Сжатие с учётом контекста. Метод кодирования длин повторений (Run Length E Словарные методы, где словарём является несжатый текст — семейство LZ77

Александра Игоревна Кононова

МИЭТ

29 октября 2024 г. — актуальную версию можно найти на https://gitlab.com/illinc/otik

О терминах

Символ — элемент качественной информации $a \in A$ (множество A — алфавит).

Текст — последовательность $m \in A^+$ таких элементов.

На практике для всех алгоритмов, где алфавит может быть произвольным, для ЭВМ с 8-битным байтом **символ кодирования = байт** (00...FF),
исходный текст = любой бинарный файл, сжатый текст — тоже бинарный файл:

- использование в программе для ЭВМ символов, не кратных байту — неудобно;
- использование символов в два байта и более $\Longrightarrow$ слишком большой алфавит.
- использование в качестве символа кодирования печатного символа ASCII или koi8r/cp1251/dos/iso/maccyrillic не позволяет рассматривать в качестве исходного текста произвольный файл и приводит к труднодиагностируемым ошибкам;
- использование печатного символа UTF-8 (144 697 символов Unicode в 2023 г.) — то же самое + большой алфавит + переменная длина = проигрыш в объёме.

В книгах для наглядности используются обозначения $A, B, \ldots$ и т. п. (маленький алфавит + визуальное отличие символа от индекса или частоты), но в программе это всё равно байты!

Размер байта – 3 бита

Для компактной иллюстрации ограничений алгоритмов примем, что для устройства «доска» байт (символ кодирования) составляет не 8 бит — октет (как для Intel x86/amd64), и не 6 бит (как для IBM 7030 Stretch), а 3 бита — триаду, или одну 8-ричную цифру:

$0 = 000$
$1 = 001$
$2 = 010$
$3 = 011$
$4 = 100$
$5 = 101$
$6 = 110$
$7 = 111$

Методы кодирования, коды и алгоритмы с учётом контекста

Далее рассматриваются **методы сжатия с учётом контекста:** RLE, LZ77 и LZ78.

1. Оптимальный код источника Маркова, где $|code(c_1 \dots c_n)| \to I(c_1 \dots c_n)$ (Хф/АС с усл. вер.) не используется как метод сжатия: даже для 1 порядка нужны $256^2 = 65536$ частот.

2. Если метод Хаффмана однозначно лучше Шеннона—Фано и тем более Шеннона, то для сжатия с учётом контекста **нет однозначно лучшего метода.**

3. Для заданного метода — много кодов, **нет однозначно лучшей реализации.**

   Есть файлы, которые «наивный» RLE сжимает лучше и быстрее всего.

4. Среди схожих реализаций одного метода однозначно **худшей** (но обычно более наглядной) является та, где часть значений недопустима. «Наивный» RLE (L, c) хуже, чем $(L-1, c)$.

5. Для сложных кодов RLE и **любого кода LZ77 кодирование принципиально неоднозначно** (разные алгоритмы $\implies$ разные длины и скорости). Декодирование — однозначно.

Семейство RLE (Run Length Encoding) — концепция

Модель источника данных — Маркова первого порядка (аналоговый сигнал), при этом:

$$\forall a \neq b\colon \begin{cases} p(a|a) = p(b|b) = r, \\ p(a|b) = p(b|a) = s, \end{cases} \quad r \gg (T-1)s, \quad \text{где } T \text{ — размер алфавита.}$$

Run Length Encoding (RLE): $\mathrm{AAAAAAAABCCCC} \to 8 \times \mathrm{A},\ 1 \times \mathrm{B},\ 4 \times \mathrm{C}$

Повторение символа c подряд L раз $(L \times c)$ — цепочку длины L, $L_{\min} \leqslant L \leqslant L_{\max}$ — будем записывать как пару $\begin{Bmatrix} L \\ c \end{Bmatrix}$ (сжатая цепочка):

- цепочки длины более $L_{\max}$ символов — делятся на несколько;
- последовательности символов, где ни один не повторяется $L_{\min}$ раз подряд — несжатый текст.

RLE — не код, а **семейство кодов,** основанных на одном принципе сжатия и похожих моделях источника.

Способы отделения сжатых цепочек от несжатого текста; RLE-н-Δ

1. Несжатого текста нет: $L_{\min} = 1$ («наивная» реализация RLE). Для наивного RLE:
   - порядок записи L и c может быть любым: $\left\{ \begin{matrix} L \\ c \end{matrix} \right\} \sim (L - \Delta, c)$ или $(c, L - \Delta)$;
   - так как $L \geqslant L_{\min} = 1$, можно записывать L кодом со смещением:
     $\left\{ \begin{matrix} L \\ c \end{matrix} \right\} \sim (L - 1, c)$, $(c, L - 1)$ [макс. смещ.: $\Delta = 1$]; (L, c) или (c, L) [без смещения: $\Delta = 0$].

   Рассматриваем $|L| = |c| = k$ бит, код с максимальным смещением и порядок $(L - 1, c)$:
   $0 \leqslant L - 1 \leqslant 2^k - 1 \implies 1 \leqslant L \leqslant 2^k \implies L_{\max} = 2^k$; для трёхбитного байта ($k = 3$) $L_{\max} = 8$.

   $m = 7777\,7000\,0000\,0000\,0123\,4567\,0011\,2233\,4455\,6677$ (40 трёхбитных байтов)
   $\to \left\{ \begin{matrix} L = 5 \\ c = 7 \end{matrix} \right\} \left\{ \begin{matrix} L = 8 \\ c = 0 \end{matrix} \right\} \left\{ \begin{matrix} L = 4 \\ c = 0 \end{matrix} \right\} \left\{ \begin{matrix} L = 1 \\ c = 1 \end{matrix} \right\} \left\{ \begin{matrix} L = 1 \\ c = 2 \end{matrix} \right\} \cdots \left\{ \begin{matrix} L = 1 \\ c = 7 \end{matrix} \right\} \left\{ \begin{matrix} L = 2 \\ c = 0 \end{matrix} \right\} \left\{ \begin{matrix} L = 2 \\ c = 1 \end{matrix} \right\} \left\{ \begin{matrix} L = 2 \\ c = 2 \end{matrix} \right\} \cdots \left\{ \begin{matrix} L = 2 \\ c = 7 \end{matrix} \right\}$
   $code(m) = 4770\,3001\,0203\,0405\,0607\,1011\,1213\,1415\,1617$ (36 трёхбитных байтов)
2. Несжатый текст тоже группируется в цепочки, различение — флаг-биты (RLE с флаг-битом).
3. Несжатый текст записывается как есть, сжатые цепочки предваряются односимвольным префиксом (RLE с префиксом).

