[1]University of Geneva, Switzerland
[2]CERN, Geneva, Switzerland

Towards Language Independent (Dynamic) Symbolic Execution

Stefan Klikovits [1,2]

Manuel Gonzalez-Berges [2]
Didier Buchs [1]

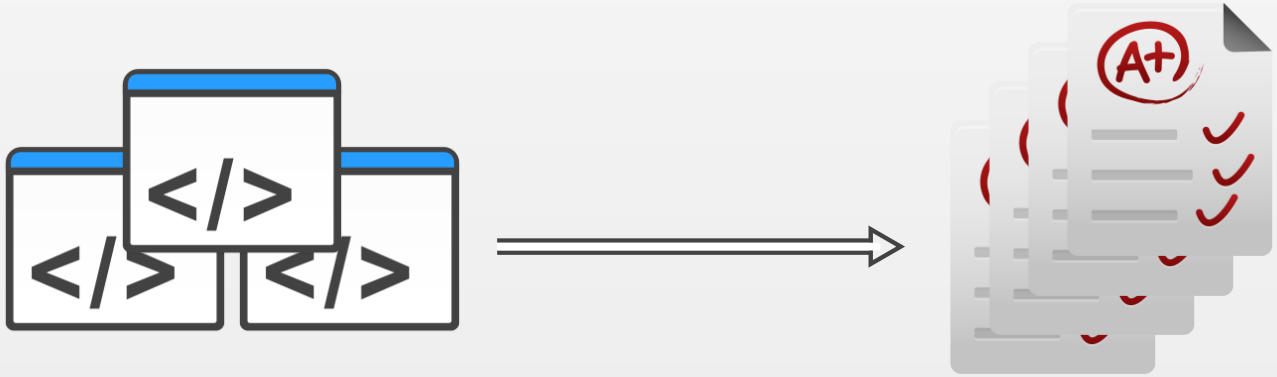
stefan.klikovits@unige.ch

What are we doing?

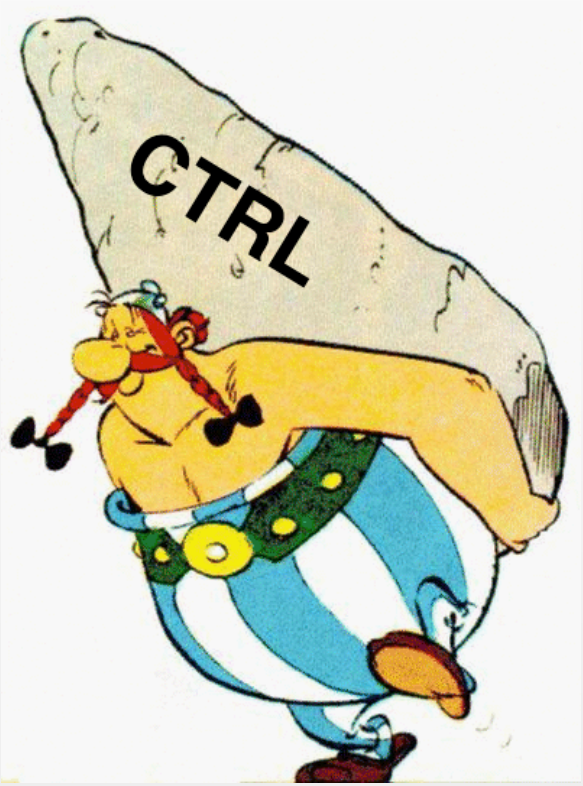
- 1 MLOC code *(Control)*
- no automated unit testing until three ago
- frequent changes in execution environment
- (mostly) manual verification
- big expenses (time) on QA side

What are we doing?

- 1 MLOC code *(Control)*
- no automated unit testing until three ago
- frequent changes in execution environment
- (mostly) manual verification
- big expenses (time) on QA side

