CS 722 Fall 2016
Homework Assignment #2
Solutions

1. Consider the NTM in Question 3 here. Derive the exact formula of the worst-case time complexity function, $W_M(n)$, of this NTM. Justify your answer.

   $W_M(n) = n+1$. Since the tape head only moves to right, there will be at most one transition on the leftmost blank symbol "_" leading to q_{accept}. So every branch of the NTM reaches a dead end in at most n transitions or else reads the leftmost "_" in q_1 or q_2 in n transitions. In the latter case, the NTM enters q_{accept} in n+1 transitions.

2. This question concerns Theorem VtoN applied to the NTM in Question 3 here. Consider the following verifier V for the language $L = \{ wab^i : i \geq 0 \} \cup \{ wac^j : j \geq 0 \}$ where $\Sigma = \{ a, b, c \}$, $w \in \Sigma^*$. A proposed certificate encodes a pair of integers $<n_1, n_2>$.

   step 1: Read n_1. Move the head back to the leftmost cell on tape. Scan the initial n_1 symbols on tape (skip w).
   step 2: Check if the current tape symbol is "a"; if it is not, reject.
   step 3: Read n_2.
   If $n_2 = 0$, check if "a" is followed by b^i, $i \geq 0$; if so, accept, o.w. reject.
   If $n_2 = 1$, check if "a" is followed by c^j, $j \geq 0$; if so, accept, o.w. reject.

   (a) Prove that V verifies L.

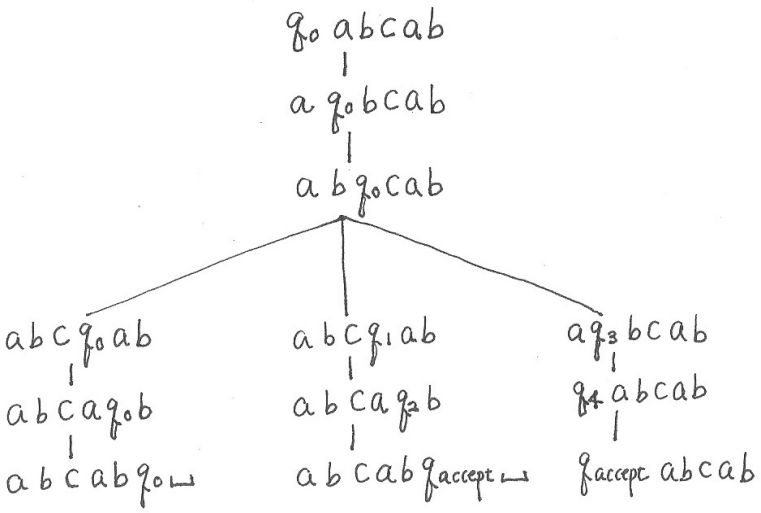
   We need to prove L = L(V).
   Suppose $x \in L$. Then $x = wab^i$ for some $w \in \Sigma^*$ and $i \geq 0$, or $x = wac^j$ for some $w \in \Sigma^*$ and $j \geq 0$. In the first case, let a proposed certificate be $<|w|, 0>$; in the second case, let a proposed certificate be $<|w|, 1>$. In either case, V accepts <x, the proposed certificate>, hence $x \in L(V)$.
   Conversely suppose $x \in L(V)$. Then there exits a certificate $<n_1, n_2>$ s.t. V accepts $<x, <n_1, n_2>>$. By the operational definition of V, this means that $x = wab^i$ for some $w \in \Sigma^*$ and $i \geq 0$, or $x = wac^j$ for some $w \in \Sigma^*$ and $j \geq 0$. Hence $x \in L$.

   (b) Describe the operation of an NTM simulating V.

