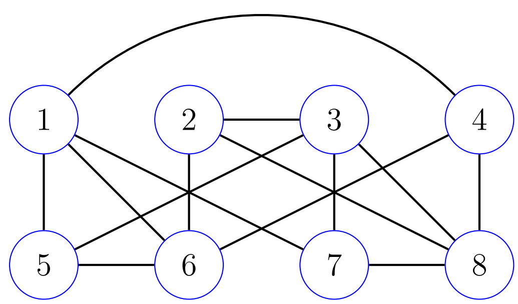
CECS 329, Learning Outcome Makeup Problems, December 7th, Fall 2023, Dr. Ebert

Problems

LO1. Answer/solve the following.

(a) Provide the definition of what it means to be a mapping reduction fromm problem A to problem B.

(b) Suppose $(G, k = 3)$ is an instance of the `Vertex Cover` decision problem, where G is drawn below. Draw $f(G, k)$, where f is the mapping reduction from `Vertex Cover` to the `Half Vertex Cover` decision problem.



(c) Verify that f is valid for input (G, k) in the sense that both (G, k) and $f(G)$ are either both positive instances or both negative instances of their respective problems. Defend your answer. Hint: it takes two vertices to cover each edge of a triangle in a graph.

LO2. Recall that an instance of the `Vertex Cover` 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 **vertex cover** of size k, i.e. a set $C \subseteq V$ for which every edge $e \in E$ is incident with at least one vertex in C.

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

(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.

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

(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.

LO3. An instance $\mathcal{C}$ of `3SAT` consists of clauses $c_1 = (x_1, \overline{x}_2, x_3)$, $c_2 = (\overline{x}_1, x_2, x_3)$, $c_3 = (\overline{x}_1, x_2, \overline{x}_3)$, and $c_4 = (\overline{x}_1, \overline{x}_2, x_3)$. Answer the following questions about the mapping reduction $f(\mathcal{C}) = (G, k)$ provided in lecture from `3SAT` to `Clique` and applied to instance $\mathcal{C}$.

(a) How many vertices and edges does G have? Explain and show work. Hint: there are six different vertex-group pairs.

(b) What is the value of k?

(c) Given that $\alpha = (x_1 = 0, x_2 = 0, x_3 = 1)$ satisfies $\mathcal{C}$, provide a clique set C for G that certifies (G, k) is a positive instance of `Clique`. For each clique member, indicate the vertex group to which it belongs.

LO4. Answer the following. Note: scoring 14 or more points counts for passing.

(a) Write either P, NP, or co-NP next to each of the following decision problems in terms of which one best characterizes the complexity of the problem. Note: there is only one best answer for each. Two points each.

i. An instance of `Reachability` is a simple graph $G = (V, E)$ and two vertices $u, v \in V$, and the problem is to decide if v is reachable from u, meaning there is a path from u to v.

ii. An instance of the `3-Dimensional Matching (3DM)` decision problem takes as input three sets A, B, and C, each having size n, along with a set S of triples of the form (a, b, c) where $a \in A$, $b \in B$, and $c \in C$. We assume that $|S| = m \geq n$. The problem is to decide if there exists a subset $T \subseteq S$ of n triples for which each member from $A \cup B \cup C$ belongs to exactly one of the triples.

iii. An instance of `UNSAT` is a Boolean formula $F(x_1, \ldots, x_n)$, and the problem is to decide if F is unsatisfiable (i.e. admits no satisfying assignments).

iv. An instance of `Bounded Cliques`] is a simple graph $G = (V, E)$ and an integer $k \geq 0$, an the problem is to decide if G has no k-cliques?

(b) In the complexity lecture, to show that `TSP` is `NP`-complete, we provided a direct polynomial-time mapping reduction from ____ to `TSP`. (8 points)

i. `Directed Hamilton Path`
ii. `3SAT`
iii. `Hamilton Cycle`
iv. `Clique`

(c) Historically speaking, what was the first problem ever to be shown `NP`-complete? (9 points)

i. `Subset Sum`
ii. `SAT`
iii. `Hamilton Cycle`
iv. `Clique`

LO5. Provide the instructions for a URM program that computes the function $f(x) = 3x + 2$. For each register that is used by your program, described its purpose in one or more sentences.

LO6. Solve the following problems.

(a) Compute the Gödel number for program $P = T(5, 2), Z(7), S(6), J(1, 2, 3)$. Write your answer as a sum of powers of two minus 1 (see part b for an example). Show all work.

