Reviewing Automatic Test Generation Tools (Auger, Deal, Klara, Pynguin)

Evelyn Kuo

Overview

Python

Automatic test generation tools

Ease of Use

Quality of Test Cases

Tools

Contract-based

DEAL

Function history

Automated

Unittest

Gen

E

Rator

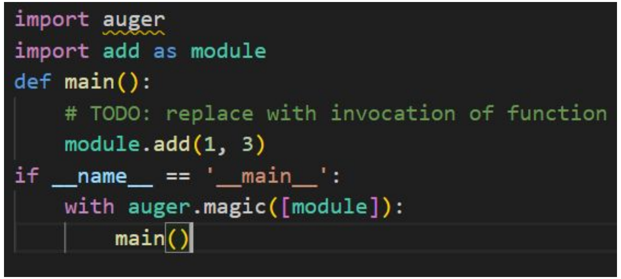
Coverage of return cases

Klara

Code coverage

Pynguin

Tools

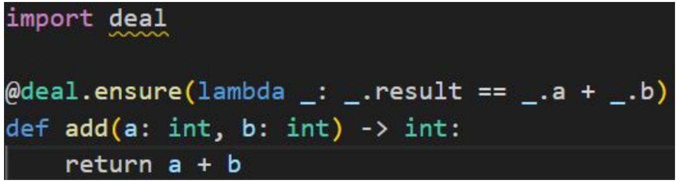


```python
import auger
import add as module
def main():
    # TODO: replace with invocation of function
    module.add(1, 3)
if __name__ == '__main__':
    with auger.magic([module]):
        main()
```

Automated
Unittest
Gen
E
Rator

Contract-based

DEAL

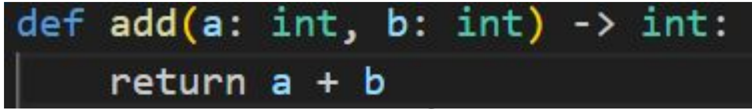


```python
import deal

@deal.ensure(lambda _: _.result == _.a + _.b)
def add(a: int, b: int) -> int:
    return a + b
```

