Logical Expressions

CS 350: Computer Organization & Assembler Language Programming

[2/2: Associativity/Commutativity]

A. Why?

- Logical operations on bits is the lowest level of computation we do.
- Converting truth tables to expressions tells us how to do logical calculations.

B. Outcomes

At the end of today, you should:

- Be able to perform logical operations on individual bits using truth tables and simplifications.

C. Logical Values, Logical Operations/Connectives

- **Boolean logic** — named after George Boole
- The logical constants are true and false (typically written 1 and 0 or T and F).
- The logical operations/connectives are functions on 1 or 2 logical values.
  - Analogous to +, −, etc. on numbers.
  - *NOT* (unary), *AND*, *OR*, *XOR*, *NAND*, *NOR* (binary)
  - Others often used: *IMPL*, *IFF*
  - Truth tables are often used to show the possible results of logical expressions.

X	***Y***	***X AND Y***	***X OR Y***	***X XOR Y***	***X NAND Y***	***X NOR Y***	***X IFF Y***	***X IMPL Y***		***X***	***NOT X***
0	0	0	0	0	1	1	1	1		0	1
0	1	0	1	1	1	0	0	1		1	0
1	0	0	1	1	1	0	0	0			
1	1	1	1	0	0	0	1	1			

- *NOT X* = 1 iff $X = 0$
- *X AND Y = 1* iff *X = Y = 1*
- *X OR Y = 1* iff *X = 1* or *Y = 1* or both ("inclusive" or)
- *X XOR Y = 1* iff $X \neq Y$ [one is 1, the other is 0] ("exclusive" or)
- *X NAND Y = NOT (X AND Y) = 0* iff *X = Y = 1*
- *X NOR Y = NOT (X OR Y) = 1* iff *X = Y = 0*

- More terminology:
  - *X AND Y* is the **conjunction** of *X* and *Y*; its conjuncts are *X* and *Y*.
  - *X OR Y* is the **disjunction** of *X* and *Y*; its disjuncts are *X* and *Y*.
  - *X XOR Y* is the **exclusive** *OR* of *X* and *Y*; its disjuncts are *X* and *Y*.
  - *IMPL* is the **conditional** operator
    - If true and false are 1 and 0, then *IMPL* behaves like ≤.
      - *0 ≤ 0*, *0 ≤ 1*, *1 ≤ 1*, but not *1 ≤ 0*. I.e., *X IMPL Y* behaves like *(NOT X) OR Y*.
  - *IFF* is the **biconditional**
    - *IFF* behaves like equality: *X IFF Y* is true when *X* and *Y* are both true or both false.
- **Commutative Operators:** (X *op* $Y =$ Y *op* X) [Added 2/2/14]
  - AND, OR, NAND, NOR, XOR, IFF are commutative
  - IMPL is not commutative
- **Associative Operators:** ((X *op* Y) *op* Z = X *op* (Y *op* Z)) [Added 2/2/14]
  - AND, OR are associative
  - NAND, NOR, XOR, IMPL, IFF not associative
- **Precedences** (higher/stronger to lower/lower): *NOT*, *AND/NAND*, *OR/XOR*, *IMPL*, *IFF*.
  - **Example**: *NOT X AND Y OR Z* means *((NOT X) AND Y) OR Z.*
  - **Example**: *NOT (X AND NOT Y OR Y)* means *NOT ((X AND (NOT Y)) OR Y).*

- **Associativity**: [Updated 2/2/14]
  - For *AND*, *OR*, *NAND*, *NOR*, use **left associativity**.
    - $X\ op\ Y\ op\ Z = (X\ op\ Y)\ op\ Z$ by default
    - (For *AND* and *OR* this isn't critical.)
  - For *IMPL* and *IFF*, use **right associativity**.
    - $X\ op\ Y\ op\ Z = X\ op\ (Y\ op\ Z)$ by default
- **Truth tables for larger expressions**
  - A table with n variables has 2^n rows.
  - For expressions with > 1 operator, there are two styles of truth tables.
- Style 1: Each column contains a whole expression. Example:

X	***Y***	***NOT Y***	***X AND NOT Y***	***X AND NOT Y OR Y***	***NOT (X AND NOT Y OR Y)***
0	0	1	0	0	1
0	1	0	0	0	1
1	0	1	1	1	0
1	1	0	0	1	0

- **Style 2**: Columns can contain just the main operator of an expression.
  - We use extra parentheses to make it clearer what operands an operator has. Using this style, the equivalent table is

X	***Y***	***NOT***	***((X***	***AND***	***NOT Y)***	***OR***	***Y)***
0	0	1	0	0	1	0	0
0	1	0	0	0	0	1	1
1	0	0	1	1	1	1	0
1	1	0	1	0	0	1	1

- The first style involves more writing, but it's easy to see what each column's expression is. The second style is briefer but a bit harder to read and write.

