Combinatorial Algorithms

Unate Covering

Binate Covering

Graph Coloring

Maximum Clique

Example

- As an Example, let's consider the formula:
  $F(x,y,z) = x'y'z' + x'yz' + x'yz + xyz + xy'z'$
- The complete sum of formula:
  $F(x,y,z) = x'y + x'z' + y'z' + yz$
- Let $P_1 = x'y$ $P_2 = x'z'$ $P_3 = y'z'$ $P_4 = yz$
  $\Rightarrow F(x,y,z) = P_1 + P_2 + P_3 + P_4$
- These are the subset of primes that satisfy the given requirement and are called SOP COVER or COVER.
- To Solve the equation for the minimum cost that covers all rows, we express the above equations in matrix form and that matrix is called CONSTRAINT MATRIX.

Constraint Matrix OR Covering Matrix

- The CONSTRAINT MATRIX is:

P.I / Min Terms	P_1	P_2	P_3	P_4
x'y' z'	0	1	1	0
x'y z'	1	1	0	0
x'y z	1	0	0	1
xyz	0	0	0	1
xy'z'	0	0	1	0

- A CONSTRAINT MATRIX is a matrix that describes the conditions or the constraint that a cover must satisfy, and is also sometimes called Covering Matrix.

Constraint Equation

- Constraint Matrix can be written as Switching function :
  $(P_1 + P_2)(P_2 + P_3) (P_1 + P_4) P_3 \ P_4$
  The equation is called Constraint Equation of the Covering Problem.
- From the above Matrix or Equation we find the subset of columns of minimum cost that covers all the rows."

Defining Set Covering Problem

- Definition: The Set Covering problem <X,Y,Cost> consists of finding a minimal cost subset of Y that covers X.
  Where X is a Set , $Y \subseteq 2^X$ and Cost is the Cost of function defined on Y.
- The covering matrix associated with the set covering problem

< X,Y,Cost > has rows labeled with elements of X and columns labeled with elements of Y , such that the element [x,y] of the matrix is equal to 1 iff y covers x, where x , y are the elements of set X and Y respectively.

X

x'y' z'	0	1	1	0
x'y z'	1	1	0	0
x'y z	1	0	0	1
xyz	0	0	0	1
xy'z'	0	0	1	0

Y

Unate Covering and Binate Covering

- In the constraint equation for example:
  $(P_1 + P_2)(P_2 + P_3) (P_1 + P_4) P_3 \ P_4 = 1$
- All the variables appear in positive form (Nocomplement)---
  - We have a Unate function and the kind of problem is called Unate covering(UCP).
- In the constraint equation for example:

  f(x1,x2,x3,x4) =
  (x1+x3+x4)*(x1+$\overline{x2}$+x4)*($\overline{x2}$+x3+x4)*(x2+$\overline{x3}$+x4)
- All the variables appear in positive form and negative form -----
  - We have a Binate function and the kind of problem is called Binate covering(BCP).

Methods that simplify the solution of CP

- For any problems in logic synthesis(2-Level or Multilevel minimization) the general steps required are:

  (i)Finding a table of all prime implicants.

  (ii) Form a covering matrix (PI vs Minterm) and solve the covering problem.

- For solving covering problem various techniques and algorithms are used.
- The Quine's theorem helps us in finding a minimum cover that is prime!.

Methods that simplify the solution of CP

- Petrick's method can also be applied to find minimum solution
- In the constraint equation for example:

  (P1 + P2)*(P2 + P3) *(P1 + P4) * P3 * P4 = 1

- If solved using Petrick's method gives us the minimum solution,which is simplified to

  (P1 + P2) P3 P4 = 1

- $\Rightarrow$ Each product term is a solution!
- But this method is not practical for large problems as opening the parenthesis involves an *exponential* number of operations.

Methods that simplify the solution of CP

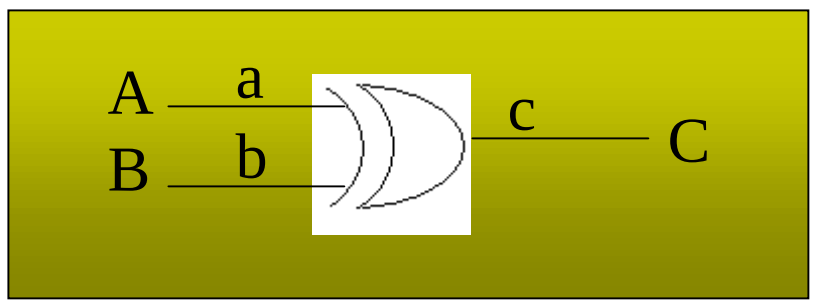
- Branch and Bound algorithm can also be used to choose the best combination of PI'S in case we have a cyclic core.
- Reduction of covering matrix using branch and bound algorithm:
  - **Step (i)** From the constraint matrix find the essential PI'S and Eliminate the rows covered by "Essential columns".
  - **Step (ii)** Elimination of the rows through "Row dominance".
  - **Step (iii)** Elimination of the rows through "Column dominance".

