CS6160 Theory of Computation
Problem Set 3
Department of Computer Science, University of Virginia

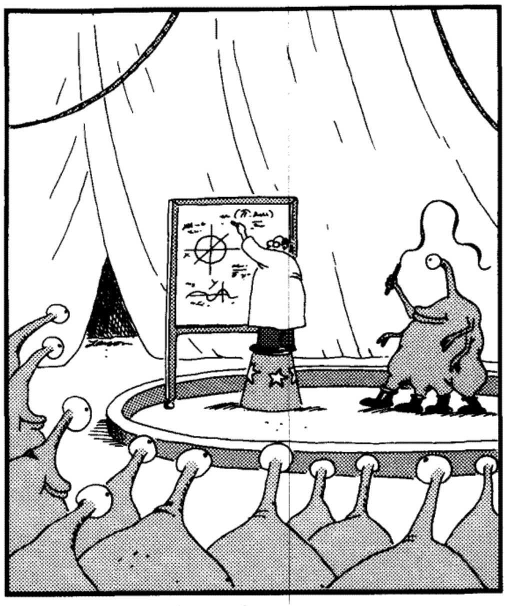
Gabriel Robins

Please start solving these problems immediately, don't procrastinate, and work in study groups. Please prove all your answers; informal arguments are acceptable, but please make them precise / detailed / convincing enough so that they can be easily made rigorous if necessary. To review notation and definitions, please read the "Basic Concepts" summary posted on the class Web site, and also read the corresponding chapters from the Sipser textbook and Polya's "How to Solve It".

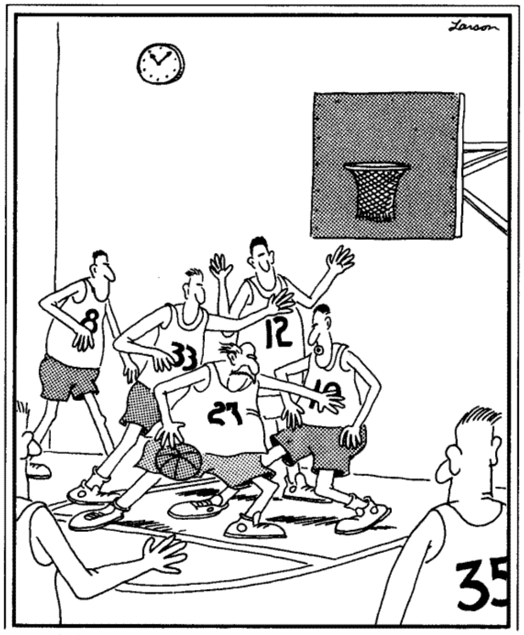
Please do not simply copy answers that you do not fully understand; on homeworks and on exams we reserve the right to ask you to explain any of your answers verbally in person (and we have exercised this option in the past). Please familiarize yourself with the UVa Honor Code as well as with the course Cheating Policy summarized on page 3 of the Course Syllabus. To fully understand and master the material of this course typically requires an average effort of at least six to ten hours per week, as well as regular meetings with the TAs and attendance of the weekly problem-solving sessions.

This is not a "due homework", but rather a "pool of problems" meant to calibrate the scope and depth of the knowledge / skills in CS theory that you (eventually) need to have for the course exams, becoming a better problem-solver, be able to think more abstractly, and growing into a more effective computer scientist. You don't necessarily have to completely solve every last question in this problem set (although it would be great if you did!). Rather, please solve as many of these problems as you can, and use this problem set as a resource to improve your problem-solving skills, hone your abstract thinking, and to find out what topics you need to further focus on and learn more deeply. Recall that most of the midterm and final exam questions in this course will come from these problem sets, so your best strategy of studying for the exams in this course is to solve (including in study groups) as many of these problems as possible, and the sooner the better!

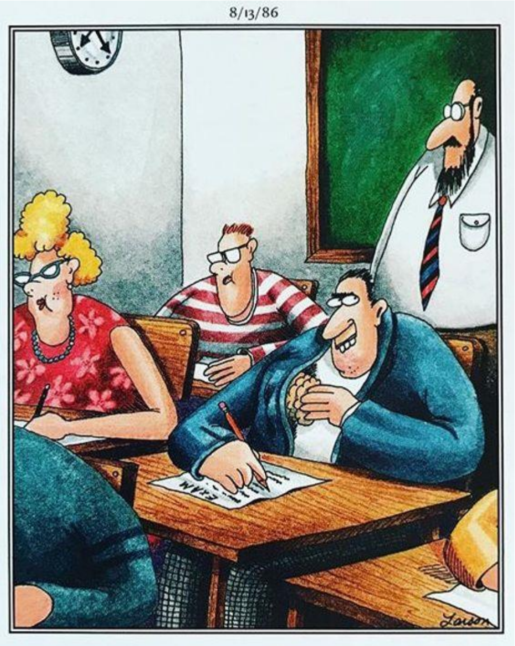
Advice: Please try to solve the easier problems first (where the meta-problem here is to figure out which are the easier ones ☺). Don't spend too long on any single problem without also attempting (in parallel) to solve other problems as well. This way, solutions to the easier problems (at least easier for you) will reveal themselves much sooner (think about this as a "hedging strategy" or "dovetailing strategy").



