Programming Languages 2024/25

Project 2 Report

Daniel Oliveira - Ist102854

I. Static Type-Checker

ASTTypes

As suggested by the code given a new construct ASTType was created to represent the type system of the language. This abstraction captures the variety of types expressible in the language. The following specialized ASTType variants were implemented:

- ASTTArrow – Describes a Lambda function type as ArgumentType → ReturnType
- ASTTBool – Describes a boolean
- ASTTId – Describes an Identifier
- ASTTInt – Describes an Integer
- ASTTList – Describes a List (Eager and Lazy)
- ASTTRef – Describes a reference
- ASTTString – Describes a String
- ASTTStruct – Describes a product type (structs)
- ASTTUnion – Describes a sum type (union)
- ASTTUnit – Describes a unit (null)

ASTNode

To start a new Enviroment<ASTType> was created. Now all AST nodes have a new function called typecheck(). This function mirrors the structure of the eval() but works with types and not Nodes. This function asserts that all the type in the code lead to a safe execution.

To support additional language features introduced in the type system, the following AST node types were added:

- ASTMatchUnion – Match construct for Union variables
- ASTString – Creation of Strings, VString and an update to the + operation
- ASTStruct – Creation of product types (structs) (Also comes with a new VStruct value)
- ASTUnion – " " " " but for unions
- ASTUnit – Creation of unit's (Also comes with a new ASTUnit value)
- ASTTypeDef – Let Logic for types

Type Bindings

To handle field labeling in structures and variant labeling in unions, a new class named TypeBindList was introduced. This class manages the association between labels and their corresponding ASTTypes.

Handling Ids

To handle id's int ASTType's a method called unfold was added, this method checks for ASTTId on his arguments like the type that Ref points to, and unfolds them with the enviroment.

END