# SSIE 501 Take Home Exam

Grant T. Aguinaldo

November 28, 2020

Grant Aguinaldo

# SSIE 501 Introduction to Systems Science
## Fall 2020
## Test #3 Take-Home Exam
## November 24 – December 3, 2020

**Permitted:** Calculator, computer, books, any notes you collected in class.

**NOT Permitted:** Consultation with anyone else other than the professor.

**NOTES:**

1. Be sure to explain in detail not just what your answer is, but also WHY you feel that is the correct answer.
2. Some of these questions are of an open-ended nature. If you cannot come to a definitive answer, just answer the best you can.
3. For some of these questions there may be more than one right way to answer the question.
4. If the meaning of a question is unclear to you, you can make your best effort to interpret the original intention. State your assumptions about the meaning of the question in detail.
5. Turn in your answers as a single legible file (pdf or MS Word) email attachment to hlewis@binghamton.edu no later than 1:15 PM Thursday, December 3.

**Problem 1: (25 points)** A chess club has 11 members, whose names (by an incredible coincidence) just happen to start with the first 11 letters of the alphabet—Arthur, Betty, . . . , Katherine. Thus, for the sake of convenience we will represent the members by their first initials, $a$ through $k$. Over time each member has played other members a number of times. Furthermore, many members have held practice games against themselves on the days when not paired off against one of the other members. Let us define a relation to mean "has defeated". That is to say, for any two members $x$ and $y$, $(x, y)$ is an element of $R_1$ if and only if $x$ has defeated $y$ on at least one previous occasion. (Because of the practice games played against oneself, we will consider these to be a member defeating himself and therefore the relation is not anti-reflexive. Also, given any two members, they may have played each other more than once with the result sometimes the first player won and in other cases the other did, and for this reason, we cannot assume the relation to be anti-symmetric either.) The relation can be expressed by its characteristic matrix as shown below:

| $R_1$ | *a* | *b* | *c* | *d* | *e* | *f* | *g* | *h* | *i* | *j* | *k* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *a* | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 |
| *b* | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| *c* | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 3 |
| *d* | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| *e* | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 3 |
| *f* | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 3 |
| *g* | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 |
| *h* | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 3 |
| *i* | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 4 |
| *j* | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 3 |
| *k* | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 4 |

Let the above set of members $T_1 = \{a, b, c, \ldots, k\}$ and $R_1$ as shown be the system $S_1$.

a. Given $S_2 = (T_2, R_2)$ where $T_2 = \{1, 2, 3, 4, 5\}$ and $R_2 = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (3, 1), (3, 3), (3, 5), (4, 1), (4, 2), (4, 4), (5, 4), (5, 5)\}$, demonstrate that $S_2$ is a strong homomorphism of $S_1$ including a clear statement of the function $h$: $T_1 \rightarrow T_2$.

b. Give an example of how if $R_2$ were to be modified slightly it would result in $S_2$ being a weak homomorphism, but not a strong homomorphism of $S_1$.

c. Why is it true that a system that is a strong homomorphism of the original system is a good simplifying model, but one that is a weak homomorphism is not? Difference between onto and one-to-one.

d. Sometimes we use systems that are isomorphic to the original system as system models, yet this does not lead to a simplification in the sense of the example above. List several reasons why we might want to use an isomorphic (and thus non-simplifying in the sense of system structure) model. (These reasons need not apply to the case of the chess club.)

**Problem 2: (15 points)** Consider a digital black box with $n$ switches on the front (for some natural number $n$) and an output port that produces some real number output value based on the combination of the settings of each of the individual switches. Each of the switches has five settings, very low, low, medium, high, and very high. By "black box" we mean that the only way to know the output value under some possible combination of switch settings is actually to try out that combination. By "digital black box" we mean that the setting of the switches and the reading of the output value can be performed automatically by the computer. Assume that it is essential to find the combination of switch settings that produces the absolutely highest possible output value.

a. If it takes 0.001 second (one one-thousandth of a second) to try out one possible combination of switch settings, how long will it take to determine the highest possible output value when the number of switches is 5? 10? 15? 20? 25? 30?
b. Suppose that there are $10n$ (the number of switches times 10) bit evaluations necessary to try out a combination and see the output value. At what number of switches does the problem become transcomputational (in the sense of Bremermann's limit)?

**Problem 3: (10 points)** Suppose that for some event that has not yet occurred we can expect to get a value (outcome) for variable $x$ and a value for variable $y$. The state space (the set of possible outcomes) for variable $x$ is $X = \{\alpha, \beta, \gamma\}$ and that the state space for $y$ is $Y = \{a, b, c, d\}$. We know the probabilities for each possible outcome for both $x$ and $y$ to be the following: $p(\alpha) = 0.20$, $p(\beta) = 0.30$, $p(\gamma) = 0.50$, $p(a) = 0.10$, $p(b) = 0.20$, $p(c) = 0.30$, and $p(d) = 0.40$.

a. What are the Hartley information measures $\mathrm{I}(X)$ and $\mathrm{I}(Y)$ in bits?
b. What are the Shannon information measures $\mathrm{H}(X)$ and $\mathrm{H}(Y)$ in bits?
c. Suppose that the joint probabilities of the outcomes are the following what is the information transmission $\mathrm{T}(X, Y)$?

| | | *y* | | | |
|---|---|---|---|---|---|
| | $p(x, y)$ | $a$ | $b$ | $c$ | $d$ |
| | $\alpha$ | 0.05 | 0.10 | 0.02 | 0.03 |
| $x$ | $\beta$ | 0.03 | 0.04 | 0.18 | 0.05 |
| | $\gamma$ | 0.02 | 0.06 | 0.10 | 0.32 |

**Problem 4: (25 points)** You wish to study the training plan of a rival triathlete by understanding his pattern of distances he runs, swims, and cycles each week in miles. You didn't start collecting information on his training plan until week 5 (May 3 to May 9) of the 2020 training season.

- Week 5, he runs 10 miles, swims 3.0 miles, and cycles 50 miles.
- Week 6, he runs 13 miles, swims 2.5 miles, and cycles 60 miles.
- Week 7, he runs 16 miles, swims 2.0 miles, and cycles 70 miles.
- Week 8, he runs 19 miles, swims 1.5 miles, and cycles 80 miles.
- Week 9, he runs 22 miles, swims 1.0 mile and cycles 90 miles.
- Week 10, he runs 0 miles, swims 3.0 miles, and cycles 100 miles.
- Week 11, he runs 19 miles, swims 2.5 miles, and cycles 30 miles.
- Week 12, he runs 22 miles, swims 2.0 miles, and cycles 40 miles.
- Week 13, he runs 25 miles, swims 1.5 miles, and cycles 50 miles.
- Week 14, he runs 28 miles, swims 1.0 miles, and cycles 60 miles.
- Week 15, he runs 6 miles, swims 3.0 miles, and cycles 70 miles.
- Week 16, he runs 9 miles, swims 2.5 miles, and cycles 80 miles.
- Week 17, he runs 12 miles, swims 2.0 miles, and cycles 90 miles.
- Week 18, he runs 15 miles, swims 1.5 miles, and cycles 100 miles.
- Week 19, he runs 34 miles, swims 1.0 miles, and cycles 30 miles.
- Week 20, he runs 12 miles, swims 3.0 miles, and cycles 40 miles.

a. How much do you predict he might run, swim, and cycle in Week 21?
b. How much would you retrodict that he ran in Weeks 4, 3, 2, and 1?
c. Completely state a source system according to Klir's epistemological hierarchy.
d. Completely state a data system.
e. Completely state a generative system.

**Problem 5: (10 points)** Given sets $A = \{1, 2, 3\}$, $B = \{\beta\}$

a. What is the number of relations that can be defined on $A \times B$? - 8 entries
b. What is $\mathscr{P}(A \times B)$ where $\mathscr{P}$ denotes power set? Write it out explicitly. - 8 entries
c. How are parts a and b of this problem related to each other?
d. What is $\mathscr{P}(\mathscr{P}(A))$? Write it out explicitly. - 256 entries
e. What is $\mathscr{P}(\mathscr{P}(\mathscr{P}(B)))$? Again, write it out explicitly. - 16 entries

**~~Problem~~ 6: (25 points)**

Consider two systems, $\boldsymbol{S_1} = (T_1, R_1)$ and $\boldsymbol{S_2} = (T_2, R_2)$. Let $T_1$ represent a set of nurses who have worked in a particular department in a particular hospital. Here, we will just anonymously label them Nurse #1 through Nurse #8. The nurses' shift assignments vary from week to week, so that some nurses work on the same schedule one week and may work with other nurses the next. Let $R_1 \subseteq T_1^2$ be a relation defined such that for any $x, y \in T_1$, let $(x, y) \in R_1$ if and only if Nurse $x$ has at some point worked on the same shift assignment as Nurse $y$. We are also provided with the following information.

