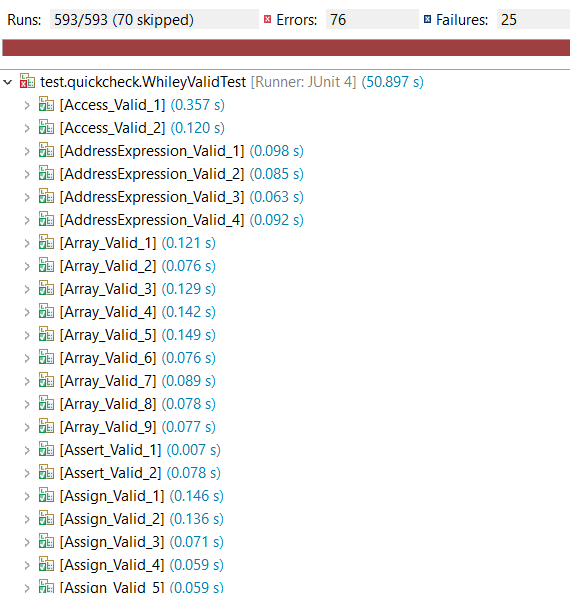
# Whiley Valid Test Results

Total number of tests: 593 tests.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Failed | Passed | Skipped | Executed |
| 25/05/18  No optimisation | 76 errors, 25 failures | 422 | 70 | 523 |
| 26/06/18  No optimisation | 75 errors, 88 failures (only 16 failures, others due to precondition) | 360 (432) | 70 | 523 |
| 26/06/18  Optimisation | 97 errors, 78 failures (only 17 failures, others due to precondition) | 348 (409) | 70 | 523 |



## Needs tweaking of limits but passed

72 tests, due to precondition failing on some or all inputs

|  |  |  |
| --- | --- | --- |
| Access\_Valid\_1 | Ensures\_Valid\_1 | While\_Valid\_12 |
| Access\_Valid\_2 | Ensures\_Valid\_4 | While\_Valid\_22 |
| Array\_Valid\_1 | Function\_Valid\_16 | While\_Valid\_23 |
| Array\_Valid\_3 | FunctionRef\_Valid\_4 | While\_Valid\_28 |
| Array\_Valid\_4 | Import\_Valid\_2 | While\_Valid\_30 |
| Array\_Valid\_5 | IntDiv\_Valid\_1 | While\_Valid\_33 |
| Array\_Valid\_6 | IntDiv\_Valid\_5 | While\_Valid\_38 |
| Array\_Valid\_7 | ListAccess\_Valid\_1 | While\_Valid\_43 |
| Array\_Valid\_8 | ListAccess\_Valid\_3 | While\_Valid\_45 |
| Array\_Valid\_9 | ListAssign\_Valid\_3 | While\_Valid\_47 |
| BoolAssign\_Valid\_3 | ListAssign\_Valid\_5 | While\_Valid\_49 |
| BoolAssign\_Valid\_5 | ListAssign\_Valid\_11 | While\_Valid\_50 |
| BoolRequires\_Valid\_1 | ListAssign\_Valid\_12 | While\_Valid\_52 |
| ConstrainedInt\_Valid\_21 | ListGenerator\_Valid\_1 | While\_Valid\_55 |
| ConstrainedList\_Valid\_1 | ListGenerator\_Valid\_3 | While\_Valid\_56 |
| ConstrainedList\_Valid\_2 | Property\_Valid\_7 | While\_Valid\_59 |
| ConstrainedList\_Valid\_7 | Quantifiers\_Valid\_1 | While\_Valid\_60 |
| ConstrainedList\_Valid\_9 | RecordAssign\_Valid\_9 | While\_Valid\_61 |
| ConstrainedList\_Valid\_16 | RecordAssign\_Valid\_14 | While\_Valid\_62 |
| ConstrainedList\_Valid\_23 | Remainder\_Valid\_1 | While\_Valid\_63 |
| ConstrainedList\_Valid\_25 | Requires\_Valid\_1 | While\_Valid\_64 |
| ConstrainedList\_Valid\_28 | Switch\_Valid\_3 | While\_Valid\_65 |
| DoWhile\_Valid\_5 | Switch\_Valid\_7 | While\_Valid\_66 |
| DoWhile\_Valid\_6 | Update\_Valid\_2 | While\_Valid\_67 |
|  | VarDecl\_Valid\_2 | While\_Valid\_68 |

## Failed postcondition

* Assume\_Valid\_1
  + Failing on negative values, due to assumption on value on line 8 causes output type invariant to fail so not quickcheck’s fault!
* While\_Valid\_58\_result
  + Failing on negative values, due to assumption on value on line 2 which causes assertion failure on line 8 so not quickcheck’s fault!

## Assertion Failure

25 assertion failures as of 25/5/18.

