Multi-Valued Logic

- Up to now...two-valued synthesis
  - Binary variables take only values $\{0, 1\}$
- Multi-valued synthesis
  - Multivalued variable X_i can take on values $P_i = \{0, ..., |P_i| - 1\}$ (integers - but no order implied)
  - Symbolic variables take values from symbolic set, e.g. state: $\{s_0, s_1, \ldots, s_n\}$ or $X : \{a, b, c\}$.
- Formally: $\mathcal{F} : P_1 \times ... \times P_n \mapsto \{0, 1, *\}$ (*sometimes called an* **mv-function**).
  $\mathcal{F} = (f, d, r)$
- **Problem:** find the minimum (SOP) form for an incompletely-specified function of this kind

Big News: *Nothing (much) changes*

Example "Truth Table"

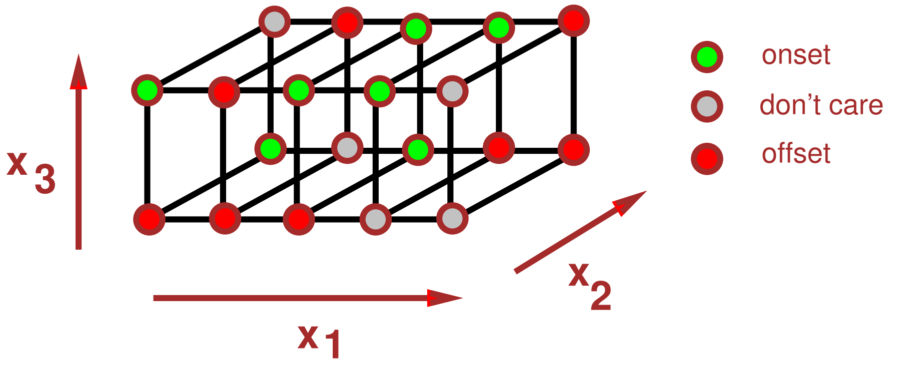
$P_1 = \{0, 1, 2\}, P_2 = \{0, 1\}$

X_1	X_2	$\mathcal{F}$
0	0	1
0	1	1
1	0	0
1	1	1
2	0	ϕ
2	1	1

Here "2" means the value 2 and not $\{0, 1\}$.

$$\begin{aligned} f(0,0) &= 1 \\ f(2,1) &= 1 \\ f(1,0) &= 0 \\ f(2,0) &= \phi \textit{ unspecified} \text{ (don't cares)} \end{aligned}$$

MV Function (different one):



Terminology

- **Vertex**: A point of the space $P_1 \times ... \times P_n$
- **Cube**: A set $c = c_1 \times ... \times c_n$, $c_i \subseteq P_i$
- **Containment**: $c \supseteq d$ (c, d cubes) iff $c_i \supseteq d_i, \forall i$
  - When c is a vertex, $d \supset c$, also say $c \in d$
- **Minterm**: Vertex v such that $f(v) = 1$. If $v = (c_1, \ldots, c_n)$ then each c_i has exactly one 1.
- **Implicant**: Cube c such that for all $v \in c, \mathcal{F}(v) \neq 0$
- **Prime Implicant**: Implicant c such that there is no implicant d such that $d \supset c$
- **Cover** of $\mathcal{F}$: A set of implicants which together contain every minterm of f
- **Prime Cover** of $\mathcal{F}$: A cover, each of whose elements is prime
- **Distance** of cubes c, d:

$$\delta(c, d) = \sum_{i=1}^{n} (c_i \cap d_i = \emptyset)$$

- **Consensus** of cubes c, d, $c \odot d = \emptyset$ if $\delta(d, c) > 1$. Otherwise:

$$c \odot d = \sum_{i=1}^{n} c_1 \cap d_1 \times ... \times c_i \cup d_i \times ... \times c_n \cap d_n$$

  Note: If $\delta(c, d) = 1$, one cube; $\delta(d, c) = 0, n$ cubes

- **Supercube of** c, d,

$$\gamma(c, d) = c_1 \cup d_1 \times ... \times c_n \cup d_n$$

Note: All these definitions are exactly as they were in the (binary) case, (except consensus)

Notation – MV-Literals

- As before, want a cover of $\mathcal{F}$ (in SOP form).

- Cube $c = c_1 \times ... \times c_n$ written:

  $$X_1^{c_1} X_2^{c_2} ... X_n^{c_n}$$

  (*product of "mv-literals"*). Literal is a logic function of form

  $$X_i^{c_i} = (X_i = \gamma_1) + \ldots + (X_i = \gamma_k)$$

  where $\gamma_j \in c_i \subseteq P_i$.

- If $c_i = P_i$ we may omit $X_i^{P_i}$ from the expression (*since* $X_i^{P_i} = 1$)

- Note analogy to two-valued case:

  $$\begin{array}{rcl} X_i^{\{0\}} & = & \overline{X}_i \\ X_i^{\{1\}} & = & X_i \\ X_i^{\{0,1\}} & = & - \quad = 1 \end{array}$$

Actually, multi-valued notation is superior to old (binary).