Reflexive and symmetric.

Nurse #1 has worked a shift with Nurses# 2, 3, and 4.
Nurse #2 has worked a shift with Nurses# 3, 4, 5, and 6.
Nurse #3 has worked a shift with Nurses# 4, 5, 7, and 8.
Nurse #4 has worked a shift with Nurse# 6.
Nurse #5 has worked a shift with Nurses# 6, 7, and 8.
Nurse #6 has worked a shift with Nurse# 8.
Nurse #7 has worked a shift with Nurse# 8.

Let $T_2$ represent a set of 8 prescription drugs, and we'll just label these Drug *a* through Drug *h*. Let $R_2 \subseteq T_2^2$ be a relation defined such that for any $x, y \in T_2$, let $(x, y) \in R_2$ if and only if Drug $x$ is taken together with Drug $y$ by at least one patient. We also know the following.

Drug *a* is sometimes taken together with Drugs *c, d, f,* and *h*
Drug *b* is sometimes taken together with Drugs *c, e, f, g,* and *h*.
Drug *c* is sometimes taken together with Drugs *e* and *h*.
Drug *d* is sometimes taken together with Drugs *f* and *h*.
Drug *e* is sometimes taken together with Drugs *f* and *g*.
Drug *f* is sometimes taken together with Drugs *g* and *h*.

Is System $\boldsymbol{S_1}$ isomorphic to $\boldsymbol{S_2}$? Explain in detail. (Including the bijective function used.)

YES

**The End.**

**Thanks to all of you for a thoroughly enjoyable experience teaching this class.**

| b | f | h | D | G | E. | a. | c |
|---|---|---|---|---|---|---|---|
| 2 | 3 | 5 | 7 | 1 | 4 | 8 | 6 |

# 1 Problem 1

**Part a.** Using the mapping below, I am able to demonstrate that $S_2$ is a strong homomorphism of $S_1$. All of the mappings to show that $S_2$ is a strong homomorphism of $S_1$ is provided in the Appendix for Problem 1.

In particular, the map below is able to satisfy the first requirement that [1] $\forall x_1, x_2 \in R_1,$ if $(x_1, x_2) \in R_1$ then $[h(x_1), h(x_2) \in R_2]$ and [2] that $\forall y_1, y_2$ in $R_2$ $\exists$ some $x_1, x_2 \in R1$ such that $h(x_1) = y_1,\ h(x_2) = y_2$. In passing, I believe that there are 14 mappings that show that $S_2$ is a strong homomorphism of $S_1$. All proposed mappings that demonstrate that $S_2$ is a strong homomorphism of $S_1$ is provided in the Appendix for Problem 1 (Pages 21 and 22).

By background, to solve this problem, I developed a `Python` script that searched through all possible 48,828,125 mappings (as determined by $5^{11}$) to determine which of these permutations are candidates for a homomophisim (Page 23). Once I found the mappings that are a homomophisim, I then analyzed this subset to determine which ones were strong homomorphisms. The code for my approach is also included in the Appendix for Problem 1 (Page 26).

1. $a \to 4$
2. $b \to 2$
3. $c \to 1$
4. $d \to 1$
5. $e \to 1$
6. $f \to 5$
7. $g \to 1$
8. $h \to 5$
9. $i \to 1$
10. $j \to 3$
11. $k \to 5$

**Part b.** Based on my analysis, I don't believe that any of the 14 proposed homomorphic solutions are weak homomorphisms. Again, the code to answer this question is provided in the Appendix for Problem 1. If a mapping was a weak homomorphism, the mapping would only need to satisfy the requirement that $\forall x_1, x_2 \in R_1,$ if $(x_1, x_2) \in R_1$ then $[h(x_1), h(x_2) \in R_2]$. In other words, if during the reverse decide from $R_2$ to $R_1$ if none of the elements in $R_1$ can be decoded back to $R_2$ then the mapping is not a strong homomorphism and is weak (Page 25).

**Part c.** From page 96 of Facets, it notes that for two systems, $S_1 = (X, R)$ and $S_2 = (Y, Q)$, $S_1$ will be a weak homomorphic image of $S_2$ if $\exists$ an onto function, $h$ such that for all $(x_1, x_2) \in X$, $(x_1, x_2) \in R$ implies $[h(x_1), h(x_2)] \in Q$. Further Klir notes that the conditions for a strong homomorphic image include, that for all $(y_1, y_2) \in Y$, $(y_1, y_2) \in Q$ implies $(x_1, x_2) \in R$ for some $x_1 \in h^{-1}(y_1)$ and $x_2 \in h^{-1}(y_2)$. The difference between a weak and strong homomorphic image is that a weak image only requires an onto function from $R$ to $Q$, whereas a strong image not only requires the relational preservation from from $R$ to $Q$ but also requires that the function also preserve the relation from $Q$ to $R$. In other words, a weak image preserves the relation in one direction (unidirectional), but a strong image preserves the relation in two directions (bidirectional). Within the context of the GSPS, strong homomorphic images are desired since one needs to not only abstract the original problem/system at hand into a general systems problem, but must also interpret the solution to the general systems problem back to the original problem. By having a strong strong homomorphic image, one is guaranteed that the properties of the original system is preserved during the abstraction process, to create the general systems model, and also preserved during the interpretation process to solve the original problem at hand. Moreover, if the abstracted model is a simplifying and strong homomorphic image, then we are working an ideal model since the abstracted model is, in theory, less complex, than the original, but still retains many of the relations found in the original model. Again, this does not hold, if we are working with a weak homomorphic image.

**Part d**. There can be several reasons why we would want to use a non-simplifying isomorphic model. First, when we simplify any model, we run the risk of losing information. In paper 17, Klir writes about the "identification problem" which can be defined as the seeking to understand the extent that a one or more subsystems can be used to adequately represent the overall system. Klir also note that when it comes to the "identification problem" there is "...no guarantee that the overall system is adequately represented by the subsystems." Further Klir notes that the identification problem is a general class of problems that require reasoning from incomplete information. Taken together, when working with a simplified isomorphic model, it is possible that the simplified model may not capture all of the fine structure of the original system and we may be working with incomplete information when trying to either reconstruct the original system, or infer properties of the original system based on the simplified model. For this reason, it may be useful and desirable to work with a non-simplifying isomorphic model. Second, when we simply the system, we run the risk of having a simplified model that takes on very different properties than the original model. This is noted in Paper 23 where Rosen notes that when it comes to abstractions "...we must note that the abstracted subsystem may exhibit capacities of it's own, which do not pertain to the fact that it is in fact a subsystem of the larger system..."[1] Due to the fact that a simplifying isomorphic model is "simplified," it is possible that the properties of the simplified model may be different than that of the original model, even thought we are working with an isomorphic system. Taken together, one way to ensure that the properties of the original model are translated into the model, is to work with a non-simplified version of the model.

# 2 Problem 2

**Part a.** For each switch, there are five possible outcomes (or events). This means that the total number of events possible by $n$ switches is given by $5^n$. Further, if it takes 0.001 seconds to evaluate a single outcome (event), then the total amount of time, in seconds, to evaluate $n$ switches is given by $(0.001 \cdot 5^n)$. Therefore, the time to evaluate 5, 10, 15, 20, 25, and 30 switches is given in the table below along with an example calculation for 5 switches.

Example Calculation:

$$0.001 \times 5^5 = 3.13 \text{ seconds} \tag{1}$$

| Num. of Switches | Total Time (sec) |
|---|---|
| 5 | $3.13 \times 10^0$ |
| 10 | $9.77 \times 10^3$ |
| 15 | $3.05 \times 10^7$ |
| 20 | $9.54 \times 10^{10}$ |
| 25 | $2.98 \times 10^{14}$ |
| 30 | $9.31 \times 10^{17}$ |

**Part b.** If there are $10n$ bit evaluations, where $n$ is the number of switches, the expression to find the number of switches that need to be present for the problem to be transcomputational is given by $5^{10n} > 10^{93}$. Solving this expression for $n$, can be done by taking the $log_5$ of $5^{10n}$ which affords $10n > 10^{92}$. Finally, solving for $n$ yields $> 10^{92}$ switches. **In other words, $> 10^{92}$ switches need to be present for the problem to become transcomputational** (referenced method outlined in Page 146 of Facets).

# 3 Problem 3

**Part a.** The Hartley Information is calculated below.

$$I(X) = log_2(|X|) = 1.585 \; bits \tag{2}$$

[1]Abstractions are mentioned here since it is a way to simplify a system.

Where $|X| = 3$.

$$I(Y) = log_2(|Y|) = 2.000\ bits \tag{3}$$

Where $|Y| = 4$.

