lib/algebra/function_unittest.ath

```
load "group"
load "algebra/function"

assert (theory-axioms Function.theory)

define fun-prop :=
  method (P) (!prove-property P no-renaming Function.theory)

(get-precedence Function.o)
(get-precedence Function.at)

(!claim Function.compose-definition)

(!fun-prop Function.associative)
(!fun-prop Function.right-identity)
(!fun-prop Function.left-identity)

define fun-monoid :=
  (renaming |{Monoid.+ := Function.o, Monoid.<0> := Function.identity}|)

pick-any f:(Function.Fun 'T 'T)
         g:(Function.Fun 'T 'T)
         h:(Function.Fun 'T 'T)
  (!instance Function.associative [f g h])

(print-instance-check fun-monoid Monoid.theory)

(!fun-prop Function.identity-surjective)
(!fun-prop Function.identity-injective)
(!fun-prop Function.identity-bijective)
(!fun-prop Function.compose-injective-preserving)
(!fun-prop Function.compose-surjective-preserving)
(!fun-prop Function.compose-bijective-preserving)
```