

Test your tests

Evaluate the quality of your tests and ultimately of your code.
March 2020

Pol Dellaiera
Analysis, development, research
PHPCC – DIGIT.B.4.003

The basics

The why, the what, and the how.

Why testing ?

Why testing ?

- Good practice,
- Please our colleagues,
- Ensure reliability across updates,
- Continuous learning mindset,
- To improve software internal quality,
 - The less you care, the harder it will be to add new features,
 - The more you delay, the more expensive it gets.

Why testing ?

«Take care of your codebase, and your codebase will take care of you when you're older.»

Tomas Votruba - [We Do Not Need Senior Developers, We Need Senior Code Bases](#)

What is a test ?

What is a test ?



What is a test ?

A procedure intended to establish the quality, performance, or reliability of something, especially before it is taken into widespread use.

In our context, a test is a piece of code that test specific code of an application.

What is a test ? Code example

```
<?php

function guessWhatIDo($number)
{
    return array_reduce(
        str_split((string) $number),
        static function ($carry, $digit) {
            return $carry + $digit;
        },
        0
    );
}

var_dump(guessWhatIDo(1234)); // ??
var_dump(guessWhatIDo('what is happening here ?')); // ??
```

What is a test ? Code example

<https://3v4l.org/FphES>

What is a test ? Code example

```
<?php
```

```
function testGuessWhatIDo($input, $expectedOutput): bool  
{  
    return assert(guessWhatIDo($input) === $expectedOutput);  
}
```

```
var_dump(testGuessWhatIDo(1234, 10));  
var_dump(testGuessWhatIDo(1234, 11));
```

What is a test ? Code example

<https://3v4l.org/fMfbd>

guessWhatIDo()

testGuessWhatIDo()

true

false



guessWhatIDo()



testGuessWhatIDo()



true



false

Testing frameworks

Overview

Testing frameworks

- PHPUnit
 - Unit testing framework.
- PHPSpec
 - Behavior Driven Development framework
- Behat
 - Behavior Driven Development framework
- Atoum
 - Unit testing framework

Tests metric ?

Metric

In mathematics, a metric or distance function is a function that defines a distance between each pair of elements of a set.

In our context, the test metric is a function that defines the "*distance*" between zero and "*the current amount <something> tested*".

<something> could be replaced with: line code coverage, type code coverage, mutation code coverage... there are many metrics available.

Line code coverage providers

- Xdebug
 - A PHP extension for debugging and profiling. ([website](#))
- PHPdbg
 - A command line PHP debugger. ([website](#))
- PCOV
 - A PHP extension. ([website](#))

Line code coverage metric

```
1  <?php
2
3  declare(strict_types=1);
4
5  namespace EcPhp\Factorial;
6
7  class Factorial
8  {
9      public function of($integer)
10     {
11         $factorial = 1;
12
13         foreach (range(1, $integer) as $n) {
14             $factorial *= $n;
15         }
16
17         return $factorial;
18     }
19 }
```

```
1  <?php
2
3  declare(strict_types=1);
4
5  namespace EcPhp\Factorial;
6
7  class Factorial
8  {
9      public function of($integer)
10     {
11         $factorial = 1;
12
13         foreach (range(1, $integer) as $n) {
14             $factorial *= $n;
15         }
16
17         return $factorial;
18     }
19 }
```

Code coverage

coverage 100%

coverage 50%

coverage 0 %

Are we doing it right ?

Metric meaning

Line code coverage

coverage 100%

coverage 50%

coverage 0%

Line code coverage

coverage 100%



Uber

coverage 50%



Bring'em on !

coverage 0%



Can I play daddy ?

Ok still, but are we doing it
right ?

How to test the quality of your tests ?

