

**MOCUSN: UM PROGRAMA PARA A DETERMINAÇÃO DE "PRIME IMPLICANTS"**

Antonio Souza Vieira Neto.

Divisão de Mecânica, Estrutura e Análise de Confiabilidade  
Instituto de Pesquisas Energéticas e Nucleares - IPEN-CNEN/SP  
São Paulo - SP

**Sumário**

Uma nova versão do programa MOCUS, capaz de determinar "prime implicants" de sistemas não-coerentes, é descrito e comparado com o programa WAMCUT.

**Abstract**

In this paper a new MOCUS algorithm for finding the prime implicants of non-coherent systems fault trees is described and compared with the WAMCUT algorithm.

# MOCUSN: UM PROGRAMA PARA A DETERMINAÇÃO DE "PRIME IMPLICANTS"

## 1. INTRODUÇÃO

A avaliação de características de confiabilidade de sistemas coerentes tem sido frequentemente efetuada a partir de cortes mínimos obtidos na análise de árvores de falhas. Diversos programas computacionais, tal como o MOCUS [1], têm sido utilizados com este objetivo. O método de cortes mínimos, no entanto, não se aplica a árvores de falhas de sistemas com estrutura não-coerente e, neste caso, deverão ser determinados os "prime implicants" [2] que fornecem todos os modos de falha do sistema, permitindo que a análise qualitativa e quantitativa do sistema seja efetuada.

Em particular, desenvolveu-se o programa MOCUSN, que é uma nova versão do programa MOCUS, para operar não apenas com estruturas coerentes, mas também com estruturas não-coerentes.

O programa MOCUSN determina os "prime implicants" de ordem igual ou inferior a um valor previamente estipulado pelo usuário. Este programa foi desenvolvido no Departamento de Tecnologia de Reatores do IPEN-CNEN/SP, no final de 1984.

## 2. PRINCIPAIS MODIFICAÇÕES INTRODUZIDAS NO PROGRAMA MOCUS

Embora a estrutura computacional do programa MOCUS tenha sido mantida, a construção do programa MOCUSN exigiu que, praticamente, todas as suas etapas de cálculo fossem alteradas ou ampliadas com a inclusão de novas subrotinas e arquivos.

As principais alterações implementadas no programa MOCUS, foram:

- a) introdução de novos algoritmos e adaptação dos já existentes para permitir a avaliação de árvores de falhas com representação booleana não-coerente;
- b) inclusão de comandos de leitura e gravação de dados que permitissem o acoplamento do programa MOCUSN ao programa RALLY-M [3];
- c) melhoria do relatório de saída, que passou a ser totalmente escrito em português e a contar com novas tabelas e com informações adicionais às tabelas já existentes.

## 3. DESCRIÇÃO DO PROGRAMA MOCUSN

Estruturalmente o programa MOCUSN é dividido em duas partes, uma de preparação de dados e outra de determinação dos "prime implicants". A parte de preparação de dados inclui a leitura e

análise da consistência dos dados, reestruturação da árvore de falhas e preparação de variáveis de entrada para o algoritmo de determinação dos "prime implicants". A tabela 1 apresenta a descrição das principais variáveis preparadas.

Tabela 1 Variáveis Definidas na Fase de Preparação de Dados

VARIÁVEL	DESCRIÇÃO
NG	Número de portões da árvore efetiva (portões "OR", "AND", "NAND", "NOR" e "NOT").
P	Indicador do portão lógico considerado. P = 1, 2 ... NG
IKC(P)	Número de entradas do portão P.
KC(P,J)	Indicador do J-ésimo elemento de entrada do portão P. J = 1, 2 ... IKC(P)
NROW	Indica o maior número de "prime implicants" que poderão existir sem que haja insuficiência de área de memória.

Nota : Uma árvore de falhas é dita efetiva após ter passado pelo processo de reestruturação.

