Computer Language Processing

Lab 4

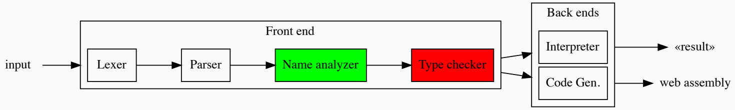
Alexandre Pinazza

Fall 2021

Labs overview

- Lab01 – Interpreter
- Lab02 – Lexer
- Lab03 – Parser
- *Lab04 – Type Checker*
- Lab05 – Codegen (Code Generator)
- Lab06 – Compiler extension

Pipeline


input
Front end
Lexer
Parser
Name analyzer
Type checker
Back ends
Interpreter
Code Gen.
«result»
web assembly

Prelude: Name Analyser

- Transforms a Nominal Tree into a Symbolic Tree
- Checks that all variables, functions and data types respects Amy naming rules
- Populates the symbol table (a dictionary of symbols for the program)
- It is provided to you (but we strongly suggest to read and understand it :-))

Type Checker

- Catches *(some)* errors in the program at compile time
  For example, it does not catches division by zero errors
- Last stage of the compiler frontend
- Does not modify the program so their is no expected outputs for the tests

Implementation

Travers a program and generate all the typing constraints

```
def genConstraints(e: Expr, expected: Type)
(implicit env: Map[Identifier, Type]): List[Constraint]
```

Unifies the constraints until none is left

```
def solveConstraints(constraints: List[Constraint]): Unit
```

Wrong example

- Input :

```
object Bogus
    "Amy <3" || 5
end Bogus
```

- Constraints

```
TypeVar(0) == BooleanType   // Top level type
BooleanType == StringType   // LHS of Or
BooleanType == IntType      // RHS of Or
```

- Error :
  The last two constraints can't be unified so the type checkers reporsts them both and crashed

Correct example

- Input :

```
object Correct
    3 + 4 == 5
end Correct
```

- Constraints

```
TypeVar(0) == BooleanType // result of equality
TypeVar(1) == IntType     // LHS of equality
TypeVar(1) == IntType     // RHS of equality
IntType == IntType        // LHS of addition
IntType == IntType        // RHS of addition
```

- Unification succeeded

```
TypeVar(0) := BooleantType
TypeVar(1) := IntType
```

Some advice

- Don’t terminate compilation directly when an error is found
- Read the handout carefully
- Write as many tests as possible
- You have one week for this lab

Finally

Good Luck !