

1. I've posted on the website a compendium of all my suggestions for term or master's paper topics from the drafts of class outlines that I've completed so far. It will grow as I work on the remaining outlines and brainstorm additional topics.
2. *Debrief assignments*
 - a. I'll collect the remaining assignments 1, post the solutions, and discuss them.
 - i. The top score among the dozen that I've graded so far was 40.5/43. Here's the breakdown:

39...	****
35...39	***
30...34	***
...29	**

The most common error was failure to note what happens when one of the sets involved is empty. (One of my professors would appoint a student as guardian of the empty set, to help him avoid blunders in class!)

- i. The top group of students is where you ought to be; the next group is ok. The third group need to study a little to learn to avoid those blunders: they'll prevent you from understanding less trivial matters. The bottom group need to study basics more deeply: you're missing something, and may consequently miss the point of many later explanations. See me if you can't figure out what you're missing.
 - ii. Experience in programming showed me that design errors *usually* betray themselves in the *simplest* cases. For example, if your sorting algorithm is wrong, it will probably fail to sort a list of length two or less. That is why I use this trivial assignment as a diagnostic.
 - b. I'll ask for final responses to the collective nouns exercise. [Click here](#) for a cartoon. Lipton 1991 is a famous source of such lore. [Click here](#) for sample solutions.
 - c. I encourage you to discuss further assignments with each other. Your solution writeups must be your own, however. *If a solution reflects consultation, you must acknowledge that.*
2. *In class*
 - a. *Basic Set Theory* substantial problem 3 (assigned) shows how to speak about injectivity or surjectivity of a function without referring explicitly to its arguments and values. That technique is used in category theory.
 - b. For me, the natural numbers begin with zero.
 - c. There's an arbitrary left-right choice in defining an *ordered triple*. I could have done it the other way.
 - d. Each ordered triple is an ordered pair. Is that a disadvantage? It would be in some kinds of software, but doesn't seem to be so here.
 - e. I discussed *Basic Set Theory* substantial problem 4.

- i. This tool is used repeatedly. In particular, the proofs of the principles of recursive definition are often based on it.
 - ii. A family of functions F_i is called *directed* if for any indices i, j there's an index k such that $F_i, F_j \subseteq F_k$.
 - iii. To prove that the union of a directed family of functions is a function, let $\langle x, y \rangle, \langle x, y' \rangle \in \bigcup_i F_i$. It suffices to prove that $y = y'$. By the definition of union, there are indices i, j such that $\langle x, y \rangle \in F_i, \langle x, y' \rangle \in F_j$. By the definition of directed family, $F_i, F_j \subseteq F_k$ for some index k , so $\langle x, y \rangle, \langle x, y' \rangle \in F_k$. Since F_k is a function, $y = y'$ as required.
3. In a previous meeting, someone noted that the *second principle of recursive definition* at the end of the *Basic Set Theory* unit is phrased awkwardly, as follows:

Given any set Y and any function $G : \mathbb{N} \times \mathcal{P}(\mathbb{N} \times Y) \rightarrow Y$,
 there's a unique $F : \mathbb{N} \rightarrow Y$ such that for each $n \in \mathbb{N}$,
 $\forall n [n \in \mathbb{N} \Rightarrow F(n) = G(\langle n, \{ \langle m, F(m) \rangle : m \in \mathbb{N} \ \& \ m < n \} \rangle)]$.

- a. This permits a great variety of applications. Some texts formulate different versions of the principle for different sorts of applications, at the cost of repeating a tedious argument each time with slightly different premisses.
- b. The most familiar application is definition of the sequence of Fibonacci numbers, which starts with 0, 1 then continues with each number being the sum of its two predecessors. For this application, define G as follows. If $\langle n, E \rangle \in \mathbb{N} \times \mathcal{P}(\mathbb{N} \times Y)$, set

$$G(\langle n, E \rangle) = \begin{cases} 0 & \text{if } n = 0, \\ 1 & \text{if } n = 1, \\ E(n-1) + E(n-2) & \text{if } n > 1 \text{ and } E : \{m : m < n\} \rightarrow Y, \\ \phi & \text{otherwise.} \end{cases}$$

Here, ϕ plays only the role of placeholder; you could put any other set there. I could have omitted the “otherwise” clause had I used in the theorem a more complicated restriction on G , namely

$$G : \{ \langle n, E \rangle : n \in \mathbb{N} \ \& \ E : \{m \in \mathbb{N} : m < n\} \rightarrow Y \}.$$

I could also have avoided using n as a variable to G , since it is recoverable from E (one more than the largest left-hand entry of a member of E , or zero if E is empty). But in order to write the recursion formula in the third clause I'd have to define a function to do that recovery, and use it to get the n in the recursion formula. The moral is, generality sometimes leads to complexity, and even striving for simplicity in one part of a task may breed complexity in another.

- c. You can easily modify this to get other sequences with more precified initial values and more complicated recursion formulas involving more, even all, preceding entries of the sequence.