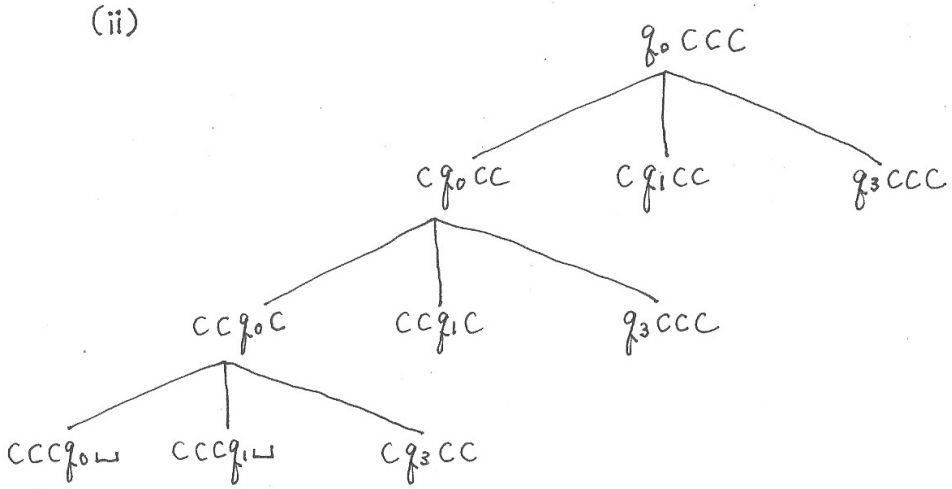
   The simulating NTM nondeterministically generates a string c encoding a pair of integers $<n_1, n_2>$ and writes it after w# where w is the input string to be verified, then runs V on w#c.

3. This question concerns two examples of the access-path encoding of computation trees' branches used in Theorem NtoV. Consider the NTM in Question 2 here, and the following computation trees of this NTM on the input strings "abcab" and "ccc":

(i)

