INTRODUZIONE
ALLA SINTESI AUTOMATICA DI CIRCUITI LOGICI

Docente: Tiziano Villa

Introduction

Motivation

- commercial success - used almost everywhere VLSI is done
- but more general - generally - discrete functions of discrete valued variables
- body of useful and general techniques - can be applied to other areas
- foundation for:
  combinational and sequential synthesis
  testing
  timing and false paths
  formal verification
  asynchronous synthesis
  automata theory
  optimal clocking schemes
  hazard analysis
  power estimation
  general combinatorics.

Outline of Class

1. Introduction
   logic functions and their representation
   unate recursive paradigm, Quine McCluskey

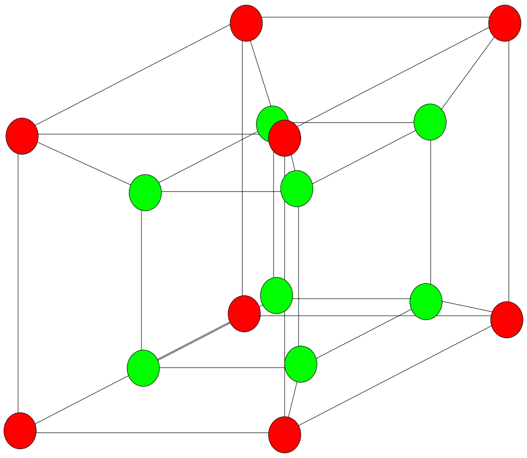
2. Two Level Minimization
   ESPRESSO two level minimization
   multivalued minimization
   BDD's

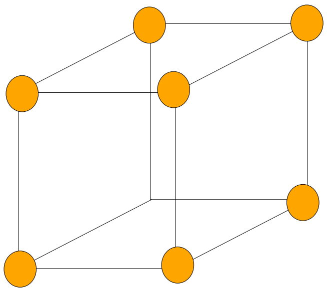
3. Multi-Level Logic Synthesis
   introduction (Boolean networks, factored forms)
   division
   simplification
   full simplify
   testing
   technology mapping
   advanced technology mapping
   delay analysis
   timing optimization

4. Sequential Logic Synthesis
   introduction (FSM networks)
   node minimization
   state minimization
   retiming and resynthesis

The Boolean n-cube

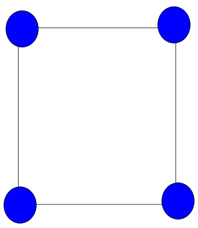
$\mathbf{B}^{\mathbf{n}}$





$\mathbf{B}^{\mathbf{3}}$

$\mathbf{B}^{\mathbf{4}}$



$\mathbf{B}^{\mathbf{0}}$ $\mathbf{B}^{\mathbf{1}}$ $\mathbf{B}^{\mathbf{2}}$

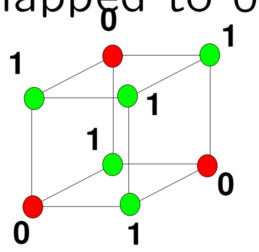
- $\mathcal{B} = \{0, 1\}$

- $\mathcal{B}^2 = \{0, 1\} \times \{0, 1\} = \{00, 01, 10, 11\}$

Boolean Functions

$f(x) : \mathcal{B}^n \to \mathcal{B}$
$\mathcal{B} = \{0, 1\}$, $x = (x_1, x_2, \ldots, x_n)$

- each **vertex** of $\mathcal{B}^n$ is mapped to 0 or 1.



- the **onset** of f is $\{x \mid f(x) = 1\} = f^1 = f^{-1}(1)$
- the **offset** of f is $\{x \mid f(x) = 0\} = f^0 = f^{-1}(0)$
- if $f^1 = \mathcal{B}^n$, f is the **tautology**, i.e. $f \equiv 1$
- if $f^0 = \mathcal{B}^n$, $(f^1 = \phi)$ f is **not satisfiable**
- if $f(x) = g(x)$ for all $x \in \mathcal{B}^n$, then f and g are **equivalent**
- $x_1, x_2, \ldots$ are **variables**
- $x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots$ are **literals**

