Equivalence of C.F. Grammars and PDAs

We now note that C.F. grammars and PDAs are equivalent characterizations of C.F. languages. I.e. we note

Theorem 1 If $L = L(G)$ is defined by a c.f. grammar G, then L is recognized by some PDA M.

Theorem 2 If a language L is recognized by a PDA M, then $L = L(G)$ is defined by some c.f. grammar G.

In other words, a c.f. grammar can be simulated by a PDA and vice versa. We only prove Theorem 1 here and that too only via an illustrative example.

Theorem 1, Proof by Example

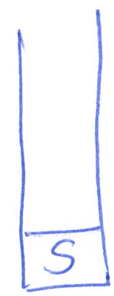
Suppose the grammar is

$$S \to aTbb \mid b$$

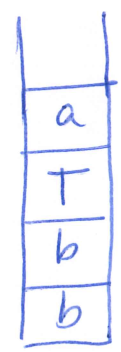
$$T \to Tab \mid \varepsilon .$$

The PDA simulates derivation of a string in $\{a,b\}^*$ as follows:

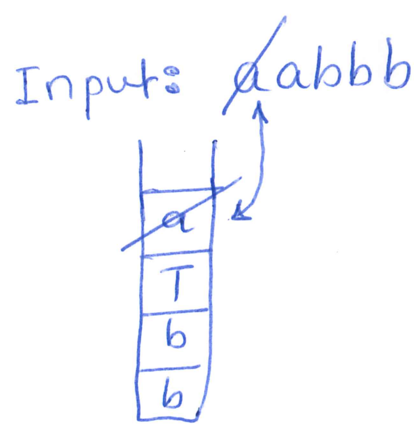
- Push S onto the stack.



→ If the top-stack symbol is a variable, replace it with right hand side of some rule for that variable.



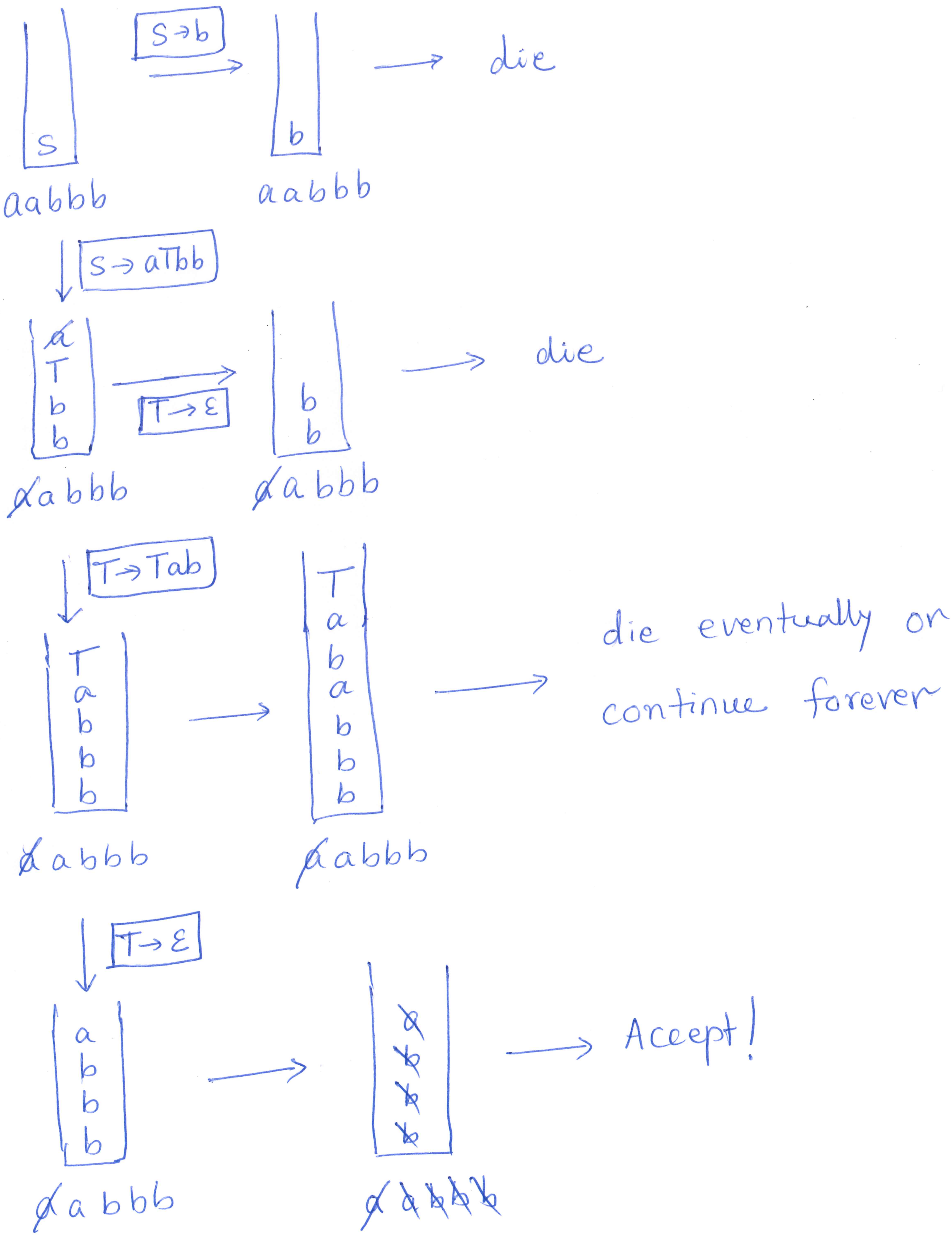
→ If the top-stack symbol is a terminal, match/pop it with the input symbol.