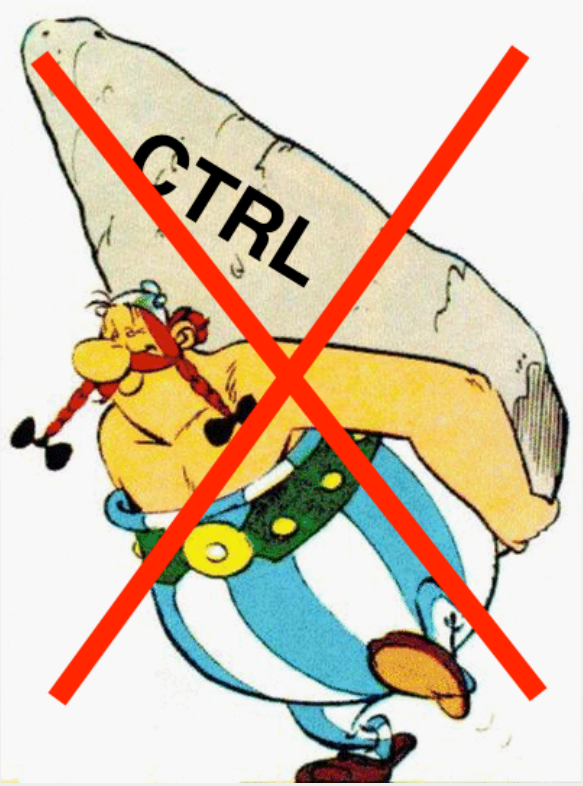


Language Independent Test Case Generation



https://thriftytraveller.files.wordpress.com/2013/11/asterix_obelix3.gif

Language Independent Test Case Generation



https://thriftytraveller.files.wordpress.com/2013/11/asterix_obelix3.gif

Language Independent Test Case Generation

1. Develop generic tool

Language Independent Test Case Generation

1. Develop generic tool

http://asterix.wikia.com/wiki/Asterix_and_Cleopatra

Language Independent Test Case Generation

1. Develop generic tool
2. Modify parser and execution

Language Independent Test Case Generation

1. Develop generic tool
2. Modify parser and execution

https://www.efiliale.de/efiliale/images/aktionen/asterix/5624_Troubadix.png

Language Independent Test Case Generation

1. Develop generic tool
2. Modify parser and execution
3. Translate to existing tool language

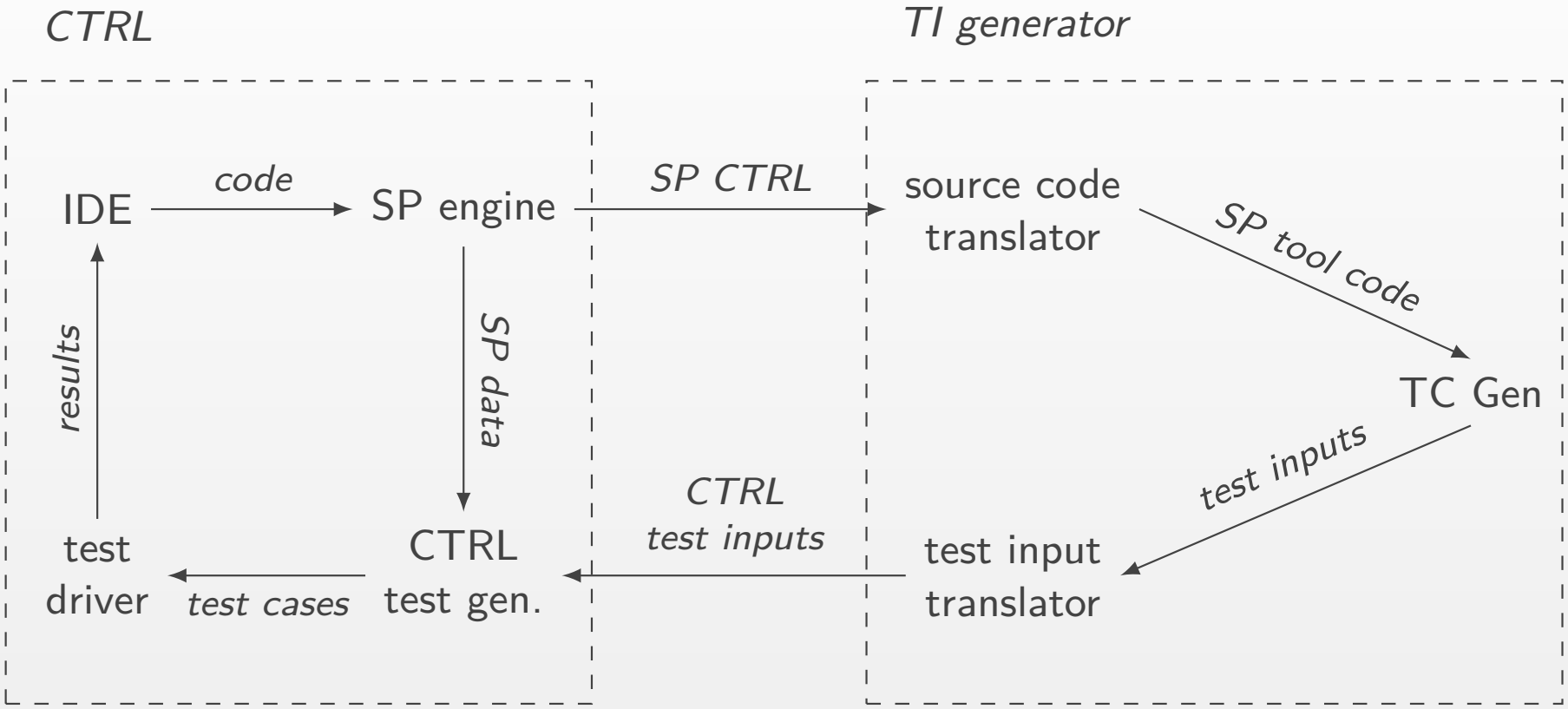
http://www.asterix.com/asterix-de-a-a-z/les-personnages/perso/a43b.gif

