CS 444 - Compiler Construction **Winter 2020**

Lecture 10: February 5th , 2020

Lecturer: Ondřej Lhoták *Notes By: Harsh Mistry*

10.1 Weeding

Definition 10.1 *Weeding checks language rules that could be checked by CFG, but easier to check with code*

10.2 Intermediate Representations

Definition 10.2 ***Intermediate representation*** *(IR) is the data structure representing program being compiled*

- A parse tree has many redundant nodes with non-terminals
- A abstract tree is a structure which optimizes the parse tree and stores types and values
- Build AST using recursive traversal of parse trees
- Other types of IRs
  - Instruction-based IRs
  - Graph-based IRs

10.3 Context-sensitive analysis

10.3.1 Name Resolution

- The goal is to connect every use of a name to the true meaning
- To determine which instance of a name is associated with a reference
- Class methods and inherited methods need to be caught

Definition 10.3 *An* ***environment*** *is a map from names to their meanings*

Definition 10.4 *A* ***Scope*** *is an area of the program where a declaration has effect (is visible)*

- *Associated with AST node*
- *each scope has an environment*
- *pointer to outer (enclosing scope)*

10.3.1.1 Name Lookup (use)

- Search in current environment
- not found, recursively search enclosing scopes
- if not found throw error

10.3.1.2 Declaring a name

- if name in current environment, throw error
- Insert into current environment

10.3.1.3 Name spaces

- There is usually multiple environments for some scope
- The environment for decoding/lookups is determined syntactically
- Namespaces in JAVA
  - Package Name Space
  - Type (Class or Interface)
  - Method Name Space
  - Expression Name Space (Variable, parameters, fields)