Universidade de São Paulo – USP
Instituto de Ciências Matemáticas e de Computação – ICMC
Departamento de Sistemas de Computação – SSC

SSC5793 - Especificacao Formal de Software

Notas de Aula

Prof.: Adenilso Simão

Turma 2/2015

Segunda-Feira, 16 de Novembro 2015

Escribas: Diego Damasceno, Sidgley Camargo, Stevão Andrade

1 Considerações Iniciais da Aula

A aula foi iniciada fazendo uma breve revisão da aula anterior. O foco inicial da aula foi recapitular os conceitos sobre a propriedade magica (PM), que possui a característica de garantir que um conjunto de sequencias é capaz de distinguir uma *FSM* de todas as outras implementações possíveis que contenham o mesmo numero de estados.

Formalmente a propriedade magica (PM) é definida da seguinte forma:

$\forall N \in DF_n(S), N \neq S \rightarrow \exists t \in T, N \neq S$

Lê-se: Se para todo n pertencente ao domínio de maquinas de estado com N estados, **n** é diferente de S, então, existe um t pertencente a T, capaz de provar que n é diferente de S.

Assim, ao aplicar a propriedade magica, é possível atestar que um dado conjunto de sequencias é *N-Complete*. A propriedade magica passa a ser chamada adiante de **propriedade N-Complete**.

Um dado conjunto de teste T é definido como **N-Complete**, se para cada *FSM* $N \in DF_n(S)$, tal que N e S sejam distinguíveis, existe uma sequencia de teste t capaz de distingui-las.

Para um dado conjunto de testes T de uma *FSM* M, duas sequencias, $\alpha, \beta \in T$ são **T-distinguíveis**, se existe $\alpha\gamma, \beta\gamma \in T$, tal que, $\lambda(\delta(s_0, \alpha), \gamma) \neq \lambda(\delta(s_0, \beta), \gamma)$.

Ou seja, dado um conjunto de casos de casos de teste T, duas sequencias, α e β são **T-distinguíveis**, se existem as sequencias $\alpha\gamma, \beta\gamma \in T$, tal que, γ consegue distinguir os estados que as sequências α e β alcançam.

2 Método por confirmação

Consiste na abordagem para verificar se um dado conjunto de teste T é **N-Complete**.

Para garantir que um dado conjunto de testes é **N-Complete**, é necessário respeitar algumas condições:

Definição:

Um conjunto de sequencias é dito como confirmado quando contem sequencias de transferências para todos os estados de uma dada *FSM M* e quaisquer sequencias convergindo (ou seja, que levem para o mesmo estado) em qualquer *FSM* que tenha a mesma saída para T e tenha os mesmos estados que M, também convirja em M.

Seja T um conjunto de teste de uma dada *FSM M* com n estados. T é **N-Complete** para M, se existe um conjunto confirmado K contido em T com as seguintes propriedades:

1. $\varepsilon \in K$ - O subconjunto vazio está contido em K.
2. Para cada $(s,x) \in D$, existe α, $\alpha x \in K$, tal que ao aplicar a sequencia α a partir do estado inicial, chega-se no estado s.

Para garantir que um dado conjunto é **N-Complete**, deve-se então seguir 3 lemas que são subdivididos no seguinte conjunto de passos: (1) Gerar árvore de testes para as sequencias de transição; (2) Realizar enumeração de todos os nós; (3) Gerar um grafo de distinção; (4) Realizar inferência sobre cada um dos estados; (5) Encontrar um clique dentro do grafo de distinção; (6) Confirmação de estados por meio de identificação de sufixo comum; e (7) Reconstruir a *FSM* com base nas informações inferidas.

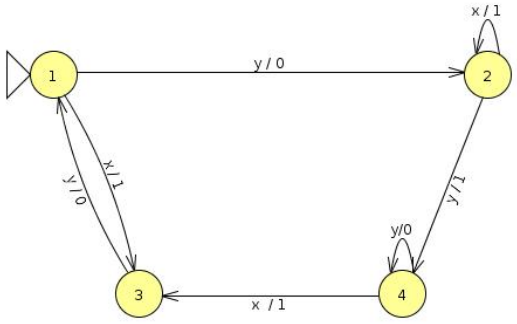

Figura 1: Representação Gráfica da MEF 1

Dada a*FSM* 1 e o conjunto de casos de teste T formado pelas sequencias abaixo:

-> xyyxy
-> yyyyyyxyyy

Deseja-se verificar se o conjunto de sequencias é dito como **N-Complete**.

Passo 1: Gerar árvore de teste para as sequencias de transição

Na árvore de testes, os nós da árvore correspondem aos estados do grafo de distinção e as arestas correspondem a cada sequencia de transferência contida no conjunto de casos de teste.

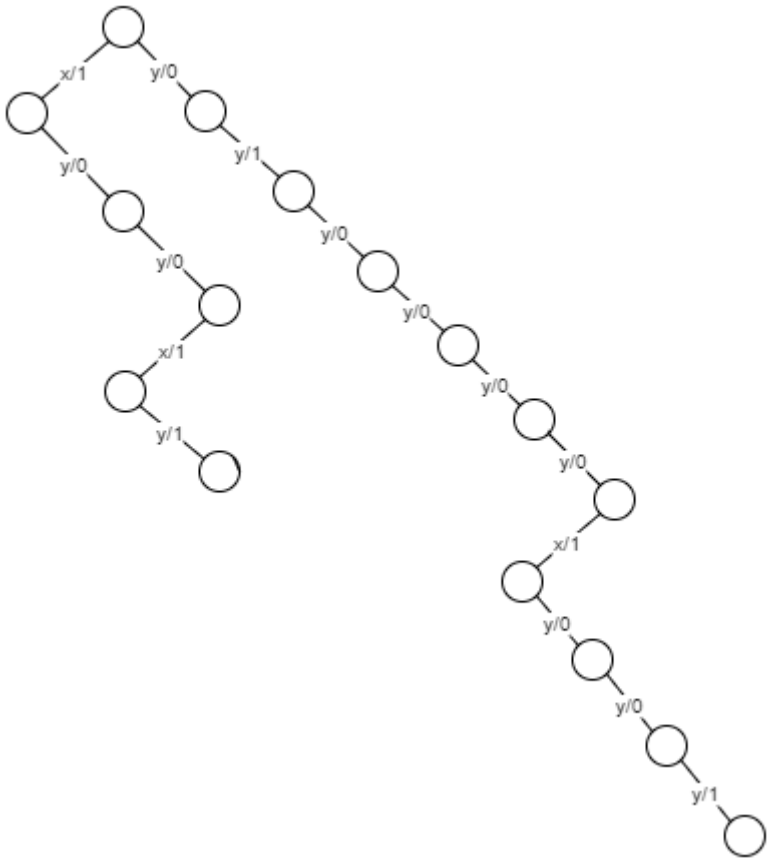

Figura 2: Árvore de teste.

