phiC

Phase III - Semantic Analysis

Team Members and Roles

Srinith Dasari - Language Guru

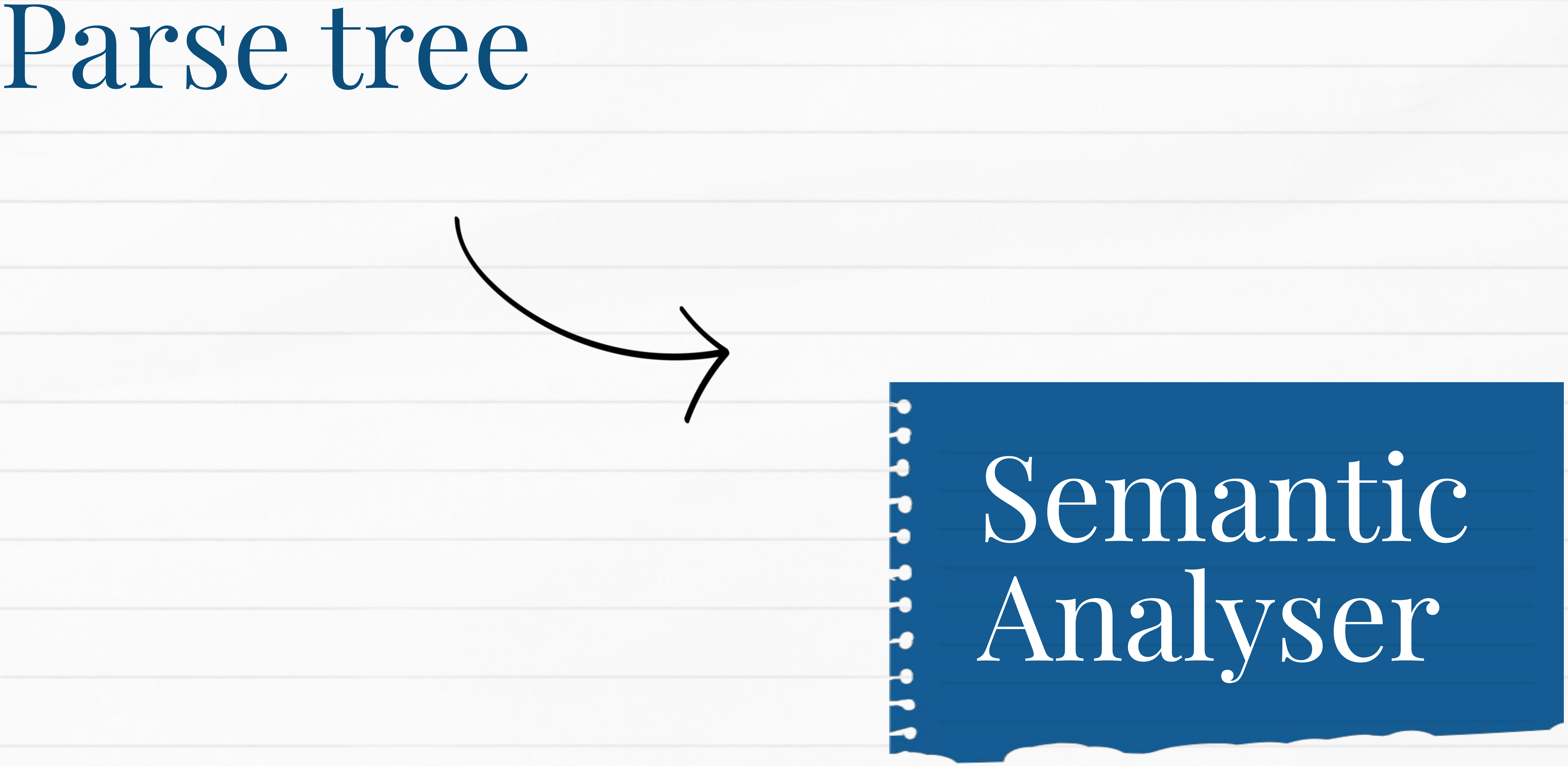
Pranav Varma Pericherla - System Integrator

