How Efficient Can Fully Verified Functional Programs be - a Case Study of Graph Traversal Algorithms

Mirko Stojadinović Filip Marić
{mirkos,filip}@matf.bg.ac.rs

Department of Computer Science
Faculty of Mathematics
University of Belgrade

Fourth Workshop on Formal and Automated Theorem Proving and Applications

Outline

Outline

Introduction

Goals

- Make programs that are efficient, and prove their correctness
- Get an insight for current possibilities to do that
- Compare efficiency between imperative and functional programs

Today's presentation focus

- Shallow embedding
  - Writing functional programs within proof assistant Isabelle/HOL
  - Proving their correctness
  - Exporting executable code to functional language ML (Haskell and Scala also supported)
- Comparing exported code efficiency to efficiency of imperative programs implementing the same algorithm
- Case study demonstrating these things: BFS algorithm (Breadth First Search) for graphs

Outline

First version of BFS - using fsets

- Write program in Isabelle/HOL using fsets (a kind of sets) and than export code
- Good properties: not so difficult to do
- Bad properties: poor efficiency (due to recreation of structures)

Outline

Monads

Monads

- Concept of monads in functional programming allows us to:
  - make imperative data structures (arrays)
  - write code similarly structured to one in imperative languages
  - achieve better efficiency

New features in Isabelle/HOL

- Isabelle 2011 includes new framework Imperative HOL employing monadic features

New idea

- prove the correctness of program using fsets (with poor efficiency)
- prove this program is equivalent to one using monads (which is much more efficient)

Outline

Efficiency comparison

Imperative vs Functional

- We compared implementation of BFS algorithm in C and ML (exported from Isabelle, using monads)
- In order to make a fair comparison, we write codes that are really similar to each other

Results

Vertices	Edges	C time	ML time
3493	6036534	0.03s	0.3s
4500	10143032	0.05s	0.5s
5555	15450463	0.07s	0.7s
70570	705700	0.02s	0.7s

Outline

Conclusions

- From the example of graph alghoritms we conclude that wide range of programs can be implemented using monads
- Their correctness can be proved in Isabelle/HOL

Future work

- Prove that fset and monad programs are equivalent
- Compare efficiency between more programs
- See the possibilities of other tools (Microsoft Spec Sharp)