CSCI 2011: Introduction and Basic Logic

Chris Kauffman

Last Updated:
Fri Jun 15 16:55:22 CDT 2018

Logistics

Reading: Rosen

- Now: Ch 1.1 - 1.3, Skim Ch 12
- Next: Ch 1.4 - 1.6

Goals

- Motivations for Discrete Math
- Propositional Logic
- Course Mechanics

Exercise: Continuous vs Discrete Mathematics

continuous (adjective) forming an unbroken whole.

discrete (adjective) individually separate and distinct.

- Calculus focuses primarily on continuous mathematics involving the Real and Complex sets of numbers
- Computer science gets much more mileage out of discrete math. Why?
- After a year of programming, everyone should be familiar with two common forms of division.

$$9 \div 2 \quad \rightarrow 4.5 \qquad \div_{real} \text{ a continuous function}$$

$$9 \div 2 \quad \rightarrow 4 \text{ rem } 1 \qquad \div_{integer} \text{ a discrete function}$$

- Aside from the set of Integers, give some other common discrete objects you've seen in computer science

Answers: Continuous vs Discrete Mathematics

Computer scientists get much more mileage out of discrete math.
Why?

- Discrete math focuses on objects that may be finite or infinite but are not infinitely divisible.
- CS deals with 0's and 1's which are inherently discrete.
- CS deals with algorithms that have discrete steps and often operate on discrete objects like...

Common Discrete Objects seen in CS

- Memory Cells, Arrays, Strings, Linked Lists, Boolean variables, Trees, Graphs, **algorithms** in some language
- Sets of any of theses
- Maps of one kind to another
  `name2string : dictionary of string to int`
- Equivalently Functions which map one set to another:
  `length : function 'a list -> int`

First Steps

- Start with basic propositional logic with true/false
- Learn some notation, build up to predicate logic and proofs
- Then move on to consider a wider variety of objects as the course progresses
- Relate the objects to practice of CS
- **Prove** properties of note about them

Propositions and Connectives

- **Propositions** are statements about the world
- Often they have a variable name associated with them as in
  - *p: Jane is studying Computer Science*
  - *q: Jane is taking CSCI 2011 in Summer 2018*
  - *r: Jane is in Keller Hall Room 3-115*
- Variable names allow symbolic manipulation with :

$\neg p$ **Negation or Inverse**, NOT p as in
Jane is NOT studying computer science.

$p \wedge q$ **Conjunction**, p AND q as in
Jane is studying Computer Science AND is taking CSCI 2011 in Summer 2018.

$p \vee r$ **Disjunction** , p OR r as in
Jane is studying Computer Science OR she is in Keller Hall Room 3-115.

$q \rightarrow r$ **Implication**, if p then q as in
If Jane is taking CSCI 2011 in Summer 2018 then she is in Keller Hall Room 3-115.

Exercise: Propositions To and From English

Propositions:

- p : Variable x is an even integer.
- q : Variable y is an odd integer.
- r : The sum of x and y is even.

Symbols to English

In English express

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

What about expressing the statement x *is odd* in symbols?

English to Symbols

In symbols express

- *Variable y is NOT an odd number.*
- *x is even AND y is NOT odd.*
- *IF the sum of x and y is even THEN x is even AND y is NOT odd OR x is NOT even AND y is odd.*

Answers: Propositions To and From English

- p : Variable x is an even integer.
- q : Variable y is an odd integer.
- r : The sum of x and y is even

Symbols to English

In English express

- $p \wedge q$
  x is even AND y is odd
- $\neg r$
  y is not odd
- $(\neg p \wedge q) \rightarrow \neg r$
  IF x is NOT even AND y is odd THEN the sum of x and y is NOT even.