Semantics, semantics, semantics

- small differences – big impacts

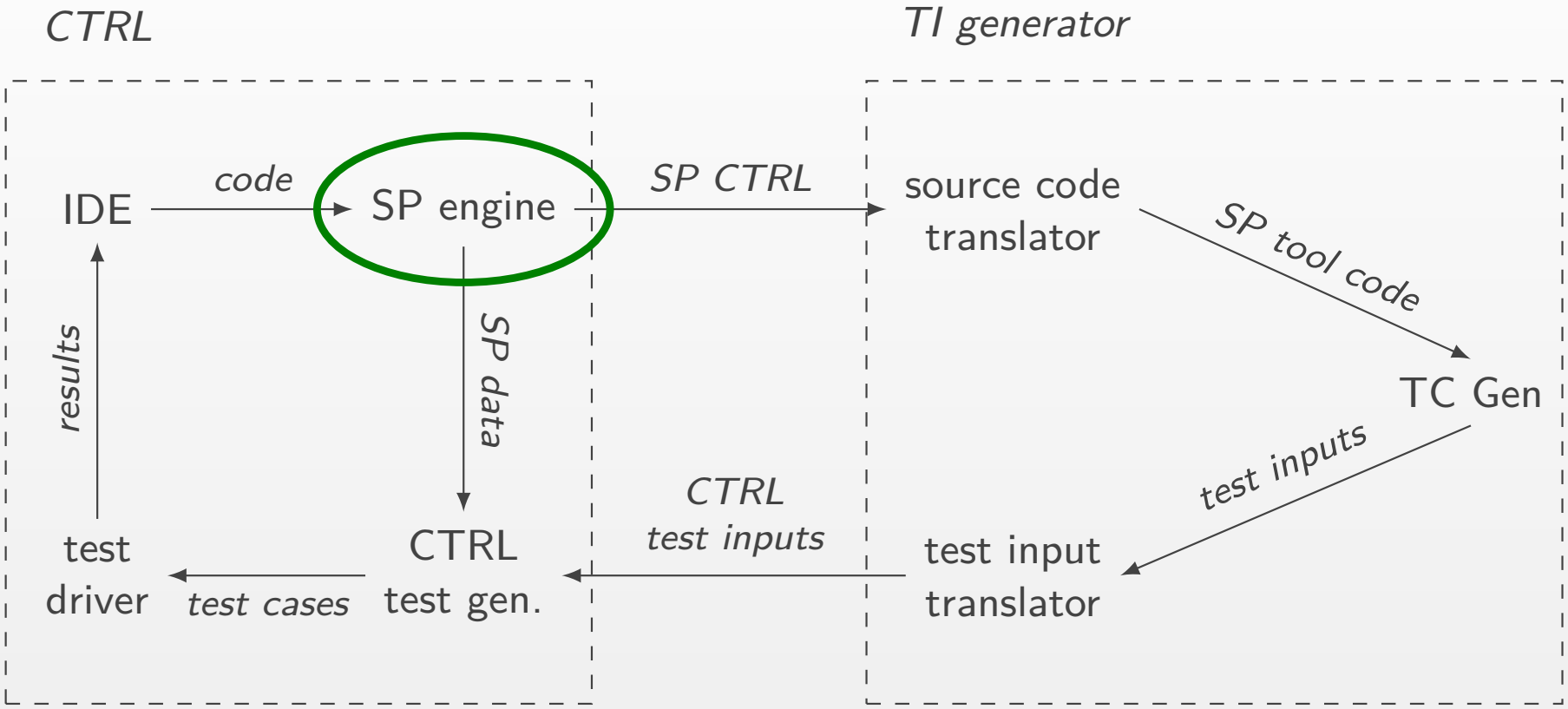
http://samcnitt.tumblr.com/

How do we generate TCs?



ITEC workflow

How do we generate TCs?



ITEC workflow

Considering Execution Environment Resilience: A White-Box Approach

Klikovits et. al., Proc. SERENE 2015, Paris

Semi-purification

- replace dependencies with parameters

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

A non-pure function

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

Semi-purified `f(x)`

```
test_f_sp(){
  x = f("test",True,5) //act
  assert(x == 5) //assert
}
```

Test case

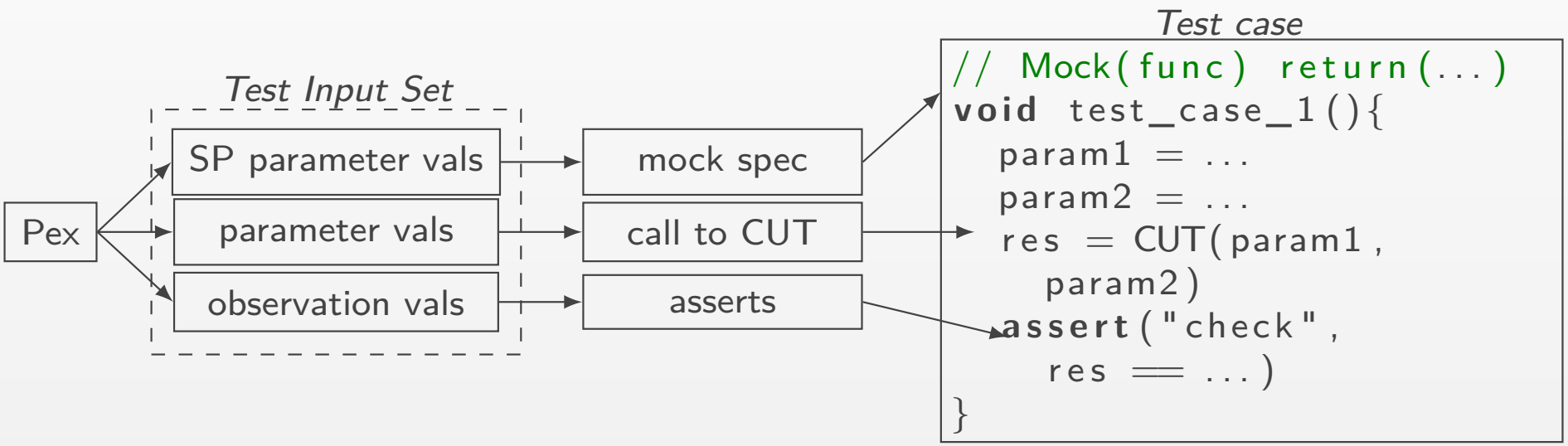
CUT translation & TC Gen

- Pex (Microsoft Research)
- Dynamic Symbolic Execution

- translate CUT, generate PUT
- manually create Pex factories, data types, built-in functions

