpha) \xrightarrow{1,M} (q, v, \beta)$.

(b)

This part can be proved by using the result of (a) along with an induction argument on the number of steps.

This completes the proof of Proposition 2.49.

Proposition 2.50. Let $M = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ be a PDA. It is true that

$\forall p, q \in Q, x, y, w \in \Sigma^*, \alpha, \beta, \gamma \in \Gamma^*$, integer $n \geq 1$,

$$\left((p, x, \alpha) \xrightarrow{n,M} (q, y, \beta)\right) \Rightarrow \left((p, xw, \alpha\gamma) \xrightarrow{n,M} (q, yw, \beta\gamma)\right)$$

Proof. The proof is by induction on n.

For $n = 1$, assume $(p, x, \alpha) \xrightarrow{1,M} (q, y, \beta)$.

$\exists a \in \Sigma_\epsilon$, $A \in \Gamma_\epsilon$ and $\eta, \theta \in \Gamma^*$ such that

$x = ay$, $\alpha = A\eta$, $\beta = \theta\eta$ and $p \xrightarrow{a, A \to \theta, \delta} q$.

Since $p \xrightarrow{a, A \to \theta, \delta} q$, and $xw = ayw$, $\alpha\gamma = A\eta\gamma$, and $\beta\gamma = \theta\eta\gamma$,

$(p, xw, \alpha\gamma) \xrightarrow{1,M} (q, yw, \beta\gamma)$.

 doi:10.20944/preprints202506.0158.v1

Therefore, the statement is true for $n = 1$.

For induction hypothesis,

$\left((p,x,\alpha) \xrightarrow{k,M} (q,y,\beta)\right) \Rightarrow \left((p,xw,\alpha\gamma) \xrightarrow{k,M} (q,yw,\beta\gamma)\right)$ for any integer $k \geq 1$.

For $n = k+1$, assume $(p,x,\alpha) \xrightarrow{k+1,M} (q,y,\beta)$.

$\exists p' \in Q,\ x' \in \Sigma^*,\ \alpha' \in \Gamma^*$ such that

$(p,x,\alpha) \xrightarrow{k,M} (p',x',\alpha')$ and $(p',x',\alpha') \xrightarrow{1,M} (q,y,\beta)$.

By induction hypothesis, we have

$(p,xw,\alpha\gamma) \xrightarrow{k,M} (p',x'w,\alpha'\gamma)$.

Since the statement is true for $n = 1$, we also have

$(p',x'w,\alpha'\gamma) \xrightarrow{1,M} (q,yw,\beta\gamma)$.

Combining the two computations, we have

$(p,xw,\alpha\gamma) \xrightarrow{k+1,M} (q,yw,\beta\gamma)$.

This completes the proof of Proposition 2.50.

The PDAs that we have dealt with thus far accept an input by entering an accept state upon reading the entire input. We call this kind of PDA a PDA that accepts by final state. There is another kind of PDA that accepts an input by popping the last symbol off the stack (without pushing any other symbol back on) upon reading the entire input. We call this kind of PDA a PDA that accepts by empty stack. It turns out that the two kinds of PDAs are equivalent in that given one, we can construct the other such that the two recognize the same language. Before we prove the equivalence of these two kinds of PDAs, we need a formal definition for PDAs that accept by empty stack.

Definition 2.51. A PDA that accepts by empty stack is a 6-tuple, $M_e = (Q,\Sigma,\Gamma,\delta,q_0,\perp_e)$ where $Q,\Sigma,\Gamma,\delta,q_0,\perp_e$ are defined similarly as in a PDA that accepts by final state.

M_e computes as follows:

Let $w = w_1w_2\cdots w_m$ where $w_i \in \Sigma_\epsilon$ for $1 \leq i \leq m$ & $m \geq 1$.

M_e accepts w iff $(q_0,w,\perp_e) \xrightarrow{*,M_e} (q,\epsilon,\epsilon)$ for any $q \in Q$.

(Note that the set of accept states, namely F, is not needed in the definition of acceptance by empty state.)

Lemma 2.52. For any PDA, M_e, that accepts by empty stack, there is a PDA, M_f, that accepts by final state such that $L(M_e) = L(M_f)$.

Proof. Let $M_e = (Q,\Sigma,\Gamma,\delta,q_0,\perp_e)$ where $\perp_e \in \Gamma$ is the initial stack symbol of M_e.

Construct $M_f = (Q_f,\Sigma,\Gamma_f,\delta_f,q_{start},\perp_f,\{q_{accept}\})$ where q_{start} and q_{accept} are newly created states (not in Q) with q_{start} serving as the start state of M_f and $\{q_{accept}\}$ serving as the set of accept states of M_f.

$\perp_f$ is a newly created stack symbol (not in Γ) serving as the initial stack symbol of M_f.

$Q_f = Q \cup \{q_{start},q_{accept}\}$

$\Gamma_f = \Gamma \cup \{\perp_f\}$

The transition function δ_f of M_f is defined as follows.

T1: $\delta_f(q_{start},\epsilon,\perp_f) = \{(q_0,\perp_e\perp_f)\}$ $\left(\Leftrightarrow q_{start} \xrightarrow{\epsilon,\perp_f\to\perp_e\perp_f,\delta_f} q_0\right)$

T2: $\delta_f(q,\epsilon,\perp_f) = \{(q_{accept},\epsilon)\}\ \forall q \in Q$ $\left(\Leftrightarrow q \xrightarrow{\epsilon,\perp_f\to\epsilon,\delta_f} q_{accept}\right)$

T3: $\delta_f(q,a,A) = \delta(q,a,A)$ for any $(q,a,A) \in Q \times \Sigma_\epsilon \times \Gamma_\epsilon$ where

$\delta: Q \times \Sigma_\epsilon \times \Gamma_\epsilon \to \wp(Q \times \Gamma^*)$.

T4: $\delta_f(q,a,A) = \emptyset$ for any other $(q,a,A) \in Q_f \times \Sigma_\epsilon \times (\Gamma_f)_\epsilon$.

The construction is now complete. It remains to show $L(M_e) = L(M_f)$.

 doi:10.20944/preprints202506.0158.v1

Suppose $w \in L(M_e)$.

$(q_0, w, \perp_e) \xrightarrow{n,M_e} (q, \epsilon, \epsilon)$ for some $n \geq 0$ & $q \in Q$.

By T1, $\delta_f(q_{start}, \epsilon, \perp_f) = \{(q_0, \perp_e \perp_f)\}$.

Therefore, $(q_{start}, w, \perp_f) \xrightarrow{1,M_f} (q_0, w, \perp_e \perp_f)$.

By Proposition 2.49, we have

$\{(q_0, w, \perp_e) \xrightarrow{n,M_e} (q, \epsilon, \epsilon)\} \Leftrightarrow \{(q_0, w, \perp_e) \xrightarrow{n,M_f} (q, \epsilon, \epsilon)\}$.

By Proposition 2.50, we have

$\{(q_0, w, \perp_e) \xrightarrow{n,M_f} (q, \epsilon, \epsilon)\} \Rightarrow \{(q_0, w\epsilon, \perp_e \perp_f) \xrightarrow{n,M_f} (q, \epsilon\epsilon, \epsilon \perp_f)\}$.

That is, $\{(q_0, w, \perp_e) \xrightarrow{n,M_f} (q, \epsilon, \epsilon)\} \Rightarrow \{(q_0, w, \perp_e \perp_f) \xrightarrow{n,M_f} (q, \epsilon, \perp_f)\}$.

Also by T2, $\delta_f(q, \epsilon, \perp_f) = \{(q_{accept}, \epsilon)\}$.

Therefore, $(q, \epsilon, \perp_f) \xrightarrow{1,M_f} (q_{accept}, \epsilon, \epsilon)$.

Combining, we have

$(q_{start}, w, \perp_f) \xrightarrow{1,M_f} (q_0, w, \perp_e \perp_f) \xrightarrow{n,M_f} (q, \epsilon, \perp_f) \xrightarrow{1,M_f} (q_{accept}, \epsilon, \epsilon)$.

Therefore, $(q_{start}, w, \perp_f) \xrightarrow{*,M_f} (q_{accept}, \epsilon, \epsilon)$.

Therefore, M_f accepts w.

$w \in L(M_f)$

Therefore, $L(M_e) \subset L(M_f)$.

Conversely, assume $w \in L(M_f)$.

$(q_{start}, w, \perp_f) \xrightarrow{*,M_f} (q_{accept}, \epsilon, \gamma)$ for some $\gamma \in \Gamma_f^*$.

Since there exists no transition in one step to go from q_{start} to q_{accept}, there must exist configurations (q_1, u_1, γ_1), (q_2, u_2, γ_2), $\cdots (q_i, u_i, \gamma_i)$, $\cdots (q_n, u_n, \gamma_n)$ where $n \geq 1$, $u_i \in \Sigma^*$, $\gamma_i \in \Gamma_f^*$ for $1 \leq i \leq n$, such that

$(q_{start}, w, \perp_f) \xrightarrow{1,M_f} (q_1, u_1, \gamma_1) \xrightarrow{1,M_f} \cdots \xrightarrow{1,M_f} (q_i, u_i, \gamma_i) \xrightarrow{1,M_f} \cdots \xrightarrow{1,M_f} (q_n, u_n, \gamma_n) \xrightarrow{1,M_f} (q_{accept}, \epsilon, \gamma)$.

Note that $q_i \neq q_{start}$ because each q_i has both incoming and outgoing arrows whereas q_{start} has only outgoing arrows and $q_i \neq q_{accept}$ because q_{accept} has only incoming arrows.

Therefore, for $1 \leq i \leq n$, $q_i \in Q$.

Claim 1. $(q_1, u_1, \gamma_1) = (q_0, w, \perp_e \perp_f)$.

$\delta_f(q_{start}, \epsilon, \perp_f) = \{(q_0, \perp_e \perp_f)\}$ by T1.

Therefore, $(q_{start}, w, \perp_f) \xrightarrow{1,M_f} (q_0, w, \perp_e \perp_f)$.

Since $(q_{start}, w, \perp_f) \xrightarrow{1,M_f} (q_1, u_1, \gamma_1)$, and by T4, $\delta_f(q_{start}, a, A) = \emptyset$ for any other combination of $(a, A) \neq (\epsilon, \perp_f)$, we must have

$(q_1, u_1, \gamma_1) = (q_0, w, \perp_e \perp_f)$.

Claim 2. For $1 \leq i \leq n$, $\exists \gamma_i' \in \Gamma^*$ such that $\gamma_i = \gamma_i' \perp_f$.

Claim 2 can be proved by induction on i.

For $i = 1$,

$(q_1, u_1, \gamma_1) = (q_0, w, \perp_e \perp_f)$ (By Claim 1)

Therefore, $\gamma_1 = \perp_e \perp_f$.

Take $\gamma_1' = \perp_e$.

$\gamma_1 = \gamma_1' \perp_f$.

$\perp_e \in \Gamma \Rightarrow \perp_e \in \Gamma^* \Rightarrow \gamma_1' \in \Gamma^*$.

The statement is true for $i = 1$.

For induction hypothesis ($i = k$), assume $\gamma_k = \gamma_k' \perp_f$ for $1 \leq k \leq n-1$, $\gamma_k' \in \Gamma^*$.

Consider configuration move of

$(q_k, u_k, \gamma_k) \xrightarrow{1,M_f} (q_{k+1}, u_{k+1}, \gamma_{k+1})$ which is equivalent to $q_k \xrightarrow{a,b \to c,\delta_f} q_{k+1}$ where

$a \in \Sigma_\epsilon$, $b \in (\Gamma_f)_\epsilon$, $c \in \Gamma_f^*$, $u_k = a u_{k+1}$, $\gamma_k = b\gamma_k''$, $\gamma_{k+1} = c\gamma_k''$, $\gamma_k'' \in \Gamma_f^*$.

Since $1 \leq k < k+1 \leq n$, $q_k, q_{k+1} \in Q$.

