1 paper of cheat sheet

April 19th Quiz 9-11 in lecture room

Set Theory

- unordered collection of objects
- objects called "members" or "elements"
- { } seperated by commas $\{1, 2, 3\}$

* $\mathbb{Z}$	Integers	$-2, -1, 0, 1, 2$
$\mathbb{N}$	Natural #'s	$1, 2, 3, 4$
$\mathbb{Q}$	Rational #'s	$\frac{a}{b}$
$\mathbb{R}$	Real #'s	$1, 1.25, 3.1456$
$\mathcal{U}$	Universal set	All possible values

Set "V" Vegtables — potato $\in V$

Set "F" fluffy animals — lizard $\notin F$

Set "V" Vegtables — potato, carrot $\in V$

Explicit List	Pattern
$S = \{\text{moe, larry, curly}\}$	$A = \{0, 1, 2, \ldots, 100\}$

Set Builder Notation

variable name, pipe symbol, T/F expression

"All x where x is an even integer"

$\{x \mid x \text{ is an even integer}\}$

$\{x \mid x \in \mathbb{N} \text{ and } x < 7\} \rightarrow \{1, 2, 3, 4, 5, 6\}$

The set $\{1, 4\}$ is a subset of the set $\{1, 3, 4, 5\}$

$\{1, 4\} \subseteq \{1, 3, 4, 5\}$ ← subset

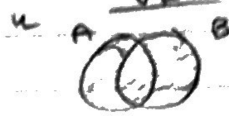
$\{3, 5\} \not\subseteq \{1, 3, 7\}$ ← not a subset

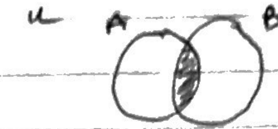
The null set is always a subset

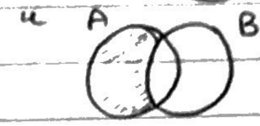
$\emptyset \subseteq \{2, 3\}$

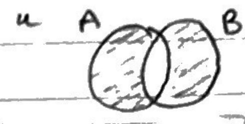
null set = $\{\ \}$ empty

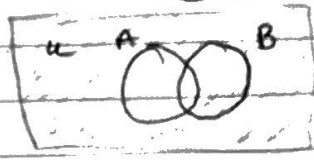
Symbol | VD

$A \cup B$ combines all members into 1 set 

$A \cap B$ contains elements in both sets 

$A - B$
$A \setminus B$ excludes elements found in 1 set from another. (A minus those in B.) 

$A \oplus B$ items in either set BUT NOT BOTH 

A' or $\overline{A}$ or A^c all items not in A 

convert $\cap$ to $\cup$

DeMorgan's Law $(A \cap B)' \equiv A' \cup B'$ $(A \cup B)' \equiv A' \cap B'$

Communitive Law $A \cap B \equiv B \cap A$ $A \cup B \equiv B \cup A$

Idempotent Law $A \cap A \equiv A$ $B \cup B \equiv B$

Involution Law $(A')' \equiv A$

Compliment Law $A \cap A' \equiv \emptyset$ $A \cup A' \equiv \mathcal{U}$
$\emptyset' \equiv \mathcal{U}$ $\mathcal{U}' \equiv \emptyset$

Identity Law $A \cap \mathcal{U} \equiv A$ $A \cup \emptyset \equiv A$

Domination Law $A \cap \emptyset \equiv \emptyset$ $A \cup \mathcal{U} \equiv \mathcal{U}$

Associative Law $A \cap (B \cap C) \equiv (A \cap B) \cap C$
$A \cup (B \cap C) \equiv (A \cup B) \cup C$

Distributive Law $A \cap (B \cup C) \equiv (A \cap B) \cup (A \cap C)$
$A \cup (B \cap C) \equiv (A \cup B) \cap (A \cup C)$

Tuples

- order is important
- same rules as sets but:
  - duplicates are important
  - order is important

n-tuples where "n" is # of elements

→ 2-tuples are also called ordered pairs

tuples → (1,2,3) or <1,2,3> or [1,2,3]

$$(1,2,3) \neq (3,2,1)$$
$$(1,2,3) \neq (1,2,2,3)$$

Duals

Fundamental Products

1. 2^n products where n is # of sets
2. Any 2 fundamental products are disjoint
3. Union of all fundamental products is $\mathcal{U}$

w/ 2 sets

Binary

	A	B
$P_1 = A \cap B$	0	0
$P_2 = A \cap B'$	0	1
$P_3 = A' \cap B$	1	0
$P_4 = A' \cap B'$	0	0

w/ 3 sets

Binary

	A	B	C
$P_1 = A \cap B \cap C$	0	0	0
$P_2 = A \cap B \cap C'$	0	0	1
$P_3 = A \cap B' \cap C$	0	1	0
$P_4 = A \cap B' \cap C'$	0	1	1
$P_5 = A' \cap B \cap C$	1	0	0
$P_6 = A' \cap B \cap C'$	1	0	1
$P_7 = A' \cap B' \cap C$	1	1	0
$P_8 = A' \cap B' \cap C'$	1	1	1

