CECS 329, Exam 2 Fall 2023, Dr. Ebert

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION ALLOWED. Submit solutions to **AT MOST 6 PROBLEMS**. Please use **BOTH SIDES** of each answer sheet to save paper. Make sure your name and Class ID are on each answer sheet.

Problems (25 Points Each)

1. The base-3 representation of a natural number x is a sequence $\alpha_k\alpha_{k-1}\cdots\alpha_1\alpha_0$, where each $\alpha_i \in \{0,1,2\}$ and
$$x = \alpha_k 3^k + \alpha_{k-1}3^{k-1} + \cdots + \alpha_1 3 + \alpha_0.$$
Note: this problem counts for passing LO5.

   (a) Let $\text{tern}(x,i)$ denote the ith ternary digit (i.e. α_i) of x. Use any known primitive recursive functions from the Models of Computation lecture to show that $\text{tern}(x,i)$ is primitive recursive. (12 pts)

   (b) It can be shown that any number x can be written in ternary using exactly $\lfloor \log_3 x \rfloor + 1$ ternary digits. Show that this function is primitive recursive with the help of the least-satisfying function and the power function. (13 pts)

2. Answer/solve the following. Note: this problem counts for passing LO7.

   (a) When simulating the computation $P_x(y)$, why must universal URM program P_U know the amount of registers that P_x uses in its computations? Explain. (7 pts)

   (b) A universal program P_U is simulating a program that has 754 instructions and whose Gödel number is
$$x = 2^7 + 2^{23} + 2^{63} + 2^{71} + 2^{105} + 2^{141} + \cdots + 2^{c_{754}} - 1.$$
If the current configuration of the computation of P_x on some input has encoding
$$\sigma = 2^3 + 2^5 + 2^{10} + 2^{13} + 2^{16} - 1,$$
then provide the next configuration of the computation *and* its encoding. (18 pts)

3. Answer the following. Note: this problem counts for passing LO8.

   (a) Define what it means to be a positive instance of decision problem `Total`. (5 pts)

   (b) The goal is to show that `Total` is undecidable. We assume it is decidable by assuming that its characteristic function
$$f(x) = \begin{cases} 1 & \text{if } x \text{ is a positive instance of } \texttt{Total} \\ 0 & \text{if } x \text{ is a negative instance of } \texttt{Total} \end{cases}$$
is total computable. Provide the definition for how to compute the "antagonist" function $g(x)$ based on the value of $f(x)$. (8 pts)

(c) By writing the values of $g(0), g(1), \ldots$ in the appropriate cells, verify that function g is different from each computable function ϕ_i, $i = 0, 1, \ldots$, which is a contradiction since g is computable and thus should be equal to at least one of the function. (5 pts)

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

(d) How can we be certain that $g(x) \neq \phi_2(x)$? (7 pts)

4. Solve the following.

(a) Let MaxRegJump(x, i) denote the maximum register index used by the i th instruction of P_x which we may assume is a Jump instruction. Show that MaxRegJump(x, i) is primitive recursive. Hint: you may use the fact that all encoding and decoding functions are primitive recursive. (12 pts)

(b) Let IncrementComponent(x, i) be the function that takes as input a tuple encoding x and a component index $1 \leq i \leq k(x)$, and outputs the encoding of the tuple

$$(a(1, x), \ldots, a(i, x) + 1, \ldots, a(k(x), x)).$$

Show that IncrementComponent(x, i) is primitive recursive. Hint: make use of the function $c(i, x)$, which gives the ith power-of-two exponent in the encoding of x. (13 pts)

5. Prove that both $\pi_1(z)$ and $\pi_2(z)$ are primitive recursive, where π_1 and π_2 satisfy the equation $\pi(\pi_1(z), \pi_2(z)) = z$. In other words, $\pi_i(z)$ returns the i th component of $\pi^{-1}(z)$, $i = 1, 2$. Note: solving this problem counts for passing LO5.

(a) Show $\pi_1(z)$ is primitive recursive. Hint: it equals the largest power of 2 that can divide into $z + 1$. (15 pts)

(b) Show that $\pi_2(z)$ is also primitive recursive. (10 pts)

6. Suppose P is a program that has 9 instructions, uses registers $R_1, \ldots, R_5$, and computes the function $f(u, v) = u \cdot v$. Write another program Q that makes use of P (as a block of instructions within Q) for computing the function x^y. Use the fact that x^y has the following recursive definition.

