Property-based-testing

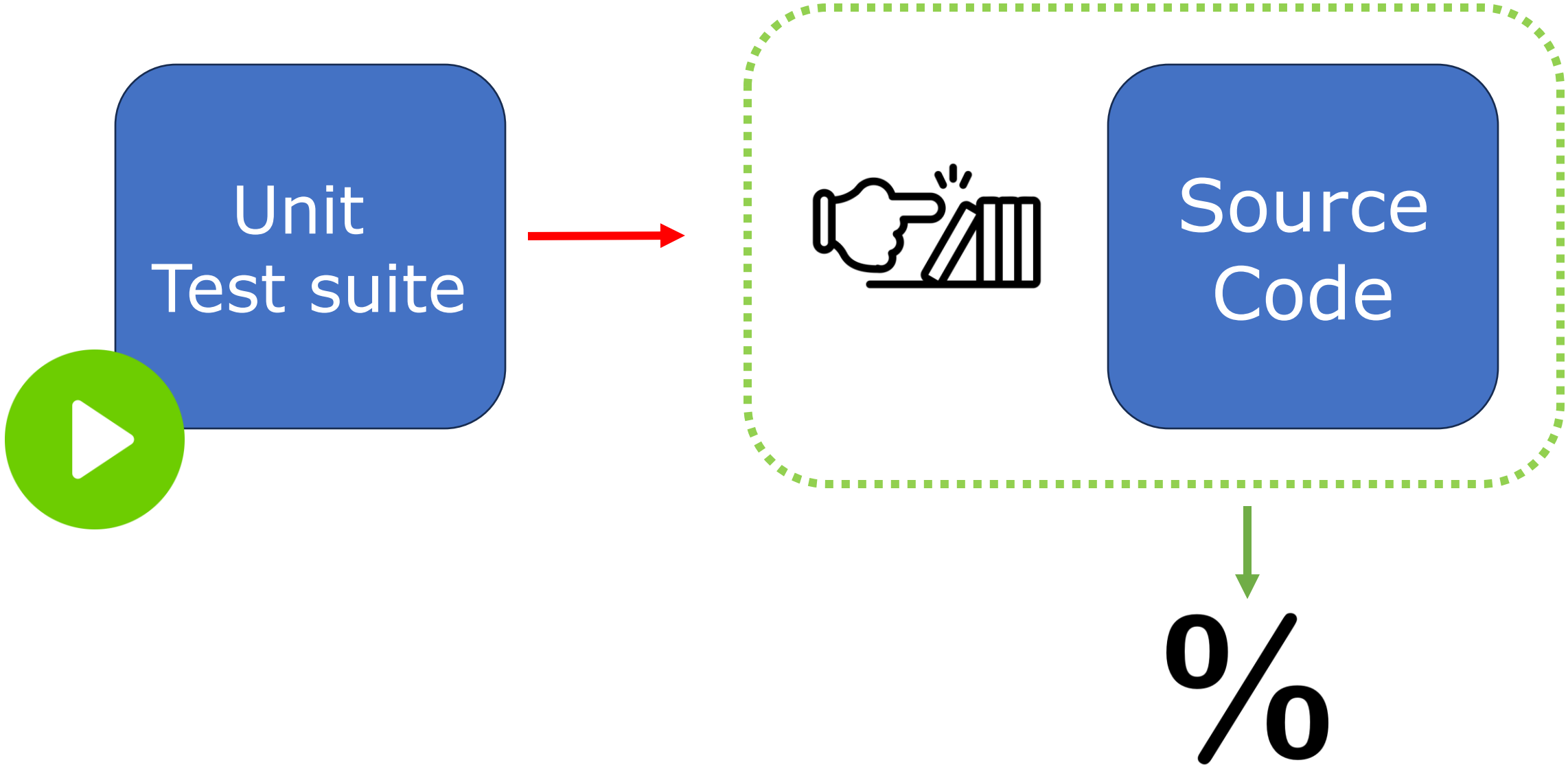
- Unveiling the truth on test reduction -

When have you done enough testing?...


Test coverage
Statement coverage
Line coverage
Function Call Coverage
Code coverage
Decision Coverage
Function Coverage
Modified Condition/Decision Coverage (MCDC)
Condition Coverage
Cyclomatic Complexity
Relational Boundary Coverage
Test coverage
Test Code Coverage

