String DECLARE NEWVAR 'bar := true declare int ‘a := 94 int

float ERROR on line 2: Cannot perform BOOL + INT

DECLARE FUNCTION('square, 'x) case a: Int => RETURN (’average)

'x := 'x**2 WHILE(2)

ENDFUNCTION

'value := CALLFUNCTION(’square, ‘value)

ENDWHILE declare boolean ‘flag := true

def >>(rhs: Any): Symbol = bitshift(rhs, ">>")

Type System for mySIMPL

CS 345H Programming Languages Final Project

Matthew Lau, Lyee Chong, Kendall Ahrendsen, Albert Haque

December 4, 2013

bool ERROR on line 4: Conditional expected type BOOL got type INT

RETURN ('average) def >>(rhs: Any): Symbol = bitshift(rhs, ">>") float declare int 'a := 94 case a: Int => int
String DECLARE NEWVAR 'bar := true ERROR on line 2: Cannot perform BOOL + INT

Project Scope

Implement a type system for mySIMPL:

- Type inferencing for variables and primitive types
- Dynamic function parameter types
- Coercion, operator, and expression rules
- Type checking
- Informative syntax errors

GitHub

`https://github.com/lyeechong/my-simple`

bool ERROR on line 4: Conditional expected type BOOL got type INT DECLARE FUNCTION('square, 'x) declare boolean 'flag := true

RETURN ('average') def >>(rhs: Any): Symbol = bitshift(rhs, ">>") float declare int 'a := 94 case a: Int => int

String DECLARE NEWVAR 'bar := true ERROR on line 2: Cannot perform BOOL + INT

Project Plan

```
/**
    #######        ##      ##    ##    ######     ####     ######     ##   ##
    ##    ##      ####      ##  ##    ##           ##     ##    ##    ##  ##
    #######      ##  ##       ##       ######      ##     ##          ####
    ##    ##    ########      ##            ##     ##     ##    ##    ##  ##
    #######    ##      ##     ##       ######     ####     ######     ##   ##
 */
```

1. Implement mySIMPL in Scala
   - Start with BAYSICK code
2. Add explicit type declarations/primitive types

   `declare string 'temp := "Hello World"`
3. Incorporate type inferencing with primitive types

   DECLARE NEWVAR 'rellermeyer := "awesome"
4. Enforce typing rules

   `ERROR on line 2: Conditional expected type BOOL got type INT`
5. Test the system (JUnit)

bool ERROR on line 4: Conditional expected type BOOL got type INT DECLARE FUNCTION('square, 'x) declare boolean 'flag := true

RETURN ('average') def >>(rhs: Any): Symbol = bitshift(rhs, ">>") float declare int 'a := 94 case a: Int => int
String DECLARE NEWVAR 'bar := true ERROR on line 2: Cannot perform BOOL + INT

Type System Overview

Primitive Types

- Integer, Boolean, String, and Float

Our language is:

- ***Strongly-Typed***: you can't intermix values with differing data types
- ***Dynamically Typed***: variable types are not known until runtime
- Similar to Python

python

bool ERROR on line 4: Conditional expected type BOOL got type INT DECLARE FUNCTION('square, 'x) declare boolean 'flag := true

Binary Operators & Expressions

- *Arithmetic:* +, -, /, *, ** (exponent)
- *Comparison:* >, >=, <, <=, ===, =/=,
- *Logical*: &&, ||
- *Bitshift*: <<, >>, >>>
- Strings are a black hole

Input:

```
DECLARE NEWVAR 'happy := (1.0 + 4) + 2 + 3**2
```

Input:

```
DECLARE NEWVAR ‘tax := 128 << 4
```

Input:

```
DECLARE NEWVAR ‘sunny := true
DECLARE NEWVAR ‘cold := true
DECLARE NEWVAR ‘whatToWear := “”
IF (‘cold === true && ‘sunny =/= true)
    ‘whatToWear := “Sweatshirt, Sunglasses”
ENDIF
```