**Part b.** The Shannon Information is calculated as follows:

$H(X) = (-0.2 \cdot log_2(0.2)) + (-0.3 \cdot log_2(0.3)) + (-0.5 \cdot log_2(0.5)) = \underline{1.485 \text{ bits}}$

$H(Y) = (-0.1 \cdot log_2(0.1)) + (-0.2 \cdot log_2(0.2)) + (-0.3 \cdot log_2(0.3)) + (-0.4 \cdot log_2(0.4)) = \underline{1.846 \text{ bits}}$

$H(X,Y) = (-0.05 \cdot log_2(0.05)) + (-0.03 \cdot log_2(0.03)) + (-0.02 \cdot log_2(0.02)) + (-0.10 \cdot log_2(0.10)) + (-0.04 \cdot log_2(0.04)) + (-0.06 \cdot log_2(0.06)) + (-0.02 \cdot log_2(0.02)) + (-0.18 \cdot log_2(0.18)) + (-0.10 \cdot log_2(0.10)) + (-0.03 \cdot log_2(0.03)) + (-0.05 \cdot log_2(0.05)) + (-0.32 \cdot log_2(0.32)) = \underline{3.026 \text{ bits}}$

**Part c.** The transmission is calculated by:

$T(X,Y) = H(X) + H(Y) - H(X,Y) = 1.485 + 1.846 - 3.026 = \underline{0.305 \text{ bits}}$

# 4 Problem 4

**Part a.** I predict that in week 21, the athlete will run for 19 miles, swim for 2.5 miles and cycle for 50 miles.

**Part b.** I retrodict that the athlete will have ran for 3 miles during week one, 6 miles during week two, 9 miles during week three and 16 miles during week four.

The data for Parts a and b are shown below.

| Mask No. | Week | Run (mi) | Swim (mi) | Cycle (mi) |
|---|---|---|---|---|
| 3 | 1 | 3 | 2.5 | 90 |
| 3 | 2 | 6 | 2.0 | 100 |
| 3 | 3 | 9 | 1.5 | 30 |
| 3 | 4 | 16 | 1.0 | 40 |
| 1 | 5 | 10 | 3.0 | 50 |
| 1 | 6 | 13 | 2.5 | 60 |
| 1 | 7 | 16 | 2.0 | 70 |
| 1 | 8 | 19 | 1.5 | 80 |
| 1 | 9 | 22 | 1.0 | 90 |
| 2 | 10 | 0 | 3.0 | 100 |
| 2 | 11 | 19 | 2.5 | 30 |
| 2 | 12 | 22 | 2.0 | 40 |
| 2 | 13 | 25 | 1.5 | 50 |
| 2 | 14 | 28 | 1.0 | 60 |
| 3 | 15 | 6 | 3.0 | 70 |
| 3 | 16 | 9 | 2.5 | 80 |
| 3 | 17 | 12 | 2.0 | 90 |
| 3 | 18 | 15 | 1.5 | 100 |
| 3 | 19 | 34 | 1.0 | 30 |
| 1 | 20 | 12 | 3.0 | 40 |
| 1 | 21 | 19 | 2.5 | 50 |

**Part c.** The source system that describes these data is provided below.

1. Variable 1: Distance the athlete runs (in miles per week)
2. Variable 2: Distance the athlete swims (in miles per week)
3. Variable 3: Distance the athlete cycles (in miles per week)
4. Variable 4: The week that the athlete is recording their data (in week number after starting training session)

Variable 4 is considered to be an input variable, and Variables 1, 2, and 3 are considered to be an output variables. I have defined the system in this way, since the amount that the athlete runs/cycles/swims is a function of the week number after starting the training sessions. Since I have defined both input and output variables, the proposed system is called a "directed" system. The state set for all of the variables includes both the positive integers (for Variable 4) and the positive real numbers (for Variables 1-3), that is the State Set = $\{\mathbb{Z}^+, \mathbb{R}^+\}$ and the methodological distinction for all of the variables are "ratio" since there is a true sense of where zero is located. The state-set for this system is further defined in the table below.

| Variable | State |
|---|---|
| Run Distance | $\mathbb{R}^+$ |
| Swim Distance | $\mathbb{R}^+$ |
| Cycle Distance | $\mathbb{R}^+$ |
| Week Number | $\mathbb{Z}^+$ |

**Part d.** The data system is presented below.

| Week | Run (mi) | Swim (mi) | Cycle (mi) |
|---|---|---|---|
| 5 | 10 | 3.0 | 50 |
| 6 | 13 | 2.5 | 60 |
| 7 | 16 | 2.0 | 70 |
| 8 | 19 | 1.5 | 80 |
| 9 | 22 | 1.0 | 90 |
| 10 | 0 | 3.0 | 100 |
| 11 | 19 | 2.5 | 30 |
| 12 | 22 | 2.0 | 40 |
| 13 | 25 | 1.5 | 50 |
| 14 | 28 | 1.0 | 60 |
| 15 | 6 | 3.0 | 70 |
| 16 | 9 | 2.5 | 80 |
| 17 | 12 | 2.0 | 90 |
| 18 | 15 | 1.5 | 100 |
| 19 | 34 | 1.0 | 30 |
| 20 | 12 | 3.0 | 40 |

Using this chart as a reference, the column name called "Week" corresponds to "Variable 4" from the source system. The column name called "Run" corresponds to "Variable 1" from the source system, the column called "Swim" corresponds to "Variable 2" from the source system, and the column called "Cycle" corresponds to "Variable 3" from the source system. .

**Part e.** I propose three different masks for the generative system and each mask covers five weeks. These masks are needed to model the data being generated that describes the amount of miles that the athlete runs each week. The models that are used to generate the data for the amount that the athlete swims and cycles each week will be the same for each of the three masks. For this generative system, Mask 1 covers weeks 5 through 9, Mask 2 covers weeks 10 through 14, and Mask 3 covers weeks 15 through 20. The data generated for week 21 will come from Mask 1, and the data being generated for weeks 1 through 4 will come from Mask 3. The pseudocode for Masks 1, 2, and 3 are presented as Figures 1, 2, and 3, respectively.

In all three masks, the model for the cycle distance is the same. In particular, each week, the athlete cycles 10 miles more than the previous week; however, the week after the athlete reaches 100 miles, the athlete will cycle for 30 miles. In other words, the smallest distance that the athlete will cycle is 30 miles per week, and the largest amount that the athlete will cycle is 100 miles per week, and each week the athlete will cycle 10 miles more than the previous week.

The pattern for the swimming is also the same for all three masks. In particular, the most that the athlete will swim is three miles, and each subsequent week, the athlete will swim a half-mile less, compared to the previous week, up until the time then the athlete swims one mile. In this case, the week after the athlete swims one mile, athlete will swim for three miles, and the cycle will repeat itself.

As noted previously, the patters for the running distance falls into three cases, or masks.

1. **Mask 1**. For the first week covered by Mask 1, the athlete runs for an arbitrary distance. Based on the source system, this is assumed to be zero or more miles. For the second through forth weeks of week of Mask 1, the athlete runs 10 more miles than they did exactly five weeks prior. For the fifth week of Mask 1, the athlete runs 6 more miles than they did exactly 5 weeks prior to the week in question.
2. **Mask 2**. For the first week covered by Mask 2, the athlete runs for an arbitrary distance. Based on the source system, this is assumed to be zero or more miles. For the second through fifth weeks of week of Mask 2, the athlete runs 6 more miles than they did exactly 5 weeks prior to the week in question.
3. **Mask 3**. For the first week covered by Mask 3, the athlete runs for an arbitrary distance. Based on the source system, this is assumed to be zero or more miles. For the second through forth weeks of week of Mask 3, the athlete runs 10 fewer miles than they did exactly 5 weeks prior to the week in question. For the fifth week of Mask 3, the athlete runs 6 more miles than they did exactly 5 weeks prior to the week in question.

# 5 Problem 5

**Part a.** The number of relations for $A \times B$ is the same as determining all possible subsets of $A \times B$. Determining all possible subsets of a Cartesian product is the same as finding the cardinality of the powerset of the Cartesian product. Since $|A| = 3$ and $|B| = 1$, the total number of relations is $2^{|A| \cdot |B|} = 8$.

**Part b.** See the Appendix for Problem 5 for all of the elements. Since $|A| = 3$ and $|B| = 1$, the total number of elements is $2^{|A| \cdot |B|} = 8$ (Page 30).

**Part c.** Parts 1 and 2 are related in that they are asking for the <u>same information</u>. When asking for the total number of relations between $A \times B$ you are asking for the total number of subsets of that Cartesian product, which is the same as asking for the powerset of the Cartesian product.

**Part d.** Since $|A| = 3$, the total number of elements for $\mathcal{P}(\mathcal{P}(A))$ is 256, determined by $(2^{2^3})$. See the Appendix for Problem 5 for all of the elements (Page 32).

**Part e.** See the Appendix for Problem 5 for all of the elements. The total number of elements is 16, as determined by $(2^{2^{2^1}})$ (Page 31).

