**Localized Deletions**

When deletions occur in a transmitted sequence, the deleted bits are completely removed from the sequence and their positions are unknown at the receiver (unlike erasures). A burst of deletions refers to the case where a certain number of consecutive bits are deleted.

Localized deletions are a more generalized form of bursts of deletions. In this setting, \( a \leq b \) deletions are localized within a certain window of length \( b \). These deletions do not necessarily occur in consecutive positions.

For the problem of correcting a single burst of exactly \( b \) deletions, Levenshtein [1] showed that the asymptotic number of redundant bits needed is at least \( \log n + b - 1 \) bits, where \( n \) is the length of the codeword. Schoeny et al. [2] derived the same bound non-asymptotically.

The problem of correcting localized deletions arises in several applications. One example is the file synchronization application where a relatively small part of a large file is edited by deleting and inserting characters. Two remote nodes communicate interactively in order to synchronize the localized edits.

**Guess & Check Codes**

![Encoding block diagram of Guess & Check codes (3) for correcting \( a \leq b \) deletions that are localized within a single window (\( z=1 \)) of size at most \( b \) bits. Block I: The binary message of length \( k \) bits is chunked into adjacent blocks of length \( \log k \) bits each, and each block is mapped to its corresponding symbol in \( GF(q) \) where \( q = 2^{\log k} = k \). Block II: The resulting string is coded using a systematic \( (k, \log k + c, k, \log k) \) \( q \)-ary MDS code where \( c \) is the number of parity symbols. Block III: The symbols in \( GF(q) \) are mapped to their binary representations. Block IV: A buffer of \( k \) zeros followed by a single one is inserted between the systematic and the parity bits. Example: Length of message: \( k = 16 \), length of window: \( b = \log k = 4 \), field size: \( GF(16) \).

1. Encoding (bits in red get deleted):
   \[
   u = \begin{bmatrix}
   1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0
   \end{bmatrix}^T
   \]
   \[
   v = \begin{bmatrix}
   1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0
   \end{bmatrix}^T
   \]
   \[
   x = \begin{bmatrix}
   1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
   \end{bmatrix}^T
   \]
   \[
   \text{Decoding (3 guesses):}
   \]
   \[
   \text{Guess 1:}
   \]
   \[
   \text{Guess 2:}
   \]
   \[
   \text{Guess 3:}
   \]

**Results: Codes for Correcting Localized Deletions**

Theorem 1 (Code properties for correcting one set of localized deletions) Guess & Check (GC) codes can correct in polynomial time \( a \leq b \) deletions that are localized within a single window of size at most \( b \) bits, where \( m \log k + 1 \leq b \leq (m + 1) \log k + k \) for some constant integer \( m \geq 0 \). Let \( c > m + 2 \) be a constant integer. The code has the following properties:

1. Redundancy: \( n - k = c \log k + b + 1 \) bits.
2. Encoding complexity is \( \mathcal{O}(k \log k) \), and decoding complexity is \( \mathcal{O}(k^2/\log k) \).
3. Probability of decoding failure:
   \[
   \Pr(F) \leq \frac{k^{m+1}}{k^m \log k} - (m + 2) \frac{k^{m+3}}{k^m}
   \]

Theorem 2 (Code properties for correcting \( z > 1 \) sets of localized deletions) Guess & Check (GC) codes can correct in polynomial time \( z > 1 \) sets of \( a \leq b \) deletions, with each set being localized within a window of size at most \( b \) bits, where \( m \log k + 1 < b \leq (m + 1) \log k + 1 \) for some constant integer \( m \geq 0 \). Let \( c > z(m + 2) \) be a constant integer. The code has the following properties:

1. Redundancy: \( n - k = zc \log k + zb + z \) bits.
2. Encoding complexity is \( \mathcal{O}(k \log k) \), and decoding complexity is \( \mathcal{O}(k^{z+2}/\log^z k) \).
3. Probability of decoding failure:
   \[
   \Pr(F) = \mathcal{O}\left(\frac{k^{(m+1)}(z-1)}{k^m z \log^z k}\right)
   \]

**Numerical Results: Simulations on the probability of decoding failure**

The graph shows the probability of decoding failure \( \Pr(F) \) for GC codes for different message lengths \( k \). The results of \( \Pr(F) \) are averaged over 10000 runs of simulations. The window position in which the deletions are localized is also chosen uniformly at random.

1. Decoding (3 guesses):
   \[
   \text{Guess 1:}
   \]
   \[
   \text{Guess 2:}
   \]
   \[
   \text{Guess 3:}
   \]

**References**