Harvard University Extension School
Computer Science E-121

Problem Set 4

Due October 21, 2016 11:59pm

Problem set by Walter Thornton

Collaboration Statement: I worked alone and only with course materials

PROBLEM 1 (6 points, suggested length of 1 page)

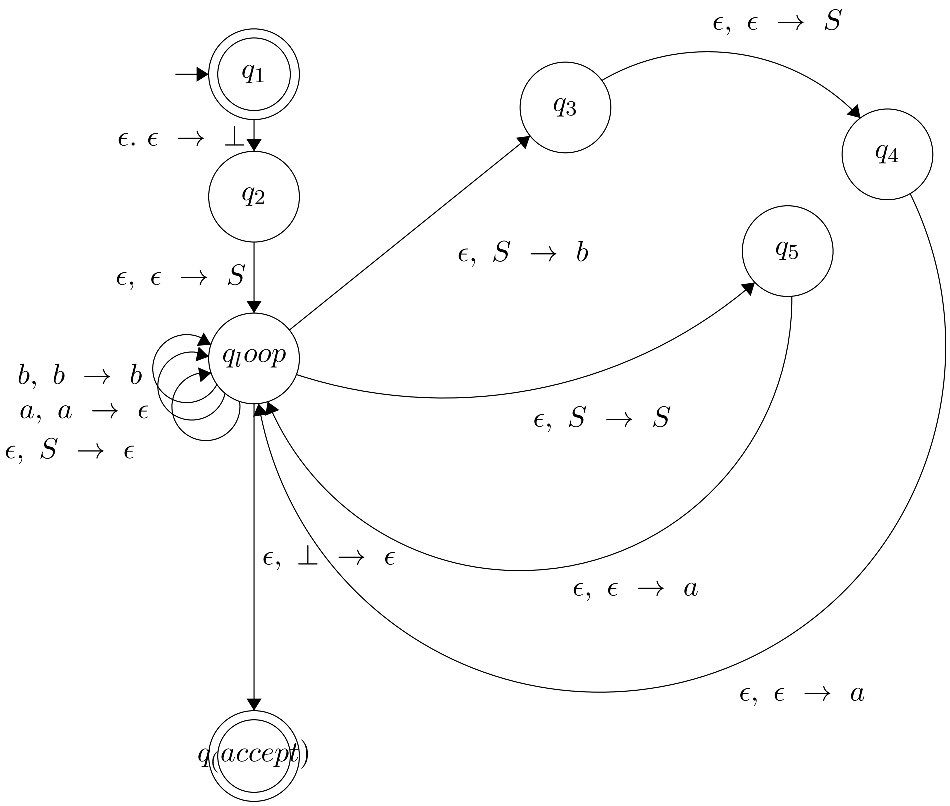
For each of the following problems, give a simple grammar constructing the described language, then construct a PDA from the grammar you have constructed. You do not have to give a formal description of each PDA; a diagram is enough. See Sipser for examples of PDA diagrams. Assume that $\Sigma = \{a, b\}$.

(A) The language $\{a^m b^n\}$, where $m, n \in \mathbb{N}$ and $m \geq n$.

