SOLUTIONS, CS301 midterm 2, McConnell, Spring '09

1. Consider the following statement: *For all context-free languages L_1 and L_2, $L_1 \cap L_2$ is also context-free.* Disprove it as follows. Give two context-free languages, show that they are context-free by giving context-free grammars for them, and then point out that their intersection is a language that we have shown to be non-context-free.

   Solution: $S_1 \to DC$; $D \to aDb|e$; $C \to cC|e$. *This is the language* $\{a^i b^i c^j | i,j \geq 0\}$.

   $S_2 \to AE$; $A \to aA|e$; $E \to bEc|e$. *This generates the language* $\{a^i b^j c^j | i,j \geq 0\}$.

   The intersection is $\{a^i b^i c^i | i \geq 0\}$, *which we have shown to be non-context-free.*

2. Consider the following statement: *For every context-free language $L \subseteq \{a,b\}^*$, $\overline{L}$ is also context-free.* Prove or disprove it. You may use the fact that the union of two context-free languages is context-free.

   Solution: *Suppose the complement of every context-free language is context-free. If L_1 and L_2 are two context-free languages, then so are $\overline{L_1}$ and $\overline{L_2}$. Since the union of two context-free languages is context-free, so is $\overline{L_1} \cup \overline{L_2}$. Since the complement of a context-free language is context-free, so is $\overline{\overline{L_1} \cup \overline{L_2}}$. This is $L_1 \cap L_2$, which is easily verified using a Venn diagram. (This is one of de Morgan's laws.) We conclude that the intersection of two context-free languages is context-free, but we know this isn't true from Problem 1. Therefore, our assumption that the complement of every context-free language is context-free must be false.*

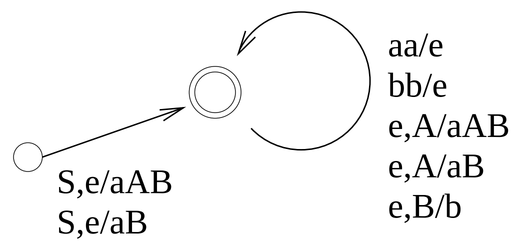
3. Draw a **deterministic** PDA for the language $\{wc^k d^j w^R | w \in \{a,b\}^*$ and $k \geq 0$ and $j \geq 1\}$ Recall that w^R is the reverse of w. Recall also that we adopted the convention that a deterministic PDA starts out with a special symbol, Z, at the bottom of the stack, and that a special symbol, \$, is appended to the input string.

   Solution:*See Homework 7 solutions.*

4. We studied an algorithm for generating a PDA that accepts the language generated by a context-free language. Use it to generate a nondeterministic PDA for the following grammar:

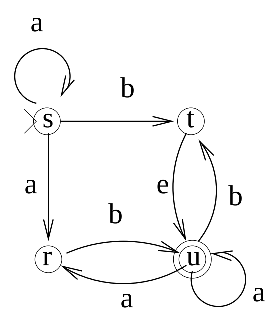
   $S \to aAB|aB$;
   $A \to aAB|aB$
   $B \to b$;

   Solution:



a

5. Give a context-free grammar that generates the language accepted by the following NFA:



Solution: $S \to aS|aR|bT$; $R \to bU$; $U \to aR|aU|bT|e$; $T \to U$.

6. Consider the following grammar:

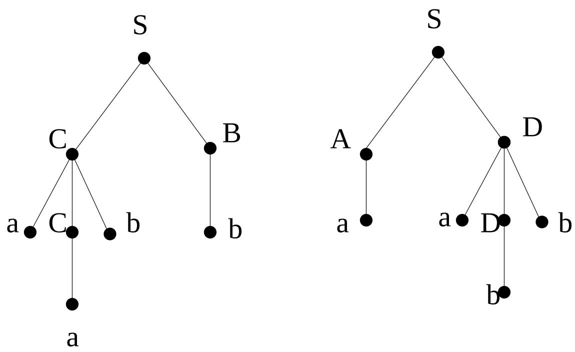
$S \to CB|AD$
$A \to aA|a$
$B \to bB|b$
$C \to aCb|a$
$D \to aDb|b$

(a) Using setbuilder notation, give the language generated by each of A, B, C, D, and S.

Solution:
A generates $\{a^i | i \geq 1\}$
B generates $\{b^i | i \geq 1\}$
C generates $\{a^{i+1}b^i | i \geq 0\}$
D generates $\{a^i b^{i+1} | i \geq 0\}$
S generates $\{a^i b^j | i, j \geq 1\}$.

(b) Show that the grammar is ambiguous by giving two parse trees for a single string of length at least four that is in the language.

Solution: A nonempty string of the form $a^i b^i$ can be generated by either CB or AD.



