19CSE205 – Program Reasoning

Assignment – 3

S. Abhishek

AM. EN. U4 CSE19147

1. **Deduce the weakest preconditions for the following programs. Show the step by step working to get full marks.**
   1. **Increment a number**

|  |  |
| --- | --- |
| P | a < 4 \/ a > 9 |
| S | a = a + 1 |
| Q | **a < 5 \/ a > 10** |

S --> a = a + 1

Q --> a < 5 \/ a > 10

P -->?

The Post-Condition Q is a < 5 \/ a > 10

* wp (a = a + 1, a < 5 \/ a > 10)
* a + 1 < 5 \/ a + 1 > 10
* a < 5 – 1 \/ a > 10 – 1
* a < 4 \/ a > 9

The Pre-Condition P is a < 4 \/ a > 9

* 1. **Absolute value**

|  |  |
| --- | --- |
| P | x >= 0 \/ x < 0 |
| S | if (x >= 0)  abs = x;  else  abs = -1 \* x; |
| Q | **abs >= 0** |

**S -->**

if (x >= 0)

abs = x;

else

abs = -1 \* x;

**Data:**

B --> x >= 0

S1 --> abs = x;

S2 --> abs = -1\*x;

Q --> abs >= 0

P -->?

**We know that:**

* wp (If B then S1 else S2, Q) 🡪 B /\ wp (S1, Q) V -B /\ wp (S2, Q)
* x >= 0 /\ wp (abs = x, abs >= 0) \/ x < 0 /\

wp (abs = -1\*x, abs >= 0)

* x >= 0 /\ wp (abs = x, abs >= 0) \/ x < 0 /\ -1\*x >= 0
* x >= 0 /\ wp (abs = x, abs >= 0) \/ x < 0 /\ x < 0
* x >= 0 /\ x >= 0 \/ x < 0
* x >= 0 \/ x < 0
  1. **Swapping two integers**

|  |  |
| --- | --- |
| P | a == 5 /\ b == 7 |
| S | temp = a;  a = b;  b = temp |
| Q | **a == 7 /\ b == 5** |

The Post-Condition Q is a == 7 /\ b ==5

* wp (temp = a; a = b; b = temp, a == 7 /\ b == 5)
* wp (temp = a; a = b, wp (b = temp, a == 7 /\ b == 5))
* wp (temp = a; a = b, a == 7 /\ temp == 5)
* wp (temp = a, wp (a = b, a == 7 /\ temp == 5))
* wp (temp = a, b == 7 /\ temp == 5)
* b ==7 /\ a == 5

The Pre-Condition P is a == 5 /\ b == 7

* 1. **Operations on a stack**

|  |  |
| --- | --- |
| P | size >= 2 |
| S | Pop the topmost element; size = size - 1;  Push element to stack; size = size + 1;  Pop the topmost element; size = size - 1;  Push element to stack; size = size + 1;  Push element to stack; size = size + 1;  Push element to stack; size = size + 1;  Pop the topmost element; size = size - 1; |
| Q | **size >= 3** |

**Post Condition - Size >= 3.**

* wp (size = size - 1; size = size + 1; size = size - 1; size = size + 1;

size = size + 1; size = size + 1; size = size – 1, size >= 3)

* wp (size = size - 1; size = size + 1; size = size - 1; size = size + 1;

size = size + 1; size = size + 1, wp (size = size – 1, size >= 3))

* wp (size = size - 1; size = size + 1; size = size - 1; size = size + 1;

size = size + 1; size = size + 1, size – 1 >= 3)

* wp (size = size - 1; size = size + 1; size = size - 1; size = size + 1;

size = size + 1; size = size + 1, size >= 4)

* wp (size = size - 1; size = size + 1; size = size - 1; size = size + 1;

size = size + 1, wp (size = size + 1, size >= 4))

* wp (size = size - 1; size = size + 1; size = size - 1; size = size + 1,

size = size + 1; size + 1>= 4)

* wp (size = size - 1; size = size + 1; size = size - 1; size = size + 1,

size = size + 1; size >= 4 - 1)

* wp (size = size - 1; size = size + 1; size = size - 1; size = size + 1,

size = size + 1; size >= 3)

* wp (size = size - 1; size = size + 1; size = size - 1; size = size + 1,

wp (size = size + 1, size >= 3))

* wp (size = size - 1; size = size + 1; size = size - 1; size = size + 1,

size + 1 >= 3)

* wp (size = size - 1; size = size + 1; size = size - 1; size = size + 1,

size >= 3 - 1)

* wp (size = size - 1; size = size + 1; size = size - 1; size = size + 1,

size >= 2)

* wp (size = size - 1; size = size + 1; size = size – 1,

wp (size = size + 1, size >= 2))

* wp (size = size - 1; size = size + 1; size = size – 1, size + 1 >= 2)
* wp (size = size - 1; size = size + 1; size = size – 1, size >= 2 - 1)
* wp (size = size - 1; size = size + 1; size = size – 1, size >= 1)
* wp (size = size - 1; size = size + 1; wp (size = size – 1, size >= 1)
* wp (size = size - 1; size = size + 1; size – 1 >= 1)
* wp (size = size - 1; size = size + 1; size >= 1 + 1)
* wp (size = size - 1; size = size + 1; size >= 2)
* wp (size = size - 1; wp (size = size + 1, size >= 2))
* wp (size = size – 1, size + 1 >= 2)
* wp (size = size – 1, size >= 2 - 1)
* wp (size = size – 1, size >= 1)
* size – 1 >= 1
* size >= 1 + 1
* size >= 2

**Pre - Condition: size >= 2**

**2. Fill the following truth table to show that B => S1 /\ -B => S2 is equivalent to (B /\ S1) \/ (-B /\ S2)**

**(B => S1) /\ (-B => S2)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| B | S1 | B => S1 | -B | S2 | -B => S2 | B => S1 /\ -B => S2 |
| T | T | T | F | T | T | T |
| T | T | T | F | F | T | T |
| T | F | F | F | T | T | F |
| T | F | F | F | F | T | F |
| F | T | T | T | T | T | T |
| F | T | T | T | F | F | F |
| F | F | T | T | T | T | T |
| F | F | T | T | F | F | F |

**(B /\ S1) \/ (-B /\ S2)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| B | S1 | -B | S2 | B /\ S1 | -B /\ S2 | (B /\ S1) \/ (-B /\ S2) |
| T | T | F | T | T | F | T |
| T | T | F | F | T | F | T |
| T | F | F | T | F | F | F |
| T | F | F | F | F | F | F |
| F | T | T | T | F | T | T |
| F | T | T | F | F | F | F |
| F | F | T | T | F | T | T |
| F | F | T | F | F | F | F |

By the truth table we have proved that (B => S1) /\ (-B => S2) is equivalent to (B /\ S1) \/ (-B /\ S2).