

# SubWeb

Search based path and Input Data  
Generation for Web Application Testing

Matteo Biagiola, Paolo Tonella, Filippo Ricca



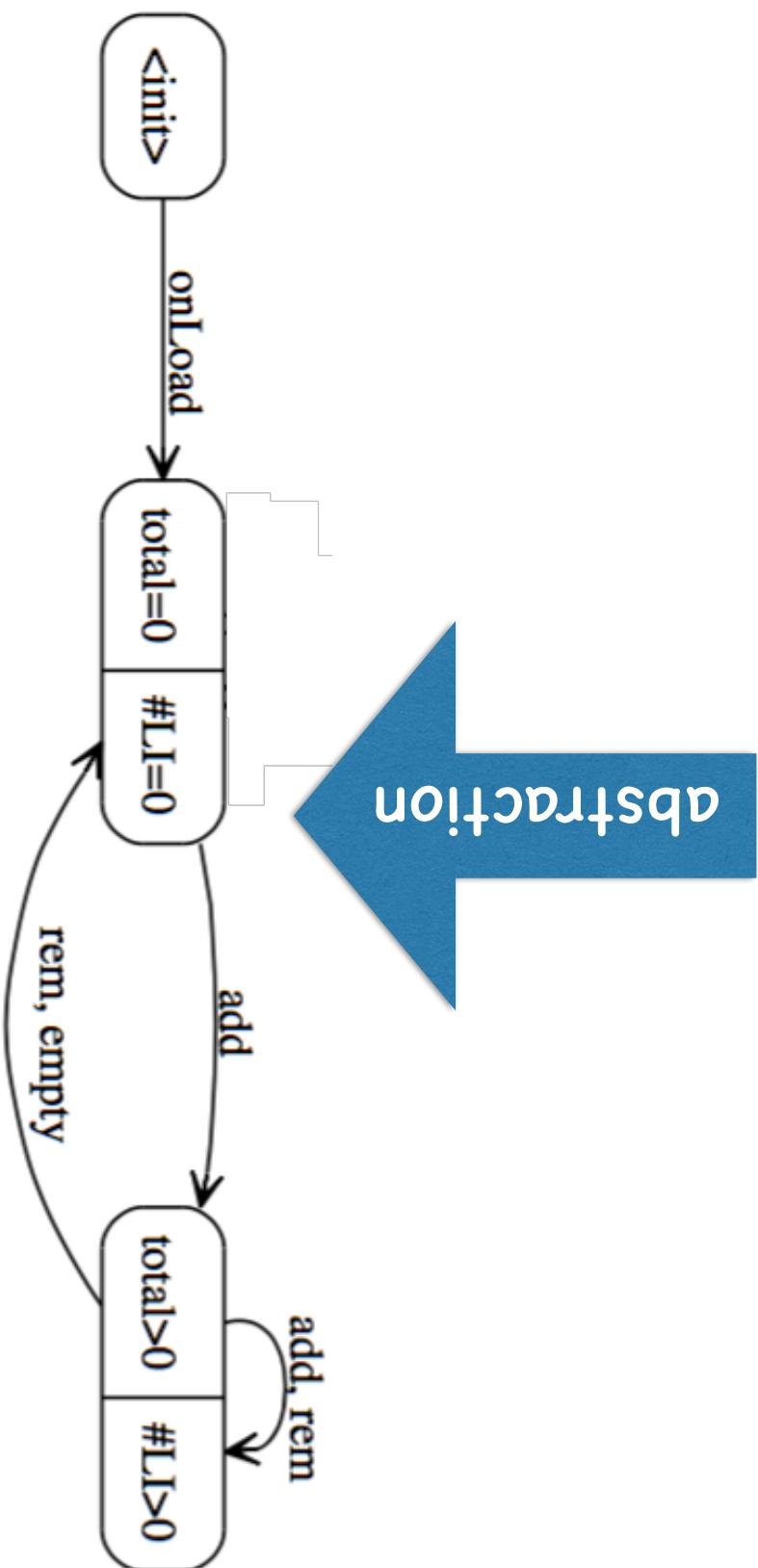
FONDAZIONE  
BRUNO KESSLER



UNIVERSITÀ DEGLI STUDI  
DI GENOVA

# Why model-based testing?

- multi-tier architecture
- asynchronism
- different technologies



# Problem definition

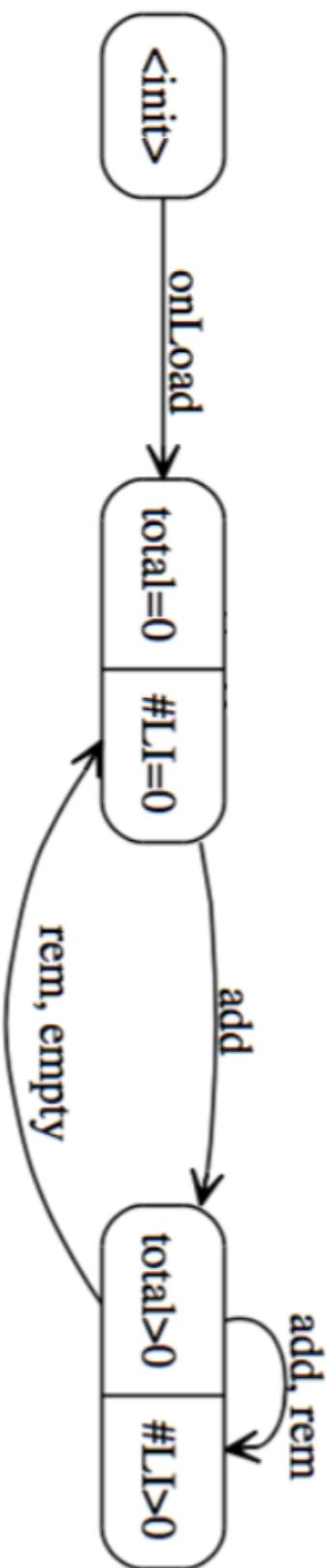
## 1) Getting the model

- manual creation
- automatic extraction

3

## 2) Using the model

- test sequences generation
- test data generation
- test case feasibility



# Existing approaches



- 1) Getting the model
  - automatic creation: web crawling based
- 2) Using the model
  - test sequences generation: graph visit algorithms
  - test data generation: manual or random
  - test case feasibility: not considered

