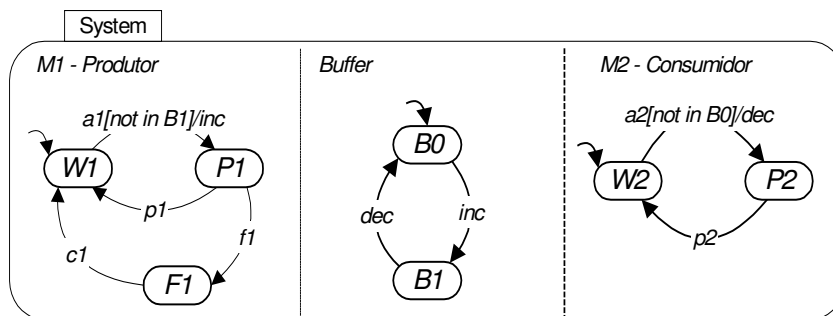


ConDado: uso para geração de casos de teste a partir de Statecharts

Ana Maria Ambrosio

Especificação em Statecharts exemplo: Produtor - Consumidor



10/12/2002

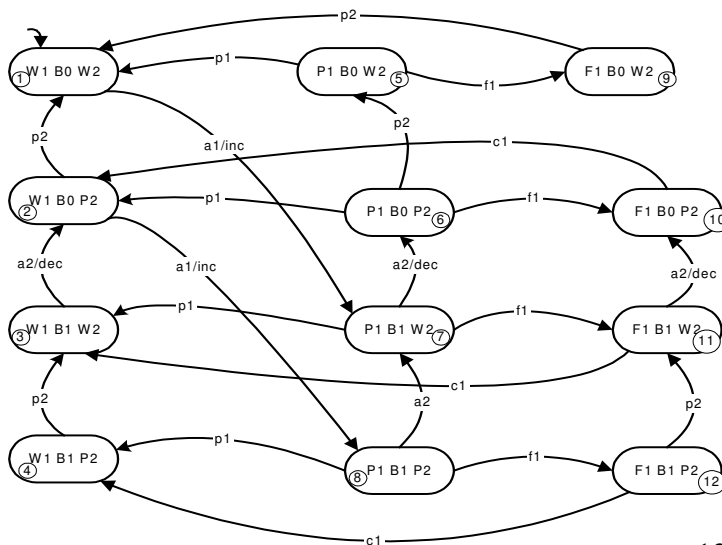
Como gerar um conjunto de casos de teste a partir de uma especificação Statecharts considerando o paralelismo dos componentes ?

A maneira natural é:

- 1 - gerar a máquina produto dos componentes paralelos ---- >> PerformCharts
- 2 - derivar casos de teste da máquina produto

10/12/2002

1- Máquina Produto



12 states
22 transitions

10/12/2002

2- Derivar casos de teste para testar o comportamento da FSM

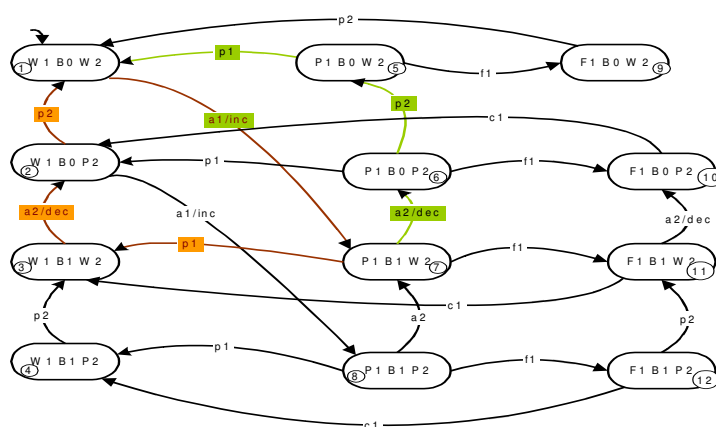
ConDado - implementa uma variação do método T
(*transitions tour*)

Consta de percorrer a máquina a partir do estado inicial e encontrar os caminhos que retornam ao estado inicial, cobrindo todas as transições.

Cada **caminho** (seqüência de transições) é um **caso de teste**.

