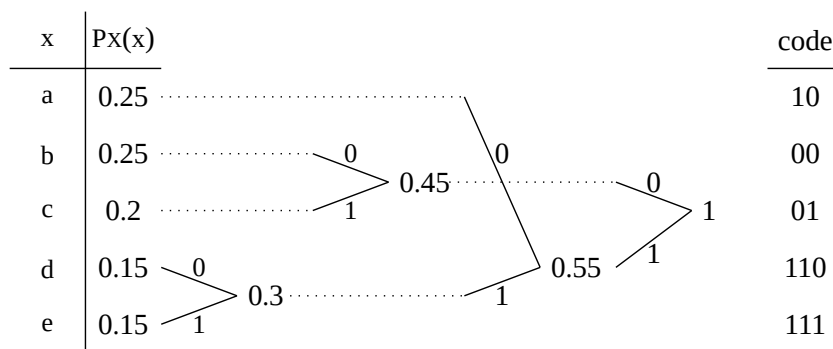


Definition: Huffman Codes

Shannon's source-coding theorem shows us that in theory, the minimal code length for a source P_X is roughly $H(X)$. In this section we will investigate **Huffman codes**, which provide an explicit and neat construction for optimal prefix-free codes. A binary Huffman code for a source P_X is constructed by iteratively pairing the two symbols with the smallest probability together, building a binary tree on the way. This is best explained by example:

Example: Binary Huffman code

Let the random variable X be given with $\mathcal{X} = \{a, b, c, d, e\}$ and $P_X(a) = P_X(b) = 0.25$, $P_X(c) = 0.2$, and $P_X(d) = P_X(e) = 0.15$. The following is a binary Huffman code for P_X :



We build up the tree from left to right, pairing the symbols (or groups of symbols) with smallest (combined) probabilities at every step. The codeword for every symbol is then determined by following the branches of the tree *from right to left* until the symbol is reached. Note that this way, the symbols with the smallest probabilities get assigned the longest codewords (paths). The average codeword length for this code is

$$0.25 \cdot 2 + 0.25 \cdot 2 + 0.2 \cdot 2 + 0.15 \cdot 3 + 0.15 \cdot 3 = 2.3.$$

This is very close to the entropy $H(X) \approx 2.285$. The average codeword length lies between $H(X)$ and $H(X) + 1$.