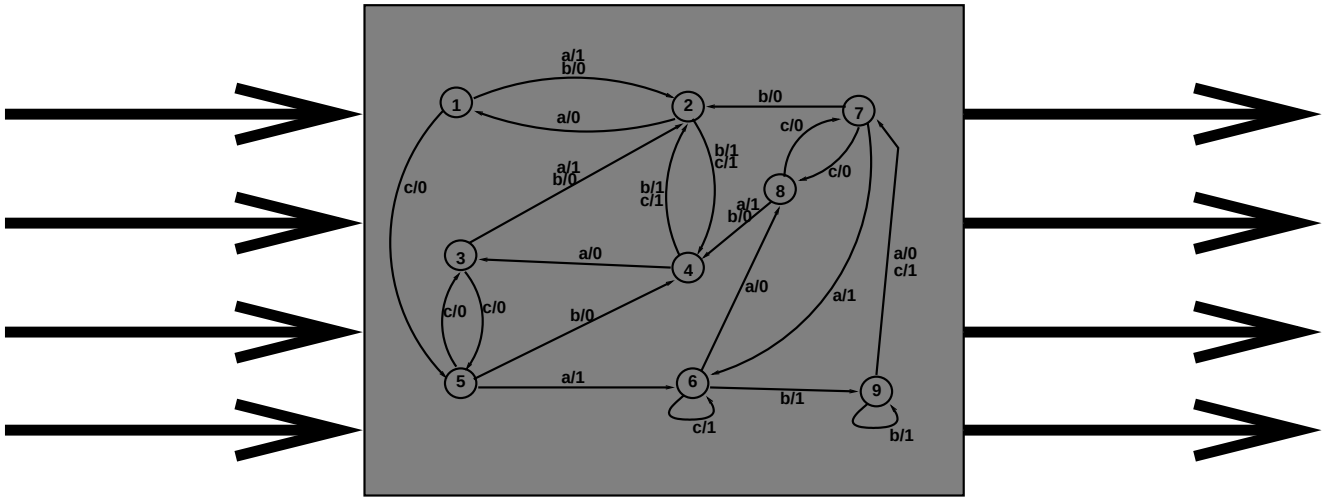
SSC5877 – Validação Verificação e Teste de Software

MEF: geração de casos de teste

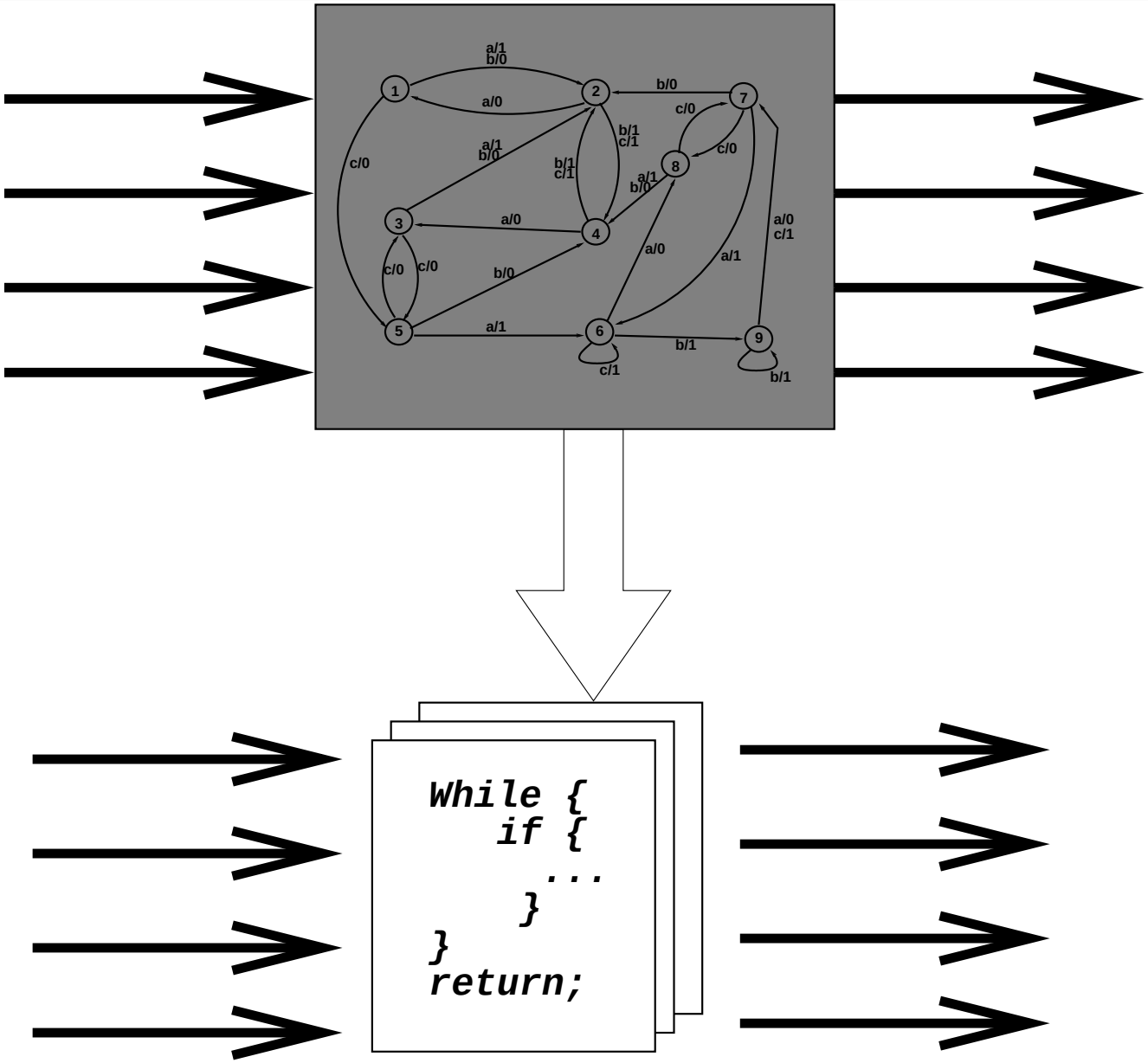
Prof. Marcio E. Delamaro

delamaro@icmc.usp.br

O problema


1
2
3
4
5
6
7
8
9
a/1
b/0
a/0
b/0
c/0
c/0
a/1
b/0
b/1
c/1
b/1
c/1
a/1
b/0
c/0
a/0
c/0
c/0
b/0
a/0
a/1
a/0
c/1
a/1
b/1
c/1
b/1

A solução


a/1
b/0
a/0
b/0
c/0
b/1
c/1
c/0
a/1
b/0
b/1
c/1
c/0
a/1
b/0
a/0
a/0
c/1
a/0
a/1
c/0
c/0
b/0
a/1
b/1
c/1
b/1
While {
if {
...
}
}
return;

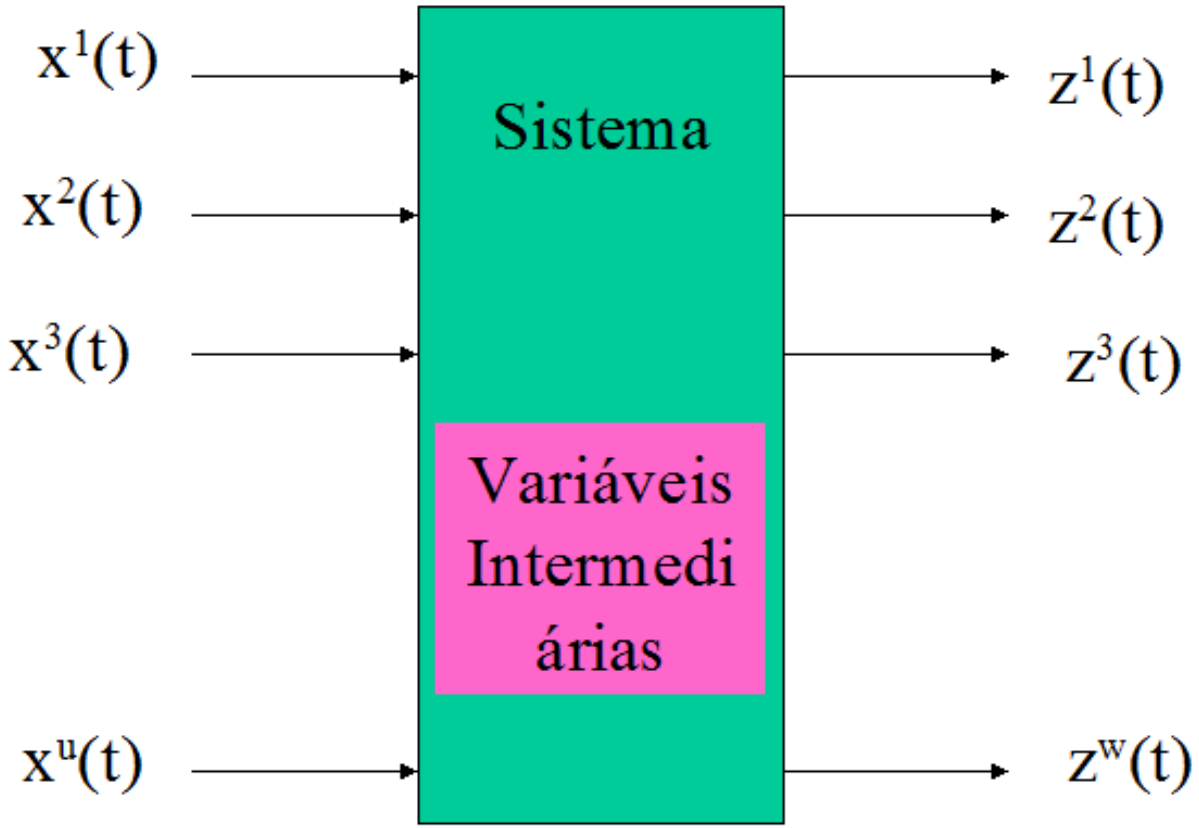
Sistema reativo

- Constante interação com o meio
- Não tem como objetivo calcular um resultado
- Não pode ser especificado através de uma função que relaciona entrada/saída
- Todo seu **comportamento** deve ser especificado

Sistemas reativos

- Exemplos
  - S.O.
  - Sistema de bordo de uma aeronave
  - Reator nuclear
  - Micro-ondas, celular, relógio
- Máquinas de estados finitos (MEF)
- Finite state machines (FSM)

MEF – modelo básico



Variáveis de Entrada

Variáveis de Saída

Suposições

- Tempo é discreto
  - As variáveis não são monitoradas constantemente
  - Existem instantes t_v em que esses valores são amostrados
  - t_v está relacionado com o número (v) da amostra e não com o tempo
  - Existem uma fonte de sincronização que determina quando a amostragem é feita

Suposições

- Alfabeto finito
  - As variáveis podem assumir um número finito de valores
  - Assim pode-se definir assumir uma única variável de entrada e uma de saída
  - Com um alfabeto de entrada e um de saída

Variáveis únicas

Variáveis únicas

$$X_{(1)} = \{0, 1\}$$
$$X_{(2)} = \{1, 2, 3\}$$
$$Z_{(1)} = \{0, 1, 2, 3, 4, 5, 6\}$$
$$Z_{(2)} = \{0, 1, 2, 3\}$$

Variáveis únicas

$$X_{(1)} = \{0, 1\}$$
$$X_{(2)} = \{1, 2, 3\}$$
$$Z_{(1)} = \{0, 1, 2, 3, 4, 5, 6\}$$
$$Z_{(2)} = \{0, 1, 2, 3\}$$

Variáveis únicas

$X_{(1)} = \{0, 1\}$
$X_{(2)} = \{1, 2, 3\}$
$Z_{(1)} = \{0, 1, 2, 3, 4, 5, 6\}$
$Z_{(2)} = \{0, 1, 2, 3\}$

$X = \{01, 02, 03, 11, 12, 13\}$
$Z = \{00, 01, 02, 03, 10, 11,$
$12, 13, 20, 21, 22, 23, 30,$
$31, 32, 33, 40, 41, 42, 43,$
$50, 51, 52, 53, 60, 61, 62, 63\}$

Estados

- Determinam como o sistema se comporta
- É definido pelo conjunto de variáveis intermediárias
- Variável de saída pode ser unicamente determinada pelo estado corrente e pelo valor da entrada
- Próximo estado pode ser unicamente determinado pelo estado corrente e pelo valor da entrada

Exemplo

- Uma moeda é jogada. 1 ponto para a 1a. cara numa seqüência de caras. Um ponto para cada coroa exceto as 2 primeiras numa seqüência de coroas.

Exemplo

- Uma moeda é jogada. 1 ponto para a 1a. cara numa seqüência de caras. Um ponto para cada coroa exceto as 2 primeiras numa seqüência de coroas.
- Fonte de sincronização: lançamento da moeda

Exemplo

- Uma moeda é jogada. 1 ponto para a 1a. cara numa seqüência de caras. Um ponto para cada coroa exceto as 2 primeiras numa seqüência de coroas.
- Fonte de sincronização: lançamento da moeda
- Alfabeto de entrada: {cara, coroa}

Exemplo

- Uma moeda é jogada. 1 ponto para a 1a. cara numa seqüência de caras. Um ponto para cada coroa exceto as 2 primeiras numa seqüência de coroas.
- Fonte de sincronização: lançamento da moeda
- Alfabeto de entrada: {cara, coroa}
- Alfabeto de saída: $\{0, 1\}$

Exemplo

- Uma moeda é jogada. 1 ponto para a 1a. cara numa seqüência de caras. Um ponto para cada coroa exceto as 2 primeiras numa seqüência de coroas.
- Fonte de sincronização: lançamento da moeda
- Alfabeto de entrada: {cara, coroa}
- Alfabeto de saída: $\{0, 1\}$
- Estados: {1 coroa, 2 coroas, cara}

Exemplo

- Uma moeda é jogada. 1 ponto para a 1a. cara numa seqüência de caras. Um ponto para cada coroa exceto as 2 primeiras numa seqüência de coroas.
- Fonte de sincronização: lançamento da moeda
- Alfabeto de entrada: {cara, coroa}
- Alfabeto de saída: $\{0, 1\}$
- Estados: {1 coroa, 2 coroas, cara}
- Verificar que dado o estado corrente e a entrada, pode-se determinar o valor da saída e o próximo estado.