10/12/2002

Caso de teste na máquina produto



10/12/2002

Casos de teste da máquina produto

1	a1/inc a2/dec p2 p1
2	a1/inc a2/dec p2 f1 c1
3	a1/inc a2/dec p1 p2
4	a1/inc a2/dec p1 a1/inc p2 a2/dec p2 p1
5	a1/inc a2/dec p1 a1/inc p2 a2/dec p2 f1 c1
6	a1/inc a2/dec p1 a1/inc p2 a2/dec p1 p2
7	a1/inc a2/dec p1 a1/inc p2 a2/dec p1 a1/inc p1 p2 a2/dec p2
8	a1/inc a2/dec p1 a1/inc p2 a2/dec p1 a1/inc p1 p2 a2/dec a1/inc f1 p2 a2/dec c1 p2
...	
765	...
766	a1/inc p1 a2/dec a1/inc f1 c1 p2 a2/dec a1/inc f1 p2 c1 a2/dec a1/inc f1 p2 a2/dec c1 a1/inc p2 p1 a2/dec p2

10/12/2002

Problemas:

- explosão de estados da máquina produto
- quantidade intratável de casos de teste






10/12/2002

Proposta

Implementar uma ferramenta para gerar casos de teste, a partir de especificações em Statecharts, que seja **autônoma** e que gere um conjunto **tratável** de casos de teste, sem interferência da pessoa que testa.






10/12/2002

Método

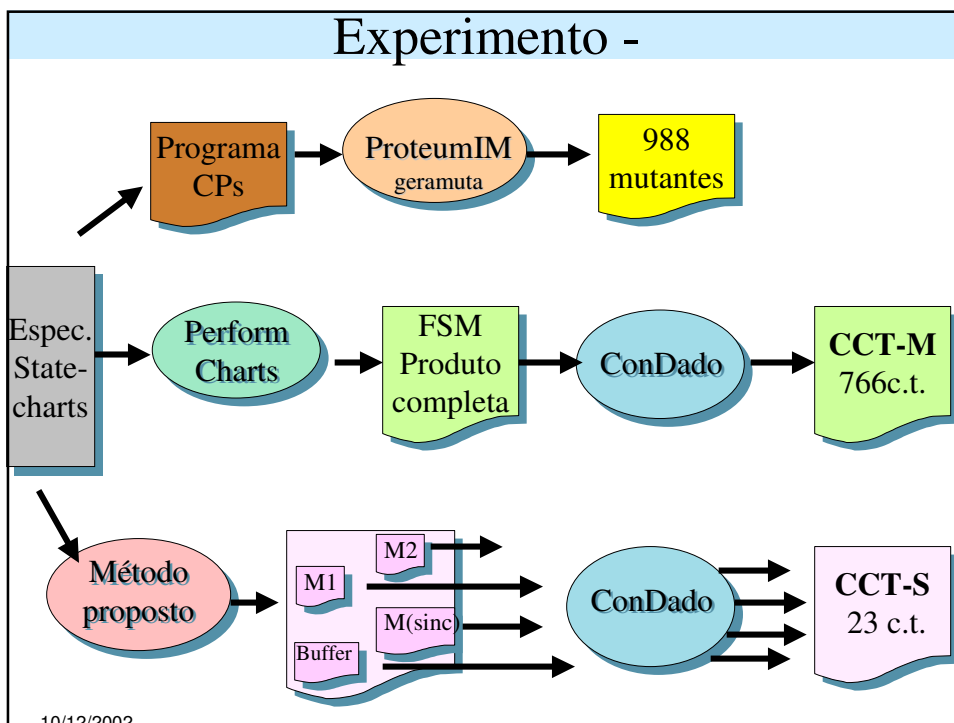
1. Gerar casos de teste p/ cada Mi individualmente 
2. Identificar a dependências dos componentes
⇒ as transições de sincronização e independentes 
3. Gerar uma EFSM considerando “apenas” as transições de sincronização = M(sinc) 
4. Gerar casos de teste p/ a M(sinc) 
5. Unir todos os casos de teste em um conjunto = CCT-S 

