Permutation Matrices

A permutation matrix is an n by n matrix with a single 1 in each row and column, 0 elsewhere.

If P is a permutation (bijection) on {1,2,..,n} let A_P be the permutation matrix with $A_{iP(i)} = 1$, $A_{ij} = 0$ for $j \neq P(i)$

Permutation Matrices

Let E_{uv} be the n by n matrix with 1 in the (u,v) position and 0 elsewhere.

* Note that $E_{uv}E_{rs} = E_{us}$ if v = r, and is the n by n zero matrix otherwise.

Then $A_P = \Sigma_{i=1}{}^{n}\ E_{iP(i)}$

If $Q = P^{-1}$ then you can check that $A_Q = (A_P)^T$, the transpose of A_P.

Permutation Matrices

Note also that

$A_P{}^T A_P = \Sigma_{i=1}{}^{n}E_{P(i)i}\Sigma_{\ t=1}{}^{n}E_{tP(t)} = \Sigma_{i=1}{}^{n}\Sigma_{\ t=1}{}^{n}E_{P(i)i}E_{tP(t)} = \Sigma_{\ t=1}{}^{n}E_{P(t)P(t)} = I_n$, the n by n identity matrix, by *

Also, $E_{ik}\ A_P = E_{ik}\ \Sigma_{\ t=1}{}^{n}E_{tP(t)} = E_{iP(k).}$

Right multiplying an m by n matrix B by A_P permutes the columns of B, moving the k^{th} column to the $P(k)^{th}$ column

Permutation Matrices

Likewise $A_P{}^T\ E_{ik} = \Sigma_{\ t=1}{}^{n}E_{P(t)t}\ E_{ik} = E_{P(i)k}$ so left multiplying an n by m matrix B by $A_P{}^T$ will permute the rows, moving the i^{th} row to the $P(i)^{th}$ place.

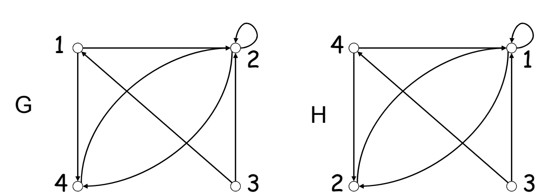
If B is n by n then $A_P{}^T\ B\ A_P$ will be B with both rows and columns permuted by P: row i → row P(i), column j → column P(j)

Isomorphisms of Graphs

Suppose G and H are graphs, each with n vertices. If G has vertices $g_1,\ldots g_n$ and H has vertices $h_1,\ldots,h_n$ then a permutation P taking g_i to $h_{P(i)}$ will give an isomorphism of graphs if $A_P{}^T\ M_G\ A_P = M_H$, where M_S is the adjacency matrix of graph S.

Example

The graphs G and H are clearly isomorphic, but can we tell that from their matrices?



Map vertices of G to those of H by P(1) = 4, P(2) = 1, P(3) = 3, P(4) = 2.

$$M_G = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad M_H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$A_P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$M_G A_P = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$A_P^T M_G A_P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} = M_H$$

Isomorphisms of Graphs

Note that if F is an isomorphism from a graph G of n vertices $v_1 \ldots v_n$ to a graph H of n vertices $w_1 \ldots w_n$ then F defines a permutation of {1,…,n} and the adjacency matrices of G and H will be related by a permutation matrix.

Isomorphisms of Graphs

But not every permutation of the vertices will produce a graph isomorphism. The permutations producing a graph isomorphism F have to map the edges appropriately because (v,u) is an edge iff (F(v), F(u)) is an edge.

Our Next Topic:

Boolean Algebra

Chapter 12

Boolean Algebra

Boolean algebra provides the operations and the rules for working with the set **{0, 1}.**

These are the rules that underlie **electronic circuits**, and the methods we will discuss are fundamental to **VLSI design**.

We are going to focus on three operations:

- Boolean complementation,
- Boolean sum, and
- Boolean product

Boolean Operations

The **complement** is denoted by a bar (on the slides, we will use a minus sign). It is defined by

-0 = 1 and -1 = 0.