### 3.1 Algoritmo de Determinação dos "Prime Implicants"

A determinação dos "prime implicants" é efetuada por um algoritmo analítico que se desenvolve no sentido do topo para a base da árvore de falhas (algoritmo do tipo "top down"). Este algoritmo considera além dos portões lógicos do tipo "AND" e "OR", permitíveis pelo programa MOCUS, os portões do tipo "NOT", "NOR" e "NAND".

O processo de cálculo deste algoritmo envolve uma série de procedimentos sobre portões lógicos em uma matriz indicadora de "prime implicants" (matriz ICS). Os portões do tipo "OR" e

"NAND" aumentam o número de linhas da matriz ICS e os do tipo "AND" e "NOR" o de colunas. Os portões do tipo "NOT" são usados apenas para indicar o complemento de eventos primários.

O algoritmo "top down" é iniciado ao se estabelecer que ICS(1,1) representa o portão "P" localizado imediatamente abaixo do evento que se deseja avaliar. A seguir, um dos algoritmos definidos na tabela 2 é aplicado sobre o portão "i" de acordo com sua função lógica. A partir daí o processo se repete para todos os portões que se situarem na primeira coluna da matriz ICS.

A aplicação dos algoritmos definidos na tabela 2 é efetuada até que existam apenas eventos primários na primeira coluna. Este processo se repete para as colunas subsequentes até que nenhum elemento da matriz ICS represente um portão lógico. As linhas de ICS representarão, então, os "prime implicants" da árvore de falhas. Este processo é intercalado por procedimentos de redução da matriz ICS, baseados nas leis de Boole e de De Morgan.

A determinação dos caminhos mínimos, caso sejam desejados, é feita da mesma maneira sobre a representação dual da árvore de falhas.

### 3.2 Características e Limitações Computacionais

A área total de memória utilizada pelo programa MOCUSN é dada pela expressão (1):

$$\text{AREA} = 113 + 8 \text{ NP} \div 1024 \text{ Kbytes,} \quad (1)$$

onde "NP" é o número de palavras de dupla precisão reservadas para as variáveis utilizadas pelo programa. Em geral, 20000 palavras são suficientes para a análise de árvores de falhas com menos de 100 eventos primários e que possuam menos de 1000 "prime implicants".

O programa MOCUSN é capaz, em uma única execução, de determinar os "prime implicants" de até 100 eventos intermediários da árvore, de usar até 100 eventos de acionamento ("house events") e processar árvores com até 2000 portões lógicos.

## 4. ANALISE COMPARATIVA ENTRE OS PROGRAMAS MOCUSN E WAMCUT

Antes de sua utilização na análise de confiabilidade de sistemas, o programa MOCUSN passou por uma série de testes de qualificação para avaliar a consistência dos algoritmos introduzidos. As soluções obtidas pelo programa MOCUSN foram comparadas com as obtidas por diferentes métodos, e em particular com as fornecidas por uma versão do programa WAMCUT [4] compatível com computadores IBM. Além das soluções, também foram comparados aspectos construtivos, metodológicos e o tempo

Tabela 2 Algoritmos para a Determinação de "Prime Implicants"