$P_i \cap P_j = \emptyset$ when $i \neq j$ (disjoint) no elements in common

$$(A \cap B) \cup (A' \cap B)$$
$$B \cap (A \cup A')$$
$$B \cap \mathcal{U}$$
$$B$$

Union of 2 fundamental products

$$P_1 \cup P_2 \cup P_3 \cup P_n = \mathcal{U}$$

union of all fund. products is universal

Cardinality

- # of distinct elements in a set

Notation: $|A| \equiv n(A)$

$A = \{1, 3, 5, 7\}$
$|A| = 4$
$B = \{1, 2, 3, 3, 3, 4\}$ ← duplicates don't count
$|B| = 4$

Power Sets

- Power set of a set A is a set of all its possible subsets
  - contains null set
  - notation: $P(A)$ ← for set A

$G = \{a, b\}$
$P(G) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$

Cardinality of a power set
$|G| = 2$
$|P(G)| = 4$

Partitions

- partition of a set A is a collection of non-empty disjoint sets whose union is A

$\{1, 2, 3, \ldots, 9\} \rightarrow \{\{1\}, \{2, 3, 5, 7\}, \{4, 6\}, \{8, 9\}\}$
Valid Partition ↗ • all disjoint • union of all = original set

each subset must be mutually exclusive unless IDENTICAL.

$\{1, 2, 3, 4\}$ $\{\{1\}, \{2\}, \{3\}, \{4\}\}$ ✓
$\{\{1, 2\}, \{1, 2\}, \{3, 4\}\}$ ✓
$\{\{1, 2, 3\}, \{2, 4\}\}$ ✗

Binary # Example

The number 1101 1011 is.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	1	0	1	1	0	1	1

$128+64+16+8+2+1 = \boxed{219}$

Numbers are tuples:
order is important
Hindu-Arabic System $123 \neq 321$

Binary numbers are tuples $1001 \neq 1110$
members of the set $\{0,1\}$

$\{1776, 1846, 1947\} \rightarrow \{(1,7,7,6), (1,8,4,6), (1,9,4,7)\}$

Bit Vectors

- A bit vector can store <u>finite</u>, countable sets using bits
  (also known as bit array, bit set, bit map)
  - compact format that can perform a set operations fast!

Each object in the universe is represented as a single bit in the string of bits

If $x \in A$, then the bit is 1, otherwise 0

Ex)

$U = \{fry, leela, zoidberg, bender, hermes\}$
$U = 11111$

$A = \{fry, leela, bender\}$
$A = 11010$

$U = \{2, 3, 5, 7, 11, 13, 17, 19\}$
$A = \{3, 5, 11, 19\}$

$U = 11111111$
$A = 01101001$

$U = \{a, b, c, d, e, f, g\}$
$A = \{b, c, d\} = 0111000$
$B = \{d, e, f\} = 0001110$

Union $\rightarrow$ $A \cup B = 0111110 = \{b, c, d, e, f\}$ OR
Intersection $\rightarrow$ $A \cap B = 0001000 = \{d\}$ AND

Compliment of Set A

- we must flip all bits from 0 to 1 and 1 to 0
- we can use binary-not or the

$$A' = \{x \mid x \notin A\}$$

So, $U = \{a, b, c, d, e, f, g\}$

$A = \{b, c, d\} = 0111000$

$$\begin{array}{l} \text{not } 0111000 \\ \hline 1000111 = \{a, e, f, g\} \end{array}$$

<u>Exclusion</u> $A - B = \{x \mid x \in A \text{ and } x \notin B\}$

Floating Point Numbers

- FPN are used to represent quantities that cannot be represented by integers

Regular Binary numbers can only stoe <u>whole</u> positive & negative integers

We use the IEEE 754 Standard to store FPN

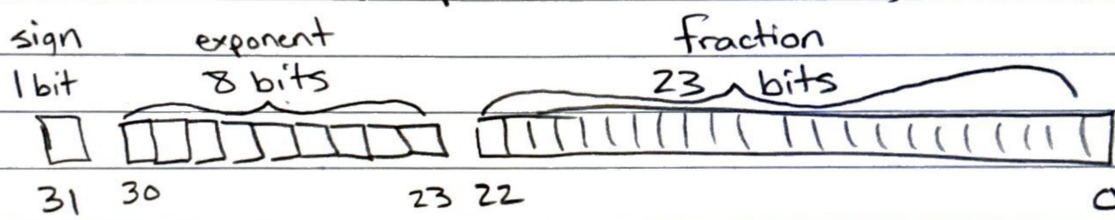
value of a # = mantissa × 2^{exponent}

3 forms

single-precision : 32 bit
double-precision : 64 bit
quad-precision : 128 bit

<u>also supports</u>
negative & positive infity
"not a #" for errors (eg. 1/0)

