Automated Test Case Generation for CTRL using Pex: Lessons Learned

David PY Lawrence \(^1\)
Manuel Gonzalez-Berges \(^2\)
Didier Buchs \(^1\)

Stefan Klikovits \(^{1,2}\)
What are we doing?

- 1 MLOC code
- no automated unit testing until two years ago
- frequent changes in execution environment
- (mostly) manual verification
- big expenses (time) on QA side
What are we doing?

- 1 MLOC code
- no automated unit testing until two years ago
- frequent changes in execution environment
- (mostly) manual verification
- big expenses (time) on QA side
How are we doing it?

**ITEC workflow**

- **IDE**
  - code
  - IDE
  - test cases
  - test driver

- **CTRL**
  - SP engine
  - SP CTRL
  - CTRL test gen.
  - SP data
  - CTRL test inputs

- **TI generator**
  - source code translator
  - test inputs
  - test input translator
  - TI generator
  - SP tool code

**Considering Execution Environment Resilience: A White-Box Approach**
Klikovits et. al.
SERENE 2015

Stefan Klikovits 3 s.k@cern.ch
How are we doing it?

IDE

SP engine

CTRL

test gen.

code

results

SP data

test cases

test driver

test inputs

CTRL

TI generator

source code translator

SP tool code

TC Gen

test input translator

test inputs

ITEC workflow

Considering Execution Environment Resilience: A White-Box Approach

Klikovits et. al. SERENE 2015

Stefan Klikovits 3 s.k@cern.ch
Recap semi-purification

- replace dependencies with parameters

A non-pure function

```python
f(x){
    if GLOBAL_VAR:
        return dbGet(x)
    else:
        return -1
}
```

Semi-purified $f(x)$

```python
f_sp(x,a,b){
    if a:
        return b
    else:
        return -1
}
```

Test case

```python
test_f_sp(){
x = f("test",True,5) //act
assert(x == 5) //assert
}
```
From Pex to test cases

Test case generation from Pex output

void test_case_1() {
    param1 = ...
    param2 = ...
    res = CUT(param1, param2)
    assert("check", res == ...)
}
Test case generation: results

CTRL Functions
1111
Semi-purification
166
Unsupported
159
SP Err
SP Functions
786
Translation
184
Translation Err
C# Code
602
ATCG (Pex)

1184
mismatching oracles
2465
matching oracles
294
TCGen Err
29 Exec Error
Test Cases
3678
Test Inputs
3972

Stefan Klikovits
s.k@cern.ch
Test case generation: update

CTRL Functions
1191

1111

Semi-purification

SP Functions
875
786

Translation

C# Code
676
602

ATCG (Pex)

Test Cases
3117
3678

TCGen

Test Inputs
3610
3972

249
166
 Unsupported

67
159
 SP Err

489
1184
 mismatching oracles

2613
2464
 matching oracles

15
29
 Exec Error

493
294
 TCGen Err

493
294
 TCGen
Number of test cases

![Bar chart displaying the number of test cases by function count.]

- 0 test cases: 44 functions
- 1-3 test cases: 295 functions
- 4-7 test cases: 231 functions
- 8-15 test cases: 87 functions
- >15 test cases: 19 functions

Stefan Klikovits 8 s.k@cern.ch
# Coverages

![Bar chart showing coverage percentages and corresponding number of functions.]

- 100% coverage: 230 functions
- 99% - 75% coverage: 129 functions
- 74% - 50% coverage: 110 functions
- 49% - 1% coverage: 118 functions
- 0% coverage: 15 functions

Stefan Klikovits

s.k@cern.ch
Coverages: updated

- 100%: 406
- 99% - 75%: 230
- 74% - 50%: 129
- 49% - 1%: 110
- 0%: 118

Stefan Klikovits
s.k@cern.ch
Coverages: matching oracles

Stefan Klikovits s.k@cern.ch
Lessons learned

- not everything can be translated (easily)
Lessons learned

- not everything can be translated (easily)
- not all features should be supported
Lessons learned

- not everything can be translated (easily)
- not all features should be supported
- C# is no silver bullet
Lessons learned

- not everything can be translated (easily)
- not all features should be supported
- C# is no silver bullet
- improving the quality of test cases?
Lessons learned

- not everything can be translated (easily)
- not all features should be supported
- C# is no silver bullet
- improving the quality of test cases?
- tools have “features”
How are we doing it?

IDE

test driver

test cases

results

test gen.

CTRL

code

SP engine

SP data

CTRL test gen.

CTRL

test gen.

results

test cases

source code translator

TC Gen

SP tool code

test inputs

test input translator

ITEC workflow

Stefan Klikovits 11 s.k@cern.ch
How are we doing it?

IDE → SP engine → CTRL

test cases

CTRL

test gen.

results

test driver

SP data

IDE

code

CTRL

test inputs

TC Gen

test input translator

CTRL

source code translator

SP tool code

test inputs

ITEC workflow

Stefan Klikovits
s.k@cern.ch
Why test the translation?

- DySyEx: execute code, cover max. paths
- small differences – big impacts
Why test the translation?

- DySyEx: execute code, cover max. paths
- small differences – big impacts

http://samcnitt.tumblr.com/
Why not test the translator?

- No formal semantics?
- Changing language?
- Effort to adapt the solution?
Why not test the translator?