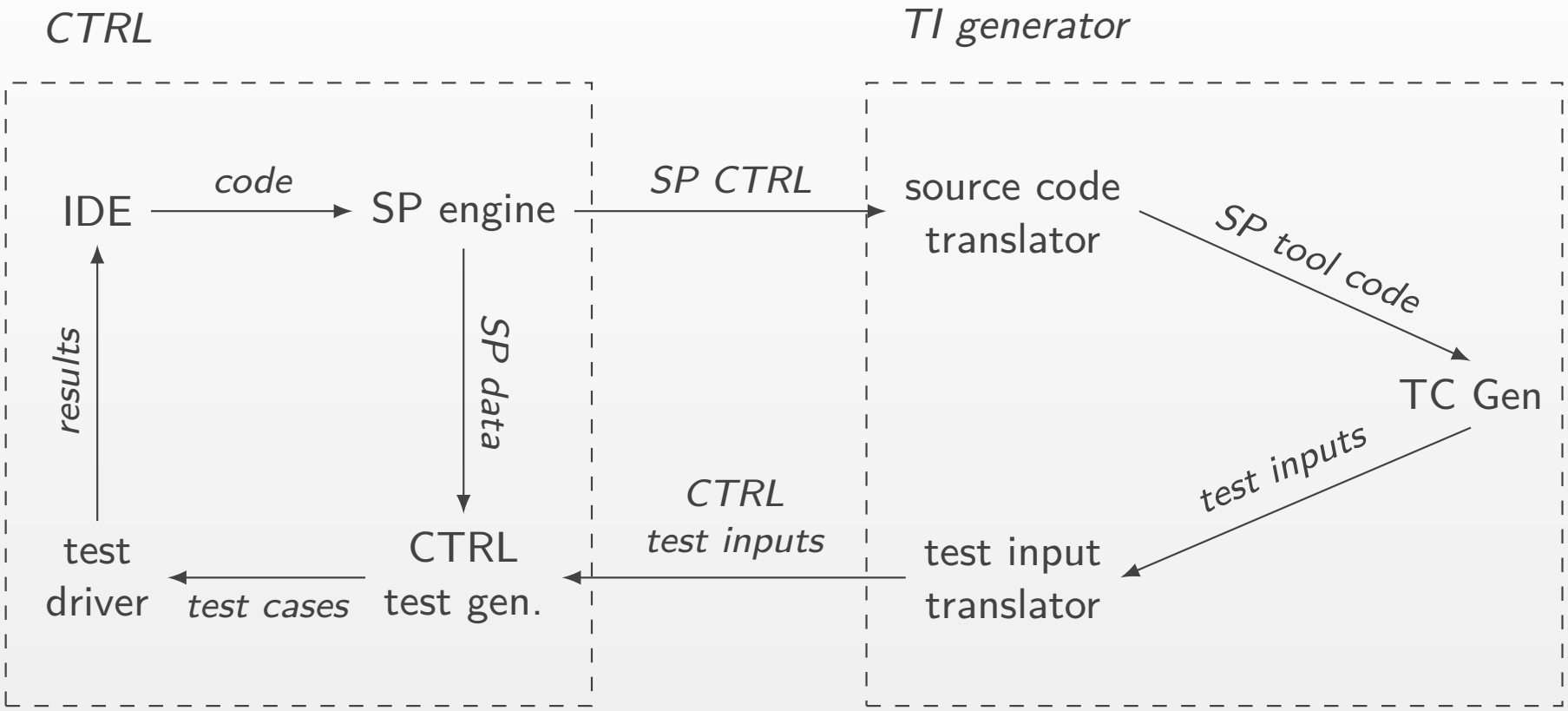
https://sites.google.com/site/diedruidenmt/_/rsrc/1367838067499/miraculix/Miraculix.jpg

