# CSE 105: Homework Set 5

Joshua Wheeler

May 5, 2014

1. Context Free Design

   Part 1A:
   The language can be recognized by the following Context Free Grammar:

   $$S \to 0S11 \mid 01S1 \mid 1S01 \mid 10S1 \mid 1S10 \mid 11S0 \mid \epsilon$$

   The language that this describes is a string that has twice as many 1's as 0's. This CFG recognizes this language through a recursion of the smallest possible strings. Every time there is one 0 in the string, we need two ones concatenated to it. There are three possibilities for this concatenation. These possibilities are 011, 101, 110. These are the smallest possible nonempty strings accepted by this language. Since we want any number of strings of any combination we must insert a recursion inside every one of these smallest strings. Hence, we have the above CFG.

   Part 1B:
   The language can be recognized by the following Context Free Grammar:

   $$S \to aSb \mid aS \mid a$$

   In this language we want to accept any string of a's followed by b's such that there are more a's than b's. The aSb term can derive an equal amount of a's to b's. This term is useful when we want to add a b to the string. The aS term allows the grammar to add any amount of a's to the string. The base case term, "a", ensures that there will always be more a's than b's.

2. Proving Ambiguity
   There are many ways to show that this grammar is ambiguous. We know a grammar is ambiguous if it has 2 or more different leftmost derivations. Below is an example of two different leftmost derivations for this grammar. The

parenthesis are to make the difference in the derivations more clear and are not part of the grammar:

$$S \rightarrow aS \rightarrow a(Sb) \rightarrow ab \tag{1}$$
$$S \rightarrow Sb \rightarrow (aS)b \rightarrow ab \tag{2}$$

Extra Credit:
If S is not the start symbol it is still possible to show that G is ambiguous. The left most derivation can still be distinct in the following way
$... \rightarrow ...aS... \rightarrow ...a(Sb)... \rightarrow ...ab... \rightarrow ...$
or $... \rightarrow ...Sb... \rightarrow ...(aS)b... \rightarrow ...ab... \rightarrow ...$
The "..." indicates that there may be more symbols in the string or more derivations before and after the current derivation step.

3. Describing Pushdown Automata
When this PDA starts reading symbols from the input it will start by reading a's. As each a is read push an "a" onto the stack. After all the a's are read, start reading b's. As each b is read, push a "b" onto the stack. Now, start reading c's. For each c read, pop a symbol off the stack. If reading the input is finished exactly when the stack is empty, accept the input. If any of the letters are read in an order different than the one described, if the stack becomes empty before reading input has been completed or if there are symbols still on the stack after the input had been completely read, reject the input.

4. Proving Closure Under Unions For CFL's
If a language is context free, then that means that there is a PDA that recognizes it. If we can prove that the union of two PDA's that recognizes a context free language is closed then that means that the union of two context-free languages must also be closed under union.

The proof to show that the union operation is closed for two PDA's is similar to that of the closure proof for a DFA with some slight modifications. This is a proof by construction.

Let $P_1$ be the PDA which recognizes the context free language $C_1$
Let $P_2$ be the PDA which recognizes the context free language $C_2$
Let $P = P_1 \cup P_2$ (this also implies that $C = C_1 \cup C_2$)

$P_1$ is the six tuple $(Q_1, \Sigma_{\epsilon 1}, \Gamma_{\epsilon 1}, \delta_1, q_1, F_1)$
$P_2$ is the six tuple $(Q_2, \Sigma_{\epsilon 2}, \Gamma_{\epsilon 2}, \delta_2, q_2, F_2)$

The union of these two PDA's, P can by defined by the following 6 tuple:

$$P = (Q, \Sigma_\epsilon, \Gamma_\epsilon, \delta, q_0, F)$$

Where

$$Q = Q_1 X Q_2$$

$$\Sigma_\epsilon = \Sigma_{\epsilon 1} \cup \Sigma_{\epsilon 2}$$

$$\Gamma_\epsilon = \Gamma_{\epsilon 1} \cup \Gamma_{\epsilon 2}$$

$$\delta : \delta((q_i, q_j), a, x) = (\delta_1((q_i, a, x)), \delta_2(q_j, a, x)) \text{ such that } (q_i, q_j) \in Q, a \in \Sigma, x \in \Gamma$$

$$q_0 = (q_1, q_2)$$

$$F = (F_1 X Q_2) \cup (Q_1 X F_2)$$

Since the PDA, P, which is the union of $P_1$ and $P_2$ exists and is context free, it follows that the union of two context free languages must also be context free.