# 6 Problem 6

To solve this problem, I made a few assumptions.

1. In both $S_1$ and $S_2$ I assumed that the relations were reflexive. For instance, for $S_1$ and $S_2$, this means that any drug can be taken with itself, and any nurse can work with themself, respectively.

2. In both $S_1$ and $S_2$ I assumed that the relations were symmetric. For instance, for $S_1$ this means that taking drug $a$, and $b$ is the same as taking $b$, and $a$. For $S_2$ this means that if if nurse 1 and nurse 2 are on a shift, this is the same as having nurse 2 and nurse 1 on the same shift (Page 39 and 40).

Once those assumptions were made, I proceeded to creating a directed graph of both systems so that I could take an inventory of the edges. See Appendix.

The edge inventory for System $S_1$ is below.

| Node | Reflexive | In | Out | Two-Way |
|---|---|---|---|---|
| 1 | Y | 0 | 0 | 3 |
| 2 | Y | 0 | 0 | 5 |
| 3 | Y | 0 | 0 | 6 |
| 4 | Y | 0 | 0 | 4 |
| 5 | Y | 0 | 0 | 5 |
| 6 | Y | 0 | 0 | 4 |
| 7 | Y | 0 | 0 | 3 |
| 8 | Y | 0 | 0 | 4 |

The edge inventory for System $S_2$ is below.

| Node | Reflexive | In | Out | Two-Way |
|---|---|---|---|---|
| a | Y | 0 | 0 | 4 |
| b | Y | 0 | 0 | 5 |
| c | Y | 0 | 0 | 4 |
| d | Y | 0 | 0 | 3 |
| e | Y | 0 | 0 | 4 |
| f | Y | 0 | 0 | 6 |
| g | Y | 0 | 0 | 3 |
| h | Y | 0 | 0 | 5 |

The possible node matches for $S_1$ and $S_2$ is shown below.

| Node | Possible Match |
|---|---|
| a | 4/6/8 |
| b | 2/5 |
| c | 4/6/8 |
| d | 1/7 |
| e | 4/6/8 |
| f | 3 |
| g | 1/7 |
| h | 2/5 |

To start, I determined, by inspection of the graphs in the the Appendix for Problem 6, that there is a mapping between $f \to 3$, $b \to 2$, $h \to 5$ (Page 40 and 41). Also, by inspection, I noticed that there are two possible choices for the mapping of $g/d$. In particular, I noticed that the following choices are possible: $d \to 1$, $g \to 1$, $d \to 7$, or $g \to 7$

Because I am assuming that the relations are symmetric, the corresponding pair is assumed to be part of the set as well, however, I am not including the elements for clarity.

Ordered pairs for the relations in $S_1$: (1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 7), (3, 8),
(4, 6), (5, 6), (5, 7), (5, 8), (6, 8), (7, 8)

Ordered pairs for the relations in $S_2$: (a, c), (a, d), (a, f), (a, h), (b, c), (b, e), (b, f), (b, g), (b, h), (c, e), (c, h),
(d, f), (d, h), (e, f), (e, g), (f, g), (f, h)

When it came to the mapping of $g/d$, I initially started with $d \to 1$ and $g \to 7$, however, when doing so, I noticed that for the element $(1, 2)$ and $(2, 1)$ there was no element $(d, b)$ and $(b, d)$ in the relation. I did notice however, that both

$(g, b)$ and $(b, g)$ were in the relation for $S_1$ and so I proceeded to assign $g \to 1$ and $d \to 7$. Again, by inspection, I noticed that for every time there was a 4 in the relations of $S_2$, I found a corresponding $e$ in $S_1$. Finally, this leaves, elements, $a$, $c$, 6, and 8. By luck, when I arbitrarily assigned $a \to 8$ and $c \to 6$, and checked the relations in $S_1$, they seemed to match what was in $S_2$. This affords the proposed bijecive function:

1. $b \to 2$
2. $f \to 3$
3. $h \to 5$
4. $d \to 7$
5. $g \to 1$
6. $e \to 4$
7. $a \to 8$
8. $c \to 6$

Gratifyingly, when using this map and checking all 42 (34 symmetric + 8 reflexive) relations, as shown in the Appendix for Problem 6, I did verify that systems $S_1$ and $S_2$ are isomorphic (Page 41).

**Algorithm 1** Mask 1

```
Running Distance for week i (r_distance_week_i)
if i = 1 then
    return Random_Positive_Integer
else if i = 5 then
    return r_distance_week_{i-5} + 6
else
    return r_distance_week_{i-5} + 10
end if
Swimming Distance for week i (s_distance_week_i)
if s_distance_week_{i-1} ≤ 1 then
    return 3
else
    return s_distance_week_{i-1} − 0.5
end if
Cycling Distance for week i (c_distance_week_i)
if c_distance_week_{i-1} < 100 then
    return c_distance_week_{i-1} + 10
else
    return 30
end if
```

Figure 1: Mask 1

**Algorithm 2** Mask 2

Running Distance for *week i* ($r\_distance\_week_i$)
**if** $i = 1$ **then**
  **return** Random_Positive_Integer
**else**
  **return** $r\_distance\_week_{i-5} + 6$
**end if**
Swimming Distance for *week i* ($s\_distance\_week_i$)
**if** $s\_distance\_week_{i-1} \leq 1$ **then**
  **return** 3
**else**
  **return** $s\_distance\_week_{i-1} - 0.5$
**end if**
Cycling Distance for *week i* ($c\_distance\_week_i$)
**if** $c\_distance\_week_{i-1} < 100$ **then**
  **return** $c\_distance\_week_{i-1} + 10$
**else**
  **return** 30
**end if**

Figure 2: Mask 2

**Algorithm 3** Mask 3

Running Distance for *week i* ($r\_distance\_week_i$)<br>
**if** $i = 1$ **then**<br>
    **return** Random_Positive_Integer<br>
**else if** $i = 5$ **then**<br>
    **return** $r\_distance\_week_{i-5} - 6$<br>
**else**<br>
    **return** $r\_distance\_week_{i-5} - 10$<br>
**end if**<br>
Swimming Distance for *week i* ($s\_distance\_week_i$)<br>
**if** $s\_distance\_week_{i-1} \leq 1$ **then**<br>
    **return** 3<br>
**else**<br>
    **return** $s\_distance\_week_{i-1} - 0.5$<br>
**end if**<br>
Cycling Distance for *week i* ($c\_distance\_week_i$)<br>
**if** $c\_distance\_week_{i-1} < 100$ **then**<br>
    **return** $c\_distance\_week_{i-1} + 10$<br>
**else**<br>
    **return** 30<br>
**end if**

Figure 3: Mask 3

# 7 Appendix: Problem 01

$h^{-1}(R_2) \longrightarrow R_1$

(4,4) ->{(a,a)}

(4,2) ->{(a,b)}

(4,1) ->{(a,c), (a,d), (a,e), (a,g), (a,i)} need 1

(2,2) ->{(b,b)}

(2,1) ->{(b,c), (b,d), (b,e), (b,g), (b,i)} need 1

(1,1) ->{(c,c), (c,d), (c,e), (c,g), (c,i),
(d,c), (d,d), (d,e), (d,g), (d,i),
(e,c), (e,d), (e,e), (e,g),
(e,i), (g,c), (g,d), (g,e),
(g,g), (g,i), (i,c), (i,d),
(i,e), (i,g), (i,i)} only need 1

(1,2) ->{(c,b), (d,b), (e,b), (g,b), (i,b)} only need 1

(1,3) ->{(c,j), (d,j), (e,j), (g,j), (i,j)} only need 1

(5,4) ->{(f,a), (h,a), (k,a)} only need 1

(5,5)->{(f,f), (f,h), (f,k), (h,f), (h,h), (h,k), (k,f), (k,h), (k,k)} only need 1

(3,1) ->{(j,c), (j,d), (j,e), (j,g), (j,i)} only need 1

(3,5) -> {(j,f), (j,h), (j,k)} only need 1

(3,3) -> {(j,j)}

S2 is a strong Homomorphism of S1

w/. mapping.

a → 4
b → 2
c → 1
d → 1
e → 1
f → 5
g → 1
h → 5
i → 1
j → 3
k → 5

$h(R_1) \longrightarrow R_2$

problem 1A

```
(a,a) -> (4,4)
(a,b) -> (4,2)
(a,i) -> (4,1)
(b,b) -> (2,2)
(b,c) -> (2,1)
(b,d) -> (2,1)
(c,c) -> (1,1)
(c,g) -> (1,1)
(c,i) -> (1,1)
(d,b) -> (1,2)
(d,d) -> (1,1)
(e,c) -> (1,1)
(e,e) -> (1,1)
(e,j) -> (1,3)
(f,a) -> (5,4)
(f,h) -> (5,5)
(f,k) -> (5,5)
(g,c) -> (1,1)
(g,d) -> (1,1)
(g,i) -> (1,1)
(h,f) -> (5,5)
(h,h) -> (5,5)
(h,k) -> (5,5)
(i,c) -> (1,1)
(i,g) -> (1,1)
(i,i) -> (1,1)
(i,j) -> (1,3)
(j,e) -> (3,1)
(j,h) -> (3,5)
(j,j) -> (3,3)
(k,a) -> (5,4)
(k,f) -> (5,5)
(k,h) -> (5,5)
(k,k) -> (5,5)
```