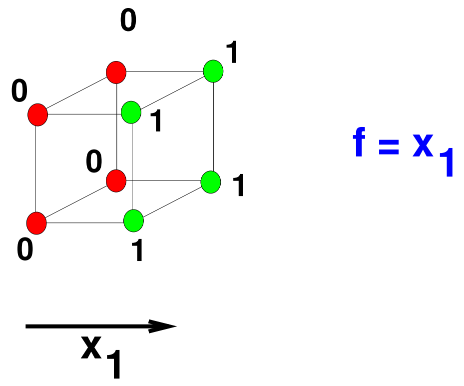
Literals

A literal is a variable or its negation

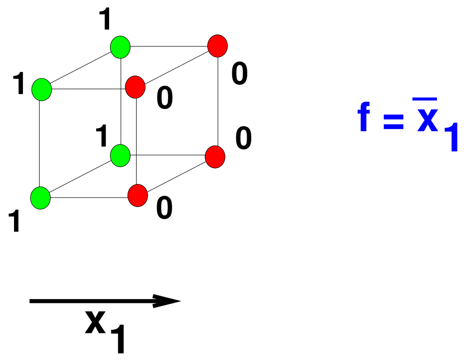
$$y, \overline{y}$$

It represents a **logic function**

Literal x_1 represents the logic function f, where
$f^1 = \{x \,|\, x_1 = 1\}$



Literal $\overline{x}_1$ represents logic function g where
$g^1 = \{x \,|\, x_1 = 0\}$



Boolean Formulas

Boolean functions can be **represented** by **formulas** defined as catenations of

- parentheses - (,)
- literals - $x, y, z, \overline{x}, \overline{y}, \overline{z}$
- Boolean operators - $+$ (OR), $\times$ (AND)
- complementation - e.g. $\overline{x+y}$

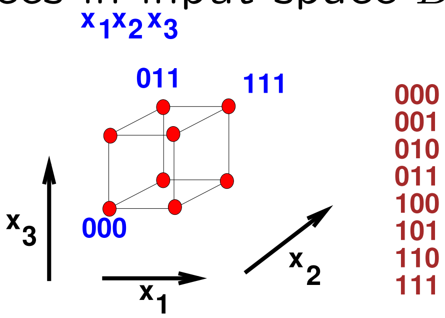
Examples:

$$\begin{aligned} f &= x_1 \times \overline{x}_2 + \overline{x}_1 \times x_2 \\ &= (x_1 + x_2) \times (\overline{x}_1 + \overline{x}_2) \\ h &= a + b \times c \\ &= \overline{\overline{a} \times (\overline{b} + \overline{c})} \end{aligned}$$

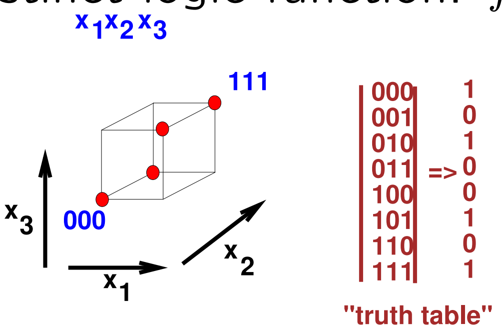
We will usually replace $\times$ by catenation, e.g. $a \times b \to ab$.

Logic Functions - $f(x) : B^n \rightarrow B$

- There are 2^n vertices in input space B^n



- There are 2^{2^n} distinct logic **functions**. Each subset of vertices is a distinct logic function: $f^1 \subseteq B^n$



- There are ∞ number of logic **formulas**

$$\begin{aligned} f &= x + y \\ &= x\overline{y} + xy + \overline{x}y \\ &= x\overline{x} + x\overline{y} + y \\ &= (x + y)(x + \overline{y}) + \overline{x}y \end{aligned}$$

SYNTHESIS = Find the "best" formula (or "representation")

Boolean Operations - AND, OR, COMPLEMENT

$$\begin{aligned} f &: B^n \rightarrow B \\ g &: B^n \rightarrow B \end{aligned}$$

- AND - $f \times g = h$ such that
  $h^1 = \{x \,|\, f(x) = 1 \text{ and } g(x) = 1\}$

