CS6160 Theory of Computation
Problem Set 4
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").

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

   Pages 83-93: 1.31, 1.36, 1.38, 1.40, 1.41, 1.42, 1.43, 1.45, 1.46, 1.48, 1.57, 1.58, 1.60, 1.61, 1.63, 1.64

2. Define a new operation ♣ on languages as follows: $♣(L) = \{w \mid \exists\, w \in \Sigma^*, ww^R \in L\}$, where w^R denotes the "reverse" of the string w. Does ♣ preserve regularity?

3. Are there two non-regular languages whose concatenation is regular? Are there a countably infinite number of such examples? Are there an uncountable number of such examples?

4. Define a new operation ♦ on languages as follows: $♦(L) = \{w \mid \exists\, z \in \Sigma^* \ni |w|=|z| \wedge wz \in L\}$. Does the operation ♦ preserve regularity?

5. We define the SHUFFLE of two strings $v,w \in \Sigma^*$ as:

   $$\text{SHUFFLE}(v,w) = \{v_1w_1v_2w_2 \ldots v_kw_k \mid v=v_1v_2\ldots v_k,\ w=w_1w_2\ldots w_k,\ \text{and for some } k \geq 1,\ v_i,w_i \in \Sigma^*,\ 1 \leq i \leq k\}$$

   For example, 212**ab**1**baa**2**b**22 ∈ SHUFFLE(**abbaab**,2121222)

   Extend the definition of SHUFFLE to two languages $L_1,L_2 \subseteq \Sigma^*$ as follows:

   $$\text{SHUFFLE}(L_1,L_2) = \{w \mid w_1 \in L_1,\ w_2 \in L_2,\ w \in \text{SHUFFLE}(w_1,w_2)\}$$

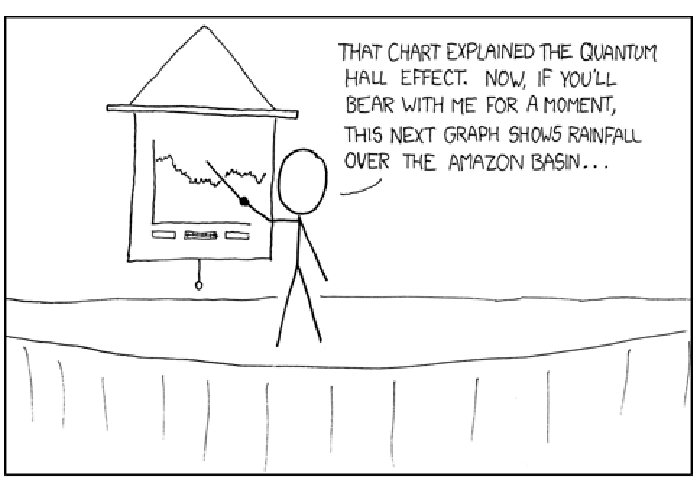
   a) Is the SHUFFLE of two finite languages necessarily finite?

   b) Is the SHUFFLE of two regular languages necessarily regular?

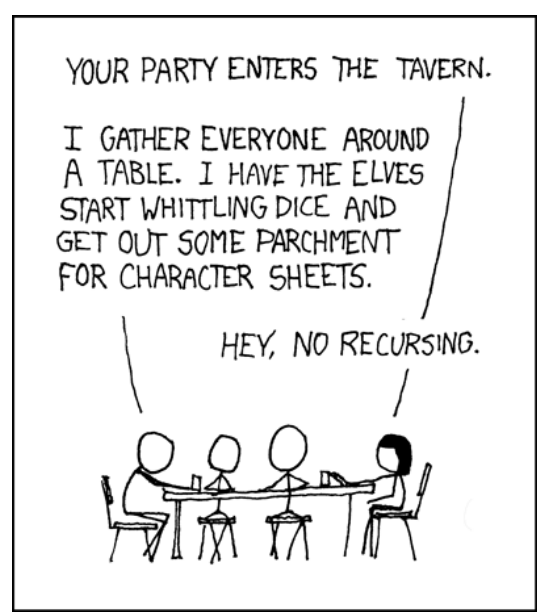
6. Define a DIVISION operator on languages as follows:

   $$\frac{L_1}{L_2} = \{w \mid w \in \Sigma^* \text{ and } \exists\, v \in L_2 \ni wv \in L_1\}$$

   Does DIVISION preserve regularity? What if L_1 is regular and L_2 is arbitrary?