# Problem 01 Python Code

November 29, 2020

[1]:
```
import pandas as pd
import numpy as np
from itertools import product, chain, combinations
import itertools
from collections import defaultdict
import time
import pickle

# Systems
R1 = ['aa', 'ab', 'ai',
'bb', 'bc', 'bd',
'cc', 'cg', 'ci',
'db', 'dd',
'ec', 'ee', 'ej',
'fa', 'fh', 'fk',
'gc', 'gd', 'gi',
'hf', 'hh', 'hk',
'ic','ig', 'ii', 'ij',
'je', 'jh', 'jj',
'ka', 'kf', 'kh', 'kk']

R2 = ['11', '12', '13', '21',
      '22', '31', '33', '35',
      '41', '42', '44', '54', '55']

#Load Pickle Files for Total Sample Space
start_time = time.time()
with open('sampleSpace.pkl', 'rb') as f:
    SAMPLE_SPACE = pickle.load(f)
end_time = time.time()
print('Elapsed Time for loading `sampleSpace.pkl`: {}'.
 ↪format(end_time-start_time))

#Load Pickle Files for Homomorphic Sample Space
start_time = time.time()
with open('homomorphicSpace.pkl', 'rb') as f:
    HOMOMORPHIC_SPACE = pickle.load(f)
```

23

```python
end_time = time.time()
print('Elapsed Time for loading `homomorphicSpace.pkl`: {}'.format(end_time-start_time))
```

```
Elapsed Time for loading `sampleSpace.pkl`: 107.2454240322113
Elapsed Time for loading `homomorphicSpace.pkl`: 0.19091582298278809
```

[ ]:
```python
#Find all Possible Mappings
start_time = time.time()
a = [list(each) for each in itertools.product('12345', repeat=11)]
space = [dict(zip([each for each in 'abcdefghijk'], i)) for i in a]
end_time = time.time()
print('Elapsed Time: {}'.format(end_time-start_time))
```

[2]:
```python
#Total Possible Mappings Between S1 and S2
len(SAMPLE_SPACE)
```

[2]: 48828125

[3]:
```python
def is_homomorphism(x, R1, R2):
    '''
    Returns True if R2 is a subset of initial_decode.
    '''
    initial_decode = [''.join(each) for each in [[x[each[0]], x[each[1]]] for each in R1]]
    return set(initial_decode) == set(R2)


def is_strong_homomorphism(x, R1, R2):
    data = sorted(list(zip(list(x.values()), list(x.keys()))))
    d = defaultdict(list)
    for r1, r2 in data:
        d[r1].append(r2)
    h_x_inv = dict(d)

    initial_decode = [''.join(each) for each in [[x[each[0]], x[each[1]]] for each in R1]]

    second_pos = []
    nested_decode = []
    dict_decode = {}

    for eachi in initial_decode:
        a = [h_x_inv[each[0][0]] for each in eachi]
        cart_prod_i = [element for element in itertools.product(a[0], a[1])]
        d = [''.join(each) for each in cart_prod_i]
        nested_decode.append(d)
```

```
        dict_decode[eachi] = d
    aaa = [i for j in [dict_decode[each] for each in R2] for i in j]
    return set(R1).issubset(aaa)
```

[4]:
```
#Find All Homomorphisms
start_time = time.time()

homomorphic_solutions = []
except_index = []
for e in SAMPLE_SPACE:
    try:
        if is_homomorphism(x=e, R1=R1, R2=R2):
            homomorphic_solutions.append(e)
        else:
            pass
    except:
        except_index.append(e)
        print('Exception: {}'.format(e))

end_time = time.time()
print('Elapsed Time: {:.3f} min'.format((end_time-start_time)/60))
print(len(homomorphic_solutions))
print(except_index)
```

```
Elapsed Time: 13.885 min
14
[]
```

[5]:
```
strong_homomorphic_solutions = []
except_index = []
for e in HOMOMORPHIC_SPACE:
    try:
        if is_strong_homomorphism(x=e, R1=R1, R2=R2):
            strong_homomorphic_solutions.append(e)
        else:
            pass
    except:
        except_index.append(e)
        print('Exception: {}'.format(e))

end_time = time.time()
print('Elapsed Time: {:.3f} min'.format((end_time-start_time)/60))
print(len(strong_homomorphic_solutions))
print(except_index)
```

```
Elapsed Time: 14.084 min
14
```

```
[]
```

```
[6]: #Example of 1 of 14 Strong Homomorphic Solutions
example_map = strong_homomorphic_solutions[0]
example_map
```

```
[6]: {'a': '4',
 'b': '2',
 'c': '1',
 'd': '1',
 'e': '1',
 'f': '5',
 'g': '1',
 'h': '5',
 'i': '1',
 'j': '3',
 'k': '5'}
```

```
[18]: #Forward Decode
initial_decode = [''.join(each) for each in [[example_map[each[0]],␣
 ↪example_map[each[1]]] for each in R1]]
forward_decode = dict(zip(R1, initial_decode))
forward_decode
```

```
[18]: {'aa': '44',
 'ab': '42',
 'ai': '41',
 'bb': '22',
 'bc': '21',
 'bd': '21',
 'cc': '11',
 'cg': '11',
 'ci': '11',
 'db': '12',
 'dd': '11',
 'ec': '11',
 'ee': '11',
 'ej': '13',
 'fa': '54',
 'fh': '55',
 'fk': '55',
 'gc': '11',
 'gd': '11',
 'gi': '11',
 'hf': '55',
 'hh': '55',
 'hk': '55',
```

```
 'ic': '11',
 'ig': '11',
 'ii': '11',
 'ij': '13',
 'je': '31',
 'jh': '35',
 'jj': '33',
 'ka': '54',
 'kf': '55',
 'kh': '55',
 'kk': '55'}
```

[17]:
```python
#Reverse Decode
second_pos = []
nested_decode = []
dict_decode = {}

for eachi in initial_decode:
    a = [h_x_inv[each[0][0]] for each in eachi]
    cart_prod_i = [element for element in itertools.product(a[0], a[1])]
    d = [''.join(each) for each in cart_prod_i]
    nested_decode.append(d)
    dict_decode[eachi] = d
reverse_decode = dict_decode
reverse_decode
```

[17]:
```
{'44': ['aa'],
 '42': ['ab'],
 '41': ['ac', 'ad', 'ae', 'ag', 'ai'],
 '22': ['bb'],
 '21': ['bc', 'bd', 'be', 'bg', 'bi'],
 '11': ['cc',
  'cd',
  'ce',
  'cg',
  'ci',
  'dc',
  'dd',
  'de',
  'dg',
  'di',
  'ec',
  'ed',
  'ee',
  'eg',
  'ei',
  'gc',
```

```
 'gd',
 'ge',
 'gg',
 'gi',
 'ic',
 'id',
 'ie',
 'ig',
 'ii'],
'12': ['cb', 'db', 'eb', 'gb', 'ib'],
'13': ['cj', 'dj', 'ej', 'gj', 'ij'],
'54': ['fa', 'ha', 'ka'],
'55': ['ff', 'fh', 'fk', 'hf', 'hh', 'hk', 'kf', 'kh', 'kk'],
'31': ['jc', 'jd', 'je', 'jg', 'ji'],
'35': ['jf', 'jh', 'jk'],
'33': ['jj']}
```

[ ]:
```
#Initial Code to Find all Homomorphisms
start_time = time.time()

homomorphic_solutions = []
except_index = []
for e in SAMPLE_SPACE:
    try:
        if is_homomorphism(x=e, R1=R1, R2=R2):
            homomorphic_solutions.append(e)
        else:
            pass
    except:
        except_index.append(e)
        print('Exception: {}'.format(e))

end_time = time.time()
print('Elapsed Time: {:.3f} min'.format((end_time-start_time)/60))
print(len(homomorphic_solutions))
print(except_index)
```

[ ]:
```
#Save var `homomorphic_solutions` as pkl file.
with open('homomorphicSpace.pkl', 'wb') as f:
    pickle.dump(homomorphic_solutions, f)
```

# 8 Appendix: Problem 05

Problem 5b.

8 total elements.

