

IB Concepts in Programming Languages

Nandor Licker

May 12, 2019

Types

1. What is the most generic type of `let x = fun y -> y` and `let x = ref (fun y -> y)`?
2. Briefly comment on emitting accurate error messages for type errors in OCaml/ML.
3. Name 3 languages with structural equality and 3 languages with nominal equality.
4. Comment on the type discipline present in JavaScript. What is lost and what is gained when a type system/type checker such as TypeScript or Flow is added on top of the language?
5. What are the differences between `std::vector<A>` in C++ and `Array<A>` in Java? Can we have covariance in the STL data structures in C++? What is different in Java?

Scripting Languages

1. In JavaScript, what is `Function`, `Function.prototype`, the prototype of `Function.prototype` and the prototype of `function f()`? How do these structures map to Java?
2. JavaScript has `instanceof`, but no classes. Comment on how this operator aligns with the prototype-based design of the language and on how it might be implemented.

Modules

1. How do modules decouple an interface from its implementation?
2. Give the type signature of an associative map in ML/OCaml. Given a module implementing an associative map in OCaml, design a functor producing a multiset.
3. Are functors compile-time or runtime constructs? Do they add a runtime overhead?

Concurrency

1. Why would an application developed for X86 crash when ported to ARM?
2. Identify issues regarding memory layout when implementing MapReduce in C++ or Java. Compare this to problems arising in OpenCL/Cuda/OpenMP.
3. Operating systems with hybrid kernels, such as NT for Windows or XNU for macOS, implement message passing. Is this actually safer, considering the language they use?

Scala

1. What does Scala sacrifice by targeting the JVM?
2. Implement a covariant binary tree in Scala with a generic `map` function.

Papers

- 2018 Paper 7 Question 1
- 2017 Paper 3 Question 5
- 2016 Paper 3 Question 5