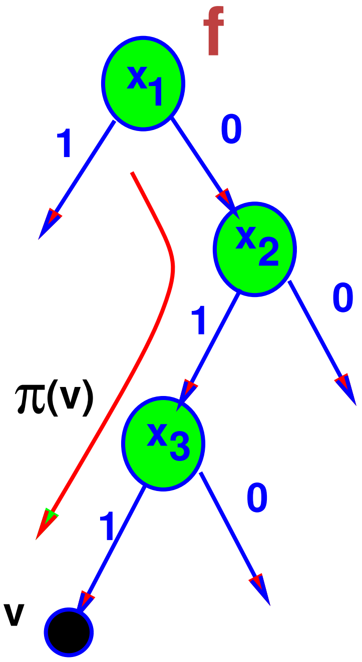
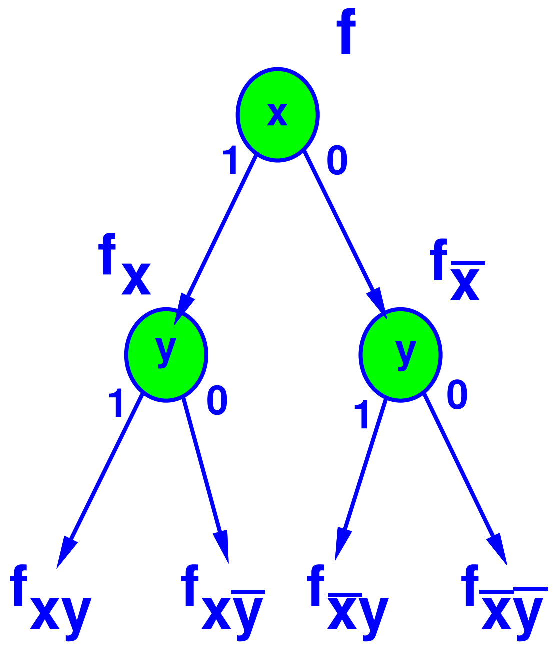
Binary Recursion Tree

The recursive Shannon expansion corresponds to a **binary recursion tree**

$$f = xf_x + \overline{x}f_{\overline{x}} = x(yf_{xy} + \overline{y}f_{x\overline{y}}) + \overline{x}(yf_{\overline{x}y} + \overline{y}f_{\overline{xy}})$$

Example:



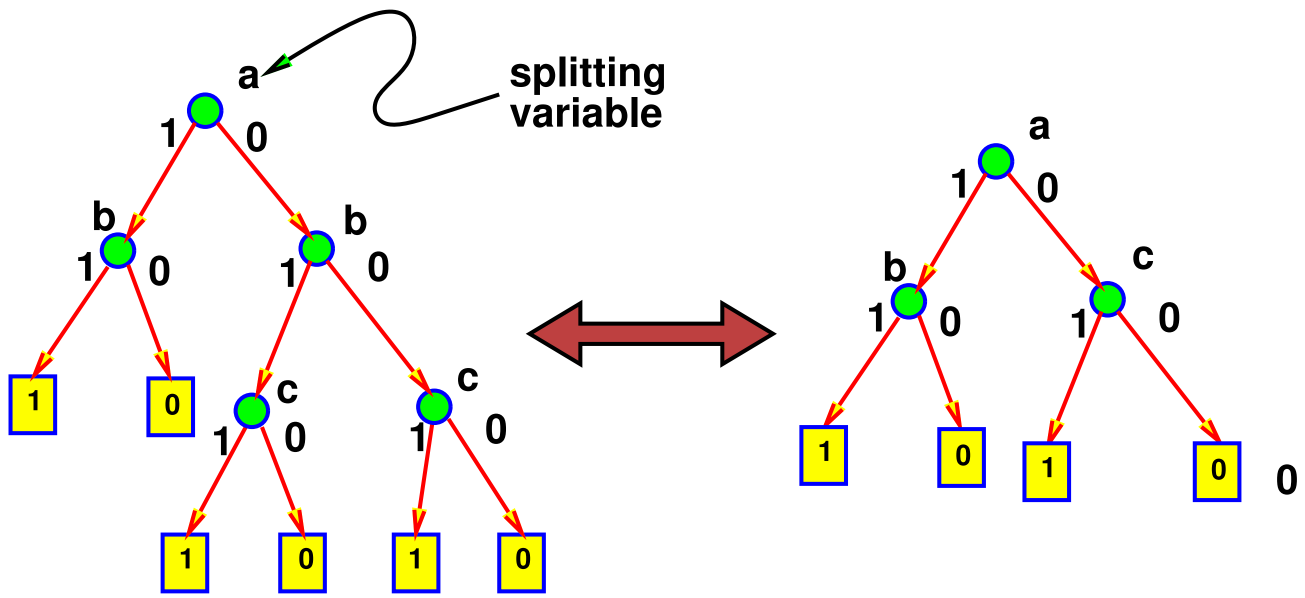


- Path $\pi(v)$ to node v corresponds to cube $c^{\pi(v)}$
  Example: $c^{\pi(v)} = \overline{x}_1 x_2 x_3$

Binary Recursion Tree

- The root represents the original function f.
- Each node v corresponds to $f_{c^{\pi(v)}}$
- If ever $f_{c^{\pi(v)}} = 1$ or 0 we can terminate the tree and replace v by 1 or 0. Such a node v is a leaf.

Example: $f = ab + \overline{a}c$

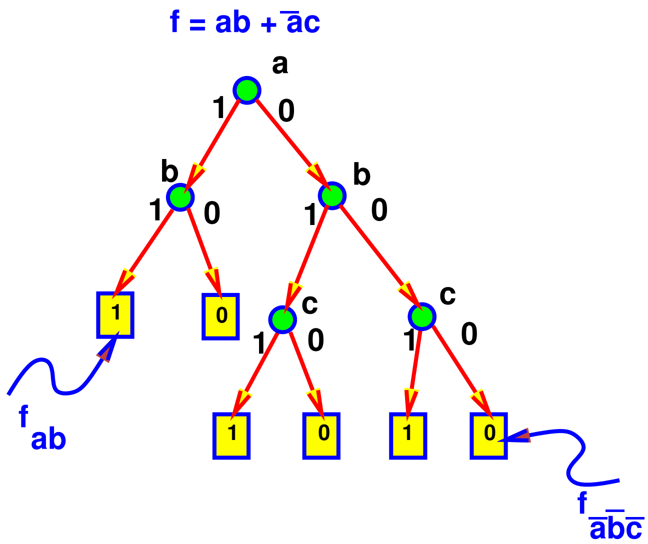


Implicit Enumeration - Branch and Bound

Checking for tautology and many other theoretically intractable problems (co-NP complete) can be **effectively** solved using implicit enumeration:

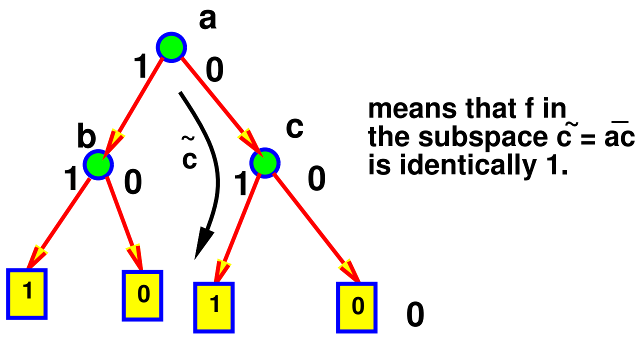
- use recursive Shannon expansion to explore B^n
- in (hopefully) large subspaces of B^n, prune the binary recursion tree by
    - exploiting properties of the node function $f_{c^{\pi(v)}}$
    - exploiting heuristic bounding techniques
- even though in the worst case the recursion tree may have 2^n nodes, in practice we typically encounter a **linear** number of nodes
- thus we say that the 2^n min-terms of f have been **implicitly enumerated**
- BDD's (Binary Decision Diagrams) are alternate representations in which implicit enumeration is performed **statically**, and nodes with identical path cofactors are identified (*very important - will discuss later*)

Example



Not a tautology

In testing for tautology, we look for a cube subspace c such that $f_c = 0$. If we can't find it then $f \equiv 1$.
Can rule out complete cube subspace $\widetilde{c}$ if $f_{\widetilde{c}} = 1$



Tautology can be checked by finding $\{c_i\}$ such that $\sum c_i = 1$ and $f_{c_i} \equiv 1$ for all c_i. We don't need that $c_i c_j = \phi$.

Unate Functions and Unate Covers

Definition 1 *A function* $f \; : \; B^n \rightarrow B$ *is* **positive unate** *in* **variable** x_i *iff*

$$f_{\overline{x}_i} \subseteq f_{x_i}$$

This is equivalent to **monotone increasing** in x_i:

$$f(m^-) \leq f(m^+)$$

for all min-term pairs (m^-, m^+) where

$$\begin{aligned} m_j^- &= m_j^+ \;\; j \neq i \\ m_i^- &= 0 \\ m_i^+ &= 1 \end{aligned}$$

For example, $m_3^- = 1001$, $m_3^+ = 1011$ (where $i = 3$).

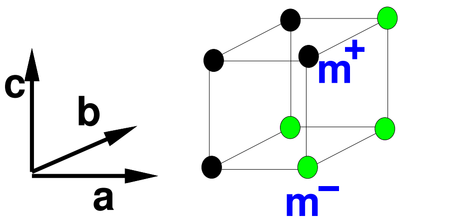
Similarly for
negative unate: $f_{x_i} \subseteq f_{\overline{x}_i}$
monotone decreasing: $f(m^-) \geq f(m^+)$

A function is **unate** in x_i if it is either positive unate or negative unate in x_i.

Unate Functions

Definition 2 *A function is* **unate** *if it is unate in each variable*

Example 1 $f = ab + b\overline{c} + a\overline{c}$



positive unate in a,b

negative unate in c

$f(m^-) = 1 \geq f(m^+) = 0$

Definition 3 *A cover* F *is* **positive unate** *in* x_i *iff* $\overline{x}_i \notin c_j$ *for all cubes* $c_j \in F$

The Unate Recursive Paradigm

- In the ESPRESSO program, the key pruning technique is based on exploiting the properties of **unate** functions.

- In particular, the **splitting** variable is chosen so that the functions at lower nodes of the recursion tree become **unate**.

- **Unate** covers F have many extraordinary properties:
  - If a cover F is minimal with respect to single-cube containment, all of its cubes are essential primes.
  - In this case F is the unique minimum cube representation of its logic function
  - A unate cover represents the tautology iff it contains a cube with no literals, i.e. a single tautologous cube.

This type of implicit enumeration applies to many subproblems (prime generation, complementation, reduction, etc.). Hence we refer to it as the **Unate Recursive Paradigm**

The Binate Select Heuristic

Tautology and other programs based on the unate recursive paradigm use a heuristic called BINATE_SELECT to choose the splitting variable in recursive Shannon expansion. The idea is for a given cover F, choose the variable which occurs, both positively and negatively, most often in the cubes of F.

Example 2 *Unate and non-unate covers:*

$$G = ac + c\bar{d} = \begin{array}{cccc} a & b & c & d \\ \hline 1 & 2 & 1 & 2 \\ 2 & 2 & 1 & 0 \end{array}$$

is UNATE

$$F = ac + \bar{c}d + bc\bar{d} = \begin{array}{cccc} a & b & c & d \\ \hline 1 & 2 & 1 & 2 \\ 2 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{array}$$

is NOT UNATE

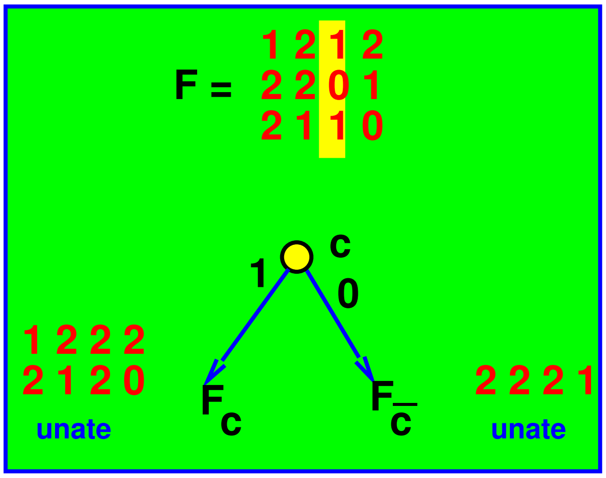
Choose c for splitting.

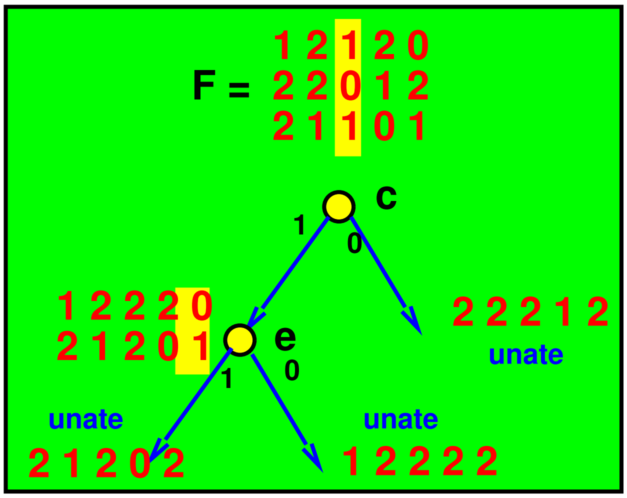
The **binate variables of a cover** are those with both 1's and 0's in the corresponding column.