P(A x B) =

1. {empty_set,
2. {{1, 'beta'}},
3. {{2, 'beta'}},
4. {{3, 'beta'}},
5. {{1, 'beta'}, {2, 'beta'}},
6. {{1, 'beta'}, {3, 'beta'}},
7. {{2, 'beta'}, {3, 'beta'}},
8. {{1, 'beta'}, {2, 'beta'}, {3, 'beta'}}}

Problem 5e.

16 total elements.

P(P(P(A))) =

1. {empty_set,
2. {empty_set},
3. {{empty_set}},
4. {{{'beta'}}},
5. {{empty_set, {beta'}}},
6. {empty_set, {empty_set}},
7. {empty_set, {{'beta'}}},
8. {empty_set, {empty_set, {'beta'}}},
9. {{empty_set}, {{'beta'}}},
10. {{empty_set}, {empty_set, {'beta'}}},
11. {{{'beta'}}, {empty_set, {'beta'}}},
12. {empty_set, {empty_set}, {{'beta'}}},
13. {empty_set, {empty_set}, {empty_set, {'beta'}}},
14. {empty_set, {{'beta',}}, {empty_set, {'beta'}}},
15. {{empty_set}, {{'beta'}}, {empty_set, {'beta'}}},
16. {empty_set, {empty_set}, {{'beta'}}, {empty_set, {'beta'}}}}

Problem 5d.

256 total elements.

P(P(P(A))) =

1. {empty_set,
2. {empty_set},
3. {{1}},
4. {{2}},
5. {{3}},
6. {{1, 2}},
7. {{1, 3}},
8. {{2, 3}},
9. {{1, 2, 3}},
10. {empty_set, {1}},
11. {empty_set, {2}},
12. {empty_set, {3}},
13. {empty_set, {1, 2}},
14. {empty_set, {1, 3}},
15. {empty_set, {2, 3}},
16. {empty_set, {1, 2, 3}},
17. {{1}, {2}},
18. {{1}, {3}},
19. {{1}, {1, 2}},
20. {{1}, {1, 3}},
21. {{1}, {2, 3}},
22. {{1}, {1, 2, 3}},
23. {{2}, {3}},
24. {{2}, {1, 2}},
25. {{2}, {1, 3}},
26. {{2}, {2, 3}},
27. {{2}, {1, 2, 3}},
28. {{3}, {1, 2}},
29. {{3}, {1, 3}},
30. {{3}, {2, 3}},
31. {{3}, {1, 2, 3}},
32. {{1, 2}, {1, 3}},
33. {{1, 2}, {2, 3}},
34. {{1, 2}, {1, 2, 3}},
35. {{1, 3}, {2, 3}},
36. {{1, 3}, {1, 2, 3}},
37. {{2, 3}, {1, 2, 3}},
38. {empty_set, {1}, {2}},
39. {empty_set, {1}, {3}},
40. {empty_set, {1}, {1, 2}},
41. {empty_set, {1}, {1, 3}},
42. {empty_set, {1}, {2, 3}},
43. {empty_set, {1}, {1, 2, 3}},
44. {empty_set, {2}, {3}},
45. {empty_set, {2}, {1, 2}},
46. {empty_set, {2}, {1, 3}},

47. {empty_set, {2}, {2, 3}},
48. {empty_set, {2}, {1, 2, 3}},
49. {empty_set, {3}, {1, 2}},
50. {empty_set, {3}, {1, 3}},
51. {empty_set, {3}, {2, 3}},
52. {empty_set, {3}, {1, 2, 3}},
53. {empty_set, {1, 2}, {1, 3}},
54. {empty_set, {1, 2}, {2, 3}},
55. {empty_set, {1, 2}, {1, 2, 3}},
56. {empty_set, {1, 3}, {2, 3}},
57. {empty_set, {1, 3}, {1, 2, 3}},
58. {empty_set, {2, 3}, {1, 2, 3}},
59. {{1}, {2}, {3}},
60. {{1}, {2}, {1, 2}},
61. {{1}, {2}, {1, 3}},
62. {{1}, {2}, {2, 3}},
63. {{1}, {2}, {1, 2, 3}},
64. {{1}, {3}, {1, 2}},
65. {{1}, {3}, {1, 3}},
66. {{1}, {3}, {2, 3}},
67. {{1}, {3}, {1, 2, 3}},
68. {{1}, {1, 2}, {1, 3}},
69. {{1}, {1, 2}, {2, 3}},
70. {{1}, {1, 2}, {1, 2, 3}},
71. {{1}, {1, 3}, {2, 3}},
72. {{1}, {1, 3}, {1, 2, 3}},
73. {{1}, {2, 3}, {1, 2, 3}},
74. {{2}, {3}, {1, 2}},
75. {{2}, {3}, {1, 3}},
76. {{2}, {3}, {2, 3}},
77. {{2}, {3}, {1, 2, 3}},
78. {{2}, {1, 2}, {1, 3}},
79. {{2}, {1, 2}, {2, 3}},
80. {{2}, {1, 2}, {1, 2, 3}},
81. {{2}, {1, 3}, {2, 3}},
82. {{2}, {1, 3}, {1, 2, 3}},
83. {{2}, {2, 3}, {1, 2, 3}},
84. {{3}, {1, 2}, {1, 3}},
85. {{3}, {1, 2}, {2, 3}},
86. {{3}, {1, 2}, {1, 2, 3}},
87. {{3}, {1, 3}, {2, 3}},
88. {{3}, {1, 3}, {1, 2, 3}},
89. {{3}, {2, 3}, {1, 2, 3}},
90. {{1, 2}, {1, 3}, {2, 3}},
91. {{1, 2}, {1, 3}, {1, 2, 3}},
92. {{1, 2}, {2, 3}, {1, 2, 3}},
93. {{1, 3}, {2, 3}, {1, 2, 3}},
94. {empty_set, {1}, {2}, {3}},
95. {empty_set, {1}, {2}, {1, 2}},
96. {empty_set, {1}, {2}, {1, 3}},
97. {empty_set, {1}, {2}, {2, 3}},
98. {empty_set, {1}, {2}, {1, 2, 3}},

99. {empty_set, {1}, {3}, {1, 2}},
100.{empty_set, {1}, {3}, {1, 3}},
101.{empty_set, {1}, {3}, {2, 3}},
102.{empty_set, {1}, {3}, {1, 2, 3}},
103.{empty_set, {1}, {1, 2}, {1, 3}},
104.{empty_set, {1}, {1, 2}, {2, 3}},
105.{empty_set, {1}, {1, 2}, {1, 2, 3}},
106.{empty_set, {1}, {1, 3}, {2, 3}},
107.{empty_set, {1}, {1, 3}, {1, 2, 3}},
108.{empty_set, {1}, {2, 3}, {1, 2, 3}},
109.{empty_set, {2}, {3}, {1, 2}},
110.{empty_set, {2}, {3}, {1, 3}},
111.{empty_set, {2}, {3}, {2, 3}},
112.{empty_set, {2}, {3}, {1, 2, 3}},
113.{empty_set, {2}, {1, 2}, {1, 3}},
114.{empty_set, {2}, {1, 2}, {2, 3}},
115.{empty_set, {2}, {1, 2}, {1, 2, 3}},
116.{empty_set, {2}, {1, 3}, {2, 3}},
117.{empty_set, {2}, {1, 3}, {1, 2, 3}},
118.{empty_set, {2}, {2, 3}, {1, 2, 3}},
119.{empty_set, {3}, {1, 2}, {1, 3}},
120.{empty_set, {3}, {1, 2}, {2, 3}},
121.{empty_set, {3}, {1, 2}, {1, 2, 3}},
122.{empty_set, {3}, {1, 3}, {2, 3}},
123.{empty_set, {3}, {1, 3}, {1, 2, 3}},
124.{empty_set, {3}, {2, 3}, {1, 2, 3}},
125.{empty_set, {1, 2}, {1, 3}, {2, 3}},
126.{empty_set, {1, 2}, {1, 3}, {1, 2, 3}},
127.{empty_set, {1, 2}, {2, 3}, {1, 2, 3}},
128.{empty_set, {1, 3}, {2, 3}, {1, 2, 3}},
129.{{1}, {2}, {3}, {1, 2}},
130.{{1}, {2}, {3}, {1, 3}},
131.{{1}, {2}, {3}, {2, 3}},
132.{{1}, {2}, {3}, {1, 2, 3}},
133.{{1}, {2}, {1, 2}, {1, 3}},
134.{{1}, {2}, {1, 2}, {2, 3}},
135.{{1}, {2}, {1, 2}, {1, 2, 3}},
136.{{1}, {2}, {1, 3}, {2, 3}},
137.{{1}, {2}, {1, 3}, {1, 2, 3}},
138.{{1}, {2}, {2, 3}, {1, 2, 3}},
139.{{1}, {3}, {1, 2}, {1, 3}},
140.{{1}, {3}, {1, 2}, {2, 3}},
141.{{1}, {3}, {1, 2}, {1, 2, 3}},
142.{{1}, {3}, {1, 3}, {2, 3}},
143.{{1}, {3}, {1, 3}, {1, 2, 3}},
144.{{1}, {3}, {2, 3}, {1, 2, 3}},
145.{{1}, {1, 2}, {1, 3}, {2, 3}},
146.{{1}, {1, 2}, {1, 3}, {1, 2, 3}},
147.{{1}, {1, 2}, {2, 3}, {1, 2, 3}},
148.{{1}, {1, 3}, {2, 3}, {1, 2, 3}},
149.{{2}, {3}, {1, 2}, {1, 3}},
150.{{2}, {3}, {1, 2}, {2, 3}},

