

Parse Trees, Ambiguity in CFG, Pumping Lemma for CFLs

Chapter 3: Grammars

Prof. Riddhi Atulkumar Mehta
Assistant Professor
Department of Computer Science and
Engineering

Content

1. Introduction.....	1
2. What is a Parse Tree?.....	2
3. Parse Tree Example.....	3
4. Ambiguity in CFG.....	4
5. Example of Ambiguity	5
6. Removing Ambiguity.....	6
7. Pumping Lemma for CFLs.....	7
8. How to Use Pumping Lemma.....	8

Introduction

This section explores:

- Structural representation of derivations using parse trees
- The concept of ambiguity in context-free grammars
- The pumping lemma for proving that certain languages are not context-free

What is a Parse Tree?

- A parse tree visually represents how a string is derived from a CFG.
- Structure:
- Root: Start symbol
- Internal nodes: Non-terminals (variables)
- Leaves: Terminals or ϵ (empty string)
- Reading the leaves from left to right gives the derived string.

Parse Tree Example

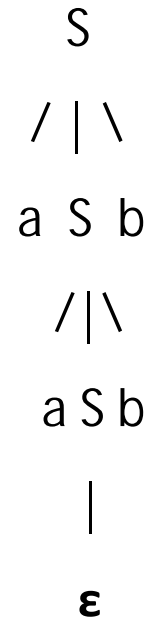
Grammar:

$S \rightarrow aSb \mid \epsilon$

String: aabb

Derivation:

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aab \Rightarrow aabb$



Each step replaces a non-terminal with its production.

Ambiguity in CFG

- A CFG is ambiguous if there exists at least one string that has:
 - More than one parse tree, or
 - More than one leftmost or rightmost derivation
- Ambiguity leads to confusion in interpretation, especially in compilers.

Example of Ambiguity

Grammar:

$E \rightarrow E + E \mid E * E \mid$

String: $\text{id} + \text{id} * \text{id}$

- Two parse trees:
 1. Interpret as $(\text{id} + \text{id}) * \text{id}$
 2. Interpret as $\text{id} + (\text{id} * \text{id})$
- Grammar does not enforce precedence or associativity.

Removing Ambiguity

- To remove ambiguity:
- Rewrite the grammar to enforce precedence (e.g., $*$ $>$ $+$)
- Separate productions by level of precedence

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow \text{id}$$

Pumping Lemma for CFLs

- Used to prove that a language is not context-free
- For every context-free language L , there exists a constant p (pumping length) such that:

Then there is a pumping length n such that any string $w \in L$ of length $\geq n$ can be written as follows –

$$|w| \geq n$$

We can break w into 5 strings, $w=uvxyz$, such as the ones given below

- $|vxy| \geq n$
- $|vy| \neq \epsilon$
- For all $k \geq 0$, the string $uv^kxy^kz \in L$

How to Use Pumping Lemma

- Assume the language is context-free.
- Choose a string s of length $\geq p$.
- Try all valid decompositions $s=uvwxy$
- Show that for some i , $uviwx^iy \notin L$ creating a contradiction.

Parul[®]
University

NAAC
GRADE **A++**



<https://paruluniversity.ac.in/>