RLE с флаг-битом (несжатый текст тоже группируется в цепочки), тезисы

Наивный RLE особенно плох на фрагментах типа 0123...: один байт невыгодно записывать двумя.

Зададимся $L_{\min}^{\text{сж}}$ таким, что $L \geqslant L_{\min}^{\text{сж}}$ повторяющихся байтов выгодно записывать двумя байтами;

такие последовательности байтов, где ни один не повторяется хотя бы $L_{\min}^{\text{сж}}$ подряд, тоже дополняем служебным байтом с общей длиной L — несжатая цепочка $\left\{ \begin{matrix} L \\ c_1 \ldots c_L \end{matrix} \right\}$, где $L \geqslant 1$ включительно ($L_{\min}^{\text{несж}} = 1$).

- Для различения сжатых и несжатых цепочек (два вида) достаточно одного бита θ — флаг-бита.
  Флаг-бит θ и длина цепочки L упаковываются в один байт (k бит):
  1. 1 бит на θ (значения 0 и 1) и $k-1$ бит на L ($0 \leqslant L - \Delta \leqslant 2^{k-1} - 1$);
  2. длина кода сжатой цепочки $\left\{ \begin{matrix} L \\ c \end{matrix} \right\}$: $|L \cup \theta| + |c| = 2$ байта,

     длина кода несжатой $\left\{ \begin{matrix} L \\ c_1 \ldots c_L \end{matrix} \right\}$: $|L \cup \theta| + |c_1 \ldots c_L| = 1 + L$ байтов.
- Длина кода несжатой цепочки непостоянна $\Longrightarrow$ байт $L \cup \theta$ в начале.

RLE с флаг-битом — варианты RLE-фб$L^{\text{сж}}_{\min}$-$\Delta_{\text{сж}}\Delta_{\text{несж}}$

1. Флаг-бит θ сж/несж может быть $0/1$ или $1/0$ — не влияет на длину кода.
2. Положение бита θ в байте $L \cup \theta$ — не влияет на длину кода; обычно θ — старший: удобнее читать дамп.
3. Выбор $L^{\text{сж}}_{\min}$ — между 2 и 3:
   - однократное вхождение символа всегда невыгодно записывать двумя символами $\implies L^{\text{сж}}_{\min} > 1$;
   - трёхкратное — всегда выгодно $\implies L^{\text{сж}}_{\min} \leqslant 3$.

   Иногда выгоднее 2 (файл 001122), иногда — 3 (файл 0112); узнать без кодирования невозможно.

   $L^{\text{сж}}_{\max}$ и $L^{\text{несж}}_{\max}$ не выбираются, а **рассчитываются** на основе разрядности $k-1$ смещённого L и $\Delta_{\text{сж}}$ и $\Delta_{\text{несж}}$: $L_{\max} = 2^{k-1} - 1 + \Delta$ $(\Delta_{\text{сж}} \neq \Delta_{\text{несж}} \implies L^{\text{сж}}_{\max} \neq L^{\text{несж}}_{\max})$.
4. Код со смещением для L:
   - без смещ. $(\Delta_{\text{сж}} = \Delta_{\text{несж}} = 0) \implies$ меньше $L_{\max} \implies$ самый длинный код для выбранного $L^{\text{сж}}_{\min}$;
   - минимальное <ненулевое> смещение: $L-1$ как для сжатых, так и для несжатых $(\Delta_{\text{сж}} = \Delta_{\text{несж}} = 1)$;
   - максимальное смещение: записывать $L - L_{\min}$ (разные для сжатых и несжатых цепочек: $\Delta_{\text{сж}} = L^{\text{сж}}_{\min}$, $\Delta_{\text{несж}} = 1$) $\implies$ самый короткий код для выбранного $L^{\text{сж}}_{\min}$;
   - смещения между макс. и мин. допустимы, но не дают выигрыша ни в длине, ни в читаемости дампа.

Относительная величина проигрыша $\Delta_{\text{сж}} = \Delta_{\text{несж}} = 0$ относительно $\Delta_{\text{сж}} = L^{\text{сж}}_{\min}$ и $\Delta_{\text{несж}} = L^{\text{несж}}_{\min} = 1$ меньше для $k = 8$ (октета), чем для $k = 3$. Но проигрыш *есть;* и найдутся файлы, где он проявится.

RLE с флаг-битом — варианты (продолжение) и выбор RLE-фб3-31

Экзотические варианты кодирования: выбор $L^{\text{сж}}_{\min} > 3$ (в частности, 4), или запись в $k-1$ бит длины не $L-\Delta$, а, например, $\frac{L}{2}$ (рассматривать только чётные L, а «лишний» байт добавлять к несжатой цепочке — технически **возможны.**

Почти для любого «безумного» кода найдётся файл, для которого именно этот код будет лучшим!
И только код без смещения L = код с неиспользуемыми кодовыми позициями = код с заведомой избыточностью всегда не лучше (и на некоторых файлах хуже) аналога без избыточности.

Но *все* варианты RLE с флаг-битом рассмотреть в рамках лекций невозможно — слишком много.