Mutation testing

Mutation testing

Mutation testing (or mutation analysis or program mutation) is used to design new software tests and evaluate the quality of existing software tests. Mutation testing involves modifying a program in small ways. Each mutated version is called a mutant and tests detect and reject mutants by causing the behavior of the original version to differ from the mutant. This is called killing the mutant. Test suites are measured by the percentage of mutants that they kill. New tests can be designed to kill additional mutants. Mutants are based on well-defined mutation operators that either mimic typical programming errors (such as using the wrong operator or variable name) or force the creation of valuable tests (such as dividing each expression by zero). The purpose is to help the tester develop effective tests or locate weaknesses in the test data used for the program or in sections of the code that are seldom or never accessed during execution. Mutation testing is a form of white-box testing.

TL; DR.

Mutation testing

Mutation testing is a type of software testing where we change certain statements in the source code and check if tests are still passing.

Mutation testing frameworks

Mutation testing frameworks

- Infection

How does it work ?

Warning, black magic ahead.

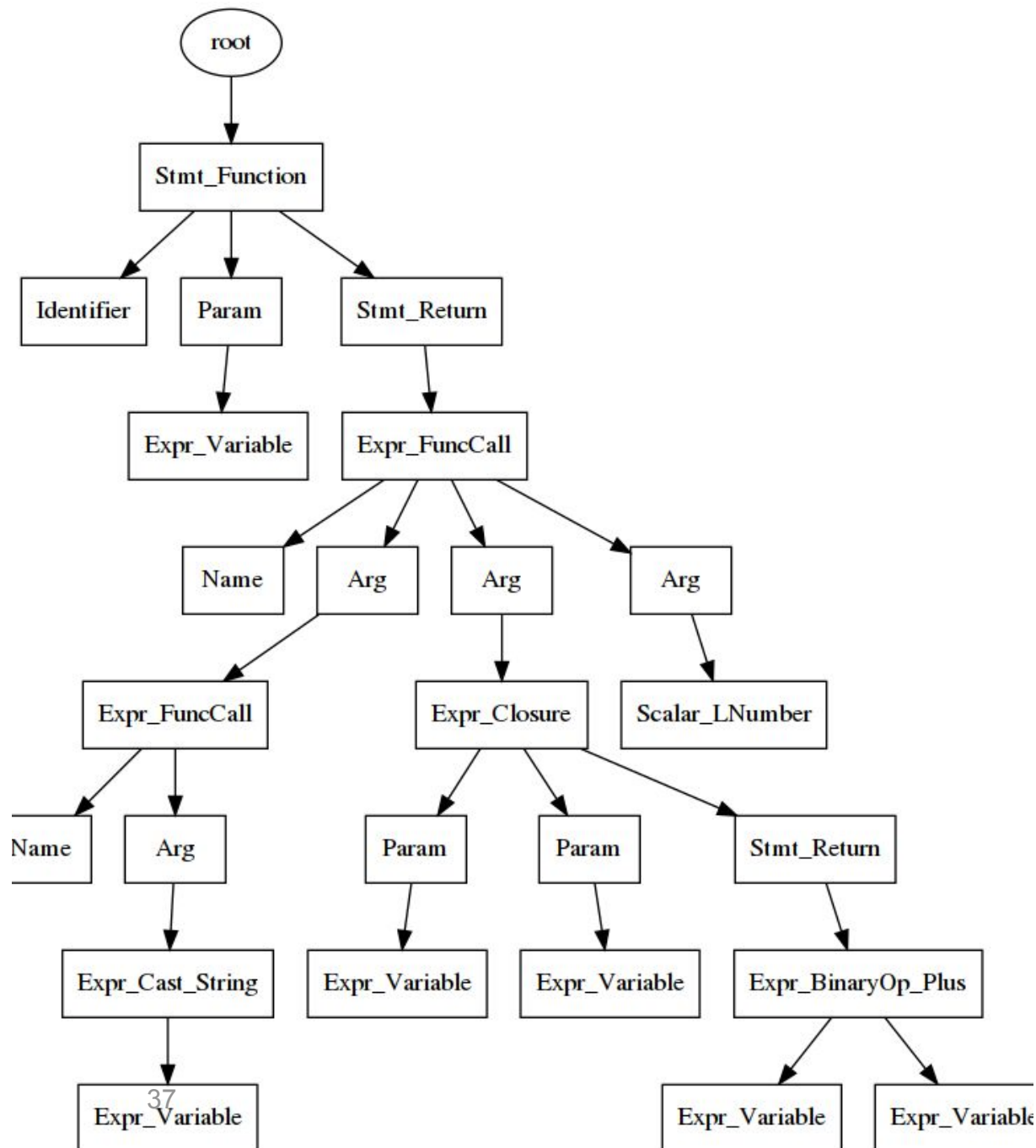
Mutation testing workflow

- Collect the source files,
- Convert files into AST,
- For each mutator, look for potential candidates,
- Generate a mutant by applying a mutator to an original source file,
- Run the actual tests using the mutant instead of the original source file,
- Inspect the result,
- Collect the metrics.

AST

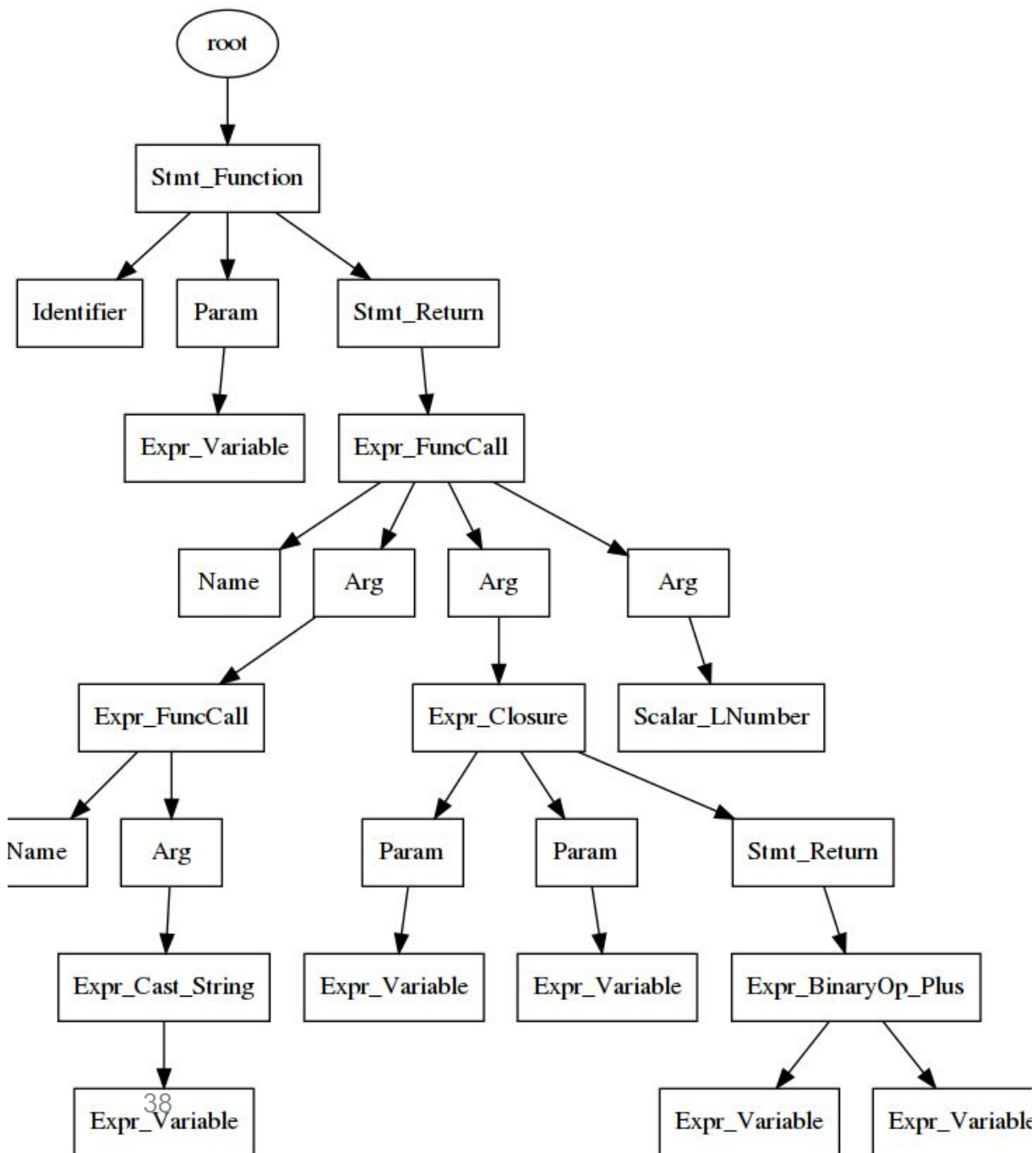
Abstract Syntax Trees - The corner stone