This configuration move could not have come from T1 because $q_k \neq q_{start}$.

 doi:10.20944/preprints202506.0158.v1

By induction hypothesis, $\gamma_k = \gamma'_k \perp_f$ & $\gamma'_k \in \Gamma^*$.
We examine two situations: (i) $\gamma'_k = \epsilon$ and (ii) $\gamma'_k \neq \epsilon$.
(i) If $\gamma'_k = \epsilon$

$$\gamma_k = \perp_f$$

$b = \epsilon$ or $b = \perp_f$.
If $b = \perp_f$, $q_k \xrightarrow{a,\perp_f \to c,\delta_f} q_{k+1}$.
This transition must have come from T2 where
$\delta_f(q_k, \epsilon, \perp_f) = \{(q_{accept}, \epsilon)\}$, $a = \epsilon$, $c = \epsilon$.
Therefore, $q_{k+1} = q_{accept}$, which contradicts $q_{k+1} \in Q$.
Therefore, $b = \epsilon$.
Therefore, $(q_k, a, b) = (q_k, a, \epsilon) \in Q \times \Sigma_\epsilon \times \Gamma_\epsilon$.
By T3, $\delta_f(q_k, a, b) = \delta(q_k, a, b)$.
Therefore, $q_k \xrightarrow{a,\epsilon \to c,\delta} q_{k+1}$.
Therefore, $c \in \Gamma^*$.
$\gamma_k = \perp_f = b\gamma''_k$.
Since $b = \epsilon$, $\gamma''_k = \perp_f$.
Therefore, $\gamma_{k+1} = c\gamma''_k = c\perp_f$.
The statement is true for $i = k + 1$.
(ii) If $\gamma'_k \neq \epsilon$
Since $\gamma'_k \in \Gamma^*$, $\perp_f$ is not a symbol in γ'_k.
Since $\gamma_k = \gamma'_k \perp_f$, the leftmost symbol of γ_k cannot be $\perp_f$.
Therefore, the configuration move of $(q_k, u_k, \gamma_k) \xrightarrow{1,M_f} (q_{k+1}, u_{k+1}, \gamma_{k+1})$ could not have come from T2.
Therefore, it must have come from T3 where $\delta = \delta_f$.
Therefore, $(q_k, u_k, \gamma_k) \xrightarrow{1,M_e} (q_{k+1}, u_{k+1}, \gamma_{k+1})$ by Proposition 2.49.
Therefore, $q_k \xrightarrow{a,b \to c,\delta} q_{k+1}$ where
$a \in \Sigma_\epsilon$, $b \in \Gamma_\epsilon$, $c \in \Gamma^*$, $u_k = au_{k+1}$, $\gamma_k = b\gamma''_k$, $\gamma_{k+1} = c\gamma''_k$, $\gamma''_k \in \Gamma_f^*$.
Note that $b \neq \perp_f$ because $\perp_f \notin \Gamma_\epsilon$.
By induction hypothesis, $\gamma_k = \gamma'_k \perp_f$ & $\gamma'_k \in \Gamma^*$.
Therefore, $\gamma_k = \gamma'_k \perp_f = b\gamma''_k$.
$\gamma''_k = \epsilon \Longrightarrow \gamma'_k \perp_f = b \Longrightarrow \gamma'_k \perp_f \in \Gamma_\epsilon$, which is a contradiction because $\perp_f \notin \Gamma_\epsilon$.
Therefore, $\gamma''_k \neq \epsilon$.
The rightmost symbol of γ''_k must be $\perp_f$.
$\exists \gamma'''_k \in \Gamma_f^*$ such that $\gamma''_k = \gamma'''_k \perp_f$.
Therefore, $\gamma'_k \perp_f = b\gamma'''_k \perp_f$.
Therefore, $\gamma'_k = b\gamma'''_k$.
Since $\gamma'_k \in \Gamma^*$ by induction hypothesis, $\gamma'''_k \in \Gamma^*$.
$\gamma_{k+1} = c\gamma''_k = c\gamma'''_k \perp_f = \gamma'_{k+1} \perp_f$ where $\gamma'_{k+1} = c\gamma'''_k$.
Since $c \in \Gamma^*$ & $\gamma'''_k \in \Gamma^*$, $\gamma'_{k+1} \in \Gamma^*$.
The statement is also true for $i = k + 1$.
Therefore, the statement is true for $i = k + 1$ whether or not $\gamma'_k = \epsilon$.
This completes the proof of Claim 2.
Claim 3. $(q_n, u_n, \gamma_n) = (q_n, \epsilon, \perp_f)$.
We know from above that $(q_n, u_n, \gamma_n) \xrightarrow{1,M_f} (q_{accept}, \epsilon, \gamma)$.
The only way to transition from a state in Q to q_{accept} is via T2 where
$\delta_f(q_n, \epsilon, \perp_f) = \{(q_{accept}, \epsilon)\}$.
Equivalently, $q_n \xrightarrow{\epsilon,\perp_f \to \epsilon,\delta_f} q_{accept}$.
Therefore, $(q_n, u_n, \gamma_n) \xrightarrow{1,M_f} (q_{accept}, u_n, \gamma''_n)$ where $\gamma_n = \perp_f \gamma''_n$ & $\gamma''_n \in \Gamma_f^*$.
Therefore, $(q_{accept}, u_n, \gamma''_n) = (q_{accept}, \epsilon, \gamma)$.

 doi:10.20944/preprints202506.0158.v1

Therefore, $u_n = \epsilon$ & $\gamma = \gamma_n^{"}$.
By Claim 2, $\gamma_n = \gamma_n' \perp_f$ where $\gamma_n' \in \Gamma^*$.
Therefore, $\gamma_n = \gamma_n' \perp_f = \perp_f \gamma_n^{"}$.
If $\gamma_n^{"} \neq \epsilon$, its rightmost symbol must be $\perp_f$.
Let $\gamma_n^{"} = \gamma_n''' \perp_f$ for some $\gamma_n''' \in \Gamma_f^*$.
Therefore, $\gamma_n' \perp_f = \perp_f \gamma_n''' \perp_f$.
Therefore, $\gamma_n' = \perp_f \gamma_n'''$, which is a contradiction because by Claim 2, $\gamma_n' \in \Gamma^*$ but $\perp_f$ is not in Γ^*.
Therefore, $\gamma_n^{"} = \epsilon$.
$\gamma = \gamma_n^{"} = \epsilon$.
$\gamma_n = \perp_f \gamma_n^{"} = \perp_f$.
$(q_n, u_n, \gamma_n) = (q_n, \epsilon, \perp_f)$ and Claim 3 is proved.

Claim 4. $(q_i, u_i, \gamma_i') \xrightarrow{1,M_e} (q_{i+1}, u_{i+1}, \gamma_{i+1}')$ for $1 \leq i \leq n-1$.
From above, we have $(q_i, u_i, \gamma_i) \xrightarrow{1,M_f} (q_{i+1}, u_{i+1}, \gamma_{i+1})$.
By Claim 2, $(q_i, u_i, \gamma_i' \perp_f) \xrightarrow{1,M_f} (q_{i+1}, u_{i+1}, \gamma_{i+1}' \perp_f)$ where
$\gamma_i' \in \Gamma^*$, $\gamma_{i+1}' \in \Gamma^*$, $a \in \Sigma_\epsilon$, $u_i = a u_{i+1}$, u_i, $u_{i+1} \in \Sigma^*$.
Equivalently, $q_i \xrightarrow{a,b \to c,\delta_f} q_{i+1}$ where $\gamma_i' \perp_f = b\eta$ & $\gamma_{i+1}' \perp_f = c\eta$, $b \in (\Gamma_f)_\epsilon$, $c \in \Gamma_f^*$, $\eta \in \Gamma_f^*$.
Since $q_i \neq q_{start}$, the above computation could not have come from T1.
Since $q_{i+1} \neq q_{accept}$, the above computation could not have come from T2.
Therefore, it must have come from T3 where $\delta_f(q_i, a, b) = \delta(q_i, a, b)$.
Therefore, $b \in \Gamma_\epsilon$ & $c \in \Gamma^*$ and $q_i \xrightarrow{a,b \to c,\delta} q_{i+1}$.
Since $\gamma_i' \perp_f = b\eta$, the rightmost symbol of η must be $\perp_f$.
Therefore, $\eta = \theta \perp_f$ for some $\theta \in \Gamma_f^*$.
Therefore, $\gamma_i' \perp_f = b\theta \perp_f$ and $\gamma_{i+1}' \perp_f = c\theta \perp_f$.
Therefore, $\gamma_i' = b\theta$ and $\gamma_{i+1}' = c\theta$.
Since $\gamma_i' \in \Gamma^*$ by Claim 2, $\theta \in \Gamma^*$.
Therefore, $(q_i, u_i, \gamma_i') \xrightarrow{1,M_e} (q_{i+1}, u_{i+1}, \gamma_{i+1}')$.
This completes the proof of Claim 4.
By Claim 4, we now have
$(q_1, u_1, \gamma_1') \xrightarrow{1,M_e} (q_2, u_2, \gamma_2') \xrightarrow{1,M_e} \cdots \xrightarrow{1,M_e} (q_{n-1}, u_{n-1}, \gamma_{n-1}') \xrightarrow{1,M_f} (q_n, u_n, \gamma_n')$.
By Claim 1, $(q_1, u_1, \gamma_1) = (q_0, w, \perp_e \perp_f)$.
Therefore, $q_1 = q_0$, $u_1 = w$, $\gamma_1 = \perp_e \perp_f$.
By Claim 2, $\gamma_1 = \gamma_1' \perp_f$.
Therefore, $\gamma_1' \perp_f = \perp_e \perp_f$.
Therefore, $\gamma_1' = \perp_e$.
Therefore, $(q_1, u_1, \gamma_1') = (q_0, w, \perp_e)$.
By Claim 3, $(q_n, u_n, \gamma_n) = (q_n, \epsilon, \perp_f)$.
$u_n = \epsilon$ & $\gamma_n = \perp_f$.
By Claim 2, $\gamma_n = \gamma_n' \perp_f$.
Therefore, $\gamma_n' \perp_f = \perp_f$.
Therefore, $\gamma_n' = \epsilon$.
$(q_n, u_n, \gamma_n') = (q_n, \epsilon, \epsilon)$.
Therefore,
$(q_0, w, \perp_e) \xrightarrow{1,M_e} (q_2, u_2, \gamma_2') \xrightarrow{1,M_e} \cdots \xrightarrow{1,M_e} (q_{n-1}, u_{n-1}, \gamma_{n-1}') \xrightarrow{1,M_f} (q_n, \epsilon, \epsilon)$.
$(q_0, w, \perp_e) \xrightarrow{*,M_e} (q_n, \epsilon, \epsilon)$.
M_e accepts w.
$w \in L(M_e)$.
$w \in L(M_e) \Leftrightarrow w \in L(M_f)$.
This completes the proof of Lemma 2.52.

 doi:10.20944/preprints202506.0158.v1

Lemma 2.53. For any PDA, M_f, that accepts by final state, there is a PDA, M_e, that accepts by empty stack such that $L(M_e) = L(M_f)$.

Proof. Let $M_f = (Q, \Sigma, \Gamma, \delta, q_0, \perp_f, F)$ where $\perp_f \in \Gamma$ is the initial stack symbol of M_f.

Construct $M_e = (Q_e, \Sigma, \Gamma_e, \delta_e, q_{start}, \perp_e)$ where

$Q_e = Q \cup \{q_{start}, q_{empty}\}$; $\Gamma_e = \Gamma \cup \perp_e$;

q_{start} and q_{empty} are newly created states (not in Q) with q_{start} serving as the start state of M_e and q_{empty} serving as the state in which M_e begins the process of emptying the stack (without further consuming input);

$\perp_e$ is a newly created stack symbol (not in Γ) serving as the initial stack symbol of M_e.

The transition function δ_e of M_e is defined as follows.

T1: $\delta_e(q_{start}, \epsilon, \perp_e) = \{(q_0, \perp_f \perp_e)\}$ $\left(\Leftrightarrow q_{start} \xrightarrow{\epsilon, \perp_e \to \perp_f \perp_e, \delta_e} q_0\right)$

T2: $\forall\ (q, a, A) \in Q \times \Sigma_\epsilon \times \Gamma_\epsilon$ where $\delta: Q \times \Sigma_\epsilon \times \Gamma_\epsilon \to \wp(Q \times \Gamma^*)$, $\delta_e(q, a, A) = \delta(q, a, A)$.

T3: $\forall\ q \in F$, $\delta_e(q, \epsilon, \epsilon) = \{(q_{empty}, \epsilon)\}$ $\left(\Leftrightarrow q \xrightarrow{\epsilon, \epsilon \to \epsilon, \delta_e} q_{empty}\right)$.

T4: $\forall A \in (\Gamma_e)_\epsilon$, $\delta_e(q_{empty}, \epsilon, A) = \{(q_{empty}, \epsilon)\}$ $\left(\Leftrightarrow q_{empty} \xrightarrow{\epsilon, A \to \epsilon, \delta_e} q_{empty}\right)$.

T5: For any other $(q, a, A) \in Q_e \times \Sigma_\epsilon \times (\Gamma_e)_\epsilon$, $\delta_e(q, a, A) = \emptyset$.

The construction is now complete. It remains to show $L(M_e) = L(M_f)$.

Suppose $w \in L(M_f)$.

$(q_0, w, \perp_f) \xrightarrow{n, M_f} (q, \epsilon, \gamma)$ for some $n \geq 0$ & $q \in F$, $\gamma \in \Gamma^*$.

By T1, $\delta_e(q_{start}, \epsilon, \perp_e) = \{(q_0, \perp_f \perp_e)\}$.

Therefore, $(q_{start}, w, \perp_e) \xrightarrow{1, M_e} (q_0, w, \perp_f \perp_e)$.

By Proposition 2.49, we have

$\{(q_0, w, \perp_f) \xrightarrow{n, M_f} (q, \epsilon, \gamma)\} \Leftrightarrow \{(q_0, w, \perp_f) \xrightarrow{n, M_e} (q, \epsilon, \gamma)\}.(Q \subset Q_e\ \&\ \delta_e(q, a, A) = \delta(q, a, A))$

By Proposition 2.50, we have

$\{(q_0, w, \perp_f) \xrightarrow{n, M_e} (q, \epsilon, \gamma)\} \Rightarrow \{(q_0, w\epsilon, \perp_f \perp_e) \xrightarrow{n, M_e} (q, \epsilon\epsilon, \gamma \perp_e)\}$.

That is, $\{(q_0, w, \perp_f) \xrightarrow{n, M_e} (q, \epsilon, \gamma)\} \Rightarrow \{(q_0, w, \perp_f \perp_e) \xrightarrow{n, M_e} (q, \epsilon, \gamma \perp_e)\}$.

Therefore, $(q_{start}, w, \perp_e) \xrightarrow{1, M_e} (q_0, w, \perp_f \perp_e) \xrightarrow{n, M_e} (q, \epsilon, \gamma \perp_e)$.

Therefore, $(q_{start}, w, \perp_e) \xrightarrow{*, M_e} (q, \epsilon, \gamma \perp_e)$.

