

# Verificação, Validação & Teste de Software

Jean Phelipe de Oliveira Lima

[jpdol.eng16@uea.edu.br](mailto:jpdol.eng16@uea.edu.br)

/jpdol

# Teste Funcional

- Técnica utilizada para se projetarem casos de teste na qual o programa ou sistema é considerado uma **caixa-preta**. Nessa técnica os detalhes de implementação não são considerados e o software é avaliado segundo o ponto de vista do usuário.

# Cr terios

- Particionamento de Equival ncia
  - An lise do Valor Limite
  - Grafo Causa-Efeito
  - Error Guessing
- 
- OBS: Como todos os cr terios da t cnica funcional baseiam-se apenas na especifica  o do produto testado, a qualidade de tais cr terios depende fortemente da exist ncia de uma boa especifica  o de requisitos.

# Critérios

- Os critérios funcionais podem ser aplicados em todas as fases de testes e em produtos desenvolvidos com qualquer paradigma de programação, pois não levam em consideração detalhes de implementação.

# Grafo Causa-Efeito

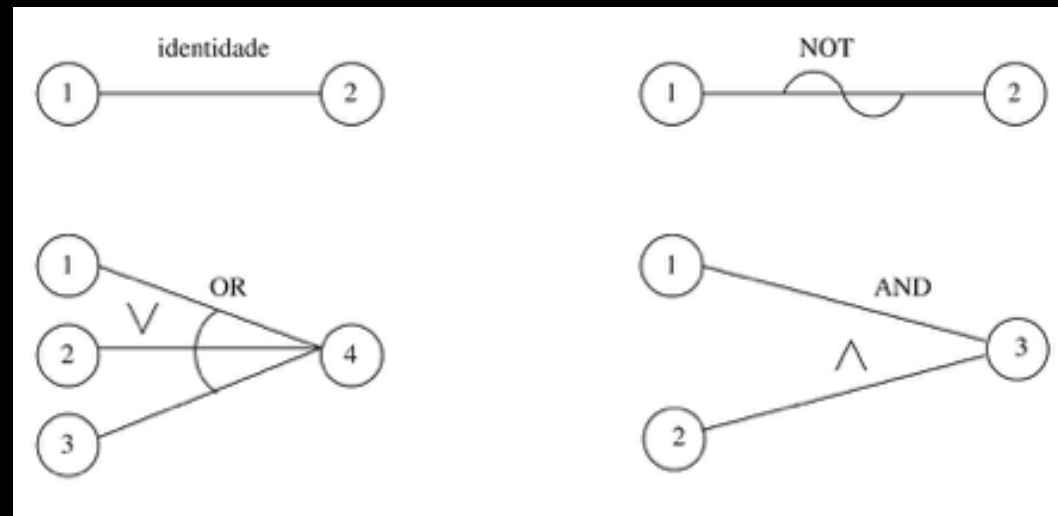
Uma das limitações dos critérios anteriores é que eles não exploram combinações dos dados de entrada. Já o critério Grafo Causa-Efeito ajuda na definição de um conjunto de casos de teste que explora ambiguidades e incompletude nas especificações.

# Grafo Causa-Efeito

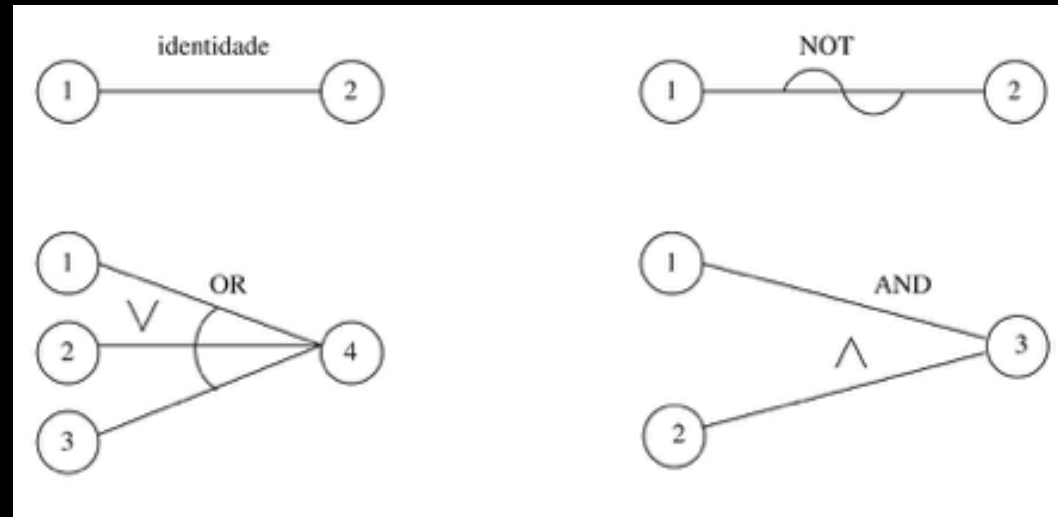
- O grafo é uma linguagem formal na qual a especificação é traduzida;
- Processo para derivar casos de teste a partir desse critério pode ser resumido nos seguintes passos:
  1. Dividir a especificação do software em partes, pois a construção do grafo para grandes especificações torna-se bastante complexa;
  2. Identificar as causas e efeitos na especificação. As causas correspondem a entradas, estímulos ou qualquer coisa que provoque uma resposta do sistema em teste, e os efeitos correspondem a saídas, mudanças no estado do sistema ou qualquer resposta observável. Uma vez identificados, a cada um deve ser atribuído um único número;
  3. Analisar a semântica da especificação e transformar em um grafo booleano – o Grafo Causa-Efeito – que liga as causas e os efeitos;
  4. Adicionar anotações ao grafo, as quais descrevem combinações das causas e efeitos devido a restrições sintáticas ou do ambiente;
  5. Converter o grafo em uma tabela de decisão, na qual cada coluna representa um caso de teste;
  6. Converter as colunas da tabela de decisão em casos de teste.

