

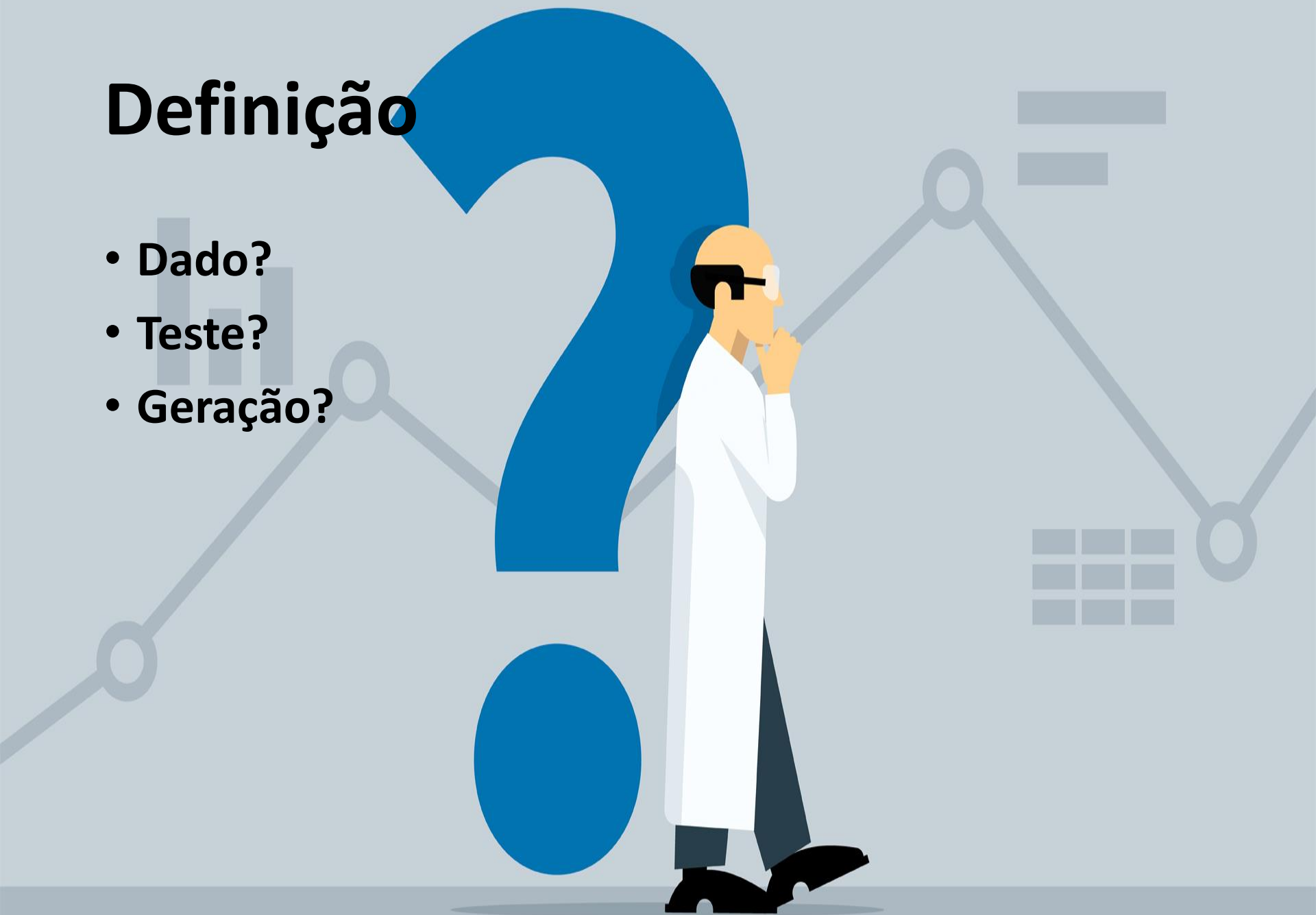
Geração de dados de teste

Ricardo F. Vilela
ricardovilela@usp.br

Simone R. S. Souza
srocio@icmc.usp.br

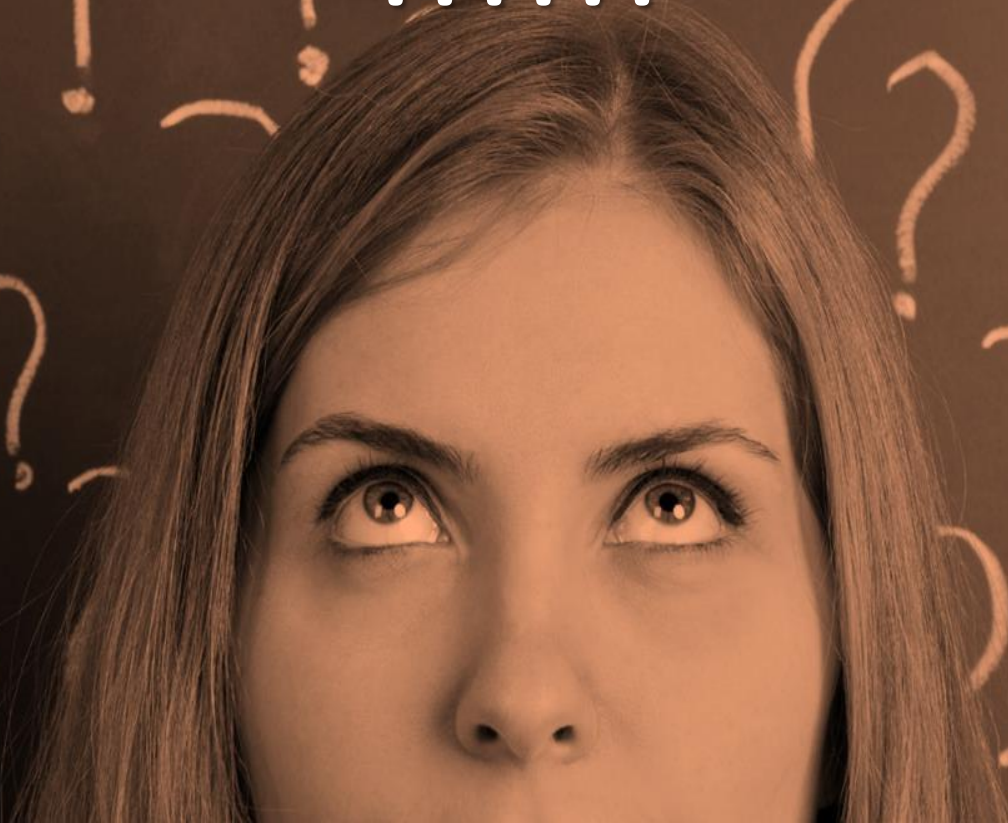
Definição

- Dado?
- Teste?
- Geração?



Espere.....

- **Dado de teste / Caso de teste
??????**



Cenário de teste



Objetivo do teste?

Objetivo do teste

- Revelar defeitos ainda não identificados!



Como testar?



Como testar?



