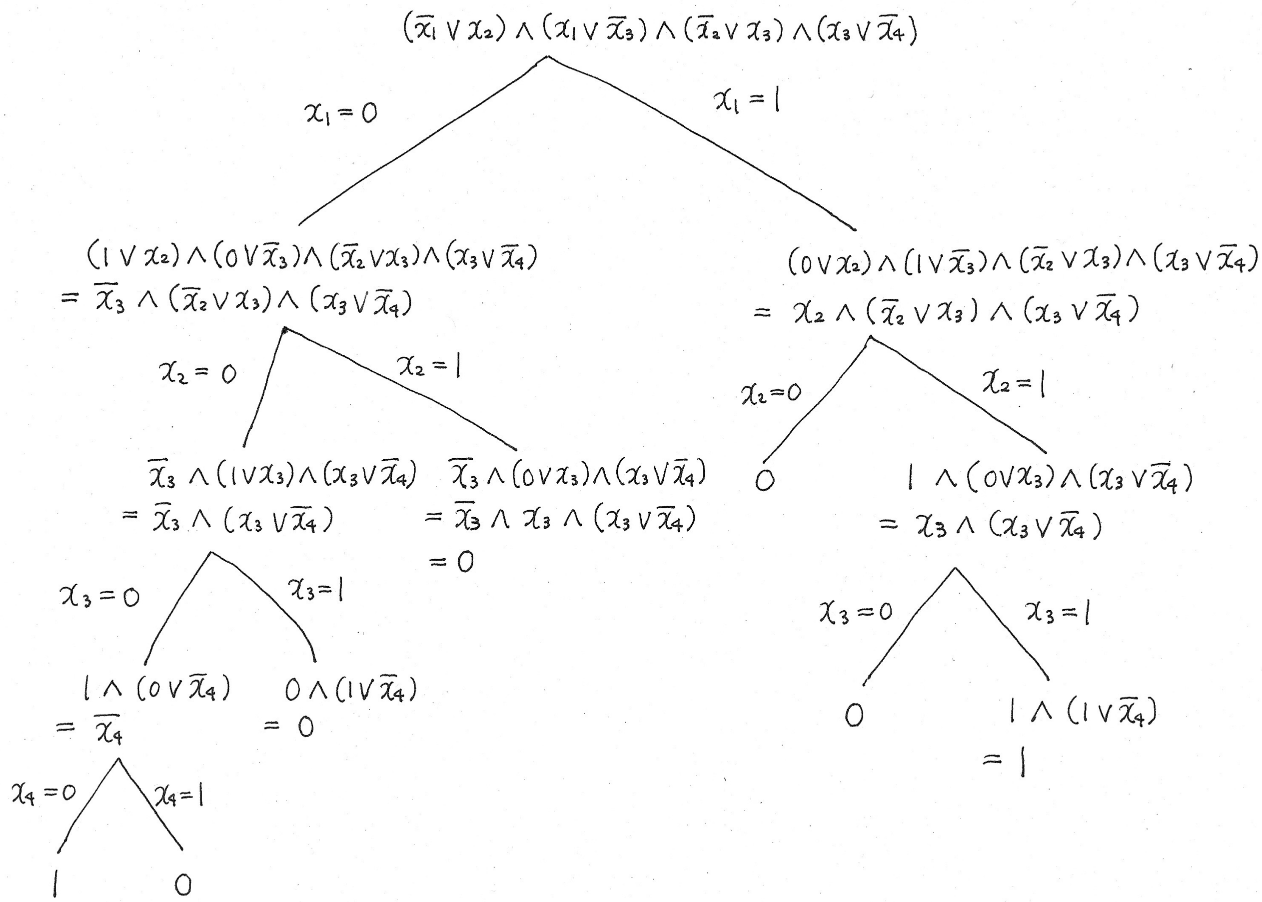
CS 722 Fall 2016
Homework Assignment #4
Solutions

1. Let

$$\psi(x_1, x_2, x_3, x_4) = (\neg x_1 \lor x_2) \land (x_1 \lor \neg x_3) \land (\neg x_2 \lor x_3) \land (x_3 \lor \neg x_4).$$

a. Give an evaluation tree for ψ by assigning $x_i = 0$ and 1, $1 \le i \le 4$. You may terminate a branch as soon as its value is known.



