CMSC210 Final/Module D Comprehensive Exercises

Due at the beginning of the designated final exam period on 12/12/03 [There are four (4) problems.]

1. Cryptarithmetic

Note: See the supplement exercise for Unit C6 Counting/Probability at:
http://www.tcnj.edu/~komagata/cmsc210/03f/units/C6-count-supp.pdf

Consider the following cryptarithmetic problems, i.e., to find an injective function from the set of letters (only those used in the problem) to the set of digits, {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} which would satisfy the arithmetic.

Problem #1

```
    S E N D
+   M O R E
-----------
  M O N E Y
```

Problem #2

```
  D O N A L D
+ G E R A L D
-------------
  R O B E R T
```

Note: How to find a solution to these problems is not the focus of this exercise. If you want to know the solutions, do a Web search with "cryptarithmetic solver" and find a program to solve these problems automatically.

One brute-force approach to solving these problems is to systematically (i.e., repeatedly) assign every digit to the letters and check whether the arithmetic is correct.

A. What would be the formulas to compute the number of different digit assignments for **Problem #1** and **Problem #2**? Explain.

   Note: Do not actually compute the numeric values. Give a formula that may involve factorial (!), permutation (*P*), combination (*C*), constants, and arithmetic operations among them. Same for Question B.

In practice, we identify certain constraints that can be used to exclude impossible cases. For example, in **Problem #1**, we can exclude **M** = 0 because the letter is used to represent the carry (of the computation of the lower digits), which must be non-zero. In addition, we know that **M** $\leq$ 1, because adding two digits cannot exceed 19, even including the carry from the computation of lower digits. As for **S**, we can say that **S** $\geq$ 8 because **S** + **M** + *carry* > 10 where **M** = 1 and *carry* $\leq$ 1.

B. We know **M** = 1 and **S** $\geq$ 8. What would be the formulas to compute the number of different digit assignments for **Problem #1** (with these constraints)? Explain.

Suppose that you create cryptarithmetic problems like **Problem #1** automatically, using random generation of letters. That is, you will assign an uppercase alphabetic letter to each of the 13 letter positions. In order for the solution (as a function) to be injective, a single problem must have *no more than* 10 different letters.

C. You are given the following formula to compute the probability of *exactly* 10 letters assigned to the 13 letter positions: $C(26, 10) \times 10^{13}/26^{13}$. Unfortunately, the formula does not compute the probability correctly. Analyze the formula and explain the problem.

   Hint: The use of "10^{13}" in the formula is supposedly dependent on the use of "$C(26, 10)$".

D. Identify and explain the logic-structure connection involved in this type of cryptarithmetic problems. That is, identify both the logic and the structure parts, and then describe how the structure would satisfy the logic.

2. Nearly-Decomposable Systems

In response to Unit D2 Summary Exercise, some of you came up with an idea that the universe is a partition with cardinality 1. That is, everything in the universe interacts with everything (including itself) directly or indirectly, e.g., if R_1 is the reflexive, transitive, and symmetric closure of {(1, 2), (2, 3), (3, 1)}, the partition is {[(1, 1)]}. While this view may be correct, the relation is not very informative as it is simply the Cartesian products of all the objects. Perhaps, inclusion of indirect interaction is problematic. So, we now focus only on ***direct*** interactions. Then, although we still want to maintain reflexivity and symmetry, we may lose transitivity if the transitivity reflects indirect interactions.

If everything directly interacts with everything, the cardinality of the partition is still 1. On the other hand, if a system is separated into multiple, completely independent, highly-connected subsystems, the cardinality of the partition is greater than 1. But Herbert Simon (Nobel Memorial Prize in Economic Science, 1978) points out that neither of these cases are realistic for modeling many real-world phenomena. Instead, he argues that many real-world phenomena are more loosely-coupled (called **nearly-decomposable systems**). This view can be described in terms of the following hypotheses.