Akhila Kumbha - System Architect

Santoshi Gayatri Mavuru - Project Manager

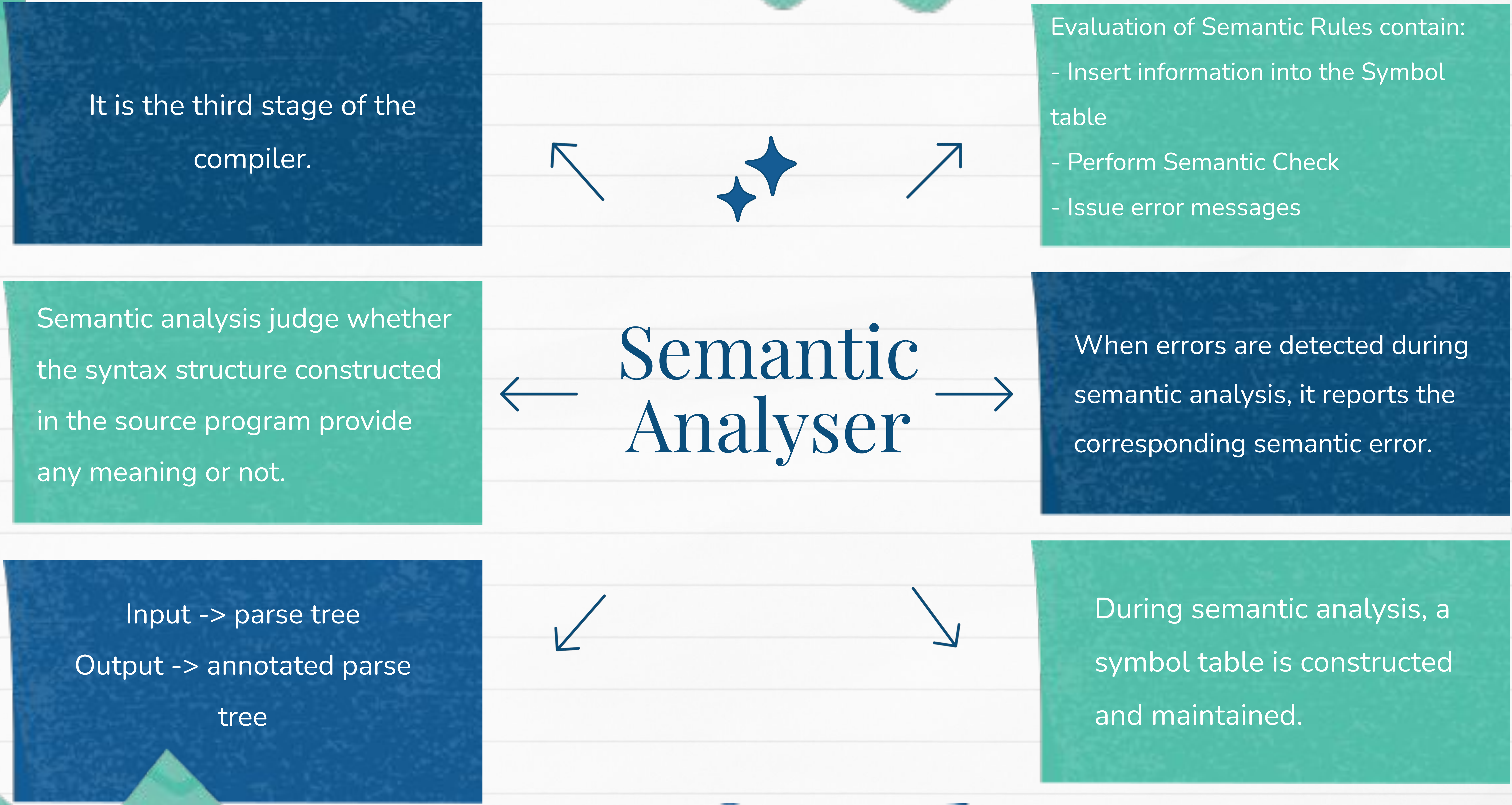
OBJECTIVES AND GOALS

- The main objective of semantic analysis is ensure that a program is not only syntactically correct but also semantically meaningful.
- To identify and address the errors that may lead to unexpected behavior or runtime issues in the program.



Parse tree

Semantic Analyser



Semantic Analyser

It is the third stage of the compiler.

Semantic analysis judge whether the syntax structure constructed in the source program provide any meaning or not.

Input -> parse tree
Output -> annotated parse tree

Evaluation of Semantic Rules contain:

- Insert information into the Symbol table
- Perform Semantic Check
- Issue error messages

When errors are detected during semantic analysis, it reports the corresponding semantic error.

