Verificação, Validação & Teste de Software

Jean Phelipe de Oliveira Lima

jpdol.eng16@uea.edu.br

/jpdol

Teste Baseado em Modelo

- Especificação:
- Linguagem Natural X Linguagem Formal

  - Natural:
    - Mais compreensiva;
    - Mais ambígua;

  - Formal:
    - Mais complexa;
    - Mais objetiva.

Máquina de Estados Finitos

- Uma MEF, ou Máquina de Estados Finitos, é uma máquina hipotética composta por estados e transições.
- A cada instante, a máquina pode estar em apenas um de seus estados. Em resposta a um evento de entrada, a máquina gera um evento de saída e muda de estado.
- Tanto o evento de saída gerado quanto o novo estado são definidos unicamente em função do estado atual e do evento de entrada.
- Uma Máquina de Estados Finitos pode ser representada tanto por um diagrama de transição de estados quanto por uma tabela de transição.
- Em um diagrama de transição de estados, os estados são representados por vértices, e as transições por arcos direcionados entre estados.
- Cada arco é rotulado com a entrada que gera a transição e a saída que é produzida, usando o formato entrada: saída (ou entrada / saída). Em uma tabela de transição, os estados são representados por linhas, e as entradas, por colunas.

Máquina de Estados Finitos

- **Definição 1:** Formalmente, uma Máquina de Estados Finitos é uma tupla $M = (X, Z, S, s_0 , f_z , f_s)$, sendo que:

  - X é um conjunto finito não-vazio de símbolos de entrada;
  - Z é um conjunto finito de símbolos de saída;
  - S é um conjunto finito não-vazio de estados;
  - $s_0 \in S$ é o estado inicial;
  - $f_z : (S \times X) \rightarrow Z$ é a função de saída; e
  - $f_s : (S \times X) \rightarrow S$ é a função de próximo estado.

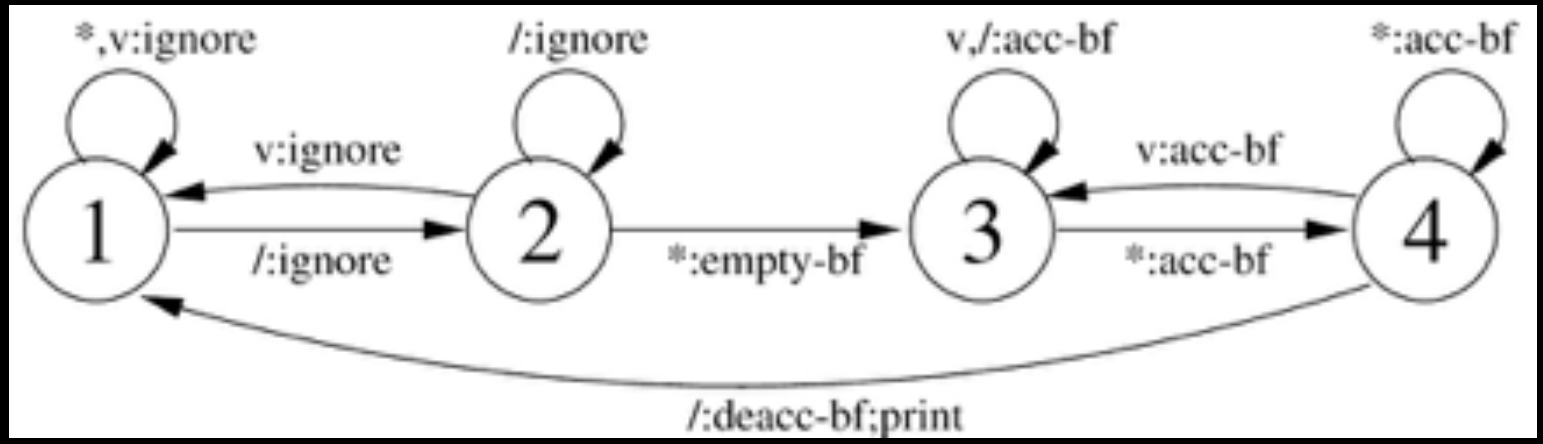
Exemplo – Comment Printer

- Ideia: printar todo comentário de um código. Um comentário é representado entre as strings ”/*” e ”*/”.

