# Whiley Valid Test Results

Total number of tests: 593 tests.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Failed | Passed | Skipped | Executed | Execution Time |
| 25/05/18  No optimisation | 76 errors, 25 failures | 422 | 70 | 523 | None |
| 26/06/18  No optimisation | 75 errors, 88 failures (only 16 failures, others due to precondition) | 360 (432) | 70 | 523 | 43.404s  44.723s  43.662s  44.068s  43.709s  = 43.913s |
| 26/06/18  With Optimisation | 97 errors, 78 failures (only 17 failures, others due to precondition) | 348 (409) | 70 | 523 | 42.457s  42.6s  43.003s  43.092s  42.911s  = 42.813s |
| 03/07/18  No optimisation | 17 errors, 102 failures (only 21 failures, others due to preconditions) | 404 (485) | 70 | 523 | 81.295s |
| 03/07/18  With Optimisation | 39 errors, 101 failures (only 31 failures)  RecursiveType\_Valid\_27 (156.165 s) | 383 (450) | 70 | 523 | 208.615s |
| 15/07/18  No optimisation | 10 errors 104 failures (only 23 failures)  Record\_valid\_5 (25 s) | 409 (490) | 70 | 523 | 97.292s |
| 15/07/18  With optimisation | 10 errors 105 failures (only 24 failures)  Record\_valid\_5 (20 s) | 408 (489) | 70 | 523 | 74.108s |

## Needs tweaking of limits but passed

81 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 | Lambda\_Valid\_7 | While\_Valid\_43 |
| Array\_Valid\_8 | ListAccess\_Valid\_1 | While\_Valid\_45 |
| Array\_Valid\_9 | ListAccess\_Valid\_3 | While\_Valid\_47 |
| BoolAssign\_Valid\_3 | ListAssign\_Valid\_3 | While\_Valid\_49 |
| BoolAssign\_Valid\_5 | ListAssign\_Valid\_4 | While\_Valid\_50 |
| BoolRequires\_Valid\_1 | ListAssign\_Valid\_5 | While\_Valid\_52 |
| ConstrainedInt\_Valid\_6 | ListAssign\_Valid\_11 | While\_Valid\_55 |
| ConstrainedInt\_Valid\_10 | ListAssign\_Valid\_12 | While\_Valid\_56 |
| ConstrainedInt\_Valid\_21 | ListGenerator\_Valid\_1 | While\_Valid\_59 |
| ConstrainedList\_Valid\_1 | ListGenerator\_Valid\_3 | While\_Valid\_60 |
| ConstrainedList\_Valid\_2 | Property\_Valid\_7 | While\_Valid\_61 |
| ConstrainedList\_Valid\_7 | Quantifiers\_Valid\_1 | While\_Valid\_62 |
| ConstrainedList\_Valid\_8 | RecordAssign\_Valid\_9 | While\_Valid\_63 |
| ConstrainedList\_Valid\_9 | RecordAssign\_Valid\_14 | While\_Valid\_64 |
| ConstrainedList\_Valid\_16 | Remainder\_Valid\_1 | While\_Valid\_65 |
| ConstrainedList\_Valid\_17 | Requires\_Valid\_1 | While\_Valid\_66 |
| ConstrainedList\_Valid\_20 | Switch\_Valid\_3 | While\_Valid\_67 |
| ConstrainedList\_Valid\_23 | Switch\_Valid\_7 | While\_Valid\_68 |
| ConstrainedList\_Valid\_25 | Update\_Valid\_2 |  |
| ConstrainedList\_Valid\_28 | VarDecl\_Valid\_2 |  |
| DoWhile\_Valid\_5 |  |  |
| DoWhile\_Valid\_6 |  |  |

## Assertion Failure

|  |  |
| --- | --- |
| Reason | Test |
| No function (11 tests) | * Lambda\_Valid\_9 * OpenRecord\_Valid\_3 * FunctionRef\_Valid\_1, 2, 5, 6, 8, 10, 11, 12, 13 |
| 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) |
| Precondition failure (81 tests) | * See Table above |
| Upper limit > lower limit (for the negative and positive tests) (6 tests) | * TypeEquals\_Valid\_33, 45 * UnionType\_Valid\_5, 9, 11, 20 |
| Failed postcondition (2 tests due to assumptions) | * 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! |
| Failing test (optimisation) | * Function\_Valid\_13 (optimisation only, positive limit) |

