Generalized Cofactor

Definition 1 *Let f, g be completely specified functions. The* **generalized cofactor of f with respect to g** *is the* incompletely *specified function*

$$co(f, g) = (fg, \overline{g}, \overline{f}g)$$

Definition 2 *Let $\mathcal{F} = (f, d, r)$ and g be given. Then*

$$co(\mathcal{F}, g) = (fg, d + \overline{g}, rg)$$

Relation with Shannon Cofactor:
Let $g = x_i$. Shannon cofactor is

$$f_{x_i} = f(x_1, \ldots, (x_i = 1), \ldots, x_n)$$

and generalized cofactor with respect to $g = x_i$ is

$$co(f, x_i) = (fx_i, \overline{x}_i, \overline{f}x_i)$$

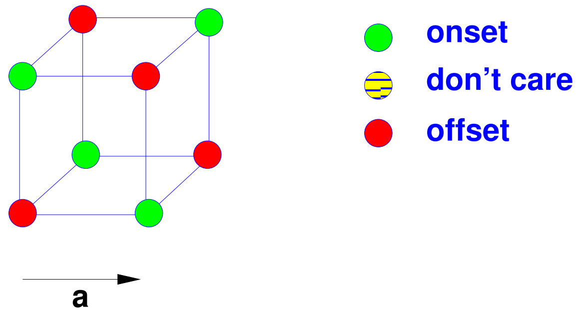
Note that

$$fx_i \subseteq f_{x_i} \subseteq fx_i + \overline{x}_i = f + \overline{x}_i$$

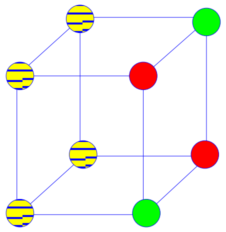
In fact f_{x_i} is the unique cover of $co(f, x_i)$ **independent** of the variable x_i.

Shannon Cofactor

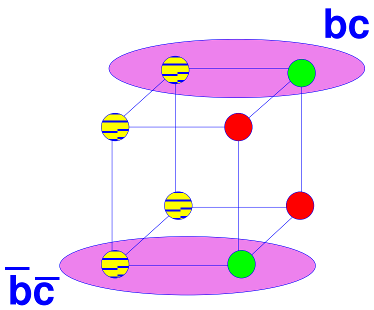
$f =$



$co(f,a) = (fa, \bar{a}, \bar{f}a) =$

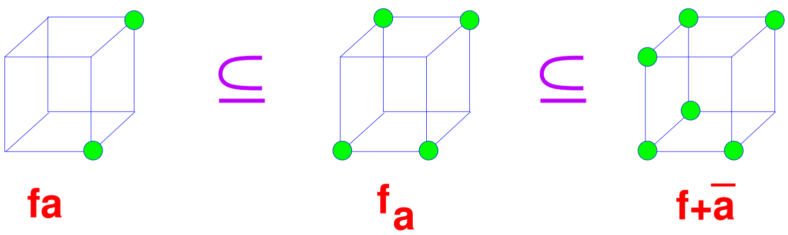


$f_a =$



$$
\begin{aligned}
f &= a\bar{b}\bar{c} + abc + \bar{a}\bar{b}c + \bar{a}b\bar{c} \\
f_a &= \bar{b}\bar{c} + bc
\end{aligned}
$$

So $fa \subseteq f_a \subseteq f + \bar{a}$



Properties of Generalized Cofactor $co(f, g)$

1. $f = g\ co(f, g) + \overline{g}\ co(f, \overline{g})$
2. $co(co(f, g), h) = co(f, gh)$
3. $co(fg, h) = co(f, h)co(g, h)$
4. $co(\overline{f}, g) = \overline{co(f, g)}$

These compare with the following properties of the Shannon cofactor:

1. $xf_x + \overline{x}f_{\overline{x}} = f$
2. $(f_x)_y = f_{xy}$
3. $(fg)_x = f_x g_x$
4. $(\overline{f})_x = \overline{(f_x)}$

We will get back to use of generalized cofactor later.

Quine-McCluskey Summarized

Q-M:

1. Generate cover of all primes

$$G = p_1 + p_2 + \ldots + p_{3^n/n}$$

2. Make G irredundant (*in optimum way*)

Note: Q-M is **exact** i.e. it gives an exact minimum.

Heuristic Methods:

1. Generate (*somehow*) a **cover** of $\mathcal{F}$ using **some** of the primes

$$G = p_{i_1} + p_{i_2} + \ldots + p_{i_k}$$

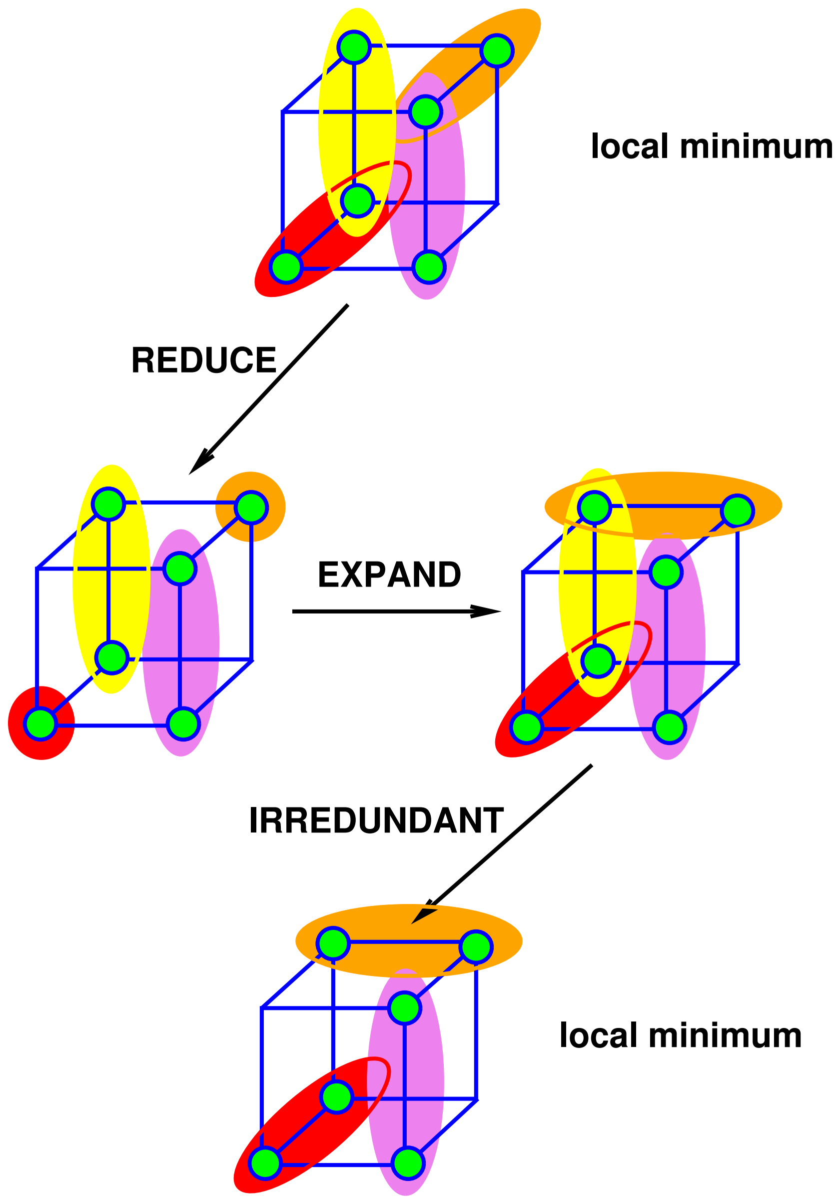
2. Make G irredundant (*maybe not optimumly*)

