CS1800

Day 7

Admin:

- hw2 due today @ 11:59 PM

Content:

- Computer Representation of sets
- Negation (DeMorgan’s Laws)
- set algebra & logic algebra (very similar!)
- Logic (digital) circuits

Computer representation of sets:

How does a computer store the following sets?

U = {10, 128, 8358, 12, 0, -100} (the universal set, contains all items another set contains
A = {10, 8358, 12, 0, -100}
B = {10, 8358, 0, -100}
C = { 128 }

Approach:

Step 1: Assign a natural number (0, 1, 2, 3...) index (position) to all the items in universal set:

0 1 2 3 4 5
U = {10, 128, 8358, 12, 0, -100}

Step 2: Represent a set as a bit string (sequence of bits).
If bit0 is 1, item0 in set.
If bit1 is 0, item1 not in set.

C = 010000

C ONLY CONTAINS 2ND ITEM

Computer representation of sets: Why is the bit-string a good idea?

1. We need only store every item once, which is important if some of our items would take a lot of memory to store:

$A = \{901824918240192491283938\}$
$B = \{901824918240192491283938, 1\}$
$C = \{901824918240192491283938, 1, 2\}$

2. Our set operation have a natural correspondance with logical operations:

Consider $U = \{\text{blue, yellow, red}\}$
$A = \{\text{blue}, \quad \}$
$B = \{ \quad \text{yellow}, \quad \}$
$A \cup B = \{\text{blue, yellow}, \quad \}$

ITEMS IN A OR B

$A = 100$
$B = 010$
$A \cup B = 110$

Many logical operations on bit string correspond to a set operation

Sets	Logic (on bit string)
U = {blue, yellow, red} A = {blue, } B = { yellow, }	A = 1 0 0 B = 0 1 0
A^c = { YELLOW, RED } ALL ITEMS NOT IN A	A = 1 0 0 A^c = 0 1 1 EACH BIT NEGATED
$A \cup B$ = { BLUE, YELLOW } ALL ITEMS IN A OR B	A = 1 0 0 B = 0 1 0 $A \cup B$ = 1 1 0 APPLY LOGICAL OR OPERATION
$A \cap B = \emptyset$ ALL ITEMS IN A AND B	A = 1 0 0 B = 0 1 0 $A \cap B$ = 0 0 0 APPLY LOGICAL AND OPERATION

Many logical operations on bit string correspond to a set operation

Sets	Logic (on bit string)
U = {blue, yellow, red} A = {blue, } B = { yellow, }	A = 1 0 0 B = 0 1 0
$A \Delta B$ = { BLUE, YELLOW } ALL ITEMS IN A XOR B	A = 1 0 0 B = 0 1 0 $A \Delta B$ = 1 1 0 APPLY LOGICAL XOR OPERATION
C = { RED, BLUE } D = { RED, BLUE } $C \Delta D$	C = 1 0 1 D = 1 0 1 $C \Delta D$ = 0 0 0

GOAL: SHOW $(A \cup B)^c = A^c \cap B^c$

APPROACH

$(A \cup B)^c \subseteq A^c \cap B^c$

ALL ITEMS IN 1ST SET ALSO IN 2ND

$(A \cup B)^c \supseteq A^c \cap B^c$

ALL ITEMS IN 2ND SET ALSO IN 1ST

WANT $(A \cup B)^c \subseteq A^c \cap B^c$

ASSUME $x \in (A \cup B)^c$

$x \notin A \cup B$

$x \notin A$ AND $x \notin B$

$x \in A^c$ AND $x \in B^c$

$x \in A^c \cap B^c$

SO $x \in (A \cup B)^c \rightarrow x \in A^c \cap B^c$

$(A \cup B)^c \supseteq A^c \cap B^c$

IF $x \in A^c \cap B^c$
THEN $x \in (A \cup B)^c$

ASSUME $x \in A^c \cap B^c$

$\Rightarrow x \in A^c$ AND $x \in B^c$

$\Rightarrow x \notin A$ AND $x \notin B$

$\Rightarrow x \notin A \cup B$

$\Rightarrow x \in (A \cup B)^c$

AFTER ALL THAT WORK WE'VE PROVED (ONE OF) DEMORGAN'S LAW FOR SETS

$$(A \cup B)^c = A^c \cap B^c$$

FEEL FAMILIAR?