PORTÃO LÓGICO	ALGORITMO
"AND"	$\begin{aligned} \text{ICS}(\text{IX}, \text{IY}) &= \text{KC}(\text{P}, 1) \\ \text{ICS}(\text{IX}, \text{IT}(\text{IX}) + \text{I} - 1) &= \text{KC}(\text{P}, \text{I}) \\ \text{I} &= 2, \dots, \text{IKC}(\text{P}) \\ \text{IT}(\text{IX}) &= \text{IT}(\text{IX}) + \text{IKC}(\text{P}) - 1 \end{aligned}$
"NOR"	$\begin{aligned} \text{ICS}(\text{IX}, \text{IY}) &= \overline{\text{KC}(\text{P}, 1)} \\ \text{ICS}(\text{IX}, \text{IT}(\text{IX}) + \text{I} - 1) &= \overline{\text{KC}(\text{P}, \text{I})} \\ \text{I} &= 2, \dots, \text{IKC}(\text{P}) \\ \text{IT}(\text{IX}) &= \text{IT}(\text{IX}) + \text{IKC}(\text{P}) - 1 \end{aligned}$
"OR"	$\begin{aligned} \text{ICS}(\text{IX}, \text{IY}) &= \text{KC}(\text{P}, 1) \\ \text{ICS}(\text{MAX} + \text{J} - 1, \text{I}) &= \text{ICS}(\text{IX}, \text{I}) \quad \text{para } \text{I} \neq \text{IY} \\ &= \text{KC}(\text{P}, \text{J}) \quad \text{para } \text{I} = \text{IY} \\ \text{IT}(\text{MAX} + \text{J} - 1) &= \text{IT}(\text{IX}) \\ \text{I} &= 1, \dots, \text{IT}(\text{IX}) \\ \text{J} &= 2, \dots, \text{IKC}(\text{P}) \end{aligned}$
"NAND"	$\begin{aligned} \text{ICS}(\text{IX}, \text{IY}) &= \overline{\text{KC}(\text{P}, 1)} \\ \text{ICS}(\text{MAX} + \text{J} - 1, \text{I}) &= \overline{\text{ICS}(\text{IX}, \text{I})} \quad \text{para } \text{I} \neq \text{IY} \\ &= \overline{\text{KC}(\text{P}, \text{J})} \quad \text{para } \text{I} = \text{IY} \\ \text{IT}(\text{MAX} + \text{J} - 1) &= \text{IT}(\text{IX}) \\ \text{I} &= 1, \dots, \text{IT}(\text{IX}) \\ \text{J} &= 2, \dots, \text{IKC}(\text{P}) \end{aligned}$
"NOT"	$\text{ICS}(\text{IX}, \text{IY}) = \overline{\text{KC}(\text{P}, 1)}$

Onde: IX e IY indicam, respectivamente, a linha e a coluna ocupada pelo portão P na matriz ICS; MAX é o número de linhas da matriz ICS no instante da aplicação do algoritmo sobre P; IT(IX) é o número de elementos da linha IX da matriz ICS. A barra horizontal sobre um elemento (portão ou evento) indica o seu complemento. IKC(P) é igual a IKC(P̄) e KC(P,L) é igual a KC(P̄,L) para qualquer L.

de processamento destes programas.

#### 4.1 Comparação de Aspectos Construtivos e Metodológicos

Tanto o programa WAMCUT quanto o programa MOCUSN usam métodos analíticos na determinação de "prime implicants". Seus algoritmos, entretanto, desenvolvem-se em sentidos contrários na árvore de falhas. No programa WAMCUT as avaliações são iniciadas na base e terminam no topo da árvore (algoritmo do tipo "bottom up"), enquanto que no programa MOCUSN são processadas em sentido inverso (algoritmo do tipo "top down").

As diferenças entre os algoritmos dos dois programas permitem que sejam feitas as seguintes comparações:

- a) Durante a execução do algoritmo usado pelo programa WAMCUT, "prime implicants" de eventos intermediários podem ser obtidos diretamente, enquanto que, para isto, o algoritmo do programa MOCUSN tem que ser reiniciado tantas vezes quantos forem os eventos intermediários a serem analisados.
- b) O programa MOCUSN pode facilmente eliminar ramos da árvore através de eventos de acionamento ("house events"). Já o programa WAMCUT não pode eliminar ramos da árvore antes que seus "prime implicants" tenham sido determinados.

Os programas diferem, também, na maneira com que os "prime implicants" são armazenados. No programa WAMCUT o número de palavras ("NP") necessárias para o armazenamento dos "prime implicants" é dado pela expressão (2):

$$NP = 1 + \text{INT}(n \div 32) \quad (2)$$

onde, INT extraí o valor inteiro do termo entre parenteses e "n" é o número de eventos primários da árvore de falhas. Cada "bit" das "NP" palavras corresponde a um evento primário. Um "bit" com valor 1 indica que o evento primário associado faz parte de um "implicant" e os de valor zero não.

