Proof Assistants as Macros

David Christiansen  Indiana University

Why?

• **Accurate bindings** from the macro expander’s hygiene information
• Proof rules can **reason about macro expansion**
• Seamless **integration** between proof assistant and rest of language

Accurate Bindings

Hygienic macro expanders track identifiers rather than names. This allows us to lift restrictions on fresh names, and naïve substitution becomes capture-avoiding.

Proofs About Macros

Reasoning about racket requires reasoning about macros. Macro expansion depends on the complete compile-time context and cannot be easily axiomatized or simulated. Executing proof steps within the macro expander allows them to directly use macros.

IDE Integration

DrRacket and racket-mode for Emacs provide hooks that macros can use to decorate programs. This can be used to display proof goals and provide real-time feedback.

Implementation: CPS + LCF

Tactics are partial functions from goals to the syntax of evidence, which may contain further holes. Each hole has a tactic and a tree of subgoal continuations.

```
(modus-ponens (\(\Rightarrow\) (\(\Rightarrow\) A B) A B))
```

```
(\(\lambda\) \(f\) \(x\) \(\Rightarrow\) (\(\Rightarrow\) [\(f\) (\(\Rightarrow\) A B)] \([x\ A]\) B))
```

```
(\(\lambda\) \(f\) \(x\) \(\Rightarrow\) (\(\Rightarrow\) [\(f\) (\(\Rightarrow\) A B)] \([x\ A]\))...
```

```
(\(\lambda\) \(f\) \(x\) \(\Rightarrow\) (\(\Rightarrow\) [\(f\) (\(\Rightarrow\) A B)] A))
```

Sequencing

Sequencing prepends a tactic onto the subgoal continuation tree. Then prepends the same tactic to all subtrees, while then-1 prepends different tactics to each subtree.

Failure

Failure is implemented by adding a failure continuation.

```
(modus-ponens (\(\Rightarrow\) (\(\Rightarrow\) A B) A B))
```

```
(\(\Rightarrow\) intro)
```

```
(orelse \(\Rightarrow\) intro)
```

```
(\(\Rightarrow\) intro)
```

```
(\(\Rightarrow\) intro)
```

```
(\(\Rightarrow\) intro)
```

```
(\(\Rightarrow\) intro)
```

```
(\(\Rightarrow\) intro)
```

Implementation: Tacticals

Holes track the current tactic, a tree of continuations for subgoals, and a failure continuation. Proof scripts are controlled with LCF-style tacticals that modify the continuations. Control tacticals produce a hole with the same goal.

Work supported by NSF grant 1540276