IEEE 754 single-precision (32 bit)



The fractional field represents part of the mantissa

leading 1 will always be 1 (integer portion)

The exponent field supports negative & positive values but does not use sign-magnititude or 2's compliment

Used a "biased" integer representation
- fixed value added to exponent before storing it
- when interpreting, fixed value is subtracted

Exponent Field

Bias is different depending on precision

single precision = 127
double precision = 1023
quad precision = 16383

Ex for single precision)

exponent of 12 stored as $(+12 + 127) \rightarrow 139$
exponent of -56 stored as $(-56 + 127) \rightarrow 71$

Floating Point Numbers Formula

$$\text{sign}\,(1.\text{fraction}) \times 2^{(\text{exponent} - \text{bias})}$$

Convert 5.5 to Binary

$= 5 \times 10^{-1}$ or $\frac{5}{10}$

2^2	2^1	2^0	2^{-1}	2^{-2}
1	0	1 .	1	

$2^2 + 2^0 + 2^{-1} = 5.5$

Convert 0.125 to Binary

start of decimal

. 0 $.125 \times 2 =$
. 00 $0.25 \times 2 =$
. 001   ~~$0.5 \times 2 =$~~

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
0	0	0 .	0	0	1

$= 0.125$

* Binary is base 2

Convert to Binary

4.375

4	.375
remainder
$4/2 = 2 \text{ r } \underline{0} \rightarrow 0.$	$.375 \times 2 = \underline{0}.75 \rightarrow .0$
$2/2 = 1 \text{ r } \underline{0} \rightarrow 00.$	$.75 \times 2 = \underline{1}.5 \rightarrow .01$
$1/2 = 0 \text{ r } \underline{1} \rightarrow 100.$	$.5 \times 2 = \underline{1}.0 \rightarrow .011$

So 4.375,

$$\boxed{= 100.011}$$

0.625

$.625 \times 2 = \underline{1}.25 \rightarrow .1$

$.25 \times 2 = \underline{0}.50 \rightarrow .10$

$.50 \times 2 = \underline{1}.0 \rightarrow .101$

So $.625 = 0.101$

Convert Binary to FPN

$2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0\ 2^{-1}\ 2^{-2}\ 2^{-3} \ldots$

$$\underbrace{1}_{\text{sign}}\ \underbrace{10000000}_{\text{exponent}}\ \underbrace{10000 \ldots 00}_{\text{fraction}}$$

bias = 127 (single point)

if sign = 1 then negative

if sign = 0 then positive

fraction = 2^{-1} or 0.5

exponent = 2^7 or 128

exponent − bias → 128 − 127

$$\text{sign}(1.\text{fraction}) \times 2^{(\text{exp} - \text{bias})}$$

$$-1.5 \times 2^1 = \boxed{-3}$$

$$\overset{2^7}{0\,1}\underbrace{0\,0\,0\,0\,0\,0\,0\,1}_{129-127}\overset{2^0}{}\;\overbrace{0\,1}^{.25 \;\; 2^{-2}}\,0\,0\,0\,0$$

$$\text{sign}\,(1.\text{fraction}) \times 2^{(\text{exp}-\text{bias})}$$

$$\boxed{\begin{aligned} &= +1.25 \times 2^2 \\ &= 5 \end{aligned}}$$

6.375 in FP binary?

convert to binary first ↳ $\underbrace{1}_{2}\underbrace{1}_{1}0.011$

$.375 \times 2 = \underline{0}.75$

$.75 \times 2 = \underline{1}.5 \rightarrow .01$

$.5 \times 2 = 1.0$

↳ so our exponent is $2^{2+\text{bias}}$ → true exp = 129

↳ $1.\underbrace{10011}_{\text{fraction}}$

↳ sign = 0 because positive

SO...

$$\boxed{\underbrace{0}_{\text{sign}}\;\underbrace{10000001}_{\substack{\text{exp} \\ 129 \text{ or } 2^7+2^0}}\;\underbrace{10011\,000\ldots}_{\text{fraction}}}$$

0.2 in FP binary?

$0.2 \times 2 = \underline{0}.4$

$0.4 \times 2 = 0.8$

$0.8 \times 2 = 1.6$

$.6 \times 2 = 1.2$

$.2 \times 2 = 0.4$

$.4 \times 2 = .8$

↳ $0.\overline{0011}\,0011\,0011 \rightarrow 2^{-3+127} \rightarrow 2^{124}$

↳ $1.\underbrace{10011\,0011}_{\text{fraction}}$

$124 = 2^6$

$$0\;01111100\;\overbrace{1001\,0011\ldots}^{23 \text{ bits}}$$

Special #'s in FPN

zero = 0 00000000 0000000 ... 000

negative zero = 1 00000000 000000 ... 000

+ infinity (inf) = 0 11111111 00000 ... 000

\- inf = 1 11111111 00000 ... 000

NaN = 0 11111111 $\underbrace{010110 \ldots 001}_{\text{just not only 0's}}$

(divide by 0)