The **Boolean sum**, denoted by + or by OR, has the following values:

1 + 1 = 1, 1 + 0 = 1, 0 + 1 = 1, 0 + 0 = 0

The **Boolean product**, denoted by · or by AND, has the following values:

1 · 1 = 1, 1 · 0 = 0, 0 · 1 = 0, 0 · 0 = 0

Boolean Functions and Expressions

Definition: Let B = {0, 1}. The variable x is called a **Boolean variable** if it assumes values only from B.

A function from B^n, the set $\{(x_1, x_2, \ldots, x_n) \mid x_i \in B, 1 \le i \le n\}$, to B is called a **Boolean function of degree n**.

Boolean functions can be represented using expressions made up from the variables and Boolean operations.

Boolean Functions and Expressions

The **Boolean expressions** in the variables $x_1, x_2, \ldots, x_n$ are defined recursively as follows:

- 0, 1, $x_1, x_2, \ldots, x_n$ are Boolean expressions.
- If E_1 and E_2 are Boolean expressions, then $(-E_1)$, (E_1E_2), and $(E_1 + E_2)$ are Boolean expressions.

Each Boolean expression represents a Boolean function. The values of this function are obtained by substituting 0 and 1 for the variables in the expression.

Boolean Functions and Expressions

For example, we can create Boolean expression in the variables x, y, and z using the “building blocks” 0, 1, x, y, and z, and the construction rules:

Since x and y are Boolean expressions, so is xy.

Since z is a Boolean expression, so is (-z).

Since xy and (-z) are expressions, so is xy + (-z).

… and so on…

Boolean Functions and Expressions

Example: Give a Boolean expression for the Boolean function F(x, y) as defined by the following table:

x	y	F(x, y)
0	0	0
0	1	1
1	0	0
1	1	0

Possible solution: F(x, y) = (-x)·y

Boolean Functions and Expressions

Another Example:

x	y	z	F(x, y, z)
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Possible solution I:

F(x, y, z) = -(xz + y)

Possible solution II:

F(x, y, z) = (-(xz))(-y)

Boolean Functions and Expressions

There is a simple method for deriving a Boolean expression for a function that is defined by a table. This method is based on **minterms**.

Definition: A **literal** is a Boolean variable or its complement. A **minterm** of the Boolean variables x_1, x_2, …, x_n is a Boolean product $y_1y_2…y_n$, where $y_i = x_i$ or $y_i = -x_i$.

Hence, a minterm is a product of n literals, with one literal for each variable.

Boolean Functions and Expressions

Consider F(x,y,z) again:

x	y	z	F(x, y, z)
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

F(x, y, z) = 1 if and only if:

x = y = z = 0 or

x = y = 0, z = 1 or

x = 1, y = z = 0

Therefore,

F(x, y, z) =
(-x)(-y)(-z) +
(-x)(-y)z +
x(-y)(-z)

Boolean Functions and Expressions

Definition: The Boolean functions F and G of n variables are **equal** if and only if $F(b_1, b_2, …, b_n) = G(b_1, b_2, …, b_n)$ whenever $b_1, b_2, …, b_n$ belong to B.

Two different Boolean expressions that represent the same function are called **equivalent**.

For example, the Boolean expressions xy, xy + 0, and xy·1 are equivalent.

Boolean Functions and Expressions

The **complement** of the Boolean function F is the function –F, where $–F(b_1, b_2, …, b_n) = -(F(b_1, b_2, …, b_n))$.

Let F and G be Boolean functions of degree n. The **Boolean sum F+G** and **Boolean product FG** are then defined by

$(F + G)(b_1, b_2, …, b_n) = F(b_1, b_2, …, b_n) + G(b_1, b_2, …, b_n)$

$(FG)(b_1, b_2, …, b_n) = F(b_1, b_2, …, b_n)\ G(b_1, b_2, …, b_n)$

Boolean Functions and Expressions

Question: How many different Boolean functions of degree 1 are there?

