

SSC 0721 Teste e Inspeção de Software

Teste Funcional – Grafo Causa-efeito

Prof. Marcio E. Delamaro

delamaro@icmc.usp.br

Grafo causa-efeito

- Uma das limitações dos critérios anteriores é que eles não exploram combinações dos dados de entrada
- O critério causa-efeito procura suprir essa deficiência
- O grafo é uma linguagem formal na qual a especificação é traduzida

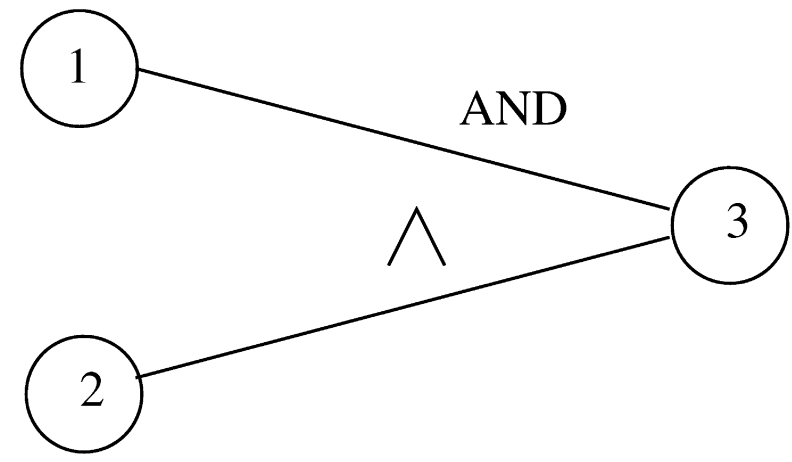
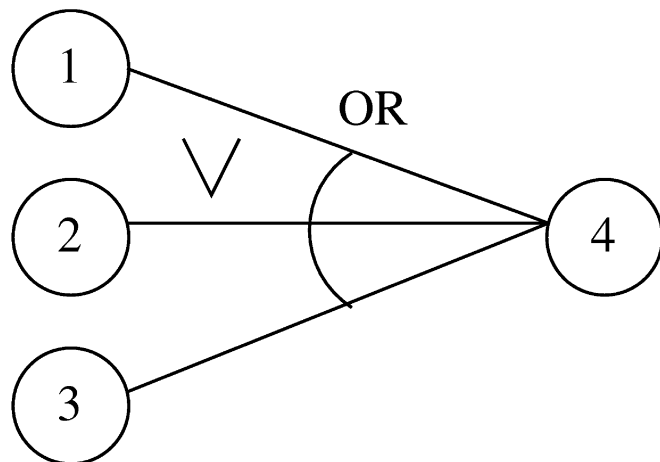
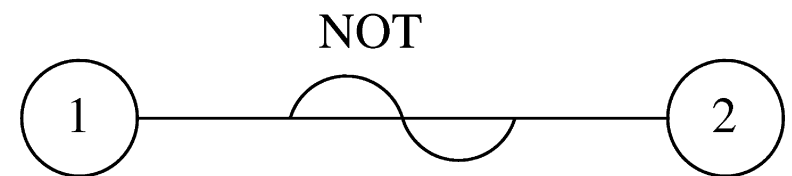
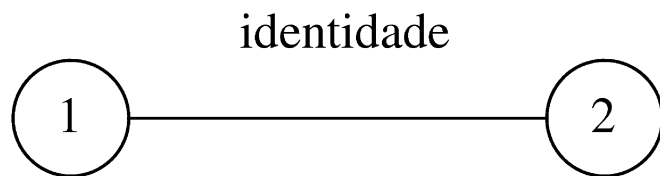
- Dividir a especificação do software em partes, pois a construção do grafo para grandes especificações torna-se bastante complexa.
- Identificar as causas e efeitos na especificação. As causas correspondem às condições de entrada, ou uma particular classe de equivalência, estímulos, ou qualquer coisa que provoque uma resposta do sistema em teste. Os efeitos correspondem às saídas, mudanças no estado do sistema ou qualquer resposta observável. Uma vez identificados, a cada um deve ser atribuído um único número.

- Analisar a semântica da especificação e transformar em um grafo booleano – o Grafo Causa-Efeito – que liga as causas e os efeitos.
- Adicionar anotações ao grafo, as quais descrevem combinações das causas e efeitos que são impossíveis devido a restrições sintáticas ou do ambiente.
- Converter o grafo em uma tabela de decisão, na qual cada coluna representa um caso de teste.
- Converter as colunas da tabela de decisão em casos de teste.