3. Keep best result - try again (i.e. go to 1)

ESPRESSO - Heuristic Two-Level Minimization

```
ESPRESSO(𝓕)
{
    (F, D, R) ← DECODE(𝓕)
    F ← EXPAND(F, R)
    F ← IRREDUNDANT(F, D)
    E ← ESSENTIAL_PRIMES(F, D)
    F ← F − E; D ← D + E
    do{
        do{
            F ← REDUCE(F, D)
            F ← EXPAND(F, R)
            F ← IRREDUNDANT(F, D)
        } while fewer terms in F
        //LASTGASP
        G ← REDUCE_GASP(F, D)
        G ← EXPAND(G, R)
        F ← IRREDUNDANT(F + G, D)
        //LASTGASP
    }while G ≠ φ
    F ← F + E; D ← D − E
    LOWER_OUTPUTS(F, D)
    RAISE_INPUTS(F, R)
    error ← (F_on ⊈ F) or (F ⊈ F_old + D)
    return(F,error)
}
```

ESPRESSO Illustrated



IRREDUNDANT

Problem:
Given a cover of cubes $\{c_i\}$, find a minimum subset $\{c_{i_k}\}$ that is also a cover, i.e.

$$f \subseteq \sum c_{i_k} \subseteq f + d$$

Idea 1:
We are going to create a function $g(y)$ and a new set of variables y_i, one for each cube c_i. A minterm in the y-space will indicate a subset of cubes.

Example 1

$$y = 011010 = \{c_2, c_3, c_5\}$$

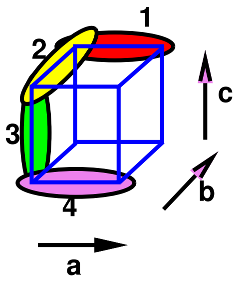
Idea 2:
$g(y)$ is a function such that:
$g(y) = 1 \iff (\sum_{y_{i_k}=1} c_{i_k})$ is a cover
i.e. $\{c_i \mid y_i = 1\}$ is a cover.

Note: $g(y)$ is positive unate (monotone increasing) in all its variables.

Example

Example 2

$$f = bc + \overline{a}c + \overline{a}\overline{b} + \overline{b}\overline{c}$$

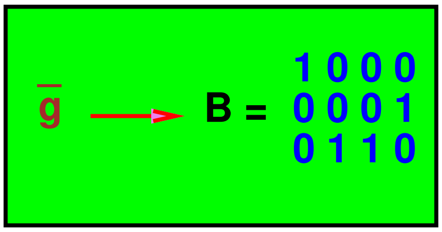


$$g(y_1, y_2, y_3, y_4) = y_1 y_4 (y_2 + y_3)$$

Note: We are after a minimum subset of cubes. Thus we want: *the largest prime of g (least literals)* Consider $\overline{g}$. It is monotone decreasing in y, e.g.

$$\overline{g} = \overline{y}_1 + \overline{y}_4 + \overline{y}_2\overline{y}_3$$

Create a Boolean matrix for $\overline{g}$:



- Recall a minimal column cover of B is a prime of $g = \overline{\overline{g}}$.
- We want a minimum cover of B

Example 3 $\{1, 2, 4\} \Longrightarrow y_1 y_2 y_4$ *(cubes 1,2,4)* $bc + \overline{a}c + \overline{b}\overline{c}$

Deriving $\overline{g}(y)$

Modify Tautology algorithm:

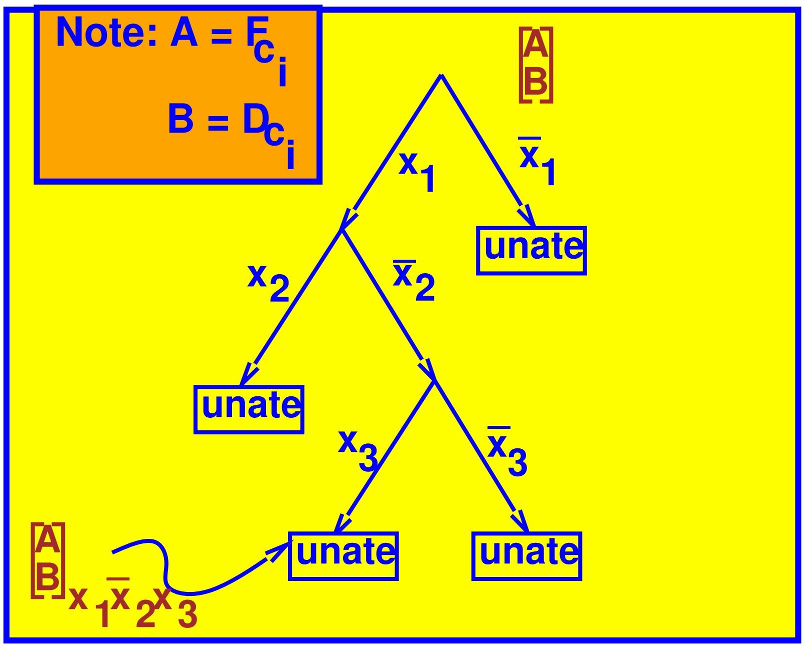
$F =$ cover of $\mathcal{F} = (f, d, r)$