English to Symbols

In symbols express

- *Variable y is NOT an odd integer.*
  - $\neg q$
- *x is even AND y is NOT odd.*
  - $p \wedge \neg q$
- *IF the sum of x and y is even THEN x is even AND y is NOT odd OR x is NOT even AND y is odd.*
  - $r \rightarrow (p \wedge \neg q) \vee (\neg p \wedge q)$

Answers: Propositions To and From English

What about expressing the statement x *is odd* in symbols?

- $\neg p$: x *is NOT even* so x must be odd
- **Assumes** x has to be an an integer, the **domain** x is integers
- If x can be real or complex, then this negation doesn't work
  - Is 4.53 even or odd?
  - What about \$$\pi = 3.14159$...\$
  - How about $i = \sqrt{-1}$?

Truth Tables

- Propositions are statement, **true or false** at a given moment.
  - *p : It is raining outside.*
  - *q : I have an umbrella.*
- A **truth table** for a logical connective shows the value of the combined statement
- Example: Truth table for Implication is **defined** as follows

p	q	$p \rightarrow q$	IF it is raining THEN I have umbrella.
T	T	T	Yes raining, yes umbrella
T	F	F	Yes raining, where's the umbrella?
F	T	T	Not raining, don't care about umbrella
F	F	T	Not raining, don't care about umbrella

- Implication is only false when left side *premise* is True and right side *conclusion* is False

Exercise: Truth Tables for

Define the truth tables for the following based on your experience with programming

- $\wedge$: Conjunction (logical ... ___ ?)
- $\vee$: Disjunction (logical ... ___ ?)

Answers: Truth Tables

p	q	Conjunction AND $p \wedge q$	Disjunction OR $p \vee q$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

- Programming language logical AND (like `a && b`) and logical OR (like `a || b`) are modeled after the mathematical notions that long preceded them.
- There are also usually **bitwise** and/or operators in programming languages (C/Java: `a & b` and `a | b`). More on these later.

A Couple other Common Logical Connectives

$p \leftrightarrow q$ **Biconditional**, p if and only if q as in
Jane gets 3% bonus credit if and only if she participates a lot during lectures.

$p \oplus q$ **Exclusive Or**, p XOR q
p or q but not both and not neither
Would you like fries or tater tots with that?

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

- Many programming languages have a **bitwise** XOR operator
  C/Java: `a ^ b`
- Few have a logical XOR or any biconditional operator

Notation Conventions

Operator Precedence

- Like all languages, logic notation has a notion of operator precedence

$$\text{Highest:} \neg, \wedge, \vee, \oplus, \rightarrow, \leftrightarrow \text{ :Lowest}$$

- Means the following are equivalent

$p \vee q \wedge r$	is	$p \vee (q \wedge r)$	
$\neg p \rightarrow q \vee r$	is	$(\neg p) \rightarrow (q \vee r)$	
$p \rightarrow q \wedge r \leftrightarrow w$	is	$(p \rightarrow (q \wedge r)) \leftrightarrow w$	*omg, stop!*

Notation Varies by Discipline

Logician:		EE/ECE	Logician:		EE/ECE
$\neg p$	is	$\overline{p}$	$p \wedge q$	is	pq
$p \vee q$	is	$p + q$	$p \wedge (\neg q \vee r)$	is	$p(\overline{q} + r)$

- Be aware of both but we'll favor the Logician Notation in

Exercise: Bitwise Operations

- In actual computers, True/False are usually represented in **bits** as 1/0
- Often "variables" are realized as memory cells which hold 8, 16, 32, or 64 bits
- Most computer hardware can perform **bitwise** operations: logical operations on all bits
- Example with **conjunction**: $1101 \wedge 1011$ which is

$$\begin{array}{cr} & 1101 \\ \wedge & 1011 \\ \hline & 1001 \end{array}$$

Compute the following results

$1\,1011 \vee 0\,0010$ which is

$$\begin{array}{cr} & 1\,1011 \\ \vee & 0\,0010 \\ \hline & \end{array}$$

$(00\,1101 \wedge 11\,0101) \oplus 01\,1011$ which is

$$\begin{array}{cr} & 00\,1101 \\ \wedge & 11\,0101 \\ \hline & \\ \oplus & 01\,1011 \\ \hline & \end{array}$$

