PROIECT MINIMIZAREA QUINE – MCCLUSKEY

Metoda Quine-McCluskey este utilă în minimizarea expresiilor logice pentru numere mari de variabile comparativ cu minimizarea prin Karnaugh Map sau Boolean algebra.

Algoritmul Quine – McCluskey sau metoda implicanților primi este o metodă folosită pentru minimizarea funcțiilor booleane. A fost dezvoltat de W.V. Quine și Edward J. McCluskey în 1956. Este funcțional identic cu minimizarea Karnaugh, dar forma lui tabulară îl face mai eficient pentru utilizarea în algoritmi de calculator și, de asemenea, oferă o modalitate deterministă de a verifica dacă forma minimă a unei funcții booleane a fost atinsă.

Metoda implică doi pași:

1. Găsirea tuturor implicaților primi ai funcției.

2. Utilizarea acelor implicați primi într-o diagramă pentru a găsi implicații primi esențiali ai funcției, precum și alți implicați primi care sunt necesari pentru acoperirea funcției.

PROCEDURA DE MINIMIZARE QUINE-McCLUSKEY

Aceasta este practic o metodă tabulară de minimizare și este potrivită pentru aplicații pe computer. Procedura este:

Pasul 1: Descrieți mintermii individuali ale expresiei date prin echivalentul lor binar.

Pasul 2: Formați un tabel prin gruparea numerelor în funcție de numărul de biți de 1 din reprezentarea lor binară.

Pasul 3: Comparați fiecare număr cu fiecare minterm din grupul următor. Dacă cele două numere diferă prin o singură cifră, puneți o bifă la dreapta ambelor numere pentru a arăta că au fost împerecheate și acoperite. Apoi introduceți numărul nou format în coloana următoare.

Pasul 4: Folosind (3) de mai sus, formați un al doilea tabel și repetați procesul până când împerecherea nu mai este posibilă.

Pasul 5: Termenii care nu au fost incluși sunt principalii implicați și sunt folosiți împreună pentru a forma funcția finală.

PROGRAMUL IMPLEMENTAT

Programul a fost scris în limbajul C++ cu ajutorul aplicației Codeblocks. Acesta urmarește exact pașii minimizării lui Quine-McCluskey.

Se citește și memorează tabelul de adevăr introdus de utilizator de la tastatură.

La primul pas, se ordonează mintermii în tabel în funcție de grupa care îi conține. Se afisează tabelul.

Pasul al doilea, face perechile de mintermi adiacenți și le salvează în alt tabel împreună cu reprezentarea binară. Se bifează în tabelul precedent mintermii utilizați.Se afișează noul tabel.

Pasul al treilea, face perechile de mintermi adiacenți din tabelul anterior si le salvează

într-un nou tabel împreună cu reprezentarea binară. Se bifează în tabelul precedent mintermii utilizați. Se afișează noul tabel.

La al patrulea pas, se găsesc mintermii nebifați din toate cele 3 tabele existente și se trec în ultimul tabel. Se afișează noul tabel.

Ultimul pas determina implicanții primi esențiali care vor fi folosiți pentru finalizarea funcției. În cazul in care aceștia nu acopera toți mintermii funcției, se aleg și alți implicanți cât mai eficient, să acopere cât mai mulți mintermi neacoperiți.

Pentru a memora tabelele se folosesc matrice în care se vor trece reprezentările binare a mintermilor, iar pentru pașii 2, 3 și 4, lista de mintermi se va memora cu ajutorul unui vector folosit ca o listă.

Pentru toti pașii algoritmului există câte o funcție care este apelată din main. În interiorulul funcțiilor se prelucrează datele care vor fi introduse în tabele.

Funcția care generează prima matrice (care reprezintă totodată și tabelul) ordonează mintermii în functie de grupă și copiază din tabelul de adevăr introdus de utilizator reprezentarea binară a fiecarui minterm. Ultima coloană este păstrată pentru a putea bifa mintermii utilizați la urmatorul pas.

A doua matrice conține numai reprezentarea binară a perechilor de mintermi adiacenți după găsirea acestora. Perechea de mintermi va fi salvată într-o listă, având acelși indice cu linia matricei care îi conține scrierea binară. Va exista, și de această data, o coloană pregatită pentru bifarea găsirii unei noi perechi de mintermi în pasul următor.

A treia funcție conține reprezentarea binară a celor 4 mintermi găsiți în a doua matrice (matrice cu 2 mintermi împerecheați cu principiul adiacenței) ca fiind adiacenți. Lista mintermilor va fi salvată în afara matricei, într-o structură de memorare separată, al cărei indice va corespunde cu linia la care se gasește reprezentarea binară a combinației de mintermi.

Ultimul tabel (tot o matrice) va prelua informații din celelalte 3 tabele. Orice minterm sau pereche de mintermi care nu a fost bifată ca fiind folosită pentru a minimiza funcția reprezintă un implicant prim, a cărui reprezentare binară va fi copiată in noul tabel. Mintermii corespunzatori implicantilor primi vor fi scriși într-o listă.

Soluția finală, funcția minimizata, va fi creată in ultima funcție a programului. Aici se determină care sunt implicanții primi esențiali, adică singurii care acoperă un anumit minterm. Se va actualiza lista mintermilor acoperiți de implicanți, iar in cazul in care mai exista mintermi neacoperiți, funcția va determina care implicant este cel mai bun pentru a acoperi un numar maxim de mintermi neacoperiți deja.

După ce se memorează implicanții care acoperă toți mintermii, funcția poate să fie afișată, aceasta fiind funcția minimizată cu ajutorul metodei lui Quine-McCluskey.

La finalul codului, la comentarii, au fost introduse niște tabele de adevăr pentru care au fost testate minimizăriile cu scopul de a reduce timpul de introducere de la tastatură a datelor.

EXEMPLE DE FUNCTIONARE