$D =$ cover of d

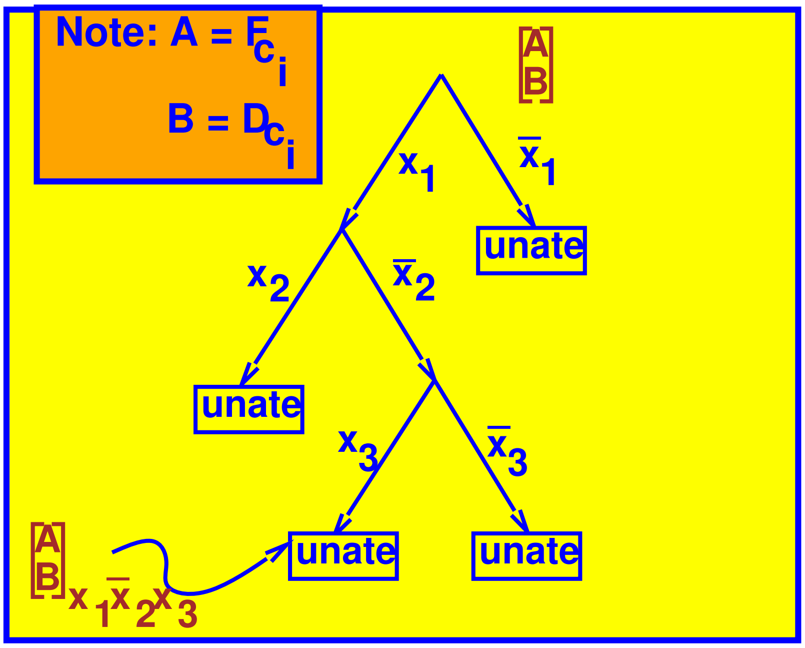
Pick a cube $c_i \in F$. (Note: $c_i \subseteq F \iff F_{c_i} \equiv 1$)

Do the following for each cube $c_i \in F$:

$$\begin{aligned} A &= F_{c_i} \\ B &= D_{c_i} \end{aligned}$$



Deriving $\overline{g}(y)$ -continued



1. Leaf must be tautology

2. $\overline{g}$ means how can we make it not the tautology. We must exactly delete **all** rows of all 2's

3. Each row came from some row of $\begin{matrix} A \\ B \end{matrix}$

4. Each row of A is associated with some cube of F.

5. Each cube of B is associated with some cube of D (we don't need to know which one).

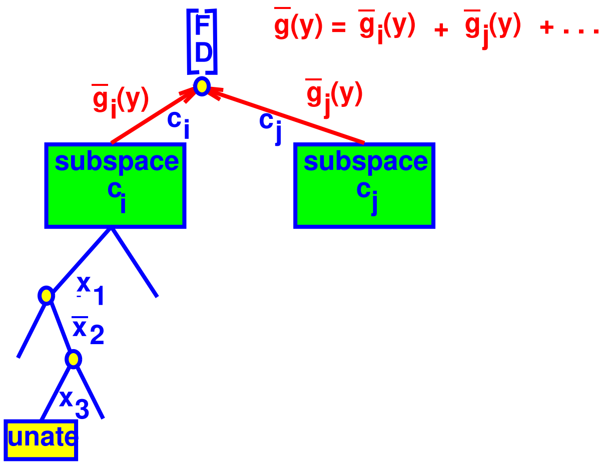
6. Rows that must be deleted are written as a cube, e.g. $\overline{y}_1\overline{y}_3\overline{y}_7 \Longrightarrow$ delete rows 1,3,7 of F.

Deriving $\overline{g}(y)$ - continued

Example 4 *Suppose unate leaf is in subspace* $x_1\overline{x}_2x_3$*:*

$$\begin{bmatrix} A \\ B \end{bmatrix}_{x_1\overline{x}_2x_3} = \begin{array}{c} \begin{bmatrix} 2 & 1 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ & & & \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 2 & 2 \end{bmatrix} \end{array} \begin{array}{l} y_2 \\ y_{10} \\ y_{18} \\ \\ \\ \\ \end{array}$$

Thus we write down: $\overline{y}_{10}\overline{y}_{18}$ (actually, $\overline{y}_i$ must be one of $\overline{y}_{10}\overline{y}_{18}$ *) Thus,* F *is* **not a cover** *if we leave out cubes* c_{10}, c_{18}.



Note: *If in*

$$\begin{bmatrix} A \\ B \end{bmatrix}_{x_1\overline{x}_2x_3} = \begin{bmatrix} 2 & 1 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ & & & \\ 2 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 \end{bmatrix} \begin{array}{l} y_2 \\ y_{10} \\ y_{18} \\ \\ \\ \\ \end{array}$$

row of all 2's in don't cares

then there is no way not to have tautology.

Summary

$$\begin{aligned}\overline{g}(y) &= \overline{y}_1(\overline{y}_{10}\overline{y}_{18} + \overline{y}_6\overline{y}_7 + \ldots) + \overline{y}_2(\overline{y}_6\overline{y}_{14} + \ldots) + \ldots \\ &\quad + \overline{y}_{11}(\overline{y}_3\overline{y}_7\overline{y}_8 + \ldots)\end{aligned}$$

Convert $\overline{g(y)}$ into a Boolean matrix B (note that $g(y)$ is unate). Then find a minimum cover of B. For example, if $= y_1 y_3 y_{18}$ is a minimum cover, then the set of cubes

$$\{c_1, c_3, c_{18}\}$$

is a minimum subcover of $\{c_i \mid i = 1, \ldots, k\}$. (Recall that a minimal cover of B is a prime of $g(y)$, and $g(y)$ gives all possible sub-covers of F.

Note: We are just doing tautology in constructing $\overline{g}(y)$, so unate reduction should work:

$$\begin{bmatrix} A & B \\ T & C \end{bmatrix}, \quad A \neq 1$$

Other efficiencies?

Back to Q-M

$$B = \begin{array}{cc} & \text{all primes} \\ \begin{array}{c}\text{minterms}\\ \text{of}\\ \text{f}\end{array} & \begin{bmatrix} 1\ 0\ 1\ 1\ 0\ 1\ 0 \\ \vdots \\ \vdots \end{bmatrix}\end{array}$$

$$B \cong \overline{g}(y) = \overline{y}_1\overline{y}_3\overline{y}_4\overline{y}_6 + \ldots$$

We want maximum prime of $g(y)$.

Note: A row of B says if we leave out primes

$$p_1, p_3, p_4, p_6$$

we **cease** to have a cover.

