

# Formalization of Basic Results in Category Theory

JV Siratt

December 26, 2016

## 1 Main Files

These files make up the core of the formalization.

### 1.1 `category.pvs`

The main category file is parameterized by the data necessary to build a specific category. The assumptions section includes the basic axioms of category theory, and the body contains category theory results.

NEXT UP: define isos, epis, and monos, then prove the proposition "If an arrow is both epi and mono, then it is iso."

### 1.2 `group_as_category.pvs`

This is an abstraction from `example_particular_group.pvs` that is parameterized. It acts as a simple interface for casting a group in category theory terms.

## 2 Example Files

These files are particular examples of categories used to both test the efficacy of the formalization in `category.pvs` and as stepping stones toward fully parameterized abstract interfaces for instantiating certain types of categories.

### 2.1 `example_category_3.pvs`

This is the simple category containing three objects,  $A, B, C$ , and the arrows  $f : A \rightarrow B, g : B \rightarrow C, g \circ f : A \rightarrow C$ . There will be no abstraction from this file since instantiating a category by sets works directly with the base file.

### 2.2 `example_particular_poset.pvs`

This is the simple poset category where the objects are the integers  $\{1, 2, 3, 4, 5\}$  with each arrow representing a less-than-or-equal-to relation. This should lead to an abstract interface for poset categories (not categories of posets, which would be subcategories of the category of sets).

NEXT UP: Abstract to `poset_as_category.pvs`.

### **2.3   `example_int_sets.pvs`**

This is a subcategory of the category of sets containing sets of integers and functions between them. Currently this is unfinished, but should lead to an abstract interface for subcategories of the category of sets.

NEXT UP: Rework identity and composition type signatures to try to simplify unity and associativity assumption `tccs`.

### **2.4   `example_particular_group.pvs`**

This is an implementation of the group of integers under addition. This abstracted directly to `groups_as_categories.pvs`.