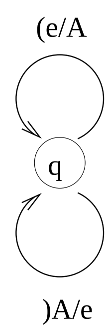
- $S \Rightarrow CB \Rightarrow aCbB \Rightarrow aabB \Rightarrow aabb.$
- $S \to AD \Rightarrow aD \Rightarrow aaDb \Rightarrow aabb.$

7. Show that the language $\{a^k b^{2k} a^{3k} | k \geq 0\}$ is not context-free.

   Solution: *Suppose the language is context-free. Then the pumping lemma applies. Let $w = a^n b^{2n} a^{3n}$. Because the number of characters from the beginning of v to the end of y is at most n, v and y can reside in at most two of the blocks a's and b's. After pumping v and y, the third block's size won't have the right ratio with the other two's sizes. The string is not in the language, contradicting the pumping lemma. Our assumption that the language is context-free must be wrong.*

8. We proved that for every PDA, there is a context-free grammar that generates the same language. This was the subject of a writeup I posted.

   (a) Write down the grammar that the algorithm generates for the following NPDA, which accepts a language on the alphabet $\{(,)\}$ of parentheses:

   Use the mnemonic notation we used for the nonterminals. (Each nonterminal except the start symbol is of the for $[q, e, r]$ or $[q, A, r]$.) *Though the algorithm can generate grammars that are large, this PDA is so simple that the grammar is fairly small.*

   Solution: $S \to [q, e, q]$; $[q, e, q] \to ([qAq]|e$; $[q, A, q] \to)[q, e, q]|([q, A, q][q, A, q]$.

   Give a leftmost derivation by this grammar of the string $(()())()$

   $S \Rightarrow [q, e, q] \Rightarrow ([q, A, q] \Rightarrow (([q, A, q][q, A, q] \Rightarrow (()[q, e, q][q, A, q] \Rightarrow (()([q, A, q][q, A, q]$
   $\Rightarrow (()()[q, E, q][q, A, q] \Rightarrow (()()[q, A, q] \Rightarrow (()())[q, e, q] \Rightarrow (()())([q, A, q] \Rightarrow (()())()[q, e, q],$
   $\Rightarrow (()())().$

9. Consider the following statement: *For every regular language L, L is context-free.* There are a number of ways to prove this. Recall that we proved two related results:

- For every context-free grammar, there is a nondeterministic PDA that accepts the same language. The proof is illustrated by the solution to problem 4, above.
- For every nondeterministic PDA, there is a context-free language that generates the same language. The proof is illustrated by the solution to problem 8, above.

(a) Prove that every regular language is context-free using an approach based on machines and language classes.

Solution: *Regular languages are those accepted by NFA's. An NFA is a special case of a PDA that never pushes or pops from a stack. The language accepted by a PDA is context-free.*

(b) *Would the proof still work if we hadn't shown that for every context-free language, there is a PDA that accepts the same language?*

Solution: *Yes. If this weren't true, it would only mean that some context-free languages can't be regular.*

(c) *Would the proof still work if we hadn't shown that for every PDA, there is a context-free language that generates the same language?*

Solution: *No. We used the fact that every PDA accepts a context-free language.*

10. Consider the language $\{a^k w a^k | k \geq 0$ and w is a palindrome on $\{a,b\}^*$ that has length $k\}$.

Here is an incorrect proof that the language is not context free.

Proof: Suppose it is context-free. Then the pumping lemma applies. Consider the word $a^n b^n a^n$, where n is the number referred to in the pumping lemma.

(a) Since b^n is a palindrome, $a^n b^n a^n$ is in the language.

(b) Since the length of each block is n, v and y are both in the same block of uniform letters, or else they're confined to two consecutive blocks.

(c) If at least one of them has letters from one of the blocks of a's, pumping out v and y results in a string that's not in the language.

(d) If v and y only have letters from the block of b's, pumping out v and y results in a string that's not in the language.

(e) There are no other options for where v and y could lie.

We conclude that we have proved that the language is not context-free.

(a) Which of the above numbered statements are gaffes? (Why?)

Solution: *Statement 10d is incorrect. Suppose you pump three characters out of the block of b's. Then the resulting string $a^n b^{n-3} a^n$ can be written as $a^{n-1}(ab^{n-3}a)a^{n-1}$, which is a block of $n-1$ a's, followed by a palindrome of size $n-1$, followed by a block of $n-1$ a's. The string is in the language, and we have failed to discredit the pumping lemma.*

(b) How can we be smarter about using the pumping lemma, and get a legitimate proof, without changing our example string $a^n b^n a^n$?

Solution: *The pumping lemma says* **for all** *strings w in the language,* **there exists** *a way to write w as uvxyz, such that* **for all** $k \geq 0$, $uv^k xy^k z$. **We get to**

pick k**.** *We failed above because we picked* $k = 0$*. Let's pick* $k = 2$ *for Case d, neither* v *nor* y *has any* a*'s. Since at least one of them is nonempty, the blocks of* a*'s in* uv^2xy^2z *are each less than one third of the whole string, so this string is not in the language.*