Applications of Covering Problem (Unate or Binate)

- The Covering problems(CP) (Unate or Binate) has several important applications in logic synthesis such as:
- (i) 2-level minimization(Logic or Boolean relation).
- (ii) Three level NAND implementation.
- (iii)Three level CDEC minimization(Conditional decoder).
- (iv) Minimum test set generation.
- (v) Boolean and multiple -valued Decomposition.
- (vi) Physical and technology mapping.
- (vii) State minimization.
- (viii) Exact encoding
- (ix) DAG covering

Example of
Set covering Problem in Testing

- Problem: Finding the minimum test set to detect all possible stuck-at faults in a circuit that has 3 lines, each line can be stuck-at-0 or stuck -at-1.
- Assumption : The gate is an EX_OR gate.
- Logic Diagram of Gate:



Circuit Under Test

Problem

- Explanation of the circuit: The circuit has two input lines (A,B) and the output(C).
- Truth Table:

A	B	a_0	a_1	b_0	b_1	c_0	c_1	C
0	0	0	1	0	1	0	1	0
0	1	1	0	0	1	0	1	1
1	0	0	1	1	0	0	1	1
1	1	1	0	1	0	0	1	0

Problem

- Here a_0,a_1,b_0,b_1,c_0,c_1 are the possible Inputs and Outputs to the circuit.
- If at all there is a fault in the Input or the Output then the fault is represented by x in the below table.

	A	B	a_0	a_1	b_0	b_1	c_0	c_1	C
T_0	0	0		x		x		x	0
T_1	0	1		x	x		x		1
T_2	1	0	x			x	x		1
T_3	1	1	x		x			x	0

Problem

- Here we can say
  a_0 is detected by T_2, T_3
  a_1 is detected by T_0, T_1
  b_0 is detected by T_1, T_3
  b_1 is detected by T_0, T_2
  c_0 is detected by T_2, T_1
  c_1 is detected by T_1, T_3
  $\Rightarrow (T_2+T_3)(T_0+T_1)(T_1+T_3)(T_0+T_2)(T_1+T_2)(T_0+T_3)=1$ This is the unate covering problem as the formula contains no letters that appears in both phases.

Problem

- The Best Covering!
- Many solution are present which has the same cost.
- Here we take the solution to be $T_0T_1T_2$ which would detect all the stuck at faults in the circuit.

Binate Covering Problem

- Definition: The Binate Covering Problem(BCP) consists of finding a minimal cost n -tuple that values f to 1.
- The covering matrix M of the binate covering problem is a matrix made of m rows labeled with the sum s_i, and of n columns labeled with the variables x_j.
- An element M[i,j] of the matrix is equal to 1 if ($x_i \Rightarrow s_i$), to 0 if ($x_j \Rightarrow \underline{\ } s_i$) and to -- otherwise.
- Initially the BCP was known as COVERING WITH CLOSURE in which there was a restrictive assumption that one literal at most appeared complemented in the clause of the function.

Binate Covering Problem

- The binate covering matrix is represented as for function $(x_1+x_3+x_4)(x_1'+x_2+x_4')(x_2+x_3'+x_4)(x_2'+x_3+x_4')$

	x_1	x_2	x_3	x_4
$x_1+x_3+x_4$	1	-	1	1
$x_1+x_2'+x_4'$	0	1	-	0
$x_2+x_3'+x_4$	-	1	0	1
$x_2'+x_3+x_4'$	-	0	1	0

Binate Covering Problem

x_1	x_2	x_3	x_4	x_5	x_6
-	-	0	0	-	-
-	-	-	0	0	-
-	-	1	1	1	-
-	1	-	1	-	1
-	1	-	-	1	-
-	-	-	-	-	1

- Consider the problem **$(x_3'+x_4')*(x_4'+x_5')*(x_3+x_4+x_5)$ $(x_2+x_4+x_6)*(x_2+x_5)*(x_6)$**
- The Covering Matrix is shown with horizontal rows corresponding to each term of the expression .
- From the shown matrix we find $x_1=0$ and $x_6 = 1$ removing the said row and columns we get 4 by 4 matrix which is shown in next slide.

