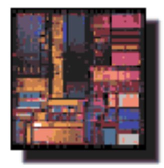
Digital Systems Design Automation

Unit 2: Advanced Boolean Algebra

Lecture 2.6: Boolean Difference and Quantifiation



Anand Raghunathan

raghunathan@purdue.edu

Outline

2.1 Boolean algebra: Quick review

2.2 Boolean spaces and functions

2.3 Boolean function representations

2.4 Conversion of Boolean function representations

2.5 Co-factors of Boolean functions

2.6 Boolean difference and Quantification

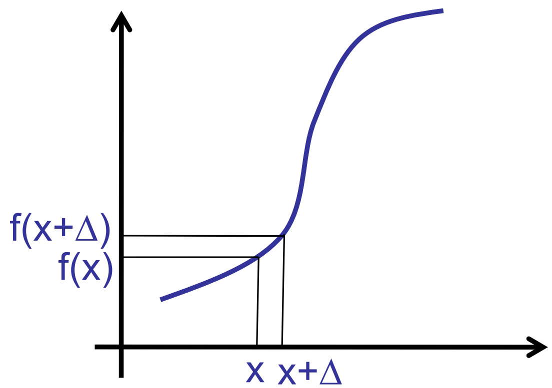
Combinations of Co-factors

- Combining f_x and $f_{x'}$ in different ways leads to useful new functions
  - $f_x \oplus f_{x'} = ?$
  - $f_x \cdot f_{x'} = ?$
  - $f_x + f_{x'} = ?$

Another analogy to the "Real" world

- The derivative of a function measures how much it changes when it's input changes

- What is the analogy in the case of Boolean functions (which only take values 0 and 1)?
  - Does a function change when it's input changes?



$$f'(x) = \lim_{\Delta \to 0} \frac{f(x+\Delta) - f(x)}{\Delta}$$

Boolean Difference

- Boolean difference of a function w.r.t. a variable is the exclusive-OR of the Shannon co-factors w.r.t. the variable

$$\frac{\partial f}{\partial x} = f_x \oplus f_{x'}$$

- Interpretation: $\frac{\partial f}{\partial x} = 1 \rightarrow f$ is sensitive to the value of x
- A new function that does not depend on x

Example:

$f = xy + w'z + x'w'$

$f_x = y + w'z$

$f_{x'} = w'z + w' = w'$

$$\frac{\partial f}{\partial x} = (y + w'z) \oplus w'$$

$= (y + w'z)' \cdot w' + (y + w'z)w$

$= y'(w + z')w' + wy$

$= w'y'z' + wy$

Boolean Difference

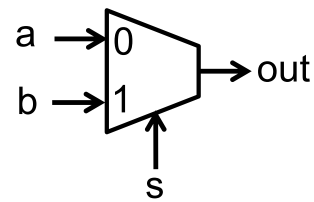
- Examples:

$S = a \oplus b \oplus c_{in}$

$c_{out} = ab + bc_{in} + ac_{in}$

$$\frac{\partial s}{\partial a} = S_a \oplus S_{a'} = (b \oplus c_{in})' \oplus (b \oplus c_{in}) = 1$$

$$\frac{\partial c_{out}}{\partial c_{in}} = (a+b) \oplus ab = (a'b')'ab + (a+b)(a'+b') = a'b + ab' = a \oplus b$$



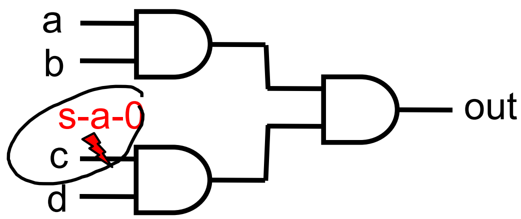
out = s'a + sb

$$\frac{\partial out}{\partial a} =$$

$$\frac{\partial out}{\partial s} = a \oplus b$$

Application of Boolean Difference

- Manufacturing test
  - Apply test vectors to ensure that each fabricated instance of an IC is functional
  - Cannot apply exhaustive test set (too big!)
- Fault model: Abstraction of physical defects that could impact the IC
  - Most commonly used: "stuck-at" fault model
  - Signals in the circuit are stuck-at-0, stuck-at-1



How do you derive a test vector to detect the fault c s-a-0?