Example

X_1	X_2	$\mathcal{F}$		
0	0	1	a	
0	1	1	a	b
1	0	0		
1	1	1		b
2	0	ϕ	a	
2	1	1	a	b

Can form rows marked a (b) as a single mv-cube implicant:

- Cube $a = \{0,2\} \times \{0,1\} = X_1^{\{0,2\}} X_2^{\{0,1\}} = X_1^{\{0,2\}}$
- Cube $b = \{0,1,2\} \times \{1\} = X_1^{\{0,1,2\}} X_2^{\{1\}} = X_2^{\{1\}}$
- Both a and b are prime implicants
- $X_1^{\{0\}} X_2^{\{0,1\}} = X_1^{\{0\}}$ is an implicant of $\mathcal{F}$.
- The following are cube covers of $\mathcal{F}$. F_2 is a prime cover.

$$
\begin{aligned}
F_1 &= X_1^{\{0\}} + X_1^{\{1,2\}} X_2^{\{1\}} \\
F_2 &= X_2^{\{1\}} + X_1^{\{0,2\}}
\end{aligned}
$$

Positional Notation

- Example: cube $X_1^{\{A,B\}} X_2^{\{R\}}$
  $P_1 = \{A, B, C, D\}, P_2 = \{R, S\} \leftarrow$ symbolic

		A	B	C	D	R	S
cube	$\rightarrow$	1	1	0	0	1	0

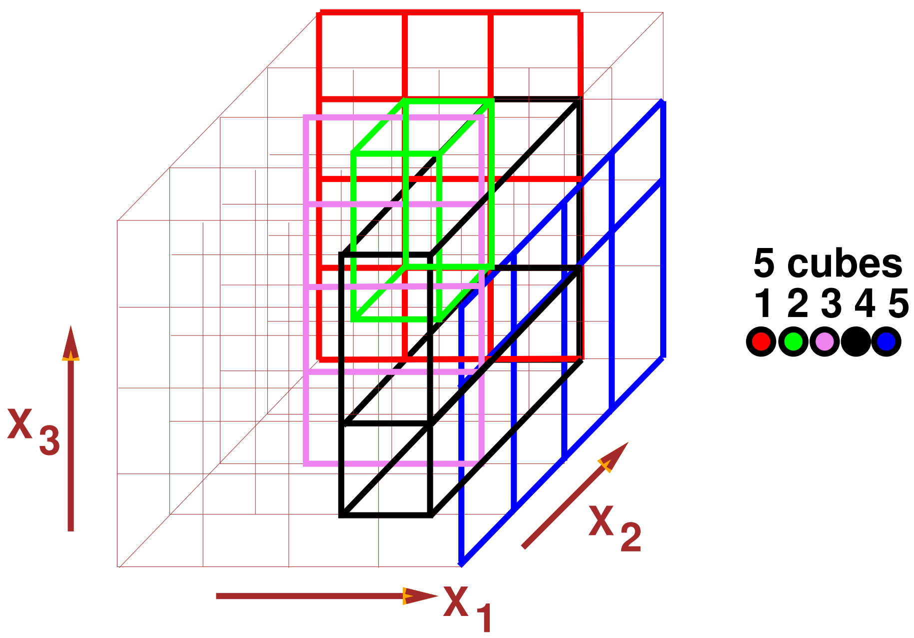
- A cube does not depend on variable X_i if it has all 1's in its set of columns associated with X_i.

- Each of the columns of a variable is called a **part** of that variable. There is one part for each value a variable can take.

- Extension of Espresso notation

$(value = 0)$	$(value = 1)$		
0	1	$\leftarrow$	1
1	0	$\leftarrow$	0
1	1	$\leftarrow$	2

Example



	X_1	X_2	X_3
c_1	11110	00001	11111
c_2	01100	00011	01010
c_3	01010	00100	11111
c_4	00110	01001	11010
c_5	00001	11111	10110

Minimization Problem for Multi-Valued Logic

- **Given** a cover F of $\mathcal{F}$ and D a cover of the don't-care set d,
- **Find** A minimum sum-of-products form for $\mathcal{F}$

Same problem as for two-valued

- Generate primes of $f + d$
- Generate covering table
- Solve the covering table

Same algorithms as for two-valued (except for small details) can be used.

Applications of Multi-Valued Logic

Theorem (Hong): minimizing two-valued n input $\times$ m output logic is equivalent to minimizing a function:

$$f : \{0,1\} \times ... \times \{0,1\} \times \{0,...,m-1\} \mapsto \{0,1\}$$

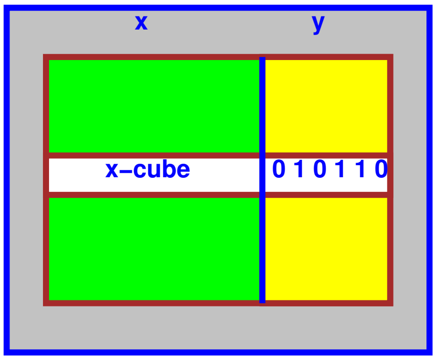
i.e., a single-output function with n binary inputs and one multi-valued input

Right way to do multiple output functions

Proof (sketch): Let $g = \{f_0, \ldots, f_{m-1}\}$ be the multiple output function. Consider the **characteristic** function f of the multiple output function, defined on $n+1$ variables with the last one being multi-valued $\{0, 1, \ldots, m-1\}$

$$f(x,y) = 1 \Longleftrightarrow \prod_{i=0}^{m-1} ((y = i) \Rightarrow f_i(x))$$

Now note that an implicant of g (the multi-output function) is a cube c in the x-space where each output is turned **on** only if $f_i(c) = 1$. The output not turned on is **no information**, since the final output is the OR of all the cube outputs.



