



## Gradiance Online Accelerated Learning

Zayd

- [Home Page](#)
- [Assignments Due](#)
- [Progress Report](#)
- [Handouts](#)
- [Tutorials](#)
- [Homeworks](#)
- [Lab Projects](#)
- [Log Out](#)

Help

---

**Submission number:** 83963  
**Submission certificate:** CF935065  
**Submission time:** 2014-05-08 01:51:36 PST (GMT - 8:00)

---

**Number of questions:** 5  
**Positive points per question:** 3.0  
**Negative points per question:** 1.0  
**Your score:** 7

---

Based on Sections 10.2 and 10.3 of HMU.

---

1. In the following expressions, - represents negation of a variable. For example, -x stands for "NOT x", + represents logical OR, and juxtaposition represents logical AND (e.g., (x+y)(y+z) represents (x OR y) AND (y OR z)).

Identify the expression that is satisfiable, from the list below.

- a)  $(-x+z)(x)(-z-y)(y+x)$
- b)  $(x+-z)(z+y)(-y+x)(-x)$
- c)  $(-x+z)(-x+-z)(x+z)(x+-z)$
- d)  $(-x+z)(-x+-z)(z+y)(x)$

Answer submitted: **a)**

You have answered the question correctly.

---

Question Explanation:

All choices fall into one of four categories (possibly with clauses reordered):

$(x)(-x+-z)(-z+y)(-y+-x)$  is satisfiable. Let  $x=1$  and  $z=y=0$ .

$(x+z)(x+-z)(-x+z)(-x+-z)$  is not satisfiable. If  $x=1$  and  $z=1$ , the fourth clause is false. If  $x=1$  and  $z=0$ , the third clause is false. If  $x=0$  and  $z=1$ , the second clause is false. If  $x=0$  and  $z=0$ , the first clause is false.

$(x)(-x+z)(-z+y)(-y+-x)$  is not satisfiable. The first clause forces  $x$  to be true if the whole expression is to be satisfied. Then, the second clause forces  $z$  to be true and the third clause forces  $y$  to be true. But then the fourth clause is false.

$(x)(-x+z)(-x+-z)(z+y)$  is not satisfiable. The first clause forces  $x$  to be true if the whole expression is to be satisfied. Then, the second clause forces  $z$  to be true. But then the third clause is false.

---

2. The Boolean expression  $wxyz+u+v$  is equivalent to an expression in 3-CNF (a product of clauses, each clause being the sum of exactly three

literals). Find the simplest such 3-CNF expression and then identify one of its clauses in the list below. Note:  $\neg e$  denotes the negation of  $e$ . Also note: we are looking for an expression that involves only  $u, v, w, x, y$ , and  $z$ , no other variables. Not all boolean expressions can be converted to 3-CNF without introducing new variables, but this one can.

- a)  $(x+y+u)$
- b)  $(y+z+v)$
- c)  $(w+u+v)$
- d)  $(u+v+\neg w)$

Answer submitted: c)

You have answered the question correctly.

#### Question Explanation:

The simplest way to proceed is to use the distributing law of OR over AND, three times, to distribute  $u+v$  over  $wxyz$ . The result is  $(w+u+v)(x+u+v)(y+u+v)(z+u+v)$ .

3. The NOT-ALL-EQUAL 3SAT problem is defined as follows: Given a 3-CNF formula  $F$ , is there a truth assignment for the variables such that each clause has at least one true literal and at least one false literal? The NOT-ALL-EQUAL 3SAT problem is NP-complete.

This question is about trying to reduce the NOT-ALL-EQUAL 3SAT problem to the MAX-CUT problem defined below to show the latter to be NP-complete.

A cut in an undirected graph  $G=(V,E)$  is a partitioning of the set of nodes  $V$  into two disjoint subsets  $V_1$  and  $V_2$ . The size of a cut is the number of edges  $e = (u,v)$  where  $u$  is in  $V_1$  and  $v$  is in  $V_2$ . The MAX-CUT problem is defined as follows: Given an undirected graph  $G=(V,E)$  and a positive integer  $k$ , does  $G$  have a cut of size  $k$  or more?

Given a 3CNF expression  $E$ , we create the graph  $G = (V,E)$  using the transformation given by Theorem 10.18 in Section 10.4.2 on p. 460 of the text. Then given an assignment  $A$ , create a cut  $C$  in  $G$  by partitioning the set of nodes  $V$  as follows: the nodes corresponding to the uncomplemented literals are in set  $V_1$  and those corresponding to the complemented variables are in set  $V_2$ .

