lib/main/nat-less-swo.ath

```
# Prove theorems about N.< using proofs in SPO.Theory and SWO.Theory

load "nat-less"
load "order"

#.....................................................................

# We already proved N.Less.asymmetric in nat-less.ath, but the
# following shows we could have gotten it from the abstract SPO
# theory in order.ath.

# For the following proofs we use an instance of the abstract SWO
# theory, but there's a complication that arises when replacing E by =.
# We can't use = directly, as it is too generic.  Instead we have to
# introduce a Nat version, Nat-Equal, as a replacement for E, then
# replace Nat-Equal with =.

declare Nat-Equal: [N N] -> Boolean

define Nat-order := (o (renaming |{Nat-Equal := =}|)
                       (renaming |{SPO.< := N.<, SPO.E := Nat-Equal}|))

conclude N.Less.asymmetric
  (!prove-property Strict-Partial-Order.asymmetric Nat-order SPO.theory)


# We also need the following transitivity property of =, a Nat-order
# instance of (SWO 'E-Transitive).

conclude (forall ?x:N ?y:N ?z:N . ?x = ?y & ?y = ?z ==> ?x = ?z)
  pick-any x:N y:N z:N
    assume (x = y & y = z)
      (!chain [x --> y  [(x = y)] --> z [(y = z)]])

# Now we can:

conclude N.Less.transitive1
  (!prove-property SWO.<-transitive-not-1 Nat-order SWO.theory)

conclude N.Less.transitive2
  (!prove-property SWO.<-transitive-not-2 Nat-order SWO.theory)

conclude N.Less.transitive3
  (!prove-property SWO.<-transitive-not-3 Nat-order SWO.theory)
```