5.二叉树

Huffman编码树

PFC编码

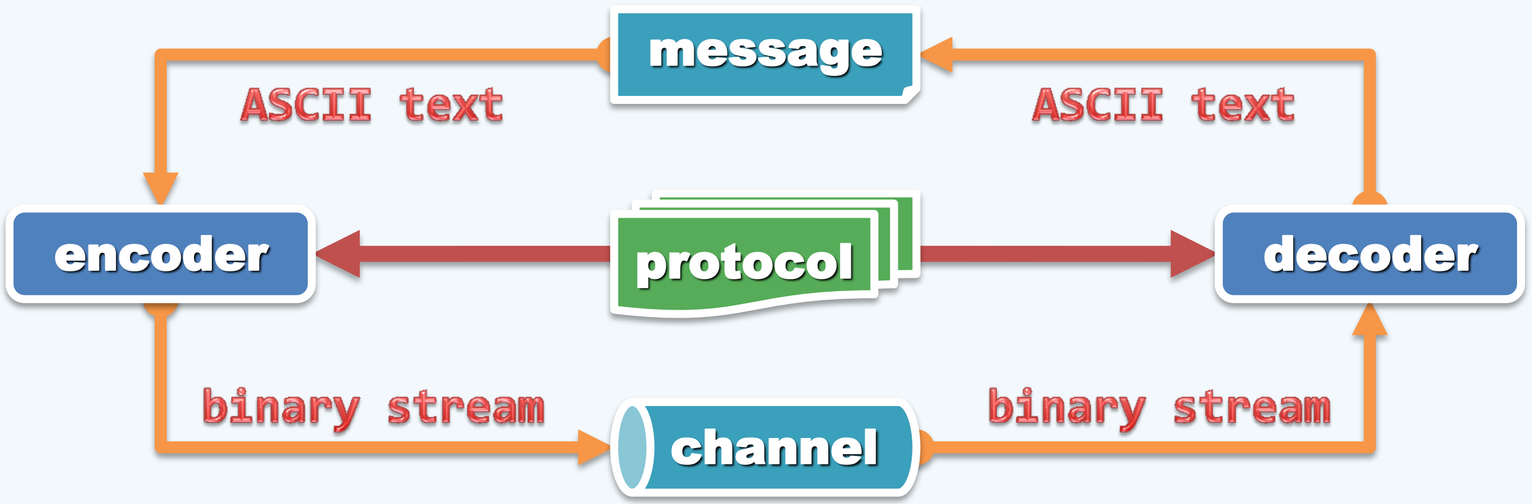
句读之不知，惑之不解，或师焉，或不焉，
小学而大遗，吾未见其明也

邓俊辉

deng@tsinghua.edu.cn

应用

❖ 通讯 / 编码 / 译码



❖ 二进制编码

- 组成数据文件的字符来自字符集Σ
- 字符被赋予互异的二进制串

❖ 文件的大小取决于

- 字符的数量 × 各字符编码的长短

❖ 通讯带宽有限时

- 如何对各字符编码，使文件最小？

"$1^{010}011^{00}$" = "MAIN"

M	A	I	N
1	010	011	00

二叉编码树

❖ 将Σ中的字符组织成一棵二叉树

以 0 / 1 表示 左 / 右 孩子

各字符 x 分别存放于对应的叶子 v(x) 中

❖ 字符x的编码串rps(v(x)) = rps(x)