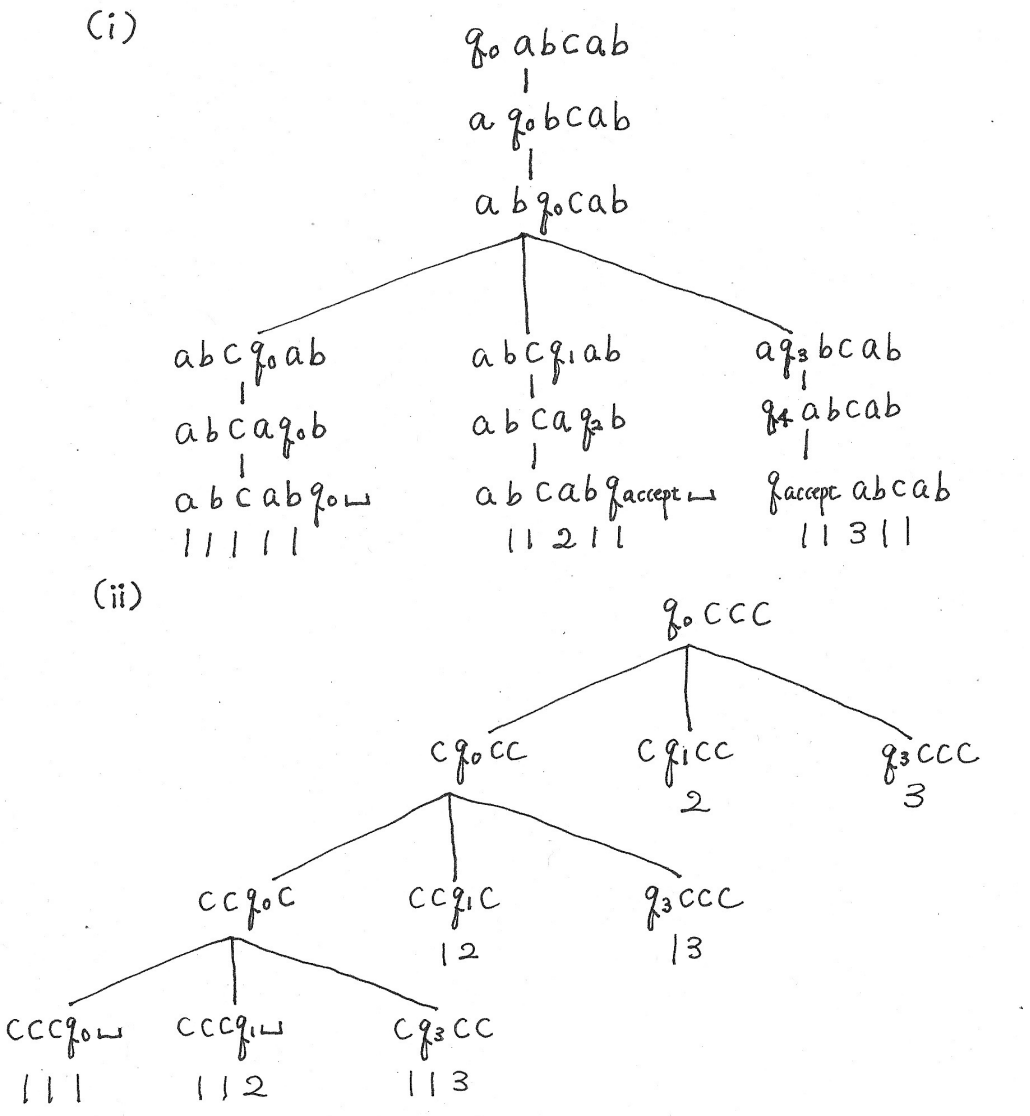


(ii)



In this NTM, $\delta(q_0, c) = \{(q_0, c, R), (q_1, c, R), (q_3, c, L)\}$. Sequentially number these three nondeterministic transitions by 1, 2, 3.

a. Give the access-path strings that encode the branches of the above computation trees.



b. Which of these strings are the certificates for "abcab", if any?

11211 and 11311 are the certificates as they encode accepting branches.

c. Which of these strings are the certificates for "ccc", if any?

None of them is a certificate as they do not encode accepting branches.

4. Show that the following problems are in NP. You don't have to show they are NP-complete.

a. LPATH in Problem 7.21 on page 324 (Problem 7.20 in the 2nd edition).

A proposed certificate is a path of G from a to b. The verifier determines if this path is simple and has the length at least k. This checking can be done in polynomial time in the number of vertices and edges of G.

b. DOUBLE-SAT in Problem 7.22 on page 324 (Problem 7.21 in the 2nd edition).

A proposed certificate is a pair of two distinct assignments for ϕ. The verifier determines if both the assignments satisfy ϕ. If so, it accepts ϕ; o.w., it rejects ϕ. This checking can be done in polynomial time in

the size of ϕ.

c. $\neq$SAT in Problem 7.26 on page 324 (Problem 7.24 in the 2nd edition).

A proposed certificate is an assignment for ϕ. The verifier determines if it satisfies ϕ without making the three literals true in any clause. This checking can be done in polynomial time in the size of ϕ.

d. MAX-CUT in Problem 7.27 on page 325 (Problem 7.25 in the 2nd edition).

Let V be the set of all vertices of the given graph G. A proposed certificate is a set of vertices $S \subseteq V$. The verifier checks all edges in G to determine if they have one endpoint in S and the other in V−S, and counts the total number of such edges. If the total number is $\geq k$, the verifier accepts S; o.w., the verifier rejects S. This checking can be done in polynomial time in the number of vertices and edges of G.

e. SET-SPLITTING in Problem 7.30 on page 326 (Problem 7.28 in the 2nd edition).

A proposed certificate is an assignment of red or blue to the elements of S. The verifier determines if every C_i has an element colored with red and an element colored with blue. If this is the case, the verifier accepts the color assignment; o.w., the verifier rejects it. This checking can be done in polynomial time in $|S|+|C_1|+\cdots+|C_k|$.