Binate Covering Problem

x_2	x_3	x_4	x_5
-	0	0	-
-	-	0	0
-	1	1	1
1	-	-	1

x_2	x_3	x_5
-	1	1
1	-	1

x_5
1
1

- In figure shown By the column dominance $x_4=0$ and other rows are removed as all the elements are zeros and we are left with x_2 ,x_3 and x_5.
- By column dominance we can say that $x_2=x_3=0$ finally which yields $x_5=1$.
- The complete solution $x_1=x_2=x_3=x_4=0$ and $x_5=x_6=1$

Binate Covering Problem

- **Unate Covering** always has a solution whereas Binate covering **may or may not have a solution** .

- For Example:

(x1+x2)(x1+x2')(x1'+x2)(x1'+x2')

if simplified represents a function that is ***identically zero***. Hence the problem has **no solution**.

Graph Coloring

- A Graph G(V,E) consists of a set of nodes V and a set of edges E.
- Graph Coloring: Is an assignment of colors to the vertices of G, one color to each vertex, so that adjacent vertices are assigned different colors.
- Chromatic number : Minimum number of colors needed for coloring.
- Given a graph G = (V,E) a subgraph H=(U,F) can be constructed by setting U subset of V and F subset of E.
- A Clique is complete subgraph of G.

Graph Coloring

- A Clique is maximal size clique or maximal clique if is is not a subgraph of another clique.
- Maximum Clique is a clique whose size is the largest possible.
- There are two graph coloring problems:
  - Exact Program (EXOC) and
  - approximate program DOM.

Graph Coloring

- Applications for Graph Coloring:
  - Two level minimization
  - Three level minimization
  - Minimum test Set
  - Boolean and Multiple valued decomposition.
- As an Example lets considering graph coloring problem.

Graph Coloring

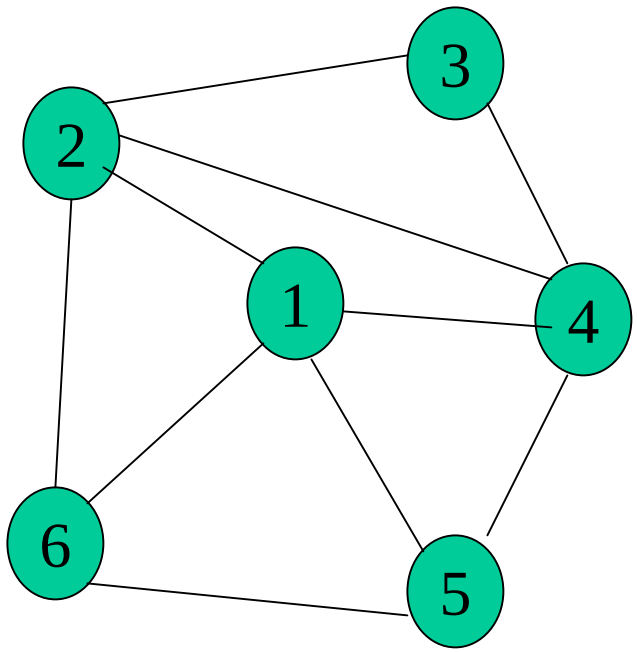
- Concept of **dominations** to color an non incompatibility graph.
- **Definition 1:** Node A in the incompatibility graph covers node B if
  - (i) A and B have no common edges.
  - (ii) A has edges with all the nodes that B has edges with.
  - (iii) A has at least one more edge than B.
- If any node A in the incompatibility graph covers any other node B in the graph, then node B can be removed from the graph.

Graph Coloring

- **Definition 2:** if the above conditions (i) and (ii) are true and A and B have the same number of nodes , then it is called *Pseudo-covering.*
- If there is a pseudo covering any one of the nodes A and B can be removed.