Passo 2: Realizar enumeração de todos os nós

Os nós da arvore de teste são devidamente enumerados:

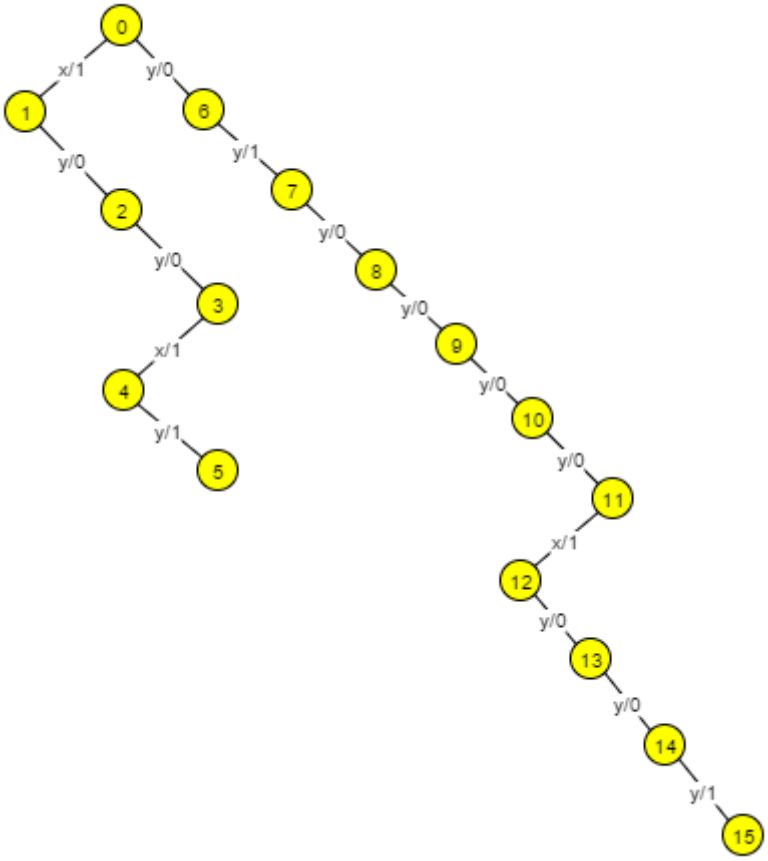

Figura 3: Árvore de teste com nós enumerados.

Passo 3: Gerar grafo de distinção

O grafo de distinção é criado, o objetivo deste passo é modelar o problema como um grafo e consiste na representação visual da aplicação do **Lema 1** para verificar se uma dada sequencia é **N-Complete**.

É realizada uma análise par a par entre os nós do grafo, cujo o objetivo é encontrar uma sequência de distinção que identifica unicamente os pares de nó. Quando existe essa sequência, os pares de nós são conectados no grafo de distinção.

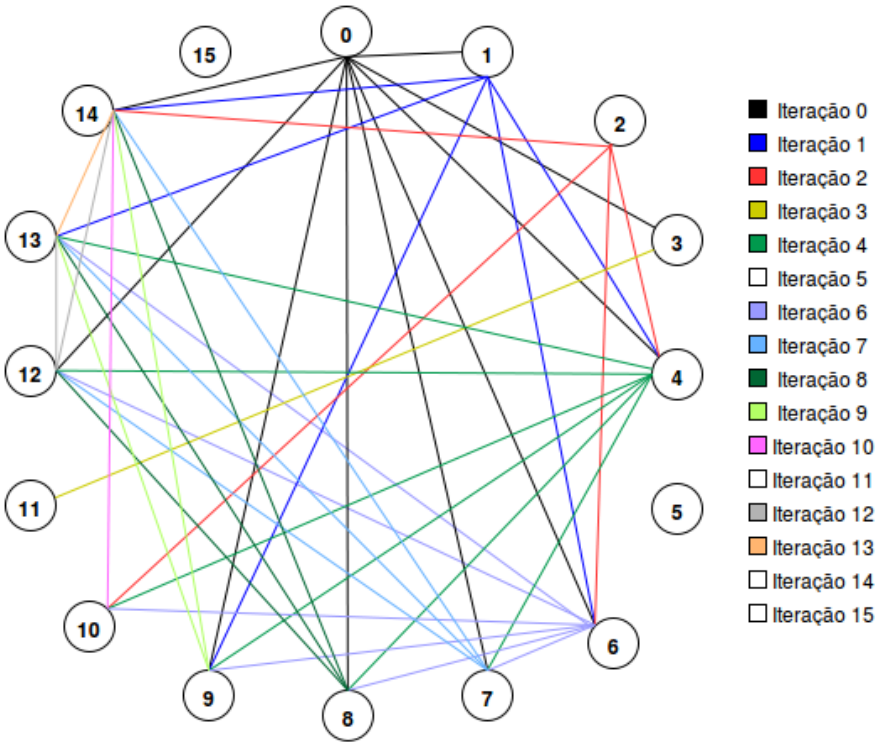

Figura 4: Grafo de distinção

As tabelas abaixo (Iteração 1 – 13) descrevem a sequência de teste utilizada para identificar a distinção entre pares de nós. As iterações 5, 11, 14 e 15 não geram ligações entre nós. Isso não significa que não exista distinção, apenas que não foi possível identificar nesse etapa do processo de confirmação.

Tabela 1: Nós e sequências distinguíveis – iteração 0

Nós	**Sequência de distinção**	**Saída distinguível dos nós**
0 - 1	yy	0={y/0,y/1}; 1={y/0,y/0}
0 - 3	xy	0={x/1,y/0}; 3={x/1,y/1}
0 - 4	y	0={y/0}; 4={y/1}
0 - 6	y	0={y/0}; 6={y/1}
0 - 7	yy	0={y/0,y/1}; 7={y/0,y/0}
0 - 8	yy	0={y/0,y/1}; 8={y/0,y/0}
0 - 9	yy	0={y/0,y/1}; 9={y/0,y/0}
0 - 12	yy	0={y/0,y/1}; 12={y/0,y/0}
0 - 14	y	0={y/0}; 14={y/1}

Tabela 2: Nós e sequências distinguíveis – iteração 1

Nós	**Sequência de distinção**	**Saída distinguível dos nós**
1 - 4	y	1={y/0}; 4={y/1}
1 - 6	y	1={y/0}; 6={y/1}
1 - 9	yyxy	1={y/0,y/0,x/1 y/1}; 9={y/0,y/0,x/1 y/0}
1 - 13	y	1={y/0,y/0}; 13={y/0,y/1}
1 - 14	y	1={y/0}; 14={y/1}

Tabela 3: Nós e sequências distinguíveis – iteração 2

Nós	**Sequência de distinção**	**Saída distinguível dos nós**
2 - 4	y	2={y/0}; 4={y/1}
2 - 6	y	2={y/0}; 6={y/1}
2 - 10	yxy	2={y/0,x/1,y/1}; 10={y/0,x/1,y/0}
2 - 14	y	2={y/0}; 14={y/1}