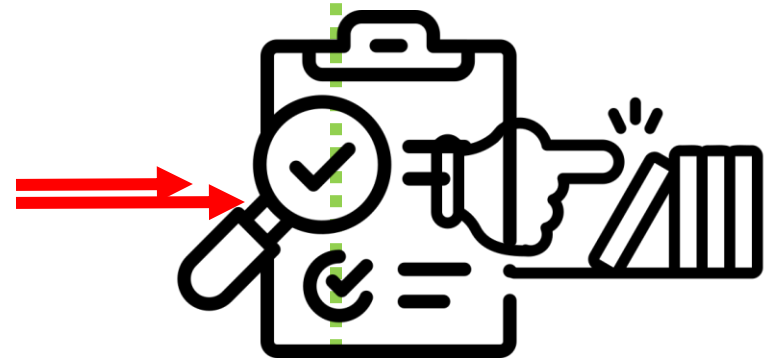
Code Coverage != Test Coverage

Code coverage



Test coverage

Unit
Test suite



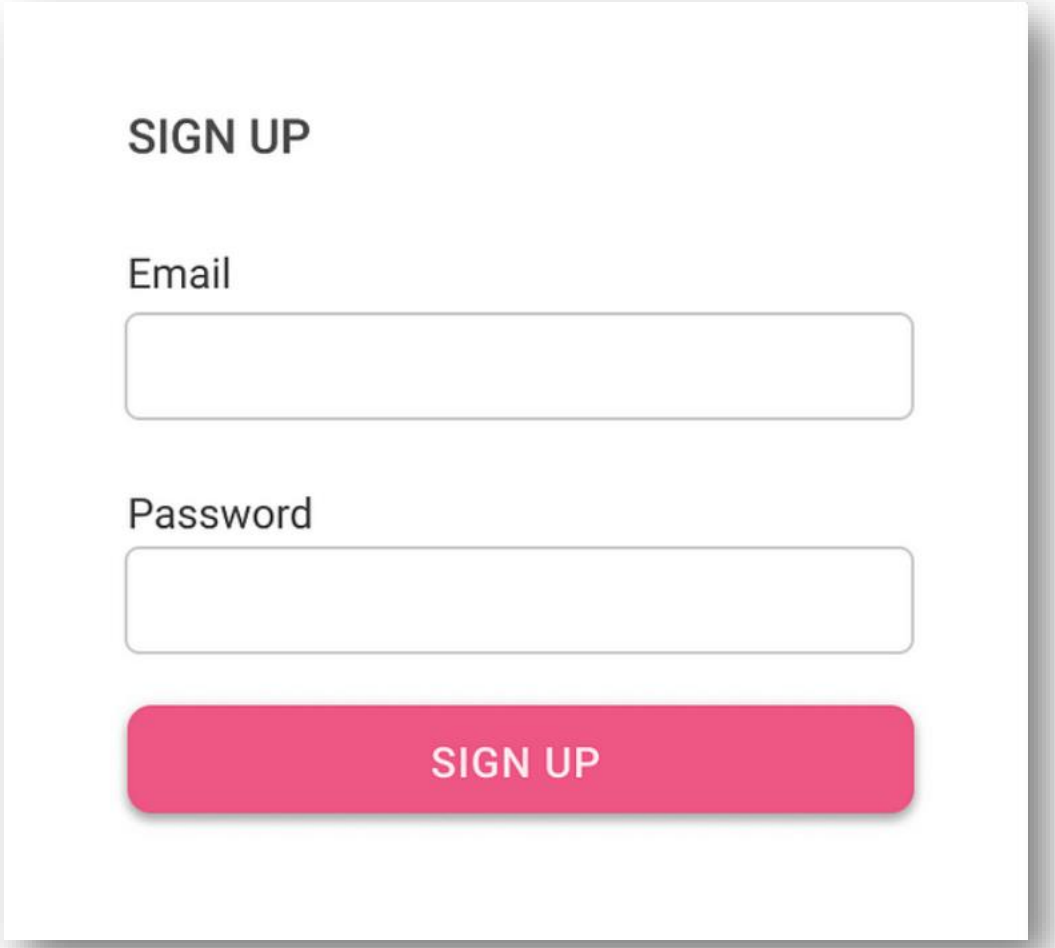
Source
Code

%

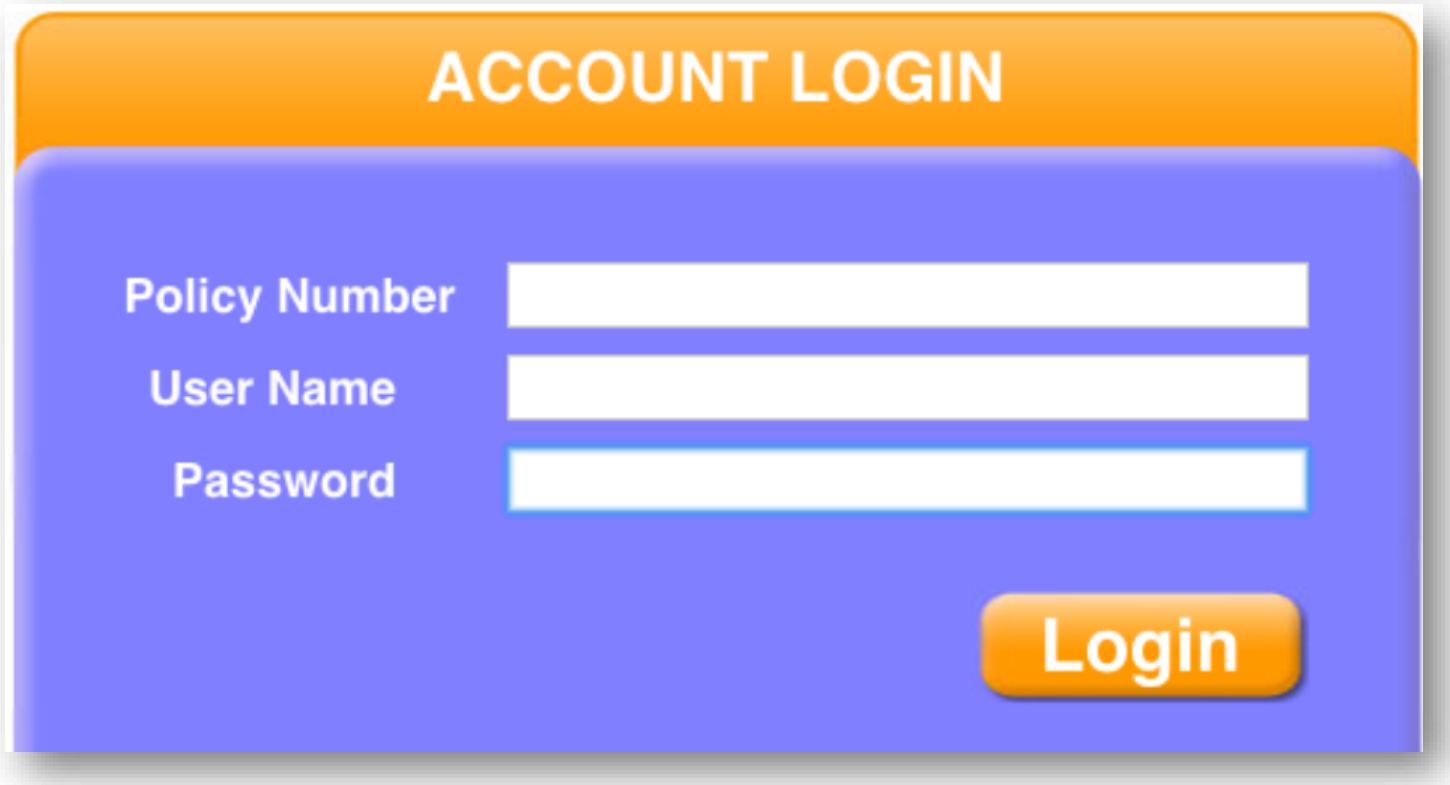
How to get a high(er) coverage?

Error Guessing

Decision table



Conditions	Case #1	Case #2	Case #3	Case #4
#1	True	False	False	True
#2	False	True	False	True