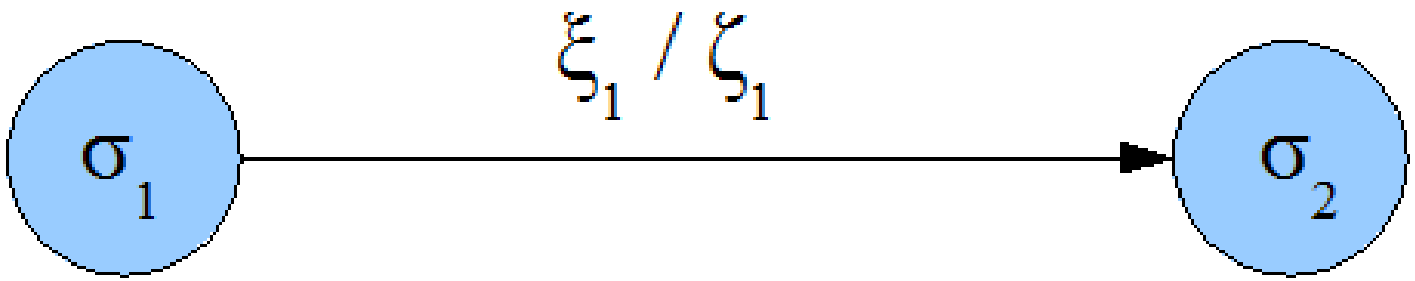
Definição: MEF

- alfabeto de entrada: $X = \{\xi_1, \xi_2, ...\xi_p\}$
- alfabeto de saída: $Z = \{\zeta_1, \zeta_2, ...\zeta_q\}$
- consjunto de estadod:$S = \{\sigma_1, \sigma_2, ..., \sigma_n\}$
- Funções de caracterização:
  - função de saída $z_v = f_z(x_v, s_v)$
  - função de transição $s_{v+1} = f_s(x_v, s_v)$

Representação: tabular

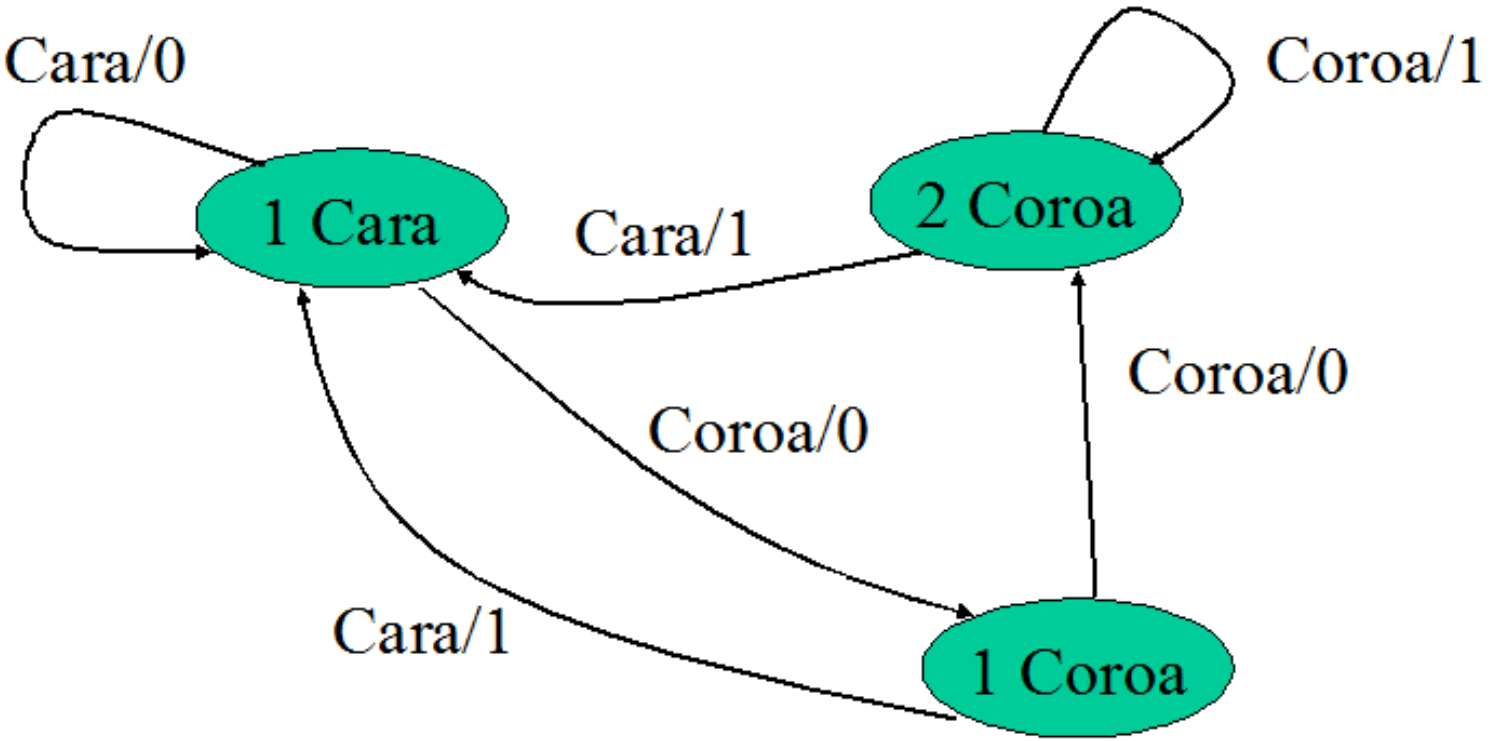
	z_v				s_{v+1}			
S_v	ξ_1	ξ_2	...	ξ_p	ξ_1	ξ_2	...	ξ_p
σ_1								
σ_2								
.								
.								
.								
σ_n								

Diagrama de transição

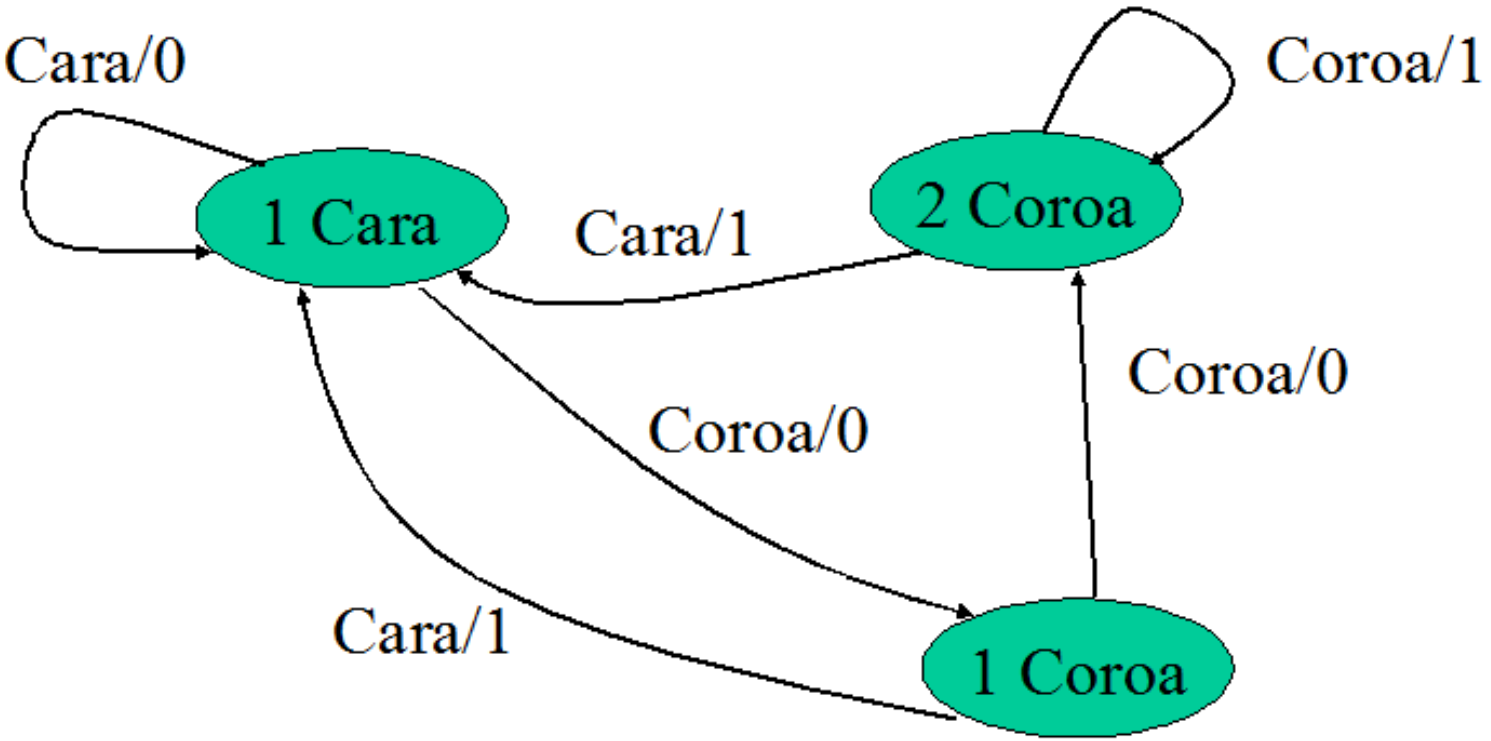
- Estados: círculos
- Transição: seta rotulada
- Transição representa a ocorrência de um evento
- Rotulo: evento/saída
- Exemplo:
  $\zeta_1 = f_z(\xi_1, \sigma_1)$
  $\sigma_2 = f_s(\xi_1, \sigma_1)$
- 

Exemplo: cara e coroa

Exemplo: cara e coroa



Exemplo: cara e coroa



ca	co	co	co	co	ca	co	ca	ca
0	0	0	1	1	1	0	1	0

Exemplo: cara e coroa

Exemplo: cara e coroa

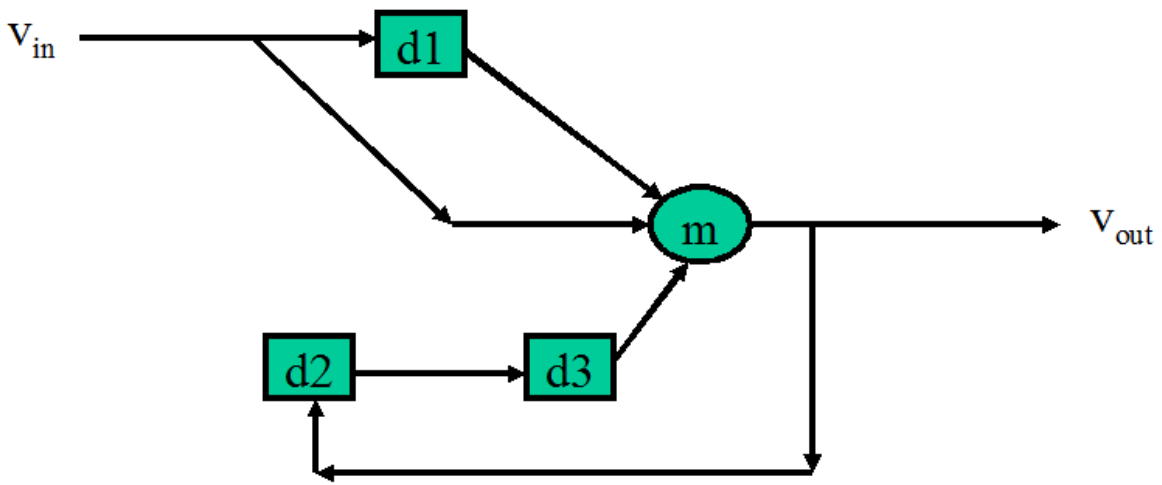
	z_v		s_{v+1}	
S_v	cara	coroa	cara	coroa
1 cara	0	0	1 cara	1 coroa
1 coroa	1	0	1 cara	2 coroa
2 coroa	1	1	1 cara	2 coroa

Exemplo: cara e coroa

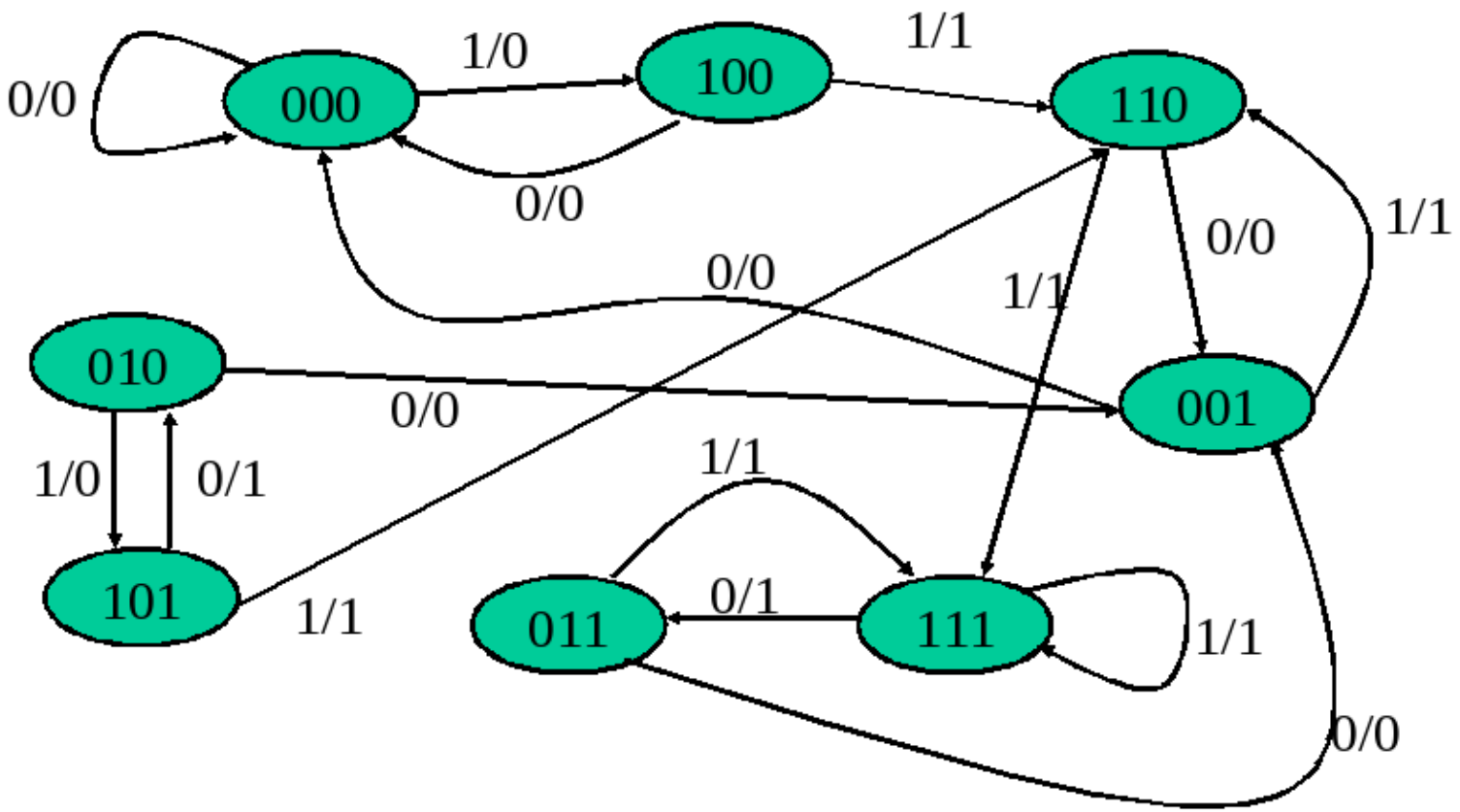
	z_v		s_{v+1}	
S_v	cara	coroa	cara	coroa
1 cara	0	0	1 cara	1 coroa
1 coroa	1	0	1 cara	2 coroa
2 coroa	1	1	1 cara	2 coroa

ca	co	co	co	co	ca	co	ca	ca
0	0	0	1	1	1	0	1	0

Exemplo: dispositivo

- v_{in} gera pulso (0 ou 1) a cada T milisegundos
- Componentes d seguram o pulso recebido pot T milisegundos
- Componente M é um votador
- X = {0,1}, Z = {0, 1}, S = {000, 001, 010, 011,..., 111}
- 

Exemplo: dispositivo

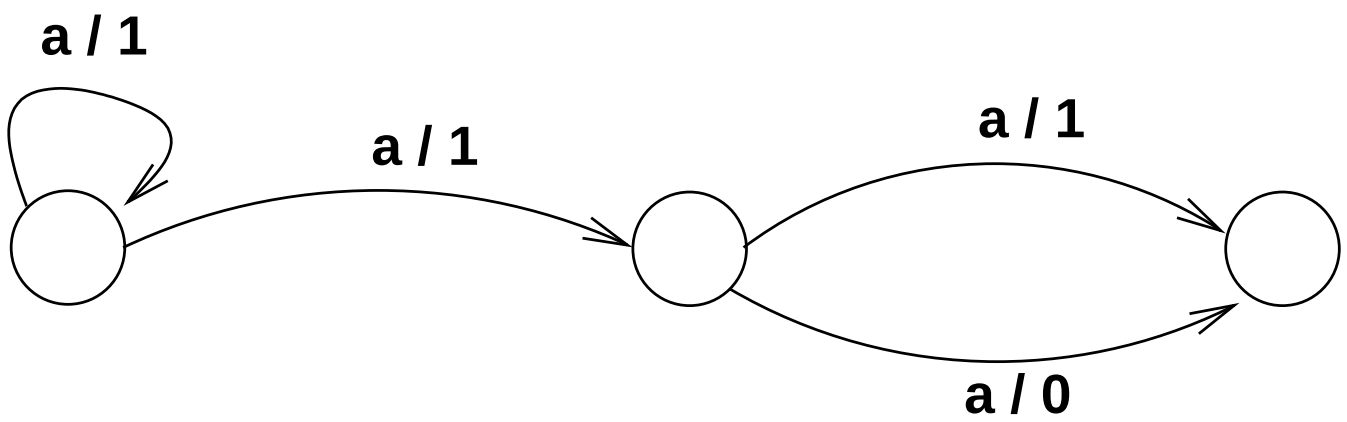


Propridades da MEF

- Muitas vezes é interessante saber se uma MEF possui ou não certa propriedade.
- Por exemplo, quando se quer aplicar determinada técnica de geração de casos de teste (que estudaremos mais adiante).
- Assim, iremos estudar alguma dessas propriedades

Determinismo

- Uma MEF é determinística se para todos os estados existe apenas um comportamento especificado para cada uma das entradas.

