Matrix representation of logic covers

- Used in logic minimizers
- Different formats
- Usually one row per implicant
- Symbols: 0,1,*…

10	11	11	10
10	01	11	11
01	10	11	11
01	11	10	01

1

CS220: Synthesis of Digital System, Fall 06

The positional cube notation

- Encoding scheme

∅	00
0	10
1	01
*	11

- Operations
  - Intersection – AND
  - Union – OR
- ∅ symbol means implicant is void and should be removed
- * means implicant is full in that variable
  - Don't care

2

CS220: Synthesis of Digital System, Fall 06

Example

- f=a'd'+a'b+ab'+ac'd

10	11	11	10
10	01	11	11
01	10	11	11
01	11	10	01

- a'd' AND a'b = 10 01 11 10 =a'bd'
- a'd' AND ab' = 00 10 11 10 = ∅
  - Can be removed
  - Not in the on-set

3

CS220: Synthesis of Digital System, Fall 06

Multi-value input functions

- Input variables can have many values
- Representations
  - Literals: set of valid values
  - Sum of products of literals
    - Let's say a, b are binary, but f can be 0,1, or 2
    - $a^{\{0\}}b^{\{0\}}f^{\{0\}}+a^{\{0\}}b^{\{1\}}f^{\{2\}}+a^{\{1\}}b^{\{0\}}f^{\{2\}}+a^{\{1\}}b^{\{1\}}f^{\{0,1\}}$
- Extension of positional cube notation

10	10	100
10	01	001
01	10	001
01	01	110

4

CS220: Synthesis of Digital System, Fall 06

Multiple output functions

- Representation by its characteristic function
  - y=f(x) can be written as $\chi(x,y)=1$
- $f_0=a'b'+ab$; $f_1=ab$; $f_2=ab'+a'b$

10	10	100
10	01	001
01	10	001
01	01	110

- Represented in as multi-value input function
  - $a^{\{0\}}b^{\{0\}}f^{\{0\}}+a^{\{0\}}b^{\{1\}}f^{\{2\}}+a^{\{1\}}b^{\{0\}}f^{\{2\}}+a^{\{1\}}b^{\{1\}}f^{\{0,1\}}$
- Input/output parts are encoded together
  - No need to talk about multiple output functions anymore
  - Simply transform it to multi-value input single Boolean output functions

5

CS220: Synthesis of Digital System, Fall 06

Notations for operations between functions

- Size of a literal is the number of 1's in its field
  - 110 size 2
- Size of implicant is the product of sizes of its literals
  - 110 11 size 4
- Intersection
  - Compute by bitwise product
  - 110 11 and 100 01 is 100 01
- Supercube of two implicants is the smallest containing cube
  - Compute by bitwise or
  - Supercube(10 10 100; 10 01 001) is 10 11 101
- Distance of two implicants is the number of empty fields in their bitwise and
  - Distance (110 11; 100 01) is 0; Distance (110 11;001 01) is 1

6

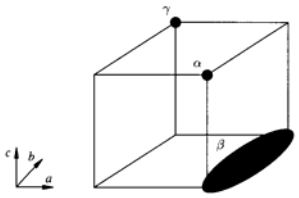
CS220: Synthesis of Digital System, Fall 06

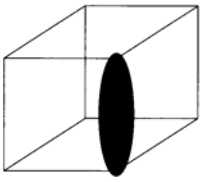
Consensus

- If two implicants has distance larger or equal to 2
  - Consensus return 0
- If two implicants have distance equal to 1
  - Consensus return a single implicant
    - Parts that is "really close"

$$CONSENSUS(\alpha, \beta) = \begin{cases} a_1 + b_1 & a_2 \cdot b_2 & \dots & a_n \cdot b_n \\ a_1 \cdot b_1 & a_2 + b_2 & \dots & a_n \cdot b_n \\ \dots & \dots & \dots & \dots \\ a_1 \cdot b_1 & a_2 \cdot b_2 & \dots & a_n + b_n \end{cases}$$

α 01 10 01
β 01 11 10
γ 10 01 01





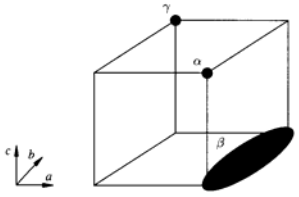
Consensus(α,β)

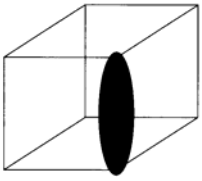
Consensus

- Consensus (ab'c,ac')=ab' = 01 10 11
  - Parts that don't differ
- Consensus (ab'c,a'b'c')=0
  - Too different

$$CONSENSUS(\alpha, \beta) = \begin{cases} a_1 + b_1 & a_2 \cdot b_2 & \dots & a_n \cdot b_n \\ a_1 \cdot b_1 & a_2 + b_2 & \dots & a_n \cdot b_n \\ \dots & \dots & \dots & \dots \\ a_1 \cdot b_1 & a_2 \cdot b_2 & \dots & a_n + b_n \end{cases}$$

α 01 10 01
β 01 11 10
γ 10 01 01





Consensus(α,β)