So basically, the only difference between Q-M and IRREDUNDANT is that for the latter, we just constructed a $\overline{g}(y)$ where we did not consider all primes, but only those in some cover

$$F = \{c_1, c_2, \ldots, c_k\}$$

EXPAND

$F \leftarrow \text{EXPAND}(F, R)$
Problem: Take a cube c and make it prime by removing literals.

a) Greedy way: (uses D and not R)

- remove literal l_i from c (results in $c \rightarrow \widetilde{c}$)
- test if $\widetilde{c} \subseteq f + d$ (same as $(f + d)_{\widetilde{c}} \equiv 1$)
- repeat, removing valid literals in order found.

b) Better way: (uses R and not D)
Want to see all possible ways to remove maximal subset of literals.
Idea: Create a function $g(y)$ such that $g(y) = 1$ if literals

$$\{l_i \mid y_i = 0\}$$

can be **removed** (*or* $\{l_i \mid y_i = 1\}$ *is a subset of literals such that if* **kept** *in* c, *will still make* $\widetilde{c} \subseteq f + d$ - *i.e.* $\widetilde{c} \cap r = \phi$).

Main Idea of EXPAND

1. Expand one cube ,c_i, at a time.

2. Build blocking matrix $B = B^{c_i}$

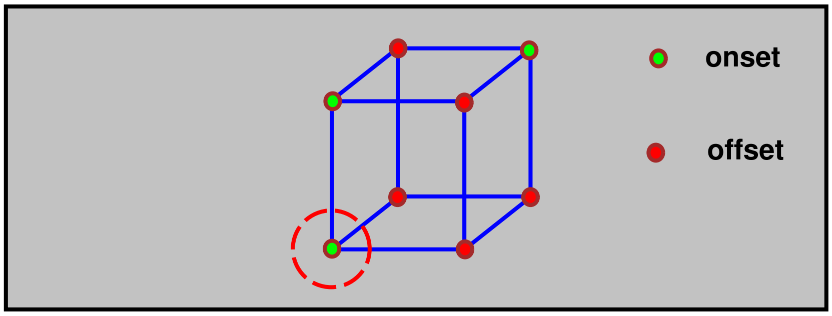
3. See which other cubes c_j can be feasibly covered using B.

4. Choose expansion (literals to be removed) to cover most c_j.

Note:

- $g(y)$ is monotone increasing
- $B \cong \overline{g}(y)$ is easily built if we have R, a cover of r.
- We do not need all of R. (*reduced offset*).

Example: (of "reduced" offset)



$$r = \overline{a}b + a\overline{b} + a\overline{c}$$

Make r unate by adding (1,1,1). then $R_{red} = a + b \cong \overline{g}(y)$ This is simpler and easier to deal with. (see paper A. Malik et. al. in reader)

Blocking Matrix B

Given $R = \{r_i\}$

$$B_{ij} = 1 \Longleftrightarrow \begin{cases} l_j \in C \text{ and } \bar{l}_j \in r_i \\ \bar{l}_j \in C \text{ and } l_j \in r_i \end{cases}$$

where $R = \{r_i\}$ is a cover of r. ($\mathcal{F} = (f, d, r)$).

What does row i of B say?
It says that if literals $\{j \mid B_{ij} = 1\}$ *are removed from* c*, then* $\widetilde{c} \cap r_i \neq \phi$, i.e. $B_{ij} = 1$ is one **reason** why c is orthogonal to offset cube r_i.

Thus $B \rightarrow \overline{g}(y) = \overline{y}_1\overline{y}_3\overline{y}_{10} + \ldots$ gives all ways that literals of c can be removed to get $\widetilde{c} \not\subseteq f + d$.
($\equiv \widetilde{c} \cap r \neq \phi$).

EXPAND Example

$$\begin{aligned} c &= ab\overline{d} \\ r_i &= \overline{a}bd\overline{e} \\ y_1 y_2 y_3 &\sim a, b, \overline{d} \end{aligned}$$

Suppose $\overline{g}(y) = 1$. If $y_1 = 1$, we keep literal a in cube c

$$\begin{aligned} y_1 = 1 &\iff keep\ a \\ y_2 = 1 &\iff keep\ b \\ y_3 = 1 &\iff keep\ \overline{d} \end{aligned}$$

$$(B_i) = 101 = \overline{y}_1 \overline{y}_3 + \ldots = \overline{g}(y)$$

means do not keep literals 1 and 3 of c (*implies that subsequent* $\widetilde{c}$ *is not an implicant*) If literals 1,3 are removed we get $c \to \widetilde{c} = b$. But $\widetilde{c} \cap r_i \neq \phi$:

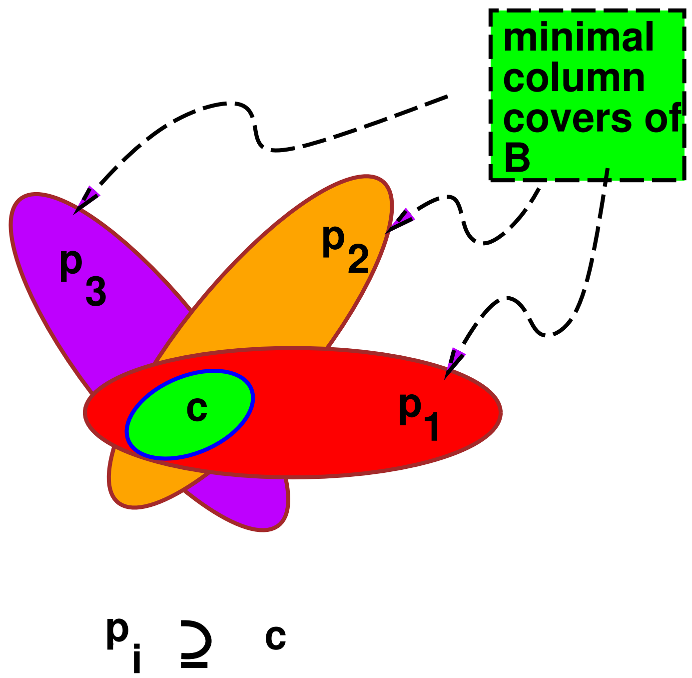
$$b \times \overline{a}bd\overline{e} = \overline{a}bd\overline{e} \neq \phi$$

so obviously b is not an implicant.