In the unate recursive paradigm, the BINATE_SELECT heuristic chooses a (most) binate variable for splitting, which is thus eliminated from the sub-covers.

Examples

$$F = ac + \overline{c}d + bc\overline{d}$$





Tautology

$$
\begin{aligned}
F &= acd + bcd + \overline{a}b\overline{d} + \overline{ac}\overline{d} \\
&+ \overline{c}d + a\overline{c} + a\overline{d} + \overline{b}c\overline{d} + \overline{a}\overline{b}d + \overline{a}\overline{b}c
\end{aligned}
$$

Is $F = 1$?
NOT EASY!

$$
F = \begin{array}{cccc}
1 & 2 & 1 & 1 \\
2 & 1 & 1 & 1 \\
0 & 1 & 2 & 0 \\
0 & 2 & 0 & 0 \\
2 & 2 & 0 & 1 \\
1 & 2 & 0 & 2 \\
1 & 2 & 2 & 0 \\
2 & 0 & 1 & 0 \\
0 & 0 & 2 & 1 \\
0 & 0 & 1 & 2
\end{array} == 1?
$$

Two Useful Theorems

Theorem 1

$$F \equiv 1 \iff (F_{x_j} \equiv 1) \wedge (F_{\overline{x}_j} \equiv 1)$$

Theorem 2 *Let A be a unate cover matrix. Then $A \equiv 1$ if and only if A has a row of all 2's.*

Proof.
if: A row of all 2's is the tautology.

only if: Assume no row of all 2's. Without loss of generality (WLOG), suppose function is positive unate. Then each row has at least one "1" in it. Consider the point (vertex) $x = (0, 0, \ldots, 0)$. This is not contained in any row of A. Hence $A \not\equiv 1$. □

Unate Reduction for Tautology Checking

Let $F(x)$ be a cover. Let (a, x') be a partition of the variables x, and let

$$F = \begin{bmatrix} A & X \\ T & F' \end{bmatrix}$$

where:

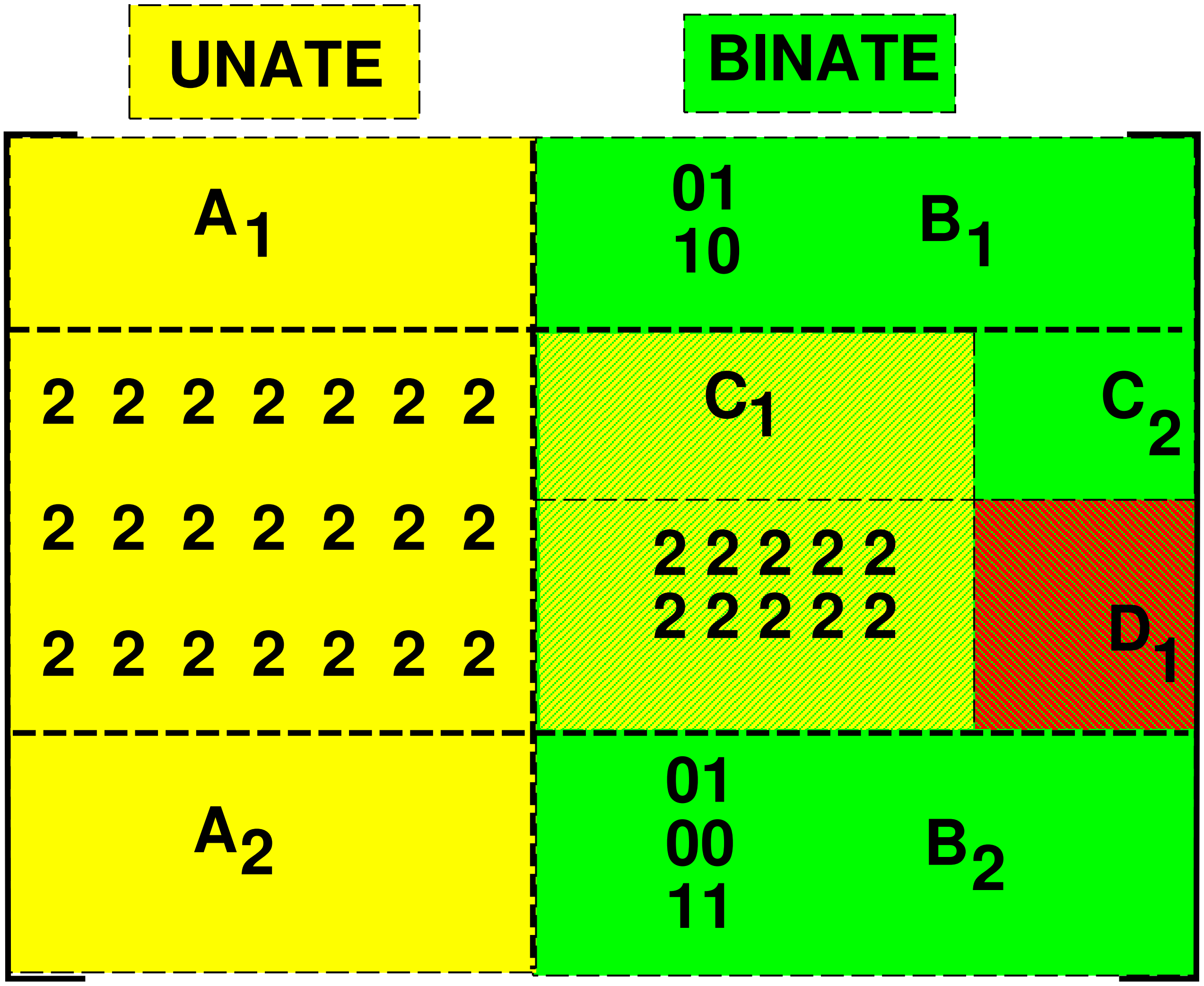
- the columns of A correspond to variables a of x;
- $A \not\equiv 1$, i.e. the function represented by A **is not** the tautology
- T is a matrix of all 2's

Theorem 3 *Assume $A \not\equiv 1$. Then $F \equiv 1 \iff F' \equiv 1$*

Example 3

$$F = \begin{bmatrix} A^1 & X^1 \\ T^1 & F^1 \end{bmatrix} = \begin{matrix} 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \\ & & & \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 1 \end{matrix} \longrightarrow \begin{matrix} 1 & 1 \\ & \\ 2 & 0 \\ 2 & 1 \end{matrix}$$

Unate Reduction



Result: Only have to look at D_1, to test if this is a tautology.
Note A_1, A_2 has no row of all 2's and is a unate cover. Hence $(A_1, A_2) \not\equiv 1$.