By T3, $(q, \epsilon, \gamma \perp_e) \xrightarrow{1, M_e} (q_{empty}, \epsilon, \gamma \perp_e)$.

By repeated application of T4, $(q_{empty}, \epsilon, \gamma \perp_e) \xrightarrow{*, M_e} (q_{empty}, \epsilon, \epsilon)$.

Combined, $(q_{start}, w, \perp_e) \xrightarrow{*, M_e} (q, \epsilon, \gamma \perp_e) \xrightarrow{1, M_e} (q_{empty}, \epsilon, \gamma \perp_e) \xrightarrow{*, M_e} (q_{empty}, \epsilon, \epsilon)$.

Therefore, $(q_{start}, w, \perp_e) \xrightarrow{*, M_e} (q_{empty}, \epsilon, \epsilon)$.

Therefore, M_e accepts w.

$w \in L(M_e)$

Therefore, $L(M_f) \subset L(M_e)$.

Conversely, assume $w \in L(M_e)$.

There exist configurations (q_1, u_1, γ_1), (q_2, u_2, γ_2), $\cdots (q_i, u_i, \gamma_i)$, $\cdots (q_n, u_n, \gamma_n)$ where $n \geq 0$, $u_i \in \Sigma^*$, $\gamma_i \in \Gamma_e^*$ for $1 \leq i \leq n$, $q_1, q_1 \cdots q_n \in Q_e$ such that

$(q_{start}, w, \perp_e) \xrightarrow{1, M_e} (q_1, u_1, \gamma_1) \xrightarrow{1, M_e} \cdots \xrightarrow{1, M_e} (q_i, u_i, \gamma_i) \xrightarrow{1, M_e} \cdots \xrightarrow{1, M_e} (q_n, u_n, \gamma_n) \xrightarrow{1, M_e} (q, \epsilon, \epsilon)$.

Note that $q_{start} \notin \{q_1, q_1 \cdots q_n\}$ because q_{start} doesn't have incoming arrows.

If $n = 0$, $(q_{start}, w, \perp_e) \xrightarrow{1, M_e} (q, \epsilon, \epsilon)$.

By T1, $(q_{start}, w, \perp_e) \xrightarrow{1, M_e} (q_0, w, \perp_f \perp_e)$ and this is the only configuration move out of $(q_{start}, w, \perp_e)$ because $\delta_e(q_{start}, \epsilon, \perp_e) = \{(q_0, \perp_f \perp_e)\}$.

Therefore, $(q, \epsilon, \epsilon) = (q_0, w, \perp_f \perp_e)$, which is a contradiction because $\perp_f \perp_e \neq \epsilon$.

Therefore, $n \geq 1$.

In addition, we have $(q_1, u_1, \gamma_1) = (q_0, w, \perp_f \perp_e)$.

 doi:10.20944/preprints202506.0158.v1

To move from $(q_0, w, \perp_f \perp_e)$ to the final configuration (q, ϵ, ϵ), M_e must pop $\perp_e$ at some point which can only be done by T4 because T1 does not pop $\perp_e$ and T2 and T3 cannot move on $\perp_e$.

Therefore, we must have a q_{empty} somewhere between $(q_0, w, \perp_f \perp_e)$ and (q, ϵ, ϵ).

However, we can only transition into q_{empty} via a state $p \in F$ (T3).

Therefore, there must be a $p \in F$ somewhere between $(q_0, w, \perp_f \perp_e)$ and (q, ϵ, ϵ).

$\exists m =$ Max$\{i | 1 \leq i \leq n; q_i \in F\}$.

Claim 1. For $1 \leq i \leq m$, $q_i \neq q_{empty}$ and hence γ_i has $\perp_e$ as its rightmost symbol.

To prove Claim 1, we assume for contradiction $\exists k$ such that $1 \leq k \leq m$ & $q_k = q_{empty}$.

By T4, $q_{k+1} = q_{k+2} = \cdots = q_m = q_{empty}$.

This contradicts $q_m \in F$.

Therefore, $q_i \neq q_{empty}$ for $1 \leq i \leq m$.

As mentioned above, $(q_1, u_1, \gamma_1) = (q_0, w, \perp_f \perp_e)$.

Therefore, $\gamma_1 = \perp_f \perp_e$.

Since the only way to pop $\perp_e$ is by using T4 that requires the existence of q_{empty} and we cannot have any q_{empty} between q_1 and q_m, we must conclude that $\perp_e$ remains sitting at the bottom of the stack as the machine moves from q_1 to q_m.

Therefore, γ_i has $\perp_e$ as its rightmost symbol for $1 \leq i \leq m$.

Claim 2. For configuration (q_m, u_m, γ_m), $u_m = \epsilon$.

To prove Claim 2, we assume for contradiction $u_m \neq \epsilon$.

At this configuration, there are two possible ways for the machine to move: (i) is to continue to simulate M_f using T2 and (ii) is to enter q_{empty} using T3.

For (i), the machine will continue to read the input but will never enter an accept state again because it has passed q_m which is the highest accept state in this computation. By the time the machine completes reading the entire input it will come to a stop and has never had a chance to enter the state q_{empty}. Thus, $\perp_e$ remains sitting in the stack when everything comes to stop. This is contradictory to the assumption that the computation ends at (q, ϵ, ϵ).

For (ii), the machine enters q_{empty} using T3 transition:

$q_m \xrightarrow{\epsilon, \epsilon \to \epsilon, \delta_e} q_{empty}$.

$q_{m+1} = q_{empty}$, $u_{m+1} = u_m$, $\gamma_{m+1} = \gamma_m$.

$(q_m, u_m, \gamma_m) \xrightarrow{1, M_e} (q_{empty}, u_m, \gamma_m)$

Once the machine has entered q_{empty}, it will follow T4, which is $q_{empty} \xrightarrow{\epsilon, A \to \epsilon, \delta_e} q_{empty}$, to continue to pop symbols from the stack while remaining in q_{empty} and not reading any input.

Therefore, $q_{m+1} = q_{m+2} = \cdots = q_n = q = q_{empty}$ &

$u_m = u_{m+1} = u_{m+2} = \cdots = u_n = \epsilon$ (Last configuration is (q, ϵ, ϵ)).

Therefore, both (i) and (ii) contradict the original assumption that $u_m \neq \epsilon$.

Therefore, $u_m = \epsilon$.

Claim 3. For $1 \leq i \leq m$, $\exists \gamma_i' \in \Gamma^*$ such that $\gamma_i = \gamma_i' \perp_e$.

The proof of Claim 3 is by induction on i.

We show at the beginning that $(q_1, u_1, \gamma_1) = (q_0, w, \perp_f \perp_e)$

Therefore, $\gamma_1 = \perp_f \perp_e$.

Since $\perp_f \in \Gamma \subset \Gamma^*$, $\perp_f \in \Gamma^*$.

If we take $\gamma_1' = \perp_f$, $\gamma_1 = \gamma_1' \perp_e$.

The statement is true for $i = 1$.

For induction hypothesis, $\gamma_k = \gamma_k' \perp_e$ for $1 \leq k \leq m-1$, $\gamma_k' \in \Gamma^*$.

We show at the beginning that $q_{start} \notin \{q_1, q_2 \cdots q_n\}$ & $q_{empty} \notin \{q_1, q_2 \cdots q_m\}$ by Claim 1.

Therefore, $q_1, q_1 \cdots q_m \in Q$.

Consider configuration move of

$(q_k, u_k, \gamma_k) \xrightarrow{1, M_e} (q_{k+1}, u_{k+1}, \gamma_{k+1})$.

This move could not have come from T3 or T4 because $q_k \neq q_{empty}$ & $q_{k+1} \neq q_{empty}$.

The move must have come from T2 where $\delta_e = \delta$.

 doi:10.20944/preprints202506.0158.v1

By Proposition 2.49, we have
$(q_k, u_k, \gamma_k) \xrightarrow{1,M_f} (q_{k+1}, u_{k+1}, \gamma_{k+1})$.
Therefore, $q_k \xrightarrow{a,b\to c,\delta} q_{k+1}$ where
$a \in \Sigma_\epsilon,\ b \in \Gamma_\epsilon,\ c \in \Gamma^*,\ u_k = au_{k+1},\ \gamma_k = b\gamma_k^{"},\ \gamma_{k+1} = c\gamma_k^{"},\ \gamma_k^{"} \in \Gamma_e^*$.
Note that $b \in \Gamma_\epsilon \Longrightarrow b \neq \perp_e$.
By induction hypothesis, $\gamma_k = \gamma_k'\perp_e\ \&\ \gamma_k' \in \Gamma^*$.
Therefore, $\gamma_k'\perp_e = b\gamma_k^{"}$.
$\gamma_k^{"} = \epsilon \Longrightarrow \gamma_k'\perp_e = b \Longrightarrow (\gamma_k' = \epsilon)\ \&\ (b = \perp_e) \Longrightarrow$ Contradiction.
Therefore, $\gamma_k^{"} \neq \epsilon$.
The rightmost symbol of $\gamma_k^{"}$ must be $\perp_e$ (because $\gamma_k'\perp_e = b\gamma_k^{"}$)
$\exists \gamma_k''' \in \Gamma_e^*$ such that $\gamma_k^{"} = \gamma_k'''\perp_e$.
Therefore, $\gamma_k'\perp_e = b\gamma_k'''\perp_e$.
Therefore, $\gamma_k' = b\gamma_k'''$.
Since $\gamma_k' \in \Gamma^*$ by induction hypothesis, $\gamma_k''' \in \Gamma^*$.
$\gamma_{k+1} = c\gamma_k^{"} = c\gamma_k'''\perp_e = \gamma_{k+1}'\perp_e$ where $\gamma_{k+1}' = c\gamma_k'''$.
Since $c \in \Gamma^*\ \&\ \gamma_k''' \in \Gamma^*,\ \gamma_{k+1}' \in \Gamma^*$.
The statement is also true for $i = k+1$.
This completes the proof of Claim 3.

Claim 4. $(q_i, u_i, \gamma_i') \xrightarrow{1,M_f} (q_{i+1}, u_{i+1}, \gamma_{i+1}')$ for $1 \le i \le m-1$.
By assumption, $(q_i, u_i, \gamma_i) \xrightarrow{1,M_e} (q_{i+1}, u_{i+1}, \gamma_{i+1})$.
By Claim 3, $(q_i, u_i, \gamma_i'\perp_e) \xrightarrow{1,M_e} (q_{i+1}, u_{i+1}, \gamma_{i+1}'\perp_e)$
By Claim 1, $q_i \neq q_{empty}\ \&\ q_{i+1} \neq q_{empty}$ for $1 \le i \le m-1$.
Also, we point out at the beginning that $q_i \neq q_{start}$ for $1 \le i \le n$.
Therefore, this computation must have come from T2 where $\delta_e = \delta$.
By Proposition 2.49, $(q_i, u_i, \gamma_i'\perp_e) \xrightarrow{1,M_f} (q_{i+1}, u_{i+1}, \gamma_{i+1}'\perp_e)$.
Equivalently, $q_i \xrightarrow{a,b\to c,\delta} q_{i+1}$ where
$a \in \Sigma_\epsilon,\ b \in \Gamma_\epsilon, c \in \Gamma^*, \gamma_i'\perp_e = b\eta\ \&\ \gamma_{i+1}'\perp_e = c\eta,\ \eta \in \Gamma_e^*,\ u_i = au_{i+1}$.
Note that $b \in \Gamma_\epsilon \Longrightarrow b \neq \perp_e$.
$\eta = \epsilon \Longrightarrow \gamma_i'\perp_e = b \Longrightarrow (\gamma_i' = \epsilon)\ \&\ (b = \perp_e) \Longrightarrow$ Contradiction.
Therefore, $\eta \neq \epsilon$.
The rightmost symbol of η must be $\perp_e$ (because $\gamma_i'\perp_e = b\eta$).
Let $\eta = \theta\perp_e$ where $\theta \in \Gamma_e^*$.
Therefore, $\gamma_i'\perp_e = b\theta\perp_e\ \&\ \gamma_{i+1}'\perp_e = c\theta\perp_e$.
Therefore, $\gamma_i' = b\theta\ \&\ \gamma_{i+1}' = c\theta$.
Since $\gamma_i' \in \Gamma^*$ by Claim 3, $\theta \in \Gamma^*$.
$c \in \Gamma^*\ \&\ \theta \in \Gamma^* \Longrightarrow \gamma_{i+1}' \in \Gamma^*$.
Therefore, $(q_i, u_i, \gamma_i') \xrightarrow{1,M_f} (q_{i+1}, u_{i+1}, \gamma_{i+1}')$.
This completes the proof of Claim 4.
By Claim 4, we now have,
$(q_1, u_1, \gamma_1') \xrightarrow{1,M_f} (q_2, u_2, \gamma_2') \xrightarrow{1,M_f} \cdots \xrightarrow{1,M_f} (q_{m-1}, u_{m-1}, \gamma_{m-1}') \xrightarrow{1,M_f} (q_m, u_m, \gamma_m')$.
$(q_1, u_1, \gamma_1) = (q_0, w, \perp_f\perp_e)$ as shown at the beginning.
Therefore, $\gamma_1 = \perp_f\perp_e$.
By Claim 3, $\gamma_1 = \gamma_1'\perp_e$.
Therefore, $\gamma_1' = \perp_f$.
Therefore, $(q_1, u_1, \gamma_1') = (q_0, w, \perp_f)$.
By definition, $q_m \in F$.
By Claim 2, $u_m = \epsilon$.
Therefore, $(q_m, u_m, \gamma_m') = (q_m, \epsilon, \gamma_m')$.
Therefore, $(q_0, w, \perp_f) \xrightarrow{*,M_f} (q_m, \epsilon, \gamma_m')$ where $q_m \in F$.

 doi:10.20944/preprints202506.0158.v1

Therefore, M_f accepts w.
$w \in L(M_f)$.
$L(M_e) \subset L(M_f)$.
This completes the proof of Lemma 2.53.
Combining Lemma 2.52 and Lemma 2.53, we have the following theorem.