# Sintaxe

- Cada nó no grafo pode assumir os valores 0 ou 1, que representam ausência ou presença no estado.
- A notação utilizada no Grafo Causa-Efeito é composta dos operadores apresentados a seguir:



# Sintaxe



- Função identidade: se nó “1” é 1, então nó “2” é 1; senão nó “2” é 0;
- Função not: se nó “1” é 1, então nó “2” é 0; senão nó “2” é 1;
- Função or: se nó “1” ou “2” ou “3” é 1, então nó “4” é 1; senão nó “4” é 0;
- Função and: se ambos nós “1” e “2” são 1, então nó “3” é 1; senão nó “3” é 0.



# Exemplo – Imprime Mensagem

- O programa lê dois caracteres e, de acordo com eles, mensagens serão impressas da seguinte forma:
  - O primeiro caractere deve ser um “A” ou um “B”;
  - O segundo caractere deve ser um dígito. Nessa situação, o arquivo deve ser atualizado;
  - Se o primeiro caractere é incorreto, enviar a mensagem X;
  - Se o segundo caractere é incorreto, enviar a mensagem Y.

# Exemplo – Imprime Mensagem

- Causas:
  1. Caractere na coluna 1 é "A";
  2. Caractere na coluna 1 é "B";
  3. Caractere na coluna 2 é um dígito.

# Exemplo – Imprime Mensagem

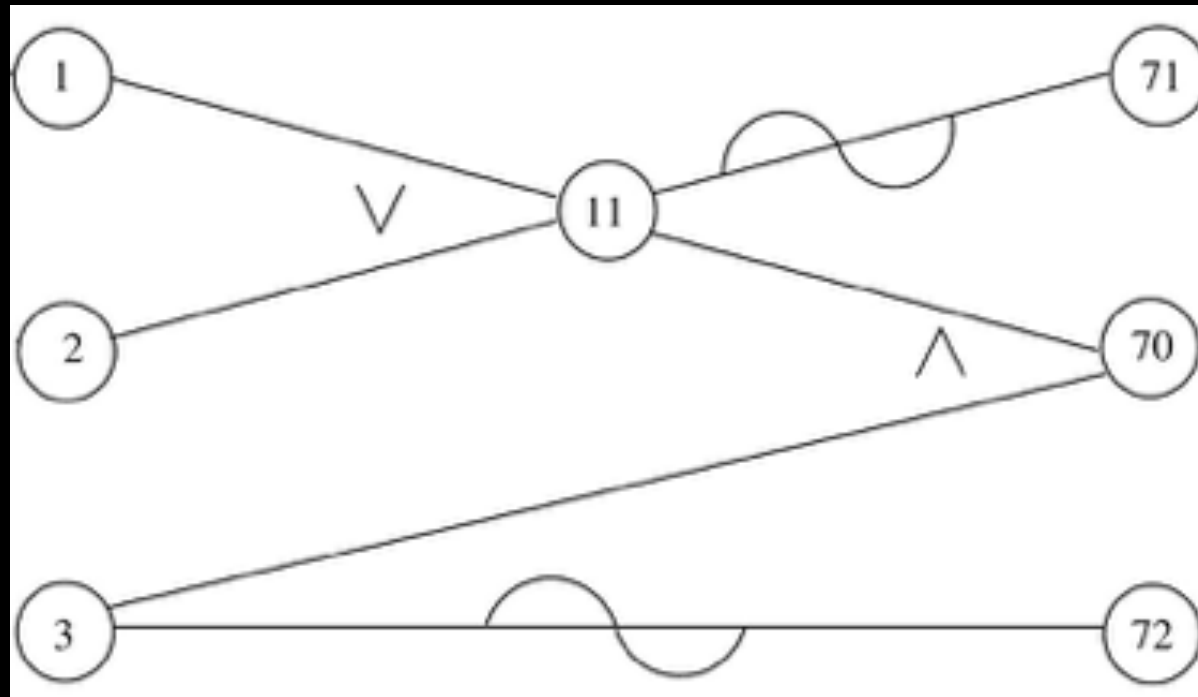
- Causas:

1. Caractere na coluna 1 é "A";
2. Caractere na coluna 1 é "B";
3. Caractere na coluna 2 é um dígito.

- Efeitos:

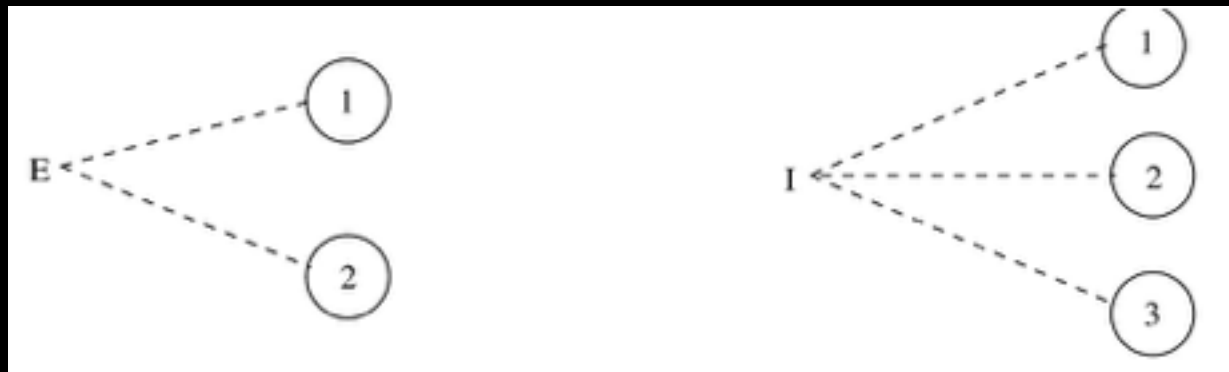
70. A atualização é realizada;
71. A mensagem X é enviada;
72. A mensagem Y é enviada.

# Exemplo – Imprime Mensagem



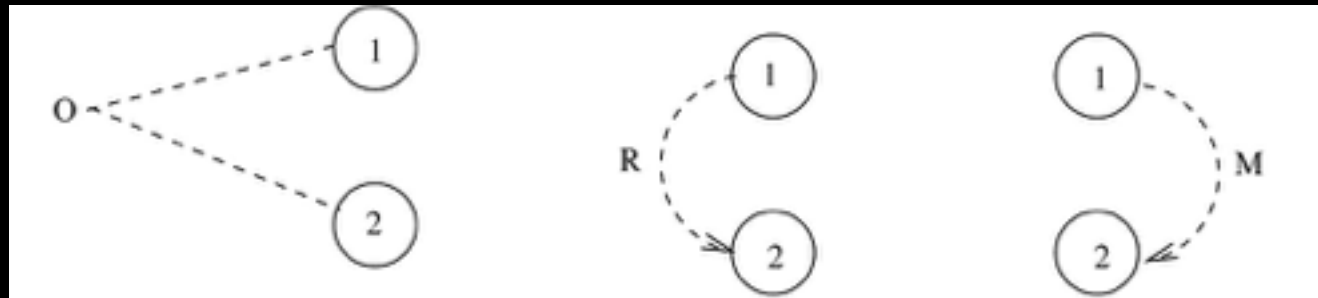
# Exemplo – Imprime Mensagem

- Restrições:
  - Restrição E: no máximo, um entre “1” e “2” pode ser igual a 1 (ou seja, “1” e “2” não podem ser 1 simultaneamente);
  - Restrição I: no mínimo, um entre “1”, “2” e “3” deve ser igual a 1 (ou seja, “1”, “2” e “3” não podem ser 0 simultaneamente);



