Minimização de funções booleanas - parte II

Última revisão em 04 de abril de 2016

Este documento é parte das notas de aula da disciplina MAC0329 (Álgebra Booleana e Circuitos Digitais). Nesta parte continuaremos a abordar a minimização de funções boolenas e, mais especificamente, a minimização lógica dois-níveis.

1 Revisão de alguns conceitos

Para a leitura deste documento é conveniente relembrarmos alguns conceitos:

Função lógica de n variáveis: qualquer função binária definida em n variáveis binárias, isto é, uma função da forma $f : \{0,1\}^n \to \{0,1\}$

Produtos. Um produto é uma expressão booleana que consiste de uma conjunção de literais, não envolvendo uma mesma variável mais de uma vez. Em particular, o produto canônico (ou mintermo) é um produto em que cada variável ocorre exatamente uma vez, ou na forma barrada ou na forma não-barrada. Um produto canônico em n variáveis toma valor 1 em apenas um elemento do conjunto $\{0,1\}^n$.

Por exemplo, dadas três variáveis a, b e c, os produtos abc e $\overline{a}bc$ são canônicos. A disjunção deles, $abc + \overline{a}bc = (a + \overline{a})bc = bc$, é também um produto, porém não é canônico. Observe que o produto bc toma valor 1 para os elementos 011 e 111; o valor da variável a não afeta o valor desse produto.

Somas. Dual aos produtos. A soma $a + \overline{b} + c$ toma valor zero apenas quando $a = 0$, $b = 1$ e $c = 0$.

Expressão na forma SOP e POS: qualquer função lógica pode ser expressa como soma de produtos canônicos ou como produto de somas canônicas. As expressões nessas formas podem ser obtidas, por exemplo, diretamente da tabela-verdade da função.

Mintermo e maxtermo: correspondem a, respectivamente, produto e soma canônicos

Implicantes. Dadas duas funções f e g, dizemos que f implica g se $f(\mathbf{x}) = 1 \implies g(\mathbf{x}) = 1, \forall \mathbf{x} \in \{0,1\}^n$. Em particular, qualquer produto p que faz parte da forma SOP de uma função f implica f e é denominado "implicante de f".

Cubos. Na função $f(a,b,c) = abc + \overline{a}bc$, os mintermos na notação compacta são m_7 (pois $111_{(2)} = 7_{(10)}$) e m_3 (pois $011_{(2)} = 3_{(10)}$). Assim, é usual escrevermos $f(a,b,c) = \sum m(3,7)$. Uma vez que existe uma correspondência um-para-um entre os mintermos em n variáveis e os elementos de $\{0,1\}^n$, uma função expressa como soma de mintermos pode ser vista como um subconjunto de $\{0,1\}^n$.

Soma de mintermos	subconjunto de $\{0,1\}^3$
$\overline{a}bc$	$\{011\}$
abc	$\{111\}$
$\overline{a}bc + abc$	$\{011, 111\}$

Os elementos de $\{0,1\}^n$ (aqui denotados como sequências ou *strings* de n dígitos binários) podem ser vistos como pontos no n-espaço. A coleção dos 2^n elementos de $\{0,1\}^n$ forma os vértices de um hipercubo. A figura 1 mostra um hipercubo no espaço de dimensão 3.

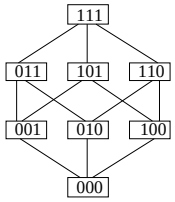

Figura 1: Um hipercubo de dimensão 3. (que feio, preciso melhorar isso!)

No contexto de circuitos lógicos, hipercubos são denominados **n-cubos**. Os vértices de um n-cubo são denominados 0-cubos. Dois 0-cubos formam um 1-cubo se eles diferem em apenas uma coordenada. Quatro 0-cubos formam um 2-cubo se eles são iguais a menos de duas coordenadas. De modo geral, 2^k 0-cubos formam um k-cubo se eles são exatamente iguais a menos de k coordenadas.

Intervalo. Observe que cubos não são subconjuntos arbitrários de $\{0,1\}^n$. Um cubo é um conjunto de elementos em $\{0,1\}^n$ para os quais um produto toma valor 1, i.e., se p é um produto então o cubo correspondente a p é o conjunto $p\langle 1\rangle = \{\mathbf{x} \in \{0,1\}^n : p(\mathbf{x}) = 1\}$.

Cubos podem também ser vistos como **intervalos**. Um intervalo no n-cubo é caracterizado por dois extremos: o menor e o maior elementos contidos nele. Assim, no cubo $\{0,1\}^3$, o intervalo de extremo inferior 100 e extremo superior 101 é denotado $[100, 101]$ e definido[1] por $[100, 101] = \{\mathbf{x} \in \{0,1\}^3 : 100 \leq \mathbf{x} \leq 101\}$.

[1] A relação $\leq$ usada na definição de intervalos não é a relação usual entre números. Por exemplo, "100" $\leq$ "101". Mas "011" não é menor que "101"; estes dois não são comparáveis. Formalmente, definimos que $(a_1, a_2, \ldots, a_n) \leq (b_1, b_2, \ldots, b_n)$ se, e somente se, $a_i \leq b_i, \forall i = 1, 2, \ldots, n$.

Denotamos um k-cubo ou intervalo de dimensão k colocando um X nas coordenadas que não são iguais. Assim, no caso de três variáveis a, b e c, o 1-cubo $\{000, 100\}$ (ou, equivalentemente, o intervalo $[000, 100]$), que correponde ao produto $\overline{b}\,\overline{c}$, é representado por $X00$. O 2-cubo $\{000, 001, 100, 101\}$ (ou, equivalentemente, o intervalo $[000, 101]$), que corresponde ao produto $\overline{b}$, é representado por $X0X$. Um intervalo contém necessariamente 2^k elementos, onde $0 \leq k \leq n$. Quanto maior a dimensão de um cubo, menor o número de literais presentes no correspondente produto.

Exemplo: A figura 2 mostra alguns cubos. Dizemos que o 0-cubo 000 está *contido* no (ou é coberto pelo) 1-cubo $X00$, ou ainda, que o 1-cubo $X00$ *cobre* o 0-cubo 000. Analogamente, dizemos que o 1-cubo $X00$ está contido no 2-cubo $X0X$ ou que o 2-cubo $X0X$ cobre o cubo $X00$.

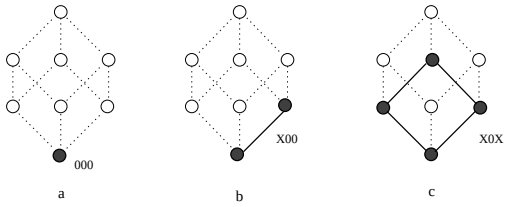

Figura 2: Os cubos 000, $X00$ e $X0X$.

Produto, implicante, cubo, e intervalo. Esses termos são equivalentes. Um produto canônico (também chamado mintermo), é equivalente a 0-cubo ou intervalo trivial. Por exemplo, dadas três variáveis lógicas, a, b e c, o produto canônico $a\,\overline{b}\,c$ corresponde ao 0-cubo 101, ou ao intervalo $[101, 101]$. Já o produto (não canônico) $a\,\overline{b}$ corresponde ao 1-cubo $10X$ ou ao intervalo $[100, 101]$.