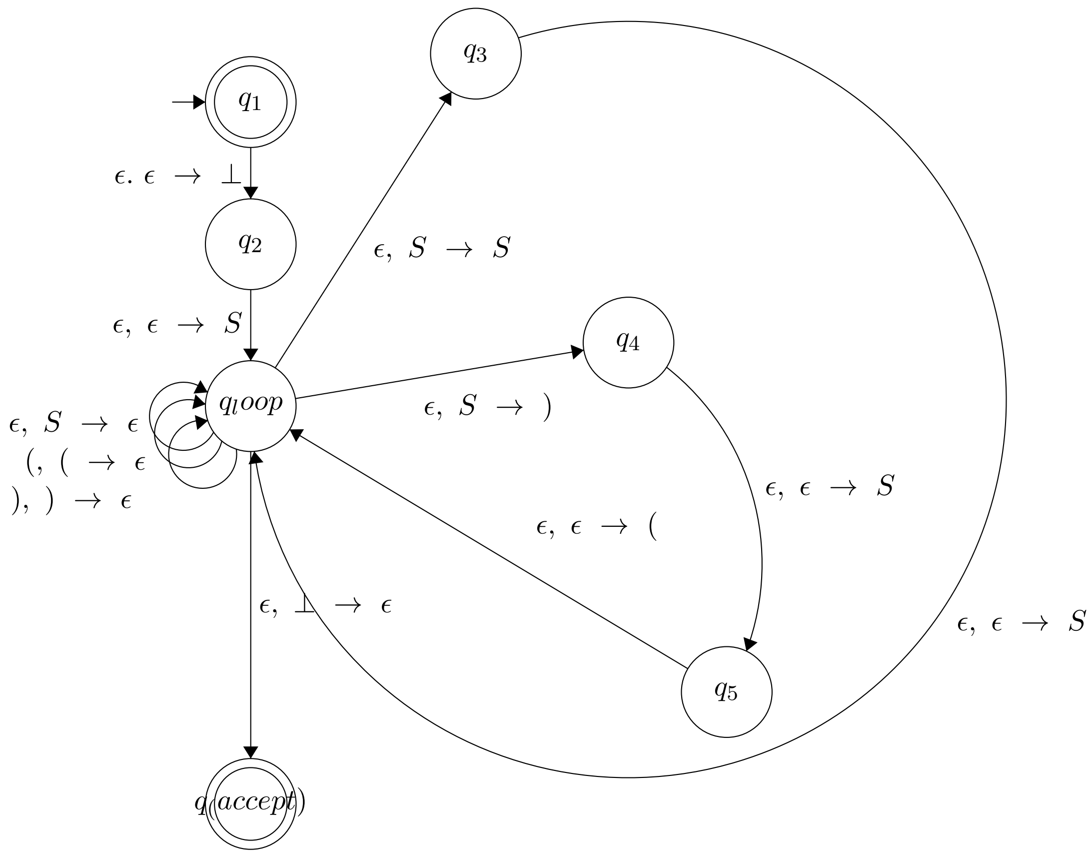
(B) The set of strings with balanced parentheses, where $\Sigma = \{(,)\}$.

Solution.

(A) $S \rightarrow aSb$
$S \rightarrow aS$
$S \rightarrow \epsilon$



(B) $S \to SS \mid (S) \mid \epsilon$



PROBLEM 2 (2+4+2 points, suggested length of 1 page)

In this problem, we show that a 2PDA (a PDA equipped with two stacks) accepts languages that a PDA cannot. A 2PDA can pop symbols from either stack at any state and push symbols to either stack as well. Our transition function now is defined on $Q \times (\Sigma \cup \{\epsilon\}) \times (\Gamma \cup \{\epsilon\})^2 \to P(Q \times (\Gamma \cup \{\epsilon\})^2)$. The other components remain the same as those of a conventional PDA.

(A) Prove that the language $\{a^n b^n c^n\}$ where $n \geq 0$ is not a context-free language. If you cite results from class, please summarize the proof here.

(B) Let $\Sigma = \{a, b, c\}$. Define a 2PDA 6-tuple that accepts the above language. You may draw a diagram for your transition function instead of explicitly writing out each possible transition.

(C) Explicitly state the relationship between CFLs and PDAs, and use it to show that there exist languages that 2PDAs accepts but a PDA cannot.

Solution.

(A) Assume $\{a^n b^n c^n\}$ where $n \geq 0$ is context free. Using the pumping lemma, we select string

$s = a^p b^p c^p$ where p is the pumping length, satisfying the condition $\bar{u}vxyz\bar{\leq}p$.
Condition 2 states that of uvxyz, either v or y in nonempty, and condition 3 states that $\mid vxy \mid \leq p$.

