CSE396, Spring 2021 Problem Set 6 Due Tue. 3/30, 11:59pm

Reading: On Tuesday we will finish section 2.3 and move on to chapter 3, so please read section 3.1 for Tuesday. Note that we will come back to section 2.2 after section 3.2.

Homework—part online (TopHat), part written, and all *individual work*:

(1) Using *TopHat*, the "Worksheet" titled *S21 HW6 Online Part* (10 Qs, 20 pts.)

Answers given there:

The other two problems are to be submitted as PDFs using the *CSE Autograder* system.

(2) Let E be the language of all strings over $\Sigma = \{a, b\}$ that do not have the substring bb, and let G be the following context-free grammar:

$$\begin{aligned} S &\longrightarrow \epsilon \mid b \mid BS \mid SA \\ A &\longrightarrow aS \mid AA \\ B &\longrightarrow a \mid bAaB \end{aligned}$$

(a) Show that the string $babab$ is ambiguous in the grammar G, by giving two different parse trees. (6 pts.)

(b) Is any other variable besides S capable of deriving ϵ? Give one(s) if so. (3 pts.)

(c) Prove by structural induction that $L(G) \subseteq E$. *Hint:* Ask yourself what additional properties, besides not allowing a bb substring themselves, must the variables A and B maintain? (15 pts., for 24 total)

Answer: To describe the trees in prose: Start $S \to SA$ in both cases. In the first, derive the S to b and A to aS. This gives baS, which repeats to give $babaS$ and finally $babab$. The second derives the A to AA instead, giving bAA. The next level of the tree has $baSaS$, and deriving both S-es to b completes the tree. (There are other parses too.)

(b) No, only S is nullable. In particular, A is not nullable since it eventually must derive at least one a.

(c) [Same key as from Spring 2019, which was same key as from earlier year that I assembled the parts of this problem from.] Define $P_S \equiv$ "Every x that I derive has no substring bb," which is just the "vanilla" statement of membership in E. A top-down, goal-oriented way to get the stronger properties needed for A and B was to note that by the rule $S \to b$, one must allow that a substring y derived from S might *both* begin *and* end with b. Hence $S \to SA$ requires that strings z such that $A \Longrightarrow^* z$ cannot begin with b—which since A is not nullable means z must begin with a. Likewise $S \to BS$ mandates that any string w derived from B (which is not nullable either) must end in a. Thus the properties you want for P_A and P_B both *include* P_S, and *add* the clause "and if $A \Longrightarrow^* x$ then x begins with a" for P_A and "if $B \Longrightarrow^* x$ then x ends in a" to P_B. The key further point required for full credit was that we need to uphold these extra clauses in the rules for A and B as well.

- $S \to \epsilon \mid b$: These rules uphold P_S immediately.

- $S \to BS$: Suppose $S \Longrightarrow^* x$ using this rule first. Then $x =: yz$ where $B \Longrightarrow^* y$ and $S \Longrightarrow^* z$. By IH P_B, P_S on the right-hand side, neither y nor z has a bb substring *and* y ends in a, so no bb appears where they concatenate either. So x has no bb, which upholds P_S on the left-hand side.

- $S \to SA$: Suppose $S \Longrightarrow^* x$ utrf. Then $x =: yz$ where $S \Longrightarrow^* y$ this time and $A \Longrightarrow^* z$. By IH P_S, P_A on RHS, y and z each have no internal bb *and* z begins with a. So x has no bb at the boundary either, which upholds P_S on LHS. (That finishes S, but we still need to maintain the stronger properties in the other rules.)

- $A \to aS$: Suppose $A \Longrightarrow^* x$ utrf. Then $x = ay$ where $S \Longrightarrow^* y$. By IH P_S on RHS, y has no bb so neither does x, *and* the leading a makes $ay = x$ immediately uphold the added clause of P_A. So P_A is upheld on LHS.

- $A \to AA$: Suppose $A \Longrightarrow^* x$ utrf. Then $x =: yz$ where $A \Longrightarrow^* y$ and $A \Longrightarrow^* z$. By IH P_A on RHS (twice), neither y nor z has an internal bb, and y begins with a which implies the same for x. Is that all we need to say, i.e., is it immaterial that z begins with a? No, that is needed too, to say that no bb occurs at the boundary between y and z. So P_A holds on the LHS.

- $B \to a$: Both clauses of P_B are immediately upheld.

- $B \to bAaB$: Suppose $B \Longrightarrow^* x$ utrf. Then $x =: byaz$ where $A \Longrightarrow^* y$ and $B \Longrightarrow^* z$. By IH P_A, P_B on RHS, those strings have no internal bb, and they don't touch and in fact surround an a, so the only bb danger can come from the leading b. That is averted, however, because P_A on RHS implies that y begins with a. We're not done—we need to uphold "ends with a" as well to get P_B on LHS, but this follows by "self-induction" since z ends in a by IH P_B on RHS.

