load "ring.ath"

module Test-Ring {
  open Ring
  define Ring1 := no-renaming
  assert (theory-axioms Ring.theory)
  (!prove-property Group.left-inverse Ring1 Ring.theory)
  (!prove-property Group.neg-plus no-renaming Group.theory)
  (!prove-property Abelian-Group.neg-plus no-renaming Abelian-Group.theory)
}

module Test-Commutative-Ring {
  open Commutative-Ring
  declare Times1: (T) [T T] -> T
  declare Plus1: (T) [T T] -> T
  declare Zero1: (T) [] -> T
  declare Negate1: (T) [T] -> T
  define Ring2 := (renaming |{Ring.* := Times1, Group.+ := Plus1,
    Group.<0> := Zero1, Group.U- := Negate1}|
  assert (Ring2 (theory-axioms Commutative-Ring.theory))
  (!prove-property Group.left-inverse Ring2 Commutative-Ring.theory)
  (!prove-property Group.left-inverse no-renaming Commutative-Ring.theory)
  (!prove-property Group.left-inverse no-renaming Commutative-Ring.theory)
  (!prove-property Abelian-Group.neg-plus no-renaming Commutative-Ring.theory)
}