b. For each of the following formulas, show its truth value and which player (E or A) has a winning strategy. Justify your answers.

i. $\exists x_1 \forall x_2 \exists x_3 \forall x_4 \psi(x_1, x_2, x_3, x_4)$

A has a winning strategy.
If E selects $x_1=0$, A selects $x_2=1$, and ψ evaluates to 0 regardless of the values of x_3, x_4.
If E selects $x_1=1$, A selects $x_2=0$, and ψ evaluates to 0 regardless of the values of x_3, x_4.
Hence the formula is false.

ii. $\forall x_1 \exists x_2 \forall x_3 \exists x_4 \psi(x_1, x_2, x_3, x_4)$

A has a winning strategy.
A selects $x_1=0$.
If E selects $x_2=0$, A selects $x_3=1$, and ψ evaluates to 0 regardless of the value of x_4.
If E selects $x_2=1$, ψ evaluates to 0 regardless of the values of x_3, x_4.
Hence the formula is false.

iii. $\forall x_1 \exists x_2 \exists x_3 \forall x_4 \psi(x_1, x_2, x_3, x_4)$

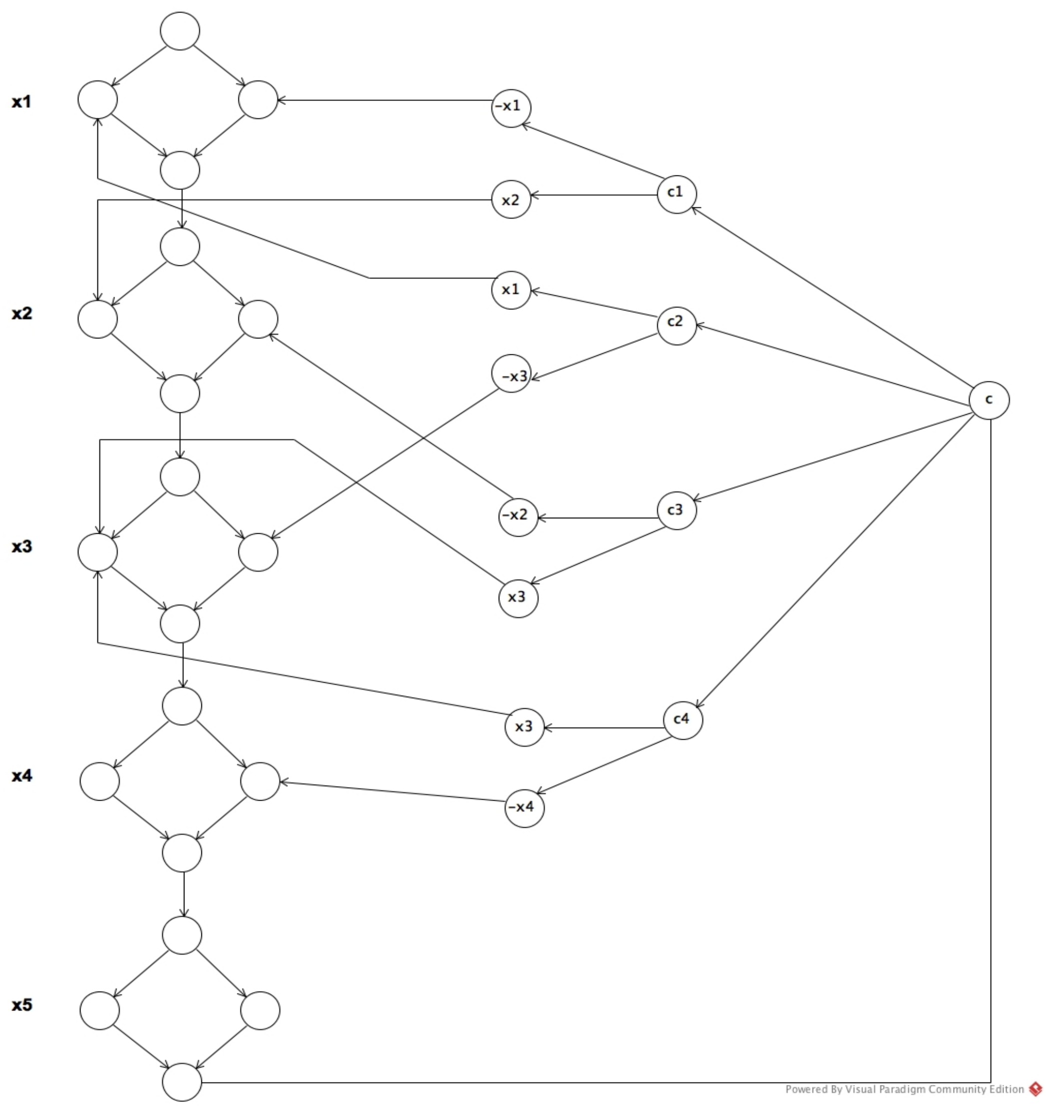
A has a winning strategy.

