Architektury systemów komputerowych

(studia dzienne)

Lista 12

$x_{12} = 0$ (minimum na bdb)

Zgodnie z obietnicą daję jeszcze możliwość zdobycia kilku punktów. Materiał na tej liście jest nadprogramowy i nie obowiązuje na egzaminie.

Najprostszym sposobem wykrywania pojedynczych błędów podczas transmisji jest dodawanie do przesyłanych informacji tzw. bitu parzystości. Oczywiście ten sposób nie daje możliwości automatycznego poprawienia błędu.

Na wykładzie wspomniałem o kodach korekcyjnych, czyli takich, które potrafią nie tylko wykrywać błędy, ale także automatycznie je poprawiać.

1. (2 pkt.) Opisz działanie kodu Hamminga poprawiającego pojedyncze błędy. Kod Hamminga można zastosować np. do znaków ASCII - do każdego kodu znaku dopisywane są cztery dodatkowe bity; jeżeli podczas transmisji wystąpi pojedynczy błąd, błąd ten można automatycznie poprawić, bez potrzeby ponownego przesyłania (odczytu).

2. W kodzie korekcyjnym/detekcyjnym tylko niektóre ciągi bitów reprezentują bezpośrednio znaki oryginalnego alfabetu. W kodzie parzystości są to te ciągi, które mają parzystą liczbę jedynek. Odległością Hamminga dla pary ciągów bitów nazywamy liczbę pozycji, na których się one różnią. Np. odległość Hamminga pomiędzy 010101010 a 010101100 wynosi 2. Odległością Hamminga dla całego kodu (rozumianego jako zbiór ciągów bitów tej samej długości) nazywamy minimalną odległość Hamminga dla dwóch ciągów reprezentujących znaki oryginalnego alfabetu. Np. dla kodu parzystości odległość Hamminga wynosi 2.

   Jaką odległość Hamminga powinien mieć kod, który umie

   (a) Wykrywać p przekłamań

   (b) Poprawiać p przekłamań

3. Załóżmy, że oryginalny alfabet jest kodowany na m bitach, a kod korygujący pojedyncze błędy dodaje r bitów nadmiarowych. Wykaż, że zachodzi następująca zależność:

$$m + r + 1 \leq 2^r.$$

   Korzystając z tej zależności pokaż, że nie da się zaprojektować kodu korygującego pojedyncze błędy dla alfabetu 128-znakowego używającego mniej niż 4 bitów dodatkowych.

Emanuel Kieroński