Proof

Theorem: Assume $A \not\equiv 1$. Then $F \equiv 1 \Longleftrightarrow F' \equiv 1$

$$F = \begin{array}{c} \begin{array}{cc} \text{unate} & \text{binate} \end{array} \\ \begin{bmatrix} A & X \\ T & F' \end{bmatrix} \end{array}$$

A =/= 1

T = 2's

Proof.
if: Assume $F' \equiv 1$. Then can replace F' by all 2's. Then last row of F becomes a row of all 2's, so tautology.

only if: Assume $F' \not\equiv 1$. Then there is a min-term m_2 such that $F'(m_2) = 0$, i.e. $m_2 \notin$ cube of F'. Similarly m_1 exists where $A(m_1) = 0$, $m_1 \notin$ cube of A. Now the minterm (m_1, m_2) in the full space satisfies $F(m_1, m_2) = 0$ since $m_1 m_2 \notin AX$ and $m_1 m_2 \notin TF'$.

$(a, \widetilde{x})$ is any row of first part
$a(m_1) \times \widetilde{x}(m_2) = 0 \times \widetilde{x}(m_2) = 0$

(t, f') is any row of last part
$t(m_1) \times f'(m_2) = t(m_1) \times 0 = 0$

So $m_1 m_2$ is not in any cube of F. $\square$

Unate Reduction for Tautology

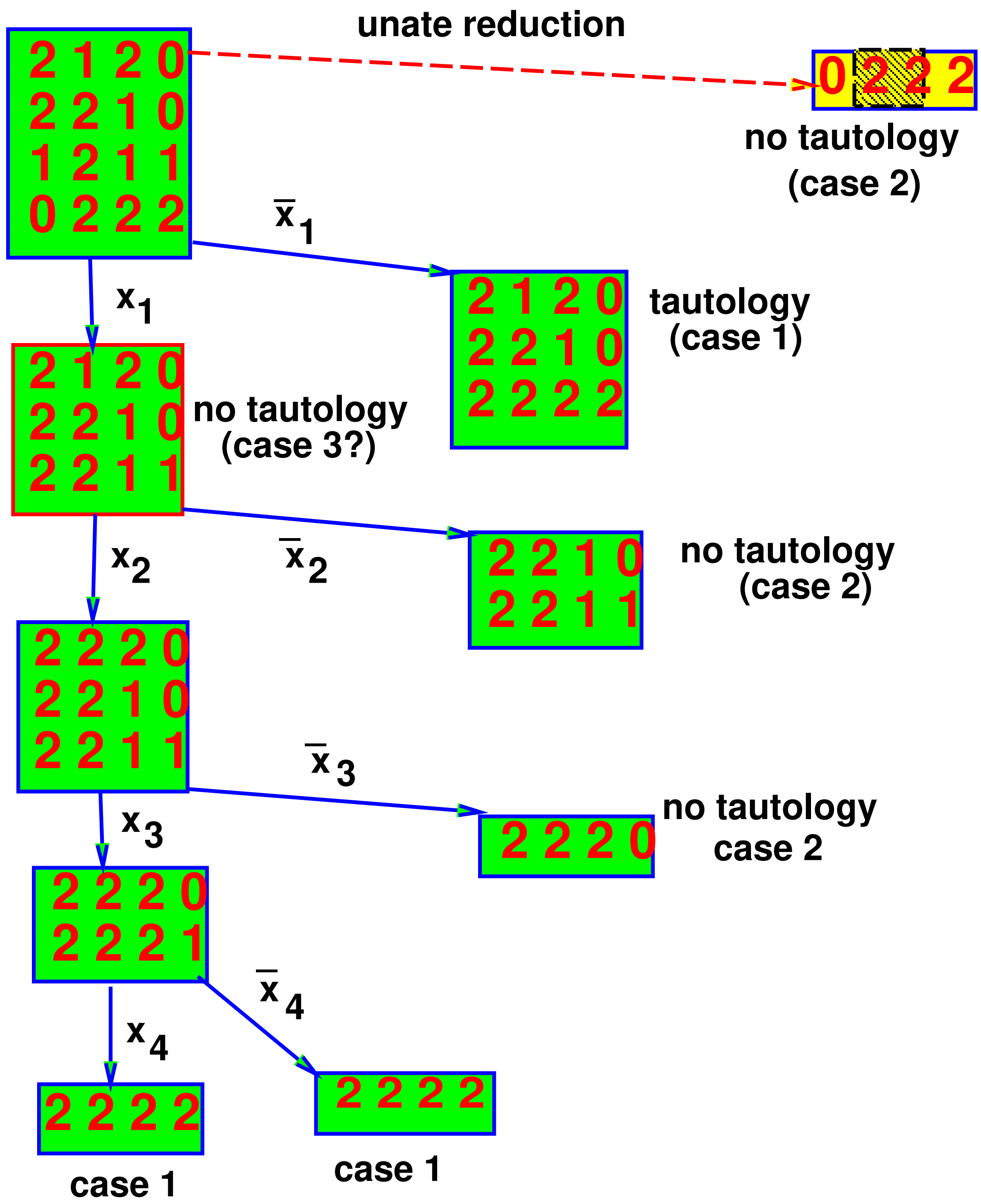
Procedure TAUTOLOGY(F, C)
// C is a cube returned if $F \not\equiv 1$. Then C contains
// a minterm m where $F(m) = 0$
    $T \leftarrow$ SPECIAL_CASES(F)
    if$(T \neq -1)$ return (T)
    $F \leftarrow$ UNATE_REDUCTION(F)
    if$(F = \phi)$ print(C), return(0)
    $j \leftarrow$ BINATE_SELECT(F)
    $T \leftarrow$ TAUTOLOGY$(F_{x_j}, C \cup \{x_j\})$
    if$(T = 0)$ print$(C \cup \{x_j\})$, return(0)
    $T \leftarrow$ TAUTOLOGY$(F_{\overline{x}_j}, C \cup \{\overline{x}_j\})$
    if $(T = 0)$ print$(C \cup \{\overline{x}_j\})$, return(0)
    return(1)
end

Notes: $T = 1(0)$ if F is the tautology (is empty), else $T = -1$

SPECIAL_CASES $(T = -1$ unless)

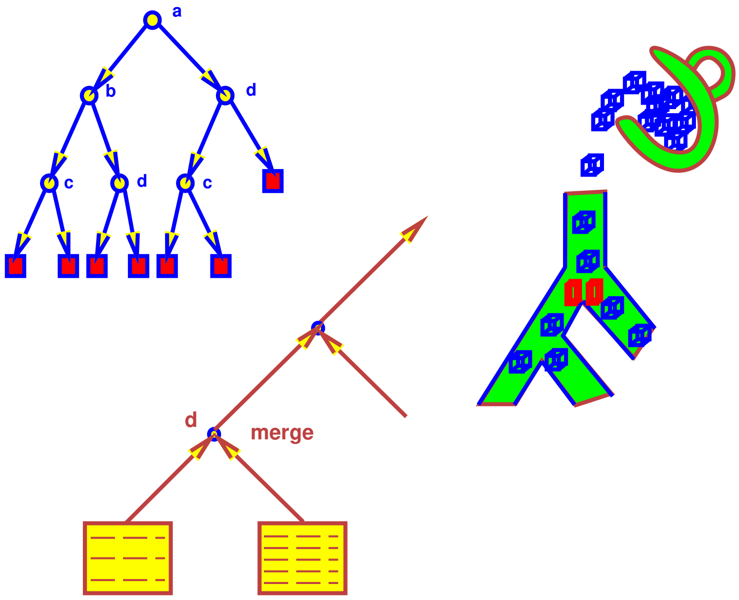
1. $T = 1:$ F contains a cube with no literals (all 2's)
2. $T = 0:$ F contains same literal in every cube
3. $T = 0$ if # minterms $< 2^n$.