Theorem 2.54. For any PDA, M_e, that accepts by empty stack, there is a PDA, M_f, that accepts by final state such that $L(M_e) = L(M_f)$.
Conversely, For any PDA, M_f, that accepts by final state, there is a PDA, M_e, that accepts by empty stack such that $L(M_e) = L(M_f)$.

2.4. Equivalence of CFG and PDA

In this section, we shall prove that context-free grammars and pushdown automata are equivalent in power in that any language that is context-free is recognized by a pushdown automata and vice versa.

Definition 2.54. Let $G = (V, \Sigma, R, S)$ be a CFG.
Let $A \in V$, $y \in (V \cup \Sigma)^*$.
A is called the leftmost variable in y iff $\exists x \in \Sigma^*$ and $\alpha \in (V \cup \Sigma)^*$ such that $y = xA\alpha$.
x is called the head of y (written as $x = Head(y)$), $A\alpha$ is called the body of y (written as $A\alpha = Body(y)$) and α is called the tail of y (written as $\alpha = Tail(y)$).
It is clear from this definition that $y = Head(y)ATail(y) = Head(y)Body(y)$ and if $y \in \Sigma^*$, then $Head(y) = y$ and $Body(y) = Tail(y) = \epsilon$.

Definition 2.55. Let $G = (V, \Sigma, R, S)$ be a CFG.
$\forall x, y \in (V \cup \Sigma)^*$, x is a prefix of y (written as $x \leq_{PRE} y$) iff $\exists z \in (V \cup \Sigma)^*$ such that $xz = y$.

Proposition 2.56. $\leq_{PRE}$ is a reflexive and transitive relation from $(V \cup \Sigma)^*$ to $(V \cup \Sigma)^*$.

Proof. $\forall x \in (V \cup \Sigma)^*$, $\exists \epsilon \in (V \cup \Sigma)^*$ and $x\epsilon = x$.
Therefore, $x \leq_{PRE} x$ and $\leq_{PRE}$ is a reflexive.
$\forall x, y, z \in (V \cup \Sigma)^*$, if $x \leq_{PRE} y$ and $y \leq_{PRE} z$,
$\exists x', y' \in (V \cup \Sigma)^*$ such that $xx' = y$ and $yy' = z$.
Therefore, $xx'y' = z$.
Since $x'y' \in (V \cup \Sigma)^*$, $x \leq_{PRE} z$.
Therefore, $\leq_{PRE}$ is transitive.
This completes the proof of Proposition 2.56.

Proposition 2.57. Let $G = (V, \Sigma, R, S)$ be a CFG.
Let $w \in \Sigma^*$, $\gamma_i \in (V \cup \Sigma)^*$ for all $i \in \{1,2,3, \cdots n\}$.
Let A_i be the leftmost variable in γ_i and $A_i \longrightarrow \beta_i$ be the rule such that $\gamma_i \xrightarrow{A_i \to \beta_i, lm} \gamma_{i+1}$ for all $i \in \{1,2,3, \cdots n\}$.
Let $S = \gamma_1 \overset{lm}{\Rightarrow} \gamma_2 \overset{lm}{\Rightarrow} \gamma_3 \cdots \overset{lm}{\Rightarrow} \gamma_n = w$.
The following statements are true:

(a) $Head(\gamma_i) \leq_{PRE} Head(\gamma_{i+1})$

(b) $\forall 1 \leq i < j \leq n$, $Head(\gamma_i) \leq_{PRE} Head(\gamma_j)$ & hence
$Head(\gamma_1) \leq_{PRE} Head(\gamma_2) \leq_{PRE} \cdots \leq_{PRE} Head(\gamma_n) = w$

(c) $Head(\gamma_{i+1}) = Head(\gamma_i)Head(\beta_i)$ & $Body(\gamma_{i+1}) = Body(\beta_i)Tail(\gamma_i)$

 doi:10.20944/preprints202506.0158.v1

(d) If $y_i \in (V \cup \Sigma)^*$ such that $Head(\gamma_i)y_i = w$, then $Head(\beta_i) \leq_{PRE} y_i$.

Proof. Since $\gamma_i \xrightarrow{A_i \to \beta_i, lm} \gamma_{i+1}$, & $\gamma_i = Head(\gamma_i)A_i Tail(\gamma_i)$, we have $\gamma_{i+1} = Head(\gamma_i)\beta_i Tail(\gamma_i)$.

Therefore, $\gamma_{i+1} = Head(\gamma_i)Head(\beta_i)B_i Tail(\beta_i)Tail(\gamma_i)$ where B_i is the leftmost variable in β_i.
Since $Head(\gamma_i) \in \Sigma^*$ & $Head(\beta_i) \in \Sigma^*$, B_i is also the leftmost variable in γ_{i+1}.
Therefore, $Head(\gamma_{i+1}) = Head(\gamma_i)Head(\beta_i)$.
Therefore, $Head(\gamma_i) \leq_{PRE} Head(\gamma_{i+1})$.

(a) This follows (a) to be true because $\leq_{PRE}$ is transitive.

(b) $Head(\gamma_{i+1}) = Head(\gamma_i)Head(\beta_i)$ is established in the proof of (a).

It is also established in the proof of (a) that $\gamma_{i+1} = Head(\gamma_i)Head(\beta_i)B_i Tail(\beta_i)Tail(\gamma_i)$.

$Body(\gamma_{i+1}) = B_i Tail(\beta_i)Tail(\gamma_i)$.

$= Body(\beta_i)Tail(\gamma_i)$ (B_i is the leftmost variable in β_i)

(c) From (b), $Head(\gamma_{i+1}) \leq_{PRE} Head(\gamma_n) = w$.
Therefore, $\exists y_{i+1} \in \Sigma^*$ such that $Head(\gamma_{i+1})y_{i+1} = w$.
Therefore, $Head(\gamma_{i+1})y_{i+1} = Head(\gamma_i)y_i$.
By (c), $Head(\gamma_i)Head(\beta_i)y_{i+1} = Head(\gamma_i)y_i$.
Therefore, $Head(\beta_i)y_{i+1} = y_i$.
Therefore, $Head(\beta_i) \leq_{PRE} y_i$.

Lemma 2.58. For any $CFG\ G$, $\exists$ a $PDA\ M_e$ such that $L(G) = L(M_e)$.

Proof. Let $G = (V, \Sigma, R, S)$ be a CFG.

Construct $M_e = (\{q\}, \Sigma, V \cup \Sigma, \delta, q, S)$ where δ is the transition function defined as follows.
T1: $\delta(q, \epsilon, A) = \{(q, \beta) | A \to \beta$ is a rule in $R\}$.
T2: $\delta(q, a, a) = \{(q, \epsilon)\}\ \forall a \in \Sigma_\epsilon$.
T3: For all other $(q, a, A) \in \{q\} \times \Sigma_\epsilon \times (V \cup \Sigma)_\epsilon$, $\delta(q, a, A) = \emptyset$.
Note that the start variable of G is the start stack symbol of M_e.
It remains to show $L(G) = L(M_e)$.
To prove $L(G) \subset L(M_e)$, suppose $w \in L(G)$.
$\exists \gamma_1, \gamma_2, \cdots \gamma_n \in (V \cup \Sigma)^*$ such that
$S = \gamma_1 \overset{lm}{\Rightarrow} \gamma_2 \overset{lm}{\Rightarrow} \gamma_3 \cdots \gamma_i \overset{lm}{\Rightarrow} \gamma_{i+1} \cdots \overset{lm}{\Rightarrow} \gamma_n = w$.
$\forall i \in \{1,2,3, \cdots n\}$, $Head(\gamma_i) \leq_{PRE} w$ by Proposition 2.57(b).
Therefore, $\exists y_i$ such that $Head(\gamma_i)y_i = w$.
Claim. $\forall i \in \{1,2,3, \cdots n\}$, $(q, w, S) \xrightarrow{*, M_e} (q, y_i, Body(\gamma_i))$ where $Head(\gamma_i)y_i = w$.
This Claim can be proved by induction i.
For $i = 1$, $S = \gamma_1$ because $S = \gamma_1 \overset{lm}{\Rightarrow} \gamma_2 \overset{lm}{\Rightarrow} \gamma_3 \cdots \gamma_i \overset{lm}{\Rightarrow} \gamma_{i+1} \cdots \overset{lm}{\Rightarrow} \gamma_n = w$.
$Head(\gamma_1) = Head(S) = \epsilon$. $Body(\gamma_1) = Body(S) = S$.
$(q, w, S) \xrightarrow{0, M_e} (q, w, S) = (q_1, y_1, \alpha_1)$.
Therefore, $q = q_1$, $y_1 = w$ & $\alpha_1 = S$.
Therefore, $q = q_1$, $Head(\gamma_1)y_1 = \epsilon w = w$ & $\alpha_1 = Body(\gamma_1)$.
$(q, w, S) \xrightarrow{*, M_e} (q, y_1, Body(\gamma_1))$.
The statement is true for $i = 1$.
For induction hypothesis, we have
$(q, w, S) \xrightarrow{*, M_e} (q, y_k, Body(\gamma_k))$ where $Head(\gamma_k)y_k = w$ for $1 \leq k < n - 1$.
Let A_k be the leftmost variable in γ_k.
Since $\gamma_k \overset{lm}{\Rightarrow} \gamma_{k+1}$, $\exists A_k \to \beta_k$ where $\beta_k \in (V \cup \Sigma)^*$.
$(q, w, S) \xrightarrow{*, M_e} (q, y_k, Body(\gamma_k))$
$= (q, y_k, A_k Tail(\gamma_k))$

 doi:10.20944/preprints202506.0158.v1

$\xrightarrow{1,M_e} (q, y_k, \beta_k Tail(\gamma_k))$ (by T1)

$= (q, y_k, Head(\beta_k)Body(\beta_k)Tail(\gamma_k))$

By Proposition 2.57, $Head(\beta_k) \leq_{PRE} y_k$.

$\exists y_{k+1} \in (V \cup \Sigma)^*$ such that $Head(\beta_k)y_{k+1} = y_k$.

Therefore, $(q, y_k, Head(\beta_k)Body(\beta_k)Tail(\gamma_k)) = (q, Head(\beta_k)y_{k+1}, Head(\beta_k)Body(\beta_k)Tail(\gamma_k))$

$\xrightarrow{|Head(\beta_k)|,M_e} (q, y_{k+1}, Body(\beta_k)Tail(\gamma_k))$ (by repeated applications of T2 $|Head(\beta_k)|$ times)

$= (q, y_{k+1}, Body(\gamma_{k+1}))$ (by Proposition 2.57(c))

Therefore, $(q, w, S) \xrightarrow{*,M_e} (q, y_{k+1}, Body(\gamma_{k+1}))$.

Since $Head(\beta_k)y_{k+1} = y_k$,

$Head(\gamma_k)Head(\beta_k)y_{k+1} = Head(\gamma_k)y_k$.

By Proposition 2.57(c), $Head(\gamma_{k+1}) = Head(\gamma_k)Head(\beta_k)$.

By induction hypothesis, $Head(\gamma_k)y_k = w$.

Therefore, $Head(\gamma_{k+1})y_{k+1} = w$.

Therefore, the statement is true for $i = k + 1$.

To complete the proof of $L(G) \subset L(M_e)$, set $i = n$ in Claim.

$(q, w, S) \xrightarrow{*,M_e} (q, y_n, Body(\gamma_n))$ where $Head(\gamma_n)y_n = w$.

Since $\gamma_n = w$, $Head(\gamma_n) = Head(w) = w$ & $Body(\gamma_n) = Body(w) = \epsilon$.

$(Head(\gamma_n) = w)$ & $(Head(\gamma_n)y_n = w) \Longrightarrow wy_n = w \Longrightarrow y_n = \epsilon$.

Therefore, $(q, w, S) \xrightarrow{*,M_e} (q, \epsilon, \epsilon)$.

$w \in L(M_e)$.

Therefore, $L(G) \subset L(M_e)$.

(Note that at this final configuration of (q, ϵ, ϵ), we could have used the transition, $\delta(q, \epsilon, \epsilon) = \{(q, \epsilon)\}$ or $q \xrightarrow{\epsilon,\epsilon \to \epsilon,\delta} q$, to loop on without stopping. However, this machine is nondeterministic, which means that we don't have to take an option which is a bad one. On the other hand, if bad choices are made, we can loop ourselves to infinity.)

To prove $L(M_e) \subset L(G)$, let $w \in L(M_e)$.

$(q, w, S) \xrightarrow{*,M_e} (q, \epsilon, \epsilon)$.

Claim. $\forall x \in \Sigma^*$, if $(q, x, A) \xrightarrow{*,M_e} (q, \epsilon, \epsilon)$, then $A \overset{*}{\Rightarrow} x$.

Proof of this Claim is by induction on the number of steps.

$\exists n \geq 1$, such that $(q, x, A) \xrightarrow{n,M_e} (q, \epsilon, \epsilon)$.

For $n = 1$, $(q, x, A) \xrightarrow{1,M_e} (q, \epsilon, \epsilon)$.