- Símbolos: ”v”; ”*”; ”/”;
- Ações:
  - ignore → Ação Nula
  - empty-bf → buffer = []
  - acc-bf → buffer.concat()
  - deacc-bf → truncamento a direita
  - print → Imprime o conteúdo do buffer

Exemplo – Comment Printer

- Diagrama MEF:



1: Estado Inicial

Exemplo – Comment Printer

- Tabela de transição de Estados:

EstadoEntrada	*	/	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

Propriedades de MEF

- **Propriedade 1**: Diz-se que a MEF é completamente especificada se ela trata todas as 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.
- **Propriedade 2**: Uma MEF é fortemente conectada se, para cada par de estados (s_i, s_j) , existir um caminho por transições que vai de s_i a s_j. Ela é inicialmente conectada se, a partir do estado inicial, for possível atingir todos os demais estados da MEF.

Propriedades de MEF

- **Propriedade 3**: Uma MEF é minimal (ou reduzida) se na MEF não existem quaisquer dois estados equivalentes. Dois estados são equivalentes se produzem as mesmas sequências de saídas para as mesmas sequências de entradas.

- **Propriedade 4**: Uma MEF é determinística quando, para qualquer estado e para dada entrada, a MEF permite uma única transição para um próximo estado. Caso contrário, é não determinística.

Sequências Básicas

- Os métodos de geração de casos de teste são fortemente baseados em algumas sequências básicas, as quais são utilizadas para se obter um resultado parcial importante.

Sequências de Sincronização

- Uma sequência de sincronização para um estado s (expressa por SS(s)) é uma sequência de entradas que leva a MEF ao estado s, independentemente do estado original da MEF.

Exemplo – Comment Printer

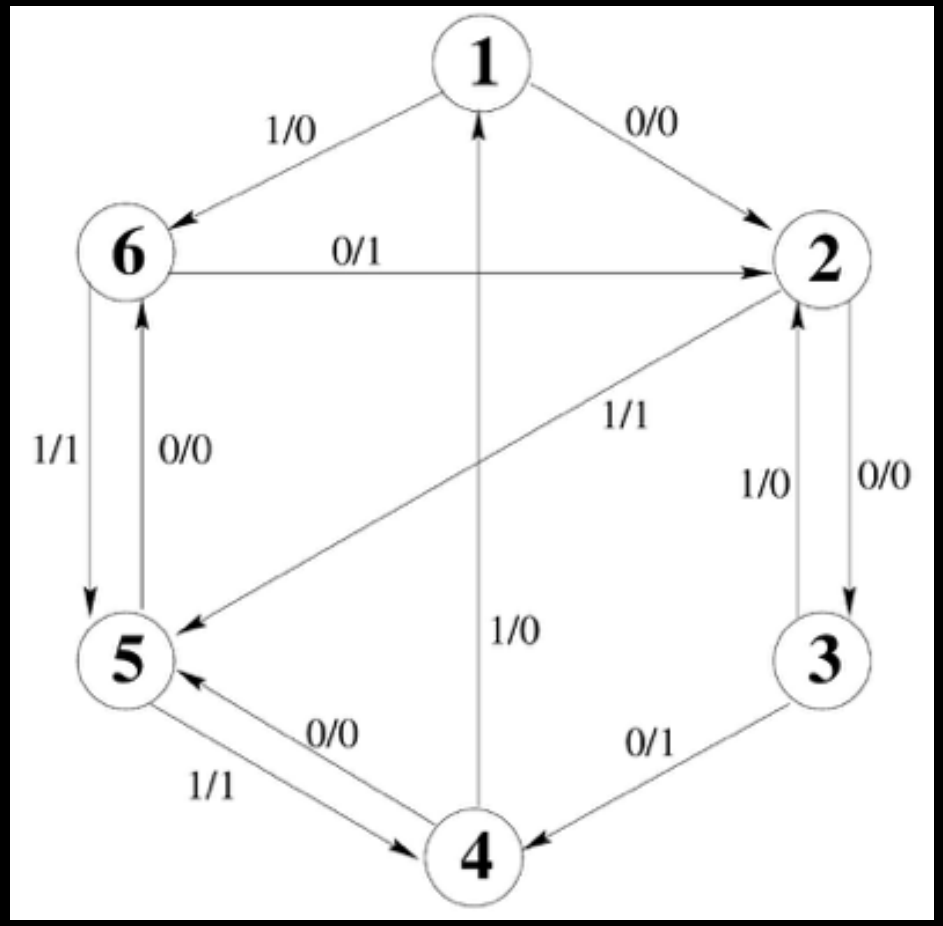
- SS(s) = [* , * , /, v]