EXPAND -continued

Thus **all minimal column covers** ($\cong g(y)$) of B are the minimal subsets of literals of c that must be kept to ensure that $\tilde{c} \subseteq f + d$ (i.e. $\tilde{c} \cap r = \phi$)

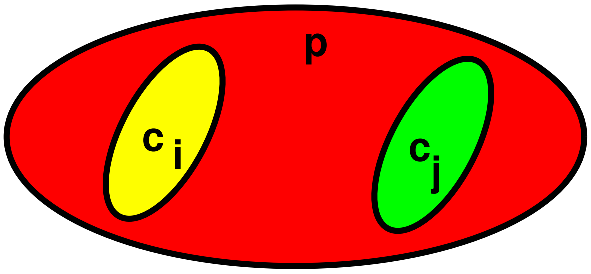
Thus, each minimal column cover is a prime p that covers c, i.e. $p \supseteq c$.



Expanding c_i

$F = \{c_i\},\ \ \mathcal{F} = (f, d, r)\ \ f \subseteq F \subseteq f + d$

Q: Why do we want to expand c_i?
A: To cover some other c_j's



Q: Can we cover c_j?
A: If and only if (*SCC = "smallest cube containing" also called "supercube"*)

$$SCC(c_i \cup c_j) \subseteq f + d$$

equivalent to:

$$SCC(c_i \cup c_j) \cap r = \phi$$

equivalent to:
literals conflicting between c_i, c_j can be removed and still have an implicant

Can check this with blocking matrix:

$$\begin{aligned} c_i &= 12012 \\ c_j &= 12120 \end{aligned}$$

implies that literals 3 and 4 must be removed for $\tilde{c}_i$ to cover c_j. Check if column 3,4 of B can be removed without causing a row of all 0's.

Covering Function

The objective of EXPAND is to expand c_i to cover as many cubes c_j as possible. The blocking function $\overline{g}(y) = 1$ whenever the subset of literals $\{l_j \mid y_j = 1\}$ yields a cube $\widetilde{c} \not\subseteq f + d$. (Note that $\widetilde{c} = \prod_{(y_j=1)} l_j$).

We now build the covering function, h, such that

$$h(y) = 1$$

whenever the cube $\widetilde{c} \supseteq c_i$ covers another cube $c_j \in F$.

Thus a minterm $m \in g(y)h(y)$ is such that it gives $\widetilde{c} \subseteq f + d$ $(g(m) = 1)$ and covers at least one cube $(h(m) = 1)$. In fact every cube $\widetilde{c}_m \supseteq c_l$ is covered. We seek m which results in the most cubes covered.

Thus define $h(y)$ by a set of cubes where $d_k = k^{th}$ cube is:

$$d_k = \phi \text{ if } SCC[c_i \cup c_k] \not\subseteq f + d \text{ else}$$

$$d_k^j = \begin{cases} \overline{y}_j \text{ if } c_k^j \not\subseteq c_i^j \text{ i.e. } \begin{array}{l} 2 \not\subseteq 1 \\ 0 \not\subseteq 1 \\ 1 \not\subseteq 0 \end{array} \\ 2 \text{ otherwise} \end{cases}$$

(*Thus each d_k gives minimal expansion to cover c_k*)

Thus $d_k \neq \phi$ if cube c_k can be feasibly covered by expanding cube c_i. d_k says which literals we have to leave out to minimally cover c_k. Thus $h(y)$ is defined by $h(y) = d_1 + d_2 + \ldots + d_{|F|-1}$ (one for each cube of F, except c_i).

We want $m \in h(y)g(y)$ contained in a maximum number of d_l's.

In the "Espresso" book, we build a Boolean covering matrix C (note that $h(y)$ is unate negative) representing $h(y)$ and solve this problem with greedy heuristics. (Note that h is unate.)

$$C = \begin{pmatrix} \cdots \\ 010110 \\ 101011 \\ 100101 \\ \cdots \end{pmatrix}$$

$$B = \begin{pmatrix} \cdots \\ 110110 \\ 101010 \\ 101001 \\ \cdots \end{pmatrix}$$

Note: $B \simeq \overline{g}(y)$ but $C \simeq \widetilde{h}(y) \supseteq h(y)$

Want set of columns such that if eliminated from B and C results in no empty rows of B and a maximum of empty rows in C.

(**Note:** A 1 in C can be interpreted as a reason why $\widetilde{c}$ does not cover c_j).

Endgame

What do we do if $h(y) \equiv 0$?

Some things to try:

- generate largest prime
- cover most care points of another cube
- coordinate two or more cube expansions, i.e. try to cover another cube by a combination of several other cube expansions

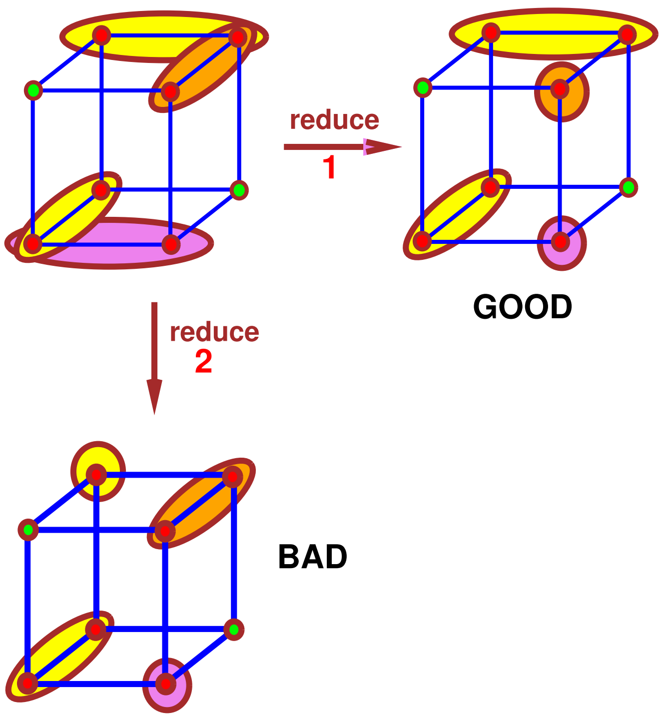
So far we have not really come up with a really satisfactory endgame. This could be important in many hard problems, since it is often the case that $h(y) \equiv 0$.

REDUCE

Problem: Given a cover F and $c \in F$, find the **smallest** cube $\underline{c} \subseteq c$ such that

$$F + \{\underline{c}\} - \{c\}$$

is still a cover. $\underline{c}$ is called the **maximally** reduced cube of c.



