Ренормализация в арифметическом кодировании

5 января 2023 г. 1:23

Реализация арифметического кодирования
F - кумулятивная вероятность, которую вычисляем на каждом шаге
G - произведение вероятностей

- Обозначим через L (low) регистр, в котором хранится F, через R ($range$) регистр, в котором хранится G и введем регистр $H = L + R$ ($high$).
- В начале работы алгоритма $L = 0, H = 2^b - 1$, где b разрядность алгоритма.
- В результате операции $R = p_j R$ регистр R может обнулиться.
- Задача ренормализации держать регистры $H \geq \frac{3}{4}2^b$, $L \leq \frac{1}{4}2^b$, т.е. $R \geq \frac{1}{2}2^b$.

В алгоритме присутствует умножение R на вероятность p, то есть на число от 0 до 1, это приводит к тому, что R становится дробным, и если из раза в раз выполнять это умножение, в какой-то момент регистр R обнулится. Чтобы этого избежать, выполняется ренормализация. Ее суть в том, чтобы держать регистры в положении, из которого они не могут переполниться. Если они выходят из этого положения, то мы выдаем соответствующее количество бит на выход выполняем сдвиг в регистрах.

Принимаем во внимание модифицированную кумулятивную вероятность (формула была в "Код Гилберта-Мура", $\sigma_i = q_i + \frac{p_i}{2}$).

$\sigma = L + \frac{R}{2} = \frac{L}{2} + \frac{H}{2}$ ($H = L+R$, т.е. $\frac{L}{2} + \frac{H}{2} = \frac{L}{2} + \frac{L}{2} + \frac{R}{2} = L + \frac{R}{2}$)

После очередного умножения R на p возможны три сценария:

1. $H < \frac{1}{2}2^b$.
   - Тогда $\sigma < \frac{1}{2}2^b$.
   - Выдаём 0.
   - $H = H \times 2$, $L = L \times 2$.
2. $H \geq \frac{1}{2}2^b$, $L \geq \frac{1}{2}2^b$
   - Тогда $\sigma > \frac{1}{2}2^b$.
   - Выдаём 1.
   - $H = H \times 2 - 2^b$, $L = L \times 2 - 2^b$.
3. $\frac{1}{2}2^b < H < \frac{3}{4}2^b$, $\frac{1}{4}2^b \leq L < \frac{1}{2}2^b$
   - Тогда $\sigma <> \frac{1}{2}2^b$ (неопределённость).
   - $btf = btf + 1$
   - $H = H \times 2 - \frac{1}{2}2^b$, $L = L \times 2 - \frac{1}{2}2^b$.

Х — просто умножение как в коде Гилберта-Мура: делим отрезок пополам и смотрим, слева или справа σ

3 случай - неопределенность, т.е. не выполняется ни 1, ни 2. Старшие разряды кодового слова не определены на данном шаге. Это происходит в случае, когда у нас запись либо ...011111..., либо ...100000... Тогда мы ничего не делаем, а только накапливаем bits to follow - число, которое показывает, сколько раз в записи встречается 1 или 0 соответственно в ...011111... или ...100000...

См. ренормализацию двоичного арифметического кодирования в "Реализация двоичного арифметического кодирования без умножений"

Байтовая ренормализация (range coder) для недвоичного[3] и двоичного[4] алфавита

два вида байтовой ренормализации

while $(L \oplus (L + R)) < 2^{24}$ *or* $R < 2^{16}$ **do**
  if $R < 2^{16} \wedge (L \oplus (L + R)) \geq 2^{24}$ **then**
    $R \leftarrow (!L + 1) \wedge (2^{16} - 1)$
  end if
  PUTBYTE $(L \gg 24)$
  $R \leftarrow R \ll 8$
  $L \leftarrow L \ll 8$
end while

if $(L \oplus (L + R)) < 2^{24}$ **then**
  PUTBYTE $(L \gg 24)$
  $R \leftarrow R \ll 8$
  $L \leftarrow L \ll 8$
else if $R < 2^{16}$ **then**
  $R \leftarrow (!L + 1) \wedge (2^{16} - 1)$
  PUTBYTE $(L \gg 24)$
  $R \leftarrow R \ll 8$
  $L \leftarrow L \ll 8$
end if