Swapping operators: sets and logic

SETS

COMPLEMENT ^{c}

INTERSECTION $\cap$

UNION $\cup$

LOGIC

NEGATION $\neg$

AND $\wedge$

INCLUSIVE OR $\vee$

$$(A \cup B)^c = A^c \cap B^c$$

$$\neg(A \vee B) = \neg A \vee \neg B$$

FEEL FAMILIAR YET?

In Class Assignment (not for today, this is complete from day 4's notes):

Build a truth table for each of the two expressions below. Results for both might feel familiar, thats ok :)

$\neg(A \vee B)$

A	B	$A \vee B$	$\neg(A \vee B)$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

$\neg A \wedge \neg B$

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

$$3+7=7+3$$

$$10(1+2) = 10\cdot 1 + 10\cdot 2$$

<take a look at logic_set_identities.pdf together>

(available on course website next to today's notes)

Absorption Laws

P ∧ (P ∨ Q) = P

P ∨ (P ∧ Q) = P

$A \cap (A \cup B) = A$

$A \cup (A \cap B) = A$

Complement Laws

P ∨ ¬ P = T

P ∧ ¬ P = F

$A \cup A^C = U$

$A \cap A^C = \emptyset$

Idempotent Laws

P ∨ P = P

P ∧ P = P

$A \cup A = A$

$A \cap A = A$

Identity

False ∨ P = P

True ∧ P = P

$\emptyset \cup A = A$

$U \cap A = A$

Domination:

True ∨ P = True

False ∧ P = False

$U \cup A = U$

$\emptyset \cap A = \emptyset$

Associative Laws

$(P \vee Q) \vee R = P \vee (Q \vee R)$

$(P \wedge Q) \wedge R = P \wedge (Q \wedge R)$

$(A \cup B) \cup C = A \cup (B \cup C)$

$(A \cap B) \cap C = A \cap (B \cap C)$

Double Negation

$\neg \neg P = P$

$(A^C)^C = A$

DeMorgan's Laws

$\neg(P \vee Q) = \neg P \wedge \neg Q$

$\neg(P \wedge Q) = \neg P \vee \neg Q$

$(A \cup B)^C = A^C \cap B^C$

$(A \cap B)^C = A^C \cup B^C$

Distributive Laws

$P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$

$P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Simplifying boolean or set expressions (set / logic algebra)

$$(X \cup Y) \cap (X \cup Y^c) = X \cup (Y \cap Y^c)$$ DISTRIBUTIVE

$$= X \cup \emptyset$$ COMPLEMENT

$$= X$$ IDENTITY

Simplifying boolean or set expressions (set / logic algebra)

$$\neg(X \vee Y) = \neg X \wedge \neg Y$$

$$\boxed{\neg(\neg A \vee B)} \wedge \neg B = (\neg\neg A \wedge \neg B) \wedge \neg B \qquad \text{DE MORGAN}$$

$$= (A \wedge \neg B) \wedge \neg B \qquad \text{DOUBLE NEGATIVE}$$

$$= A \wedge (\neg B \wedge \neg B) \qquad \text{ASSOCIATIVE}$$

$$= A \wedge \neg B \qquad \text{IDEMPOTENT}$$

IN CLASS ACTIVITY

SIMPLIFY

$$(A \cup B) \cap A^c$$

$$= (A \cap A^c) \cup (B \cap A^c)$$

$$= \emptyset \cup (B \cap A^c)$$

$$= B \cap A^c$$

FEELS LIKE A STEP BACK BUT NOTICE $A \cap A^c$

DISTRIBUTIVE

COMPLEMENT

IDENTITY

$$(\neg X \wedge X) \vee (Y \vee \neg\neg X) = F \vee (Y \vee \neg\neg X)$$ COMPLEMENT