Implicantes primos. Um produto ou implicante p de f é maximal se não existe outro produto ou implicante p' de f tal que p seja também implicante de p'. Implicantes maximais são denominados implicantes primos.

No caso do mapa de Karnaugh, os agrupamentos de 1's que construímos no mapa são implicantes primos (produtos maximais).

2 Minimização Tabular de Quine-McCluskey

Definição: Uma expressão lógica escrita na forma soma de produtos é **minimal** se

1. não existe nenhuma outra expressão equivalente[2] na forma soma de produtos com um número menor de termos, e

2. não existe nenhuma outra expressão equivalente na forma soma de produtos com igual número de termos mas com menor número de literais.

Denominaremos tal expressão de **SOP minimal**[3]. Ao se remover, de uma forma SOP minimal, um produto ou um literal de qualquer um dos produtos, a expressão resultante não mais representa a mesma função.

O problema pode ser analogamente definido para a forma POS. O problema de encontrar a forma SOP ou POS minimal de uma função é denominada **minimização lógica dois-níveis** devido ao fato de funções na forma SOP ou POS poderem ser realizadas em circuitos de dois níveis (isto é, no caso da forma SOP, o primeiro nível é formado pelas portas E, uma para cada produto, e o segundo nível consiste de uma porta OU que recebe como entradas as saídas das portas E).

Mapas de Karnaugh, vistos no outro documento, representam uma maneira visual e intuitiva de se minimizar funções booleanas. No entanto, eles só se aplicam a funções com até 6 variáveis e não são sistemáticos (adequados para programação). O algoritmo tabular de Quine-McCluskey para minimização de funções Booleanas é um método clássico que sistematiza este processo de minimização para um número arbitrário de variáveis.

Tanto os mapas de Karnaugh como o algoritmo de Quine-McCluskey (QM), em sua formulação clássica, requerem que a função booleana a ser minimizada esteja na forma SOP canônica. A idéia básica do algoritmo QM consiste em encarar os mintermos da SOP canônica como pontos no n-espaço, ou seja, como vértices de um n-cubo. A partir do conjunto desses vértices (ou 0-cubos) procura-se gerar todos os 1-cubos possíveis combinando-se dois deles (equivale a gerar as arestas do cubo que ligam dois 0-cubos da função). A partir da combinação de dois 1-cubos procura-se gerar todos os possíveis 2-cubos e assim por diante, até que nenhum cubo de dimensão maior possa ser gerado a partir da combinação de dois cubos de dimensão menor. Os cubos resultantes (aqueles que não foram combinados com nenhum outro) ao final de todo o processo são os **implicantes primos** (ou seja, cubos maximais) da função.

Esse processo de combinar dois cubos pode ser facilmente associado ao processo algébrico de simplificação. Os mintermos da expressão na forma canônica inicial corres-

[2]Duas expressões são equivalentes se definem uma mesma função.
[3]SOP é contração de *Sum of products*

pondem aos 0-cubos. Combinar dois 0-cubos para gerar um 1-cubo corresponde a combinar dois mintermos para eliminar uma variável e gerar um termo com menos literais para substituí-los, como mostramos no seguinte exemplo :

$$x_1x_2x_3 + x_1x_2\overline{x}_3 = x_1x_2(x_3 + \overline{x}_3) = x_1x_2 \cdot 1 = x_1x_2$$

Quando considerados no 3-espaço, o processo mostrado na expressão algébrica acima corresponde ao processo de agruparmos os 0-cubos 111 e 110 para geração do 1-cubo $11X$, como ilustra a figura 3.

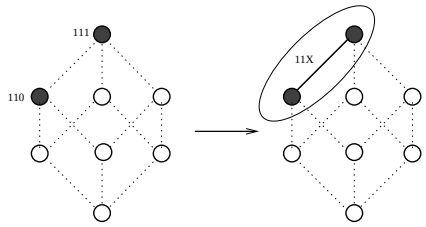

Figura 3: Passo elementar do algoritmo de Quine-McCluskey

À primeira vista, poderíamos afirmar que a soma de todos os implicantes primos corresponde à expressão minimal da função Booleana. No entanto, existem casos em que a soma de dois ou mais implicantes primos cobre um outro. Neste caso, este último termo é redundante, no sentido de que ele pode ser eliminado do conjunto de implicantes primos, sem que a expressão resultante deixe de ser equivalente à expressão original. Podemos ilustrar esta situação no seguinte exemplo.

Exemplo: Considere a expressão Booleana $f(a, b, c) = \sum m(0, 1, 3, 7)$. Os implicantes primos dessa função são $00X$, $0X1$ e $X11$ (calcule usando o mapa de Karnaugh). Graficamente, estes implicantes primos (ou cubos) correspondem respectivamente aos intervalos $[000, 001]$, $[001, 011]$ e $[011, 111]$ ilustrados na figura 4(a). Note, porém, que o

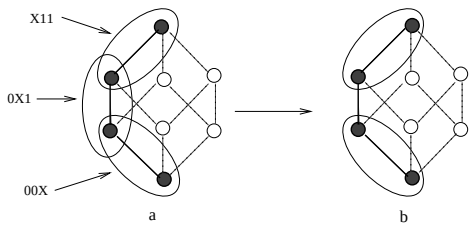

Figura 4: Os (a) implicantes primos e uma (b) cobertura mínima .

intervalo $[001, 011]$ é redundante, ou seja, a mesma expressão pode ser expressa apenas pelos implicantes primos $00X$ e $X11$ (figura 4(b)).

Teorema: Qualquer produto em uma expressão minimal na forma soma de produtos é um implicante primo.

Dem.: A prova deste teorema é simples. Suponha que exista algum produto p na expressão que não seja um implicante primo. Por definição, existe um produto p' tal que $p < p'$ e tal que p' implica a função. Então, ao substituirmos p por p' na expressão, obtemos uma expressão equivalente, porém com custo menor. Isto contradiz com o fato de que a expressão era minimal. □

Este teorema diz, em outra palavras, que para encontrarmos uma expressão minimal de uma função, basta considerarmos apenas os produtos que são implicantes primos da função. Porém, como nem todos os implicantes primos de uma função f fazem parte da forma SOP minimal de f, é ainda necessário selecionar o "menor subconjunto" de implicantes primos suficientes para expressar f. Portanto, um procedimento para obter a forma SOP minimal de uma função pode ser:

1. Calcular todos os implicantes primos da função
2. Calcular uma cobertura mínima dos 1's da função (menor subconjunto de implicantes primos cuja soma representa a função)

O ponto central da segunda etapa é o cálculo de um menor subconjunto do conjunto de implicantes primos suficientes para cobrir[4] todos os mintermos da função. Tal conjunto é denominado uma **cobertura mínima**.

No caso de mapas de Karnaugh, estas duas etapas são realizadas conjuntamente de forma um tanto "intuitiva". No caso do algoritmo QM, estas etapas são realizadas explícita e separadamente. As etapas são descritas a seguir, por meio de um exemplo.

2.1 Cálculo de implicantes primos

A primeira etapa do algoritmo QM consiste de um processo para determinação de todos os implicantes primos. A seguir descrevemos os passos que constituem esta etapa, mostrando

[4]Um conjunto de implicantes primos (cubos maximais) cobre um mintermo (0-cubo) se este é coberto por pelo menos um dos implicantes primos.

como exemplo o cálculo dos implicantes primos da função $f(x_1, x_2, x_3) = \sum m(0, 1, 4, 5, 6)$.

