COMP 520 - Compilers

Lecture 12a (Thu Mar 31, 2022)

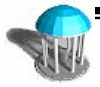
Contextual Analysis using the Visitor interface

(PLPJ pp153 – 168)

Contextual Analysis

1. Identification
   - traversal order top-down in AST
   - link identifiers to declarations
   - use scoped identification table (see lec 12)

2. Type checking
   - traversal order is bottom-up in AST
   - assign types at leaves
     - identifiers have a declared type
     - integer constants have manifest type: int or boolean
     - new T (T is a class type)
   - determine parent types
     - miniJava operators are
       - int x int → int, int x int → bool, bool x bool → bool
     - user-defined functions
       - type x …. x type → type or void



Visitor interface implements Contextual Analysis

- Contextual Analysis
  - `public class Identification implements Visitor<ScopedIdTable>`
    - maintains scoped IdTable
    - top-down traversal of AST starting from package
    - identification errors include missing or duplicate classes
  - `public class TypeChecking implements Visitor<Type,Type>`
    - checks type compatibility of values with operations
    - bottom-up traversal of AST starting from leaves
- Errors in contextual should be reported
  - if possible, continue traversal