A selects x_1=0.
E must then select x_2=0, x_3=0, for otherwise ψ evaluates to 0 and E loses.
Then A selects x_4=1, and ψ evaluates to 0.
Hence the formula is false.

2. Study GENERALIZED GEOGRAPHY in §8.3 of the book, especially the polynomial-time reduction from FORMULA-GAME to GENERALIZED GEOGRAPHY. Recall that FORMULA-GAME is a 2-player game interpretation of TQBF. For each of the formulas (i) and (ii) in Question 1, (b), answer the following questions:

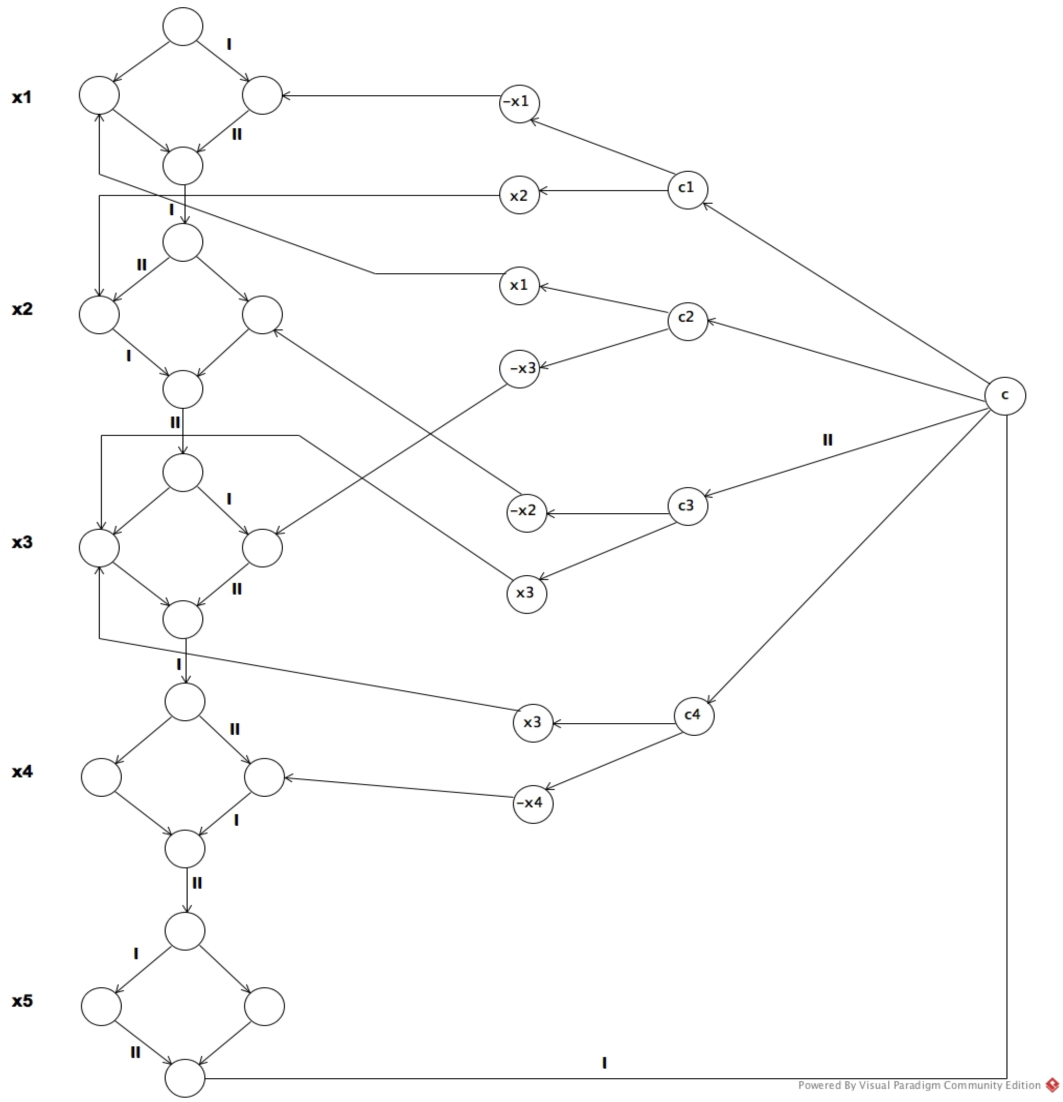
   a. Give the directed graph generated from the formula by the polynomial-time reduction.
   b. Give an assignment for x_i, $1 \le i \le 4$, representing a winning strategy for the formula, and show the corresponding path selected by players I and II in the graph. Mark the edges of the path by "I" or "II" according as they are selected by player I or II, respectively.