## Error

|  |  |
| --- | --- |
| Reason | Test |
| StackOverflow (1 test)    Due to recursive types (arrays and records within records) | * ConstrainedInt\_Valid\_20 (function is recursive, never ends if input < 0) |
| No possible values generated due to nominal invariant (5 tests) | * TypeEquals\_Valid\_16 * Coercion\_Valid\_8 (recursive) * Contractive\_Valid\_1 (recursive) * RecursiveType\_Valid\_24, 26 (recursive) |
| Upper limit <= lower limit  (4 tests)  Just need to tweak limits | * Complex\_Valid\_1, 11, 5 * Method\_Valid\_1 |

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 | Execution Time |
| 25/05/18 | 4 errors, 18 failures | 316 | 17 | 338 | None |
| 27/06/18 (no optimisation) | 4 errors | 331 | 17 | 338 | 9.459s  9.06s  9.354s  9.497s  9.045s  = 9.283 s |
| 27/06/18 (With optimisation) | 18 errors | 320 | 17 | 338 | 8.764s  8.671s  8.607s  8.779s  8.623s  = 8.689s |
| 05/07/18 (no optimisation) | 3 errors (6 if ignoring preconditions) | 336 | 17 | 339 | 11.264s |
| 05/07/18 (With optimisation) | 15 errors (18 errors if ignoring precondition) | 323 | 17 | 339 | 9.943s |
| 15/07/18 (no optimisation) | 3 errors (6 if ignoring preconditions) | 323 | 17 | 339 | 10.665s |
| 15/07/18 (With optimisation) | 3 errors (6 if ignoring preconditions) | 323 | 17 | 339 | 11.154s |

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

|  |  |
| --- | --- |
| Reason | Test |
| ArrayIndexOutOfBounds is good? (2 tests) | * ArrayUpdate\_Invalid\_1 * While\_Invalid\_10 * Array\_Invalid\_10 (only if ignore precondition fail) |
| DivisionByZero is good? (1 test) | * Remainder\_Invalid\_1 * VarDecl\_Invalid\_2 (only if ignore precondition fail) * IntDiv\_Invalid\_1 (only if ignore precondition fail) |

# Whiley Benchmark Tests

Total number of tests: 36 tests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Failed | Passed | Skipped | Executed | Execution Time |
| 2/07/18 | 5 errors 31 failures | 0 | 0 | 36 | None |
| 15/07/18  (No optimisation) | 9 errors, 15 failures (9 failures, others due to precondition) (004\_matrix takes 13 s) | 9 (15) | 1 | 33 | 41.18s  40.794s  40.812s  41.345s  39.091s  = 40.64s |
| 15/07/18  (With optimisation) | 9 errors, 15 failures (9 failures, others due to precondition) (004\_matrix takes 255 s) |  | 2  (skip bipmatch) | 32 | 327.53s |

## Assertion Failure

|  |  |
| --- | --- |
| Reason | Test |
| Due to precondition – otherwise tests do pass | * 001\_average\main (negative limits) * 004\_matrix\main (positive limits) * 006\_queens\main (positive limits) * 010\_sort\main (negative limits) (toString) * 012\_cyclic\main (positive limits) (toString) * 014\_lights\main (negative limits) (toString) * 016\_date\main (Bug in the limits of type day, should be 1 <= d) * 017\_math\main (positive limits) (toString) * 025\_tries\main (positive limits) (cannot generate ascii::string) * 026\_reverse\main (negative limits) (toString) * 102\_conway\main (positive limits) – Should throw error, technically correct as tried to access array out of bounds. * 022\_cars\main (positive limits) * 023\_microwave\main (positive limits) * 028\_flag\main (positive limits) * 030\_fractions\main (Bug in the compare function! In r > 0, should (f1.numerator \* f2.denominator) > (f2.numerator \* f1.denominator) ) |

## Error

|  |  |
| --- | --- |
| Reason | Test |
| NegativeArrayException  Not supported when array size is negative in the condition. Due to negative int range | * 018\_heap\main |
| ArrayIndexOutOfBounds  Due to empty array in m.left | * 029\_bipmatch\main |
| StackOverflowError  Due to ith.next being the same index (0) instead of the next array index | * 032\_arrlist\main |
| No values can be generated | * 008\_scc\main (as cannot generate collection::Stack due to State?) |
| Due to upper > lower | * 015\_cashtill\main * 104\_tictactoe\board * 104\_tictactoe\game |
| RuntimeException  Cannot find library | * 011\_codejam\main (parser::skipWhiteSpace) |
| NullPointerException | * 013\_btree\main (Can’t find elts input/output definition in typesystem). |