A trivial Haskell vs C++ timing comparison

Andrew Shallue

Department of Computer Science
Illinois Wesleyan University

ANTS 14 Rump session, July 1, 2020

Characteristics of Haskell

- Lazy evaluation - can work with infinite objects
- Type inference
- Purely functional - no side effects
- Clean syntax
- Using tail recursion helps keep code efficient

I'm curious to see whether it would be worthwhile to do computational number theory in Haskell

Previous work: Lobachev, Loogen, Towards an implementation of a computer algebra system in a functional language, AISC 2008.

Comparisons of a single base-2 Fermat primality test:

decimal digits	time in seconds, C++	time in seconds, Haskell
10	1.1×10^{-4}	3.5×10^{-4}
100	1.4×10^{-2}	3.2×10^{-4}
1000	7.7	1.2×10^{-2}
10000	6206	4.0

Comparisons of a single base-2 Fermat primality test:

decimal digits	time in seconds, C++	time in seconds, Haskell
10	1.1×10^{-4}	3.5×10^{-4}
100	1.4×10^{-2}	3.2×10^{-4}
1000	7.7	1.2×10^{-2}
10000	6206	4.0

This is ludicrous. I really should not have used Visual Studio.

A more reasonable comparison

Thanks to Bobby Koirala, the following uses the exact same code on a laptop with linux:

decimal digits	time in seconds, C++	time in seconds, Haskell
10000	3.4	4.5

This is unfair to Haskell, because the C++ implementation has fast multiplication (through NTL and GMP) while the Haskell implementation does not.