Notação

- Cada nó tem o valor 0 ou 1
- Valor 1 indica que aquele determinado estado existe (é verdadeiro)
- Valor 0 indica que estado não é verdadeiro
- 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.

Notação



Um exemplo

- “Imprime Mensagens”: o programa lê dois caracteres e, de acordo com eles, mensagens serão impressas
- 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

1 - caractere na coluna 1 é “A”

Causas

- 1 - caractere na coluna 1 é “A”
- 2 - caractere na coluna 1 é “B”

Causas

- 1 - caractere na coluna 1 é “A”
- 2 - caractere na coluna 1 é “B”
- 3 - caractere na coluna 2 é um dígito

70 - a atualização é realizada

- 70 - a atualização é realizada
- 71 - a mensagem X é enviada

- 70 - a atualização é realizada
- 71 - a mensagem X é enviada
- 72 - a mensagem Y é enviada

O grafo

1

71

2

70

3

72

O grafo

1

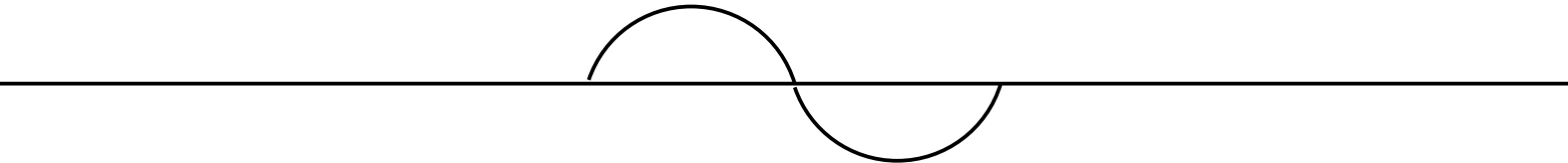
71

2

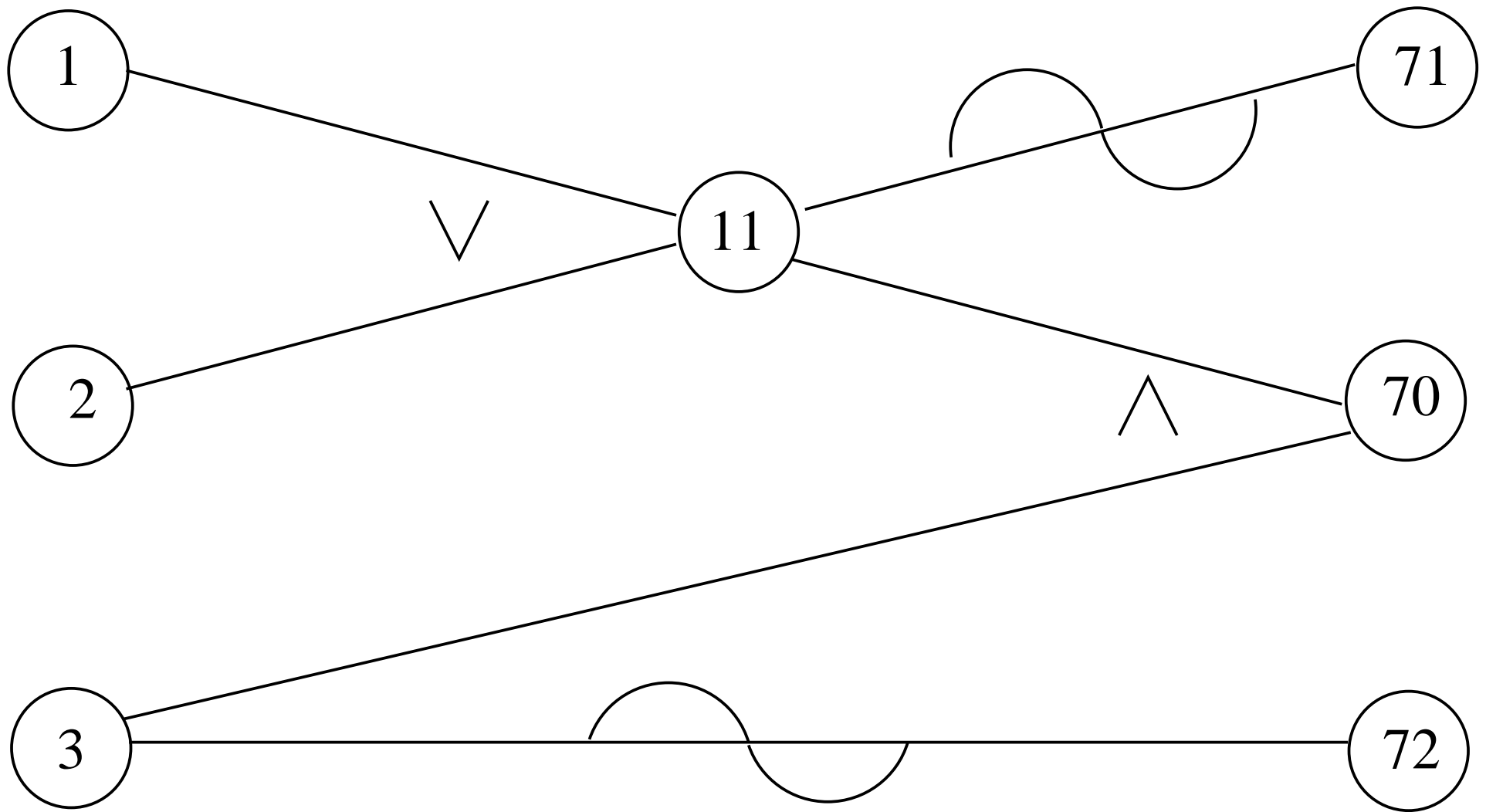
70

3

72



O grafo

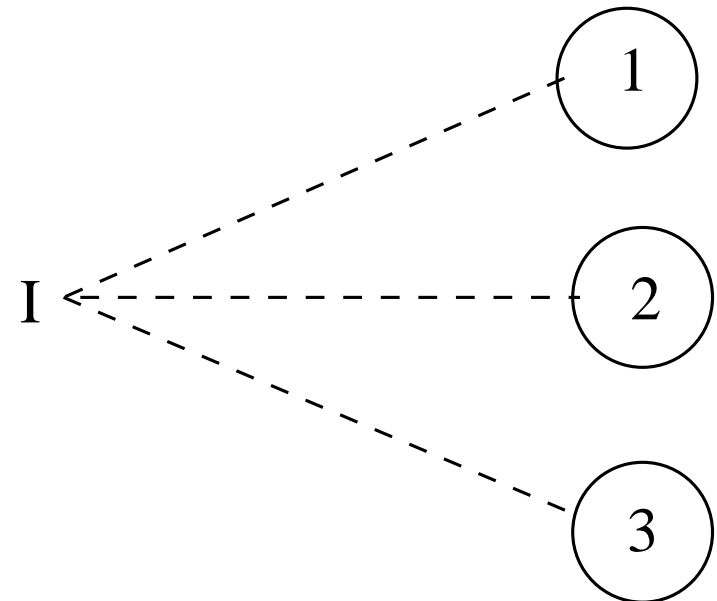
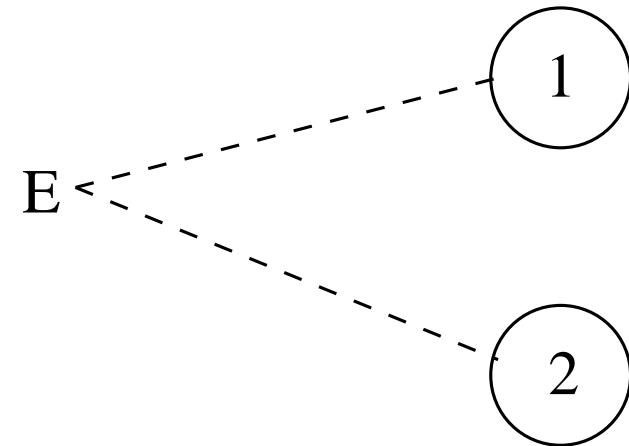


Restrições

- Para verificar se o grafo está correto, deve-se atribuir valores 0 e 1 para as causas e verificar se os efeitos assumem o valor correto
- Existem combinações que não são possíveis
- Essas restrições devem ser anotadas no grafo
- Por exemplo, causas 1 e 2 não podem ser verdadeiras ao mesmo tempo