(b) Provide the URM program P whose Gödel number equals

$$2^{21} + 2^{30} + 2^{98} + 2^{112} - 1.$$

Show all work.

LO7. Answer/solve the following.

(a) When universal program P_U simulates the computation $P_x(y)$, how is y used by P_U as part of the simulation?

(b) A universal program P_U is simulating a program that has 93 instructions and whose Gödel number is

$$x = 2^{31} + 2^{32} + 2^{44} + 2^{68} + 2^{256} + \cdots + 2^{c_{93}} - 1.$$

If the current configuration of the computation of P_x on some input has encoding

$$\sigma = 2^5 + 2^9 + 2^{11} + 2^{18} + 2^{21} - 1,$$

then provide the next configuration of the computation *and* its encoding.

LO8. Answer the following.

(a) Define what it means to be a positive instance of decision problem `Self Accept`. Hint: "Gödel number x is a positive instance of `Self Accept` iff ____."

(b) The goal is to show that `Self Accept` is undecidable. We assume it is decidable by assuming that its characteristic function

$$f(x) = \begin{cases} 1 & \text{if } x \text{ is a positive instance of } \texttt{Self Accept} \\ 0 & \text{if } x \text{ is a negative instance of } \texttt{Self Accept} \end{cases}$$

is total computable. Provide the definition for how to compute the "antagonist" function $g(x)$ based on the value of $f(x)$.

(c) By writing the values of $g(0), g(1), \ldots$ in the appropriate table cells (next to the number in that cell), verify that function g is different from each computable function ϕ_i, $i = 0, 1, \ldots$, Why does this create a contradiction?

index\input x	0	1	2	$\cdots$	i	$\cdots$	**self accepting?**
$\phi_0(x)$	$\uparrow$	12	7	$\cdots$	$\uparrow$	$\cdots$	no
$\phi_1(x)$	8	87	36	$\cdots$	96	$\cdots$	yes
$\phi_2(x)$	7	5	0	$\cdots$	$\uparrow$	$\cdots$	yes
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$\phi_i(x)$	0	32	65	$\cdots$	$\uparrow$	$\cdots$	no
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$

LO9. An instance of the decision problem `Prime Range` is a Gödel number x, and the problem is to decide if function ϕ_x has a range that consists only of prime numbers, i.e. each value that it outputs is a member of the set $\{2, 3, 5, 7, 11, 13, \ldots\}$. Note: it is OK if the range is a proper subset of the set of prime numbers. Consider the function

$$g(x) = \begin{cases} 1 & \text{if } \phi_x \text{ has a prime range} \\ 0 & \text{otherwise} \end{cases}$$

(a) Evaluate $g(x)$ for each of the following Gödel number's x. Note: 2 out of 3 correct is considered passing.

i. $x = e_1$, where e_1 is the Gödel number of the program that computes the constant function $\phi_{e_1}(y) = 29$.

ii. $x = e_2$, where e_2 is the Gödel number of the program that computes $\phi_{e_2}(y) = y^2$.

iii. $x = e_3$, where e_3 is the Gödel number of the program that computes $g(x)$.

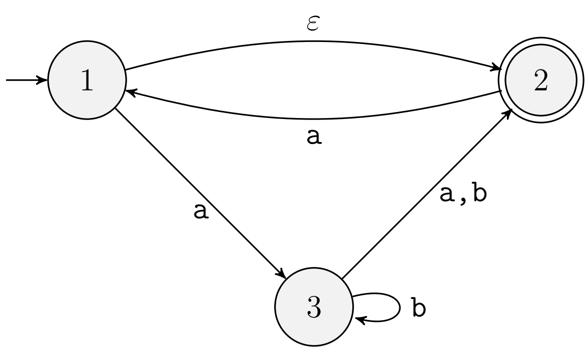
(b) Prove that $g(x)$ is not URM computable. In other words, there is no URM program that, on input x, always halts and either outputs 1 or 0, depending on whether or not ϕ_x has a range consisting of prime numbers. Do this by writing a program P that uses g and makes use of the `self` programming concept. Then show how P creates a contradiction.

LO10. Solve the following.

(a) Provide the state diagram for a DFA M that accepts a binary word w iff has an odd number of 0's and ends with a 1.

(b) Show the computation of M on inputs i) $w_1 = 10110101$ and ii) $w_2 = 01001101$.

LO11. Do the following for the NFA N whose state diagram is shown below.



(a) Provide a table that represents N's δ transition function.

(b) Use the table from part a to convert N to an equivalent DFA M using the method of subset states. Draw M's state diagram.

(c) Show the computation of M on input `abbab`.