Binary Recursion with and without Unate Reduction



Unate Recursive Paradigm

1. Create cofactoring tree stopping at **unate covers**
   - choose, at each node, the "most binate" variable for splitting
   - recurse til no binate variable left (unate leaf)
2. "Operate" on the unate cover at each leaf to obtain the result for that leaf. Return the result.
3. At each non-leaf node **merge** (appropriately) the results of the two children.



Main Idea: "Operation" on unate leaf is easy!

Operations: complement, simplify, tautology, generate-primes,...

Complement of a Unate Cover

1. Map cube matrix M into Boolean matrix B

$$\begin{array}{ccccc} a & b & c & d & e \\ \hline 2 & 1 & 2 & 0 & 2 \\ 2 & 2 & 0 & 0 & 1 \\ 1 & 1 & 2 & 2 & 1 \\ 1 & 2 & 0 & 2 & 1 \end{array} \longrightarrow \begin{matrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{matrix} = B$$

Thus: non – 2 $\longrightarrow$ 1
2 $\longrightarrow$ 0

2. Find all minimal column covers of B. (*A column cover is a set of columns J such that for each row i, $\exists j \in J$ such that $B_{ij} = 1$*).

   Example 4 $\{1, 4\}$ *is a minimal column cover for:*

$$\begin{matrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{matrix} \longrightarrow \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \end{matrix}$$

   All rows "covered" by at least one 1.

3. For each **minimal** column cover create a cube with opposite column literal from M.

Example

Example 5 $\{1,4\} \rightarrow \overline{a}d$ *is a cube of* $\overline{f}$

4. The set of all minimal column covers = cover of $\overline{f}$.

Example 6 :

$$\begin{array}{ccccc} a & b & c & d & e \\ \hline 2 & 1 & 2 & 0 & 2 \\ 2 & 2 & 0 & 0 & 1 \\ 1 & 1 & 2 & 2 & 1 \\ 1 & 2 & 0 & 2 & 1 \end{array} \quad \longrightarrow \quad \begin{array}{ccccc} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{array}$$

{(1,4),(2,3),(2,5),(4,5)}

is the set of all minimal column covers

This translates into:

$$\overline{f} = \overline{a}d + \overline{b}c + \overline{b}\,\overline{e} + d\overline{e}$$

Unate Complement Theorem

Theorem 4 *Let M be a unate cover of f. The set of cubes associated with the minimal column covers of B_M is a cube cover of $\overline{f}$.*

Proof. We first show that **any** such **cube** c generated is in the **offset** of f, by showing that the cube c is orthogonal (has empty intersection) with any cube of M. Note, the literals of c are the complemented literals of M. (Since M is a unate cover, the literals of M are just the union of the literals of each cube of M) For each cube $m_i \in M$, $\exists\ j \in J$ such that $B_{ij} = 1$. *(J is column cover associated with c)* Thus $(m_i)_j = x_j \Longrightarrow (c)_j = \overline{x}_j$ and $(m_i)_j = \overline{x}_j \Longrightarrow (c)_j = x_j$. Thus $m_i c = \phi$. Thus $c \subseteq \overline{f}$.

We now show that any **min-term** $\mu \in \overline{f}$ is **contained** in **some cube** c generated. First, μ must be orthogonal to each cube of M. So for each row of M, there is at least one literal of μ that conflicts with that row. The union of all columns (literals) where this happens is a column cover of B_M; hence it contains at least one minimal cover and the associated cube contains μ. □

Example

Example 7 :

$$f = b\overline{d} + \overline{c}\overline{d} + abe + a\overline{c}e$$

$$M = \begin{array}{c} a\ b\ c\ d\ e \\ \hline \begin{matrix} 2&1&2&0&2 \\ 2&2&0&0&1 \\ 1&1&2&2&1 \\ 1&2&0&2&1 \end{matrix} \end{array} \longrightarrow \begin{array}{c} B_M \\ \begin{matrix} 0&1&0&1&0 \\ 0&0&1&1&1 \\ 1&1&0&0&1 \\ 1&0&1&0&1 \end{matrix} \end{array}$$

$\{(1,4),(2,3),(2,5),(4,5)\}$

is the set of all minimal column covers

This translates into:

$$\overline{f} = \overline{a}d + \overline{b}c + \overline{b}\overline{e} + d\overline{e}$$

A minimal column cover is (1,4) $\longrightarrow$ $\overline{a}d$

$$\overline{a}d \perp f = b\overline{d} + \overline{c}\overline{d} + abe + a\overline{c}e$$

Consider min-term $\overline{a}bcd\overline{e} \in \overline{f}$. *It conflicts in literals* $\overline{a}, c, d, \overline{e}$. *Thus* $\{1, 3, 4, 5\}$ *is a column cover. It contains* $\{1, 4\}$ *and* $\{4, 5\}$. *Thus*

$$\begin{aligned} \overline{a}bcd\overline{e} &\in \overline{a}d \\ &\in d\overline{e} \end{aligned}$$

Unate Covering

Definition 4 *The problem, given a Boolean matrix B, find a minimum column cover, is called a* **unate covering** *problem.*

The problem of unate complementation was our first example of the unate covering problem appearing. We will see it **often** in this course.

Unate Covering Problem:
Given B, $B_{ij} \in \{0, 1\}$ find x, $x_i \in \{0, 1\}$ such that

$$Bx \geq 1$$

and $\sum_j x_j$ is **minimum**.

Sometimes we want to minimize:

$$\sum_j c_j x_j$$

where c_j is a cost associated with column j.

