Compilation Project

Presentation
Les Quatre MousquetEngineers
Master M1 MOSIG, Grenoble Universities

Cristian HARJA Yassine JAZOUANI Lina MARSS0
Clément MOMMESSIN

Joseph Fourier Grenoble University

March 11, 2015

Summary

Functionalities implemented

- Free variables
- Type Checking
- Alpha Conversion
- Beta Reduction
- Constant Folding
- Inlining
- Elimination
- K-normalization
- Closure conversion
- Intermediate Code Generation
- Assembly Code Generation

Choice of Algorithms

- AST
- Type Checking & Free Variables
- Transformation
- Intermediate Representation
- Code Generation

Transformation

- Inlining
- Elimination

K-Normalization

- K-Normalized Tree
- Insert Let*

Closure

- Special Instructions
- Visit of K-Normalized Tree

Intermediate Code Generation

- Intermediate Representation
- Separate functions
- Instr / Operand
- ASSIGN
- ADD_I, SUB_I, ADD_F, SUB_F, MUL_F, DIV_F
- LABEL, JUMP, BRANCH
- CALL, RETURN, CLS_MAKE and CLS_APPLY
- ARRAY_NEW, ARRAY_GET, ARRAY_PUT

Assembly Code Generation

- ARM assembly
- Each register has a specific role
- Register allocation
- Stack and Heap Management

Demonstration

It's demo time.

Bibliography

- MinCaml: A Simple and Efficient Compilerfor a Minimal Functional Language, Eijiro Sumii from Tohoku University.

End

Thank you for your attention, do you have any question ?