CS6160 Theory of Computation
Problem Set 2

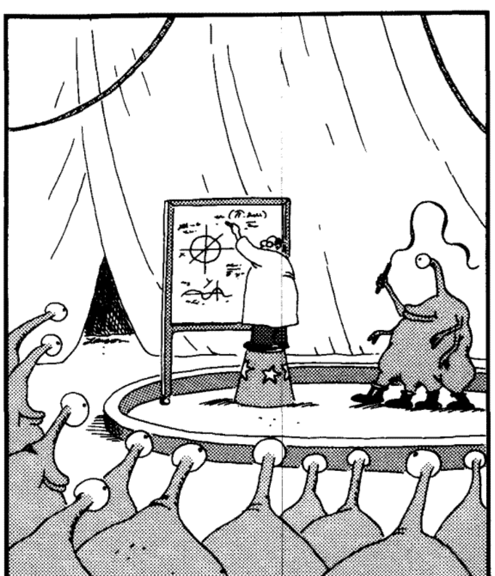
Department of Computer Science, University of Virginia

Gabriel Robins

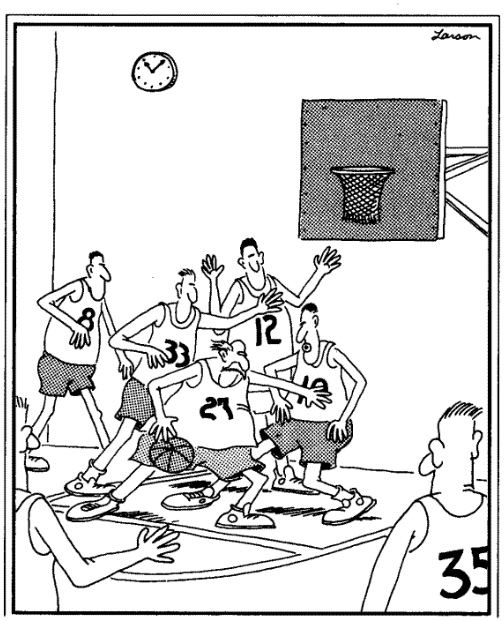
Please start solving these problems immediately, and work in study groups. Please prove all your answers; informal arguments are acceptable, but please make them precise / detailed / convincing enough so that they can be easily made rigorous.

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 exams, PhD quals, becoming a better problem-solver, thinking more abstractly and generally, performing more effective research, 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 list 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!

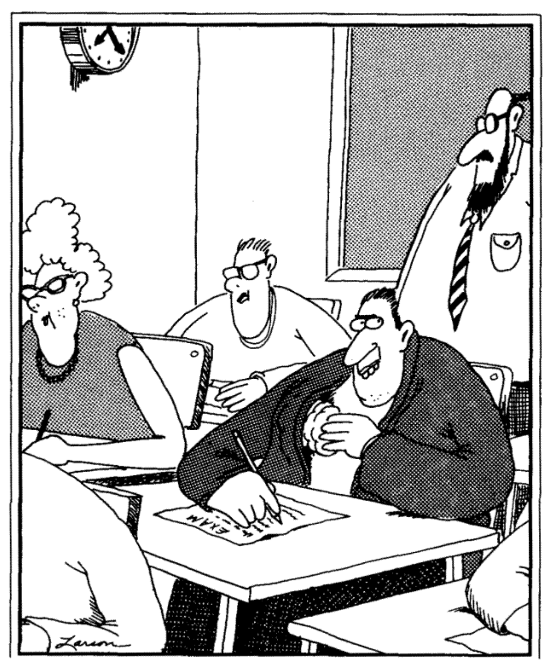
Advice: Please try to solve the easier problems first (where the meta-problem here is to figure out which are the easier ones. ☺) Please don’t spend too long on any single problem without also attempting (in parallel) to solve other problems as well; this way, the easiest problems (at least to you) will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



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

   Page 27: 0.10, 0.12

   Pages 83-92: 1.6, 1.7, 1.11, 1.18, 1.20, 1.21, 1.45, 1.46, 1.48, 1.58, 1.63

   Pages 128-132: 2.4, 2.5, 2.9, 2.16, 2.17, 2.21, 2.22, 2.24, 2.27, 2.32, 2.33, 2.36, 2.37, 2.40, 2.41, 2.42, 2.43, 2.44, 2.45