- d. This kind of definition is analogous to a standard recursive programming technique. For example, here is a *Mathematica* definition of F :

$$F[n] := \text{If } [n == 0, 0, \text{If } [n == 1, 1, F[n - 1] + F[n - 2]]]$$

The statement of the principle of recursive definition and proof of the theorem that justifies it are analogous to the design and implementation of the underlying software that permits *Mathematica* users to define functions recursively.

4. *In class*

- a. I responded to all questions about the *Equivalences and Partitions* unit.
 - b. *You* were prepared to discuss its routine exercise 2: the second example relation is not transitive.
 - c. I discussed its routine exercise 1 and substantial problem 4. The first is set up to facilitate, using various “trivialities” from *Basic Set Theory*, the following clean solution of the second using *relation algebra*, a technique you probably haven’t seen before.
 - i. Suppose E, F are equivalences on a set X that *commute*: $E | F = F | E$.
 - ii. Then $E | F$ is reflexive: $I_X \subseteq E, F \Rightarrow I_X = I_X | I_X \subseteq E | F$.
 - iii. And $D = E | F$ is symmetric: $\bar{D} = \bar{F} | \bar{E} = F | E = E | F$.
 - iv. So, too, is D transitive: $D | D = D | (E | F) = (D | E) | F = ((E | F) | E) | F = ((F | E) | E) | F = (F | (E | E)) | F \subseteq (F | E) | F = F | (E | F) = F | (F | E) = (F | F) | E = F | E = E | F = D$.
 - v. D is an upper bound of E, F : $E = E | I_X \subseteq E | F$; similarly, $F \subseteq E | F$.
 - vi. Finally, D is the supremum (least upper bound) of E, F : if G is an equivalence on X such that $E, F \subseteq G$, then $E | F \subseteq G | G \subseteq G$.
 - d. The intersection of any two equivalences on X is an equivalence on X . In fact, it’s their *infimum* (greatest lower bound).
 - e. There is an extensive theory of commuting equivalence relations. It is applied to algebra through use of the equivalence relation E^* that corresponds to a subgroup E of a group G : $x E^* y \Leftrightarrow xy^{-1} \in E$, for any $x, y \in G$. Such an equivalence is called a *congruence*.
 - f. *Theorem*: the congruence corresponding to a normal subgroup F commutes with the congruences corresponding to any subgroup E of G . *Proof*. Suppose $x, z \in G$ and $x (E^* | F^*) z$. Then $x E^* y$ and $y F^* z$ for some $y \in G$, so $xy^{-1} \in E$ and $yz^{-1} \in F$. Consider $y' = xy^{-1}z$, for which $y'z^{-1} = xy^{-1} \in E$, and $xy'^{-1} = xz^{-1}yx^{-1} = x(yz^{-1})x^{-1} \in F$ because F is normal. Thus $x (F^* | E^*) z$. In short, $E^* | F^* \subseteq F^* | E^*$. The reverse inclusion is proved similarly.
5. I’ll discuss *assignment 3* briefly here; I didn’t get to it in class.
- a. *Routine exercise 3*. Think small.
 - b. *Substantial problem 1*. I’ve given an example of the type of reasoning needed.
 - c. *Substantial problem 5*. Early in most number theory or beginning algebra courses, the following result is presented: *given nonzero $m, n \in \mathbb{N}$, there exist $a, b \in \mathbb{Z}$ such that $am + bn$ is the greatest common divisor of m, n* . You should always be prepared to use that fact in considering divisibility questions.

6. *Project*, that could lead to a term paper or a master's expository paper. Kosmák 1980 presents many extensions of exercises in the *Equivalences and Partitions* unit and related elementary results. Some of those could form the core of a paper about equivalence relations. He didn't give many applications, but an expository paper should also show how the elementary theory of equivalences finds applications in other areas. It should be possible to unearth some references to that. I have copies of several French papers by Paul Dubreil and Marie-Louise Dubreil-Jacotin that may be appropriate. Finberg et al. 1996 is intriguing and understandable.
7. *History*
 - a. Construction of the various number systems from the natural numbers was completed by Georg Cantor, Richard Dedekind, Hermann Grassmann, Giuseppe Peano, and Karl Weierstrass during 1850–1890.
 - b. Responding to the 1960s “New Math” movement, SFSU devoted a whole course to that material, required for perspective high-school teachers and based on Cohen and Ehrlich 1963. Its appropriateness is controversial.
 - c. The 19th-century works just mentioned started an analysis of recursive techniques but left many questions unanswered. Mario Pieri pursued that more deeply in [1907] 2007, but his work was neglected. The validity of recursive definitions was finally proved in the 1920s. Kurt Gödel and others founded recursive function theory in the 1930s to study recursion in great detail.
 - d. The material in the *Equivalence Relations and Partitions* unit, but not all the terminology, was in common use by about 1890.