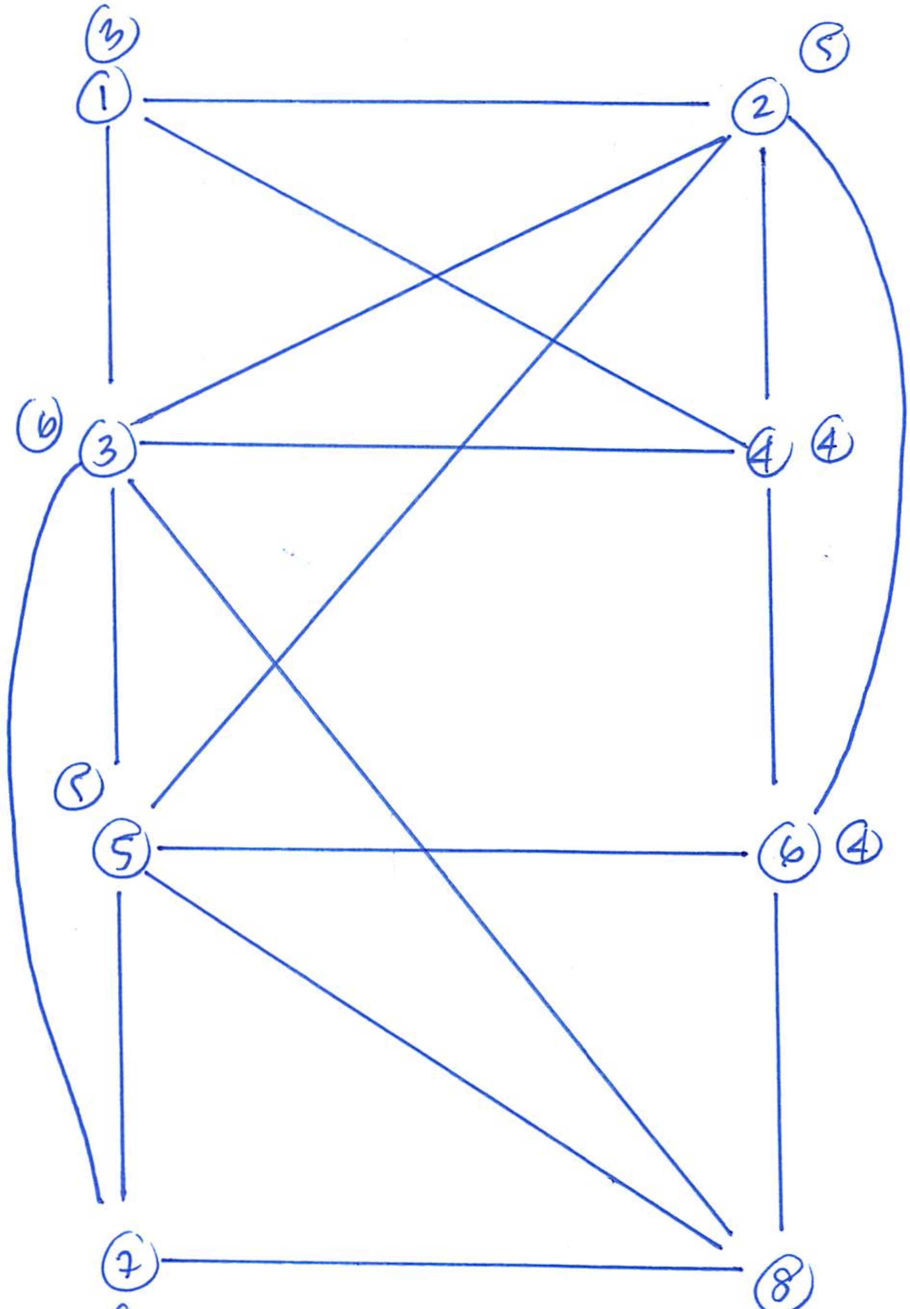
151.{{2}, {3}, {1, 2}, {1, 2, 3}},
152.{{2}, {3}, {1, 3}, {2, 3}},
153.{{2}, {3}, {1, 3}, {1, 2, 3}},
154.{{2}, {3}, {2, 3}, {1, 2, 3}},
155.{{2}, {1, 2}, {1, 3}, {2, 3}},
156.{{2}, {1, 2}, {1, 3}, {1, 2, 3}},
157.{{2}, {1, 2}, {2, 3}, {1, 2, 3}},
158.{{2}, {1, 3}, {2, 3}, {1, 2, 3}},
159.{{3}, {1, 2}, {1, 3}, {2, 3}},
160.{{3}, {1, 2}, {1, 3}, {1, 2, 3}},
161.{{3}, {1, 2}, {2, 3}, {1, 2, 3}},
162.{{3}, {1, 3}, {2, 3}, {1, 2, 3}},
163.{{1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
164.{empty_set, {1}, {2}, {3}, {1, 2}},
165.{empty_set, {1}, {2}, {3}, {1, 3}},
166.{empty_set, {1}, {2}, {3}, {2, 3}},
167.{empty_set, {1}, {2}, {3}, {1, 2, 3}},
168.{empty_set, {1}, {2}, {1, 2}, {1, 3}},
169.{empty_set, {1}, {2}, {1, 2}, {2, 3}},
170.{empty_set, {1}, {2}, {1, 2}, {1, 2, 3}},
171.{empty_set, {1}, {2}, {1, 3}, {2, 3}},
172.{empty_set, {1}, {2}, {1, 3}, {1, 2, 3}},
173.{empty_set, {1}, {2}, {2, 3}, {1, 2, 3}},
174.{empty_set, {1}, {3}, {1, 2}, {1, 3}},
175.{empty_set, {1}, {3}, {1, 2}, {2, 3}},
176.{empty_set, {1}, {3}, {1, 2}, {1, 2, 3}},
177.{empty_set, {1}, {3}, {1, 3}, {2, 3}},
178.{empty_set, {1}, {3}, {1, 3}, {1, 2, 3}},
179.{empty_set, {1}, {3}, {2, 3}, {1, 2, 3}},
180.{empty_set, {1}, {1, 2}, {1, 3}, {2, 3}},
181.{empty_set, {1}, {1, 2}, {1, 3}, {1, 2, 3}},
182.{empty_set, {1}, {1, 2}, {2, 3}, {1, 2, 3}},
183.{empty_set, {1}, {1, 3}, {2, 3}, {1, 2, 3}},
184.{empty_set, {2}, {3}, {1, 2}, {1, 3}},
185.{empty_set, {2}, {3}, {1, 2}, {2, 3}},
186.{empty_set, {2}, {3}, {1, 2}, {1, 2, 3}},
187.{empty_set, {2}, {3}, {1, 3}, {2, 3}},
188.{empty_set, {2}, {3}, {1, 3}, {1, 2, 3}},
189.{empty_set, {2}, {3}, {2, 3}, {1, 2, 3}},
190.{empty_set, {2}, {1, 2}, {1, 3}, {2, 3}},
191.{empty_set, {2}, {1, 2}, {1, 3}, {1, 2, 3}},
192.{empty_set, {2}, {1, 2}, {2, 3}, {1, 2, 3}},
193.{empty_set, {2}, {1, 3}, {2, 3}, {1, 2, 3}},
194.{empty_set, {3}, {1, 2}, {1, 3}, {2, 3}},
195.{empty_set, {3}, {1, 2}, {1, 3}, {1, 2, 3}},
196.{empty_set, {3}, {1, 2}, {2, 3}, {1, 2, 3}},
197.{empty_set, {3}, {1, 3}, {2, 3}, {1, 2, 3}},
198.{empty_set, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
199.{{1}, {2}, {3}, {1, 2}, {1, 3}},
200.{{1}, {2}, {3}, {1, 2}, {2, 3}},
201.{{1}, {2}, {3}, {1, 2}, {1, 2, 3}},
202.{{1}, {2}, {3}, {1, 3}, {2, 3}},

