Huffman Coding

Eun Bae Kong
Department of Computer Science and Engineering
Chungnam National University
Taejon, 305-764
Republic of Korea
keb@cnu.ac.kr

Fixed-Length Encoding Schemes

- ASCII
- Waste in bits
- Use a small number of bits for the frequent letters, and a larger number of bits for the less frequent ones.

Variable-Length Encoding Schemes

- Morse code
  - e: 0
  - t: 1
  - a: 01

    $\vdots$
- 0101
  - eta
  - aa
  - $etet$
  - aet

Prefix Codes

- *Prefix code* for a set S of letters is a function γ that maps each letter $x \in S$ to some sequence of zeros and ones, in such a way that for distinct $x, y \in S$, the sequence $\gamma(x)$ is not a prefix of the sequence $\gamma(y)$.
- For $S = \{a, b, c, d, e\}$,

$$\begin{aligned} \gamma_1(a) &= 11 \\ \gamma_1(b) &= 01 \\ \gamma_1(c) &= 001 \\ \gamma_1(d) &= 10 \\ \gamma_1(e) &= 000 \end{aligned}$$

- $cecab \Longrightarrow 0010000011101$

Decoding Prefix Code

- Scan the bit sequence from left to right.
- Match sufficiently enough bits to some letters.
- Delete the corresponding bits from the front and iterate.

$$cecab \Longrightarrow \underbrace{001}_{c}\underbrace{000}_{e}\underbrace{001}_{c}\underbrace{11}_{a}\underbrace{01}_{b}$$

Optimal Prefix Codes

- n: total number of letters
  f_x: fraction of letter x
- encoding length $= \Sigma_{x \in S}\, n f_x |\gamma(x)| = n\, \Sigma_{x \in S}\, f_x |\gamma(x)|$.
- Average number of bits per letter (ABL(γ)) $= \Sigma_{x \in S}\, f_x |\gamma(x)|$
- Want to minimize ABL.

Prefix Codes as Binary Trees

- The number of leaves is equal to the size of the alphabet S.
- Each leaf is labeled with a distinct letter of S.
- Left path is labeled with 0, and right path is labeled with 1.

