Boolean Logic

المنطق البولياني

Epp, sections 1.1 and 1.2

Applications of Boolean logic

- Computer programs
- And computer addition
- Logic problems
- Sudoku

Boolean propositions

- A proposition is a statement that can be either true or false
  - “The sky is blue”
  - “I is a Engineering major”
  - “x == y”
- Not propositions:
  - “Are you Bob?”
  - “x := 7”

Boolean variables

- We use Boolean variables to refer to propositions
  - Usually are lower case letters starting with p (i.e. *p, q, r, s*, etc.)
  - A Boolean variable can have one of two values true (T) or false (F)
- A proposition can be…
  - A single variable: p
  - An operation of multiple variables: $p \wedge (q \vee \neg r)$

Introduction to Logical Operators

- About a dozen logical operators
  - Similar to algebraic operators + * - /
- In the following examples,
  - *p* = "Today is Friday"
  - *q* = "Today is my birthday"

Logical operators: Not

- A not operation switches (negates) the truth value
- Symbol: $\neg$ or ~
- In C++ and Java,
  the operand is !
- $\neg p$ = "Today is not Friday"

p	$\neg p$
T	F
F	T

Logical operators: And

- An and operation is true if both operands are true
- Symbol: $\wedge$
  - It's like the 'A' in And
- In C++ and Java, the operand is &&
- $p \wedge q$ = "Today is Friday and today is my birthday"

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Logical operators: Or

- An or operation is true if either operands are true
- Symbol: $\vee$
- In C++ and Java, the operand is ||
- $p \vee q$ = "Today is Friday or today is my birthday (or possibly both)"

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Logical operators: Exclusive Or

- An exclusive or operation is true if one of the operands are true, but false if both are true
- Symbol: $\oplus$
- Often called XOR
- $p \oplus q \equiv (p \vee q) \wedge \neg(p \wedge q)$
- In Java, the operand is ^ (but not in C++)
- $p \oplus q$ = "Today is Friday or today is my birthday, but not both"

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Inclusive Or versus Exclusive Or

- Do these sentences mean inclusive or exclusive or?
  - Experience with C++ or Java is required
  - Lunch includes soup or salad
  - To enter the country, you need a passport or a driver's license
  - Publish or perish

Logical operators: Nand and Nor

- The negation of And and Or, respectively
- Symbols: | and ↓, respectively
  - Nand: $p | q \equiv \neg(p \wedge q)$
  - Nor: $p \downarrow q \equiv \neg(p \vee q)$

p	q	$p \wedge q$	$p \vee q$	$p \mid q$	$p \downarrow q$
T	T	T	T	F	F
T	F	F	T	T	F
F	T	F	T	T	F
F	F	F	F	T	T

Logical operators: Conditional 1

- A conditional means "if *p* then *q*"
- Symbol: $\to$
- $p \to q$ = "If today is Friday, then today is my birthday"
- $p \to q = \neg p \vee q$

p	q	$p \to q$	$\neg p \vee q$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

the antecedent **the consequence**

Logical operators: Conditional 2

- Let p = "I am elected" and q = "I will lower taxes"
- I state: $p \rightarrow q$ = "If I am elected, then I will lower taxes"
- Consider all possibilities
- Note that if p is false, then the conditional is true regardless of whether q is true or false

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Logical operators: Conditional 3

				Conditional	Inverse	Converse	Contra-positive
p	q	$\neg p$	$\neg q$	$p \rightarrow q$	$\neg p \rightarrow \neg q$	$q \rightarrow p$	$\neg q \rightarrow \neg p$
T	T	F	F	T	T	T	T
T	F	F	T	F	T	T	F
F	T	T	F	T	F	F	T
F	F	T	T	T	T	T	T

Logical operators: Conditional 4

- Alternate ways of stating a conditional:
  - *p* implies *q*
  - If *p*, *q*
  - *p* is sufficient for *q*
  - *q* if *p*
  - *q* whenever *p*
  - *q* is necessary for *p*
  - *p* only if *q*