   Pages 159-162: 3.7, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19, 3.20, 3.22

   Pages 182-184: 4.2, 4.3, 4.4, 4.6, 4.7, 4.10, 4.12, 4.15, 4.17, 4.18, 4.19, 4.24, 4.26, 4.27, 4.28

   Pages 211-214: 5.2, 5.4, 5.9, 5.12, 5.13, 5.14, 5.15, 5.16, 5.20, 5.26, 5.27, 5.28, 5.29, 5.33, 5.35

   Pages 242-243: 6.1, 6.2, 6.4, 6.6, 6.14, 6.15, 6.16, 6.17, 6.18, 6.20, 6.21, 6.22, 6.23, 6.24

2. Prove or disprove: a countable set of parabolas (arbitrarily oriented and placed) can completely cover (every point in) 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?

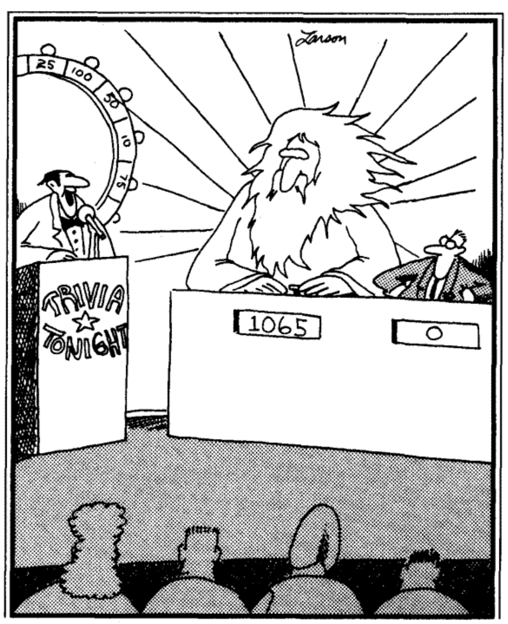
4. What is the cardinality of the set of all finite-sized matrices with rational entries?

5. What is the cardinality of the set of all infinite matrices with Boolean entries?

6. Determine as precisely as possible when is the following true: $L^+ = L^* - \{\varepsilon\}$

7. Describe exactly what happens if we apply the "powerset construction" to an automaton that is already deterministic?

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."

8. Define the set of all prefixes of L as PREFIX(L) = $\{w \mid wy \in L \text{ for some } w,y \in \Sigma^*\}$. Does PREFIX preserve regularity? Context-freeness? Decidability? Turing-recognizability?

9. Define the set of all suffixes of L as SUFFIX(L) = $\{w \mid yw \in L \text{ for some } y,w \in \Sigma^*\}$. Does SUFFIX preserve regularity? Context-freeness? Decidability? Turing-recognizability?

10. Define the set of all subsequences of L as SUBSEQ(L) = $\{w_1w_2w_3\ldots w_k \mid \exists\ k \in \mathbb{N}, \exists\ w_i \in \Sigma^*$ for $1 \leq i \leq k$, and $\exists\ x_j \in \Sigma^*$ for $0 \leq j \leq k$ such that $x_0w_1x_1w_2x_2w_3x_3\ldots w_kx_k \in L\}$. Does SUBSEQ preserve regularity? Context-freeness? Decidability? Turing-recognizability?

11. Define the set of all supersequences of L as SUPERSEQ(L) = $\{ x_0w_1x_1w_2x_2w_3x_3\ldots w_kx_k \mid \exists\ k \in \mathbb{N}$, $\exists\ w_i \in \Sigma^*$ for $1 \leq i \leq k$, and $\exists\ x_j \in \Sigma^*$ for $0 \leq j \leq k$ such that $w_1w_2w_3\ldots w_k \in L\}$. Does SUPERSEQ preserve regularity? Context-freeness? Decidability? Turing-recognizability?

12. A language L is said to be "definite" if there exists some fixed integer k such that for any string w, whether $w \in L$ depends only on the last k (or less) symbols of w. (a) State this definition more formally. (b) Is a definite language necessarily regular? (c) Is the set of definite languages closed under union? Intersection? Complementation? Concatenation? Kleene closure?