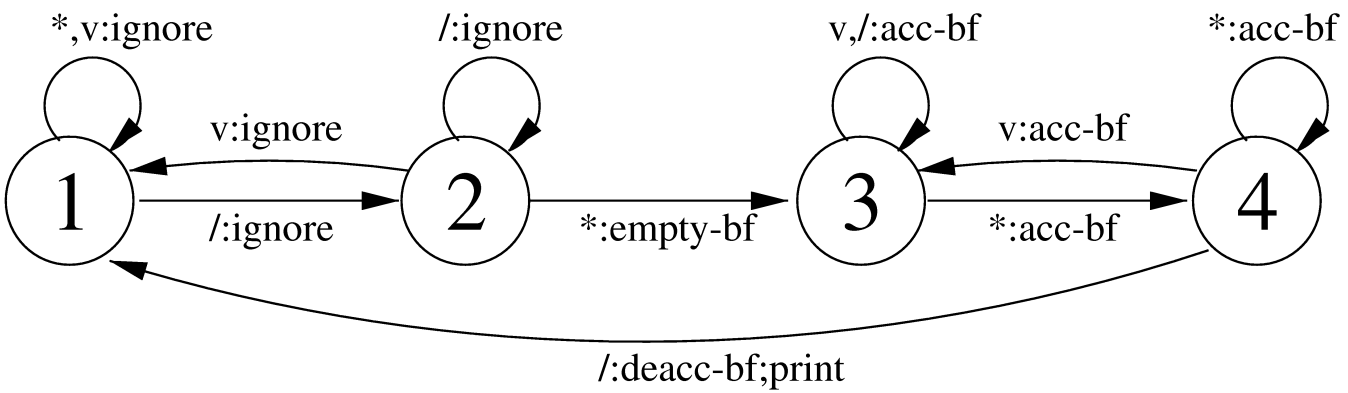


- Estamos interessados nas MEFs determinísticas

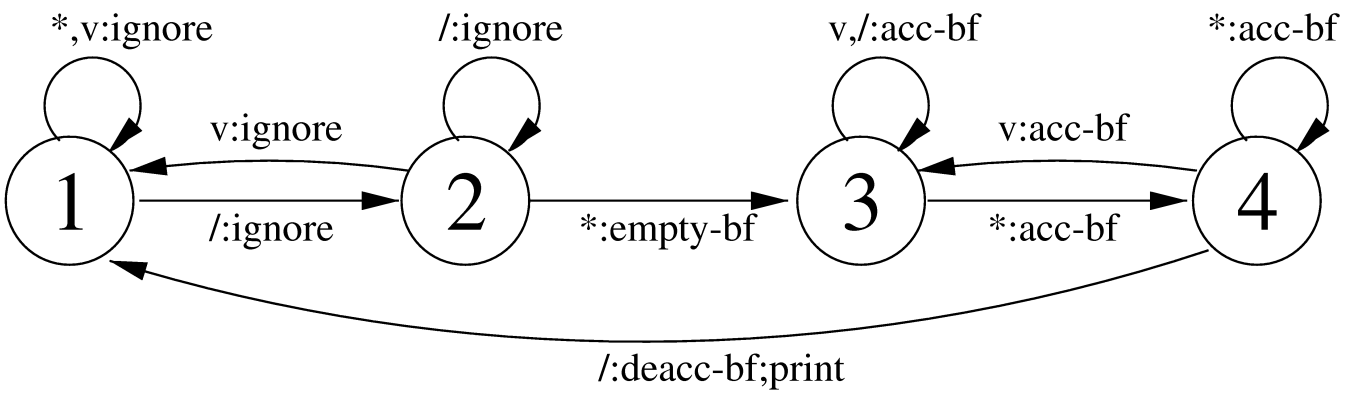
Completude

- Diz-se que a MEF é **completamente especificada** se ela trata todas a entradas em todos os estados. Caso contrário, ela é **parcialmente especificada**.
- Quando as transições não especificadas de uma MEF são completadas com fins de realização dos testes, diz-se que assume uma suposição de completude.

Completamente especificada?

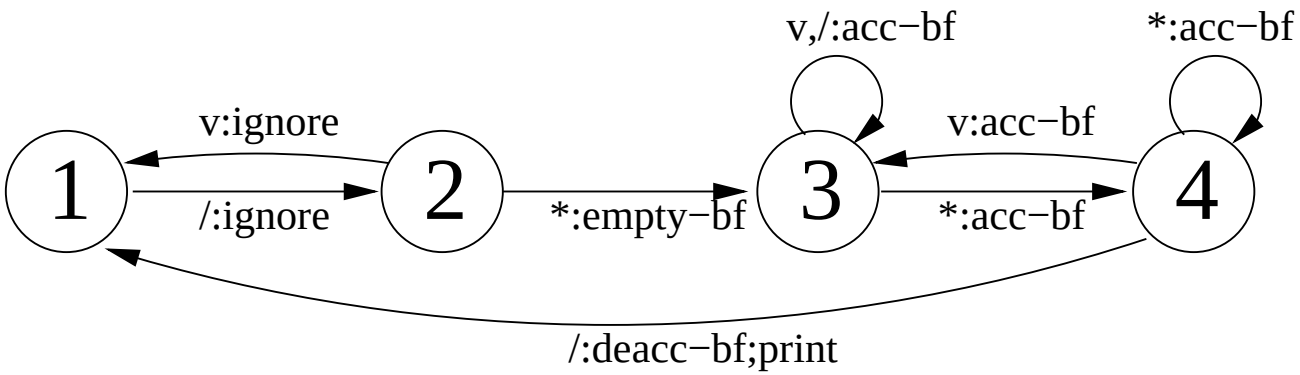


Completamente especificada?



Estado \ Entrada	*	/	v	*	/	v
1	ignore	ignore	ignore	1	2	1
2	empty-bf	ignore	ignore	3	2	1
3	acc-bf	acc-bf	acc-bf	4	3	3
4	acc-bf	deacc-bf; print	acc-bf	4	1	3

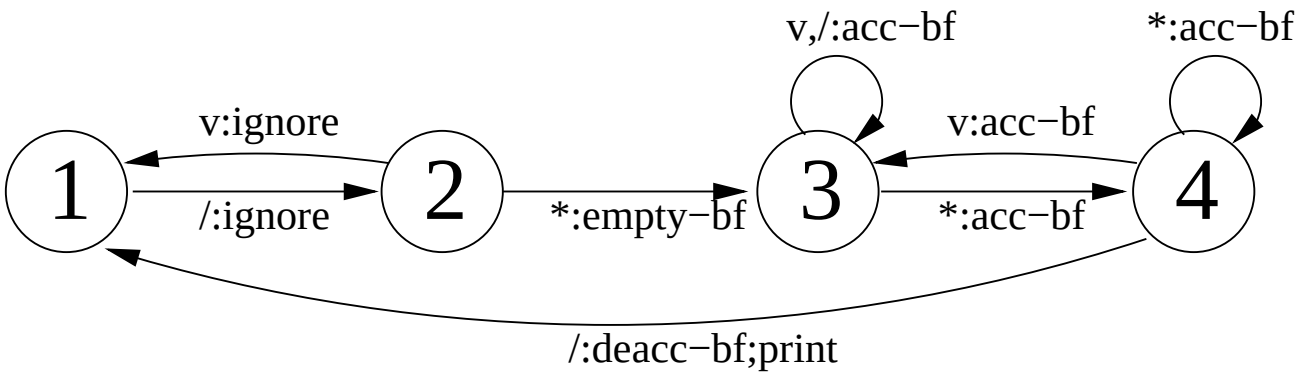
Parcialmente especificada



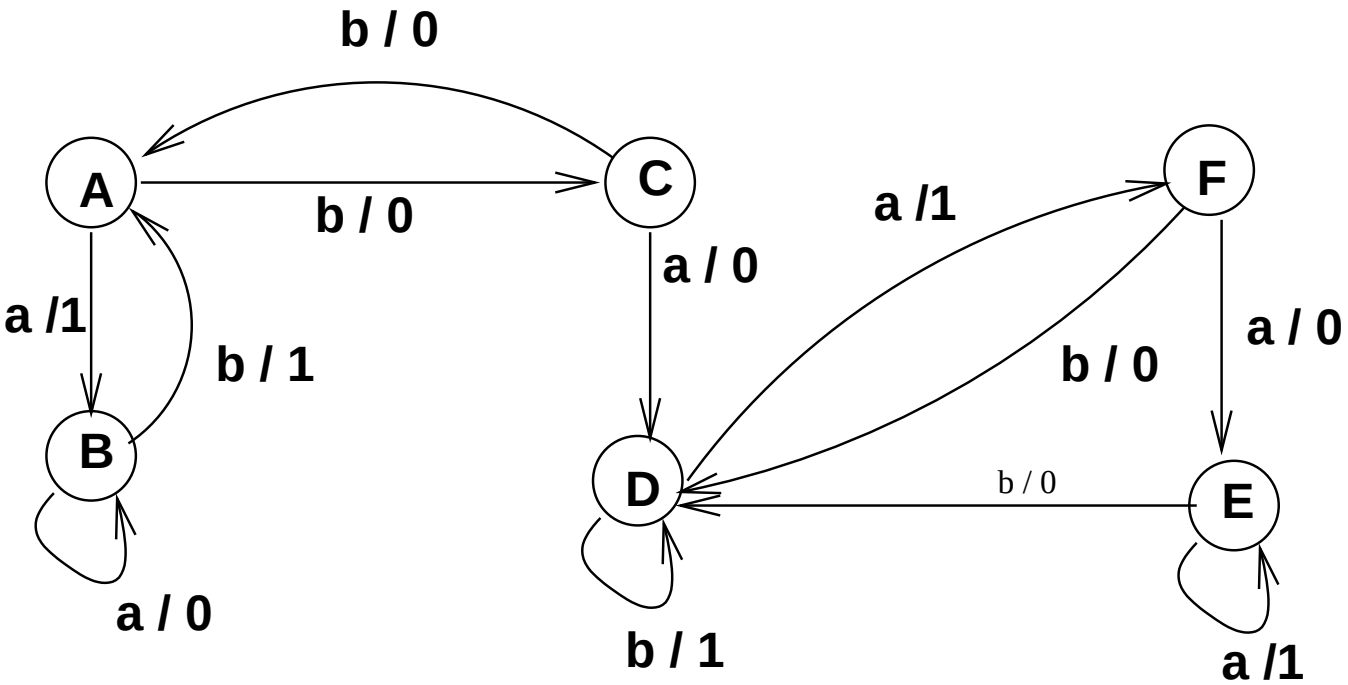
Conectividade

- Uma MEF é **fortemente conectada** se para cada par de estados (s_i, s_j) existe um caminho por transições que vai de s_i a s_j .
- Ela é **inicialmente conectada** se a partir do estado inicial (ou de algum estado) é possível atingir todos os demais estados da MEF.

Fortemente conexa



Não fortemente conexa



Equivalência

- Duas MEFs são ditas equivalentes se para qualquer seqüência de entrada ambas geram a mesma seqüência de saída.
- Dois estados (possivelmente em máquinas diferentes) são ditos equivalentes se para qualquer seqüência de entrada ambos geram a mesma seqüência de saída.
- Dois estados (MEFs) que não são equivalentes são **distinguíveis**.
- Equivalência: $\sigma_1 = \sigma_2$. Distinção: $\sigma_1 \neq \sigma_2$

Minimalidade

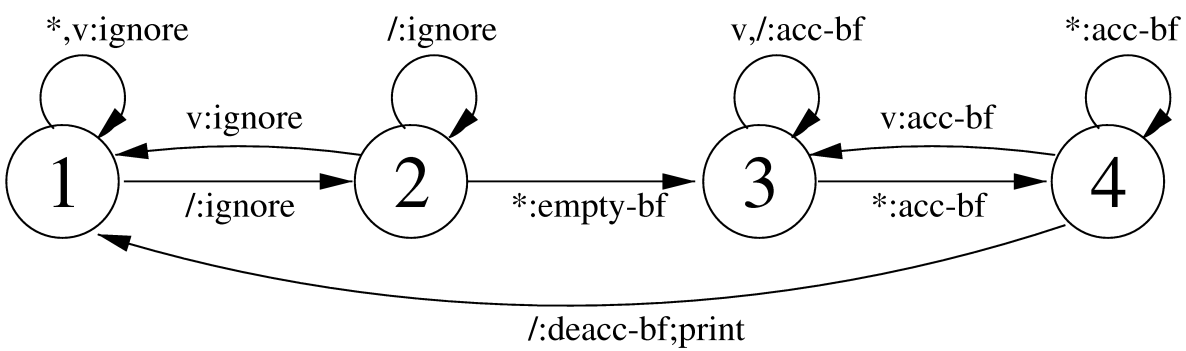
- Sejam $\Sigma_1, \Sigma_2, ...\Sigma_n$ as partições formadas pelos estados equivalentes, uma MEF é minimal se $|\Sigma_i| = 1$ para $i = 1, ..., n$
- Não existem dois estados equivalentes entre si.
- Dado uma MEF é possível verificar se é minimal ou computar uma MEF minimal **equivalente**.
- MEF minimal é **única**.

Seqüências básicas

- Os métodos de geração de casos de teste são fortemente baseados em algumas seqüências básicas, as quais são utilizadas para se obter um resultado parcial importante.
- Seqüências de sincronização
- Seqüências de distinção
- Seqüências UIO (Única de Entrada e Saída)
- Conjunto de caracterização

Seqüência de sincronização (SS)

- Uma seqüência de sincronização para o estado s é uma seqüência que leva a MEF ao estado s, independentemente do estado original.
- Por exemplo, para a máquina que extrai comentários, temos para o estado 1, a SS $\langle *, *, /, v \rangle$



-

	$*$	$*$	$/$	v
1	1	1	2	**1**
2	3	4	1	**1**
3	4	4	1	**1**
4	4	4	4	**1**

Como calcular SS

- A SS é util para colocar a MEF num estado específico (p.e.,um estado inicial) sem precisar efetuar um reset
- Para calcular, podemos o conceito de conjunto de estados, que indica os possíveis estados que a MEF pode estar.
- Definimos também $f'_s : \mathcal{P}(S) \times X \rightarrow \mathcal{P}(S)$ tal que

$$f'_s(I, x) = \{s' \mid s \in I \wedge f_s(s, x) = s'\}$$

- Por exemplo, $f'_s(\{3, 4\}, *) = \{4\}$
- Assim a definição formal de SS é:

$$f'_s(S, SS(s)) = \{s\}$$

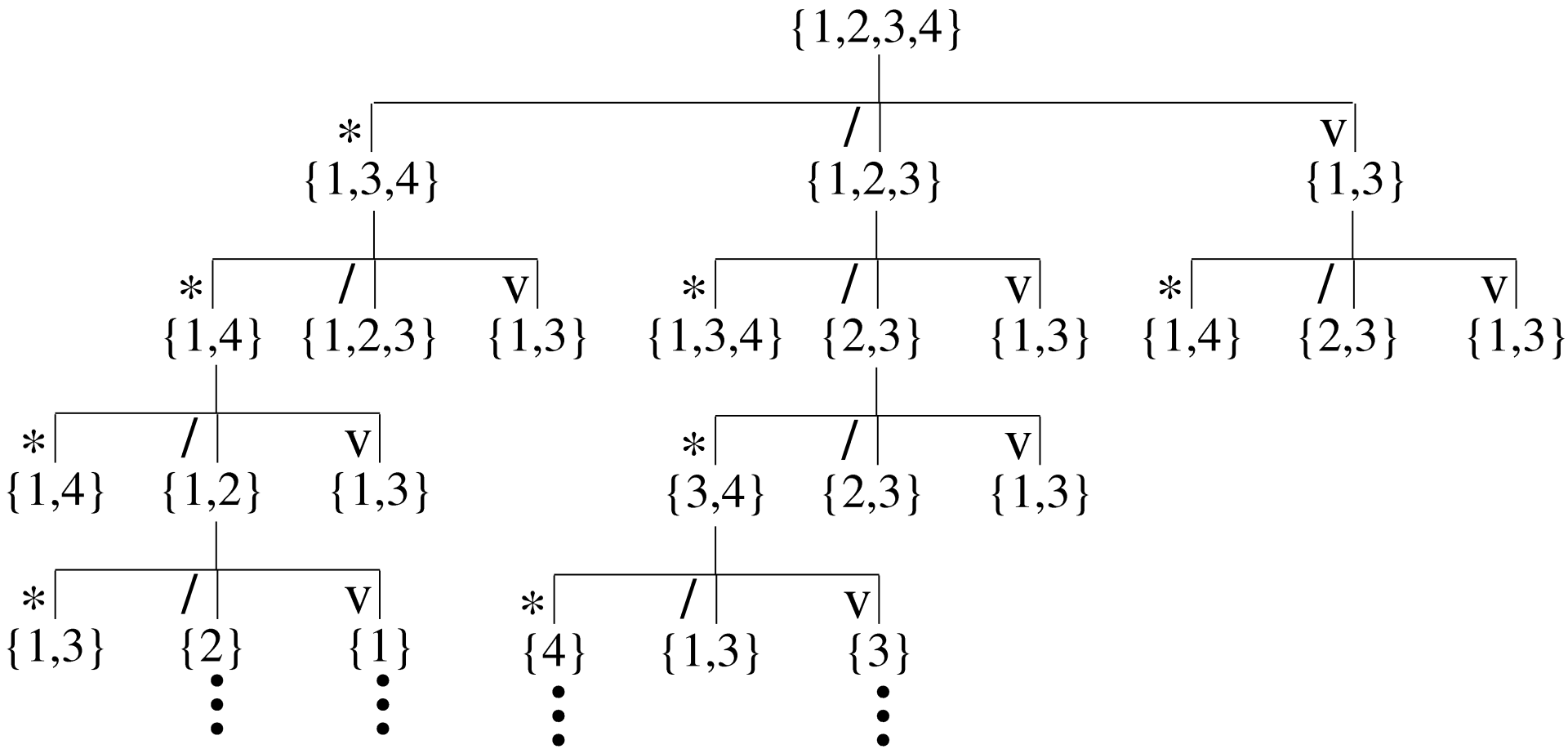
- $f'_s(\{1, 2, 3, 4\}, \langle *, *, /, v\rangle) = \{1\}$