For variable  $a$ , let  $a'$  denote  $\text{NOT}(a)$ . Let

$$E = (a + b + c)(a + b' + c)(a' + b' + d)(c' + d' + e)$$

be an instance of NOT-ALL-EQUAL 3SAT. Suppose a cut separates the true nodes from false nodes according to some truth assignment applied to  $E$ . How many edges between nodes corresponding to the literals in the same clause are cut? How many other edges are cut? Find out how the cut-size can be computed for an arbitrary instance of NOT-ALL-EQUAL 3SAT. Then for the instance  $E$ , determine in which of the cases below, the cut-size  $C$  corresponds to the satisfiable assignment given.

- a)  $a = T, b = T, c = F, d = T, e = T, C = 8$
- b)  $a = T, b = F, c = F, d = T, e = F, C = 15$
- c)  $a = T, b = T, c = F, d = F, e = T, C = 13$
- d)  $a = F, b = F, c = T, d = T, e = T, C = 15$

Answer submitted: **b)**

You have answered the question correctly.

Question Explanation:

If each clause has at least two literals with different truth values, they end up in different partitions of the cut. The maximum contribution from a single clause to the total cut-size is hence 2. For 4 clauses, the total is 8. Complemented and uncomplemented literals for the same variable will necessarily be in different partitions of the cut and hence each such pair contributes 1 to the cut-size. There are 7 such pairs in  $E$  with a total contribution of 7. Hence maximum cut-size =  $8 + 7 = 15$ .

4. Use the construction from Theorem 10.15 (p. 457) to convert the following clauses:

1.  $(a+b)$
2.  $(c+d+e+f)$
3.  $(g+h+i+j+k+l+m)$

to products of 3 literals per clause. In each case, the new clauses must be satisfiable if and only if the original clause is satisfiable. For the first clause, introduce variables  $x_1, x_2, \dots$  in that order from the left; for the second introduce  $y_1, y_2, \dots$  in that order from the left, and for the third introduce  $z_1, z_2, \dots$  in that order from the left. Use  $-w$  as shorthand for NOT  $w$ . Then identify, in the list below, the one clause that would appear among the clauses generated by the construction.

- a)  $(e+f+y_1)$
- b)  $(g+-h+z_1)$
- c)  $(e+y_1+-y_2)$
- d)  $(l+z_4+-z_5)$

Answer submitted: **b)**

Your answer is incorrect.

Possible mistake: note that none of the original literals are negated in the new clauses. The reduction of CSAT to 3SAT is in Section 10.3.4 (p. 456).

Question Explanation:

$(a+b)$  becomes  $(a+b+x_1)(a+b+-x_1)$ .

$(c+d+e+f)$  becomes  $(c+d+-y_1)(e+f+y_1)$ .

$(g+h+i+j+k+l+m)$  becomes  $(g+h+-z_1)(i+z_1+-z_2)(j+z_2+-z_3)(k+z_3+-z_4)(l+m+z_4)$ .

The correct choice is: **a)**

5. The polynomial-time reduction from SAT to CSAT, as described in Section 10.3.3 (p. 452), needs to introduce new variables. The reason is

that the obvious manipulation of a boolean expression into an equivalent CNF expression could exponentiate the size of the expression, and therefore could not be polynomial time.

Suppose we apply this construction to the expression  $(u+(vw))+x$ , with the parse implied by the parentheses. Suppose also that when we introduce new variables, we use  $y_1, y_2, \dots$

After constructing the corresponding CNF expression, identify one of its clauses from the list below. Note: logical OR is represented by +, logical AND by juxtaposition, and logical NOT by -.

- a)  $(-y_2+y_1+w)$
- b)  $(x)$
- c)  $(y_2+-y_1+w)$
- d)  $(y_1+v)$

Answer submitted: **b)**

Your answer is incorrect.

Possible error: you may have reversed the constructions for AND and OR. Hint: remember that when you take the OR of two CNF expressions (i.e., AND's of clauses), introduce a new variable  $y$  that is added positively (as  $y$ ) to each clause of the first expression and negatively (as  $\text{NOT}y$ ) to each clause of the second expression. You should consult the reduction from SAT to CSAT in Section 10.3.3 (p. 452).

---

#### Question Explanation:

The first subexpression to which we apply the transformation is  $vw$ . The AND rule is simple: take the AND of the clauses for each side. That gives us  $(v)(w)$  as the CNF expression.

Next, we work on  $u+(vw)$ . The rule for OR requires us to introduce variable  $y_1$ . It is added positively to all the clauses on the left side and negatively to all clauses on the right side. That gives us  $(y_1+u)(-y_1+v)(-y_1+w)$ .

Finally, we apply the same transformation to  $(u+(vw))+x$ , introducing  $y_2$ . The final answer is  $(y_2+y_1+u)(y_2+-y_1+v)(y_2+-y_1+w)(-y_2+x)$ .

The correct choice is: **c)**

---