A ==> B <==> !A||B

**A ==> true <==> true**

A ==> false <==> !A

**true ==> B <==> B**

false ==> B <==> true

**A ==> (B ==> C) <==> (A && B) ==> C**

wp(t := M(E), Q) = P[x\E] && forall y’ :: R[x, y\E, y’] ==> Q[t\y’]^

**(forall y :: y == E ==> P) <==> P[y\E]**

(A && B) || C <==> (A || C) && (B || C)

^ -

method Triple(x: int) returns (y: int)

requires x >= 0

ensures y == 3 \* x

{u == 15}

{u + 3 >= 0 && 3 \* (u + 3) == 54}

{u + 3 >= 0 && forall y’ :: y’ == 3 \* ( u + 3) ==> y’ == 54}

t := Triple(u + 3);

{t == 54}

Representation set: ghost var Repr: set<object>

**Invariant:**

predicate Valid()

reads this, Repr

ensures Valid() ==> this in Repr

{

this in Repr && …

}

a in Repr && a.Valid()

b in Repr && b.Repr <= Repr && this !in b.Repr && b.Valid()

a0 != a1 && {a0, a1} !! b0.Repr !! b1.Repr

where a, a0, a1 are objects with simple frames

b, b0, b1 are objects with dynamic frames

**Constructor:**

constructor()

ensures Valid() && fresh(Repr)

{

…

new;

Repr := {this, a, b} + b.Repr;

}

**Functions:**

function F(x: X): Y

requires Valid()

reads Repr

**Methods (Mutating):**

method M(x: X) returns (y: Y)

requires Valid()

modifies Repr

ensures Valid() && fresh(Repr – old(Repr))

When looping over some array and checking each element against some constraint ‘P’ (such as max element), make sure to check (as an invariant) that all elements that have already been checked through previous loop iterations did not agree with the constraint and the variable housing the element to be returned contains an element that IS IN the array (as an invariant) where necessary.

When looping over a 2d array (matrix) and modifying it through some constant expression (copying, incrementing, …):

* Outer loop:
  + First invariant checks iterator against array length constraints
  + Second checks all already modified elements in whole matrix
  + Third checks all elements not modified in whole matrix
  + Fourth is termination metric
* Inner loop:
  + First invariant checks iterator against array length constraints
  + Second checks all already modified elements in whole matrix
  + Third checks all elements not modified in whole matrix
  + Fourth checks all already modified elements in just this loop array
  + Fifth checks all elements not modified in just this loop array
  + Sixth is termination metric