

21.1 More on Compression

In Huffman encoding, the dictionary is not fixed, but it is *static*: the dictionary is the same for the entire encoding/decoding. Now we're going to look at another encoding method: adaptive encoding

21.1.1 Lempel-Ziv

The main idea is that each character in the coded text C either refers to a single character in Σ_S or a substring of S that both the encoder and decoder have already seen.

Encoding

after encoding a substring x of S , add xc to D , where c is the character that follows x . We keep a dictionary table of ASCII keys and their respective values (the keys go up to 127). Starting at 128, we create our own dynamic dictionary that stores the values of the text we're reading. Refer to the LZW example in the course slides in module 10 for a live example.

Decoding

after decoding a substring y of S , add xc to D , where x is previously encoded/decoded substring of S , and c is the first character of y .

21.1.2 Efficient Compression

Move-to-Front

When retrieving encoding information from our dictionary, we can rearrange our dictionary using the move-to-front heuristic.