Ниже используются:

1. флаг-бит θ сж/несж — $1/0$;
2. в k-битном байте $|L \cup \theta|$ флаг θ — старший бит байта $L \cup \theta$, $L - L_{\min}$ в $(k-1)$ младших; $k = 3$;
3. $L^{\text{сж}}_{\min} = 3$;
4. Макс. смещение:
$$\begin{cases} \begin{Bmatrix} L^{\text{сж}} \\ c \end{Bmatrix} \to \Big((\theta = 1, L^{\text{сж}} - 3), c\Big), & 3 \leqslant L^{\text{сж}} \leqslant 2^{k-1} + 2 \\ \begin{Bmatrix} L^{\text{несж}} \\ c_1 \ldots c_L \end{Bmatrix} \to \Big((\theta = 0, L^{\text{несж}} - 1), c_1 \ldots c_L\Big), & 1 \leqslant L^{\text{несж}} \leqslant 2^{k-1} \end{cases}$$

RLE с флаг-битом, кодирование с выбранными параметрами

$$\begin{cases} \left\{ \begin{matrix} L^{\text{сж}} \\ c \end{matrix} \right\} \to \Big((\theta = 1, L^{\text{сж}} - 3), c \Big), & 3 \leqslant L^{\text{сж}} \leqslant 6 \\ \left\{ \begin{matrix} L^{\text{несж}} \\ c_1 ... c_L \end{matrix} \right\} \to \Big((\theta = 0, L^{\text{несж}} - 1), c_1 ... c_L \Big), & 1 \leqslant L^{\text{несж}} \leqslant 4 \end{cases}$$

$m = 7777\,7000\,0000\,0000\,0123\,4567\,0011\,2233\,4455\,6677$ (40 трёхбитных байтов)

$$\left\{ \begin{matrix} 1 \\ 5 \\ 7 \end{matrix} \right\} \left\{ \begin{matrix} 1 \\ 6 \\ 0 \end{matrix} \right\} \left\{ \begin{matrix} 1 \\ 6 \\ 0 \end{matrix} \right\} \left\{ \begin{matrix} 0 \\ 4 \\ 1234 \end{matrix} \right\} \left\{ \begin{matrix} 0 \\ 4 \\ 5670 \end{matrix} \right\} \left\{ \begin{matrix} 0 \\ 4 \\ 0112 \end{matrix} \right\} \left\{ \begin{matrix} 0 \\ 4 \\ 2334 \end{matrix} \right\} \left\{ \begin{matrix} 0 \\ 4 \\ 4556 \end{matrix} \right\} \left\{ \begin{matrix} 0 \\ 3 \\ 677 \end{matrix} \right\}$$

$$\Big((1, 5-3=2), 7\Big) \Big((1, 6-3=3), 0\Big) \Big((1, 6-3=3), 0\Big) \Big((0, 4-1=3), 1234\Big) \Big((0, 4-1=3), 5670\Big)$$
$$\Big((0, 4-1=3), 0112\Big) \Big((0, 4-1=3), 2334\Big) \Big((0, 4-1=3), 4556\Big) \Big((0, 3-1=2), 677\Big)$$

$$(110, 7)\ (111, 0)\ (111, 0)\ (011, 1234)\ (011, 5670)\ (011, 0112)\ (011, 2334)\ (011, 4556)\ (010, 677)$$

$code(m) = 677070312343567030112323343455626 77$ (35 трёхбитных байтов)

RLE с односимвольным префиксом, тезисы

Текст, который не имеет смысла записывать как сжатые цепочки (несжатый текст) — пишется как есть, без дополнительных служебных байтов.

Чтобы отделить сжатые цепочки $\left\{ \begin{matrix} L \\ c \end{matrix} \right\}$ от несжатого текста, они предваряются *префиксом* — выбираемым для каждого файла индивидуально байтом p; то есть цепочка записывается тремя байтами.

- длина кода цепочки $\left\{ \begin{matrix} L \\ c \end{matrix} \right\}$ 3 байта $\implies$ имеет смысл только при $L \geqslant 4$;
- в несжатом тексте может встретиться любой байт, в том числе p $\implies$ нужно экранировать p, pp и ppp (четырёхкратное $pppp$ в любом случае цепочка $\left\{ \begin{matrix} L = 4 \\ p \end{matrix} \right\}$);
- любой способ экранирования одного байта p в несжатом тексте длиннее байта $\implies$ p выбирается для конкретного файла как самый редкий байт (в идеале — отсутствующий); следовательно, $count(p) \leqslant \frac{n}{2^k}$;
- кодирование в два прохода по файлу: 1) поиск p, 2) сжатие RLE (наивный и флаг-биты — в один проход);
- значение p сохраняется в заголовке файла — нужно для декодирования.

RLE с префиксом — варианты RLE-$p1p/p0p0p0/p0p2p/p0p0p3p$-$\Delta_{c\neq p}\Delta_p$

1. Способ экранирования в несжатом тексте p, pp и ppp:
   1. p как цепочка $\begin{Bmatrix}1\\p\end{Bmatrix}$, pp как $\begin{Bmatrix}2\\p\end{Bmatrix}$, ppp как $\begin{Bmatrix}3\\p\end{Bmatrix}$ $\Longrightarrow$ а) $L^p_{\min}=1$; $L^{c\neq p}_{\min}=4$; б) порядок любой: $(p, L-\Delta, c)$ или $(p, c, L-\Delta)$, причём $\Delta \leqslant L_{\min}$;
   2. p как $p0$, pp как $p0p0$, ppp как $p0p0p0$ $\Longrightarrow$ а) $L^p_{\min}=L^{c\neq p}_{\min}=4$; б) код $\begin{Bmatrix}L\\c\end{Bmatrix}$ не должен начинаться с $p0$ $\Longrightarrow$ следующий после p байт ненулевой $\Longrightarrow$ порядок $(p, \underbrace{L-\Delta}_{\neq 0}, c)$, причём $\Delta \leqslant L_{\min}-1=3$;
   3. p как $p0$, но pp как цепочка $\begin{Bmatrix}2\\p\end{Bmatrix}$ $\Longrightarrow$ а) $L^p_{\min}=2$; $L^{c\neq p}_{\min}=4$; б) порядок $(p, \underbrace{L-\Delta}_{\neq 0}, c)$, $\Delta \leqslant L_{\min}-1$;
   4. p как $p0$, pp как $p0p0$, ppp как $\begin{Bmatrix}3\\p\end{Bmatrix}$ $\Longrightarrow$ а) $L^p_{\min}=3$...
2. Код со смещением для L — аналогично предыдущим; макс. Δ_p и $\Delta_{c\neq p}$ — в общем случае разные.

   При макс. смещении L *для любого* из способов экранирования 1–4 найдётся файл, где этот способ будет лучше трёх остальных (где хуже код p, pp и ppp — там лучше $L_{\max}$).

