Cpt S 317 Homework #8

Please print your name!

Explicit construction == Psuedo-code in the form of δ or instructions like "on state p, when read a, the top of stack is b, the next move is to ....."

1. Construct a DPDA (yes, deterministic DPA) to accept language $\{0^n 1^{3n} : n \geq 1\}$. You need write down the explicit construction.

2. Construct a PDA to accept language $\{w \in \{0,1\}^* : \#_0(w) \geq \#_1(w)\}$. (It is fine you describe how your machine works in English.)

3. Construct a PDA to accept language $L = \{w \in \{0,1\}^* :$ each prefix of w is in the language defined in Problem 2. $\}$. You need also write down the explicit construction. (This is actually an easy problem. First, you have to understand the language. For instance $0001100111001 \in L$, but $001110 \notin L$. why? since $w \in L$ requires that each prefix of w contains more or the same number of 0's than 1's.)

4. Let L be a language accepted by a PDA M. Define $Prefix(L) = \{x :$ there exists y such that $xy \in L\}$. Describe a construction of a PDA M' accepting $Prefix(L)$. (You only need to describe in English how M' works.)

5. Consider the following pseudo-C code:

```
int x=0;
char d;

while (1){
  d=getchar();
  if (d==EOF) break; //when reads EOF (end of the input)
                     //get out of the while-loop
  if (d=='a' || d=='b') x++;
  if (d=='c') x--;
}

//get here after breaking the loop
if (x==0) putchar('y'); //write yes to stdout
```

This program writes 'y' (yes) to stdout (your screen) for some particular sequences of input characters (i.e., strings). These strings form a language L. Construct a PDA to accept L. (What we learn from this problem is that a program with one integer variable can always be simulated by a PDA.)