(i) Set c = 1
(ii) Set other inputs such that output of good and faulty circuits are different

Looks familiar?

Quantification

- Two more functions of Shannon co-factors
  - $f_{x_i} \cdot f_{x_i'} = 1$ specifies when f = 1 independent of the value of x_i

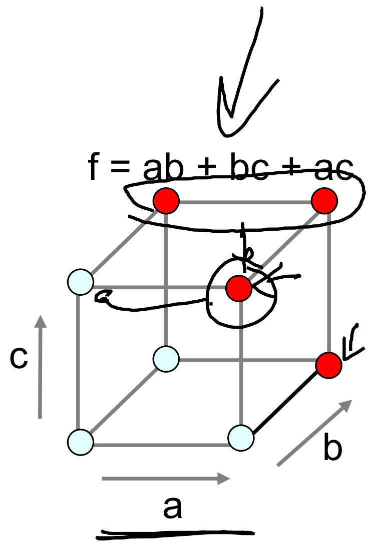
    $f(x_1 \ldots x_{i-1}, \mathbf{x_i = 1}, x_{i+1} \ldots x_n) = 1$ AND

    $f(x_1 \ldots x_{i-1}, \mathbf{x_i = 0}, x_{i+1} \ldots x_n) = 1$
  - Called **Universal quantification** or **Consensus**

