Proof that $L_{a's=b's} = \{ w \mid n_a(w) = n_b(w) \}$ is a Context Free Language
By William Gasarch

1 Introduction

Def 1.1 If $F$ is a finite set then a string over $F$ is a sequence of elements of $F$. For example, if $F = \{a, b, A, B\}$ then $aaAba$ is a string, $BaBBaAba$ is a string.

Def 1.2 The string $e$ is the empty string. Its main property is that for any string of symbols $\alpha$, $\alpha e = \alpha$.

Def 1.3 A language is a set of strings.

Def 1.4 If $L$ is a language then $LL = L^2 = \{ xy : x \in L \land y \in L \}$. You can define $L^3$, $L^4$, etc. Note that $L^1 = L$ and $L^0 = \{e\}$.

Def 1.5 If $L$ is a language then $L^* = L^0 \cup L^1 \cup \cdots$. Note that $L^*$ is the set of all strings of symbols from $L$.

Def 1.6 If $\alpha$ is a string then $\#_a(\alpha)$ is the number of $a$'s in $\alpha$. Note that $\alpha$ may have $b$'s and even nonterminals (like $S$) in it. $\#_b(\alpha)$ is defined similarly.

Def 1.7

1. A Context Free Grammar (henceforth CFG) is a tuple $(N, \Sigma, R, S)$ where
   (a) $N$ is a finite set of nonterminals. We will denote these by capitol letters.
   (b) $N$ is a finite set of terminals, also called the alphabet. We will denote these by small letters.
   (c) $R$ is a set of rules of the form $A \rightarrow \alpha$ where $A$ is a nonterminal and $\alpha$ is a string of terminals and nonterminals.
   (d) $S$ is the start nonterminal.

2. Let $G$ be a CFG. We write $S \Rightarrow \alpha$ to mean that if you start with $S$ you apply the rules (perhaps many times) and you end up with $\alpha$. If you use $n$ rules then we write this as $S \Rightarrow_n \alpha$. We all this a derivation of length $n$.

3. $L(G)$ is the set of nonterminals that can be generated from $S$. Formally
   \[ L(G) = \{ \alpha : S \Rightarrow \alpha \} \cap \Sigma^* \]
A CFG for \( L_{a's=b's} = \{ w \mid n_a(w) = n_b(w) \} \)

Let \( G \) be the CFG:

\[
\begin{align*}
S & \rightarrow aSa \\
S & \rightarrow bSa \\
S & \rightarrow SS \\
S & \rightarrow e
\end{align*}
\]

**Theorem 2.1** \( L_{a's=b's} = L(G) \).

**Proof:**

1) \( L(G) \subseteq L_{a's=b's} \).

**KEY:** Look at the set \( L'(G) = \{ \alpha : S \Rightarrow \alpha \} \). Note that we DID NOT intersect with \( \Sigma^* \). This is ALL of the sequences that can be generated, including those that have \( S \) in them.

**Claim:** For all \( n \), if \( S \Rightarrow_n \alpha \) then \( \#_a(\alpha) = \#_b(\alpha) \).

We prove this by induction on \( n \).

**Base Case:** \( n = 1 \). \( S \Rightarrow_1 \alpha \) means just \( S \rightarrow \alpha \). The only \( \alpha \) are \( aSa \) and \( bSa \) and \( SS \) and \( e \).

All of these strings have an equal number of \( a \)'s and \( b \)'s.

**IH:** If \( S \Rightarrow_{n-1} \alpha \) then \( \#_a(\alpha) = \#_b(\alpha) \).

**IS:** We show that if \( S \Rightarrow_n \alpha \) then \( \#_a(\alpha) = \#_b(\alpha) \).

We decompose \( S \Rightarrow_n \alpha \) into its first \( n-1 \) steps and its \( n \)th step. Since there is an \( n \)th step the \((n-1)\)st step must result in a string that has an \( S \) in it which is then used in a rule to get the \( n \)th step. So we have

\[
S \Rightarrow_{n-1} \beta S \gamma
\]

and then we have the next step. By the IH \( \#_a(\beta S \gamma) = \#_b(\beta S \gamma) \). The \( n \)th step will be to replace \( S \) with either \( aSB \) or \( bSa \) or \( SS \) or \( e \). Clearly the resulting string will have the same number of \( a \)'s as \( b \)'s.

**End of Proof of Claim**

Since

\[
L'(G) = \{ \alpha \in \{ S,a,b \}^* : \#_a(w) = \#_b(w) \}
\]

Clearly

\[
L(G) = \{ \alpha \in \{ a,b \} : \#_a(w) = \#_b(w) \}.
\]

Hence \( L(G) \subseteq L_{a's=b's} \).

2) \( L_{a's=b's} \subseteq L(G) \).

We proof this by induction on \( |w| \).

**Base Case:** If \( |w| = 0 \) then use \( S \rightarrow e \).

**IH:** All \( w' \) such that \( |w'| = n-1 \) and \( w' \in L_{a's=b's} \) are in \( L(G) \).

**IS:** Let \( w \) such that \( |w| = n \) and \( w \in L_{a's=b's} \). We show that \( w \in L(G) \).
Case 1: $w = aw'b$. Clearly $w' \in L(G)$. By the IH $S \Rightarrow w'$. To obtain $w$ we do the following:

$$S \rightarrow aSb \Rightarrow aw'b = w.$$  

Case 2: $w = bw'a$. Similar to Case 1.

Case 3: $w = axa$.

**Claim:** $w = w''w'''$ where $|w'|, |w''| < n$, $w', w'' \in L_{a' = b' s}$.

Look at the strings

- $w_0 = a$
- $w_1 = a\sigma_1$
- $\vdots$
- $w_i = a\sigma_1\sigma_2 \cdots \sigma_i$
- $\vdots$
- $w_{n-2} = a\sigma_1 \cdots \sigma_{n-2}$
- $w_{n-1} = a\sigma_1 \cdots \sigma_{n-2}a$

Note that

$$\#_a(w_0) - \#_b(w_0) = 1 > 0$$
$$\#_a(w_{n-1}) - \#_b(w_{n-1}) = 0$$

Since $w_{n-1} = w_{n-2}a$, we must also have

$$\#_a(w_{n-2}) - \#_b(w_{n-2}) < 0$$

We rewrite just two of the equations:

- $\#_a(w_0) - \#_b(w_0) > 0$
- $\#_a(w_{n-2}) - \#_b(w_{n-2}) < 0$

Since each $w_i$ is obtained by adding just one letter there must be an $i$ such that

$$\#_a(w_i) - \#_b(w_i) = 0$$

This $w_i \in L_{a' = b' s}$. Since $w \in L_{a' = b' s}$, we must also have that $w = w_iw'''$ and $w'' \in L_{a' = b' s}$.

Let $w_i = w'$.

**End of Proof of Claim**

So we now have $w = w'w''$ where $w' \in L_{a' = b' s}$ and $w'' \in L_{a' = b' s}$. By the IH $S \Rightarrow w'$ and $S \Rightarrow w''$. To derive $w$ use

$$S \rightarrow SS \Rightarrow w'S \Rightarrow w''w' = w$$

Case 4: $w = bxb$. Similar to Case 3.