COMP 314 Homework Assignment B

This is version 3 of the assignment, published February 6, 2015. Question B2 has been updated, and some other minor changes were made.

Chapter 3

Question B1. (Ungraded) Do all of the practical exercises recommended in the text of Chapter 3. In particular:

- run `countLines(rf('countLines.py'))`
- Run all the commands in Figure 3.3, and verify the output.
- Use `yesOnStringApprox.py` to verify the outputs in Figure 3.5 wherever possible. Which lines produce the result "`unknown`", and why?
- Use `yesOnSelfApprox.py` to verify the outputs in Figure 3.5 wherever possible.

Question B2. (10 points) As stated in Figure 3.3, the output of the command

```
>>> containsGAGA(rf('containsGAGA.py'))
```

is "`yes`". Write a new version of `containsGAGA`, called `containsGA_GA.py`. The new version should be equivalent to the old one (that is, it produces exactly the same outputs given the same inputs). However, the result of the command

```
>>> containsGA_GA(rf('containsGA_GA.py'))
```

should be "`no`".

Question B3. (10 points) Consider a hypothetical Python program called `noOnString.py`. This program is similar to `yesOnString.py`, but detects "`no`" outputs instead. Specifically, `noOnString.py` takes two parameters (P, I) and outputs "`yes`" if and only if $P(I) =$ "`no`". As you might expect, `noOnString.py` cannot exist, but we can approximate it for a few particular values of P, using the same technique as `yesOnStringApprox.py`. Write a program called `noOnStringApprox.py` which has this desired behavior. Your approximation should work correctly for the same inputs as `yesOnStringApprox.py`.

Question B4. (Ungraded) Similarly to the previous question, consider a hypothetical Python program called `noOnSelf.py`, which takes a parameter

P and outputs "`yes`" if and only if $P(P)$ = "`no`". Write an approximation `noOnSelfApprox.py` which approximates `noOnSelf.py` for the same values of P as in the previous question.

Question B5. (Ungraded) Define `noOnSelf.py` as in the previous question. Assuming that `noOnSelf.py` exists, what should be the output of the following commands?

(a) `noOnSelf(rf('noOnSelf.py'))`

(b) `noOnSelf(rf('yesOnSelf.py'))`

(c) `yesOnSelf(rf('noOnSelf.py'))`

Question B6. (5 points) Define `noOnSelf.py` as in the previous questions. Although `noOnSelf.py` is similar to `notYesOnSelf.py`, these programs are not equivalent. Give an example of an input for which these two programs (if they existed!) would give different outputs. Briefly explain your answer.

Question B7. (15 points) Consider a Python program `definedOnString.py`. This program takes two parameters (P, I). It returns "`yes`" if $P(I)$ is defined and "`no`" otherwise. Using an approach similar to Figure 3.12 and the claim on page 54, prove that `definedOnString.py` cannot exist.

Question B8. (Ungraded) Consider a Python program `longerThan10.py`. This program takes two parameters (P, I). It returns "`yes`" if $P(I)$ is a string of more than 10 characters, and "`no`" otherwise. Using an approach similar to Figure 3.12 and the claim on page 54, prove that `longerThan10.py` cannot exist.

Question B9. (Ungraded) Consider a Python program `startsWithZ.py`. This program takes two parameters (P, I). It returns "`yes`" if $P(I)$ is a string that starts with the character `Z`, and "`no`" otherwise. Using an approach similar to Figure 3.12 and the claim on page 54, prove that `startsWithZ.py` cannot exist.

Question B10. (5 points) Using the previous three questions as inspiration, describe another impossible Python program. Try to come up with an interesting program: for example, `longerThan5.py` and `startsWithABC.py` are indeed impossible, but they are too similar to the earlier questions to be interesting.

Question B11. (10 points) Suppose a new videogame called "Best Game Ever" is released. You play the game with a friend and are disappointed to discover the game is rather buggy. Your friend, who has taken a course on the theory of computation, says: "Actually, it's impossible for a computer program to automatically find all the bugs in the Best Game Ever software." Do you agree with your friend? Explain your answer.

