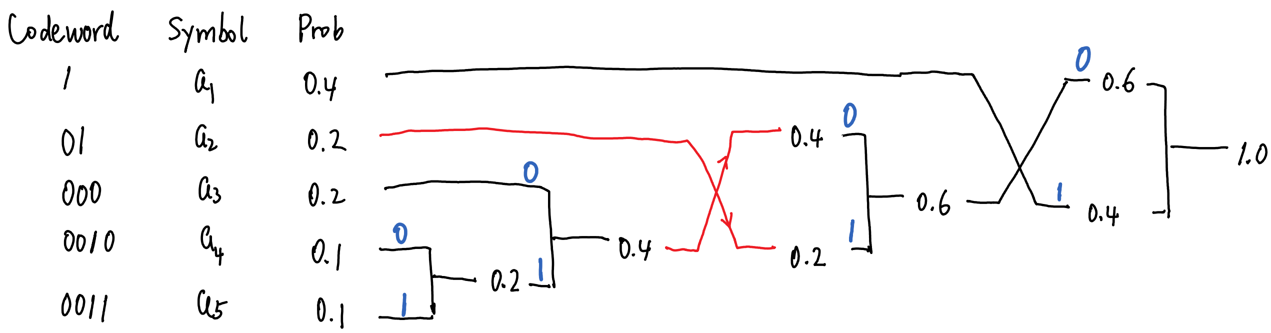
# Lecture 12

Huffman Code (cont'd)

Consider another example:
Alphabet = $\{a_1, a_2, a_3, a_4, a_5\}$




Avg. Codeword Length (ACL)

= 1 x 0.4 + 2 x 0.2 + 3 x 0.2 + 4 x 0.1 + 4 x 0.1 = 2.2  bits/symbol

Entropy of the source entropy:
H(0.4, 0.2, 0.2, 0.1, 0.1) = 2.1219 bits/symbol

```
>> -0.4*log2(0.4)-2*0.2*log2(0.2)-2*0.1*log2(0.1)
ans =
   2.1219
```

Redundancy = ACL - H( ) =  2.2 - 2.1219 = 0.0781 bit/symbol

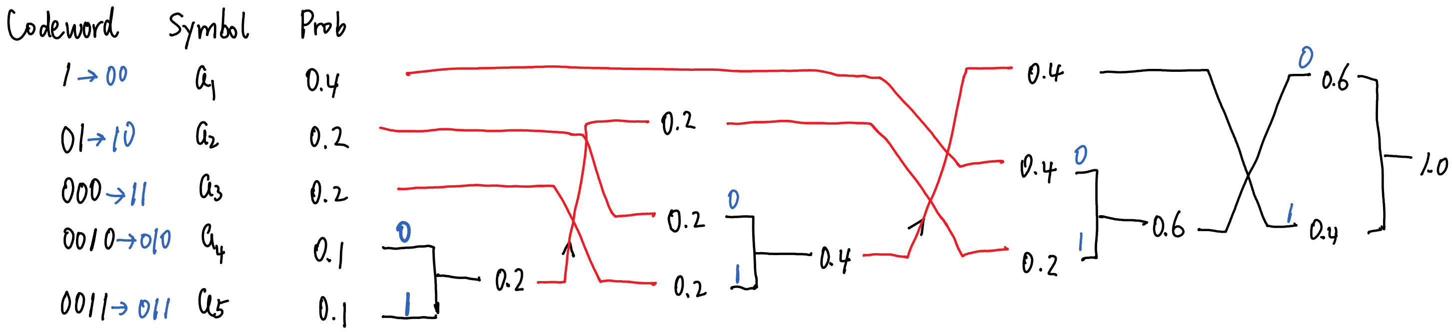
$Var[X] = E[(X - E[X])^2]$

Variance of the codeword lengths
= (1 - 2.2)^2 x 0.4 + (2 - 2.2)^2 x 0.2 + (3 - 2.2)^2 x 0.2 + (4 - 2.2)^2 x 0.1 x 2 = 1.36

Next, minimum-variance Huffman codes:
Break the probability tie by moving the **composite symbol upward**.

