CS 4384 — Automata Theory, Homework 5

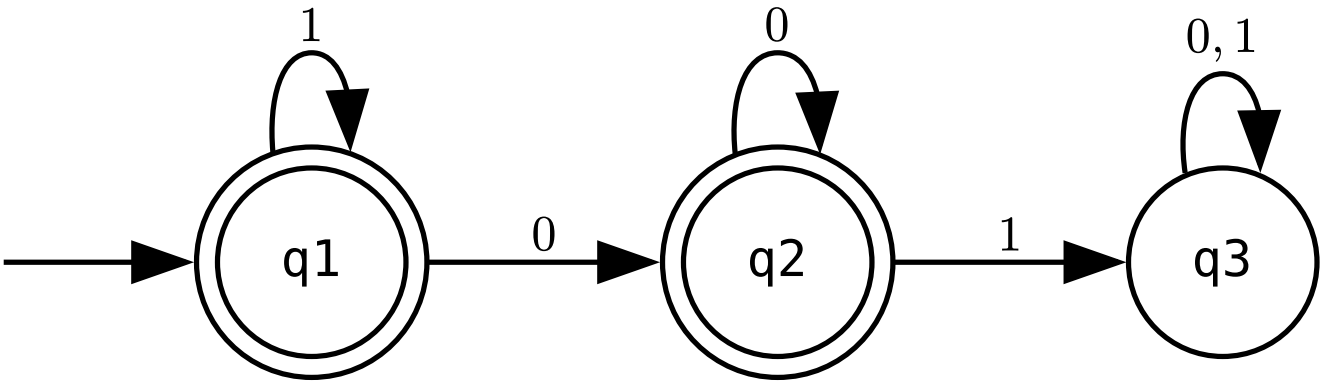
https://iniyanjoseph.com | iniyanijoseph@gmail.com

Let $\Sigma = \{0, 1\}$ be an alphabet, $L \subseteq \Sigma^*$ a CFL and, $w \in \Sigma^*$ a string. Prove that the language

$$L_1 = \{v \in L \mid v \text{ does not contain } 0w1 \text{ as subtring}\}$$

is a CFL.

This language is regular, thus it is a CFL. To show it is regular, let us create a DFA for this language.
Observe that the word w in the definition of the language cannot contain the character "1". This means that the string can be described by the regular expression `1*0*`, with the corresponding DFA:



Prove that the following language is not context-free.

$$L = \{wcw \mid w \in \{a, b\}^*\}$$

Let $n > 0$ Let $z = a^n bca^n b$ Let $z = u \cdot v \cdot w \cdot x \cdot y$ such that $|v \cdot w \cdot x| \leq n$ and $|v \cdot x| \geq 1$ Let $i = 0$ Then then there are several cases:

- $v \cdot w \cdot x \subseteq a^n$. This is clearly not in the language, as $a^{n-\ell}b \neq a^n b$.
- $c = w$. Here, we get $a^{n-\ell+1}ca^{n-\ell}b$, which is not in the language.
- $c \in v \cdot w \cdot x$ and $c \subseteq w$. This is not in the language, since c is no longer in the string.

Thus this language is not context-free by the pumping lemma for CFLs.

Prove that the following language is not context-free without applying the pumping lemma.

$$L = \{w \in \{a, b, c, d\}^* \mid \#_a(w) = \#_b(w) = \#_c(w) = \#_d(w)\}$$

This applies because of the closure of CFLs under intersection with regular languages. Take the regular language $M = L(a^*b^*c^*)$. $L \cap M = \{a^n b^n c^n\}$, which is not context-free. Thus, L is not context free.

Show that if G is a CFG in Chomsky normal form, then for any string $w \in L(G)$ of length $n \geq 1$, exactly $2n - 1$ steps are required for any derivation of w

For every internal vertex (nonterminal) in the derivation tree, there are exactly 2 children, and every leaf (terminal) in the tree, there are exactly 0 children. This is a full binary tree. Because

the string has n characters, there must be exactly n terminals in the derivation tree. Based on this, we would like to count the number of internal nodes in the tree. In a full binary tree, there are $\frac{m\ell-1}{m-1}$ vertices. Solving for n, we get $\frac{2n-1}{1} = 2n-1$ steps in the derivation tree.