Chapter 4

Question B12. (Ungraded) Do all of the practical exercises recommended in the text of Chapter 4. In particular:

- Run each of the sorting programs in Figure 4.2 on a few test inputs. Verify that the first two work correctly on all inputs, but the last works only on some inputs.
- *To do: Anything else to go here?*

Question B13. In this question, we use the conventions of Figure 4.3 for converting between graphs and ASCII strings.

(a) (3 points) Let G be the undirected graph represented by "`a1,a2 a1,b1 a2,b1 a2,b2 a2,b3 a2,a3 a3,b2 a3,b3`". What is the shortest path from `a1` to `b3` in G?

(b) (1 point) Is G a tree?

(c) (3 points) Let H be the directed graph represented by "`apple,apple apple,banana banana,coconut coconut,banana coconut,apple`". List all (directed) cycles present in H. Hint: there are three cycles. Clarifications: (i) we are only interested in cycles that have no repeated vertices; (ii) the start vertex is irrelevant e.g. `x,y,z` and `y,z,x` represent the same cycle.

(d) (4 points) Consider the tree T represented by "`p,y y,v y,x y,z z,w z,u`". We will regard T as a rooted tree with root u.

  (i) Which vertices are at level 2?

  (ii) Which vertices are leaves of the tree?

  (iii) List the vertices in depth-first order.

  (iv) List the vertices and breadth-first order.

Question B14.

(a) (1 point) How many symbols are in the ASCII alphabet, as it is defined in this book?

(b) (1 point) Consider the following set S of symbols, which consists of all horizontal lines whose length is some multiple of 2 millimeters: $S = \{\_, \_\_, \_\_\_, \ldots\}$. Is S an alphabet?

Question B15. Consider the alphabet $\Sigma = \{\texttt{C}, \texttt{A}, \texttt{G}, \texttt{T}\}$.

(a) (2 points) How many strings of length 0,1,2 and 3 can be made using Σ?

(b) (2 points) How many strings of length n can be made using Σ?

(c) (2 points) How many strings of length at most n can be made using Σ?

(d) (2 points) Approximately how long would it take for a modern computer to iterate over all strings on Σ of length exactly 20? (Assume each string can be processed in 1 ns.) How about for strings of length exactly 50?

Question B16. (4 points) Assume we using the ASCII alphabet and take $s =$ "abc", $t =$ "yz". Determine the following strings or values:

(a) sts

(b) $t^2 s^0 t \lambda$

(c) $|(st^2)^9|$

(d) λ^4

Question B17. (5 points) Assume we using the 4-symbol alphabet of genetic bases, ordered according to $\texttt{A} < \texttt{C} < \texttt{G} < \texttt{T}$. List the following set of strings in (a) shortlex and (b) lexicographical order: {CGG, AAG, AAGA, T, TATA, TTT}.

Question B18. (Ungraded) How many strings are in each of the following languages?

(a) Binary strings of length at most 3.

(b) Binary strings containing exactly three 1s and exactly two 0s.

(c) Binary strings containing exactly three 1s.

(d) $\varnothing$

(e) $\{\texttt{Japan}, \texttt{Korea}, \texttt{China}, \texttt{Vietnam}, \lambda\}$

Question B19. (Ungraded) Take our alphabet Σ to be $\{\texttt{C}, \texttt{A}, \texttt{G}, \texttt{T}\}$ and let L be the language on Σ consisting of all strings accepted by the program containsGAGA.py (see Figure 2.1).

(a) Is $\lambda \in L$?

(b) How many strings of length 4 are in L?

(c) How many strings of length 6 are in L?

Question B20. (Ungraded) Suppose that L is defined as in the previous question, except that the ASCII alphabet is used. According to this definition of L:

(a) Is the source code of `containsGAGA.py` an element of L? Explain your answer.

(b) Suppose `another.py` is a Python program that is equivalent to `containsGAGA.py`. Can we conclude that `another.py` is an element of L? Explain your answer. Hint: See question xxx. *To do: fix*

Question B21. (Ungraded) Let J be the language of Java programs as defined in this book. That is, J is the set of ASCII strings that produce no errors when processed by some particular choice of the Java compiler. Conduct an experiment to determine whether $\lambda \in J$. Briefly describe your experiment and your conclusion. If you're not familiar with Java, you can do a similar experiment for any other programming language.