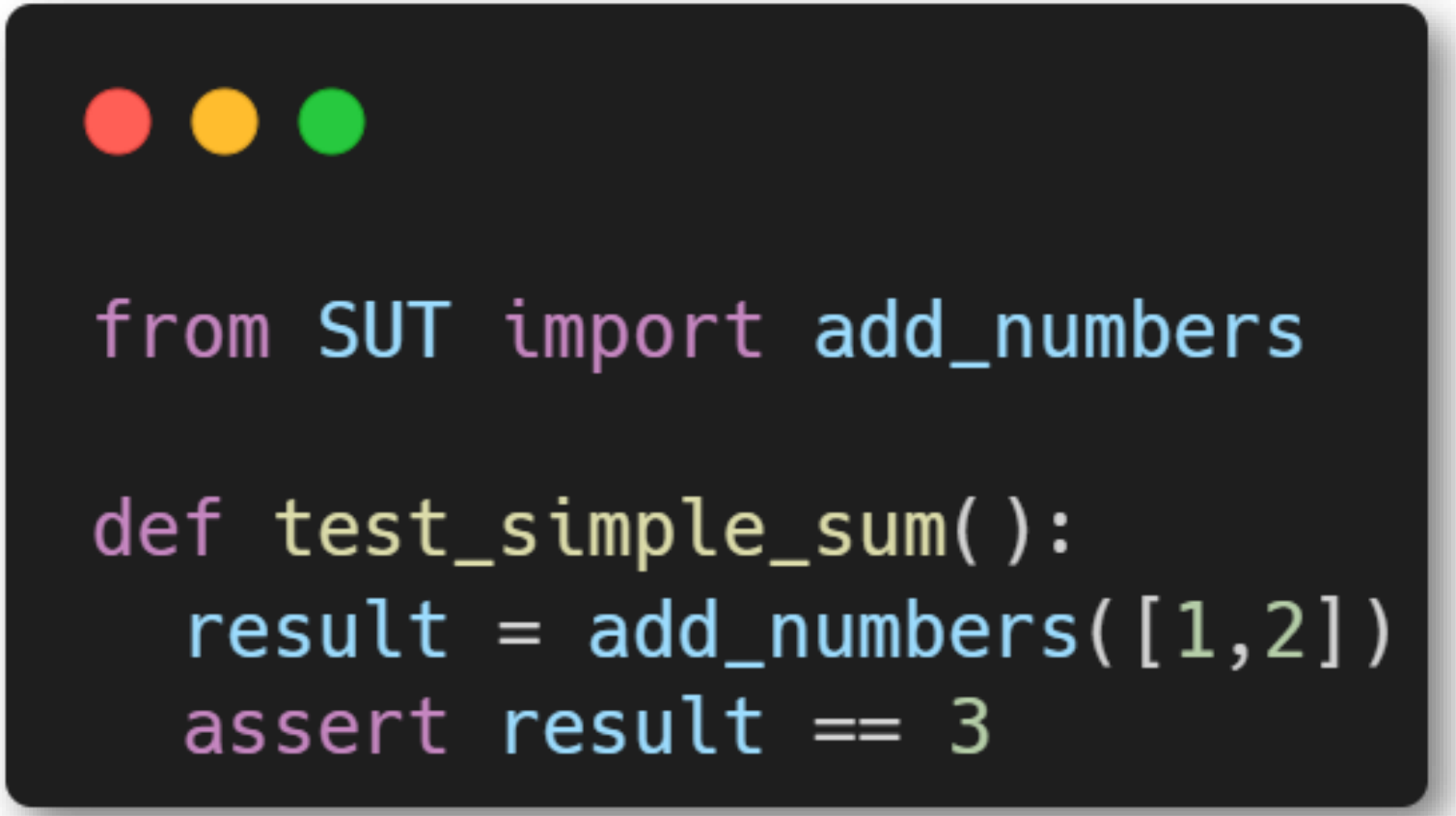


Conditions	Case #1	Case #2	Case #3	Case #4	Case #5	Case #6	Case #7	Case #8
#1	True	True	True	True	False	False	False	False
#2	False	True	False	True	False	True	False	True
#3	False	False	True	True	False	True	True	False

$$x = 2^N$$

N	X
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	…

How many conditions can you define?



```python
from SUT import add_numbers

def test_simple_sum():
    result = add_numbers([1,2])
    assert result == 3
```

SPECIFIC

[1, 2] == 3

```
from SUT import add_numbers

def test_simple_sum():
    result = add_numbers([1,2])
    assert result == 3
```

GENERIC --> DESCRIBING

“list of integers”

“integer”
“sum(*list of integers*)”

```python
from hypothesis import given
import hypothesis.strategies as st
from SUT import add_numbers

@given(st.lists(st.integers))
def test_simple_sum(example):
    result = add_numbers(example)
    assert isinstance(result, int)
    assert added_nums == sum(example)
```

Build a dragon

1. Array of integers
2. Minimum of 2 items

1. Type (output)
2. Content input VS content output
3. All items are 'lesser then or eaqual to' the next item

```python
from hypothesis import given
from collections import Counter
import hypothesis.strategies as st
from SUT import sort_algo

@given(st.lists(st.integers, min_size=2))
def test_simple_sum(l):
  s = sort_algo(l)

  print(f'list: {l}')

  assert isinstance(s, list)
  assert Counter(l) == Counter(s)
  assert all(x <= y for x, y in zip(s,s[1:]))
```

Shrinking

```
============================= test session starts ==============================
platform win32 -- Python 3.9.2, pytest-6.2.2, py-1.10.0, pluggy-0.13.1
rootdir: C:\Users\ewald\OneDrive - TestCoders B.V\Documenten\codechallenges\sudoku
plugins: hypothesis-6.12.0
collected 1 item

tests.py list: [0, 0]
list: [0, 0]
list: [-16, 1446194188, -5608, 30468, -21342, -1340026860858963418224843187799620263 88, 10]
list: [-7005, -92, 3656995437321075716]
list: [-1971873499119115395, 3656995437321075716]
list: [-281872320870016, -281872319939201, 16188]
list: [-281872320870016, -281872319939201, -385, -105, 3768, 841, -8299584243642436951, -103, -11269, 16316,
-18, 2153, -18745, -27367, 31966, 119, 19224]
list: [-3827198291545035 65, 23044, 26937, 112, 1781]
list: [17357, -10616, -14567, 1375094212723311558619512986334724558 64, -23629]
list: [17357, -10616, -14567, 1375094212723311558619512986334724558 64, -23629]
list: [-3827198291545035 65, 23044, 26937, 112, 1781]
list: [-3827198291545035 65, 23044, 26937, 112, 1781]
list: [-3827198291545035 65, 23044, 129]
list: [4, 129]
list: [-3827198291545035 65, 23044]
list: [-3827198291545035 65, 129]
