13.4. Universal Data Compression

13.4.1. The Lempel–Ziv Algorithm

Definition 13.7. Let $X := \{0, 1\}$ and consider the sequence $x \in X^n$. The procedure of parsing $x$ into the shortest substrings that have not appeared so far is called the Lempel–Ziv parsing.

Example 13.8. Suppose $x = 1011010100010$. We parse it using Lempel–Ziv parsing as

shortest phrases list = 1, 0, 11, 01, 010, 00, 10.

Note that each phrase in the list is unique.

The Lempel–Ziv Algorithm consists of two steps.

Parsing: Parse the sequence $x$ using the Lempel–Ziv parsing to obtain the “shortest phrases list.”

Coding: Represent each phrase found in $x$, say $x_i \cdots x_{j-1} x_j$, by the ordered pair $(p, x_j)$, where $p$ is the position of $x_i \cdots x_{j-1}$ in the “shortest phrases list.” Thus, to decode $(p, x_j)$ we concatenate the $p$th entry in the “shortest phrases list” with the bit $x_j$. Positions in the “shortest phrases list” are numbered starting at $p = 1$. One-bit phrases of the form $x_j$ are coded as $(0, x_j)$ and are decoded as $x_j$ without looking at the “shortest phrases list.”

Example 13.9. Let $x$ and its “shortest phrases list” be as in the previous example. Applying coding procedure to $x$, the corresponding code pairs are

$$(0, 1), (0, 0), (1, 1), (2, 1), (4, 0), (2, 0), (1, 0).$$

If $x \in X^n$, we denote the number of code pairs by $c_n(x)$. For the sequence $x$ of Example 13.8, $n = 13$ and $c_{13}(x) = 7$. 

129
Example 13.10. Let $x = 10110101001010$. This is same as the previous example, with an additional two bits appended at the end. The result of Lempel–Ziv parsing is

shortest phrases list = 1, 0, 11, 01, 010, 00, 10.

Note that the last phrase 10 in $x$ is already present in the “shortest phrases list,” so it is not repeated there. The code pairs in this case are

$$(0, 1), (0, 0), (1, 1), (2, 1), (4, 0), (2, 0), (1, 0), (1, 0).$$

The code corresponding to the last phrase 10 is the same as the one for the preceding 10 phrase. Since this time $x \in X_{15}$ and there are 8 code pairs, $c_{15}(x) = 8$. Notice that the length of the “shortest phrases list” may be shorter than the number of code pairs $c_n(x)$.

13.4.2. Length of Lempel–Ziv Codes

Since the length of the “shortest phrases list” is at most $c_n(x)$, pointer values range at most over $0, 1, \ldots, c_n(x)$. Hence, to represent each pointer requires at most $\lceil \log_2(c_n(x) + 1) \rceil$ bits. Each code pair requires this number of bits plus 1. Since there are a total of $c_n(x)$ code pairs, to represent all of them requires at most

$$\ell_{LZ}(x) := c_n(x) \left\{ \lceil \log_2(c_n(x) + 1) \rceil + 1 \right\} \text{ bits.}$$

Theorem 13.11. If $\{X_k\}$ is a stationary, ergodic, binary source, then

$$\lim_{n \to \infty} \frac{\ell_{LZ}(X_1 \cdots X_n)}{n} \leq H, \quad \text{a.s.}$$