Abducted by an alien circus company, Professor Doyle is forced to write calculus equations in center ring.



Unbeknownst to most historians, Einstein started down the road of professional basketball before an ankle injury diverted him into science.



Midway through the exam, Allen pulls out a bigger brain.

1. Solve the following problems from the [Sipser, Second Edition] textbook:

   Pages 83-93: 1.4, 1.5, 1.6, 1.7, 1.11, 1.12, 1.13, 1.16, 1.17, 1.18, 1.19, 1.20, 1.21, 1.23, 1.29, 1.30

2. Prove or disprove: a countable set of parabolas (arbitrarily oriented and placed) can completely cover (every point inside) the unit square in the plane (i.e., the interior and boundary of a square of side 1)

3. Prove or disprove: an uncountable set of pairwise-disjoint line segments can completely cover (every point in) the unit disk in the plane (i.e., the interior and boundary of a circle of diameter 1). What if the segments could intersect each other, but must all have unique slopes?

4. What is the cardinality of the set of all finite-sized matrices with rational entries?

5. What is the cardinality of the set of all infinite matrices (i.e., matrices with a countably-infinite number of rows and columns) with Boolean entries?

6. Is every subset of a regular language necessarily regular?
   Is every superset of a regular language necessarily non-regular?

7. Does every regular language have a proper regular subset?
   Does every regular language have a proper regular superset?

8. Are the regular languages closed under infinite union? Infinite intersection?

9. Is a countable union of regular languages necessarily regular?

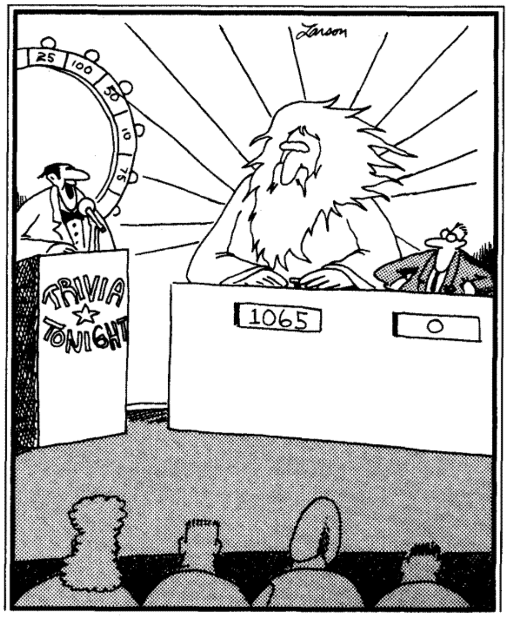
10. Is a countable intersection of regular languages necessarily regular?

11. Solve problems 1.6(b), 1.6(h), 1.6(i) on page 84 of [Sipser]. Use JFLAP to implement and test each of these deterministic finite automata on various representative input strings.

12. Solve problem 1.17 on page 86 of [Sipser]. Use JFLAP to implement the NFA and DFA of this question, and test both of them on various representative input strings.

BONUS QUESTION: (50 points)
What's the name of that thing that hangs down in the back of our throats?

Final page of the Medical Boards



"Yes! That's right! The answer *is* 'Wisconsin!' Another 50 points for God, and … uh-oh, looks like Norman, our current champion, hasn't even scored yet."

13. Prove or disprove: the set of all languages (i.e. 2^{Σ^*}) is countable.

14. Prove or disprove: a given regular language is a countable set.

15. Prove or disprove: the set of all regular languages is a countable set.

16. Determine as precisely as possible, for a language L, when is the following true: $L^+ = L^* - \{\varepsilon\}$

17. What is the infinite union of all of the regular languages?
What is the infinite intersection of all of the regular languages?

18. Let $\text{YESNO}(L)=\{xy \mid x\in L \text{ and } y\notin L,\ x,y\in\Sigma^*\}$. Does YESNO preserve regularity?

19. Let $\text{PALI}(L)=\{w \mid w\in L \text{ and } w^R\in L\}$. Does PALI preserve regularity?

20. Describe an algorithm that determines for a given pair of regular expressions whether they denote the same language. What is the time complexity of your algorithm?

21. True or false: for any given regular language, there exists a linear-time algorithm for testing whether an arbitrary input string is a member of that language.

22. What is the smallest language, closed under concatenation, containing the languages L_1 and L_2?

23. Give a sufficient condition (but as general as possible) for $L_1^* + L_2^* = (L_1 + L_2)^*$ to hold.

24. Given two arbitrary languages S and T, find a new language R (in term of S and T) so that the equation R = SR + T holds.

25. Describe exactly what happens if we apply the "powerset construction" to a finite automaton that is already deterministic?

26. Show that the intersection of two sets of languages can be empty, finite (of arbitrarily large cardinality), countably infinite, or uncountably infinite.

"Frankly, I have a lot of trouble relating to all this symbolism."

"It's hard to tell you this, Melnik, but you're being sent down to the Institute for Pretty Hard Thinking."