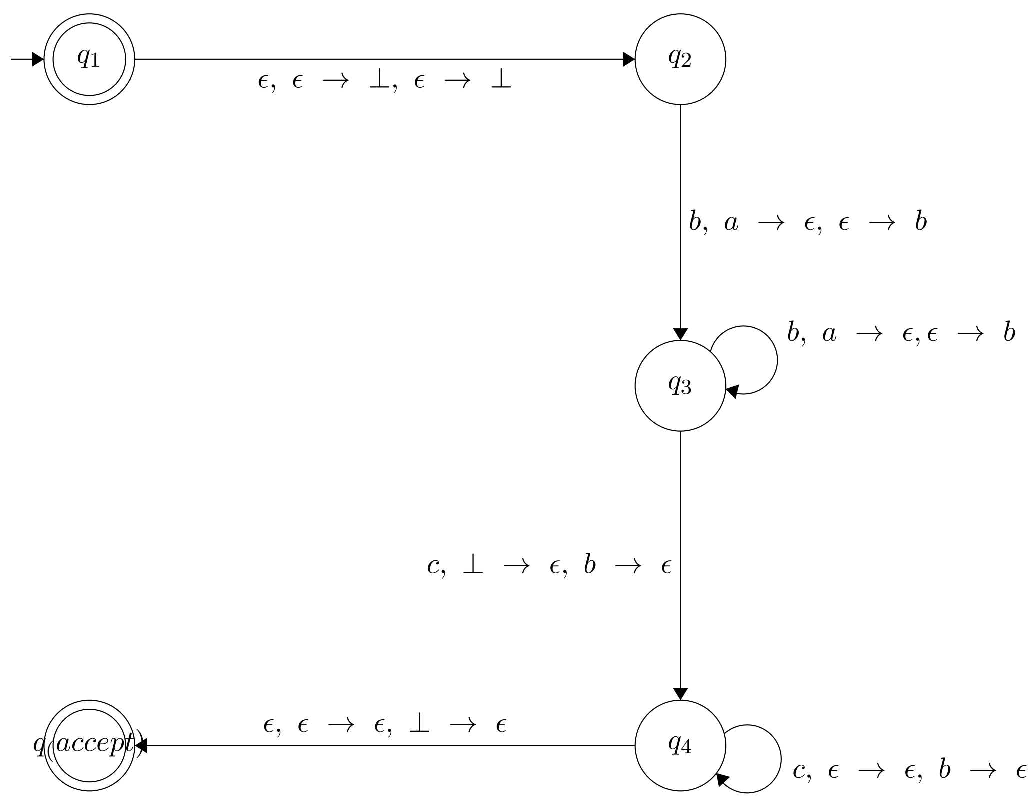
If we select s so that v and y contain a single alphabet symbol each, when pumped, the resulting string would have a disproportionate amount of that symbol. Our string in our language must have an equal number of a's b's and c's.
If we select s so that v contains ab or y contains bc, or any combination where v or y are multiple alphabet symbols, when pumped, the correct proportion of symbols could be perserved, but not in the correct order. The resulting string is not in our language.

In either case we arrive at a contradiction.
Therefore $\{a^n b^n c^n\}$ where $n \geq 0$ is not context free.
Sipser 128

(B) $M = (\{q_1, q_2, q_3, q_4, q_5\}, \{a, b, c\}, \{a, b.\perp\}, \delta, q_1, \{q_1, q_5\})$

Place all a's on stack1 until first b. For each b, push b onto stack2, pop a from stack1. If if b's are exhausted as stack1 empties, continue For each c read, pop b from stack 2. if last c is reached as stack2 empties, accept

Transitions are of the form: input, stack1, stack2



A language is context free if and only if some PDA recognizes it.

Since $\{a^n b^n c^n\}$ where $n \geq 0$ is not context free (Sipser 128) there is no PDA that recognizes it. From 2B, we see there is a 2PDA that recognizes it. Therefore, 2PDAs can recognize some languages that PDAs cannot.