Quine-McCluskey Procedure (Exact)

Given $\widetilde{G}$ and D (covers for $\mathcal{F} = (f, d, r)$ and d), find a **minimum** cover G of primes where:

$$f \subseteq G \subseteq f + d$$

(G is a prime cover of $\mathcal{F}$)

Q-M Procedure:

1. Generate all the primes of $\mathcal{F}$, $\{P_i\}$ (*i.e. primes of* $f + d = \widetilde{G} + D$)

2. Generate all the minterms of $f = \widetilde{G}\overline{D}$, $\{m_i\}$

3. Build Boolean matrix B where

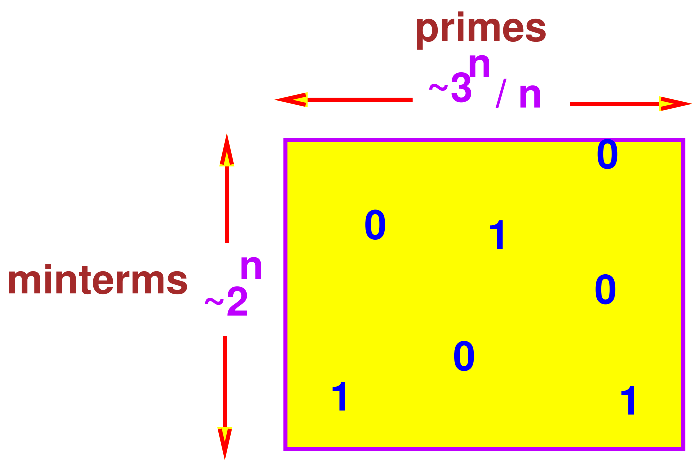
$$\begin{aligned} B_{ij} &= 1 \; if \; m_i \in P_j \\ &= 0 \; otherwise \end{aligned}$$

4. Solve the column covering problem for B (*unate covering problem*)

Difficulty

Note:

- $\sim 2^n$ minterms
- $\sim \frac{3^n}{n}$ primes

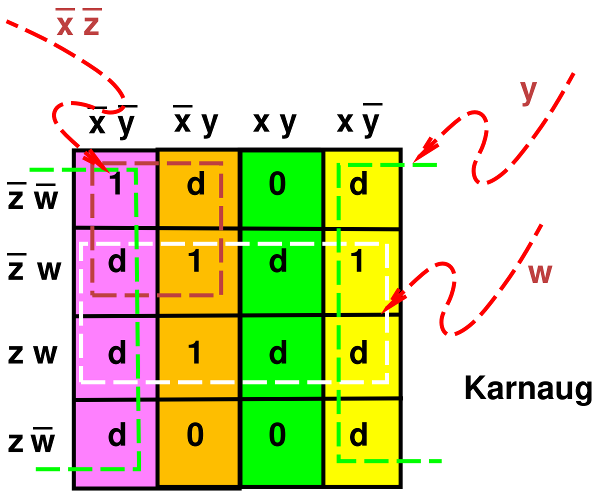


Thus $O(2^n)$ rows and $O(\frac{3^n}{n})$ columns AND minimum covering problem is NP-complete. (*Hence probably exponential in* 3^n, *i.e. difficulty is* $O(2^{3^n})$).

Example

Example 8

$$
\begin{aligned}
F &= \overline{xyzw} + \overline{x}y\overline{z}w + x\overline{y}\overline{z}w + \overline{x}yzw \\
D &= \overline{y}z + xyw + \overline{xyz}w + x\overline{yw} + \overline{x}y\overline{zw}
\end{aligned}
$$



Karnaugh Map

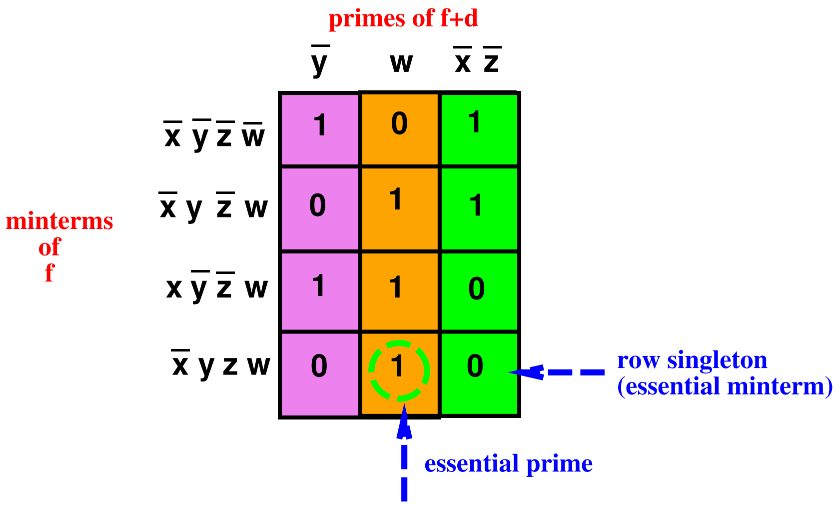
Primes: $\overline{y} + w + \overline{xz}$

Covering Table:

		primes of f+d		
		$\overline{y}$	w	$\overline{x}\,\overline{z}$
minterms of f	$\overline{x}\,\overline{y}\,\overline{z}\,\overline{w}$	1	0	1
	$\overline{x}\,y\,\overline{z}\,w$	0	1	1
	$x\,\overline{y}\,\overline{z}\,w$	1	1	0
	$\overline{x}\,y\,z\,w$	0	1	0

Solution: $\{1, 2\} \Longrightarrow \overline{y} + w$ *is minimum prime cover.*
(also: $w + \overline{xz}$)

Covering Table



Definition 5 *An* **essential prime** *is any prime that uniquely covers a minterm of f.*

Row Equality: In practice, many rows are identical. That is there exist minterms that are contained in the same set of primes.

$$\begin{array}{ll} m_1 & 0101101 \\ m_2 & 0101101 \end{array}$$

These can be grouped into a cube - called a **signature cube**.

$$cube = \prod_{m_i \in P_j} P_j$$

e.g. $m_1 \cup m_2 \subseteq P_2 P_4 P_5 P_7$

Row and Column Dominance

Definition 6 *A row i_1 whose set of primes is contained in the set of primes of row i_2 is said to* **dominate** *i_2.*

Example 9

$$\begin{array}{ll} i_1 & 011010 \\ i_2 & 011110 \end{array}$$

i_1 *dominates* i_2

We can **remove** row i_2, because we have to choose a prime to cover i_1, and any such prime also covers i_2. So i_2 is automatically covered.

Definition 7 *A column j_1 whose rows are a superset of another column j_2 is said to* **dominate** *j_2.*

Example 10

$$\begin{array}{cc} j_1 & j_2 \\ 1 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 1 & 1 \end{array}$$

j_1 *dominates* j_2

We can **remove** column j_2 since j_1 covers all those rows and more. We would never choose j_2 in a minimum cover since it can always be replaced by j_1.