- OR - $f + g = h$ such that
  $h^1 = \{x \,|\, f(x) = 1 \text{ or } g(x) = 1\}$

- COMPLEMENT - $\overline{f} = h$ such that
  $h^1 = \{x \,|\, f(x) = 0\}$

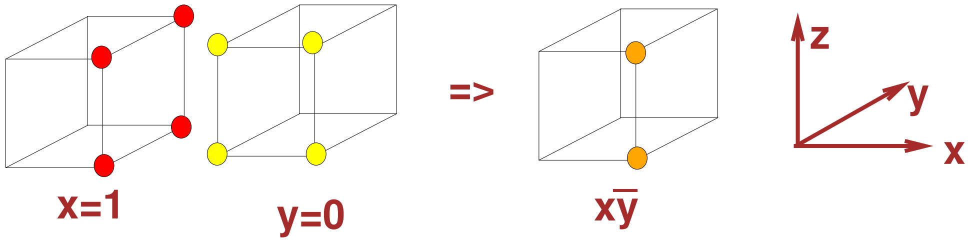
Cubes

- The AND of a set of literal functions ("conjunction" of literals) is a **cube**

$$C = x\overline{y}$$

is a cube

$$C = (x = 1)(y = 0)$$



- if $C^1 \subseteq f^1$, C a cube, then C is an **implicant** of f

- if $C^1 \subseteq B^n$ and C has k literals, then $|C^1|$ has 2^{n-k} vertices

Example 1 $C = x\overline{y} \subseteq B^3$
$k = 2$
$n = 3$

$$C^1 = \{100, 101\}$$

$|C^1| = 2 = 2^{3-2}$

- if $k = n$, the cube is a **min-term**

Representation of Boolean Functions

The **truth table** of a function $f : B^n \to B$ is a tabulation of its value at each of the 2^n vertices of B^n.

For

$$f = \overline{a}\overline{b}\overline{c}d + \overline{a}\overline{b}cd + \overline{a}b\overline{c}d + a\overline{b}\overline{c}d + a\overline{b}cd + ab\overline{c}d + abc\overline{d} + abcd$$

the truth table is

	abcd	f		abcd	f
0	0000	0	8	1000	0
1	0001	1	9	1001	1
2	0010	0	10	1010	0
3	0011	1	11	1011	1
4	0100	0	12	1100	0
5	0101	1	13	1101	1
6	0110	0	14	1110	1
7	0111	0	15	1111	1

This is intractable for large n (but canonical).

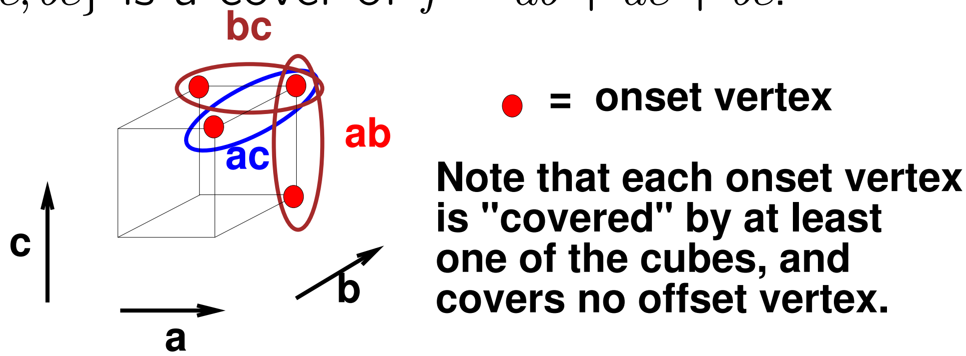
Canonical means that if two functions are the same, then the canonical representations of each are isomorphic.

Sum-of-Products Representation - SOP

- A function can be represented by a sum of cubes (products):
$$f = ab + ac + bc$$
Since each cube is a product of literals, this is a "sum-of-products' representation

- A SOP can be thought of as a set of cubes
$$f = \{ab, ac, bc\} = \mathcal{C}$$

- A set of cubes that represents f is called a **cover** of f. $\{ab, ac, bc\}$ is a cover of $f = ab + ac + bc$.



- Covers (SOP's) can efficiently represent many logic functions (i.e. for many, there exist small covers).

- Two-level minimization seeks the minimum size cover (least # of cubes).