Repeat

Accept if input is exhausted and the stack is empty

Input : aabbb


S
aabbb
$S \to b$
b
aabbb
die
$S \to aTbb$
~~a~~
T
b
b
~~a~~abbb
$T \to \varepsilon$
b
b
~~a~~abbb
die
$T \to Tab$
T
a
b
b
b
~~a~~abbb
T
a
b
a
b
b
b
~~a~~abbb
die eventually or
continue forever
$T \to \varepsilon$
a
b
b
b
~~a~~abbb
~~a~~
~~b~~
~~b~~
~~b~~
~~a~~ ~~a~~ ~~b~~ ~~b~~ ~~b~~
Acept!

The figure shows many possible computation paths of the PDA on input aabbb.

- Since there is a choice of two rules to apply for variable S (and also for T), the PDA is non-deterministic.

- The accepting computation of the PDA corresponds precisely to the leftmost derivation of the input string.

Exercise. Convince yourself of this conclusion and that this simulation works for every c.f. grammar.

We will henceforth take for granted the equivalence of CFGs and PDAs.

Pumping Lemma for Context free Languages

Similar to the pumping lemma for regular languages, we have a pumping lemma for c.f. languages, used to show that certain languages are not context free.

Lemma (Pumping Lemma for CFLs)

Let L be a C.F.L. There exists an integer p (= pumping length) such that the following holds.

Every $s \in L$, $|s| \geq p$ can be written as

$$s = uvxyz \quad \text{such that}$$

1. $uv^ixy^iz \in L \quad \forall i \geq 0$
2. $|vy| \geq 1$
3. $|vxy| \leq p$.

Proof Let G be a c.f. grammar for L.

