Chapter 1: Introduction

1 Compiler Definition

What's a Compiler?

A compiler is a software program that takes a piece of code as input, analyzes it for errors, and generates output in the form of messages or logs. Additionally, it produces an executable file that can be run on a computer.

2 Steps of Compilation

Steps

The process of compilation is done following these steps chronologically:

1. Lexical Analysis
2. Syntactic Analysis (Parsing)
3. Semantic Analysis
   - Generation of Intermediate Code
   - Syntax-Directed Translation
4. Generation of Object Code

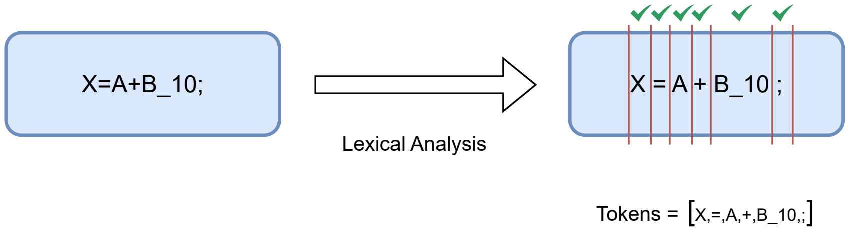
2.1 Lexical Analysis

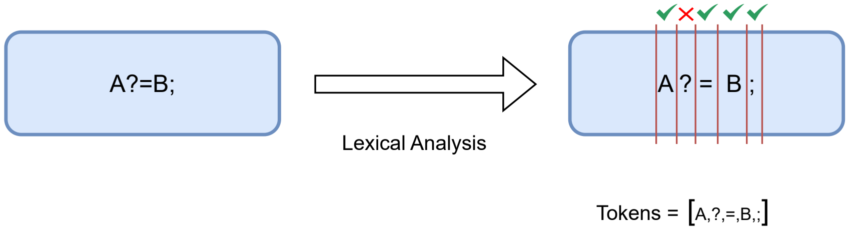
Lexical Analysis

Lexical analysis consists of dividing the entire code into lexical entities (tokens) and evaluating each of them independently. Tokens can be divided based on:

- Spaces
- Logical operators (AND, OR, NOT, >=, >, <=, <, =)
- Arithmetic operators (+, -, *, /)
- Separators ([ ], { }, ( ))

Example





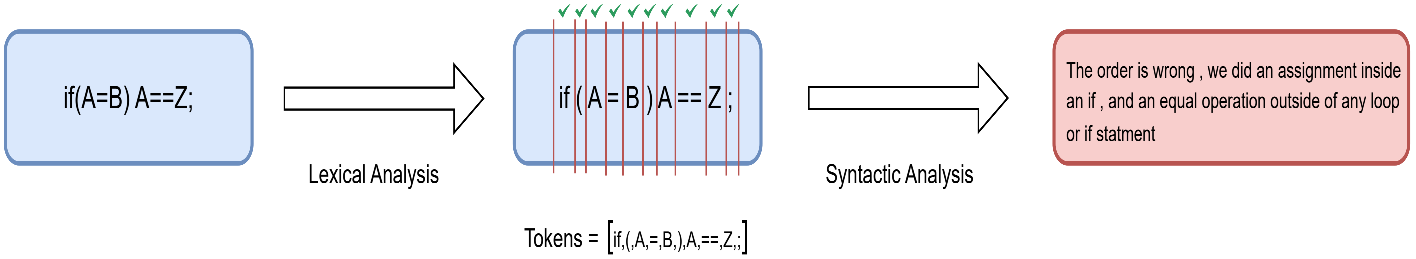
2.2 Syntactic Analysis (Parsing)

Syntactic Analysis

Syntactic analysis checks if the order of tokens is correct and adheres to the required format (using automata or regular expressions). There are two methods:

- Descending methods
- Ascending methods

Example


if(A=B) A==Z;
Lexical Analysis
✓✓✓✓✓✓✓ ✓ ✓✓
if (A = B) A == Z ;
Tokens = [if,(,A,=,B,),A,==,Z,;]
Syntactic Analysis
The order is wrong , we did an assignment inside an if , and an equal operation outside of any loop or if statment

Note

If a code is lexically correct, it does not guarantee that it will be syntactically correct. However, the opposite is true:

Correct Syntactically → Correct Lexically
Correct Lexically ↛ Correct Syntactically

2.3 Semantic Analysis

Semantic Analysis

This phase ensures the code is logically correct and adheres to the language's rules. It verifies **type mismatches, variable declarations, function calls, and more**.

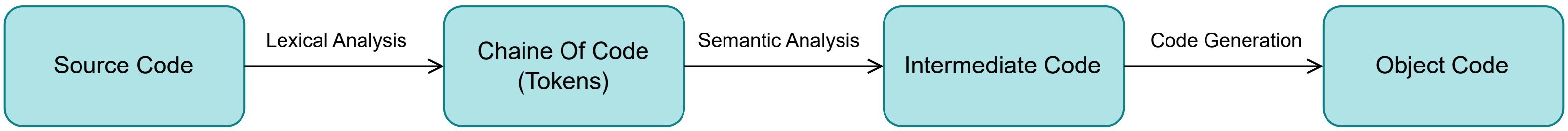
- **Generation of Intermediate Code**: Produces a simplified, platform-independent representation of the program. This step is crucial for optimizing the code and converting it into machine code later.
  - **Quadruple Form**
  - **Postfix and Prefix Forms**
- **Syntax-Directed Translation**: Generates intermediate representations, ensuring that the semantics (meaning) of the program guide the creation of subsequent steps.

2.4 Object Code Generation

Object Code Generation

The final step of compilation, where the object code is generated to execute the program. This is the only step in the compilation process that we will not cover in this course.

3 Compilation Steps Diagram


Source Code
Lexical Analysis
Chaine Of Code (Tokens)
Semantic Analysis
Intermediate Code
Code Generation
Object Code