

Отчет по лабораторной работе №1. Алгоритм РРМА

Мясников Владислав, М4106

Информация о вероятностном распределении символов представлена в виде дерева, где каждый узел соответствует некоторому контексту, который уже встречался при считывании (корень дерева соответствует пустому контексту длины 0). При этом с каждым узлом связан массив частот всех символов (то есть сколько раз тот или иной символ встречался в считанной последовательности после соответствующего контекста).

Помимо дерева узлов, также имеется дополнительное равномерное распределение тех символов, которые еще не встречались в последовательности. С данным распределением также связан массив частот.

Также вводится дополнительный ESC-символ для каждого узла (данный символ имеет частоту, равную единице), так как впоследствии он также будет кодироваться арифметическим кодировщиком.

Алгоритм работы архиватора:

1. Записываем в выходную последовательность первые 8 байт - количество кодируемых символов
2. Для каждого символа x из исходной последовательности, который нужно закодировать, выполняем следующее:
 - 2.1. Кодируем x при помощи арифметического кодировщика:
 - ищем наибольшую длину $d \leq D$ для текущего контекста, такую что контекст длины d ранее встречался
 - если для найденной длины контекста d символ x ни разу не встречался, то кодируем ESC-символ в текущем контексте длины d и декрементируем значение d (итерируемся до тех пор, пока контекст не станет пустым или частота x не станет положительной)
 - затем кодируем символ x , используя одно из распределений в зависимости от величины d и частоты символа x (распределение для непустого или пустого контекста, либо равномерное распределение для символов, которые ранее не встречались)
 - 2.2. Если x встретился впервые, то сбрасываем частоту, соответствующую этому x в модели равномерного распределения среди символов, которые ранее не встречались
 - 2.3. Инкрементируем частоту для x :
 - проходим по дереву контекстных распределений символов, учитывая текущий контекст, и инкрементируем частоту вхождения x
 - 2.4. Обновляем текущий контекст:
 - контекст представлен в виде массива байт (символов)
 - очередной символ добавляем в начало массива, сдвигая все остальные на одну позицию вправо
 - если количество символов стало больше максимальной длины D , то удаляем последний символ из массива (он встретился раньше остальных)

Алгоритм работы деархиватора

1. Считываем первые 8 байт входной последовательности и вычисляем n (количество закодированных символов)
2. Выполняем n раз декодирование символа x :

[illegible]