# Our approach

## SubWEB

(Search Based WEB test generator)

- 1) Getting the model
  - extraction from Page Objects
- 2) Using the model
  - joint generation of feasible test sequences and test data

# PO design pattern (Web Testing)

App API

```
selectProductWithName()  
getPrice()  
updateProductPrice('52$')
```

Page Objects

```
Product  
Page  
ProductList  
Page
```

► reduce fragility  
► increase reusability



HTML API

```
findElementsWithXpath('/html/div/h2/table/div[3]')  
getText()  
click()
```

```
setText('52$')
```

HTML  
Wrapper

```
product: Headphones  
price: 24$
```

```
product: DVD-RW  
price: 2$
```

# Shopping cart running example

Your Shop Name Products purchased

Shop Name

Category 1

Category 2

Category 3

0 - Items

Thumbnail	Name	Price	Description
320x150	Bitchip	\$29.89	praesent lectus vestibulum quam sapien varius ut blandit
320x150	Zathin	\$42.84	egert eleifend luctus ultricies eu nibh quisque id justo
320x150	Flexidly	\$35.43	donec quis orci egest orci vehicula condimentum curabitur
320x150	Tempsoft	\$47.90	tempus sit amet sem fusce consequat nulla nisi
320x150	Solarbreeze	\$46.66	accumsan odio curabitur convallis duis consequat dui

7

# Navigation model – specification via Page Objects

```
public class ProductsPage implements PageObject {  
    public WebDriver driver;  
    public ProductsPage(WebDriver driver) {...}  
    public int getActiveCategory() {...}  
  
    public ProductDetailPage selectProduct(int id, int category) {  
        if((id >= 1 && id <= 6) &&  
            (category >= 1 && category <= 3) &&  
            (this.getActiveCategory() == category)) {  
            this.driver.findElement(By.id("product-" + id + "-" +  
                category)).click();  
            return new ProductDetailPage(this.driver);  
        } else {  
            throw new IllegalArgumentException("Invalid parameter values");  
        }  
    }  
}
```

2nd assumption

1st assumption

# Navigational model – automatic extraction

```
public class ProductsPage implements PageObject {  
    public WebDriver driver;  
    public ProductsPage(WebDriver driver) {...}  
    public int getActiveCategory() {...}
```

```
    public ProductDetailPage selectProduct(int id, int category) {  
        if((id >= 1 && id <= 6) &&  
            (category >= 1 && category <= 3) &&  
            (this.getActiveCategory() == category)) {  
            this.driver.findElement(By.id("product-" + id + "_" +  
                category)).click();  
        }  
        return new ProductDetailPage(this.driver);  
    }  
    else {  
        throw new IllegalArgumentException("Invalid parameter values");  
    }  
}
```

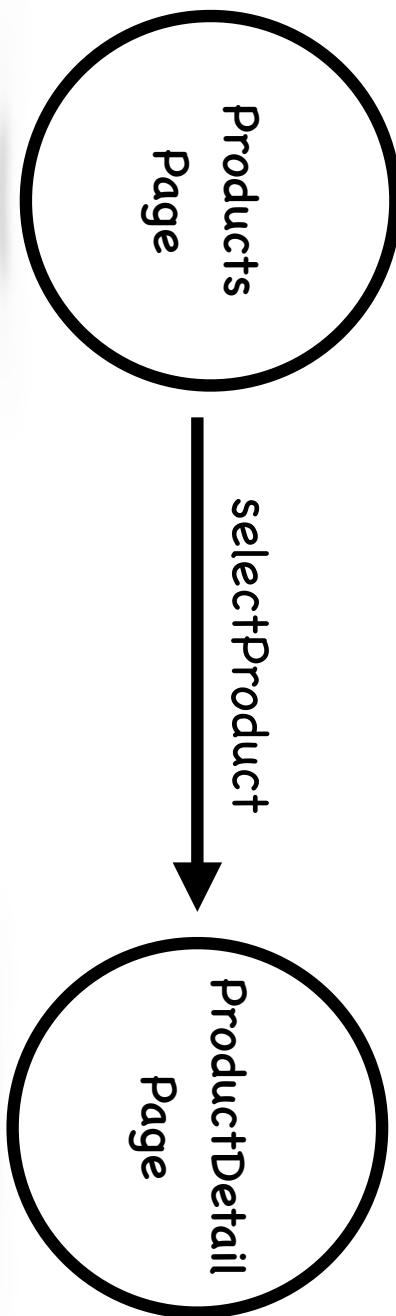
Products  
Page

selectProduct

ProductDetail  
Page

# Path

$$p = \langle ns, es, pr \rangle \quad ns \in N^+ \quad es \in E^* \quad pr \in V^*$$



$ns = \langle \text{ProductsPage}, \text{ProductDetailPage} \rangle$   $pr = \langle \text{id}, \text{category} \rangle$