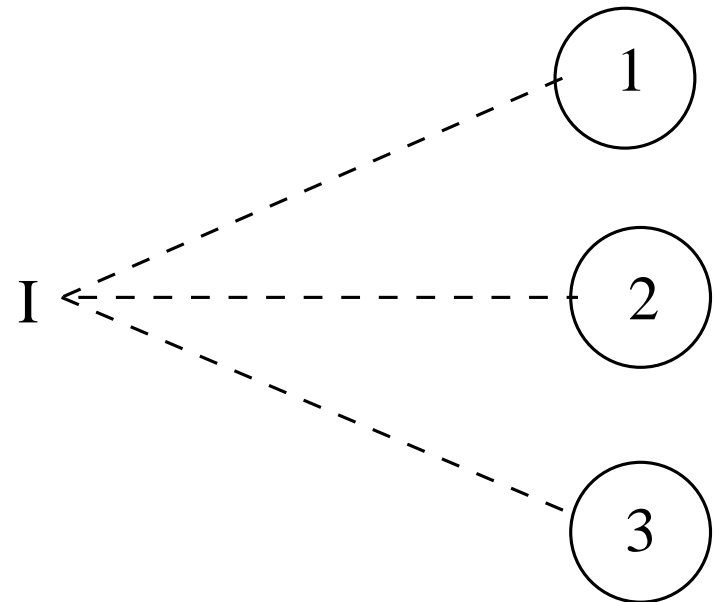
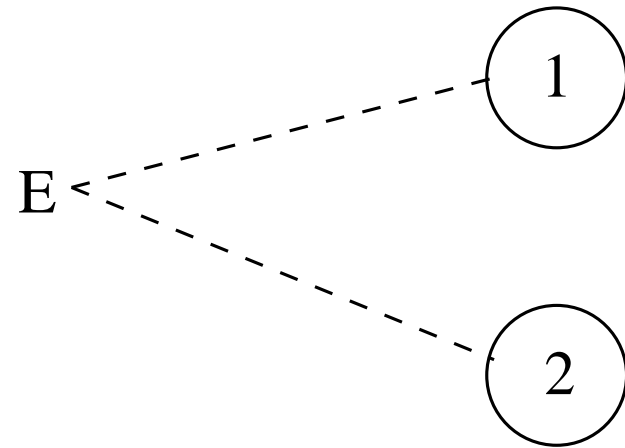
Restrições no grafo

- Restrição E: no máximo um entre “1” e “2” pode ser igual a 1 (ou seja, “1” e “2” não podem ser 1 simultaneamente).



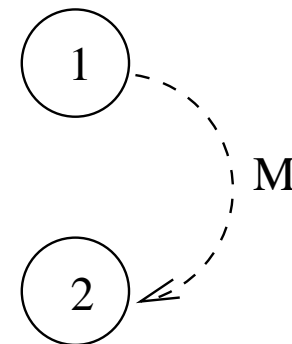
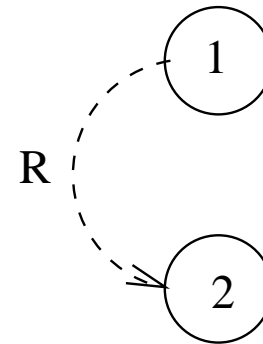
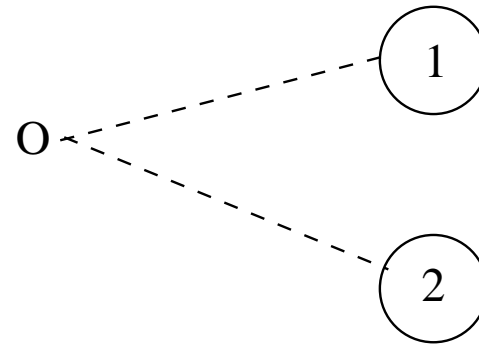
Restrições no grafo

- 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).



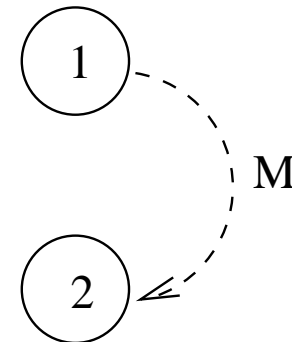
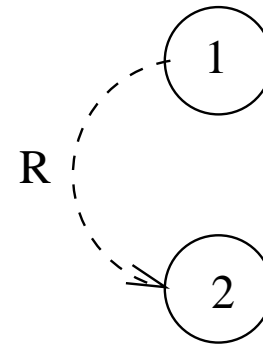
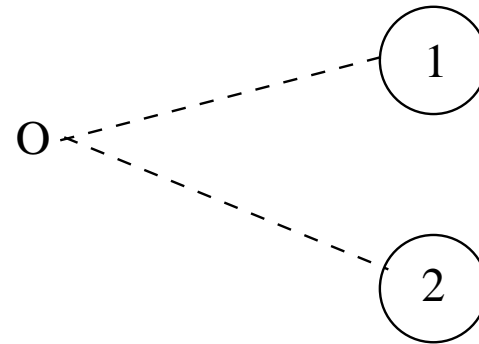
Restrições no grafo

- Restrição O: um e somente um entre “1” e “2” deve ser igual a 1.