   i.
      a. The dummy quantifier $\exists x_5$ is added to get: $\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \psi(x_1, x_2, x_3, x_4)$.



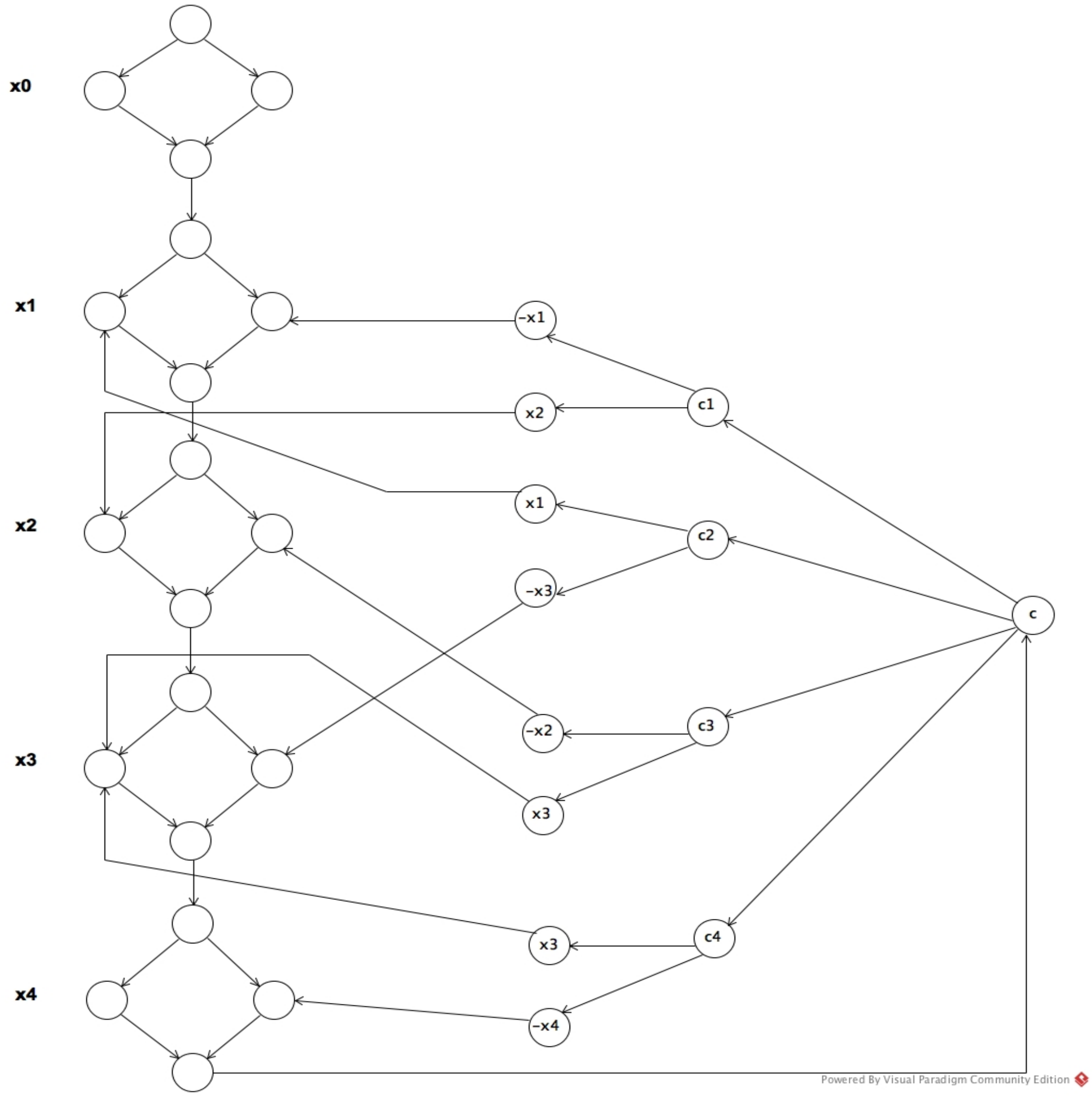
      b. An example assignment where A wins: $x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 0$.
      A's (II's) winning strategy: II goes to false clause C_3. Whether I goes to $\neg x_2$ or x_3, II wins.