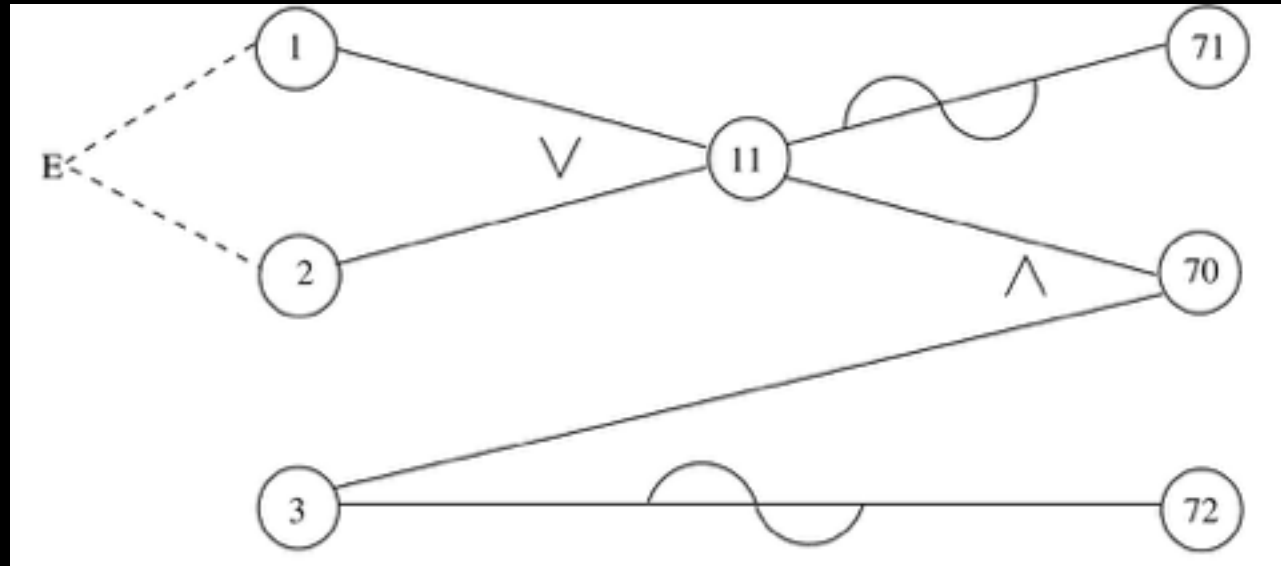
# Exemplo – Imprime Mensagem

- Restrições:
  - Restrição O: um e somente um entre “1” e “2” deve ser igual a 1;
  - Restrição R: para que “1” seja igual a 1, “2” deve ser igual a 1 (ou seja, é impossível que “1” seja 1 se “2” for 0);
  - Restrição M: se o efeito “1” é 1 o efeito “2” é forçado a ser 0.



# Exemplo – Imprime Mensagem

- Grafo atualizado:



# Exemplo – Imprime Mensagem

- O próximo passo é estudar sistematicamente o grafo e construir uma tabela de decisão, a qual mostra os efeitos (ações) que ocorrem para todas as possíveis combinações de causas (condições).
- Portanto, se podem ocorrer  $n$  causas, a tabela de decisão contém até  $2^n$  entradas.



# Exemplo – Imprime Mensagem

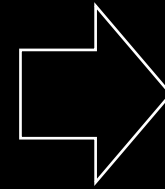
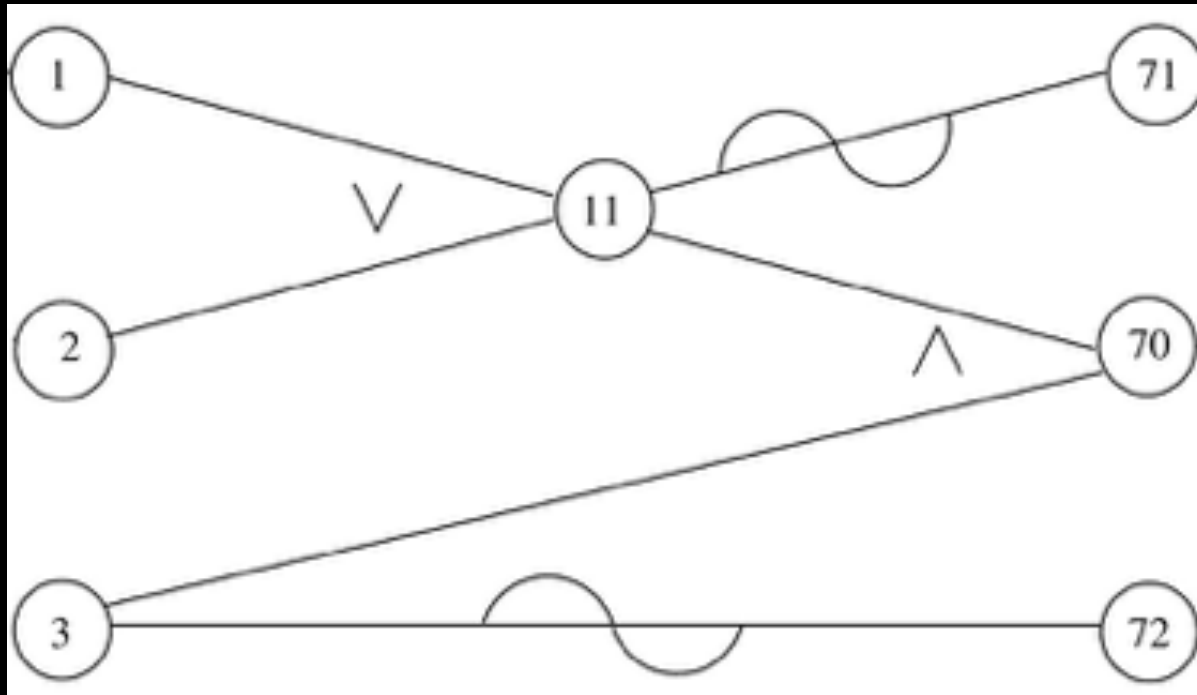
- Meyers define o seguinte procedimento para elaborar a tabela de decisão:
  1. Selecionar um efeito para estar no estado presente, isto é, com valor 1.
  2. Rastrear o grafo para trás, encontrando todas as combinações de causas (sujeitas a restrições) que fazem com que esse efeito seja 1.
  3. Criar uma coluna na tabela de decisão para cada combinação de causa.
  4. Determinar, para cada combinação, os estados de todos os outros efeitos, anotando na tabela.