Tabela 4: Nós e sequências distinguíveis – iteração 3

Nós	**Sequência de distinção**	**Saída distinguível dos nós**
3 - 11	xy	3={x/1,y/1}; 11={x/1,y/0}

Tabela 5: Nós e sequências distinguíveis – iteração 4

Nós	**Sequência de distinção**	**Saída distinguível dos nós**
4 - 7	y	4={y/1}; 7={y/0}
4 - 8	y	4={y/1}; 8={y/0}
4 - 9	y	4={y/1}; 9={y/0}
4 - 10	y	4={y/1}; 10={y/0}
4 - 12	y	4={y/1}; 12={y/0}
4 - 13	y	4={y/1}; 13={y/0}

Tabela 6: Nós e sequências distinguíveis – iteração 6

Nós	**Sequência de distinção**	**Saída distinguível dos nós**
6 - 7	y	6={y/1}; 7={y/0}
6 - 8	y	6={y/1}; 8={y/0}
6 - 9	y	6={y/1}; 9={y/0}
6 - 10	y	6={y/1}; 10={y/0}
6 - 12	y	6={y/1}; 12={y/0}
6 - 13	y	6={y/1}; 13={y/0}

Tabela 7: Nós e sequências distinguíveis – iteração 7

Nós	**Sequência de distinção**	**Saída distinguível dos nós**
7 - 12	yyy	7={y/0,y/0,y/0}; 12={y/0,y/0,y/1}
7 - 13	yy	7={y/0,y/0}; 13={y/0,y/1}
7 - 14	y	7={y/0}; 14={y/1}

Tabela 8: Nós e sequências distinguíveis – iteração 8

Nós	**Sequência de distinção**	**Saída distinguível dos nós**
8 - 12	yyy	8={y/0,y/0,y/0}; 12={y/0,y/0,y/1}
8 - 13	yy	8={y/0,y/0}; 13={y/0,y/1}
8 - 14	y	8={y/0}; 14={y/1}

Tabela 9: Nós e sequências distinguíveis – iteração 9

Nós	**Sequência de distinção**	**Saída distinguível dos nós**
9 - 13	yy	9={y/0,y/0}; 13={y/0,y/1}
9 - 14	y	9={y/0}; 14={y/1}

Tabela 10: Nós e sequências distinguíveis – iteração 10

Nós	**Sequência de distinção**	**Saída distinguível dos nós**
10 - 14	y	10={y/0}; 14={y/1}

Tabela 11: Nós e sequências distinguíveis – iteração 12

Nós	**Sequência de distinção**	**Saída distinguível dos nós**
12 - 13	yy	12={y/0,y/0}; 13={y/0,y/1}
12 - 14	y	12={y/0}; 14={y/1}

Tabela 12: Nós e sequências distinguíveis – iteração 13

Nós	**Sequência de distinção**	**Saída distinguível dos nós**
13 - 14	y	13={y/0}; 14={y/1}

Passo 4: Encontrar *clique* dentro do grafo de distinção

Identificar um *clique* no grafo de distinção consiste em verificar um conjunto mínimo de sequências confirmadas dentro do grafo.

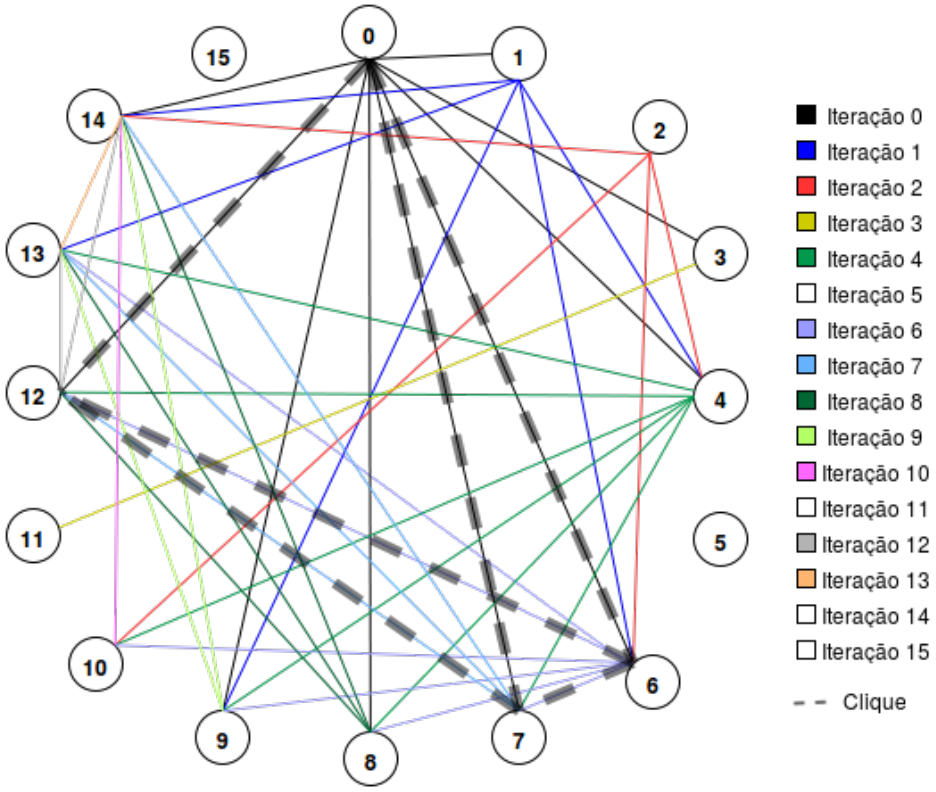

Figura 5: Clique no grafo de distinção

Analisando visualmente, um *clique* pode ser definido como um conjunto de N nós onde todos se ligam entre si dentro do grafo de distinção. Neste caso os nós {0, 6, 7, 12}. A linha pontilhada na cor preta mostra visualmente o *clique* na Figura 5.

Passo 5: Realizar inferência sobre os estados

Como o número de estados da *FSM* é previamente conhecido por meio do *clique*, é possível identificar um conjunto de N estados que sejam distintos entre si. Assim, é possível concluir que esses N estados correspondem aos estados da máquina. Para que seja possível inferir a respeito dos demais estados, a primeira etapa consiste em rotular os estados encontrados com o *clique*. Isso é feito com os estados {0, 6, 7, 12}, assumindo, respectivamente, os rótulos {A, B, C, D}.

Em seguida, o grafo é percorrido e, para cada estado não conhecido, é inferido seu rotulo (atribuindo letra dentre as definidas no clique).

Tabela 13: Inferência a partir do clique - Iteração 1

Nó do grafo	**Ligações**	**Possível rotulo (inferência)**
1	{0/A, 6/B}	{C, D}
2	{6/B}	{A, C, D}
3	{0/A}	{B, C, D}
4	{0/A, 7/C, 12/D}	{B}
5	{-}	{A, B, C, D}
8	{0/A, 4/B, 12/D}	{C}
9	{0/A, 4/B, 6/B}	{C, D}
10	{4/B, 6/B}	{A, C, D}
11	{-}	{A, B,C ,D}
13	{4/B, 6/B, 7/C, 8/C, 12/D}	{A}
14	{0/A, 7/C, 8/C, 12/D, 13/A}	{B}
15	{-}	{A, B, C, D}

Ao analisar o estado 1, percebe-se que o mesmo liga-se aos nós {0, 6} que foram rotulados respectivamente com as letras {A, B}. Logo, por inferência, deduz-se que os possíveis rotulo para o nó 1 são {C, D}. Contudo, como não há informação suficiente, essa análise deve ser postergada e o próximo nó do grafo de distinção deve ser analisado.