Árvore de sincronização

- Cada nó representa um conjunto de estados
- A raiz contém todos os estados da MEF
- Cada nó tem um filho para cada símbolo de entrada
- Os filhos do nó A são dados por $f'_s(A, x)$
- Um nó só precisa ser “expandido” uma única vez.
- A SS de um nó é dada pelo caminho que vai desse nó à raiz

Exemplo de árvore de sincronização

Exemplo de árvore de sincronização



Seqüência de distinção (DS)

- Um seqüência de distinção (DS) é uma seqüência de símbolos de entrada que, quando aplicada aos estados da MEF, produzem saídas distintas para cada um dos estados.

Seqüência de distinção (DS)

- Um seqüência de distinção (DS) é uma seqüência de símbolos de entrada que, quando aplicada aos estados da MEF, produzem saídas distintas para cada um dos estados.
- Observando-se a saída produzida pela MEF como resposta à DS, pode-se determinar em qual estado a MEF estava originalmente.

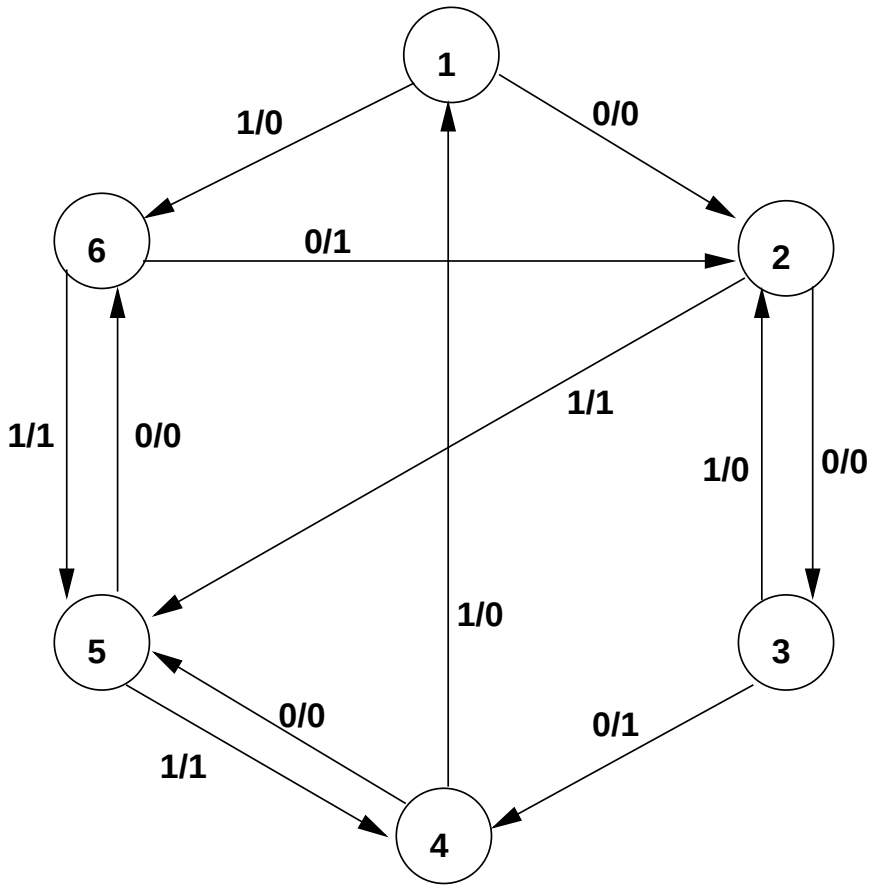
Seqüência de distinção (DS)

- Um seqüência de distinção (DS) é uma seqüência de símbolos de entrada que, quando aplicada aos estados da MEF, produzem saídas distintas para cada um dos estados.
- Observando-se a saída produzida pela MEF como resposta à DS, pode-se determinar em qual estado a MEF estava originalmente.
- Qual seria a DS para a MEF que extrai comentários?

Seqüência de distinção (DS)

- Um seqüência de distinção (DS) é uma seqüência de símbolos de entrada que, quando aplicada aos estados da MEF, produzem saídas distintas para cada um dos estados.
- Observando-se a saída produzida pela MEF como resposta à DS, pode-se determinar em qual estado a MEF estava originalmente.
- Qual seria a DS para a MEF que extrai comentários?
- A MEF anterior não possui uma DS

Exemplo de DS



	0	0	1
1	0	0	0
2	0	1	0
3	1	0	1
4	0	0	1
5	0	1	1
6	1	0	0

Como calcular

- Um grupo de incerteza é um conjunto de estados que ainda não foram distinguidos e para os quais ainda é necessário que alguma entrada seja anexada à seqüência para ela se tornar uma DS.
- Um grupo $[\{2,3,5,6\},\{2,4\}]$ indica que os estados 2, 3, 5 e 6 precisam ser distinguidos, assim como os estados 2 e 4.
- Um grupo é trivial se todos os conjuntos de estados são unitários ou vazios.
- Um grupo g pode ser particionado em relação a um símbolo de entrada x, criando um grupo g' no qual todos os conjuntos de estados são subconjuntos dos conjuntos em g.

Como calcular

- No particionamento, são mantidos juntos os estados que produzem a mesma saída em relação a x.
- Por exemplo, o grupo $[\{2,3,5,6\},\{2,4\}]$ pode ser particionado em relação a 0, criando o grupo $[\{2,5\},\{3,6\},\{2,4\}]$.
- Após ser particionado considerando-se a saída, devem-se calcular os próximos estados correspondentes ao grupo obtido em relação à entrada em questão.
- Por exemplo, após ser particionado e atualizado em relação a 0, o grupo $[\{2,3,5,6\},\{2,4\}]$ gerará o grupo $[\{3,6\},\{4,2\},\{5,3\}]$.

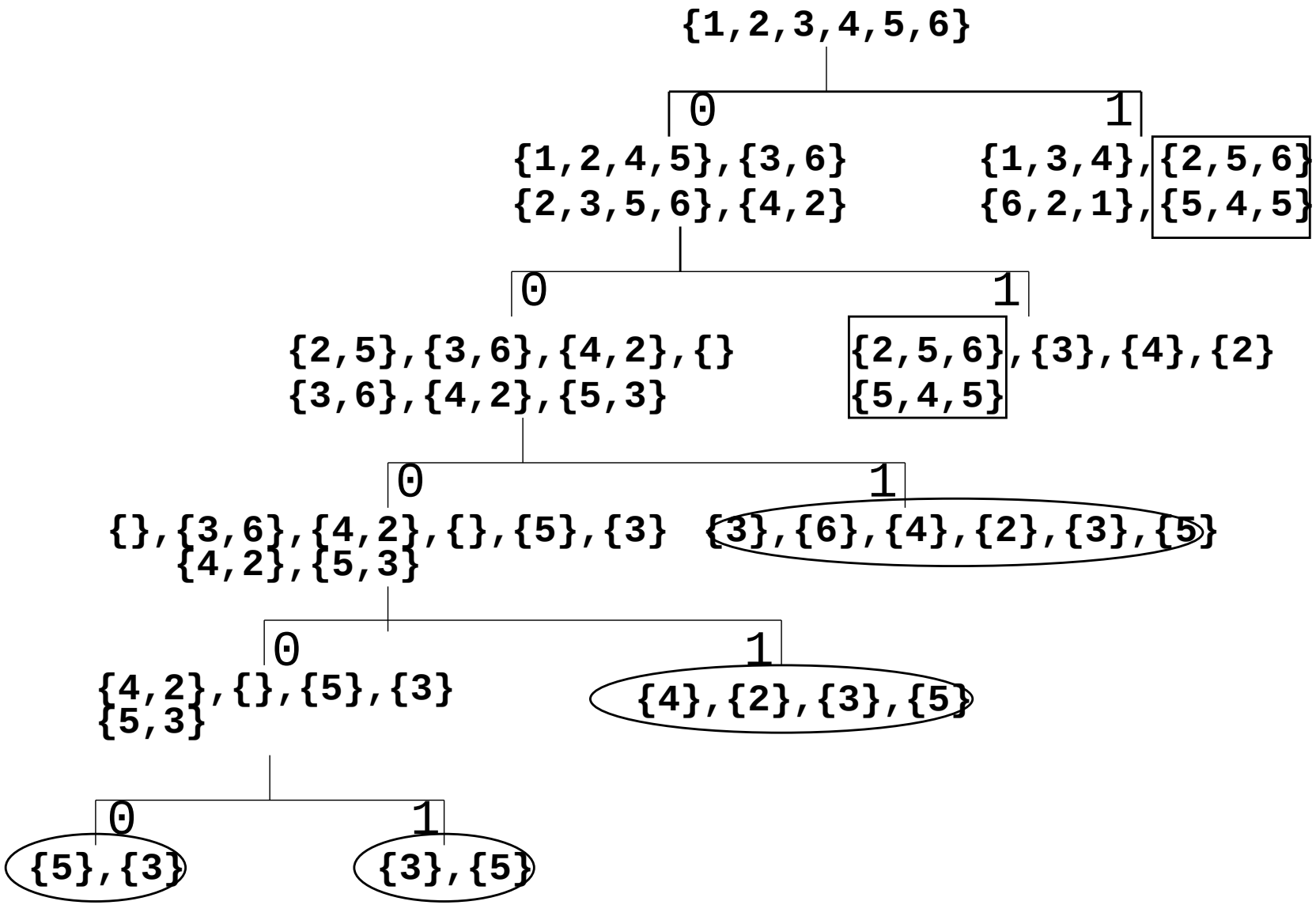
Como calcular

- Ao se atualizar um grupo, pode ocorrer de o grupo ficar inconsistente.
- Um grupo é inconsistente caso, ao calcular os próximos estados, se perceba que dois estados que ainda não foram distinguidos levam ao mesmo estado (e, portanto, não poderão ser mais distinguidos!).
- É possível verificar que, ao se particionar o grupo $[\{2,3,5,6\},\{2,4\}]$ em relação a 1, se obtém o grupo $[\{2,5,6\},\{3\},\{4\},\{2\}]$, que é inconsistente, pois tanto o estado 2 quanto o estado 6 levam ao estado 5 quando a entrada 1 é aplicada.

Árvore de distinção

- Cada nó é formado por um grupo de incerteza.
- O nó raiz é formado pelo grupo no qual todos os estados ainda não foram distinguidos (ou seja, $[S]$)
- Cada nó possui um filho para cada símbolo de entrada.
- O grupo do nó filho é calculado a partir do grupo de nó pai, particionando-o e, caso o grupo resultante não seja inconsistente, atualizando-o.
- A árvore é construída por nível a partir da raiz.
- Um nó é folha se for inconsistente ou se for trivial.
- Uma DS é a seqüência de símbolos de entrada presentes no caminho que vai da raiz da árvore a um nó cujo grupo seja trivial.

Exemplo de árvore de distinção



Seqüências UIO

- Nem todas as MEFs possuem DS.
- Mesmo nessas situações, pode-se fazer a distinção entre um estado e os demais.
- Seqüência Única de Entrada e Saída (UIO) é utilizada para verificar se a MEF está em um estado particular.
- Assim, para cada estado da MEF, pode-se ter uma UIO distinta.

Seqüências UIO

- Isso é verdade apenas se forem consideradas tanto as entradas como as saídas.
- A seqüência de entradas da UIO de um estado pode ser igual à seqüência de entradas da UIO de outro estado.
- Se uma MEF possui uma DS, ela também possui UIOs para todos os seus estados.
- Contudo, em geral, pode-se obter UIOs mais curtas.

Exemplo

	1	1
1	0	1
2	1	1
3	0	1
4	0	0
5	1	0
6	1	1

- A seqüência $\langle 1/0, 1/0 \rangle$ é uma UIO para o estado 4.
- Aplica-se a seqüência de entrada $\langle 1, 1 \rangle$. Se a seqüência de saída for $\langle 0, 0 \rangle$, então pode-se garantir que o estado original era o estado 4.
- Se a saída for $\langle 0, 1 \rangle$, não há como saber se o estado original era o estado 1 ou o estado 3.
- A $\langle 1, 1 \rangle$ também pode formar uma UIO para o estado 5 (que, nesse caso, seria a seqüência $\langle 1/1, 1/0 \rangle$)

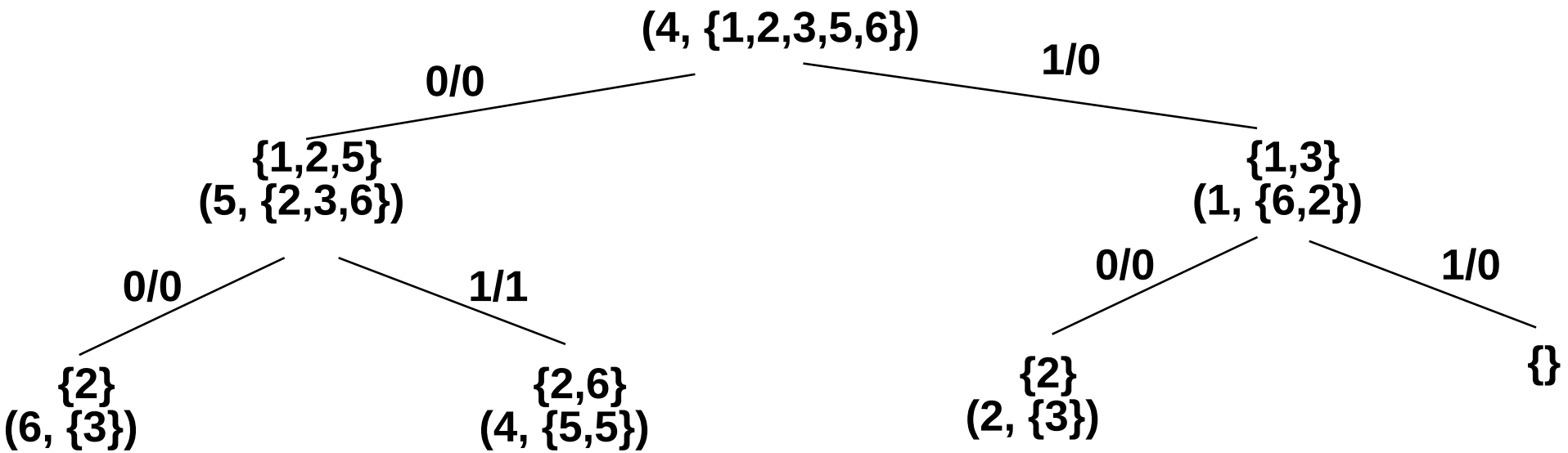
Como calcular

- Usa-se a árvore sucessora
- Cada nó é formado por um estado de interesse e os nós que ainda não foram distinguidos.
- Para cada símbolo de entrada cria-se um filho para um nó.
- No nó filho são colocadas duas “linhas”. A primeira com aqueles estados que têm a mesma saída do estado de interesse e a segunda com os sucessores destes estado e do estado de interesse.