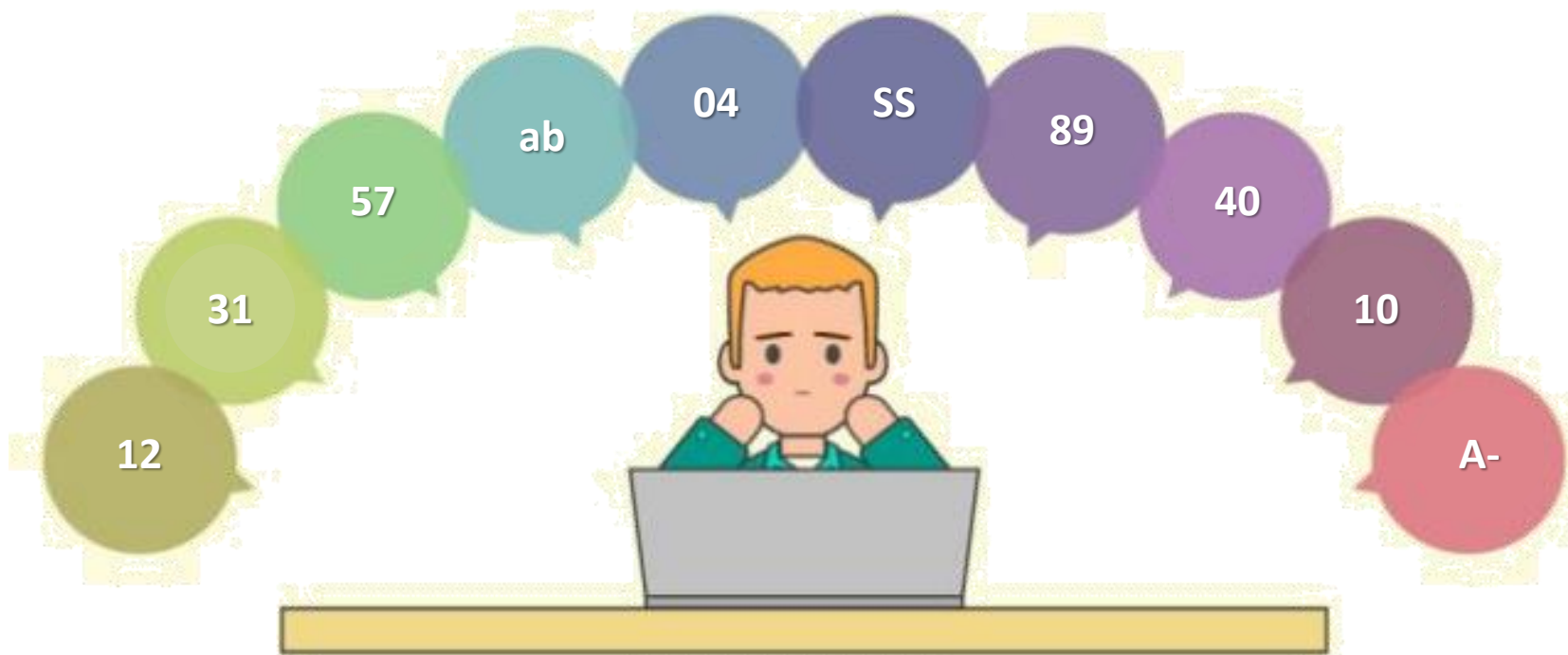
Como testar?



- 1+1
- 1+2
- 1+3
- .
- .
- .
- 1+N

Teste
exaustivo

Seleção de dados de teste



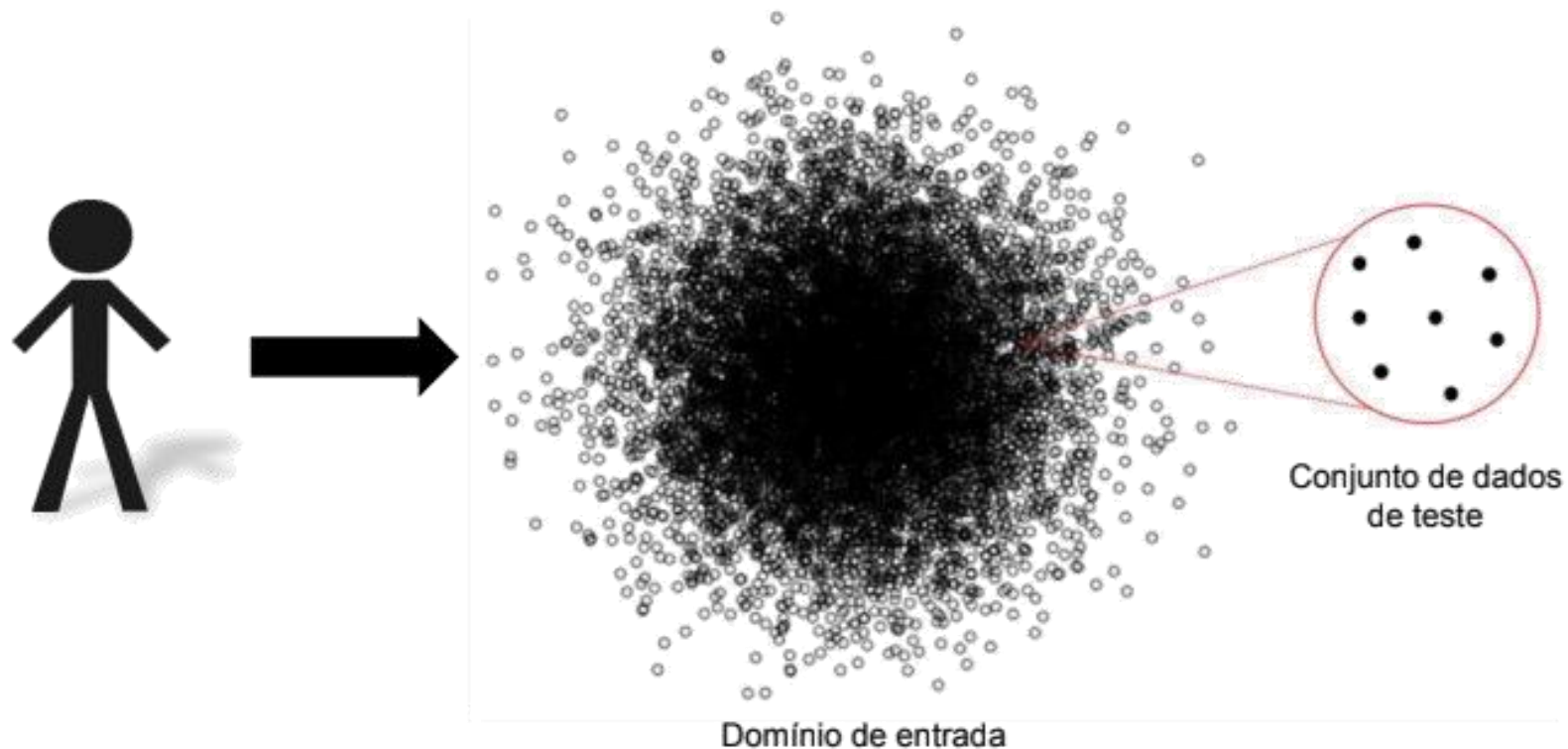
Seleção de dados de teste



Geração de dados de teste - Definição

- É um processo de identificação de **dados do domínio de entrada válidos** para um determinado programa de acordo com os **critérios de teste**.