Abstract Syntax Tree



`<>` guessWhatIDo.php

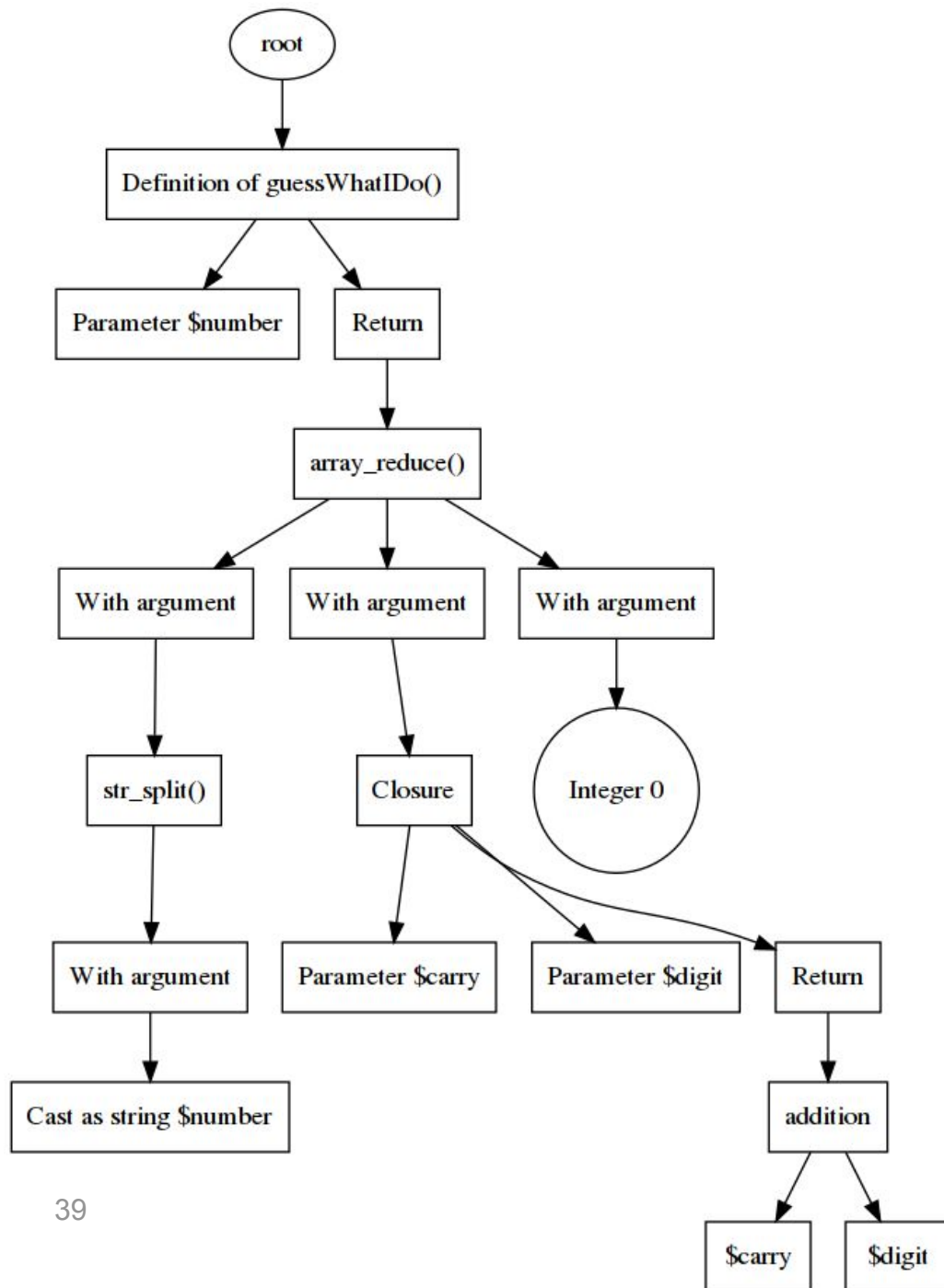
```
1  <?php
2
3  function guessWhatIDo($number)
4  {
5      return array_reduce(
6          str_split((string) $number),
7          static function ($carry, $digit) {
8              return $carry + $digit;
9          },
10         0
11     );
12 }
```



