MA/CSSE 474   Homework #1 (69 points total)

HW1 notes: There are several "no turn-in" problems in this assignment (and in later assignments). This is partly because some of this week's material is supposed to be review of previous courses. I do not want to burden those who remember that material with a too many problems to write up and turn in, but I also want to make sure that those who need to do a lot of review during this course can discover that fact right away. Most of the ten problems that you are required to turn in have answers that are short and easy once you understand the definitions involved. This will not necessarily be true of future assignments!

For all assignments: You may do the problems on your computer and submit them, or hand-write them and scan them. If you need a scanner, a networked scanner is available 24 hours per day in F-217; there are also scanners in Logan Library. If you are off-campus, you can buy a USB scanner for about 2% of what you pay for tuition for this course. You can scan documents and email them to yourself. If your phone takes clear pictures (and you can reduce their file size and have them still be readable), you may use that. Be sure to check to make sure everything is there and is legible before you submit to the drop box. If you make multiple documents, please combine them into one file(a PDF or ZIP file, for example) before you submit. **Please make your file size less than 5 MB if you can do so and still have it be readable.** The "hard limit" for all assignments is 10 MB. If you use a scanner or your phone**, be sure to actually view your PDF** file to make sure that it is readable and that no parts are missing/cut off.

Please use notation and terminology that is consistent with the textbook's.

Please read the part of the syllabus that describes the 0-1-2-3 approach to 474 homework grading.

Please number the problems in your submission the same as they are numbered here (I.e., the turnin problems on this assignment are 3, 4, 6, 7, 9, 11, and 12).

A separate document contains the statements of the textbook problems, so you do not need to have your textbook with you in order to work on problems. This document, not that one, is the primary assignment document; if there is ever a discrepancy between the homework document and the problems document; the homework document is the one to believe (and please point out the discrepancy to me). The problems document will not necessarily contain all of the non-textbook info (such as Q&A from previous Piazza courses) that is contained here.
In the textbook problem numbers on the assignments, 2.4 means Exercise 4 from the end of Chapter 2.

Key to parenthesized expressions after problem numbers:

(No parentheses) Not required to be turned in. Just be sure that you could do it if asked.
(t -9) Turn it in, worth 9 points (multi-part problems may have different point values for each part)

Each problem (or problem part, in some cases), will be graded on the three-point scale described in the syllabus. So your score will be a multiple of one of the numbers in that scale.

1. 2.2   *concatenation of languages* (was also on prep quiz for day 2).
2. 2.3   *List the elements of language concatenation in lexicographic order.*
3. (t-3) 2.4   *A language of strings whose length is a multiple of 3.*
4. (t-6-3) 2.5   *At least two different substrings of length 2.*
5. 2.6   *Strings in or not in a language* Simple English description.
6. (t-6) 2.6(c)   **Simple** English description,
7. (t-15) 2.7   *Closed or not closed?* If closed, you only have to write "closed." If not, what is the closure (smallest closed set that contains the given set)?

Note: In the DISCO book, Grimaldi uses an unusual terminology (P 136). He says "addition and multiplication are closed binary operations on $\mathbf{Z}^+$." A more standard way to say it (and the way Elaine Rich says it) would be "$\mathbf{Z}^+$ is closed under addition and multiplication." See the definition of *closed under a binary operation* and several examples in section A.5.

8. 2.8 *True/False.* No need to turn in all of these. Do them; discuss with other student(s). Problems like these are among my favorites for exams. In problem 9, you will write up a few.

9. (t-15) 2.8 (a,c,d,g,l) Don't forget the "Prove your answer" part. To prove false, give a counterexample.

There is one part of the prerequisite material (mathematical induction) that is so important that I do want to make sure that you are up-to-speed on it.

10. A.21d,e *Proofs of numeric problems by induction.*

11. (t-9) A.22 *Lines through rectangles.* Be sure that your proof (for this and every induction problem) makes it clear where and how the induction assumption is used. *3 points for base case, 6 for induction step*. **Hint:** The induction should be on the number of lines. For the induction step, assume it can be 2-colored after adding k lines, and explain how to get a 2-coloring after adding another line.

12. (t-12) *parentheses and ops.* (The note on the previous problem applies here also.)
A language E of arithmetic expressions is defined recursively from variables {a, b}, operators {+, -} and parentheses as follows:

BASE CASES: $a \in E$ and $b \in E$.