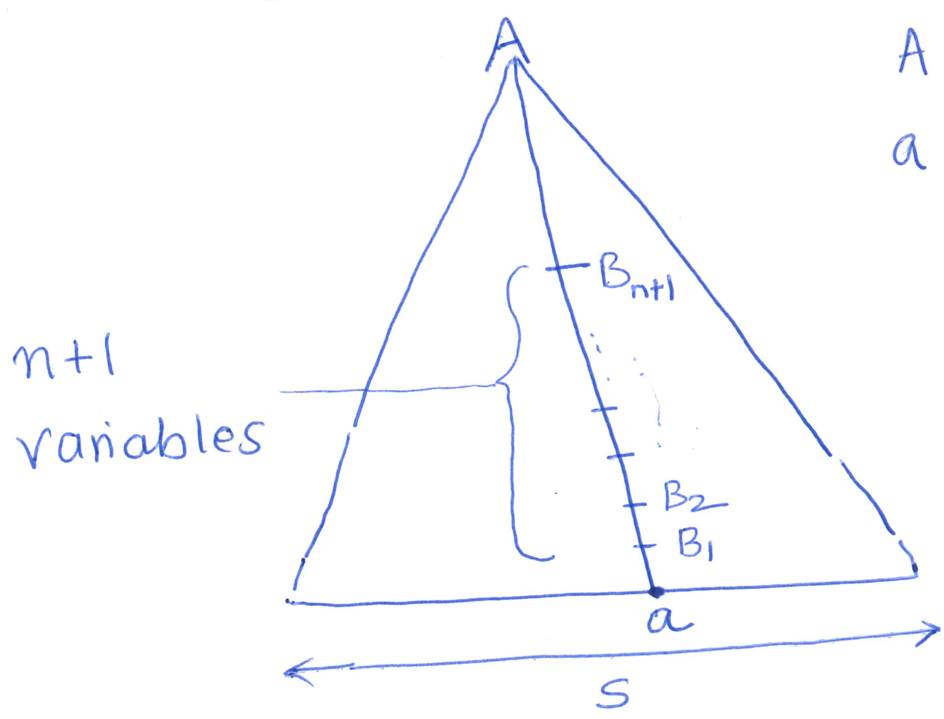
Let n = # variables in G.

b = Maximum length of right hand side of a rule in G.

Let $p = b^{n+2}$.

Consider any string $s \in L$, $|s| \geq p$ and consider the parse tree for s that has the minimum number of nodes.

The tree is "b-ary" and since $|s| \geq p = b^{n+2}$, has height at least $n+2$.

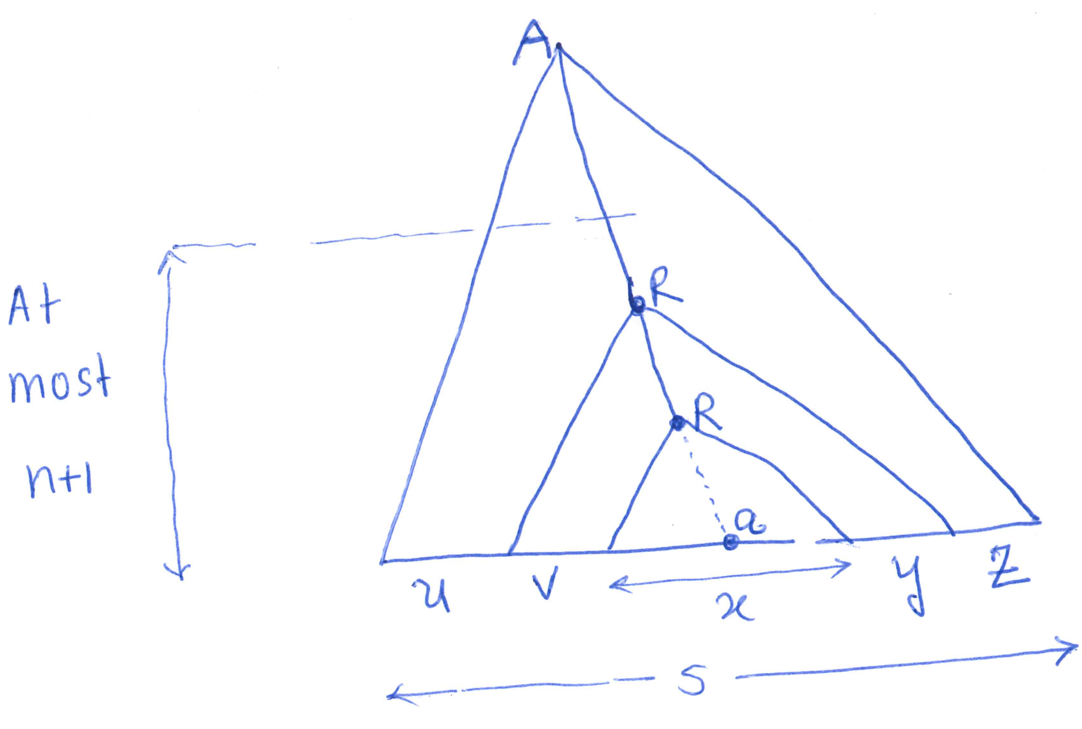


A : Start variable

a : terminal / leaf at depth at least $n+2$.