5. Prove each of the following:

These can be proved by NTMs or verifiers. The following uses verifiers.

a. If $A, B \in NP$, then $A \cup B \in NP$.

Let V_A, V_B be polynomial-time verifiers for A and B with $W_{V_A}(n) = O(n^{k_1})$ and $W_{V_B}(n) = O(n^{k_2})$. We define a polynomial-time verifier $V_{A \cup B}$ for $A \cup B$ using V_A and V_B. We need to show:

$$A \cup B = \{ w \mid \exists c \text{ s.t. } V_{A \cup B} \text{ accepts } \langle w, c \rangle \}.$$

definition of $V_{A \cup B}$

given $\langle w, c \rangle$:
1. run V_A on $\langle w, c \rangle$
2. run V_B on $\langle w, c \rangle$
3. **if** V_A or V_B accepts $\langle w, c \rangle$ **then** accept **else** reject

Suppose $w \in A \cup B$. Then $w \in A$ or $w \in B$, so $\exists c_A$ s.t. V_A accepts $\langle w, c_A \rangle$ or $\exists c_B$ s.t. V_B accepts $\langle w, c_B \rangle$. Note that c_A and c_B may be distinct. In the former case, $V_{A \cup B}$ accepts $\langle w, c_A \rangle$, and in the latter case, $V_{A \cup B}$ accepts $\langle w, c_B \rangle$. In either case, w is verified by $V_{A \cup B}$.

Conversely, suppose w is verified by $V_{A \cup B}$. Then $\exists c$ s.t. $V_{A \cup B}$ accepts $\langle w, c \rangle$. This means, by definition of $V_{A \cup B}$, V_A or V_B accepts $\langle w, c \rangle$. Hence w is verified by V_A or V_B, and so $w \in A$ or $w \in B$. Thus $w \in A \cup B$.

Clearly, $W_{V_{A \cup B}}(n)$ has an asymptotic order of $W_{V_A}(n) + W_{V_B}(n) = O(n^{k_1}) + O(n^{k_2})$, which is polynomially bounded.

b. If $A, B \in NP$, then $A \cdot B \in NP$, where $A \cdot B = \{ xy \mid x \in A \text{ and } y \in B \}$.

Let V_A, V_B be polynomial-time verifiers for A and B with $W_{V_A}(n) = O(n^{k_1})$ and $W_{V_B}(n) = O(n^{k_2})$. We define a polynomial-time verifier $V_{A \cdot B}$ for $A \cdot B$ using V_A and V_B. We need to show:

$$A \cdot B = \{ w \mid \exists c \text{ s.t. } V_{A \cdot B} \text{ accepts } \langle w, c \rangle \}.$$

$V_{A \cdot B}$ uses a 3-component certificate $c = \langle c_{sp}, c_A, c_B \rangle$. The string c_{sp} encodes a split position of the input w. If $c_{sp} = i$, $0 \le i \le n$, $w = w_1 \cdots w_n$ will be split into $x = w_1 \cdots w_i$ and $y = w_{i+1} \cdots w_n$.

definition of $V_{A \cdot B}$

given $\langle w, c = \langle c_{sp}, c_A, c_B \rangle \rangle$:

1. extract 3 components c_{sp}, c_A, c_B from c
2. split w into xy using the split point encoded in c_{sp}
3. run V_A on $\langle x, c_A \rangle$
4. run V_B on $\langle y, c_B \rangle$
5. **if** V_A accepts $\langle x, c_A \rangle$ and V_B accepts $\langle y, c_B \rangle$ **then** accept **else** reject

Suppose $w \in A \cdot B$. By definition of $A \cdot B$, $w = xy$, $x \in A$, and $y \in B$. So $\exists c_A$ s.t. V_A accepts $\langle x, c_A \rangle$ and $\exists c_B$ s.t. V_B accepts $\langle y, c_B \rangle$. Let c_{sp} encode the split position of $w = xy$, and let $c = \langle c_{sp}, c_A, c_B \rangle \rangle$. Then $V_{A \cdot B}$ will accept $\langle w, c = \langle c_{sp}, c_A, c_B \rangle \rangle$.