From Pex to test cases



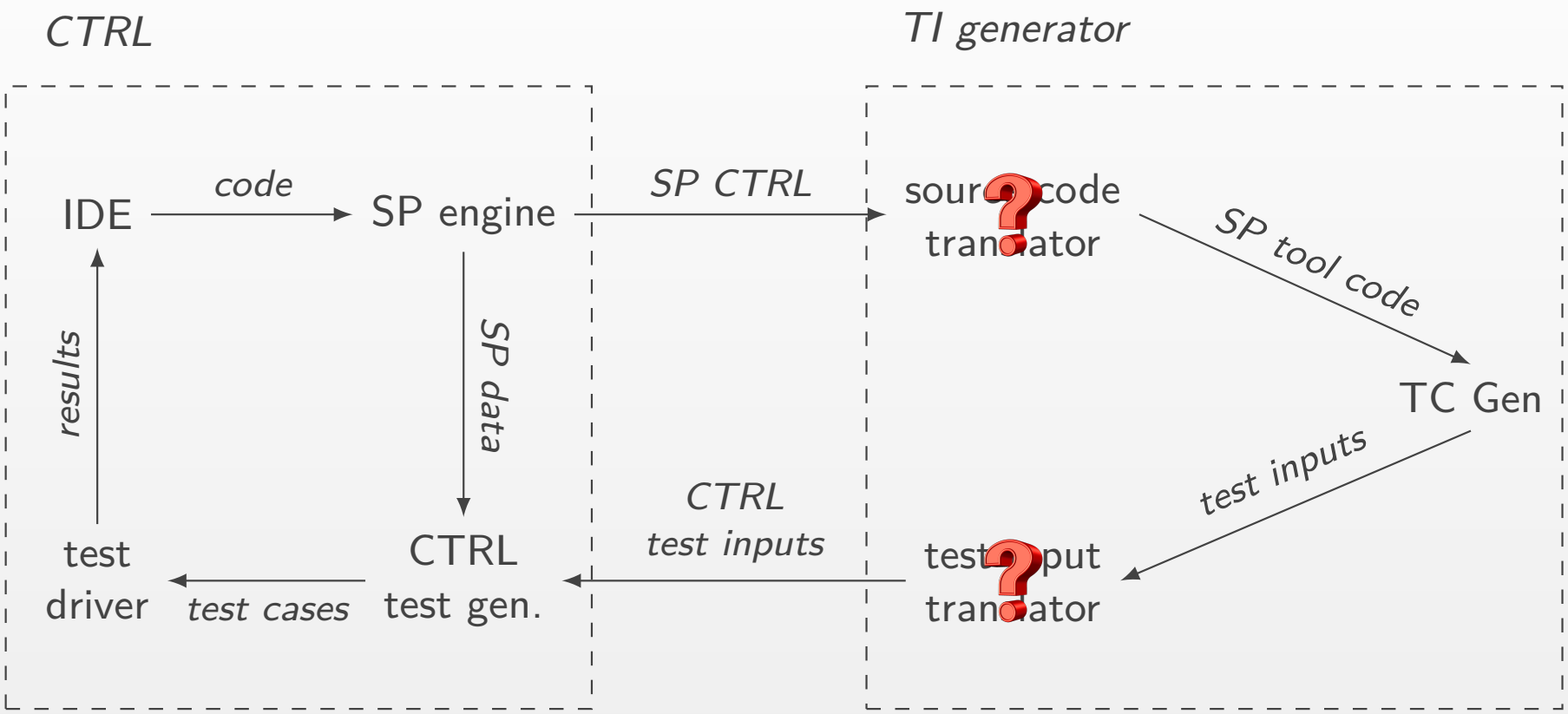
Test case generation from Pex output

How are we doing it?



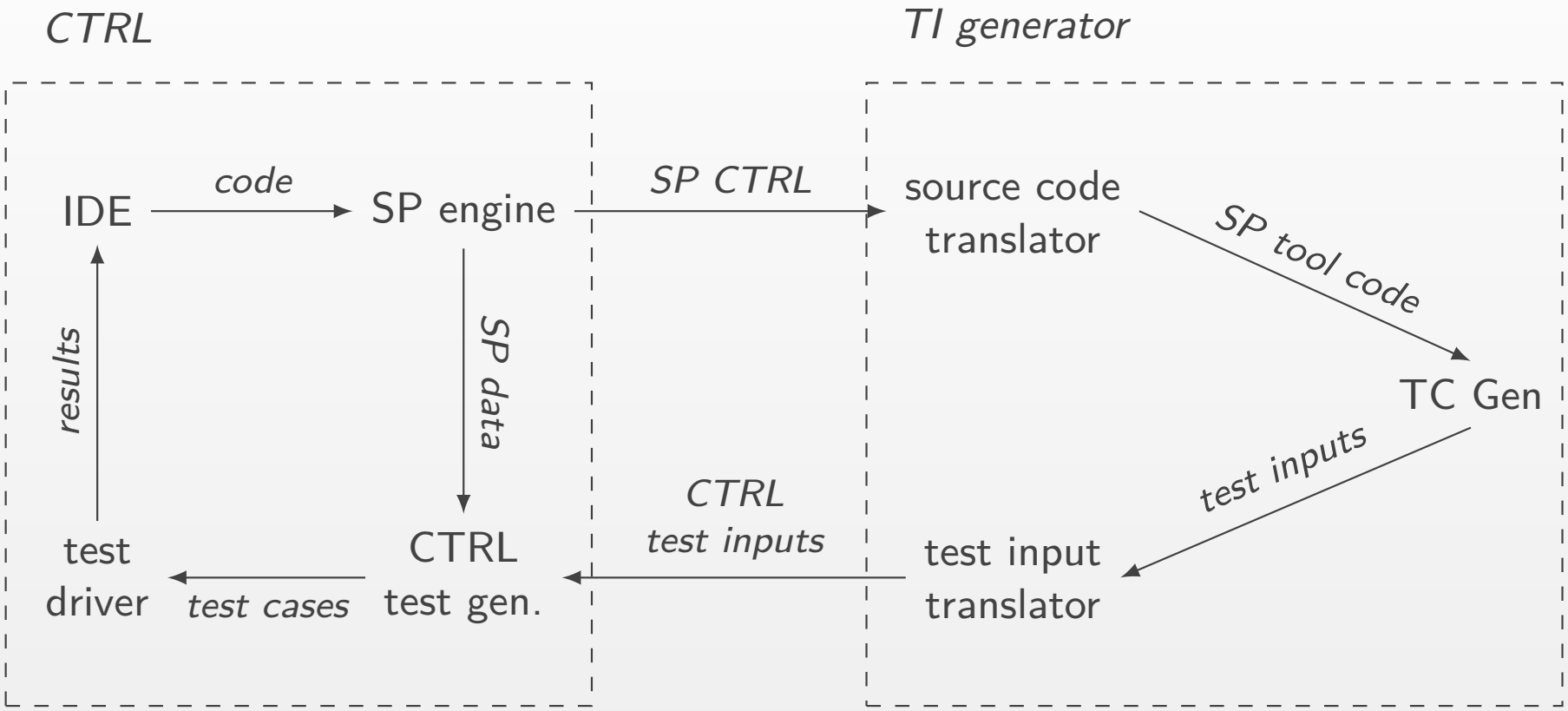
ITEC workflow

How are we doing it?