Selection for x_5 is immaterial as it is a dummy variable.

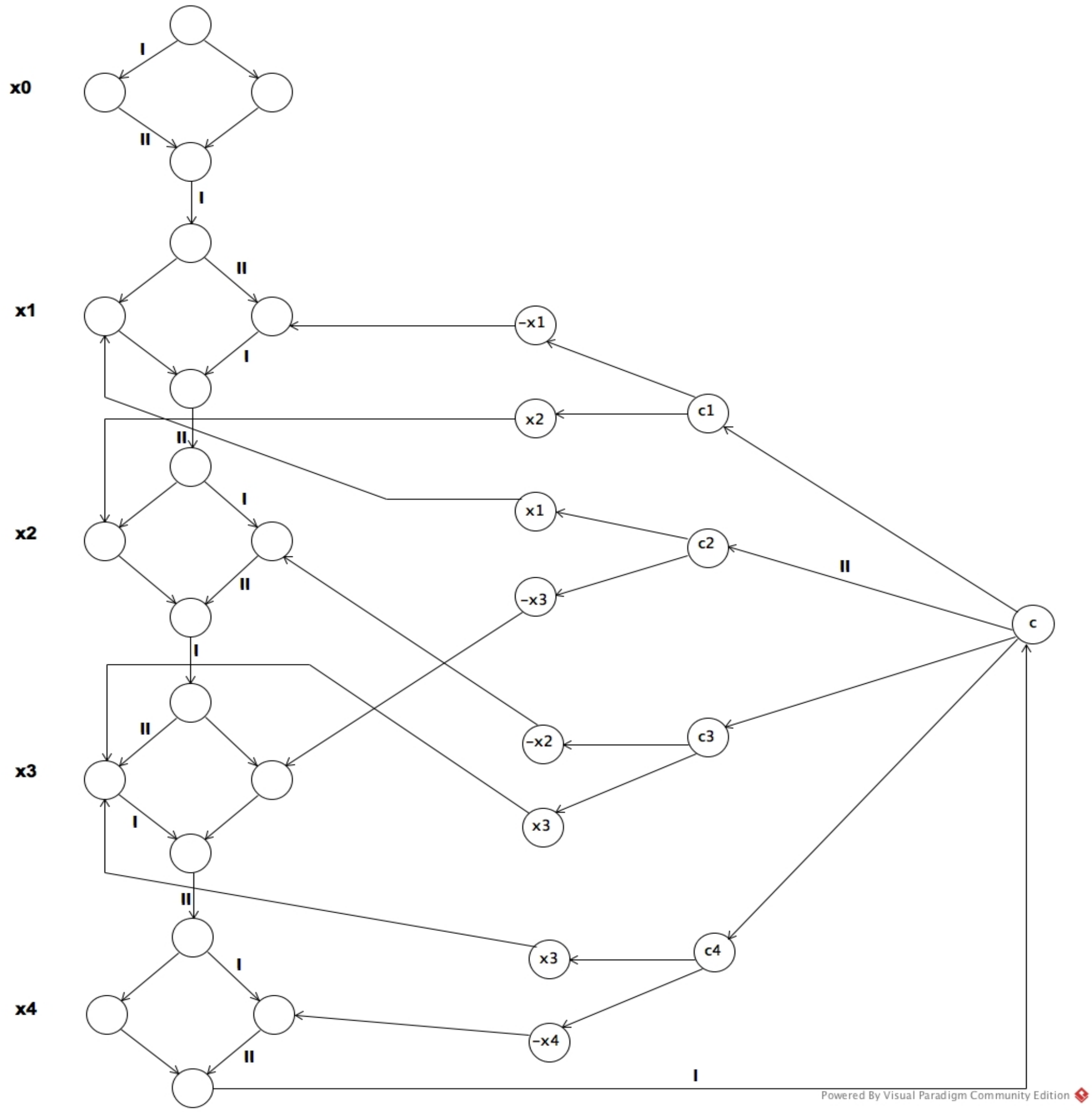


ii.

a. The dummy quantifier $\exists x_0$ is added to get: $\exists x_0 \forall x_1 \exists x_2 \forall x_3 \exists x_4 \psi(x_1, x_2, x_3, x_4)$.



b. An example assignment where A wins: $x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 0$.
A's (II's) winning strategy: II goes to false clause C_2. Whether I goes to x_1 or $\neg x_3$, II wins.
Selection for x_0 is immaterial as it is a dummy variable.



3. The NFA problem is to decide if a nondeterministic finite automaton (NFA) accepts an input string w. (If necessary, review NFAs in §1.2.) Formally, the language L_{NFA} is defined as follows:

   L_{NFA} = { <M, w> | M is an NFA that accepts input string w },

   where <M, w> is a string encoding the pair of M and w. Prove that L_{NFA} is NL-complete by answering the following questions:

   a. Show that $L_{NFA} \in$ NL by describing a nondeterministic log-space algorithm.

      The NTMR, N, implementing the nondeterministic algorithm has <M, w> on the read-only input tape. Let $Q = \{q_0, \ldots, q_{m-1}\}$ be the set of control states of M where q_0 is the start state and $m = |Q|$. Encode $q_0, \ldots, q_{m-1}$ in binary notation of $0, \ldots, m-1$. Then each q_i is encoded in $O(\log_2 m)$ bits. Let $w = a_1 \cdots a_n$.

