In the next section we're going to start our discussion on how to actually engineer the bit encodings we'll use in our circuitry, but first we'll need a way to evaluate the efficacy of an encoding.

The entropy of a random variable is average amount of information received when learning the value of the random variable.

The mathematician's name for "average" is "expected value"; that's what the capital E means.

We compute the average in the obvious way: we take the weighted sum, where the amount of information received when learning of particular choice i -- that's the log-base-2 of 1/p_i -- is weighted by the probability of that choice actually happening.

Here's an example.

We have a random variable that can take on one of four values: A, B, C or D.

The probabilities of each choice are shown in the table, along with the associated information content.

Now we'll compute the entropy using the probabilities and information content.

So we have the probability of A (1/3) times its associated information content (1.58 bits), plus the probability of B times its associated information content, and so on.

The result is 1.626 bits.

This is telling us that a clever encoding scheme should be able to do better than simply encoding each symbol using 2 bits to represent which of the four possible values is next.

Food for thought!

We'll discuss this further in the third section of this chapter.

So, what is the entropy telling us?

Suppose we have a sequence of data describing a sequence of values of the random variable X.

If, on the average, we use less than H(X) bits to transmit each piece of information in the sequence, we will not be sending enough information to resolve the uncertainty about the values.

In other words, the entropy is a lower bound on the number of bits we need to transmit.

Getting less than this number of bits wouldn't be good if the goal was to unambiguously describe the sequence of

values -- we'd have failed at our job!

On the other hand, if we send, on the average, more than H(X) bits to describe the sequence of values, we will not be making the most effective use of our resources, since the same information might have been able to be represented with fewer bits.

This is okay, but perhaps with some insights we could do better.

Finally, if we send, on the average, exactly H(X) bits, then we'd have the perfect encoding.

Alas, perfection is, as always, a tough goal, so most of the time we'll have to settle for getting close.

In the final set of exercises for this section, try computing the entropy for various scenarios.