Soundness Across Formalizations of Decentralized Information Flow Control

Swetabh Changkakoti, CSCI 1951x

In this capstone project for CSCI 1951x: Formal Proof and Verification, I modeled the semantics of Decentralized Information Flow Control (DIFC), as presented in the paper 'Complete, Safe Information Flow with Decentralized Labels' (1998) by Andrew C. Myers and Barbara Liskov. I used Lean 3 as an automated proof assistant and modeling language throughout this project.

I used inductive predicates to model incremental and larger relabelings for a simplified relabeling rule (the subset relabeling rule) and a complete one (the complete relabeling rule). I included and proved a set of example theorems to show how we can use these inductive predicates to prove the validity of certain relabelings. Further, I defined closed-form rules and functions that help model the complete relabeling rule, and close out the project by proving that this rule is sound.

```
-- Example: here, this relabeling replaces an owner with a principal that acts for
-- the previous owner. It also removes a reader, as we could earlier.
def ph_single : PHierarchy := [10 ⩾ 1]
def cl1 : Label := {policies := [{owner := 1, readers := {2}}]}
def cl2 : Label := {policies := [{owner := 10, readers := {}}]}

theorem eg_can_replace_owner_and_remove_reader : CompleteRelabel ph_single cl1 cl2 :=
  by
    apply CompleteRelabel.trans ph_single cl1 cl2 {policies := [{owner := 1, readers := {}}]}
    {
      apply CompleteRelabel.direct
      have hincr : IncrSubRelabel cl1 {policies := [{owner := 1, readers := {}}]} :=
        by
          apply IncrSubRelabel.remove_reader
          apply Exists.intro {owner := 1, readers := {2}}
          apply Exists.intro {owner := 1, readers := {}}
          apply Exists.intro 2
          simp
          intro m_in_cl1
          apply Exists.intro 2
          simp
      apply IncrRelabel.subset
      exact hincr
    }
    {
      apply IncrRelabel.replace_owner
      apply Exists.intro {owner := 1, readers := {}}
      apply Exists.intro {owner := 10, readers := {}}
      apply Exists.intro 1
      apply Exists.intro 10
      simp
    }
    done
```

Pictured above is an example theorem I stated and proved using my formalization of labeling rules for DIFC.