TÉCNICO LISBOA

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. Like ASTTArrow (Lambda function), ASTTBool, ASTTId, ASTTInt, ASTTList, ASTTRef (pointer), ASTTString, ASTTStruct product type, ASTTUnion (sum type), ASTTUnit.

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.

Together with this some extra Nodes were create to handle extra functionalities of the language like ASTStruct, ASTMatchUnion.

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 in ASTType a method called specialEquals and specialIsSubTypeOf was added, this method checks for ASTTId on his arguments and then calls the equals and isSubTypeOf functions respectabely.

This also helped with Recursive Types that by only checking them when .equals, isSubTypeOf and some extra action are called. Leads to the possibility of having Recursive Types.

END