由根到v(x)的通路（root path）确定

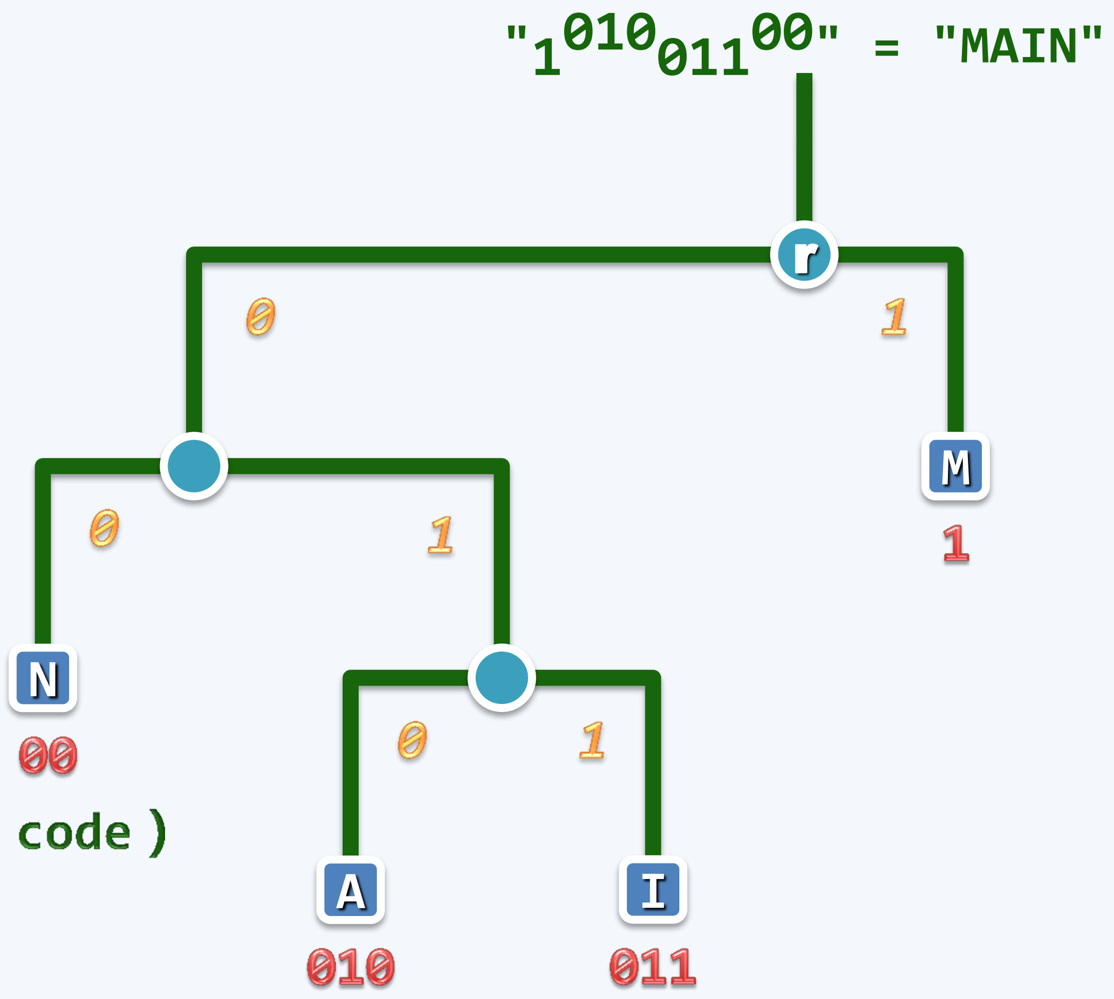
❖ 优点：字符编码不必等长，且

不致出现解码歧义

❖ 这属于“前缀无歧义”编码（prefix-free code）

不同字符的编码 互不为前缀 ，故不致歧义

❖ 缺点：你能发现吗？



编码长度 vs. 叶节点平均深度

- $|rps(x)| = depth(v(x))$
- 编码总长 $= \Sigma_x depth(v(x))$

  平均编码长度

  $= \Sigma_x depth(v(x))/|\Sigma|$

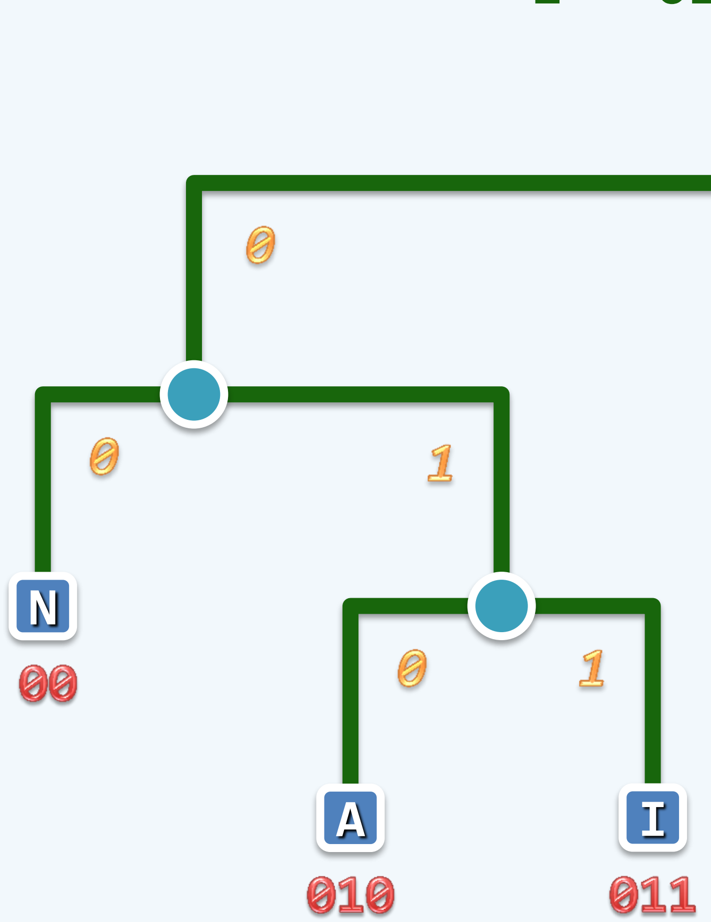