Base Case: $x^0 = 1$

Recursive Case: $x^{y+1} = x \cdot x^y$

Thus, your program should (non-recursively) implement the above recursive definition by making use of P within a `while` loop. Note: with the exception of the abstract P-block of instructions, all other instructions should be concrete. Hint: the P-block may be assigned a single instruction number (as opposed to 9 instructions).

(a) Provide the instructions of URM program Q. (12 pts)

(b) Write a paragraph that explains how your program works. (13 pts)

Learning Outcome Makeup Problems (0 Points Each)

LO1. Answer the following.

(a) Provide the definition of what it means to be a mapping reduction from decision problem A to decision problem B. (10 pts)

(b) Is $f(n) = 3n^2 + 7$ a valid mapping reduction from the `Even` decision problem to the `Odd` decision problem? Justify your answer.

LO2. An instance of the `Composite` decision problem is a natural number n, and the problem is to decide if n is composite, i.e. $n \geq 2$ and there is a number $2 \leq d < n$ that divides evenly into n.

(a) For a given instance n of `Composite` describe a certificate in relation to n.

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

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

(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. Hint: assume division of two k-bit numbers may be performed in $O(k^2)$ steps.

LO3. Recall the mapping reduction $f(\mathcal{C}) = (S, t)$, where f maps an instance of `3SAT` to an instance of the `Subset` decision problem. Given `3SAT` instance

$$\mathcal{C} = \{(x_1, \overline{x}_2, x_5), (x_2, x_3, \overline{x}_4), (x_1, x_2, x_4), (\overline{x}_1, \overline{x}_3, \overline{x}_5)\}$$

answer the following questions about $f(\mathcal{C})$. Hint: to answer these questions you are *not* required to draw the table.

(a) What is the value of t?

(b) How many numbers (counting repeats) are in S? What is the largest (in terms of numerical value) number in S?

(c) Determine a satisfying assignment for $\mathcal{C}$ and use it to identify a subset of S that sums to t. List all the members of S. Hint: there are multiple possible answers, but the subset you choose must correspond with your chosen satisfying assignment.

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 `Half Clique` is a graph $G = (V, E)$ and the problem is to decide if G has a clique of size $|V|/2$.

ii. An instance of `Set Cover (SC)` is a triple $(\mathcal{S}, m, k)$, where $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a collection of n subsets, where $S_i \subseteq \{1, \ldots, m\}$, for each $i = 1, \ldots, n$, and a nonnegative integer k. The problem is to decide if there are k subsets $S_{i_1}, \ldots, S_{i_k}$ for which

$$S_{i_1} \cup \cdots \cup S_{i_k} = \{1, \ldots, m\}.$$

iii. An instance of `UNSAT` is a Boolean formula F and the problem is to decide if F is unsatisfiable, meaning that F cannot be satisfied by any assignment over its variables.

iv. An instance of the `Composite` decision problem is a natural number n, and the problem is to decide if n is composite, i.e. $n \geq 2$ and there is a number $2 \leq d < n$ that divides evenly into n.

(b) Based on mapping reductions from the Mapping Reducibility and Computational Complexity lectures, the three polynomial-time mapping reductions $A \leq_m^p B$, $B \leq_m^p C$, and $C \leq_m^p$ `Traveling Salesperson` establish that `Traveling Salesperson` is an NP-complete problem. Provide the specific names of decision problems A, B, and C. Hint: A does *not* equal `SAT`. (3 points each)

(c) Which of the following decision problems is *not* `P`? (9 points)

i. `2SAT`

ii. `Composite`

iii. `Set Partition`

iv. `Prime`

LO6. Answer and solve the following.

(a) Compute the Gödel number for program $P = T(3,2), J(1,2,3), Z(3), S(6)$. Write your answer as a sum of powers of two minus 1.

(b) Provide the instructions of the program whose Gödel number is

$$x = 2^4 + 2^{14} + 2^{301} + 2^{381} - 1.$$

Solutions to 25-Point Problems

1. The base-3 representation of a natural number x is a sequence $\alpha_k \alpha_{k-1} \cdots \alpha_1 \alpha_0$, where each $\alpha_i \in \{0, 1, 2\}$ and
$$x = \alpha_k 3^k + \alpha_{k-1} 3^{k-1} + \cdots + \alpha_1 3 + \alpha_0.$$
Note: this problem counts for passing LO5.