list: [23044, 129]
list: [0, 90]
list: [129, 23044]
list: [23044, 0]
list: [23044, 0]
list: [0, 0]
list: [0, 0]
list: [6660, 0]
list: [2564, 0]
list: [516, 0]
list: [4, 0]
list: [2, 0]
list: [1, 0]
list: [0, 0]
list: [1, 0]
list: [0, 0]
list: [1, 0]
list: [1, 0]
list: [1, -2]
list: [-2, 0]
list: [0, 1]
Falsifying example: test_sort_algo(
    l=[1, 0],
)
list: [1, 0]
F

================================== FAILURES ===================================
_____________________________ test_sort_algo ______________________________

    @given(st.lists(st.integers(), min_size=2))
>   def test_sort_algo(l):

tests.py:7:

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

l = [1, 0]

    @given(st.lists(st.integers(), min_size=2))
    def test_sort_algo(l):
        s = sort_algo(l)
        print(f'list: {l}')
        assert isinstance(s, list)
        assert Counter(l) == Counter(s)
>       assert all(x <= y for x, y in zip(s, s[1:]))
E       assert False
E        +  where False = all(<generator object test_sort_algo.<locals>.<genexpr> at 0x000001C913D4DE80>)

tests.py:12: AssertionError
---------------------------------- Hypothesis ----------------------------------
Falsifying example: test_sort_algo(
    l=[1, 0],
)
=========================== short test summary info ===========================
FAILED tests.py::test_sort_algo - assert False
============================== 1 failed in 0.62s ==============================
```

```
tests.py list: [0, 0]
list: [0, 0]
list: [-16, 1446194188, -5608, 30468, -21342, -1340026860858963418224843187799620263 88, 10]
list: [-7005, -92, 3656995437321075716]
list: [-1971873499119115395, 3656995437321075716]
list: [-281872320870016, -281872319939201, 16188]
list: [-281872320870016, -281872319939201, -385, -105, 3768, 841, -8299584243642436951, -103, -11269, 16316,
-18, 2153, -18745, -27367, 31966, 119, 19224]
list: [-3827198291545035 65, 23044, 26937, 112, 1781]
list: [17357, -10616, -14567, 1375094212723311558619512986334724558 64, -23629]
list: [17357, -10616, -14567, 1375094212723311558619512986334724558 64, -23629]
list: [-3827198291545035 65, 23044, 26937, 112, 1781]
list: [-3827198291545035 65, 23044, 26937, 112, 1781]
list: [-3827198291545035 65, 23044, 129]
list: [4, 129]
list: [-3827198291545035 65, 23044]
list: [-3827198291545035 65, 129]
list: [23044, 129]
list: [0, 90]
list: [129, 23044]
list: [23044, 0]
list: [23044, 0]
list: [0, 0]
list: [0, 0]
list: [6660, 0]
list: [2564, 0]
list: [516, 0]
list: [4, 0]
list: [2, 0]
list: [1, 0]
list: [0, 0]
list: [1, 0]
list: [0, 0]
list: [1, 0]
list: [1, 0]
list: [1, -2]
list: [-2, 0]
list: [0, 1]
Falsifying example: test_sort_algo(
    l=[1, 0],
)
list: [1, 0]
F
```