Solution: There are four of them, F_1, F_2, F_3, and F_4:

x	F_1	F_2	F_3	F_4
0	0	0	1	1
1	0	1	0	1

Boolean Functions and Expressions

Question: How many different Boolean functions of degree 2 are there?

Solution: There are 16 of them, F_1, F_2, …, F_{16}:

x	y	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Boolean Functions and Expressions

Question: How many different Boolean functions of degree n are there?

Solution:

There are 2^n different n-tuples of 0s and 1s.

A Boolean function is an assignment of 0 or 1 to each of these 2^n different n-tuples.

Therefore, there are 2^{2^n} different Boolean functions.

Boolean Identities

There are useful identities of Boolean expressions that can help us to transform an expression A into an equivalent expression B (see Table 5 on page 815 [6th edition: page 753] in the textbook).

--x = x, law of double complement

x+x = x,idempotent laws
x·x = x

x+0 = x, identity laws
x·1 = x

x+1 = 1, domination laws
x·0 = 0

x+y = y+x, commutative laws
x·y = y·x

x+(y+z) = (x+y)+z, associative laws
x·(y·z) = (x·y)·z

x+yz = (x+y)(x+z), distributive laws
x·(y+z) = (x·y)+(x·z)

-(xy) = -x + -y, De Morgan's laws
-(x+y) = (-x)(-y)

x+xy = x, Absorption laws
x(x+y) = x

x+-x = 1, unit property

x(-x) = 0, zero property

Duality

We can derive additional identities with the help of the **dual** of a Boolean expression.

The dual of a Boolean expression is obtained by interchanging Boolean sums and Boolean products and interchanging 0s and 1s.

Duality

Examples:

The dual of x(y + z) is x + yz.

The dual of -x·1 + (-y + z) is (-x + 0)((-y)z).

The dual is essentially the complement, but with any variable x replaced by -x. (exercise 29, p. 881)

The **dual of a Boolean function F** represented by a Boolean expression is the function represented by the dual of this expression.

This dual function, denoted by F^d, **does not depend** on the particular Boolean expression used to represent F. (exercise 30, page 881 [6th ed. p.756])

Duality

Therefore, an identity between functions represented by Boolean expressions **remains valid** when the duals of both sides of the identity are taken.

We can use this fact, called the **duality principle**, to derive new identities.

For example, consider the absorption law $x(x + y) = x$.

By taking the duals of both sides of this identity, we obtain the equation $x + xy = x$, which is also an identity (and also called an absorption law).

Definition of a Boolean Algebra

All the properties of Boolean functions and expressions that we have discovered also apply to **other mathematical structures** such as propositions and sets and the operations defined on them.

If we can show that a particular structure is a Boolean algebra, then we know that all results established about Boolean algebras apply to this structure.

For this purpose, we need an **abstract definition** of a Boolean algebra.

Definition of a Boolean Algebra

Definition: A Boolean algebra is a set B with two binary operations $\vee$ and $\wedge$, elements 0 and 1, and a unary operation – such that the following properties hold for all x, y, and z in B:

$x \vee 0 = x$ and $x \wedge 1 = x$ (identity laws)

$x \vee (-x) = 1$ and $x \wedge (-x) = 0$ (domination laws)

$(x \vee y) \vee z = x \vee (y \vee z)$ and
$(x \wedge y) \wedge z = x \wedge (y \wedge z)$ and (associative laws)

$x \vee y = y \vee x$ and $x \wedge y = y \wedge x$ (commutative laws)

$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ and
$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ (distributive laws)

Boolean Algebras

Examples of Boolean Algebras are:

1. The algebra of all subsets of a set U, with $+ = \cup$, $\cdot = \cap$, $- =$ complement, $0 = \emptyset$, $1 = U$.
2. The algebra of propositions with symbols $p_1, p_2, \ldots, p_n$, with $+ = \vee$, $\cdot = \wedge$, $- = \neg$, $0 = F$, $1 = T$.
3. If $B_1, \ldots, B_n$ are Boolean Algebras, so is $B_1 \times \ldots \times B_n$, with operations defined coordinate-wise.