   (a) Let $\text{tern}(x, i)$ denote the ith ternary digit (i.e. α_i) of x. Use any known primitive recursive functions from the Models of Computation lecture to show that $\text{tern}(x, i)$ is primitive recursive. (12 pts)

   Solution. We have
$$\text{tern}(x, i) = (x/3^i) \bmod 3.$$

   (b) It can be shown that any number x can be written in ternary using exactly $\lfloor \log_3 x \rfloor + 1$ ternary digits. Show that this function is primitive recursive with the help of the least-satisfying function and the power function. (13 pts)

   Solution. We have
$$\lfloor \log_3 x \rfloor + 1 = [\underset{z \leq x}{\lambda} (3^z > x)] - 1.$$

2. Answer/solve the following. Note: this problem counts for passing LO7.

   (a) When simulating the computation $P_x(y)$, why must universal URM program P_U know the amount of registers that P_x uses in its computations? Explain. (7 pts)

   Solution. P_U know the amount of registers used by P_x in order to properly encode each configuration of the computation of P_x on input y, since the configuration is a vector whose number of components equals one more than the number of registers used by P_x.

   (b) A universal program P_U is simulating a program that has 754 instructions and whose Gödel number is
$$x = 2^7 + 2^{23} + 2^{63} + 2^{71} + 2^{105} + 2^{141} + \cdots + 2^{c_{754}} - 1.$$
If the current configuration of the computation of P_x on some input has encoding
$$\sigma = 2^3 + 2^5 + 2^{10} + 2^{13} + 2^{16} - 1,$$
then provide the next configuration of the computation *and* its encoding. (18 pts)

   Solution.
$$c = \tau^{-1}(\sigma) = (3, 1, 4, 2, 2).$$
Also, $\beta(I_2) = 15$ and $15 \bmod 4 = 3$ implies that I_2 is a jump instruction $J(i, j, k)$, where $\xi(i - 1, j - 1, k - 1) = 3 = (15 - 3)/4$. Finally, to get $\xi^{-1}(3)$ we see that
$$3 + 1 = 4 = 2^2(2(0) + 1),$$

and $\pi^{-1}(2) = (0, 1)$ to give $\beta^{-1}(15) = J(1, 2, 1)$. Therefore,

$$c_{\text{next}} = (3, 1, 4, 2, 3)$$

and

$$\tau(c_{\text{next}}) = 2^3 + 2^5 + 2^{10} + 2^{13} + 2^{17} - 1.$$

3. Answer the following. Note: this problem counts for passing LO8.

(a) Define what it means to be a positive instance of decision problem `Total`. (5 pts)

Solution. See Section 3.1 of the Undecidability and Diagonalization Method lecture.

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

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

is total computable. Provide the definition for how to compute the "antagonist" function $g(x)$ based on the value of $f(x)$. (8 pts)

Solution. See Section 3.1 of the Undecidability and Diagonalization Method lecture (but $f(x)$ instead of $g(x)$ since the function roles have been reversed).

(c) By writing the values of $g(0), g(1), \ldots$ in the appropriate cells, verify that function g is different from each computable function ϕ_i, $i = 0, 1, \ldots$, which is a contradiction since g is computable and thus should be equal to at least one of the function. (5 pts)

Solution. The values of $g(x)$ are in red and show that g differs from all computable functions ϕ along the diagonal, except for those ϕ_x which are not total and for which $\phi_x(x) = 0$. See part c) for an explanation as to why $g(x)$ is different from these functions.

index\input x	0	1	2	$\cdots$	i	$\cdots$	**total?**
$\phi_0(x)$	$2 \to 0$	12	7	$\cdots$	$\uparrow$	$\cdots$	no
$\phi_1(x)$	8	$87 \to 88$	36	$\cdots$	96	$\cdots$	yes
$\phi_2(x)$	7	5	$0 \to 0$	$\cdots$	$\uparrow$	$\cdots$	no
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$\phi_i(x)$	0	32	65	$\cdots$	$5 \to 6$	$\cdots$	yes
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$

(d) How can we be certain that $g(x) \neq \phi_2(x)$? (7 pts)

Solution. $g(x)$ is total but $\phi_2(x)$ is undefined on input i, and so the functions will disagree on input $x = i$.