  $=$ 叶节点平均深度ald(T)
- 对于特定的Σ

  ald()最小者即为最优编码树T_{opt}
- 最优编码树必然存在，但不见得唯一

  它们具有哪些特征？

ald(T)*4 = 2+3+3+1 = 9

"$_{1}{}^{010}{}_{011}{}^{00}$" = "MAIN"

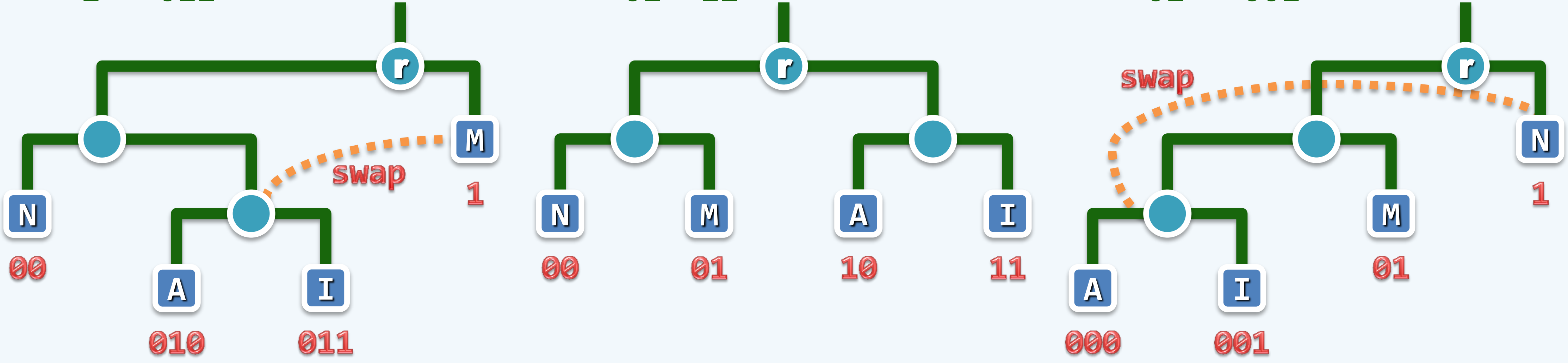


最优编码树

- ❖ $\forall\ v \in T_{opt}$, $deg(v) = 0$ $\boxed{\text{only if}}$ $depth(v) \geq depth(T_{opt})-1$
  亦即，叶子只能出现在**倒数两层**内——否则，通过节点**交换**即可...