IF YOU KEEP SAYING "BEAR WITH ME FOR A MOMENT", PEOPLE TAKE A WHILE TO FIGURE OUT THAT YOU'RE JUST SHOWING THEM RANDOM SLIDES.



7. Which of the following modifications / restrictions to finite automata would change the class of languages accepted relative to "normal" (unmodified) finite automata? (Assume a fixed alphabet of say $\Sigma = \{a,b\}$)

a) The ability to move the read head backwards (as well as forwards) on the input.

b) The ability to write on (as well as read from) the input tape.

c) Both a) and b) simultaneously.

d) Having 2 read-heads moving (independently, left-to-right) over the input.

e) Having no more than one billion different states.

8. Determine whether each of the following languages is regular:

a) $\{a^n a^n a^n \mid n>0\}$

b) $\{www \mid w \in \{x,y,z\}^*, |w| < 10^{100}\}$

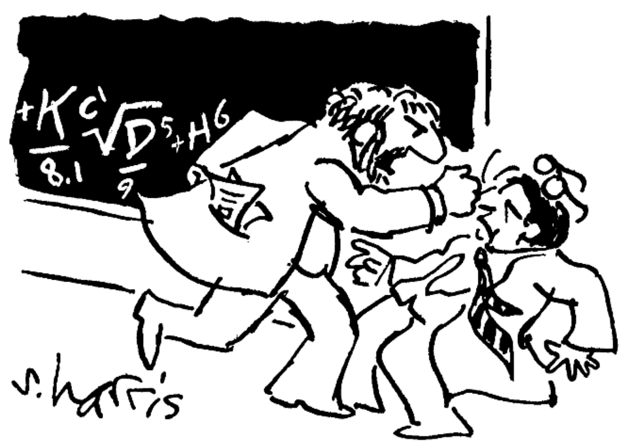
c) $\{vw \mid v,w \in \{a,b\}^*\}$

d) $\{ww \mid w \in \{a\}^*\}$

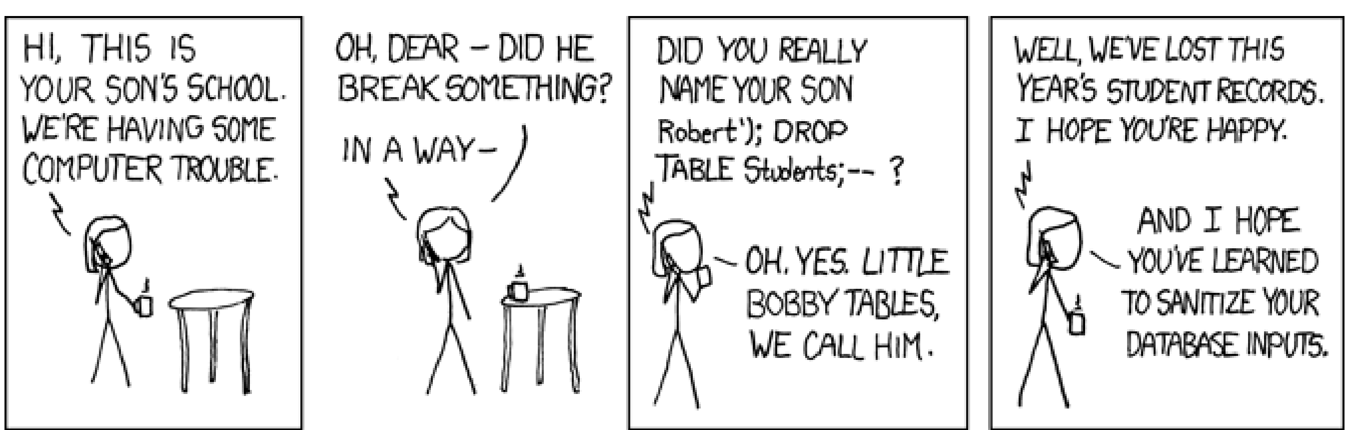
9. Are there two non-finitely-describable languages whose concatenation is regular? Are there a countably infinite number of such examples? Are there an uncountable number of such examples?