I don’t like this one

Logical operators: Bi-conditional 1

- A bi-conditional means "*p* if and only if *q*"
- Symbol: ↔
- A [illegible]
  " [illegible]
  ([illegible]
- $p \leftrightarrow q \equiv p \rightarrow q \wedge q \rightarrow p$
- Note that a bi-conditional has the opposite truth values of the exclusive or

p	*q*	*p*↔*q*
T	T	T
T	F	F
F	T	F
F	F	T

Logical operators: Bi-conditional 2

- Let *p* = "You take this class" and *q* = "You get a grade"
- Then $p \leftrightarrow q$ means "You take this class if and only if you get a grade"
- Alternatively, it means "If you take this class, then you get a grade and if you get a grade then you take (took) this class"

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Boolean operators summary

		not	not	and	or	xor	nand	nor	conditional	bi-conditional
p	q	$\neg p$	$\neg q$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \mid q$	$p \downarrow q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	F	F	T	T	F	F	F	T	T
T	F	F	T	F	T	T	T	F	F	F
F	T	T	F	F	T	T	T	F	T	F
F	F	T	T	F	F	F	T	T	T	T

- Learn what they mean, don't just memorize the table!

Precedence of operators

- Just as in algebra, operators have precedence
  - 4+3*2 = 4+(3*2), not (4+3)*2
- Precedence order (from highest to lowest): $\neg \wedge \vee \rightarrow \leftrightarrow$
  - The first three are the most important
- This means that $p \vee q \wedge \neg r \rightarrow s \leftrightarrow t$
  yields: $(p \vee (q \wedge (\neg r))) \leftrightarrow (s \rightarrow t)$
- Not is *always* performed before any other operation

Translating English Sentences

- Problem:
  - p = "It is below freezing"
  - q = "It is snowing"

- It is below freezing and it is snowing — $p \wedge q$
- It is below freezing but not snowing — $p \wedge \neg q$
- It is not below freezing and it is not snowing — $\neg p \wedge \neg q$
- It is either snowing or below freezing (or both) — $p \vee q$
- If it is below freezing, it is also snowing — $p \rightarrow q$
- It is either below freezing or it is snowing, but it is not snowing if it is below freezing — $(p \vee q) \wedge (p \rightarrow \neg q)$
- That it is below freezing is necessary and sufficient for it to be snowing — $p \leftrightarrow q$

Translation Example 1

- Heard on the radio:
  - A study showed that there was a correlation between the more children ate dinners with their families and lower rate of substance abuse by those children
  - Announcer conclusions:
    - If children eat more meals with their family, they will have lower substance abuse
    - If they have a higher substance abuse rate, then they did not eat more meals with their family

Translation Example 1

- Let p = “Child eats more meals with family”
- Let q = “Child has less substance abuse
- Announcer conclusions:
  - If children eat more meals with their family, they will have lower substance abuse
    - $p \to q$
  - If they have a higher substance abuse rate, then they did not eat more meals with their family
    - $\neg q \to \neg p$
- Note that $p \to q$ and $\neg q \to \neg p$ are logically equivalent

Translation Example 1

- Let *p* = "Child eats more meals with family"
- Let *q* = "Child has less substance abuse"
- Remember that the study showed a *correlation*, not a *causation*

p	*q*	result	conclusion
T	T	T	T
T	F	?	F
F	T	?	T
F	F	T	T

Translation Example 2

- "I have neither given nor received help on this exam"
  - Rephrased: "I have not given nor received …"
  - Let p = "I have given help on this exam"
  - Let q = "I have received help on this exam"
- Translation is: $\neg p \downarrow q$

p	q	$\neg p$	$\neg p \downarrow q$
T	T	F	F
T	F	F	T
F	T	T	F
F	F	T	F

Translation Example 2

- What they mean is "I have not given and I have not received help on this exam"
  - Or "I have not (given nor received) help on this exam"

p	q	$\neg p \wedge \neg q$	$\neg(p \downarrow q)$
T	T	F	F
T	F	F	F
F	T	F	F
F	F	T	T

- The problem: ¬ has a higher preced[…]c, but not always in English
- Also, "neither" is vague