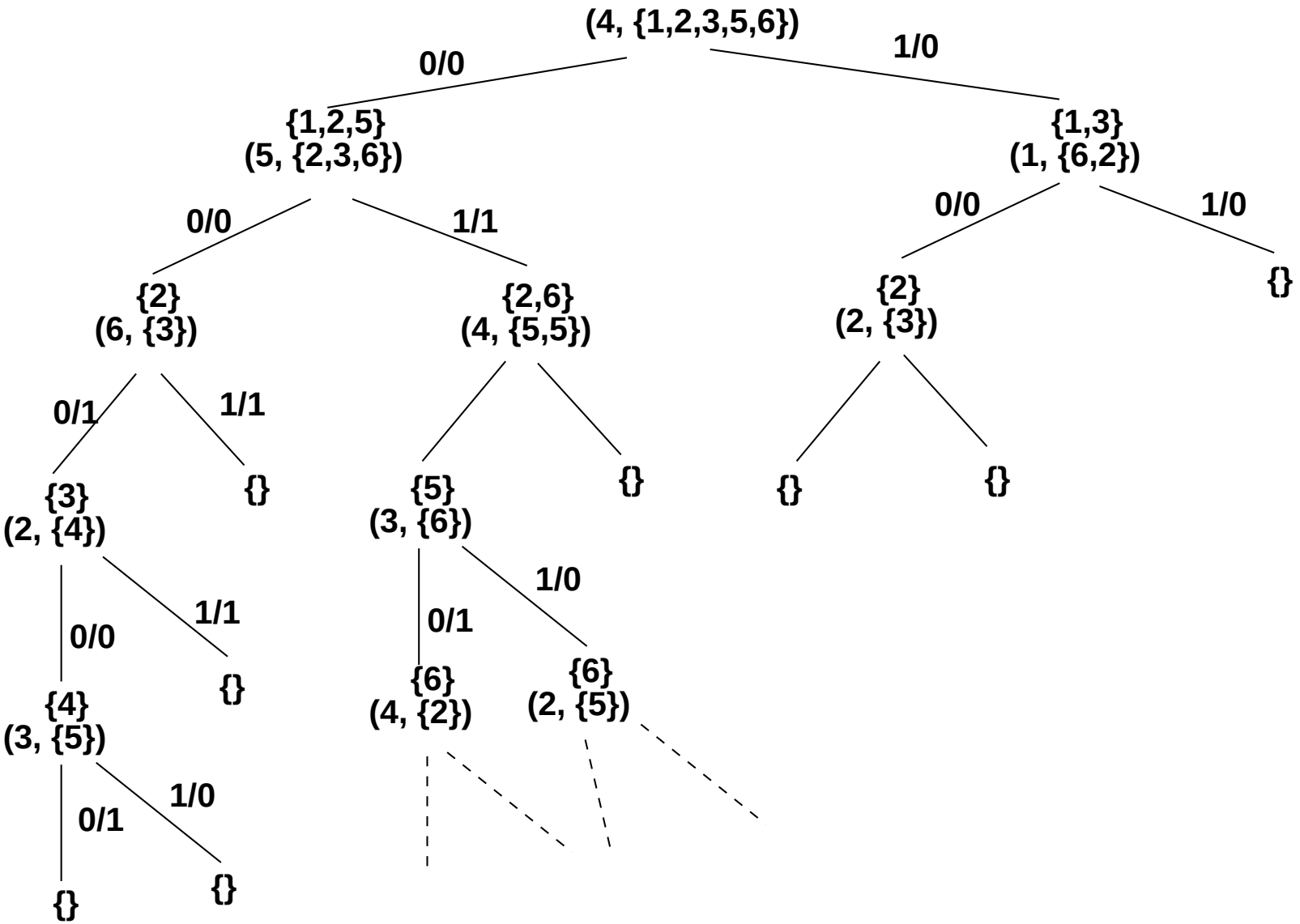
Como calcular

- A raiz possui apenas uma linha.
- A raiz contém o estado do qual se deseja computar a UIO como estado de interesse junto com os demais estados que devem ser distinguidos.
- Quando o conjunto de estados a ser distinguido for vazio, achou-se uma solução.
- Quando o conjunto de estados a ser distinguido inclui o estado de interesse, esse nó não é expandido pois um estado não pode ser diferenciado dele próprio.

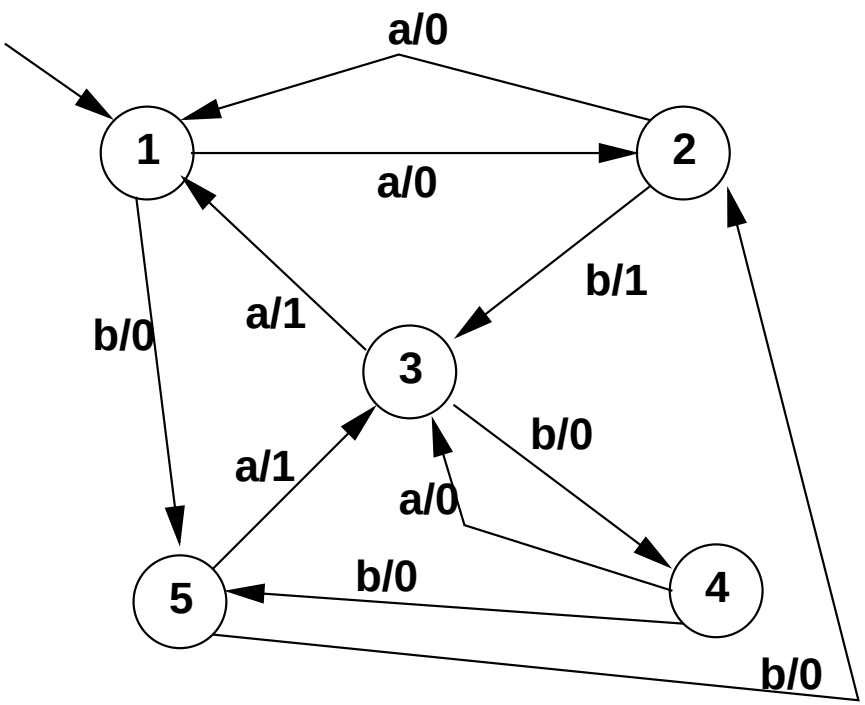
Exemplo

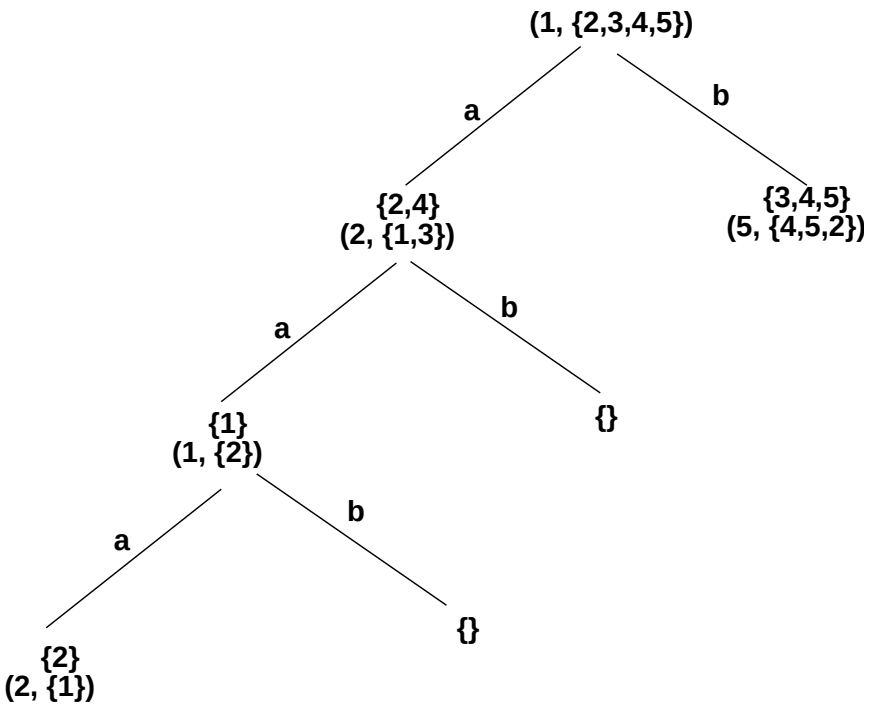


Exemplo


(4, {1,2,3,5,6})
0/0
1/0
{1,2,5}
(5, {2,3,6})
{1,3}
(1, {6,2})
0/0
1/1
0/0
1/0
{2}
(6, {3})
{2,6}
(4, {5,5})
{2}
(2, {3})
{}
0/1
1/1
{3}
(2, {4})
{}
{5}
(3, {6})
{}
{}
{}
1/0
1/1
0/0
0/1
{}
{4}
(3, {5})
{6}
(4, {2})
{6}
(2, {5})
1/0
0/1
{}
{}

Exemplo 2


a/0
1
2
a/0
b/1
b/0
a/1
3
b/0
a/1
a/0
5
b/0
4
b/0


(1, {2,3,4,5})
a
b
{2,4}
(2, {1,3})
{3,4,5}
(5, {4,5,2})
a
b
{1}
(1, {2})
{}
a
b
{2}
(2, {1})
{}

Conjunto de caracterização

- Um conjunto de caracterização é um conjunto de seqüências de entrada que podem, em conjunto, identificar qual é o estado que a MEF se encontra
- Por exemplo, $W = \{\langle 0 \rangle, \langle 1, 1 \rangle\}$ é um conjunto de caracterização para a MEF do nosso exemplo
- Cada seqüência deve ser aplicada a partir do mesmo estado.
- Por isso, para se utilizar um conjunto de caracterização não unitário, deve-se possuir uma forma de restaurar a MEF ao estado original.

Conjunto de caracterização

- Na tabela podemos ver que cada par estados pode ser distinguido pela aplicação das seqüências

	0	1	1
1	0	0	1
2	0	1	1
3	1	0	1
4	0	0	0
5	0	1	0
6	1	1	1

Como calcular

- Deseja-se distinguir cada par de estados através de seqüências de entrada.
- Usa-se o conceito de estados k-equivalentes visto no algoritmo para minimizar uma MEF.
- Computando-se P_1 pode-se estabelecer quais pares de estados podem ser distinguidos por uma seqüência de tamanho 1.
- Computando-se P_2 pode-se estabelecer quais pares de estados podem ser distinguidos por uma seqüência de tamanho 2 e assim por diante.

Exemplo

- $P1 = \{\{1,4\},\{2,5\},\{3\},\{6\}\}$
- Assim, podemos distinguir os pares (1,2), (1,3), (1,5), (1,6), (2,3), (2,4), (2,6), (3,4), (3,5), (3,6), (4,5), (4,6), (5,6) com seqüências de tamanho 1.
- $P_2 = \{\{1\},\{4\},\{2\},\{5\},\{3\},\{6\}\}$
- E os pares (1,4), (2,5) com seqüências de tamanho 2

Métodos de geração

Método TT (Transition Tour) Para uma dada MEF, o transition tour é uma seqüência que parte do estado inicial atravessa todas as transições pelo menos uma vez e retorna ao estado inicial. Permite a detecção de erros de saída, mas não garante erros de transferência, ou seja, erros em que a transição leva a MEF a um estado diferente do qual ela deveria levar.

Método UIO (Unique input/output) Produz uma seqüência de identificação de estado. Não garante cobertura total dos erros, pois a seqüência de entrada leva a uma única saída na especificação correta, mas isso não é garantido para as implementações com erro.

Métodos de geração

Método DS (Seqüência de distinção) Uma seqüência de entrada é uma DS se a seqüência de saída produzida é diferente quando a seqüência de entrada é aplicada a cada estado diferente. A desvantagem desse método é que a DS nem sempre pode ser encontrada para uma dada MEF.

Método W É um conjunto de seqüências que permite distinguir todos os estados e garante detectar os erros estruturais, sempre que a MEF for completa, mínima e fortemente conexa.

Método TT

- O método TT é relativamente simples.
- Ele gera um conjunto de seqüências que passam por todas as transições da MEF.
- Em alguns casos, o conjunto de seqüências é unitário.
- Para algumas MEFs, pode não ser possível calcular uma única seqüência.
- Algoritmos de busca em grafos.

Método DS

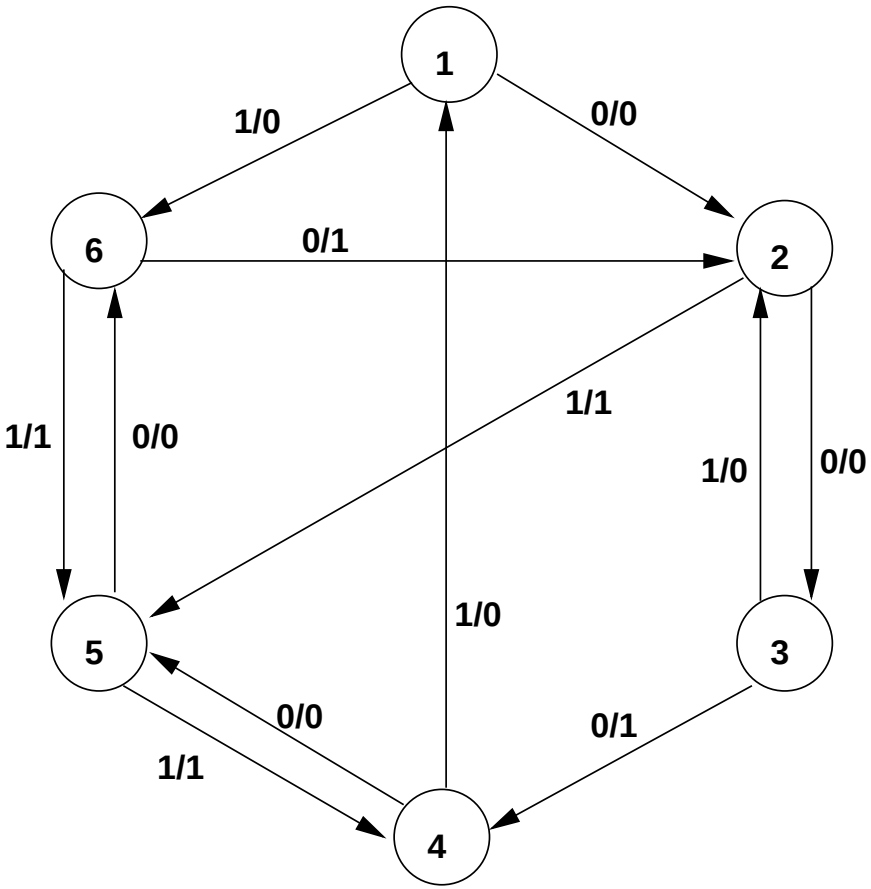
- Utiliza a seqüência DS para geração dos casos de teste.
- Usa também as seqüências de sincronização.
- Por isso, para que possa ser utilizado, o método requer que a MEF possua essas seqüências (para todos os estados).

