

## Milestone 6

### Changes

Since my last milestone update, we have pivoted my focus from repair to inferring policies.

### Accomplishments Since Last Update

Since my last update, I have made good progress towards verifying database queries in Yesod. I have figured out how to verify that relationships hold by using Liquid Haskell measures-- things that look like functions that may be used in the refinement logic.

For example, suppose the haskell function `isAuthor :: Paper -> User -> Bool` returns true if and only if the user is an author for the paper. We can create the following measure and refinement on `isAuthor` to use this relationship in a refinement:

```
{-@ measure author :: Paper -> User -> Bool @-}  
{-@ isAuthor :: p:Paper -> u:User -> {v:Bool | v ⇔ author p u} @-}
```

Now a function, `getPapersForAuthor :: User -> [Paper]`, can have the following refinement type:  
`{-@ getPapersForAuthor :: u:User -> [{p:Paper | author p u}]`.

Since this is done, all that is left on my milestones is repair. However, our collaborator who focuses on program synthesis has been very busy interviewing for academic positions, so we decided to postpone that part of my research. Instead, I am focusing on how to *infer* policies from the code. This would make our tool much easier to use, since no work would be needed to use it!

### Meeting Milestones

Since we are no longer focusing on repair, my milestones at this point are complete.

### Looking Ahead

Over the next two weeks, I will be focusing more on defining just what we want for policy inference, and figuring out how to use Liquid Haskell to get it. This involves collaborating more closely with our Liquid Haskell friends at UCSD.

### Resources Needed

I require no additional resources to continue working on my project.