Abstract Syntax Tree

- The file is "tokenized"
- Each token has a name and only one parent
- A PHP script is then a tree !
- ❤️ nikic/php-parser
- ❤️ microsoft/tolerant-php-parser
- ❤️ nikic/php-ast

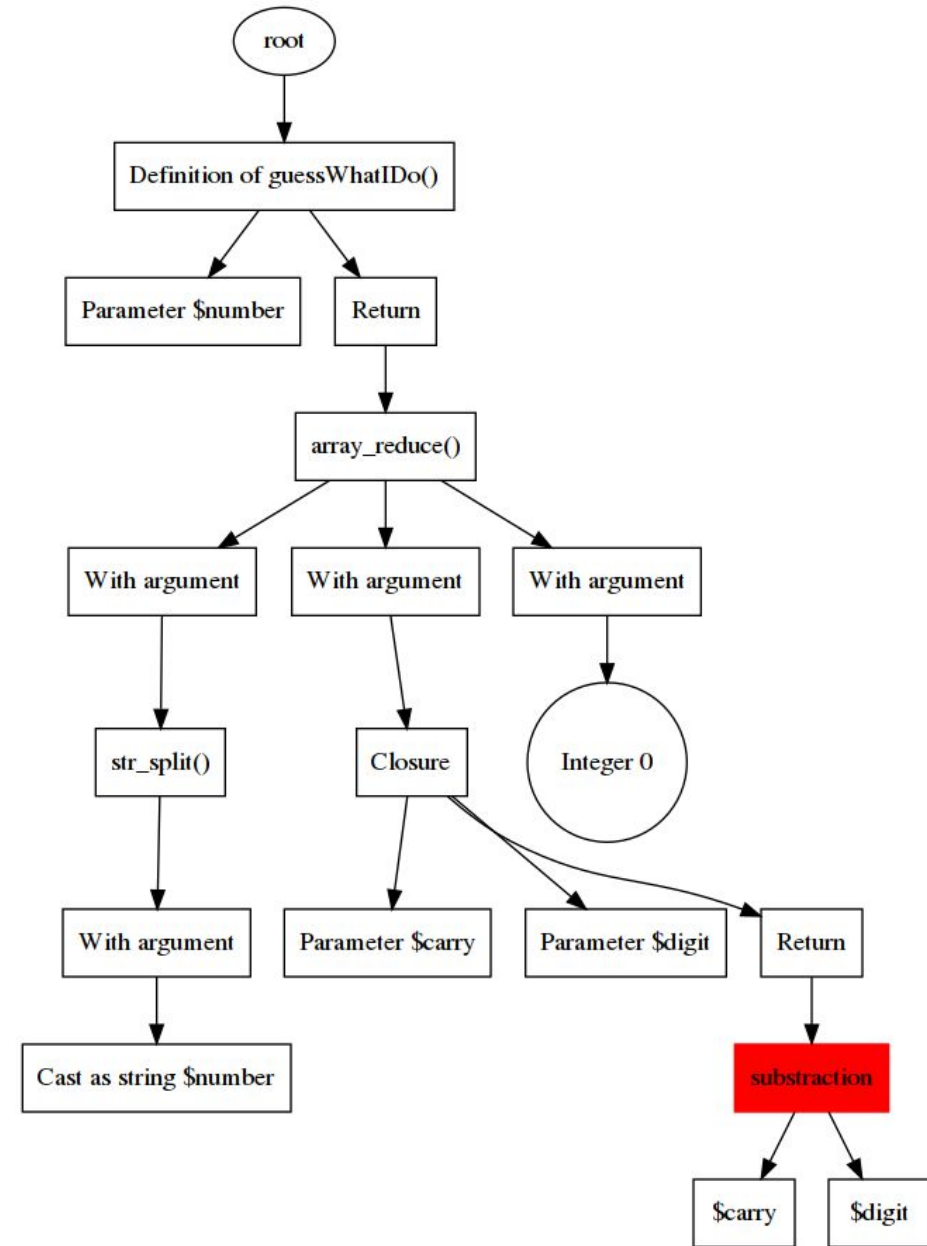
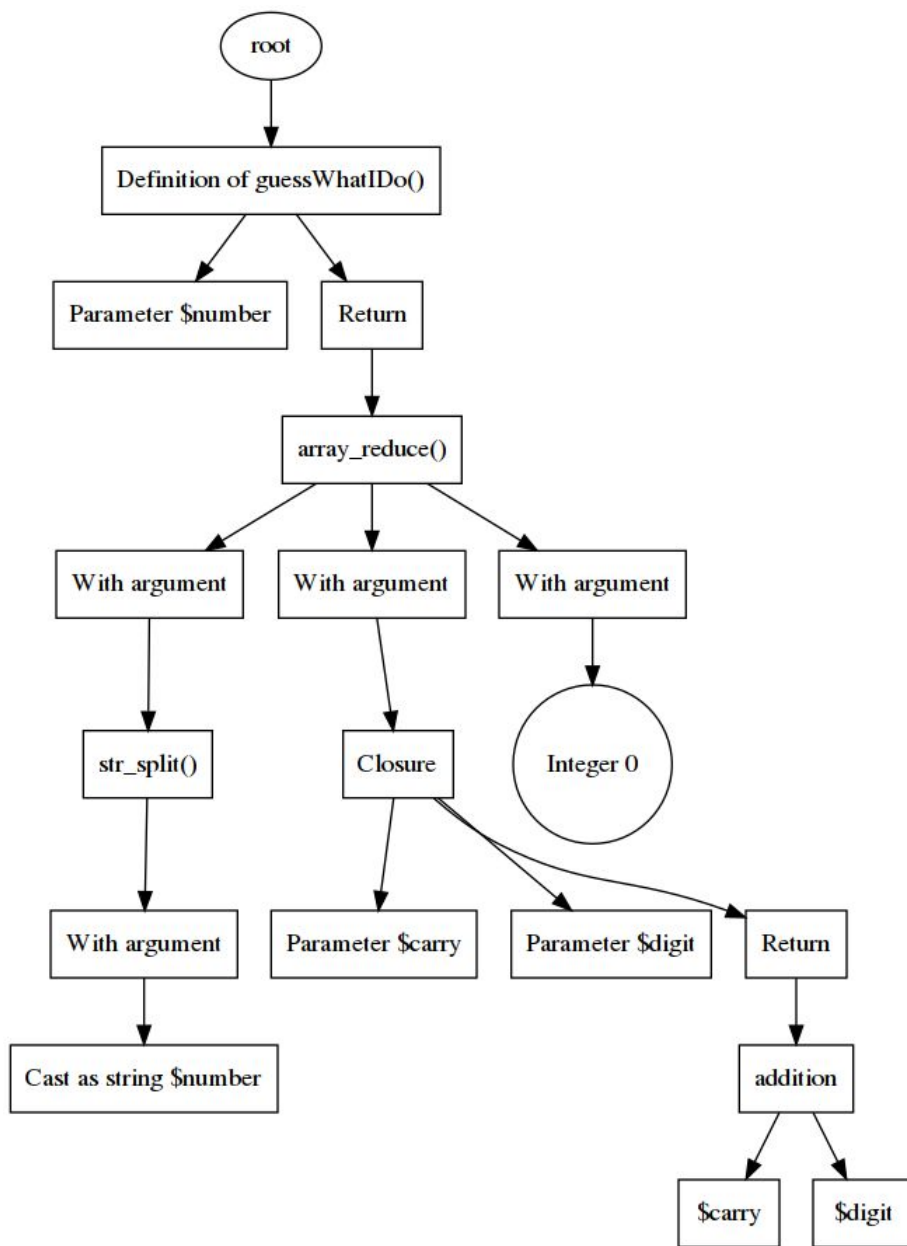
Abstract Syntax Tree