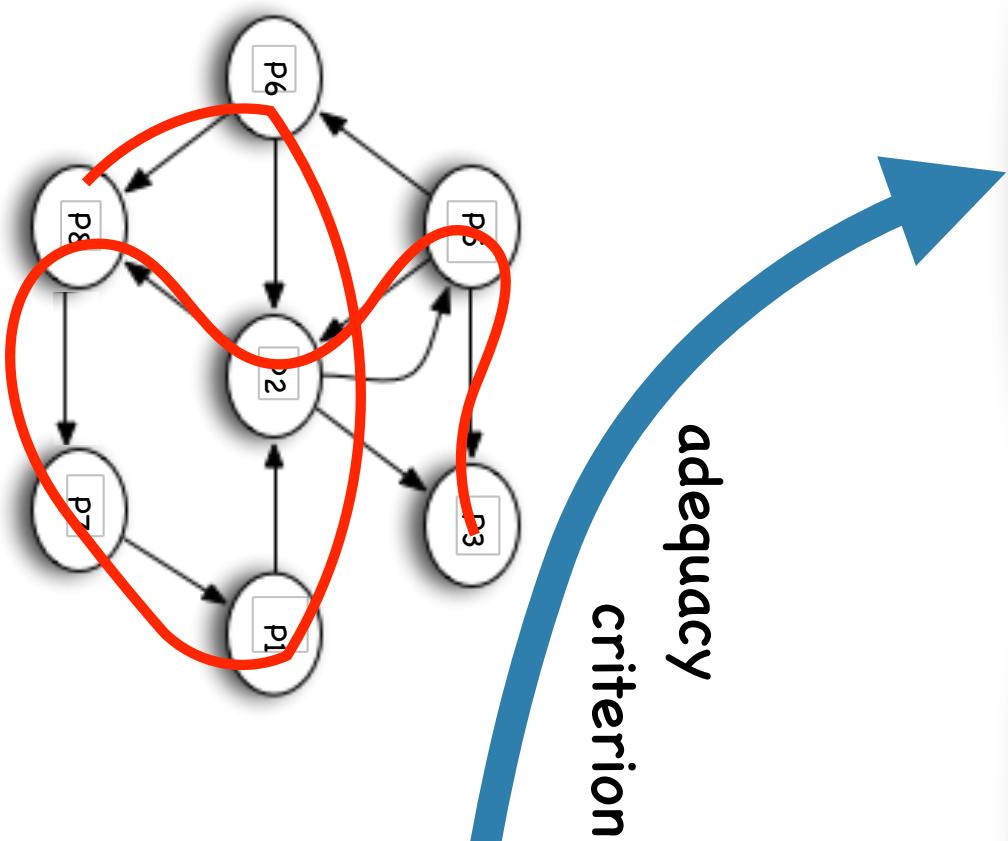
$es = \langle \text{selectProduct} \rangle$

# Test generation problem

Navigation graph

Paths

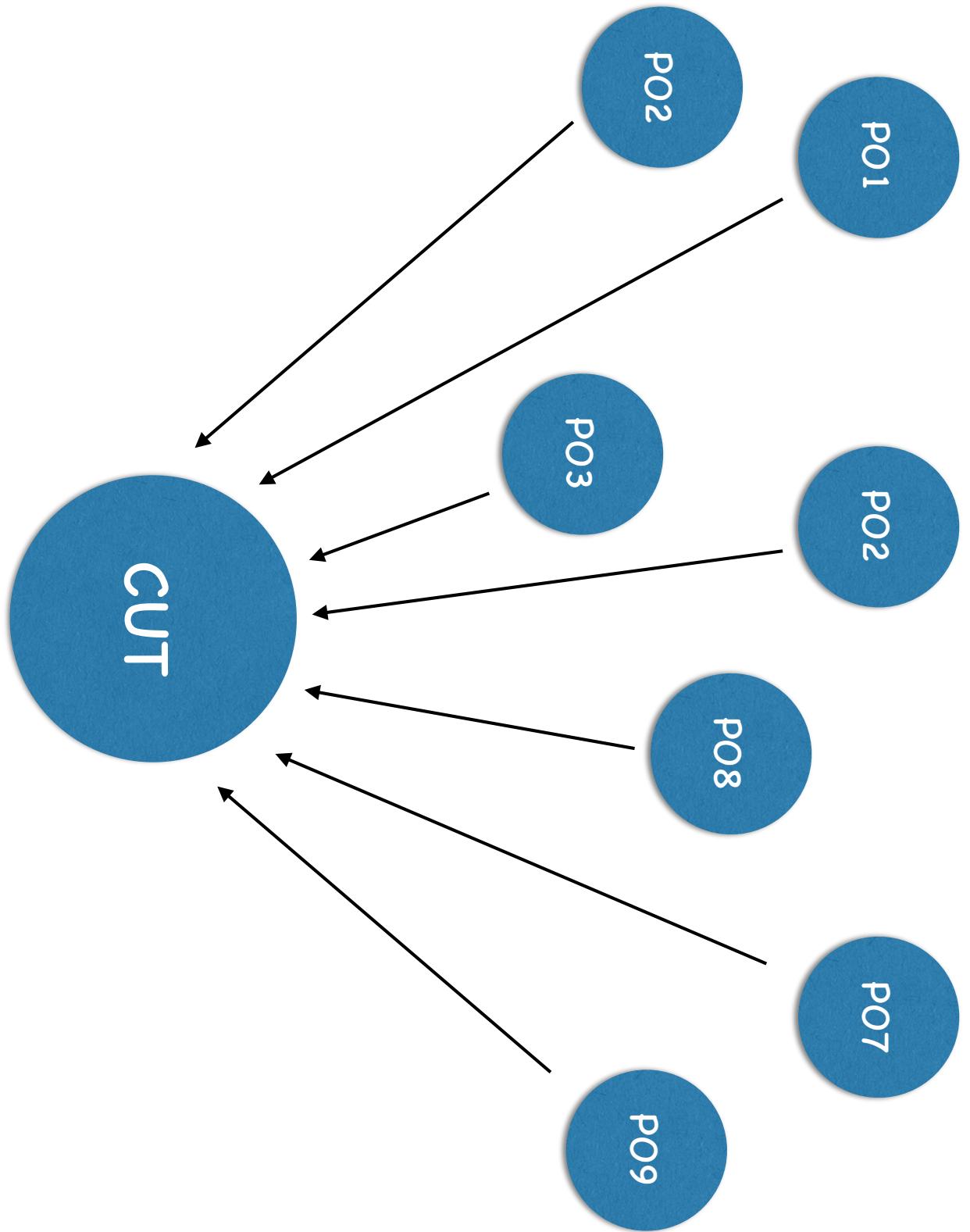
Executable Paths



# Test generation problem

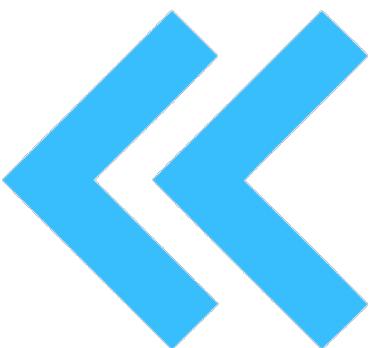


# Problem reformulation – program transformation

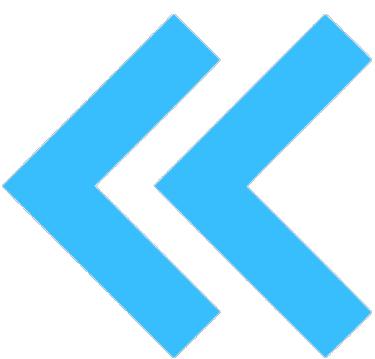


# Problem reformulation – program transformation

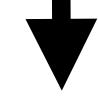
```
public class ProductsPage implements PageObject {  
    public WebDriver driver;  
    public ProductsPage(WebDriver driver) {...}  
    public int getActiveCategory() {...}  
  
    public ProductDetailPage selectProduct(int id, int category) {  
        if((id >= 1 && id <= 6) &&  
            (category >= 1 && category <= 3) &&  
            (this.getActiveCategory() == category)) {  
            this.driver.findElement(By.id("product-" + id + "-" +  
                category)).click();  
            return new ProductDetailPage(this.driver);  
        } else {  
            throw new IllegalArgumentException("Invalid parameter values");  
        }  
    }  
}
```