$ald(T)*4 = 2+3+3+1 = 9$　　$ald(T)*4 = 2+2+2+2 = 8$　　$ald(T)*4 = 2+3+3+1 = 9$

"$1^{010}011^{00}$" = "MAIN"　　"$01^{10}11^{00}$" = "MAIN"　　"$01^{000}001^{1}$" = "MAIN"



- ❖ 特别地，**真**完全树即是最优编码树
- ❖ 实际上，字符的**出现频率**不尽相同，例如w('E') >> w('Z')

带权编码长度 vs. 叶节点平均带权深度

❖ 已知各字符的$\boxed{\text{期望频率}}$，如何构造最优编码树？

❖ 文件长度 ∝ 平均带权深度

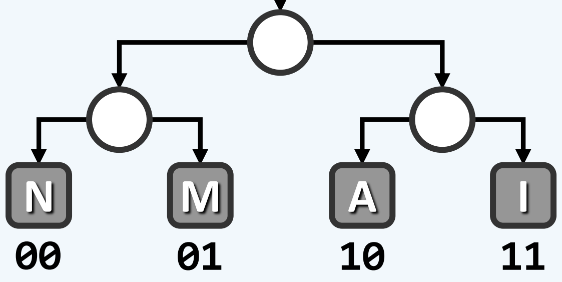
$= \mathrm{wald}(T) = \sum_x \mathrm{rps}(x) \times w(x)$

❖ 此时，完全树$\boxed{\text{未必}}$就是最优编码树

比如，考查"mamani"和"mammamia"...

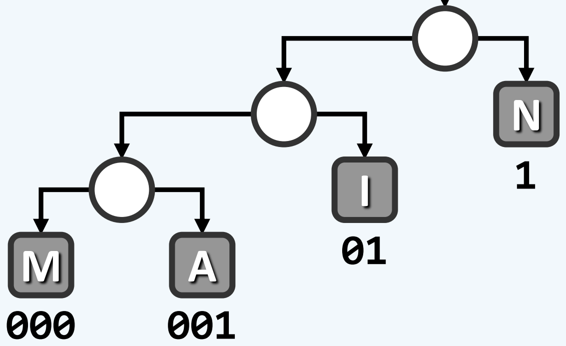
|"011001100011"| = 12

|"0110010110011110"| = 16



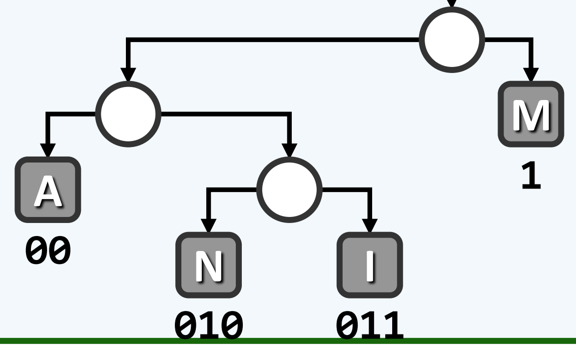
|"000001000001101"| = 15

|"00000100000000100001001"| = 23



|"1001000100011"| = 12

|"1001100101100"| = 13



最优带权编码树

❖ 同样，频率高/低的（超）字符，应尽可能放在高/低处

❖ 故此，通过交换，同样可以缩短wald(T)