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

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

15. Does every contex-free language have a proper contex-free subset?
Does every contex-free language have a proper contex-free superset?

16. Is every subset of a contex-free language necessarily contex-free?
Is every superset of a contex-free language necessarily non-contex-free?

17. Are the regular languages closed under infinite union? Infinite intersection?
Are the context-free languages closed under infinite union? Infinite intersection?
Are the decidable languages closed under infinite union? Infinite intersection?

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

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

20. Are the decidable languages closed Kleene closure? Concatenation? Union? Complementation? Are the non-decidable languages closed Kleene closure? Concatenation? Union?

21. Are the non-finitely-describable languages closed under concatenation? Kleene closure? Complementation? Union?

22. Can a uncomputable number be rational? Must an irrational number be non-computable?

23. Is the set of non-finitely-describable real numbers closed under addition? Squaring?

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

25. Let $PALI(L)=\{w \mid x\in L \text{ and } x^R\in L\}$. Does PALI preserve regularity? Context-freeness? Decidability? Turing-recognizability?

26. Define the density of a language to be the function $D_L(n) = |\{w \mid w\in L \text{ and } |w|\leq n\}|$. What is the density of $(a+b)^*$? What is the density of a^*b^* ? Show that the density of a regular language is either bounded from above by a polynomial, or bounded from below by an exponential (i.e., a function of the form 2^{cn} for some constant c). In other words, densities of regular languages can not be functions of intermediate growth such as $n^{\log n}$.

27. True or false: the densities of the decidable languages can be any computable function.

"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."

28. Define the "Busy Beaver" function $BB:\mathbb{N}\rightarrow\mathbb{N}$ 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?

29. 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. ☺)

30. A string w is square-free if it can not be written in the form $w=xy^2z$ for some $x,z\in\Sigma^*$ and $y\in\Sigma^+$. Are there arbitrarily long square-free strings on a two-letter alphabet? How about a three-letter alphabet?

31. A string w is cube-free if it can not be written in the form $w=xy^3z$ for some $x,z\in\Sigma^*$ and $y\in\Sigma^+$. Are there arbitrarily long cube-free strings on a two-letter alphabet?

32. True or false: if $|\Sigma|=1$ then the set of all cube-free strings in Σ^* is regular.
True or false: if $|\Sigma|=2$ then the set of all square-free strings in Σ^* is regular.
True or false: if $|\Sigma|>2$ then the set of all square-free strings in Σ^* is not regular.

33. 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?

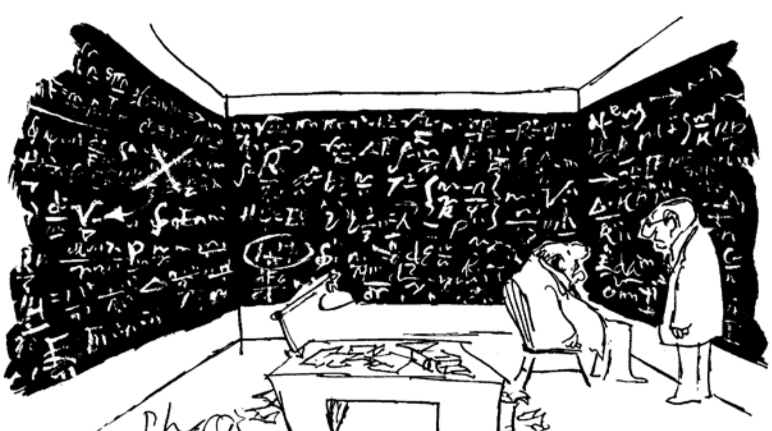
34. True or false: for any given regular language, there exists a linear-time algorithm for testing whether an arbitrary input string is a member of that language.

