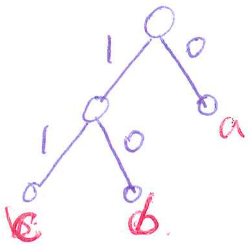
4. (a) $\{\underset{a}{0}, \underset{b}{10}, \underset{c}{11}\}$



① $f_a = \frac{1}{2}$, $f_b = \frac{1}{4}$, $f_c = \frac{1}{4}$

② $f_a = \frac{5}{8}$, $f_b = \frac{1}{8}$, $f_c = \frac{1}{4}$

⋮

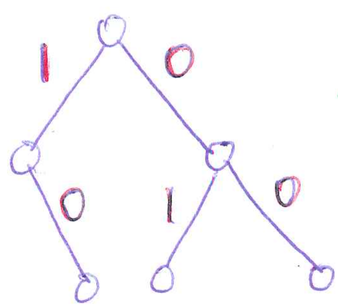
Just make sure. $f_a + f_b + f_c = 1$

f_b and. f_c least frequecy.

(b). $\{0, 1, 00\}$.

$\underline{0}$ is prefix of $\underline{0}0$, this is not prefix-free code.

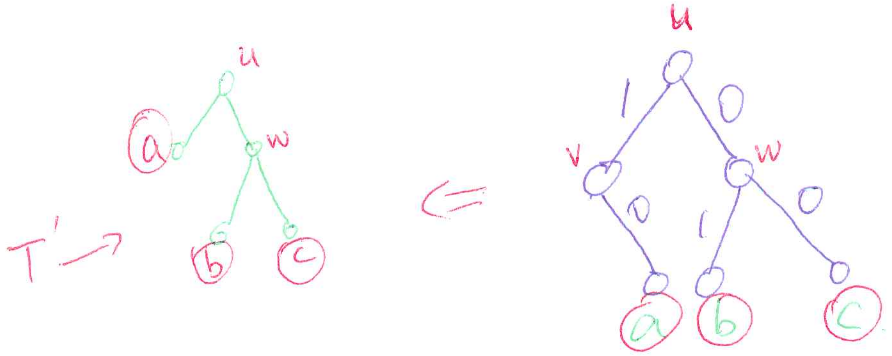
(c). $\{10, 01, 00\}$.



← This is not a full binary tree!

Lecture slides : Huffman_code, check. page 10!

a full tree may not necessarily represent optimal code, but optimal code must be full binary tree, why?



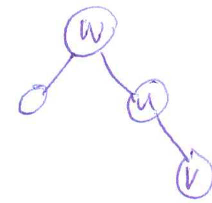
$cost(T) = f_a \times 2 + f_b \times 2 + f_c \times 2$

Suppose T is binary tree of optimal prefix code and is not Full.

Delete "v", let "a" be the child of "u". $cost(T') = \sum_{i=1}^{n} f_i \times depth_T(c_i)$ decrease!

$cost(T') = f_a \times 1 + f_b \times 2 + f_c \times 2 < cost(T)$

Clearly, the new tree T' has smaller cost than T, contradiction!



General proof: case $\boxed{1}$: u is the root.

delete u and use v as the root.

case $\boxed{2}$: u is not the root.

- assume w be the parent of u.
- delete u and make v be a child of w in place of u.

In both case the cost will decrease! the new tree T' has smaller cost encoding than T, contradict to our assumption that T is the binary tree of optimal prefix code.