Geração de dados de teste



Geração de dados de teste

- Na prática...
 1. Programador implementa a funcionalidade



Geração de dados de teste

- Na prática...
 2. O testador inicia os testes das funcionalidades implementadas e...



Geração de dados de teste

- Na prática...
 3. Gera os dados e executa os testes **MANUALMENTE**



Geração de dados de teste - Automatização

- Por que, então, **NÃO** automatizar essa atividade?



Geração de dados de teste - Automatização

- Por que, então, **NÃO** automatizar essa atividade?
 - Exemplo: <http://www.generatedata.com/>

Geração de dados de teste - Automatização

- Por que, então, **NÃO** automatizar essa atividade?
 - Exemplo: <http://www.generatedata.com/>

Muito simples???

Geração de dados de teste - Automatização

- Por que, então, **NÃO** automatizar essa atividade?
 - **Exemplo 2:**
<https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd>

Geração de dados de teste - Automatização

- Exercício
 - Grupos
 - Automação site: <http://testpar.icmc.usp.br/pessoas>
 - <https://www.seleniumhq.org/selenium-ide/docs/en/api/commands/>
 - Verificar Pesquisadores [Simone do Rocio Senger de Souza]
 - Verificar Alunos de mestrado: [Rodolfo] = Concluído
 - Verificar Alunos de doutorado [Silvana] = Concluído
 - Verificar Alunos de doutorado [Ricardo Ferreira Vilela]

Geração de dados de teste – Manual x Automática

Geração Manual	Geração Automática
Muitos dados de teste	Menor quantidade de dados de teste
Muito trabalhosa	Reduz significativamente o esforço
Demanda muito tempo	Mais rápida
Dispendiosa	O custo é reduzido
Sujeito a falhas	Alto grau de confiabilidade
	Deteção de regressões

Geração de dados de teste – Manual x Automática

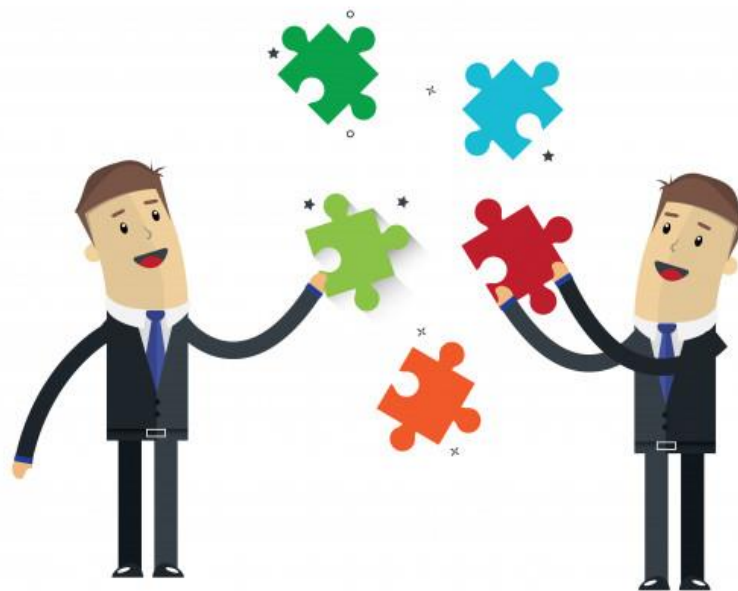
- Humanos não são bons em realizar a **mesma tarefa repetidamente**;
- Humanos não são bons em realizar a **mesma tarefa igualmente todas as vezes**;
- **Stress e fadiga** mascaram **falhas**.

Com a geração automática é possível gerar vários conjuntos de dados de teste de acordo com um critério sem ocasionar exaustão.

Com a automatização, o teste é executado igualmente todas as vezes garantindo maior confiabilidade.

Geração de dados de teste

- Problema resolvido??



Geração de dados de teste - Limitações

- Problema resolvido??

NÃO!!!

Problema indecidível



Geração de dados de teste - Limitações

- **Problema indecidível**

- Embora a automatização da tarefa de geração de dados de teste seja desejável, não existe um algoritmo de propósito geral para determinar um conjunto de dados de teste que satisfaça um dado critério.
- Em geral, não é possível nem mesmo determinar automaticamente se esse conjunto existe (Delamaro et al., 2016).

Geração de dados de teste - Limitações

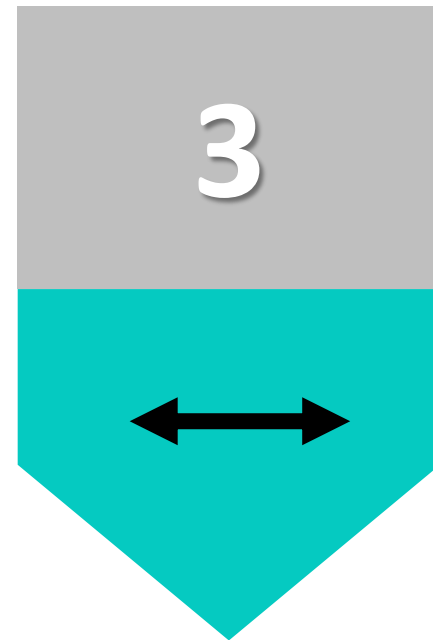
- Limitações



**Correção
coincidente**



**Caminhos não
executáveis**



**Equivalência de
programas**

Geração de dados de teste - Limitações



**Correção
coincidente**

- Uma parte incorreta é executada pelo dado de teste, um estado intermediário incorreto é produzido, mas o programa coincidentemente produz uma saída correta.

Geração de dados de teste - Limitações



Correção
coincidente

```
1 void calculo (int x, int y){  
2     if (y < 0) // (y<=0)  
3         printf ("%d",x+x+6-y*y);  
4     else  
5         printf ("%d",x*x+3-y*y);  
6 }
```

Caso de teste (X,Y)	Caminho esperado	Caminho percorrido	Saída esperada	Saída obtida

Geração de dados de teste - Limitações

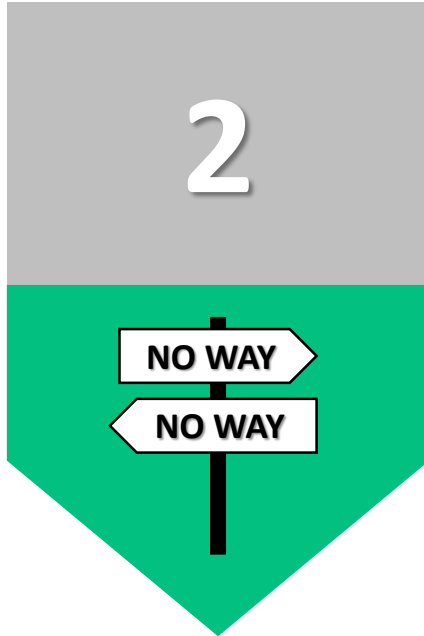


**Correção
coincidente**

```
1 void calculo (int x, int y){  
2     if (y < 0) // (y<=0)  
3         printf ("%d",x+x+6-y*y);  
4     else  
5         printf ("%d",x*x+3-y*y);  
6 }
```

