Final

DECEMBER 10, 1992. 5:30–8:00 PM

There are five problems for a total of 100 points. Good luck.

Name: ______________________________

Problem	Credit
1	
2	
3	
4	
5	
Total	

1. (20 Points.)

   Consider a program which plays tick-tack-toe. One can encode a game board B as follows. There are nine boxes in tick-tack-toe which we will number $i = 0, \ldots, 8$. Define function τ which takes board B and integer i and equals,

$$\tau(B, i) = \begin{cases} 1 & \text{if box } i \text{ is empty in board } B, \\ 2 & \text{if box } i \text{ has an } X \text{ in board } B, \\ 3 & \text{if box } i \text{ has an } O \text{ in board } B. \end{cases}$$

   The game board is then represented as,

$$\sigma(B) = \prod_{i=0}^{8} pr(i)^{\tau(B,i)}.$$

   Consider the function $f : \mathbf{N} \to \mathbf{N}$,

$$f(b) = \begin{cases} 1 & b = \sigma(B) \text{ and it is possible for X to win given board } B. \\ 0 & \text{else} \end{cases}$$

   Show that f is primitive recursive. (HINT: Loop-programs.)

2. (20 Points.)

   Mark in each of the following boxes "Y" if the computation system is equivalent in power to while-programs, or "N" if it is less powerful.

   - [ ] Repeat-programs: These are while-programs except the basic control construction is *repeat ... until* rather than *while ...* .
   - [ ] A PS/2 computer whose FORTRAN, for reason of programming style, has forbidden the use of GOTO's. For our purposes, define FORTRAN as:
     - Having all the usual arithmetic capabilities on an infinite supply of integer variables.
     - Having an if-then construction of the form,

       ```
       IF (condition) THEN
          statements
       END IF
       ```

     - Having a do-loop construction of the form,

       ```
       DO variable=initial,final,step
          statements
       END DO
       ```

       The limits to the do-loop are fixed upon entry to the loop.
     - Having subroutine and functions calls only when they cannot result in recursion. (E.g. "A" calls "B" which calls "A" is not allowed.)
   - [ ] A while-program whose arithmetic rules (set to zero, successor and predecessor) have been replaced by the following string manipulation rules. A variable can be initialized to the empty string, `X:=''`. An "a" can be appended to the string contained in a variable, `X:=X||'a'`. Two strings can be checked for unequal length. (The length of the empty string is zero.) While and compound statements are as before.
   - [ ] A Turing Machine whose tape is infinitely long *only in one direction.*

3. (20 Points.) Show that the *Integer Programming Decision Problem* is in NP. (Do not attempt to show it NP-complete!)

   An instance of an Integer Programming Decision Problem is given by a set of variables $\{ X_1, \ldots, X_n \}$, a set of linear equations using these variables, called the *constraints,*

$$\begin{aligned} b_1 &\geq a_{1,1}X_1 + a_{1,2}X_2 + \ldots + a_{1,n}X_n \\ b_2 &\geq a_{2,1}X_1 + a_{2,2}X_2 + \ldots + a_{2,n}X_n \\ &\vdots \\ b_m &\geq a_{m,1}X_1 + a_{m,2}X_2 + \ldots + a_{m,n}X_n, \end{aligned}$$

   where all the b_i and $a_{i,j}$ are integers, a cost function,

$$c(X_1, \ldots, X_n) = c_1X_1 + c_2X_2 + \ldots + c_nX_n,$$

   where all the c_i are integers, and an integer B.

   The decision problem is to answer Yes if there exists an assignment of *integers* to the X_i such that all of the constraints are true and the cost c is greater or equal to B.

4. (20 Points.)

   Give a bijection from pairs of integers to the naturals,

   $$\mu(i,j) = k, \;\; i,j \in \mathbf{Z}, \; k \in \mathbf{N}.$$

   (HINT: Use Kfoury, Moll and Arbib's pairing function $\tau : \mathbf{N}^2 \to \mathbf{N}$ and build from there.)

5. (20 Points.)

   If $f, g : \mathbf{N} \to \mathbf{N}$ are two functions, recall that $f \geq g$ by extension ordering if,

   (a) The domain of definition of f includes that of g, and,

   (b) for any i in the domain of definition of g, $g(i) = f(i)$.

   The greatest-lower-bound of a family of partial functions

   $$\mathcal{G} = \{g_i : \mathbf{N} \to \mathbf{N} \mid i = 0, 1, \ldots\}$$

   is a partial function G such that,

   (a) $g_i \geq G$ for all i. (That is, G is a lower bound of the g_i.)

   (b) For any other G' such that $g_i \geq G'$ for all i, $G \geq G'$. (That is, among all lower bounds for the g_i, G is the greatest.)

   Show that, under the assumption that the g_i form a descending chain,

   $$g_0 \geq g_1 \geq g_2 \geq \ldots,$$

   the greatest-lower-bound of family $\mathcal{G}$ exists.

   Show that the greatest-lower-bound need not be computable even if all the g_i are.