CS3102 Theory of Computation
Problem Set 3

Department of Computer Science, University of Virginia

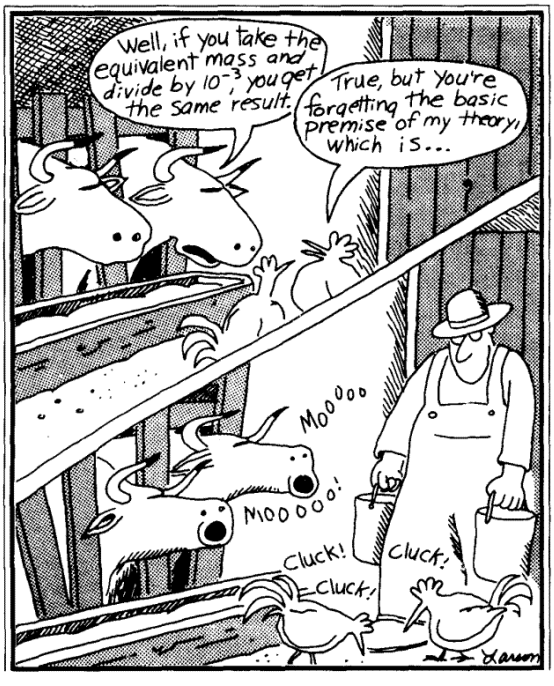
Gabriel Robins

Please start solving these problems immediately, 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. To review notation and definitions, please read the "Basic Concepts" summary posted on the class Web site, and also read the second chapter (on regular languages) from the Sipser textbook.

Important note: 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, thinking more abstractly, performing more effectively as a computer scientist, etc. 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, abstract thinking, and to find out what topics you need to further focus & improve on. Recall that most (and perhaps even all) 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 (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. ☺) Please don't spend too long on any single problem without also attempting (in parallel) to solve other problems as well; this way, the easiest problems (at least to you) will reveal themselves much sooner (think about this as a "hedging strategy").

Before paper and scissors



1. The following problems are from [Sipser, Second Edition]:

   Pages 128-132: 2.4, 2.5, 2.6, 2.9, 2.10, 2.13, 2.16, 2.17, 2.20, 2.21, 2.22, 2.27, 2.36, 2.43, 2.44

   Pages 159-162: 3.4, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19, 3.22

   Pages 182-184: 4.2, 4.3, 4.4, 4.6, 4.7, 4.9, 4.12, 4.15, 4.18, 4.19, 4.22, 4.26, 4.27, 4.28

   Pages 211-214: 5.2, 5.4, 5.6, 5.7, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.20, 5.28, 5.29, 5.30, 5.31

   Pages 242-243: 6.23, 6.24

   Page 294-300: 7.6, 7.7, 7.9, 7.10, 7.11, 7.13, 7.17, 7.21, 7.26, 7.36, 7.38, 7.42, 7.44

2. Does every context-free language have a proper context-free subset?
   Does every context-free language have a proper context-free superset?

3. Is every subset of a context-free language necessarily context-free?
   Is every superset of a context-free language necessarily non-context-free?

4. Are the context-free languages closed under infinite union? Infinite intersection?
   Are the Turing-decidable languages closed under infinite union? Infinite intersection?
   Are the Turing- recognizable languages closed under infinite union? Infinite intersection?

5. Is a countable union of regular languages necessarily context-free? Decidable?
   Is a countable union of decidable languages necessarily Turing-recognizable?

6. What is the infinite union of all context-sensitive languages? Decidable languages?
   What is the infinite intersection of all context-sensitive languages? Decidable languages?

7. Let YESNO(L)=$\{xy \mid x\in L \text{ and } y\notin L, x,y\in\Sigma^*\}$. Does YESNO preserve context-freeness? Turing-recognizability?

8. Let PALI(L)=$\{w \mid w\in L \text{ and } w^R\in L\}$. Does PALI preserve context-freeness? Turing-recognizability?

9. Define the "Busy Beaver" function BB:$\mathbb{N}\to\mathbb{N}$ as follows: BB(n) is the maximum number of 1's printed on the tape of any Turing machine with n states which halts when running on the blank tape (i.e., with no input). Is BB finitely describable? Is BB computable? How fast does BB grow asymptotically?

10. If we had free access to an oracle that computes the Busy Beaver function for us in constant time, prove either that all functions (mapping naturals to naturals) are computable relative to such an oracle, or else give a counter-example. (Please don't do both. ☺)

11. Given the alphabet $\Sigma=\{a,b,(,),+,*,\emptyset,\varepsilon\}$ construct a context-free grammar that generates all strings in Σ^* that correspond to regular expressions over {a,b}.

12. Construct the smallest possible (in terms of the number of non-terminals and/or production rules) context-free grammar that generates all well-formed parenthesis.

13. Construct a (small) context-free grammar that generates all well-formed nestings of parenthesis ( ) and brackets [ ].

14. Characterize as precisely as you can the class of languages accepted by deterministic pushdown automata with two stacks.

15. Characterize as precisely as you can the class of languages accepted by deterministic pushdown automata with a single "counter" (i.e., stack with only a single-letter stack alphabet).

16. Characterize as precisely as you can the class of languages accepted by deterministic pushdown automata with two "counters" (i.e., two stacks with only a single-letter stack alphabet).

17. Construct a context-sensitive grammar that generate $\{a^n b^n c^n \mid 1 \leq n\}$. Make your grammar as "small" as possible in terms of the number of non-terminals and productions in it.

