Intuition for the Z Algorithm

The string $S = S[1..n]$ is processed left to right. In the general case, $3 \geq k \geq n$, and we know the Z_i, l_i, and r_i values for $2 \leq i \leq k-1$.

The string

$$\alpha = S[l_{k-1}..r_{k-1}]$$

is the "furthest" Z-box found so far. Hence,

$$\begin{aligned} \alpha &= S[l_{k-1}..r_{k-1}] \\ &= S[1..Z_{k-1}]; \end{aligned}$$

if $l_{k-1} = 0$, then all of these are the empty string ϵ. The value

$$k' = k - l_{k-1} + 1$$

is the length of the prefix of $S[l_{k-1}..r_{k-1}]$ that is before $S(k)$. We can write these equalities:

$$\begin{aligned} S[l_{k-1}..r_{k-1}] &= S[1..Z_{k-1}] \\ S(1) &= S(l_{k-1}) \\ S(k') &= S(k) \\ S(Z_{k-1}) &= S(r_{k-1}). \end{aligned}$$

Since the algorithm has matched through $S(Z_{k-1}) = S(r_{k-1})$, the interval

$$\beta = S[k..r_{k-1}]$$

is the "furthest" interval starting at position that has been explored so far. Moreover, we know that

$$S[k..r_{k-1}] = S[k'..Z_{k-1}].$$

We can illustrate our status as this:

$$\overbrace{S(1)\cdots \underbrace{S(k')\cdots S(Z_{k-1})}_{\beta}}^{\alpha}\cdots\overbrace{S(l_{k-1})\cdots \underbrace{S(k)\cdots S(r_{k-1})}_{\beta}}^{\alpha}.$$

Since we know Z'_k, we can compare it to $r_{k-1} - k + 1$ to determine how much of $S[k..r_{k-1}]$ can be part of a prefix of S. The cases in the algorithm follow from these observations.