Teste Baseado em Modelos

PPGCC12-Teste de Software

Reginaldo Ré

Aula 07

Introdução

Introdução

- Uma especificação de requisitos pode ser escrita de diversas formas:
  - Linguagem natural, métodos formais, modelos…;
  - Todas possuem maior ou menor risco de imprecisão;
- Linguagem natural, per se, não implica em problemas:
  - Maior risco de imprecisão e incosistências.

Introdução

Modelos

- Modelos podem ser úteis pelo aumento da compreensão do sistema modelado:
  - Além da possibilidade de análise algorítmica da especificação;
- Esforço de busca pelo “o que o sistema deve fazer”:
  - Ou seja, ao invés de responder se o resultado está correto, estabelecer o que é o correto;
- Alguns modelos permitem execução (ou simulação):
  - Pode ser usado como oráculo de teste;
  - A questão que surge é: em caso de um resultado divergente, quem está correto o SUT ou o modelo?
    - Em outras palavras, como saber se o modelo está correto?
- Modelos (executáveis) podem ser testados assim como o SUT:
  - Estando correto podem ser oráculos e gerados de scripts e cenários de teste;

Teste baseado em Modelos

Teste baseado em Modelos

- Durante o teste, como estabelecer quais itens serão testados?
  - e como?
  - Uma especificação clara torna-se importante;
- Testadores sempre constroem modelos:
  - Ainda que informais;
- Modelos informais (ou na cabeça do testador) dificulta os testes;

Passos da Aplicação do Teste Baseado em Modelos

1. Modelagem: um modelo que especifica o SUT;
2. Geração de Casos de Teste: sequências de teste (abstratos) são geradas para testar o próprio modelo e o SUT:
- Existem vários métodos de geração, que dependem do tipo de modelo usado.
3. Concretização: codificação dos testes abstratos para executar no SUT:
- Existem várias estratégias de concretização;
4. Exdcução dos testes: casos de teste concretizados são executados no SUT.

Máquinas de transição de estados

- Alguns técnicas usam modelos baseados em grafos;
- Nós representam “estados” (ou configurações);
- Arestas representam transições entre os estados;
- São comumente chamadas de “máquinas de transição de estados”;
- Algumas são hierarquicas:
  - Permitem a decomposição de um estado complexo em sub-estados.

Máquinas de Estados Finitos (MEFs)

- São um tipo de máquina de transição de estados;
- Em uma MEF, estados e transições são explicitamente declarados:
  - Problema da explosão de estados
- MEF estendida possuem:
  - Variáveis de contexto;
  - Transições parametrizadas;

Detalhes de uma MEF

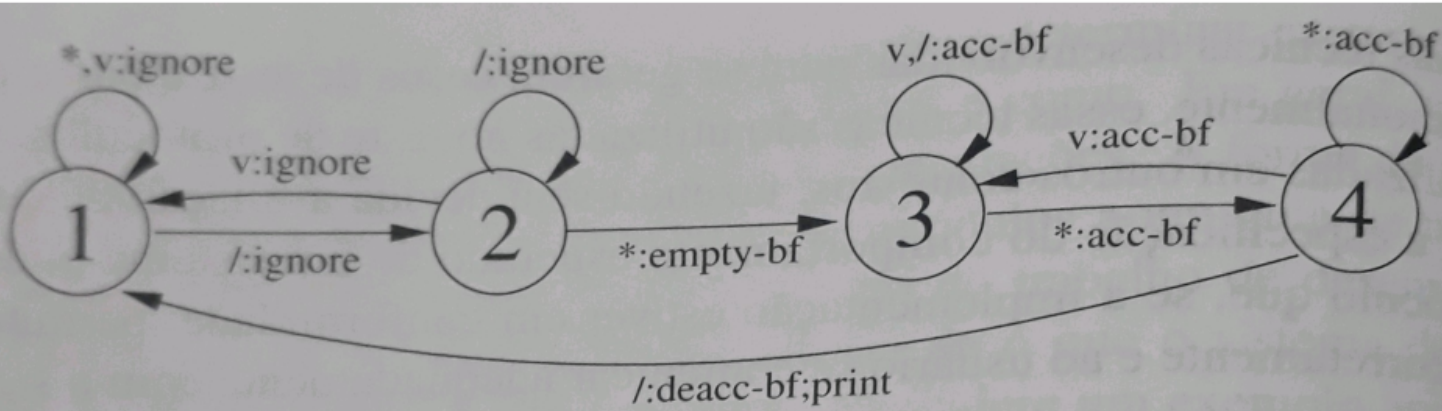
- Máquina hipotética;
- Composta por estados e transições:
  - Transições ligam estados;
- Em um dado instante a máquina só pode estar em um estado;
- Ao ocorrer um evento de entrada:
  - Um evento de saída é gerado;
  - Ocorre uma mudança de estado;
- Podem ser representadas por:
  - Diagramas de transição de estados; ou,
  - Tabelas de transição.

Representações de uma MEF

- Diagrama de Transição de estados
  - Estados são círculos;
  - Transições são arcos:
    - Cada arco é rotulado com:
      - A entrada que gera a transição; e,
      - A saída que é produzida;
      - O formato é *entrada:saída*;
- Tabela de Transição:
  - Estados são as linhas da tabela;
  - Entradas e as transições são as colunas.