Question B22. (10 points) Write a Python program that decides the language of prime numbers represented in decimal as ASCII strings. That is, your program should output "`yes`" on inputs like "`5`" and "`41`", but "`no`" on inputs like "`8`" and "`39`". Your program need only work correctly on inputs that represent positive integers.

Question B23. (3 points) Let $\Sigma = \{\texttt{C}, \texttt{A}, \texttt{G}, \texttt{T}\}$, and let F be a solution-set function on Σ mapping a string s to all substrings of s that begin and end with `A`. For each value of s below, give $F(s)$.

(a) `CAGGATACC`

(b) `GAGA`

(c) `CGCGATT`

Question B24. (4 points) Write out the full solution set for

(a) ShortestPath("`1,2 1,3 1,4 2,4 2,5 3,5 6,7; 1 ; 5`")

(b) ShortestPath("`1,2 1,3 1,4 2,4 2,5 3,5 6,7; 3 ; 6`")

(c) ShortestPath("`1,2 1,3 1,4 2,4 2,5 3,5 6,7; 3`")

Question B25. (4 points) For each of the instances of ShortestPath given below, state whether the instance is positive or negative.

(a) `x,y x,z u,v; x; u`

(b) `x,y x,z u,v; y; z`

(c) `x,y x,z u,v; u; v`

(d) `x,y,x z,u,v; y; w`

Question B26. (Ungraded) Give the solution set for each of the problem instances below.

(a) ShortestPath("`a,b b,c c,d d,e e,f f,a; a; d`")

(b) ShortestPathDecision("`a,b b,c c,d d,e e,f f,a; a; d; 2`")

(c) ShortestPathDecision("`a,b b,c c,d d,e e,f f,a; a; d; 3`")

(d) Multiply("`3 4`")

(e) MultiplyDecision("`3 4 10`")

(f) MultiplyDecision("`3 4 12`")

Question B27. (3 points) Let CountGs be a computational problem that takes an ASCII string I as input. The (unique) solution is the number of times `G` occurs in I, written in decimal notation.

(a) Does CountGs have any positive instances? If so, give an example.

(b) Does CountGs have any negative instances? If so, give an example.

(c) Suggest a definition of CountGsDecision, a decision problem that intuitively seems to require the same computational process as the non-decision problem CountGs.

Question B28. (Ungraded) Compute the following values:

(a) ESS("`CGGATT`", "`CATTA`")

(b) ESS(λ, "`CATTA`")

(c) ESS("`CGGATT`", λ)

(d) ESS("`GG`", ESS("`CC`", "`GG`"))

(e) DESS("`3 CAGGAC`")

(f) DESS("`6 CAGGAC`")

(g) DESS("`0 CAGGAC`")

(h) For any strings s, t: DESS(ESS(s, t))

Question B29. (Ungraded) Write out a formal definition of the computational problem NoOnString, which decides whether a program outputs "`no`" on a given input.

Question B30. (4 points) Let Q be the language of ASCII strings that end in a question mark.

(a) Is Q a decidable language? Explain your answer.

(b) Is Q a recognizable language? Explain your answer.

(c) Give the solution of: IsMember_Q("`Am I in Q?`")

(d) Give the solution of: IsMember_Q("`I am in Q!`")

Question B31. (5 points) Let TwoZs be the decision problem which decides whether an ASCII string contains exactly two `Z` characters.

(a) Is TwoZs a decidable problem? Explain your answer.

(b) Is L_{TwoZs} a decidable language? Why?

(c) Is L_{TwoZs} a recognizable language? Why?

(d) Is "`XYZ`" $\in L_{\text{TwoZs}}$?

(e) Assuming the problem TwoYs is defined similarly for the character `Y`, give an example of the string that is a member of $L_{\text{TwoZs}} \cap L_{\text{TwoYs}}$.

Question B32. (Ungraded) Give an example of an undecidable language that is not $L_{\text{YesOnString}}$ or $L_{\text{CrashOnString}}$. (Hint: look back to exercises xxx.) Briefly explain why your example is undecidable.

Question B33. (10 points) Let L be the language of ASCII strings that contain the character `Z`. Write a Python program that recognizes L, but does not decide L.

Total points on this assignment: 128