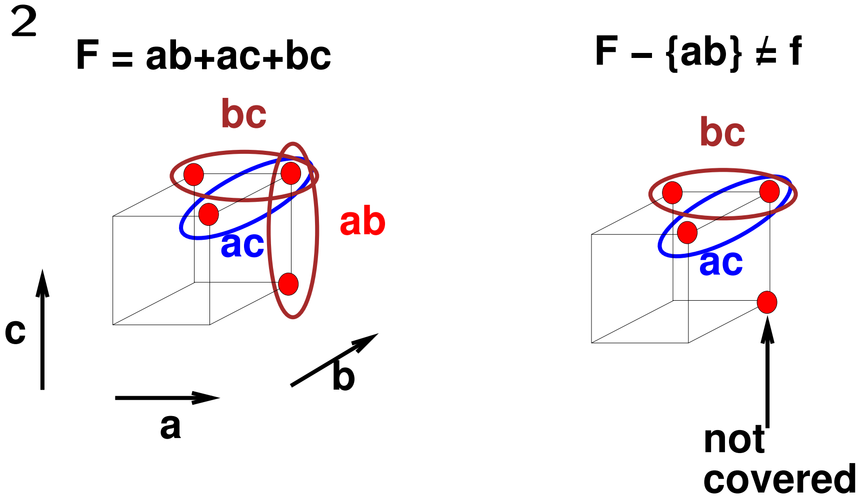
Irredundant

Let $F = \{c_1, c_2, \ldots, c_k\}$ be a cover for f

$$f = \sum_{i=1}^{k} c_i$$

A cube $c_i \in F$ is **irredundant** if $F - \{c_i\} \neq f$

Example 2



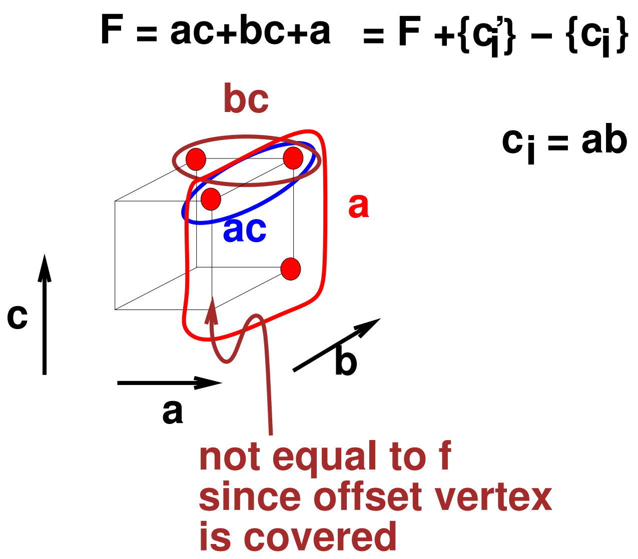
Prime

An implicant (cube) of f is **prime** if

$$f + c_i' \neq f$$

where c_i' is c_i with any literal of c_i deleted.

Example 3
$f = ab + ac + bc$
$c_i = ab$
$c_i' = a$ *(literal b deleted)*
$F \cup \{c_i'\} - \{c_i\} = a + ac + bc$

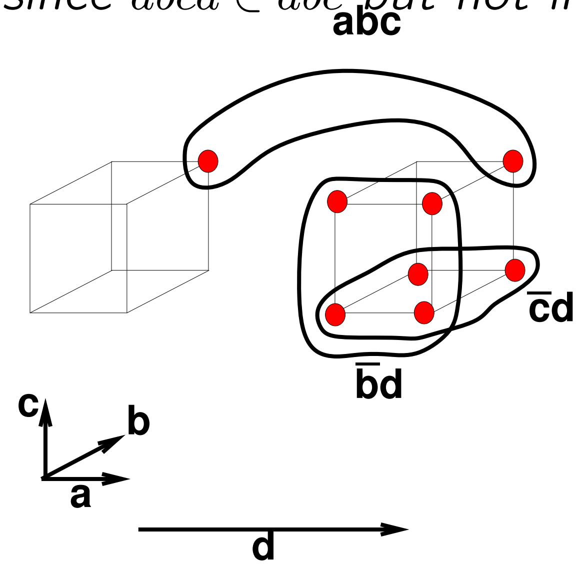


Prime and Irredundant Covers

Definition 1 *A* **cover** *is prime* **irredundant** *if all its cubes are prime (irredundant).*

Definition 2 *A prime of f is* **essential** *(essential prime) if there is a min-term (essential vertex) in that prime but in no other prime.*