Other Applications: Encoding Problems

Problems

- Input Encoding problem
  - bit-paired PLA structure
- Output encoding problem
  - output phase optimization
- State encoding problem
  - Minimize symbolically to get constraints on binary code
  - Solve constraints to derive binary code
  - Re-minimize binary problem
  - Implement in binary

Multi-Valued Minimization Example

$$
\begin{aligned}
f_0 &= \overline{X}_1\overline{X}_2\overline{X}_3 + \overline{X}_1 X_2\overline{X}_3 + \overline{X}_1 X_2 X_3 + X_1 X_2 X_3 \\
f_1 &= \overline{X}_1\overline{X}_2\overline{X}_3 + \overline{X}_1\overline{X}_2 X_3 + \overline{X}_1 X_2 X_3 + X_1\overline{X}_2 X_3 \\
f_2 &= \overline{X}_1\overline{X}_2\overline{X}_3 + \overline{X}_1\overline{X}_2 X_3 + \overline{X}_1 X_2\overline{X}_3 + X_1 X_2 X_3
\end{aligned}
$$

$$
\begin{array}{lccccccccc}
f: & \{0,1\} & \times & \{0,1\} & \times & \{0,1\} & \times & \{0,1,2\} & \mapsto & \{0,1\} \\
f: & P_1 & \times & P_2 & \times & P_3 & \times & P_4 & \mapsto & \{0,1\}
\end{array}
$$

$$
\begin{aligned}
f = \; & X_1^0X_2^0X_3^0X_4^0 + X_1^0{X_2}^1X_3^0X_4^0 + X_1^0X_2^1X_3^1X_4^0 + \\
& X_1^1X_2^1X_3^1X_4^0 + X_1^0X_2^0X_3^0X_4^1 + X_1^0{X_2}^0X_3^1X_4^1 + \\
& X_1^0X_2^1X_3^1X_4^1 + X_1^1X_2^0X_3^1X_4^1 + X_1^0X_2^0X_3^0X_4^2 + \\
& X_1^0{X_2}^0X_3^1X_4^2 + X_1^0X_2^1X_3^0X_4^2 + X_1^1X_2^1X_3^1X_4^2
\end{aligned}
$$

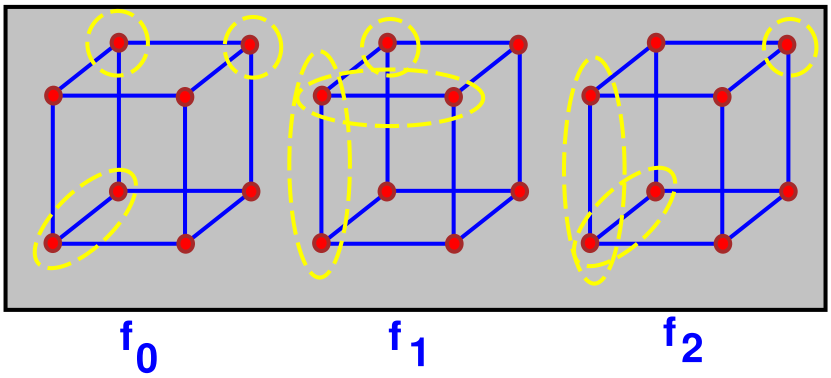
Example - After Minimization:

Prime and irredundant SOP of f:

$$\begin{aligned} f &= X_1^0 X_2^1 X_3^1 X_4^{\{0,1\}} + X_1^1 X_2^1 X_3^1 X_4^{\{0,2\}} + \\ & X_1^0 X_2^{\{0,1\}} X_3^0 X_4^{\{0,2\}} + X_1^0 X_2^0 X_3^{\{0,1\}} X_4^{\{1,2\}} \\ & + X_1^{\{0,1\}} X_2^0 X_3^1 X_4^1 \end{aligned}$$

(five cubes $1+2+3+4+5$)
Equivalent to:

$$\begin{aligned} f_0 &= 1+2+3 \\ &= \overline{X_1} X_2 X_3 + X_1 X_2 X_3 + \overline{X_1 X_3} \\ f_1 &= 1+4+5 \\ &= \overline{X_1} X_2 X_3 + \overline{X_1 X_2} + \overline{X_2} X_3 \\ f_2 &= 2+3+4 \\ &= X_1 X_2 X_3 + \overline{X_1 X_3} + \overline{X_1 X_2} \end{aligned}$$



Note: $X_1 X_2 X_3$ is **not** a prime of f_0, but $X_1 X_2 X_3 X_4^{\{0,2\}}$ is a prime of f. Similarly for $\overline{X_1} X_2 X_3$.

Shannon Cofactor

Cofactor of cube c with respect to cube d (c_d)

$$c_d = \begin{cases} \emptyset & \text{if } c \cap d = \emptyset \\ c_1 \cup \overline{d_1} \times \ldots \times c_n \cup \overline{d_n} & \text{otherwise} \end{cases}$$

Note: $\overline{d}_i = P_i - d_i$

- Notice this agrees with "standard" cofactor in the case of two-valued
  - Hint: check cases on d_i, c_i, e.g. if $d_i = c_i = 1$, then $c_i \cup \overline{d}_i = 2$
- Rationale: Only care about value of c on places *addressed* by d ($\overline{d}$ *is don't care*)
- Example: space is $\{0, 1\} \times \{0, 1, 2\}$

$$(X_1^0 X_2^{\{0,2\}})_{X_1^{\{0,1\}} X_2^{\{2\}}} = X_1^0 X_2^{\{0,1,2\}} = X_1^0$$

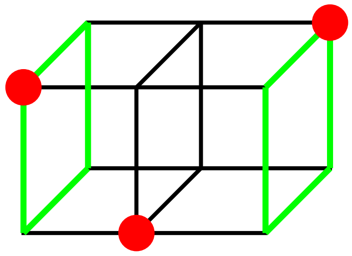
Cofactor of cover $F = c^1 + \ldots + c^m$ with respect to cube d is

$$F_d = c_d^1 + \ldots + c_d^m$$

Note that f_c can be obtained by representing f in truth table form and cofactoring each term.