Answers: Bitwise Operations

- In actual computers, True/False are usually represented in **bits** as 1/0
- Often "variables" are realized as memory cells which hold 8, 16, 32, or 64 bits
- Most computer hardware can perform **bitwise** operations: logical operations on all bits
- Example with **conjunction**: $1101 \wedge 1011$ which is

$$\begin{array}{cr} & 1101 \\ \wedge & 1011 \\ \hline & 1001 \end{array}$$

Compute the following results

$1\ 1011 \vee 0\ 0010$ which is

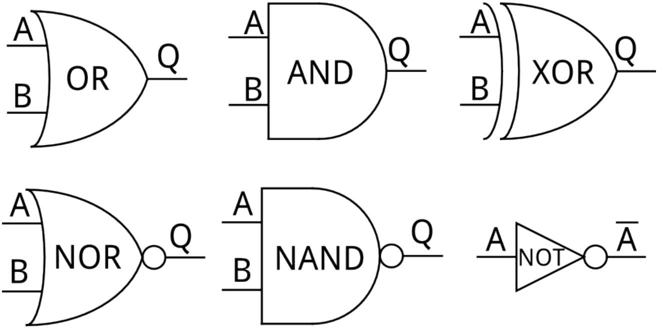
$$\begin{array}{cr} & 1\ 1011 \\ \vee & 0\ 0010 \\ \hline & 1\ 1011 \end{array}$$

$(00\ 1101 \wedge 11\ 0101) \oplus 01\ 1011$ which is

$$\begin{array}{cr} & 00\ 1101 \\ \wedge & 11\ 0101 \\ \hline & 00\ 0101 \\ \oplus & 01\ 1011 \\ \hline & 01\ 1110 \end{array}$$

Logic Gates

- Abstract physical device that implements a logical connectives
- May be implemented with a variety of physical devices including **transistors**, vacuum tubes, mechanical devices, and water pressure
- Physical implementations have many trade-offs: cost, speed, difficulty to manufacture, wetness



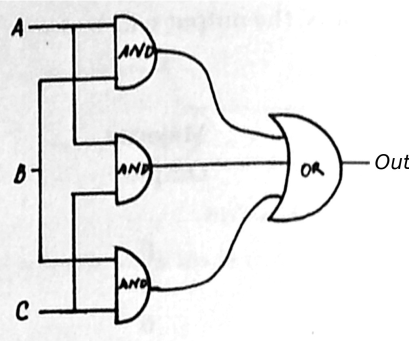
Combinatorial Circuits

- Combination of wires/gates with output solely dependent on input
- No storage of information involved / **stateless**
- Distinguished from **sequential** circuits which involve storage
- Can compute any **Boolean functions** of inputs
  - Set inputs as 0/1
  - After a delay, outputs will be set accordingly
- Examples: AND, OR, NOT are obvious

Exercise: Example Combinatorial Circuit

- Fill in the Truth Table for this circuit with
  - 0 = False
  - 1 = True

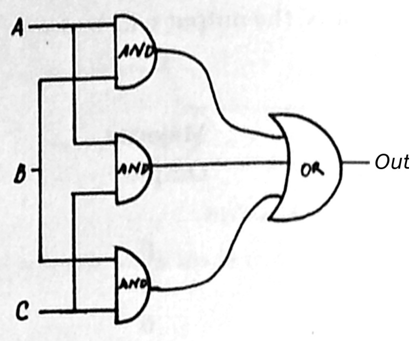
A	B	C	Out
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?