4. Solve the following.

(a) Let MaxRegJump(x, i) denote the maximum register index used by the i th instruction of P_x which we may assume is a Jump instruction. Show that MaxRegJump(x, i) is primitive recursive. Hint: you may use the fact that all encoding and decoding functions are primitive recursive. (12 pts)

Solution. We have

$$\text{MaxRegJump}(x, i) = \text{Max}(\pi_1(\pi_1((a(i, x) - 3)/4)), \pi_2(\pi_1((a(i, x) - 3)/4))) + 1.$$

(b) Let IncrementComponent(x, i) be the function that takes as input a tuple encoding x and a component index $1 \leq i \leq k(x)$, and outputs the encoding of the tuple

$$(a(1, x), \ldots, a(i, x) + 1, \ldots, a(k(x), x)).$$

Show that IncrementComponent(x, i) is primitive recursive. Hint: make use of the function $c(i, x)$, which gives the ith power-of-two exponent in the encoding of x. (13 pts)

Solution. We have

$$\text{IncrementComponent}(x, i) = \sum_{z=1}^{i-1} 2^{c(i,x)} + \sum_{z=i}^{k(x)} 2^{c(i,x)+1}.$$

5. Prove that both $\pi_1(z)$ and $\pi_2(z)$ are primitive recursive, where π_1 and π_2 satisfy the equation $\pi(\pi_1(z), \pi_2(z)) = z$. In other words, $\pi_i(z)$ returns the i th component of $\pi^{-1}(z)$, $i = 1, 2$. Note: solving this problem counts for passing LO5.

(a) Show $\pi_1(z)$ is primitive recursive. Hint: it equals the largest power of 2 that can divide into $z + 1$. (15 pts)

Solution. We have

$$\pi_1(z) = [\lambda_{p \leq z+1} \overline{\text{Div}(z + 1, 2^p)}] - 1.$$

(b) Show that $\pi_2(z)$ is also primitive recursive. (10 pts)

Solution. We have

$$\pi_2(z) = (z + 1)/\pi_1(z).$$

6. Suppose P is a program that has 9 instructions, uses registers $R_1, \ldots, R_5$, and computes the function $f(u, v) = u \cdot v$. Write another program Q that makes use of P (as a block of instructions within Q) for computing the function x^y. Use the fact that x^y has the following recursive definition.

Base Case: $x^0 = 1$

Recursive Case: $x^{y+1} = x \cdot x^y$

Thus, your program should (non-recursively) implement the above recursive definition by making use of P within a `while` loop. Note: with the exception of the abstract P-block of instructions, all other instructions should be concrete. Hint: the P-block may be assigned a single instruction number (as opposed to 9 instructions).

(a) Provide the instructions of URM program Q. (12 pts)

Solution.

1. $T(1,6)$
2. $T(2,7)$
3. $Z(2)$
4. $S(2)$ //place 1 in R_2
5. $J(7,8,14)$ //while the number of multiplications performmed is $< y$
6. P
7. $T(1,2)$
8. $T(6,1)$
9. $Z(3)$
10. $Z(4)$
11. $Z(5)$
12. $S(8)$
13. $J(1,1,5)$
14. $T(2,1)$

(b) Write a paragraph that explains how your program works. (13 pts)

Solution. The program uses a loop to successively compute

$$x \cdot 1, x \cdot x, x \cdot x^2, \ldots, x \cdot x^{y-1} = x^y.$$

It uses program P to perform each multiplication. It first places x and y in safe registers R_6 and R_7, respectively. It also uses R_8 to count up to y, the number of multiplications that must be performed. It then places 1 in R_2 and uses P to compute the initial product $x \cdot 1$. This result is then moved to R_2 in order to prepare for the next multiplication. Also, x is transferred to R_1 from R_6. Furthermore, before the next use of P, registers R_3, R_4, and R_5 must be cleared. Finally, after y multiplications, the result x^y is stored in R_1 as output.

Solutions to Learning Outcome Makeup Problems

LO1. Answer the following.

(a) Provide the definition of what it means to be a mapping reduction from decision problem A to decision problem B. (10 pts)

Solution. See Definition 2.1 of Turing and Mapping Reducibility lecture.

(b) Is $f(n) = 3n^2 + 7$ a valid mapping reduction from the `Even` decision problem to the `Odd` decision problem? Justify your answer.

Solution. This is a valid reduction since even number $2k$ maps to

$$12k^2 + 7 = 2(6k^2) + 2(3) + 1 = 2(6k^2 + 3) + 1,$$

which is odd, and odd number $2k + 1$ maps to

$$3(4k^2 + 4k + 1) + 7 = 12k^2 + 12k + 10 = 2(6k^2 + 6k + 5)$$

which is even.