RLE с префиксом, одиночный p как цепочка [RLE-p1p-41]

Выбираем порядок (p, L, c) и код с максимальным смещением:

$$\begin{cases} \left\{\begin{matrix} L \\ c \neq p \end{matrix}\right\} \to \Big(p, L-4, c\Big), & 0 \leqslant L^{c\neq p} - 4 \leqslant 2^k - 1 \iff 4 \leqslant L^{c\neq p} \leqslant 2^k + 3 \\ \left\{\begin{matrix} L \\ p \end{matrix}\right\} \to \Big(p, L-1, p\Big), & 0 \leqslant L^{p} - 1 \leqslant 2^k - 1 \iff 1 \leqslant L^{p} \leqslant 2^k \end{cases}$$

$k = 3$: $\quad 4 \leqslant L^{c\neq p} \leqslant 11, \quad 1 \leqslant L^{p} \leqslant 8$

$m = 7777\,7000\,0000\,0000\,0123\,4567\,0011\,2233\,4455\,6677$ (40 трёхбитных байтов)
самые редкие $\{1, 2, 3, 4, 5, 6\}$ — выбираем префикс $p = 1$

$$\left\{\begin{matrix} 5 \\ 7 \neq p \end{matrix}\right\} \left\{\begin{matrix} 11 \\ 0 \neq p \end{matrix}\right\} 0 \left\{\begin{matrix} 1 \\ 1 = p \end{matrix}\right\} 23456700 \left\{\begin{matrix} 2 \\ 1 = p \end{matrix}\right\} 223344556677$$

$$\left\{\begin{matrix} 5-4=1 \\ 7 \neq p \end{matrix}\right\} \left\{\begin{matrix} 11-4=7 \\ 0 \neq p \end{matrix}\right\} 0 \left\{\begin{matrix} 1-1=0 \\ 1 = p \end{matrix}\right\} 23456700 \left\{\begin{matrix} 2-1=1 \\ 1 = p \end{matrix}\right\} 223344556677$$

$code(m) = 11717001012345670011122334455667 7$ (33 трёхбитных байта)

RLE с префиксом, одиночный p как $p0$, сдвоенный как цепочка [RLE-p0p2p-31]

Выбираем код с максимальным смещением:

$$\begin{cases} \begin{Bmatrix} L \\ c \neq p \end{Bmatrix} \to \Big(p, L-3, c\Big), & 1 \leqslant L^{c\neq p} - 3 \leqslant 2^k - 1 \iff 4 \leqslant L^{c\neq p} \leqslant 2^k + 2 \\ \begin{Bmatrix} 1 \\ p \end{Bmatrix} \to \Big(p, 0\Big), & \\ \begin{Bmatrix} L \geqslant 2 \\ p \end{Bmatrix} \to \Big(p, L-1, p\Big), & 1 \leqslant L^p - 1 \leqslant 2^k - 1 \iff 2 \leqslant L^p \leqslant 2^k \end{cases}$$

$k = 3$: $\quad 4 \leqslant L^{c\neq p} \leqslant 10, \quad 2 \leqslant L^p \leqslant 8$

$m = 7777\,7000\,0000\,0000\,0123\,4567\,0011\,2233\,4455\,6677$ (40 трёхбитных байтов), выбираем $p = 1$

$$\begin{Bmatrix} 5 \\ 7 \neq p \end{Bmatrix} \begin{Bmatrix} 10 \\ 0 \neq p \end{Bmatrix} 00 \begin{Bmatrix} 1 \\ 1 = p \end{Bmatrix} 23456700 \begin{Bmatrix} 2 \\ 1 = p \end{Bmatrix} 223344556677$$

$$\begin{Bmatrix} 5-3=2 \\ 7 \neq p \end{Bmatrix} \begin{Bmatrix} 10-3=7 \\ 0 \neq p \end{Bmatrix} 00 \Big(p, 0\Big) 23456700 \begin{Bmatrix} 2-1=1 \\ 1 = p \end{Bmatrix} 223344556677$$

$code(m) = 12717000102345670011122334455667 7$ (33 трёхбитных байта)

Лучший и худший случаи RLE (для октетов: $k = 8$)

Лучший исходный файл длины n для всех RLE — $m_л = \underbrace{0000...0000}_{n}$;

худший — $m_х = \underbrace{0123...0123...}_{n}$ (все байты от 0 до $2^k - 1$ [октеты от 00 до FF]; нет ни одного повторения).

Считаем $n \gg L_{\max}$ («хвостом» можно пренебречь);
$L_{\max}$ оцениваем по порядку величины (без смещения будет на 1 меньше, с макс. — немного больше).
Все длины $|code|$ оцениваются приблизительно (кроме $|code(m_х)|$ для RLE-н).

$code$	$L_{\max}$	$L_{\max}\|_{k=8}$	$\|code(m_л)\|$	$\|code(m_л)\|_{k=8}$	$\|code(m_х)\|$	$\|code(m_х)\|_{k=8}$
RLE-н	2^k	256	$2 \cdot \frac{n}{L_{\max}} \approx \frac{n}{2^{k-1}}$	$\frac{n}{128} = 2 \cdot \frac{n}{256}$	$= 2 \cdot n$	$2 \cdot n$
RLE-фб	2^{k-1}	128	$2 \cdot \frac{n}{L_{\max}} \approx \frac{n}{2^{k-2}}$	$\frac{n}{64} = 4 \cdot \frac{n}{256}$	$(1 + \frac{1}{L_{\max}}) \cdot n$	$(1 + \frac{1}{128}) \cdot n$
RLE-$p0*$	2^k	256	$3 \cdot \frac{n}{L_{\max}} \approx \frac{3n}{2^k}$	$3 \cdot \frac{n}{256}$	$n + count(p) \leqslant (1 + \frac{1}{2^k}) \cdot n$	$(1 + \frac{1}{256}) \cdot n$
RLE-$p1p$					$n + 2count(p) \leqslant (1 + \frac{1}{2^{k-1}}) \cdot n$	$(1 + \frac{1}{128}) \cdot n$

RLE-$p*$ лучше RLE-фб в наилучшем и наихудшем случаях; но есть файлы, где RLE-фб лучше.
RLE-н лучше всех прочих в наилучшем случае (все байты входят сериями по $L_{\max}$ раз).

Модель RLE-н: построение

Зададимся конкретным кодом $code$ семейства RLE — «наивным» $(L-1, c)$ (RLE-н-1) для k-битного байта (алфавит — из $T = 2^k$ байтов; $L_{max} = 2^k = T$)

и попробуем построить модель X (аналоговый сигнал по определению RLE), для которой $|code(C)| \to I_X(C)$ не только в среднем по X, но и для любого C.