ITEC workflow

How are we doing it?



ITEC workflow

Automated Test Case Generation for CTRL using Pex: Lessons Learned

Klikovits et. al., Proc. SERENE 2016, Gothenburg

How to test translation?

How to test translation?

Divide

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

How to test translation?

Divide

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

Anonymise

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

How to test translation?

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 translation?

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/336784878358770673/

How to test translation?

```
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 translation?

```
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/

How to test translation?

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

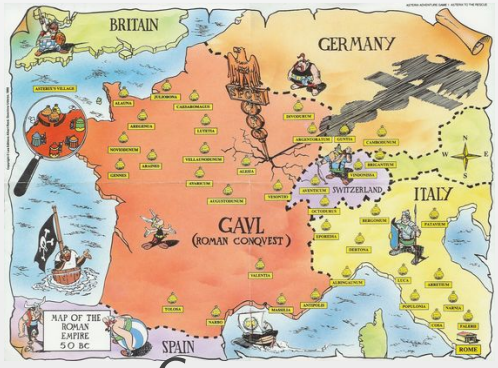
Divide

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

Anonymise

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

Analyse Blocks



Conquer

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

How to test translation?

```
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

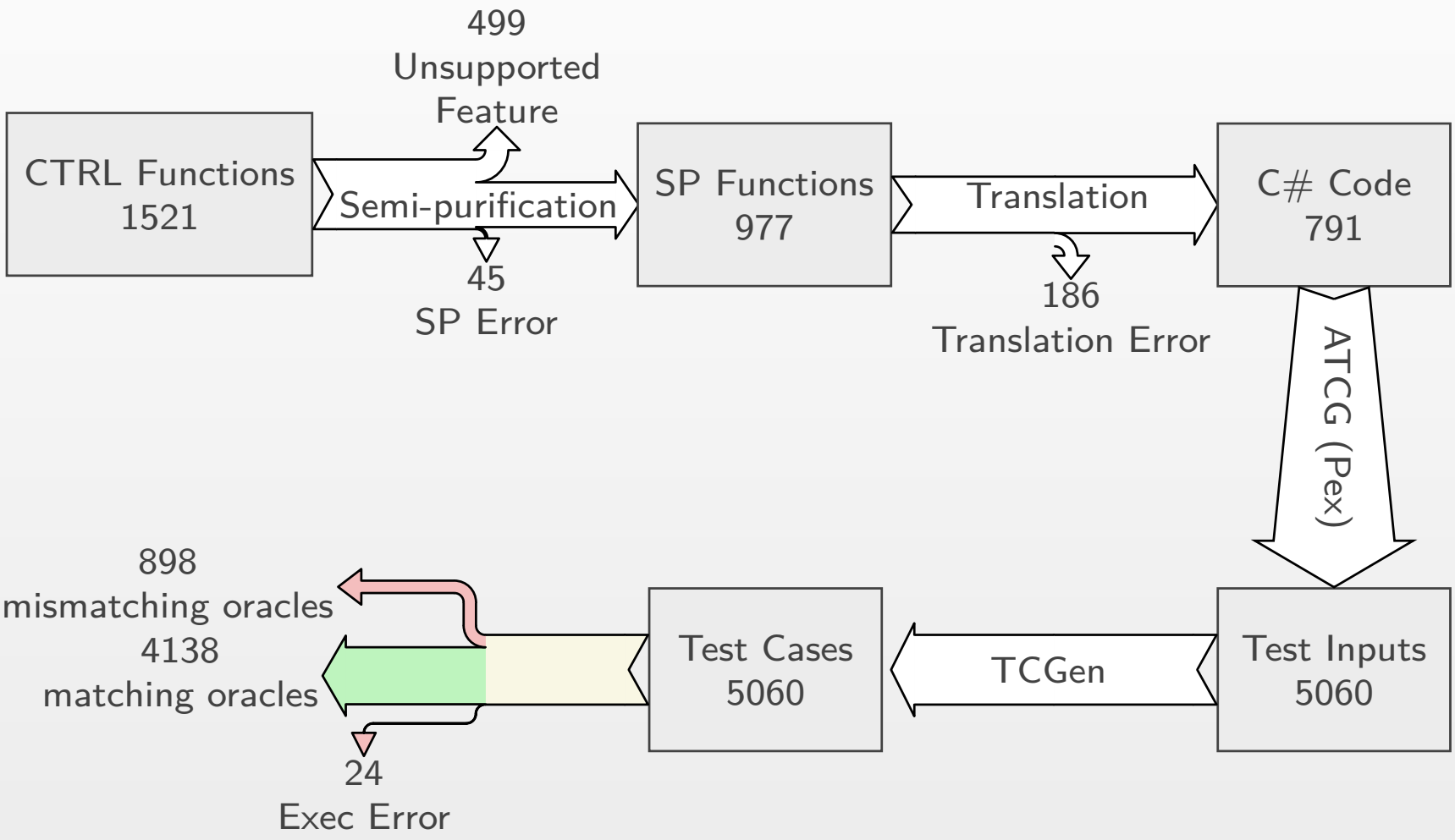
```
int func(int a, int b) {
  int++                  1
  int++                  1
  int = int+int          1
  if(int > int){         0
    return int % int     0
  } else {
    return int + int     1
  }
}
```

Analyse Blocks

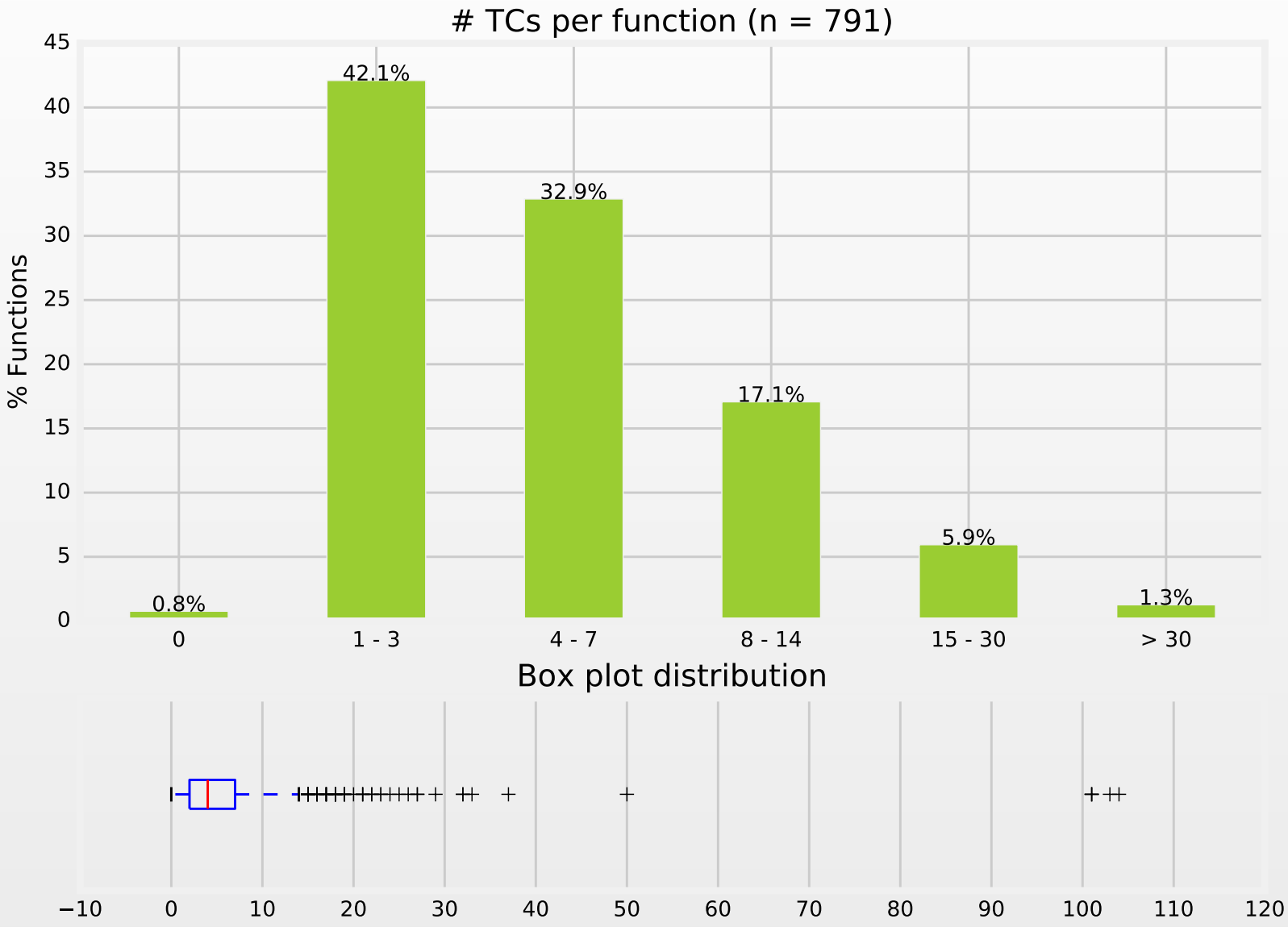
$$\phi = \frac{\sum \phi(L_i)}{|L|}$$

Conquer

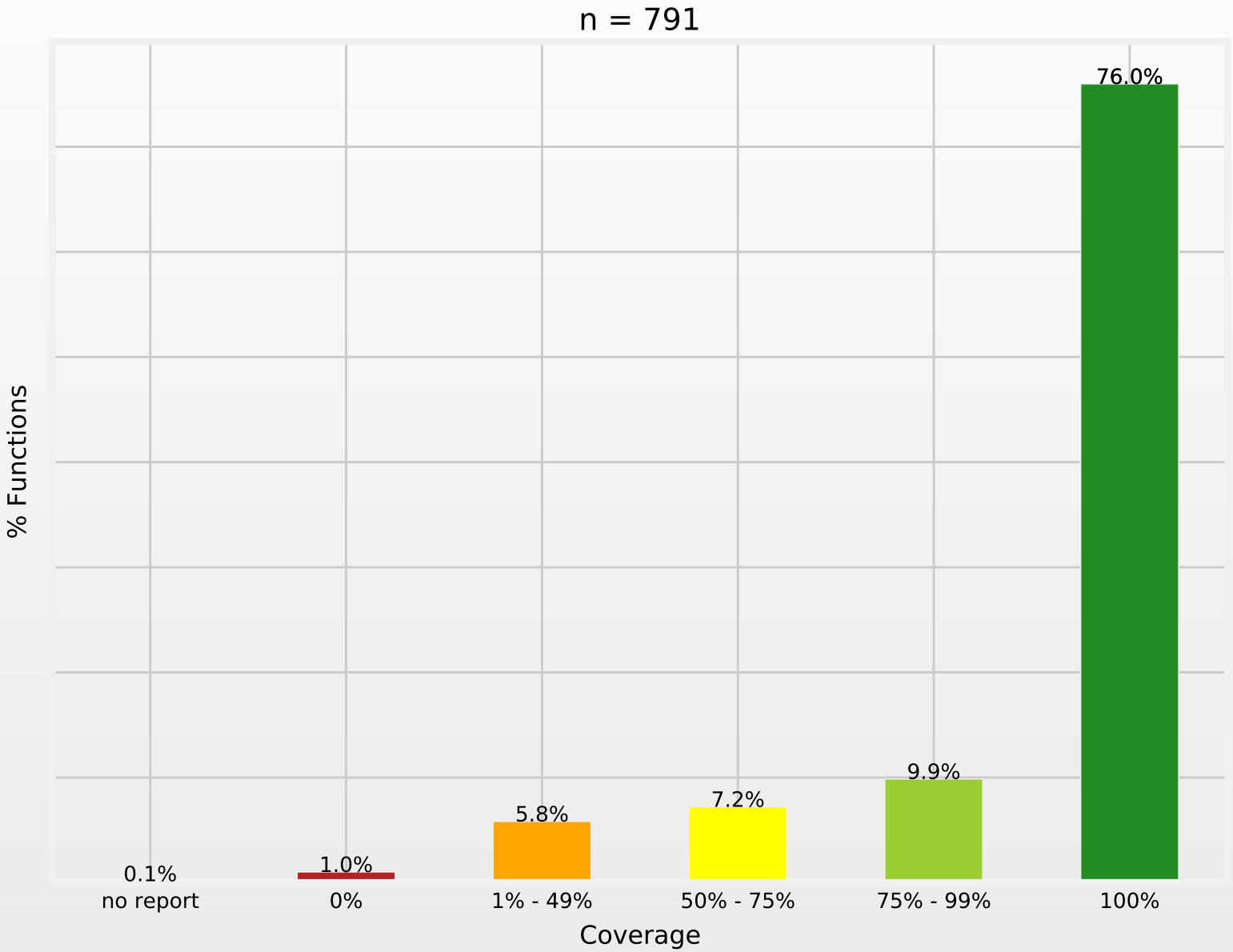
Test case generation: results



Number of Test Cases



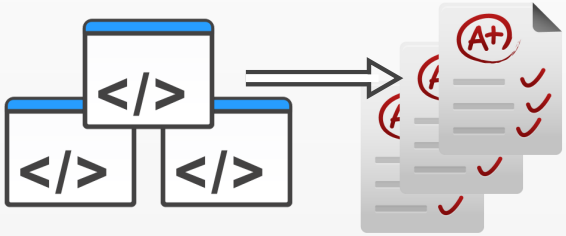
Coverage


n = 791
% Functions
0.1%
1.0%
5.8%
7.2%
9.9%
76.0%
