

Homework #2: Pumping and CFL's
Due: see Canvas for due date and requirements

This handout starts with some definitions, followed by a list of problems. Again, you do not have to do all the problems! See Canvas for exact requirements. These problems concern Chapter 2 (and a bit of Chapter 1). Students in CS424 can skip Section 2.4, on DCFL's. Students in CS524 should read the first few pages of Section 2.4, but you can stop where it says "Deterministic Context Free Grammars" (a big topic, we will not cover it).

Definitions

Given a language $L \subseteq \Sigma^*$, we define the following two-player "pumping games". In both games, the winner is the last player to make a legal move. So if they finish the game, N wins.

Regular Pumping Game for L :

1. R chooses an integer $p \geq 0$.
2. N chooses a string $s \in L$ such that $|s| \geq p$.
3. R chooses strings x, y, z such that $s = xyz$, $|xy| \leq p$, and $|y| > 0$.
4. N chooses an integer $i \geq 0$ such that $xy^iz \notin L$.

Pumping Claim 1: If L is regular, then R has a winning strategy. (This is equivalent to the Chapter 1 pumping lemma, page 78.)

Context-Free Pumping Game for L :

1. C chooses an integer $p \geq 0$.
2. N chooses a string $s \in L$ such that $|s| \geq p$.
3. C chooses strings u, v, x, y, z such that $s = uvxyz$, $|vxy| \leq p$, and $|vy| > 0$.
4. N chooses an integer $i \geq 0$ such that $uv^ixy^iz \notin L$.

Pumping Claim 2: If L is context-free, then C has a winning strategy. (This is equivalent to the Chapter 2 pumping lemma, page 125.)

Roughly speaking: N tries to pick a unpumpable string s , while R (or C) tries to find a pumpable substring (or pair of substrings) in s . We often use the above pumping claims in the contrapositive direction:

If N has a winning strategy for the regular game, then L is not regular.

If N has a winning strategy for the context-free game, then L is not context-free.

The *converses* of the two pumping claims are false! We consider counterexamples below.

Problems

You may cite results which are stated in the main text of Chapters 1 and 2 (but not the problems). You may consult wikipedia where directed below, but your work should otherwise be your own.

Problem 1 Exercises 2.6 (page 155, grammars, just the unsolved parts (b) and (d)).

Problem 2 Exercises 2.9 (page 155). If your grammar is ambiguous, give an example string with two different parse trees.

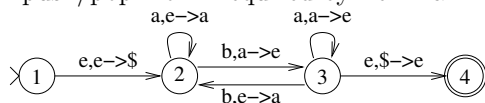
Problem 3 Exercise 2.30 (page 157, pumping, unsolved parts (a) and (d)). State an explicit winning strategy for N , and give a brief argument that it works.

Problem 4 Convert the following grammar to Chomsky Normal Form (CNF), using the process described by Sipser. Show at least the following: (1) the grammar after you have fixed the start symbol and eliminated ε -rules, (2) the grammar after you have also eliminated unit rules, and (3) the final CNF grammar.

$S \rightarrow aSA \mid AB \mid aB$
 $A \rightarrow abb \mid \varepsilon$
 $B \rightarrow bbA \mid AA$

Problem 5 Consider the language F in Exercise 1.54 (page 91). Argue that it is not regular, and find a winning strategy for R, in the regular pumping game for F . (Remark: this shows the converse of Pumping Claim 1 is false.)

Problem 6 We'll consider a similar problem in lecture. The following PDA P is already in the nice "push/pop" form required by Lemma 2.27 (ϵ denotes ϵ):



Problem 6(a). Trace out an accepting computation of P on the input $x = aabbabaababa$ (you might use my "mountain diagram" format). For each step show the input consumed (if any), the resulting state, and the resulting stack.

Problem 6(b). Using the textbook construction, convert the PDA P to a grammar G , with start symbol A_{14} . Try to omit all rules from G that are not necessary to generate $L(P)$.

Problem 6(c). Draw a parse tree for $aabbabaababa$, corresponding to the computation in the first part.

Problem 7 Let $L = \{a^i b^j c^k d^l : i=0 \text{ or } j=k=l\}$. Consider the context-free pumping game for L .

Problem 7(a). State a winning strategy for C.

Problem 7(b). See "Ogden's Lemma" on wikipedia. State a "game" version of it. (That is, state a modified version of our context-free pumping game, which Ogden's Lemma says C can win, if L is context-free.)

Problem 7(c). Describe a winning strategy for N in your game. (Remark: this shows the converse of Pumping Claim 2 is false.)

Problem 8 Consider this modified version of the regular pumping game:

1. R' chooses an integer $p \geq 0$.
2. N chooses a string $s \in L$ such that $|s| \geq p$.
3. R' chooses strings x, y, z such that $s = xyz$, $|xz| \leq p$, and $|y| > 0$.
4. N chooses an integer $i \geq 0$ such that $xy^i z \notin L$.

(The difference is that R is renamed R', and we require $|xz| \leq p$ instead of $|xy| \leq p$.) If L is regular, must R' have a winning strategy? Either prove this is true (by modification of the book argument, pages 78–79), or give a counterexample (a specific regular L , and a winning strategy for N).

Problem 9 We are given a PDA P . From Chapter 2, we know there is an equivalent CNF grammar G .

Problem 9(a). Lookup "shift-reduce parser" on wikipedia (also known as "bottom-up"). Using that, construct a PDA P' accepting $L(G)$. You do not have to give P' as a tuple, just a generic description and diagram of P' (like Sipser's parser) is enough. (Unlike Sipser's parser, for this one it is easier to imagine the stack is a string, and the top is the *right* end of the string.)

Problem 9(b). Argue that on an input of length n , any computation of P' (accepting or not) takes at most $O(n)$ steps. (Remark: this problem shows that we can convert a PDA to an equivalent PDA that always stops within $O(n)$ steps.)

Problem 10 In Sipser's Proof of Theorem 2.9 (page 109), he gives a construction to convert a given CFG G to Chomsky normal form. In the second paragraph, he describes how to eliminate ϵ -rules of the form $A \rightarrow \epsilon$ (where A is not the start variable). In particular, he says "... we add $R \rightarrow \epsilon$ unless we had previously removed the rule $R \rightarrow \epsilon$." Give a careful argument that this is correct, i.e. that we do not change the language in this paragraph. (You want to argue that if w had a parse tree before, then it has one afterwards, and vice-versa.)