$$= F \vee (Y \vee X)$$ DOUBLE NEG

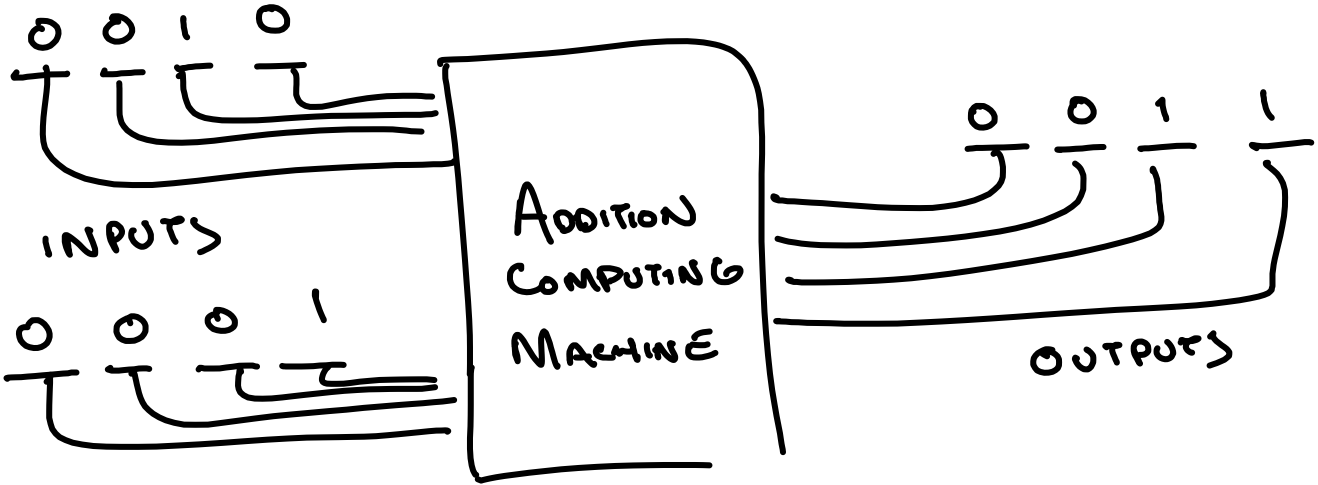
$$= Y \vee X$$ IDENTITY

<lego logic gate video https://youtu.be/RA2po1xk_0A?t=5 >

You can build logic gates (AND, OR, NOT) out of real life things!

- legos
    - (0 = pin pushed in, 1=pin pulled out)
- electronics
    - (0=low voltage, 1=high voltage)
- water
    - (0 = empty tube, 1 = tube has water)
- mechanical switches & gears
    - (0 = lever is down, 1 = lever is up)

Why would you want to build logic gates out of real-life things?

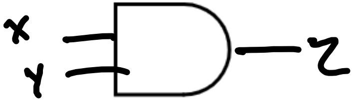


MORE GENERALLY ... COMPUTERS!

Digital Logic (another way of expression boolean algebra)

Many of these gates have to consider the physical layout of their inputs (pins, water, cable etc) so they can be arranged to produce intended behavior.

These "logic gates" emphasize the physical layout and connections between gates:

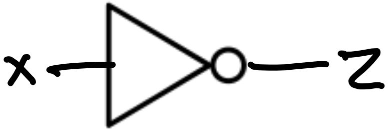


AND

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

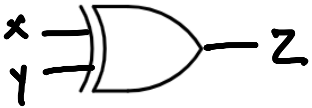
OR

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1



NOT

x	z
0	1
1	0

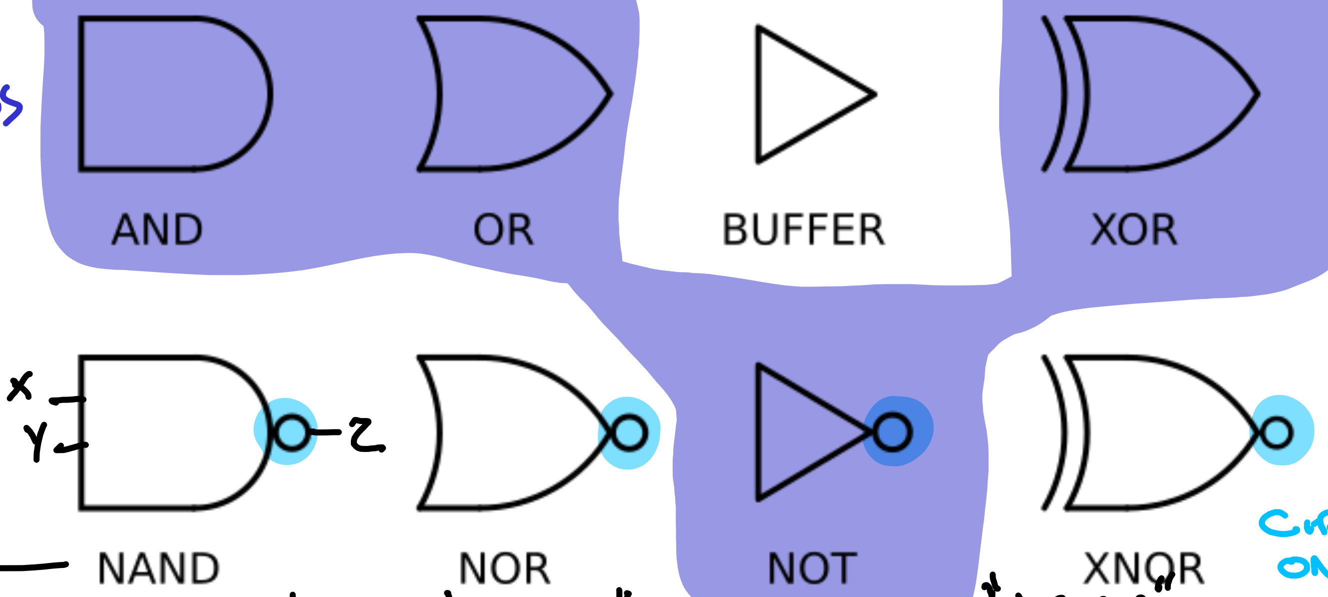


XOR

x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

Digital Logic: some other symbols (you'll never see again in CS1800...)

OLD FRIENDS



X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

"NOT AND"

"NOT OR"

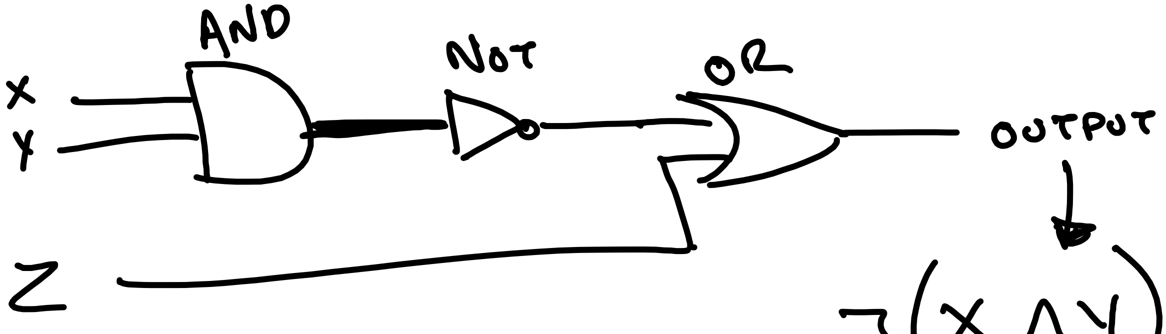
"NOT XOR"

CIRCLES ON BOTTOM ROW NEGATE OUTPUT

Digital Logic: circuits

A circuit is a collection of logic gates which have been connected.

