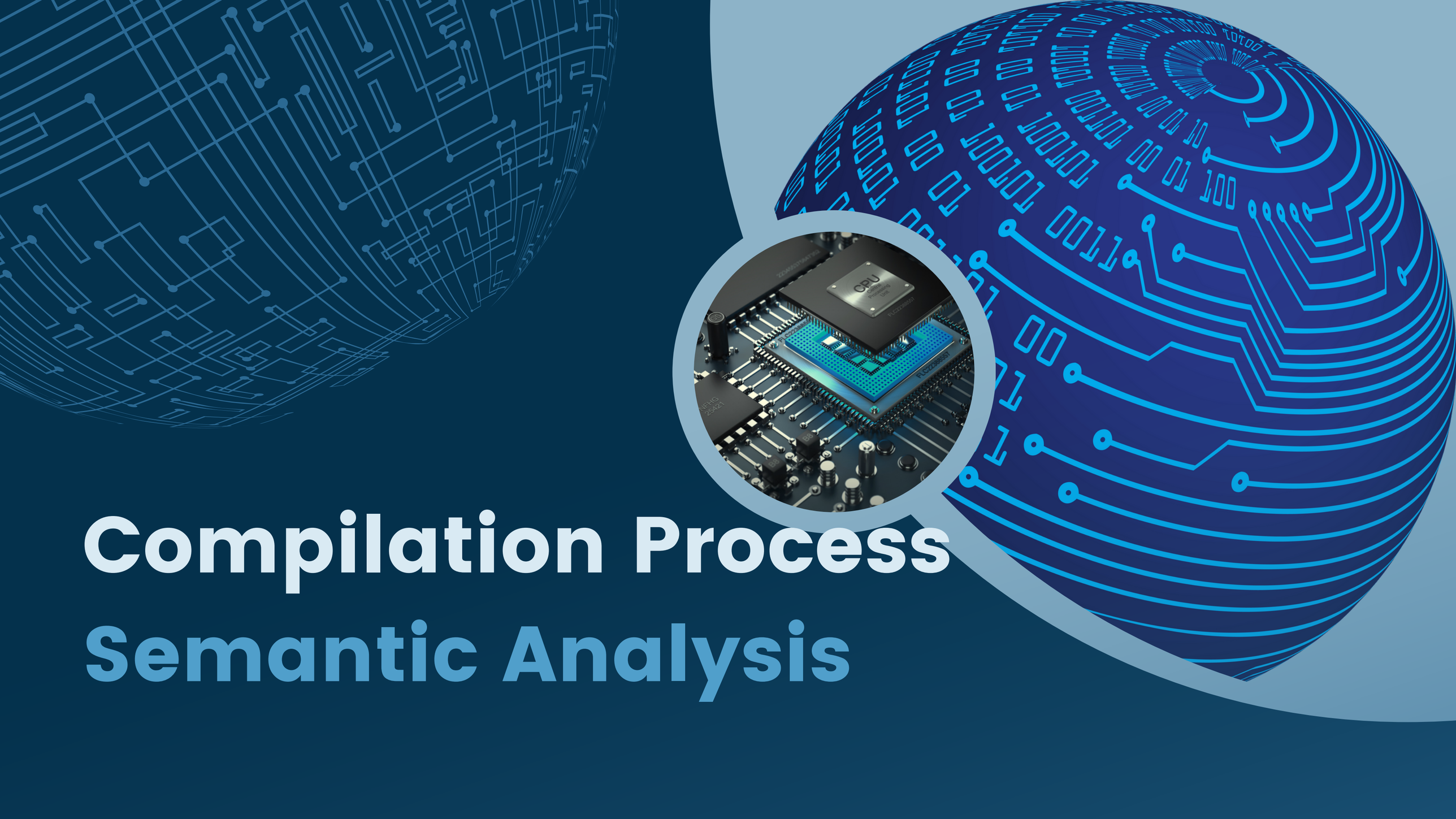

Compilation Process
Semantic Analysis

Table Of Content

Introduction

Semantic analysis is the process of analyzing and validating the meaning of a program's code. It ensures that the program code meets the requirements and constraints of the programming language in use.

It is a critical component of the compilation process, as it enables compilers to generate efficient and correct code. Without semantic analysis, compilers would not be able to identify and report errors or optimize code.

Compilation process

The compilation process involves several phases, including lexical analysis, syntax analysis, semantic analysis, code generation, and code optimization. Semantic analysis occurs in the middle phase of the compilation process, after lexical and syntax analysis, and before code generation and optimization.

01 Lexical analysis

02 Syntax analysis

03 Semantic analysis

Syntax vs Semantic

Syntax refers to the structure and grammar of a programming language. Semantics refers to the meaning and behavior of code.

Syntax analysis

Syntax analysis can identify errors related to the structure of code, but it cannot detect errors related to the meaning and behavior of code.

Semantic analysis

Semantics is crucial in programming languages because it defines the meaning and behavior of code.

Semantic analysis

Semantic analysis in programming involves steps like type checking, name resolution, scope checking, and control flow analysis. Its purpose is to identify errors and optimize code. Examples of semantic analysis include type and scope checking.

Other Applications

NLP Semantic parsing, is a technique used in natural language processing (NLP) to analyze and understand the meaning of text or speech.

Thank You