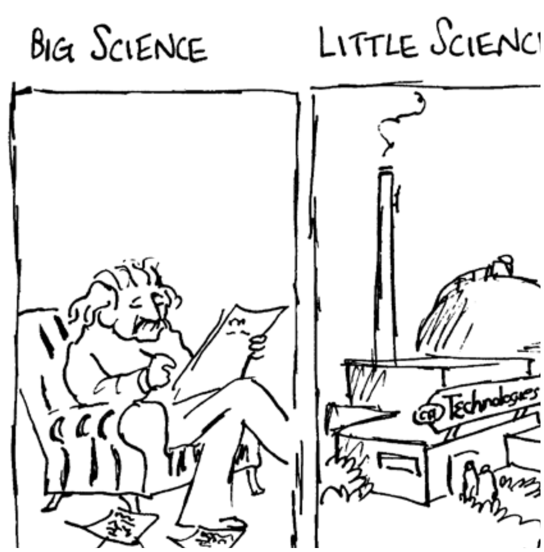
35. Given the alphabet $\Sigma=\{a,b,(,),+,*,\emptyset,\varepsilon\}$ construct a context-free grammar that generates all strings in Σ^* that correspond to regular expressions over {a,b}.

36. 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.

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

38. Characterize as precisely as you can the class of languages accepted by deterministic push-down automata with two stacks.


"Whatever happened to *elegant* solutions?"



39. Characterize as precisely as you can the class of languages accepted by deterministic push-down automata with a single “counter” (i.e., stack with only a single-letter stack alphabet).

40. Characterize as precisely as you can the class of languages accepted by deterministic push-down automata with two “counters” (i.e., two stacks with only a single-letter stack alphabet).

41. 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.

42. 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.

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

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

45. 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.

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

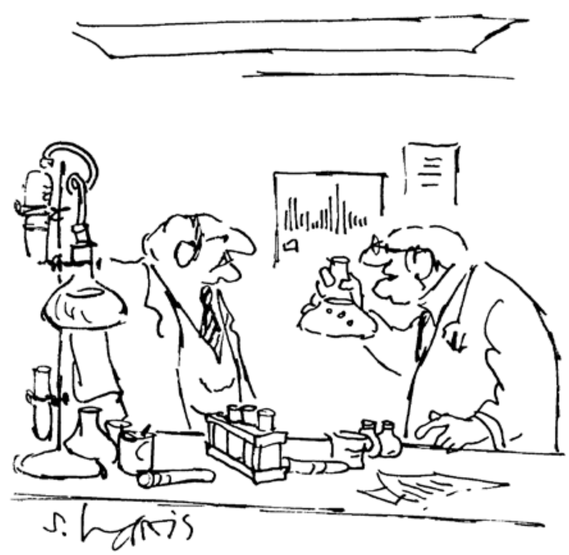
47. Let $L=\{0^n 1^n \mid n \geq 0\}$. Is $\overline{L}$ (i.e. the complement of L) a regular language? Is $\overline{L}$ context-free?

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

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

50. 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?

“ ‘Look’, I would say to Leonardo. ‘See how far our technology has taken us.’ Leonardo would answer, ‘You must explain to me how everything works.’ At that point, my fantasy ends.”



“It may very well bring about immortality, but it will take forever to test it.”

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

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

53. 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?

54. Define an "**infinite automata**" similarly to finite automata, but where the state set Q is no longer restricted to be finite. Characterize precisely the class of languages accepted by deterministic infinite automata. Is the characterization any different for non-deterministic infinite automata? Do oracles increase the power of infinite automata?

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

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

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:

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

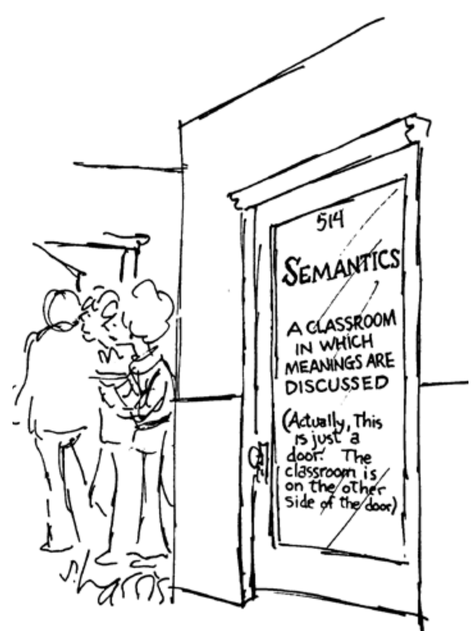
a) Is the SHUFFLE of two regular languages necessarily regular?