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

Sequências de Distinção

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

Exemplo

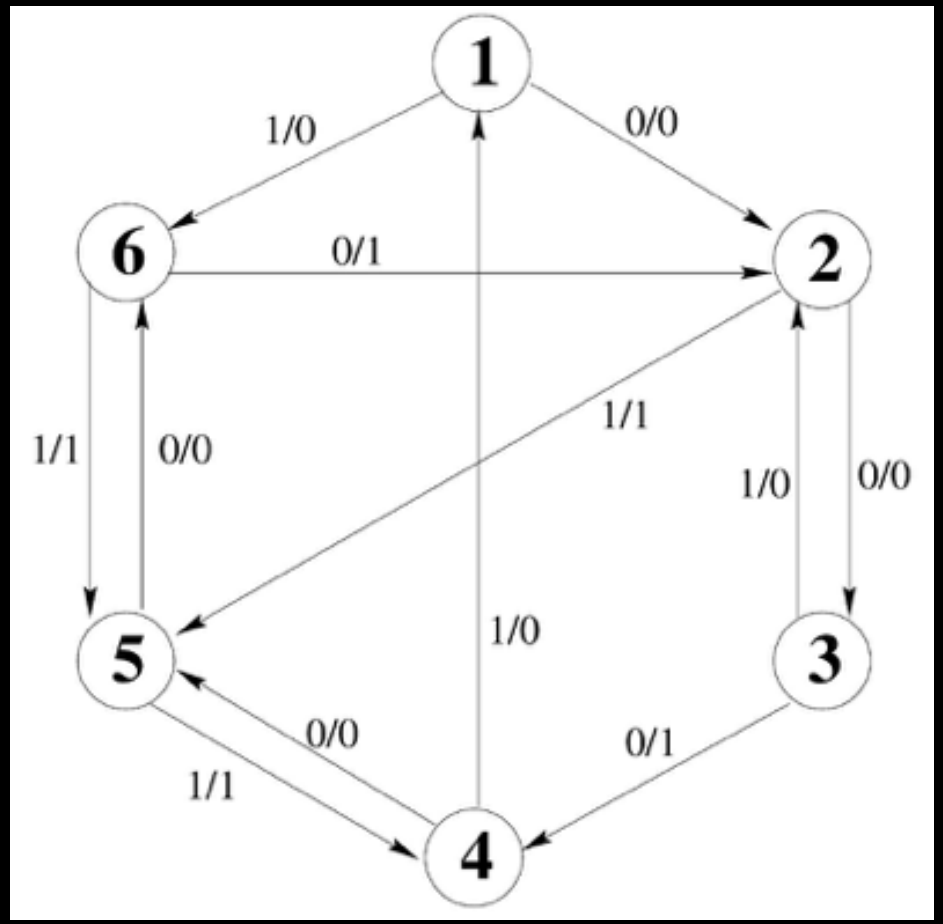


	0	0	1
1	0	0	0
2	0	1	0
3	1	0	1
4	0		0 1
5	0	1	1
6	1	0	0

Sequências UIO

- Nem todas as MEFs possuem DSs. Entretanto, mesmo nessas situações, pode-se fazer a distinção entre um estado e os demais. Para isso, usa-se uma Sequência Única de Entrada e Saída (UIO), a qual é utilizada para verificar se a MEF está em um estado particular.
- Para cada estado da MEF, pode-se ter uma UIO distinta. Isso é verdade apenas se forem consideradas tanto as entradas como as saídas. A sequência de entradas da UIO de um estado pode ser igual à sequência de entradas da UIO de outro estado. Assim, pode-se concluir que se uma MEF possui uma DS, ela também possui UIOs para todos os seus estados. Contudo, em geral, podem-se obter UIOs mais curtas.

