Arithmetic Encoder/Decoder Example

Assume:

Fixed probabilities : a=3/12, b=4/12, c=5/12
Binary output
8-bit code values
Message sequence = **bac**

Initialization:

low = 0	a:	Cum-Freq[3] = 0
First-qtr = 64	b:	Cum-Freq[2] = 3
Half = 128	c:	Cum-Freq[1] = 7
Third-qtr = 192		Cum-Freq[0] = 12
high = 255		

Encode b:	range	= 256
	high = 0 + (256 * 7) / 12 - 1	= 148
	low = 0 + (256 * 3) / 12	= 64
Expand Range:	low = 2 * (64 - 64)	= 0
	high = 2 * (148 - 64) + 1	= 169
Encode a:	range	= 170
	high = 0 + (170 * 3) / 12 - 1	= 41
	low = 0 + (170 * 0) / 12	= 0
Output 01 and scale:	low = 2 * 0	= 0
	high = 2 * 41 + 1	= 83
Output 0 and scale:	low = 2 * 0	= 0
	high = 2 * 83 + 1	= 167
Encode c:	range	= 168
	high = 0 + (168 * 12) / 12 - 1	= 167
	low = 0 + (168 * 7) / 12	= 98
Expand Range:	low = 2 * (98 - 64)	= 68
	high = 2 * (167 - 64) + 1	= 207

Termination is 10, except we must include the deferred 0 after the 1. Thus the code is 010100. Notice that the probability of the message bac is .03472, and the optimal encoding would require $-\log_2 .03472 = 4.85$ bits. To decode, we begin with code 01010000 = 80 and note that the rightmost 2 bits could be anything.

Decode b:	value	= 80
	range	= 256
	cum = ((80 - 0 + 1) * 12 - 1 / 256	= 3
	symbol	= 2 or b
Encode b:	range	= 256
	high = 0 + (256 * 7) / 12 - 1	= 148
	low = 0 + (256 * 3) / 12	= 64
Expand Range:	value = 2 * (80 - 64) + 0	= 32
	low = 2 * (64 - 64)	= 0
	high = 2 * (148 - 64) + 1	= 169
Decode a:	value	= 32
	range	= 170
	cum = ((32 - 0 + 1) * 12 - 1 / 170	= 2
	symbol	= 1 or a
Encode a:	range	= 170
	high = 0 + (170 * 3) / 12 - 1	= 41
	low = 0 + (170 * 0) / 12	= 0
Expand Range:	value = 2 * 32 + 0	= 64
	low = 2 * 0	= 0
	high = 2 * 41 + 1	= 83
Expand Range:	value = 2 * 64 + 0	= 128
	low = 2 * 0	= 0
	high = 2 * 83 + 1	= 167
Decode c:	value	= 128
	range	= 168
	cum = ((128 - 0 + 1) * 12 - 1 / 168	= 9
	symbol	= 3 or c