Let a be a terminal symbol that occurs as a leaf of this tree and has depth at least $n+2$.

Consider the path upwards from a to the root of the tree (= start variable) A. Let $B_1, B_2, \ldots, B_{n+1}$ be the successive variables (upwards) on this path as shown in the picture. Since the grammar has only n variables, by Piegeon-Hole Principle, for some $1 \leq i < j \leq n+1$, $B_i = B_j = R$ is the same variable. We can redraw as:



Let $S = uvxyz$ be the splitting as shown. Clearly, we have derivations

$$A \overset{*}{\Rightarrow} uRz$$
$$R \overset{*}{\Rightarrow} vRy \mid x$$

We now prove the three conclusions of the lemma.

① $\underline{i=0}$ $\quad A \overset{*}{\Rightarrow} uRz \overset{*}{\Rightarrow} uxz$

Hence $uxz \in L$.

$\underline{i \geq 1}$ $\quad A \overset{*}{\Rightarrow} uRz \overset{*}{\Rightarrow} uv\underline{R}yz$

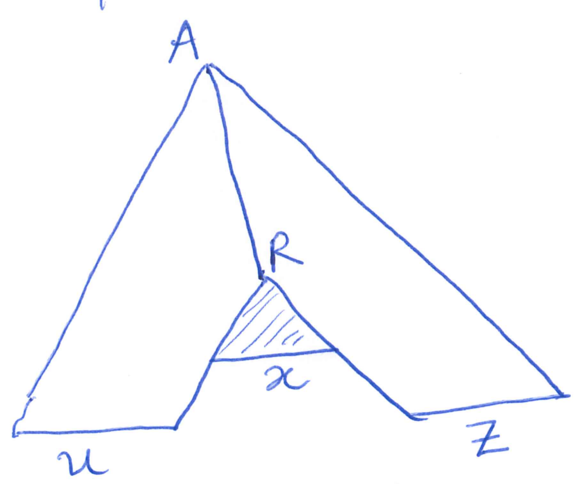
$$\overset{*}{\Rightarrow} uv \cdot \underline{vRy}\, yz$$
$$\vdots$$
$$\overset{*}{\Rightarrow} uv^iRy^iz$$
$$\overset{*}{\Rightarrow} uv^ixy^iz$$

Hence $uv^ixy^iz \in L$.

③ $|vxy| \leq p$. This is because vxy is generated by the parse tree rooted at variable R (that appears "above") and having height at most $n+1$. Since the tree is "b-ary", we have

$$|vxy| \leq b^{n+1} \leq b^{n+2} = p.$$

③ $|vy| \geq 1$. Suppose on the contrary that vy equals the empty string ε. Then $s = uxz$ is also generated by the parse tree



This parse tree has fewer nodes than the one before, contradicting the minimality of the tree before! ▨

Now we'll apply the pumping lemma to show that:

<u>Claim</u> $L = \{a^n b^n c^n \mid n \geq 0\}$ is not context free.

<u>Proof</u> Suppose on the contrary that L is context free. Let p be the pumping length and let $s = a^{5p} b^{5p} c^{5p}$

$$s = aaa \cdots a \cdot bbb \cdots b\, ccc \cdots c.$$

Since $s \in L$ and $|s| \geq p$, by the pumping lemma, s can be written as

$$s = uvxyz, \quad uv^i xy^i z \in L \ \forall i \geq 0$$
$$|vy| \geq 1$$
$$|vxy| \leq p.$$

Since $|vxy| \leq p$, the sub-string vxy can be <u>either</u> contained in the prefix $a^{5p} b^{5p}$

<u>or</u> contained in the suffix $b^{5p} c^{5p}$.