Ao analisar o nó 4, percebe-se que o mesmo possui ligação com os nós {0, 7, 12}, que foram rotulado respectivamente como {A, B, C}. Assim, por inferência deduz-se que o rotulo para o nó 4 é {D}.

Após identificar o rotulo do nó 4 (B), as análises seguintes levam em consideração não apenas os rótulos definidos inicialmente pelo *clique*, mas também o rotulo do nó 4 recém descoberto. O processo é iterativo e a medida que novos rótulos são descobertos, essas informações também são utilizadas para a identificação dos rótulos seguintes.

As novas informações sobre os rótulos que são descobertas durante a iteração são destacadas na Tabela 14 pela cor vermelha.

Ao completar uma iteração e analisar todos os nós, conforme analise feita na tabela 14,

tem-se o grafo de distinção representado na Figura 11 com os estados {4, 8, 13, 14} com seus respectivos rótulos {D, C, A, B} identificados.

Este passo de identificação dos nós por meio de inferência no grafo corresponde à representação visual do **Lema 2** para verificar se uma dada sequencia é **N-Complete**.

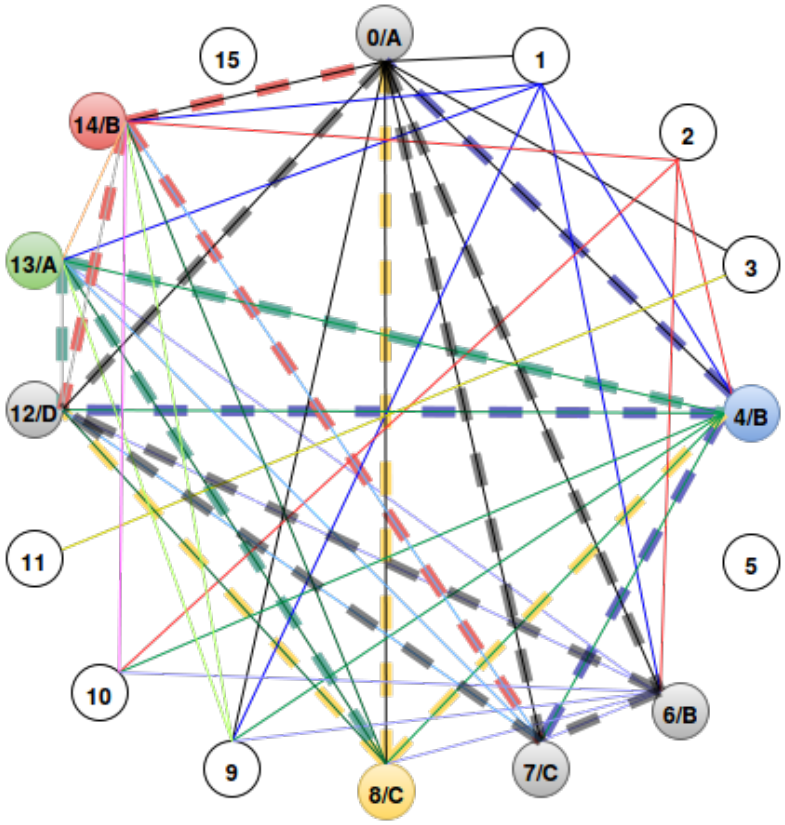

Figura 6: Inferência sob o clique

Passo 6: Confirmação de estados por meio de identificação de sufixo comum

Após identificar os rótulos por meio de inferência no grafo, é necessário atualizar a árvore de teste com os rótulos identificados. Os novos rótulos, 4, 8, 12, 13, identificados são destacados pela cor verde:

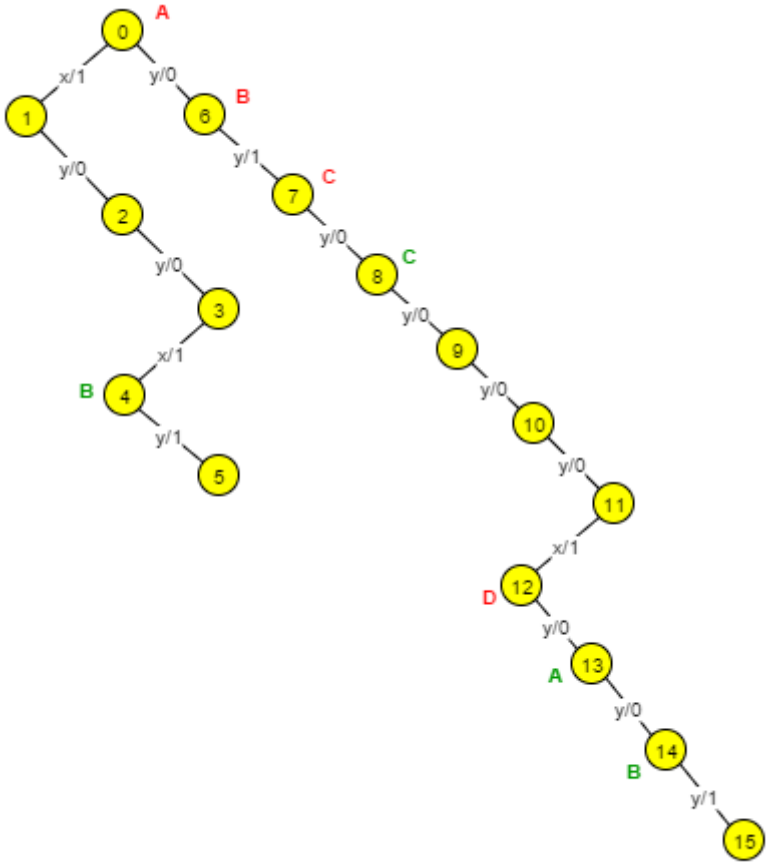

Figura 7: Árvore de teste com nós rotulados - 1 iteração.

Com as novas informações na árvore de teste, é possível agora identificar novos rótulos com base na verificação da árvore de teste por meio de uma avaliação de sufixo comum. Este passo de identificação dos nós por meio de verificação de sufixo comum na árvore de teste corresponde à representação visual do **Lema 3** para verificar se uma dada sequência é **N-Complete**.

Ao analisar a árvore de testes é possível observar que ao aplicar Y/0 ao nó 7, rotulado como C, chega-se ao nó 8, que também é rotulado por C. Logo, percebe-se, nesta situação, que na árvore de testes ao encontra-se em um nó rotulado como C e aplica-se um Y/0, torna-se a atingir um nó rotulado por C.