Tautology and Contradiction

- A tautology is a statement that is always true
  - $p \vee \neg p$ will always be true (Negation Law)
- A contradiction is a statement that is always false
  - $p \wedge \neg p$ will always be false (Negation Law)

p	$p \vee \neg p$	$p \wedge \neg p$
T	T	F
F	T	F

Logical Equivalence

- A logical equivalence means that the two sides always have the same truth values
  - Symbol is ≡ or ⇔
    - We'll use ≡, so as not to confuse it with the bi-conditional

Logical Equivalences of And

- $p \wedge \mathbf{T} \equiv p$ Identity law

p	T	$p \wedge$T
T	T	T
F	T	F

- $p \wedge \mathbf{F} \equiv \mathbf{F}$ Domination law

p	F	$p \wedge$F
T	F	F
F	F	F

Logical Equivalences of And

- $p \wedge p \equiv p$ Idempotent law

p	p	$p \wedge p$
T	T	T
F	F	F

- $p \wedge q \equiv q \wedge p$ Commutative law

p	q	p∧q	q∧p
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

Logical Equivalences of And

- $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$ Associative law

p	q	r	$p \wedge q$	$(p \wedge q) \wedge r$	$q \wedge r$	$p \wedge (q \wedge r)$
T	T	T	T	T	T	T
T	T	F	T	F	F	F
T	F	T	F	F	F	F
T	F	F	F	F	F	F
F	T	T	F	F	T	F
F	T	F	F	F	F	F
F	F	T	F	F	F	F
F	F	F	F	F	F	F

Logical Equivalences of Or

• $p \vee \mathbf{T} \equiv \mathbf{T}$	Identity law
• $p \vee \mathbf{F} \equiv p$	Domination law
• $p \vee p \equiv p$	Idempotent law
• $p \vee q \equiv q \vee p$	Commutative law
• $(p \vee q) \vee r \equiv p \vee (q \vee r)$	Associative law

Corollary of the Associative Law

- $(p \wedge q) \wedge r \equiv p \wedge q \wedge r$
- $(p \vee q) \vee r \equiv p \vee q \vee r$
- Similar to (3+4)+5 = 3+4+5
- Only works if ALL the operators are the same!

Logical Equivalences of Not

- $\neg(\neg p) \equiv p$ — Double negation law
- $p \vee \neg p \equiv T$ — Negation law
- $p \wedge \neg p \equiv F$ — Negation law

DeMorgan's Law

- Probably the most important logical equivalence
- To negate $p \wedge q$ (or $p \vee q$), you "flip" the sign, and negate BOTH p and q
  - Thus, $\neg(p \wedge q) \equiv \neg p \vee \neg q$
  - Thus, $\neg(p \vee q) \equiv \neg p \wedge \neg q$

p	q	$\neg p$	$\neg q$	$p \wedge q$	$\neg(p \wedge q)$	$\neg p \vee \neg q$	$p \vee q$	$\neg(p \vee q)$	$\neg p \wedge \neg q$
T	T	F	F	T	F	F	T	F	F
T	F	F	T	F	T	T	T	F	F
F	T	T	F	F	T	T	T	F	F
F	F	T	T	F	T	T	F	T	T

Yet more equivalences

- Distributive:

  $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

  $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$

- Absorption

  $p \vee (p \wedge q) \equiv p$

  $p \wedge (p \vee q) \equiv p$

How to prove two propositions are equivalent?

- Two methods:
  - Using truth tables
    - Not good for long formulae
    - In this course, only allowed if specifically stated!
  - Using the logical equivalences
    - The preferred method

- Example: show that:

$$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$$

Using Truth Tables

جَامعة
المَنارة
MANARA UNIVERSITY

$$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$$

p	q	r	$p \rightarrow r$	$q \rightarrow r$	$(p \rightarrow r) \vee (q \rightarrow r)$	$p \wedge q$	$(p \wedge q) \rightarrow r$
T	T	T	T	T		T	
T	T	F	F	F		T	
T	F	T	T	T		F	
T	F	F	F	T		F	
F	T	T	T	T		F	
F	T	F	T	F		F	
F	F	T	T	T		F	
F	F	F	T	T		F	