- Primeiro passo : converter os mintermos para a notação binária.

  000, 001, 100, 101, 110

- Segundo passo : Separar os mintermos em grupos de acordo com o número de 1's em sua representação binária e ordená-los em ordem crescente, em uma coluna, separando os grupos com uma linha horizontal.

000
001
100
101
110

- Terceiro passo : combinar todos os elementos de um grupo com todos os elementos do grupo adjacente inferior para geração de cubos de dimensão maior. Para cada 2 grupos comparados entre si, gerar um novo grupo na próxima coluna e colocar os novos cubos. Marcar com $\surd$ os cubos que foram usados para gerar novos cubos.

$\surd$	000
$\surd$	001
$\surd$	100
$\surd$	101
$\surd$	110

$\Longrightarrow$

$00X$
$X00$
$X01$
$10X$
$1X0$

  Observação : o novo cubo gerado será inserido no novo conjunto se e somente se ele ainda não estiver contido nele.

  Repetir o processo para cada nova coluna formada, até que nenhuma combinação mais seja possível.

$\surd$	000
$\surd$	001
$\surd$	100
$\surd$	101
$\surd$	110

$\Longrightarrow$

$\surd$	$00X$
$\surd$	$X00$
$\surd$	$X01$
$\surd$	$10X$
	$1X0$

$\Longrightarrow$ $X0X$

- Quarto passo : Listar os implicantes primos. Os implicantes primos são os cubos que não foram combinados com nenhum outro, ou seja, aqueles que não estão com a marca $\surd$.

  $1X0$ e $X0X$

2.2 Cálculo de uma cobertura mínima

Uma cobertura mínima pode ser calculada com a ajuda de uma tabela denominada **Tabela de Implicantes Primos**. Este processo é mostrado a seguir, usando como exemplo a minimização da função $f(x_1, x_2, x_3, x_4, x_5) = \sum(1, 2, 3, 5, 9, 10, 11, 18, 19, 20, 21, 23, 25, 26, 27))$.

1. Construir a Tabela de Implicantes Primos: No topo das colunas desta tabela deve-se colocar os mintermos de f e, à esquerda de cada linha, os implicantes primos. Os implicantes primos devem ser listados em ordem decrescente de acordo com a sua dimensão, isto é, em ordem crescente de acordo com o número de literais. Deve-se acrescentar uma coluna à esquerda e uma linha na parte inferior da tabela.

   Em cada linha, marcar as colunas com $\surd$ quando o implicante primo da linha cobre o mintermo da coluna.

		1	2	3	5	9	10	11	18	19	20	21	23	25	26	27
	XX01X		$\surd$	$\surd$			$\surd$	$\surd$	$\surd$	$\surd$					$\surd$	$\surd$
	X10X1					$\surd$		$\surd$						$\surd$		$\surd$
	0X0X1	$\surd$		$\surd$		$\surd$		$\surd$								
	00X01	$\surd$			$\surd$											
	X0101				$\surd$							$\surd$				
	1010X										$\surd$	$\surd$				
	10X11									$\surd$			$\surd$			
	101X1											$\surd$	$\surd$			

2. Selecionar os implicantes primos essenciais: deve-se procurar na tabela as colunas que contém apenas uma marca $\surd$. A linha na qual uma dessas colunas contém a marca $\surd$ corresponde a um implicante primo essencial. Em outras palavras, este implicante primo é o único que cobre o mintermo da coluna e, portanto, não pode ser descartado. Então, deve-se marcar com um asterisco (*) esta linha na coluna mais à esquerda, para indicar que este é um implicante primo essencial. A seguir, deve-se marcar, na última linha da tabela, todas as colunas cujo mintermo é coberto pelo implicante primo selecionado.

   No exemplo, o mintermo 2 é coberto apenas pelo implicante primo $XX01X$. Logo $XX01X$ é essencial.

		1	2	3	5	9	10	11	18	19	20	21	23	25	26	27
*	XX01X		√	√			√	√	√	√					√	√
	X10X1					√		√						√		√
	0X0X1	√		√		√		√								
	00X01	√			√											
	X0101				√							√				
	1010X										√	√				
	10X11									√			√			
	101X1											√	√			
			√	√			√	√	√	√					√	√

A linha correspondente a um implicante primo essencial, bem como as colunas cujos mintermos são cobertos por esse implicante primo, devem ser descondirados no prosseguimento do processo.

Deve-se repetir o processo enquanto existir, na tabela restante, algum implicante primo essencial.

No exemplo, vemos que o mintermo 25 é coberto apenas pelo implicante primo $X10X1$ e que o mintermo 20 é coberto apenas pelo implicante primo $1010X$. Logo, esses dois implicantes primos também são essenciais.

		1	2	3	5	9	10	11	18	19	20	21	23	25	26	27
*	XX01X		√	√			√	√	√	√					√	√
*	X10X1					√		√						√		√
	0X0X1	√		√		√		√								
	00X01	√			√											
	X0101				√							√				
*	1010X										√	√				
	10X11									√			√			
	101X1											√	√			
			√	√		√	√	√	√	√	√	√		√	√	√

3. Reduzir a tabela: eliminar as colunas cujos mintermos já foram cobertos (ou seja, manter apenas as colunas correspondentes aos mintermos não cobertos pelos implicantes primos essenciais). Eliminar as linhas correspondentes aos implicantes primos essenciais e as linhas que não cobrem nenhum dos mintermos restantes na tabela.

No exemplo, após a redução, temos a seguinte tabela:

		1	5	23
	0X0X1	√		
	00X01	√	√	
	X0101		√	
	10X11			√
	101X1			√

4. Selecionar os implicantes primos secundariamente essenciais: eliminar as linhas dominadas e as colunas dominantes e, em seguida, selecionar os essenciais.

Colunas dominantes: Diz-se que uma coluna β na Tabela de Implicantes Primos domina uma coluna α se e somente se todos os implicantes que cobrem o mintermo da coluna α cobrem também o mintermo da coluna β. Se β domina α, então a coluna β pode ser removida da tabela. (Por quê?)

Linhas dominadas ou equivalentes: Sejam A e B duas linhas na Tabela de Implicantes Primos reduzida. Então dizemos que a linha A domina B se o implicante da linha A cobre, ao menos, todos os mintermos cobertos pelo implicante da linha B. Dizemos que as linhas A e B são equivalentes se os respectivos implicantes primos cobrem exatamente os mesmos mintermos na tabela. Se A domina B, ou se A e B são equivalentes, e **se o custo do implicante da linha A é menor ou igual ao da linha** B (ou seja, o implicante da linha A possui não mais literais que o implicante da linha B), então a linha B pode ser eliminada da tabela (Por quê?).

Após a eliminação de colunas dominantes e linhas dominadas (ou equivalentes), deve-se repetir o mesmo processo do passo 2, porém os implicantes primos essenciais serão chamados secundariamente essenciais e marcados com dois asteriscos (**).

No exemplo, a linha do implicante primo $X0101$ pode ser eliminada pois é dominada pela linha do implicante $00X01$. A linha do implicante $101X1$ pode ser eliminada pois é equivalente a do implicante $10X11$. Neste último caso, note que, alternativamente, podemos eliminar a linha do implicante $10X11$ em vez da linha do implicante $101X1$.