# Problem reformulation – program transformation



```
public class CUT {  
    private PageObject currentPage;  
    public void selectProduct(Id id, Category category) {  
        if (this.currentPage instanceof ProductsPage) {  
            ProductsPage page = (ProductsPage) this.currentPage;  
            if (page.getActiveCategory() == category.value){  
                this.currentPage.findElement(By.id("product-" + id.value + " -" + category.value)).click();  
            } else { throw new IllegalArgumentException("Invalid parameter values"); }  
        } else { throw new IllegalArgumentException("You are not in the right page"); }  
    }  
    ...  
}
```



```
}
```

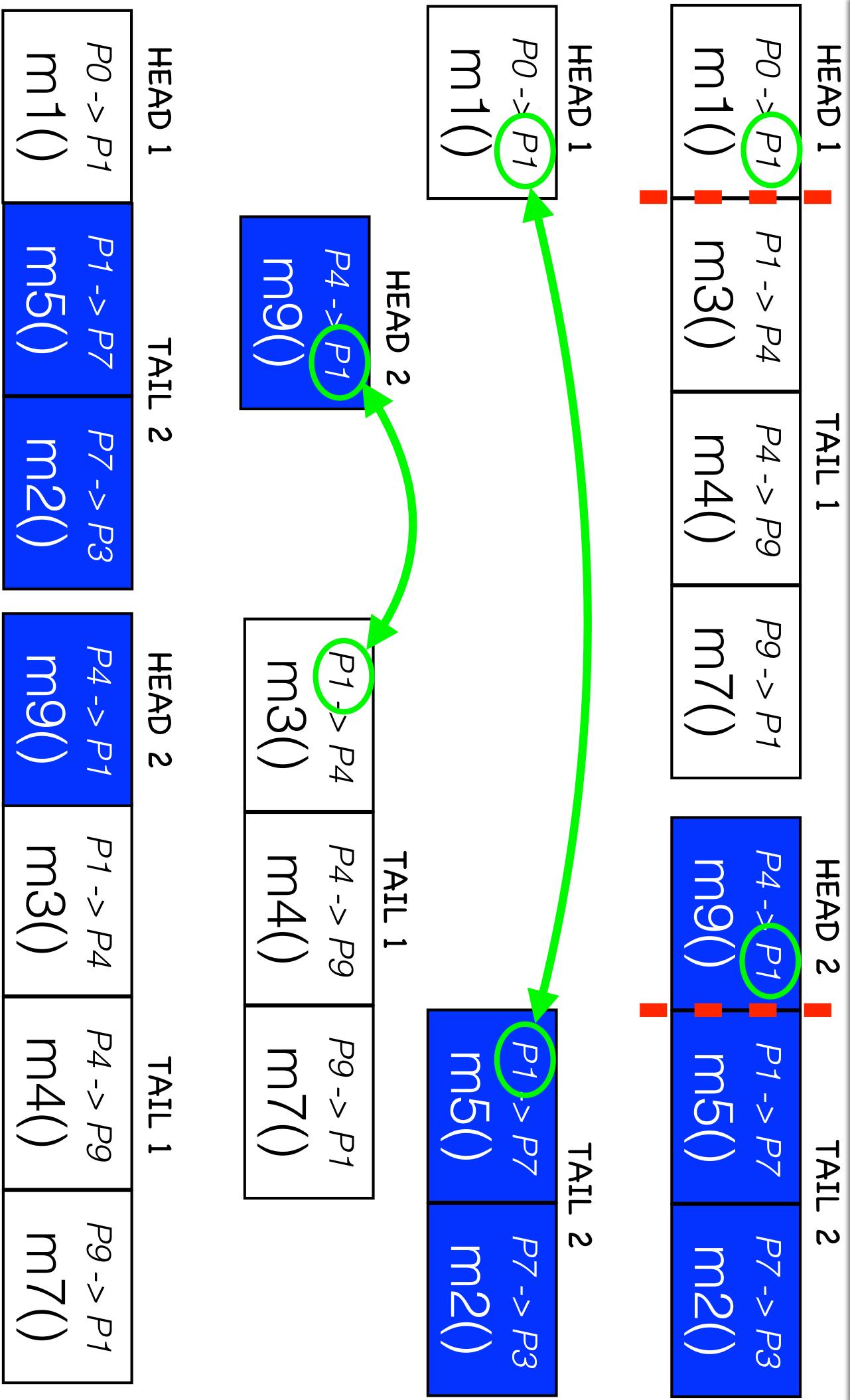
# Problem reformulation - program transformation



```
public class CUT  
private PageC  
public void s  
if (this.c  
Product.  
if (page  
page.  
this.  
} else .  
} else { t  
}  
...  
}
```

```
alue) .click();  
}; }  
}
```

# Genetic operators – crossover targets matched



# Genetic operators – mutation

## Starting chromosome

$P0 \rightarrow P1$	$P1 \rightarrow P4$	$P4 \rightarrow P9$	$P9 \rightarrow P1$	$P4 \rightarrow P11$
$m1()$	$m3()$	$m4()$	$m7()$	$m8()$

## Delete

$P0 \rightarrow P1$	$P1 \rightarrow P4$	$P4 \rightarrow P9$	$P9 \rightarrow P1$	$P4 \rightarrow P11$
$m1()$	$m3()$	$m4()$	$m7()$	$m8()$

