

Critérios de Teste Caixa-Branca

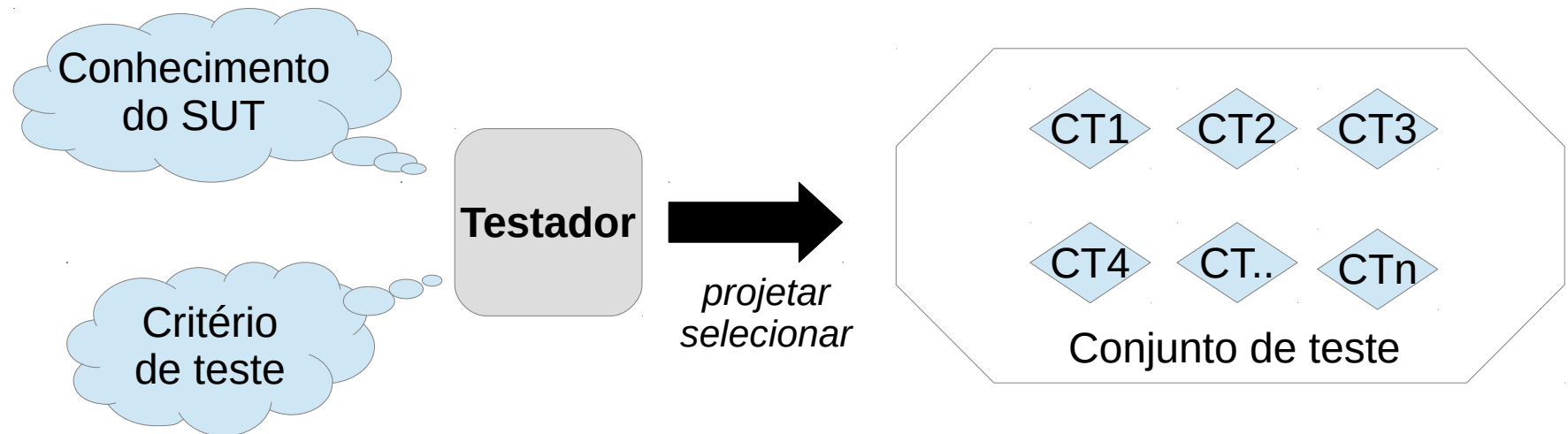
Prof. André Takeshi Endo

Técnicas de Teste

- **CrITÉrios de Teste**
- Define uma ***maneira sistemática*** e planejada para conduzir testes
- Quais casos de teste com **maior chance** de revelar defeitos