18. Construct a context-sensitive grammar that generate $\{a^{(n \wedge n)} \mid 1 \leq n\}$. Make your grammar as "small" as possible in terms of the number of non-terminals and productions in it.

19. Let $L=\{0^n 1^n \mid n \geq 0\}$. Is $\overline{L}$ (i.e. the complement of L) context-free?

20. Let $L= \{0^i 1^j \mid i \neq j\}$. Is L a context-free language?

21. Does there exist a context-free grammar for $\{0^i 1^j \mid 1 \leq i \leq j \leq 2i\}$?

22. Is $\{w \in \{a,b\}^* \mid$ w contains an equal number of a's and b's$\}$ a context-free language?

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

$$\text{SHUFFLE}(v,w) = \{v_1 w_1 v_2 w_2 \ldots v_k w_k \mid v=v_1 v_2 \ldots v_k,\ w=w_1 w_2 \ldots w_k,$$
$$\text{and for some } k \geq 1,\ v_i, w_i \in \Sigma^*,\ 1 \leq i \leq k\}$$

For example, 212**ab**1**baa**2**b**22 $\in$ 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 context-free languages necessarily context-free?

b) Is the SHUFFLE of a context-free language with a regular language necessarily context-free?

c) Is the SHUFFLE of two Turing-recognizable languages necessarily Turing-recognizable?

24. Which of the following modifications / restrictions to PDA's would change the class of languages accepted, relative to "normal" PDA's?

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) Having 2 read-heads moving (independently, left-to-right) over the input.
d) Having three stacks instead of one.
e) Having a stack alphabet of at most two symbols.
f) Having a stack alphabet of one symbol.
g) Having a FIFO queue instead of a stack (i.e., write-only at the top of the queue, and read-only at the bottom of the queue).

25. Is $\{v\$w \mid v,w \in \{a,b\}^*, v \neq w\}$ a context-free language?

26. Is $\{vw \mid v,w \in \{a,b\}^*, v \neq w\}$ a context-free language?

27. Is $\{vw \mid v,w \in \{a,b\}^*, v \neq w, |v|=|w|\}$ a context-free language?

28. Determine whether each of the following languages is context-free.

a) $\{a^n a^n a^n \mid n>0\}$
b) $\{www \mid w \in \{x,y,z\}^*, |w| < 10^{100}\}$
c) $\{vw \mid v,w \in \{a,b\}^*\}$

29. Give (and prove) several example non-Turing-recognizable languages.

30. Describe a Turing machine that prints out its own description (regardless of its input).

31. Prove whether given a TM M and string w, each of the following properties is decidable, Turing-recognizable, or not Turing-recognizable:

a) an arbitrary given string w causes M to enter state 3.
b) there exists some string that causes M to enter state 3.
c) an arbitrary given string w causes M to enter each and every one of its states.
d) a given string w causes M to move its head to the left at least once when M runs on w.
e) M accepts a finite language.
f) M accepts a regular language.
g) M accepts a decidable language.
h) M accepts a Turing-recognizable language.
i) M never writes a nonblank symbol on its tape when it runs on a given string w.
j) M never overwrites a nonblank symbol when it runs on a given string w.
k) M never overwrites a nonblank symbol when it runs on any string.
l) M is a universal Turing machine.

32. Let $L=\{0^k \mid k \text{ is a Fibonacci number}\}$. Describe a Turing machine that accepts L. Give a (context-sensitive) grammar that generates L.

33. Describe a two-tape Turing machine that prints out on its second tape only prime numbers (in either binary or unary, separated by commas), such that every prime number will eventually be printed there.

34. Describe a two-tape Turing machine that prints out on its second tape valid encodings of all Turing machines (separated by commas), such that every Turing machine (including itself!) will eventually be printed there.

35. Is it decidable whether given a one-state PDA accepts all input strings? How about a three-state PDA?

36. Two cyborgs walk into your home, both claiming to be oracles for the graph 3-colorability decision problem. They both always give a yes/no answer in constant time for any instance, and are each self-consistent (i.e. each always gives the same answer for the same instance). However, one is a true oracle and the other is a shameless impostor, and you have a large instance of 3-colorability upon which they disagree. Prove whether it is possible to expose the impostor within time polynomial in the size of that instance.

37. Modify Turing machines so that they can insert new tape cells into their tape(s), and also remove old tape cells from their tape(s), instead of only (over)writing existing tape cells. (a) Define carefully the transition function and the computational behavior of such machines. (b) Show that such a machine can be simulated by an ordinary Turing machine with at most a quadratic loss of efficiency.

38. True or false: any two-tape Turing machine that uses constant space (aside from the read-only space occupied by the input string) recognizes a regular language.

39. True or false: if L is Turing recognizable, then there is a Turing machine M that enumerates L without ever repeating an element of L.

40. True or false: If a rooted binary tree has infinitely many nodes, then it has an infinitely long path from the root.

41. True or false: Most Boolean functions on N inputs have an exponentially long (as a function of N) minimal description (in any fixed reasonable encoding / formalism).

42. True or false: Most Boolean functions on N inputs require an exponential (as a function of N) number of 2-input Boolean gates to implement.

43. Is NP closed under the Kleene-star operator?

44. Is P closed under the Kleene-star operator?

45. Explain whether or not matrix inversion is NP-complete.

46. Is NP countable?