RETURN ('average') def >>(rhs: Any): Symbol = bitshift(rhs, ">>") float declare int 'a := 94 case a: Int => int
String DECLARE NEWVAR 'bar := true ERROR on line 2: Cannot perform BOOL + INT

Error Handling

- Goal: Assist user with debugging and syntax errors

1. Identify the line number and error (assignment, declaration, loop, etc.)
2. Show the type mismatch and intended operation

Example:

```
DECLARE NEWVAR ‘foo := 2.1
‘foo := ‘foo + 3 + true
ENDALL
```

Output:

```
‘foo:INT
Error on line 2: Cannot perform INT + BOOL
Error on line 2: Attempted to assign type INCOMPATIBLE to ‘foo:INT
```

bool ERROR on line 4: Conditional expected type BOOL got type INT DECLARE FUNCTION('square, 'x) declare boolean 'flag := true

Error Handling

- Goal: Assist user with debugging and syntax errors

1. Identify the line number and error (assignment, declaration, loop, etc.)
2. Show the type mismatch and intended operation

Example:

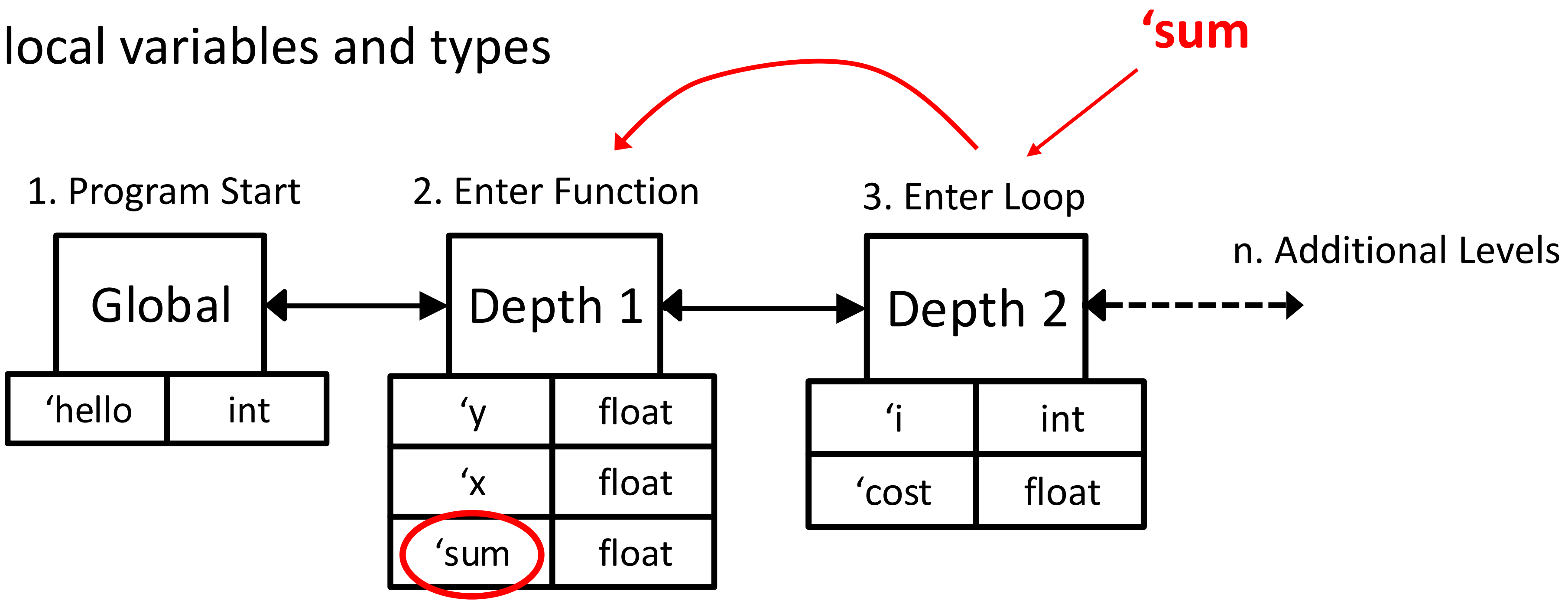
```
WHILE (1)                          WHILE (true)
    'foo := 'foo + 1    ───────▶       'foo := 'foo + 1
ENDWHILE                           ENDWHILE
```

