
Mull

Release 0.6.0

Alex Denisov <alex@lowlevelbits.org>, Stanislav Pankevich <s.pa

Dec 05, 2019

CONTENTS

1	Getting Started	1
2	Introduction into Mutation Testing	3
3	Supported Mutation Operators	5
4	Command Line Reference	7
5	Hacking On Mull	11

GETTING STARTED

INTRODUCTION INTO MUTATION TESTING

SUPPORTED MUTATION OPERATORS

Operator Name	Operator Semantics
cxx_add_assign_to_sub_assign	Replaces += with -=
cxx_add_to_sub	Replaces + with -
cxx_and_assign_to_or_assign	Replaces &= with =
cxx_and_to_or	Replaces & with
cxx_assign_const	Replaces 'a = b' with 'a = 42'
cxx_bitwise_not_to_noop	Replaces ~x with x
cxx_bitwise_rshift_assign_to_lshift_assign	Replaces >>= with <<=
cxx_div_assign_to_mul_assign	Replaces /= with *=
cxx_div_to_mul	Replaces / with *
cxx_eq_to_ne	Replaces == with !=
cxx_ge_to_gt	Replaces >= with >
cxx_ge_to_lt	Replaces >= with <
cxx_gt_to_ge	Replaces > with >=
cxx_gt_to_le	Replaces > with <=
cxx_init_const	Replaces 'T a = b' with 'T a = 42'
cxx_le_to_gt	Replaces <= with >
cxx_le_to_lt	Replaces <= with <
cxx_logical_and_to_or	Replaces && with
cxx_logical_or_to_and	Replaces with &&
cxx_lshift_assign_to_rshift_assign	Replaces <<= with >>=
cxx_lshift_to_rshift	Replaces << with >>
cxx_lt_to_ge	Replaces < with >=
cxx_lt_to_le	Replaces < with <=
cxx_minus_to_noop	Replaces -x with x
cxx_mul_assign_to_div_assign	Replaces *= with /=
cxx_mul_to_div	Replaces * with /
cxx_ne_to_eq	Replaces != with ==
cxx_or_assign_to_and_assign	Replaces = with &=
cxx_or_to_and	Replaces with &
cxx_post_dec_to_post_inc	Replaces x- with x++
cxx_post_inc_to_post_dec	Replaces x++ with x-
cxx_pre_dec_to_pre_inc	Replaces -x with ++x
cxx_pre_inc_to_pre_dec	Replaces ++x with -x
cxx_rem_assign_to_div_assign	Replaces %= with /=
cxx_rem_to_div	Replaces % with /
cxx_rshift_to_lshift	Replaces >> with <<
cxx_sub_assign_to_add_assign	Replaces -= with +=

Continued on next page

Table 1 – continued from previous page

Operator Name	Operator Semantics
cxx_sub_to_add	Replaces - with +
cxx_xor_assign_to_or_assign	Replaces ^= with =
cxx_xor_to_or	Replaces ^ with
negate_mutator	Negates conditionals !x to x and x to !x
remove_void_function_mutator	Removes calls to a function returning void
replace_call_mutator	Replaces call to a function with 42
scalar_value_mutator	Replaces zeros with 42, and non-zeros with 0

COMMAND LINE REFERENCE

--workers number How many threads to use

--dry-run Skips real mutants execution. Disabled by default

--cache-dir directory Where to store cache (defaults to /tmp/mull-cache)

--disable-cache Disables cache (enabled by default)

--report-name filename Filename for the report (only for supported reporters). Defaults to <times-tamp>.<extension>

--report-dir directory Where to store report (defaults to '.')

--reporters reporter Choose reporters:

- IDE** Prints compiler-like warnings into stdout
- SQLite** Saves results into an SQLite database
- Elements** Generates mutation-testing-elements compatible JSON file

--compdb-path filename Path to a compilation database (compile_commands.json) for junk detection

--compilation-flags string Extra compilation flags for junk detection

--ld-search-path directory Library search path

--include-path regex File/directory paths to whitelist (supports regex)

--exclude-path regex File/directory paths to ignore (supports regex)

--sandbox sandbox Choose sandbox approach:

- None** No sandboxing
- Watchdog** Uses 4 processes, not recommended
- Timer** Fastest, Recommended

--test-framework framework Choose test framework:

- GoogleTest** Google Test Framework
- CustomTest** Custom Test Framework
- SimpleTest** Simple Test (For internal usage only)

--mutators mutator Choose mutators:

Groups:

- all** cxx_all, experimental

cxx_all cxx_assignment, cxx_increment, cxx_decrement,
cxx_arithmetic, cxx_logical, cxx_comparison, cxx_boundary

cxx_arithmetic cxx_minus_to_noop, cxx_add_to_sub,
cxx_sub_to_add, cxx_mul_to_div, cxx_div_to_mul,
cxx_rem_to_div

cxx_arithmetic_assignment cxx_add_assign_to_sub_assign,
cxx_sub_assign_to_add_assign,
cxx_mul_assign_to_div_assign,
cxx_div_assign_to_mul_assign,
cxx_rem_assign_to_div_assign

cxx_assignment cxx_bitwise_assignment,
cxx_arithmetic_assignment, cxx_const_assignment

cxx_bitwise cxx_bitwise_not_to_noop, cxx_and_to_or,
cxx_or_to_and, cxx_xor_to_or, cxx_lshift_to_rshift,
cxx_rshift_to_lshift

cxx_bitwise_assignment cxx_and_assign_to_or_assign,
cxx_or_assign_to_and_assign, cxx_xor_assign_to_or_assign,
cxx_lshift_assign_to_rshift_assign,
cxx_bitwise_rshift_assign_to_lshift_assign

cxx_boundary cxx_le_to_lt, cxx_lt_to_le, cxx_ge_to_gt,
cxx_gt_to_ge

cxx_comparison cxx_eq_to_ne, cxx_ne_to_eq, cxx_le_to_gt,
cxx_lt_to_ge, cxx_ge_to_lt, cxx_gt_to_le

cxx_const_assignment cxx_assign_const, cxx_init_const

cxx_decrement cxx_pre_dec_to_pre_inc,
cxx_post_dec_to_post_inc

cxx_default cxx_increment, cxx_arithmetic, cxx_comparison,
cxx_boundary

cxx_increment cxx_pre_inc_to_pre_dec,
cxx_post_inc_to_post_dec

cxx_logical cxx_logical_and_to_or, cxx_logical_or_to_and

experimental negate_mutator, remove_void_function_mutator,
scalar_value_mutator, replace_call_mutator

Single mutators:

cxx_add_assign_to_sub_assign Replaces += with -=

cxx_add_to_sub Replaces + with -

cxx_and_assign_to_or_assign Replaces &= with |=

cxx_and_to_or Replaces & with |

cxx_assign_const Replaces 'a = b' with 'a = 42'

cxx_bitwise_not_to_noop Replaces ~x with x

cxx_bitwise_rshift_assign_to_lshift_assign Replaces >>= with
<<=

cxx_div_assign_to_mul_assign Replaces /= with *=

cxx_div_to_mul Replaces / with *
cxx_eq_to_ne Replaces == with !=
cxx_ge_to_gt Replaces >= with >
cxx_ge_to_lt Replaces >= with <
cxx_gt_to_ge Replaces > with >=
cxx_gt_to_le Replaces > with <=
cxx_init_const Replaces 'T a = b' with 'T a = 42'
cxx_le_to_gt Replaces <= with >
cxx_le_to_lt Replaces <= with <
cxx_logical_and_to_or Replaces && with ||
cxx_logical_or_to_and Replaces || with &&
cxx_lshift_assign_to_rshift_assign Replaces <<= with >>=
cxx_lshift_to_rshift Replaces << with >>
cxx_lt_to_ge Replaces < with >=
cxx_lt_to_le Replaces < with <=
cxx_minus_to_noop Replaces -x with x
cxx_mul_assign_to_div_assign Replaces *= with /=
cxx_mul_to_div Replaces * with /
cxx_ne_to_eq Replaces != with ==
cxx_or_assign_to_and_assign Replaces |= with &=
cxx_or_to_and Replaces | with &
cxx_post_dec_to_post_inc Replaces x- with x++
cxx_post_inc_to_post_dec Replaces x++ with x-
cxx_pre_dec_to_pre_inc Replaces -x with ++x
cxx_pre_inc_to_pre_dec Replaces ++x with -x
cxx_rem_assign_to_div_assign Replaces %= with /=
cxx_rem_to_div Replaces % with /
cxx_rshift_to_lshift Replaces >> with <<
cxx_sub_assign_to_add_assign Replaces -= with +=
cxx_sub_to_add Replaces - with +
cxx_xor_assign_to_or_assign Replaces ^= with |=
cxx_xor_to_or Replaces ^ with |
negate_mutator Negates conditionals !x to x and x to !x
remove_void_function_mutator Removes calls to a function returning void
replace_call_mutator Replaces call to a function with 42

scalar_value_mutator Replaces zeros with 42, and non-zeros with 0

HACKING ON MULL