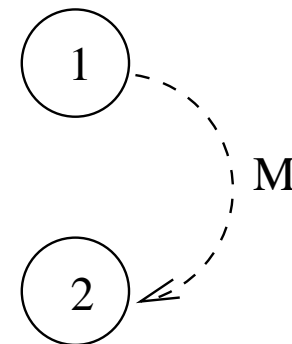
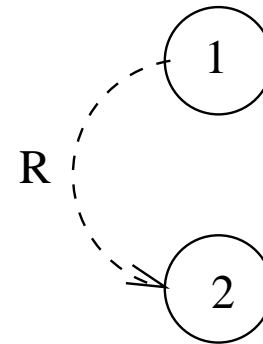
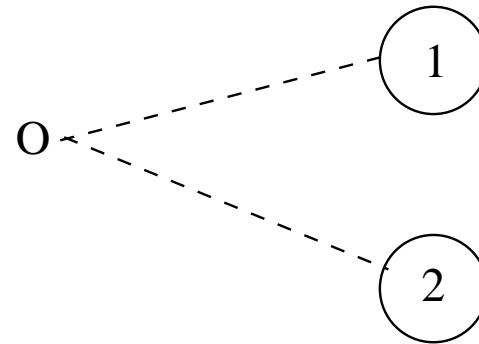
Restrições no grafo

- 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).



Restrições no grafo

- 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.