1. Из симметрии RLE-н $\quad \forall a \neq b$: $\begin{cases} p(a|a) = p(b|b) = r, \\ p(a|b) = p(b|a) = s. \end{cases}$ (*)
2. Из предположений RLE $r \gg (T-1)s$.
3. Код RLE-н-1 не зависит от конкретных частот — r и s не оцениваются по файлу (фиксированы);

Найдём r и s такие, что $|code(C)| = I_X(C)$ для достаточно длинного сообщения ($\alpha \to \infty$):

Случай	C	$\lvert code(C)\rvert = I_X(C)$	$p(C)$ по $I_X(C)$	$p(C)$ по C	Итог
лучший	$\underbrace{aaa..aaa}_{\alpha \cdot L_{max}}$	2α байтов ($2\alpha k$ бит)	$\frac{1}{2^{2\alpha k}} = \frac{1}{T^{2\alpha}}$	$\frac{1}{T} \cdot r^{\alpha \cdot L_{max} - 1}$	$r = T^{-\frac{2\alpha-1}{\alpha T-1}} = T^{-\frac{2-\frac{1}{\alpha}}{T-\frac{1}{\alpha}}}$
худший	$\underbrace{abc...}_{\alpha}$	2α байтов ($2\alpha k$ бит)	$\frac{1}{2^{2\alpha k}} = \frac{1}{T^{2\alpha}}$	$\frac{1}{T} \cdot s^{\alpha - 1}$	$s = T^{-\frac{2\alpha-1}{\alpha-1}} = T^{-\frac{2-\frac{1}{\alpha}}{1-\frac{1}{\alpha}}}$

Модель RLE-н: расчёт r и s лучшего и худшего случаев $\Longrightarrow$ нет точной модели

Для октетов $k = 8$, $T = 256$ (для $k \neq 8$ значения другие, но $s_{\text{х}\alpha} < s_{\text{л}\alpha}$, $s_{\text{х}\infty} < s_{\text{л}\infty}$ сохраняется)

Случай	лучший, $\alpha \to \infty$ (л∞)	лучший, $\alpha = 1$ (л1)	худший, $\alpha \to \infty$ (х∞)	худший, $\alpha = 2$ (х2)
r	$0{,}9576 \approx 1 - 0{,}04$	$0{,}9785 \approx 1 - 0{,}02$	$0{,}9961 \approx 1 - 0{,}004$	$1 - 1{,}5 \cdot 10^{-5}$
s	$1{,}7 \cdot 10^{-4}$	$8 \cdot 10^{-5}$	$1{,}5 \cdot 10^{-5}$	$6 \cdot 10^{-8}$

- на л1 и х2 сильно влияет первая вероятность, которую приняли $\frac{1}{T}$ $\Longrightarrow$ не рассматриваем их;
- $s_{\text{л}\infty} \neq s_{\text{х}\infty}$ $\Longrightarrow$ не существует такого X вида (*), чтобы $|code(x)| \to I_X(x)$ для всех x из X; **то есть нет такой модели X вида (*), чтобы оценка $|code(x)|$ как $I_X(x)$ была, как для Хаффмана — всегда близкой и иногда достижимой.**
- невозможно сделать вывод, существует ли такой X вида (*), что $|code(X)| \to I(X)$ в среднем по X.

Для источника с $p(a|b) \in [s_{\text{х}\infty}, s_{\text{л}\infty}]$ или того же порядка RLE-н-1 с 8-битным $\widetilde{L}$ **субоптимален** и лучше других RLE-н (скорее всего — лучше всех RLE и даже всех LZ77 и всех LZ78).

При $p(a|b) \lll s_{\text{х}\infty}$ субоптимальным будет RLE-н, где $|\widetilde{L}| > 8$ бит (но $|c| = 8$, символ = байт).

При $1 \gg p(a|b) \gg s_{\text{л}\infty}$ сложные RLE лучше «наивного» (если $p(a|b)$ не малы — все RLE плохи).
Оценка по худшему случаю строже $\Longrightarrow$ в сложных RLE необходимо уменьшать $|code(\text{х}\alpha)|$.

Задачи: модель источника RLE

1. Оценить $I(m)$ по моделям $X_л$, $X_х$ для сообщений длины $n \to \infty$ $m_1 = 0000...0000$ (наилучший случай) и $m_2 = 0123...4567$ (наихудший случай); сравнить с длиной наивного RLE $(L-1, c)$ этих сообщений $\left(\text{рассчитать } \frac{code(m)}{I_i(m)}\right)$.
2. При каком r имеет смысл наивный RLE с двухоктетным L?
3. Построить приближённую модель для RLE с флаг-битом (считать, что $L_{\max} \approx 2^{k-1} = \frac{T}{2}$ для всех цепочек).
   Как отличаются вероятности от модели наивного RLE?

Концепция LZ77, код Зива–Лемпеля LZ77/LZ1

1977 г., Якоб Зив (Jacob Ziv) и Абрахам Лемпель (Abraham Lempel)

J. Ziv and A. Lempel, «A universal algorithm for sequential data compression», in IEEE Transactions on Information Theory, vol. 23, no. 3, pp. 337-343, May 1977, doi: 10.1109/TIT.1977.1055714.

— идея замены слова ссылкой $\left\{\begin{matrix} S \\ L \end{matrix}\right\}$ и концепт $(S, L, c)/(0, 0, c)$:

если цепочка символов (не обязательно одинаковых) длины $L_{\min} \leqslant L \leqslant L_{\max}$ (слово) встречается более одного раза, то каждое **следующее вхождение** слова **заменяется ссылкой на предыдущее.**

Ссылка $\left\{\begin{matrix} S \\ L \end{matrix}\right\}$ состоит из:

- относительного смещения $S \geqslant 1$ предыдущего вхождения слова относительно текущей позиции;
- длины L слова.

Скользящее окно: область перед текущей позицией кодирования, в которой можем искать и адресовать ссылки ($S_{\min} \leqslant S \leqslant S_{\max}$).

Поиск выполняется по несжатому тексту (при декодировании — по **уже разжатой** части)!

В окне поиска нет ни ссылок $\left\{\begin{matrix} S \\ L \end{matrix}\right\}$, ни флаг-байтов, ни экранирующих нулей, ни иных служебных структур.