A árvore de teste representada na Figura 8 mostra a avaliação dos rótulos para os nós 7 e 8:

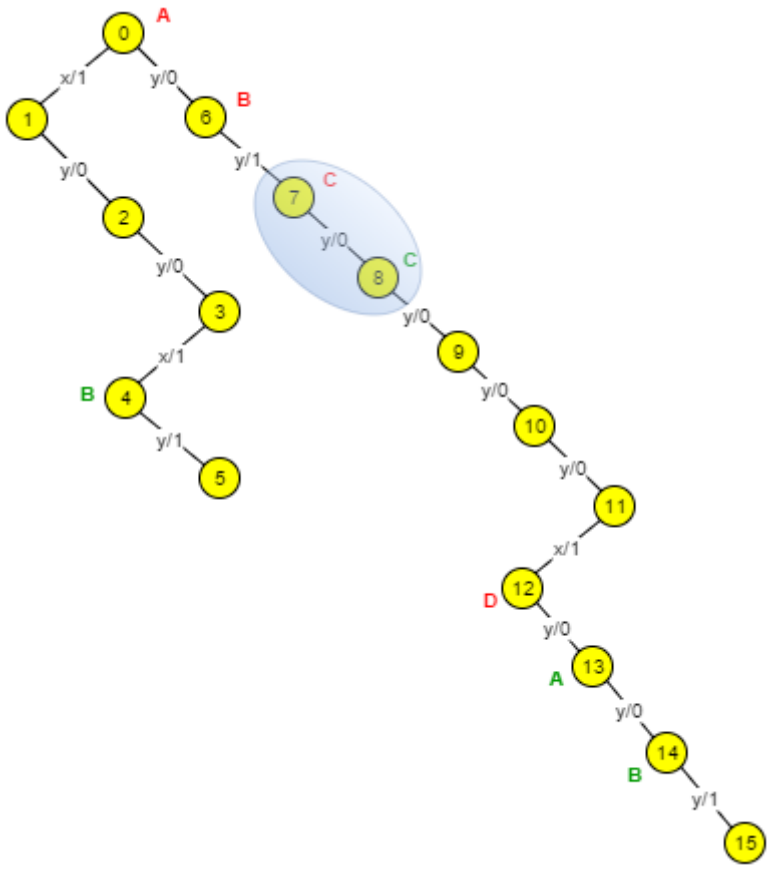

Figura 8: Árvore de teste - Nós 7 e 8 destacados.

Assim, por sufixo comum, é possível garantir que os nós 9, 10, 11 da árvore de testes também possuem o rotulo C.

Portanto, na sequência devem ser atualizados tanto a árvore de testes como o grafo de distinção com os novos rótulos descobertos:

Os novos rótulos identificados na árvore de teste são destacados na cor azul.

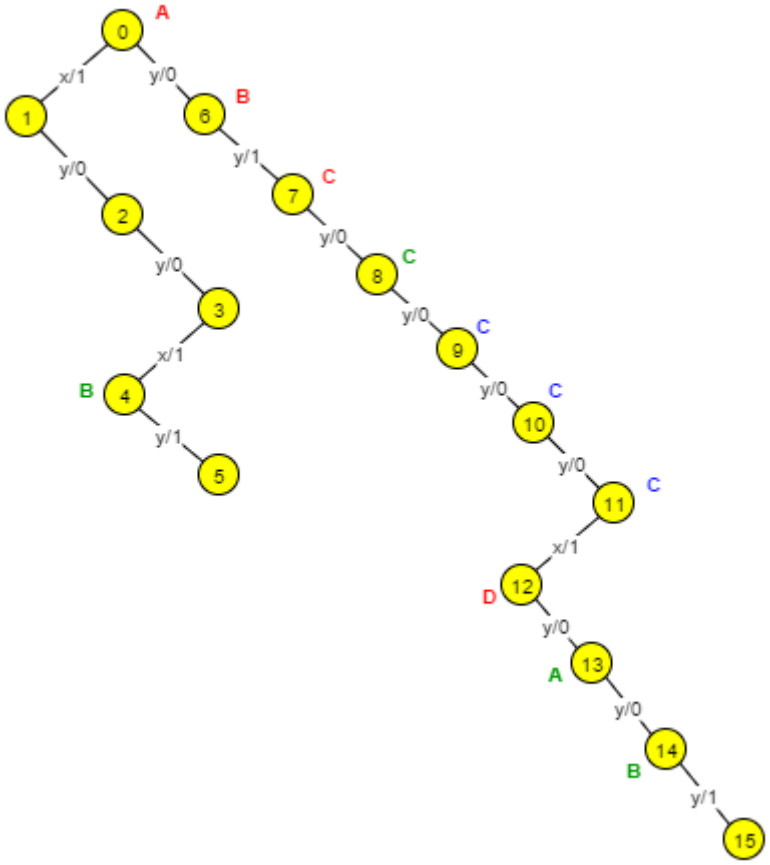

Figura 9: Árvore de teste - Nós 9, 10, 11 rotulados por sufixo comum (Azul).

O grafo de distinção é atualizado com as informações sobre os rótulos descobertos na árvore de teste por meio da análise do sufixo comum:

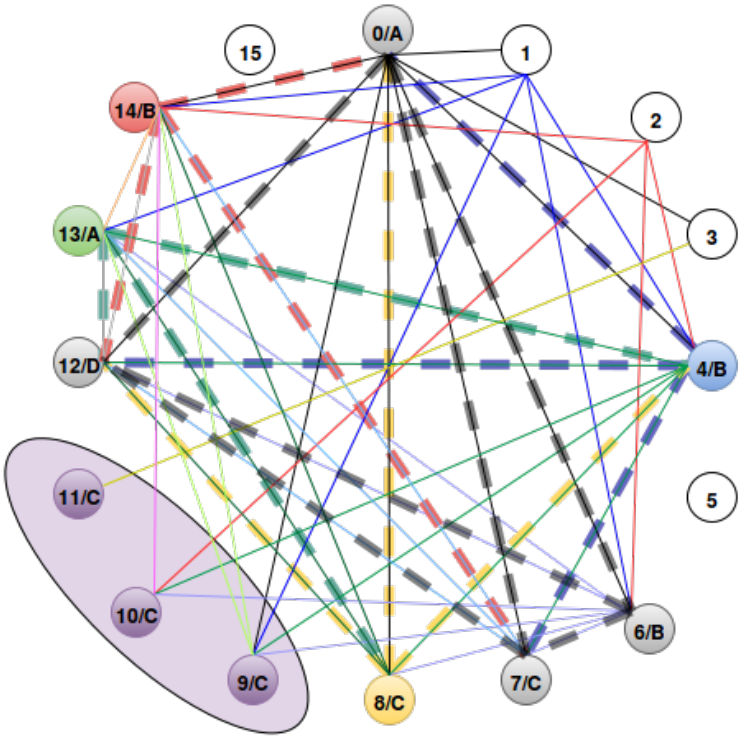

Figura 10: Grafo de distinção - Informações atualizadas pela analise de sufixo comum