Output:

```
'foo:INT
ERROR on line 2: Conditional expected type BOOL got type INT
```

Variable Scoping

- Conditionals and loops
- Must end all if statements and while loops inside the function
  1. Checks you're exiting at the correct scope
  2. Destroys local variables and types



Parameter Type Inference

```
DECLARE FUNCTION ('sqrt, 'x)
    DECLARE NEWVAR ‘result := 'x ** 0.5
    RETURN (‘result)
ENDFUNCTION
DECLARE NEWVAR 'y := CALLFUNCTION(‘sqrt, -4)
```

```
ERROR on line 5:
function 'sqrt expected type FLOAT, got INT
```

```
DECLARE FUNCTION ('sqrt, 'x)
    DECLARE NEWVAR ‘result := 'x ** 0.5
    RETURN (‘result)
ENDFUNCTION
DECLARE NEWVAR 'y := CALLFUNCTION(‘sqrt, 9.0)
```

```
'result:FLOAT
'y:FLOAT
```

✓

Functions

- Scoping
- Infer the function return type
- Infer parameter type
  - Look at function body

Example:

```
DECLARE FUNCTION('hello, 'hi)
     DECLARE NEWVAR ‘foo := ‘hi + 2.0
     RETURN (‘foo)
ENDFUNCTION
DECLARE NEWVAR ‘x := CALLFUNCTION(‘hello, 3.2)
```

```
'foo:FLOAT
'x:FLOAT
```

✓

RETURN ('average) def >>(rhs: Any): Symbol = bitshift(rhs, ">>") float declare int ‘a := 94 case a: Int => int
String DECLARE NEWVAR 'bar := true ERROR on line 2: Cannot perform BOOL + INT

Functions

- Scoping
- Infer the function return type
- Infer parameter type
  - Look at function body

Example:

```
DECLARE FUNCTION('hello, 'hi)
    DECLARE NEWVAR ‘foo := ‘hi + 2.0
    RETURN (‘foo)
ENDFUNCTION
DECLARE NEWVAR ‘x := CALLFUNCTION(‘hello, 3.2)
DECLARE NEWVAR ‘y := CALLFUNCTION(‘hello, “text”)
```

```
ERROR on line 6: function 'hello expected type FLOAT, got STRING
```

bool ERROR on line 4: Conditional expected type BOOL got type INT DECLARE FUNCTION('square, 'x) declare boolean ‘flag := true

```
String DECLARE NEWVAR 'bar := true declare int ‘a := 94 int
float ERROR on line 2: Cannot perform BOOL + INT

DECLARE FUNCTION('square, 'x) case a: Int => RETURN (’average)
    'x := 'x**2
ENDFUNCTION
WHILE(2)
    'value := CALLFUNCTION(’square, ‘value)
ENDWHILE declare boolean ‘flag := true
def >>(rhs: Any): Symbol = bitshift(rhs, ">>")
```

Live Demo

```
bool ERROR on line 4: Conditional expected type BOOL got type INT
```

Conclusion

Difficulties

- No experience with Scala
- Function Parameter Types
  - Hard to infer unknown types
  - Need to "guess" type

Possible Future Extensions

- Incorporate PEMDAS order of operations
- Add full support for nested functions
- Include common math functions:
  - MAX, MIN, ABS, etc.
- Add error corrections/suggestions
  - Can borrow from parameter type inference

String DECLARE NEWVAR 'bar := true declare int ‘a := 94 int

float ERROR on line 2: Cannot perform BOOL + INT

```
DECLARE FUNCTION('square, 'x)    case a: Int =>    RETURN (’average)
    'x := 'x**2
ENDFUNCTION               WHILE(2)
                             'value := CALLFUNCTION(’square, ‘value)
                          ENDWHILE              declare boolean ‘flag := true
def >>(rhs: Any): Symbol = bitshift(rhs, ">>")
```

Thank You

bool ERROR on line 4: Conditional expected type BOOL got type INT