## Insert

$P0 \rightarrow P1$	$P1 \rightarrow P4$	$P4 \rightarrow P9$	$P9 \rightarrow P1$	$P4 \rightarrow P11$
$m1()$	$m3()$	$m4()$	$m7()$	$m8()$

Diagram illustrating the mutation process:

- The starting chromosome is shown in the top table.
- The mutation process involves two main steps: **Delete** and **Insert**.
- Delete:** A red arrow points from the  $P1 \rightarrow P4$  segment to a large red X, indicating its removal.
- Insert:** A green arrow points to the  $P4 \rightarrow P11$  segment, which is circled in green. This segment is then moved to the  $P1 \rightarrow P4$  position.
- Further Insertions:** Subsequent mutations involve moving segments between positions. A red arrow points to the  $P11 \rightarrow P8$  segment, which is circled in red and moved to the  $P1 \rightarrow P4$  position. Another red arrow points to the  $P8 \rightarrow P7$  segment, which is circled in pink and moved to the  $P4 \rightarrow P9$  position. A final red arrow points to the  $P7 \rightarrow P12$  segment, which is circled in pink and moved to the  $P9 \rightarrow P1$  position.
- The final mutated chromosome is shown in the bottom table, where the original segments have been replaced by the newly inserted ones.

# Results – manual cost

## App: AddressBook

PHP LOC	Javascript LOC
30223	1288

non trivial  
POs

POS LOC	Number of POs	Navig. methods
764	1288	73

↑  
2%  
Preconds

Methods LOC	Total number	Logic operators
75	16	54

0.2%

# Results - models

PO graph

States	Transitions
12	73

Crawled graph

States	Transitions	Missing states transitions
329	927	0

27x states  
13x transitions

# Results - test suite size

## SuBWEB

Test cases	Divergent test cases
54	0

## Ext-Crawljax

Test cases	Divergent test cases
598	104

**11x size**



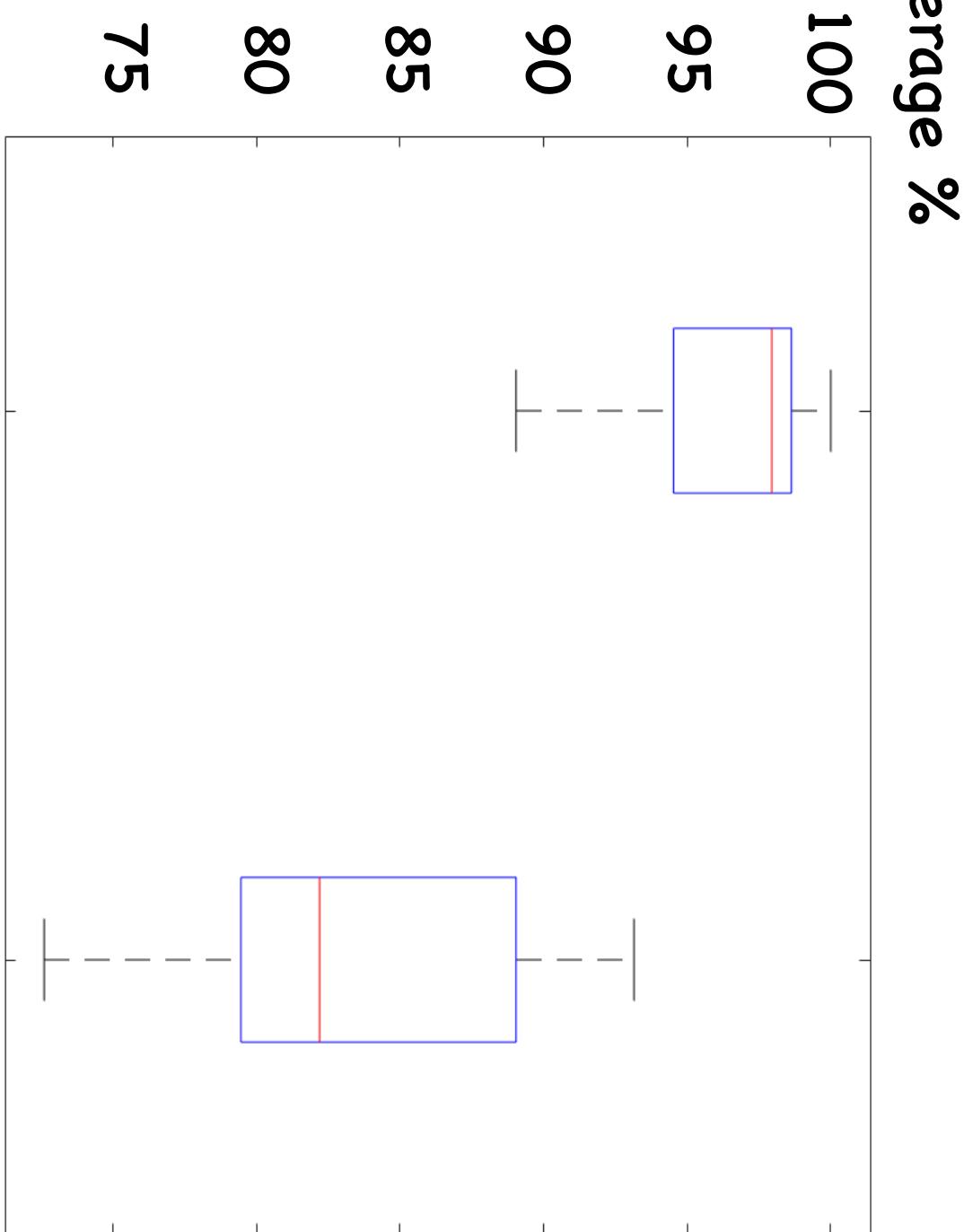
**17%**



Divergence happens if an element existing at crawling time is no longer found at test time (different state): step in test case triggers an error so the desired path cannot be followed.