**variance — Variance for Huffman code**
'min' | 'max'
Variance for Huffman code, specified as one of these values.

- 'min' — This function generates N-ary Huffman code dictionary with the minimum variance. If you do not specify the variance input argument, the function uses this case (default).
- 'max' — This function generates N-ary Huffman code dictionary with the maximum variance.



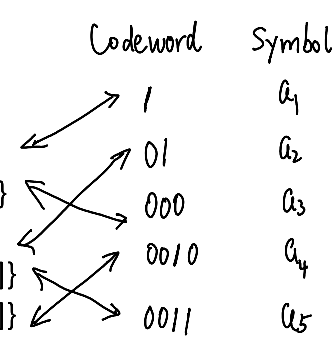

Avg. Codeword Length (ACL) = 2 x 0.4 + 2 x 0.2 + 2 x 0.2 + 3 x 0.1 + 3 x 0.1 = 2.2 bits/symbol -- no change

Variance of the codeword lengths
= (2 - 2.2)^2 x 0.4 + (2 - 2.2)^2 x 0.2 + (2 - 2.2)^2 x 0.2 + (3 - 2.2)^2 x 0.1 x 2 = 0.16 < 1.36 previously

Matlab,
Huffmandict ( )
Generate Huffman code dictionary

```
>> symbols = (1:5)
symbols =
   1    2    3    4    5
>> p = [.4 .2 .2 .1 .1]
p =
   0.4000    0.2000    0.2000    0.1000    0.1000
>> sum(p)
ans =
   1
>> [dict,avglen] = huffmandict(symbols, p);
>> avglen
avglen =
   2.2000
```

```
>> dict
dict =
 5×2 cell array
   {[1]}    {[      1]}
   {[2]}    {[  0 0 0]}
   {[3]}    {[    0 1]}
   {[4]}    {[0 0 1 1]}
   {[5]}    {[0 0 1 0]}
```



- Color image (R,G,B components) -- HW3 problem

```
>> I = imread('football.jpg');
>> imshow(I)
>> whos I
  Name       Size              Bytes  Class    Attributes

  I          256x320x3         245760  uint8

>> IR = I(:,:,1);
>> figure; imshow(IR)
```

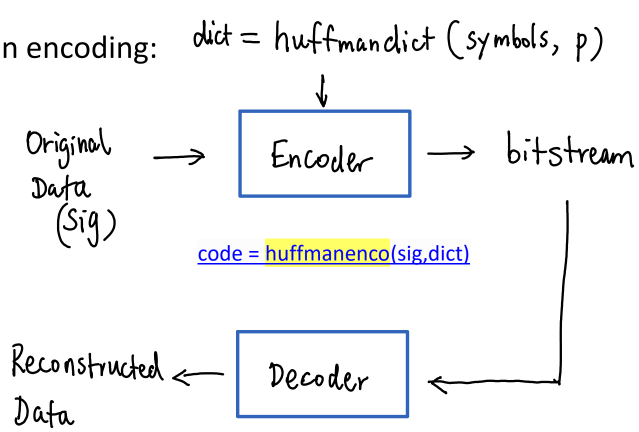
Go back to minimum-variance Huffman code:

```
>> [dict,avglen] = huffmandict(symbols, p, 2, 'min');
>> avglen
avglen =
    2.2000
>> dict
dict =
  5×2 cell array
    {[1]}    {[  0 0]}
    {[2]}    {[  1 1]}
    {[3]}    {[  1 0]}
    {[4]}    {[0 1 1]}
    {[5]}    {[0 1 0]}
```

2 ← binary

| Codeword | Symbol |
|---|---|
| 1 → 00 | $a_1$ |
| 01 → 10 | $a_2$ |
| 000 → 11 | $a_3$ |
| 0010 → 010 | $a_4$ |
| 0011 → 011 | $a_5$ |

Huffman encoding:




code = huffmanenco(sig,dict)

```
>> symbols = 1:6;
p = [.5 .125 .125 .125 .0625 .0625];
>>[dict,avglen] = huffmandict(symbols,p);
>> inputSig = randsrc(100,1,[symbols;p]);

>> code = huffmanenco(inputSig,dict);
>> 224/100
ans =
    2.2400
```

```
>> avglen
avglen =
    2.1250
```