

Idea: Use GADT's to capture the type system:

```
{-# LANGUAGE GADTs #-}

-- LANGUAGES
data Haskell
data Java

data OutputCompiler
data OutputProgram
data OutputInterpreter

-- LANGUAGE TYPE
data Language l where
  LangHaskell :: Language Haskell
  LangJava :: Language Java

-- PROGRAM
data Program l where
  Program :: String -> Language l -> Program l

exampleProg = Program "hello" LangHaskell

progtest :: Program a -> String
progtest (Program name l) = name

-- PLATFORMS
data l686

-- PLATFORM
data Platform p where
  Platformi686 :: Platform l686

-- INTERPRETER
data Interpreter l p where
  Interpreter :: String -> Language l -> Platform p -> Interpreter l p

data Execution l p where
  ProgramOnPlatform :: Program l -> Platform p -> Execution l p
  ProgramOnInterpreter :: Program l -> Interpreter l p -> Execution l p
  InterpreterOnPlatform :: Interpreter l p -> Platform p -> Execution l p
  InterpreterOnInterpreter :: Interpreter l p -> Interpreter l p -> Execution l p
  CompilerOnPlatform :: Compiler l1 l2 p -> Platform p -> Execution l2 p
  CompilerOnInterpreter :: Compiler l1 l2 p -> Platform p -> Execution l2 p
  CompilationOnPlatform :: Compilation l1 l2 p o -> Platform p -> Execution l2 p
  CompilationOnInterpreter :: Compilation l1 l2 p o -> Interpreter l2 p -> Execution l2 p
  ProgramOnCompilation :: Program l2 -> Compilation l1 l2 p OutputInterpreter -> Execution l2 p

data Compiler l1 l2 p where
  Compiler :: String -> Language l1 -> Language l2 -> Platform p -> Compiler l1 l2 p

data Compilation l1 l2 p output where
  ProgramWithCompiler :: Program l1 -> Compiler l1 l2 p -> Compilation l1 l2 p OutputProgram
  ProgramWithPlatform :: Program l1 -> Platform p -> Compilation l1 l2 p OutputProgram
  CompilerWithPlatform :: Compiler l1 l2 p -> Platform p -> Compilation l1 l2 p OutputCompiler
  CompilerWithInterpreter :: Compiler l1 l2 p -> Interpreter l1 p -> Compilation l1 l2 p OutputCompiler
  InterpreterWithPlatform :: Interpreter l1 p -> Platform p -> Compilation l1 l2 p OutputInterpreter
  ProgramWithCompilation :: Program l1 -> Compilation l1 l2 p OutputCompiler -> Compilation l1 l2 p OutputProgram

hugs = Interpreter "hugs" LangHaskell Platformi686
jvm = Interpreter "jvm" LangJava Platformi686
```