Haozhe(Jason) Zhang
CS 5002 Discrete Math

Homework #3

Problem 1 [20 pts (5 pts each)]: Predicate Logic
For this problem, the domain is the set of all solar system objects:

P(x) means x is a Planet

M(x) means x is a Moon

O(x, y) means x orbits y

Formulate the following statements using predicate logic. You may use $x \neq y$ to indicate that the x and y are different.

1. All planets orbit the sun and all moons orbit a planet.
2. Some planets have no moon.
3. Some planets have two or more moons.
4. Some objects orbit the sun that are not planets
5. Everything that orbits the sun is a planet.
   (Also prove that this statement is the negation of the previous statement)

6. Pluto is a planet!

Solution: ($\forall$, $\exists$, $\wedge$, $\vee$, $\rightarrow$, $\neg$, $\equiv$)

1. $\forall x, z$: $(P(x) \rightarrow O(x, sun)) \wedge (M(z) \rightarrow \exists y: P(y) \wedge O(z, y))$
2. $\exists x$: $P(x) \wedge \forall y$: $M(y) \rightarrow \neg O(y, x)$
3. $\exists x, y, z$: $P(x) \wedge M(y) \wedge M(z) \wedge O(y, x) \wedge O(z, x) \wedge (y/=z)$
4. $\exists x\, (O(x, Sun) \wedge \neg P(x))$
5. $\forall x\, (O(x, Sun) \rightarrow P(x))$

$\neg\forall x\, (O(x, Sun) \rightarrow P(x)) \equiv \neg\forall x\, (\neg O(x, Sun) \vee P(x)) \equiv \exists x\, \neg (\neg O(x, Sun) \vee P(x)) \equiv \exists x\, (O(x, Sun) \wedge \neg P(x))$

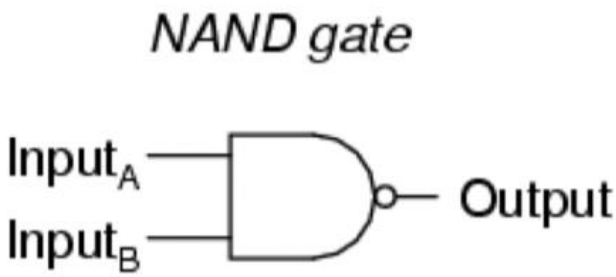
6. P(Pluto)

Haozhe(Jason) Zhang
CS 5002 Discrete Math

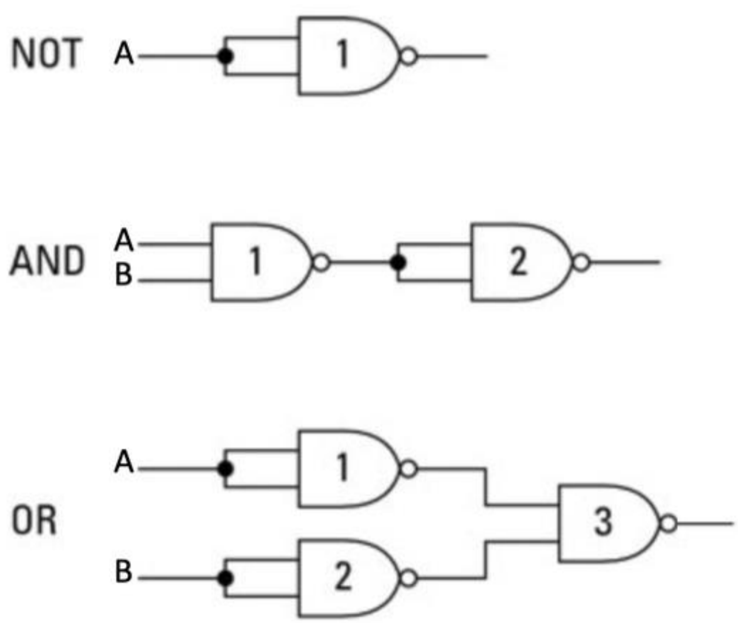
Homework #3

Problem 2 [20 pts] (10, 10): NAND: The universal gate
In this problem we’ll explore the fact that all logical circuits can be implemented using just

NAND gates. The figure below shows you the symbol for a NAND gate and its truth table. We then show you how NAND gates can be wired together to perform the equivalent of a NOT gate, an AND gate, and an OR gate.



A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0



i. Let’s denote p NAND q as p□q. Write a logical expression for the three circuits corresponding to AND, OR, and NOT.
ii. Validate your three logical expressions with three truth tables. For clarity and full credit, show each variable and distinct sub-clause in a separate column, culminating in your final formula.

