PhD in Programming Languages (Ownership and Immutability)

Unification of Immutability and Ownership

Object-oriented programs at run-time consist of *objects* – small software components that are created, changed, and destroyed as the program runs. The uncontrolled ability of objects to change other objects is well known to result in many errors in large systems.

Immutability – an ability to prevent changes to objects – is a mechanism that could be used to provide such control. While immutability controls *whether* an object can be changed, it is not enough: we need to be able to also control *which* objects can perform the changes.

Ownership – an ability to control *which* objects can access a given object – is a mechanism that can be used to control which objects are allowed to perform the changes. Immutability benefits greatly from ownership that prevents unauthorised objects from accessing and changing the object.

Immutability and ownership evolved separately: while some proposed languages attempt to support both, they treat them as independent concepts.

We propose to treat them as facets of a more general unified concept. This work aims to advance the science of programming by providing a unified treatment of immutability and ownership, simplifying both the theory and practice of object-oriented programming.

The Student

This project requires a strong mathematical logic or formal methods in Computer Science background.

Contact

Dr Alex Potanin
alex@mcs.vuw.ac.nz
http://homepages.mcs.vuw.ac.nz/~alex/