D. Alternate Notations

- Instead of words as operators, people often use symbols.

Operator	***Alternatives***
AND	$\wedge$, juxtaposition, *
OR	$\vee$, $+$
XOR	$\oplus$
IMPL	$\rightarrow$, $\Rightarrow$
IFF	$\leftrightarrow$, $\Leftrightarrow$
NOT	$\sim$, $\neg$, !, overbar, prime

- **Note**: Be careful to distinguish between $\overline{X}\ \overline{Y}$, i.e., *NOT X AND NOT Y*, and $\overline{X\ \ Y}$, i.e., *NOT (X AND Y)*.

E. Tautologies, Contradictions, and Contingencies

- A logical expression is a
  - **Tautology** if it is always true (regardless of the values we assign its variables). I.e., its truth table has 1 in all rows.
  - **Contradiction** if it is always false (0's in all rows) .
  - **Contingency** if it has a mix of rows with 0's and 1s
    - (I.e., it's neither a tautology nor a contradiction.)
  - If something is not a tautology, it's a contradiction or contingency
    - Not of "always true" means "sometimes, possibly always false."
  - If something is not a contradiction, it's a tautology or contingency.
    - Not of "always false" means "sometimes, possibly always true."
- Two expressions are **logically equivalent** if their truth tables match.
  - You can substitute logically equivalent expressions for each other.
    - (Substitute equals for equals.)
  - If P and Q are equivalent, then P *IFF* Q is a tautology, and vice versa.
  - If P is a tautology, then *NOT* P is a contradiction, and vice versa.

- If P is a contingency, then $NOT\ P$ is also a contingency, and vice versa.
- But,
  - If P is a not a tautology, then P is a contingency or contadiction
    - (Column not all 1's = Column mix of 0's and 1's or all 0's)
  - If P is a not a contradiction, then P is a contingency or tautology

Logical Expressions

CS 350: Computer Organization & Assembler Language Programming

A. Why?

- Logical operations on bits is the lowest level of computation we do.
- Mathematical calculations on bits can be viewed as logical operations on bits.

B. Outcomes

After this activity, you should be able to

- Perform logical operations on individual bits using truth tables and simplifications.

C. Questions

1. What is the minimal parenthesization for *((NOT X) OR (NOT Y)) AND Z*? (I.e., which parentheses are redundant because of the precedence/associativity rules?)
2. What do we get if we add the redundant parentheses back to *NOT X AND Y OR Z*? (Don't bother adding parentheses around individual variables.)
3. Write a truth table for *NOT (X AND Y)* and *NOT X OR NOT Y*. (You should find they are equivalent; i.e., have exactly the same result columns.) [In the next lecture, we'll see that this is one of DeMorgan's laws.]
4. Write a truth table for *((X NAND Y) NAND Z)* and *(X NAND (Y NAND Z))*. Are they equivalent?

Solution

1. *(NOT X OR NOT Y) AND Z*
2. *((NOT X) AND Y) OR Z*
3. The two expressions are indeed equivalent.

X	***Y***	***NOT***	***(X***	***AND***	***Y)***	***((NOT X)***	***OR***	***(NOT Y))***
0	0	1	0	0	0	1	1	1
0	1	1	0	0	1	1	1	0
1	0	1	1	0	0	0	1	1
1	1	0	1	1	1	0	0	0

4. They are not equivalent. (They're equal iff $X = Z$.)

X	***Y***	***Z***	***((X***	***NAND***	***Y)***	***NAND***	***Z)***	***(X***	***NAND***	***(Y***	***NAND***	***Z))***
0	0	0	0	1	0	1	0	0	1	0	1	0
0	0	1	0	1	0	0	1	0	1	0	1	1
0	1	0	0	1	1	1	0	0	1	1	1	0
0	1	1	0	1	1	0	1	0	1	1	0	1
1	0	0	1	1	0	1	0	1	0	0	1	0
1	0	1	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	1	0	1	0	1	1	0
1	1	1	1	0	1	1	1	1	1	1	0	1

