есть вначале исходная строка полностью помещается в буфер, в конце обработки строка полностью окажется в словаре.

Для наглядности покажем те последовательности буфера, которые выделит алгоритм:

compres[s]i[o]n a[n]d[ ]d[e]c[ompression and]c[ompression and decompressio]n

Далее каждое из этих совпадений будет заменено на комбинацию [смещение, длина, следующий символ]:

compres[1,1,i][8,1,n] a[3,1,d][4,1,d][12,1,c][18,15,c][34,25,n]

Можно увидеть, что количество элементов в выходной последовательности – 30, в отличии от количества элементов во входной последовательности – 63. Коэффициент сжатия составил 0.47.

Однако у алгоритма есть и недостаток: способ формирования кодов сравнительно неэффективен и позволяет сжимать только сравнительно длинные последовательности.

Также важной особенностью LZ77 является сильная несимметричность по времени – кодирование значительно медленнее декодирования, поскольку при компрессии значительное количество времени тратится на поиск совпадающих последовательностей[1)] [1].

2.3 Набор файлов Canterbury Corpus

Решение задачи сравнения алгоритмов по достигаемой ими степени сжатия требует введения некоторого критерия, так как нельзя сравнивать производительность реализаций на каком-то абстрактном файле. Следует осторожно относиться к теоретическим оценкам, так как они вычисляются с точностью до констант. Величины этих констант на практике могут колебаться в очень больших пределах, особенно при сжатии небольших файлов.

В 1997 году группой исследователей был предложен набор файлов, специально отобранных, чтобы служить в качестве эталона при проведении исследований алгоритмов сжатия. Этот набор был назван Canterbury Corpus (информационный фонд Кентербери). Отбор файлов осуществлялся на основании того, что результаты их обработки подтверждали теоретические исследования алгоритмов. Это давало надежду, что результаты обработки этих файлов новыми алгоритмами, которые будут изобретены в будущем, будут также достоверными [6]. Описание файлов, входящие в состав Canterbury Corpus, представлено в таблице 1.

[1)] Данный факт будет показан при исследовании алгоритмов в разделе 6