Caso de teste (X,Y)	Caminho esperado	Caminho percorrido	Saída esperada	Saída obtida
(-1,0)	1,2,3	1,4,5	4	4
(2,0)	1,2,3	1,4,5	10	7

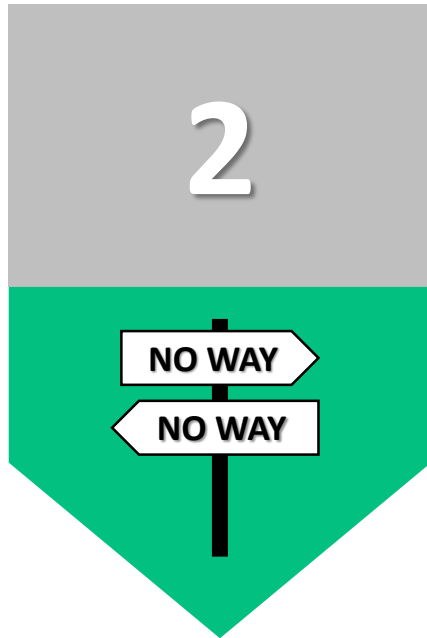
Geração de dados de teste - Limitações



**Caminhos não
executáveis**

- Não existe dado de entrada capaz de executá-lo.

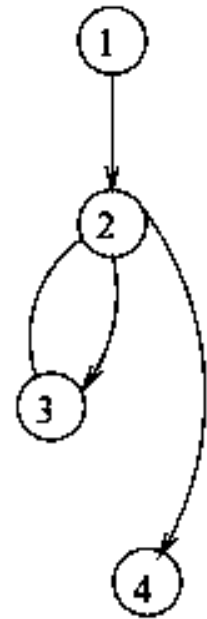
Geração de dados de teste - Limitações



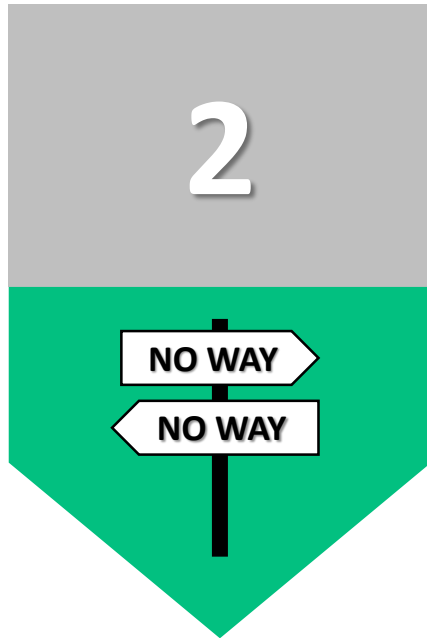
Caminhos não executáveis

- Descubra a associação de $\langle i \rangle$ não executável

```
1 void soma()  
2 {  
3   int i, s=0;  
4   for (i = 0; i < 10; i++)  
5     s = s + pow(i, 2);  
6   printf("%d", s);  
7 }
```



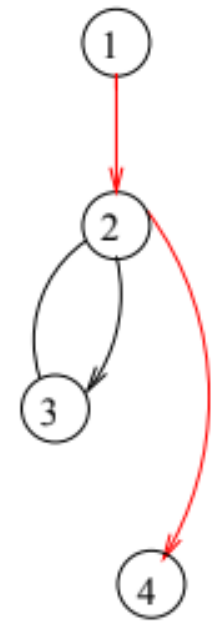
Geração de dados de teste - Limitações



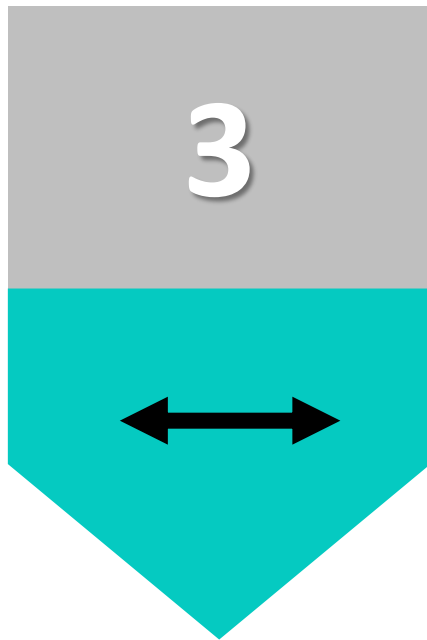
Caminhos não executáveis

- Associação $\langle i, 1, (2, 4) \rangle$ é não executável

```
1 void soma()  
2 {  
3   int i, s=0;  
4   for (i = 0; i < 10; i++)  
5     s = s + pow(i, 2);  
6   printf("%d", s);  
7 }
```



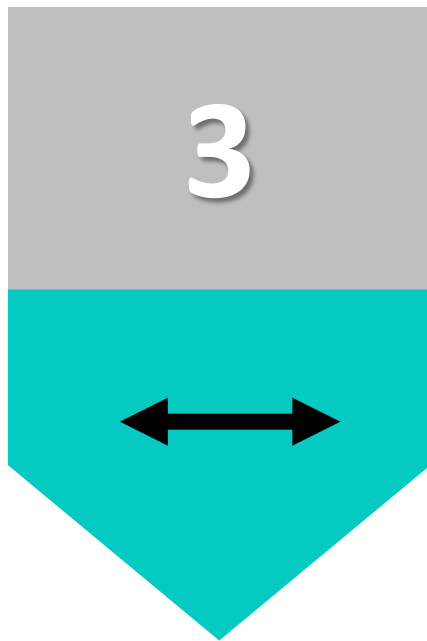
Geração de dados de teste - Limitações



**Equivalência de
programas**

- Mutantes que computam a mesma função que o programa original. Não existe dado de teste para diferenciar seu comportamento.

Geração de dados de teste - Limitações



Equivalência de programas

- Mutantes que computam a mesma função que o programa original. Não existe dado de teste para diferenciar seu comportamento.

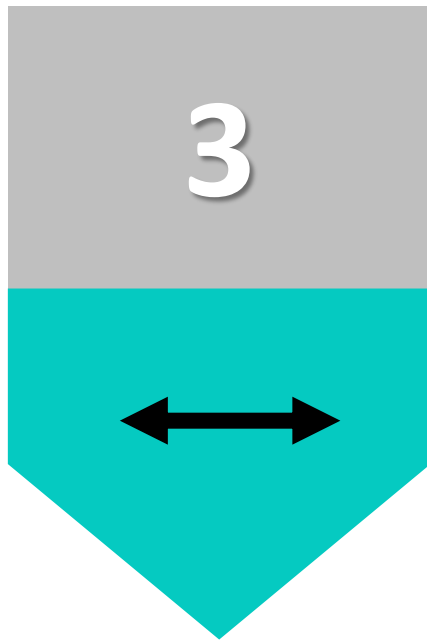
```
1 void soma()  
2 {  
3     int i, s=0;  
4     for (i = 0; i < 10; i++)  
5         s = s + pow (i,2);  
6     printf ("%d",s);  
7 }
```

Programa original

```
1 void soma()  
2 {  
3     int i, s*=0;  
4     for (i = 0; i < 10; i++)  
5         s = s + pow (i,2);  
6     printf ("%d",s);  
7 }
```

Programa mutante (equivalente)

Geração de dados de teste - Limitações



Equivalência de programas

- Mutantes que computam a mesma função que o programa original. Não existe dado de teste para diferenciar seu comportamento.

```
1 void soma()  
2 {  
3   int i, s=0;  
4   for (i = 0; i < 10; i++)  
5     s = s + pow (i,2);  
6   printf ("%d",s);  
7 }
```

Programa original

```
1 void soma()  
2 {  
3   int i, s*=0;  
4   for (i = 0; i < 10; i++)  
5     s = s + pow (i,2);  
6   printf ("%d",s);  
7 }
```

Programa mutante (equivalente)

Geração de dados de teste - Limitações

- Existe solução?