		1	5	23
	0X0X1	√		
**	00X01	√	√	
**	10X11			√
		√	√	√

Deve-se repetir o processo descrito neste passo até que não seja mais possível fazer qualquer eliminação ou até que a tabela fique vazia. Se a tabela não ficar vazia, a tabela restante é chamada **tabela cíclica**.

5. Resolver a tabela cíclica: Para isso, uma possível abordagem é o método de Petrick, um método de busca exaustiva. Ele fornece todas as possíveis combinações dos implicantes primos restantes que são suficientes para cobrir os mintermos ainda não cobertos pelos implicantes primos já selecionados. Deve-se escolher, dentre essas combinações, uma que envolve o menor número de termos. Caso existam mais de uma nestas condições, deve-se escolher a de custo mínimo (aquela que envolve menor número de literais).

   Exemplo: Considere a tabela cíclica a seguir:

		0	4	13	15	10	26	16
a	0X10X		√	√				
b	011XX			√	√			
c	01X1X				√	√		
d	1X0X0						√	√
e	00X00	√	√					
f	X1010					√	√	
g	X0000	√						√

   Para que todos os mintermos da tabela cíclica sejam cobertos, a seguinte expressão deve ser verdadeira.

$$(e+g)(a+e)(a+b)(b+c)(c+f)(d+f)(d+g) = 1$$

   Transformando esta expressão em soma de produtos, obtemos todas as possíveis soluções (cada produto é uma solução viável). Dentre os produtos, deve-se escolher aquele(s) de menor custo (menor número de implicantes primos e implicantes com menor número de literais). (Se eu não errei nos cálculos, as soluções são $\{a, c, d, e\}$, $\{b, c, d, e\}$ e $\{a, c, d, g\}$ pois os outros tem custo maior).

   Outro exemplo: Considere a tabela cíclica a seguir e suponha que o custo de A é menor que o de B e que o custo de C é menor que o de D.

	m_1	m_2	m_3
A	√		
B	√	√	
C			√
D		√	√

Então as possíveis soluções são $(A + B)(B + D)(C + D) = (AB + AD + B + BD)(C + D) = (B + AD)(C + D) = BC + BD + ACD + AD$. Dos que envolvem dois implicantes, certamente o custos de BC e de AD são menores que o custo de BD. Então a escolha final fica entre BC e AD.

Resumo do Procedimento para cálculo de cobertura mínima:

1. Montar a tabela de implicantes primos

2. Identificar todos os implicantes primos essenciais e eliminar as linhas correspondentes, bem como as colunas dos mintermos cobertos por esses implicantes.

3. Eliminar colunas dominantes: Se uma coluna β tem $\surd$ em todas as linhas que uma outra coluna α tem $\surd$, a coluna β é dominante e pode ser eliminada (pois se escolhermos um implicante primo que cobre α, β será necessariamente coberto também).

4. Eliminar linhas dominadas ou equivalentes: se uma linha A tem $\surd$ em todas as colunas em que a linha B tem $\surd$, então a linha A domina a linha B. Se elas tem $\surd$ exatamente nas mesmas colunas, então elas são equivalentes. Se, além disso, o número de literais de A é menor que o de B, então a linha B pode ser eliminada (pois se tivéssemos uma cobertura envolvendo B, ao trocarmos B por A na cobertura teríamos uma cobertura de menor custo).

   Observação: Se o objetivo da minimização é encontrar apenas UMA solução minimal (e NÃO TODAS), então podemos eliminar uma linha B se existe uma linha A tal que A domina B, ou A é equivalente a B, e ambos têm um mesmo custo.

5. Identificar os implicantes essenciais secundários e eliminar as linhas correspondentes, bem como as colunas dos mintermos cobertos por esses implicantes.

6. Repetir 3, 4 e 5 enquanto possível

7. Se a tabela não estiver vazia, aplicar o método de Petrick (que lista todas as possíveis soluções para o restante da tabela) e escolher uma solução de custo mínimo.

Exemplo: Considere a função $f(a, b, c) = a\,\overline{b}\,c + \overline{a}\,b\,\overline{c} + a\,b\,\overline{c} + a\,b\,c = \sum m(2, 5, 6, 7)$.

Podemos realizar a simplificação algébrica da seguinte forma:

$$\begin{aligned} f(a,b,c) &= a\,\overline{b}\,c + \overline{a}\,b\,\overline{c} + a\,b\,\overline{c} + a\,b\,c \\ &= a\,\overline{b}\,c + a\,b\,c + \overline{a}\,b\,\overline{c} + a\,b\,\overline{c} + a\,b\,\overline{c} + a\,b\,c \\ &= a\,c + b\,\overline{c} + a\,b \\ &= a\,c + b\,\overline{c} \end{aligned}$$

Por QM temos

√	010
√	101
√	110
√	111

$\Longrightarrow$

$X10$
$1X1$
$11X$

Os implicantes primos são $X10$ ($b\,\overline{c}$), $1X1$ ($a\,c$) e $11X$ ($a\,b$). Uma cobertura mínima pode ser calculada usando-se a Tabela de Implicantes Primos.

		2	5	6	7
*	X10	√		√	
*	1X1		√		√
	11X			√	√
		√	√	√	√

Os implicantes primos $X10$ e $1X1$ são essencias e cobrem todos os mintermos da função. Logo formam uma cobertura mínima.

2.3 Funções incompletamente especificadas

Em algumas situações, o valor de uma função para algumas entradas não são relevantes (tipicamente porque tais entradas nunca ocorrerão na prática). Em tais situações, tanto faz se a função toma valor 0 ou 1 nessas entradas, que serão denominadas de **don't cares**.

Para minimizar uma função incompletamente especificada pelo algoritmo QM, é interessante considerarmos todos os don't cares na etapa de cálculo dos implicantes primos, pois isto aumenta a chance de obtermos cubos maiores (portanto produtos com menos literais). Observe que, durante as iterações para a geração dos implicantes primos, um

cubo que cobre apenas don't cares não deve ser eliminado pois ele pode, eventualmente em iterações futuras, se juntar a outro cubo para formar outro cubo maior.

De forma mais genérica do que a vista anteriormente, podemos caracterizar uma função booleana f através dos seus conjuntos um, zero e dc (de don't care), definidos respectivamente por $f\langle 1\rangle = \{\mathbf{x} \in \{0,1\}^n : f(\mathbf{x}) = 1\}$, $f\langle 0\rangle = \{\mathbf{x} \in \{0,1\}^n : f(\mathbf{x}) = 0\}$ e $f\langle *\rangle = \{0,1\}^n \setminus (f\langle 1\rangle \cup f\langle 0\rangle)$.

Na parte de cálculo dos implicantes primos devem ser utilizados todos os elementos de $f\langle 1\rangle \cup f\langle *\rangle$. Na parte de cálculo de uma cobertura mínima devem ser considerados apenas os elementos de $f\langle 1\rangle$.