Since $A \in V$, we must use T1 and that is, $\delta(q, \epsilon, A) = \{(q, \beta) | A \to \beta$ is a rule in $R\}$.

Therefore, $(q, x, A) \xrightarrow{1,M_e} (q, x, \beta) = (q, \epsilon, \epsilon)$.

Therefore, $x = \beta = \epsilon$.

Therefore, $A \to \epsilon$ is a rule.

$A \Longrightarrow \epsilon$ by Proposition 2.8(i).

Therefore, $A \overset{*}{\Rightarrow} \epsilon$.

The statement is true for $n = 1$.

For induction hypothesis, assume the statement is true for all $n \leq k$ with $k \geq 1$.

That is, if $(q, x, A) \xrightarrow{n,M_e} (q, \epsilon, \epsilon)$, then $A \overset{*}{\Rightarrow} x$ for all $n \leq k$.

For $n = k + 1$, assume $(q, x, A) \xrightarrow{k+1,M_e} (q, \epsilon, \epsilon)$.

Since $A \in V$, the first move must be based on T1.

Therefore, $(q, x, A) \xrightarrow{1,M_e} (q, x, Y_1Y_2 \cdots Y_m) \xrightarrow{k,M_e} (q, \epsilon, \epsilon)$ where

$A \to Y_1Y_2 \cdots Y_m$ & $Y_i \in V \cup \Sigma$ for $i \in \{1,2, \cdots m\}$.

Since $(q, x, Y_1Y_2 \cdots Y_m) \xrightarrow{k,M_e} (q, \epsilon, \epsilon)$, the machine must pop all the $Y_i s$ off the stack by the time it finishes reading input x and empties the stack.

Let x_i be the portion of x that the machine consumes while popping Y_i off the stack and returning its stack head to the position right before popping Y_{i+1} off for $i = 1,2,3, \cdots m - 1$.

 doi:10.20944/preprints202506.0158.v1

Let also x_m be the last portion of x that the machine consumes while popping Y_m off the stack and emptying the stack eventually.

Note that if Y_i is a terminal, $x_i = Y_i$. The PDA will pop Y_i using T2 and then scan the same symbol x_i from the input. The stack head will then point at Y_{i+1}.

By these assumptions, we have $x = x_1 x_2 \cdots x_m$.

In addition, we have the following sequence of computations: $(q, x_1 x_2 \cdots x_m, Y_1 Y_2 \cdots Y_m) \xrightarrow{*,M_e} (q, x_2 x_3 \cdots x_m, Y_2 Y_3 \cdots Y_m) \xrightarrow{*,M_e} (q, x_3 x_4 \cdots x_m, Y_3 Y_4 \cdots Y_m)$
$\xrightarrow{*,M_e} \cdots \xrightarrow{*,M_e} (q, x_i x_{i+1} \cdots x_m, Y_i Y_{i+1} \cdots Y_m) \xrightarrow{*,M_e} (q, x_{i+1} \cdots x_m, Y_{i+1} \cdots Y_m)$
$\xrightarrow{*,M_e} \cdots \xrightarrow{*,M_e} (q, x_m, Y_m) \xrightarrow{*,M_e} (q, \epsilon, \epsilon)$.

Since the stack head does not go below Y_{i+1} while the PDA consumes x_i, we have the following equivalent computations:

$(q, x_1, Y_1) \xrightarrow{*,M_e} (q, \epsilon, \epsilon)$
$(q, x_2, Y_2) \xrightarrow{*,M_e} (q, \epsilon, \epsilon)$
$\vdots$
$(q, x_i, Y_i) \xrightarrow{*,M_e} (q, \epsilon, \epsilon)$
$\vdots$
$(q, x_m, Y_m) \xrightarrow{*,M_e} (q, \epsilon, \epsilon)$

Since the sum of all the numbers of steps in all these computations is equal to k, the number of steps in each of these computations is less than or equal to k.

Therefore, we can use the induction hypothesis to derive the following:

$Y_1 \overset{*}{\Rightarrow} x_1$
$Y_2 \overset{*}{\Rightarrow} x_2$
$\vdots$
$Y_i \overset{*}{\Rightarrow} x_i$
$\vdots$
$Y_m \overset{*}{\Rightarrow} x_m$

Since $A \longrightarrow Y_1 Y_2 \cdots Y_m$, $A \Longrightarrow Y_1 Y_2 \cdots Y_m$ by Proposition 2.8(i).

Since $Y_i \overset{*}{\Rightarrow} x_i \ \forall i \in \{1,2,\cdots m\}$, $Y_1 Y_2 \cdots Y_m \overset{*}{\Rightarrow} x_1 x_2 \cdots x_m$ by Proposition 2.16(d).

Therefore, $A \overset{*}{\Rightarrow} x_1 x_2 \cdots x_m$.

Therefore, $A \overset{*}{\Rightarrow} x$.

The statement is true for $n = k + 1$.

To complete the proof of $L(M_e) \subset L(G)$, put $A = S$ & $x = w$ in Claim.

$\left[(q, w, S) \xrightarrow{*,M_e} (q, \epsilon, \epsilon)\right] \Longrightarrow [S \overset{*}{\Rightarrow} w]$.

$w \in L(M_e) \Longrightarrow \left[(q, w, S) \xrightarrow{*,M_e} (q, \epsilon, \epsilon)\right] \Longrightarrow [S \overset{*}{\Rightarrow} w] \Longrightarrow w \in L(G)$.

This completes the proof of $L(M_e) \subset L(G)$ and hence the proof of Lemma 2.58.

Lemma 2.59. For any PDA M_e, $\exists$ a CFG G such that $L(G) = L(M_e)$.

Proof.

Let $M_e = (Q, \Sigma, \Gamma, \delta, q_0, \perp_e)$ be a PDA that accepts by empty stack.

Construct CFG $G = (V, \Sigma, R, S)$ where V & R are defined as follows.

$V = \{S\} \cup \{[pXq] | p, q \in Q, X \in \Gamma\}$

Note that V is finite because Q & Γ are finite.

Let (P) be the procedure for creating rules in R defined as follows.

$\forall (q, a, X) \in Q \times \Sigma_\epsilon \times \Gamma$, if $\delta(q, a, X) \neq \emptyset$, $\exists (r_0, Y_1 Y_2 \cdots Y_m) \in \delta(q, a, X)$.

That is, $q \xrightarrow{a, X \to Y_1 Y_2 \cdots Y_m} r_0$.

For every $r_1, r_2, \cdots r_m \in Q$, $a \in \Sigma_\epsilon$, let

$[qXr_m] \longrightarrow a[r_0 Y_1 r_1][r_1 Y_2 r_2] \cdots [r_{m-1} Y_m r_m]$ be a rule in R.

Note that the total number of rules thus created based on each $(r_0, Y_1 Y_2 \cdots Y_m)) \in \delta(q, a, X)$ is finite because Q, m, & Σ_ϵ are finite.

 doi:10.20944/preprints202506.0158.v1

Furthermore, the set $\delta(q,a,X)$ is finite & the total number of such sets, $\delta(q,a,X)$ is finite because the total number of $(q,a,X) \in Q \times \Sigma_\epsilon \times \Gamma$ is finite.

Therefore, the total number of rules thus created for any given PDA, M_e is finite.

Let $R_1 =$ the set of rules created by (P).

$R_2 = \{S \longrightarrow [q_0 \perp_e p] | p \in Q\}$.

$R = R_1 \cup R_2$.

The construction of G is complete and we now proceed to prove $L(G) = L(M_e)$.

Claim 1. $S \overset{*}{\Rightarrow} w$ iff $[q_0 \perp_e p] \overset{*}{\Rightarrow} w$ for some $p \in Q$.

<Proof of Claim 1>

"If $S \overset{*}{\Rightarrow} w$"

$\exists n \geq 1$ such that $S \overset{n}{\Rightarrow} w$.

$S \overset{1}{\Rightarrow} \beta \overset{n-1}{\Longrightarrow} w$ where $\beta \in (V \cup \Sigma)^*$.

By Proposition 2.8(i), $S \longrightarrow \beta$ is a rule.

This rule must be from R_2.

Therefore, $S \longrightarrow [q_0 \perp_e p]$ for some $p \in Q$.

Therefore, $S \overset{1}{\Rightarrow} [q_0 \perp_e p] \overset{n-1}{\Longrightarrow} w$.

Therefore, $[q_0 \perp_e p] \overset{n-1}{\Longrightarrow} w$.

Therefore, $[q_0 \perp_e p] \overset{*}{\Rightarrow} w$ for some $p \in Q$.

"If $[q_0 \perp_e p] \overset{*}{\Rightarrow} w \; for \; some \; p \in Q$"

By construction, $S \longrightarrow [q_0 \perp_e p]$ is a rule in R_2.

By Proposition 2.8(i), $S \overset{1}{\Rightarrow} [q_0 \perp_e p]$.

Therefore, $S \overset{1}{\Rightarrow} [q_0 \perp_e p] \overset{*}{\Rightarrow} w$.

Therefore, $S \overset{*}{\Rightarrow} w$.

This completes the proof of Claim 1.

Claim 2. $\forall p, q \in Q$, $X \in \Gamma$, $w \in \Sigma^*$, $[qXp] \overset{*}{\Rightarrow} w$ iff $(q,w,X) \overset{*,M_e}{\longrightarrow} (p,\epsilon,\epsilon)$.

<Proof of Claim 2>

"If"

Assume $(q,w,X) \overset{*,M_e}{\longrightarrow} (p,\epsilon,\epsilon)$.

$\exists n \geq 1$ such that $(q,w,X) \overset{n,M_e}{\longrightarrow} (p,\epsilon,\epsilon)$.

The proof of $[qXp] \overset{*}{\Rightarrow} w$ is by induction on n.

For $n = 1$, $(q,w,X) \overset{1,M_e}{\longrightarrow} (p,\epsilon,\epsilon)$

Therefore, $q \overset{a,X \to \epsilon}{\longrightarrow} p$ where $a \in \Sigma_\epsilon$ & $w = a\epsilon = a$.

By (P), if $q \overset{a,X \to Y_1 Y_2 \cdots Y_m}{\longrightarrow} r_0$, then $\exists$ a rule $[qXr_m] \longrightarrow a[r_0 Y_1 r_1][r_1 Y_2 r_2] \cdots [r_{m-1} Y_m r_m]$ for some $r_1, r_2, \cdots r_m \in Q$.

In this case, $Y_1 Y_2 \cdots Y_m = \epsilon$ which means $m = 0$ and hence $r_m = r_0$.

Therefore, $\exists$ a rule $[qXr_0] \longrightarrow a$.

Since $p = r_0$ & $w = a$, the rule becomes $[qXp] \longrightarrow w$.

By Proposition 2.8(i), $[qXp] \Longrightarrow w$.

Therefore, $[qXp] \overset{*}{\Rightarrow} w$.

The statement is true for $n = 1$.

Assume the statement is true for all $n \leq k$ where $k \geq 1$.

That is, $\left\{(q,w,X) \overset{n,M_e}{\longrightarrow} (p,\epsilon,\epsilon)\right\} \Rightarrow \left\{[qXp] \overset{*}{\Rightarrow} w\right\}$ for all $n \leq k$.

For $n = k+1$, assume $(q,w,X) \overset{k+1,M_e}{\longrightarrow} (p,\epsilon,\epsilon)$.

$\exists Y_1, Y_2, \cdots Y_m \in \Gamma$, $a \in \Sigma_\epsilon$, $x \in \Sigma^*$, $w = ax$, $r_0 \in Q$ such that $q \overset{a,X \to Y_1 Y_2 \cdots Y_m}{\longrightarrow} r_0$.

Therefore, $(q,w,X) \overset{1,M_e}{\longrightarrow} (r_0, x, Y_1 Y_2 \cdots Y_m) \overset{k,M_e}{\longrightarrow} (p,\epsilon,\epsilon)$.

Since $(r_0, x, Y_1 Y_2 \cdots Y_m) \overset{k,M_e}{\longrightarrow} (p,\epsilon,\epsilon)$, using the same argument as used in the proof of Lemma 2.58, we can deduce the following computations:

 doi:10.20944/preprints202506.0158.v1

$(r_0, x_1, Y_1) \xrightarrow{*,M_e} (r_1, \epsilon, \epsilon)$
$(r_1, x_2, Y_2) \xrightarrow{*,M_e} (r_2, \epsilon, \epsilon)$

$\vdots$

$(r_{i-1}, x_i, Y_i) \xrightarrow{*,M_e} (r_i, \epsilon, \epsilon)$

$\vdots$

$(r_{m-1}, x_m, Y_m) \xrightarrow{*,M_e} (r_m, \epsilon, \epsilon)$ where
$r_1,\ r_2 \cdots r_{m-1} \in Q,\ r_m = p,$

x_i is the portion of x that the machine consumes while popping Y_i off the stack and returning its stack head to the position right before popping Y_{i+1} off for $i = 1,2,3, \cdots m-1$ and x_m is the last portion of x that the machine consumes while popping Y_m off the stack and emptying the stack eventually.

Note that the machine goes from state r_{i-1} to state r_i after completing the above actions & $x = x_1 x_2 \cdots x_m$.

Since each computation $(r_{i-1}, x_i, Y_i) \xrightarrow{*,M_e} (r_i, \epsilon, \epsilon)$ is part of the computation $(r_0, x, Y_1 Y_2 \cdots Y_m) \xrightarrow{k,M_e} (p, \epsilon, \epsilon)$, each one makes no more than k moves.