Неоднозначность кодирования LZ77

В окне $1 \leqslant S \leqslant S_{\max}$ может быть несколько подходящих совпадений $L_{\min} \leqslant L \leqslant L_{\max}$:

$S_{\min} = 1$ всегда. Пусть $S_{\max} = 16$, $L_{\min} = 4$, $L_{\max} = 6$.

																16	15	14	13	12	11	10	09	08	07	06	04	03	02	01	↓
...	7	6	5	1	1	2	3	1	2	5	1	2	3	1	6	1	2	3	1	2	3	1	2	3	1	1	...		

1. $\begin{cases} S_1 = 15 \\ L_1 = 5 \end{cases}$, 2. $\begin{cases} S_2 = 9 \\ L_2 = 4 \end{cases}$, 3. $\begin{cases} S_3 = 3 \\ L_3 = 6 \end{cases}$ (при бóльшем $L_{\max}$ было бы 7).

Любая одна из этих пар будет допустимым кодом слова в текущей (↓) позиции.

- полный перебор всех смещений ($S_{\min}...S_{\max}$) или ускоренный, по вспомогательным структурам данных (как в LZSS)
- поиск первого совпадения длины $L_{\min}$ (слабее сжатие) или поиск наилучшего (медленно);
- по возрастанию абсолютного адреса (от $S_{\max}$ к $S_{\min}$) или по убыванию и т. д.

Поиск предыдущего вхождения — основная сложность семейства LZ77 и неоднозначность кодирования.

Декодирование любого кода семейства LZ77 однозначно.

Семейство LZ77 — варианты

LZ77, как и RLE — **семейство кодов,** основанных на одной идее; так как семейство LZ77 чаще применяется на практике — многие коды семейства получили собственные имена. Отличаются:

1. способом отделения ссылок $\left\{\begin{matrix}L\\S\end{matrix}\right\}$ от несжатого текста $\Big($концепт Зива–Лемпеля $\Big/$ флаг-байт ссылка/символ $\Big/$ флаг-бит ссылка/цепочка $\Big/$ префикс и т. д.$\Big)$;
2. порядком $(L,S)/(S,L)$ для тех случаев, где порядок не определён способом отделения;
3. выбором $L_{\min}$: значение $\inf(L_{\min})$, то есть наименьшее L такое, что ссылка $\left\{\begin{matrix}L\\S\end{matrix}\right\}$ занимает меньший объём, чем несжатые L символов, определяется способом кодирования; но можно задать $L_{\min} > \inf(L_{\min})$ (в частности, $L_{\min} = 4$ вместо 3 — хуже сжатие, но быстрее поиск).
4. разрядностью L и S: для сильно избыточных файлов лучше $|L| = |S| = k$ бит (больше $L_{\max}$), для малоизбыточных $|L| = k - \delta$ бит, $|S| = k + \delta$ бит (больше окно).
   Но слишком большое окно замедляет кодирование — 16 разрядов много.
5. (не)использованием для L и S кода со смещением и другими деталями реализации.

Алгоритм семейства LZ77 = код + алгоритм поиска оригинала в окне. Так, алгоритм LZSS (Сторер, Сжимански) — флаг-бит ссылка/символ и битовый выходной поток + дерево для ускорения поиска.

Выбор для примеров ниже (реализовать иначе — можно, но не всё стоит)

1. Символ=байт, $|c| = 1$ байт $= k$ бит (примеры для $k = 3$). Выходной поток байтовый. $S_{\min} = 1$.

2. Принимаем $L_{\min} = \inf(L_{\min})$, не увеличиваем (если способ кодирования не даёт однозначного $\inf(L_{\min})$ — берём наибольший из вариантов).

3. Рассматриваем на примере малоизбыточных файлов $\Longrightarrow$ окно $S_{\max}$ желательно вчетверо больше алфавита, минимум вдвое $\Longrightarrow$ разрядность S от $k+2$, минимум от $k+1$.

   Считаем, что L и S в сумме занимают **ровно два байта.** 3-битный байт, примеры: $|L| = 2$ бита, $|S| = 4$ бита ($|L| : |S| = 2 : 4$) — двухбитное L дополняется старшим битом S до байта; Октет: есть реализация $|L| : |S| = 6 : 10$ (LZJB).

   Флаг-биты ссылка/цепочка рассматриваем на примере $|S| = 1$ байт $= 3$ бита.

4. Везде, где можно, **для L и S используется код со смещением.**
   Обозначим смещённое S как $\widetilde{S} = S - \Delta_S$ и смещённое L как $\widetilde{L} = L - \Delta_L$.

Концепт Зива— Лемпеля [LZ77-к]

Ссылки чередуются с несжатыми символами: ссылка $\begin{Bmatrix} S \\ L \end{Bmatrix}$, за которой следует c — тройка (S, L, c);

если в окне не на что дать ссылку — тройка $(0, 0, c)$.

При необходимости исходный текст дополняется (обычно нулями).

$L_{\min} = S_{\min} = 1$, ноль используется как специальное $\implies$ код со смещением невозможен.
Примем $k = 3$, $|L| : |S| = 2 : 4 \implies S_{\max} = 2^4 - 1 = 15$, $L_{\max} = 2^2 - 1 = 3$

$m = 7777\,7000\,0000\,0000\,0123\,4567\,0123\,4343\,7012\,1077$ (40 трёхбитных байтов)

$$\begin{Bmatrix} 0 \\ 0 \\ - \end{Bmatrix} 7 \begin{Bmatrix} S=1 \\ L=3 \\ 777 \end{Bmatrix} 7 \begin{Bmatrix} 0 \\ 0 \\ - \end{Bmatrix} 0 \begin{Bmatrix} S=1 \\ L=3 \\ 000 \end{Bmatrix} 0 \begin{Bmatrix} S=1 \\ L=3 \\ 000 \end{Bmatrix} 0 \begin{Bmatrix} S=1 \\ L=3 \\ 000 \end{Bmatrix} 1 \begin{Bmatrix} 0 \\ 0 \\ - \end{Bmatrix} 2 \begin{Bmatrix} 0 \\ 0 \\ - \end{Bmatrix} 3 \begin{Bmatrix} 0 \\ 0 \\ - \end{Bmatrix} 4 \begin{Bmatrix} 0 \\ 0 \\ - \end{Bmatrix} 5 \begin{Bmatrix} 0 \\ 0 \\ - \end{Bmatrix} 6 \begin{Bmatrix} 0 \\ 0 \\ - \end{Bmatrix} 7$$

$$\begin{Bmatrix} S=8 \\ L=3 \\ 012 \end{Bmatrix} 3 \begin{Bmatrix} S=8 \\ L=1 \\ 4 \end{Bmatrix} 3 \begin{Bmatrix} S=2 \\ L=2 \\ 43 \end{Bmatrix} 7 \begin{Bmatrix} S=9 \\ L=3 \\ 012 \end{Bmatrix} 1 \begin{Bmatrix} S=4 \\ L=1 \\ 0 \end{Bmatrix} 7 \begin{Bmatrix} S=1 \\ L=1 \\ 7 \end{Bmatrix} 0$$

— 18 троек $\begin{Bmatrix} S \\ L \end{Bmatrix} + c$, каждая записывается тремя байтами $\implies$ 54 байта в коде.