Theorem 4.1 *The encoding of S constructed from T is a prefix code.*

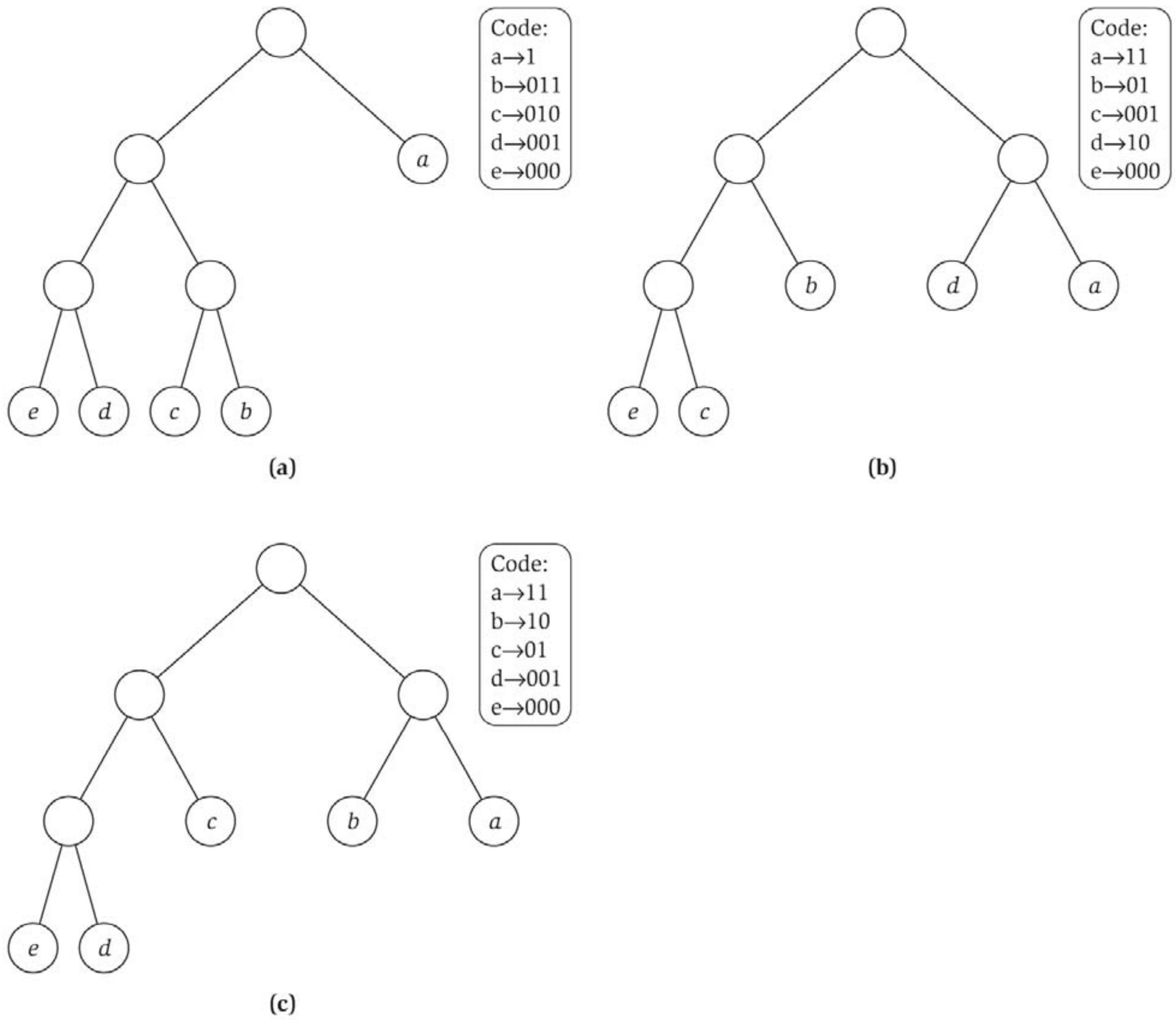

Figure 4.16 Parts (a), (b), and (c) of the figure depict three different prefix codes for the alphabet $S = \{a, b, c, d, e\}$.

Copyright © 2005 Pearson Addison-Wesley. All rights reserved.

Optimal Trees

- Search for an optimal prefix code is the search for a binary tree that minimizes the average number of bits per letter.
- The length of the encoding of x is equal to the depth of x in the tree.
- We are seeking the labeled tree that minimizes the weighted average of the depths of all leaves (ABL(T)).

Full Tree

A binary tree is *full* if each node that is not a leaf has two children.

Theorem 4.2 *The binary tree corresponding to the optimal prefix code is full.*

Proof. Let T denote the binary tree corresponding to the optimal prefix code, and suppose it contains a node u with exactly one child v.

- Replace node u with v.
- This change decreases the number of bits needed to encode any leaf in the subtree rooted at node u.
- T cannot be optimal.

Exchange Argument

Theorem 4.3 *Suppose that u and v are leaves of T^*, such that depth(u) < depth(v). Suppose that leaf u is labeled with y and leaf v is labeled with z. Then $f_y \geq f_z$.*

Proof. If $f_y < f_z$, exchange the labels at the nodes u and v.

- The change to the overall sum is $(depth(v) - depth(u))(f_y - f_z)$.
- If $f_y < f_z$, this change is negative, contradiction.

Don't place a lower-frequency letter at a strictly smaller depth than some other higher-frequency letter.

Siblings

Theorem 4.4 *Let v be a leaf of maximum depth and u be the parent of v. Since T^* is a full binary tree, u has another child w Then, w is a leaf of T^*.*

Proof.

- If w were not a leaf, there would be some leaf w' in the subtree below it.
- Then w' would have a depth greater than that of v.
- Contradictingour assumption that v is a leaf of maximum depth.

Optimal Prefix Code

Theorem 4.5 *There is an optimal prefix code in which the two lowest-frequency letters are assigned to leaves that are siblings in* T^*.