10. Let F denote some finite language, R denote some regular language, C denote some context-free language, and N denote some non-context-free languages. For each one of the following statements, prove whether it is always true, sometimes true, or never true:

a) RC is regular
b) N – R is regular
c) $N \cap F$ is not regular
d) N – F is regular
e) C^* is infinite
f) R is context-free
g) $R \cup C$ is finite
h) C is regular
i) N is infinite



"You want proof? I'll give you proof!"



11. Give algorithms (i.e., a well-defined, deterministic, always-terminating decision procedures, and state their time complexities) to determine whether for a given finite automaton M, L(M) is:

a) countable

b) empty

c) Σ^*

d) finite

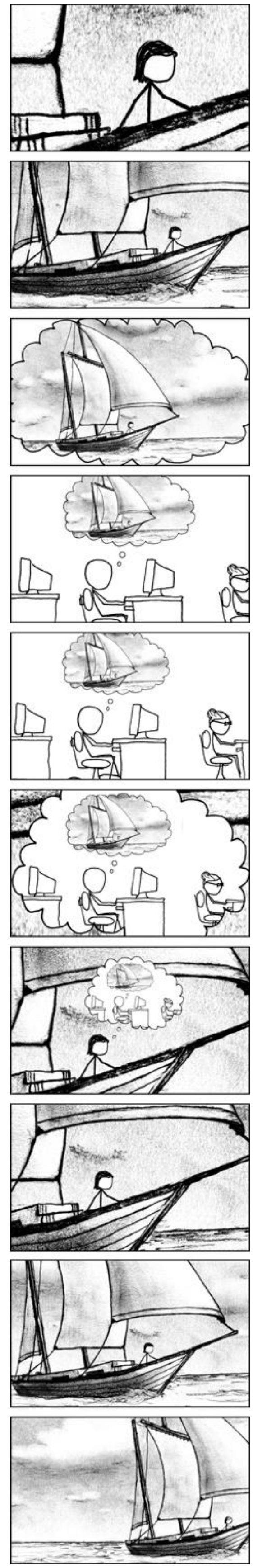
e) infinite

f) co-finite (i.e., with a finite complement)

g) regular

h) context-free

i) also accepted by a smaller FA (i.e., with fewer states than M)



12. Give algorithms (and state their time complexities) to determine whether for a given pair of finite automata:

a) they both accept the same language

b) the intersection of their languages is empty

c) the intersection of their languages is finite

d) the union of their languages is finite

e) the intersection of their languages is infinite

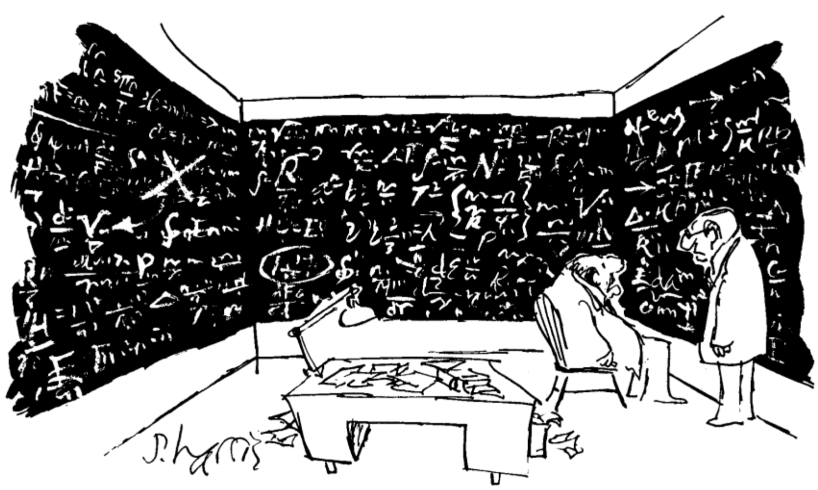
f) the union of their languages is infinite

g) the intersection of their languages is Σ^*

h) the difference of their languages is finite

13. Let $L=\{0^n1^n \mid n\geq 0\}$. Is $\overline{L}$ (i.e. the complement of L) a regular language?

14. Let $L=\{0^i1^j \mid i\neq j\}$. Is L a regular language?



"Whatever happened to *elegant* solutions?"