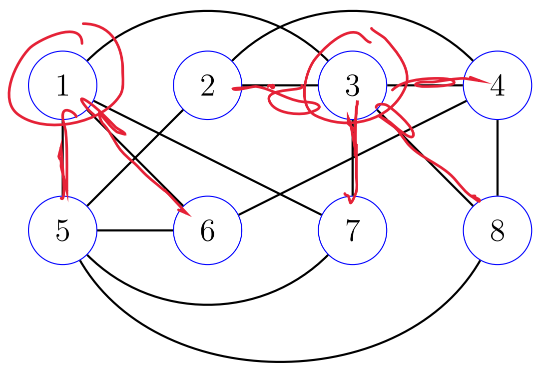
CECS 329, Learning Outcome Assessment 3, February 14th, Spring 2024, Dr. Ebert

Problems

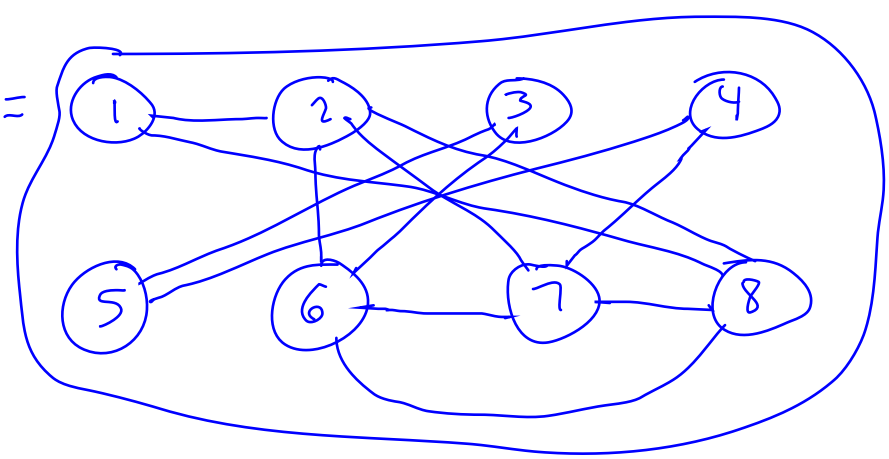
LO1. Answer the following.

(a) Provide the definition of what it means to be a mapping reduction fromm problem A to problem B. Hint: do *not* assume that A and B are decision problems.

(b) The simple graph $G = (V, E)$ shown below is an instance of the `Maximum Independent Set (MIS)` optimizationn problem. Draw $f(G)$, where f is the mapping reduction from `MIS` to `Maximum Clique` provided in lecture.



$f(G) = \bar{G} =$



(c) Verify that f is valid for input G in the sense that both G and $f(G)$ have the same solution. Hint: for both problems please assume that a "solution" is represented by a subset of vertices.

f is valid for input G since $\{2,6,7,8\}$ is a maximum independent set for G, while it is also a maximum clique for $f(G)$.

LO2. An instance of the `Dominating Set` decision problem is a pair (G, k), where $G = (V, E)$ is a simple graph, $k \geq 0$ is an integer, and the problem is to decide if G has a **dominating set** of size k, i.e. a set $D \subseteq V$ for which every vertex $v \in V$ is adjacent to at least one vertex in D. For example, for the graph G shown above, $(G, k = 2)$ is a positive instance of `Dominating Set` since $D = \{1, 3\}$ is a dominating set for G.

(a) For a given instance (G, k) of `Dominating Set` describe a certificate in relation to (G, k).

A certificate is a subset D of k vertices.

(b) Provide a semi-formal verifier algorithm that takes as input i) an instance (G, k), ii) a certificate for (G, k) as defined in part a, and decides if the certificate is valid for G.

For each $u \in V$,
  If $u \in D$, then continue.
  found = 0. // Find a $v \in D$ for which $(u,v) \in E$
  For each $v \in D$
    If $(u,v) \in E$, then
      found = 1.
      break.
  If found = 0, return 0.
Return 1. // every $u \in V$ is either in D or is adjacent to a vertex in D

(c) Provide size parameters that may be used to measure the size of an instance of `Dominating Set`.

$m = |E|$ $n = |V|$

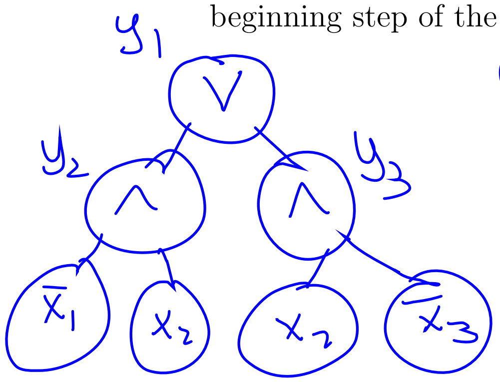
(d) Use the size parameters from part c to describe the running time of your verifier from part b. Defend your answer in relation to the algorithm you provided for the verifier.