Geração de dados de teste - Limitações

- Existe solução?
- Qual?
- Como você faria?



Geração de dados de teste - Técnicas

- Apesar das limitações.....

Na literatura encontram-se trabalhos que permitem a geração automática de dados de teste.

- Seleção ao acaso (aleatória) ou direcionada;
- Necessidade de executar ou não o programa
 - Dinâmica
 - Estática
 - Híbrida

Geração de dados de teste - Técnicas

- Seleção ao acaso (aleatória)
 - Pontos do domínio de entrada são selecionados aleatoriamente;
 - Fácil automação;
 - Simples;
 - Baixo custo de “geração”;
 - Dificuldade para alcançar cobertura desejada.



Randoop

Automatic unit test generation for Java

<https://randoop.github.io/randoop/>

Geração de dados de teste - Técnicas

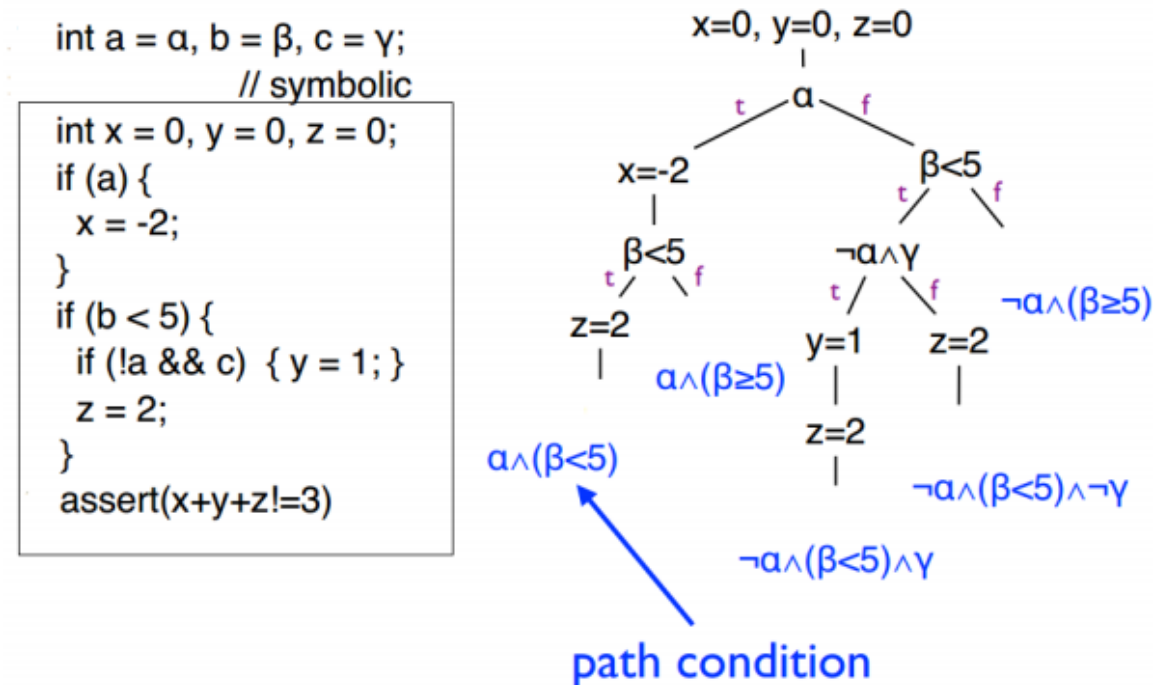
- Geração estática
 - Não requer a execução do programa
 - Geralmente utiliza execução simbólica

Execução Simbólica

- O programa é executado com valores simbólicos para suas variáveis de entrada; os comandos são avaliados simbolicamente.
- Geralmente é utilizada para:
 - Executar caminhos requeridos no teste estrutural
 - Matar mutantes no teste baseado em defeitos

Geração de dados de teste - Técnicas

- Execução Simbólica
 - **Computação do caminho:** expressão algébrica que representa um dado caminho, na qual as variáveis de saída são expressas em termos das variáveis de entrada
 - **Condição do caminho:** condição para que um caminho seja executado (a ser resolvida para gerar os dados)



Geração de dados de teste - Técnicas

- Execução Simbólica
 - Interpretador simbólico
 - Simplificador de Expressões
 - Solucionador de sistemas de equações
 - Provador de Teoremas
 - se $x < 0$ & $x > 0$ então “caminho impraticável”



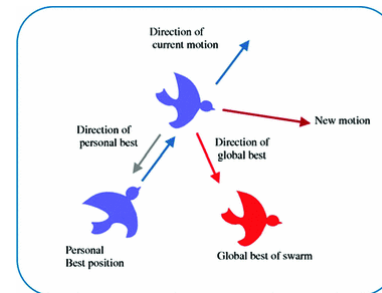
Geração de dados de teste - Técnicas

- Geração de dados baseada em busca
 - Algoritmos de Busca (Search-based software testing) SBST
 - Aplicação de técnicas de busca meta-heurísticas para geração de dados de teste