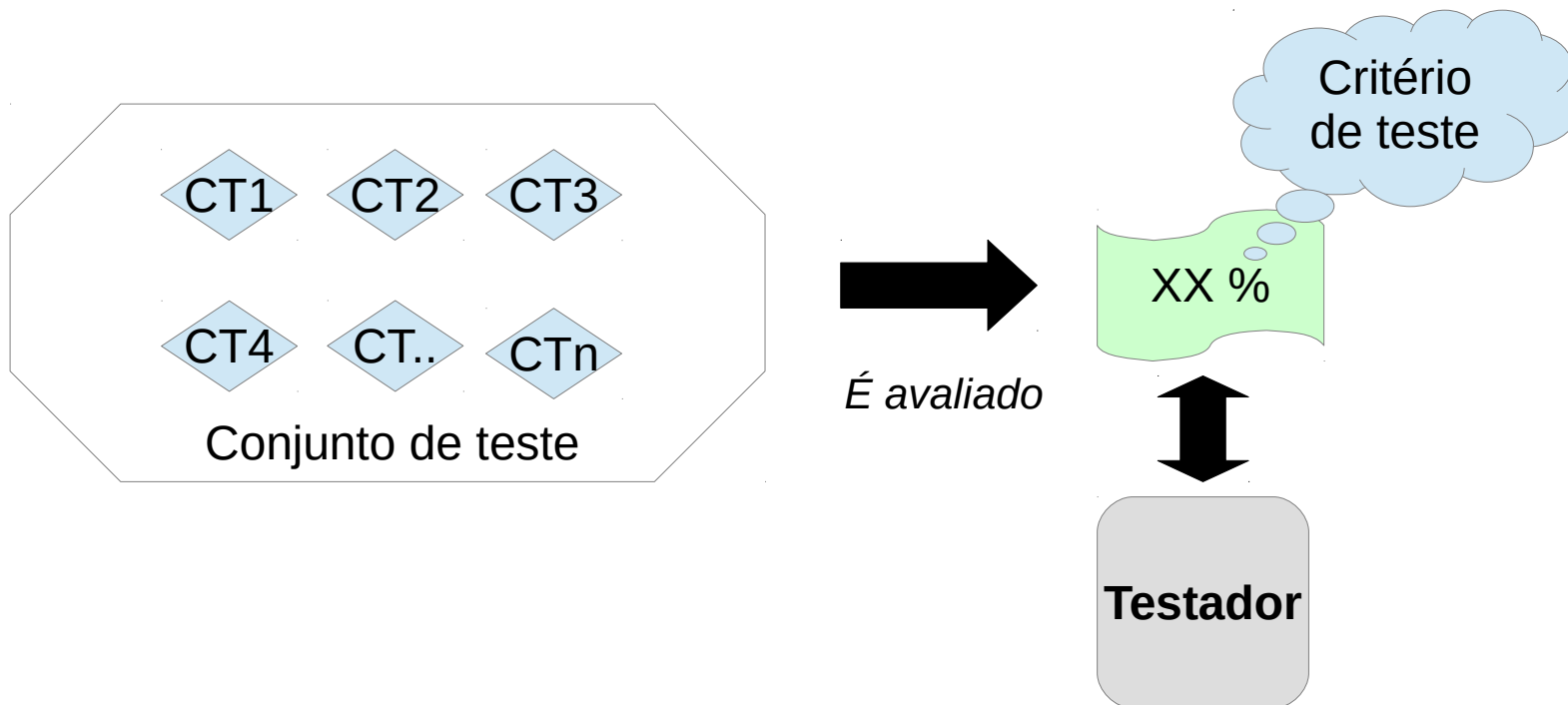
Critérios de Teste

- Um critério pode ser usado para selecionar/projetar os casos de teste
 - ***Critério de seleção de teste***



Critérios de Teste

- Um critério pode ser usado para avaliar a qualidade dos casos de teste
 - ***Critério de adequação de teste***



Técnicas de Teste

- **Teste Caixa Preta**

- Testes baseados na especificação (de requisitos)
- A funcionalidade testada é considerada uma caixa preta

- **Teste Caixa Branca**

- Testes baseados na estrutura interna do programa
- Analisar o código fonte (caixa branca)

Teste Caixa Branca

- Vários critérios de teste:
- Critérios baseado em fluxo de controle
- Teste de mutação

Grafo de Fluxo de Controle

- Grafo direcionado $G = (V, E, s)$
- Construído com base no fluxo de controle do programa

Grafo de Fluxo de Controle

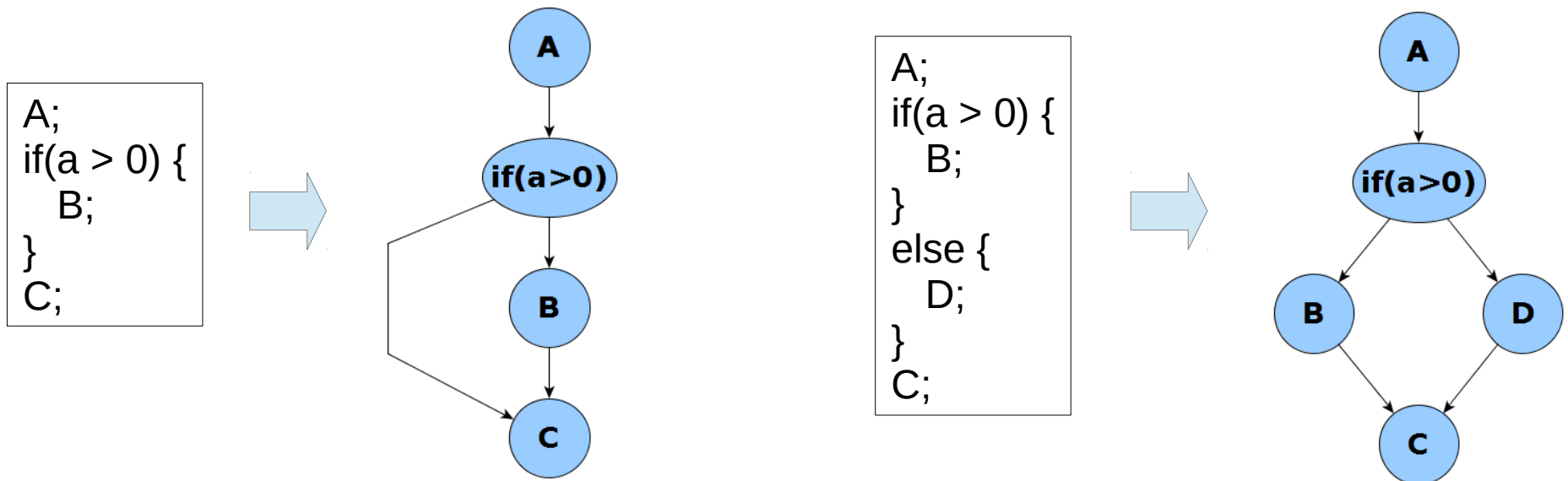
- $V \rightarrow$ Nós (Vértices)
 - Blocos de instrução que não possuem desvios de execução
- $E \rightarrow$ Arcos (Arestas)
 - Representa mudanças no fluxo de execução
- $s \in V$ é o nó de entrada

