lib/main/nat-times-minus.ath

```
load "nat-times-less.ath"
load "nat-minus.ath"

extend-module N {

extend-module Minus {

define Times-Distributivity :=
  (forall x y z . x * y - x * z = x * (y - z))

conclude Times-Distributivity
  pick-any x y z
    (!two-cases
      assume A := (z <= y)
        (!chain->
          [(x * y)
           = (x * ((y - z) + z))   [Plus-Cancel]
           = (x * (y - z) + x * z) [Times.left-distributive]
           = (x * z + x * (y - z)) [Plus.commutative]
           ==> (x * y - x * z = x * (y - z))
                                   [Plus-Minus-property]])
      assume A := (~ z <= y)
        let {C := (!chain-> [A ==> (y < z)  [Less=.trichotomy1]])}
          (!combine-equations
           (!chain->
            [C ==> (C | y = z)       [alternate]
               ==> (y <= z)          [Less=.definition]
               ==> (x * y <= x * z)  [Times.<=-cancellation-conv]
               ==> (x * y - x * z = zero)
                                     [second-greater-or-equal]])
           (!chain
            [(x * (y - z))
             = (x * zero)       [second-greater]
             = zero             [Times.right-zero]]))))
} # Minus
} # N
```