      N nondeterministically simulates state transitions of M by keeping the binary code of the current state on the work tape until all symbols in w have been read. The only potential problem is that this process may fail to terminate if M's transition diagram has a loop of transitions all labeled by ε and the simulation happens to follow this loop indefinitely. Since a loop of this kind periodically repeats the same transitions, however, it is sufficient to simulate *simple* paths of transitions all labeled by ε. Hence it is sufficient to simulate the fictitious input string of the form $e_0a_1e_1a_2e_2 \cdots e_{n-1}a_ne_n$ where each e_i is a string of ε of length at most $m-1$. The length of this fictitious input string is at most $n+(n+1)(m-1)$. This leads to the following algorithm.

```
compute n+(n+1)(m-1) in binary and set p = n+(n+1)(m-1);
i = 0; // i holds a binary number at most p
q = q0; // q holds a binary number representing a state
while ( i < p )
{
```

```
        nondeterministically select q' = δ(q, a) where a is the current input symbol (possibly ε)
        and update q to q';
        if ( q is an accept state )
              output accept and terminate the algorithm;
        i = i + 1;
}
output reject;
```

The values of the variables p, i, q are binary numbers of size $O(\log_2[(n+(n+1)(m-1)])$, $O(\log_2[n+(n+1)(m-1)])$, $O(\log_2 m)$. Thus the algorithm runs in log space.

b. Prove that PATH $\leq_L$ L_{NFA}. Hint: Reduce an input <G, s, t> for PATH to an NFA whose transition diagram reflects the graph G. You need to prove your reduction can be run by a log-space transducer and the equivalence condition holds.