O grafo

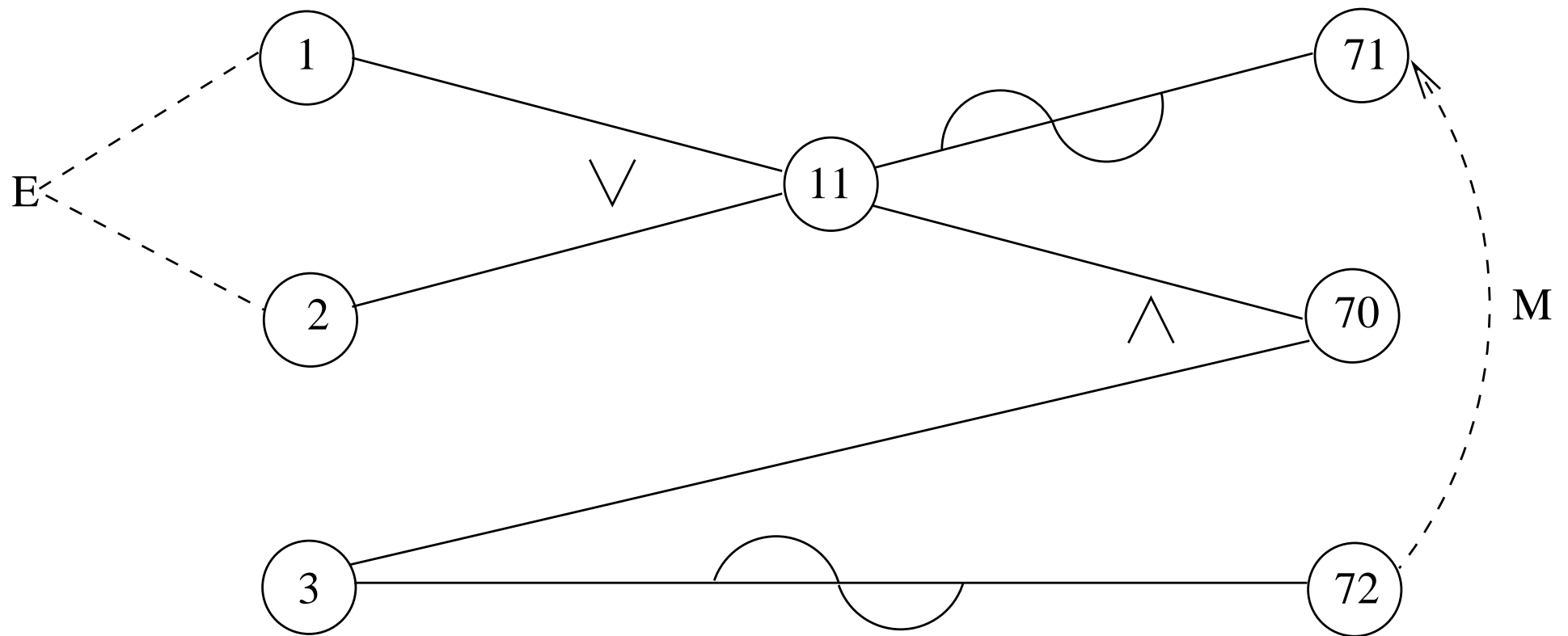


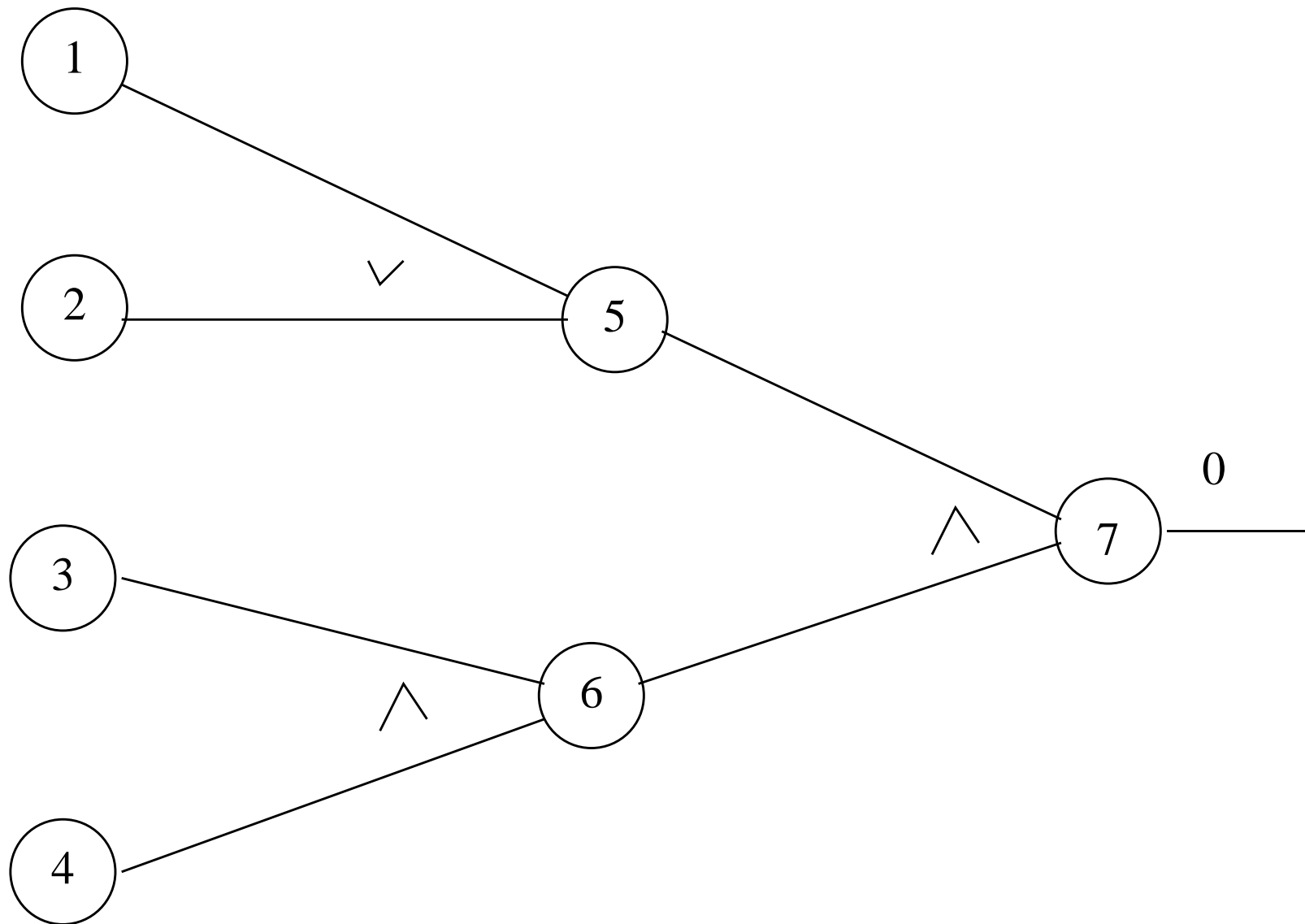
Tabela de decisão

- Forma de representar condições e ações
- O próximo passo é estudar sistematicamente o grafo e construir uma tabela de decisão.
 - 1 Selecionar um efeito para estar 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.