By induction hypothesis, $[r_{i-1} Y_i r_i] \overset{*}{\Rightarrow} x_i$ for $i = 1,2, \cdots m$.
As shown above, $q \xrightarrow{a, X \to Y_1 Y_2 \cdots Y_m} r_0$ & since $r_1,\ r_2 \cdots r_{m-1}, p \in Q$, by (P),
$\exists$ a rule $[qXp] \to a[r_0 Y_1 r_1][r_1 Y_2 r_2] \cdots [r_{m-1} Y_m p]$.
By Proposition 2.8(i), $[qXp] \Rightarrow a[r_0 Y_1 r_1][r_1 Y_2 r_2] \cdots [r_{m-1} Y_m p]$.
Since $a \overset{0}{\Rightarrow} a$ & $[r_{i-1} Y_i r_i] \overset{*}{\Rightarrow} x_i$ for $i = 1,2, \cdots m$.
$a[r_0 Y_1 r_1][r_1 Y_2 r_2] \cdots [r_{m-1} Y_m p] \overset{*}{\Rightarrow} a x_1 x_2 \cdots x_m$ by Proposition 2.16(d).
Therefore, $[qXp] \overset{*}{\Rightarrow} a x_1 x_2 \cdots x_m$.
Since $w = ax = a x_1 x_2 \cdots x_m$, $[qXp] \overset{*}{\Rightarrow} w$.
Therefore, $\left\{(q, w, X) \xrightarrow{k+1,M_e} (p, \epsilon, \epsilon)\right\} \Rightarrow \left\{[qXp] \overset{*}{\Rightarrow} w\right\}$
Therefore, the statement is true for $n = k + 1$.
This completes the proof of the "If" part of Claim 2.
"Only if"
Assume $[qXp] \overset{*}{\Rightarrow} w$.
$\exists n \geq 1$ such that $[qXp] \overset{n}{\Rightarrow} w$.
The proof of $(q, w, X) \xrightarrow{*,M_e} (p, \epsilon, \epsilon)$ is by induction on n.
For $n = 1$, $[qXp] \overset{1}{\Rightarrow} w$.
By Proposition 2.8(i), $[qXp] \to w$ is a rule in R_1 (It's not in R_2 because $[qXp] \neq S$).
Since every rule in R_1 is of the form $[qXr_m] \to a[r_0 Y_1 r_1][r_1 Y_2 r_2] \cdots [r_{m-1} Y_m r_m]$ where
$q \xrightarrow{a, X \to Y_1 Y_2 \cdots Y_m} r_0$ & $r_1,\ r_2 \cdots r_{m-1}, r_m \in Q$.
In this particular case, w is not a variable.
Therefore, $[r_0 Y_1 r_1][r_1 Y_2 r_2] \cdots [r_{m-1} Y_m r_m] = \epsilon$ or $m = 0$.
Therefore, $a = w$ & $r_m = r_0$.
Since $[qXp] = [qXr_m]$, $p = r_m = r_0$.
We must have $q \xrightarrow{a, X \to \epsilon} p$.
Therefore, $(q, w, X) \xrightarrow{1,M_e} (p, x, \epsilon)$ where $w = ax$.
As shown above, $a = w$.
Therefore, $x = \epsilon$.
Therefore, $(q, w, X) \xrightarrow{1,M_e} (p, \epsilon, \epsilon)$.
Therefore, $(q, w, X) \xrightarrow{*,M_e} (p, \epsilon, \epsilon)$.
The statement is true for $n = 1$.
For induction hypothesis, assume it is true that
$\left\{[qXp] \overset{n}{\Rightarrow} w\right\} \Rightarrow \left\{(q, w, X) \xrightarrow{*,M_e} (p, \epsilon, \epsilon)\right\}$ for all $n \leq k$ where $k \geq 1$.

 doi:10.20944/preprints202506.0158.v1

For $n = k+1$, assume $[qXp] \overset{k+1}{\Longrightarrow} w$.

Therefore, $[qXp] \overset{1}{\Rightarrow} \beta \overset{k}{\Rightarrow} w$ where $\beta \in (V \cup \Sigma)^*$.

By Proposition 2.8(i), $[qXp] \longrightarrow \beta$ is a rule in R_1.

This rule must be of the form $[qXp] \longrightarrow a[r_0Y_1r_1][r_1Y_2r_2]\cdots[r_{m-1}Y_mp]$ where $r_0, r_1, r_2 \cdots r_{m-1} \in Q$, $a \in \Sigma_\epsilon$, $Y_1, Y_2, \cdots Y_m \in \Gamma$, $q \xrightarrow{a, X \to Y_1Y_2\cdots Y_m} r_0$.

$[qXp] \overset{1}{\Rightarrow} a[r_0Y_1r_1][r_1Y_2r_2]\cdots[r_{m-1}Y_mp]$ by Proposition 2.8(i).

Therefore, $[qXp] \overset{1}{\Rightarrow} a[r_0Y_1r_1][r_1Y_2r_2]\cdots[r_{m-1}Y_mp] \overset{k}{\Rightarrow} w$.

By Proposition 2.28(ii), $\exists w_1, w_2 \cdots w_m \in \Sigma^*$ such that $[r_{i-1}Y_ir_i] \overset{*}{\Rightarrow} w_i$ in no more than k steps for $i = 1,2,\cdots m$ and $w = aw_1w_2\cdots w_m$.

By induction hypothesis, $(r_{i-1}, w_i, Y_i) \overset{*,M_e}{\longrightarrow} (r_i, \epsilon, \epsilon)$ for $i = 1,2,\cdots m$.

By Proposition 2.50,

$(r_{i-1}, w_iw_{i+1}\cdots w_m, Y_iY_{i+1}\cdots Y_m) \overset{*,M_e}{\longrightarrow} (r_i, w_{i+1}\cdots w_m, Y_{i+1}\cdots Y_m)$ for $i = 1,2,\cdots m$.

$(r_0, w_1w_2\cdots w_m, Y_1Y_2\cdots Y_m) \overset{*,M_e}{\longrightarrow} (r_1, w_2\cdots w_m, Y_2\cdots Y_m)$ for $i = 1$.

$(r_1, w_2\cdots w_m, Y_2\cdots Y_m) \overset{*,M_e}{\longrightarrow} (r_2, w_3\cdots w_m, Y_3\cdots Y_m)$ for $i = 2$.

$\vdots$

$(r_{m-1}, w_m, Y_m) \overset{*,M_e}{\longrightarrow} (r_m, \epsilon, \epsilon)$ for $i = m-1$ where $r_m = p$.

Furthermore, as shown above, $q \xrightarrow{a, X \to Y_1Y_2\cdots Y_m} r_0$.

Therefore, $(q, aw_1w_2\cdots w_m, X) \overset{1,M_e}{\longrightarrow} (r_0, w_1w_2\cdots w_m, Y_1Y_2\cdots Y_m)$.

Since $w = aw_1w_2\cdots w_m$, $(q, w, X) \overset{1,M_e}{\longrightarrow} (r_0, w_1w_2\cdots w_m, Y_1Y_2\cdots Y_m)$.

Connecting all the computations, we have

$(q, w, X) \overset{1,M_e}{\longrightarrow} (r_0, w_1w_2\cdots w_m, Y_1Y_2\cdots Y_m) \overset{*,M_e}{\longrightarrow} (r_1, w_2\cdots w_m, Y_2\cdots Y_m)$
$\overset{*,M_e}{\longrightarrow} (r_2, w_3\cdots w_m, Y_3\cdots Y_m) \overset{*,M_e}{\longrightarrow} \cdots \overset{*,M_e}{\longrightarrow} (r_{m-1}, w_m, Y_m) \overset{*,M_e}{\longrightarrow} (p, \epsilon, \epsilon)$.

Therefore, $(q, w, X) \overset{*,M_e}{\longrightarrow} (p, \epsilon, \epsilon)$.

Therefore, $\left\{[qXp] \overset{k+1}{\Longrightarrow} w\right\} \Longrightarrow \left\{(q, w, X) \overset{*,M_e}{\longrightarrow} (p, \epsilon, \epsilon)\right\}$.

This completes the proof of Claim 2.

We now get back to the proof of $L(G) = L(M_e)$.

$w \in L(G) \Leftrightarrow S \overset{*}{\Rightarrow} W$

$\Leftrightarrow [q_0 \perp_e p] \overset{*}{\Rightarrow} w$ for some $p \in Q$ (Claim 1)

$\Leftrightarrow (q_0, w, \perp_e) \overset{*,M_e}{\longrightarrow} (p, \epsilon, \epsilon)$ (Claim 2)

$\Leftrightarrow M_e$ accepts w

$\Leftrightarrow w \in L(M_e)$

Therefore, $L(G) = L(M_e)$.

This completes the proof of Lemma 2.59.

Combining Lemma 2.58 and Lemma 2.59, we have the following theorem.

Theorem 2.60. For any $CFG\ G$, $\exists$ a $PDA\ M_e$ such that $L(G) = L(M_e)$.
Conversely, for any $PDA\ M_e$, $\exists$ a $CFG\ G$ such that $L(G) = L(M_e)$.

2.5. The Pumping Lemma for Context Free Languages

In this section, we shall develop a tool for showing that a language is not context free. This tool is called "The Pumping Lemma for context free languages." This Pumping Lemma is analogous to the pumping lemma we study in Chapter 1 for regular languages. The difference this time is we are pumping two strings rather than one and the string that we are dealing with is broken down into five substrings in contrast to three substrings in the case of regular languages.

Theorem 2.61. Let $G = (V, \Sigma, R, S)$ be a CFG in Chomsky Normal Form.

 doi:10.20944/preprints202506.0158.v1

Let $Pt(A,w,h)$ be the parse tree corresponding to this grammar in accordance with the meaning of Theorem 2.33 where $A(\in V)$ is the root, $w(\in \Sigma^*)$ is the yield and h is the height of the parse tree. Then it is true that $|w| \leq 2^{h-1}$.

Proof. The proof of this theorem is by induction on h.

For $h = 1$, $Pt(A,w,h)$ is a 1-level tree with A at the zero level and w at the first level.

The only forms of rules in Chomsky Normal Form are:

$A \longrightarrow BC$ where $A\epsilon V$ and $B,C\epsilon V\backslash\{S\}$

$A \longrightarrow a$ where $a\epsilon\Sigma \subset \Sigma^*$

$S \longrightarrow \epsilon$ where $S =$ Start Variable.

Since $w \in \Sigma^*$, we have either $A \longrightarrow a$ or $S \longrightarrow \epsilon$.

Therefore, $w = a$ or $w = \epsilon$.

$w = a \Rightarrow |w| = 1 \Rightarrow |w| = 2^0 \leq 2^{h-1}$.

$w = \epsilon \Rightarrow |w| = 0 \Rightarrow |w| \leq 2^0 \leq 2^{h-1}$.

Either case, we have statement being true for $h = 1$.

For induction hypothesis, assume the statement is true for all $h \leq k$ where $k \geq 1$.

Consider a parse tree, $Pt(A,w,k+1)$, that correspond to G according to the meaning of Theorem 2.33.

Since $k \geq 1$, the height of $Pt(A,w,k+1)$ is greater than or equal to 2. Hence the children of A which appear in the first level cannot be a or ϵ.

They must be B and C with $B,C\epsilon V\backslash\{S\}$.

Using similar argument as we use in proving Theorem 2.33, we can show the following:

(i) The combination of all branches of B (respectively C) form a subtree $Pt(B,w_1,h_1)$ (respectively $Pt(C,w_2,h_2)$)

(ii) $h_1 \leq k$ & $h_2 \leq k$

(iii) $w = w_1w_2$.

By (ii) and induction hypothesis, $|w_1| \leq 2^{h_1-1}$ & $|w_2| \leq 2^{h_2-1}$.

$$
\begin{aligned}
|w| &= |w_1| + |w_2| \\
&\leq 2^{h_1-1} + 2^{h_2-1} \\
&\leq 2^{k-1} + 2^{k-1} \\
&= 2 \cdot 2^{k-1} \\
&= 2^k \\
&= 2^{(k+1)-1}
\end{aligned}
$$

This completes the induction proof of Theorem 2.61.

Proposition 2.62. Let $Pt(A,z,h)$ be a parse tree for $CFG, G = (V,\Sigma,R,S)$ and $Pt(B,w,k)$ be the largest subtree of $Pt(A,z,h)$ where w, $z \in \Sigma^*$. Then $\exists x,y \in \Sigma^*$ such that $z = xwy$. Furthermore, the nodes on any path from A to x (respectively y) cannot be a node in $Pt(B,w,k)$.

Proof. By T13, every leaf of a subtree is also a leaf of the parent tree.

Therefore, $w \sqsubset z$.

Therefore, $z = xwy$ for some $x,y \in \Sigma^*$.

Let $(A,v_1,v_2,\cdots v_n,l)$ be a path from A to l where l is a symbol in x.

There exists a $v_i(i = 1,2,\cdots n)$ on this path such that v_i and B are at the same level.

If v_i and B are the same node, $(v_i,\cdots v_n,l)$ is a branch rooted at B and by T11, it is a path inside $Pt(B,w,k)$.

This means that l is a symbol in w and this contradicts the assumption that l is a symbol in x.

Therefore, v_i cannot be the same node as B.

 doi:10.20944/preprints202506.0158.v1

Since l is to the left of every symbol in w and v_i is an ancestor of l, by T12, v_i is to the left of B.

Let v_j be a node on the path $(A, v_1, v_2, \cdots v_i \cdots v_n, l)$.

If $j < i$, v_j is above the level of B and hence v_j is not a node in $Pt(B, w, k)$.