Let G = (V, E), s, t be an instance of PATH problem. This is reduced to <M, w> where the transition graph of M is isomorphic to G, all transitions are labeled by ε, and the start and accept states are s and t, respectively. Also set w = ε. Formally, G, s, t is reduced to $M = (Q, \Sigma, \delta, q_0, F)$ where $Q = V$, $\Sigma = \{\}$, $q_0 = s$, $F = \{t\}$, and $v \in \delta(u, \varepsilon)$ for each edge $(u, v) \in E$, i.e., $\delta(u, \varepsilon) = \{ v \mid (u, v) \in E \}$.

The log-space transducer simply copies G = (V, E), s, t from the input tape to the output tape to produce N without using the work tape: Copy V and set it to be Q, write $\Sigma = \{\}$, copy s and t and set them to be q_0 and F, and copy all edges (u, v) in E and set $v \in \delta(u, \varepsilon)$. And write w = ε.

The proof of the equivalence condition. Suppose that there is a directed path from s to t in G. By definition of M, there is a transition sequence from q_0 to the accept state t where all the transitions are ε-transitions. So M accepts ε. Conversely, suppose that M accepts ε. Then there is a transition sequence of M from q_0 to the accept state t where all the transitions are ε-transitions. By definition of the reduction, this transition sequence corresponds to a path from s to t in G.

4. The complexity class Co-NP is defined as follows:

Co-NP = $\{ -L \mid L \in NP \}$

where –L is the complement set of L. Informally, –L is obtained from L by switching accept and reject. For example, –SAT, call it NON-SAT, accepts ϕ if it is *not* satisfiable and rejects if it is satisfiable. Like PSPACE-Complete, nondeterminism does not seem to help in deciding complements of NP-complete problems. No polynomial-time NTM or verifier is known for the complement of any NP-complete problem – what could be a polynomial-size certificate to show that ϕ is not satisfiable?

a. Prove: NON-SAT $\leq_p$ TQBF.

Given an instance $\phi(x_1, \ldots, x_n)$ of NON-SAT, the reduction produces $\forall x_1 \cdots \forall x_n \neg\phi(x_1, \ldots, x_n)$. Clearly this runs in polynomial time of the size of ϕ. The proof of the equivalence condition is as follows:

$\phi(x_1, \ldots, x_n)$ is unsatisfiable ⇔
for all possible 0/1 values of $x_1, \ldots, x_n$, $\phi(x_1, \ldots, x_n)$ is false ⇔
for all possible 0/1 values of $x_1, \ldots, x_n$, $\neg\phi(x_1, \ldots, x_n)$ is true ⇔
$\forall x_1 \cdots \forall x_n \neg\phi(x_1, \ldots, x_n)$ is true.

b. Prove: For $\forall L \in$ Co-NP, $L \leq_p$ NON-SAT.

First we prove

Theorem For all L_1, L_2, $L_1 \leq_p L_2$ iff $-L_1 \leq_p -L_2$.

Proof: Any polynomial-time reduction from L_1 to L_2 is a polynomial-time reduction from $-L_1$ to $-L_2$. Formally,

$L_1 \leq_p L_2$ ⇔
∃polynomial-time reduction f s.t. for $\forall w \in \Sigma^*$, $w \in L_1$ iff $f(w) \in L_2$ ⇔
∃polynomial-time reduction f s.t. for $\forall w \in \Sigma^*$, $w \notin L_1$ iff $f(w) \notin L_2$ ⇔
∃polynomial-time reduction f s.t. for $\forall w \in \Sigma^*$, $w \in -L_1$ iff $f(w) \in -L_2$ ⇔
$-L_1 \leq_p -L_2$

Let L ∈ Co-NP. Then –L ∈ NP. Since SAT ∈ NP-Complete, $-L \leq_p$ SAT. By the above Theorem, $L \leq_p$ –SAT.

c. Prove: For $\forall L_1 \in$ Co-NP, $\forall L_2 \in$ PSPACE-Complete, $L_1 \leq_p L_2$.

Let $L_1 \in$ Co-NP, $L_2 \in$ PSPACE-Complete. By a) and b), $L_1 \leq_p$ NON-SAT $\leq_p$ TQBF $\leq_p L_2$.

Another way to prove this is by proving Co-NP $\subseteq$ PSPACE (left as exercise).