Conversely, suppose $V_{A \cdot B}$ accepts $\langle w, c = \langle c_{sp}, c_A, c_B \rangle \rangle$. By definition of $V_{A \cdot B}$, this means that w is split into xy using the split position encoded in c_{sp}, V_A accepts $\langle x, c_A \rangle$, and V_B accepts $\langle y, c_B \rangle$. Hence, $x \in A$ and $y \in B$, and thus $w = xy \in A \cdot B$.

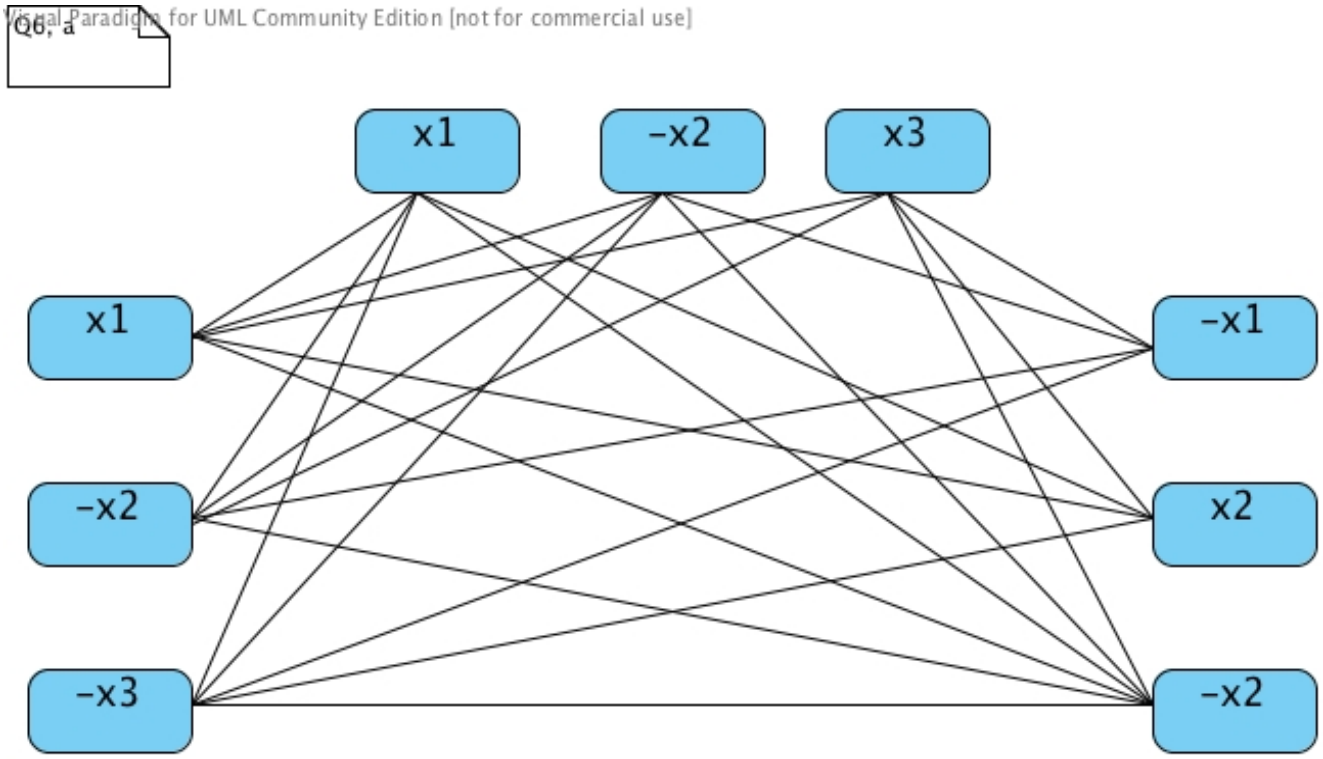
A split point ranges from 0 to n and can be encoded by a string of length at most n, so $|c_{sp}| \le n$. Also, $|c_A| \le W_{V_A}(n) = O(n^{k_1})$ and $|c_B| \le W_{V_B}(n) = O(n^{k_2})$. Hence $|c| = O(n + O(n^{k_1}) + O(n^{k_2}))$, which is polynomially bounded. Thus, Step 1 can be done in polynomial time of $n = |w|$. Step 2 can be done in polynomial time of $n = |w|$. Since $|x| \le n$ and $|y| \le n$, Step 3 takes at most $W_{V_A}(n) = O(n^{k_1})$ and Step 4 takes at most $W_{V_B}(n) = O(n^{k_2})$. Thus, the total runtime is polynomially bounded.