no report
0%
1% - 49%
50% - 75%
75% - 99%
100%
Coverage

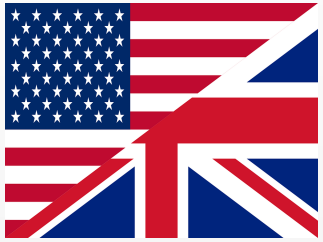
Lessons learned

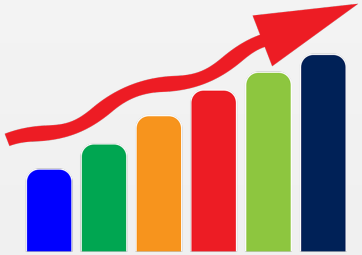
- not everything can be translated (easily)
- *nor should it ... (?)*
- C# is no silver bullet
- improving the quality of test cases ?
- tools have “features”

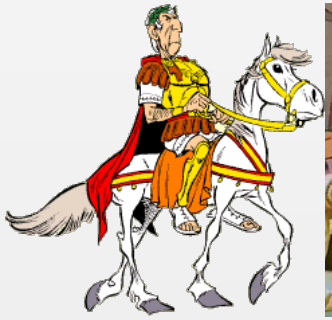
Summary

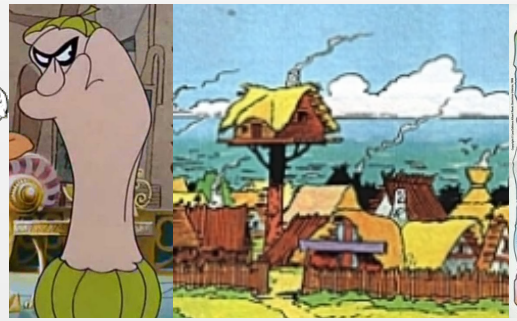

A+


CLACK!










BRITAIN
GERMANY
SWITZERLAND
ITALY
GAVL
(ROMAN CONQVEST)
SPAIN
MAP OF THE ROMAN EMPIRE 50 BC

[1]University of Geneva, Switzerland
[2]CERN, Geneva, Switzerland

Towards Language Independent (Dynamic) Symbolic Execution

Stefan Klikovits [1,2]

Manuel Gonzalez-Berges [2]
Didier Buchs [1]

stefan.klikovits@unige.ch

What next?

- expand TC generation
- other/different use cases
- trade-off complexity vs. usefulness
- research unsupported features