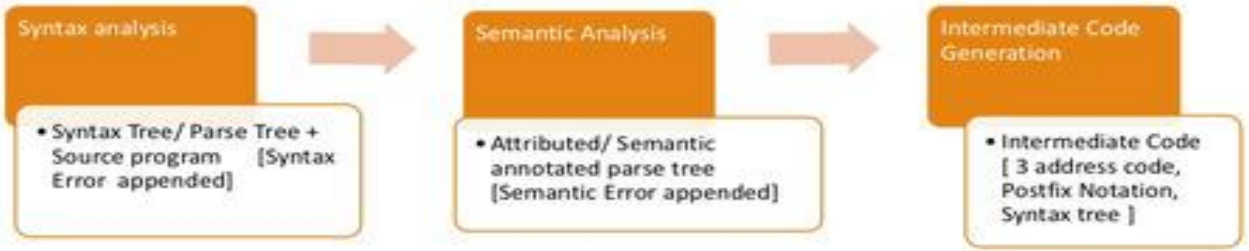
Semantic Analyser

By Team 8

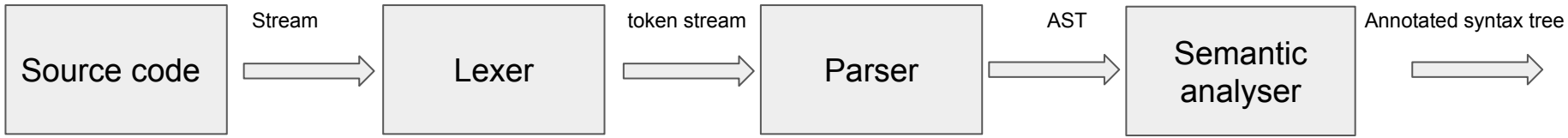
Manoj Gayala - CS19BTECH11011
Nisha M - CS19BTECH11012
Sharanya Gupta - CS19bTECH11020
Sravanthi Reddy - ES19BTECH11007
Suraj Gubbala - CS19BTECH11042
Vedika Verma - CS19BTECH11057
Jatin Kumar - CS19BTECH11036

Semantic Analysis

In this phase we ensure that the declarations and statements of a program are semantically correct. That is that their meaning is clear and consistent with the way in which control structures and data types are supposed to be used. Semantic analysis judges whether the syntax structure constructed in the source program derives any meaning or not.



Pipeline:


Source code
Stream
Lexer
token stream
Parser
AST
Semantic analyser
Annotated syntax tree

What does a semantic analyser do?

- Type checking: To make it sure that the operator is applied to compatible operands
- Label Checking: Labels references in a program must exist.
- Flow control checks: Control structures must be used in their proper fashion.
- Uniqueness Checking: Make it sure unique declaration of variable in a scope.
- Scope resolution

Semantic errors:

- Type mismatch
- Reserved identifier misuse.
- Multiple declaration of variable in a scope.
- Accessing an out of scope variable.
- Function overloading
- Actual and formal parameter mismatch.
- Unknown function calls
- Undeclared identifiers
- Errors in non primitive data types:
    - Matrix: different row sizes, wrong index type
    - Struct: empty declaration, illegal member access

A small snippet of our code:

```
let check_for_void errormsg = function
   (_, Datatype(typ), name, _, _) when typ = Void -> raise (Failure (errormsg name))
 | _ -> ()


let check_duplicate errormsg lst =
 let rec help_check = function
     elem1 :: elem2 :: _ when elem1 = elem2 -> raise (Failure (errormsg elem2))
   | _ :: rem -> help_check rem
   | [] -> ()
 in help_check (List.sort compare lst)
```

Example 1: empty struct declaration

```
int a;

struct x {

}

def void main ()
{
    int b;

    return;
}
```

```
(base) sharanya@sharanya-Swift-SF314-55G:~/Documents/compilers-2-project-team-8-aug21-main(1)/compile
rs-2-project-team-8-aug21-main$ ./a.out testcases/fail_emptystruct.tz
Fatal error: exception Failure("Found struct without fields 'x'")
```

Example 2: Function overloading

```
def void print (int x)
{
    int y;
}
```

```
gayalamanoj@manojgayala:~/Desktop/SEMESTER 5/COMPILERS/compilers-2-project-team-
8-aug21-main$ ./xyz test.tz
Fatal error: exception Failure("Function print cannot be defined, it is built-in
```

Example 3: No main

```
def int incr(int a)
{
    return a+1;
}
```

```
(base) sharanya@sharanya-Swift-SF314-55G:~/Documents/compilers-2-project-team-8-aug21-main(1)/compile
rs-2-project-team-8-aug21-main$ ./a.out testcases/func4.tz
Fatal error: exception Failure("Unknown function main call")
```

Example 4: Duplicate detection

```
int x;
int x;
```

```
gayalamanoj@manojgayala:~/Desktop/SEMESTER 5/COMPILERS/compilers-2-project-team-
8-aug21-main$ ./xyz test.tz
Fatal error: exception Failure("Found duplicate global 'x'")
```

Thank you!