**Tutorial Nov 30**

* Ex:

assert(true)

assert(((B ⊕ {(3, 5}) ⊕ {3, 6})[3] = 6)

B[3] := 5;

assert((B ⊕ {(3, 6)})[3] = 6) % array asn

B[3] := 6;

assert(B[3] = 6) % array asn

* Ex:

assert(∀k . B[k] = B0[k])

% implied VC 1

assert( ( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[i] = B0[j] ∧ ( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[j] = B0[i] )

x = B[i];

assert( ( (B ⊕ {(i, B[j])}) ⊕ {(j, x)} )[i] = B0[j] ∧ ( (B ⊕ {(i, B[j])}) ⊕ {(j, x)} )[j] = B0[i] ) % asn

B[i] = B[j];

assert((B ⊕ {(j, x)})[i] = B0[j] ∧ (B ⊕ {(j, x)})[j] = B0[i]) % array asn

B[j] = x;

assert(B[i] = B0[j] ∧ B[j] = B0[i]) % array asn

% VC 1

∀k . B[k] = B0[k] |−

( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[i] = B0[j] ∧

( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[j] = B0[i]

1) ∀k . B[k] = B0[k] premise

2) ( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[j] = B[i] by set % rel override ← this is a tautology

3) B[i] = B0[i] by forall\_e on 1

4) ( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[j] = B0[i] by eq\_e on 2, 3

5) (i = j) ∨ !(i = j) by lem

6) case (i = j) {

7) ( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[i] = B0[j] by eq\_e on 4, 6

}

8) case !(i = j) {

9) ( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[i] = ( B ⊕ {(i, B[j])} )[i] by set % rel override

10) ( B ⊕ {(i, B[j])} )[i] = B[j] by set % rel override

11) ( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[i] = B[j] by eq\_e on 9, 10

% 9, 10, 11 can be done in one step

12) B[j] = B0[j] by forall\_e on 1

13) ( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[i] = B0[j] by eq\_e on 11, 12

}

14) ( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[i] = B0[j] by cases on 5, 6-7, 8-13

15) ( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[i] = B0[j] ∧ ( (B ⊕ {(i, B[j])}) ⊕ {(j, B[i])} )[j] = B0[i] by and\_i …