No programa MOCUSN, cada componente de um "implicant" ocupa a área de uma palavra.

As principais desvantagens computacionais da versão do programa WAMCUT analisado, em relação ao programa MOCUSN, estão relacionadas com a área de memória necessária, fixada em 1320 Kbytes, e com o dimensionamento interno que limita o programa a determina no máximo 2000 "prime implicants". O uso excessivo de memória pelo programa WAMCUT pode ser creditado a:

- a) problemas na adaptação do programa fonte para computadores IBM, uma vez que foi originalmente escrito para computadores CDC.
- b) abrangência de cálculos já que o programa também reserva

áreas de memória para efetuar estimativas aproximadas dos momentos de primeira e segunda ordem da indisponibilidade do sistema.

#### 4.2 Tempos de Processamento

Embora a análise efetuada por Rosenthal [5] tenha indicado que os algoritmos do tipo "botton up" geralmente são mais velozes do que os do tipo "top down", existem fatores ligados à implementação computacional que podem alterar esta tendência. Em vista disto, duas séries de 5 modelos, série S e série P, foram avaliadas pelos dois programas com a finalidade de comparar os seus tempos de processamento. As principais características destes modelos são apresentadas nas tabelas 3 e 4, respectivamente.

Numa primeira fase procurou-se comparar os tempos de processamento na determinação de todos os "prime implicants" de cada modelo. Nos modelos da série P, cujos tempos de processamento são apresentados na tabela 5, nota-se uma nítida vantagem do programa MOCUSN sobre o programa WAMCUT. Observa-se nestes casos que o tempo de processamento requerido pelo programa MOCUSN cresce linearmente com o tamanho da árvore (número de portões e eventos) enquanto que para o programa WAMCUT este tempo cresce exponencialmente.

Para os modelos da série S, o aumento exponencial do número de "prime implicants" e o crescimento de sua ordem máxima, fizeram com que o tempo de processamento requerido pelo programa MOCUSN tivesse um comportamento semelhante ao do programa WAMCUT, também aumentando exponencialmente. Nestes casos, o aumento foi tão acentuado que 60 minutos de processamento foram insuficientes para a obtenção dos "prime implicants" do modelo S5. Os tempos de processamento requeridos pelo programa WAMCUT limitaram-se aos modelos S1, S2 e S3, já que os modelos S4 e S5 possuem mais de 2000 "prime implicants", limite máximo dimensionado para o programa. A tabela 6 apresenta os tempos de processamento obtidos.

Na segunda fase da análise, a mesma comparação foi feita introduzindo-se uma limitação na ordem dos "prime implicants". Foram determinados os tempos de processamento para obter todos os "prime implicants" de ordem menor ou igual a  $j$  ( $j = 1, 2, \dots, 5$ ) para cada um dos modelos da série S. Nestes casos o programa MOCUSN apresentou-se mais veloz que o programa WAMCUT apenas na determinação dos "prime implicants" de ordem 1 (vide tabelas 7 e 8), indicando a forte dependência do programa MOCUSN com a ordem máxima dos "prime implicants" da árvore.

As análises comparativas foram feitas sob as seguintes condições:

Tabela 3 Características dos Modelos da Série S

CASO	NF	NC	NÚMERO DE "PRIME IMPLICANTS"					TOTAL GERAL
			ordem 1	ordem 2	ordem 3	ordem 4	ordem 5	
S1	20	41	2	87	16	0	0	105
S2	39	80	2	87	277	48	0	414
S3	58	119	2	87	277	831	144	1341
S4	77	158	2	87	277	831	2493	4122
S5	96	197	2	87	277	831	4122	4122

Tabela 4 Características dos Modelos da Série P

CASO	NF	NC	NÚMERO DE "PRIME IMPLICANTS"					TOTAL GERAL
			ordem 1	ordem 2	ordem 3	ordem 4	ordem 5	
P1	20	41	2	87	16	0	0	105
P2	39	81	2	174	32	0	0	209
P3	58	120	3	261	48	0	0	312
P4	77	159	4	348	64	0	0	416
P5	96	158	5	435	80	0	0	0

Notas: "NF" é o número de porções lógicas e "NC" é o número de eventos primários da árvore de falhas.