8-bit FPN → bias = 3

$\underbrace{0}_{\text{sign}} \quad \underbrace{000}_{\text{exp}} \quad \underbrace{0000}_{\text{fraction}}$

What is 01000000 10110000 00000000 00000000 in decimal?

$$\text{sign}\,(1.\text{fraction}) \times 2^{(\text{exp} - \text{bias})}$$

$129 - 127 = 2$

$$\boxed{\begin{aligned} &+ (1.375) \times 2^2 \\ &= 5.5 \end{aligned}}$$

$2^{-2} + 2^{-3} =$

$0.250 + 0.125 = .375$

Relations

Cross product - a set of ordered pairs (order of operand is important)

Ex) $A = \{ \text{real}, \text{zoom} \}$
$B = \{ \text{teacher}, \text{lawyer} \}$

$A \times B = \{ (\text{real}, \text{teacher}), (\text{real}, \text{lawyer}), (\text{zoom}, \text{teacher}), (\text{zoom}, \text{lawyer}) \}$

tuples

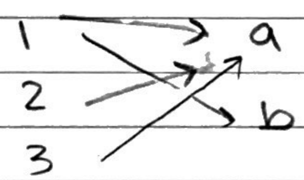
Binary Relations

- T/F statements
- A fact is called a "predicate"

Ex) "x $\underbrace{\text{is bigger than}}_{\text{(relation)}}$ y"

" x <u>and</u> y <u>are</u> ... "

Relation Chart



Ex) Capitols
↳ A is set of all cities
B is set of all states

The domain of the relation is a set of all the first elements in tuple

" the range is 2^{nd} element

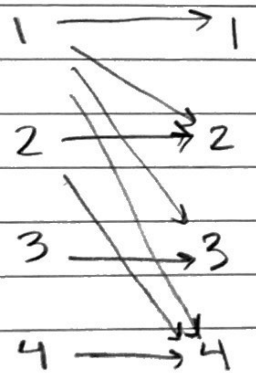
The inverse relation swaps the first & last element in each tuple.

Relations can be ∞ large

Finite: for a set w/ n elements
\# of relations = $n \times n = n^2$

We can represent a relation using set notation

$R_1 = \{ (a,b) \mid a \text{ is bigger than } b \}$
$R_2 = \{ (a,b) \mid a \leq b \}$



Relation of Multiples of Eachother

A reflexive relationship means every element in the domain must be related to itself
(equals, subset of, $\leq$)

Math24.net

Reflexive Relation

- Check domain of tuples

Look for all duals of domains

$R = \{1, 2, 3\}$

$R_1 = \{(1,1), (1,2), (2,2), (3,3)\}$ ✓

$R_2 = \{(1,1), (1,2), (3,2), (3,3)\}$ ✗

No $(2,2)$

Symmetric Relation

- Check tuples as they relate to eachother

Look for (a,b) but no (b,a) means its non-symmetric

$R = \{1, 2\}$

$R_1 = \{(1,2), (2,1), (2,2)\}$ ✓

$R_2 = \{(1,1), (1,2), (2,2)\}$ ✗

Transitive Relation

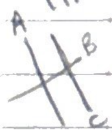
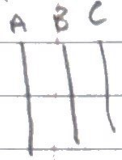
So if (a,b) \$ (b,c) then (a,b) must exist

$R = \{1,2,3,4\}$

✓ $R_1 = \{(2,1), (3,1), (3,2), (4,1), (4,2), (4,3)\}$

Equivalence Relation

\- has to have all 3 true to be equivalence.

	(aRa) Reflexive	(aRb → bRa) Symmetric	(aRb, bRc → aRc) Transitive	E...
perpendicular lines 	✗	✓	✗	✗
parallel lines 	✓	✓	✓	✓
A brother B	✗	✗	✓	✗

Joe

Matt

✓ x x

$\{(1,2), (3,4), (3,2), (2,1)\}$ Closure for Trans...

$(3,1)$ $(2,2)$

$C = \{(1,1), (3,1), (2,2)\}$

Closures

- A closure of relation R is the smallest set that gives R the desired property

$R = \{1, 2, 3; 4\}$

Reflexive closure $C = \{(1,1), (2,2), (3,3), (3,4)\}$

Functions

- must be defined for every element in domain
- only mapped to 1 element in range

function name

domain range

$f : N \to N$

(left can't match to right twice)

(D, R)

$f : Z \to Z$

$f(x) = x^2$

Composition

- Output of one function is the input of another

$f \circ g(x) \equiv f(g(x))$

Cross Products & Databases

SQL is set notation

Fields contain the smallest unit of data
eg Number, Text, Set tuple

- Each field has a unique field name
  eg Name, Age, ID #, Major

- A record is a tuple of data fields
- Records are stored in sets called Tables

A query language is used to:
- locate information
- sort records
- change data in records