$code(m) = 007137000130\ldots$

LZ77 с флаг-байтами ссылка/символ [LZ77-cc]

Символы и ссылки группируются по k штук, каждая группа предваряется байтом, где каждый бит показывает тип соответствующего объекта (флаг-бит). Без группировки — битовый выходной поток.

При необходимости исходный текст дополняется (обычно нулями).

- флаг-бит θ ссылка/символ может быть $0/1$ или $1/0$; ниже рассматривается $1/0$;
- порядок S и L в ссылке любой; примем (S, L);
- длина кода ссылки $\begin{Bmatrix} S \\ L \end{Bmatrix}$ с учётом флаг-бита $2 + \frac{1}{k}$ байта, длина кода несжатого символа — $1 + \frac{1}{k}$ $\Longrightarrow$ $L_{\min} = 2$ (ссылка на двухбайтовое слово занимает $2 + \frac{1}{k}$ байта, два несжатых байта $2 + \frac{2}{k}$).
- возможен код со смещением: $\widetilde{S} = S - S_{\min} = S - 1, \quad \widetilde{L} = L - L_{\min} = L - 2.$

В наихудшем случае к файлу из n байтов добавляется $\frac{n}{k}$ флаг-байтов.

Примем $k = 3$, $|L| : |S| = 2 : 4$ $\Longrightarrow$ $\begin{cases} 0 \leqslant S - 1 \leqslant 2^4 - 1 \\ 0 \leqslant L - 2 \leqslant 2^2 - 1 \end{cases}$ $\Longrightarrow$ $\begin{cases} 1 \leqslant S \leqslant 2^4 \\ 2 \leqslant L \leqslant 2^2 + 1 \end{cases}$ $\Longrightarrow$ $\begin{cases} S_{\max} = 16 \\ L_{\max} = 5 \end{cases}$

$m = 7777\,7000\,0000\,0000\,0123\,4567\,0123\,4343\,7012\,1077$ (40 трёхбитных байтов)

$$7 \begin{Bmatrix} S=1 \\ L=4 \\ 7777 \end{Bmatrix} 0 \begin{Bmatrix} S=1 \\ L=5 \\ 00000 \end{Bmatrix} \begin{Bmatrix} S=1 \\ L=5 \\ 00000 \end{Bmatrix} 01234567 \begin{Bmatrix} S=8 \\ L=5 \\ 01234 \end{Bmatrix} \begin{Bmatrix} S=2 \\ L=3 \\ 343 \end{Bmatrix} \begin{Bmatrix} S=9 \\ L=4 \\ 7012 \end{Bmatrix} 1077$$

LZ77 с флаг-байтами ссылка/символ [LZ77-сс] (продолжение): ссылка/символ $1/0$

$$7\begin{Bmatrix}S=1\\L=4\\7777\end{Bmatrix}0\begin{Bmatrix}S=1\\L=5\\00000\end{Bmatrix}\begin{Bmatrix}S=1\\L=5\\00000\end{Bmatrix}01234567\begin{Bmatrix}S=8\\L=5\\01234\end{Bmatrix}\begin{Bmatrix}S=2\\L=3\\343\end{Bmatrix}\begin{Bmatrix}S=9\\L=4\\7012\end{Bmatrix}1077$$

группировка по $k=3$ объекта (дополняем до $3x$ нулём); код со смещением $\widetilde{S}=S-1,\quad \widetilde{L}=L-2$:

$$\big(010\big)7\begin{Bmatrix}1-1\\4-2\end{Bmatrix}0\big(110\big)\begin{Bmatrix}1-1\\5-2\end{Bmatrix}\begin{Bmatrix}1-1\\5-2\end{Bmatrix}0\big(000\big)123\big(000\big)456\big(001\big)7\begin{Bmatrix}8-1\\5-2\end{Bmatrix}\begin{Bmatrix}2-1\\3-2\end{Bmatrix}\big(100\big)\begin{Bmatrix}9-1\\4-2\end{Bmatrix}10\big(000\big)770$$

7 флаг-байтов + 6 двухбайтовых ссылок + 15 несжатых символов — всего $7+6\cdot 2+15=34$ байта

$$27\begin{Bmatrix}0\\2\end{Bmatrix}06\begin{Bmatrix}0\\3\end{Bmatrix}\begin{Bmatrix}0\\3\end{Bmatrix}00123045617\begin{Bmatrix}7\\3\end{Bmatrix}\begin{Bmatrix}1\\1\end{Bmatrix}4\begin{Bmatrix}8\\2\end{Bmatrix}100770$$

$$27\begin{Bmatrix}0000\\10\end{Bmatrix}06\begin{Bmatrix}0000\\11\end{Bmatrix}\begin{Bmatrix}0000\\11\end{Bmatrix}00123045617\begin{Bmatrix}0111\\11\end{Bmatrix}\begin{Bmatrix}0001\\01\end{Bmatrix}4\begin{Bmatrix}1000\\10\end{Bmatrix}100770$$

старший бит $\widetilde{S}$ записывается в байт, хранящий $\widetilde{L}$:

$$27\begin{Bmatrix}000\\010\end{Bmatrix}06\begin{Bmatrix}000\\011\end{Bmatrix}\begin{Bmatrix}000\\011\end{Bmatrix}00123045617\begin{Bmatrix}111\\011\end{Bmatrix}\begin{Bmatrix}001\\001\end{Bmatrix}4\begin{Bmatrix}000\\110\end{Bmatrix}100770$$

$\begin{Bmatrix}S\\L\end{Bmatrix}$ записывается в порядке (S,L): $code(m)=27020603030012304561773114061100770$ (34 трёхбитных байта)

LZ77 с флаг-битами ссылка/цепочка [LZ77-сц, предложен студентом МИЭТ]

Несжатые символы объединяются в цепочки аналогично RLE с флаг-битом.