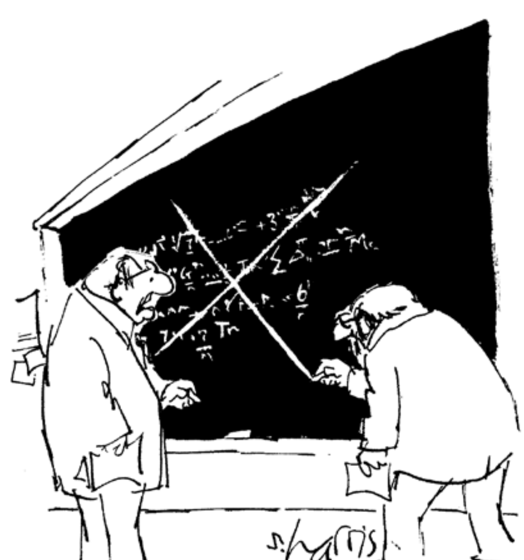
b) Is the SHUFFLE of two context-free languages necessarily context-free?

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

d) Is the SHUFFLE of two decidable languages necessarily decidable?

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





56. Which of the following modifications / restrictions to finite automata would change the class of languages accepted relative to "normal" finite automata?

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 one billion or less different states.

57. 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).

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

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

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

61. Determine whether each of the following is regular, context-free, or both.

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

"Very creative. Very imaginative. Logic…*that's* what's missing."

62. Given an arbitrary alphabet $\Sigma = \{a_1, a_2, \ldots, a_n\}$, we can impose a total ordering on it in the sense that we can define $<$ so that $a_1 < a_2 < \ldots < a_n$. We now proceed to define a new operation called the SORT of a string $w = w_1 w_2 \ldots w_k \in \Sigma^*$ (where $w_i \in \Sigma$ and $k = |w|$) as:

$$\mathrm{SORT}(w) = w_{\sigma(1)} w_{\sigma(2)} \ldots w_{\sigma(k)} \quad \text{so that } w_{\sigma(i)} < w_{\sigma(i+1)} \text{ for } 1 \le i \le k-1$$
$$\text{and } \sigma \text{ is a permutation (i.e., a 1-to-1 onto mapping } \sigma:[1..k] \rightarrow [1..k])$$

For example, SORT(11210010120)=00001111122. Now extend the definition of SORT to languages, so that SORT(L) = {SORT(w) | w∈L}. For each one of the following statements, state whether it is true or false and explain:

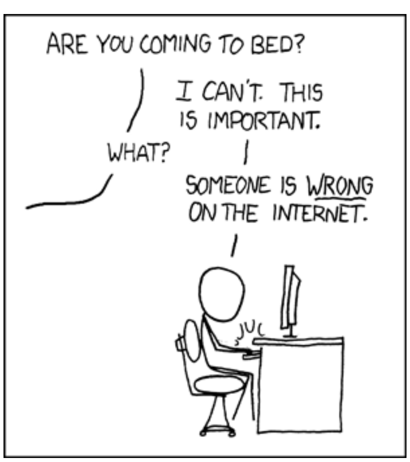
a) $\mathrm{SORT}(\Sigma^*)$ is regular.
b) $\mathrm{SORT}(L) \subseteq L$
c) SORT(SORT(L))=SORT(L)
d) $\mathrm{SHUFFLE}(L_1,L_2)=\mathrm{SHUFFLE}(L_2,L_1)$
e) $\mathrm{SORT}(\mathrm{SHUFFLE}(L_1,L_2)) = \mathrm{SORT}(L_1 L_2)$
f) $\exists$ L such that SORT(L)=SHUFFLE(L,L)=L
g) SORT preserves regularity.
h) SORT preserves context-freeness.
i) SORT preserves decidability
j) SORT preserves Turing-recognizability
k) SORT preserves non-decidability
l) SORT preserves non-finite-describability