Como não é possível inferir novas informações sobre os rótulos com base na árvore de testes, então retorna-se para o **passo 5** e uma nova iteração para análise por inferência deve ser realizada. Desta vez serão considerados os nós rotulados descobertos:

Tabela 14: Inferência a partir dos novos rótulos - Iteração 2

Nó do grafo	**Ligações**	**Possível rotulo (inferência)**
1	{0/A, 4/B, 6/B, 9/C, 13/A, 14/B}	{D}
2	{4/B, 6/B, 10/C}	{A, D}
3	{0A, 11/C }	{B, D}
5	{-}	{A,B,C,D}
15	{-}	{A,B,C,D}

Ao realizar a segunda iteração no grafo de distinção, é possível inferir informações para o rotulo do nó 1. Como o mesmo faz ligação com os nós 0, 4, 6, 9, 13, 14 que são respectivamente rotulados como A, B, B, C, A, B, por inferência, garante-se que o nó 1 deve passar a ser rotulado como D.

Assim, atualiza-se novamente o grafo de distinção e a árvore de teste:

Grafo de distinção com informação atualizada sobre o rotulo do nó 1:

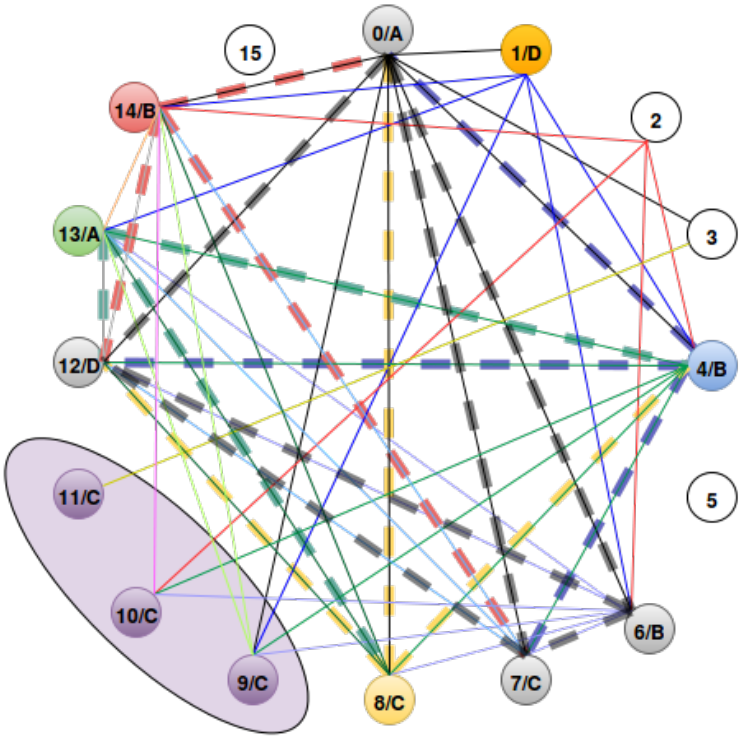

Figura 11: Grafo de distinção - Inferência realizada sobre o nó 1

Árvore de testes atualizada com informações sobre o rotulo para o nó 1:

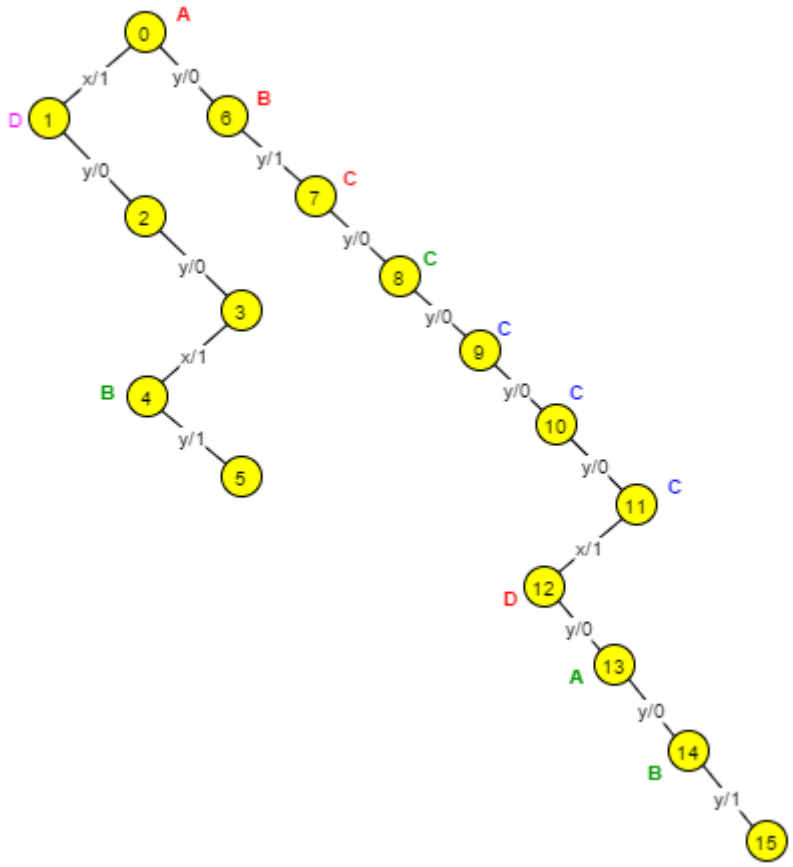

Figura 12: Árvore de teste - Nó 1 rotulado por inferência (Rosa).

Com as novas informações na árvore de teste, é novamente possível aplicar as informações contidas no **Lema 3** e identificar novos rótulos por meio da verificação de sufixo comum:

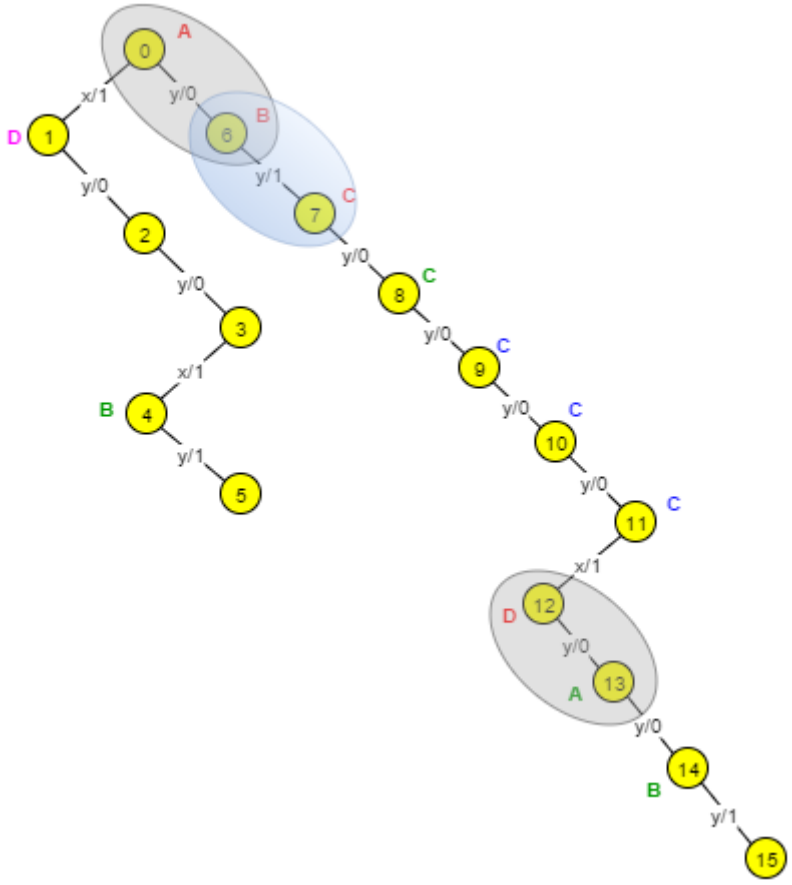

Figura 13: Árvore de teste - Pares de Nó (0, 6), (6, 7), (12, 13) destacados.

Percebe-se que ao encontrar-se em um estado rotulado como D e aplicar a transição Y/0, atinge-se um estado rotulado por A. Assim, por sufixo comum, é possível concluir que o estado 3 também possui rotulo A.

Percebe-se que ao encontra-se em um estado rotulado por A, ao aplicar uma transição Y/0, atinge-se um nó rotulado por B. Assim, por sufixo comum, é possível concluir que o estado 3 também possui rotulo B.

Percebe-se que ao encontrar-se em um estado rotulado por B e aplicar uma transição Y/1, atinge-se um nó rotulado por C. Assim, por sufixo comum, é possível concluir que os estado 5 e 15 também possui rotulo C.

Após a dedução dos novos rótulos utilizando sufixo comum, todos os nós da arvore de testes, bem como do grafo de dependência, possuem rótulos identificados. As Figuras 14 e 15 ilustram a árvore de teste e o grafo de distinção após a descoberta de todos os nós.

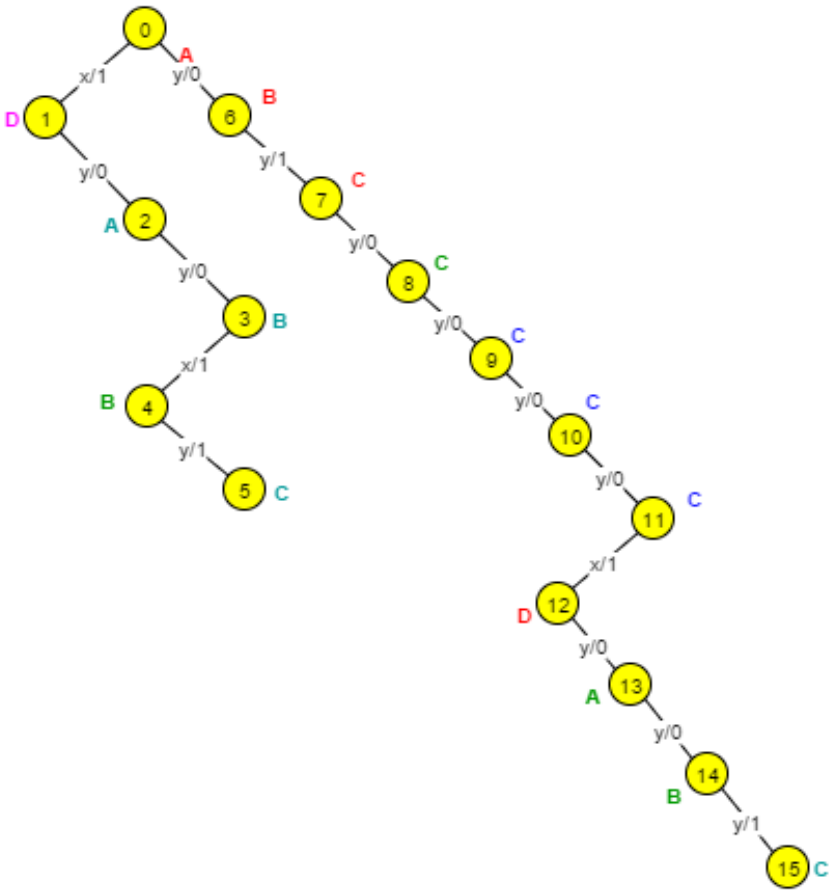

Figura 14: Árvore de teste - Nós 2, 3, 5, 15 rotulado por sufixo comum.