**Tabela 5** Tempos para Obtenção de Todos os "Prime Implicants" das Ávores do Grupo P

PROGRAMA	TEMPOS DE PROCESSAMENTO (s)				
	P1	P2	P3	P4	P5
WAMCUT	2,0	4,3	7,5	14,8	19,1
MOCUSN	1,5	2,8	4,8	7,4	10,6

**Tabela 6** Tempos para Obtenção de Todos os "Prime Implicants" das Árvores do Grupo S

PROGRAMA	TEMPOS DE PROCESSAMENTO (s)				
	S1	S2	S3	S4	S5
WAMCUT	2,0	6,6	63,0	-	-
MOCUSN	1,5	10,1	48,0	720,0	3600

Tabela 7 Tempos de Processamento Requeridos pelo Programa MOCUSN

MÁXIMA ORDEM CALCULADA	TEMPOS DE PROCESSAMENTO (s)				
	S1	S2	S3	S4	S5
1	1,3	1,8	3,0	3,7	4,4
2	2,2	5,4	7,9	8,5	9,4
3	2,2	10,7	49,4	67,5	68,4
4	-	40,7	77,0	437,0	648,7
5	-	-	77,2	711,0	1273,0

Tabela 8 Tempos de Processamento Requeridos pelo Programa WAMCUT

MÁXIMA ORDEM CALCULADA	TEMPOS DE PROCESSAMENTO (s)				
	S1	S2	S3	S4	S5
1	1,6	2,6	3,8	5,1	6,8
2	2,0	3,6	5,4	8,0	11,5
3	2,0	5,9	14,7	28,4	43,4
4	-	6,6	51,2	109,0	224,0
5	-	-	63,0	250,0	594,0

- a) os tempos de obtidos referem-se apenas ao de execução dos programas, em um computador IBM-4341, não incluindo os tempos de compilação e "linkedição";
- b) não foi considerado que o programa WAMCUT tem condições de eliminar os "implicants", cujo produto da indisponibilidade de seus componentes for menor que um valor pré-estabelecido pelo usuário. Esta decisão foi tomada principalmente pela dificuldade de se estabelecer critérios para a comparação com o programa MOCUSN.

## 5. CONCLUSÃO

O programa MOCUSN pode ser utilizado na análise de confiabilidade de sistemas que apresentam estruturas não-coerentes. Para árvores que contenham "prime implicants" com ordem inferior a 5 o programa MOCUSN pode ser usado com vantagens sobre o programa WAMCUT.

## 6. BIBLIOGRAFIA

- [1] FUSSEL, J. B.; HENRY, E. B.; MARSHALL, N. H. MOCUS: a computer program to obtain minimal sets from fault trees. Idaho Falls, Idaho, Aerojet Nuclear Company, Mar. 1974. (ANCR-1156).
- [2] JOHNSTON, B. D. & MATTHEWS, R. H. Noncoherent structure theory: a review and its role in fault tree analysis, Safety and Reliability Directorate, Oct. 1983. (SRD R 245).
- [3] VIEIRA Neto, A. S. Análise e modificações do programa RALLY, usado na avaliação de confiabilidade de sistemas. Dissertação de mestrado a ser apresentada em 1987. Instituto de Pesquisas Energéticas e Nucleares. São Paulo - SP.
- [4] ERDMANN, R.C.; LEVERENZ, F. L.; KIRCH, H. WAMCUT: a computer code for fault tree evaluation. Electric Power Reserch Institute. Jun. 1978. (EPRI-NP-803).
- [5] ROSENTHAL, A. Approaches to comparing cut-set enumeration algorithms, IEEE Trans. Reliability, vol R-28(1):62-5. Apr. 1979.