2.4 Cálculo da forma POS minimal

Similarmente ao que já vimos na minimização por mapas de Karnaugh, para se obter a forma POS minimal de uma função por QM procede-se da seguinte forma. No cálculo dos implicantes primos, em vez de listar os mintermos, listamos os 0s da função e aplicamos o método tabular (a primeira parte do algoritmo QM). Em seguida, realizamos o cálculo da cobertura mínima utilizando-se nas colunas os 0s da função. Ao final, expressa-se o resultado como produto dos implicantes primos selecionados complementados.

A explicação é a seguinte: ao tomarmos os 0s da função em vez dos 1s, estamos considerando a minimização SOP de $\overline{f}$. Agora, uma vez que $f = \overline{\overline{f}}$, ao complementarmos a forma SOP minimal de $\overline{f}$, obtemos a forma POS minimal de f.

Exemplo: Minimizar na forma POS a função $f(a,b,c) = \prod M(3,6,7)$. Algebricamente temos:

$$\begin{aligned} f(a,b,c) &= (a+\overline{b}+\overline{c})(\overline{a}+\overline{b}+c)(\overline{a}+\overline{b}+\overline{c}) \\ &= (a+\overline{b}+\overline{c})(\overline{a}+\overline{b}+\overline{c})(\overline{a}+\overline{b}+c)(\overline{a}+\overline{b}+\overline{c}) \\ &= (a\,\overline{a}+\overline{b}+\overline{c})(\overline{a}+\overline{b}+c\,\overline{c}) \\ &= (\overline{b}+\overline{c})(\overline{a}+\overline{b}) \end{aligned}$$

Por QM temos:

√	011
√	110
√	111

$\Longrightarrow$

$X11$
$11X$

Os implicantes são $X11$ e $11X$, que escritos na forma de produtos correspondem respectivamente a bc e ab. Complementando estes produtos temos: $\overline{b}+\overline{c}$ e $\overline{a}+\overline{b}$, que são as somas que aparecem na forma POS minimal.

3 Minimização dois-níveis de múltiplas funções*

No outro documento vimos brevemente a ideia de compartilhar portas E quando existem duas ou mais funções a serem realizadas, definidas sobre as mesmas variáveis. Vimos também que, nessas condições, não necessariamente a minimização individual das funções leva a uma melhor solução. Isto é, há situações nas quais minimizar individualmente as funções impede o compartilhamento de portas E, embora existam produtos que sejam implicantes de mais de uma função.

Quando pensamos em minimizar um conjunto de funções, diferentes critérios podem ser considerados. Entre eles, alguns são intuitivos:

- reduzir o número total de produtos (ou seja, portas E).
- reduzir o número de entradas para as portas E (tentar usar produtos com o menor número possível de literais)
- reduzir o número de entradas para as portas OU (ou seja, utilizar o menor número possível de produtos para cada função)

Quando consideramos a minimização de múltiplas funções, podemos aplicar um processo análogo ao do algoritmo QM. A noção de implicante primo é agora definida com relação às múltiplas funções. Devemos considerar não apenas os implicantes primos de cada uma das funções, mas também todos os implicantes maximais que podem ser compartilhados por 2 ou mais funções. Vamos esclarecer isso analisando um exemplo concreto.

Exemplo 3: Considere as funções, escritas na forma SOP canônica

$$f_1(x_1,x_2,x_3)=\overline{x}_1\overline{x}_2\overline{x}_3+\overline{x}_1\overline{x}_2x_3+x_1\overline{x}_2x_3=\sum m(0,1,5)$$

$$f_2(x_1, x_2, x_3) = x_1\overline{x}_2x_3 + x_1x_2\overline{x}_3 + x_1x_2x_3 = \sum m(5, 6, 7)$$

Escritos na notação cúbica, os mintermos de f_1 são 000, 001 e 101 enquanto os de f_2 são 101, 110 e 111. Os implicantes primos (com respeito a f_1 e f_2) são:

a) os implicantes primos de f_1: $00X$ e $X01$

b) os implicantes primos de f_2: $1X1$ e $11X$

c) os implicantes de f_1 e de f_2 que não são cobertos por outro implicante de f_1 e de f_2: 101 (ou seja, 101 é implicante (não é primo!) de ambas as funções e não existe nenhum outro implicante de f_1 e de f_2 que o cobre).

Na minimização de múltiplas funções, compartilhar produtos é uma forma de se minimizar o número de portas E no circuito. Os implicantes do tipo (c) são, em outras palavras, os maiores subcubos compartilhados entre duas ou mais funções. Portanto, o critério de minimização deve levar em consideração não apenas os implicantes primos de cada função, mas também todos os implicantes do tipo do item (c).

O algoritmo QM adaptado para o cálculo de implicantes primos de múltiplas funções

Considere novamente as funções do exemplo 1: $f_1(x_1, x_2, x_3) = \sum m(0, 1, 5)$ e $f_2(x_1, x_2, x_3) = \sum m(5, 6, 7)$. A tabela 1 mostra o cálculo dos implicantes primos de $\mathbf{f} = (f_1, f_2)$. Inicialmente, listam-se todos os mintermos (independente da função a qual eles pertencem), um em cada linha, agrupados de acordo com o número de ocorrência de 1s. Ao lado, cria-se uma coluna para cada uma das funções e, para cada coluna, marcam-se as linhas correspondentes aos mintermos da função. Assim, por exemplo, na coluna correspondente à função f_1 aparece 1 nas linhas dos mintermos 000, 001 e 101.

Para gerar os implicantes primos, procede-se de forma análoga ao da minimização de uma única função, tomando-se cuidado para (1) combinar somente os pares de subcubos que fazem parte de pelo menos uma mesma função e (2) quando dois cubos C e C' são combinados e um novo cubo C'' é gerado, marcar o subcubo C para descarte somente se o cubo gerado C'' é também cubo das mesmas funções das quais C é cubo (idem para C'). Por exemplo, 001 pode ser combinado com 101 para gerar o cubo $X01$. Neste momento, o cubo 001 (da função f_1) pode ser descartado pois ele é coberto por $X01$ (que também é cubo da função f_1). No entanto, o cubo 101 (das funções f_1 e f_2) NÃO pode ser descartado pois o cubo $X01$ que o cobre não é cubo da função f_2. Quando dois cubos

são combinados e um novo cubo é gerado, ele deve ser colocado em uma outra tabela (mais a direita na figura 1), indicando-se a qual das funções ele pertence. O processo deve ser repetido até que nenhuma combinação seja mais possível. Ao final do processo, os implicantes candidatos são aqueles que não foram marcados para descarte ao longo do processo. No exemplo da figura 1 são aqueles marcados por a, b, c, d e e.

$x_1x_2x_3$	f_1	f_2	
000	1		$\surd$
001	1		$\surd$
101	1	1	e
110		1	$\surd$
111		1	$\surd$

$x_1x_2x_3$	f_1	f_2	
00X	1		a
X01	1		b
1X1		1	c
11X		1	d

Tabela 1: Método tabular para cálculo dos implicantes primos de $\mathbf{f} = (f_1, f_2)$.