Grafo de Fluxo de Controle

- Mapeando o programa para o GFC
 - If
 - If-else
 - While
 - For
 - Do-While
 - Switch
 - Condições compostas
 - And (&&) e Or (||)

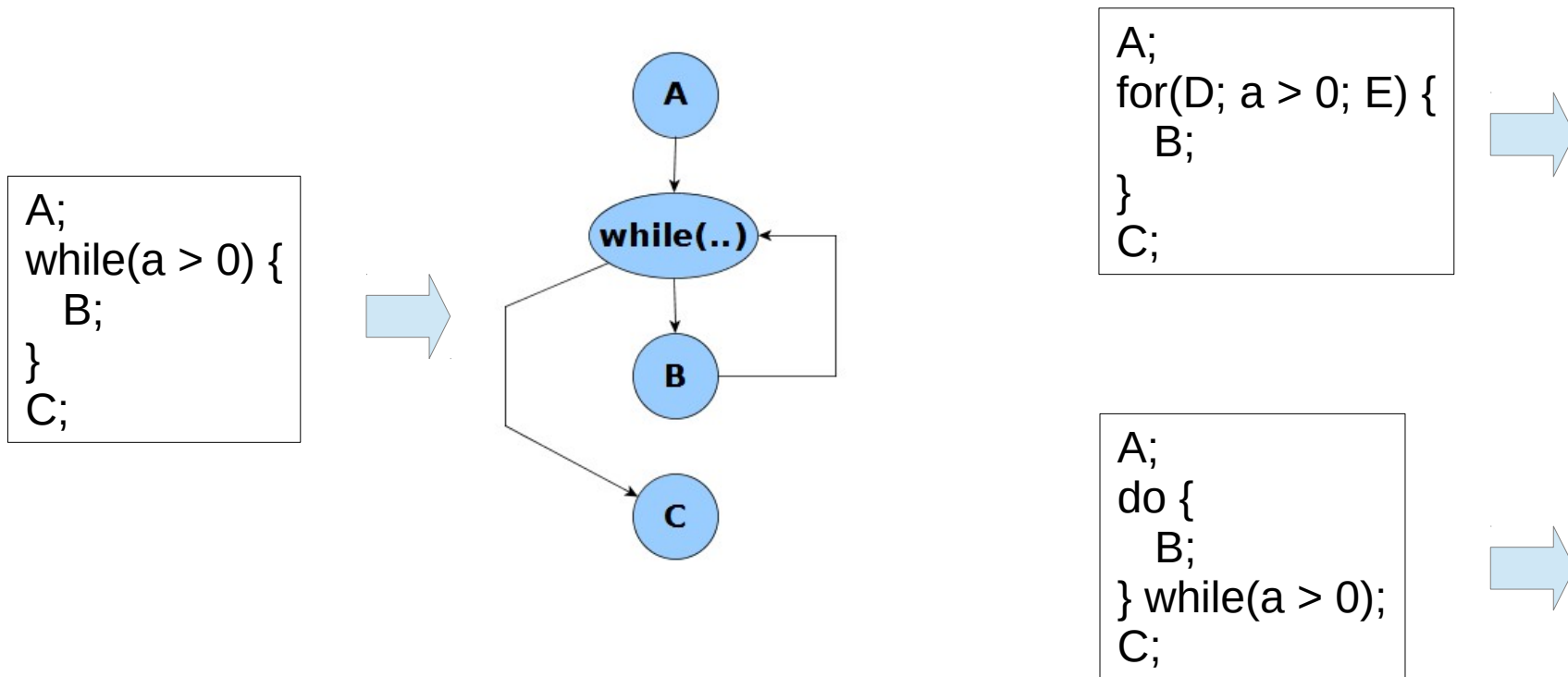
Grafo de Fluxo de Controle

- Mapeando o programa para o GFC
 - If e If-else



Grafo de Fluxo de Controle

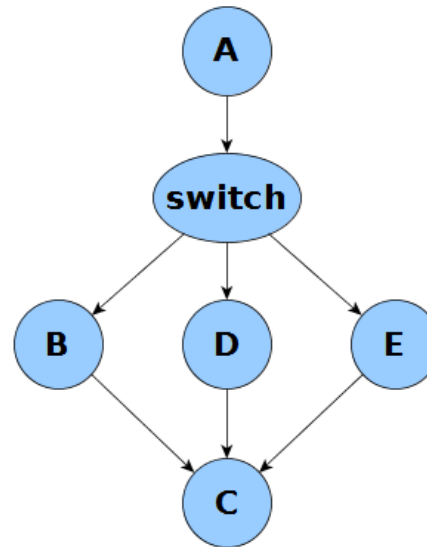
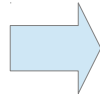
- Mapeando o programa para o GFC
 - While, For, Do-While



Grafo de Fluxo de Controle

- Mapeando o programa para o GFC
 - Switch

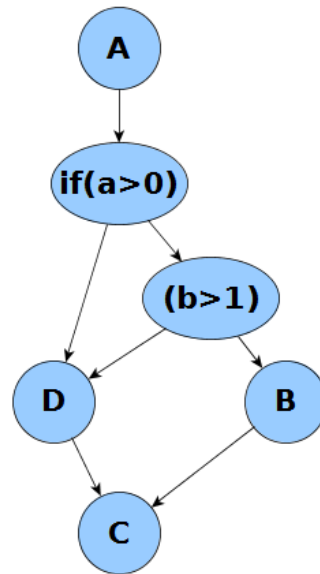
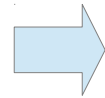
```
A;  
switch(a) {  
  case 1: B; break;  
  
  case 2: D; break;  
  
  default: E;  
}  
C;
```



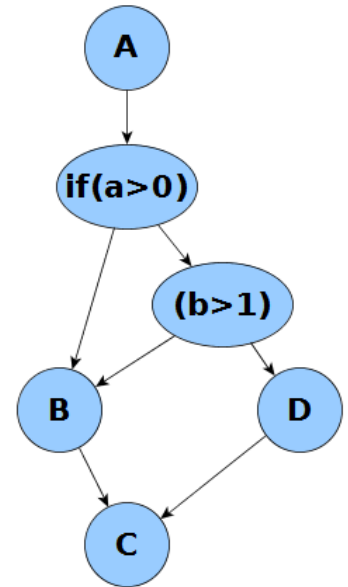
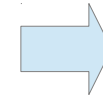
Grafo de Fluxo de Controle

- Mapeando o programa para o GFC
 - Condições compostas
 - And (&&) e Or (||)

```
A;  
if(a > 0 && b > 1) {  
  B;  
}  
else {  
  D;  
}  
C;
```



```
A;  
if(a > 0 || b > 1) {  
  B;  
}  
else {  
  D;  
}  
C;
```



Grafo de Fluxo de Controle

- Mapeando o programa para o GFC
 - Return?
 - Try-catch?
 - Operador ternário?
 - Laço com condição composta?

Exemplo

- Aglutinar caracteres de '\n'
 - Adaptado do Apache Velocity [Pezze]

```
public class NovaLinha {  
  
    /**  
     * @param argStr string da qual os caracteres de new line serao aglutinados  
     * @return String  
     */  
    public String collapseNewlines(String argStr) {  
        char last = argStr.charAt(0);  
        StringBuffer sBuf = new StringBuffer();  
  
        for (int i = 0; i < argStr.length(); i++) {  
            char ch = argStr.charAt(i);  
            if(ch != '\n' || last != '\n') {  
                sBuf.append(ch);  
                last = ch;  
            }  
        }  
  
        return sBuf.toString();  
    }  
}
```