$$\forall x(f) = f_x \cdot f_{x'}$$

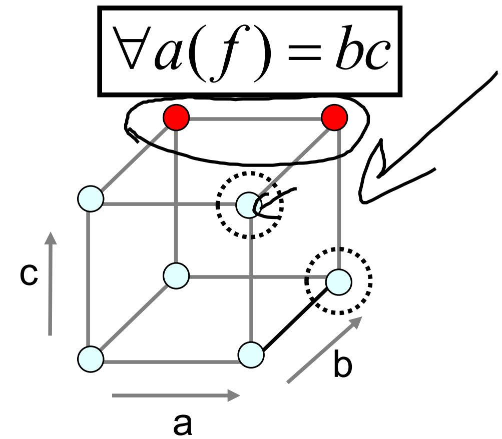
$C_x(f)$

Universal Quantification / Consensus: Geometric Interpretation



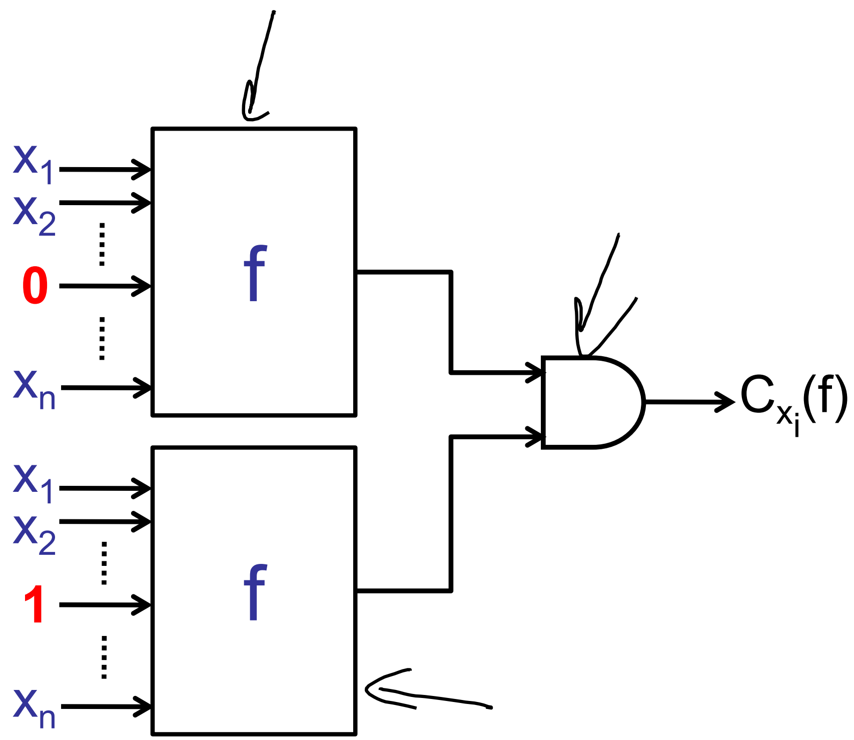
$f_a = b + c$

$f_{a'} = bc$



Keep vertices where f = 1 independent of a in the on-set of consensus function

Universal Quantification / Consensus: Circuit interpretation



Quantification

- Two more functions of Shannon co-factors
  - $f_{x_i} \cdot f_{x_i'} = 1$ specifies when f = 1 independent of the value of x_i

    $f(x_1 \ldots x_{i-1}, \mathbf{x_i = 1}, x_{i+1} \ldots x_n) = 1$ AND

    $f(x_1 \ldots x_{i-1}, \mathbf{x_i = 0}, x_{i+1} \ldots x_n) = 1$
  - Called **Universal quantification** or **Consensus**
  - $f_x + f_{x'} = 1$ specifies when f = 1 for at least one value of x_i

    $f(x_1 \ldots x_{i-1}, \mathbf{x_i = 1}, x_{i+1} \ldots x_n) = 1$ OR

    $f(x_1 \ldots x_{i-1}, \mathbf{x_i = 0}, x_{i+1} \ldots x_n) = 1$
  - Called **Existential quantification** or **Smoothing**

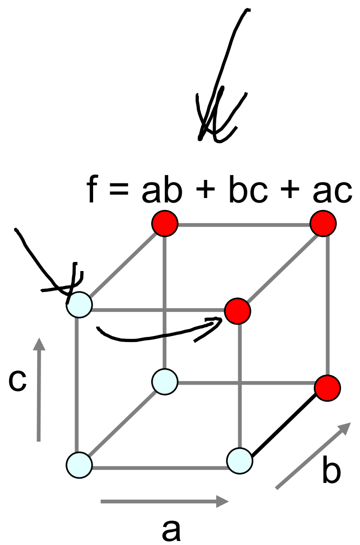
$$\forall x(f) = f_x \cdot f_{x'}$$

$C_x(f)$

$$\exists x(f) = f_x + f_{x'}$$

$S_x(f)$

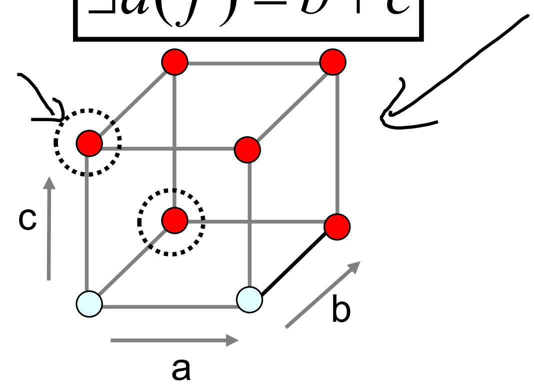
Existential Quantification / Smoothing: Geometric interpretation