- Write a symbolic expression for the output of the circuit
- Speculate on the "meaning" of this circuit

Answer: Example Combinatorial Circuit

A	B	C	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



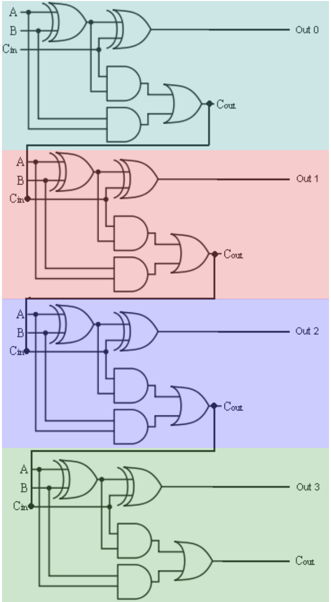
- Output is: $(A \wedge B) \vee (A \wedge C) \vee (B \wedge C)$
- A "majority" circuit: Out is 1 when two or more of A,B,C are 1

Adding with Gates

- Gates can be used for adding multi-bit numbers
- Basis for computer arithmetic
- Make use of a chain of circuits call **full adders** with the resulting truth table

A	B	Out	Carry
0000	0000	0000	0
0000	0001	0001	0
0001	0000	0001	0
0001	0001	0010	0
0001	0010	0011	0
0010	0001	0011	0
0001	0011	0100	0
0011	0011	0110	0
0001	1110	1111	0
0001	1111	0000	1
...			

Chain of 4-bit Adders



Propositional Logic

	Logic Type	AKA	Features
Now	Propositional Logic	Boolean Logic	Truth Vars, Connectives
Later	Predicate Logic	First Order	Predicates, Quantifiers over Vars
Nope	Higher Order Logic(s)	Brain-breaker	Quantifiers over Predicates/Funcs

Will seek to understand some of the techniques and properties of Propositional Logic, then do some work with First Order Logic

Terminology

Tautology A statement that is true regardless of how truth values are assigned to propositions/variables
Ex: $p \vee \neg p$

Contradiction A statement that is false regardless of how truth values are assigned to propositions/variables
Ex: $p \wedge \neg p$

Exercsie: Tautology, Contradiction, or Neither

Determine if each of the following symbolic statements is a Tautology, Contradiction, or Neither

1. $\neg p \vee q \vee p$
2. $p \rightarrow (q \wedge p)$
3. $p \rightarrow (q \vee p)$
4. $(p \rightarrow q) \vee (q \rightarrow p)$
5. $\neg(\neg p \vee q) \leftrightarrow (p \rightarrow q)$
6. $(p \oplus \neg q) \leftrightarrow (\neg p \oplus q)$

Answers: Tautology, Contradiction, or Neither

Determine if each of the following symbolic statements is a Tautology, Contradiction, or Neither

1. $\neg p \vee q \vee p$
   - Tautology: true with any truth assignment, mostly due to $p \vee \neg p$
2. $p \rightarrow (q \wedge p)$
   - Neither: $p = T, q = F$ yields False, $p = F$ yields True
3. $p \rightarrow (q \vee p)$
   - Tautology: true with any truth assignment
4. $(p \rightarrow q) \vee (q \rightarrow p)$
   - Tautology: true with any truth assignment
5. $\neg(\neg p \vee q) \leftrightarrow (p \rightarrow q)$
   - Contradiction: false with any truth assignment
6. $(p \oplus \neg q) \leftrightarrow (\neg p \oplus q)$
   - Tautology: true with any truth assignment

Logical Equivalences

- Two statements A and B are **logically equivalent** if any of the following hold about them
  1. A and B have the same truth table
  2. $A \leftrightarrow B$ is a tautology (always true)
  3. A can be transformed into B using equivalences

Example: Show that $p \to q$ is equivalent to $\neg p \vee q$

Via truth table equivalence / biconditional tautology

p	q	$\neg p \vee q$	$p \to q$	$(\neg p \vee q) \leftrightarrow (p \to q)$
T	T	T	T	T
T	F	F	F	T
F	T	T	T	T
F	F	T	T	T
				Tautology