Hypotheses

1. Complex systems should not be characterized as a partition with cardinality 1.
2. Complex systems should not be characterized as a partition with cardinality greater than 1.
3. Complex systems are still considered as loosely-coupled subsystems. That is, the relation would still define subsystems like equivalence classes while they are *not completely disjoint*.

A. Prove that there are no equivalence relations that would satisfy both **Hypotheses** 1 and 2. Use the proof pattern discussed in class.

   Note: Exclude the empty relation, which could be interpreted as an equivalence relation with a partition with cardinality 0.

B. A randomly generated relation representing direct interaction between objects is not likely to satisfy **Hypothesis** 3. Explain why.

C. Give a tabular representation of an example relation that would characterize a nearly-decomposable system. Explain, referring **Hypotheses** 1 through 3.

   Hint: See Unit D4 slides regarding the tabular representation of a relation.

   Note: To be able to clearly visualize all the **Hypotheses**, you will need a set of several objects, not just, say, 2.

Reference: Simon, Herbert Alexander. 1969. *The sciences of the artificial*. M.I.T. Press. [available in the TCNJ library; the 3rd ed. (1996) is the most current; not essential for this exercise, though]

3. Slime Scene

Some scientists observed that the following obscure scientific law applies to the small amount of slime (certain sticky matter, which a lay person would not bother to analyze) found in their lab.

Law

1. $\exists x\, f(x)$
2. $\big(\neg\exists x\,(c(x) \wedge v(x))\big) \rightarrow \big(\neg\exists x\, f(x)\big)$
3. $\big(\exists x\,(c(x) \wedge v(x))\big) \rightarrow \big(\exists x \exists y\,(c(x) \wedge c(y) \wedge x \neq y)\big)$
4. $\neg\exists x \exists y\,(h(x) \wedge c(y) \wedge s(x, y))$
5. $\forall x\,((h(x) \wedge v(x)) \rightarrow s(x, x))$

This **Law** is expected to explain the composition of the slime, which can be represented as a structure **Slime** = (O, C, P, F, V, S) where

- O contains all the involved objects.
- C and P are subsets of O, and interpret the unary predicate symbols c and h, respectively. For example, $h(a)$ is true if and only if $a \in P$.
- F defines the meaning of the unary predicate symbol f. For example, $f(a)$ is true if and only if $a \in F$.
- V defines the meaning of the unary predicate symbol v. For example, $v(a)$ is true if $a \in V$.
- S defines the meaning of the binary predicate symbol s. For example, $s(a, b)$ is true if and only if $(a, b) \in S$.

In addition, the scientists also noted that the structure satisfies the following fact as well. **Fact**: $|C| \leq |P|$

Now, we see the logic-structure connection. However, we do not know what exactly the predicate symbols (in the logic) and the corresponding relation symbols (in the structure) mean. Thus, we should not introduce additional assumptions beyond what the **Law** and **Fact** specify. For example, you do not know whether the sets C and P intersect or are disjoint.

A. Formally define all the structure components of the *smallest* instance of **Slime** (call it **Slime**$_0$) that would satisfy all the statements in **Law** and the **Fact** shown above. For relations/functions, give their types as well. Explain how you came to that conclusion.

   Note: The smallest instance would include the minimal number of objects.

B. Suppose that the smallest instance **Slime**$_0$ contains C_0 such that $|C_0| = k_0$. Prove the following, using *Mathematical Induction*: For any natural number $i \geq k_0$, there is an instance of **Slime**$_i$ containing C_i with $|C_i| = i$ which satisfies the **Law** and **Fact** (i.e., there are infinitely many satisfying structures). Follow the Mathematical Induction proof pattern discussed in class.

   Hint: Do not forget that the **Fact**, $|C| \leq |P|$, must also be satisfied.

