HW Solution

CS/ECE 374: Algorithms & Models of Computation, Spring 2019 Version: **1.0**

Submitted by:

- **≪Alan Lee≫**: **≪alanlee2≫**
- **≪Joshua Burke≫**: **≪joshuab3≫**
- **≪Gerald Kozel≫**: **≪gjkozel2≫**

(100 PTS.) As easy as 1,2,3,6.

Let $L = \left\{ a^i b^j c^k \;\middle|\; k = i + j \right\}$.

1 (20 PTS.) Prove that L is context free by describing a grammar for L.

2 (80 PTS.) Prove that your grammar is correct. (See extra problems for an example of how this is done.)

12 Solution:

1. $G = (V, T, P, S)$

   $V = \{S, A, B\}$

   $T = \{a, b, c\}$

   $P =$

   $$S \to A$$
   $$A \to aAc \mid B \mid \epsilon$$
   $$B \to bBc \mid \epsilon$$

2. To prove our grammar is correct, we will prove L $\subseteq L(G)$ and $L(G) \subseteq L$. We will separately prove these statements.

   Claim 1. L(G) $\subseteq L$, meaning every string in L(G) has the property that the number of a's and b's is equal to the number of c's.

   Proof: we will prove that for any string w, we have $\Delta(w) = (\#_a(w) + \#_b(w)) - \#_c(w)$. We will prove that $\Delta(w) = 0$ for every string w $\in$ L(G).

   Inductive Hypothesis: let w be an arbitrary string in L(G) and consider an arbitrary derivation of w of length k. Assume that $\Delta(u) = 0$ for every string u in L(G) that can be derived with fewer than k production rules. There are 3 cases to consider depending on the first production in the derivation of w.

   Base Case: $w = \epsilon$ from some set of production rules, then by definition, $\Delta(w) = (\#_a(w) + \#_b(w)) - \#_c(w) = 0$.

   Sub case 1: Suppose derivation of w begins with $S \to^* w$. Then w = u for some string u $\in L(G)$, that can be derived with fewer than k production rules. The inductive hypothesis implies $\Delta(u) = 0$. It then follows that $\Delta(w) = 0$.

   Sub case 2: Suppose w uses $S \to A$ making $w = awc$. Then $w = auc$ for some string u $\in L(G)$ The inductive hypothesis implies $\Delta(u) = 0$. It then follows that $\Delta(w) = 0$.

   Sub case 3: Suppose w uses $S \to A \to B$ making $w = bwc$. Then $w = buc$ for some string u $\in L(G)$. The inductive hypothesis implies $\Delta(u) = 0$. It then follows that $\Delta(w) = 0$.

   In all cases, all strings produced will hold the property $\Delta(w) = 0$.

   Claim 2. L $\subseteq$ L(G), meaning G generates every string with the property that the number of a's and b's is equal to the number of c's.

   Proof: we will prove for any arbitrary string w in language L, w is an element of L(G).

   Base Case: $w = \epsilon$, then w is an element of L(G) because of the production rule $S \to A \to \epsilon$.

   Non-Empty Case:

   Sub case 1: By using production rules, w uses $S \to A$ making $w = auc$ where u is a nonempty string with the $\#_a(u) + \#_b(u) = \#_c(u)$. This implies u $\in$ L(G). Thus the production rule A $\to aAc$ implies that w $\in L(G)$.

   Sub case 2: $w = buc$ where u is a non empty string with $\#_b(u) = \#_c(u)$ and $\#_a(u) = 0$. this implies that u $\in$ L(G) thus the rule B $\to bBc$ implies w $\in L(G)$.

   Sub case 3: $w = u$ where u is a non empty string with $\#_b(u) = \#_c(u)$ and $\#_a(u) = 0$. This is based on the result of sub case 2 that B will always satisfy the condition $\#_b(u) = \#_c(u)$ and $\#_a(u) = 0$. This implies that u $\in$ L(G) thus the rule A $\to B$ implies w $\in L(G)$.

   In all the cases, we conclude that G generates w.

   With claim 1 and claim 2 together, it implies that L = L(G).