Truth Table

$\neg(a \wedge b) \equiv \neg b \vee \neg a$

a	b	$\neg a$	$\neg b$	$a \wedge b$	$\neg(a \wedge b)$	$\neg a \vee \neg$
T	T	F	F	T	F	F
T	F	F	T	F	T	T
F	T	T	F	F	T	T
F	F	T	T	F	T	T

equivalent

Tautology

a	b	$\neg a$	$a \wedge b$	$\neg a \wedge b$	$(a \wedge b) \vee (\neg a \wedge b$
T	T	F	T	F	T
T	F	F	F	F	F
F	T	T	F	T	T
F	F	T	F	F	F

$p \ominus a \wedge \neg p) \vee a \rightarrow p$

p	a	$a \wedge \neg p$	$p \ominus (a \wedge \neg p)$	$\vee p$	$\rightarrow p$
T	T	F	T	T	T
T	F	F	T	T	T
F	T	T	T	T	F
F	F	F	F	F	T

Boolean Logic

statement - declarative statement results in T/F

p and q	T if both are T
p or q	T if 1 is T
not p	T is p is F
p xor q	T if p & q are different
p implies q	T unless p is T & q is F

Logic:	C:	Statement:
$p \wedge q$	p && q	and
$p \vee q$	p || q	or
$\neg p$	!p	not
$p \oplus q$	N/A	xor

Precedence Levels (PEMDAS for logic)

1) $\neg$
2) $\wedge$
3) $\vee$, $\oplus$
4) implies

$a \wedge b = a \cap b$

$a \vee b = a \cup b$

$\neg a = A'$

Statement Always T = Tautology

Statement Always F = Contradiction

Implication

For "p implies q"

p is called the antecendant (hypothesis)
q is called the conclusion

"p implies q" is false only when
p is true and q is false

lowest precedence of boolean operaters

ways to say implies:

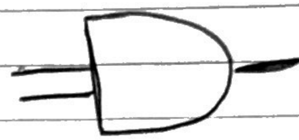
- A implies B
- $A \rightarrow B$
- B if A
- If A then B
- B given A

Circuits

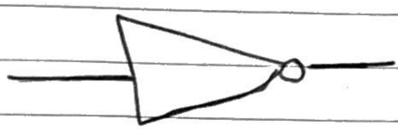
- electronics are made up of gates
- gates take 2 inputs and produce 1 output

Or gate

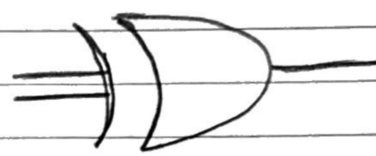
And gate



Not gate



XOR gate



computer engineering notation

$T \equiv 1$

$F \equiv 0$

$\wedge \equiv *$

$\vee \equiv +$

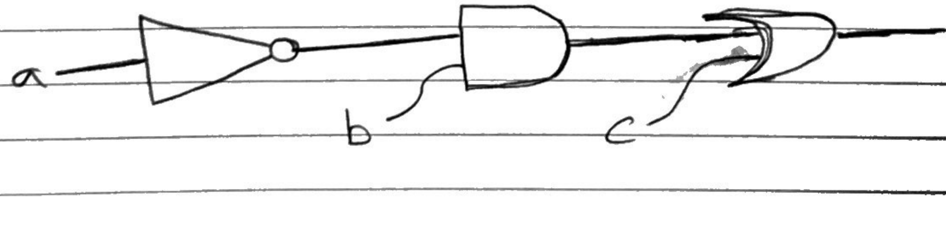
$\neg \equiv '$

Convert Bool Logic to Circuits

1) Choose last operation evaluated
2) Draw gate hook up output
3) Repeat for all operations
4) Attach expression inputs

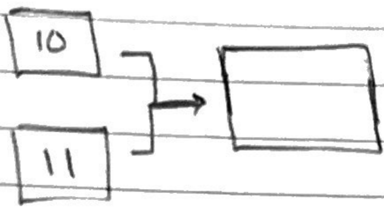
$((\neg a) \wedge b) \vee c$

$a' * b + c$

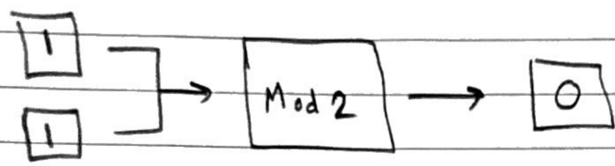


Creating an Arbitrary Circuit

Two Bit Multiplier



Ex) 1 Bit Add Mod 2

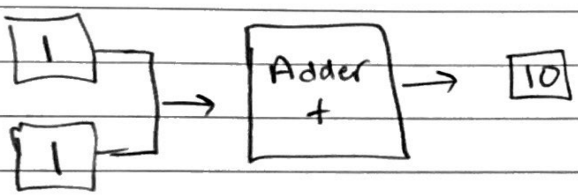


when True

a	b		R
0	0		0
0	1		1
1	0		1
1	1	1	0

$R = a'b + ab'$

Ex) One Bit Adder



when True

a	b	R_1	R_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$R_1 = ab$

Karnaugh Maps

- n variables ⇒ a grid of 2^n squares

Literals are ordered using gray code
- values in the table are not ordered in normal ascending order

c \ ab	00	01	11	10
0	1	0	1	1
1	0	0	1	1

You can overlap squares as long as you cover all 1's

2) Select a minimal set of rectangles where: each rectangle has a power of 2 area and is as large as possible

