Metoda Quine'a-McCluskeya

Mamy przykładową funkcję F(A,B,C,D). Wybrano cztery zmiennych, gdyż metoda ta dla większej liczby zmiennych jest bardziej pracochłonna i jest możliwa do praktycznego wykorzystania w zasadzie tylko jako algorytm komputerowy. Funkcję opiszemy przy pomocy zbiorów F^1 i F^*.

F^1={1,5,7,8,9,13,15}-zbiór argumentów dla których nasza funkcja przyjmuje wartość logiczną 1, oraz F^*={4,14}- zbiór argumentów gdzie funkcja jest nieokreślona czyli 0 lub 1.

Naszym zadaniem będzie znaleźć minimalną postać dysjunkcyjną (sumę iloczynów) tej funkcji.

Łączymy zbiór F^1 ze zbiorem F^* i otrzymujemy zbiór F={1,4,5,7,8,9,10,14,15}. Argumenty tego zbioru wypisujemy w kolejności dziesiętnej w postaci kolumny ***A*** (poniżej) i dla każdego z nich odpowiednio postać binarną-wektory.

Następnie grupujemy te wektory o tej samej liczbie jedynek. Grupy szeregujemy rosnąco według liczby jedynek, tak jak w kolumnie ***B***

A

1	0001
4	0100
5	0101
7	0111
8	1000
9	1001
13	1011
14	1110
15	1111

B

1	0001	+
4	0100	+
8	1000	+
5	0101	+
9	1001	+
7	0111	+
13	1011	+
14	1110	+
15	1111	+

Teraz porównujemy wektory z postaci ***B*** o k-tej liczbie jedynek z wektorami o k+1 liczbie jedynek i łączymy te które różnią się tylko jednym bitem, np. 0001 i 0101 czyli1 i 5 róznią się jedną pozycją zatem tworzymy nowy wektor 0*01 gdzie symbol * wstawiamy w miejsce gdzie te wektory się różnią (w pozycję różniących bitów wpisujemy *). Tak postępujemy aż do wyczerpania się dalszych łączeń jak przedstawia poniższa tabela ***C.*** Wektory z ***B*** użyte w ***C*** oznaczamy symbolem +. Połączenia, które powstały są przedstawione w ***C.***

C

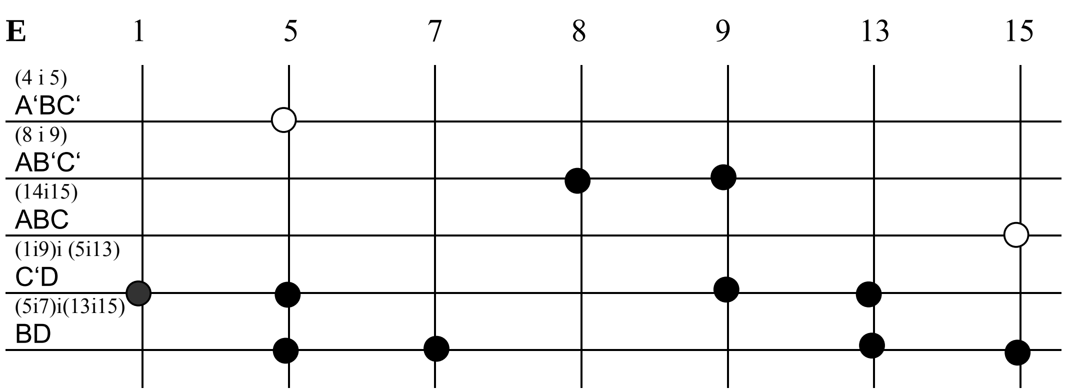
		Implikanty niewykorzystanych Połączeń
1 i 5	0*01	+
1 i 9	*001	+
4 i 5	010*	A‘BC‘
8 i 9	100*	AB‘C‘
5 i 7	01*1	+
5 i 13	*101	+
9 i 13	1*01	+
7 i 15	*111	+
13 i 15	11*1	+
14 i 15	111*	ABC

Z tabelą ***C*** postępujemy podobnie jak z ***B*** tzn. porównujemy np. (1i5)i(1i9) czyli 0*01 i *001 widzimy że różnią się więcej niż jedną pozycją zatem nie możemy ich połączyć, w przypadku różnicy na jednej pozycji łączymy te wektory i na tą pozycje dajemy *. Postępujemy tak aż do wyczerpania się możliwych połączeń (analogicznie do powstawania tabeli ***C***). Implikanty z ***C*** użyte w ***D*** oznaczamy symbolem +. Połączenia, które powstały są przedstawione w ***D.***

D

(1 i 5) i (9 i 13)	**01	C‘D
(1 i 9) i (5 i 13)	**01	C‘D
(5 i 7) i (3 i 15)	*1*1	BD
(5 i 13) i (7 i 15)	*1*1	BD

Postępowanie kończymy gdy nie będzie możliwości dalszych łączeń, tzn. kiedy nie będzie możliwości połączenia wektorów z ***D*** .Z tabeli ***C*** bierzemy wektory bez + i tworzymy ich implikanty (niewykorzystane połączenia) , z ***D*** bierzemy wszystkie implikanty (jeśli są identyczne to bierzemy jeden z nich). Tworzymy tabelę, w której w wierszach będą powyższe implikanty , a w kolumnach argumenty zbioru F^1. Zakreślamy odpowiednie wektory w wierszach dla odpowiednich implikantów np. dla 4 i5 zakreślimy tylko 5 ponieważ 4 nie należy do zbioru F^1 (zakreślamy pustym kółkiem).



Wybieramy teraz kolumny z jednym kółkiem ,bo to z obliguje nas do wybrania całego wiersza (wypełnienia wszystkich kółek w tym wierszu). Jeżeli po tej czynności nie będzie w każdej kolumnie występowało co najmniej jedno kółko zapełnione to czynność powtarzamy dla kolumn bez wypełnionych kółek czyli dla kolumn z dwoma pustymi ,itd. aż do uzyskania pokrycia co najmniej jednego kółka w każdej kolumnie. Z tabeli ***E*** widzimy że pierwszy i trzeci wiersz nie są oznaczone, więc nie będą należeć do zestawu implikantów, a będą należeć pozostałe. Mamy zestaw obowiązkowych implikantów: AB‘C‘, C‘D, BD.

Postacią minimalną funkcji F(X,Y,Z,W) jest suma powyższych implikantów: F(x,y,z,w)= AB‘C‘+ C‘D+BD.