Shrinking

[-38271929154503565, 23044, 26937, 112, 1781] 

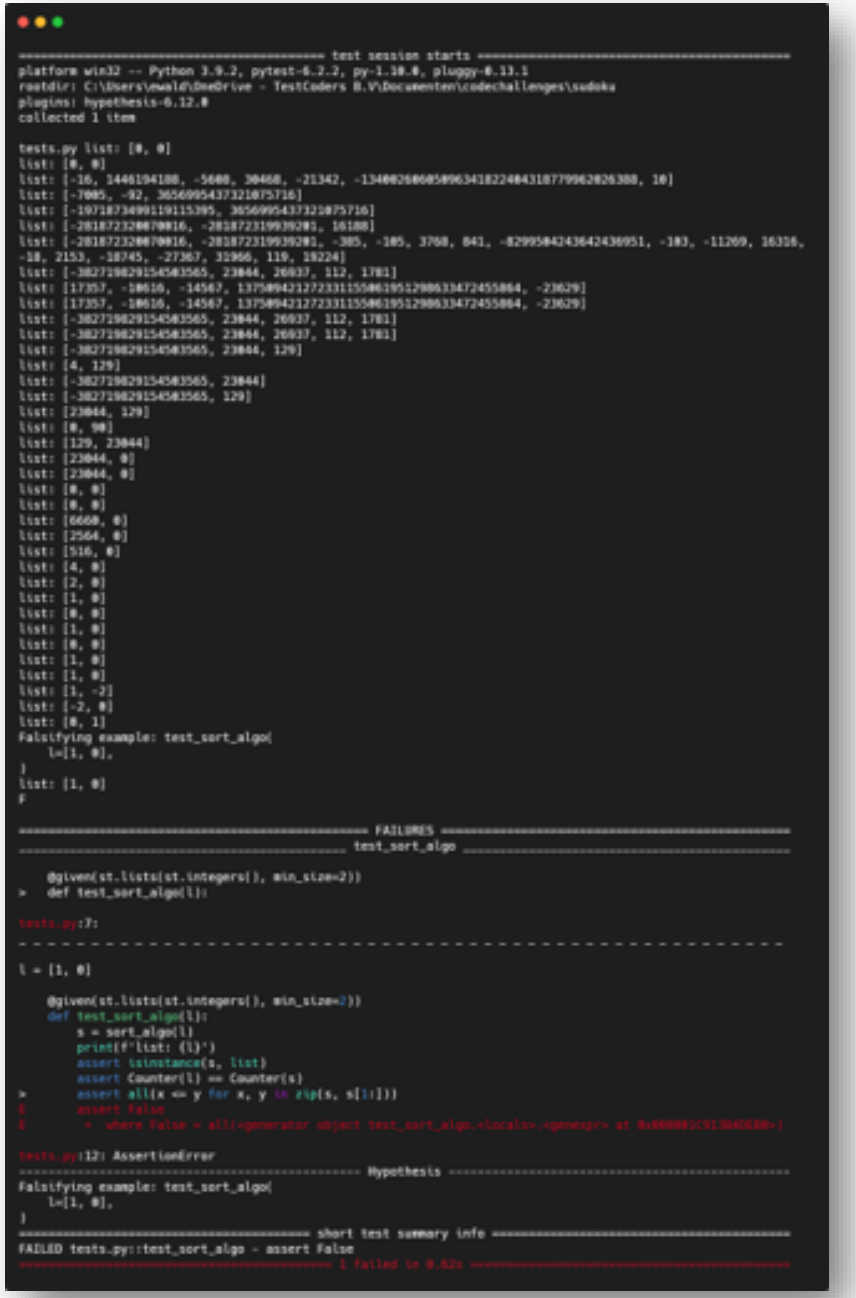
[-38271929154503565, 23044, 129] 

