Haskell & LLVM

Why compilers/interpreters?

1. Some of the largest Javascript performance improvements
in recent years are due to better interpreters see V8 (http://en.wikipedia.org/wiki/V8_(JavaScript_engine)).

2. It's a fun subject! You get play with really cool tools that are practically magical.

What is Haskell?

1. Functional
2. Pattern Matching
3. High Level & Succinct
4. Lazy
5. Math focused

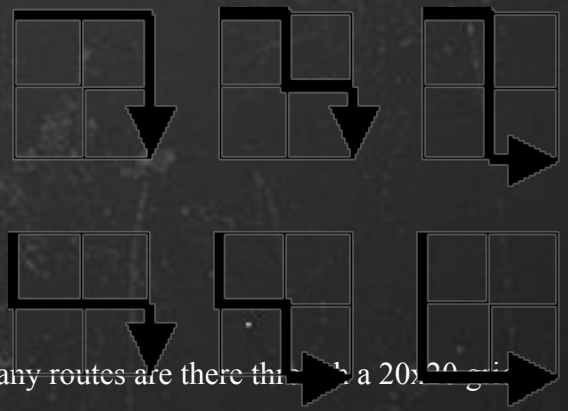
Example Program in Haskell

Definition of the Fibonacci squence.
f(0) = 0, f(1) = 1, f(n) = f(n - 1) + f(n - 2).

In Haskell
```
f 0 = 0
f 1 = 1
f n = f (n - 1) + f (n - 2)
```
(show demo of writing f in emacs & WinGHCI with guards)

A more interesting example (project euler #15)

Starting in the top left corner of a 2x2 grid, there are 6 routes (without backtracking) to the bottom right corner.



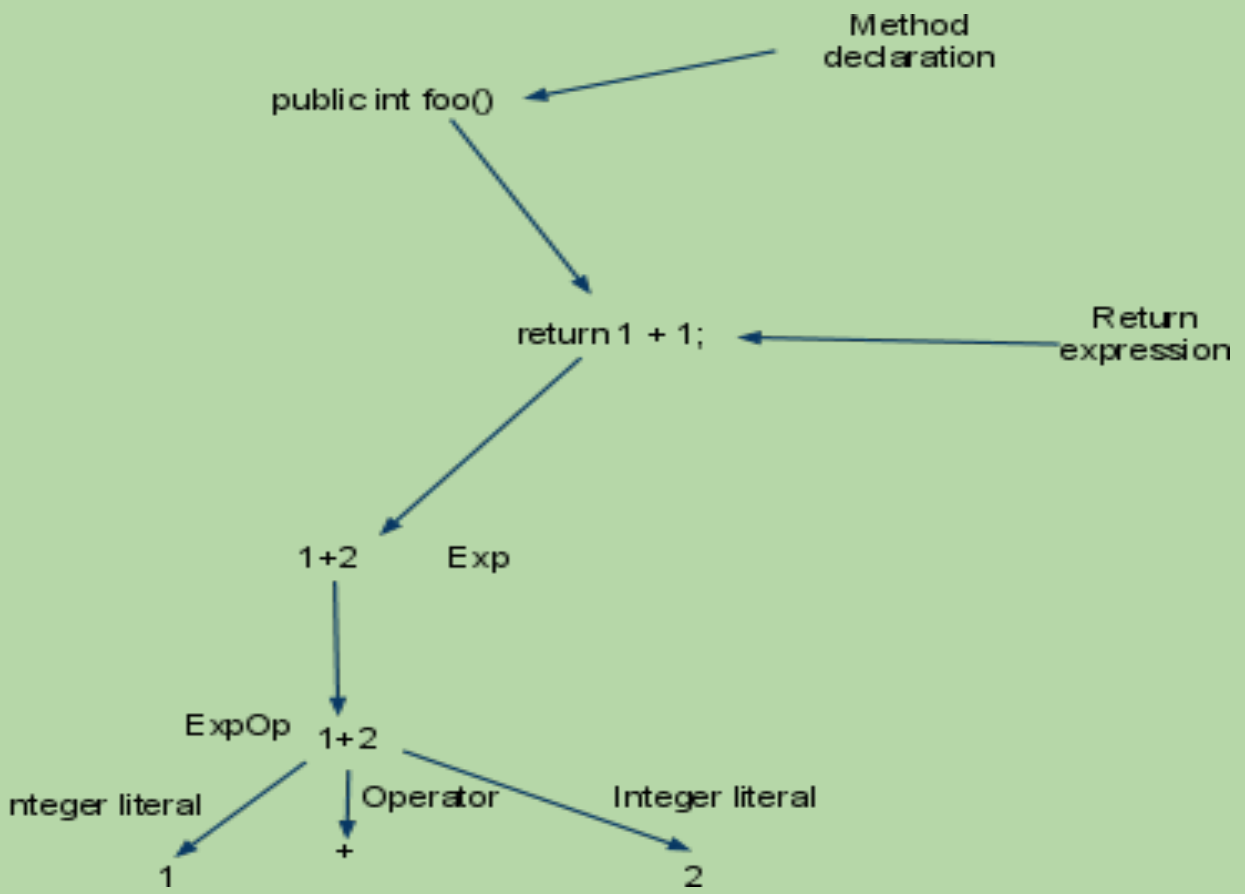
How many routes are there through a 20x20 grid?

Answer **137846528820**

Think recursively.
(See euler_15.hs)

Why Use Haskell for Writing Compilers?

Recursion & pattern matching naturally matches the structure of parse trees:



Technical Details of Haskell + LLVM

1. Defined the grammar for Java like language using Backus-Naur Form. http://bloggingmath.wordpress.com/2010/01/20/writing-a-compiler-in-haskell-compiler-series-part-i/
2. Used the Haskell version of Lex for scanning and the Haskell version of Yacc for parsing. (go over data types, http://bloggingmath.wordpress.com/2010/04/06/having-fun-with-happy-compiler-series-part-iii/).
3. Added a symbol table for type checking
4. Added a type checking pass on the parse tree.
5. Created functions to traverse the parse tree and call LLVM to generate the LLVM assembly code.
6. LLVM does the work of generating machine code.

Using LLVM for Code Generation

1. Code generation is implemented as a pass over the parse tree
2. The result of the code generation is LLVM IR (intermediate representation) code.
3. LLVM IR Example:

```
define i64 @foo()
{
    entry:
      ret i64 3
}
```

(entry: is a basic block)

LLVM Code Generation Pass

See for the full code listing http://bloggingmath.wordpress.com/2010/08/13/the-basics-of-llvm-compiler-series-part-x/

LLVM Haskell API Bindings

LLVM is a C++ library. They have a set of C API bindings that the Haskell bindings use for calling LLVM.

Bryan O'Sullivan (of Real World Haskell fame) and Lennart Augustsson created the Haskell bindings see http://hackage.haskell.org/package/llvm for the latest version.

Thank You

QA?

1. Haskell (http://haskell.org)
2. LLVM (http://llvm.org).