- No formal semantics?
- Changing language?
- Effort to adapt the solution?

http://asterix.wikia.com/wiki/Asterix_and_Cleopatra
How to test?
How to test?

Divide

http://chapleau.us/Img/caesar_asterix.gif
How to test?

Divide

http://chapleau.us/lmg/caesar_asterix.gif

Anonymise

https://www.youtube.com/watch?v=UF6E-4G4n_M
How to test?

Divide
http://chapleau.us/Img/caesar_asterix.gif

Anonymise
https://www.youtube.com/watch?v=UF6E-4G4n_M

Analyse Blocks
https://en.gamigo.com/game/asterix
How to test?

Divide

http://chapleau.us/Img/caesar_asterix.gif

Anonymise

https://www.youtube.com/watch?v=UF6E-4G4n_M

Analyse Blocks

https://en.gamigo.com/game/asterix

Conquer

https://www.pinterest.com/pin/33678487835870673/
How to test?

```c
int func(int a, int b) {
    a++
    a++
    b = b+2
    if (a > b) {
        return a % b
    } else {
        return a + b
    }
}
```

Divide

Anonymise

https://www.youtube.com/watch?v=UF6E-4G4n_M

Analyse Blocks

https://en.gamigo.com/game/asterix

Conquer

https://www.pinterest.com/pin/336784878358770673/
How to test?

```
int func(int a, int b) {
    a++
    a++
    b = b+2
    if (a > b) {
        return a % b
    } else {
        return a + b
    }
}
```

Divide

```
int func(int , int){
    int++
    int++
    int = int + int
    if (int > int) {
        return int % int
    } else {
        return int + int
    }
}
```

Anonymise

Analyse Blocks

https://en.gamigo.com/game/asterix

Conquer

https://www.pinterest.com/pin/336784878358770673/

Stefan Klikovits 14 s.k@cern.ch
How to test?

```c
int func(int a, int b) {
    a++
    a++
    b = b+2
    if (a > b) {
        return a % b
    } else {
        return a + b
    }
}
```

Divide

```c
int func(int , int) {
    int++
    int++
    int = int + int
    if (int > int) {
        return int % int
    } else {
        return int + int
    }
}
```

Anonymise

```c
int func(int a, int b) {
    int++
    int++
    int = int+int
    if (int > int) {
        return int % int
    } else {
        return int + int
    }
}
```

Analyse Blocks

Conquer

https://www.pinterest.com/pin/336784878358770673/
How to test?

Int func(int a, int b) {
    a++
    a++
    b = b + 2
    if (a > b) {
        return a % b
    } else {
        return a + b
    }
}

Divide

Int func(int, int) {
    int++
    int++
    int = int + int
    if (int > int) {
        return int % int
    } else {
        return int + int
    }
}

Anonymise

\[ \phi = \frac{\sum \phi(L_i)}{|L|} \]

Conquer

Analyse Blocks
Anonymisation & basic blocks

- Equivalence classes
- Exhaustive testing of features
- Confidence in basic blocks
- Calculate confidence for CUT
The quality metric

For translated source code

\[
conf(c) = \frac{\sum_{i=1}^{n} \phi(anon_i)}{n}, \text{anon}_i \in \{\text{Anon}\}
\]
The quality metric

For translated source code

\[
\text{conf}(c) = \frac{\sum_{i=1}^{n} \phi(\text{anon}_i)}{n}, \text{anon}_i \in \{\text{Anon}\}
\]

For test cases

\[
\text{conf}_{tc}(c, \sigma) = \prod_{\forall s_i \in S | \sigma} \phi(\text{anonymize}(s_i))
\]
Example calculation

```c
int func(int a, int b) {
    int ++
    ++
    int = int + int
    if (int > int) {
        return int % int
    } else {
        return int + int
    }
}
```

\[ \phi(func) = \frac{1L_2 + 1L_4 + 0L_5 + 0L_6}{4} = 0.5 \]
Example calculation

```c
int func(int a, int b) {
    a++;  // 1
    a++;  // 1
    b = b+2  // 1
    if (a > b) {
        return a % b  // 0
    } else {
        return a + b  // 1
    }
}
```

\[ \phi(func) = \frac{1L_2 + 1L_4 + 0L_5 + 0L_6}{4} = 0.5 \]

\[ \phi(func, \langle 3, 5 \rangle) = 1L_2 \ast 1L_3 \ast 1L_4 \ast 0L_5 \ast 1L_8 = 0 \]
Lessons learned

- how to find exhaustive test suites?
Lessons learned

- how to find exhaustive test suites?
- granularity of basic blocks?
Lessons learned

- how to find exhaustive test suites?
- granularity of basic blocks?
- there are thousands (!!) of possibilities
Lessons learned

- how to find exhaustive test suites?
- granularity of basic blocks?
- there are thousands (!!) of possibilities
- automation?
What next?

- expand TC generation
- exhaustive testing for basic blocks
- representative study for quality metric
- trade-off complexity vs. usefulness
- research unsupported features
Conclusion
Conclusion
Conclusion
Automated Test Case Generation for CTRL using Pex: Lessons Learned

Stefan Klikovits $^{1,2}$

David PY Lawrence $^1$
Manuel Gonzalez-Berges $^2$
Didier Buchs $^1$