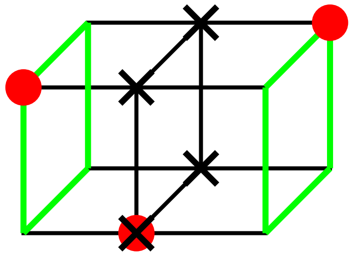
Example

$\mathcal{F} = (f, r)$ and cube $d = X_1^{\{0,2\}}$

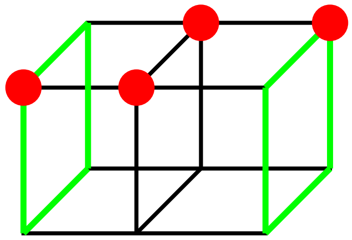


Consider the generalized cofactor:

$$Co(\mathcal{F}, d) = (fd, \overline{d}, rd)$$



$$\begin{aligned} f_d &= (X_1^{\{0\}}X_2^{\{0\}}X_3^{\{1\}} + X_1^{\{1\}}X_2^{\{0\}}X_3^{\{0\}} + X_1^{\{2\}}X_2^{\{1\}}X_3^{\{1\}})_d \\ &= X_1^{\{0,1\}}X_2^{\{0\}}X_3^{\{1\}} + X_1^{\{1,2\}}X_2^{\{1\}}X_3^{\{1\}} \end{aligned}$$



We try to keep all the onset and project the care onset to $\overline{d}$

Note: $f \not\subseteq f_c$.

Shannon Cofactor Expansion Theorem (General Case)

Let f any function and $\{c^1, \ldots, c^t\}$ any set of cubes such that

$$\begin{aligned} \sum_{i=1}^{t} c^i &= 1 \textit{ and} \\ c^i \cap c^j &= \emptyset \textit{ for } i \neq j \end{aligned}$$

i.e. a cube partition of space. Then

$$f = \sum_{i=1}^{t} c^i f_{c^i}$$

We immediately have:

- $f \equiv 1$ iff $f_{c^i} = 1$ for every i
- $\overline{f} = \sum_{i=1}^{t} c^i \overline{f_{c^i}}$. Note that $(\overline{f})_c \neq \overline{(f_c)}$, but $c(\overline{f})_c = c\overline{(f_c)}$.
- ...etc..

i.e. most Shannon cofactor results continue to hold. However, note $(\overline{f})_c \neq \overline{(f_c)}$, but $c(\overline{f})_c = c\overline{(f_c)}$.

Recursive Paradigm: Multi-Valued Version

Tautology(f)
  if $f \equiv 1$ return true;
  if $f \equiv 0$ return false;
  Choose $c^1, ..., c^t$ such that
    $c^i \neq c^j$ (*don't need* $c^i \cap c^j = \phi$*for this*)
    $1 = \sum_{i=1}^{t} c^i$
  for($i = 1, \ldots, t$)
    if (Tautology(f_{c^i}) = false)
      return false;
  return true;

Still open:

- Unate leaves (*what does unateness mean?*)
- Splitting choice (*i.e. which* c^i)
- Unate Reduction?

Unateness: Multi-Valued

Definition 1 *f is said to be* **weakly unate** *in X_i if there exists some value $= j$ such that changing X_i from value $= j$ to any other value causes f to change (if it changes) from 0 to 1 (f does not decrease)*

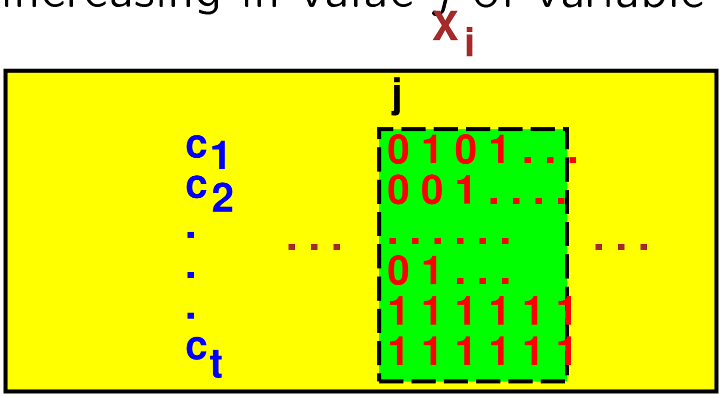
- Analog to unateness in two-valued case set $j = 0$ and get monotone increasing; set $j = 1$ and get monotone decreasing
- In general: detecting unateness is hard (obviously)
- Special case: unate cover.