3) Convert to sum of minterms

a	b	R
0	0	1
0	1	1
1	0	1
1	1	0

b \ a	0	1
0	1	1
1	1	0

$*$ = Don't Care (can be either 0 or 1 whichever is convenient)

2 bit adder

Ex)

a_1	a_0	b_1	b_0	S_2	S_1	S_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

one

S_2

b_1b_0 \ a_1a_0	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	1

$$S_2 = a_1b_1 + a_1a_0b_0 + b_1b_0a_0$$

Functional Completeness

$x' = x \text{ nand } x$

$xy = (x \text{ nand } y) \text{ nand } (x \text{ nand } y)$

$x + y =$

Proofs

Prove $A \to B$ by showing
- whenever a is T B must be T

Proof by Contrapositive – prove the opposite of original equation
- negate both hypothesis & conclusion
- reverse the original implication & solve

for $p \to q$
contrapositive is $\neg q \to \neg p$

Proof by Contradiction – prove theorem by showing it can't be false
- assume its false
- show that if false something Impossible happens

Ex) For all int n, n^3+5 is odd then n is even.

if n^3+5 is odd then n is odd. X

$$n^3 + 5 = 2k+1 \qquad n = 2i+1$$

$$(2i+1)^3 + 5 = 2k+1$$
$$8i^3 + 12i^2 + 6i + 1 + 5 = 2k + 1$$
$$\frac{8i^3 + 12i^2 + 6i + 5}{2} = \frac{2k}{2}$$
$$k = 4i^3 + 6i^2 + 3i + \frac{5}{2}$$

non-integer

$\boxed{\text{So original statement is True.}}$

Arguments

Notation : write a premise on each line

$$\begin{array}{l} p \to q \\ q \to r \\ \hline p \to r \end{array}$$

$p \to q$, $q \to r$ ← premises

$p \to r$ ← consequence

can also use the $\vdash$ symbol.

Arguments are really implications with each premise connected with $\wedge$.

An argument is only invalid is the hypothesis is true \$ conclusion is false

Rules of Inference:

(Law of Detachments)

① Modus Ponens -

- implication true
- implications hypothesis true
- implication conclusion is true

$$\begin{array}{c} p \to q \\ p \\ \hline q \end{array}$$

② Modus Tollens

- implication true
- imp conclusion is false
- imp hypothesis must be false

$$\begin{array}{c} p \to q \\ \neg q \\ \hline \neg p \end{array}$$

③ Disjunctive Syllogism

- an or-statement is true
- one opperands false
- other opperand must be true

$$\begin{array}{c} p \vee q \\ \neg p \\ \hline q \end{array}$$

(4) Hypothetical Syllogism

- gives a logical "chain" of events
- so if $a \to b$ and $b \to c$ then $a \to c$

Ex)

$p \vee a$	$p \vee a$ DS	p
$a \to r$ MP	$p \vee s \to t$	$p \vee s \to t$
$p \vee s \to t$	$\neg a$	$\neg q$
$\neg r$	$\neg q \to u \wedge s$	$u \wedge s$
$\neg q \to u \wedge s$		
t	t	t

simple
first

$t = t$

Ex) The crop is good, but there is not enough water. If there is not a lot of rain, or not a lot of sun, then there is enough water. Therefore the crop is good & there is a lot of sun.

C = crop is good W = enough water S = alot of sun R = alot

#1	True	C
#2		$\neg W$
#3		$(\neg R \vee \neg S) \to W$
		$C \wedge S$

true because #1 & #8
Statement is true

Mod Tollens on #2 & #3

#4	$\neg(\neg R \vee \neg S)$	Demorgans Law
#5	$(\neg(R \vee S))$	Double Neg
#6	$R \vee S$	conj simplification
#7	R	
#8	S	

Logical Fallacies

Fallacy of the Converse
- assumption that is the conclusion is true then the hypothesis is true

Fallacy of Affirming a Disjunct
- assumption that if there are 2 attributes and one is true, the other must be false.