Example 5 $F = ac + bc + \bar{b}\bar{c} + \bar{a}\bar{c}$
Two orders:

1. *REDUCE*$(F = \{ac, bc, \bar{b}\bar{c}, \bar{a}\bar{c}\}) = a\bar{b}c + bc + a\bar{b}\bar{c} + \bar{a}\bar{c}$

2. *REDUCE*$(F = \{bc, \bar{b}\bar{c}, ac, \bar{a}\bar{c}\}) = \bar{a}bc + a\bar{b}\bar{c} + ac + \bar{a}\bar{c}$

REDUCE is **order dependent**.

REDUCE Algorithm

```
REDUCE(F, D) {
    F ← ORDER( F )
    for (1 ≤ j ≤ |F|) do {
        c_j ← MAX_REDUCE(c, F, D)
        F ← F ∪ {c_j} − c_j
    }
    return F
}
```

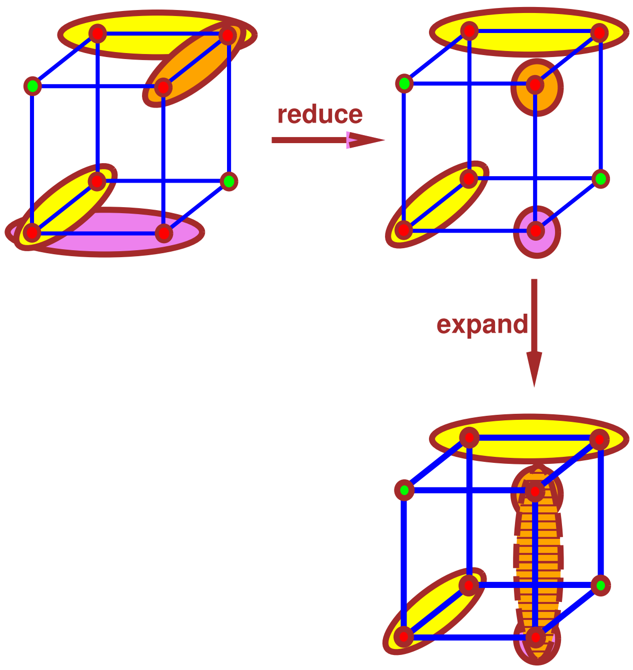
REDUCE

Main Idea: Make a prime not a prime but still maintain a cover: $\{c_1, \ldots, c_i, \ldots, c_k\} \to \{c_1, c_2, \ldots, \underline{c}_i, c_{i+1}, \ldots, c_k\}$
But

$$f \subseteq \sum_{j=0}^{i-1} c_j + \underline{c}_i + \sum_{j=i+1}^{k} c_j \subseteq f + d$$

- Get out of a local minimum (*prime and irredundant is local minimum*)

- Then have nonprimes, so can expand again in different directions. (*Since EXPAND is "smart", it will know which direction*)



REDUCE - continued

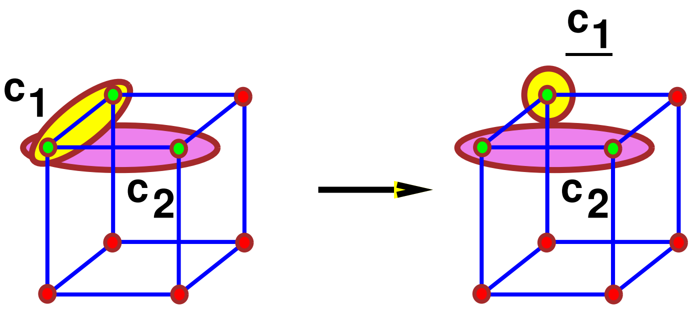
$$
\begin{aligned}
F &= \{c_1, c_2, \ldots, c_k\} \\
F(i) &= F + D - \{c_i\} \\
&= \{c_1, c_2, \ldots, c_{i-1}, c_{i+1}, \ldots, c_k\} + D
\end{aligned}
$$

Reduced cube:

$$\underline{c_i} = \text{smallest cube containing}(c_i \cap \overline{F(i)})$$

Note that $c_i \cap \overline{F(i)}$ is the set of points uniquely covered by c_i (and not by any other c_j or D.)

Thus, $\underline{c_i}$ is the smallest **cube** containing the minterms of c_i which are not in $F(i)$.



(*SCC == "supercube"*)

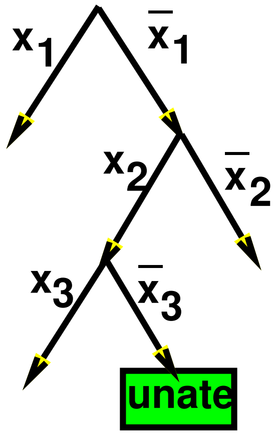
$$
\begin{aligned}
\underline{c_i} &= \text{SCC}(c_i \cap \overline{F(i)}) \\
&= \text{SCC}(c_i \overline{F_{c_i}(i)}) \\
&= c_i \text{SCC}(\overline{F_{c_i}(i)}) \\
&= c_i \text{SCCC}(F_{c_i}(i))
\end{aligned}
$$

where SCCC = "smallest cube containing complement".

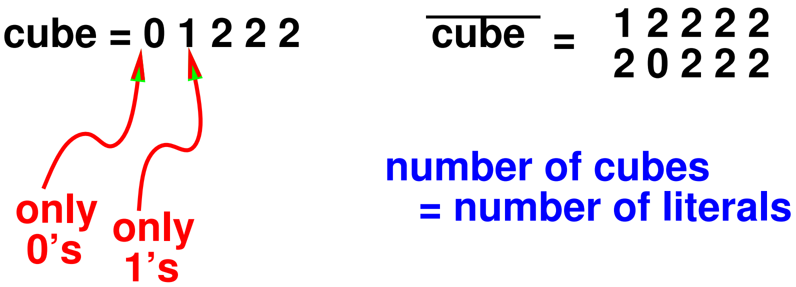
Efficient Algorithm for SCCC

Unate Recursive Paradigm:

- select most binate variable
- cofactor
- until unate leaf



What is SCCC(unate cover)?
Note that the SCCC(c), where c is a cube with at least 2 literals, is the universe:



SCCC

SCCC(U) = γ.
Claim: if unate cover has

- row of all 2's except one 0, then complement is in x_i, i.e. $\gamma_i = 1$
- all 2's except one 1, complement is in $\overline{x}_i$, i.e. $\gamma_i = 0$
- otherwise in both subspaces, i.e. $\gamma_i = 2$.

$$\begin{aligned} SCCC(c_1 + c_2 \ldots c_k) &= SCC(\overline{c}_1 \cap \ldots \cap \overline{c}_k) \\ &= SCC(\overline{c}_1) \cap \ldots \cap SCC(\overline{c}_k) \end{aligned}$$

implies that only need to look at 1-literal cubes.