Exemplo

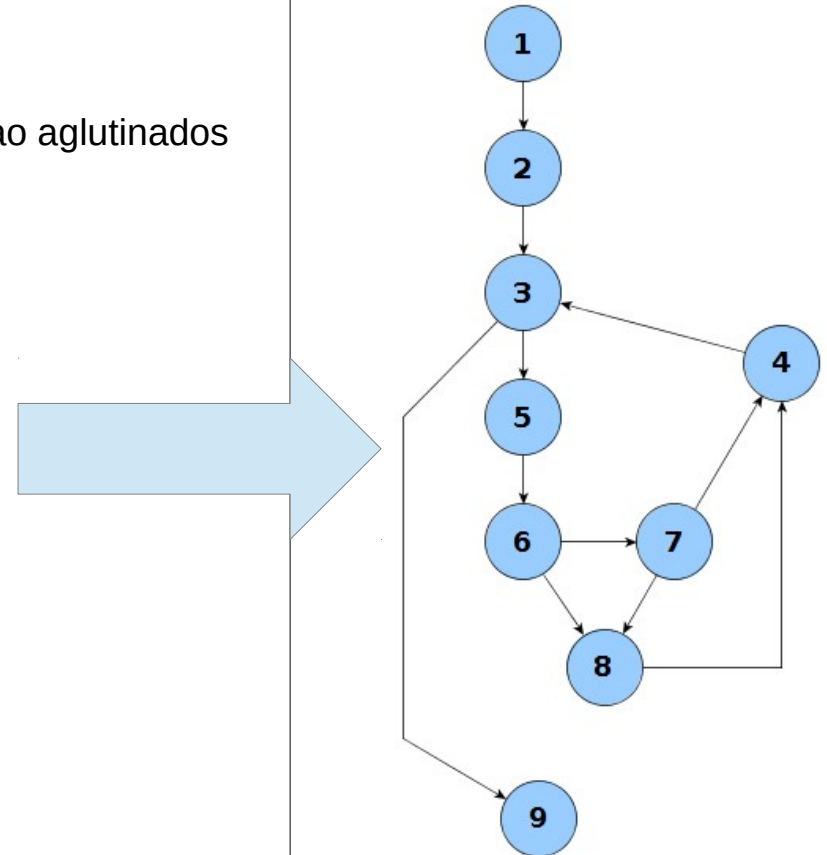
- Aglutinar caracteres de '\n'
 - Adaptado do Apache Velocity [Pezze]

```
public class NovaLinha {  
  
    /**  
     * @param argStr string da qual os caracteres de new line serao aglutinados  
     * @return String  
     */  
    public String collapseNewlines(String argStr) {  
        1 char last = argStr.charAt(0);  
        StringBuffer sBuf = new StringBuffer();  
        2      3      4  
        for (int i = 0; i < argStr.length(); i++) {  
            5 char ch = argStr.charAt(i);  
            6 if(ch != '\n' || last != '\n') { 7  
                sBuf.append(ch);  
            8 last = ch;  
            }  
        }  
        9 return sBuf.toString();  
    }  
}
```


Exemplo

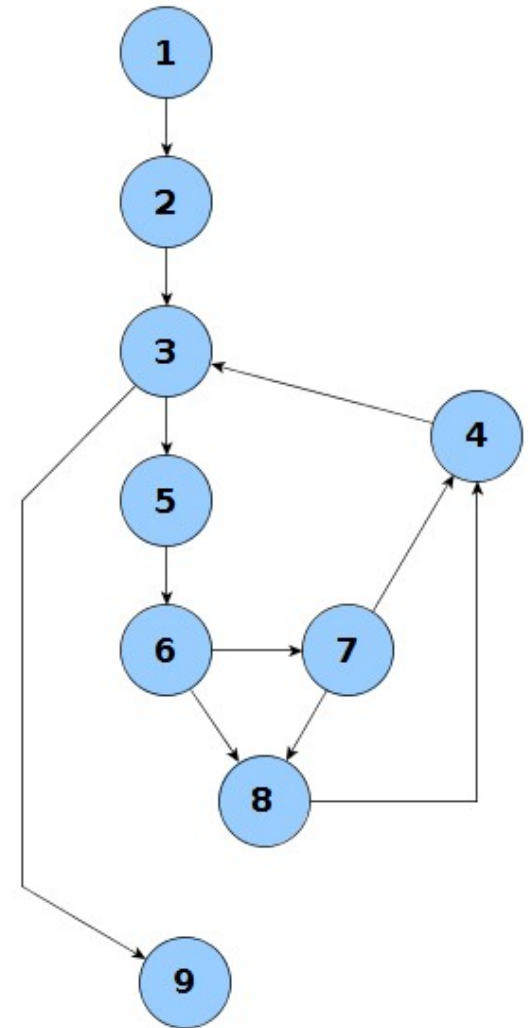
- Aglutinar caracteres de '\n'
 - Adaptado do Apache Velocity [Pezze]

```
public class NovaLinha {  
  
    /**  
     * @param argStr string da qual os caracteres de new line serao aglutinados  
     * @return String  
     */  
    public String collapseNewlines(String argStr) {  
1      char last = argStr.charAt(0);  
        StringBuffer sBuf = new StringBuffer();  
        for (int i = 0; i < argStr.length(); i++) {  
2          char ch = argStr.charAt(i);  
3          if (ch != '\n' || last != '\n') {  
4              sBuf.append(ch);  
5              last = ch;  
6          }  
7      }  
9      return sBuf.toString();  
    }  
}
```