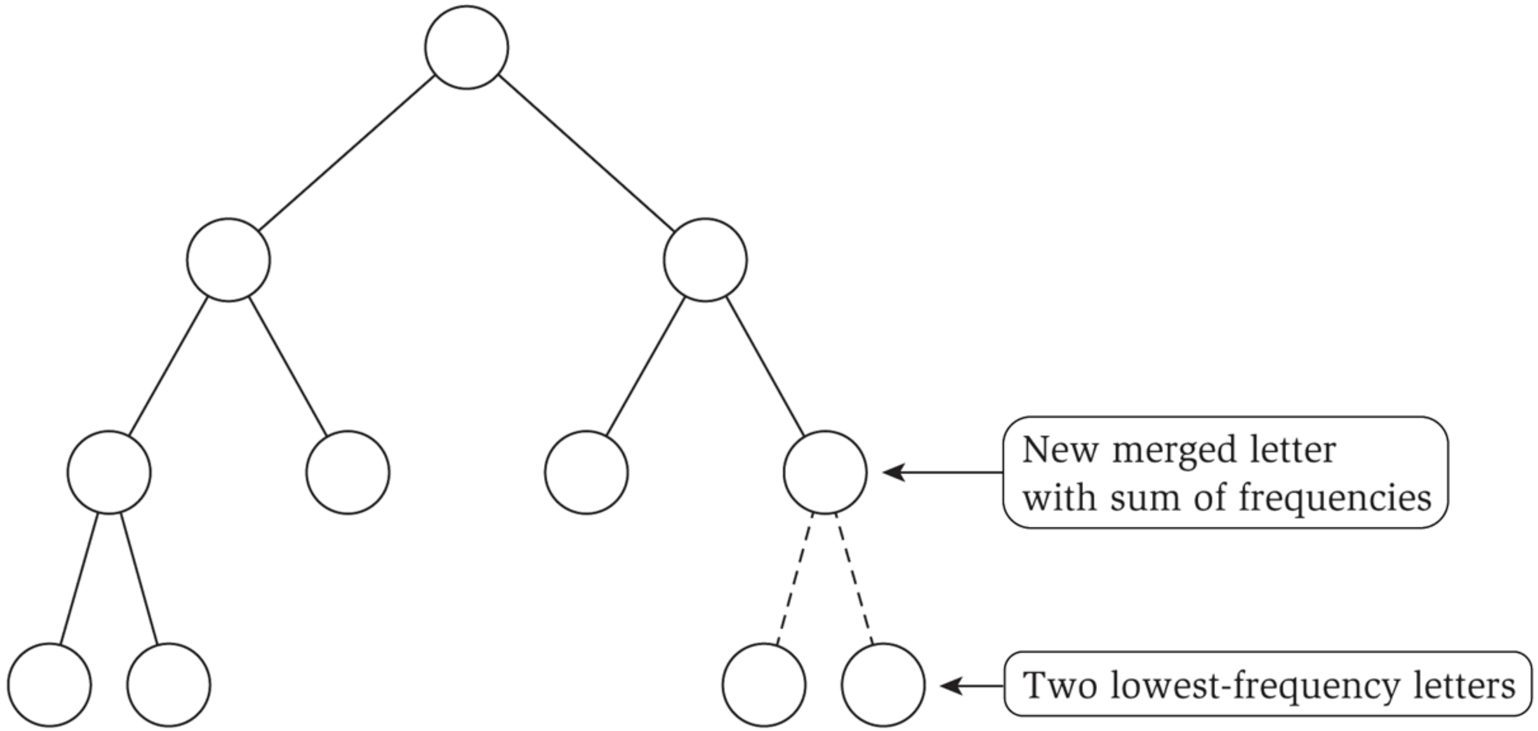

Figure 4.17 There is an optimal solution in which the two lowest-frequency letters label sibling leaves; deleting them and labeling their parent with a new letter having the combined frequency yields an instance with a smaller alphabet.

Copyright © 2005 Pearson Addison-Wesley. All rights reserved.

Algorithm 4.6, page 172

```
To construct a prefix code for an alphabet S, with given frequencies:
  If S has two letters then
    Encode one letter using 0 and the other letter using 1
  Else
    Let y* and z* be the two lowest-frequency letters
    Form a new alphabet S' by deleting y* and z* and
      replacing them with a new letter ω of frequency f_{y*} + f_{z*}
    Recursively construct a prefix code γ' for S', with tree T'
    Define a prefix code for S as follows:
      Start with T'
      Take the leaf labeled ω and add two children below it
        labeled y* and z*
   Endif
```

Copyright © 2005 Pearson Addison-Wesley. All rights reserved.

Optimality

Let T be a tree for S and T' be a tree that merges the two lowest-frequency letters $y^*, z^* \in S$ into a single letter ω.

Theorem 4.6 $ABL(T') = ABL(T) - f_\omega$.

Proof.

- The depth of each letter $x \neq y^*, z^*$ is the same in both T and T'.
- The depths of y^* and z^* in T are each one greter than that of ω in T'.
- $f_\omega = f_{y^*} + f_{z^*}$.

$$\begin{aligned}
\mathrm{ABL}(T) &= \sum_{x\in S} f_x \cdot depth_T(x) \\
&= f_{y^*} \cdot depth_T(y^*) + f_{z^*} \cdot depth_T(z^*) + \sum_{x\neq y^*,z^*} f_x \cdot depth_T(x) \\
&= (f_{y^*} + f_{z^*}) \cdot (1 + depth_{T'}(\omega)) + \sum_{x\neq y^*,z^*} f_x \cdot depth_{T'}(x) \\
&= f_\omega \cdot (1 + depth_{T'}(\omega)) + \sum_{x\neq y^*,z^*} f_x \cdot depth_{T'}(x) \\
&= f_\omega + f_\omega \cdot depth_{T'}(\omega) + \sum_{x\neq y^*,z^*} f_x \cdot depth_{T'}(x) \\
&= f_\omega + \sum_{x\in S'} f_x \cdot depth_{T'}(x) \\
&= f_\omega + \mathrm{ABL}(T').
\end{aligned}$$

Optimality

Theorem 4.7 *The Huffman code for a given alphabet achieves the minimum average number of bits per letter of any prefix code.*

Proof. Suppose that T produced by greedy algorithm is not optimal.

- This means that there is some labeled tree Z such that $\mathrm{ABL}(Z) < \mathrm{ABL}(T)$.
- In Z, two lowest-frequency letters y^* and z^* are siblings.
- Delete the leaves labeled y^* and z^* from Z, and label their former parent with ω.
- We get a tree Z' that is a prefix code for S'.
- $\mathrm{ABL}(Z') = \mathrm{ABL}(Z) - f_\omega$.
- We have assumed that $\mathrm{ABL}(Z) < \mathrm{ABL}(T)$; subtracting f_ω from both sides we get $\mathrm{ABL}(Z') < \mathrm{ABL}(T')$.
- That contradicts the optimality of T' as a prefix code for S'.