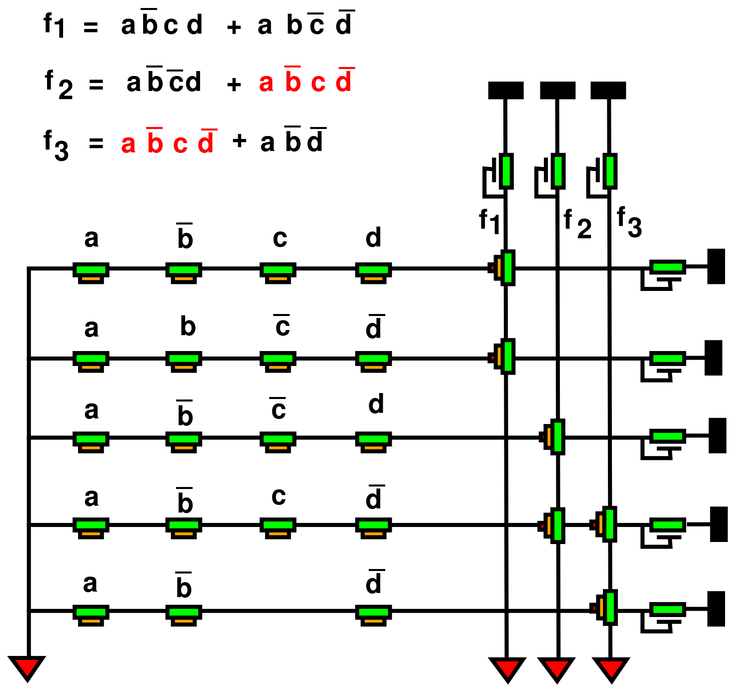
Example 4 $F = abc + \bar{b}d + \bar{c}d$
is prime and irredundant
abc *is essential since* $abc\bar{d} \in abc$ *but not in* $\bar{b}d$ *or* $\bar{c}d$ *or* ad



Why is $abcd$ not an essential point of abc?
What is an essential vertex of abc?
What other cube is essential?
What cube is not essential?

PLA's - Multiple Output Boolean Functions

A PLA is a function $f : B^n \rightarrow B^m$ represented in SOP form:



- Each distinct cube appears just once in the **AND-plane**, and can be shared by (multiple) outputs in the **OR-plane**, e.g. cube $a\bar{b}c\bar{d}$.
- Extensions of single output minimization theory are straightforward.
- Multilevel logic can be viewed mathematically as a connection of single output functions.

Shannon (Boole) Cofactors

Let $f : B^n \rightarrow B$ be a Boolean function, and $x = (x_1, \ldots, x_n)$ the variables in the support of f. The **cofactor** f_a of f by a literal $a = x_i$ or $a = \overline{x}_i$ is

$$
\begin{aligned}
f_{x_i}(x_1, \ldots, x_n) &= f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) \\
f_{\overline{x}_i}(x_1, \ldots, x_n) &= f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)
\end{aligned}
$$

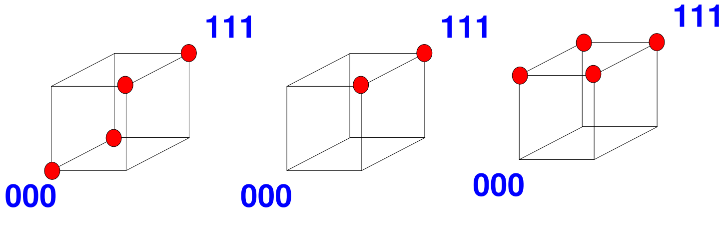
Notes:

- The cofactor f_C of f by a cube C is just f with the fixed values indicated by the literals of C, e.g. if $C = x_i \overline{x}_j$, then $x_i = 1$ and $x_j = 0$.

- If $C = x_1 \overline{x}_4 x_6$, f_C is just the function f restricted to the subspace where $x_1 = x_6 = 1$ and $x_4 = 0$.

- As a function, f_C does not depend on x_1, x_4, or x_6. (*However, we still consider* f_C *as a function of all* n *variables, it just happens to be independent of* x_1, x_4, x_6)

- $x_1 f \neq f_{x_1}$

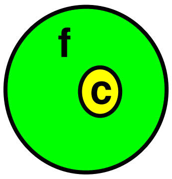


$f = ac + \overline{a}\overline{c}$ $af = ac$ $f_a = c$

Fundamental Theorem

Theorem 1 *Let c be a cube and f a function. Then $c \subseteq f \iff f_c \equiv 1$.*

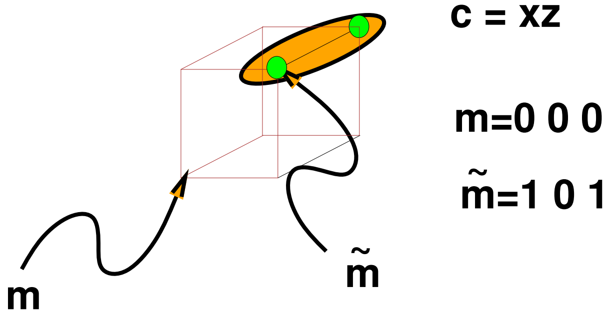
Proof. We use: $xf_x = xf$ and f_x is independent of x.
if: Suppose $f_c \equiv 1$. Then $f_c c = c$ and $c = cf$. Thus $c \subseteq f$.



