IB Concepts in Programming Languages

Nandor Licker

May 5, 2019

FORTRAN

1. How can a function return multiple values in FORTRAN? Compare the FORTRAN implementation with C, Go and OCaml. Comment on which options are more readable and which options are more efficient.
2. Provide examples of errors which are not detected statically in FORTRAN, but could actually be identified ahead of runtime? Comment on the runtime behaviour of such constructs. Does C handle these problems better?
3. Comment on the safety of pass-by-reference parameters in C++. How is this better than FORTRAN?

LISP

1. Why does LISP require garbage collection? Describe the first garbage collector implemented for LISP.
2. What are the challenges of statically compiling LISP to machine code?
3. In OCaml, define a data structure capable of representing any LISP program. Write a simple parser for it in OCaml.
4. How does argument passing differ from FORTRAN?

Algol and Pascal

1. Comment on the similarities and differences regarding named and optional parameters in Ada, C++, Python and OCaml.
2. What are the challenges in implementing `out` and `in out` parameters in Ada, C++, C#, Java and OCaml?
3. What are the challenges in using both stack and heap storage in a language?
4. Why is Pascal's type system stronger than that of C?

SIMULA and Smalltalk

1. Why do compiler writers strongly dislike dynamic dispatch? Enumerate 3 optimisations that are rendered ineffective due to dynamic dispatch.
2. Enumerate features which complicate the execution and compilation of `a.x` in JavaScript.
3. Enumerate 3 advantages and 3 disadvantages of reflection.
4. In Smalltalk everything is an object. How could this interfere with reference counting?

Papers

- 2007 Paper 6 Question 7
- 2009 Paper 3 Question 2