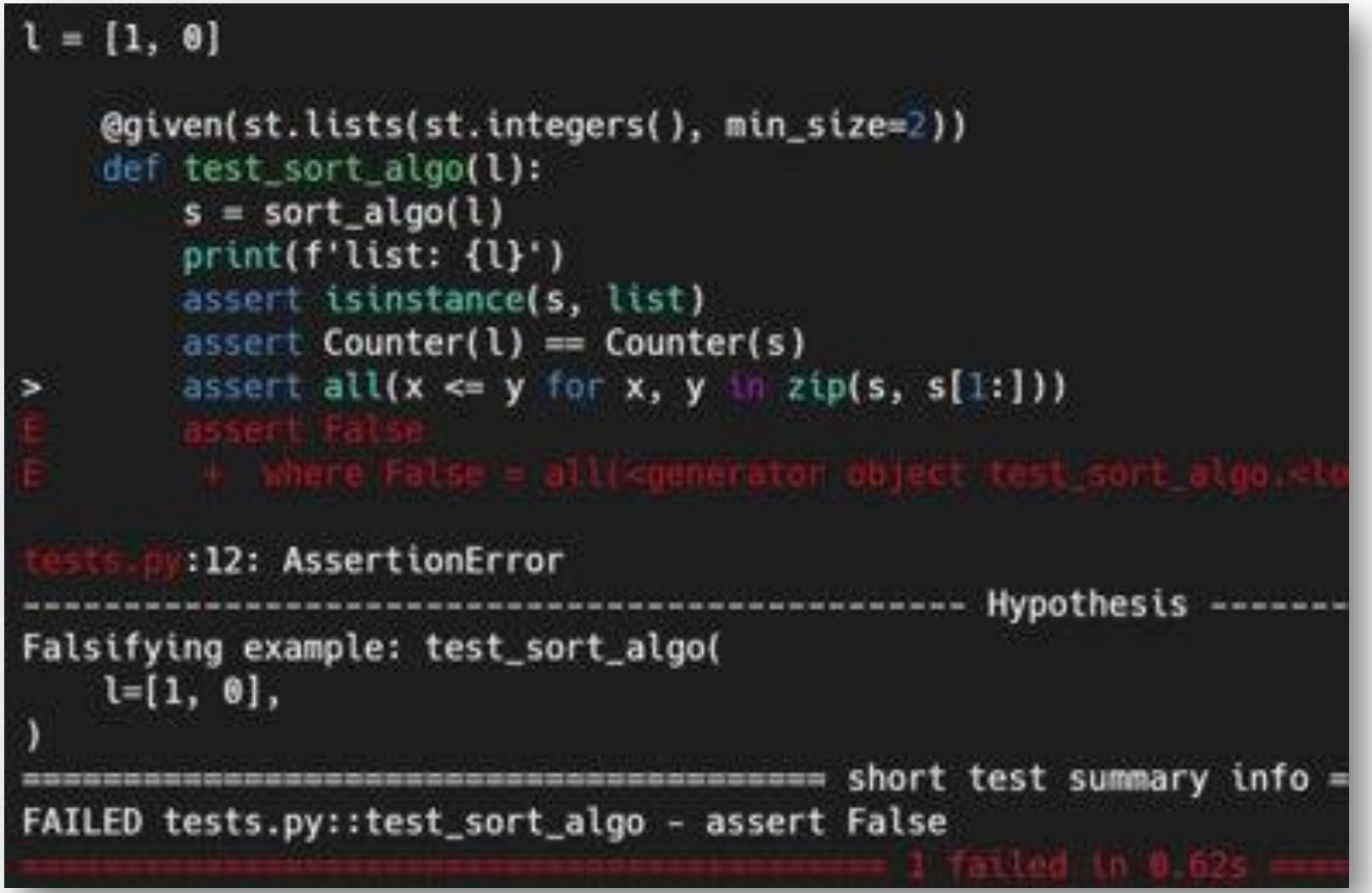
[23044, 129] 

[23044, 0] 

!!! >>> [1, 0] <<< !!!

