Consider the language *L* that generates strings with twice as many a's as b's over the input alphabet $\Sigma = \{a, b\}$. The language does not care about the order in which the symbols a's and b's occur.

The CFG for the language *L* is as follows:

$$T \to Saab \mid aSab \mid aaSb \mid aabS \mid Saba \mid aSba \mid abSa \mid abaS \mid Sbaa \mid bSaa \mid baSa \mid baaS$$
$$S \to T \mid \varepsilon$$

Now, prove the grammar is correct using the induction.

The smallest possible strings that are generated by the grammar are $\{aab, aba, baa\}$. Let w be the string from the set of smallest possible strings, such that $f_a(w) = 2f_b(w)$, where $f_a(w)$ is the number of a's in the string w and $f_b(w)$ is the number of b's in the string w. Hence, all the smallest possible strings have twice as many a's as b's.

$$f_a(w_n) = 2f_b(w_n) \quad \text{......(1)}$$

(where w_n represents the string of length n)

Now, show that $f_a(w_{n+1}) = 2f_b(w_{n+1})$ holds.

Obtain the string w_{n+1} by inserting any of the strings $\{\varepsilon, aab, aba, baa\}$ in to w_n. The insertions may result in addition of 0 *a*'s and 0 *b*'s or 2 *a*'s and 1 *b*.

Case 1:

When ε is inserted, (inserting 0 a's and 0 b's)

$$f_a(w_{n+1}) = f_a(w_n) + f_a(\varepsilon) = f_a(w_n) + 0 = f_a(w_n) \quad \text{......(2)}$$
$$f_b(w_{n+1}) = f_b(w_n) + f_b(\varepsilon) = f_b(w_n) + 0 = f_b(w_n) \quad \text{......(3)}$$

Now, substitute (2) and (3) in (1).

$$f_a(w_n) = 2f_b(w_n)$$
$$f_a(w_{n+1}) = 2f_b(w_{n+1})$$

Case 2:

When aab or aba or baa is inserted, (inserting 2 a's and 1 b)

$$f_a(w_{n+1}) = f_a(w_n) + f_a(w) = f_a(w_n) + 2 \quad \text{......(4)}$$
$$f_b(w_{n+1}) = f_b(w_n) + f_b(w) = f_b(w_n) + 1 \quad \text{......(5)}$$
$$(\text{where } w \text{ is } aab \text{ or } aba \text{ or } baa)$$

Using (4),

$$\begin{aligned} f_a(w_{n+1}) &= f_a(w_n) + 2 && (\text{from } (4)) \\ &= 2f_b(w_n) + 2 && (\text{from } (1)) \\ &= 2(f_b(w_n) + 1) \\ f_a(w_{n+1}) &= 2f_b(w_{n+1}) && (\text{from } (5)) \end{aligned}$$

From both the cases, it is proved that $f_a(w_{n+1}) = 2f_b(w_{n+1})$.

Hence, from the principle of mathematical induction the grammar is correct.

Therefore, the CFG generates the language of strings with twice as many *a*'s as *b*'s.