Weakly-Unate Cover

Definition 2 *A* **cover** $F = c^1 + ... + c^t$ *is said to be* **weakly unate** *in* X_i *iff there is some* j *such that, for each cube* c^k*, either*

$$\begin{aligned} j &\notin c_i^k \text{ or} \\ c_i^k &= P_i \text{ (universal set)} \end{aligned}$$

*(*monotone increasing in value j of variable X_i*).*



Analogy to two-value:

- Rewrite

$$\begin{aligned} \overline{X}_i &\rightarrow X_i^{\{0\}} \\ X_i &\rightarrow X_i^{\{1\}} \\ 2 &\rightarrow X_i^{\{0,1\}} \end{aligned}$$

- Example:

$$\begin{array}{ccc} X_i & & X_i \\ 0 & & 10 \\ 2 & \rightarrow & 11 \\ 0 & & 10 \\ 2 & & 11 \end{array}$$

Here $j = 1$ i.e. monotone increasing from $j = 1$.

$$\begin{array}{ccc} X_i & & X_i \\ 1 & & 01 \\ 1 & \rightarrow & 01 \\ 2 & & 11 \end{array}$$

Here $j = 0$ i.e. monotone increasing from $j = 0$

Weakly-Unate Cover

Easy to detect:

for each variable X_i
    $\nu_i = P_i$ (*start with all values*)
    for each cube c^k
        if $c_i^k \neq P_i$ (*if not all values* P_i)
            $\nu_i = \nu_i - c_i^k$ (*remove any value in* c^k)

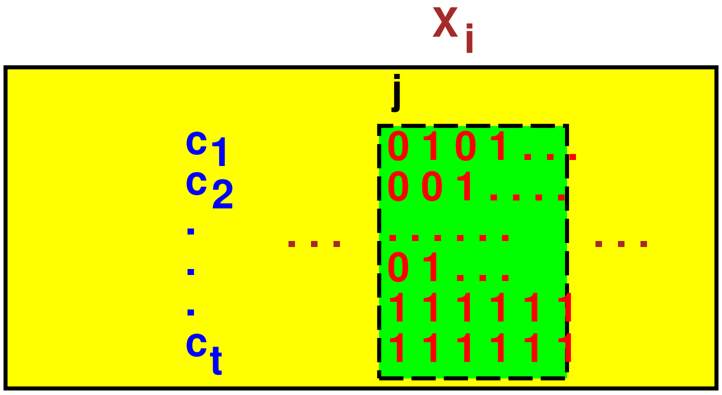
Unate variables are those for which $\nu_i \neq \emptyset$

(*Just looking for a column with all 0's, except for rows of all 1's*)

Also another way:

1. throw out rows of all 1's

2. look for column of all 0's



Example

$$\begin{aligned}F &= X_1^{\{0,1,2,3,4\}} X_2^{\{4\}} X_3^{\{0,1,2,3\}} + \\ & X_1^{\{1,2\}} X_2^{\{3,4\}} X_3^{\{1,3\}} + \\ & X_1^{\{1,3\}} X_2^{\{2\}} X_3^{\{0,1,2,3,4\}} + \\ & X_1^{\{2,3\}} X_2^{\{1,4\}} X_3^{\{0,1,3\}} + \\ & X_1^{\{4\}} X_2^{\{0,1,2,3,4\}} X_3^{\{0,2,3\}} \\ P_1 &= P_2 = P_3 = \{0,1,2,3,4\}\end{aligned}$$

- $\nu_1 = \nu_2 = \{0\}, \nu_3 = \{4\}$
- F is weakly-unate in every variable

	X_1	X_2	X_3
c_1	11111	00001	11110
c_2	01100	00011	01010
c_3	01010	00100	11111
c_4	00110	01001	11010
c_5	00001	11111	10110

Application to Tautology

Theorem 1 *Let* $\{c^1, \ldots, c^t\}$ *be a cube partition as in Shannon expansion theorem. Then*

$$f = 1 \text{ iff } f_{c^k} = 1 \text{ for every } k \tag{1}$$

Proof follows two-valued case exactly

$$f = c^1 f_{c^1} + c^2 f_{c^2} + \ldots + c^t f_{c^t}$$

$$c^i \cap c^j = \phi, i \neq j$$

$$\sum c^i \equiv 1$$

Monotone Theorem

Theorem 2 *Let f be weakly unate in variable x_i from value j. Then*

$$f_{X_i^{\{j\}}} \subseteq f_{X_i^{\{k\}}} \text{ forall } k \in P_i \tag{2}$$

Analogous to $f_{\overline{x}} \subseteq f_x$ for monotone increasing (from 0).

Proof: We first show that $f_{x_i^{\{k\}}}$ is independent of x_i. Let $\{c^l\}$ be any cover for f. Then $\{c^l_{x_i^{\{k\}}}\}$ is a cover for $f_{x_i^{\{k\}}}$. Each cube $c^l_{x_i^{\{k\}}}$ is either ϕ or has all 1's in the x_i part. (*When cofactoring by cube $x_i^{\{k\}}$, $c^l_{ik} = 0 \Longrightarrow \phi$, $c^l_{ik} = 1 \Longrightarrow$ all 1's, because we add 1's to all values $P_i - \{k\}$*).