Trace view


test session starts
platform win32 -- Python 3.9.2, pytest-6.2.2, py-1.10.0, pluggy-0.13.1
rootdir: C:\Users\ewald\OneDrive - TestCoders B.V\Documenten\codechallenges\sudoku
plugins: hypothesis-6.12.0
collected 1 item
tests.py list: [0, 0]
list: [0, 0]
list: [-16, 1446194188, -5680, 30468, -21342, -13408268685896341822484318779962026388, 10]
list: [-7085, -92, 365609543732187571 6]
list: [-197187349911911539 5, 365609543732187571 6]
list: [-281872320870816, -28187231993928 1, 16188]
list: [-281872320870816, -28187231993928 1, -385, -105, 3768, 841, -829958424364243695 1, -103, -11269, 16316, -18, 2153, -18745, -27367, 31966, 119, 19224]
list: [-382719829154583565, 23044, 26937, 112, 1781]
list: [17357, -18616, -14567, 13758942127233115586195129863347245588 4, -23629]
list: [17357, -18616, -14567, 13758942127233115586195129863347245588 4, -23629]
list: [-382719829154583565, 23044, 26937, 112, 1781]
list: [-382719829154583565, 23044, 26937, 112, 1781]
list: [-382719829154583565, 23044, 129]
list: [4, 129]
list: [-382719829154583565, 23044]
list: [-382719829154583565, 129]
list: [23044, 129]
list: [0, 90]
list: [129, 23044]
list: [23044, 0]
list: [23044, 0]
list: [0, 0]
list: [0, 0]
list: [6660, 0]
list: [2564, 0]
list: [516, 0]
list: [4, 0]
list: [2, 0]
list: [1, 0]
list: [0, 0]
list: [1, 0]
list: [0, 0]
list: [1, 0]
list: [1, 0]
list: [1, -2]
list: [-2, 0]
list: [0, 1]
Falsifying example: test_sort_algo(
l=[1, 0],
)
list: [1, 0]
F
FAILURES
test_sort_algo
@given(st.lists(st.integers(), min_size=2))
def test_sort_algo(l):
tests.py:7:
l = [1, 0]
@given(st.lists(st.integers(), min_size=2))
def test_sort_algo(l):
s = sort_algo(l)
print(f'list: {l}')
assert isinstance(s, list)
assert Counter(l) == Counter(s)
> assert all(x <= y for x, y in zip(s, s[1:]))
E assert False
E + where False = all(<generator object test_sort_algo.<locals>.<genexpr> at 0x000001C913040E80>)
tests.py:12: AssertionError
Hypothesis
Falsifying example: test_sort_algo(
l=[1, 0],
)
short test summary info
FAILED tests.py::test_sort_algo - assert False
1 failed in 0.62s


l = [1, 0]
@given(st.lists(st.integers(), min_size=2))
def test_sort_algo(l):
s = sort_algo(l)
print(f'list: {l}')
assert isinstance(s, list)
assert Counter(l) == Counter(s)
> assert all(x <= y for x, y in zip(s, s[1:]))
E assert False
E + where False = all(<generator object test_sort_algo.<lo
tests.py:12: AssertionError
Hypothesis
Falsifying example: test_sort_algo(
l=[1, 0],
)
short test summary info =
FAILED tests.py::test_sort_algo - assert False
1 failed in 0.62s

Build **custom** strategies

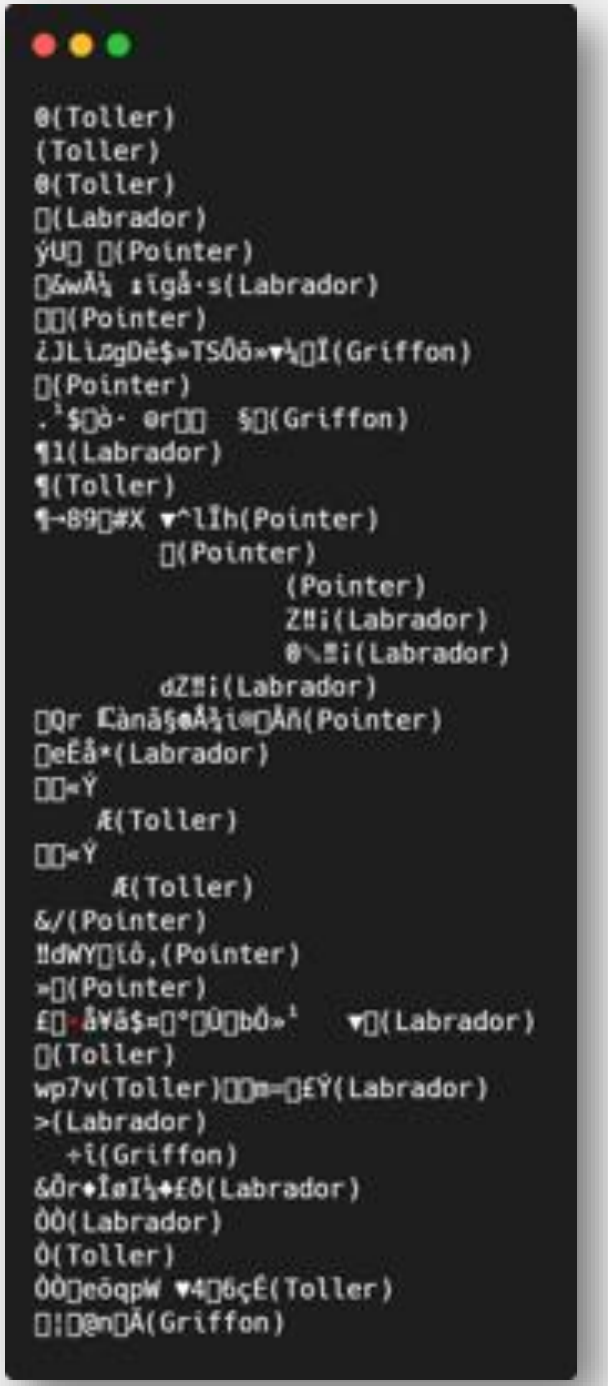
```
class Dog:

    def __init__(self, name, breed):
        self.name = name
        self.breed = breed

    def __str__(self) -> str:
        return f'{self.name}({self.breed})'
```

```
st_dogs = st.builds(Dog,
                    name=st.text(),
                    breed=st.sampled_from(MY_BREEDS))

@given(st_dogs)
def test_animals(animal):
  print(animal)
```



Omg, that is sooo hypothesis!..