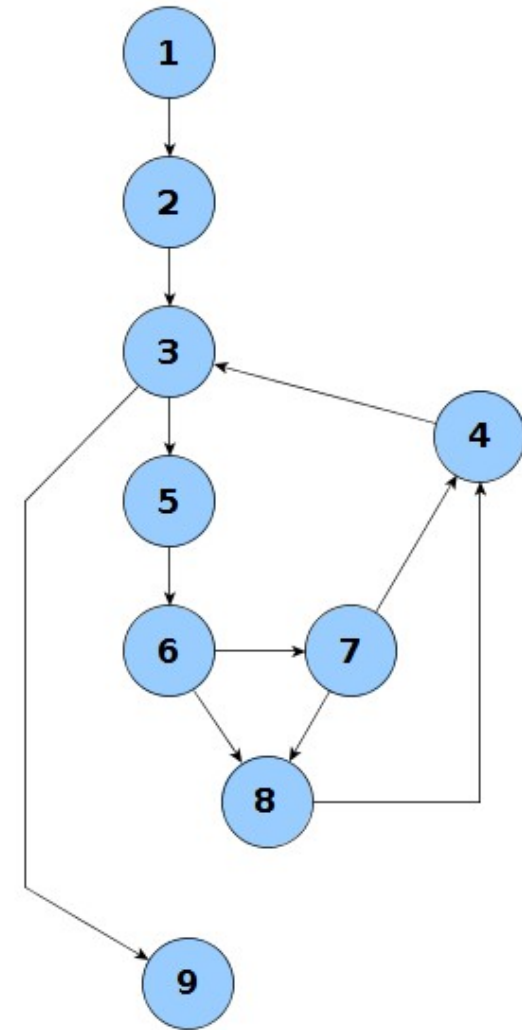
Critérios de Teste

- **Todos-Nós** → CTs que executem cada nó ao menos uma vez (*line coverage*)
- **Todos-Arcos** → CTs que executem cada arco ao menos uma vez (*branch coverage*)



Exercício

- Elabore um caso de teste (JUnit) que execute o arco (3,9)
 - Qual o caminho que ele executa?
 - Qual o % de cobertura do todos-nós?
 - Qual o % de cobertura do todos-arcos?



Exercício

- Elabore outros casos de teste (JUnit) para cobrir 100% do critério todos-arcos.
- Para cada caso de teste, deixe na forma de comentário o caminho que ele executou segundo o GFC apresentado.
- Use a ferramenta JaCoCo para confirmar o 100% de cobertura do todos-arcos (*branch coverage*)

Bibliografia

- [Pfleeger07] S. L. Pfleeger, “Engenharia de Software: Teoria e Prática”, 2007.
- [Pressman11] R. S. Pressman, “Engenharia de Software: uma abordagem profissional”, 2011.
- [Sommerville03] I. Sommerville, “Engenharia de Software”, 2003.
- [Brooks87] “No Silver Bullet: Essence and Accidents of Software Engineering”, 1987.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1663532
- [IEEE90] “IEEE Standard Glossary of Software Engineering Terminology”, 1990.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=159342

Bibliografia

- [Myers] G. J. Myers, T. Badgett, C. Sandler, “The art of software testing”, 2012.
- [Pezze] M. Pezze, M. Young, “Teste e análise de software: Processos, princípios e técnicas”, 2008.
- [DMJ07] DELAMARO, Márcio Eduardo; MALDONADO, José Carlos; JINO, Mario. Introdução ao teste de software. Rio de Janeiro, RJ: Elsevier, 2007. 394 p. ISBN 9788535226348.
- [UUU] Materiais didáticos elaborados pelos grupos de engenharia de software do ICMC-USP, DC-UFSCAR e UTFPR-CP.