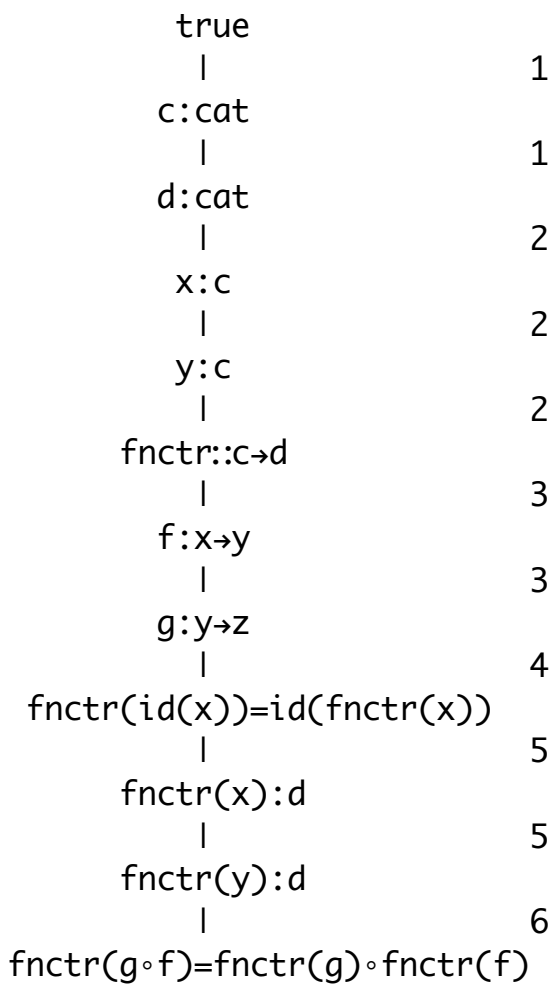
```
/*
                        CATEGORY LOGIC
                            Functor
                    AutoLog Indexical Forcing
                           12/23/2024
  * How to specify AutoLog Rules for a covariant Functor
  * corresponding to category theory diagrams.
  * Notes regarding operators '::' vs ':'
  *   1st order category indexical
  *    F:X→Y   category mapping F from object X to object Y
  *    C:cat   C is category
  *   2nd order category indexical
  *    F::C→D   Functor F from cat C to cat D
  * Consider the following autoLog theory Functor.auto
  */
     true => c:cat, d:cat.                    // 1
     true => x:c, y:c, fnctr::c→d.            // 2
     true => f:x→y, g:y→z.                    // 3

     C:cat, D:cat, Fnctr::C→D, X:C => Fnctr(id(X))=id(Fnctr(X)). // 4

     C:cat, D:cat, Fnctr::C→D, X:C => Fnctr(X):D.                // 5

     C:cat, D:cat, Fnctr::C→D, X:C, Y:C, Z:C, F:X→Y, G:Y→Z =>
         Fnctr(G◦F)=Fnctr(G)◦Fnctr(F).                           // 6
/*
   The AutoLog theory Functor.auto compiles as is ....
   Next: Use rules 4,5,6 to force conclusions from 1,2,3,
   using AutoLog Inference trees. This idea is intended to illustrate
```

how autolog indexical forcing is applicable to deriving logical conclusions inferred from the category diagram. The intention is to display what graphical output to expect from the AutoLog Skolem Machine inferences for the Functor computations.



```
   So far it appears that the Functor related rules for indexical
   forcing generally work as intended. Carefully presenting these
   autolog inferences as proof branches are fairly easy to follow.
   Notice that rules 4,5, and 6 are applied to inference branch facts
   generated earlier in the tree by applying rules 1,2,and 3 first.
*/
```