Parul® University

NAAC A++

Context-Sensitive Languages: Context-Sensitive Grammars (CSG) and Languages
Chapter 3: Grammars

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

Content

1. Context-Sensitive Languages (CSLs)............. 1
2. Properties of Context-Sensitive Languages... 2
3. Example..3
4. Differences with Other Grammars 4

INDEX

Context-Sensitive Languages (CSLs)

Before we enter context-sensitive languages, it's essential to understand the context-free languages (CFLs), as they form the foundation for our discussion.

What is Context-Free Grammar?

A context-free language is generated by a context-free grammar (CFG). In a CFG, production rules have the form: A → X, Where –

- A is a variable (non-terminal)
- X is any string of terminals or variables

Context-Sensitive Languages (CSLs)

- CSLs are languages generated by Context-Sensitive Grammars
- Can be recognized by Linear Bounded Automata (LBA)
- Proper superset of CFLs: CFL $\subset$ CSL
- Used to describe more complex syntax rules, like type agreement in natural language
- The context-sensitive languages extend the concept of CFLs by allowing production rules to depend on the context in which variables appear.
- This seemingly small change leads to a significant increase in expressive power.
- A context-sensitive grammar has production rules of the form: $\alpha A\beta \rightarrow \alpha X\beta$, where –
- **α**, **β** are strings of terminals and/or variables (can be empty)
- A is a variable
- X is a non-empty string of terminals or variables

Context-Sensitive Languages (CSLs)

- Context Sensitive Grammar
- A context-sensitive grammar has production rules of the form: $\alpha A\beta \rightarrow \alpha X\beta$, where –
- $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ are strings of terminals and/or variables (can be empty)
- A is a variable
- X is a non-empty string of terminals or variables

Properties of Context-Sensitive Languages

Listed below are some of the important properties of context-sensitive languages –

- Context Preservation – The production process maintains the same context (α and β) on both sides, ensuring that the replacement of A with X only occurs within the defined context.
- Non-Contracting – The grammar's property X cannot be empty, ensuring it doesn't reduce string length during derivation. However, the start variable S can generate an empty string if it's part of the language.
- Increased Expressive Power – CSLs can describe patterns that CFLs cannot, such as matching multiple repeated substrings.

Example

Consider the following CSG.

S → abc/aAbc

Ab → bA

Ac → Bbcc

bB → Bb

aB → aa/aaA

What is the language generated by this grammar?

Example

Solution:

S → aAbc

→ abAc

→ abBbcc

→ aBbbcc

→ aaAbbcc

→ aabAbcc

→ aabbAcc

→ aabbBbccc

→ aabBbbccc

Differences with Other Grammars

Grammar Type	Example Language	Automaton
Regular	a*	Finite Automaton
Context-Free (CFG)	a^nb^n	Pushdown Automaton
Context-Sensitive	A^nb^nc^n	Linear Bounded Automaton
Recursively Enumerable	All computable languages	Turing Machine

Parul® University

NAAC GRADE A++

https://paruluniversity.ac.in/