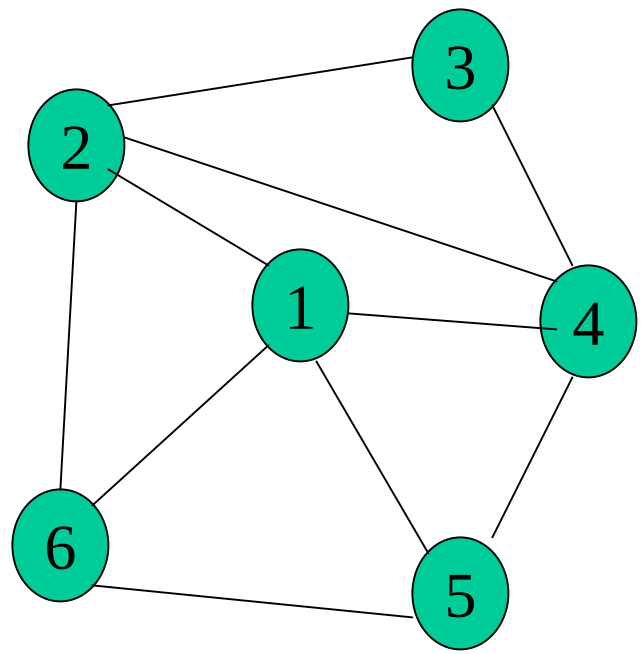
Example of Graph Coloring

- Graph:



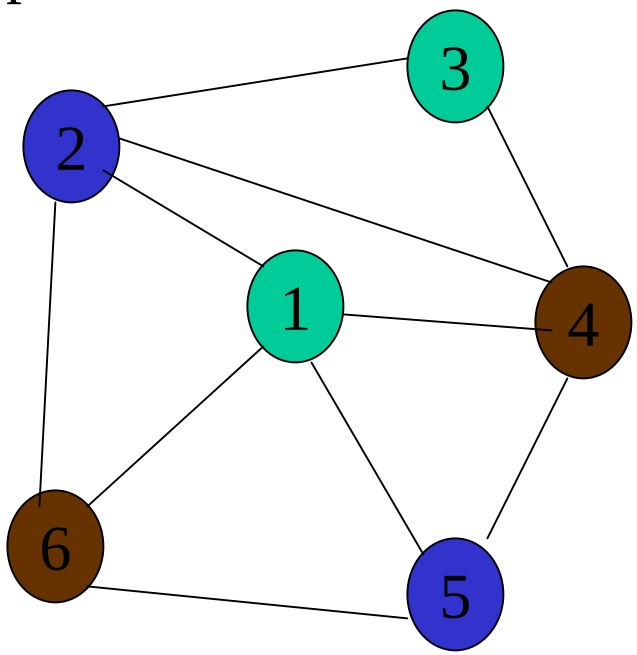
Example of Graph Coloring

- In the given graph
  1 dominates 3 and 3 can be removed.
  We are left with the graph whose nodes are 1,2,4,5,6.
- 2 and 5 are pseudo covering nodes ⇒ 2 and 5 of same color.
  and 4 and 6 are pseudo covering nodes ⇒ 4 and 6 of same color.
  the final graph contains nodes 1,2,4.
  It is a complete graph and can be colored with 3 colors. and since 2 and 5 are pseudo nodes they are given same color 4 and 6 are pseudo nodes ⇒ 4,6 are of same color.



Example of Graph Coloring

- Final Graph:



Maximum Clique

- Coloring of the incompatability graph ⇔ Finding cover with maximum clique ⇔ Clique partitioning of ***compatibility graph***.
- Advantage of using maximum clique is that it helps in finding neighbouring information.

Conclusions

- All the four combinatorial algorithms can be used in logic synthesis for minimization of a function,
- These four methods can also be used to finding the column multiplicity in functional decomposition.
- Graph coloring can be used for FSM minimization.

What to learn?

- How to see the possibility of using any of these problem formulations as parts of new problems from real life that may be given to you
- Descartes formulated the method:“Every problem can be converted to a set of equations. Next the set can be converted to one equation and solved.”
- Boole proposed to use this method to logic problems.
- The Petrick function, tautology, satisfiability are special problems within the area of Boolean Equations.
- These methods, based on tree branchings or solving algebraically formulas, can be extended to multiple-valued and other types of logic, they are very general.
- It is more important to understand how to convert any problems to these combinational problems than to remember efficient algorithms. Efficient algorithms can be found in hundreds of books and software collections, many in public domain.
- The covering, coloring, maximum clique and Boolean equations are used in hundreds of industrial and academic programs not only in logic synthesis but also in test, communication, computer networking, robotics, industrial applications, image processing and pattern recognition. They are also solved using mathematical programming, Genetic Algorithm and Artificial Intelligence techniques

Sources

- Seyda Mohsida