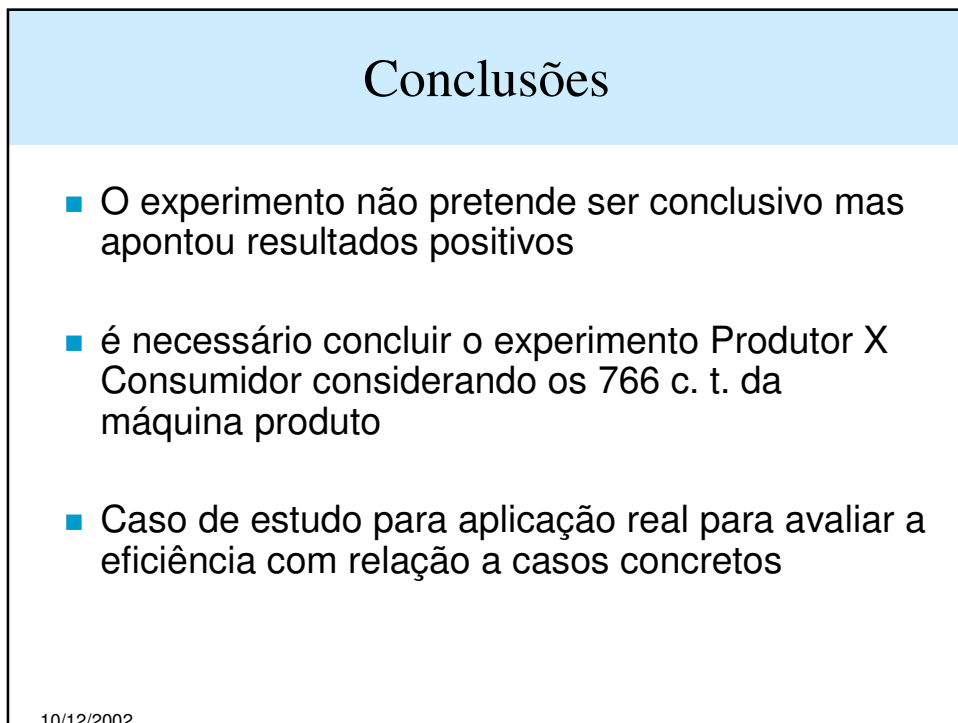
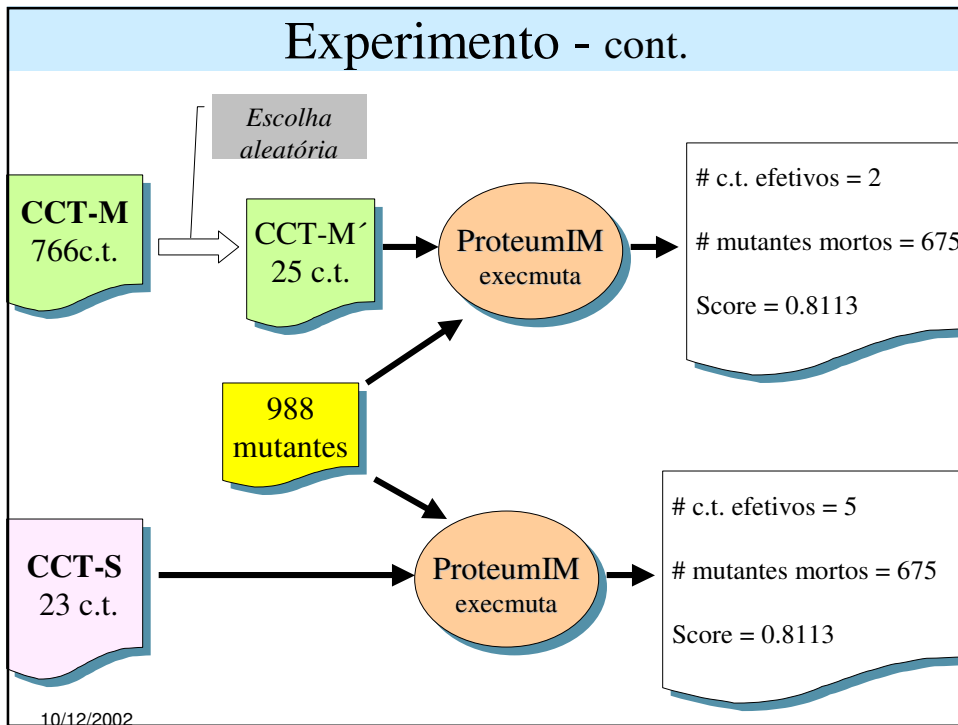
10/12/2002

Como avaliar a eficiência do conjunto de casos teste gerado - CCT-S ?