Pruning the Covering Table

1. Remove all rows covered by essential primes. Put these primes in the cover G.

2. Group identical rows together and remove dominated rows.

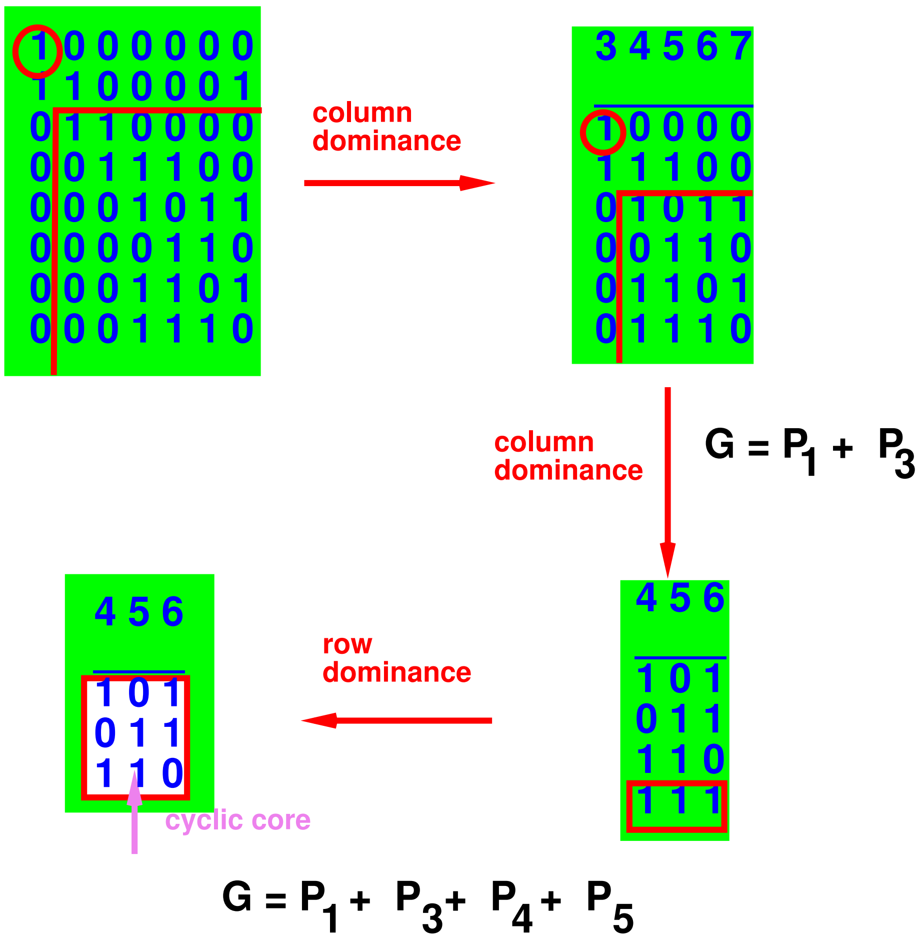
3. Remove dominated columns. For equal columns keep one prime to represent them.

4. Newly formed row singletons define n**-ary essential primes**.

5. Go to 1 if covering table decreased.

The resulting reduced covering table is called the **cyclic core**. This has to be solved (unate covering problem). Its minimum solution is added to G - the set of n-ary essential primes. The resulting G is **a** minimum cover.

Example

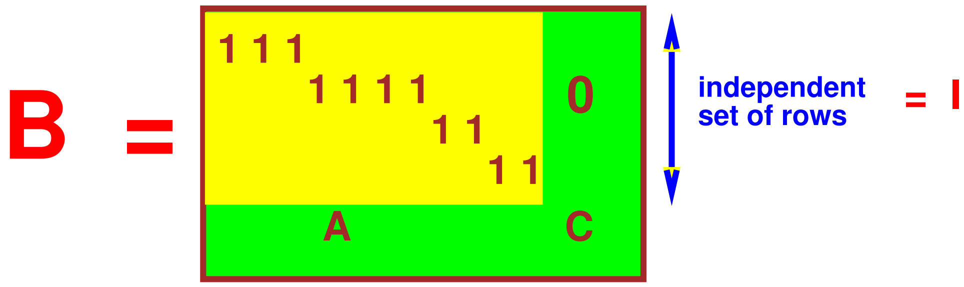

1 0 0 0 0 0 0
1 1 0 0 0 0 1
0 1 1 0 0 0 0
0 0 1 1 1 0 0
0 0 0 1 0 1 1
0 0 0 0 1 1 0
0 0 0 1 1 0 1
0 0 0 1 1 1 0
column dominance
3 4 5 6 7
1 0 0 0 0
1 1 1 0 0
0 1 0 1 1
0 0 1 1 0
0 1 1 0 1
0 1 1 1 0
column dominance
G = P1 + P3
4 5 6
1 0 1
0 1 1
1 1 0
1 1 1
row dominance
4 5 6
1 0 1
0 1 1
1 1 0
cyclic core
G = P1 + P3 + P4 + P5

Solving the Cyclic Core

Best known method (for unate covering) is **branch and bound** with some clever bounding heuristics.

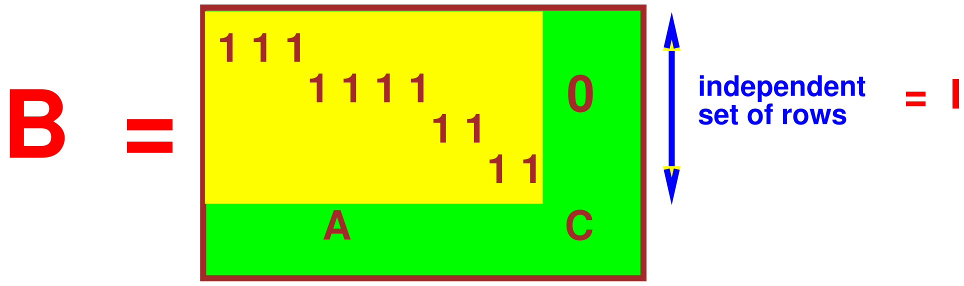
Independent Set Heuristic:
Find a maximum set of "independent" rows I. Two rows B_{i_1}, B_{i_2} are **independent** if $\nexists j$ such that $B_{i_1 j} = B_{i_2 j} = 1$. (*They have no column in common*).

Example 11 *Covering matrix B rearranged with independent sets first.*



Lemma 1 $|$*Solution of Covering*$| \geq |I|$

Heuristic



Let $I = \{I_1, I_2, \ldots, I_k\}$

1. choose $j \in I_i$ which covers the most rows of A. Put $j \to J$
2. eliminate all rows covered by column j
3. $I \leftarrow I - \{I_j\}$
4. go to 1 if $|I| > 0$
5. if B is empty, then done (*in this case we have the guaranteed minimum solution-* **IMPORTANT**).
6. if B is not empty, choose an independent set of B and go to 1

Can you think of some improved heuristics?

Generating Primes

We use the unate recursive paradigm. The following is how the merge step is done. (Assumption: we have just generated all primes of f_{x_i} and $f_{\overline{x}_i}$.)