Exercise: De Morgan's Laws

- These two are important enough to get their own category
- "Distribution" of a negation to parenthesized terms

Show $\neg(p \wedge q)$ and $\neg p \vee \neg q$ are equivalent

Show $\neg(p \vee q)$ and $\neg p \wedge \neg q$ are equivalent

p	q	¬ (p ∧ q)	¬ p ∨ ¬ q
T	T		
T	F		
F	T		
F	F		

Answers: De Morgan's Laws

- These two are important enough to get their own category
- "Distribution" of a negation to parenthesized terms

Show $\neg(p \wedge q)$ and $\neg p \vee \neg q$ are equivalent

p	q	¬ (p ∧ q)	¬ p ∨ ¬ q
T	T	F	F
T	F	T	T
F	T	T	T
F	F	T	T

So $\neg(p \wedge q) \leftrightarrow \neg p \vee \neg q$ is a tautology

Show $\neg(p \vee q)$ and $\neg p \wedge \neg q$ are equivalent

p	q	¬ (p ∨ q)	¬ p ∧ ¬ q
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

So $\neg(p \vee q) \leftrightarrow \neg p \wedge \neg q$ is a tautology

Logical Equivalences

- Large number of common logical equivalences
- Equivalences denoted by the "equivalence" symbol $\equiv$ as in

$$p \vee q \equiv q \vee p$$

- Reads: $p \vee q$ *is equivalent to* $q \vee p$
- Note that $\equiv$ is NOT a logical connective itself; its "higher level" than that, outside and above of the language of Propositional Logic

TABLE 6 **Logical Equivalences.**

Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

Derivations with Equivalence

- Can show equivalence by **deriving** one statement from another
- Start with one statement
- Substitute logically equivalent expressions until the other expression is found

Show $p \rightarrow (q \vee \neg p) \equiv \neg p \vee q$

$$
\begin{array}{rcll}
p \rightarrow (q \vee \neg p) & \equiv & (\neg p) \vee (q \vee \neg p) & \text{by } A \rightarrow B \equiv \neg A \vee B \\
& \equiv & (\neg p) \vee (\neg p \vee q) & \text{by Commutative Laws} \\
& \equiv & (\neg p \vee \neg p) \vee q & \text{by Associative Laws} \\
& \equiv & (\neg p) \vee q & \text{by Idempotent Laws} \\
& \equiv & \neg p \vee q & \text{Equivalence Proved } \blacksquare
\end{array}
$$

There is a convention in mathematical writing to end "proofs" with a symbol like ■ or □. This may be because it *helps readers skip past the proofs more easily.* 😁

Exercise: Equivalence via Derivation

Show: $\neg(p \to q) \equiv p \land \neg q$ using a derivation

Show: $(p \land q) \to (p \lor q)$ is a Tautology

This can be done by showing $(p \land q) \to (p \lor q) \equiv \mathit{True}$

Answers: Equivalence via Derivation

Show: $\neg(p \to q) \equiv p \land \neg q$ using a derivation

$$\begin{aligned}
\neg(p \to q) &\equiv \neg(\neg p \lor q) && \text{by } A \to B \equiv \neg A \lor B \\
&\equiv \neg\neg p \land \neg q && \text{by De Morgan's Laws} \\
&\equiv p \land \neg q && \text{by Double Negation Law } \blacksquare
\end{aligned}$$

Show: $(p \land q) \to (p \lor q)$ is a Tautology

This can be done by showing $(p \land q) \to (p \lor q) \equiv \mathit{True}$

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

$$\begin{aligned}
&\equiv \neg(p \land q) \lor (p \lor q) && \text{by } A \to B \equiv \neg A \lor B \\
&\equiv (\neg p \lor \neg q) \lor (p \lor q) && \text{by De Morgan's Laws} \\
&\equiv (\neg p \lor p) \lor (\neg q \lor q) && \text{by Associative/Commutative Laws} \\
&\equiv (\mathit{True}) \lor (\mathit{True}) && \text{by Negation Laws} \\
&\equiv \mathit{True} && \text{by Idempotent Laws } \blacksquare
\end{aligned}$$