LO2. An instance of the `Composite` decision problem is a natural number n, and the problem is to decide if n is composite, i.e. $n \geq 2$ and there is a number $2 \leq d < n$ that divides evenly into n.

(a) For a given instance n of `Composite` describe a certificate in relation to n.

Solution. A certificate is an integer c in the interval $[2, n-1]$.

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

Solution. One line: Return $\text{Div}(n, c)$.

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

Solution. The size of integer n is $\lfloor \log n \rfloor + 1$, i.e. the number of bits needed to represent n. So we use $\log n$ as a size parameter.

(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. Hint: assume division of two k-bit numbers may be performed in $O(k^2)$ steps.

Solution. Since we must perform a single division and each one requires $O(\log^2 n)$ steps, the running time is $O(\log^2 n)$.

LO3. Recall the mapping reduction $f(\mathcal{C}) = (S, t)$, where f maps an instance of `3SAT` to an instance of the `Subset` decision problem. Given `3SAT` instance

$$\mathcal{C} = \{(x_1, \overline{x}_2, x_5), (x_2, x_3, \overline{x}_4), (x_1, x_2, x_4), (\overline{x}_1, \overline{x}_3, \overline{x}_5)\}$$

answer the following questions about $f(\mathcal{C})$. Hint: to answer these questions you are *not* required to draw the table.

(a) What is the value of t?

Solution. $t = 111, 113, 333$

(b) How many numbers (counting repeats) are in S? What is the largest (in terms of numerical value) number in S?

Solution. We have $|S| = 2(m + n) = 2(4 + 5) = 18$. Largest value is $y_1 = 100, 001, 010$.

(c) Determine a satisfying assignment for $\mathcal{C}$ and use it to identify a subset of S that sums to t. List all the members of S. Hint: there are multiple possible answers, but the subset you choose must correspond with your chosen satisfying assignment.

Solution. Since $\alpha = (x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 0)$ satisfies $\mathcal{C}$, a subset that sums to t is

$$A = \{y_1, y_2, z_3, y_4, z_5, g_1, h_1, g_2, h_2, g_4\}.$$

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 `Half Clique` is a graph $G = (V, E)$ and the problem is to decide if G has a clique of size $|V|/2$.

ii. An instance of `Set Cover (SC)` is a triple $(\mathcal{S}, m, k)$, where $\mathcal{S} = \{S_1, \dots, S_n\}$ is a collection of n subsets, where $S_i \subseteq \{1, \dots, m\}$, for each $i = 1, \dots, n$, and a nonnegative integer k. The problem is to decide if there are k subsets $S_{i_1}, \dots, S_{i_k}$ for which

$$S_{i_1} \cup \cdots \cup S_{i_k} = \{1, \dots, m\}.$$

iii. An instance of `UNSAT` is a Boolean formula F and the problem is to decide if F is unsatisfiable, meaning that F cannot be satisfied by any assignment over its variables.

iv. An instance of the `Composite` decision problem is a natural number n, and the problem is to decide if n is composite, i.e. $n \geq 2$ and there is a number $2 \leq d < n$ that divides evenly into n.

Solution. i) NP, ii) NP, iii) co-NP, iv) P

(b) Based on mapping reductions from the Mapping Reducibility and Computational Complexity lectures, the three polynomial-time mapping reductions $A \leq_m^p B$, $B \leq_m^p C$, and $C \leq_m^p$ `Traveling Salesperson` establish that `Traveling Salesperson` is an NP-complete problem. Provide the specific names of decision problems A, B, and C. Hint: A does *not* equal `SAT`. (3 points each)

Solution. $A =$ `DHP`, $B =$ `UHP`, $C =$ `HC`.

(c) Which of the following decision problems is *not* `P`? (9 points)

i. `2SAT`

ii. `Composite`

iii. `Set Partition`

iv. `Prime`

Solution. `Set Partition`

LO6. Answer and solve the following.

(a) Compute the Gödel number for program $P = T(3,2), J(1,2,3), Z(3), S(6)$. Write your answer as a sum of powers of two minus 1.

Solution. We have

$$\gamma(P) = 2^{46} + 2^{226} + 2^{235} + 2^{247} - 1.$$

(b) Provide the instructions of the program whose Gödel number is

$$x = 2^{4} + 2^{14} + 2^{301} + 2^{381} - 1.$$

Solution. $P = Z(1), S(3), T(4,5), J(1,2,3)$.