Example:

$$f = a + bc + \overline{d} \Longrightarrow \overline{f} = \overline{a}(\overline{b} + \overline{c})d \subseteq \overline{a}d$$

Note: 0101 and 0001 are both in $\overline{f}$. So SCCC could not have literal b or $\overline{b}$ in it.

Example

$$
U \text{ (unate)} = \begin{array}{ccccc} 2 & 2 & 2 & 2 & 0 \\ 0 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 2 & 2 \\ 2 & 1 & 2 & 1 & 0 \\ \uparrow & & & & \uparrow \end{array}
$$

Note that columns 1 and 5 are essential: they must be in every minimal cover. So $\overline{U} = x_1 x_5(\ldots)$. Hence

$$
\text{SCCC}(U) = x_1 x_5
$$

Proof. (sketch)

$$
U = \overline{x}_1 + \overline{x}_5 + x_2(x_3 + x_4)
$$

$$
\overline{U} = x_1 x_5(\overline{x}_2 + \overline{x}_3 \overline{x}_4)
$$

$$
\overline{U} = \begin{array}{ccccc} 1 & 0 & 2 & 2 & 1 \\ 1 & 2 & 0 & 0 & 1 \\ & \uparrow & \uparrow & \uparrow & \end{array} \subseteq 12221
$$

The marked columns contain both 0's and 1's. But every prime of $\overline{U}$ contains literals x_1, x_5

$$
\overline{U} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}
$$

□

SCCC

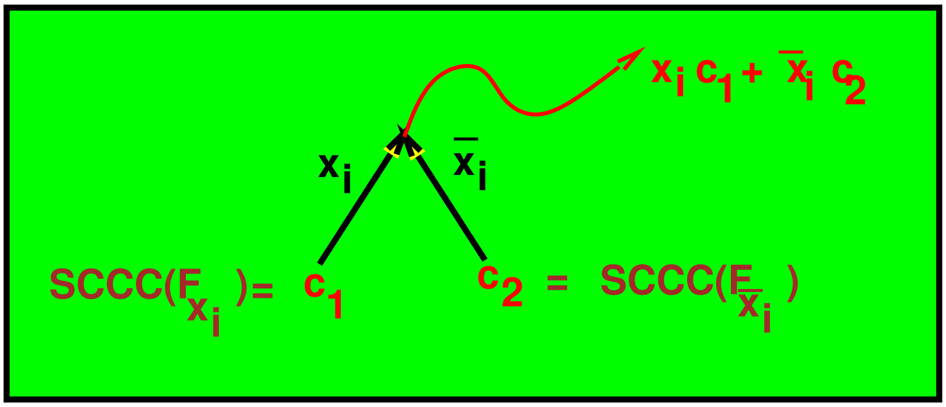
Thus:

$$\gamma = \text{SCCC(unate)} = \phi \text{ if row of all 2's}$$

$$\gamma_j = \begin{cases} l_j & \text{if } j \text{ has a row singleton with a 0 in it} \\ \bar{l}_j & \text{if } j \text{ has a row singleton with a 1 in it} \\ 2 & \text{otherwise} \end{cases}$$

Thus unate leaf is easy!

Merging



We need to produce SCCC(f) =

$$\text{SCC}(x_i c_1 + \overline{x}_i c_2) = \gamma$$

$$\begin{array}{rcl} \gamma & = & \{l_1, l_2, \ldots, l_k\} \\ x_i \in \gamma & \Longleftrightarrow & c_2 = \phi \\ \overline{x}_i \in \gamma & \Longleftrightarrow & c_1 = \phi \\ l_{j \neq i} \in \gamma & \Longleftrightarrow & (l_j \in c_1) \wedge (l_j \in c_2) \end{array}$$

If $c_1 \wedge c_2 \neq \phi$, then points in both x_i and $\overline{x}_i$, so

$$(\text{SCC}(x_i c_1 + \overline{x}_i c_2))_i = 2$$

If $l_j \notin c_1$, or $l_j \notin c_2$, then both x_j and $\overline{x}_j$ points exist, hence $\gamma_j = 2$.

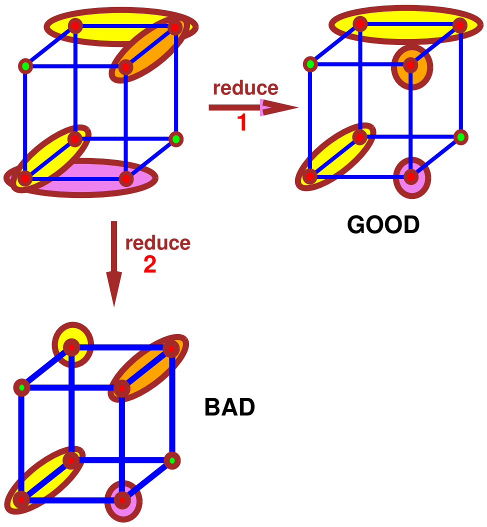
Also if $l_j \in c_1$ and $\bar{l}_j \in c_2$, then $\gamma_j = 2$.

ESPRESSO - Heuristic Two-Level Minimization

```
ESPRESSO(𝓕)
{
    (F, D, R) ← DECODE(𝓕)
    F ← EXPAND(F, R)
    F ← IRREDUNDANT(F, D)
    E ← ESSENTIAL_PRIMES(F, D)
    F ← F − E; D ← D + E
    do{
        do{
            F ← REDUCE(F, D)
            F ← EXPAND(F, R)
            F ← IRREDUNDANT(F, D)
        } while fewer terms in F
        //LASTGASP
        G ← REDUCE_GASP(F, D)
        G ← EXPAND(G, R)
        F ← IRREDUNDANT(F + G, D)
        //LASTGASP
    }while G ≠ φ
    F ← F + E; D ← D − E
    LOWER_OUTPUTS(F, D)
    RAISE_INPUTS(F, R)
    error ← (F_on ⊈ F) or (F ⊈ F_old + D)
    return(F,error)
}
```

LASTGASP

Reduce is order dependent:



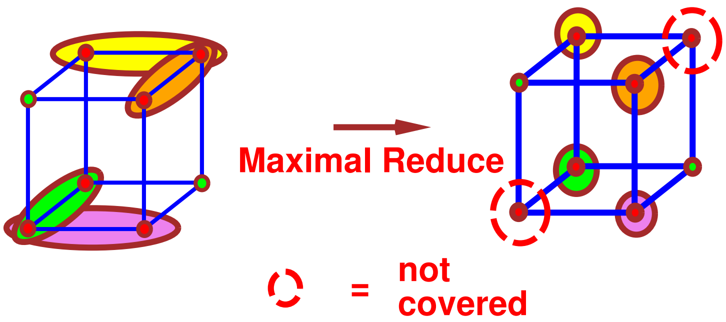
Expand can't do anything with the reduction produced by REDUCE 2.