Let v be an arbitrary point in the space excluding x_i. Let $u = x_i^{\{j\}}v$, $w = x_i^{\{k\}}v$. Since f is weakly unate from j, $f(u) \le f(w)$. By Shannon expansion

$$f = \sum_{l=0}^{p_i-1} x_i^{\{l\}} f_{x_i^{\{l\}}}$$

So

$$f(u) = f_{x_i^{\{j\}}}(v) \le f(w) = f_{x_i^{\{k\}}}(v)$$

(*since* $f_{x_i^{\{l\}}}$ *is independent of* x_i). Since v is an arbitrary point, then

$$f_{x_i^{\{j\}}} \subseteq f_{x_i^{\{k\}}}$$

Weakly Unate Reduction Theorem

Theorem 3 *(unate reduction) f is weakly-unate in X_i, and the "unate value" is j. Then $f = 1$ iff $f_{X_i^{\{j\}}} = 1$*

Proof.
($\Longrightarrow$) Write

$$f = \sum_{k=0}^{p_i - 1} X_i^{\{k\}} f_{X_i^{\{k\}}}.$$

From (1), $f = 1$ iff $f_{X_i^{\{k\}}} = 1$ for every k, and hence $f = 1$ implies $f_{X_i^{\{j\}}} = 1$

($\Longleftarrow$) If $f_{X_i^{\{j\}}} = 1$, then by (2) $f_{X_i^{\{k\}}} \supseteq f_{X_i^{\{j\}}} = 1$ and hence $f_{X_i^{\{k\}}} = 1$, and so $f = 1$ by (1) $\square$

Note this is *exactly* two-valued theorem:

$$A \neq 1 \Longrightarrow \left(\begin{pmatrix} A & X \\ T & B \end{pmatrix} \equiv 1 \Longleftrightarrow B \equiv 1 \right)$$

except that here we are doing one unate variable at a time.

Also note that

$$F = \ldots \begin{array}{c} X_i \\ 01101 \\ 01010 \\ 00110 \\ 11111 \\ 11111 \end{array} \ldots \qquad F_{X_i^{\{0\}}} = \ldots \begin{array}{c} X_i \\ 11111 \\ 11111 \end{array} \ldots$$

because $X_i^{\{0\}}$ does not intersect with the first 3 cubes.

Tautology for Weakly Unate Cover

Definition 3 *Cover* $c^1 + \ldots + c^t$ *is weakly-unate iff it is weakly-unate in all variables*

Theorem 4 $c^1 + \ldots + c^t$ *weakly-unate. Then* $c^1 + \ldots + c^t = 1$ *iff* $c^j \equiv 1$ *for some cube* j*. (*$c^j \equiv 1$ means $c^j = X_1^{P_1} \ldots X_n^{P_n}$*)*

Proof. Follows from reduction theorem □

Thus for weakly unate cover, can tell immediately.

$$\begin{array}{lccccc} & 0101 & 1011 & 0011 & & \\ & 0011 & 1010 & 1011 & = & 1 \\ \rightarrow & 1111 & 1111 & 1111 & & \end{array}$$

$$\begin{array}{ccccc} 0101 & 1011 & 0011 & & \\ 0011 & 1010 & 1011 & \neq & 1 \\ 0101 & 1011 & 1001 & & \end{array}$$

vertex 1000 0100 0100 not covered.

Reduction in One Step

Form cube $c = X_1^{\mu_1} \dots . X_n^{\mu_n}$ where

$$\mu_i = \begin{cases} P_i & \nu_i = \emptyset \text{ (\textit{i.e. if} } X_i \text{ \textit{is "binate"})} \\ \nu_i & \text{ (\textit{unate part of variable} } X_i \text{)} \end{cases}$$

$f = 1$ iff $f_c = 1$
(*Unate reduction. Gets rid of cubes with 0 in* ν_i *part*)

c *exactly* as in two-valued algorithm

$$f = \begin{pmatrix} A & X \\ T & B \end{pmatrix}$$

c is cube of unate variables, e.g.

$$A = \begin{array}{c} 1201 \\ 2022 \\ 1021 \end{array} \Longrightarrow c = 0110$$

then $A_c = 0$. Hence

$$f_c = (T \ \ B)$$

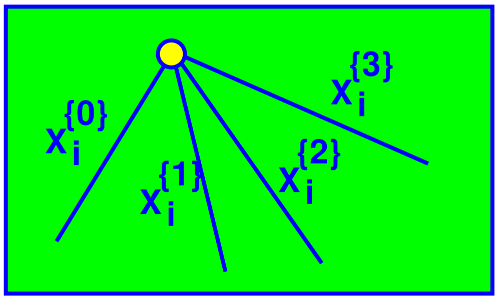
Revised Tautology

Tautology(F)
   while(F is weakly unate in some variables)
      $c = X_1^{\mu_1} .... X_n^{\mu_n}$
      $F \leftarrow F_c$
      if $F \equiv 1$ return true;
      if $F \equiv 0$ return false;
   Choose $c^1, ..., c^t$ (*some cube partition of space*)
   for($i = 1, \ldots, t$)
      if (Tautology(F_{c^i}) = false)
         return false;
   return true;

Left open: how to split?
(*i.e. how to choose* $c^1, c^2, \ldots, c^t$ *where* $c^i \cap c^j = \phi$, *and* $\sum c^i = 1$.)

Methods of Splitting

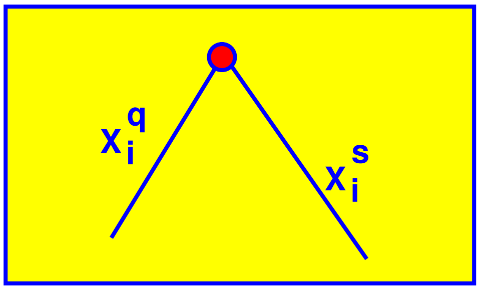
- **"Split by value"**



$$f = \sum_{k=0}^{p_i - 1} X_i^{\{k\}} f_{X_i^{\{k\}}}$$

  - Gets rid of variable X_i in a single step

- **"Split by parts"** q, s partition P_i (*e.g.* $q = \{0,1\}, s = \{2,3\}$)



$$f = X_i^q f_{X_i^q} + X_i^s f_{X_i^s}$$

  - May get to unate leaves (somewhat) more quickly

- More freedom to choose good partitions – don't need to entirely eliminate variable X_i before splitting on X_k.