203.{{1}, {2}, {3}, {1, 3}, {1, 2, 3}},
204.{{1}, {2}, {3}, {2, 3}, {1, 2, 3}},
205.{{1}, {2}, {1, 2}, {1, 3}, {2, 3}},
206.{{1}, {2}, {1, 2}, {1, 3}, {1, 2, 3}},
207.{{1}, {2}, {1, 2}, {2, 3}, {1, 2, 3}},
208.{{1}, {2}, {1, 3}, {2, 3}, {1, 2, 3}},
209.{{1}, {3}, {1, 2}, {1, 3}, {2, 3}},
210.{{1}, {3}, {1, 2}, {1, 3}, {1, 2, 3}},
211.{{1}, {3}, {1, 2}, {2, 3}, {1, 2, 3}},
212.{{1}, {3}, {1, 3}, {2, 3}, {1, 2, 3}},
213.{{1}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
214.{{2}, {3}, {1, 2}, {1, 3}, {2, 3}},
215.{{2}, {3}, {1, 2}, {1, 3}, {1, 2, 3}},
216.{{2}, {3}, {1, 2}, {2, 3}, {1, 2, 3}},
217.{{2}, {3}, {1, 3}, {2, 3}, {1, 2, 3}},
218.{{2}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
219.{{3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
220.{empty_set, {1}, {2}, {3}, {1, 2}, {1, 3}},
221.{empty_set, {1}, {2}, {3}, {1, 2}, {2, 3}},
222.{empty_set, {1}, {2}, {3}, {1, 2}, {1, 2, 3}},
223.{empty_set, {1}, {2}, {3}, {1, 3}, {2, 3}},
224.{empty_set, {1}, {2}, {3}, {1, 3}, {1, 2, 3}},
225.{empty_set, {1}, {2}, {3}, {2, 3}, {1, 2, 3}},
226.{empty_set, {1}, {2}, {1, 2}, {1, 3}, {2, 3}},
227.{empty_set, {1}, {2}, {1, 2}, {1, 3}, {1, 2, 3}},
228.{empty_set, {1}, {2}, {1, 2}, {2, 3}, {1, 2, 3}},
229.{empty_set, {1}, {2}, {1, 3}, {2, 3}, {1, 2, 3}},
230.{empty_set, {1}, {3}, {1, 2}, {1, 3}, {2, 3}},
231.{empty_set, {1}, {3}, {1, 2}, {1, 3}, {1, 2, 3}},
232.{empty_set, {1}, {3}, {1, 2}, {2, 3}, {1, 2, 3}},
233.{empty_set, {1}, {3}, {1, 3}, {2, 3}, {1, 2, 3}},
234.{empty_set, {1}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
235.{empty_set, {2}, {3}, {1, 2}, {1, 3}, {2, 3}},
236.{empty_set, {2}, {3}, {1, 2}, {1, 3}, {1, 2, 3}},
237.{empty_set, {2}, {3}, {1, 2}, {2, 3}, {1, 2, 3}},
238.{empty_set, {2}, {3}, {1, 3}, {2, 3}, {1, 2, 3}},
239.{empty_set, {2}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
240.{empty_set, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
241.{{1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}},
242.{{1}, {2}, {3}, {1, 2}, {1, 3}, {1, 2, 3}},
243.{{1}, {2}, {3}, {1, 2}, {2, 3}, {1, 2, 3}},
244.{{1}, {2}, {3}, {1, 3}, {2, 3}, {1, 2, 3}},
245.{{1}, {2}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
246.{{1}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
247.{{2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
248.{empty_set, {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}},
249.{empty_set, {1}, {2}, {3}, {1, 2}, {1, 3}, {1, 2, 3}},
250.{empty_set, {1}, {2}, {3}, {1, 2}, {2, 3}, {1, 2, 3}},
251.{empty_set, {1}, {2}, {3}, {1, 3}, {2, 3}, {1, 2, 3}},
252.{empty_set, {1}, {2}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
253.{empty_set, {1}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
254.{empty_set, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},

255.{{1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}},
256.{empty_set, {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}}}

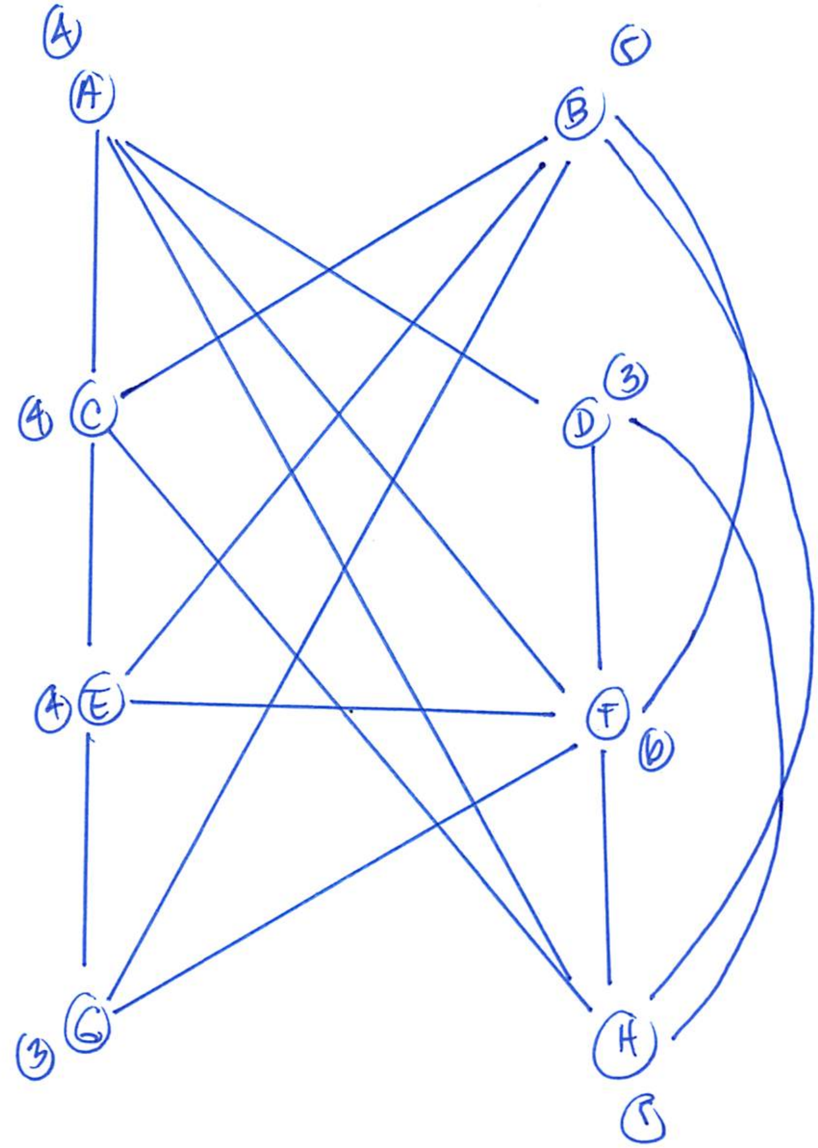
# 9 Appendix: Problem 06

S1




* All EDGES are symmetric
* All NODES are reflexive

52




* all edges are symmetric

* all nodes are reflexive.

(1,2) — (g,b) — (1,2)
(2,1) — (b,g) — (2,1)
(1,3) — (g,f) — (1,3)
(3,1) → (f,g) — (3,1)
(1,4) — (g,e) — (1,4)
(4,1) — (e,g) — (4,1)
(2,3) — (b,f) — (2,3)
(3,2) — (f,b) — (3,2)
(2,4) — (b,e) — (2,4)
(4,2) — (e,b) — (4,2)
(2,5) — (b,h) — (2,5)
(5,2) — (h,b) — (5,2)
(2,6) → (b,c) — (2,6)
(6,2) → (c,b) — (6,2)
(3,4) — (f,e) — (3,4)
(4,3) — (e,f) — (4,3)
(3,5) — (f,h) — (3,5)
(5,3) — (h,f) — (5,3)
(3,7) — (f,d) — (3,7)
(7,3) — (d,f) — (7,3)
(3,8) — (f,a) — (3,8)
(8,3) — (g,f) — (8,3)

(4,6) — (e,c) — (4,6)
(6,4) — (c,e) — (6,4)
(5,6) — (h,c) — (5,6)
(6,5) — (c,h) — (6,5)
(5,7) — (h,d) — (5,7)
(7,5) — (d,h) — (7,5)
(5,8) — (h,a) — (5,8)
(8,5) — (a,h) — (8,5)
(6,8) — (c,a) — (6,8)
(8,6) — (a,c) — (8,6)
(7,8) — (d,a) — (7,8)
(8,7) — (a,d) — (8,7)
(1,1) — (g,g) — (1,1)
(2,2) — (b,b) (2,2)
(3,3) — (f,f) — (3,3)
(4,4) — (e,e) — (4,4)
(5,5) — (h,h) — (5,5)
(6,6) — (c,c) — (6,6)
(7,7) — (d,d) — (7,7)
(8,8) — (a,a) — (8,8)



$S_1$ is isomorphic to $S_2$.