Theorem 5 *p is a prime of f if and only if p is* **maximal** *among the set consisting of:*

1. *$p = x_i q$, q is a prime of f_{x_i}, $q \not\subseteq f_{\overline{x}_i}$*

2. *$p = \overline{x}_i r$, r is a prime of $f_{\overline{x}_i}$, $r \not\subseteq f_{x_i}$*

3. *$p = qr$, q is a prime of f_{x_i}, r is a prime of $f_{\overline{x}_i}$*

Example 12 *Assume $q = abc$ is prime of f_{x_i}. Form $p = x_i abc$. Suppose $r = ab$ is prime of $f_{\overline{x}_i}$. Thus $q \subseteq f_{\overline{x}_i}$. Then $\overline{x}_i ab$ is an implicant of f (by Shannon expansion, $f = x_i abc + \overline{x}_i ab + \ldots$) Thus, abc and $\overline{x}_i ab$ are implicants, so $\overline{x}_i abc$ is* **not prime** *($\overline{x}_i abc \subseteq abc$).*

Note: abc is prime because if not, $ab \subseteq f$ (assume remove c). Then $ab \subseteq f_{x_i}$ contradicting abc prime of f_{x_i}.
Note: $\overline{x}_i ab$ is prime, since if not then either $ab \subseteq f$ or $\overline{x}_i a \subseteq f$. The first contradicts abc prime of f_{x_i} and the second contradicts ab prime of $f_{\overline{x}_i}$.

Unate Recursion for Primes

```
PRIMES(G = F + D) (i.e. G is a cover of f + d)
    if (G is unate) return UPRIMES(G)
    x_i = BINATE_SELECT(G)
    P_1 = PRIMES(G_{x_i})
    P_0 = PRIMES(G_{x̄_i})
    C = φ
    foreach p ∈ P_1
        C ← C ∪ {x_i p}
        foreach q ∈ P_0, C ← C ∪ {pq}
    foreach q ∈ P_0, C ← C ∪ {x̄_i q}
    sort C by decreasing number of 2's
    foreach c ∈ C
        foreach d ∈ C, d ≻ c
            if c ⊇ d, C ← C − {d}
    return C
end
```

Note: UPRIMES is a special function for unate covers. It uses the following facts:

1. Every prime of a unate function is essential.
2. The primes of a unate cover are obtained simply by: $C \leftarrow C - \{d\}$ if $d \subset c \in C$.

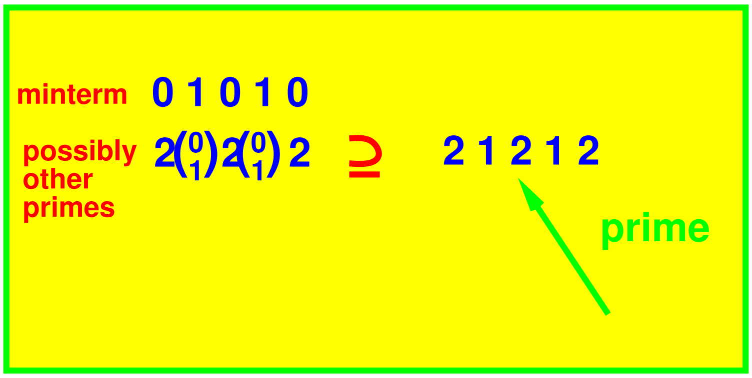
Note: $q \not\subseteq f_{\overline{x}_i} \Leftrightarrow q \not\subseteq$ prime of $f_{\overline{x}_i}$.

Primes of Unate Functions

Theorem 6 *Every prime of a unate function is essential*

Intuition: Consider a positive unate function.

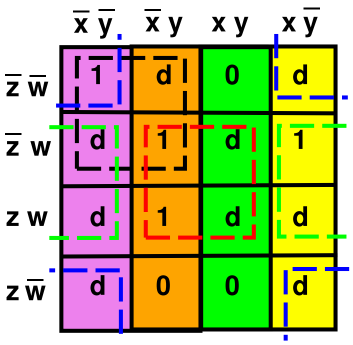
- Then primes are cubes with only 1's and 2's: e.g. 2 1 2 1 2
- Any such cube contains the minterm where 2's are replaced by 0's: e.g. $01010 \subseteq 21212$.
- If cube is not contained in any other cube of cover, no other cube contains this minterm because if so, it must have a 2 wherever a 0 occurs (because prime only has 1's and 2's).



So no other prime contains this minterm $\Longrightarrow$ prime is **essential**. Thus every prime of a unate function has such an (essential) minterm and hence the prime is essential.

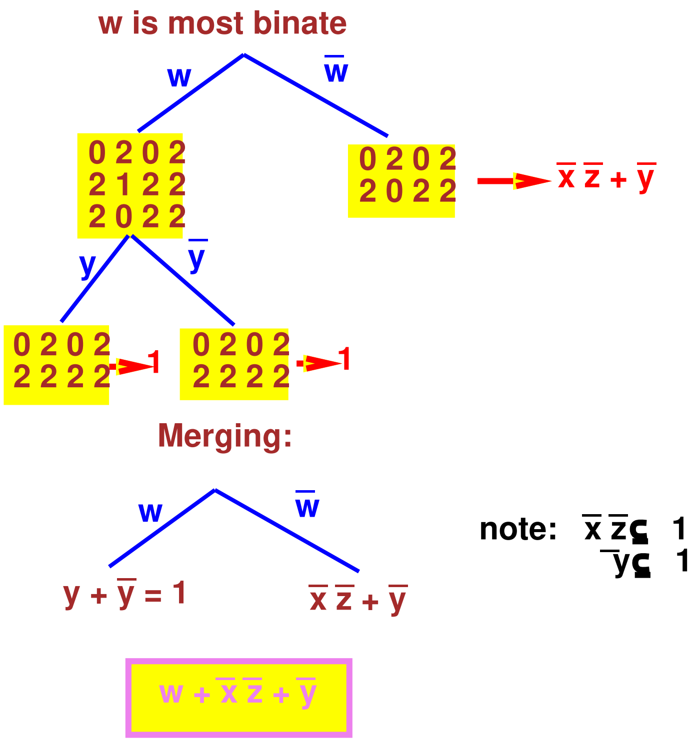
Note: Assuming it is prime, 21212 is the only positive unate implicant that contains 01010.

Example - Generate all Primes



Assume cover $F + D = \overline{xz} + \overline{yw} + yw + \overline{y}w$

x	y	z	w
0	2	0	2
2	0	2	0
2	1	2	1
2	0	2	1



New Exact Methods

For 40 years, Q-M method basically unchanged. In 1992, two fundamentally new methods developed:

1. McGeer's Method (McGeer, Sanghavi, et.al.)
   - generate all essential signature cubes (ESC) (if c a cube, $\sigma(c)$ is the intersection of all primes containing c; an essential signature cube is a maximal cube of $\{\sigma(m_i)\}$ where m_i is a minterm of the onset.)
   - generate **only** those primes that contain these cubes (ESP)
   - form covering table ESC vs ESP and solve
2. Implicit Q-M based on BDD's (Coudert and Madre, McGeer, Swamy, Brayton)
   - form characteristic BDD of all primes
   - form characteristic BDD of all minterms of f
   - formulate row dominance and column dominance as BDD operations
   - iterate dominance and prime and minterm elimination until no change
   - generate covering table and solve

Great Result: In both cases, superior to ESPRESSO-EXACT (*both in speed and number of problems solved*).