Haozhe(Jason) Zhang
CS 5002 Discrete Math

Homework #3

Solution:

i.

NOT: not (A and A) $\equiv \neg(A \land A) \equiv \neg A$

AND: not (not (A and B) and not (A and B)) $\equiv \neg(\neg(A \land B) \land \neg(A \land B)) \equiv (A \land B) \lor (A \land B) \equiv A \land B$

OR: not (not (A and A) and not(B and B)) $\equiv \neg(\neg(A \land A) \land \neg(B \land B)) \equiv \neg(\neg A \land \neg B) \equiv A \lor B$

ii. Table shown below

		NOT (¬A)
A	A ∧ A	¬(A ∧ A)
0	0	1
0	0	1
1	1	0
1	1	0

					AND (A ∧ B)
A	B	A ∧ B	¬(A ∧ B)	(A ∧ B) ∧ ¬(A ∧ B)	¬(¬(A ∧ B) ∧ ¬(A ∧ B))
0	0	0	1	1	0
0	1	0	1	1	0
1	0	0	1	1	0
1	1	1	0	0	1

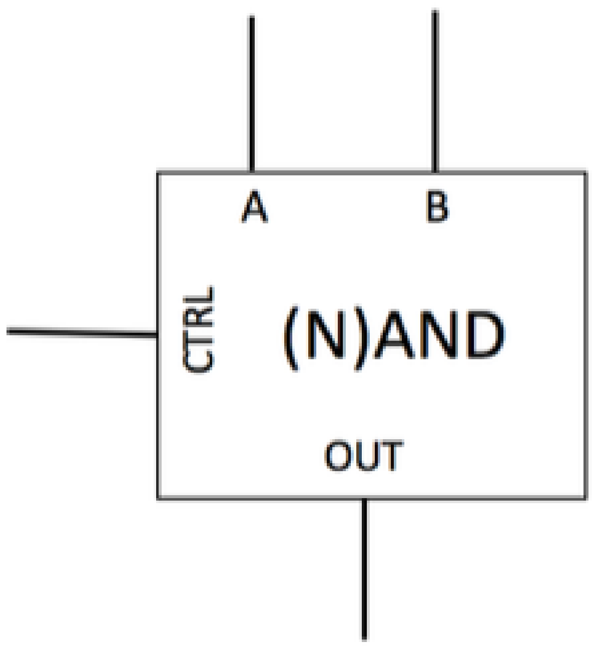
							OR (A ∨ B)
A	B	A ∧ A	¬(A ∧ A)	B ∧ B	¬(B ∧ B)	¬(A ∧ A) ∧ ¬(B ∧ B)	¬(¬(A ∧ A) ∧ ¬(B ∧ B))
0	0	0	1	0	1	1	0
0	1	0	1	1	0	0	1
1	0	1	0	0	1	0	1
1	1	1	0	1	0	0	1

Haozhe(Jason) Zhang
CS 5002 Discrete Math

Homework #3

Problem 3 [25 pts (8,8,9)]: More fun with NAND

In the figure below we depict an unusual little device that we'll call a (N)AND gate. It takes two inputs A and B and a control input labelled CTRL, while producing one output labelled OUT. If the CTRL input is 0, the device works like an AND gate. If the CTRL input is 1, the device works like a NAND gate.



1. Create a truth table for the (N)AND device.
2. Write down an unsimplified disjunctive normal form (DNF) expression and explain why a conjunctive normal form (CNF) expression would have the same number of clauses. (You may use C for the control input.)
3. draw the circuit using the canonical format shown in class.

Haozhe(Jason) Zhang
CS 5002 Discrete Math

Homework #3

Solution:

1. When C(CTRL) input is 0 or 1, the gate works like an AND or NAND. Truth table is shown below.

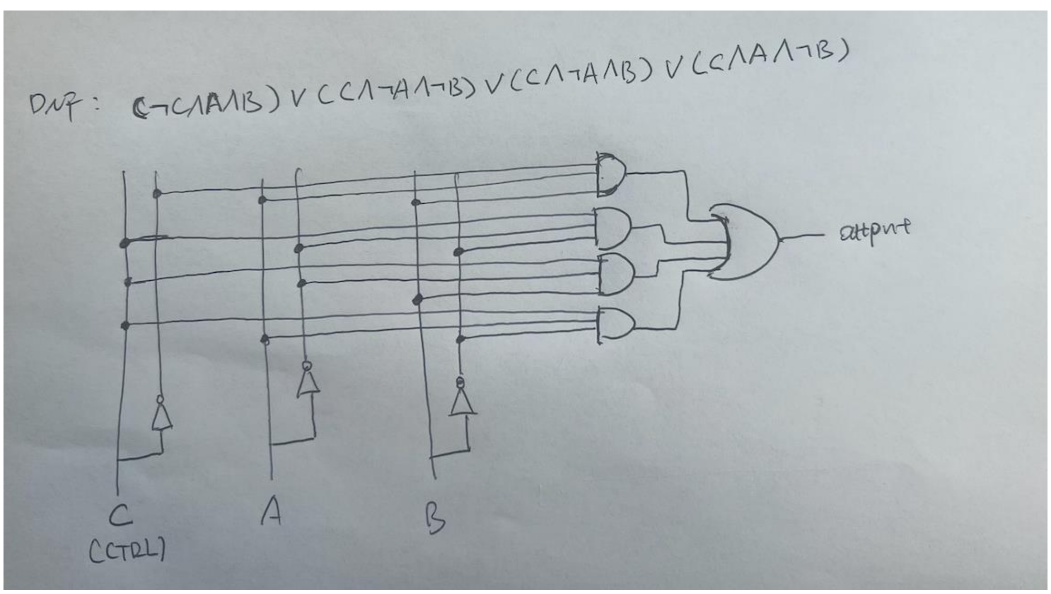
			CTRL = 0	CTRL = 1
C	A	B	A and B	A NAND B
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	
1	0	0		1
1	0	1		1
1	1	0		1
1	1	1		0

2. Disjunctive normal form (DNF). (∧, ¬, ≡, ∨)

DNF form: $(\neg C \wedge A \wedge B) \vee (C \wedge \neg A \wedge \neg B) \vee (C \wedge \neg A \wedge B) \vee (C \wedge A \wedge \neg B)$

DNF have values of 1, while CNF have values of 0, the CNF expression basically means not DNF (¬ DNF) in this problem. When the DNF has 4 clauses, the CNF should have total 8 – 4 = 4 clauses reading from the truth table above. It is easy to approve by adding a not(¬) sign to the DNF expression above to acquire the CNF form.

3.

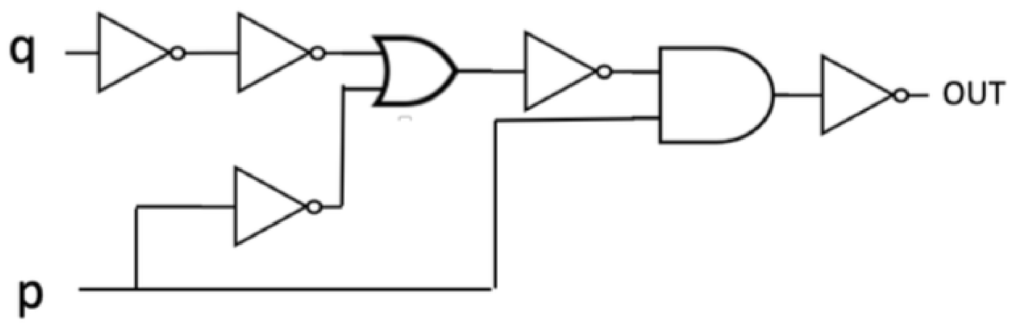


Haozhe(Jason) Zhang
CS 5002 Discrete Math

Homework #3

Problem 4 [25 pts (10,10,5)]: Circuits

i. Write the logic expression that represents the output of the following circuit as a function of its inputs. (Do not simplify your expression). (∧, ¬, ≡, ∨,→)



ii. Use logic rules to simplify your expression. Show each step and state the law of logical equivalence that are applying at each step.

iii. Draw the circuit corresponding to your simplified expression.

Solution:

i. $\neg(\neg(\neg\neg q \vee \neg p) \wedge p)$

ii.

$\neg(\neg(\neg\neg q \vee \neg p) \wedge p) \equiv \neg\neg(\neg\neg q \vee \neg p) \vee \neg p$	De Morgan's
$\neg\neg(\neg\neg q \vee \neg p) \vee \neg p \equiv (\neg\neg q \vee \neg p) \vee \neg p$	Double negation
$(\neg\neg q \vee \neg p) \vee \neg p \equiv (q \vee \neg p) \vee \neg p$	Double negation
$(q \vee \neg p) \vee \neg p \equiv q \vee \neg p$	Identity

iii.