Automated Refactoring of Client Code for Immutability

Mihai Codoban, Politehnica University of Timișoara
Jack Ma, Rose-Hulman Institute of Technology
ADVISOR: Danny Dig

Goals

- Develop analyses that verify whether the client is able to use a target class in an immutable fashion.
- Rewrite the client code to use a target class in an immutable fashion.

Research Questions / Motivating Example

- What are the best ways to change the client code in order to safely use the target class in an immutable fashion?
- Is the solution unique, or do different patterns of usage require different refactoring strategies?

Refactoring Flow

1. Generate Immutable Class

Mutable Class

Immutable Class

2. Adapt Client Code

Client

Client'

Preliminary Research Results

a) Immutable classes use factory methods to change state.
   - After mutator calls, references point to old objects.
   - Client must reassign the newly created objects back to the references.
   - Encapsulates the immutable objects.
   - Delegates calls.
   - Internally reassigns new objects back to reference.
   - The client code does not change due to wrapper’s preservation of behavior.

b) The Wrapper hides the spawning of new immutable objects upon mutation from the client.
   - Encapsulates the immutable objects.
   - Delegates calls.
   - Internally reassigns new objects back to reference.
   - The client code does not change due to wrapper’s preservation of behavior.

Related Work / Interaction with Other Projects