(C)

PROBLEM 3 (3 points, suggested length of 10 lines)

A satisfied boolean expression is a string over the alphabet $\Sigma = \{0, 1, (,), \neg, \wedge, \vee\}$ representing a boolean expression that evaluates to 1, where $\neg$ is the *not* operator, $\wedge$ is the *and* operator, and $\vee$ is the *or* operator (see Sipser p.14). For instance, 1, $(1 \wedge (\neg 0))$, and $(\neg(\neg 1))$ are satisfied boolean expressions.

Meanwhile, $)(1$, $(0\neg)$, $\wedge(0 \vee 1)$, $(0 \wedge 1)$, and $(\neg(1 \vee 0))$ are examples of strings that are *not* satisfied boolean expressions: (The first three examples are strings that aren't valid expressions and simply don't make sense. The last two are well-formed expressions, but they evaluate to 0, rather than to 1.)

Write a context-free grammar that generates the language of satisfied boolean expressions. You may assume that expressions must be completely parenthesized. For instance, your grammar need not generate $\neg(0 \wedge \neg 0 \wedge 1)$ but should generate its equivalent, $(\neg(0 \wedge ((\neg 0) \wedge 1)))$.

(Note: You need not write down a full 4-tuple; just write down a clear set of rules and indicate what the start variable is.)

Solution.
$T \to (T \wedge T) \mid (T \vee B) \mid (B \vee T) \mid (\neg F) \mid 1$
$F \to (F \vee F) \mid (F \wedge B) \mid (B \wedge F) \mid (\neg T) \mid 0$
$B \to T \mid F$
Start symbol T

PROBLEM 4 (3+3+3 points, suggested length of 1/3 page)

A DNA strand can be represented as a string over the alphabet $\{a,\ g,\ c,\ t\}$. We call two DNA strands, σ and τ, complementary when $\sigma = \sigma_1 \ldots \sigma_n$, $\tau = \tau_1 \ldots \tau_n$, and each pair (σ_i, τ_i) is equal to one of (a, t), (t, a), (g, c), or (c, g).

(A) Prove that the language $\{xy^R$: x and y are complementary strands of DNA$\}$ is not regular.
(B) Prove that the language $\{xy^R$: x and y are complementary strands of DNA$\}$ is context-free by providing a grammar that generates it, ensuring to specify the start state. Briefly justify the correctness of your grammar (1-2 sentences).
(C) Provide a derivation for the string *gtagctac* using the grammar you wrote in part (B).

Solution.

(A)
We assume the language $\{xy^R$: x and y are complementary strands of DNA$\}$ is regular.
Using the pumping lemma for regular languages, we must divide string s of at least length p, into xyz.
We have three ways we can divide any string in the language with xyz. Either y lies before the center of the string, or y lies after the center of the string, or y straddles the center of the string.

In the first two cases, y, when pumped, will increase alphabet symbols either to the left or right of the center. This will necessarily result in a string with uncomplemented symbols in the string. This is not in the language.
If y lies at the center and is not empty, than it will necessarily include symbols from both halves of the string. When pumped, the resulting string will have a repeating pattern at its center and is not in the language. In either case we reach a contradiction. The language is not regular.

(B) The language $\{xy^R$: x and y are complementary strands of DNA$\}$ is context-free.
$S \to gSc \mid cSg \mid tSa \mid aSt \mid \epsilon$

(C) $gSc \Rightarrow agSct \Rightarrow tagScta \Rightarrow gtagSctac \Rightarrow gtagctac$

PROBLEM 5 (4+4 points, suggested length of half a page)

Unlike the regular languages, the family of context-free languages is not closed under complement.
(A) Prove that the language $L = \{ww : w \in \{a, b\}^*\}$ is not context-free.
(B) Prove that the language $\bar{L} = \{a, b\}^* - L$ is context-free by giving the rules of a grammar that generates it. In a sentence or two, justify the correctness of your grammar. (Hint: argue first that $\bar{L}$ is the set of all strings of the form $xayubv$ or $xbyuav$ where $|x| = |y|$ and $|u| = |v|$, along with all strings of odd length.)