$f_a = b + c$

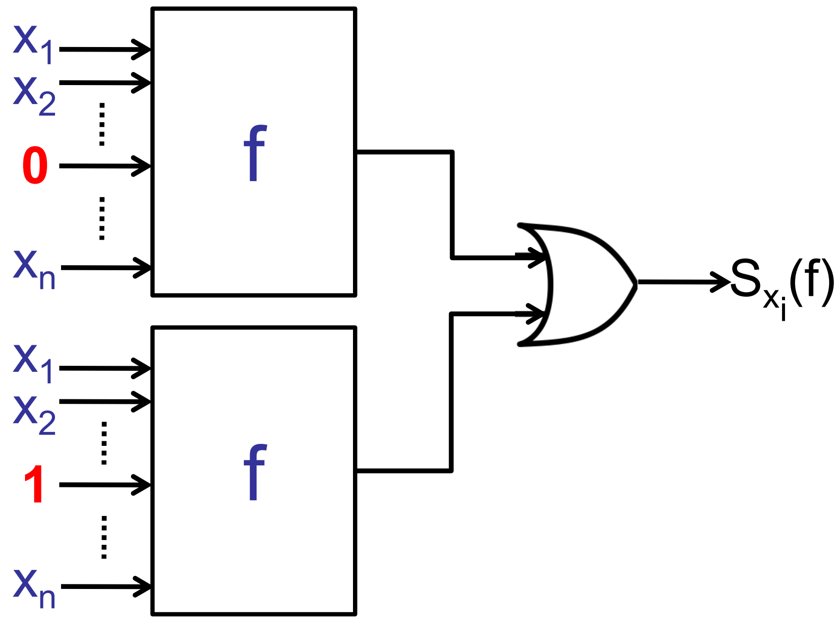
$f_{a'} = bc$

$$\exists a(f) = b + c$$



Geometric interpretation: If an off-set vertex has an on-set neighbor in the a-dimension, move it into the on-set

Existential Quantification / Smoothing: Circuit interpretation



Boolean Quantification: Examples

$S = a \oplus b \oplus c$

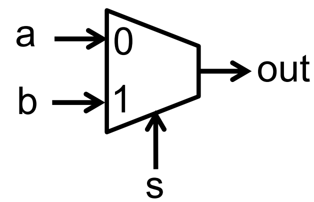
$c_{out} = ab + bc_{in} + ac_{in}$

$\forall a(s) = S_a \cdot S_{a'} = (b \oplus c)' \cdot (b \oplus c) = 0$

$\forall c_{in}(s) = 0$

$\exists a(c_{out}) = C_{out_a} + C_{out_{a'}} = b + c_{in} + bc_{in} = b + c_{in}$

$\exists c_{in}(c_{out}) =$



$out = s'a + sb$

$\forall a(out) =$

$\forall s(out) = ab$

$\exists a(out) =$

$\exists s(out) = a + b$

Properties of Boolean Quantification

- Can be applied w.r.t. multiple variables, order does not matter
  - $C_{xy}(f) = C_x(C_y(f)) = C_y(C_x(f))$
  - $S_{xy}(f) = S_x(S_y(f)) = S_y(S_x(f))$

- Containment properties
  - Consensus of a function f w.r.t. variable x is contained in f
  - Smoothing of a function f w.r.t. variable x contains f

$$\forall x(f) \subseteq f \subseteq \exists x(f)$$

$$C_x(f) \subseteq f \subseteq S_x(f)$$

Hint: For containment, think of a function in terms of it's on-set

Unit 2: Summary

- Boolean Algebra: Quick Review
- Advanced Boolean Algebra
  - Boolean spaces and functions
  - Representations of Boolean functions
  - Operations on Boolean functions
  - Co-factors and their applications
    - Shannon's expansion
    - Boolean difference
    - Existential and Universal Quantification

Reading for Unit 3: Two-level synthesis

- De Micheli, Chapter 7.1-7.4, 7.7
- Hachtel & Somenzi, Chapter 4, Chapter 5

Acknowledgments

- Prof. Sharad Malik, Princeton
- Prof. Rob Rutenbar, Carnegie Mellon University (now at University of Pittsburgh)
- Prof. Maciej Ciesielski, University of Massachusetts