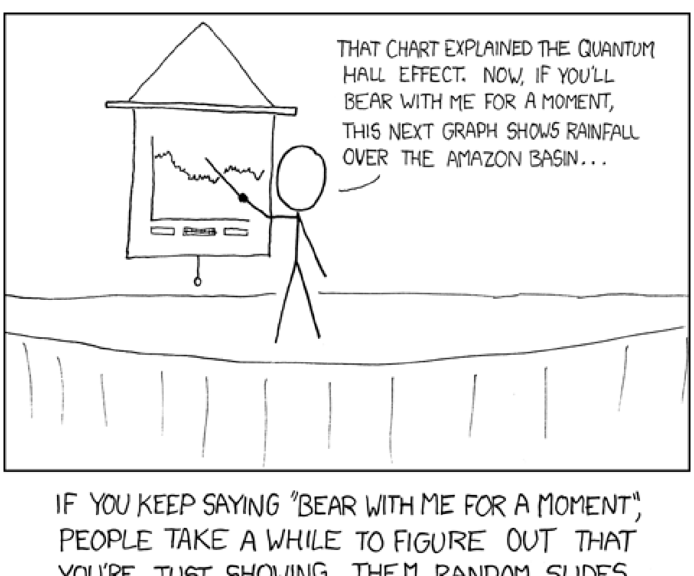
(The definition of SHUFFLE operator is the same as above.)

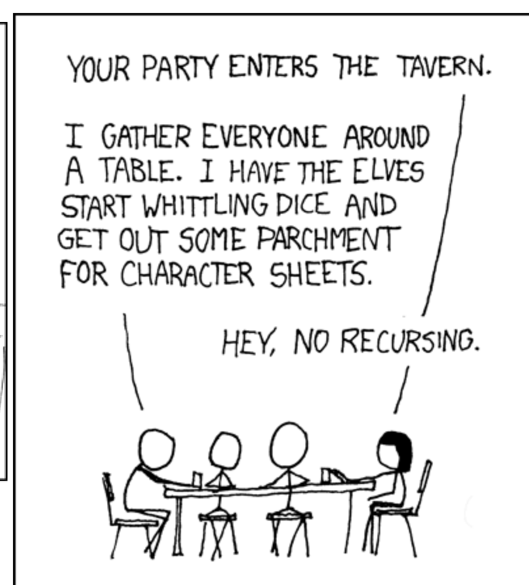
63. Define a DIVISION operator on languages as follows:

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

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







64. Give decision procedures (i.e., a well-defined, deterministic, always-terminating algorithm) to determine whether for a given finite automaton M, L(M) is:

a) empty

b) Σ^*

c) finite

d) infinite

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

f) regular

g) context-free

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

65. Give algorithms 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 Σ^*

d) the intersection of their languages is finite

e) the difference of their languages is finite

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

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

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

a) w causes M to enter state 3.

b) there exists some string that causes M to enter state 3.

c) w causes M to enter each and every one of its states.

d) 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 w.

j) M never overwrites a nonblank symbol when it runs on w.

k) M never overwrites a nonblank symbol when it runs on any string.

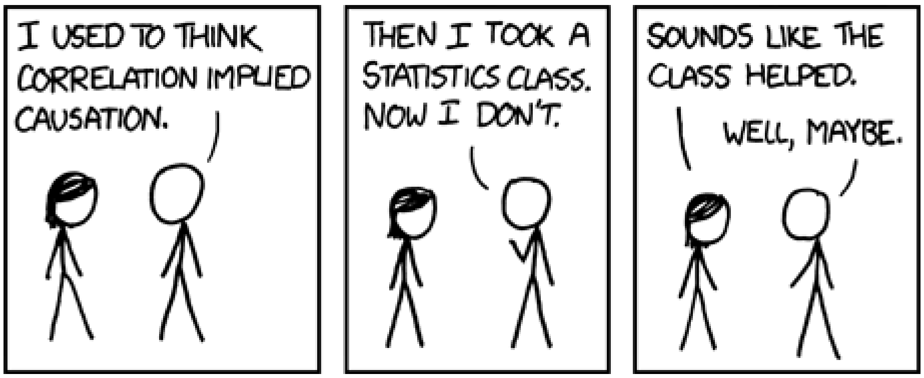
l) M is a universal Turing machine.

69. 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.