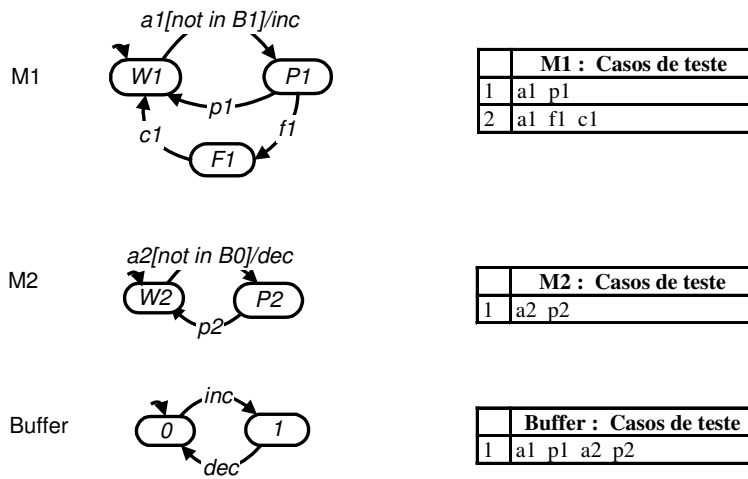
- Experimento sobre o exemplo produtor-consumidor baseado em:
 - Conceitos de:
 - Teste de Conformidade 
 - Mutantes de implementação (interface) 
 - Ferramentas existentes:
 - PerformCharts 
 - ConDado 
 - ProteumIM 

10/12/2002



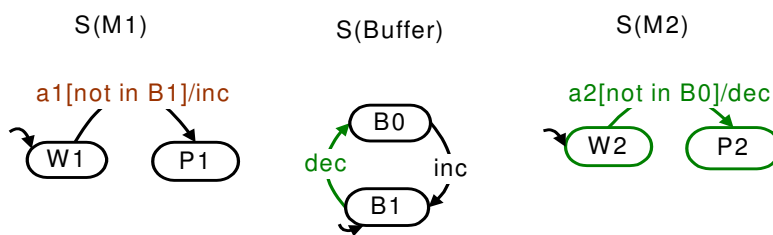


Casos de teste para cada M(i)



10/12/2002

Transições de Sincronização

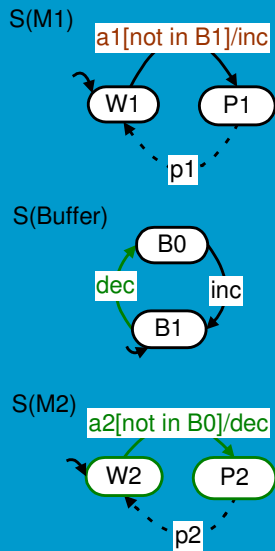


transições de sincronização:

- $input(M_i) = input(M_k) \quad \forall_i \neq_k$
- $input(M_i) = output(M_k) \quad \forall_i \neq_k$
- $condition(M_i) = not\ in\ any\ state\ s(M_k) \quad \forall_i \neq_k$

10/12/2002

Máquina de Sincronização - passo 1

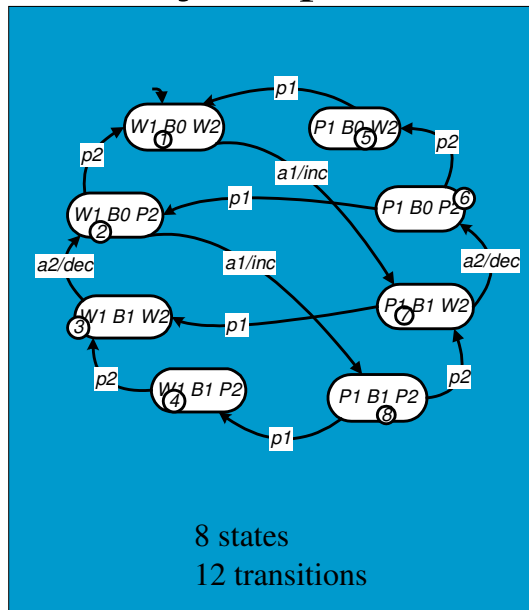


- para cada máquina ou componente, criar uma máquina simples com:
 - transitions de sincronização
 - transições para tornar a máquina inicialmente conexa $S(M_i)$;

10/12/2002

Máquina de sincronização – passo 2

Gerar a máquina produto a partir das máquinas simplificadas dos componentes



10/12/2002

Casos de teste de M(sinc)