# Results - coverage

SubWEB      Ext-Crawljax



Mann-Whitney U test

p-value

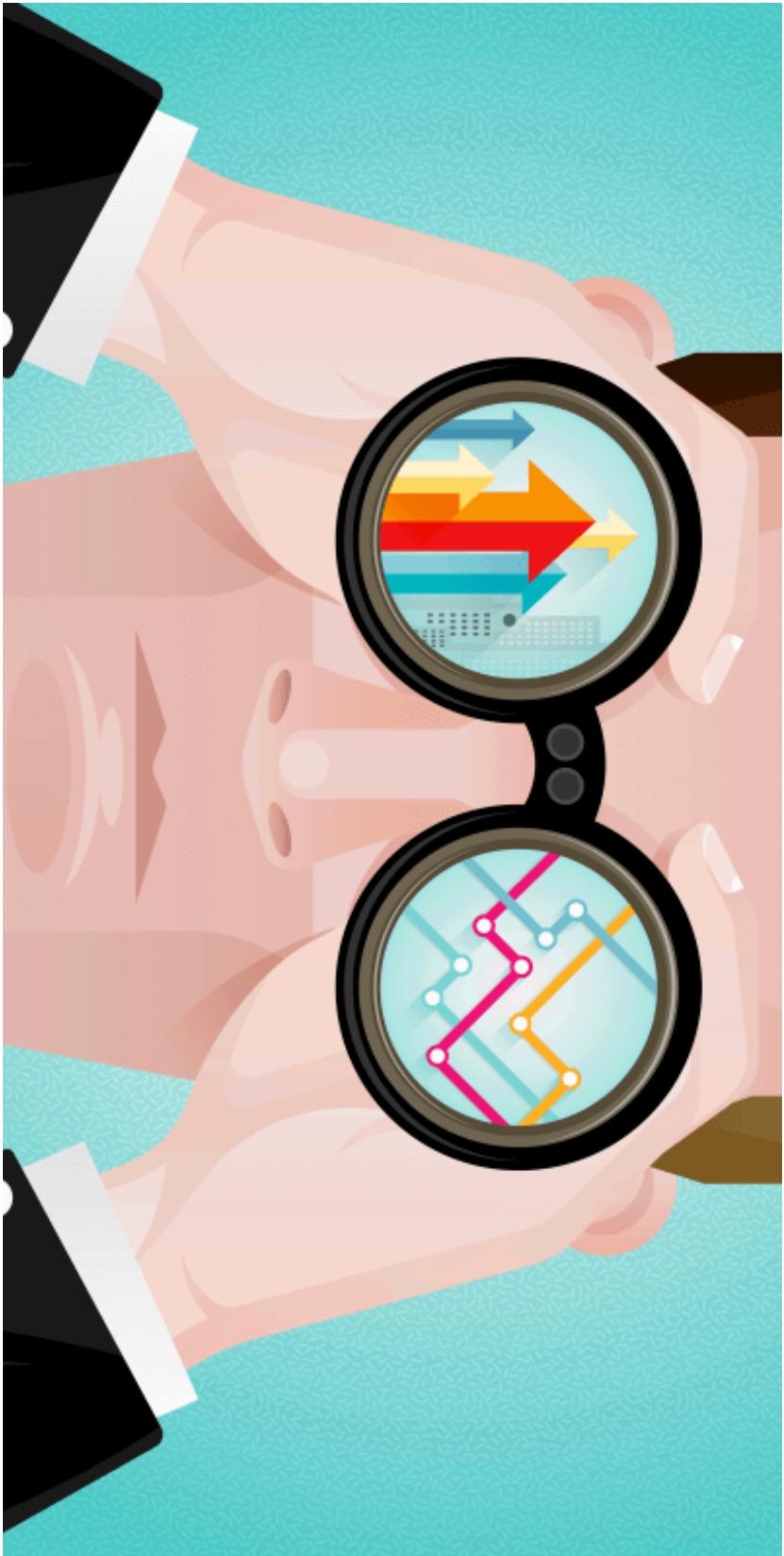
$4.70 \times 10^{-4}$

# Conclusions with comparisons

- ♦ Manual step
- ♦ Navigation graph is much smaller
- ♦ Smaller size test suites
- ♦ No divergency
- ♦ Higher transition coverage