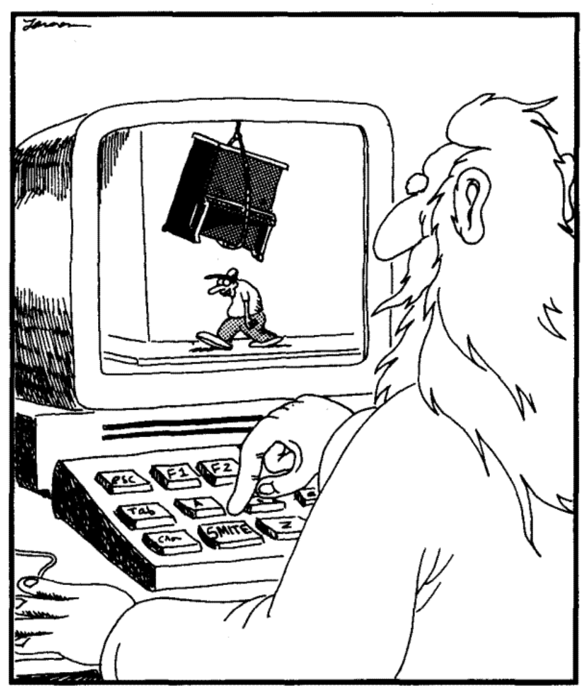
70. Describe a two-tape Turing machine that prints out on its second tape only prime numbers (in either binary or unary, separated by commas), such that every prime number will eventually be printed there.

71. 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.

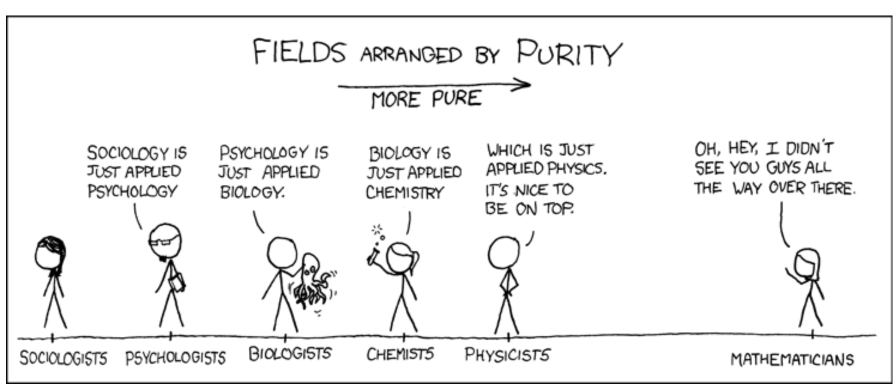
72. Is it decidable whether given a one-state PDA accepts all input strings? How about a three-state PDA?