If $j > i$, v_j is a descendant of v_i and hence by T12, v_j is to the left of all descendants of B at the same level.

Therefore, v_j cannot be a node in $Pt(B, w, k)$.

With similar argument, we can also prove that if v_j is a node on a path from A to any l' in y, v_j cannot be a node in $Pt(B, w, k)$.

This completes the proof of Proposition 2.62.

Theorem 2.63. Let $Pt(S, z, h)$ be a parse tree for $CFG, G = (V, \Sigma, R, S)$ and $Pt(A, w, k)$ be the largest subtree of $Pt(S, z, h)$ rooted at A such that $z = xwy$ where $x, y, w, z \in \Sigma^*$.

If $Pt(A, w, k)$ is replaced by another parse tree $Pt(A, w', k')$, to form a new tree $Pt(S, z', h')$, then $z' = xw'y$.

Proof. By Proposition 2.62, we can write $z' = x'w'y'$ for some x', $y' \in \Sigma^*$ because $Pt(A, w', k')$ is a subtree of $Pt(S, z', h')$.

See Figure 2.13 below.

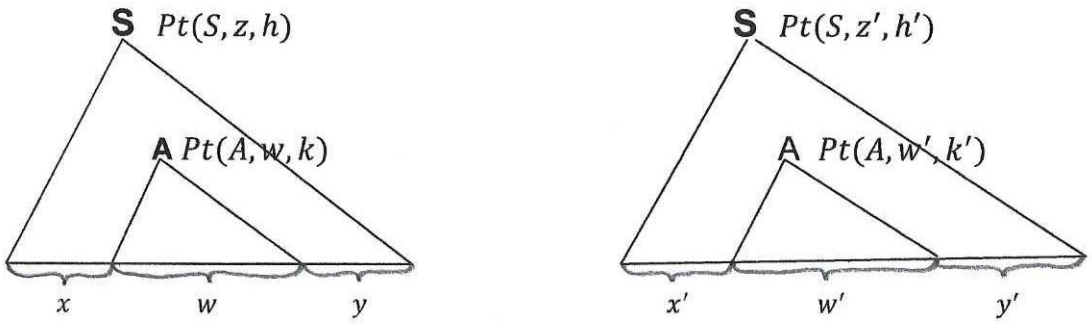


Figure 2.13. Caption.

Let l be a leaf in x.

By T8, there is a unique path from S to l in $Pt(S, z, h)$.

Let's call this path $(S, v_1, v_2, \cdots v_i \cdots v_n, l)$ where v_i is at the same level of A.

By Proposition 2.62, $(S, v_1, v_2, \cdots v_i \cdots v_n, l)$ is not affected by the removal of $Pt(A, w, k)$ and in addition, v_i is to the left of A.

Therefore, $(S, v_1, v_2, \cdots v_i \cdots v_n, l)$ remains a path in the new $Pt(S, z', h')$.

Therefore, l is a leaf in z'.

l is not in w' because w' consists of all the leaves created from the addition of $Pt(A, w', k')$.

If l is in y', the ancestor of l at the level of A, namely v_i, must be to the right of A which contradicts what we have shown above and that is v_i is to the left of A.

Therefore, l cannot be in y'.

Therefore, l is in x'.

Therefore, $x \sqsubset x'$.

Conversely, if l' is a leaf in x',

by T8, there is a unique path from S to l' in $Pt(S, z', h')$.

Let's call this path $(S, v_1', v_2', \cdots v_i' \cdots v_n', l')$ where v_i' and A are at the same level.

By Proposition 2.62, $v_1', v_2', \cdots v_i' \cdots v_n', l'$ are not in $Pt(A, w', k')$ and v_i' is to the left of A.

These nodes must have come from $Pt(S, z, h)$.

 doi:10.20944/preprints202506.0158.v1

In addition, they are not in $Pt(A,w,k)$ either because if they were, they would have been eliminated by the replacement of $Pt(A,w,k)$.

Therefore, l' is not in w.

If l' is in y, v_i' would be to the right of A, which contradicts what we have shown above and that is v_i' is to the left of A.

Therefore, l' cannot be in y.

Therefore, l' is in x.

Therefore, $x' \sqsubset x$.

Therefore, $x = x'$.

With similar argument, we can also prove that $y = y'$.

This completes the proof of Theorem 2.63.

Theorem 2.64A (The Pumping Lemma for *CFL*s). Let L be a CFL.

$\exists p > 0$ such that if $z \in L$ and $|z| \geq p$, then

$z = uvwxy$ for some $u,v,w,x,y \in \Sigma^*$ with the following conditions satisfied:

(i) $\forall i \geq 0,\ uv^iwx^iy \in L$

(ii) $vx \neq \epsilon$

(iii) $|vwx| \leq p$

Proof. By Theorem 2.37, there exists a CFG, $G = (V,\Sigma,R,S)$ in Chomsky Normal Form such that $L = L(G)$. Let $p = 2^m$ where $m = |V| =$ the number of variables in V.

If $z \in L$, $z \in L(G)$.

$S \overset{*}{\Rightarrow} z$

By Theorem 2.33, there is a parse tree for G with root S and yield z.

Let this parse tree be represented by $Pt(S,z,h)$ where h is the height of the tree.

By Theorem 2.61, $|z| \leq 2^{h-1}$.

If $|z| \geq p = 2^m$, $2^m \leq 2^{h-1}$.

$m \leq h-1$.

$h \geq m+1$.

By T9, $\exists$ a path from S to a where a is a leaf in z such that the length of this path is equal to h.

(Note that this is the longest path in the tree.)

Since $h \geq m+1$, there are at least $m+2$ nodes on this path.

Let $(V_1,V_2,\cdots\cdots V_m,V_{m+1},a)$ be the lowest portion of this path where

$V_1,V_2,\cdots\cdots V_m,V_{m+1} \in V$.

See Figure 2.14 below.

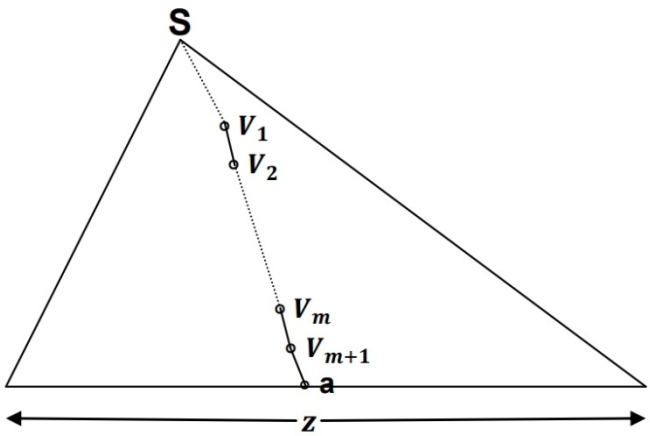


Figure 2.13. Caption.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 June 2025 doi:10.20944/preprints202506.0158.v1

Note that this is the longest path from V_1 to a leaf.

Since $m = |V|$, by the pigeonhole principle, $\exists 1 \leq i < j \leq m+1$ such that $V_i = V_j$.

Let $Pt(V_j, w, h_j)$ be the largest subtree rooted at V_j and $Pt(V_i, w', h_i)$ be the largest subtree rooted at V_i.

See Figure 2.15 below.

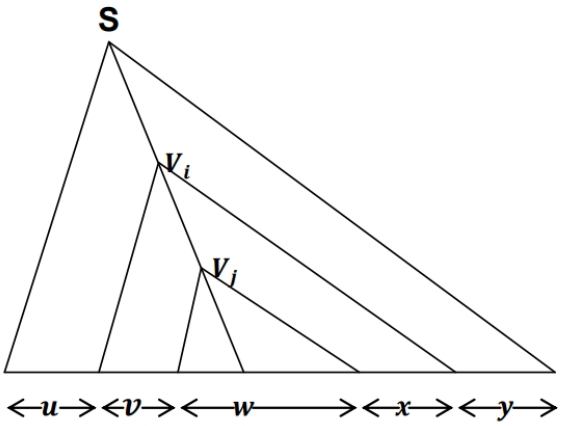


Figure 2.13. Caption.

As can be seen in Figure 2.15, $Pt(V_j, w, h_j)$ is a subtree of $Pt(V_i, w', h_i)$ which in turn is a subtree of the parent tree $Pt(S, z, h)$.

By Proposition 2.62, we can write the yield of $Pt(V_i, w', h_i)$ as vwx where $v, x \in \Sigma^*$ and the yield of $Pt(S, z, h)$ as $uvwxy$ where $u, y \in \Sigma^*$.

That is, $z = uvwxy$ and $w' = vwx$.

Since $V_i = V_j$, we can replace $Pt(V_j, w, h_j)$ by $Pt(V_i, vwx, h_i)$ to form a new parse tree.

By Theorem 2.63, the yield of this new parse tree is $uvvwxxy = uv^2wx^2y$.

By repeated application of this replacement procedure, we can create new parse trees $Pt(S, uv^iwx^iy, k_i)$ for $i \geq 2$.

By Theorem 2.33, $S \overset{*}{\Rightarrow} uv^iwx^iy$ for $i \geq 2$.

If we replace $Pt(V_i, vwx, h_i)$ by $Pt(V_j, w, h_j)$, we obtain a new parse tree $Pt(S, uwy, k_0)$.

Again by Theorem 2.33, $S \overset{*}{\Rightarrow} uwy$.

Or $S \overset{*}{\Rightarrow} uv^0wx^0y$.

When $i = 1$, $z = uvwxy$ and we know $S \overset{*}{\Rightarrow} z$.

Therefore, $S \overset{*}{\Rightarrow} uv^iwx^iy$ for $i \geq 0$.

Therefore, $uv^iwx^iy \in L$ for $i \geq 0$.

This proves Condition (i) is satisfied.

The only three forms of rules in a CFG in CNF are:

$A \longrightarrow BC$ where $A \epsilon V$ and $B, C \epsilon V \backslash \{S\}$

$A \longrightarrow a$ where $a \epsilon \Sigma$

$S \longrightarrow \epsilon$ where $S =$ Start Variable

a and ϵ cannot be the children of V_i because a and ϵ cannot have descendant V_j.

Let $B, C \in V$ be the two children of V_i.

Let $Pt(B, b, h_b)$ & $Pt(C, c, h_c)$ be the largest sub parse tree with yields $b, c \in \Sigma^*$ and roots $B, C \in V$.

Using similar argument as used in the proof Theorem 2.33, we can show that $bc = vwx$.

By T7, V_j is either a descendant of B or C.

If V_j is a descendant of B, $Pt(V_j, w, h_j)$ is a subtree of $Pt(B, b, h_b)$.

By Proposition 2.62, $w \sqsubset b$ and $b = w_1ww_2$ for some w_1, $w_2 \in \Sigma^*$.

Therefore, $vwx = bc = w_1ww_2c$.

$v = w_1$ & $x = w_2c$.

 doi:10.20944/preprints202506.0158.v1

Since C and all its descendants are not S, c cannot be ϵ.
$x \neq \epsilon$.
Therefore, $vx \neq \epsilon$.
If V_j is a descendant of C, with similar argument, we can show that $v \neq \epsilon$ and hence $vx \neq \epsilon$.
In all cases, Condition (ii) is satisfied.
Since V_i is a descendant of V_1, $Pt(V_i, vwx, h_i)$ is a subtree of $Pt(V_1, z_1, h_1)$.
By Proposition 2.62, $vwx \sqsubset z_1$.
$|vwx| \leq |z_1|$.
By Theorem 2.61, $|z_1| \leq 2^{h_1-1}$.
Therefore, $|vwx| \leq 2^{h_1-1}$.
Since $(V_1, V_2, \cdots \cdots V_m, V_{m+1}, a)$ is the longest path from V_1 to a leaf,
$h_1 =$ the length of $(V_1, V_2, \cdots \cdots V_m, V_{m+1}, a) = m + 1$.
Therefore, $|vwx| \leq 2^{m+1-1} = 2^m = p$.
Therefore, Condition (iii) is satisfied.
This completes the proof of Theorem 2.64A.

Theorem 2.64B Pumping Lemma (contra positive form). $\sim(S) \Longrightarrow L$ is not context free where $\sim(S)$ is equivalent to:

$\forall p \geq 1, \exists s \in L$ with $|s| \geq p$ such that whenever $s = uvwxy$, at least one of the conditions (i), (ii), or (iii) cannot be satisfied.

The contra positive form of the Pumping Lemma is used to prove a language is not context free. The general strategy is to find an $s \in L$ with $|s| \geq p$ for any given $p \geq 1$ so that whenever s is broken into $s = uvwxy$, at least one of the conditions of (i), (ii), or (iii) must be false. This can be usually accomplished by showing one of the following:

(1) Condition (i) alone is false.

(2) Condition (iii) $\Longrightarrow \sim$ Condition (i)

(3) (Condition (ii) and Condition (iii)) $\Longrightarrow \sim$ Condition (i).

Example 2.65. Show that $L = \{a^n b^n c^n | n \geq 0\}$ is not CFL.