# Future works



Automatic generation of preconditions

Automatic generation of assertions

Thank  
you

# My supervisors



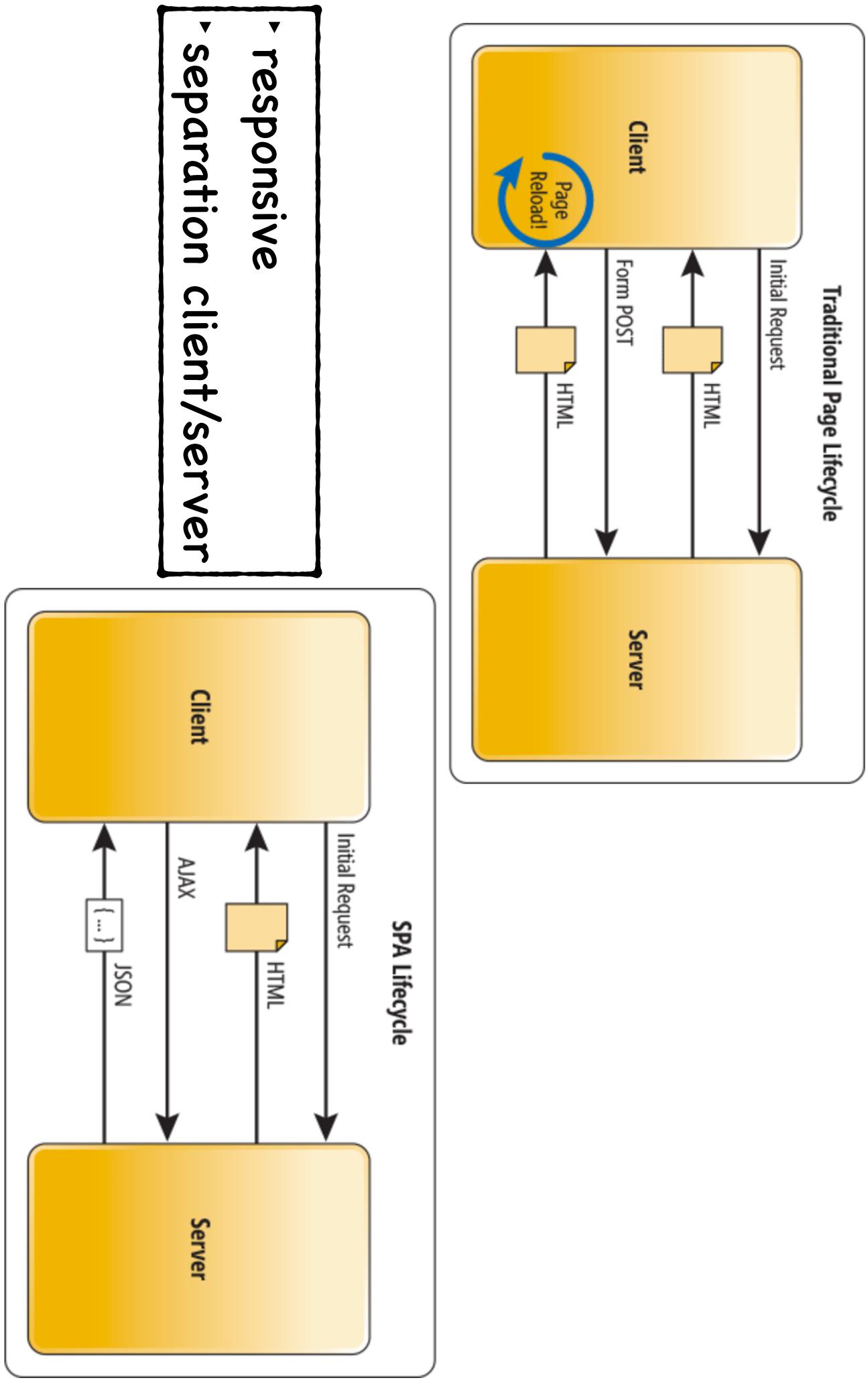
Paolo Tonella



Filippo Ricca



# Web applications



# Research questions

SubWEB

RQ1

Manual cost

POs preconditions

SubWEB vs Ext-Crawljax

RQ2

Navigation graph

States, transitions

RQ3

RQ4

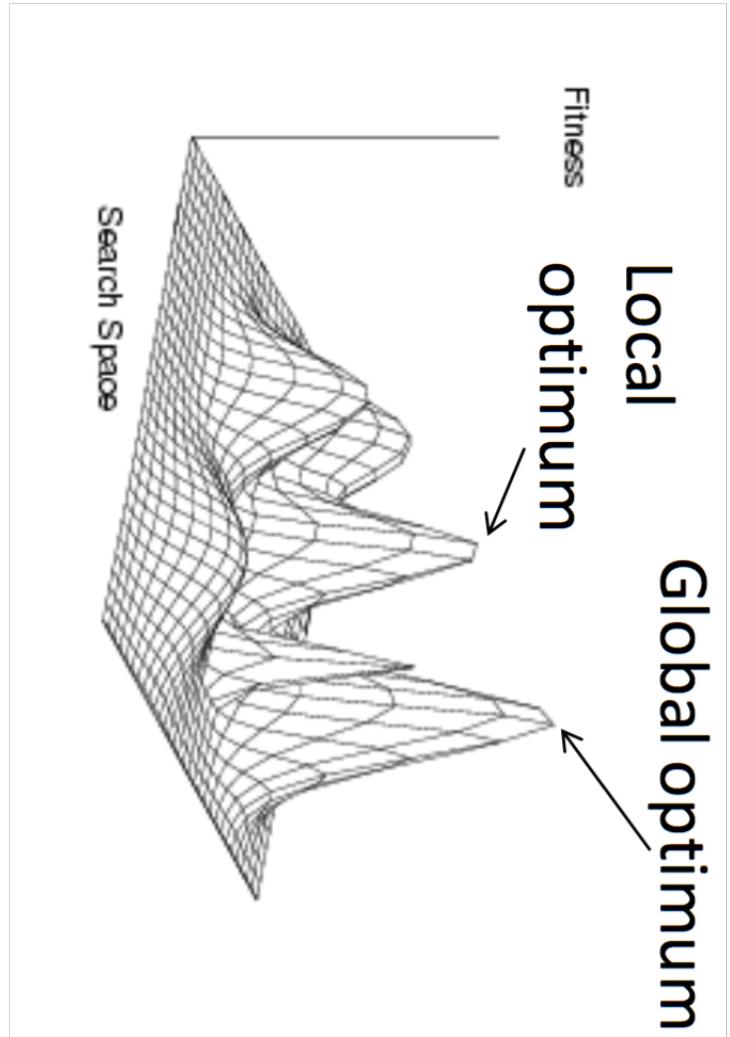
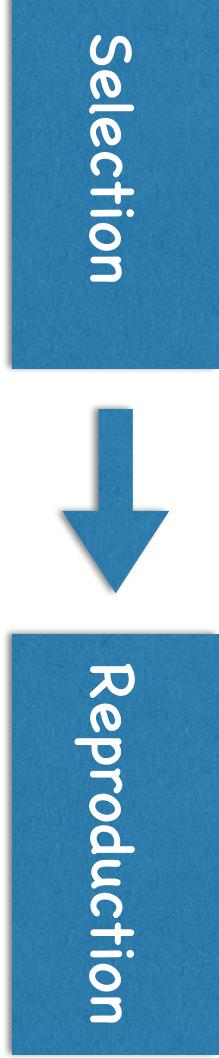
Test suite features

Coverage

Size, % divergent

Navigation graph

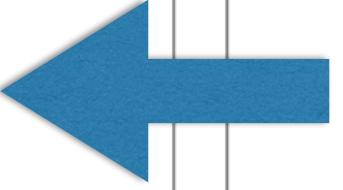
# Genetic algorithms



# Problem reformulation – program transformation

```
public class ProductsPage implements PageObject {  
    public WebDriver driver;  
    public ProductsPage(WebDriver driver) {...}  
    public int getActiveCategory() {...}
```

```
public ProductDetailPage selectProduct(int id, int category) {  
    if (id >= 1 && id <= 6) &&  
        (category >= 1 && category <= 3) &&  
        (this.getActiveCategory() == category) {  
            this.driver.findElement(By.id("product-" + id + "-" +  
                category)).click();  
            return new ProductDetailPage(this.driver);  
        } else {  
            throw new IllegalArgumentException("Invalid parameter values");  
        }  
    }  
  
public class CUT {  
    private PageObject currentPage;  
    public void selectProduct(Id id, Category category) {  
        if (this.currentPage instanceof ProductsPage) {  
            ProductsPage page = (ProductsPage) this.currentPage;  
            if (page.getActiveCategory() == category.value){  
                page.driver.findElement(By.id("product-" + id.value + "-" + category.value)).click();  
                this.currentPage = new ProductDetailPage(page.driver);  
            } else { throw new IllegalArgumentException("Invalid parameter values"); }  
        } else { throw new IllegalArgumentException("You are not in the right page"); }  
    }  
}
```



