

Haskell Virtual Machine Implementation

Group Information

- **Group Designation:** T14_G04
- **Student Number (Contributions):**
 - Maria Eduarda Sousa Rabelo up202000130 (50%)
 - Vitor Alves Moreira up201900198 (50%)

Installation and Execution

This Haskell project does not require any additional software besides the Haskell Platform.

Project Description

This project involves the implementation of a simple virtual machine in Haskell. The machine executes a set of instructions representing arithmetic and logical operations, handling both integer and boolean values.

Features of the Haskell Virtual Machine:

1. **Arithmetic Operations:** Addition, Subtraction, and Multiplication.
2. **Logical Operations:** Boolean operations including AND, OR, and NOT.
3. **Control Flow:** Branching and looping mechanisms.
4. **State Management:** Storing and fetching variables from a state.

Internal Representation

The project uses several data types to represent instructions (`Inst``), arithmetic expressions (`Aexp``), boolean expressions (`Bexp``), and statements (`Stm``).

Key Components:

- **Stack:** Manages execution stack for operations.
- **State:** Keeps track of variable assignments.
- **Instructions:** Encapsulates operations like arithmetic and control flow.

Parser and Lexer

The program includes a lexer and parser for converting a string input into an executable list of instructions:

- **Lexer:** Tokenizes a given string input.
- **Parser:** Parses tokens into abstract syntax trees representing the program.

Execution

The main module includes a demonstration of compiling and running a sample program, showcasing the virtual machine's capabilities.

Conclusions

This implementation offers a foundational understanding of virtual machine concepts in Haskell, highlighting the language's capabilities in handling complex data structures and functional programming paradigms.

Bibliography

- Haskell Language Documentation: <https://www.haskell.org/>