only if: Assume $c \subseteq f$. Then $c \subseteq fc = f_c c$. But f_c is independent of literals $l \in c$. If $f_c \not\equiv 1$, then $\exists m \in B^n$, $f_c(m) = 0$. Let
$\widetilde{m}_i = m_i$, if $x_i \notin c$ and $\overline{x}_i \notin c$
$\widetilde{m}_i = m_i$, if $m_i = 0$, $\overline{x}_i \in c$ or $m_i = 1, x_i \in c$
$\widetilde{m}_i = \overline{m_i}$, otherwise
i.e. we make the literals of $\widetilde{m}$ agree with c. But then

$$f_c(\widetilde{m}) = f_c(m) = 0$$

and $\widetilde{m} \in c$, Hence $c(\widetilde{m}) = 1$ and $f_c(\widetilde{m})c(\widetilde{m}) = 0$ contradicting $c \subseteq f_c c$.



Cofactor of Covers

Definition 3 *The* **cofactor of a cover** F *is the sum of the cofactors of each of the cubes of* F*.*

Note: If $F = \{c_1, c_2, \ldots, c_k\}$ is a cover of f, then $F_c = \{(c_1)_c, (c_2)_c, \ldots, (c_k)_c\}$ is a cover of f_c.

Suppose $F(x) : B^n \to B$ is a cover, i.e.

$$F(x) = \sum_i C_i = \sum_i \prod_j l_j^i = \{C_i\}$$

Then for $1 \leq j \leq n$,

$$F(x)_{x_j} = \sum_i (C_i)_{x_j}$$

Definition 4 *The* **cofactor** C_{x_j} *of a cube* C *with respect to a literal* x_j *is*

- C *if* x_j *does not appear in* C
- $C - \{x_j\}$ *if* x_j *appears positively in* C*, i.e.* $x_j \in C$
- ϕ *if* x_j *appears negatively in* C*, i.e.* $\overline{x}_j \in C$

Examples

Example 5 *If* $C = x_1\overline{x}_4x_6$,

- $C_{x_2} = C$ *(x_2 does not appear in C)*
- $C_{x_1} = \overline{x}_4x_6$ *(x_1 appears positively in C)*
- $C_{x_4} = \phi$ *(x_4 appears negatively in C)*

Example 6

$$\begin{aligned} F &= abc + \overline{b}d + \overline{c}d \\ F_b &= ac + \overline{c}d \end{aligned}$$

(*Just drop b everywhere and throw away cubes containing literal $\overline{b}$*).

Shannon Expansion

$f : B^n \rightarrow B$

Theorem 2 $f = x_i f_{x_i} + \overline{x}_i f_{\overline{x}_i}$

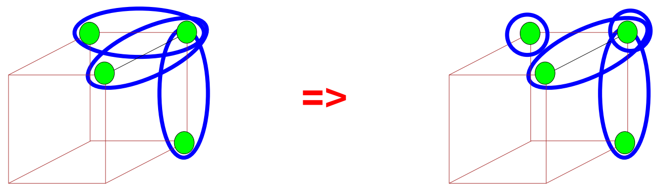
Theorem 3 *F is a cover of f. Then*

$$\widetilde{F} \equiv x_i F_{x_i} + \overline{x}_i F_{\overline{x}_i}$$

is a cover of f.

We say that f (F) is **expanded about** x_i. x_i is called the **splitting** variable.

Example 7 $F = ab + ac + bc$
$\widetilde{F} = aF_a + \overline{a}F_{\overline{a}} = a(b + c + bc) + \overline{a}(bc)$
$= ab + ac + abc + \overline{a}bc$



Cube bc got split into 2 cubes.

Cover Matrix

We sometimes use matrix notation to represent a cover:

Example 8 $F = ac + \overline{c}d =$

$$\begin{array}{r|cccc|} & a & b & c & d \\ \hline ac \longrightarrow & 1 & 2 & 1 & 2 \\ \overline{c}d \longrightarrow & 2 & 2 & 0 & 1 \\ \hline \end{array} \quad \text{or} \quad \begin{array}{|cccc|} a & b & c & d \\ \hline 1 & - & 1 & - \\ - & - & 0 & 1 \\ \hline \end{array}$$

Each row represents a cube. 1 means that the positive literal appears in the cube, 0 the negative. The 2 here represents that the variable does **not appear** *in the cube. It also represents both 0 and 1 values.*

Example 9 $a\overline{c} = 1\ 2\ 0\ 2 = \{1, \{0,1\}, 0, \{0,1\}\}$
$= \{1000, 1100, 1001, 1101\}$

2 here is sometimes called "input don't care", but this is confusing so we won't use the term.