14 assertion failures, 74 precondition failures as of 26/6/18.

|  |  |
| --- | --- |
| Reason | Test |
| No function (9 tests) | * Lambda\_Valid\_9 * OpenRecord\_Valid\_3 * FunctionRef\_Valid\_1, 2, 5, 6, 8, 12, 13 * Function\_Valid\_13 (optimisation only) |
| Fail to compile (4 tests)  Due to intersection type | * ConstrainedIntersection\_Valid\_1 (for the *is* clause, expected type byte found bool) * Intersection\_Valid\_1, 2 (intersections don’t work) * NegationType\_Valid\_3 (intersections don’t work) |
| Assertion error (1 tests) | * ConstrainedList\_Valid\_9 |
| Precondition failure (72 tests) | * See Table above |

## Error

76 errors as of 25/5/18

75 errors as of 26/6/18

|  |  |
| --- | --- |
| Reason | Test |
| StackOverflow (12 tests)    Due to recursive types? | * Coercion\_Valid\_8 (recursive) * ConstrainedInt\_Valid\_20 (function is recursive, never ends if input < 0) * Contractive\_Valid\_1, 3 (recursive) * FunctionRef\_Valid\_10, 11 (recursive + function) * RecursiveType\_Valid\_20, 23, 24, 26 (recursive) * TypeEquals\_Valid\_16, 2 (recursive) |
| Upper limit <= lower limit  (58 tests)  Just need to tweak limits | * Complex\_Valid\_1, 11, 5 * ConstrainedInt\_Valid\_10, 13, 16, 17, 19, 23, 24, 6, 8 * ConstrainedList\_Valid\_17, 19, 2, 20, 21, 22, 3, 6, 8 * ConstrainedRecord\_Valid\_1, 2 * Define\_Valid\_4 * Fail\_Valid\_3 * Function\_Valid\_18, 20, 4, 6, 8 * Lambda\_Valid\_7 * ListAssign\_Valid\_4 * Method\_Valid\_1 * RecordAssign\_Valid\_7 * RecordDefine\_Valid\_1, 3 * Record\_Valid\_5 * String\_Valid\_1 * Switch\_Valid\_4 * TypeEquals\_Valid\_1, 3, 31, 33, 39, 45 * UnionType\_Valid\_11, 14, 18, 2, 20, 22, 5, 7, 9 * While\_Valid\_16, 18, 21, 3 |
| RuntimeException (5 tests, 3 optimisation only) | * ConstrainedList\_Valid\_11 - Null operand * ConstrainedRecord\_Valid\_6 – null operand * RecordAssign\_Valid\_13, 4 – null operand * RecursiveType\_Valid\_17 – null operand * While\_Valid 55, 56, 57 – null operand (optimisation only) |
| Class Cast exception – trying to cast Decl.Property to Decl.FunctionOrMethod  (Optimisation only, 18 tests) | * DoWhile\_Valid\_9 * Property\_Valid\_1, 2, 3, 5, 6, 8, 9, 13 * While\_Valid\_59, 60, 62, 63, 64, 65, 66, 67 |
| ArrayIndexOutOfBounds  (Optimisation only, 1 test) | * RecursiveType\_Valid\_16 |

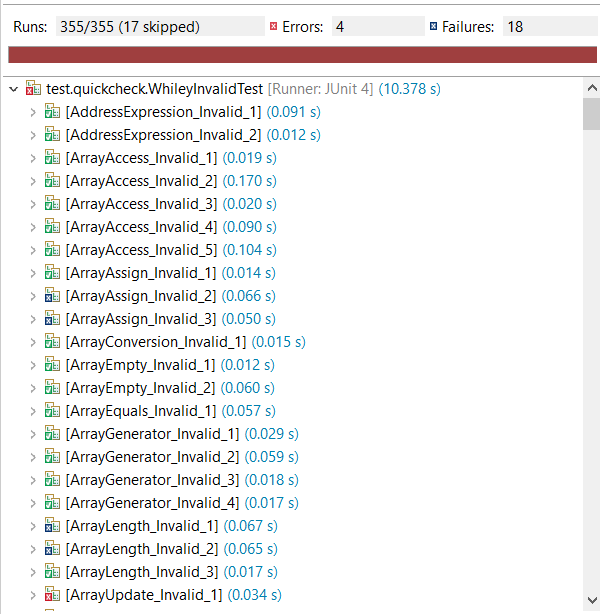
May be of interest (currently disabled)

|  |  |
| --- | --- |
| NegativeArraySizeException | * While\_Valid\_2, 26   Due to executeArrayRange not checking end – start > 0 |

# Whiley Invalid Test Results

Total number of tests: 355 tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Failed | Passed | Skipped | Total Executed |
| 25/05/18 | 4 errors, 18 failures | 316 | 17 | 338 |
| 27/06/18 | 7 errors | 331 | 17 | 338 |
| 27/06/18 | 18 errors | 320 | 17 | 338 |



Some tests could be passing due to the loop limits used.

|  |  |
| --- | --- |
| Reason | Test |
| ArrayIndexOutOfBounds is good? (3 tests, 2 optimisation, ) | * ArrayUpdate\_Invalid\_1 * Array\_Invalid\_10 (no optimisation) * ConstrainedInt\_Invalid\_7,8 (optimisation) * While\_Invalid\_10 |
| Class Cast exception – trying to cast Decl.Property to Decl.FunctionOrMethod  (Optimisation only, 12 tests) | * Do\_While\_Invalid\_6, 7, 8 * Property\_Invalid\_2, 3, 4, 8 * While\_Invalid\_19, 20, 21, 22, 23 |
| DivisionByZero is good? (3 test) | * IntDiv\_Invalid\_1 (no optimisation) * Remainder\_Invalid\_1 * VarDecl\_Invalid\_2 (no optimisation) |
| RuntimeException (1 test) | * TupleAssign\_Invalid\_2 – null operand (technically ok as it removes invalid inputs during test gen) |