6. Consider the polynomial-time reduction used to prove 3SAT $\le_p$ CLIQUE. Let:

$$\phi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_2)$$

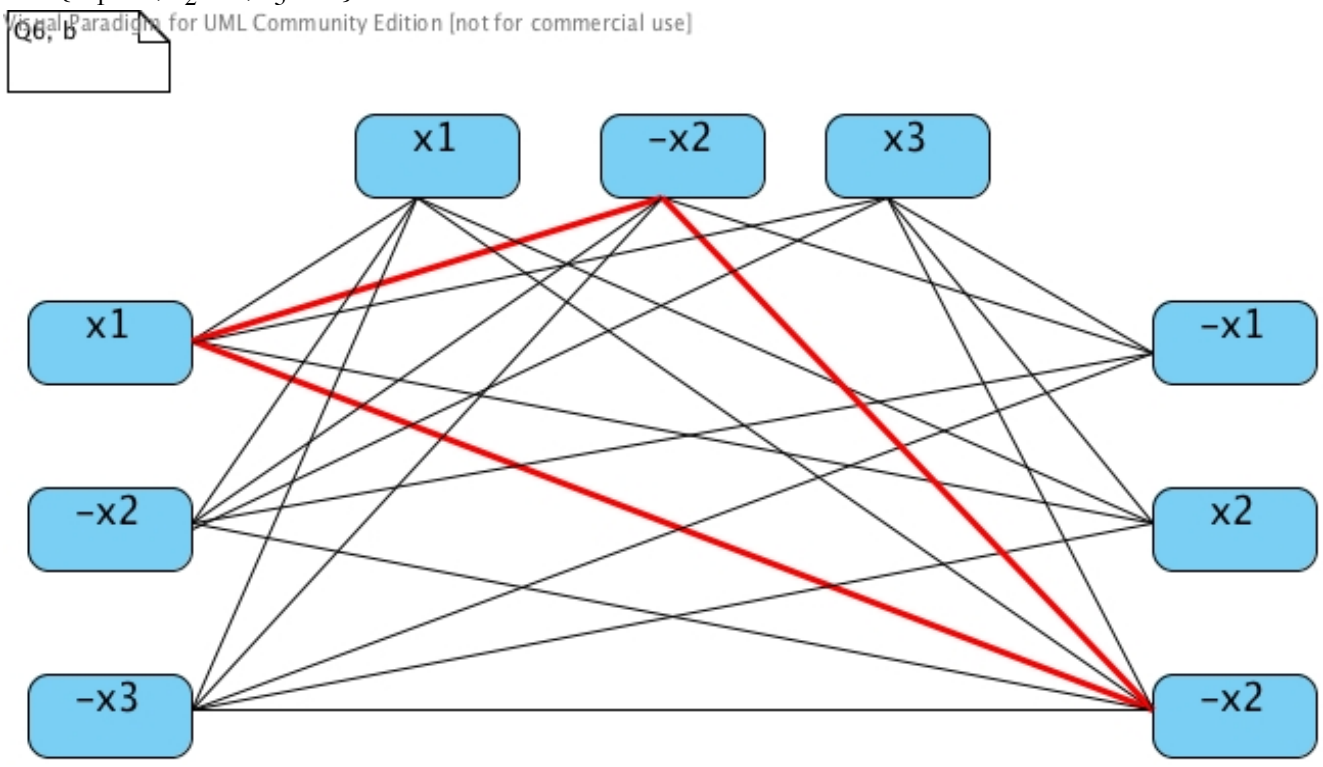
Here, $\neg x_i$ is the negation of x_i.