O segundo passo do algoritmo é a seleção de uma cobertura mínima. A tabela 2 mostra a **tabela de implicantes primos** de $\mathbf{f} = (f_1, f_2)$, utilizada para a seleção de uma cobertuta mínima. A tabela contém colunas correpondentes a cada um dos mintermos de cada uma das funções, e linhas correspondentes aos implicantes primos calculados no passo anterior. Os números ao lado esquerdo de um implicante primo indicam que o implicante é das funções com os respectivos números. Por exemplo, na coluna ao lado esquerdo de $00X$ aparece (1) para indicar que $00X$ é um implicante da função f_1; ao lado de 101 aparece $(1, 2)$ para indicar que 101 é implicante de f_1 e de f_2. Para cada linha, marcam-se com $\surd$ as colunas correspondentes aos mintermos cobertos pelo implicante primo, **desde que o implicante primo seja implicante da função correspondente ao mintermo** (por exemplo, $X01$ cobre 101, mas não se marca na coluna 101 da função f_2 pois $X01$ não é implicante de f_2).

Após a construção da tabela, deve-se primeiramente selecionar os implicantes primos essencias, que são marcados com $*$ ($00X$ e $11X$). Procede-se com a redução da tabela, eliminando-se a linha dos essenciais, bem como as colunas dos mintermos cobertos por eles. Após a redução da tabela, obtemos a tabela da direita da figura 2.

Se levarmos em consideração apenas a minimização do número de produtos, podemos dizer que o implicante (e) domina os implicantes (b) e (c). Portanto, as linhas (b) e (c) podem ser eliminadas, resultando apenas a linha (e). Temos então o resultado $00X$, $11X$ e 101.

No entanto, se levarmos em conta também, além da minimização do número de produtos, o número de literais em cada produto, não podemos dizer que a linha (e)

			f_1			f_2		
			000	001	101	101	110	111
*	1	(a) 00X	√	√				
	1	(b) X01		√	√			
	2	(c) 1X1				√		√
*	2	(d) 11X					√	√
	1,2	(e) 101			√	√		
			√	√		√		√

			f_1	f_2
			101	101
	1	(b) X01	√	
	2	(c) 1X1		√
	1,2	(e) 101	√	√

Tabela 2: Tabela de implicantes primos de $\mathbf{f} = (f_1, f_2)$.

domina as outras duas. Neste caso, temos uma table cíclica e utilizaremos o método de Petrick para resolvê-lo.

Para cobrir ambas as colunas, a seguinte igualdade deve ser verdadeira:

$$(b + e)(c + e) = 1$$

Escrevendo a expressão acima na forma SOP, temos

$$bc + be + ce + e = 1$$

Daqui podemos concluir que a solução de menor custo é escolher (e). Assim, temos o resultado $00X$, $11X$ e 101 (por coincidência, o mesmo obtido considerando o custo mais simples).

Cálculo de implicantes primos de múltiplas funções usando mapas de Karnaugh

Exemplo 5: Considere as seguintes funções.
$f_\alpha(a, b, c, d) = \sum m(2, 4, 10, 11, 12, 13)$
$f_\beta(a, b, c, d) = \sum m(4, 5, 10, 11, 13)$
$f_\gamma(a, b, c, d) = \sum m(1, 2, 3, 10, 11, 12)$

O processo de cálculo dos implicantes primos pode ser realizado usando os mapas de Karnaugh. Na figura 5 aparecem 7 mapas de Karnaugh, das funções f_α, f_β, f_γ, $f_\alpha \cdot f_\beta$, $f_\alpha \cdot f_\gamma$, $f_\beta \cdot f_\gamma$ e $f_\alpha \cdot f_\beta \cdot f_\gamma$, respectivamente.

Começa-se marcando os implicantes primos (cubos maximais) no mapa da função $f_\alpha \cdot f_\beta \cdot f_\gamma$. O único implicante primo de $f_\alpha \cdot f_\beta \cdot f_\gamma$ é $101X$. Em seguida, marcam-se os implicantes primos da interseção de duas funções, desde que os mesmos não sejam

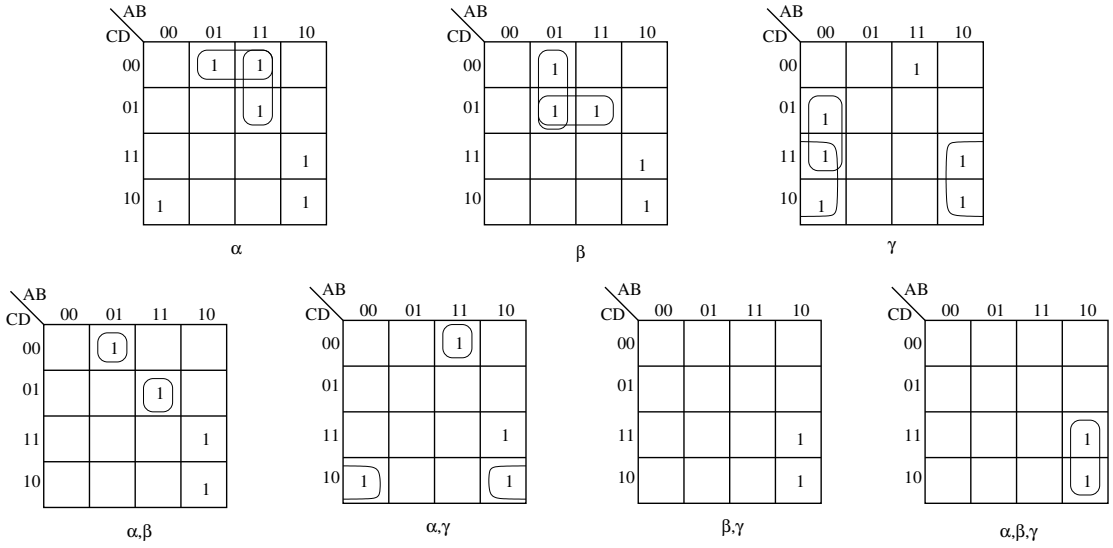

Figura 5: Implicantes primos da função $\mathbf{f} = (f_\alpha, f_\beta, f_\gamma)$.

cobertos pelos implicantes de um produto de funções de ordem maior. Por xemplo, em $f_\alpha \cdot f_\gamma$, $101X$ é um implicante primo mas ele é coberto pelo implicante primo $101X$ de $f_\alpha \cdot f_\beta \cdot f_\gamma$. Portanto, este implicante primo não é marcado. Repete-se o processo para cada uma das funções, marcando-se apenas os implicantes primos que não aparecem em nenhuma das funções $f_\alpha \cdot f_\beta$, $f_\alpha \cdot f_\gamma$, $f_\beta \cdot f_\gamma$ e $f_\alpha \cdot f_\beta \cdot f_\gamma$.

Assim, os implicantes obtidos são os mostrados na tabela a seguir. A coluna da esquerda indica as funções implicadas pelo implicante.

$101X$	$\alpha\ \beta\ \gamma$
$X010$	$\alpha\ \gamma$
1100	$\alpha\ \gamma$
0100	$\alpha\ \beta$
1101	$\alpha\ \beta$
$X01X$	γ
$00X1$	γ
$X101$	β
$010X$	β
$110X$	α
$X100$	α

O método tabular para determinação dos implicantes primos é mostrado na tabela 3.

	f_α	f_β	f_γ	IP
0001			1	√
0010	1		1	√
0100	1	1		i
0011			1	√
0101		1		√
1010	1	1	1	√
1100	1		1	j
1011	1	1	1	√
1101	1	1		k

	f_α	f_β	f_γ	IP
00X1			1	f
001X			1	√
X010	1		1	g
010X		1		d
X100	1			b
X011			1	√
X101		1		e
101X	1	1	1	h
110X	1			c

	f_α	f_β	f_γ	IP
X01X			1	a

Tabela 3: Método tabular para cálculo dos implicantes primos de $\mathbf{f} = (f_\alpha, f_\beta, f_\gamma)$.