# Exemplo – Imprime Mensagem

- Ao executar o passo 2, fazer as seguintes considerações:
  1. Quando o nó for do tipo OR e a saída deva ser 1, nunca atribuir mais de uma entrada com valor 1 simultaneamente. O objetivo disso é evitar que alguns erros não sejam detectados pelo fato de uma causa mascarar outra.
  2. Quando o nó for do tipo AND e a saída deva ser 0, todas as combinações de entrada que levem à saída 0 devem ser enumeradas. No entanto, se a situação é tal que uma entrada é 0 e uma ou mais das outras entradas é 1, não é necessário enumerar todas as condições em que as outras entradas sejam iguais a 1.
  3. Quando o nó for do tipo AND e a saída deva ser 0, somente uma condição em que todas as entradas sejam 0 precisa ser enumerada. (Se esse AND estiver no meio do grafo, de forma que suas entradas estejam vindo de outros nós intermediários, pode ocorrer um número excessivamente grande de situações nas quais todas as entradas sejam 0).

# Exemplo – Imprime Mensagem

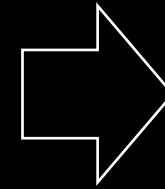
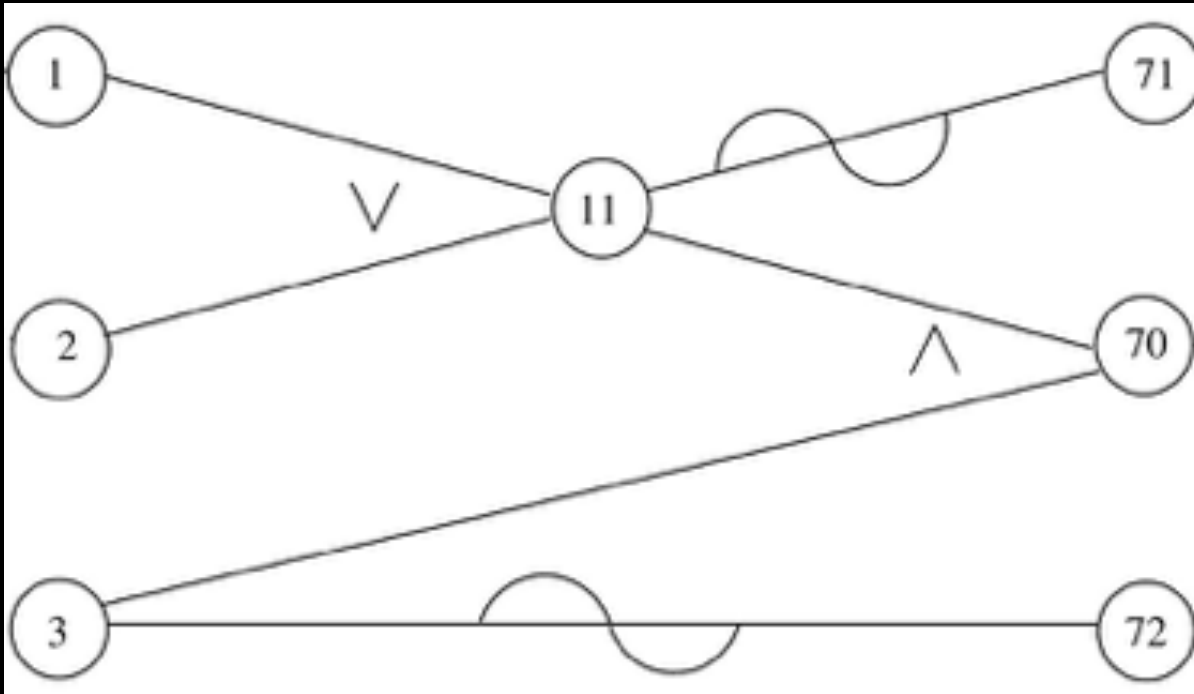
- Tabela de Decisão:



1	0	0	1	0	1
2	0	0	0	1	0
3	0	1	1	1	0
70	0	0	1	1	0
71	1	1	0	0	0
72	1	0	0	0	1

# Exemplo – Imprime Mensagem

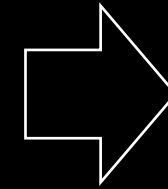
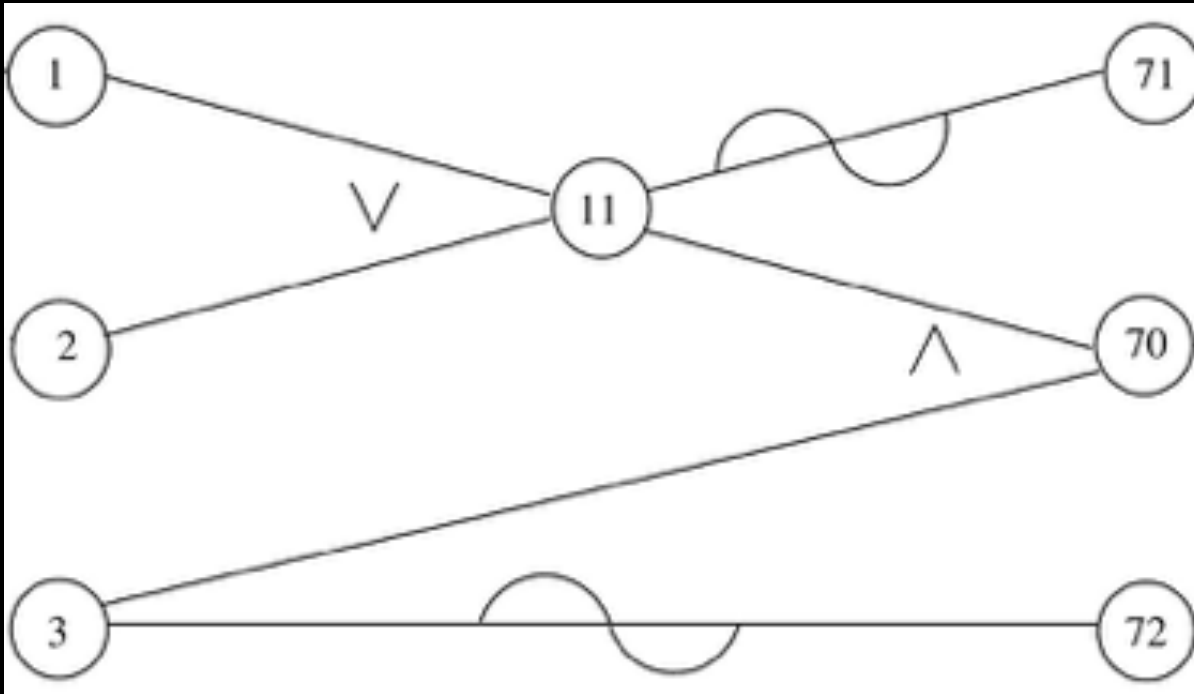
- Tabela de Decisão:



Causa	1	0	0	1	0	1
	2	0	0	0	1	0
	3	0	1	1	1	0
Efeito	70	0	0	1	1	0
	71	1	1	0	0	0
	72	1	0	0	0	1

# Exemplo – Imprime Mensagem

- Tabela de Decisão:



		Casos de Teste					
Entrada		1	0	0	1	0	1
		2	0	0	0	1	0
		3	0	1	1	1	0
Saída		70	0	0	1	1	0
		71	1	1	0	0	0
		72	1	0	0	0	1

# Exemplo - Cal

- O primeiro passo para aplicação do critério de causa-efeito é a identificação das possíveis condições de entrada (causas) e as de saída (efeitos). Para o programa cal , foram identificadas as seguintes causas:
  1. argumento m fornecido;
  2. argumento m está no intervalo esperado;
  3. argumento a fornecido;
  4. argumento a está no intervalo esperado;
  5. argumento a refere-se a ano bissexto;
  6. argumento a refere-se a ano de mudança de calendário;
  7. argumentos referem-se a mês bissexto;
  8. argumentos referem-se a mês de mudança de calendário.

# Exemplo - Cal

- Os efeitos identificados são:

- 91. mensagem de mês inválido;

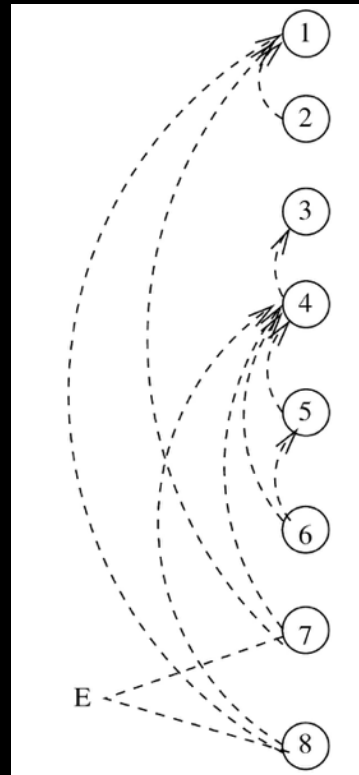
- 92. mensagem de ano inválido;

- 93. calendário do mês é apresentado;

- 94. calendário do ano é apresentado.

# Exemplo - Cal

- Restrições impostas pela Regra Negócio da Aplicação:

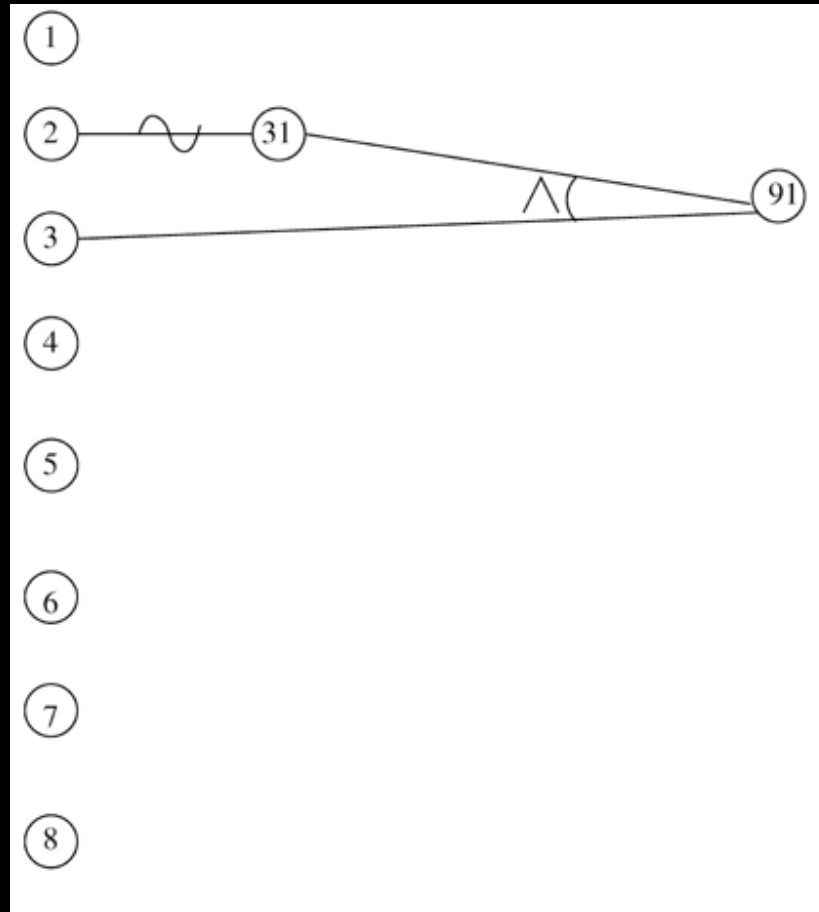




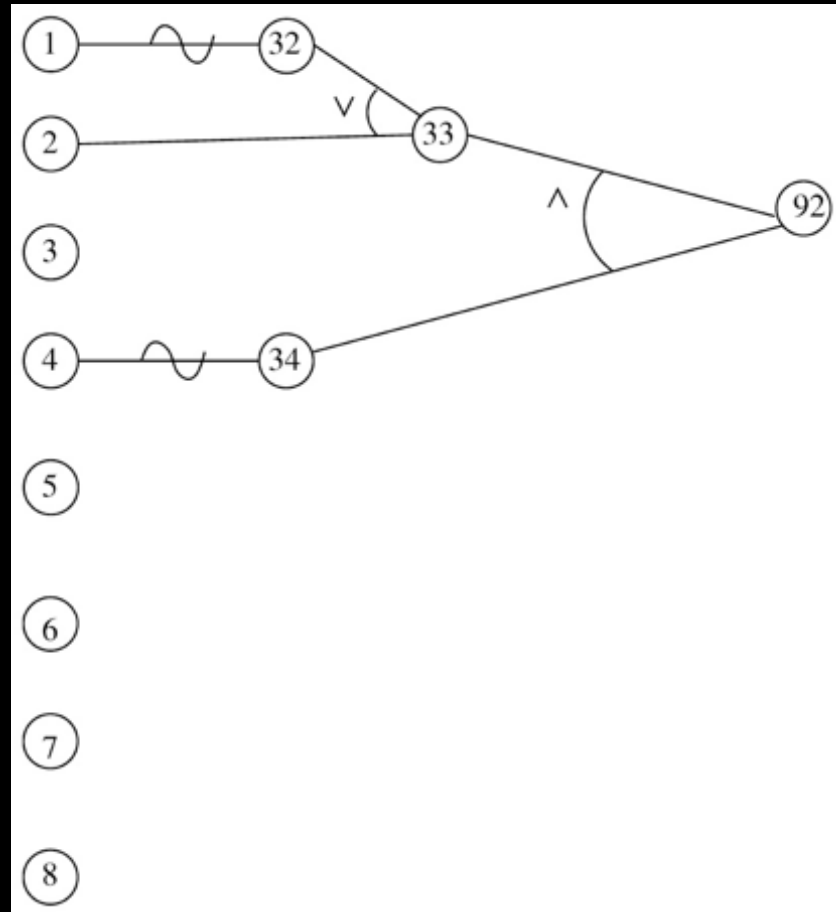
# Exemplo - Cal

- Os grafos também podem ser construídos separadamente para cada situação
- A seguir, os grafos baseados nas Causas-Efeitos Especificadas.