Proofs in Propositional Logic

- A **proof** is a formal description of why some property is true (or false, or unknowable)
- So far seen several methods to prove two propositional logic statements are equivalent, prove something is a tautology/contradiction etc.
  - Remind me: What are those methods again?
- In Propositional Logic, Proofs are relatively easy as some methods *always* result in a proof or a **counter-example**, a specific situation in which the desired property does NOT hold
- Above Propositional Logic, this is no longer true and proofs become harder

Exercise: Prove or Disprove

Prove or Disprove the following statements by any means.

A: $p \leftrightarrow q \equiv (p \rightarrow q) \land (q \rightarrow p)$

B: $\neg(p \rightarrow q) \equiv p \land \neg q$

C: $p \rightarrow q \equiv q \rightarrow \neg p$

Answers: Prove or Disprove

Prove or Disprove the following statements by any means.

A: $p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$: Proved by truth tables

p	q	$p \leftrightarrow q$	$A \wedge B$	$A : (p \to q$	$B : (q \to p)$	
T	T	T	T	T	T	
T	F	F	F	F	T	
F	T	F	F	T	F	
F	F	T	T	T	T	$\blacksquare$

B: $\neg(p \to q) \equiv p \wedge \neg q$: Proved by derivation

$$\begin{aligned} \neg(p \to q) &\equiv \neg(\neg p \vee q) && \text{by } A \to B \equiv \neg A \vee B \\ &\equiv p \wedge \neg q) && \text{by De Morgan's } \blacksquare \end{aligned}$$

C: $p \to q \equiv q \to \neg p$: Disproved by counter example

p	q	$p \to q$	$q \to \neg p$	
T	T	T	F	Not Matching

Proofs of Implication

- Common properties that are proved involve the **implication** and take the following statement structure
  Prove that IF so-and-so is true, THEN what-not is also true.
- Example:
  - *p: a and b are even numbers*
  - *q: sum of a and b is even*
  - $p \to q$: *IF a and b are even numbers, THEN their sum is even.*
  - Prove $p \to q$
- We'll prove that implication later
- Main point: often proofs have the structure of
  - Hypotheses/Assumptions (left side of implication)
  - Conclusion (right side of implication)

Converse, Contrapositive, Inverse

Several variants of implication also commonly arise in proofs. These are as follows.

Converse $q \to p$ as in
IF the sum of a and b is even, THEN a and b are even. (this is not true and can be disproved)

Contrapositive $\neg q \to \neg p$ as in
IF the sum of a and b is NOT even, then a and b are NOT (both) even. (this is true and can be proved)

Inverse $\neg p \to \neg q$ as in
If a and b are NOT even, THEN their sum is NOT even. (not true in the case that a and b are both not even).

Exercise: State in English

Exercise 1.27 c)

IF a positive integer has no divisors other than 1 and itself, THEN it is a prime.

State in English the

1. Converse: $q \rightarrow p$
2. Contrapositive: $\neg q \rightarrow \neg p$
3. Inverse: $\neg p \rightarrow \neg q$

Answers: State in English

Exercise 1.27 c)
IF a positive integer has no divisors other than 1 and itself, THEN it is a prime.
State in English the

1. Converse: $q \rightarrow p$
   IF a positive integer is prime, THEN it has no divisors other than 1 and itself.
2. Contrapositive: $\neg q \rightarrow \neg p$
   IF a positive integer is NOT prime, THEN it DOES have divisors other than 1 and itself.
3. Inverse: $\neg p \rightarrow \neg q$
   IF a positive integer DOES have divisors other than 1 and itself, THEN it is NOT prime.