Using Logical Equivalences

$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$ Original statement

Definition of Implication

$p \to q \equiv \neg p \vee q$

$(\neg p \vee r) \vee (\neg q \vee r) \equiv \neg (p \wedge q) \vee r$

DeMorgan's Law

$\neg (p \wedge q) \equiv \neg p \vee \neg q$

$\neg (p \wedge q) \vee r \equiv (\neg p \vee \neg q) \vee r$

Associativity of Or

$(\neg p \vee r) \vee (\neg q \vee r) \equiv \neg p \vee r \vee \neg q \vee r$

$(\neg p \vee \neg q) \vee r \equiv \neg p \vee \neg q \vee r$

Re-arranging

$\neg p \vee r \vee \neg q \vee r \equiv \neg p \vee \neg q \vee r \vee r$

Idempotent Law

$r \vee r \equiv r$

$\neg p \vee \neg q \vee r \vee r \equiv \neg p \vee \neg q \vee r$

Logical Thinking

- At a trial:
  - Bill says: "Sue is guilty and Fred is innocent."
  - Sue says: "If Bill is guilty, then so is Fred."
  - Fred says: "I am innocent, but at least one of the others is guilty."
- Let b = Bill is innocent, f = Fred is innocent, and s = Sue is innocent
- Statements are:
  - $\neg s \wedge f$
  - $\neg b \rightarrow \neg f$
  - $f \wedge (\neg b \vee \neg s)$

Can all of their statements be true?

- Show: (¬s ∧ f) ∧ (¬b → ¬f) ∧ (f ∧ (¬b ∨ ¬s))

b	f	s	¬b	¬f	¬s	¬s∧f	¬b→¬f	f∧(¬b∨¬s)
T	T	T	F	F	F	F	T	F
T	F	T	F	T	F	F	T	F
T	F	F	F	T	T	F	T	F
F	T	T	T	F	F	F	F	T
F	T	F	T	F	T	T	F	T
F	F	T	T	T	F	F	T	F
F	F	F	T	T	T	F	T	F

Are all of their statements true?
Show values for s, b, and f such that the equation is true

$(\neg s \wedge f) \wedge (\neg b \rightarrow \neg f) \wedge (f \wedge (\neg b \vee \neg s)) \equiv T$	Original statement
$(\neg s \wedge f) \wedge (b \vee \neg f) \wedge (f \wedge (\neg b \vee \neg s)) \equiv T$	Definition of implication
$\neg s \wedge f \wedge (b \vee \neg f) \wedge f \wedge (\neg b \vee \neg s) \equiv T$	Associativity of AND
$\neg s \wedge f \wedge f \wedge (b \vee \neg f) \wedge (\neg b \vee \neg s) \equiv T$	Re-arranging
$\neg s \wedge f \wedge (b \vee \neg f) \wedge (\neg b \vee \neg s) \equiv T$	Idempotent law
$f \wedge (b \vee \neg f) \wedge \neg s \wedge (\neg s \vee \neg b) \equiv T$	Re-arranging
$f \wedge (b \vee \neg f) \wedge \neg s \equiv T$	Absorption law
$(f \wedge (b \vee \neg f)) \wedge \neg s \equiv T$	Re-arranging
$((f \wedge b) \vee (f \wedge \neg f)) \wedge \neg s \equiv T$	Distributive law
$((f \wedge b) \vee F) \wedge \neg s \equiv T$	Negation law
$(f \wedge b) \wedge \neg s \equiv T$	Domination law
$f \wedge b \wedge \neg s \equiv T$	Associativity of AND

What if it weren't possible to assign such values to s, b, and f?

$(\neg s \wedge f) \wedge (\neg b \rightarrow \neg f) \wedge (f \wedge (\neg b \vee \neg s)) \wedge s = T$	Original statement
$(\neg s \wedge f) \wedge (b \vee \neg f) \wedge (f \wedge (\neg b \vee \neg s)) \wedge s = T$	Definition of implication
	... (same as previous slide)
$(f \wedge b) \wedge \neg s \wedge s = T$	Domination law
$f \wedge b \wedge \neg s \wedge s = T$	Re-arranging
$f \wedge b \wedge F = T$	Negation law
$f \wedge F = T$	Domination law
$F = T$	Domination law
	Contradiction!