If you developed detective skills in "Crime Scene" (Module B Comprehensive Exercise 2), you must have noticed some connection between Crime Scene and Slime Scene. In fact, although not many people would notice, the ability to analyze the logic-structure connection is an essential skill for detectives and scientists alike. For example, you may have realized that the statements in **Law** (formal) correspond to **Information** 1 through 5 (informal) in Crime Scene (in the given order), and the **Fact** correspond to **Information** 6.

C. (i) Copy (or recreate, but do not modify) the structure in your answer to Crime Scene 2F and call it **CrimeScene**. Then, (ii) analyze whether **Slime**$_0$ above is isomorphic to **CrimeScene**. Explain fully, including why or why not they are isomorphic. (iii) Redo Question 2G in Module B Comprehensive Exercise 2.

4. Egotruism

Normally, "egoism" and "altruism" are considered two extreme forms of regarding self and others (use the dictionary if you need clarification). But there are stories (fiction and non-fiction) that are not so clear cut. One might recall "Christmas Carol." Philadelphia Inquirer had an article about a philanthropist, who gave so much to charity that he was abandoned by his family. As discussed in an earlier problem, things interact in various ways. Maybe it is too simplistic to contrast these two ism's; maybe they are not so different (hence, "egotruism").

In an attempt to simulate the connection between egoism and altruism, we define mutually recursive functions *ego* (for egoist) and *alt* (for altruist) as shown below. Note that these functions have the type $\mathbf{N} \to \mathbf{N}$, where $\mathbf{N}$ is the set of natural numbers. Also note that *round* returns the nearest natural number, e.g., *round*(1.49) = 1, *round*(1.5) = 2. The idea is as follows: *ego* accesses the shared resources (represented as *r*, the input to the function), uses some of it, and returns a reduced amount of the shared resources. Due to its greedy nature, *ego* cannot sustain its operation when the shared resources are too low (under 10). On the other hand, *alt* accesses the shared resources and returns an increased amount. It cannot sustain its operation if the shared resources become abundant (above 90).

$$\begin{aligned} ego(r) &= \begin{cases} r & \text{if } r < 10 \\ alt\left(round\left(r^2/100\right)\right) & \text{otherwise} \end{cases} \\ alt(r) &= \begin{cases} r & \text{if } r > 90 \\ ego\left(round\left(r \times (200 - r)/100\right)\right) & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

A. Explain how to compute the value of *ego*(90). Do this step by step. That is, compute the first recursive step as *round*(90^2 / 100) = 81, and then compute the second recursive step, using 81 as the input to *alt*, and so on.

B. Observe the first several recursive steps of computing *ego* for the following inputs:
- *ego*(61): 61, 37, 60, 36, ...
- *ego*(62): 62, 38, 62, 38, ...
- *ego*(63): 63, 40, 64, 41, ...

Estimate and explain the behavior of *ego* as accurately as possible for all the natural number inputs.

C. Suppose that termination of the functions corresponds to the termination of both the egoist and the altruist (death, deportation, seclusion, etc.). Identify a relevant real-world story, and analyze the connection between the story and the mathematical behavior of the functions. Your story must be unique to you.

Note: If you have difficulty identifying a unique story, make up one and indicate that you did so.

D. Suppose that the function definition (1) was modified as follows (now, two separate recursive functions, with no mutual recursion):

$$\begin{aligned} ego_2(r) &= \begin{cases} r & \text{if } r < 10 \\ ego_2\left(round\left(r^2/100\right)\right) & \text{otherwise} \end{cases} \\ alt_2(r) &= \begin{cases} r & \text{if } r > 90 \\ alt_2\left(round\left(r \times (200 - r)/100\right)\right) & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

(i) Discuss whether the above functions always terminate. (ii) Discuss real-world implications of the mathematical behavior of these functions.

Note: All the above functions are fairly easy to implement in most modern programming languages as well as in Mathematica. You are encouraged to experiment with various inputs. Also try variants for real numbers.

<End>