(Pointer)
.¹$ò· 0r §(Griffon)
¶1(Labrador)
¶(Toller)
¶→89#X ▼^lÏh(Pointer)
(Pointer)
(Pointer)
Z‼¡(Labrador)
0‼¡(Labrador)
dZ‼¡(Labrador)
Qr Ćànã§0Å¾i®Åñ(Pointer)
eËå*(Labrador)
«Ý
Æ(Toller)
«Ý
Æ(Toller)
&/(Pointer)
‼dWY ïô,(Pointer)

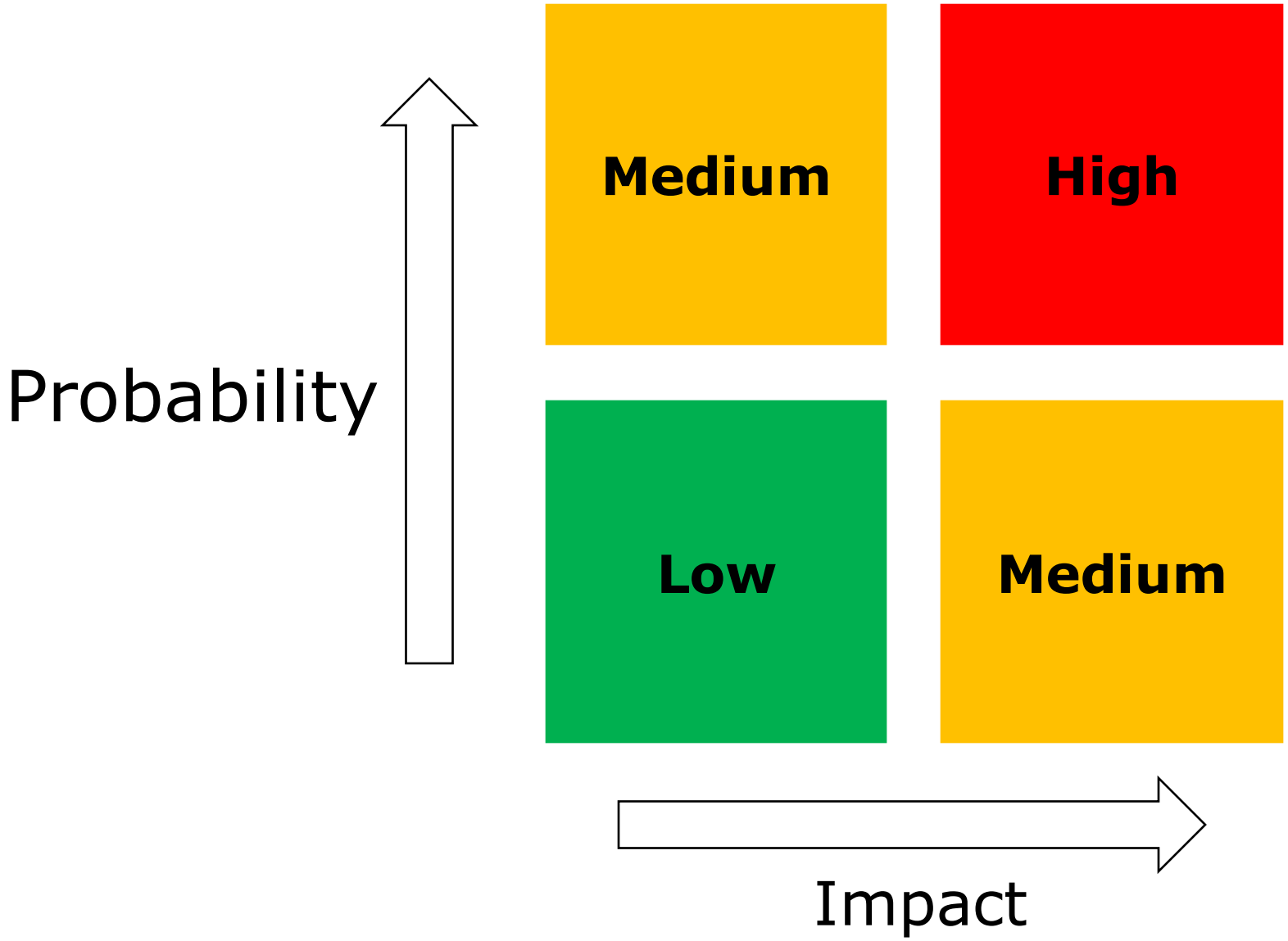
Test Reduction

PBT Characteristics

- Increase of coverage
- Possibly decrease test code (not test cases)

When have you done enough testing?...

RISK BASED TESTING



”Build a dragon”

PDF presentation

github.com/EwaldVerhoeven/NDC-Porto

ewald@testcoders.nl

NDC Porto 2023

(room 4 rocks!)