Referências

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5396326

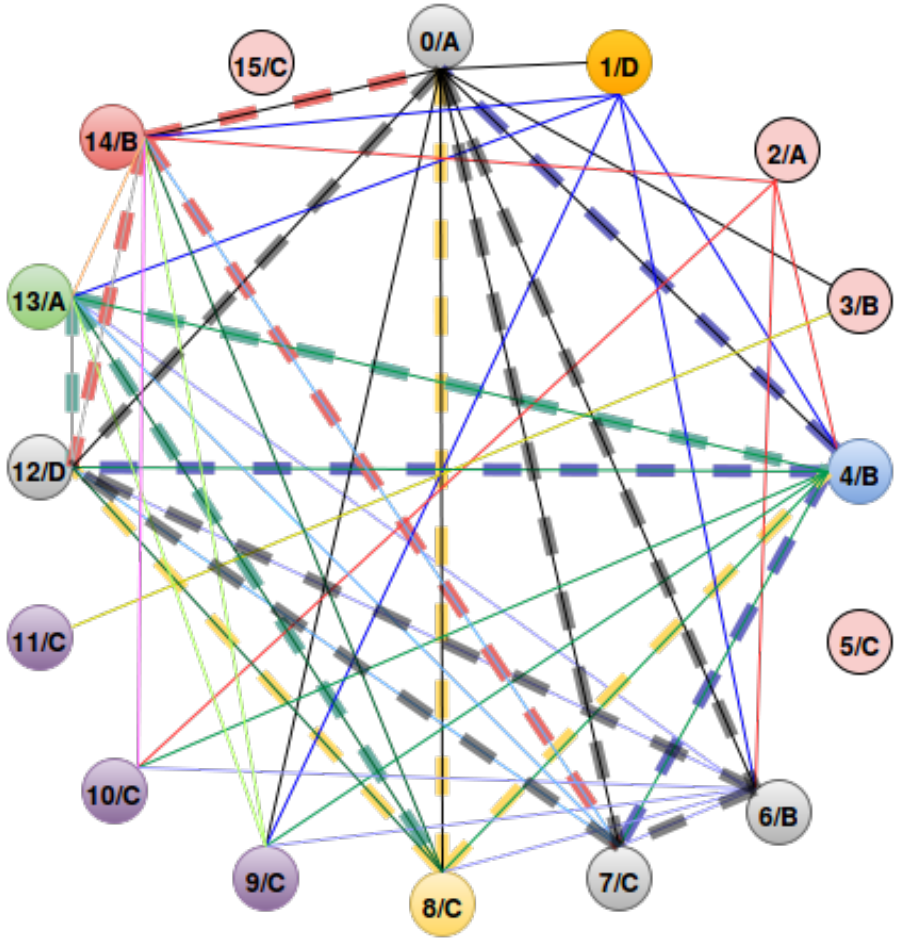

Figura 15: Grafo de distinção após inferências

3 Moore Lock

Assume-se que uma máquina M_I (Figura 16) é idêntica a uma máquina M_S exceto por possui k estados extras $q_1, q_2, ..., q_k$ e uma transição no estado s_i sob a entrada a_1 que leva a um estado extra q_1. Além disso, cada estado extra q_j tal que $1 \leq j < k$ está ligado a outro estado extra q_{j+1} por meio de uma transição com entrada a_{j+1} e, no pior caso, as demais transições retornam para os estados pertencentes a M_S e o único meio de alcançar q_k e cobrir a transição a_{k+1} é por meio de uma única sequência de comprimento $k+1$ aplicada em s_i. A esta sequência única denominamos *Moore lock*, ou *Combination lock*. O conceito de *Moore lock* serve de base para estender métodos capazes de gerar conjuntos de teste n-completos para domínios onde uma máquina M_I possui um número de estados superior ao total de estados de M_S.

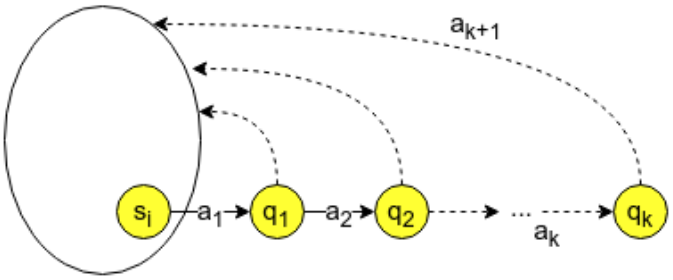

Figura 16: Moore Lock

4 Método W

O método W se utiliza do *conjunto de caracterização* (Conjunto W) e do conjunto de cobertura das transições (Transition Cover Set - P set) para testar cada uma das transições de uma FSM M_S. O método W é composto dos seguintes passos:

- Geraçao do conjunto de cobertura de estados (State cover set - Q set)
- Geraçao do conjunto de cobertura de transições (Transition cover set - P set)
- Aplicação das sequências de distinção nas sequências do P set

Na Figura 17 pode ser visto uma MEF que será usada como exemplo na aplicação do método W.

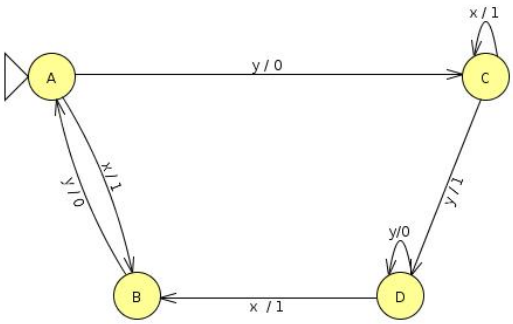

Figura 17: MEF usada como exemplo na aplicação do método W

O conjunto state cover (Q set) é constituido por $n = |S|$ sequências de teste $a_i \in T$ que permitem alcançar cada um dos estados de M_S, ou seja, $Q_{set} = \{a_1, a_2, ...a_n\}, 1 \leq i \leq n$ tal que $\delta(s_0, a_i) = s_i, \forall s_i \in S$. Na Figura 18 pode ser vista a árvore de teste gerada com as sequências do Q_{set} para a MEF da Figura 17.

Consecutivamente, o conjunto transition cover (P set) pode ser formado a partir do Q set ao serem concatenadas todas as entradas válidas para cada estado alcançado com as

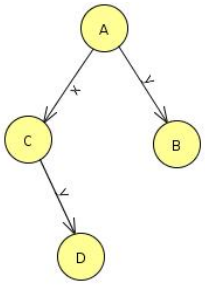

Figura 18: State Cover

sequência de Q_{set}, ou seja $P_{set} = \{b = a_i x,\ \text{tal que}, a_i \in Q_{set}, x \in I, \delta(\delta(s_0, a_i), x) = s_j, s_j \in S\}$. Na Figura 19 pode ser vista a árvore de teste gerada com as sequências do P_{set} para a MEF da Figura 17.

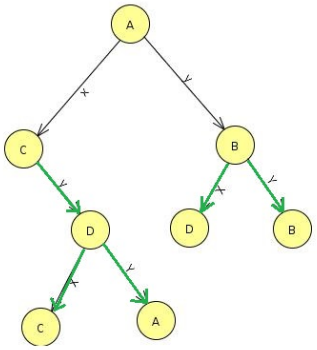

Figura 19: Transition Cover

Por ultimo, o conjunto W é concatenado ao conjunto transition cover P de modo cada um dos estados $s_j \in S$ alcançados sejam distinguidos dos demais estados de M_S. O conjunto de caracterização W é formado por sequências que permitem distinguir por meio das saídas obtidas cada par de estados $s, t \in S$, ou seja, $\exists z \in W_{set},$ tal que$\lambda(s, z) \neq \lambda(t, z)$. Em certos casos, W_{set} pode possuir mais de uma sequência de separação e, consequentemente, a sequência $a_i x \in P_{set}$ deverá ser aplicada mais de uma vez para cada $z \in W_{set}$. Na Figura 20 pode ser vista a árvore de teste gerada com as sequências do P_{set} concatenadas ao W_{set} para a MEF da Figura 17.

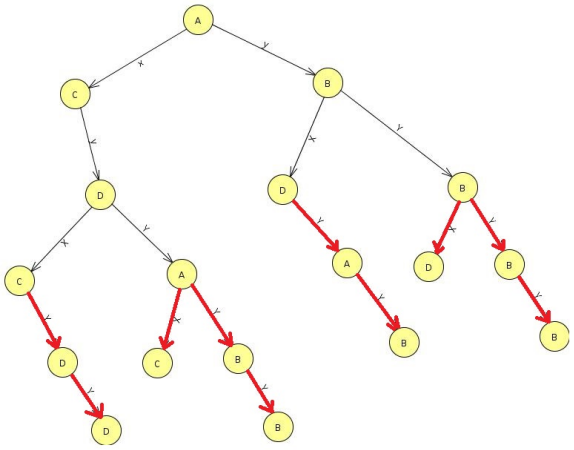

Figura 20: Metodo W -Final

Tabela de Símbolos

$\mapsto$ **Simbolo**	$\rightarrow$ **Significado**
S	Conjunto finito de estados
So	Estado inicial
T	Função de transição de estados
P	Conjunto de propriedades do estado
ℓ	Mapeamento entre um estado e suas propriedades
$\wedge$	Operador de disjunção
$\vee$	Operador de conjunção
$\neg$	Operador de negação ou inversão de valor lógico
$\mapsto$	Operador de implicação