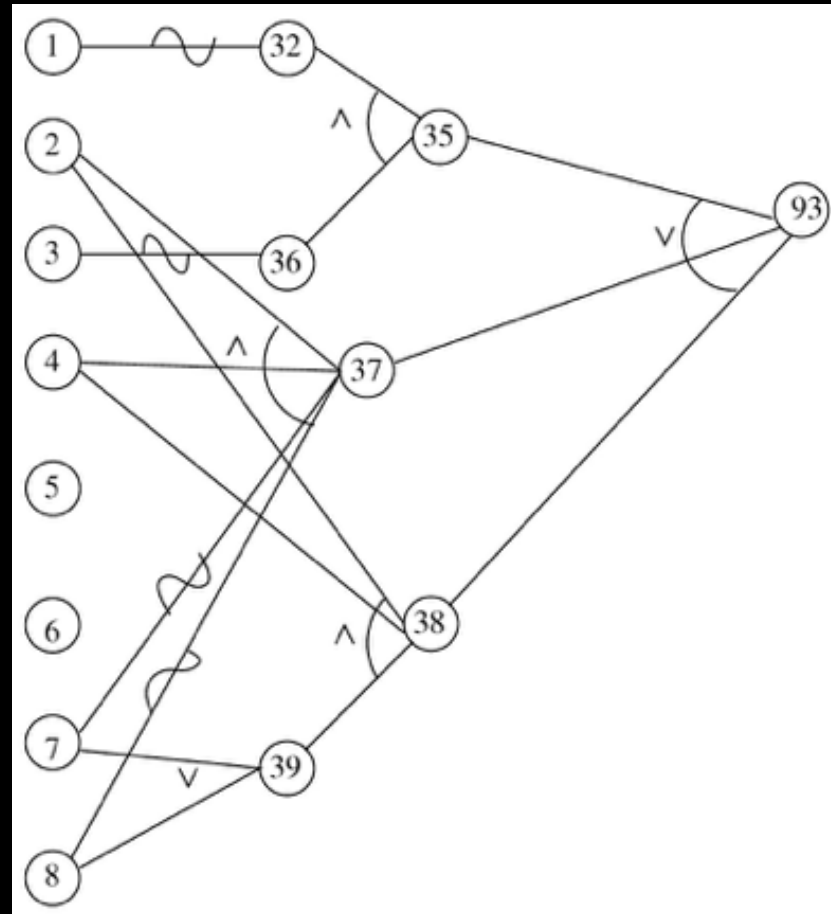
# Exemplo - Cal



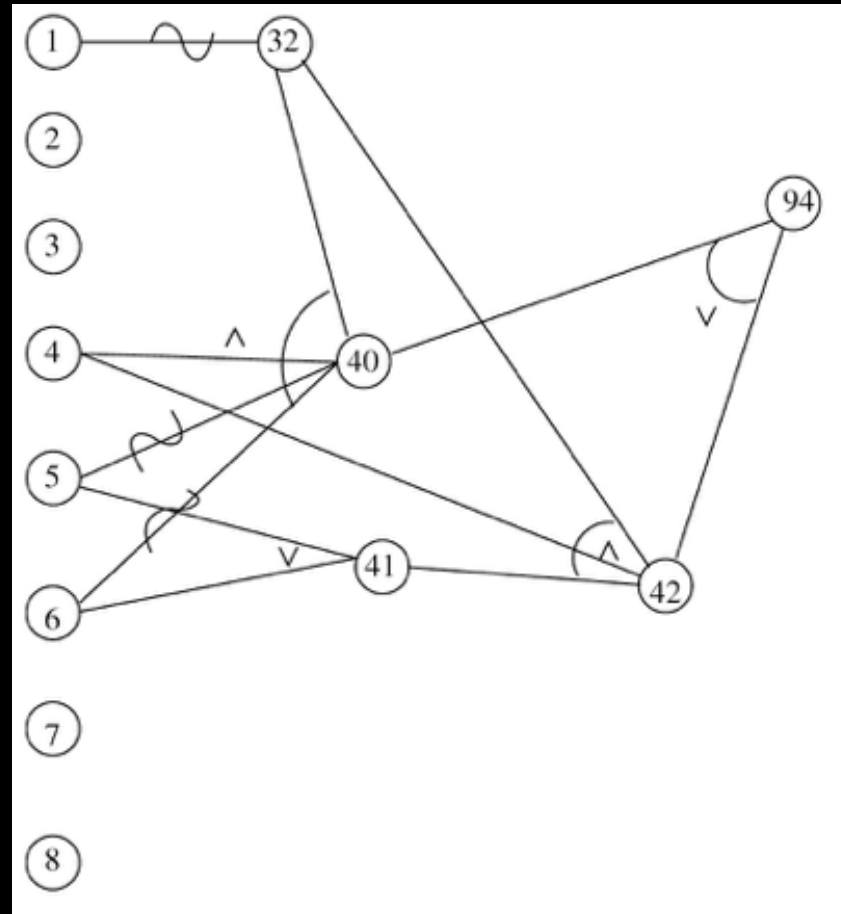
# Exemplo - Cal



# Exemplo - Cal



# Exemplo - Cal



# Error Guessing

# Error Guessing

- Adivinhação de Erros;
- A ideia do critério é enumerar possíveis erros ou situações propensas a erros e, então, definir casos de teste para explorá-las;
- Exemplo:
  - Lista vazia de entrada;
  - Lista com apenas uma entrada;
  - Todas as entradas com o mesmo valor;
  - Lista de entradas já ordenada.

# Verificação, Validação & Teste de Software

Jean Phelipe de Oliveira Lima

[jpdol.eng16@uea.edu.br](mailto:jpdol.eng16@uea.edu.br)

/jpdol