<u>case 1</u> vxy is contained in $a^{5p} b^{5p}$.

Since $|vy| \geq 1$, vy contains 'a' or 'b' or both.

Thus the string $uv^i xy^i z$ has more 'a's or 'b's than 'c's, for $i \geq 2$.

$$\underbrace{aaa\cdots}_{u}\underbrace{a\,bbb\cdots b}_{vxy}\underbrace{ccc\cdots c}_{z}$$

Hence for $i \geq 2$, $uv^i xy^i z \notin L$, contradicting the conclusion of the pumping lemma.

<u>Case 2</u> vxy is contained in $b^{5p} c^{5p}$.

$$\underbrace{aaa\cdots a}_{u}\underbrace{bbb\cdots b\,ccc}_{vxy}\underbrace{\cdots c}_{z}$$

The proof is similar. Now $uv^2 xy^2 z$ contains more 'b's or 'c's than 'a's.

The contradiction in both cases shows that L is not context free.

<u>Note</u> $L = \{a^n b^n c^n \mid n \geq 0\}$ is

$L = L_1 \cap L_2$ where

$L_1 = \{a^n b^n c^m \mid n, m \geq 0\}$

$L_2 = \{a^m b^n c^n \mid n, m \geq 0\}$.

It is easily seen that L_1, L_2 are context free. Thus intersection of context free languages need not be context free.

As another application, we show:

<u>Claim</u> $L = \{w \# w \mid w \in \{0,1\}^*\}$ $\Sigma = \{0,1,\#\}$

is not context free.

<u>Proof</u> Suppose L is context free and p

be the pumping length. Consider the string $s = 0^{5p} 1^{5p} \# 0^{5p} 1^{5p}$.

Since $s \in L$, $|s| \geq p$, by Pumping Lemma,

$s = uvxyz$ s.t. - $uv^ixy^iz \in L \ \forall i \geq 0$

- $|vy| \geq 1$

- $|vxy| \leq p$.

Since $|vxy| \leq p$ there are 4 cases.

<u>case 1</u> $$00 \cdots \underbrace{0\,11 \cdots 1}_{vxy} \# 00 \cdots 0\,11 \cdots 1.$$

I.e. vxy occurs before the $\#$ symbol.

<u>case 2</u> $$00 \cdots 0\,11 \cdots 1 \# \underbrace{00 \cdots 0\,1}_{vxy}1 \cdots 1$$

I.e. vxy occurs after the $\#$ symbol.

Case 3

$$00\cdots 0\underbrace{11\cdots 1}_{v}\underbrace{\#00\cdots}_{x}\underbrace{011\cdots}_{y}1$$

v only has '1's, x contains # symbol,
y only has '0's.

case 4 # is contained in v (or y).

$$00\cdots 011\cdots\underbrace{1\#00\cdots}_{v}\underbrace{0}_{x}\underbrace{1}_{y}1\cdots 1$$

In all the four cases $uv^2xy^2z \notin L$
contradicting the conclusion of the pumping
Lemma. The reason is that uv^2xy^2z

In case 1 has longer length before #
than after.

In case 2 has longer length after #
than before.

In case 3 has the form

$$0^{5p} 1^{5p+a} \# 0^{5p+b} 1^{5p}$$

with $(a+b) \geq 1$. Clearly this is not of the form $w \# w$.

In case 4 has two # symbols.

The contradiction shows that L is not context free.

$\blacksquare$

Exercise $L = \{w\#w \mid w \in \{0,1\}^*\}$ as above, observe that $\bar{L} = L_1 \cup L_2 \cup L_3$ where

$L_1 = \{0,1,\#\}^* \# \{0,1,\#\}^* \# \{0,1,\#\}^*$ i.e. two or more #

$L_2 = \{0,1\}^*$ i.e. no # symbol

$L_3 = \{w \# w' \mid w, w' \in \{0,1\}^*, w \neq w'\}$

Show that L_3 is context free (use PDA!) and hence $\bar{L}$ is context free. (but L is not as shown). So complement CFL need not be CFL.