$\forall p \geq 1$, construct $s = a^p b^p c^p$.
See figure below.

$$\boldsymbol{S = \underbrace{a \cdots \cdots \cdots a}_{p} \underbrace{b \cdots \cdots \cdots b}_{p} \underbrace{c \cdots \cdots \cdots c}_{p}}$$

$s \in L$ and $|s| \geq p$.
Assume $s = uvwxy$.
If Condition (iii) is true, $|vwx| \leq p$.
There are 5 cases for consideration.
(1) $vwx = a^n$ where $n \leq p$
(2) $vwx = a^n b^m$ where $n \leq p$ & $m \leq p$
(3) $vwx = b^n$ where $n \leq p$
(4) $vwx = b^n c^m$ where $n \leq p$ & $m \leq p$
(5) $vwx = c^n$ where $n \leq p$
For case (1), $vwx = a^n \Longrightarrow v^2 w x^2 = a^{n'}$.
If Condition (ii) is true, $vx \neq \epsilon$.
Either $v \neq \epsilon$ or $x \neq \epsilon$.
This means $n' > n$.
$s = uvwxy = ua^n y$ contains the same number of a's, b's and c's.
$uv^2 wx^2 y = ua^{n'} y$ contains more a's than s and therefore has more a's than b's and c's in itself.
Therefore, $uv^2 wx^2 y$ is not in L.

 doi:10.20944/preprints202506.0158.v1

Therefore, (Condition (ii) and Condition (iii)) $\Longrightarrow \sim$ Condition (i).

Similar arguments can be made in cases (3) and (5) to arrive at the same conclusion as in case (1).

For case (2), $s = uvwxy = ua^n b^m y$ contains the same number of $a's$, $b's$ and $c's$.

If Condition (ii) is true, $vx \neq \epsilon$.

Either $v \neq \epsilon$ or $x \neq \epsilon$.

$vwx = a^n b^m \Longrightarrow v^2 w x^2$ will increase the number of $a's$ or the number of $b's$ or both.

uv^2wx^2y will have more $a's$ than $c's$ or more $b's$ than $c's$.

Either way, uv^2wx^2y is not in L.

Therefore, (Condition (ii) and Condition (iii)) $\Longrightarrow \sim$ Condition (i).

Similar arguments can be made in cases (4) to arrive at the same conclusion as in case (2).

Combining all 5 cases, we conclude (Condition (ii) and Condition (iii)) $\Longrightarrow \sim$ Condition (i).

By Theorem 2.64B, L is not context free.

Example 2.66. Show that $L = \{w\#w | w \in \{0,1\}^*\}$ is not CFL.

$\forall p \geq 1$, construct $s = 0^p 1^p \# 0^p 1^p$.

$s \in L$ and $|s| = 4p + 1 > p$.

Assume $s = uvwxy$.

If both Conditions (ii) & (iii) are true, we have the following cases to consider.

(1) vwx is to the left of #

See figure below.

$$\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}\#\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}$$

$$\nvdash \boldsymbol{vwx} \nrightarrow$$

Condition (3) $\Longrightarrow |vwx| \leq p$ which makes it possible for vwx to be contained in $0^p 1^p$.

Since Condition (ii) is true, $vx \neq \epsilon$.

Pumping up to uv^2wx^2y will increase the number of symbols on the left of the # sign while not changing the symbols on the right.

This makes it impossible for uv^2wx^2y to remain in L.

Therefore, (Condition (ii) and Condition (iii)) $\Longrightarrow \sim$ Condition (i).

(2) vwx is to the right of #

Similar argument can be made to lead to same conclusion as in (1).

(3) vwx contains the # sign

(i) w contains # (See figure below.)

$$\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}\#\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}$$

$$\nvdash \boldsymbol{v} \nrightarrow\nvdash \boldsymbol{w} \nrightarrow\nvdash \boldsymbol{x} \nrightarrow$$

Condition (iii) $\Longrightarrow |vwx| \leq p \Longrightarrow \begin{pmatrix} v \text{ contains only } 1's \text{ if } v \neq \epsilon \text{ and} \\ x \text{ contains only } 0's \text{ if } x \neq \epsilon \end{pmatrix}$.

Condition (ii) $\Longrightarrow vx \neq \epsilon \Longrightarrow$ one of v or x is not ϵ.

Pumping down $\Longrightarrow uwy = 0^p 1^i \# 0^j 1^p$.

$v \neq \epsilon \Longrightarrow i < p \Longrightarrow 0^p 1^i \# 0^j 1^p \notin L \Longrightarrow uwy \notin L$.

$x \neq \epsilon \Longrightarrow j < p \Longrightarrow 0^p 1^i \# 0^j 1^p \notin L \Longrightarrow uwy \notin L$.

Therefore, (Condition (ii) and Condition (iii)) $\Longrightarrow \sim$ Condition (i).

(ii) w is to the left of # (See figure below.)

$$\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}\#\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}$$

$$\nvdash \boldsymbol{vw} \nrightarrow\nvdash \boldsymbol{x} \nrightarrow$$

Since vwx contains the # sign, $x \neq \epsilon$.

Therefore, x contains #. (See figure above.)

Pumping down will eliminate the # sign making it impossible for uwy to remain in L.

Therefore, uwy is not in L and Condition (i) cannot be satisfied.

(iii) w is to the right of #

 doi:10.20944/preprints202506.0158.v1

Similar argument will lead to the same conclusion as in case 3(ii) above.
Combining all possible cases, (Condition (ii) and Condition (iii)) $\Longrightarrow \sim$ Condition (i).
By Theorem 2.64B, L is not context free.

Example 2.67. Show that the intersection of two CFLs may not be a CFL.

Let $L_1 = \{a^n b^n c^m | n, m \in \mathbb{N}\}$

$$L_2 = \{a^n b^m c^m | n, m \in \mathbb{N}\}$$

$L_1 \cap L_2 = \{a^n b^n c^n | n \in \mathbb{N}\}$, which is not context free as shown in Example 2.65.
L_1 can be generated by the following CFG rules:
$S \longrightarrow TD$
$T \longrightarrow aTb|\epsilon$
$D \longrightarrow Dc|\epsilon$
L_2 can be generated by the following CFG rules:
$S \longrightarrow AB$
$A \longrightarrow Aa|\epsilon$
$B \longrightarrow bBc|\epsilon$
Therefore, L_1 and L_2 are CFLs.

Example 2.68. Show that Show that $L = \{ww | w \in \{0,1\}^*\}$ is not CFL.

$\forall p \geq 1$, construct $s = 0^p 1^p 0^p 1^p$.
$s \in L$ and $|s| = 4p > p$.
Assume $s = uvwxy$ for some $u, v, w, x, y \in \{0,1\}^*$.
Claim 1. If $i < p$, the strings $0^i 1^p 0^p 1^p, 0^p 1^i 0^p 1^p, 0^p 1^p 0^i 1^p, 0^p 1^p 0^p 1^i$ are not in L.
Assume for contradiction that $0^i 1^p 0^p 1^p \in L$.
$\exists r \in \{0,1\}^*$ such that $0^i 1^p 0^p 1^p = rr$.
Therefore, $|rr| = (i + p) + (p + p)$
$|r| = \frac{(i+p)+(p+p)}{2}$
Since $|r|$ is the arithmetic mean of $(i + p)$ and $(p + p)$ and $(i + p) < (p + p)$,
$|r| > i + p$ and $|r| < p + p$.
Therefore, $|r| \geq i + p + 1$.
The leftmost $i + p + 1$ symbols of $0^i 1^p 0^p 1^p$ form the substring $0^i 1^p 0$.
The leftmost $|r|$ symbols of $0^i 1^p 0^p 1^p$ form the substring r.
Since $|r| \geq i + p + 1$, $0^i 1^p 0 \sqsubset r$.
Therefore, $10 \sqsubset 0^i 1^p 0 \sqsubset r$.
Similarly, the rightmost $2p$ symbols of $0^i 1^p 0^p 1^p$ form the substring $0^p 1^p$ and the rightmost $|r|$ symbols of $0^i 1^p 0^p 1^p$ form the substring r.
Since $|r| < p + p$, $r \sqsubset 0^p 1^p$.
Therefore, $10 \sqsubset r \sqsubset 0^p 1^p$.
This is a contradiction because 10 cannot be a substring of $0^p 1^p$.
Therefore, $0^i 1^p 0^p 1^p \notin L$.
Similar arguments can be made to show $0^p 1^i 0^p 1^p, 0^p 1^p 0^i 1^p, 0^p 1^p 0^p 1^i$ are not in L.
Claim 2.
If at least one of i, j is less than p, the strings $0^i 1^j 0^p 1^p, 0^p 1^i 0^j 1^p, 0^p 1^p 0^i 1^j$ are not in L.
Assume for contradiction $0^i 1^j 0^p 1^p$ is in L.
$0^i 1^j 0^p 1^p = rr$ for some $r \in \{0,1\}^*$.
Therefore, $|rr| = (i + j) + (p + p)$
$|r| = \frac{(i+j)+(p+p)}{2}$
Since $|r|$ is the arithmetic mean of $(i + j)$ and $(p + p)$ and $(i + j) < (p + p)$,
$|r| > i + j$ and $|r| < p + p$.
Therefore, $|r| \geq i + j + 1$.

 doi:10.20944/preprints202506.0158.v1

The leftmost $i+j+1$ symbols of $0^i1^j0^p1^p$ form the substring 0^i1^j0.

The leftmost $|r|$ symbols of $0^i1^j0^p1^p$ form the substring r.

Since $|r| \geq i+j+1$, $0^i1^j0 \sqsubset r$.

Therefore, $10 \sqsubset 0^i1^j0 \sqsubset r$.

Similarly, the rightmost $2p$ symbols of $0^i1^j0^p1^p$ form the substring 0^p1^p and the rightmost $|r|$ symbols of $0^i1^j0^p1^p$ form the substring r.

Since $|r| < p+p$, $r \sqsubset 0^p1^p$.

Since $10 \sqsubset r$ & $r \sqsubset 0^p1^p$, $10 \sqsubset 0^p1^p$.

This is a contradiction because 10 cannot be a substring of 0^p1^p.

Therefore, $0^i1^j0^p1^p \notin L$.

Similar arguments can be made to show that $0^p1^i0^j1^p, 0^p1^p0^i1^j$ are not in L.

Returning to the proof that L is not CFL, we assume both Condition (ii) and Condition (iii) are true.

Since $|vwx| \leq p$, we have 7 cases to consider.

(1) vwx is a substring of the first block of 0^p.

$$\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}$$
$$\nvdash \boldsymbol{vwx} \nvdash$$

(2) vwx is a substring of the first block of 1^p.

$$\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}$$
$$\nvdash \boldsymbol{vwx} \nvdash$$

(3) vwx is a substring of the second block of 0^p.

$$\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}$$
$$\nvdash \boldsymbol{vwx} \nvdash$$

(4) vwx is a substring of the second block of 1^p.

$$\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}$$
$$\nvdash \boldsymbol{vwx} \nvdash$$

(5) vwx straddles the first block of 0^p and the first block of 1^p.

$$\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}$$
$$\nvdash \boldsymbol{vwx} \nvdash$$

(6) vwx straddles the first block of 1^p and the second block of 0^p.

$$\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}$$
$$\nvdash \boldsymbol{vwx} \nvdash$$

(7) vwx straddles the second block of 0^p and the second block of 1^p.

$$\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}\overbrace{\mathbf{00}\cdots\cdots\mathbf{00}}^{p}\overbrace{\mathbf{11}\cdots\cdots\mathbf{11}}^{p}$$
$$\nvdash \boldsymbol{vwx} \nvdash$$

In case (1), v consists of all 0's if $v \neq \epsilon$ and x consists of all 0's if $x \neq \epsilon$.

Pumping down would only affect the first block of 0^p and not the other 3 blocks.

Therefore, $uwy = 0^i1^p0^p1^p$.

Since $vx \neq \epsilon$ by Condition (ii), one of v and x is not ϵ.

Pumping down would reduce the number of 0's in the first block of 0^p.

Therefore, $i < p$.

By Claim 1, $uv^0wx^0y = uwy = 0^i1^p0^p1^p \notin L$.

Therefore, Condition (i) is not satisfied.

For cases (2), (3) and (4), similar arguments can be made to lead to the same conclusion as in (1).

For case (5), $|vwx| \leq p \implies$ pumping down can only affect the first and second blocks of symbols.

We can write $uwy = 0^i1^j0^p1^p$.

 doi:10.20944/preprints202506.0158.v1

Furthermore, the first symbol of vwx is 0 and the last symbol of vwx is 1.
If $v \neq \epsilon$, the first symbol of v is 0.
If $x \neq \epsilon$, the last symbol of x is 1.
Since Condition (ii) is true, $vx \neq \epsilon$.
One of v and x is not ϵ.
Pumping down will either reduce the number of 0's in 0^p or the number of 1's in 1^p.
Therefore, either $i < p$ or $j < p$.
By Claim 2, $uwy = 0^i 1^j 0^p 1^p$ is not in L.
Therefore, Condition (i) is not satisfied.
For cases (6) and (7), similar arguments can be made to lead to the same conclusion as in (5).
Combining all 7 cases, we conclude that
(Condition (ii) and Condition (iii)) $\Longrightarrow \sim$ Condition (i).
Hence, by Theorem 2.64B, L is not context free.

References

Sipser, Michael. *Introduction to the Theory of Computation, Third Edition*.

Dexter C. Kozen. *Automata & Computability*.

John E. Hopcroft, Rajeev Motwani, Jeffrey D Ullman. *Introduction to Automata Theory, Languages, & Computation, Third Edition.*

Seymour Lipschutz, Marc Lars Lipson. *Discrete Mathematics, Second Edition.*

Kwan, Chac. *A Mathematical Approach to the Theory of Finite Automata,* 10.6084/m9.figshare.26232644, *https://figshare.com/articles/journal_contribution/A_Mathematical_Approach_to_the_Theory_of_Finite_Automata_pdf/26232644?file=47541602*