Predicate Notation

constants start w/ lower case
predicates start w/ upper case

student is Sleepy
Sleepy(student)

$x < y$
LessThan(x, y)

For All Notation

every student is sleepy

$\forall_x P(x)$

$\forall_s A(s)$ for all

Exists Notation

$\exists_x P(x)$

$\exists_s A(s)$ exists

$\neg \exists_x P(x) \equiv \forall_x \neg P(x)$
$\neg \forall_x P(x) \equiv \exists_x \neg P(x)$

switching operands

$\exists_x P(x) \equiv \neg \forall_x \neg P(x)$
$\forall_x P(x) \equiv \neg \exists_x \neg P(x)$

Equivalence

Bound & Free Variables

- A variable is free is a value must be supplied to it before expression can be evaluated

- $(x \hat{} 2 < 4 * c)$ ← x, c are free

$\forall_x (x \hat{} 2 < 4 * c)$ ← x is dummy variable
← c is free

Multiple Quantifiers

$P(x,y) \equiv x > y$

$\forall_x \exists_y P(x,y)$

"For all x, there exists a y such that x > y"

English → Logic

"Every cat who has a buddy who has flees, will need anti-flea meds!"

c = cat   b = buddy

B = has buddy
F = has flees
M = meds

$B(c,b)$ ← Predicates
$F(b)$
$M(c)$

$\forall_c \exists_b$ (c & b are buddies, b has flees)

$\exists_b (B(c,b) \land F(b)) \rightarrow M(c)$

$\forall_c (\exists_b (B(c,b) \land F(b))) \rightarrow M(c)$ ✓

"weak"

Induction
Proof by Pattern

Helps prove $\forall_x P(x)$ where the inverse is positive

Steps

1. Prove $P(1)$ "Base case"
2. Proving that $P(n) \rightarrow P(n+1)$ "Induction Step"

Ex) Show that the sum of odd #'s is a square.

n	odd #'s	square	terms of n
1	1	1	$2n-1$
2	1+3	4	$(2n-1)+(2(n-1)-1)$
3	1+3+5	9	$(2n-1)+(2(n-1)-1)+(2(n-2)-1)$
... n	$1+3+5...+(2n-1)$	n^2	
n+1	$1+3+5...+(2n-1)+(2n+1)$	$(n+1)^2$	

1. $P(1) = 1 = 1^2$
2. $1+3+5+...(2n-1) = n^2$ ✓

prove $P(n+1) = n^2+(2n+1) = (n+1)^2$ ✓

Ex) Show that n^3-n is divisable by 3 when n is positive.

1. $P(1) = 1^3-1 = 0$
2. $P(n) \rightarrow P(n+1)$

$n^3-n = 3k$

$(n+1)^3-(n+1) = 3m$

$n^3+3n^2+3n\cancel{+1}-n\cancel{-1} = 3m$

$n^3-n+3n^2+3n = 3m$

$3k+3n^2+3n = 3m$

$3(k+n^2+n) = 3m$ ✓

multiple of 3

Ex) Show that $2^0 + 2^1 + \ldots + 2^n = 2^{n+1} - 1$ when n is positive.

① $P(1) = \quad 2^0 + 2^1 = 3 \; ; \; 2^2 - 1 = 3$ ✓

② $P(n) \rightarrow \quad 2^0 + 2^1 + \ldots 2^n = 2^{n+1} - 1$

$P(n+1) \rightarrow \quad \ldots 2^n + 2^{n+1} = 2^{n+2} - 1$

$$2^{n+1} \cancel{-1} + 2^{n+1} = 2^{n+2} \cancel{-1}$$ ✓

$$2^{n+2} = 2^{n+2}$$ ✓

Ex) Show $1 + 2 + 3 + \ldots (n-1) + n = \frac{n(n+1)}{2}$

① $P(1) = \frac{1(1+1)}{2} = 1$

② $P(n) \rightarrow \quad 1 + 2 + 3 \ldots + (n-1) + n$

$P(n+1) \rightarrow \quad \ldots (n+1) = \frac{(n+1)((n+1)+1)}{2}$

$$= \frac{n(n+1)}{2} + (n+1) = \frac{(n+1)((n+1)+1)}{2}$$

$$= \frac{n(n+1) + 2(n+1)}{2} = \frac{(n+1)(n+2)}{2}$$

$$\frac{n^2 + n + 2n + 2}{2} = \frac{n^2 + 3n + 2}{2}$$ ✓

Strong Induction

Steps
1. Show $P(1)$ is true
2. Assume $P(1), P(2), P(n)$ are true, show $P(n+1)$ is true

Ex) Show any number $n \geq 2$ can be written as the product of primes

1. $P(2) = 2$ ✓
2. $P(n+1) \leftarrow P(n) \wedge P(n+1) \;\dots\; P(2)$

$P(n+1)$ is prime
$P(n+1)$ is composite

$P(3) = 3$
$P(4) = 2 \cdot 2$
$P(5) = 5$
$P(6) = 2 \cdot 3$

Recursion

- Function calling itself

Ex) Let A = a set of #'s where $A_n = 2^n$ for $n \geq 0$.