Functional completeness

- All the "extended" operators have equivalences using only the 3 basic operators (and, or, not)
  - The extended operators: nand, nor, xor, conditional, bi-conditional
- Given a limited set of operators, can you write an equivalence of the 3 basic operators?
  - If so, then that group of operators is functionally complete

Exclusive-Or

coffee “or” tea ← ⍰ exclusive-or

How to construct a compound statement for exclusive-or?

p	q	p ⍰ q
T	T	F
T	F	T
F	T	T
F	F	F

Idea 1: Look at the true rows

$$(p \land \neg q) \lor (\neg p \land q)$$

Idea 2: Look at the false rows

$$\neg(p \land q) \land \neg(\neg p \land \neg q)$$

Idea 3: Guess and check

$$(p \lor q) \land \neg(p \land q)$$

Logical Equivalence

$$p \oplus q \equiv (p \vee q) \wedge \neg(p \wedge q)$$

p	q	$p \oplus q$	$p \vee q$	$\neg(p \wedge q)$	
T	T	F	T	F	F
T	F	T	T	T	T
F	T	T	T	T	T
F	F	F	F	T	F

Logical equivalence: Two statements have the same truth table

Writing Logical Formula for a Truth Table

Given a truth table, how to write a logical formula with the same function?

First write down a small formula for each row, so that the formula is true if the inputs are exactly the same as the row.

Then use idea 1 or idea 2.

	p	q	r	output
$p \wedge q \wedge r$	T	T	T	F
$p \wedge q \wedge \neg r$	T	T	F	T
$p \wedge \neg q \wedge r$	T	F	T	T
$p \wedge \neg q \wedge \neg r$	T	F	F	F
$\vee(\neg p \wedge \neg q \wedge r)$	F	T	T	T
$\neg p \wedge q \wedge \neg r$	F	T	F	T
$\neg p \wedge \neg q \wedge r$	F	F	T	T
$\neg p \wedge \neg q \wedge \neg r$	F	F	F	F

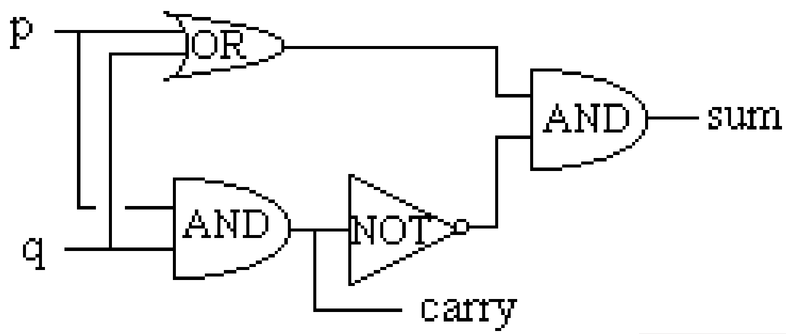
Idea 1: Look at the true rows and take the **"or".**

$$(p \wedge q \wedge \neg r)$$
$$\vee(p \wedge \neg q \wedge r)$$
$$\vee(\neg p \wedge q \wedge r)$$
$$\vee(\neg p \wedge q \wedge \neg r)$$

The formula is true iff the input is one of the true rows.

Writing Logical Formula for a Truth Table

Digital logic:



p	q	sum	carry
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

Idea 2: Look at the false rows, negate and take the **"and"**.

	p	q	r	output
$p \wedge q \wedge r$	T	T	T	F
$p \wedge q \wedge \neg r$	T	T	F	T
$p \wedge \neg q \wedge r$	T	F	T	T
$p \wedge \neg q \wedge \neg r$	T	F	F	F
$\neg p \wedge q \wedge r$	F	T	T	T
$\neg p \wedge q \wedge \neg r$	F	T	F	T
$\neg p \wedge \neg q \wedge r$	F	F	T	T
$\neg p \wedge \neg q \wedge \neg r$	F	F	F	F

$$\neg(p \wedge q \wedge r)$$
$$\wedge \neg(p \wedge \neg q \wedge \neg r)$$
$$\wedge \neg(\neg p \wedge \neg q \wedge \neg r)$$

can be simplified further

The formula is true iff the input is not one of the false row.