Alex Jia CS 205
157000846
9/24/16

HW 2

Section 1.1

1. Indicate whether each of the following is true or false
    a. True
    b. False
    c. False
2. Using letters for simple statements, translate the following compound English statements into propositional logic formulas
    a. If Jan wins or if he loses, he will be tired.

    (W \/ L) -> T; W=Jan wins, L=Jan loses, T=Jan will be tired.

    b. If prices go up, then housing will be plentiful and expensive; but if housing is not expensive then it will be plentiful.

    (P -> A /\ E) /\ (~E -> P); P=Prices go up, A=Housing will be plentiful, E=Housing will be expensive

    c. Either going to bed or going swimming is sufficient condition for changing clothes.; however, changing clothes does not mean going swimming.

    (B -> C) NAND (S -> C) /\ (C ->S); B=Going to bed, C=Changing clothes, S=Going swimming.

Section 1.1, 1.3

3. Let P be the propositional formula (w -> a) \/ (~w -> i), and let Q be the propositional formula (w -> a) /\ (~w -> i).
    a. Construct the truth table terminating with columns for P and for Q.

W	A	I	(w->a)	(~w->i)	P	Q
T	T	T	T	T	T	T
T	T	F	T	T	T	T
T	F	T	F	T	T	F
T	F	F	F	T	T	F
F	T	T	T	T	T	T

F	T	F	T	F	T	F
F	F	T	T	T	T	T
F	F	F	T	F	T	F

b. Which formula among P, or Q, is a tautology?

   P is the tautology (it's entire row is True)

c. Prove in a second way that the formula you selected in part (b) is tautology, by using an algebraic proof which shows that is is logically equivalent to True. You can use _only_ equivalence rules in Figure 6 (page 26) of Section 1.3 (also appearing in my notes). Please be sure to use the proper format, where you justify each line by a rule.

   Prove: (w -> a) \/ (~w -> i) ⊢ T

1. (~w \/ a) \/ (~w -> i)	Def. ->
2. (~w \/ a) \/ (~(~w) \/ i)	Def. ->
3. (~w \/ a) \/ (w \/ i)	Def. Double Negation law
4. (~w \/ w) \/ (a \/ i)	Def. Commutative Law
5. (T) \/ (a \/ i)	Def. Negation lows (it's called that in his slides)
6. T	Def. Domination Law

d. Take the formula that was NOT a tautology (according to what you said to (b) above), and
   i. Put it into "disjunctive normal form", starting with the distributive laws. *Make the result be as short as possible (have as few symbols as possible). Again, please show your derivation of normal DNF using an algebraic proof, as in part (c). /* This is easy */

      (*using the truth table from part a b/c I am lazy*)
      Q = (w -> a) /\ (~w -> i)

1. (~w \/ a) /\ (~(~w) \/ i)	Def. ->
2. (~w \/ a) /\ (w \/ i)	Def. Double Negation Law
3. (~w \/ w) /\ (a \/ i)	Def. Distributive Laws
4. T /\ (a \/ i)	Def. Negation lows
5. (a \/ i)	Def Domination Law

      DNF: (a \/ i)

   ii. Give a truth assignment to the symbols {w,a,i} which makes the non-tautology be false. (That is how you can tell it is not a tautology)

A, I = False, W can be whatever it wants to be.

e. Now consider the new formula Y = ‘(w -> r) /\ (~w -> ~r)’. Just like in (d(i)), use algebraic equivalences to put it into DNF, and simplify it. /* Again, this is easy. It might explain the intuition that led the class to vote for disjunction. */

Y = (w -> r) /\ (~w -> ~r)

1. (~w \/ r) /\ (~w -> ~r)   Def. ->
2. (~w \/ r) /\ (~(~w) \/ ~r)Def. ->
3. (~w \/ r) /\ (w \/ ~r)   Def. Double Negation
4. (~w \/ w) /\ (r \/ ~r)   Def. Distributive Laws
5. T /\ (r \/ ~r)   Def. Negation lows
6. T /\ T   Def. Negation lows
7. T   Def. Domination Law

DNF: T

4. The NOR operator is defined as: (p NOR q) is true if and only if both p and q are false. Show that {NOR} is _functionally complete_. (I have done in class the case of the NAND operator.)

Assuming that I don’t have to prove again why the {~, /\} operators can represent any and every proposition available (being functionally-complete), (we did this in lecture, so I see no point in wasting time), we can prove that the NOR operator can replace both the ~ and /\ operators:

For ~:

A	~A	A NOR A
T	F	F
F	T	T

For /\:   A /\ B == ~~(A /\ B) == ~(A NOR B) == (A NOR A) NOR (B NOR B)

5. Give Rosen-style proofs for the following (you must use at least one rule of inference)

1. P /\ Q |- P \/ Q

    1. P /\ Q   Given
    2. P   AND Simplification 1
    3. P \/ Q   OR Addition 2

2. ~P |- (P -> Q)

    1. ~P   Given
    2. ~P \/ Q   OR Addition 1
    3. P -> Q   Def. Implication 2

3. ((B /\ C) -> (P /\ Q)) |- ((B /\ C) -> P)

1.	((B /\ C) -> (P /\ Q))	Given
2.	~(B /\ C) \/ (P /\ Q)	Def Implication 1
3.	~(B /\ C) \/ (P)	AND Simplification 2
4.	((B /\ C) -> P)	Def. Implication 3