Differences of Proofs

John Leo

August 8, 2017

1 Introduction

This document is a place to record my thoughts on the Coq proof repair project headed by Talia Ringer, Nate Yazdani, and Dan Grossman. Everything in it is preliminary.

2 Difference of Terms

Consider the following scenario. We are given type A and a fixed term $a : A$. For a given type B (for the time being, not depending on a) we would like to construct a function (called a "patch") $f : A \to B$. Now say that you are given $b : B$. Clearly one can set $f(x) = b$ for all x. However what we would like to do is find a "minimal" body of f where we are trying to minimize some measure such as the size of its AST, and if we could replace any occurrence of a in b with x then that would reduce the size by $|a| - |x| = |a| - 1$. We can thus try to find the maximal number of occurrences of a in b that we can abstract and replace with the variable x.

2.1 Examples

For now my examples will be written in Agda, as the original examples in the user study use Ltac whereas the actual work is done in Gallina so I'd need to generate the Gallina myself; also Agda is easier to read, at least for me. But the Gallina code is very similar and can be generated later if it matters.

The first example from the user study is the following.

```
≤transWeak : {n m p : ℕ} → n ≤ m → m ≤ p → n ≤ p + 1
≤transWeak {p = p} a b = ≤-trans (≤-trans a b) (m≤m+n p 1)

≤trans : {n m p : ℕ} → n ≤ m → m ≤ p → n ≤ p
≤trans a b = (≤-trans a b)

≤transWeakPatch : ({n m p : ℕ} → n ≤ m → m ≤ p → n ≤ p) →
                  ({n m p : ℕ} → n ≤ m → m ≤ p → n ≤ p + 1)
≤transWeakPatch P {n} {m} {p} b c = ≤-trans (P b c) (m≤m+n p 1)
```

```
≤transWeak' : {n m p : ℕ} → n ≤ m → m ≤ p → n ≤ p + 1
≤transWeak' = ≤transWeakPatch ≤trans
```

In this case our original term a is $\leq$-trans.

2.2 Strengthened Conclusions

All eight of the examples in the CSE 503 project, as well as most later examples (such as oldMinimal/newMinimal in email) are of the same form, which could be characterized as strengthening or in general modifying the conclusion of a theorem given a set of hypthoses. The types involved are pi types, and we are given terms $a : (x : X) \to A$ and $b : (x : X) \to B$ where A and B can depend on x. Note that X is identical for both a and b. We view it here as a single sigma type, but in the original form of the it is curried so that a and b have pure pi types.

3 Alternate Datatype Formulations