

Mutation Testing with C++

Using C++ Mutation Test Environment.
Mutate++

https://github.com/nlohmann/mutate_cpp

Prepared by: Sergio Garcia & Jose Pastor

What is mutation testing?

- Is a method for improving software tests and consequently improve the software
- Mutation testing involves modifying (mutating) a program in small ways.
- If a test does not fail when testing a mutated software it has detected a problem in the test

Example of a SUT

```
#include "example.h"

double add_numbers(const double f1, const double f2)
{
    return f1 + f2;
}

double subtract_numbers(const double f1, const double f2)
{
    return f1 - f2;
}

double multiply_numbers(const double f1, const double f2)
{
    return f1*f2;
}
```

Example of Test

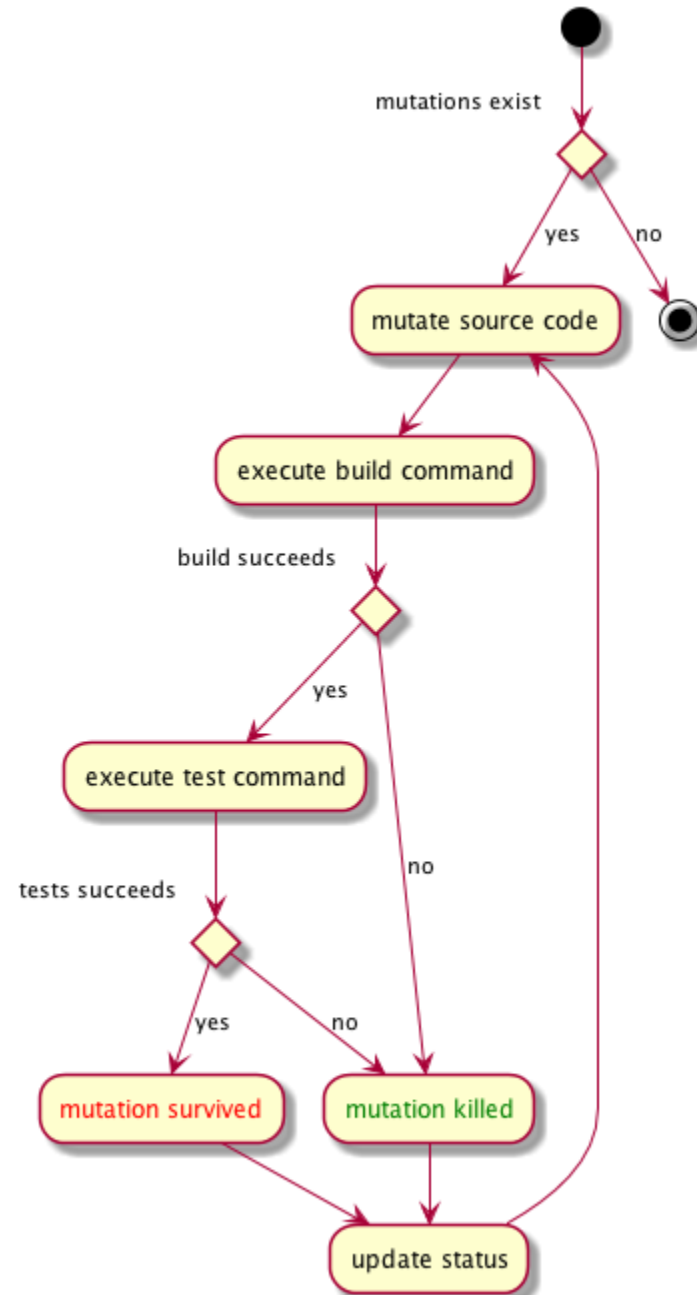
```
#include "gtest/gtest.h"
#include "example.h"

TEST(example, add)
{
    double res;
    res = add_numbers(1.0, 2.0);
    ASSERT_NEAR(res, 3.0, 1.0e-11);
}

TEST(example, add)
{
    double res;
    res = add_numbers(1.0, 2.0);
    ASSERT_NEAR(res, 3.0, 1.0e-11);
}
```

NOTE: Multiply operation not tested

https://github.com/nlohmann/mutate_cpp



Generate patches for file example.cpp

first line

first line

 $\left(\begin{array}{c} \bullet \\ \bullet \end{array} \right)$

last line

last line

$$\left(\begin{array}{c} \vdots \\ \vdots \end{array} \right)$$

- ☐ Deletes a whole line.
- ☐ Replaces logical operators.
- ☐ Replaces comparison operators.
- ☐ Swaps increment and decrement operators.
- ☐ Replaces assignment operators.
- ☐ Replaces Boolean assignment operators.
- ☐ Replaces arithmetic operators.
- ☐ Replaces Boolean arithmetic operators.
- ☐ Swaps the Boolean literals true and false.
- ☐ Changes the position where elements are inserted.
- ☐ Changes the semantics of an STL range predicate.
- ☐ Swaps STL minimum by maximum calls.

Patch 4

Patch

```
--- /private/tmp/cmake-example/src/example.cpp 2017-11-25 12:07:58.368779
+++ /private/tmp/cmake-example/src/example.cpp 2017-11-25 12:58:58.530461
@@ -3,7 +3,7 @@
```

```
double add_numbers(const double f1, const double f2)
{
-   return f1 + f2;
+   return f1 * f2;
}

double subtract_numbers(const double f1, const double f2)
```

Description

The patch is of kind [arithmeticOperator](#) and replaces arithmetic operators.

In line 6 of file example.cpp, **+** was replaced with *****.

The patch has not yet been investigated.

Confirmation



command	✓ failure	⚠ success	sum
⚙ build	5 runs 4.75 secs 0.95 secs/run	9 runs 9.79 secs 1.09 secs/run	14 runs 14.54 secs 1.04 secs/run
🔍 quickcheck	0 runs 0 secs 0 secs/run	0 runs 0 secs 0 secs/run	0 runs 0 secs 0 secs/run
🔍 test	8 runs 0.17 secs 0.02 secs/run	1 runs 0.02 secs 0.02 secs/run	9 runs 0.19 secs 0.02 secs/run
sum	13 runs 4.93 secs 0.38 secs/run	10 runs 9.81 secs 0.98 secs/run	23 runs 14.74 secs 0.64 secs/run