Lecture 16:

- Quiz 4
- Semantic analysis (cont)

Announcements:

- HW-3 due
- HW-4 out
- Proj. Part 1 due next Fri.

© S. Bowers

CPSC 326, Spring 2025

1

Basic Semantic Analysis Examples

Programs can be syntactically correct but still have many errors

• Goal: Find and report errors statically (without running program code)

Detect type errors, e.g.:

```
x = 0 + "1" // int + string isn't allowed
if 42 <= true { ... } // int <= bool isn't allowed</pre>
```

Detect "use before def" errors, e.g.:

var x: int = $42 + y$	// y isn't defined
var y: int = $x + f(x)$	<pre>// f isn't defined</pre>

Detect function call errors, e.g.:

```
int add(x: int, y: int) {return x + y}
void main() {
    var r1: int = add(1, 2, 3); // wrong number of args
    var r2: int = add(3.14, 1); // wrong argument types
    var r3: bool = add(1, 2); // wrong resulting type
}
```

and so on ...

© S. Bowers

2

Basic Semantic Analysis Examples

Examples of other errors detected during static analysis (non exhaustive)

- duplicate function names, duplicate struct names
- parameters with duplicate names, fields with duplicate names
- variable shadowing
- main function not defined
- struct type not defined (when creating an object or array)
- field not found in path expression
- dereferencing a non-struct type

Type errors often based on a set of typing rules (aka Judgements)

- the rules define how types can be "*inferred*" (inference rules)
- statements or expressions that violate the rules have type errors

```
© S. Bowers
```

CPSC 326, Spring 2025

3

Basic Idea of Semantic Analysis for HW-4

- (1) **navigate** the AST using the Visitor pattern
- (2) during navigation infer types and look for errors

```
var x: int = 10
while x > 0 {
    x = x - y
}
```

For "var x: int = 10"

check / infer rhs type, compare against declared type, remember x's type
 For "while x > 0 { ... } "

• ensure x is declared and compatible with 0 (both ints), check body

For "x = x - y"

- rhs: ensure x and y are declared and types are compatible for -
- Ihs: ensure x declared and rhs result type (int) is compatible with x's type

Basic Idea of Semantic Analysis for HW-4

Example suggests we need to keep track of names and their types!

- we'll do this using a symbol table
- data structure for managing **bindings** (id -> type) in **environments**

MyPL uses static (i.e., lexical or block) scoping

- we associate to each block an environment (set of bindings)
- blocks (i.e., environments) can be nested
- bindings are found at a location by looking through all containing blocks

(*) Note: a slight abuse of the notion of an "environment"

where an "environment" is typically all of the containing blocks

© S. Bowers

CPSC 326, Spring 2025

5

Symbol Table

Stores variable state in a "stack" of environments as program is being checked

```
public class SymbolTable {
    private Deque<Map<String,DataType>> environments = new ArrayDeque<>();
    // add and remove environments
    public void pushEnvironment() { ... }
    public void popEnvironment() { ... }
    // check if variable name is bound
    public void exists(String name) { ... }
    public boolean existsInCurrEnv(String name) { ... }
    // add a binding (overwrites existing name binding)
    public void add(String name, DataType type) { ... }
    // returns the name's binding
    public DataType get(String name) { ... }
    // print the symbol table (for debugging)
    public String toString() { ... }
}
```