Example Induction proof on strings derived from Context Free Grammars

CS 173 Lecture B Fall 2016

Gul Agha

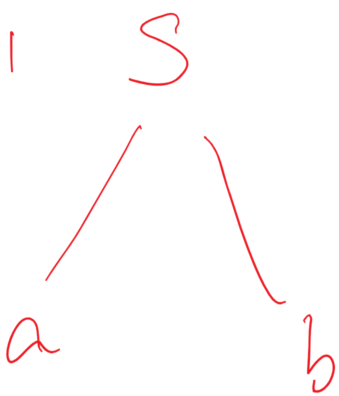
(1) $S \rightarrow a\ b$ (2) $S \rightarrow a\ c$
(3) $S \rightarrow a\ S\ b$ (4) $S \rightarrow a\ S\ S\ c$

Claim: For any string generated by the grammar G above, the number of a's will be equal to the number of b's plus the number of c's ($n_a = n_b + n_c$)

Proof by Induction on number of applications of the rules.

Base Case. Suppose the string has been generated with just application of a rule. Then it must have been generated by (2) or by (3). Thus it is either $a\ b$ or $a\ c$. In either case, $n_a = n_b + n_c$.

$n_a = 1$
$n_b = 1$
$n_c = 0$



$n_a = 1$
$n_b = 0$
$n_c = 1$

(1) $S \rightarrow a\ b$ (2) $S \rightarrow a\ c$
(3) $S \rightarrow a\ S\ b$ (4) $S \rightarrow a\ S\ S\ c$

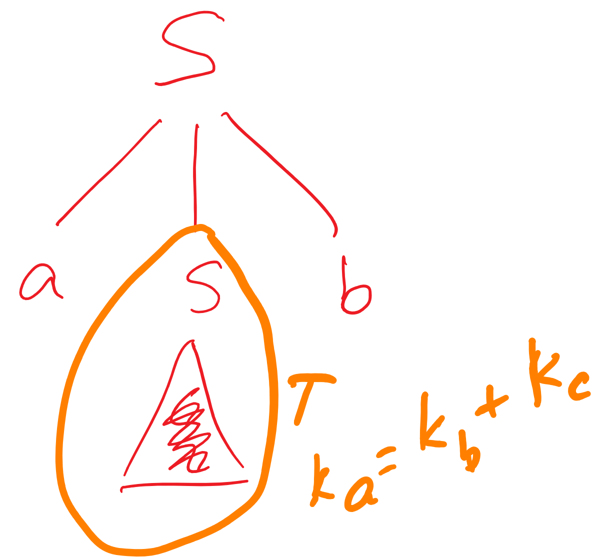
Claim: For any string generated by the grammar G above, the number of a's will be equal to the number of b's plus the number of c's ($n_a = n_b + n_c$)

Induction Hypothesis: For all $k \geq 1,\ r < k$ applications of the rules, the number of a's (call it k_a) is the sum of the number of b's (k_b) and the number of c's (k_c). $k_a = k_b + k_c$.

Want to prove: For $r = k + 1$ applications of the rules, $n_a = n_b + n_c$ where n_a, n_b, n_c are the number of a's, b's and c's in the string generated.

Proof: Consider the first application of the rule. If it is (1) or (2), we can't have another application so $r = 1$ and we are interested in $r = k + 1 > 1$. So the first rule applied must be (3) or (4).

Let $r = k + 1$be the number of applications of the rules.



Case 1. The first rule applied is (3). Now the tree T (see diagram) has k applications of the rule.
Thus by induction hypothesis, $k_a = k_b + k_c$.
But $n_a = k_a + 1$ and $n_b + n_c = k_b + k_c + 1$. Thus $n_a = n_b + n_c$.

We proceed to **Case 2** where the first rule applied is (4).

(1) $S \rightarrow a\ b$ (2) $S \rightarrow a\ c$
(3) $S \rightarrow a\ S\ b$ (4) $S \rightarrow a\ S\ S\ c$

Induction Hypothesis: For all $k \geq 1,\ r < k$ applications of the rules, assume the number of a's (call it k_a) is the sum of the number of b's (k_b) and the number of c's (k_c). $k_a = k_b + k_c$.

Want to prove: For $r = k + 1$ applications of the rules, $n_a = n_b + n_c$ where n_a, n_b, n_c are the number of a's, b's and c's in the string generated.

Recall that $r = k + 1$ is the number of applications of the rules

Case 2. The first rule applied is (4). Let the trees T_1 and T_2 (see diagram) have $r_1 > 0$ and $r_2 > 0$ applications of the rules respectively.
Since $r = r_1 + r_2 + 1, r_1 < k$ and $r_2 < k$.
By induction hypothesis, the number of a's is the sum of the number of b's and the number of c's in each of trees T_1 and T_2.
But n_a, n_c is one more than the sum of a's and c's in T_1 and T_2, respectively and the number of b's is the same as the sum of b's in T_1 and T_2.
Thus $n_a = n_b + n_c$.
QED.