`<>` `guessWhatIDo.php`

```
1  <?php
2
3  function guessWhatIDo($number)
4  {
5      return array_reduce(
6          str_split((string) $number),
7          static function ($carry, $digit) {
8              return $carry + $digit;
9          },
10         0
11     );
12 }
```

Mutators



Towards better metrics

Better metrics

$$MSI = \frac{mutantsKilled}{mutantsTotal}$$

Better metrics

$$MCC = \frac{mutantsTotal - notCoveredByTestsMutants}{mutantsTotal}$$

Better metrics

$$CCMSI = \frac{mutantsKilled}{mutantsTotal - notCoveredByTestsMutants}$$

Live demo

Live demo scenario

- Branch 1:
 - Base version, without tests.
- Branch 2:
 - A bug is found, we need more tests, and hopefully commit a fix!
- Branch 3:
 - Better tests are added to increase code coverage.
- Branch 4:
 - Further optimizations.

Other curiosity

**Nikita Popov** @nikita_ppv · 27 déc. 2019
I've implemented a fuzzer for PHP:
github.com/nikic/PHP-Fuzz...

Fuzzing is a great way to find obscure bugs in parsing libraries...



nikic/PHP-Fuzzer
Experimental fuzzer for PHP libraries. Contribute to nikic/PHP-Fuzzer development by creating an account on GitHub.
github.com

4 102 326

**Pol Dellaiera** @drupol
En réponse à @nikita_ppv
What would be the difference with infection/infection ?
5:45 PM · 27 déc. 2019 · [Twitter Web App](#)

1 J'aime

**Nikita Popov** @nikita_ppv · 27 déc. 2019
En réponse à @drupol
Infection does mutation testing, which is in a way the opposite of fuzzing. Mutation testing randomly modifies the code while keeping inputs the same (given by test suite). Fuzzing randomly generates the inputs while keeping the code the same.

1 2 16

Thanks

- Core Infection developers
 - For their tremendous work,
 - Théo Fidry for all his good advices and reviews.
 - Maks Rafalko for [his presentation](#).
- European Commission Open Source Program Office
 - For their proofreading and corrections
- You !
 - For coming here to listen to me !

Licences and images

Images used in this presentation are free of use.



© European Union 2020

Reuse of this presentation authorised
under the CC BY 4.0 license.

Software used:

Latex formula editor: <https://www.codecogs.com/latex/eqneditor.php>

AST Generator: <https://github.com/loopphp/phptree-ast-generator>

More detailed information on Creative Commons can be found [here](#).