a. Give (G_ϕ, k) constructed from ϕ by the reduction.



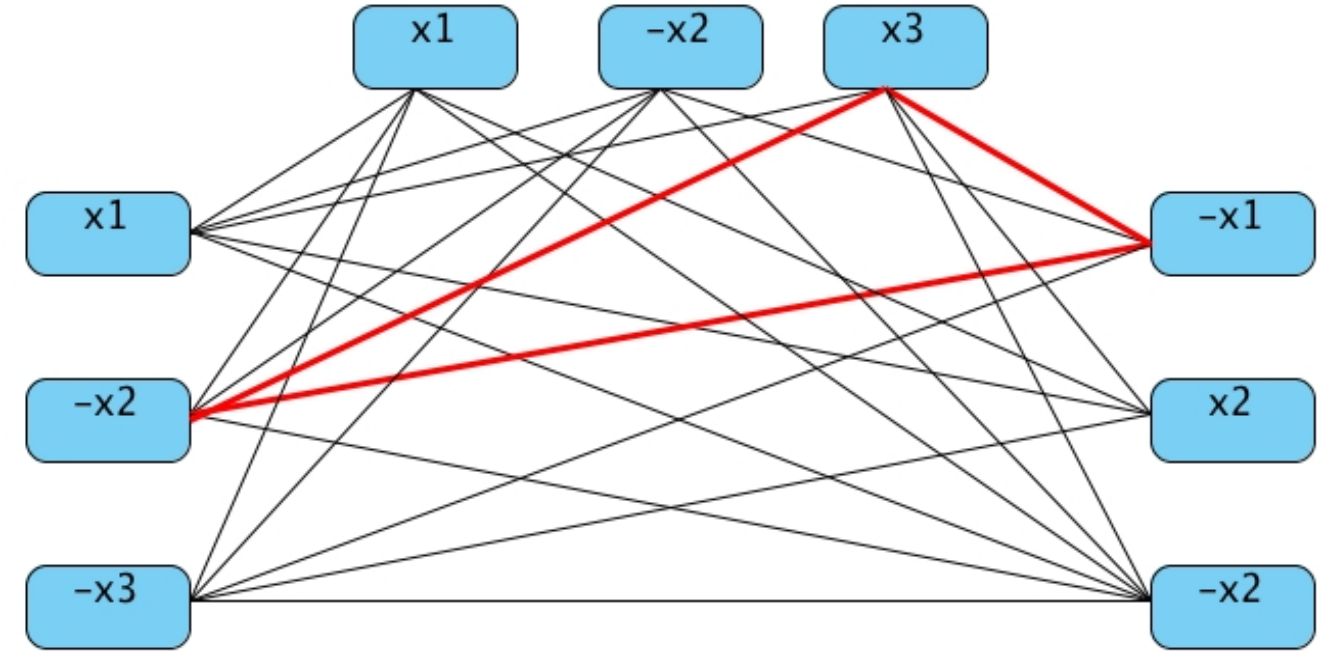
b. Give one satisfying assignment A for ϕ, and a corresponding k-clique in G_ϕ produced by the proof.

A = { $x_1 = 1, x_2 = 0, x_3 = 0$ }.



c. Give one k-clique in G_ϕ that is distinct from the one you gave in (b), and give a corresponding assignment for ϕ produced by the proof.

Q6, c



A = { $x_1 = 0, x_2 = 0, x_3 = 1$ }.