Thus the properties are upheld by all the rules, so $L(G) \subseteq E$ by structural induction.

(3) Let $A = \{ a^n b^n : n \geq 1 \}$. Define E to be the language of strings that differ *in at most one place* from a string in A. An example of a string in E is $aaba$, since changing the last a to b gives a string in A. Note that E contains A, and that the strings in E have the same lengths as strings in A. Define G to be the context-free grammar $(\{ S, T, U \}, \{ a, b \}, R, S)$, where the rules in R are:

$$
\begin{aligned}
S &\to aSb \mid aTU \mid UTb \\
T &\to aTb \mid \epsilon \\
U &\to a \mid b.
\end{aligned}
$$

(a) For each of the following strings, say whether it belongs to E, and if so, give a leftmost derivation for it (6 pts. total): (i) ϵ, (ii) bb, (iii) $aaabb$, (iv) $aabbbb$.

(b) Find an ambiguous string and draw two different parse trees for it. (6 pts.)

(c) Prove by the structural induction technique that $L(G) \subseteq E$. (You may speak in terms of E "allowing up to one error." As usual, "reasonable proof shortcuts" are OK. 18 pts.)

(d) Is $L(G) = E$? Justify your answer briefly by referring to your parsing strategies in (a,b), but you need not give a formal proof. (6 pts., for 36 total on the problem, 80 on the set)

Answer: (a) (i) $\epsilon \notin E$ since it (too) requires $n \geq 1$; moreover $\epsilon \notin L(G)$ because S is immediately not nullable—all its rules include terminal(s); (ii) $bb \in E$ since it is one-place different from ab, and G derives it: $S \Longrightarrow UTb \Longrightarrow bTb \Longrightarrow bb$; (iii) $aaabb \notin E$ because it has odd length; (iv) $aabbbb \in E$ with regard to $aaabbb$ which is in A, plus $S \Longrightarrow aSb \Longrightarrow aaSbb \Longrightarrow aaUTbbb \Longrightarrow aabTbbb \Longrightarrow aabbbb$.

(b) The string ab is ambiguous—intuitively because it has no errors and hence has an allowance it can use (or rather refuse) in two places. ASCII parse trees:

```
   (S)                (S)
  / | \              / | \
 a  T  U            U  T  b
    |  |            |  |
   eps b            a eps
```

(c) Define the properties $P_S \equiv$ "Every x I derive belongs to E" (or more intuitively, "has at most one error and is not empty"), $P_T \equiv$ "Every x I derive belongs to A or could be empty" (that is, adds no errors), and P_U is that U stands for one char, either a or b. The rules for U are immediate, as is $T \to \epsilon$, so we need only focus on the other rules.

- $S \to aSb$. Suppose $S \Longrightarrow^* x$ using this rule first. Then $x = ayb$ where $S \Longrightarrow^* y$. By IH P_S on the right-hand side, y differs from a string in A in at most one place. Adding an a in front and a b in back preserves that property (one could say, doesn't add any further error), so x belongs to E and P_S on LHS is upheld.

- $S \to aTU$: Suppose $S \Longrightarrow^* x$ using this rule first. Then $x = ayc$ where $T \Longrightarrow^* y$ and $U \Longrightarrow^* c$, so that c is a or b. By IH P_T on RHS, y is in A or $y = \epsilon$; either way $|x| \geq 2$. So even if $c = a$, x differs from a string in A in at most that one place, which is good to put $x \in E$ thus upholding P_S on LHS.

- $S \to UTb$: Suppose $S \Longrightarrow^* x$ using this rule first. Then $x = cyb$ where $T \Longrightarrow^* y$ and $U \Longrightarrow^* c$, so that c is a or b. By IH P_T on RHS, y is in A or $y = \epsilon$; either way $|x| \geq 2$. So even if $c = b$, x differs from a string in A in at most that one place, which is good to put $x \in E$ thus upholding P_S on LHS. (It would be fine to shortcut this by saying, "similar to last rule.")

- $T \to aTb$: Suppose $S \Longrightarrow^* x$ using this rule first. Then $x = ayb$ where $T \Longrightarrow^* y$. By IH P_T on the right-hand side, $y \in A$ or $y\epsilon$. Adding an a in front and a b in back preserves that property (it actually gives $x \in A$ but upholding P_T on LHS is enough).

Thus the properties are upheld in all rules, so $L(G) \subseteq E$.

(d) In fact, $L(G) = E$. Sufficient justification is that you can derive any string $x \in E$ by working from the outside in: use $S \to aSb$ until you hit a place with an error (if any), then switch to $S \to aTU$ if the error is an a in back that should be a b, or to $S \to UTb$ if a char in the first half of x is wrong. Then finish off using T. In case of there being no error, the restriction $n \geq 1$ means $x \neq \epsilon$ so that there is "time" to use one of the rules $S \to aTU$ or $S \to UTb$.