Recitation - Week 6
Module 8

CSE 355: Theory of Computation

Today’s Agenda

1. Non-Context-Free Languages
2. Pumping Lemma for Context-Free Languages
3. Strategy
4. Examples

Introduction to Non-Context-Free Languages

Definition: A **Non-Context-Free Language** is a language that cannot be generated by a context-free grammar. Such languages require more powerful computational models than those provided by context-free grammars.

Examples: Some languages have structures or dependencies that context-free grammars cannot handle, such as matching multiple levels of nested constructs or specific mirroring patterns (a^n * b^n * c^n, ss)

Pumping Lemma for Context-Free Languages

A technique to prove that certain languages are **not context-free**. It provides a condition that all context-free languages must satisfy. If a language fails this condition, it is **not context-free**.

Lemma Statement:

- For any context-free language L, there exists a constant p (pumping length) such that any string $s \in L$ with $|s| \geq p$ can be split into five parts
- $s=uvxyz$ such that:
  1. $|vxy| \leq p$
  2. $|vy| > 0$
  3. $u\, v^n\, x\, y^n\, z \in L$ for all $n \geq 0$

Strategy to Prove a Language is Non-Context-Free

- **Assume the Language is Context-Free**: Assume the language satisfies the conditions of the pumping lemma.
- **Pick a Specific String**: Choose a string s from the language that is long enough (at least p symbols).
- **Show the Contradiction**: Try to break the string into parts uvxyz and show that pumping (increasing or decreasing the number of v and y) leads to a string that is **not** in the language, thereby contradicting the assumption.

Problem 1

The Language A = {a^n * b^m * c^n | n≥0,m≥0}. Decide if the language is context free or not and prove your claim.

Solution

We define a CFG $G=(V,\Sigma,P,S)$, where:

- **Non-terminals**: S,B
- **Terminals**: a,b,c
- **Production rules**:
    1. $S \rightarrow aSc$
       (Generates matching numbers of as and cs)
    2. $S \rightarrow B$
       (Transitions to generating the middle segment of bs)
    3. $B \rightarrow bB \mid \epsilon$
       (Generates any number of bs, including none)

Problem 2

The language $B = \{a^{n} * b^{n^{2}} \mid n \in N\}$. Decide if the language is regular or not and prove your claim.

Solution

Assume B is context-free. Let p be the pumping length given by the Pumping Lemma for context-free languages. Choose a string $s = a^p b^{p^2}$ from the language B where: The number of `a`s is p, and The number of `b`s is p^2.

- Case 1: vxy lies entirely within the `a`s (the prefix a^p). In this case, pumping v and y would result in more or fewer `a`s than p, while the number of `b`s remains fixed at p^2 . This breaks the relationship between the number of `a`s and the number of `b`s, violating the structure of the language.

- Case 2: vxy lies entirely within the `b`s (the suffix b^{p^2}). In this case, pumping v and y would result in more or fewer `b`s than p^2, while the number of `a`s remains fixed at p . Again, this breaks the relationship between the number of `a`s and the number of `b`s, violating the structure of the language.

- Case 3: vxy spans across both the `a`s and `b`s. In this case, pumping v and y would change both the number of `a`s and `b`s, but it would destroy the required relationship b^{p^2}. For example, increasing or decreasing the number of `a`s would also disrupt the precise p^2 `b`s relationship, resulting in a string that no longer belongs to B.

Conclusion: In all possible cases, pumping v and y results in a string that does not belong to the language B. Therefore, the pumping lemma fails for the language B.

Problem 3

The language $C = \{ a^n b a^{2n} b a^{3n} \mid n => 0 \}$. Decide if the language is regular or not and prove your claim.

Solution

Assume C is context-free. Let p be the pumping length given by the Pumping Lemma for context-free languages. Choose a string s = a^p b a^{2p} b a^{3p} from the language C, where: The number of `a`s before the first `b` is p, The number of `a`s between the two `b`s is 2p, The number of `a`s after the second `b` is 3p .

- Case 1: vxy lies entirely within the first block of `a`s (the prefix a^p). In this case, pumping v and y will increase or decrease the number of `a`s in this block. This will disrupt the relationship between the three blocks of `a`s, as the number of `a`s in the first block will no longer be n, breaking the relationship n : 2n : 3n .

- Case 2: vxy lies entirely within the second block of `a`s (the middle section a^{2p}). In this case, pumping v and y will increase or decrease the number of `a`s in the second block, destroying the required ratio n: 2n :3n .

- Case 3: vxy lies entirely within the third block of `a`s (the suffix a^{3p}). In this case, pumping v and y will alter the number of `a`s in the third block, which will again disrupt the required ratio n : 2n : 3n .

- Case 4: vxy spans across the boundaries between different blocks. In this case, pumping v and y will change the distribution of `a`s between the blocks, which will disrupt the precise ratio required between the three blocks of `a`s.

6. Conclusion: In all possible cases, pumping v and y results in a string that no longer satisfies the ratio n : 2n : 3n . Hence, the resulting string is not in the language C, and the pumping lemma fails for C.

Questions!!