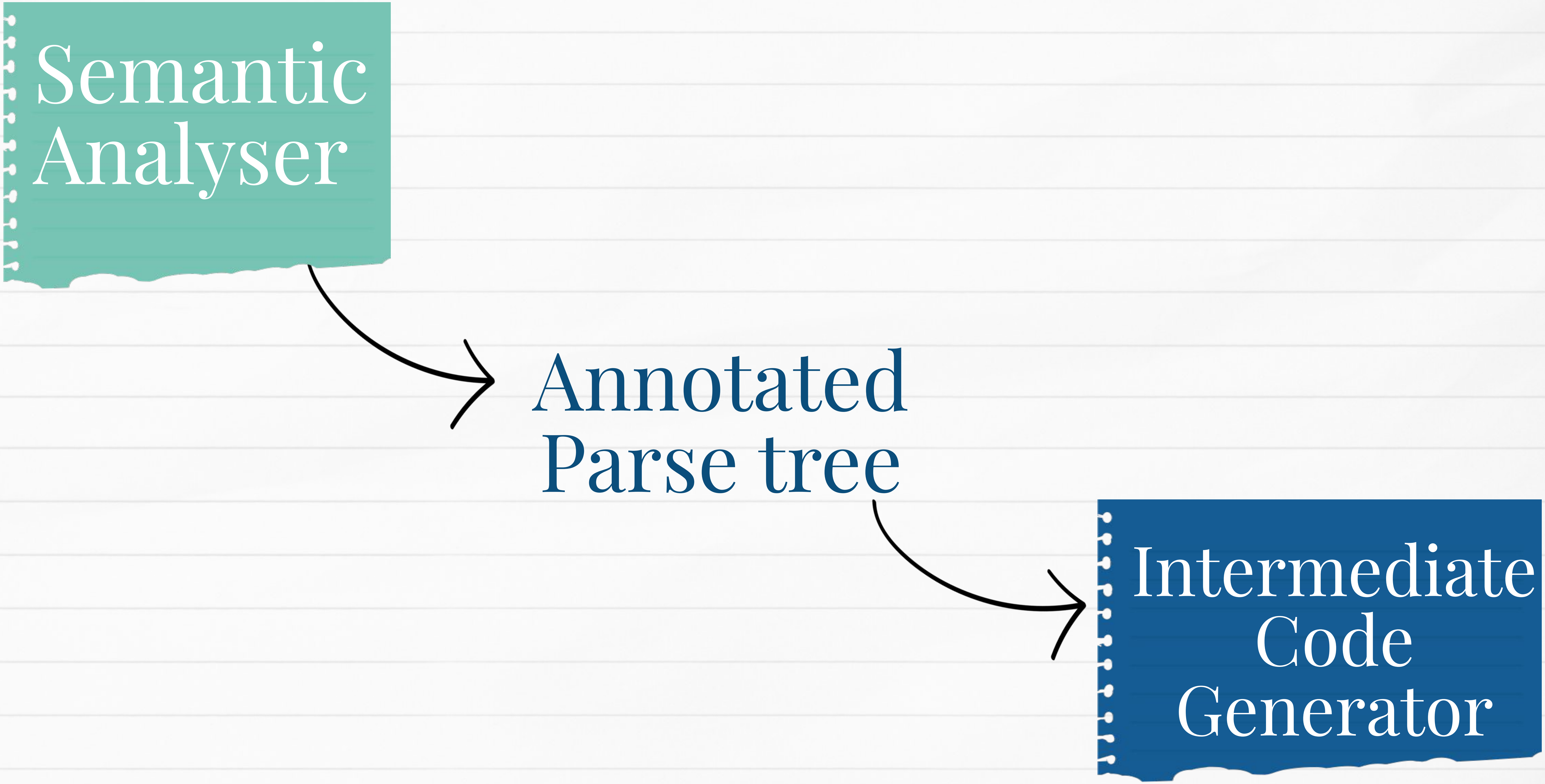
During semantic analysis, a symbol table is constructed and maintained.

Symbol Table

- A symbol table is a data structure used by a compiler to keep track of various information about identifiers (such as variables and functions).
- This information is then used in subsequent compilation phases like type checking and code generation.
- We defined two symbol tables
  - **Global Symbol table Attributes** - Name, Datatype, Type, Scope
  - **Function Symbol table Attributes** - Name, Datatype, No.of parameters, Parameters list, Variable list, Scope.
    - **Parameter List Attributes** - Name, Datatype
    - **Variable List Attributes** - Name, Datatype, Scope

Typical Semantic Errors

- **Multiple declarations:** A variable should be declared (in the same scope) at most once.
- **Undeclared variable:** A variable should not be used before being declared.
- **Type mismatch:** Type of the LHS of an assignment should match the type of the RHS.
- **Incompatible types for operation:** Operation is applied to operands of types that are not compatible.
- **Array Bounds Checking:** Ensuring that array indices are within the valid range to prevent errors at runtime.
- **Wrong arguments:** Methods should be called with the right number and types of arguments.
- **Return type:** Type of the return value of the method should match.
- **Standard Libraries:** Input parameters should be of particular datatypes only.



Semantic Analyser

Annotated Parse tree

Intermediate Code Generator

THANK YOU

Objectives:.
the main objective of semantic analysis is ensure that a program is not only syntactically correct but also semantically meaningful
To identify and address the errors that may lead to unexpected behavior or runtime issues in the program.

Semantic analysis includes Static semantic and dynamic semantic checks.

Semantics of a language provide meaning to its constructs like tokens and syntax structure. Semantic analysis judge whether the syntax structure constructed in the source program provide any meaning or not.

output of syntax analysis which is a parse tree is the input of semantic analysis which is also a parse tree with some additional attributes called annotated parse tree.

Evaluation of Semantic Rules may:
-Generate Code;
- Insert information into the Symbol Table:
- Perform Semantic Check;
- Issue error messages; etc.

Typical semantic errors:
Multiple declarations: A variable should be declared (in the same scope) at most once.
Undeclared variable: A variable should not be used before being declared.
Type mismatch: Type of the LHS of an assignment should match the type of the RHS.
Incompatible types for operation: Operation is applied to operands of types that are not compatible.

A symbol table is a data structure used by a compiler to keep track of various information about identifiers (such as variables and functions).
This information is then used in subsequent compilation phases like type checking and code generation.

A symbol table is a data structure that tracks the current bindings of identifiers.

Variable and procedure names
Literal constants and strings

- Textual name
- Data type
- Declaring procedure
- Lexical level of declaration
- If array, number and size of dimensions
- If procedure, number and type of parameters