Representações de uma MEF

Exemplo



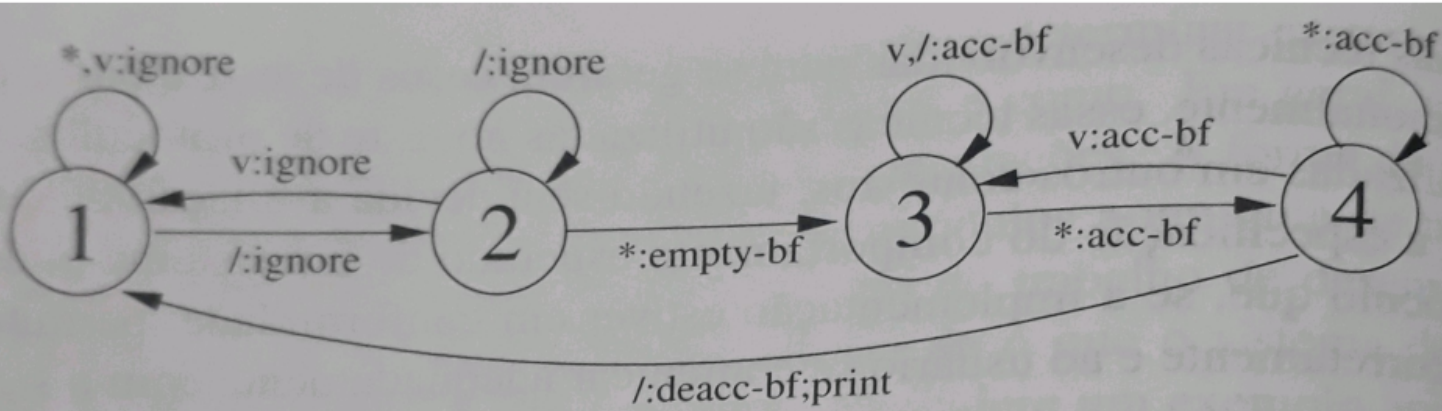
Estado \ Entrada	*	/	v	*	/	v
1	ignore	ignore	ignore	1	2	1
2	empty-bf	ignore	ignore	3	2	1
3	acc-bf	acc-bf	acc-bf	4	3	3
4	acc-bf	deacc-bf; print	acc-bf	4	1	3

Definição Formal de uma MEF

- Uma MEF é uma tupla $M = (X, Z, S, s_0, f_z, f_z)$, no qual:
  - 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;
  - $f_s : (S \times X) \rightarrow S$ é a função de próximo estado.

MEF *Comment Printer*

- Especifica o comportamento de um extrator de comentários: *Comment Printer*



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

Propriedades de uma MEF

Definição 1

Uma MEF é completamente especificada se ela trata todas as entradas em todos os estados. Em contrário, ela é parcialmente especificada.

- Em MEFs parcialmente especificadas o teste de conformidade é fraco;
  - Em contrário, o teste é forte.

Propriedades de uma MEF

Definição 2

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 até s_j.

- Ela é inicialmente conectada se a partir do estado inicial é possível atingir os demais estados da MEF;

Propriedades de uma MEF

Definição 3

Uma MEF é minimal (ou reduzida) se a não existem quaisquer dois estados equivalentes. Dois estados são equivalentes se possuem as mesmas entradas e produzem as mesmas saídas.

Propriedades de uma MEF

Definição 4

Uma MEF é determinística quando, para qualquer estado e para uma dada entrada, a MEF permite uma única transição para um próximo estado.

- Em contrário, ela é não determinística.

Geração de Sequências de Teste

Geração de Sequências de Teste

- Uma sequência de teste é uma sequência de símbolos de entrada derivada da MEF;
- Os símbolos são usados no programa implementado pela MEF para verificação de conformidade; -Saída gerada pela sequência de teste (especificação) *versus* saída gerada pelo programa;

Métodos de Geração de Sequências de Teste

Método TT (Transition Tour)

- É uma sequência que parte do estado inicial e atravessa todas as transições ao menos uma vez e retorna ao estado inicial:
- Permite a detecção de erros de saída;
- Não garante erros de transferência:
  - Erros em que a transição leva a MEF a um estado diferente do qual deveria estar.

Métodos de Geração de Sequências de Teste

Método UIO (Unique Input/Output)

- Uma sequência de identificação de estado;
- Não garante cobertura total dos erros:
  - A sequencia leva a uma única saída na especificação correta;
  - Não é garantido para implementações com erro.

Métodos de Geração de Sequências de Teste

Método DS (Distinction Sequence)

- 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:
  - Nem sempre uma DS pode ser encontrada em uma MEF.

Métodos de Geração de Sequências de Teste

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.

Considerações Finais

Pontos importantes

- Uso de modelos no teste trás diversas vantagens:
  - Possibilidade de eliminar ambiguidades:
    - Ao menos reduzi-las;
  - Possibilidde de testar o próprio modelo;
- Possibilita a automação dos testes.