Compare os implicantes primos obtidos pelos mapas de Karnaugh e pelo método QM adaptado.

A tabela de implicantes primos é mostrada na figura 4. Primeiramente são identificados os implicantes essenciais.

			f_α						f_β					f_γ					
			2	4	10	11	12	13	4	5	10	11	13	1	2	3	10	11	12
	γ	(a) X01X													√	√	√	√	
	α	(b) X100		√			√												
	α	(c) 110X					√	√											
	β	(d) 010X							√	√									
	β	(e) X101								√			√						
*	γ	(f) 00X1												√		√			
*	$\alpha\gamma$	(g) X010	√		√										√		√		
*	$\alpha\beta\gamma$	(h) 101X			√	√					√	√					√	√	
	$\alpha\beta$	(i) 0100		√					√										
*	$\alpha\gamma$	(j) 1100					√												√
	$\alpha\beta$	(k) 1101						√					√						
			√		√	√	√				√	√		√	√	√	√	√	√

Tabela 4: Tabela dos implicantes primos e seleção de uma cobertura mínima de $\mathbf{f} = (f_\alpha, f_\beta, f_\gamma)$.

Obtemos os **essenciais:** f, g, h, j

Caso 1: Minimizar APENAS o número de produtos (implementação em PLA)

			f_α		f_β		
			4	13	4	5	13
	α	(b) X100	$\surd$				
	α	(c) 110X		$\surd$			
	β	(d) 010X			$\surd$	$\surd$	
	β	(e) X101				$\surd$	$\surd$
**	$\alpha\beta$	(i) 0100	$\surd$		$\surd$		
**	$\alpha\beta$	(k) 1101		$\surd$			$\surd$
			$\surd$	$\surd$	$\surd$	$\surd$	$\surd$

- o número de literais presentes nos implicantes não é importante: b e c podem ser eliminados pois são dominados por i e k, respectivamente.
- i e k são essenciais secundários.
- Para cobrir 5 podemos selecionar d ou e.

RESULTADO (ao selecionarmos d)

$$f_\alpha : g + h + i + j + k = X010 + 101X + 0100 + 1100 + 1101$$
$$f_\beta : d + h + i + k = X101 + 101X + 0100 + 1101$$
$$f_\gamma : f + g + h + j = 00X1 + X010 + 1100 + 101X$$

Note que na expressão de f_β, o termo 1101 é redundante e portanto pode ser eliminado. Assim temos $f_\beta = d + h + i = X101 + 101X + 0100$.

Caso 2: Minimizar <u>número de produtos</u> E <u>número de literais nos produtos</u>

			f_α		f_β		
			4	13	4	5	13
	α	(b) X100	$\surd$				
	α	(c) 110X		$\surd$			
	β	(d) 010X			$\surd$	$\surd$	
	β	(e) X101				$\surd$	$\surd$
	$\alpha\beta$	(i) 0100	$\surd$		$\surd$		
	$\alpha\beta$	(k) 1101		$\surd$			$\surd$

- a tabela acima é cíclica. Não podemos eliminar a linha (b) nem a (c) pois, embora elas sejam dominadas respectivamente pelas linhas (i) e (k), o custo dessas últimas é maior.

- devemos aplicar o método de Petrick

$$(b+i)(c+k)(d+i)(d+e)(e+k) = 1$$

que resulta em

$$cei + bcde + eik + dik + bdk$$

Desses, os de menor custo são cei e bdk

- Para cada função, selecionar o menor subconjunto que a cobre.

RESULTADO (ao selecionarmos cei)

$$f_\alpha : c + g + h + i = 110X + X010 + 101X + 0100$$
$$f_\beta : e + h + i = X101 + 101X + 0100$$
$$f_\gamma : f + g + h + j = 00X1 + X010 + 101X + 1100$$

Comparando os casos 1 e 2 acima, em ambos precisamos de 7 produtos. Há, no entanto, diferença no número de produtos na função f_α. Em PLA, a porta OU dessa função terá uma entrada a mais que no caso não-PLA.

Exemplo 6: minimizar as funções
$f_1(a,b,c,d) = \sum m(3,4,5,7,9,13,15) + d(11,14)$
$f_2(a,b,c,d) = \sum m(3,4,7,9,13,14) + d(0,1,5,15)$

Os implicantes primos são mostrados na figura 6.

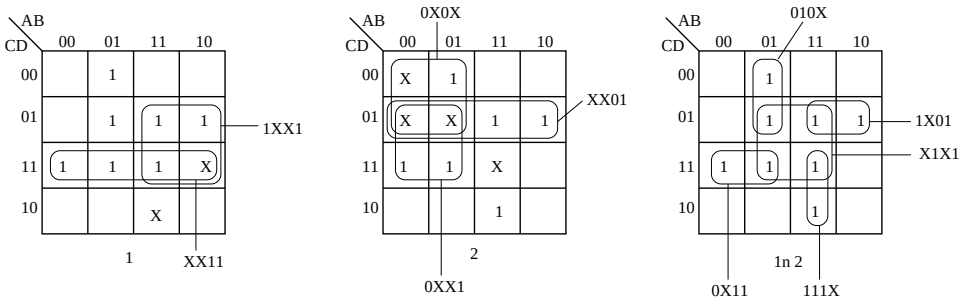

Figura 6: Implicantes primos da função $\mathbf{f} = (f_1, f_2)$.

Uma cobertura mínima pode ser encontrada com o auxílio da Tabela de Implicantes Primos. Os implicantes essenciais são $111X$ e $010X$.

			f_1							f_2					
			3	4	5	7	9	13	15	3	4	7	9	13	14
	1	1XX1					√	√	√						
	1	XX11	√			√			√						
	2	XX01											√	√	
	2	0XX1								√		√			
	2	0X0X									√				
*	1,2	111X							√						√
	1,2	0X11	√			√				√		√			
	1,2	1X01					√	√					√	√	
	1,2	X1X1			√	√		√	√			√		√	
*	1,2	010X		√	√						√				
				√	√				√		√				√

Eliminando os essenciais e colunas, e as linhas vazias, temos a seguinte tabela reduzida. Se considerarmos implementação em PLA, pdemos eliminar a linha dos implicantes $XX11$ e $0XX1$ pois estes são dominados pela linha do implicante $0X11$. Similarmente, podemos eliminar as linhas dos implicantes $1XX1$ e $XX01$.

			f_1				f_2			
			3	7	9	13	3	7	9	13
	1	1XX1			√	√				
	1	XX11	√	√						
	2	XX01							√	√
	2	0XX1					√	√		
	1,2	0X11	√	√			√	√		
	1,2	1X01			√	√			√	√
	1,2	X1X1		√		√		√		√

A tabela resultante é a seguinte. Escolhendo-se os secundariamente essenciais obtém-se uma cobertura para todos os mintermos restantes na tabela.

			f_1				f_2			
			3	7	9	13	3	7	9	13
**	1,2	0X11	√	√			√	√		
**	1,2	1X01			√	√			√	√
	1,2	X1X1		√		√		√		√
			√	√	√	√	√	√	√	√

Assim, a solução final é:
$f_1 : 0X11, 1X01, 111X, 010X$
$f_2 : 0X11, 1X01, 111X, 010X$
Ou seja, $f_1 = f_2$! (Por que isso aconteceu !??)