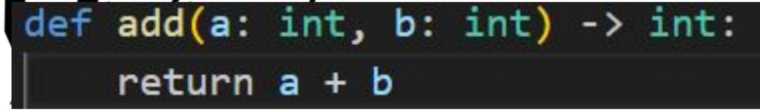
Coverage of return cases

Klara



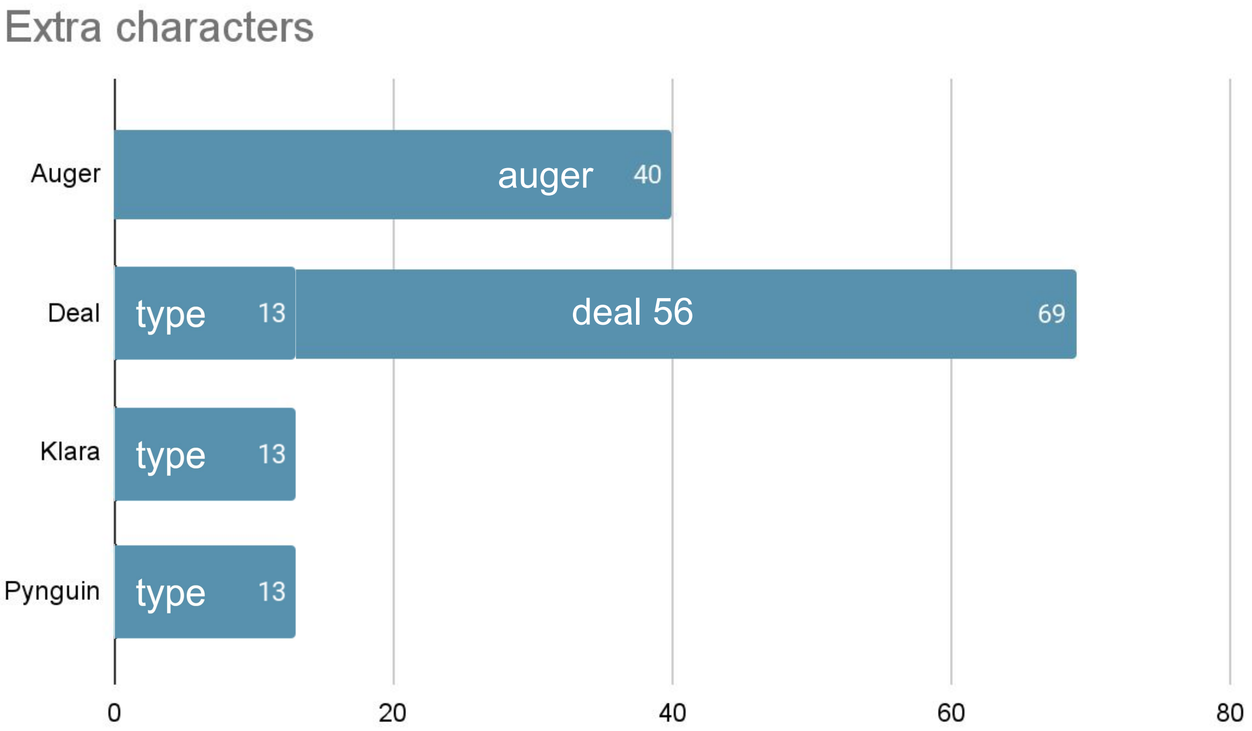
```python
def add(a: int, b: int) -> int:
    return a + b
```

Pynguin



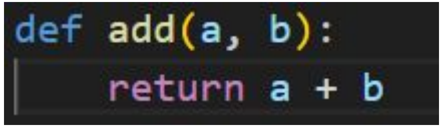
```python
def add(a: int, b: int) -> int:
    return a + b
```

Examples (Add)



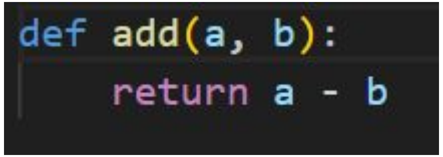
Examples (Add)

Original



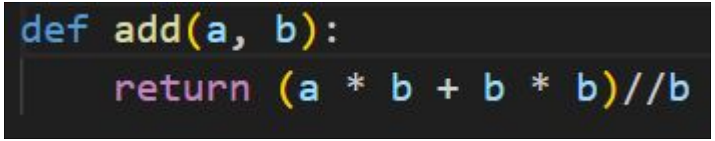
```python
def add(a, b):
    return a + b
```

Bug 1



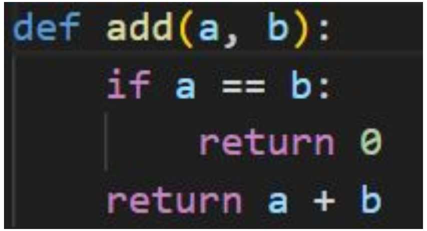
```python
def add(a, b):
    return a - b
```

Bug 2



```python
def add(a, b):
    return (a * b + b * b)//b
```

Bug 3



```python
def add(a, b):
    if a == b:
        return 0
    return a + b
```

DEAL

Caught all bugs

Klara

Automated
Unittest
Gen
E
Rator

Pynguin

Didn’t use type information for test cases

Conclusion

No free lunch

- Deal: creating good contracts in return for good test cases
- Auger/Pynguin/Klara: less effort, but assumes code as ground truth

Future

- Pynguin: not sure if type information is some local setting issue

Links

Github: https://github.com/enkokoro/program_analysis_project

Final Report (see References): https://docs.google.com/document/d/17PLXPvryhvp1SeDS1k_h2Ff4aoeoeVUFzLZeB8PKw1A/edit?usp=sharing