Considerações

- 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.)

Considerações – exemplo



Todas possibilidades

● 0 – 0 – 0 – 0 (0 – 0)

● 0 – 0 – 0 – 1 (0 – 0)

● 0 – 0 – 1 – 0 (0 – 0)

● 0 – 0 – 1 – 1 (0 – 1)

● 0 – 1 – 0 – 0 (1 – 0)

● 0 – 1 – 0 – 1 (1 – 0)

● 0 – 1 – 1 – 0 (1 – 0)

● 1 – 0 – 0 – 0 (1 – 0)

● 1 – 0 – 0 – 1 (1 – 0)

● 1 – 0 – 1 – 0 (1 – 0)

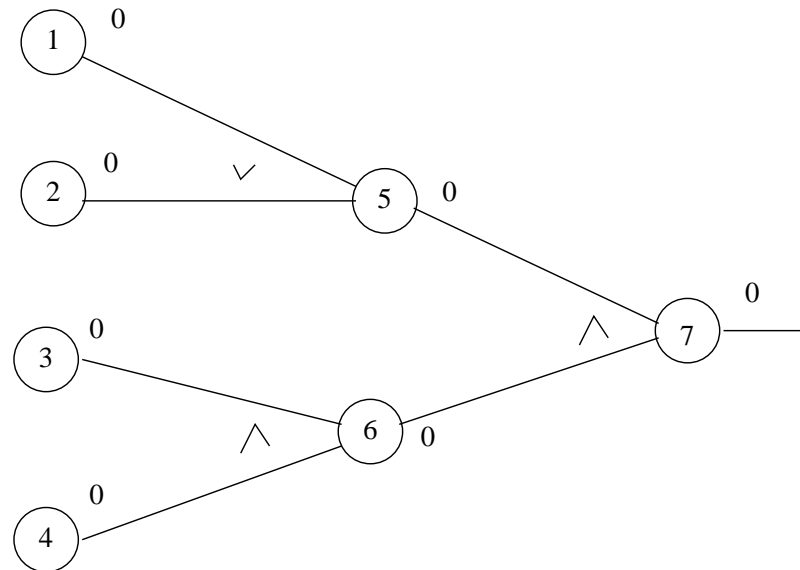
● 1 – 1 – 0 – 0 (1 – 0)

● 1 – 1 – 0 – 1 (1 – 0)

● 1 – 1 – 1 – 0 (1 – 0)

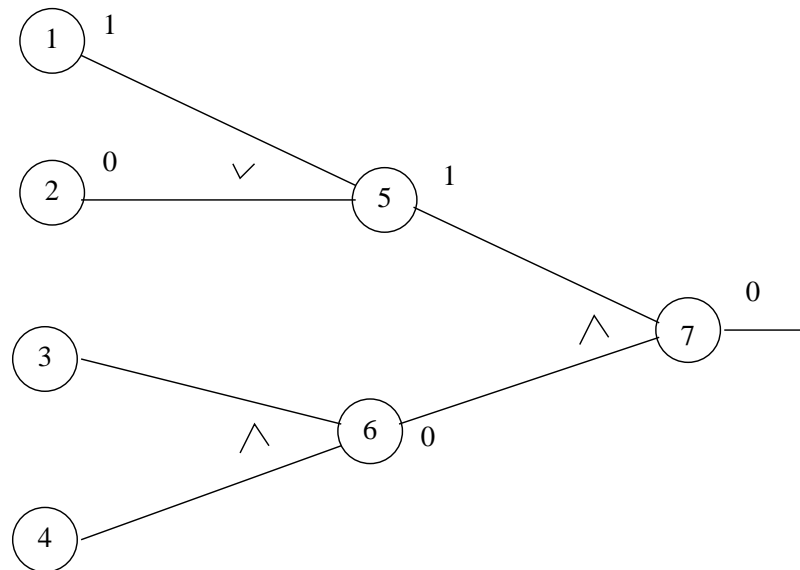
Consideração 3

- Somente uma situação em que 5 e 6 são ambos 0
- Por exemplo, selecionamos 0 – 0 – 0 – 0
- Eliminamos 0 – 0 – 0 – 1 e 0 – 0 – 1 – 0