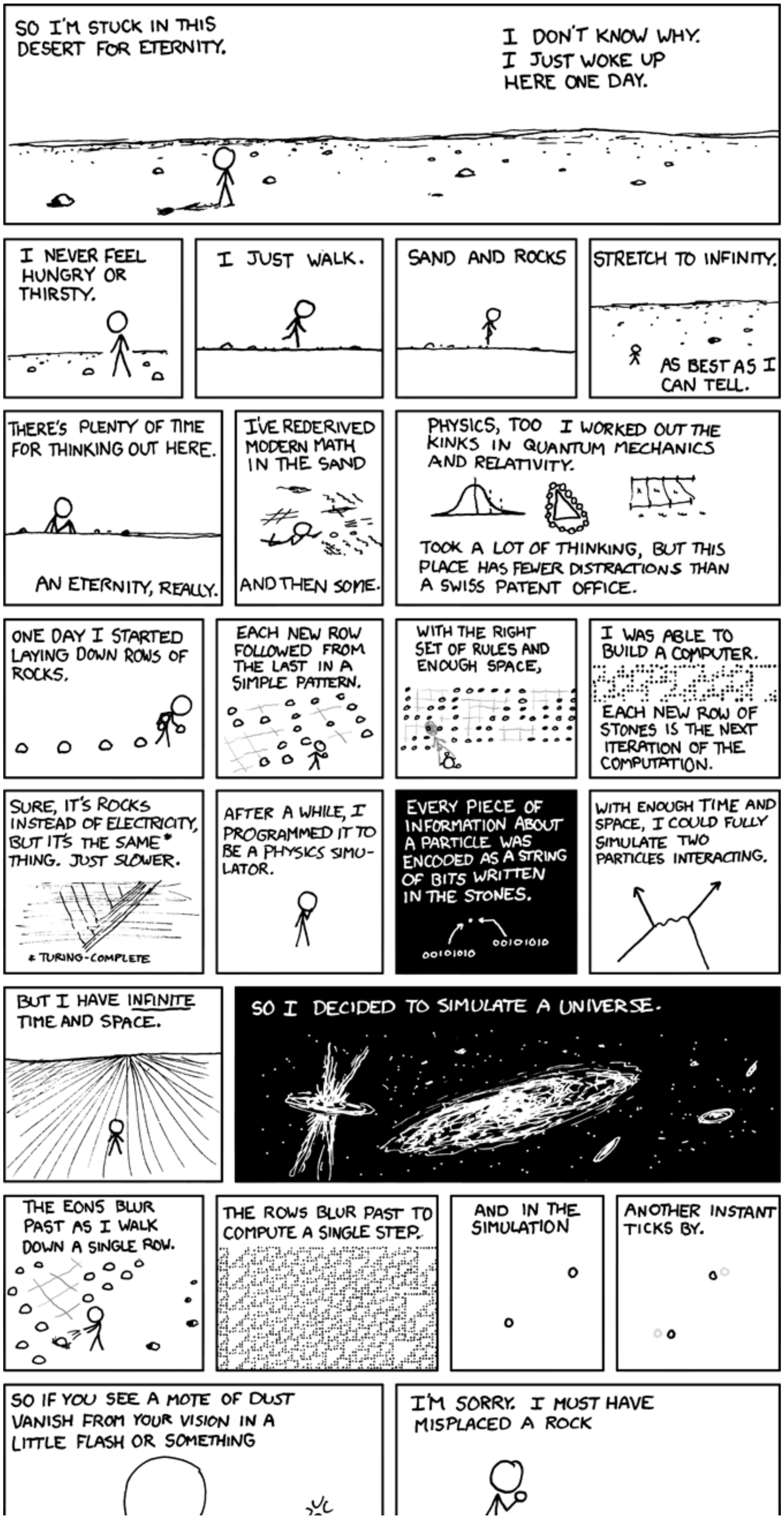


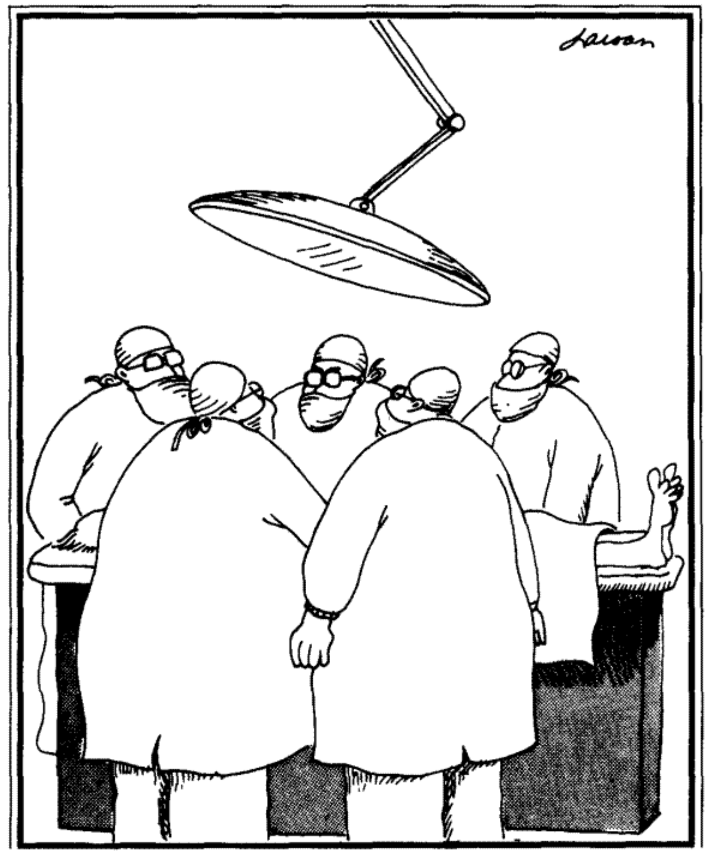
Early checkers



God at His computer







"Okay, Williams, we'll vote . . . how many here say the heart has four chambers?"

Unwittingly, Irwin has a brush with Death.