Solution.

(A)
Assume the language $L = \{ww : w \in \{a, b\}^*\}$ is context free.
Use pumping lemma with string $s = uvxyz$ with $\mid s \mid > p, \mid vxy \mid \leq pand \mid vy \mid \geq 1$. Consider $s = a^p b^p a^p b^p$ satisfying $\mid s \mid > p$.
If vxy stradles the center, then v would be b and y would be an a. When pumped, the resulting string would not in L.
If vxy is in the first half, the first string w grows disproportioinate to the second w. The resulting string is not in L.
If vxy is in the second half, the second string w grows disproportioinate to the first w. The resulting string is not in L.
In any case, the pumping lemma does not produce a string in L, therfore the language is not context free.

(B)
$S \to O \mid E$
$O \to aC \mid bC \mid a \mid b$
$C \to aO \mid bO$
$E \to AB \mid BA$
$A \to EAE \mid a$
$B \to EBE \mid b$
$E \to a \mid b$

The complement of L is the union of two sets, the set of all odd length strings and the set of even length strings of for xy where $| x |=| y |$, but $x \neq y$.
The set of odd lenght strings can be generated by the above CFG, by following the O rule. The set of even length strings can take the following forms xayubv or xbyuav where $| x |=| y |$ and $| u |=| v |$.
The CFG uses the E rule to generate even length strings in this manner.

PROBLEM 6 (3+3 points, suggested length of 1/4 page)

For each of the following propositions, state whether it is true or false, and provide a complete justification/counterexample as proof.

(A) If L is context-free and R is regular, then $L - R$ must be context-free.

(B) If L is context-free and R is regular, then $R - L$ must be context-free.

Solution.

(A) True

$L - R = L \cap \overline{R}$
Regular languages are closed under complement. This results in the intersection of a context free language with a regular language. We know that a context free language with a regular language is closed under intersection. (Lecture 10)
Therefore L - R is context free.

(B) $R - L = R \cap \overline{L}$
Consider $R = \{\Sigma^*\}$ and L is the language from problem 5B, the complement of $\{ww : w \in \{a, b\}^*\}$, which is context free.
Therefore $R - L = \{\Sigma^*\} \cap \{ww\}$.
This intersection results in {ww} which we know is not context free.

PROBLEM 7 (Challenge!! Not required; worth up to 3 points, suggested length of 1/2 page)

Show that every context-free language over a unary alphabet $\Sigma = \{a\}$ is regular.

Note: On every problem set we will provide a challenge problem, generally significantly more difficult than the other problems in the set, but worth only a few points. It is recommended that if you attempt these problems, you do so only after completing the rest of the assignment.

Solution.
the pumping lemma for context free languages uses $s_C = uvxyz$
the pumping lemma for regular languages uses $s_R = xyz$
if a language over a unary alphabet is context free. we know it satisfies the pumping lemma for context free languages.

Since it is a unary alphabet, we can use vxy of s_C as our y_R for the regular language pumping lemma. We can then pump y_r as we would normally do for regular languages. This will necessarily result in a string in the language. Therefore any context free language over a unary alphabet is also regular.

PROBLEM 8 (4 points, suggested length of 1/3 page)

Let F be a context-free grammar in Chomsky Normal Form with n variables. Prove that if there exists a string $s \in L(F)$ that takes more than 2^n steps to derive from F, then $L(F)$ contains an infinite number of strings.

Solution.

If a parse tree has n variables and takes 2^n steps to derive, than the height of the associated parse tree must be $n + 1$, allowing 1 for the terminal step.
If a particular path of the parse tree contains $n + 1$ variables, then by the pigeon hole principal, a variable must be repeated.
If a variable is repeated, it necessarily creates a loop which can produce an infinite number of strings.

PROBLEM 9 (0 points)

Please estimate the number of hours spent on this assignment to the nearest half hour.

Solution.
12