Exemplo



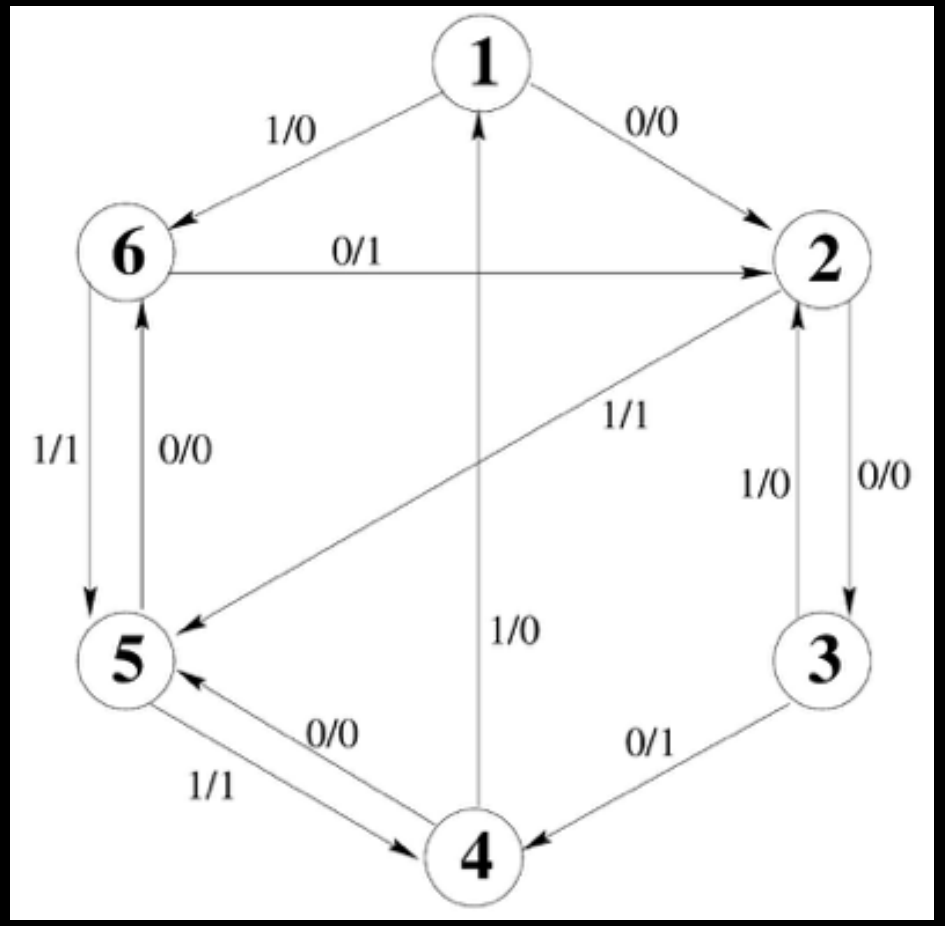
UIO(4) = {1/0;1/0}

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

Conjunto de Caracterização

- Um conjunto de caracterização é um conjunto de sequências de entrada que podem, juntas, identificar qual era o estado original.

Exemplo



W = {[0]; [1,1]}

	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

Geração de Sequências de Teste

- Uma sequência de teste consiste em uma sequência de símbolos de entrada derivada da MEF que são submetidos à implementação correspondente para verificar a conformidade da saída gerada pela implementação com a saída especificada na sequência de teste.

- Os métodos mais clássicos para a geração de sequências de teste a partir de MEFs são apresentados a seguir.

Método TT (Transition Tour)

- Para dada MEF, o transition tour é uma sequência 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 sequência de identificação de estado. Não garante cobertura total dos erros, pois a sequência de entrada leva a uma única saída na especificação correta, mas isso não é garantido para as implementações com erro.

Método DS (Sequência de Distinção)

- Uma sequência de entrada é uma DS se a sequência de saída produzida é diferente quando a sequência de entrada é aplicada a cada estado diferente. A desvantagem desse método é que a DS nem sempre pode ser encontrada para dada MEF.

Método W

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

Verificação, Validação & Teste de Software

Jean Phelipe de Oliveira Lima

jpdol.eng16@uea.edu.br

/jpdol