We must make $O(n \cdot n) = O(n^2)$ iterations over the nested For-loops. In each iteration, checking $(u,v) \in E$ can be done using a hash table in $O(1)$ steps. Building the table requires $O(m)$ steps,

$\therefore$ Verifier running time $= O(m + n^2) = O(n^2)$ is quadratic in n.

LO3. Recall the polynomial-time mapping reduction from SAT to 3SAT described in lecture.

(a) Given the SAT instance $F(x_1, x_2, x_3) = (\overline{x}_1 \wedge x_2) \vee (x_2 \wedge \overline{x}_3)$, draw its parse tree and provide the associated Boolean formula G that is satisfiability equivalent to F and serves as the beginning step of the reduction. Hint: formula G introduces y-variables.



$$G = y_1 \wedge (y_1 \leftrightarrow (y_2 \vee y_3)) \wedge (y_2 \leftrightarrow (\overline{x}_1 \wedge x_2)) \wedge (y_3 \leftrightarrow (x_2 \wedge \overline{x}_3))$$

(b) Rewrite formula G by making use of the logical identity

$$(P \leftrightarrow Q) \Leftrightarrow [(P \rightarrow Q) \wedge (Q \rightarrow P)]$$

$$y_1 \wedge (y_1 \rightarrow (y_2 \vee y_3)) \wedge ((y_2 \vee y_3) \rightarrow y_1) \wedge (y_2 \rightarrow (\overline{x}_1 \wedge x_2)) \wedge ((\overline{x}_1 \wedge x_2) \rightarrow y_2) \wedge (y_3 \rightarrow (x_2 \wedge \overline{x}_3)) \wedge ((x_2 \wedge \overline{x}_3) \rightarrow y_3)$$

(c) Rewrite the formula from part b by making use of the logical identity

$$(P \rightarrow Q) \Leftrightarrow (\overline{P} \vee Q).$$

$$y_1 \wedge (\overline{y}_1 \vee y_2 \vee y_3) \wedge (\overline{(y_2 \vee y_3)} \vee y_1) \wedge (\overline{y}_2 \vee (\overline{x}_1 \wedge x_2)) \wedge (\overline{(\overline{x}_1 \wedge x_2)} \vee y_2) \wedge (\overline{y}_3 \vee (x_2 \wedge \overline{x}_3)) \wedge (\overline{(x_2 \wedge \overline{x}_3)} \vee y_3)$$

(d) Rewrite the formula from part c by performing one or more applications of De Morgan's rule.

$$y_1 \wedge (\overline{y}_1 \vee y_2 \vee y_3) \wedge ((\overline{y}_2 \wedge \overline{y}_3) \vee y_1) \wedge (\overline{y}_2 \vee (\overline{x}_1 \wedge x_2)) \wedge (x_1 \vee \overline{x}_2 \vee y_2) \wedge (\overline{y}_3 \vee (x_2 \wedge \overline{x}_3)) \wedge (\overline{x}_2 \vee x_3 \vee y_3)$$

(e) Rewrite the formula from part d by performing one or more applications of the distributive rule in order to obtain an AND of OR's. Then convert the AND of OR's to an AND of ternary (i.e. three) OR's and use 3SAT notation to complete the reduction.

$\{(y_1, y_1, y_1), (\overline{y}_1, y_2, y_3), (\overline{y}_2, y_1, y_1), (\overline{y}_3, y_1, y_1),$
$(\overline{y}_2, \overline{x}_1, \overline{x}_1), (\overline{y}_2, x_2, x_2), (x_1, \overline{x}_2, y_2),$
$(\overline{y}_3, x_2, x_2), (\overline{y}_3, \overline{x}_3, \overline{x}_3), (\overline{x}_2, x_3, y_3)\}$

$f(F)$ has 10 clauses and $3+3=6$ variables.

Notice how F is satisfiable via $\alpha = (x_1=0, x_2=1, x_3=1)$ and, by following the computation within the parse tree, $F(f)$ is satisfiable via

$\alpha' = (x_1=0, x_2=1, x_3=1, y_1=1, y_2=1, y_3=0)$

On the other hand, if $\alpha = (x_1=0, x_2=0, x_3=0)$ then α does not satisfy F and it forces the assignment $\alpha' = (x_1=0, x_2=0, x_3=0, y_1=0, y_2=0, y_3=0)$ which does not satisfy G since G asserts that $y_1=1$.