1	a1/inc a2/dec p2 p1
2	a1/inc a2/dec p1 p2
3	a1/inc a2/dec p1 a1/inc p2 a2/dec p2 p1
4	a1/inc a2/dec p1 a1/inc p2 a2/dec p1 p2
5	a1/inc a2/dec p1 a1/inc p2 a2/dec p1 a1/inc p1 p2 a2/dec p2
6	a1/inc a2/dec p1 a1/inc p2 p1 a2/dec p2
7	a1/inc a2/dec p1 a1/inc p1 p2 a2/dec p2
8	a1/inc a2/dec p1 a1/inc p1 p2 a2/dec a1/inc p2 a2/dec p2 p1
9	a1/inc a2/dec p1 a1/inc p1 p2 a2/dec a1/inc p2 a2/dec p1 p2
18	...
19	a1/inc p1 a2/dec a1/inc p1 p2 a2/dec a1/inc p2 p1 a2/dec p2
19 casos de teste gerados	

10/12/2002



CCT-S = conjunto de casos de teste gerados pelo método proposto

1	a1/inc a2/dec p2 p1
2	a1/inc a2/dec p1 p2
3	a1/inc a2/dec p1 a1/inc p2 a2/dec p2 p1
4	a1/inc a2/dec p1 a1/inc p2 a2/dec p1 p2
5	a1/inc a2/dec p1 a1/inc p2 a2/dec p1 a1/inc p1 p2 a2/dec p2
6	a1/inc a2/dec p1 a1/inc p2 p1 a2/dec p2
7	a1/inc a2/dec p1 a1/inc p1 p2 a2/dec p2
8	a1/inc a2/dec p1 a1/inc p1 p2 a2/dec a1/inc p2 a2/dec p2 p1
9	a1/inc a2/dec p1 a1/inc p1 p2 a2/dec a1/inc p2 a2/dec p1 p2
18	...
19	a1/inc p1 a2/dec a1/inc p1 p2 a2/dec a1/inc p2 p1 a2/dec p2
19 casos de teste gerados	

M1 : Casos de teste	
1	a1 p1
2	a1 p1 c1

M2 : Casos de teste	
1	a2 p2

Buffer : Casos de teste	
1	a1 p1 a2 p2

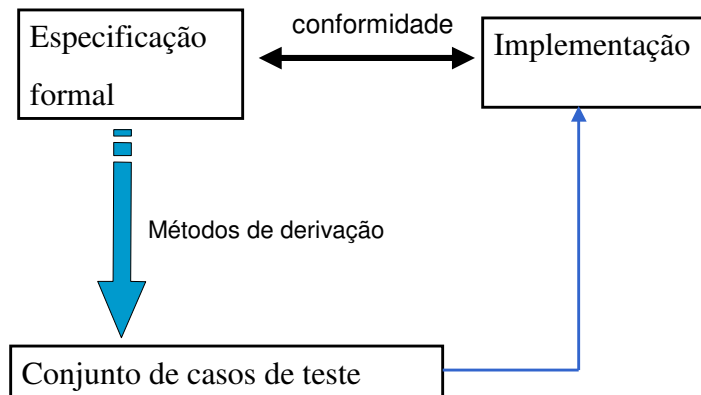
= 19 casos de teste M (sinc)
 + 2 casos de teste M1
 + 1 caso de teste Buffer
 + 1 caso de teste M2

 23 casos de teste

10/12/2002



Teste de Conformidade



10/12/2002

Mutantes

Dado um programa P que se deseja testar, são criados vários programas M_1, M_2, \dots, M_n chamados **Mutantes**.

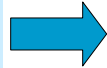
Os Mutantes possuem pequenas **alterações sintáticas** em relação à P .

10/12/2002

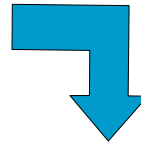
PerformCharts

LAC/INPE

Especificação
em
Statecharts



Máquina de
estados
produto
planificada



Análise de
desempenho



10/12/2002

ConDado

IC/Unicamp
Projeto ATIFS

Especificação
em
máquina de
estados
planificada



Conjunto de
casos de teste
(seq. de
transições)
Baseado no
método T



10/12/2002

Proteum IM

ICMC/USP S. Carlos

Código fonte
(ling. C)

MutaGen

Conjunto de
mutantes da
implementação

Conjunto
de casos de
teste

ExecMuta

Conjunto de
mutantes

Score de mutação =

mutantes mortos

total de mutantes - mut equivalentes

Medida de eficiência
do conjunto de c.teste

10/12/2002