Specification(s) of the Infinite Array Data Structure

This specification will take Nat = {0,1,2, , … }, plus the familiar arithmetic operations, as a pre-defined ADT. It will be conceptually simplest to specify infinite arrays whose entries are of sort Nat. The TOI is the sort Array with

Signature
CREATE: → Array
-- CREATE an infinite array whose entries are all 0

STORE: Array × Nat × Nat → Array
-- STORE(a,p,n) returns array b with same entries as a, except b[p] = n

FETCH: Array × Nat → Nat
-- FETCH(a,p) returns a[p]

Equations (for all a Array; n,p,q Nat)
FETCH(STORE(a,p,n),q) = **if** equal(p,q) **then** n **else** FETCH(a,q)

FETCH(CREATE,p) = 0

With these two equations and the *final* algebra interpretation, we obtain “arrays” as they are normally understood. Arrays are distinguished by accessing different values with the same index, but there is no equality applying to two distinct array terms.

For instance,
STORE(STORE(CREATE,0,5),1,6) ≈ STORE(STORE(CREATE,1,6),0,5), but

STORE(STORE(CREATE,0,5),1,6) ≠ STORE(STORE(CREATE,1,6),0,5).

To obtain the same abstraction with the initial algebra view, we must add the axiom (for all a Array; m,n,p,q Nat)
STORE(STORE(a,p,m),q,n) = **if** equal(p,q)
then STORE(a,p,n)
else STORE(STORE(a,q,n),p,m)