Incompletely Specified Functions

$\mathcal{F} = (f, d, r)\ :\ B^n \rightarrow \{0, 1, *\}$
where * represents "don't care". (*Sometimes we use 2 in place of **)

- $f =$ **onset** function - $f(x) = 1 \leftrightarrow \mathcal{F}(x) = 1$
- $r =$ **offset** function - $r(x) = 1 \leftrightarrow \mathcal{F}(x) = 0$
- $d =$ **don't care** function - $d(x) = 1 \leftrightarrow \mathcal{F}(x) = *$

(f, d, r) forms a **partition** of B^n, i.e.

1. $f^1 + d^1 + r^1 = B^n$
2. $f^1 d^1 = f^1 r^1 = d^1 r^1 = \phi$ - (*pairwise disjoint*)

A completely specified function g is a **cover** for $\mathcal{F} = (f, d, r)$ if

$$f \subseteq g \subseteq f + d$$

(*Thus* $gr = \phi$). Thus if $x \in d$ (i.e. $d(x) = 1$), then $g(x)$ can be 0 or 1, but if $x \in f$, then $g(x) = 1$ and if $x \in r$, then $g(x) = 0$.

(*We "don't care" which value* g *has at* $x \in d$)

Primes of Incompletely Specified Functions

Definition 5 *A cube c is* **prime** *of $\mathcal{F} = (f, d, r)$ if $c \subseteq f + d$ (*an implicant of $f + d$*), and no other implicant (of $f + d$) contains c, i.e.*

$$\forall \widetilde{c},\ \widetilde{c} \subseteq f + d,\ c \not\subseteq \widetilde{c}$$