# Branch distance

Condition $c = \text{atomic predicate}$	Distance $BD(c) = d / (d + 1)$
$a$	$d = \{0 \text{ if } a == \text{true}; K \text{ otherwise}\}$
$!a$	$d = \{K \text{ if } a == \text{true}; 0 \text{ otherwise}\}$
$a == b$	$d = \{0 \text{ if } a == b; \text{abs}(a - b) + K \text{ otherwise}\}$
$a != b$	$d = \{0 \text{ if } a != b; K \text{ otherwise}\}$
$a < b$	$d = \{0 \text{ if } a < b; a - b + K \text{ otherwise}\}$
$a <= b$	$d = \{0 \text{ if } a <= b; a - b + K \text{ otherwise}\}$
$a > b$	$d = \{0 \text{ if } a > b; b - a + K \text{ otherwise}\}$
$a >= b$	$d = \{0 \text{ if } a >= b; b - a + K \text{ otherwise}\}$

# Genetic algorithm – chromosome and fitness

## chromosome

```
CUT cut0 = new CUT();
cut0.goToShoppingCartProductsPage();
cut0.goToShoppingCartShoppingCartPage();
cut0.goToHomeShoppingCartPage();
cut0.goToProductsPurchasedProductsPage();
cut0.goToHomeProductsPurchasedPage();

int int0 = (-2140);

Id id0 = new Id(int0);

Category category0 = new Category(id0.value);
cut0.selectProductProductsPage(id0, category0);
cut0.goToProductsPurchasedProductDetailPage();
cut0.goToHomeProductsPurchasedPage();
cut0.goToProductsPurchasedProductsPage();
cut0.continueShoppingProductsPurchasedPage();
```

## branch coverage criterion

$$\text{fitness}(T) = |M| - |M_T| + \sum_{b_k \in B} d(b_k, T)$$

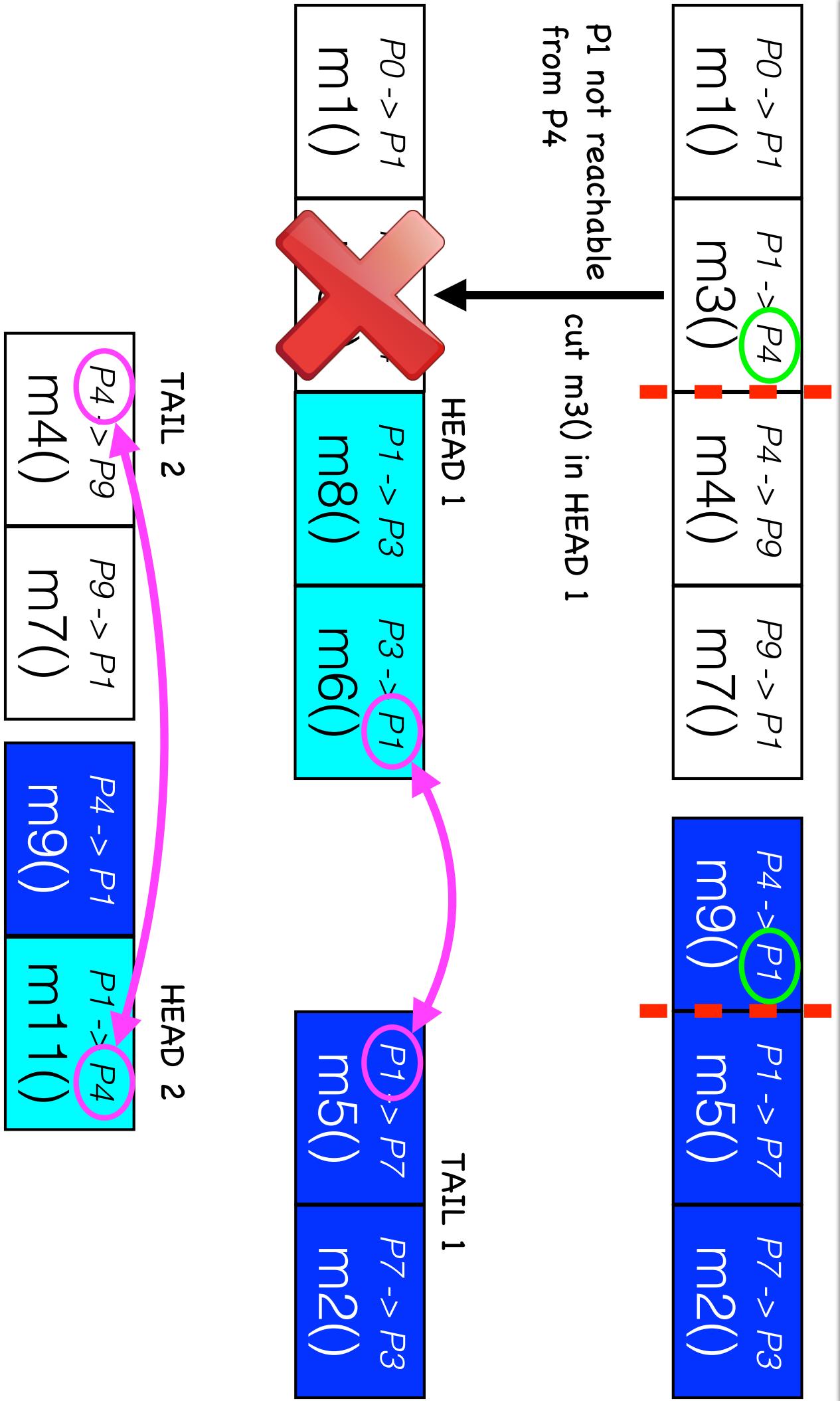
unexecuted

uncovered

methods

branches

# Genetic operators – crossover targets do not match



# Genetic operators – mutation insertion non reachability

