| Formula | | | | | | | | | | | | Variables | |
|--|--------------------------------|---------------------------------------|--|--|--|---|---|--|--|---|---|--|--|
| precondition statement postcondition | | | | | | | | | | | LOCAL int[] tmp LOCAL int i | | |
| {true} | | | | | | | {containsNewTop(data, newTop) | | | | | RAM int newTop | |
| (true) | | | | statement | | | | & containsOldElements(data, \old(data))} | | | | UBLIC int[] data | |
| Statement1 | | | | | | | | | | | | obal Conditions data != null | |
| precondition | statement | postcondit | ion | Composition | | | | | | ata.length >= 0 i >= 0 old(data) = data | | | |
| (truo) | tmp = new | {tmp.lengt data.lengt 1} | | precondition | | | postcondition | | | | tmp != null | | |
| {true} | int[data.lengtl +1]; | | | {true} | | | {containsNewTop(data, newTop) & containsOldElements(data, \old(data))} | | | | | | |
| | | | , | statement 1 interme | | | ate condition statement 2 | | | t 2 | | | |
| | | | | statement1 | | {tmp.length = data.l | | ngth + 1} statemen | | t2 | | | |
| Statement | 2 | | | | | | | | | | | | |
| precondition statement postconditi | | | | Composition | | | | | | | | ✓ ` | |
| | | {tmp.lengt data.lengt | | | precondition | | | postcon | | | ndition | | |
| data length + | tmp[tmp.lengt h-1] = newTop | 1 2 8 | | {tmp.length = data.length + 1 | | | | | | ainsNewTop(data, newTop) sOldElements(data, \old(data))} | | | |
| ., | | h-1] = | | statemen | | | intermediate | | ondition | stateme | | ment 2 | |
| newTop} | | | statemer | nt1 | | | | data.length + 1 gth-1] = newTop} | | statement2 | | | |
| Composition | | | | | | | | | sition | V | * | | |
| precondition | | | | postcondition | | | precondition | | | | postcondition | | |
| | | | | {tmp.length = data.length + 1 & tmp[tmp.length-1] = newTop orall int i; (0 <= i & i < data.length - > data[i] = tmp[i]))} | | | {tmp.length = data.length + 1 & tmp[tmp.length-1] = newTop} | | | {containsNewTop(data, newTop) & containsOldElements(data, \old(data))} | | | |
| statement 1 intermediate c | | | | | statement 1 | | | intermediate cond | | statement 2 | | | |
| statement1 & (\fo | | 1& tmp[tmp new k (\forall int j | Top (0 <= j & j < tmp[j])) & i = | | statement2 | | statement1 | | {tmp.length = d & tmp[tmp.l new] & (\forall int i; data.length - tmp[| ength-1] = [op (0 <= i & i < > data[i] = | statement2 | | |
| Repetition | Statemer | nt5 | | V | | | | Sta | tement5 | | L | <u> </u> | |
| Repetition | | | etition State | on Statement DOOD | | | ~ | precondition | | statement | | postcondition | |
| invariant tmp.length = data.length + 1 & tmp[tmp.length-1] = newTop & (\forall int j; (0 <= j & j < i -> data[j] = tmp[j])) precondition | | p\& | i < data.length | | | variant data.length - i postcondition | | | {tmp.length = data.length + 1 & tmp[tmp.length-1] = newTop & (\forall int i; (0 <= i & i < data.length - > data[i] = tmp[i]))} | | mp; | {containsNewTop(da ta, newTop) & containsOldElements (data, \old(data))} | |
| {(tmp.length = data.length + 1 & tmp[tmp.length-1] = newTop & | | | | {tmp.len | | | ength + 1 & | Statement4 | | | | | |
| tmp[tmp.length-1] = newTop & (\forall int j; (0 <= j & j < i -> data[j] = tmp[j]))) & (i < | | -> | loop | | | tmp[tmp.length-1] = newTop & (\forall int j; (0 <= j & j < i -> data[j] = tmp[j]))} | | | precondition statem | | ent | postcondition | |
| Statement | n.length)} ·3 | | | | | | | | tmp.length = | | | {tmp.length = | |
| precondition | | | statement | | postcondition | | | data.length + 1 & tmp[tmp.length-1] = newTop & (\forall int j; (0 <= j & j < i -> data[j] = tmp[j]))) & (i < data.length)} | | data.length + 1 & tmp[tmp.length-1] = | | | |
| {tmp.length = data.length + 1 & tmp[tmp.length-1] = newTop} | | | i = 0; | | {tmp.length = data.length+ 1& tmp[tmp.length-1] = newTop & (\forall int j; (0 <= j & j < i -> data[j] = tmp[j])) & i = 0} | | | | | | j; (0 <= j & j < i - data[j] = tmp[j]) | | |