RECURSIVE STEP: If u and v are in E, then (u + v), (u – v), and (-v) are in E.

CLOSURE: A string is in E if and only if it can be obtained from the base cases by a finite number of applications of the recursive step.

For example, the string (a + (b – (-a))) is obtained from three base cases and three applications of recursive rules.

Prove by induction:

$\forall w \in E$ ((number of parentheses in w) = 2*(number operators in w))

Note: Induction is usually based on some number related to the statement we are trying to prove. In this case, there are multiple possibilities for the number that we could use.

While induction on the length of the string w is doable, I believe that it is easier to use induction on the number of applications of the recursive rules that are used to obtain w from the base cases.
3 points for base case, 9 for induction step.

In the past, so many students made the same error that I want to address it in advance.
The erroneous argument went something like this: Assume that the property is true for expressions produced by n applications of the recursive rules. If we then apply another rule, we add one more operator and two more parentheses. The induction hypothesis says that there are n ops and 2n parens in the original; after one more applications of the rules, we have n+1 operators and 2n+2 = 2(n+1) parens. Can you see the problem with this approach? The induction assumption is a statement about expressions in E.

If we apply the + rule, for example, we get (E1 + E2), where E1 and E2 are expressions. The problem with the above argument is that neither E1 nor E2 has n operators, so the above argument does not apply.

For this problem, it is necessary to use strong induction, where the induction assumption is that the property is true for all expressions with fewer than n applications of the recursive rules, then use that to show that it is also true for expressions that use n rules.

All we know about E1 and E2 is that each of them uses fewer than n rules. Strong induction works here, while ordinary induction (i.e., showing the P(n) implies P(n+1)) doesn't.

Some past questions and answers from Piazza:

Actions

Hint for HW1, problem 11 (2-coloring the rectangle)

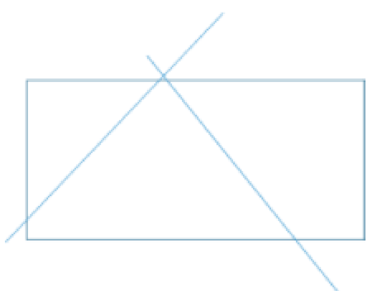
A student asked me a question, and I thought perhaps everyone might benefit from the discussion that we have had so far:

Student: I've gone over some of my induction problems from CSSE230 and MA275 to try to jog my memory on induction but I keep getting stuck at what I should choose as my inductive hypothesis.

Me: A good Inductive hypothesis is "if the rectangle is cut by k lines, it can be colored with two colors." You assume that and show that if a k+1st line is added, you can still 2-color it. Of course you have to show how to get from the 2-coloring with k lines to the 2-coloring with k+1 lines.

Student: Showing how to get from k -> k + 1, could I describe it In a paragraph. Something to the effect of "if a rectangle is cut by k lines then, then it can be 2-colored. If It is cut one more time it creates an even number of new pieces to be colored which can then be 2 colored".

Me:
You can and should describe it in a paragraph. But YOUR paragraph is much too vague (and not true; if the cut goes through an existing intersection point, it might make an odd number of new pieces -- see the picture).



You need to precisely describe (i.e. an algorithm) how to start with a 2-coloring of the regions of the k-cut rectangle and modify it to get a 2-coloring of the regions that you have after the k+1st cut.

You might think that there have to be many special cases for the induction step, but that is not the case.

Closure example (related to problem 7)

Consider the set L={1, 2, 3}. This set is not closed under subtraction, since (for example) 2-3=-1, which is not in L. What is the closure of this set under subtraction? **It's the smallest closed set that contains L.** Adding the elements that we can get by immediate subtraction, we have (-2. -1, 0, 1, 2, 3}. But this set is not closed under subtraction, either. The new numbers we can get by subtracting elements of this set are 5, 4, -5, -4, -3. And if we add these numbers to our set, we get a set that is still not closed. It's now not too hard to see that the closure must be the set of all integers.

If we asked about closure of the same set under addition, we find that it is the set of all positive integers.

The closure of {2, 5} under addition is slightly more complicated. Can you figure out what it is?

A student correctly answered: For closure of {2, 5} under addition, I guess it's all positive integers except 1 and 3.