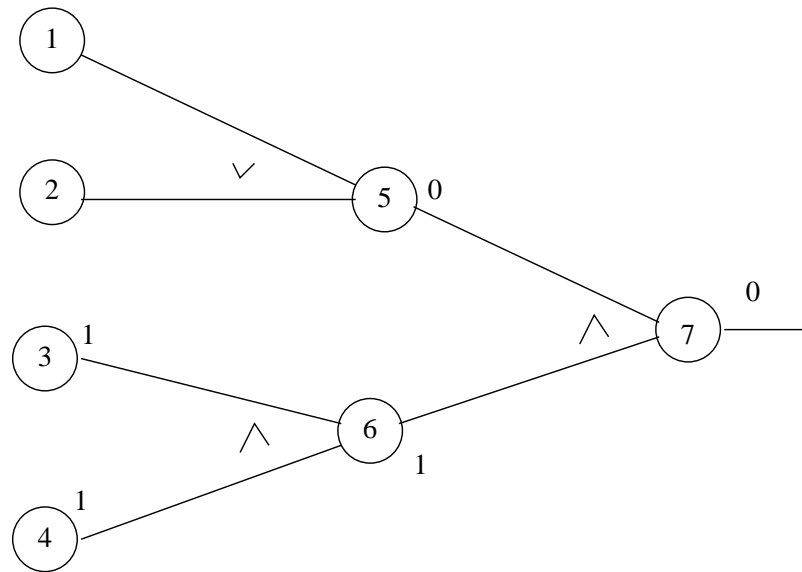
Consideração 2

- Quando 5 é 1 e 6 é 0, somente uma condição para que 5 seja 1 é requerida
- Consideração 1 diz que não se deve selecionar 1 – 1
- Por exemplo, selecionamos 1 – 0 – * – *
- Eliminamos 0 – 1 – * – * e 1 – 1 – * – *



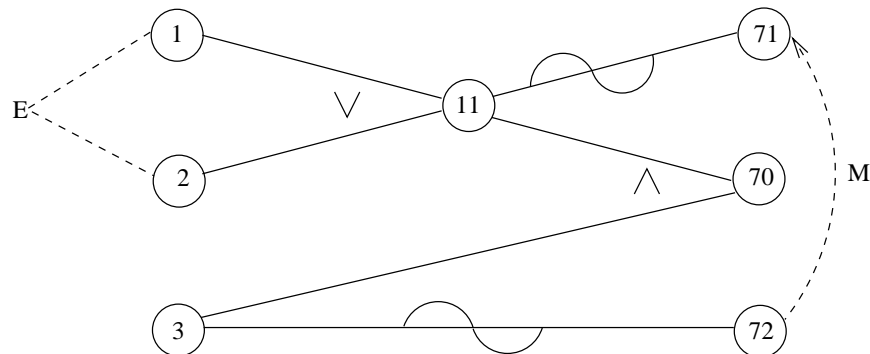
Consideração 2

- Da mesma forma, para 5 valendo 0 e 6 valendo 1, apenas uma condição para que 6 seja 1 é requerida
- Na verdade somente uma existe, que seria 0 – 0 – 1 – 1



Voltando ao exemplo

	Nós	Casos de teste				
Causas	1	0	1	0	0	1
	2	0	0	1	1	0
	3	-	1	1	0	0
Efeitos	70	0	1	1	0	0
	71	1	0	0	0	0
	72	-	0	0	1	1



Exercício

- Faça o grafo causa-efeito para o exemplo da busca na cadeia de caracteres

Exercício: causas e efeitos

- Causas

- 1 inteiro positivo no intervalo 1-20;
- 2 caractere a ser procurado está na cadeia;
- 3 procurar outro = S
- 4 procurar outro = N

- Efeitos:

- 20 inteiro fora do intervalo;
- 21 retorna posição do caractere na cadeia;
- 22 retorna caractere não encontrado;
- 23 msg opção inválida
- 24 continua execução normal

Exercício: solução

Nós	Casos de teste					
1	0	1	1	1	1	1
2	-	1	0	-	-	-
3	-	-	-	0	1	0
4	-	-	-	0	0	1
20	1	0	0	0	0	0
21	0	1	0	-	-	-
22	0	0	1	-	-	-
23	-	-	-	1	0	0
24	-	-	-	-	1	1