Maximal Reduce:

$$\underline{c}_i^M = \text{SCC}(c_i \cap \overline{F(i)}) = c_i \cap \text{SCCC}(F(i)_{c_i})$$

i.e. we reduce all cubes as if each were the first.

Note:
$\{\underline{c}_1^M, \underline{c}_2^M, \ldots\}$ is not a cover



LASTGASP - continued

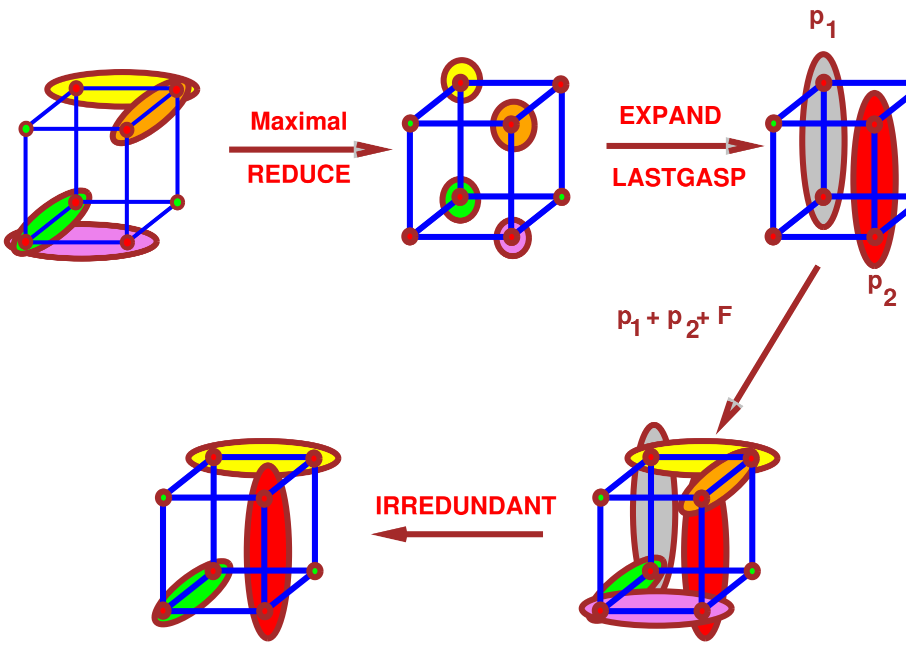
Now expand $\underline{c}_i^M$ using EXPAND, but try to cover only $\underline{c}_j^M$'s. (*Note here we call EXPAND*(G, R)*, where* $G = \{\underline{c}_1^M, \ldots, \underline{c}_k^M\}$) If a covering is possible, take the resulting prime:

$$f + d \supseteq p_i \supseteq \underline{c}_i^M \cup \underline{c}_j^M$$

Add to F:

$$\widetilde{F} = F \cup \{p_i\}$$

Since F is a cover, so is $\widetilde{F}$. Now make $\widetilde{F}$ irredundant (using IRREDUNDANT).



What about "supergasp"?
Main Idea: Generally, think of ways to throw in a few more primes and then use IRREDUNDANT. If generate all primes, then just Q-M.

Goldberg Variation

Property of LASTGASP:
Let $\underline{F}^i$ = maximal reduction of $F^i \in F$. LASTGASP searches for maximally reduced cubes $\underline{F}^i$ and $\underline{F}^k$ such that

$$\underline{F}^{ik} \equiv \text{ supercube}(\underline{F}^i, \underline{F}^k)$$

is an implicant of $\mathcal{F}$. It then produces a new prime (which is not in F) which covers supercube($\underline{F}^i, \underline{F}^k$).

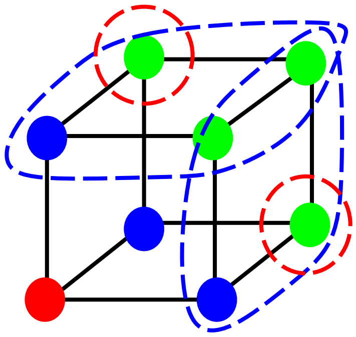
However, this does not guarantee that the size of F decreases on the next IRREDUNDANT by at least one.

EXAMPLE:

$$F = \begin{matrix} 1\ 2\ 2 \\ 2\ 2\ 1 \end{matrix} \qquad D = \begin{matrix} 1\ 0\ 0 \\ 0\ 1\ 0 \\ 0\ 0\ 1 \end{matrix}$$

$$\underline{F}^1 = 1\ 1\ 0 \qquad \underline{F}^2 = 0\ 1\ 1$$

$$\underline{F}^{12} = 2\ 1\ 2 = \text{prime}$$



Goldberg Theorem

$$\underline{F}^{ik} \equiv \text{ supercube}(\underline{F}^i, \underline{F}^k)$$

Theorem 1 *Let*

$$\begin{aligned} A^{ik} &= (f \cap F^i \cap F^k) - \{F - \{F^i, F^k\}\} \\ &= (F^i \cap F^k) - \{D + F - \{F^i, F^k\}\} \end{aligned}$$

*(*i.e. A^{ik} is the set of ON-set minterms covered by $F^i \cap F^k$ and not intersected with any other cube from F.*)*
Then F^i and F^k can be replaced by a prime $P^{ik} \supseteq \underline{F}^{ik}$ if and only if

$$P^{ik} \supseteq A^{ik}$$

Thus for a reduction in the size of the cover, we need a prime with the property

$$P^{ik} \supseteq \text{ supercube}(\underline{F}^i, \underline{F}^k) \cup A^{ik}$$

An algorithm based on this was proposed by E. Goldberg of the Byelorussion Academy of Sciences, Institute of Engineering Cybernetics in Minsk, Former Soviet Union. He reported it was less expensive than SUPER_GASP and more effective than LASTGASP.