- In practice, **"split by parts"** is used

Choice of Splitting Variable, i.e. which X_i

Cover $F = \gamma^1 + ... + \gamma^{|F|}$

- Goal: get to weakly unate leaves as fast as possible
- **Definition 4 Active** *value of variable* X_i*:*
$$\{j \mid \exists k \text{ such that } j \notin \gamma_i^k\}$$
*(*Any value with all 1's in column is **not** active *)*
- Choose variable with most active values
(*choose binate ("multi-ate") variable*)
(Note: all **inactive** values can be equivalently grouped into **one** value.)
- Tiebreaks:
  - Variables i maximizing
$$\sum_{j=1}^{|F|} |P_i - \gamma_i^j| = \sum_{j=1}^{|F|} |\overline{\gamma_i^j}|$$
("Smallest" variable = most 0's in column)
  - Variables minimizing
$$\sum_{j=1}^{|F|} (P_i = \gamma_i^j)$$
(*least "2's"*)

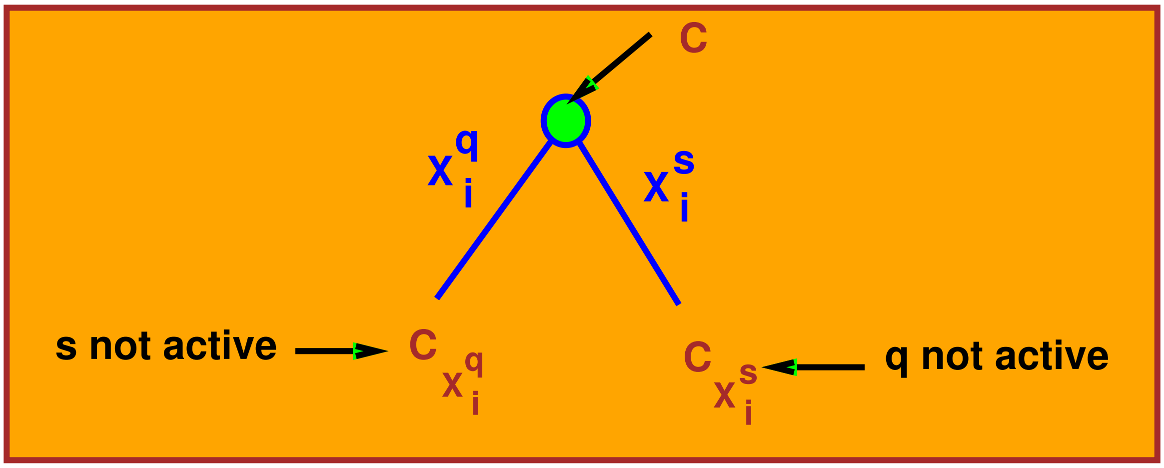
Choice of Partition (s, q)

Cover $C = c^1 + ... + c^t$, variable X_i

- Goal: Like to find **partition** q, s of P_i such that:

$$|C_{X_i^q}| + |C_{X_i^s}|$$

  is minimized



- Hard problem! Use heuristic
- "Fast to compute" more important than quality...

Heuristic:

- m **active** values in X_i
- q gets first $m/2$ active values, s the rest

This reduces the number of active values on each side by half.

Strongly Unate Functions

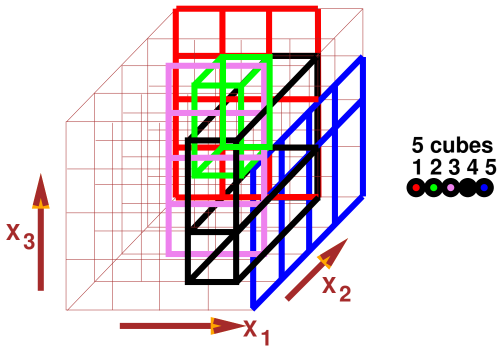
Weakly-unate good enough for tautology based algorithms, but...

- F weakly-unate $\not\Rightarrow F_c$ weakly-unate

Example: F is weakly unate cover.

$$
\begin{array}{ccllll}
 & & X_1 & X_2 & X_3 & X_4 \\
 & & 10 & 11 & 11 & 111 \\
F & = & 11 & 10 & 10 & 100 \\
 & & 11 & 11 & 10 & 010 \\
 & & & & & \\
c & = & 11 & 11 & 11 & 110 \\
 & & & & & \\
 & & 10 & 11 & 11 & 111 \\
F_c & = & 11 & 10 & 10 & 101 \\
 & & 11 & 11 & 10 & 011
\end{array}
$$

F_c is not weakly unate in X_4. (*But* f_c *is!*).
(*However, I think this also holds for* f *and* f_c *as well*).



- f weakly-unate $\not\Rightarrow$ every prime of f essential. Example: $f = \{p_1, p_2, p_3, p_4, p_5\}$

p_1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0
p_2	0	1	1	0	0	0	0	0	1	1	0	1	0	1	0
p_3	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1
p_4	0	0	1	1	0	0	1	0	0	1	1	1	0	1	0
p_5	0	0	0	0	1	1	1	1	1	1	1	0	1	1	0

$p_1, \ldots, p_5$ are all primes.

