IST210 Boot Camp

Boolean Algebra

Bo Luo
bluo@ist.psu.edu

Introduction

- **Logical Statements**
  - Jupiter is the largest planet in the Solar system.
  - 10>5
  - A>5
- In DB: query conditions.
- **Boolean Algebra**
  - the algebra of propositions
  - Basis for computation in binary computer systems

Introduction

- **Boolean Algebra**
  - the algebra of propositions
  - Basis for computation in binary computer systems
- Constants/Truth Values
  - False (0) or True (1)
- Variables / Propositions
  - *A*, *B*, *C*, …, upper case Roman letters
  - Each representing either True or False
- Operations
  - Single variable / Unary operations e.g. ***not (')***
  - Two variables/ Binary operations e.g. ***and* (·)**, ***or* (+)**

Introduction

- Boolean Constants:
  - True, T, 1
  - False, F, 0
- Boolean Operators
  - NOT A, NOT(A), A', $\overline{A}$, ~A
  - A AND B, A * B, A·B, AB, A ∧ B
  - A OR B, A + B, A ∨ B

Boolean Expressions

- Literals
  - A literal is primed (negated) or unprimed variable name
  - E.g. A, a', b, x'
- Boolean **expression**
  - A sequence of zeros, ones and literals separated by Boolean operators
  - E.g. *A·B+C' is a Boolean expression*
- Boolean **equation**
  - Used to express relationships.
  - E.g. *X= A·B+C' is a Boolean equation,* representing the relationship between the value of X and the values of A, B and C.
- Truth table
  - another way of represent a Boolean expression /equation

Boolean vs. binary

- They are different
- 1+1
  - Boolean: true and true = true
  - Binary: 1+1=10

Negation (NOT)

- ***NOT (A)* or $\bar{A}$ or *A'***
- Definition

$$\bar{A} = \begin{cases} 0 & \textbf{if } A = 1 \\ 1 & \textbf{if } A = 0 \end{cases}$$

- Truth table

A	$\bar{A}$
0	1
1	0

Negation (NOT)

- NOT means "not satisfying the condition"
  - E.g. (NOT car_model="beetle")
- ~F=T
- ~T=F
- ~(~F)=~T=F
- ~(~T)=~F=T
- ~(~A)=A

A	$\bar{A}$
0	1
1	0

Conjunction (AND)

- ***A* AND *B* ; *A·B*; *AB*; *A^B***
- ***True*** *if and only if* ***A*** *and* ***B*** *are both* ***true***

A	B	$A \wedge B$
0	1	0
1	0	0
0	0	0
1	1	1

Conjunction (AND)

- AND means to satisfy both
  - E.g. (age>21 and gender="female")
- A^A=A
- A^T=A
- A^F=F
- A^~A=F

A	B	$A \wedge B$
0	1	0
1	0	0
0	0	0
1	1	1

Disjunction (OR)

- ***A* OR *B* ; *A+B*; *A v B***
- ***False*** *if and only if* ***A*** *and* ***B*** *are both* ***false***

A	B	$A \vee B$
0	1	1
1	0	1
0	0	0
1	1	1

Disjunction (OR)

- OR means to satisfy either
  - E.g. (weather="sunny" or temperature>80)
- A v A = A
- A v T = T
- A v F = A
- A v ~A = T

A	B	$A \vee B$
0	1	1
1	0	1
0	0	0
1	1	1

Operator Precedence

- Precedence of the operators determines the order of calculation and the result of the expression

Priority From High To Low

Order of Operators
Parenthesized Expressions
NOT (')
AND (^)
OR (v)

- If same precedence: left associative

E.g. A^B^C=(A^B)^C

Laws of Boolean Algebra (1)

- Commutative Law
  - $A+B=B+A$
  - $A \cdot B = B \cdot A$
- Associate Law
  - $(A+B)+C = A+(B+C)$
  - $(A \cdot B) \cdot C = A \cdot (B \cdot C)$
- Distributive Law
  - $A \cdot (B+C) = A \cdot B + A \cdot C$
  - $A + (B \cdot C) = (A+B) \cdot (A+C)$
- Idempotent /Identity Law
  - $A \cdot A = A$
  - $A+A=A$

Laws of Boolean Algebra (2)

- Compliment Law
  - $A + A' = 1$
  - $A \cdot A' = 0$
- De Morgan's Law
  - $(A+B+C)'=A' \cdot B' \cdot C'$
  - $(A \cdot B \cdot C)'=A'+B'+C'$
- Redundancy Law
  - $A+A \cdot B = A$
  - $A \cdot (A+B)=A$

A Brief Summary of the Laws

	$\overline{\overline{A}} = A$	
	$A+0=A$	$A \cdot 1 = A$
	$A+1=1$	$A \cdot 0 = 0$
Idempient	$A+A=A$	$A \cdot A = A$
Complement	$A+\overline{A}=1$	$A \cdot \overline{A} = 0$
Commutative	$A+B=B+A$	$A \cdot B = B \cdot A$
Associative	$A+(B+C)=(A+B)+C$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$
Distributive	$A+(B \cdot C)=(A+B) \cdot (A+C)$	$A \cdot (B+C) = A \cdot B + A \cdot C$
De Morgan's Law	$(A+B+C)' = A' \cdot B' \cdot C'$	$(A \cdot B \cdot C)' = A' + B' + C'$