```
(*****************************************************************************************************************************)
(* Bijection Lemma                                                                                                           *)
(*                                                                                                                           *)
(*    Used in the proof that the mapping between the category of reifications and category of generalized arrows is in fact    *)
(*    a functor                                                                                                              *)
(*                                                                                                                           *)
(*****************************************************************************************************************************)

Generalizable All Variables.
Require Import Preamble.
Require Import General.
Require Import Categories_ch1_3.
Require Import Functors_ch1_4.
Require Import Isomorphisms_ch1_5.
Require Import ProductCategories_ch1_6_1.
Require Import OppositeCategories_ch1_6_2.
Require Import Enrichment_ch2_8.
Require Import Subcategories_ch7_1.
Require Import NaturalTransformations_ch7_4.
Require Import NaturalIsomorphisms_ch7_5.
Require Import MonoidalCategories_ch7_8.
Require Import Coherence_ch7_8.
Require Import Enrichment_ch2_8.
Require Import RepresentableStructure_ch7_2.

Section BijectionLemma.

  (* given two categories with (for simplicity) the same objects *)
  Context  {Obj:Type}.
  Context  {CHom}{C:Category Obj CHom}.
  Context  {DHom}{D:Category Obj DHom}.

  (* and a functor which is (for simplicity) identity on objects *)
  Context  {F:Functor C D (fun x => x)}.

  (* and a mapping in the opposite direction, which respects equivalence *)
  Context  {M:forall x y, (x~~{D}~~>y) -> (x~~{C}~~>y)}.
  Implicit Arguments M [ [x] [y] ].
  Context  {M_respects:forall x y, forall (f g:x~~{D}~~>y), f~~g -> M f ~~ M g}.

  Add Parametric Morphism (x y:Obj) : (@M x y)
```

```
  with signature ((@eqv D _ D x y) ==> (@eqv C _ C x y)) as parametric_morphism_fmor''.
    intros; apply M_respects; auto.
    Defined.

  (*
   * If the functor and the mapping form a bijection, then the fact
   * that the functor preserves composition forces the mapping to do so
   * as well
   *)

  Lemma bijection_lemma :
    (forall A B, forall f:(A~~{C}~~>B), M (F \ f) ~~ f) ->
    (forall A B, forall f:(A~~{D}~~>B), F \ (M f) ~~ f) ->
    forall A B C, forall f:(A~~{D}~~>B), forall g:(B~~{D}~~>C), M f >>> M g ~~ M (f >>> g).

    intro lem1.
    intro lem2.
    intros.
    rewrite <- (lem2 _ _ f).
    rewrite <- (lem2 _ _ g).

    set (@fmor_preserves_comp _ _ _ _ _ _ _ F) as q.
    rewrite q.
    rewrite lem1.
    rewrite lem1.
    rewrite lem1.
    reflexivity.
    Qed.

End BijectionLemma.
```