*(*a prime of $f + d$*)*

Definition 6 *Cube c_j of cover $F = \{c_i\}$ is* **redundant** *if*

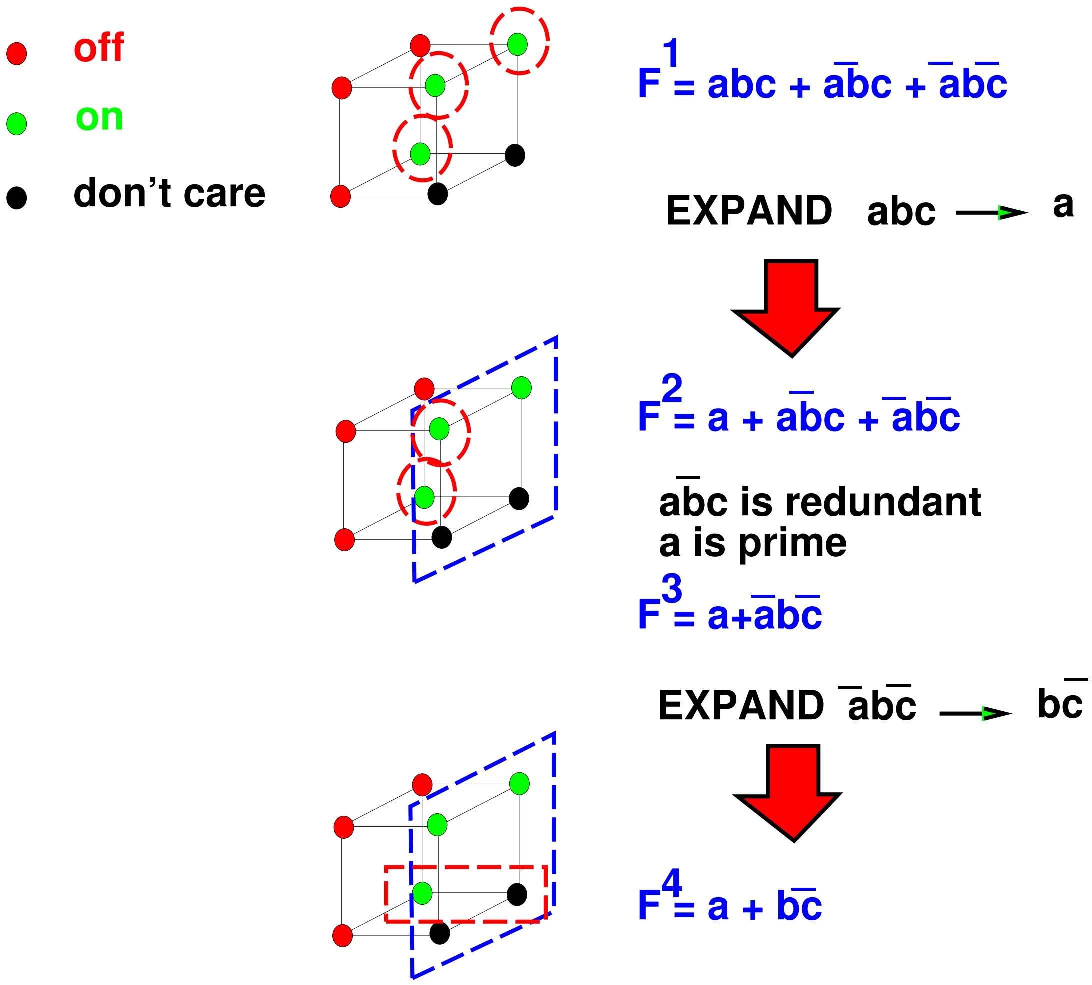
$$f \subseteq F - \{c_j\}$$

Otherwise it is **irredundant***.*

Note that $c \subseteq f + d \leftrightarrow cr = \phi$.

Example - Logic Minimization

Consider $\mathcal{F}(a,b,c) = (f,d,r)$, where $f = \{\overline{a}b\overline{c}, a\overline{b}c, abc\}$ and $d = \{a\overline{b}\overline{c}, ab\overline{c}\}$, and the sequence of covers illustrated below:



Checking for Prime and Irredundant

Let $G = \{c_i\}$ be a cover of $\mathcal{F} = (f, d, r)$. Let D be a cover for d.

1. $c_i \subseteq G$ is **redundant** iff
$$c_i \subseteq (G - \{c_i\}) \cup D \equiv G^i \qquad (1)$$
(*Since* $c_i \subseteq G^i$, *and* $f \subseteq G \subseteq f + d$ *then* $c_i = c_i f + c_i d$ *and* $c_i f \subseteq G - \{c_i\}$. *Thus* $f \subseteq G - \{c_i\}$)

2. A literal $l \in c_i$ is **prime** if $(c_i - \{l\}) = (c_i)_l$ is not an implicant of $\mathcal{F}$.
A cube c_i is a prime of $\mathcal{F}$ iff all literals $l \in c_i$ are prime.
$$\texttt{literal } l \in c_i \texttt{ is not prime } \iff (c_i)_l \subseteq f + d \qquad (2)$$

Note: Both tests (1) and (2) can be checked by **tautology**:

1. $(G^i)_{c_i} \equiv 1$ (*implies* c_i *redundant*)

2. $(F \cup D)_{(c_i)_l} \equiv 1$ (*implies* l *not prime*)

Example

$$
\begin{aligned}
F &= (\overline{a}b\overline{c}, a\overline{b}c, abc) \\
D &= (a\overline{c}) \\
R &= (\overline{a}\overline{b}, \overline{a}c)
\end{aligned}
$$

Expand $abc \rightarrow ab = (c_i)_l = (abc)_c$
Check $ab \subseteq f + d \Longleftrightarrow (f+d)_{ab} = 1$

- $(f+d)_{ab} = c + \overline{c} \equiv 1$ **OK**

Expand $ab \rightarrow a$
Check $a \subseteq f + d$

- $(f+d)_a = \overline{b}c + bc + \overline{c} \equiv 1$ **OK**

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

Check if $a\overline{b}c$ is redundant

- $a\overline{b}c \subseteq a + \overline{a}b\overline{c} + a\overline{c} = F - \{a\overline{b}c\} \cup D$
- $(a + \overline{a}b\overline{c} + a\overline{c})_{a\bar{b}c} = 1$ **OK**

Cover is now $F = a + \overline{a}b\overline{c}$ and $D = a\overline{c}$

Check if $\overline{a}b\overline{c}$ is redundant
$\overline{a}b\overline{c} \subseteq a + a\overline{c} = F - \{\overline{a}b\overline{c}\} \cup D$

- $(a + a\overline{c})_{\overline{a}b\overline{c}} = \phi$ **NOT REDUNDANT**