Como gerar

- Inicialmente calcula-se uma DS (X_d).
- A mais curta possível pois ela será utilizada diversas vezes.
- Constroe-se o grafo X_d.
  - Cada estado aparece como um vértice do grafo X_d.
  - Cada aresta representa a aplicação de X_d no estado da MEF

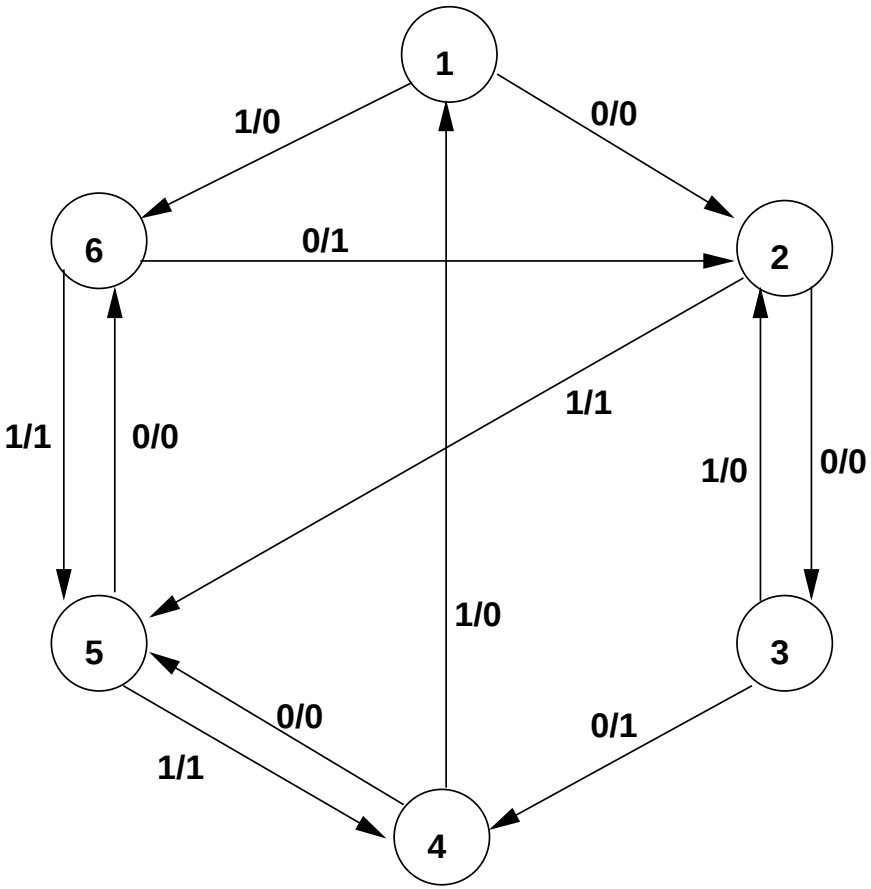
Grafo X_d

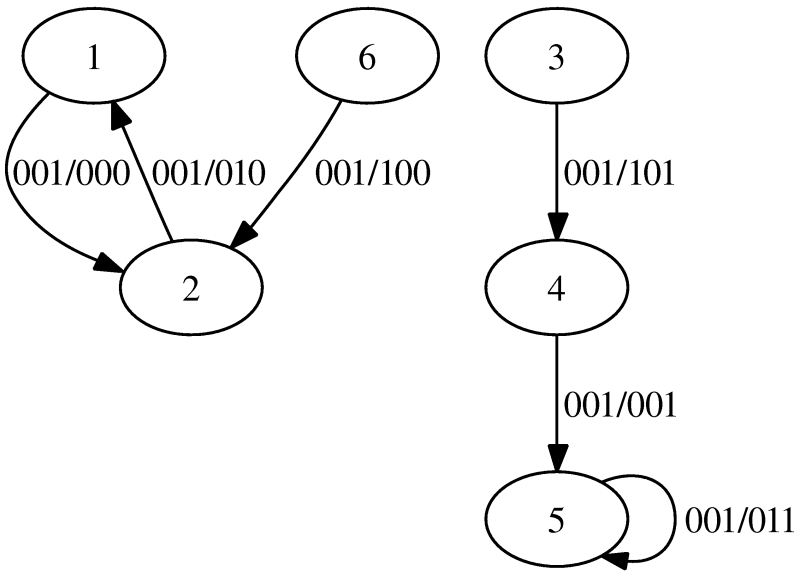
$DS = \langle 0, 0, 1 \rangle$



Grafo X_d

$DS = \langle 0, 0, 1 \rangle$





Continuando

- Gera-se a **seqüência de verificação**
  - seqüência-α que verifica todos os estados da MEF;
  - seqüência-β que verifica todas as transições.
- seqüência-α são geradas percorrendo o grafo X_d

Seqüência α

- As seqüências-α são geradas percorrendo-se o grafo X_d sem que se repitam os vértices.
- Escolhe-se um estado como sendo o inicial.
- Percorre-se o grafo marcando os estados como “reconhecido” conforme são atingidos.
- Quando um estado que já foi reconhecido é novamente visitado, precisa-se escolher outro estado inicial e levar a MEF até ele.
- Para garantir que o último estado foi atingido corretamente, aplica-se mais uma vez a seqüência X_d.
- Ao se escolher um estado inicial, deve-se primeiramente escolher um "estado origem", ou seja, um estado que não é destino de nenhuma aresta no grafo X_d.

No exemplo

- Estados origens: 3 ou 6.

No exemplo

- Estados origens: 3 ou 6.
- Escolhemos arbitrariamente o 3.

No exemplo

- Estados origens: 3 ou 6.
- Escolhemos arbitrariamente o 3.
- Marcado como "reconhecido".

No exemplo

- Estados origens: 3 ou 6.
- Escolhemos arbitrariamente o 3.
- Marcado como "reconhecido".
- Aplica-se X_d, marca 4; X_d marca 5; X_d 5 de novo.

No exemplo

- Estados origens: 3 ou 6.
- Escolhemos arbitrariamente o 3.
- Marcado como "reconhecido".
- Aplica-se X_d, marca 4; X_d marca 5; X_d 5 de novo.
- Antes de terminar, aplica-se X_d novamente.

No exemplo

- Estados origens: 3 ou 6.
- Escolhemos arbitrariamente o 3.
- Marcado como "reconhecido".
- Aplica-se X_d, marca 4; X_d marca 5; X_d 5 de novo.
- Antes de terminar, aplica-se X_d novamente.
- Por enquanto temos $\alpha = \langle 0,0,1,\ 0,0,1,\ 0,0,1,\ 0,0,1 \rangle$

No exemplo

- O novo estado inicial só pode ser 6.

No exemplo

- O novo estado inicial só pode ser 6.
- Para levar a MEF a esse estado podemos usar a seqüência $\langle 0 \rangle$

No exemplo

- O novo estado inicial só pode ser 6.
- Para levar a MEF a esse estado podemos usar a seqüência $\langle 0 \rangle$
- Repete-se o procedimento a partir de 6.

No exemplo

- O novo estado inicial só pode ser 6.
- Para levar a MEF a esse estado podemos usar a seqüência $\langle 0 \rangle$
- Repete-se o procedimento a partir de 6.
- Marca 6; X_d marca 2; X_d marca 1; X_d repete 2.

No exemplo

- O novo estado inicial só pode ser 6.
- Para levar a MEF a esse estado podemos usar a seqüência $\langle 0 \rangle$
- Repete-se o procedimento a partir de 6.
- Marca 6; X_d marca 2; X_d marca 1; X_d repete 2.
- Aplica-se mais uma vez X_d.

No exemplo

- O novo estado inicial só pode ser 6.
- Para levar a MEF a esse estado podemos usar a seqüência $\langle 0 \rangle$
- Repete-se o procedimento a partir de 6.
- Marca 6; X_d marca 2; X_d marca 1; X_d repete 2.
- Aplica-se mais uma vez X_d.
- Resultado: $\alpha =$
  $\langle 0, 0, 1,\ 0, 0, 1,\ 0, 0, 1,\ 0, 0, 1,\ 0,\ 0, 0, 1,\ 0, 0, 1,\ 0, 0, 1,\ 0, 0, 1 \rangle$

No exemplo

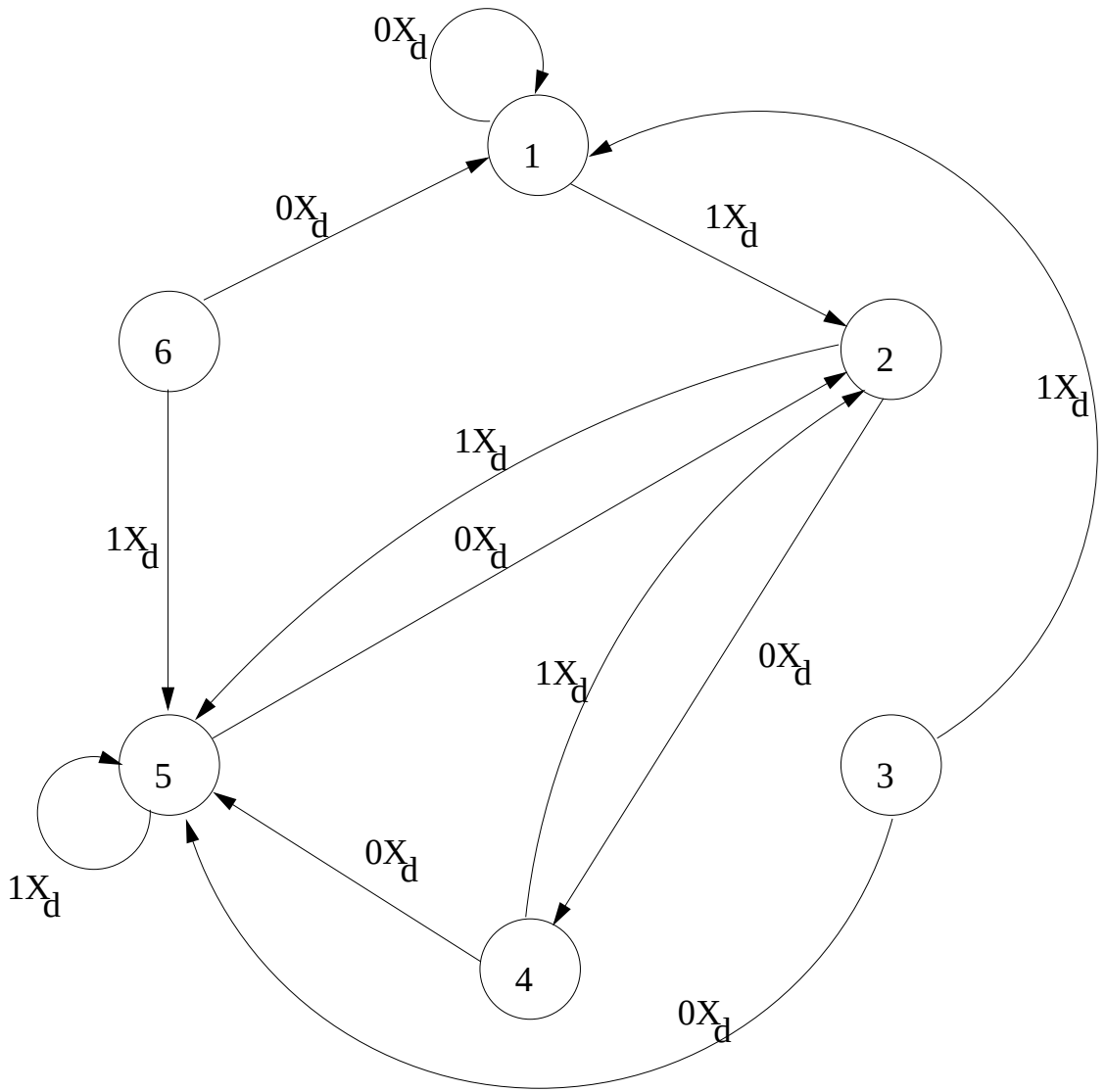
- O novo estado inicial só pode ser 6.
- Para levar a MEF a esse estado podemos usar a seqüência $\langle 0 \rangle$
- Repete-se o procedimento a partir de 6.
- Marca 6; X_d marca 2; X_d marca 1; X_d repete 2.
- Aplica-se mais uma vez X_d.
- Resultado: $\alpha =$ $\langle 0,0,1,\ 0,0,1,\ 0,0,1,\ 0,0,1,\ 0,\ 0,0,1,\ 0,0,1,\ 0,0,1,\ 0,0,1 \rangle$
- No caso desta MEF uma seqüência inicial que a colocaria no estado inicial 3, seria SS(3), ou seja, $\langle 1,1,0,1,1,1,0,0 \rangle$

Seqüência β

- A seqüência-β vai verificar se todas as transições da MEF estão corretas.
- Estratégia semelhante à utilizada para a seqüência-α.
- Em vez da seqüência X_d, utiliza-se uma seqüência da forma $x_i X_d$, para todo $x_i \in X$.
- Estando em um estado s_i que possui uma transição para o estado s_j com a entrada x_i, aplica-se primeiramente a entrada x_i e, em seguida, aplica-se a X_d para verificar (pela saída gerada) se o estado atingido foi realmente o s_j.
- Grafo-β
  - Cada vértice corresponde a um estado
  - Cada aresta corresponde à aplicação de $x_i.X_d$ para cada $x_i \in X$.

Grafo β no exemplo

Grafo β no exemplo



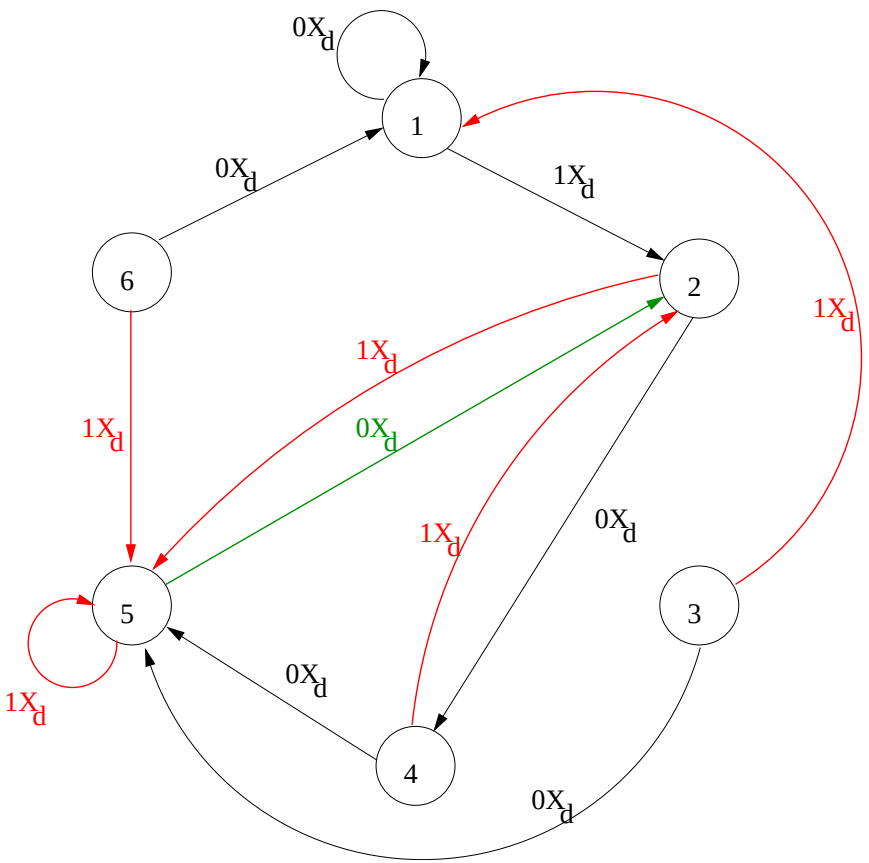
Redução do grafo β

- Na seqüência-α, todos os estados já foram verificados.
- A última transição da aplicação de cada X_d já foi verificada.
- Por exemplo, aplicando a seqüência X_d ($\langle 0, 0, 1 \rangle$) no estado 2, a MEF vai para o estado 1, passando pelos estados 3 e 4.
- Se a transição que vai de 2 a 3 com a entrada 0 e a transição que vai de 3 a 4 com entrada 0 tiverem sido verificadas, não há necessidade de se verificar a transição que vai de 4 a 1 com entrada 1.
- O mesmo vale para as seqüências que servem para ligar os componentes desconexos do grafo-α.