- Hill Climb



- Particule Swarming Optimization

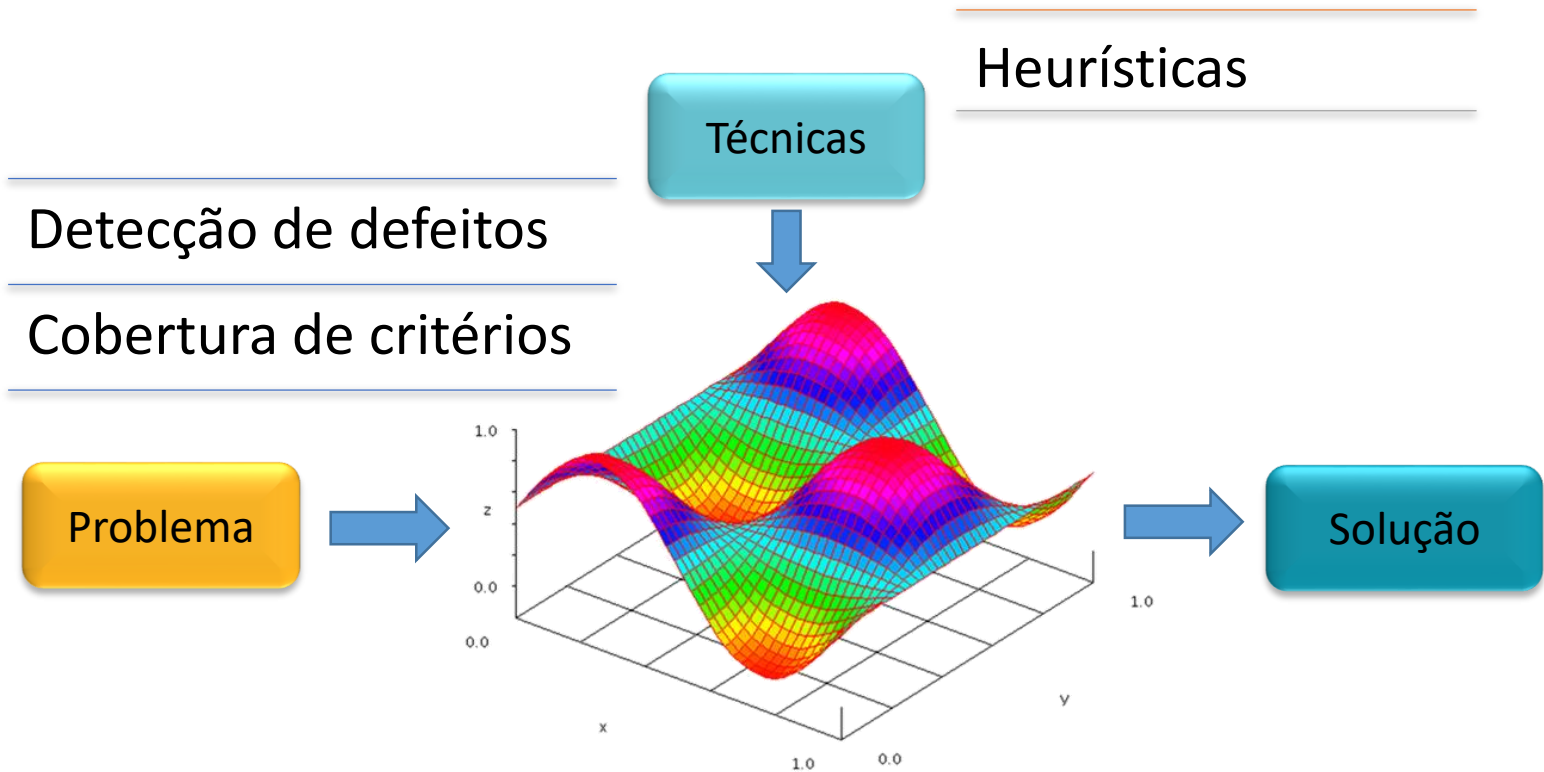


- Genetic Algorithm



Geração de dados de teste – Algoritmo Genético

- Geração de dados baseada em busca



Geração de dados de teste – Algoritmo Genético

- Geração de dados de teste por meio de Algoritmos Genéticos
 - **Computação Evolutiva:** ramo da computação que se baseia em mecanismos evolutivos encontrados na natureza.
 - Existe uma solução (espécie) diferente para cada problema (ambiente) específico.
 - As soluções (espécies) inclusive se adaptam a ambientes que mudam com o tempo.
 - **Conclusão:** Teoria da Evolução das Espécies.



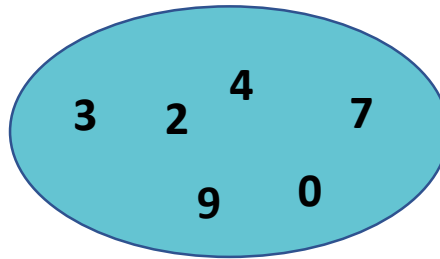
Geração de dados de teste – Algoritmo Genético

- Conceitos
 - Indivíduos
 - População
 - Função de Fitness
- Operadores
 - Seleção
 - Mutação
 - Crossover

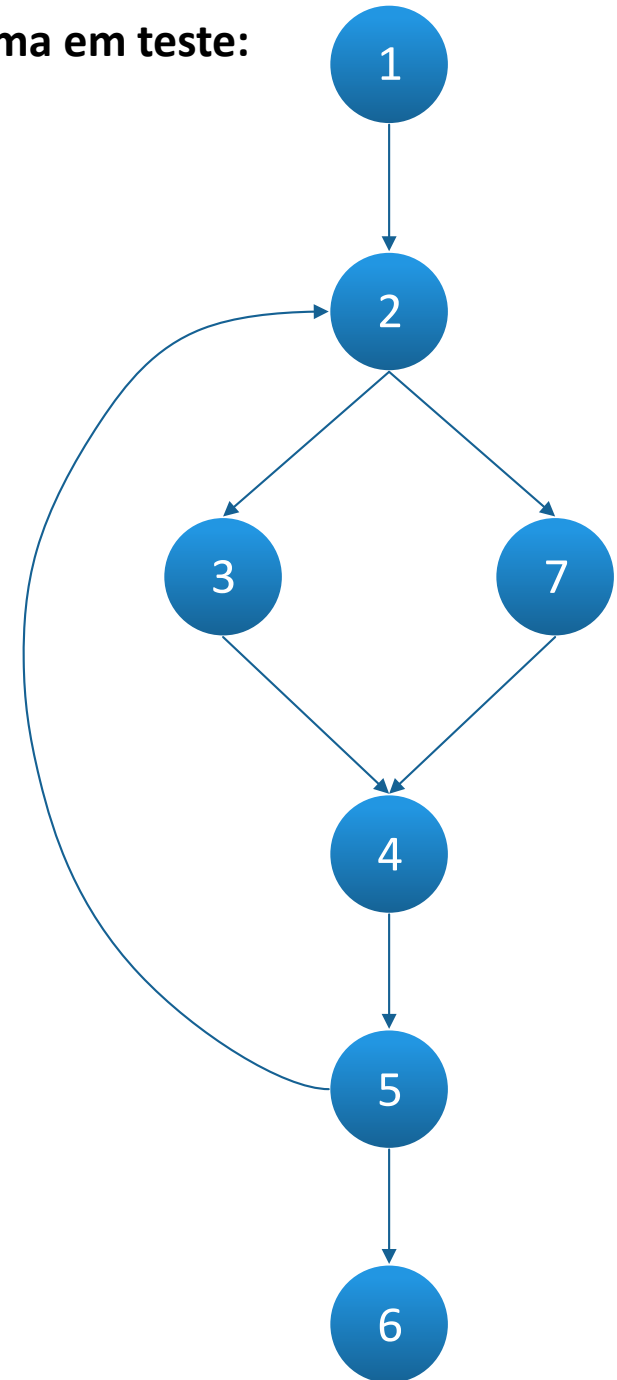


Exemplo

População aleatória de
soluções candidatas



Programa em teste:

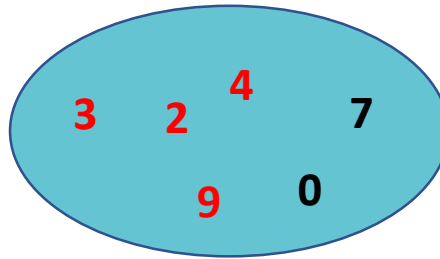


Exemplo

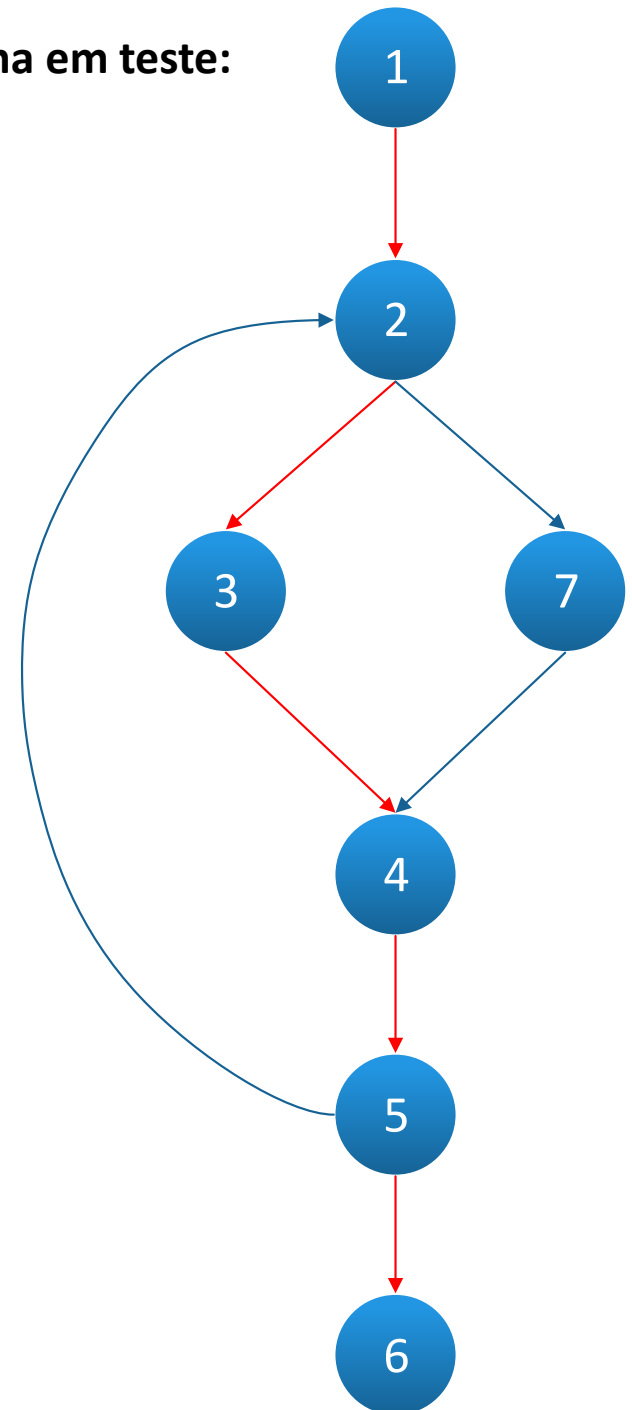
População aleatória de
soluções candidatas



Avaliação de *Fitness* 62,5%

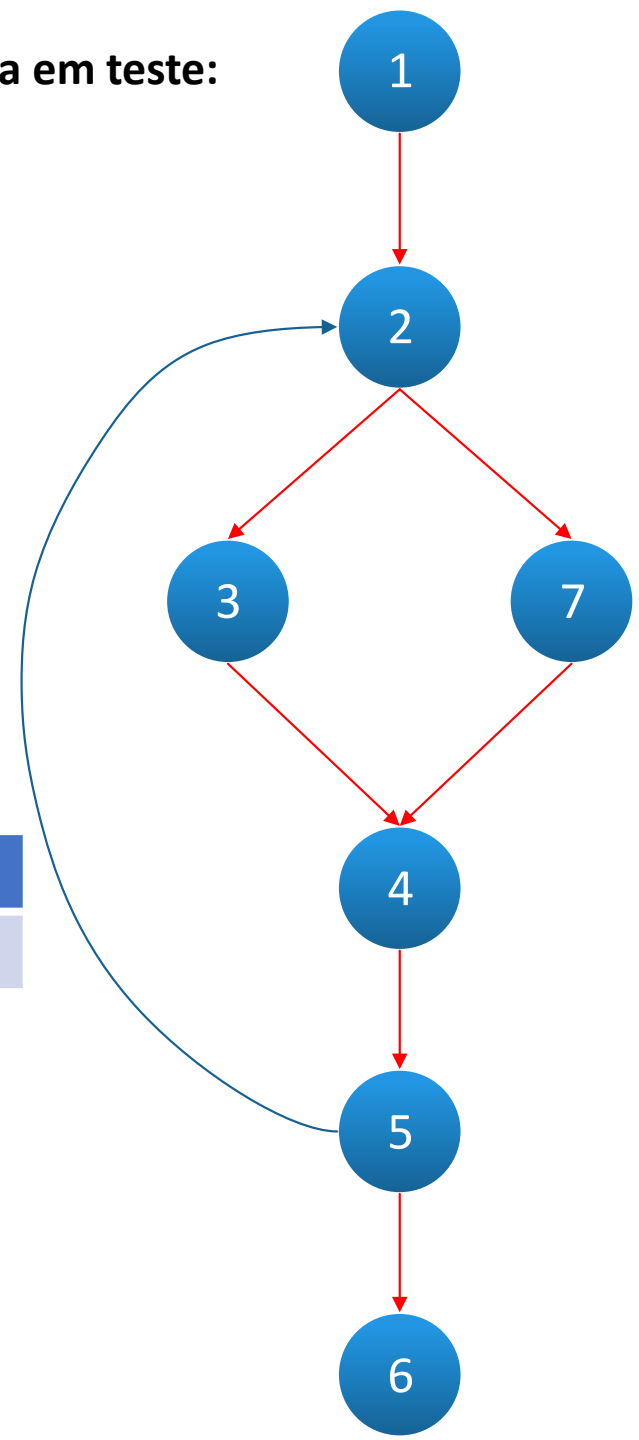


Programa em teste:

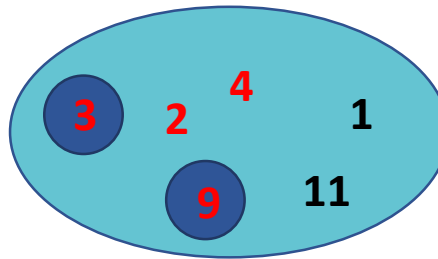


Exemplo

Programa em teste:

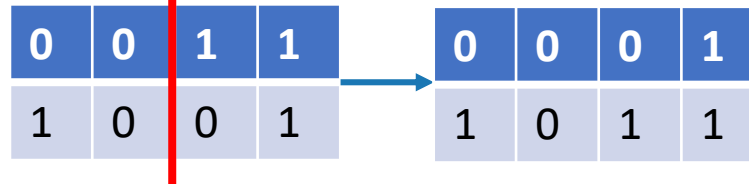


População aleatória de
soluções candidatas



Avaliação de *Fitness* 62,5%

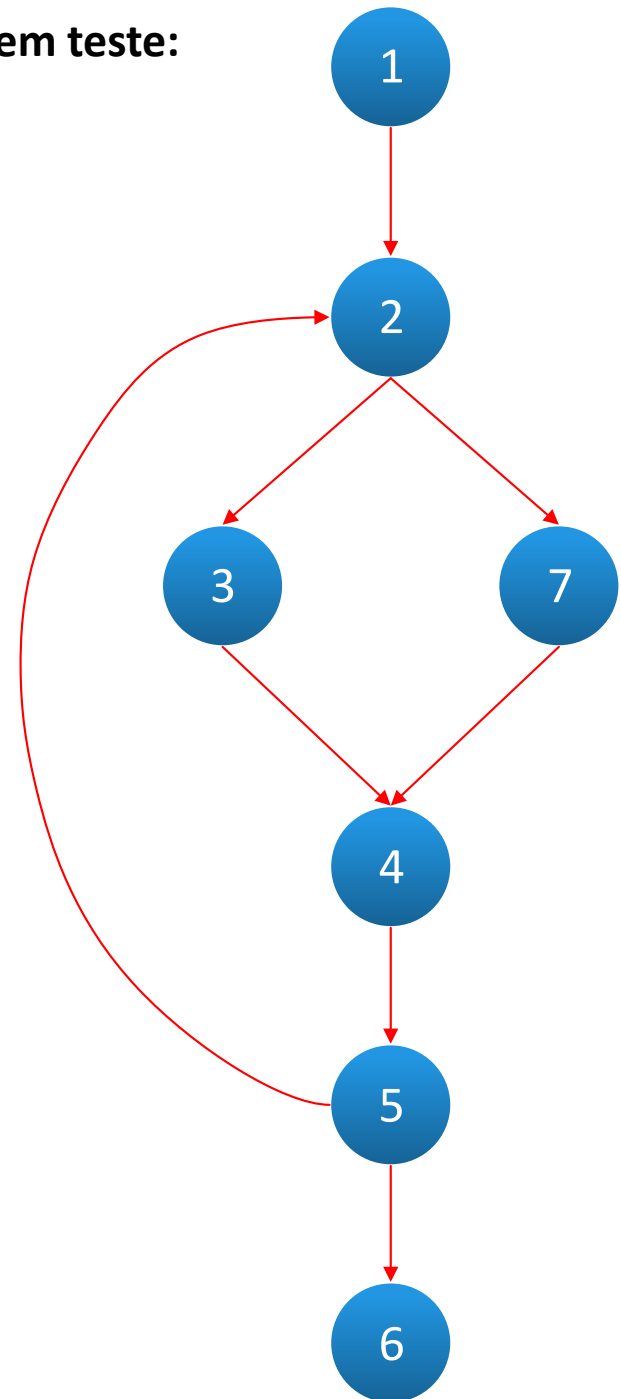
Crossover



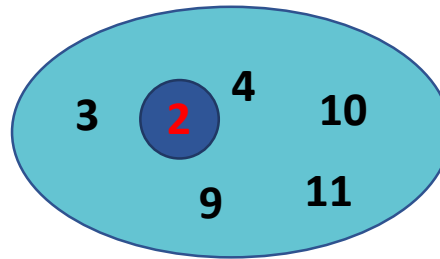
Avaliação de *Fitness* 87,5%

Exemplo

Programa em teste:

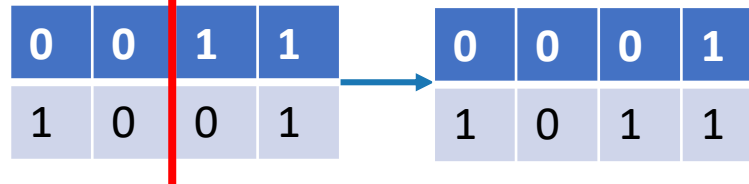


População aleatória de soluções candidatas



Avaliação de *Fitness* 62,5%

Crossover



Avaliação de *Fitness* 87,5%

Mutação

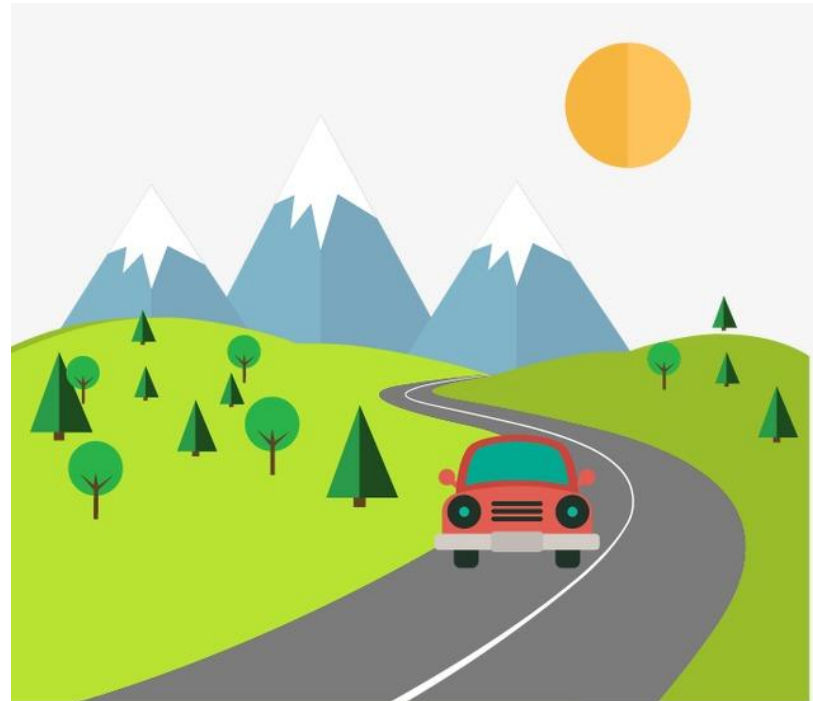


Avaliação de *Fitness* 100%

Geração de dados de teste – Algoritmo Genético

- Exemplo
 - Como definir o melhor carro para testar uma rodovia sinuosa?

https://rednuht.org/genetic_cars_2/



Geração de dados de teste – Algoritmo Genético

- Desafio
 - Definir uma configuração no Algoritmo Genético para gerar o carro capaz de percorrer a maior distância.

Save Population	Restore Saved Population
Surprise!	New Population
Fast Forward	
Create new world with seed	
Enter any string	Go!
Generation	0
Cars alive	16
Distance	17.58 meters
Height	0.42 meters
Mutation rate:	5% v
Mutation size:	100% v
Floor:	fixed v
Gravity:	Earth (9.81) v
Elite clones:	1 v

Geração de dados de teste – Algoritmo Genético

- Ferramenta de Geração
 - EvoSuite (Programas sequenciais)
 - <http://www.evosuite.org/>
 - BioConcST
 - Bio-inspired optimization in Concurrent Software Testing
- Competição (TOOLympics)



TOOLympics 2019

Competition on Software Testing (Test-Comp)

Dúvidas???

Ricardo F. Vilela
ricardovilela@usp.br

Simone R. S. Souza
srocio@icmc.usp.br