What logic expression is equivilent to the output below?

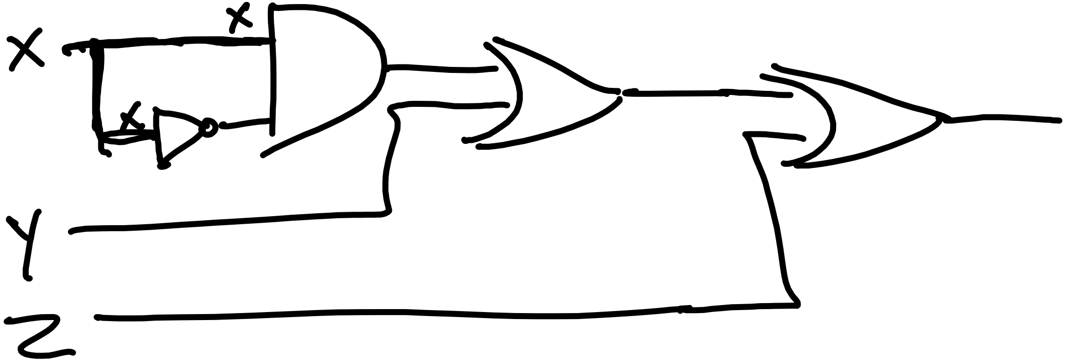


$$\neg(\underline{X \wedge Y}) \vee Z$$

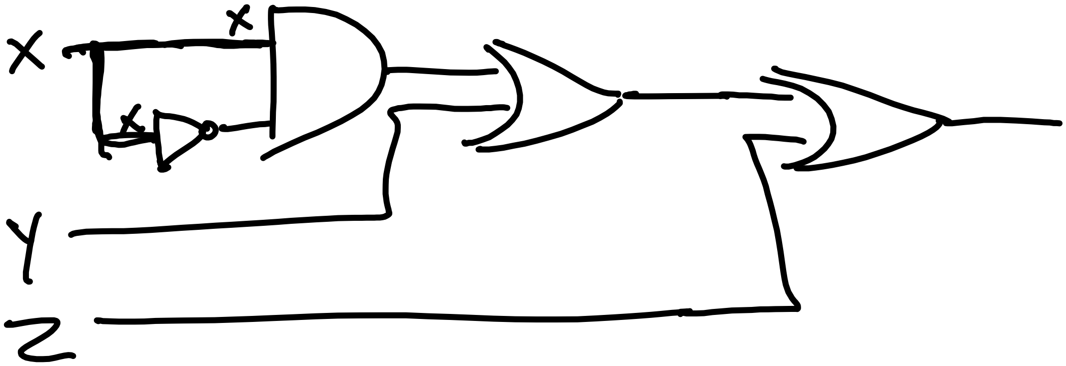
In Class Activity

For the circuit shown below:

- express it using logical symbols
- simplify this expression using the logical identities shown earlier (label each step please)
- draw a new circuit which is equivilent to your simplified expression



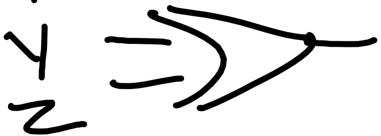
if time / for fun: design your own super complex circuit which is equivilent to something much simpler
(see also, "rube goldberg machine")



$(X \land \neg X) \lor Y \lor Z = \text{FALSE} \lor Y \lor Z$ COMPLEMENT

$= Y \lor Z$ IDENTITY

EQUIVALENTLY:



NEGATING QUANTIFIER

$$\neg\left(\forall x \quad \text{ICE CREAM}(x)\right)$$

$$= \exists x \quad \neg\, \text{ICE CREAM}(x)$$

$$\forall x \quad p(x) \rightarrow q(x)$$

$$\forall x \quad \neg q(x) \rightarrow \neg p(x)$$