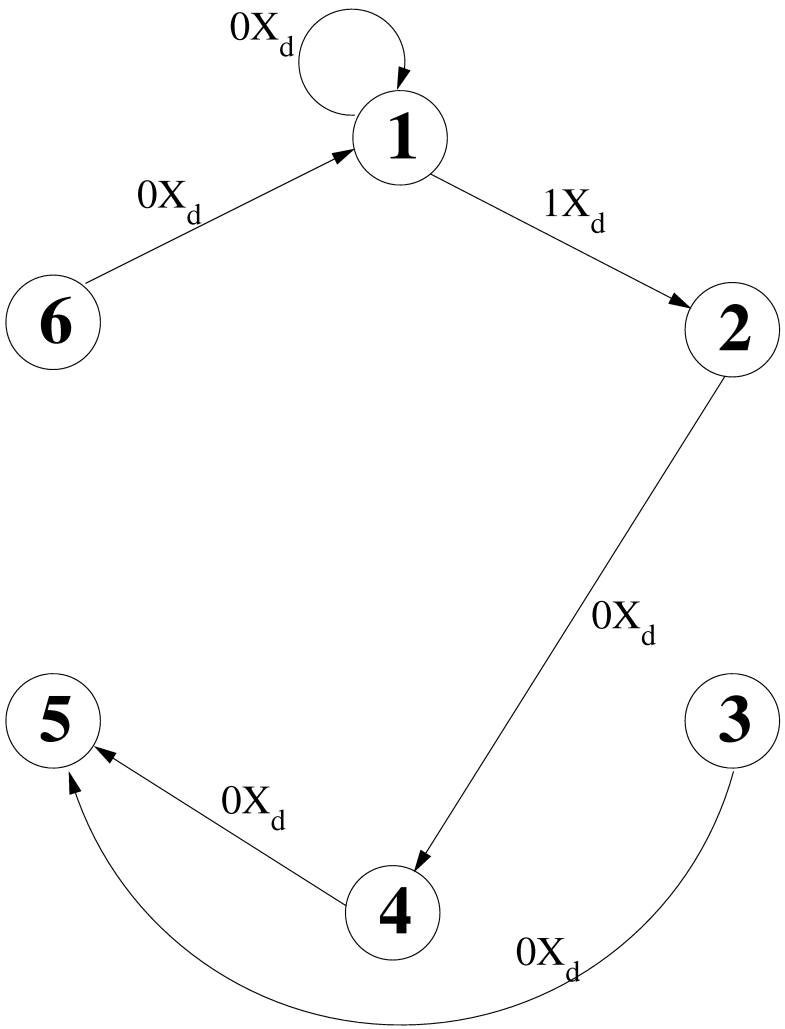
Redução do grafo β

	0	0	1
1	2	3	2
2	3	4	1
3	4	5	4
4	5	6	5
5	6	2	5
6	2	3	2

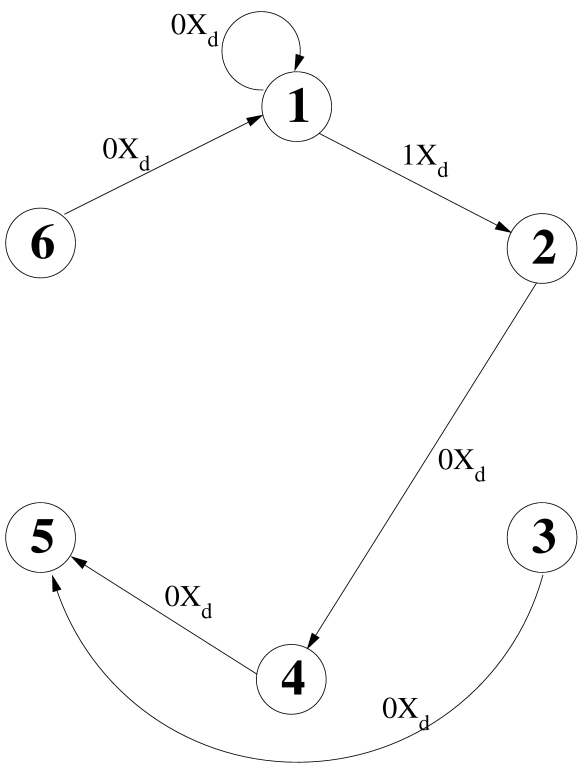
	0
5	6



Grafo reduzido

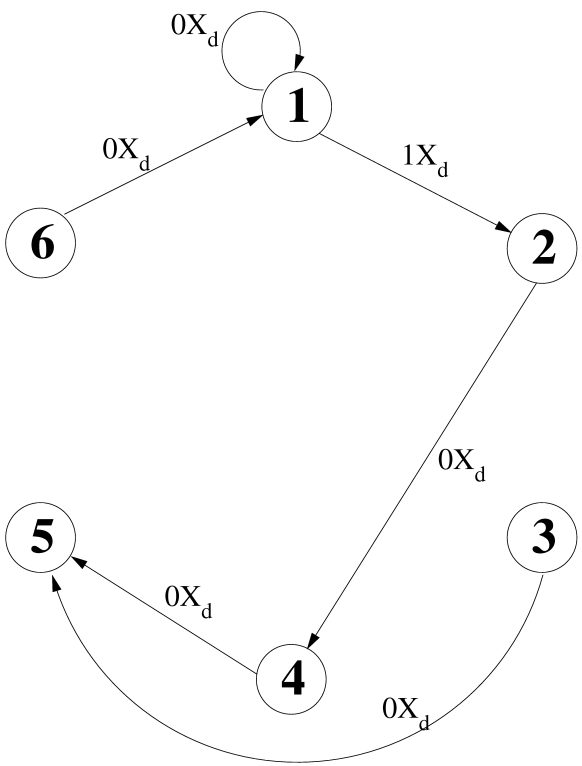


Seqüência β



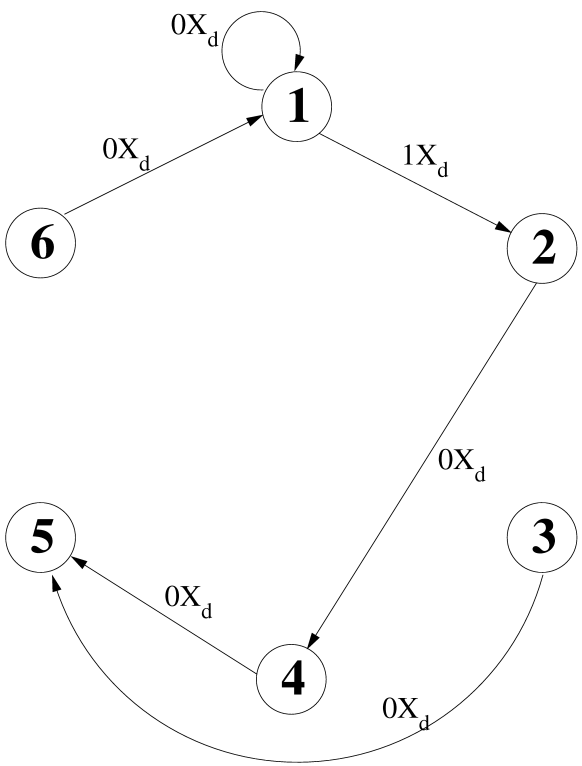
- Cobertura das arestas do grafo-β

Seqüência β



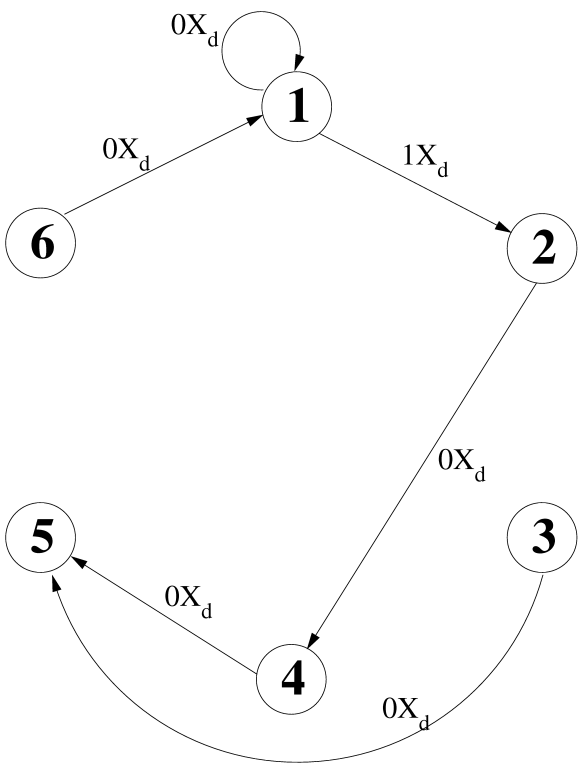
- Cobertura das arestas do grafo-β
- Do estado 1: $\langle 0X_d, 1X_d, 0X_d, 0X_d \rangle$ cobre quatro arestas, e pára no estado 5

Seqüência β



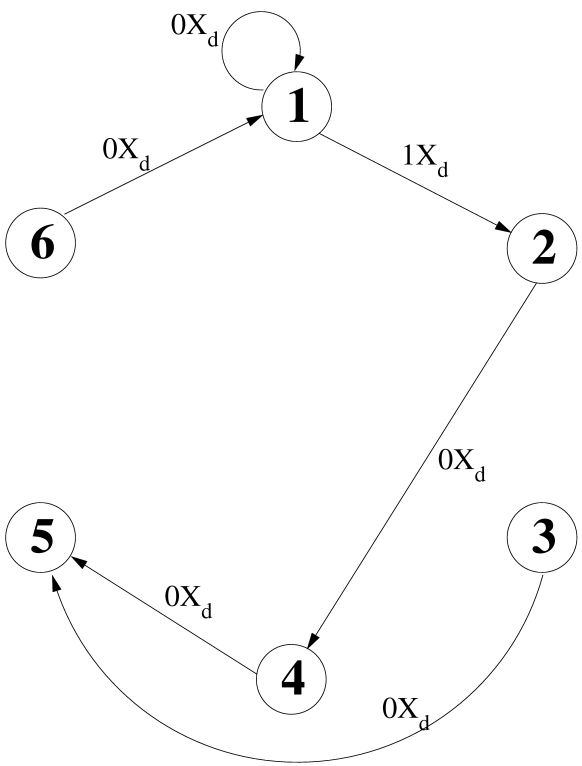
- Cobertura das arestas do grafo-β
- Do estado 1: $\langle 0X_d, 1X_d, 0X_d, 0X_d \rangle$ cobre quatro arestas, e pára no estado 5
- Seqüência $\langle 0 \rangle$ que leva ao estado 6.

Seqüência β



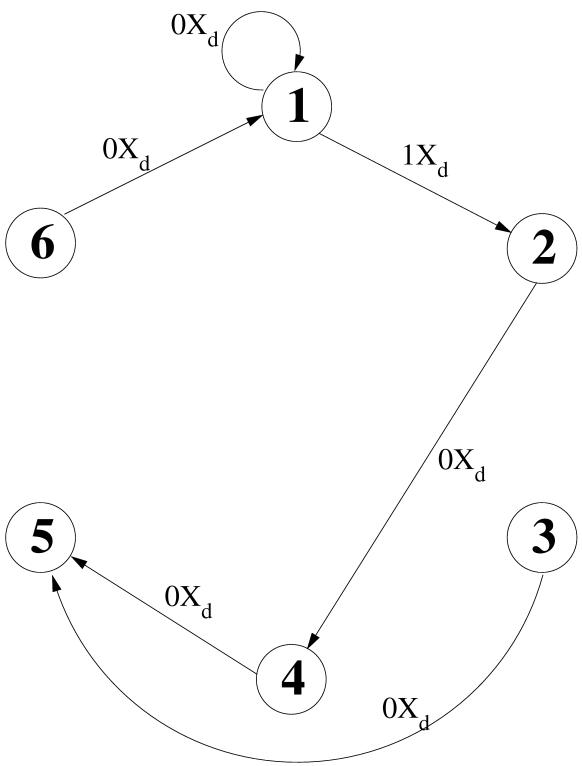
- Cobertura das arestas do grafo-β
- Do estado 1: $\langle 0X_d, 1X_d, 0X_d, 0X_d \rangle$ cobre quatro arestas, e pára no estado 5
- Seqüência $\langle 0 \rangle$ que leva ao estado 6.
- Cobre-se essa aresta com $\langle 0X_d \rangle$, parando no estado 1.

Seqüência β



- Cobertura das arestas do grafo-β
- Do estado 1: $\langle 0X_d, 1X_d, 0X_d, 0X_d \rangle$ cobre quatro arestas, e pára no estado 5
- Seqüência $\langle 0 \rangle$ que leva ao estado 6.
- Cobre-se essa aresta com $\langle 0X_d \rangle$, parando no estado 1.
- Seqüência $\langle 0, 0 \rangle$ para levar a MEF ao estado 3 (que ainda possui uma aresta não coberta).

Seqüência β



- Cobertura das arestas do grafo-β
- Do estado 1: $\langle 0X_d, 1X_d, 0X_d, 0X_d \rangle$ cobre quatro arestas, e pára no estado 5
- Seqüência $\langle 0 \rangle$ que leva ao estado 6.
- Cobre-se essa aresta com $\langle 0X_d \rangle$, parando no estado 1.
- Seqüência $\langle 0, 0 \rangle$ para levar a MEF ao estado 3 (que ainda possui uma aresta não coberta).
- Com a seqüência $\langle 0X_d \rangle$, cobre-se a última aresta.

Seqüência β

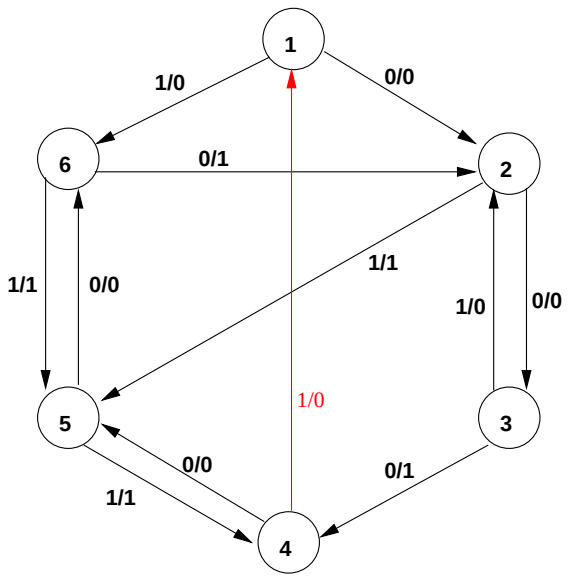
$$\langle 0,0,0,1,\quad 1,0,0,1,\quad 0,0,0,1,\quad 0,0,0,1\ \ 0,\quad 0,0,0,1\ \ 0,0,\quad 0,0,0,1\rangle$$

Método UIO

- Utiliza as seqüências UIO para testar a máquina.
- Para cada transição de um estado s_i para $s_j = f_s(s_i, x)$, com algum x, define-se uma seqüência que leva a máquina do estado inicial até s_i, aplica-se o símbolo de entrada e, em seguida, aplica-se a seqüência UIO do estado que deveria ser atingido.
- Ou seja: $P(s_i) \cdot x \cdot UIO(s_j)$
- $P(s_i)$ é uma seqüência que leva a MEF do estado inicial ao estado s_i

No exemplo

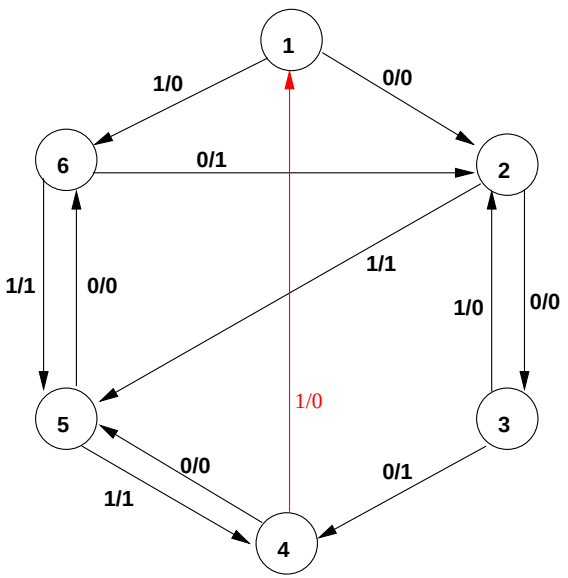
P(1)	$\langle\rangle$
P(2)	$\langle 0\rangle$
P(3)	$\langle 0,0\rangle$
P(4)	$\langle 0,0,0\rangle$
P(5)	$\langle 0,1\rangle$
P(6)	$\langle 1\rangle$



- Transição do estado 4 para o estado 1 com entrada 1.