  - p_1 essential (red)
  - p_2 nonessential (green)
  - p_3 essential (pink)
  - p_4 nonessential (black)
  - p_5 essential (blue)

a column of all 1's indicates a value that is not active.

- f weakly-unate $\not\Rightarrow \overline{f}$ weakly-unate

$$\overline{f} = \begin{array}{ccccccccccccccc}
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1
\end{array}$$

All these are primes.

Need stronger condition:

Definition 5 *f is* **strongly-unate** *in X_i iff there is some order $<$ on P_i such that, for $j < k$ in P_i,*

$$f_{X_i^{\{j\}}} \subseteq f_{X_i^{\{k\}}}$$

Thus "increasing" X_i (*from value* $j \rightarrow$ *value* k, $j < k$) doesn't decrease f

Example: strongly unate **cover** (*order is from left to right i.e.* $1 < 2 < 3$ *on all variables*)

$$f = \begin{array}{cccccccccc}
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1
\end{array}$$

Can detect strongly unate cover by the existence of a value order for each P_i, where the 0's are on the left for all cubes.

Strongly-Unate Functions: Propositions

- f strongly-unate $\Rightarrow f$ weakly-unate
- f strongly-unate $\Rightarrow \overline{f}$ strongly-unate
- f strongly-unate $\Rightarrow f_c$ strongly-unate
- f strongly-unate $\Rightarrow$ every prime of f essential...

But:

- Weakly-unate applies more often
  - easier to compute
  - good enough for tautology based algorithms

Strongly unate not used in two-level logic minimization algorithms (so far).

Exact Minimization

1. Use basic logic synthesis algorithm
2. Generate all primes
3. Form covering table
4. Solve covering table

Steps (3)-(4) are same as in binary case

Prime Generation

Theorem 5 *(prime merging) Let f be any function, l, r be any cubes such that $l \cap r = \emptyset$ and $l + r = 1$. Then the primes of f are the maximal cubes among:*

- *the primes of $lf_l = l\cap$ primes of f_l, and*
- *the primes of $rf_r = r\cap$ primes of f_r, and*
- *the cubes of $c^l \odot c^r$, where c^l is a prime of lf_l and c^r is a prime of rf^r.*

The operation $\odot$ stands for consensus:

$$c \odot d = \sum_{i=1}^{n} (c_1 \cap d_1 \times \ldots \times c_i \cup d_i \times \ldots \times c_n \cap d_n)$$

provided $\delta(c, d) \leq 1$.

Note: this easily specializes to binary theorem when all values are binary.

How do we get all the primes at a leaf? (see Rudell paper in reader, page 738)

Heuristic Minimization

- Use Espresso-II
- Irredundant, Reduce, Lastgasp
(*unchanged* because tautology based)
- Essential Primes *essentially* unchanged (minor technical differences in expression)
- Expand

Minor difference from Espresso-II

Can't use blocking matrix B^c when expanding cube c.

Can expand c_i even if column i is in a minimal column cover of B^c

Consider the case $j \notin c_i$, $j \notin r_i^k$ for each cube k of the offset

j can be added to c_i even when i is in a minimal column cover of B^c

Adding j to c_i leaves $c_i \cap r_i^k = \phi$ unchanged.

Example:

$$
\begin{array}{rcccc}
 & & \ldots..X_i..\ldots & & \\
c & = & \ldots 11000 \ldots & \Longrightarrow & B_{ki}^c = 1 \\
r^k & = & \ldots 00011 \ldots & & \\
 & & \ldots..j..\ldots & &
\end{array}
$$

This implies that $B_{ki}^c = 1$ because $c_i \cap r_i^k = \emptyset$. But we still may be able to change to $\widetilde{c} = \ldots 11100 \ldots$.

Multi-value EXPAND

We build the blocking function $g(y)$ using the cover of the offset $R = \{r^1, r^2, \ldots, r^{|R|}\}$. Suppose we are expanding cube c. Let variable y_j^k denote that in positional notation, the expanded cube $\widetilde{c}$ has a 1 in value k of variable x_j. Then $\widetilde{c}$ intersects r^i if

$$\bigcap_{j=1}^{n} \bigcup_{k=0}^{p_j-1} (r^i)_j^k y_j^k = 1$$

Hence $\widetilde{c}$ intersects the offset is given by the function

$$\overline{g}(y) = \bigcup_{i=1}^{|R|} \bigcap_{j=1}^{n} \bigcup_{k=0}^{p_j-1} (r^i)_j^k y_j^k$$

Its complement $g(y)$ gives all legitimate expansions of c. A prime of g gives a prime expansion of c. Note that g is a binary function, and is unate. It is convenient to obtain $g(y)$ in SOP form by complementing

$$\bigcap_{j=1}^{n} \bigcup_{k=0}^{p_j-1} (r^i)_j^k y_j^k$$

using our unate complementer.

Conclusions

- Adding Multi-valued permits multiple-output functions, solving encoding problems, and other applications which are naturally multi-valued.
- *Minimization fundamentals unchanged*
- *Details* changed in
  - Consensus
  - All-prime generation (minor)
  - Unateness (2 kinds now – largest change)
  - Essential primes (to accomodate consensus)
  - Expand (blocking matrix no longer an efficiency gain)
- Details in paper by Rudell and Sangiovanni (in reader).