$$A = \{ 2^n \mid n \in \mathbb{Z} \wedge n \geq 0 \}$$

$A(0) = 1$ $\quad$ $A(n+1) = 2A(n)$

Def $A(n)$ if $n = 0$ return 1
if $n > 0$ return $2 * A(n-1)$

Base Case → $n = 0$ return 1

Ex) Recursively define factorial function $f(n) = n!$

$4! = 4 \times 3 \times 2 \times 1 \rightarrow 4! = 4 \times 3!$

$n! = n \times (n-1)!$

$1! = 1$ $\quad$ $2! = 2$

Fibonacci #'s : Add 2 previous numbers to get new #

(0 1 1 2 3 5 8 13 21 34)

Stacks and Heap

Stacks → first in last out

Sum Rule - tasks mutually exclusive

Ex) 187 CS students
102 Art students

How many combos?
$187 + 102 = 289$ combos

Product Rule - tasks are dependent on eachother
order counts (tuple)
Ex) pin password w/ 4 letters

How many pins?
$26 \times 26 \times 26 \times 26 = 456{,}976$ combos

Sum Rule in Sets
sets are disjoint (don't intersect) we can union all sets and retrieve # of items (cardinality)
$|A_1 \cup A_2| \rightarrow |A_1 + A_2| =$

Product Rule in Sets
Each set dependent, given 2 sets find cross product of sets and retrieve # of items (cardinality)
$|A_1 \times A_2| =$

Inclusion & Exclusion

When counting items in sets, you must not count same element twice

The pigeonhole principle states that if $(k+1)$ objects are placed in k boxes, then there is atleast one box w/ 2 or more objects

Ex) 6 classes Mon - Fri

Then there must be atleast 1 day w/ 2 classes.

$$|A \cup B \cup C| =$$

$$|A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

$|A|$ = cardinality of set A

=

Set exclusion:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Probability

$P(A) = |A| \div |S|$

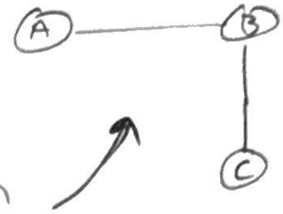
How many int from 1-300 div by 3, 5, 7?

$\frac{|A|}{|S|} = 300$ $|A| = 3, 5,$ or 7

$A = \#$ div 3 $A = 300/3 \rightarrow 100$
$B = \#$ div 5 $B = 300/5 \rightarrow 60$
$C = \#$ div 7 $C = 300/7 \rightarrow 42$

Not disjoint!

$$
\begin{aligned}
|A \cup B \cup C| = &\ |A| + |B| + |C| \\
&- |A \cap B| \\
&- |A \cap C| \\
&- |B \cap C| \\
&+ |A \cap B \cap C| \\
&= 142/300
\end{aligned}
$$

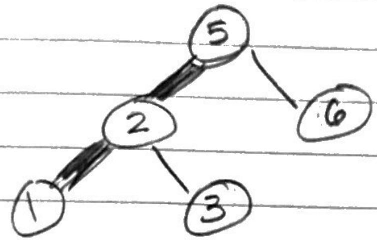


Set representation

$V = \{A, B, C\}$ Vertices

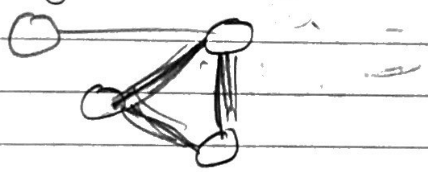
$E = \{(A,B), (B,C)\}$ Edges (Bridges)

A path is a sequence of vertices such that they are adjacent



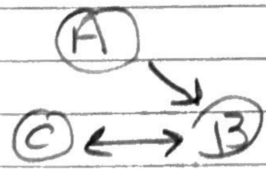
$(1, 2, 5)$

Cycles are when loops are created



A directed graph is one where each edge has a source and target value

Undirected graphs have edges that have some meaning for both directions



$V = \{A, B, C\}$

$E = \{(A,B), (B,C), (C,B)\}$

Languages

- An alphabet is a nonempty set of symbols
- A string is a sequence of zero or more symbols (a tuple)
- λ (lambda) represents a string of 0
- A language is a set of strings

Concatenation 1 string added onto end of other string

* Lexicograph order
  - String length shortest → longest
  - Then dictionary order low → high

A regular expression is a method of specifying a language

Give $A = \{a\}$

$A^* = \{\lambda, a, aa, aaa, aaaa \ldots\}$

$A^+ = \{a, aa, aaa, aaaa \ldots\}$

one or more characters from set A

Finite Automata

a|b	1	2	ba
(a|b)*	4	4	ababba
^b(a⁺|b⁺)	3	3	baaaa
^b⁺a	2	1	b
(a*|b)	5	5	aaab

[abc] {1,3}

$3 + (3 \times 3) + (3 \times 3 \times 3) =$

$3 + 9 + 27 =$