Correctness Statements Proved in Class

Stefan D Bruda

18 March 2026

1. Some random loop:

```
ASSERT(true)
i = 0;
j = 100;
while (i <= 100) {
    i = i+1;
    j = j-1;
}
ASSERT(i == 101 && j == -1)
```

   Proof:

```
ASSERT(true)
ASSERT(0 <= 101 && 0 == 0)
i = 0;
ASSERT(i <= 101 && i == 0)
ASSERT(i <= 101 && i+100 == 100)
j = 100;
ASSERT(i <= 101 && i+j == 100)
while (i <= 100) {
    ASSERT(i <= 101 && i+j == 100 && i <= 100)
    ASSERT(i+1 <= 101 && i+j == 100 && i <= 100)
    ASSERT(i+1 <= 101 && i+j == 100)
    ASSERT(i+1 <= 101 && i+1+j-1 == 100)
    i = i+1;
    ASSERT(i <= 101 && i+j-1 == 100)
    j = j-1;
    ASSERT(i <= 101 && i+j == 100)
}
ASSERT(i <= 101 && i+j == 100 && i > 100)
ASSERT(i <= 101 && j == -1 && i > 100)
ASSERT(i == 101 && j == -1 && i > 100)
ASSERT(i == 101 && j == -1)
```

2. Swap two indices in an array using extra space:

```
ASSERT(A[i]==A0[i] && A[j]==A0[j])
int z;
z = A[i];
A[i] = A[j];
A[j] = z;
ASSERT(A[i]==A0[j] && A[j]==A0[i])
```

In the first proof we tried to eliminate the array component notations as soon as they were introduced, which is generally recommended. I botched the proof in class, here is the correct version:

```
ASSERT(A[i]==A0[i] && A[j]==A0[j])
ASSERT( A[j] == A0[j] && A[i]==A0[i] )
ASSERT( true && A[j] == A0[j] && A[i]==A0[i] )
ASSERT( (i == j || i != j) && A[j] == A0[j] && A[i]==A0[i] )
// common factor
ASSERT( i == j && A[j] == A0[j] && A[i]==A0[i]||
        i != j && A[j] == A0[j] && A[i]==A0[i] )
// the case i != j is fine, we convert the other by replacing first i with j
// (which is logically fine since i == j)
ASSERT( i == j && A[i] == A0[j] && A[i]==A0[i]||
        i != j && A[j] == A0[j] && A[i]==A0[i] )
// to get a clear picture, distribute the A[i]==A0[i] term inside the disjunction
ASSERT( ( i == j && A[i] == A0[j] || i != j && A[j] == A0[j] ) && A[i]==A0[i] )
int z;
ASSERT( ( i == j && A[i] == A0[j] || i != j && A[j] == A0[j] ) && A[i]==A0[i] )
z = A[i];
ASSERT( ( i == j && z == A0[j] || i != j && A[j] == A0[j] ) && z==A0[i] )
ASSERT( ( i == j && z == A0[j] || i != j && (A | i -> A[j])[i] == A0[j] ) && z==A0[i] )
// A -> (A | i -> A[j])
A[i] = A[j];
ASSERT( ( i == j && z == A0[j] || i != j && A[i] == A0[j] ) && z==A0[i] )
ASSERT((A | j -> z)[i]==A0[j] && z==A0[i])
ASSERT((A | j -> z)[i]==A0[j] && (A | j -> z)[j]==A0[i])
// A -> (A | j -> z)
A[j] = z;
ASSERT(A[i]==A0[j] && A[j]==A0[i])
```

The second proof only eliminate the array component notation at the very end:

```
ASSERT(A[i]==A0[i] && A[j]==A0[j])
ASSERT( A[j]==A0[j] && A[i]==A0[i] )
ASSERT( ( i == j || i != j ) && A[j]==A0[j] && A[i]==A0[i] )
ASSERT( i == j && A[j]==A0[j] && A[i]==A0[i] ||
        i != j && A[j]==A0[j] && A[i]==A0[i] )
```

```
ASSERT( i == j && A[i]==A0[j] && A[i]==A0[i] ||
        i != j && A[j]==A0[j] && A[i]==A0[i] )
ASSERT(((A | i -> A[j]) | j -> A[i])[i]==A0[j] && A[i]==A0[i])
int z;
ASSERT(((A | i -> A[j]) | j -> A[i])[i]==A0[j] && A[i]==A0[i])
z = A[i];
ASSERT(((A | i -> A[j]) | j -> z)[i]==A0[j] && z==A0[i])
A[i] = A[j];
ASSERT((A | j -> z)[i]==A0[j] && z==A0[i])
ASSERT((A | j -> z)[i]==A0[j] && (A | j -> z)[j]==A0[i])
A[j] = z;
ASSERT(A[i]==A0[j] && A[j]==A0[i])
```

You decide which version is easier. I personally prefer to eliminate the array component notation as soon as possible, but this is ultimately a matter of personal preference.