No exemplo

P(1)	$\langle\rangle$
P(2)	$\langle 0\rangle$
P(3)	$\langle 0,0\rangle$
P(4)	$\langle 0,0,0\rangle$
P(5)	$\langle 0,1\rangle$
P(6)	$\langle 1\rangle$



- Transição do estado 4 para o estado 1 com entrada 1.
- A seqüência é dada por: $P(4) \cdot 1 \cdot UIO(1) = \langle 0,0,0,\ 1,\ 1,0\rangle$

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	$\langle 0 \rangle . UIO(2)$
1	1	

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	$\langle 0 \rangle . UIO(2)$
1	1	$\langle 1 \rangle . UIO(6)$
2	0	

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	$\langle 0 \rangle . UIO(2)$
1	1	$\langle 1 \rangle . UIO(6)$
2	0	$\langle 0, 0 \rangle . UIO(3)$
2	1	

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	$\langle 0 \rangle . UIO(2)$
1	1	$\langle 1 \rangle . UIO(6)$
2	0	$\langle 0, 0 \rangle . UIO(3)$
2	1	$\langle 0, 1 \rangle . UIO(5)$
3	0	

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	$\langle 0 \rangle . UIO(2)$
1	1	$\langle 1 \rangle . UIO(6)$
2	0	$\langle 0, 0 \rangle . UIO(3)$
2	1	$\langle 0, 1 \rangle . UIO(5)$
3	0	$\langle 0, 0, 0 \rangle . UIO(4)$
3	1	

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	$\langle 0 \rangle . UIO(2)$
1	1	$\langle 1 \rangle . UIO(6)$
2	0	$\langle 0, 0 \rangle . UIO(3)$
2	1	$\langle 0, 1 \rangle . UIO(5)$
3	0	$\langle 0, 0, 0 \rangle . UIO(4)$
3	1	$\langle 0, 0, 1 \rangle . UIO(2)$
4	0	

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	$\langle 0 \rangle . UIO(2)$
1	1	$\langle 1 \rangle . UIO(6)$
2	0	$\langle 0, 0 \rangle . UIO(3)$
2	1	$\langle 0, 1 \rangle . UIO(5)$
3	0	$\langle 0, 0, 0 \rangle . UIO(4)$
3	1	$\langle 0, 0, 1 \rangle . UIO(2)$
4	0	$\langle 0, 0, 0, 0 \rangle . UIO(5)$
4	1	

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	$\langle 0 \rangle . UIO(2)$
1	1	$\langle 1 \rangle . UIO(6)$
2	0	$\langle 0, 0 \rangle . UIO(3)$
2	1	$\langle 0, 1 \rangle . UIO(5)$
3	0	$\langle 0, 0, 0 \rangle . UIO(4)$
3	1	$\langle 0, 0, 1 \rangle . UIO(2)$
4	0	$\langle 0, 0, 0, 0 \rangle . UIO(5)$
4	1	$\langle 0, 0, 0, 1 \rangle . UIO(1)$
5	0	

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	$\langle 0 \rangle . UIO(2)$
1	1	$\langle 1 \rangle . UIO(6)$
2	0	$\langle 0, 0 \rangle . UIO(3)$
2	1	$\langle 0, 1 \rangle . UIO(5)$
3	0	$\langle 0, 0, 0 \rangle . UIO(4)$
3	1	$\langle 0, 0, 1 \rangle . UIO(2)$
4	0	$\langle 0, 0, 0, 0 \rangle . UIO(5)$
4	1	$\langle 0, 0, 0, 1 \rangle . UIO(1)$
5	0	$\langle 0, 1, 0 \rangle . UIO(6)$
5	1	

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	$\langle 0 \rangle . UIO(2)$
1	1	$\langle 1 \rangle . UIO(6)$
2	0	$\langle 0, 0 \rangle . UIO(3)$
2	1	$\langle 0, 1 \rangle . UIO(5)$
3	0	$\langle 0, 0, 0 \rangle . UIO(4)$
3	1	$\langle 0, 0, 1 \rangle . UIO(2)$
4	0	$\langle 0, 0, 0, 0 \rangle . UIO(5)$
4	1	$\langle 0, 0, 0, 1 \rangle . UIO(1)$
5	0	$\langle 0, 1, 0 \rangle . UIO(6)$
5	1	$\langle 0, 1, 1 \rangle . UIO(4)$
6	0	

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	$\langle 0 \rangle . UIO(2)$
1	1	$\langle 1 \rangle . UIO(6)$
2	0	$\langle 0, 0 \rangle . UIO(3)$
2	1	$\langle 0, 1 \rangle . UIO(5)$
3	0	$\langle 0, 0, 0 \rangle . UIO(4)$
3	1	$\langle 0, 0, 1 \rangle . UIO(2)$
4	0	$\langle 0, 0, 0, 0 \rangle . UIO(5)$
4	1	$\langle 0, 0, 0, 1 \rangle . UIO(1)$
5	0	$\langle 0, 1, 0 \rangle . UIO(6)$
5	1	$\langle 0, 1, 1 \rangle . UIO(4)$
6	0	$\langle 1, 0 \rangle . UIO(2)$
6	1	

No exemplo, demais transições

Estado	Entrada	Seqüência
1	0	$\langle 0 \rangle . UIO(2)$
1	1	$\langle 1 \rangle . UIO(6)$
2	0	$\langle 0, 0 \rangle . UIO(3)$
2	1	$\langle 0, 1 \rangle . UIO(5)$
3	0	$\langle 0, 0, 0 \rangle . UIO(4)$
3	1	$\langle 0, 0, 1 \rangle . UIO(2)$
4	0	$\langle 0, 0, 0, 0 \rangle . UIO(5)$
4	1	$\langle 0, 0, 0, 1 \rangle . UIO(1)$
5	0	$\langle 0, 1, 0 \rangle . UIO(6)$
5	1	$\langle 0, 1, 1 \rangle . UIO(4)$
6	0	$\langle 1, 0 \rangle . UIO(2)$
6	1	$\langle 1, 1 \rangle . UIO(5)$

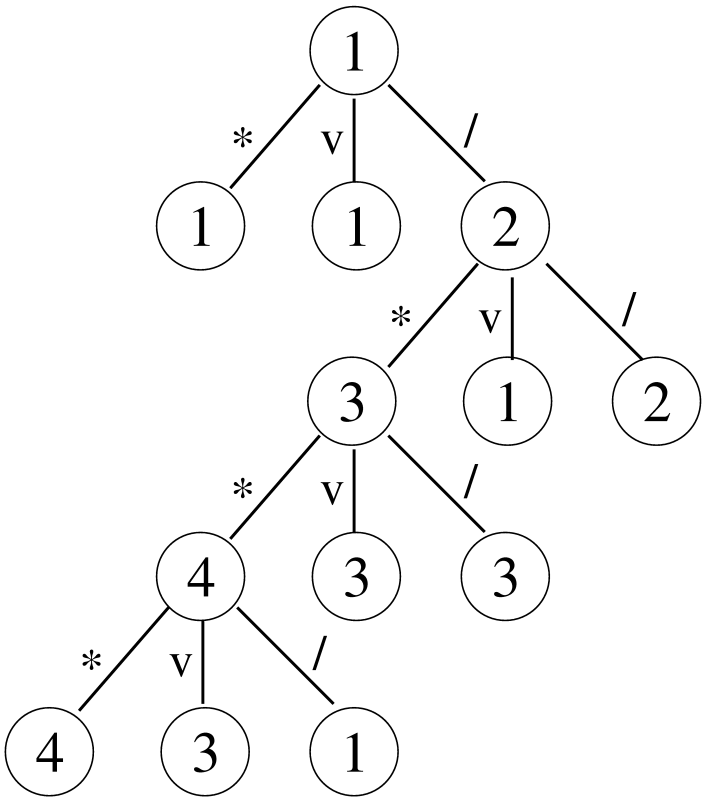
Método W

- O método W utiliza o conjunto de caracterização para verificar se todos os estados e transições estão corretamente implementados.
- Cada seqüência é composta por duas subseqüências.
- A primeira subseqüência (denominada, seqüência-T) é formada a partir da árvore de teste, e serve para garantir que cada transição foi atingida.
- A segunda subseqüência (denominada, seqüência-P) é derivada a partir do conjunto de caracterização W, e é usada para verificar se o estado atingido após a transição está correto.

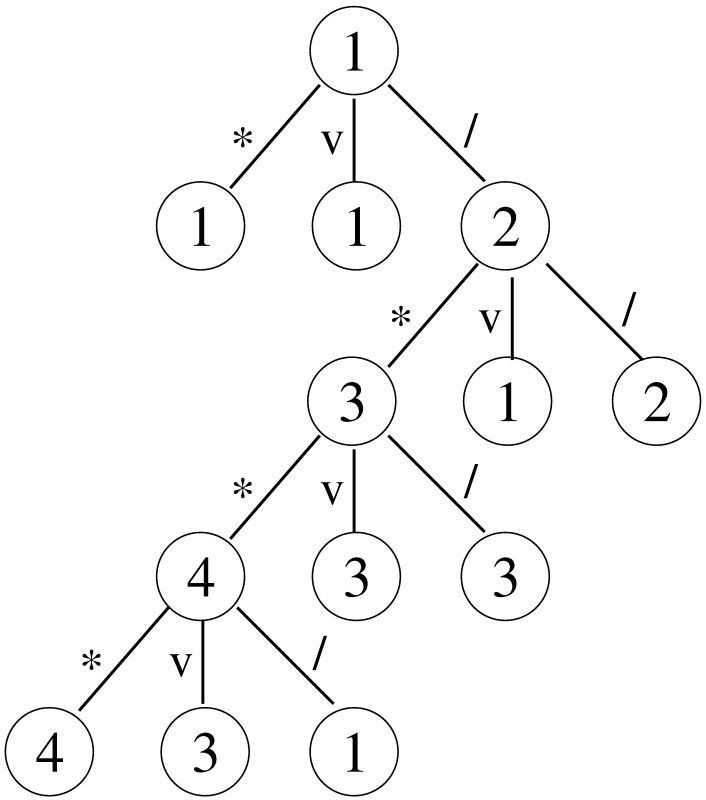
Árvore de teste

- Os nós correspondem aos estados.
- As arestas correspondem às transições.
- A raiz da árvore é o estado inicial.
- Cada transição aparece exatamente uma vez na árvore.
- As seqüências-T são todos os caminhos parciais da árvore de teste.

Árvore de teste



Árvore de teste



$$T = \{\langle\rangle, \langle *\rangle, \langle v\rangle, \langle /\rangle, \langle /, *\rangle, \langle /, v\rangle, \langle /, /\rangle, \langle /, *, *\rangle, \langle /, *, v\rangle, \langle /, *, /\rangle, \langle /, *, *, *\rangle, \langle /, *, *, v\rangle, \langle /, *, *, /\rangle\}$$

Seqüências P

- É necessário fazer uma estimativa da quantidade de estados que a implementação pode ter.
- É essencial para o método, pois, garante-se que, se a implementação possuir no máximo essa quantidade de estados e passar por todas as seqüências de teste com o resultado correto, ela estará correta.
- Por outro lado, quanto maior for esse valor, maior o número de seqüências necessárias.
- Seja g a diferença entre a estimativa e o número de estados da MEF, $P = (\bigcup_{i=0}^{g} X^i) \cdot W$
- Se o número de estados da MEF é 5 e a estimativa é 7, tem-se que $g = 2$ e $P = (\langle\rangle \cup X \cup X \cdot X) \cdot W$

No exemplo

- Conjunto de caracterização $W = \{\langle / \rangle, \langle * \rangle\}$.

No exemplo

- Conjunto de caracterização $W = \{\langle / \rangle, \langle * \rangle\}$.
- Considerando $g = 0$, temos $P = W$

No exemplo

- Conjunto de caracterização $W = \{\langle / \rangle, \langle * \rangle\}$.
- Considerando $g = 0$, temos $P = W$
- O conjunto de seqüências de teste é formado pela concatenação de T e P.

No exemplo

- Conjunto de caracterização $W = \{\langle / \rangle, \langle * \rangle\}$.
- Considerando $g = 0$, temos $P = W$
- O conjunto de seqüências de teste é formado pela concatenação de T e P.
- Ou seja:

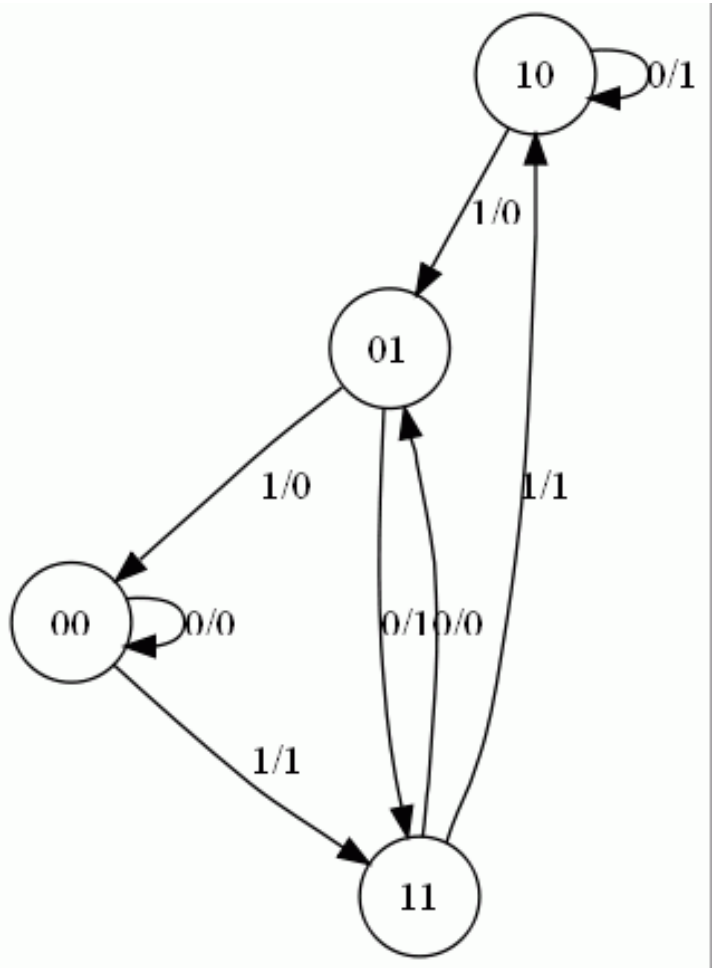
$\langle / \rangle$	$\langle * \rangle$	$\langle */ \rangle$	$\langle ** \rangle$
$\langle \mathrm{v}/ \rangle$	$\langle \mathrm{v}* \rangle$	$\langle // \rangle$	$\langle /* \rangle$
$\langle / * / \rangle$	$\langle / * * \rangle$	$\langle /\mathrm{v}/ \rangle$	$\langle /\mathrm{v}* \rangle$
$\langle /// \rangle$	$\langle //* \rangle$	$\langle / * */ \rangle$	$\langle / * ** \rangle$
$\langle / * \mathrm{v}/ \rangle$	$\langle / * \mathrm{v}* \rangle$	$\langle / * // \rangle$	$\langle / * /* \rangle$
$\langle / * * * / \rangle$	$\langle / * * * * \rangle$	$\langle / * *\mathrm{v}/ \rangle$	$\langle / * *\mathrm{v}* \rangle$
$\langle / * *// \rangle$	$\langle / * */* \rangle$		

Restrições dos métodos

	TT	UIO	DS	W
Minimalidade		✓	✓	✓
Completamente especificada		✓	✓	✓
Fortemente conectada		✓	✓	
Determinismo	✓	✓	✓	✓
Seqüências de sincronização			✓	
Seqüências de distinção			✓	
Seqüências únicas de entrada/saída		✓		
Conjunto de caracterização				✓

Observações importantes

- O método TT é o menos exigente.
- Quanto à aplicabilidade, o método W é o que pode ser aplicado a qualquer MEF, uma vez que toda MEF minimal possui um conjunto de caracterização.
- Para MEFs completamente especificadas, os métodos UIO, DS e W são capazes de detectar falhas nas saídas de todas as transições. Dessa forma, eles são capazes de detectar erros de transferência (próximo estado incorreto), de operação (símbolo de entrada ou saída da transição incorretos) e de estados em excesso ou falta
- O método UIO produz as seqüências de teste mais curtas que os métodos DS e W, ao passo que o método W produz as seqüências mais longas.

Exercício

Compute seqüências de teste usando os quatro métodos vistos na aula. Considere 00 o estado inicial. Para o método W, use g = 1.