Exemplo 7: minimizar as funções
$f_1(a,b,c,d) = \sum m(0,2,7,10) + d(12,15)$
$f_2(a,b,c,d) = \sum m(2,4,5) + d(6,7,8,10)$
$f_3(a,b,c,d) = \sum m(2,7,8) + d(0,5,13)$

O cálculo dos implicantes pelo método tabular é mostrado na tabela 5. Os impli-

	f_1	f_2	f_3	IP
0000	1		1	√
0010	1	1	1	m
0100		1		√
1000		1	1	l
0101		1	1	√
0110		1		√
1010	1	1		√
1100	1			k
0111	1	1	1	j
1101			1	√
1111	1			√

	f_1	f_2	f_3	IP
00X0	1		1	i
X000			1	h
0X10		1		g
X010	1	1		f
010X		1		√
01X0		1		√
10X0		1		e
01X1		1	1	d
X101			1	c
011X		1		√
X111	1			b

	f_1	f_2	f_3	IP
01XX		1		a

Tabela 5: Método tabular para cálculo dos implicantes primos de $\mathbf{f} = (f_1, f_2, f_3)$.

cantes 1100, $X101$ e $10X0$ cobrem apenas don't cares e podem ser desconsiderados na etapa de cálculo de cobertura mínima.

Os implicantes podem também ser obtidos pelo mapa de Karnaugh. Veja a figura 7.

Uma cobertura mínima pode ser encontrada com o auxílio da Tabela de Implicantes Primos, que é mostrada na seguinte tabela. Nesta mesma tabela, os implicantes essenciais aparecem marcados por *.

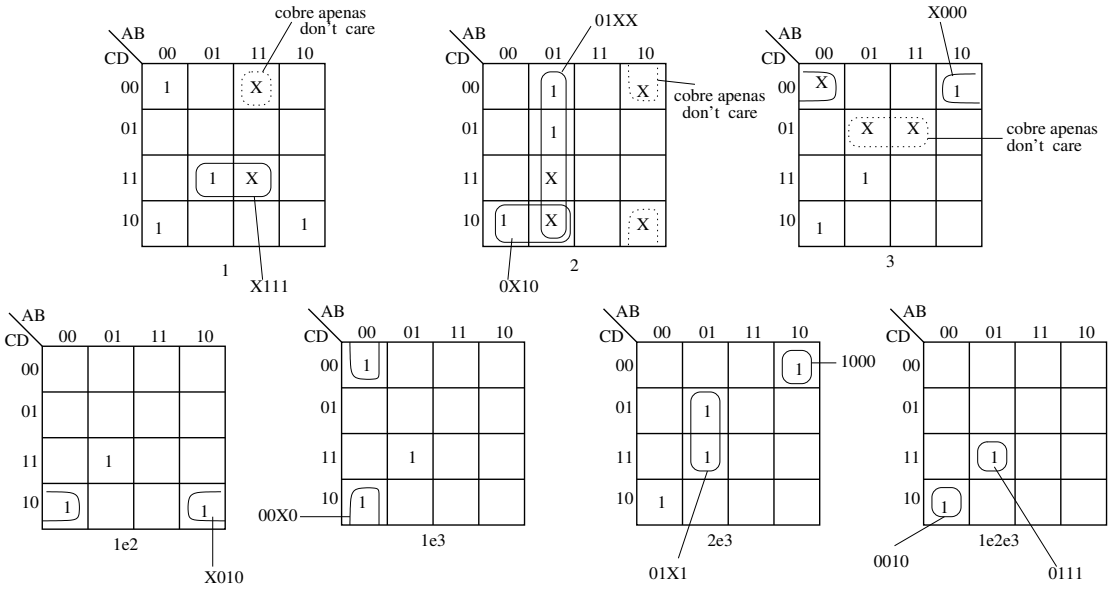

Figura 7: Implicantes primos da função $\mathbf{f} = (f_1, f_2, f_3)$.

			f_1				f_2			f_3		
			0	2	7	10	2	4	5	2	7	8
*	2	(a) 01XX						√	√			
	1	(b) X111			√							
	23	(d) 01X1							√		√	
*	12	(f) X010		√		√	√					
	2	(g) 0X10					√					
	3	(h) X000										√
*	13	(i) 00X0	√	√						√		
	123	(j) 0111			√						√	
	23	(l) 1000					√					√
	123	(m) 0010		√			√			√		
			√	√		√	√	√	√	√		

Eliminando-se a linha dos essencias e as colunas dos mintermos cobertos por eles, obtém-se a seguinte tabela, onde a linha do implicante 1000 é dominada pela linha do implicante $X000$. Ao se eliminar a linha do implicante 1000, o implicante $X000$ torna-se essencial. Das três linhas que restam, é imediato verificar que a escolha que minimiza o custo é 0111, que será marcada com ***.

			f_1	f_3		
			7	7	8	
	1	X111	√			
	23	01X1		√		
**	3	X000			√	
***	123	0111	√	√		
	23	1000			√	⟵ dominada pela linha do $X000$
			√	√	√	

Portanto obtemos:
f_1: $X010$, $00X0$, 0111
f_2: $01XX$, $X010$, $0111 \Longrightarrow 01XX$, $X010$ (pois 0111 é coberto por $01XX$)
f_3: $00X0$, $X000$, 0111.

4 Comentários

Algoritmos de minimização tabular (como o Quine-McCluskey) têm algumas desvantagens do ponto de vista computacional:

- eles listam explicitamente todos os implicantes primos, cuja quantidade pode ser de ordem exponencial no número de variáveis n
- requerem que a função a ser minimizada esteja na forma SOP canônica. Não é raro que, juntamente com os don't cares, o número de produtos canônicos seja da ordem de 2^{n-1}
- a tabela-cíclica pode ser bem grande. O cálculo da cobertura mínima é um problema computacionalmente difícil, no sentido de não haver solução eficiente.

Devido a isso, implementações existentes baseadas no algoritmo QM fazem uso de heurísticas (que produzem resultados sub-ótimos, mas muitas vezes bastante próximos aos resultados ótimos). Um dos algoritmos heurísticos bastante conhecidos é o ESPRESSO, desenvolvido na década de 1990 por um grupo da University of California – Berkeley [Brayton et al., 1984, McGreer et al., 1993]. ESPRESSO não calcula os implicantes primos explicitamente. Além disso, ele também realiza minimização de múltiplas funções e de funções multi-valoradas (lógica multi-valores).

Referências

[Brayton et al., 1984] Brayton, R. K., Hachtel, G. D., McMullen, C. T., and Sangiovanni-Vincentelli, A. L. (1984). *Logic Minimization Algorithms for VLSI Synthesis*. Kluwer Academic Publishers.

[McGreer et al., 1993] McGreer, P. C., Sanghavi, J., Brayton, R. K., and Sangiovanni-Vincentelli, A. L. (1993). Espresso-Signature : A New Exact Minimizer for Logic Functions. *IEEE trans. on VLSI*, 1(4):432–440.