- флаг-бит θ ссылка/цепочка может быть $0/1$ или $1/0$; ниже рассматривается $1/0$;
- θ помещается в байт L; L или S на один бит короче (разрядность L цепочки может отличаться от L ссылки);
- порядок (L, S);
- для ссылки: аналогично RLE $\inf(L_{\min}) \in \{2, 3\}$ — возьмём $L_{\min} = 3$;
- для цепочки несжатых символов: $L_{\min}^{\text{несж}} = 1$ по определению;
- возможен код со смещением: $\widetilde{S} = S - S_{\min} = S - 1, \quad \widetilde{L} = L - L_{\min} = L - 3;$ $\widetilde{L^{\text{несж}}} = L^{\text{несж}} - 1 \implies L_{\max}^{\text{несж}} = 2^{k-1}$.

В наихудшем случае к файлу из n байтов добавляется $\frac{n}{L_{\max}^{\text{несж}}} = \frac{n}{2^{k-1}}$ дополнительных байтов (при $k > 3$ — меньше, чем для флаг-байтов ссылка/символ).

Примем $k = 3$:

для ссылок $\begin{Bmatrix} S \\ L \end{Bmatrix}$ $|\theta| : |L| : |S| = 1 : 2 : 3$, $\begin{cases} 0 \leqslant S - 1 \leqslant 2^3 - 1 \\ 0 \leqslant L - 3 \leqslant 2^2 - 1 \end{cases} \implies \begin{cases} 1 \leqslant S \leqslant 2^3 \\ 3 \leqslant L \leqslant 2^2 + 2 \end{cases} \implies \begin{cases} S_{\max} = 8 \\ L_{\max} = 6 \end{cases}$

для цепочек несжатых символов $|\theta| : |L| = 1 : 2, \quad L_{\max}^{\text{несж}} = 4.$

LZ77 с флаг-битами ссылка/цепочка [LZ77-сц, предложен студентом МИЭТ] (продолжение): ссылка/цепочка $1/0$

$m = 7777\,7000\,0000\,0000\,0123\,4567\,0123\,4343\,7012\,1077$ (40 трёхбитных байтов)

ссылки ($\theta = 1$): $1 \leqslant S \leqslant 8$, $3 \leqslant L \leqslant 6$; цепочки несжатых символов ($\theta = 0$): $1 \leqslant L \leqslant 4$:

$$\left\{\begin{matrix}0\\L=1\\7\end{matrix}\right\}\left\{\begin{matrix}1\\S=1\\L=4\\7777\end{matrix}\right\}\left\{\begin{matrix}0\\L=1\\0\end{matrix}\right\}\left\{\begin{matrix}1\\S=1\\L=6\\000000\end{matrix}\right\}\left\{\begin{matrix}1\\S=1\\L=5\\00000\end{matrix}\right\}\left\{\begin{matrix}0\\L=4\\1234\end{matrix}\right\}\left\{\begin{matrix}0\\L=3\\567\end{matrix}\right\}\left\{\begin{matrix}1\\S=8\\L=5\\01234\end{matrix}\right\}\left\{\begin{matrix}1\\S=2\\L=3\\343\end{matrix}\right\}\left\{\begin{matrix}0\\L=4\\7012\end{matrix}\right\}\left\{\begin{matrix}0\\L=4\\1077\end{matrix}\right\}$$

смещение для ссылки: $\widetilde{S} = S - 1$, $\widetilde{L} = L - 3$; для цепочки $\widetilde{L} = L - 1$:

$$\Big((0, 1-1=0), 7\Big)\Big((1, 4-3=1), 1-1=0\Big)\Big((0, 1-1=0), 0\Big)\Big((1, 6-3=3), 1-1=0\Big)\Big((1, 5-3=2), 1-1=0\Big)$$
$$\Big((0, 4-1=3), 1234\Big)\Big((0, 3-1=2), 567\Big)\Big((1, 5-3=2), 8-1=7\Big)\Big((1, 3-3=0), 2-1=1\Big)$$
$$\Big((0, 4-1=3), 7012\Big)\Big((0, 4-1=3), 1077\Big)$$

$$(000, 7)(101, 0)(000, 0)(111, 0)(110, 0)(011, 1234)(010, 567)(110, 7)(100, 1)(011, 7012)(011, 1077)$$

$code(m) = 07500070603123425676741370123 1077$ (33 трёхбитных байта)

LZ77 с односимвольным префиксом [LZ77-p]

- $\left\{\begin{matrix} S \\ L \end{matrix}\right\}$ записывается как $\left(p, \left\{\begin{matrix} S \\ L \end{matrix}\right\}\right)$; байт p самый редкий, $p(p) \leqslant \frac{1}{2^k}$; значение p сохраняется в заголовке файла;
- $L_{\min} = 4$;
- байт p в несжатом тексте экранируется только как $(p, 0) \implies$
  $\implies$ в $\left(p, \left\{\begin{matrix} S \\ L \end{matrix}\right\}\right)$ второй байт $\neq 0 \implies$ порядок $\left(p, \widetilde{L}, \widetilde{S}\right)$ с $\widetilde{L} \neq 0 \implies$
  $\implies$ смещение не на $L_{\min} = 4$, а на $L_{\min} - 1 = 3$: $(p, L-3, S-1)$

В наихудшем случае к файлу из n байтов добавляется столько дополнительных байтов, сколько было символов p — не более $\frac{n}{2^k}$ (меньше, чем для флаг-битов).

LZ77 с односимвольным префиксом [LZ77-p]

Примем $k = 3$, $|L| : |S| = 2 : 4 \implies \begin{cases} 0 \leqslant S-1 \leqslant 2^4-1 \\ 1 \leqslant L-3 \leqslant 2^2-1 \end{cases} \implies \begin{cases} 1 \leqslant S \leqslant 2^4 \\ 4 \leqslant L \leqslant 2^2+2 \end{cases} \implies \begin{cases} S_{\max} = 16 \\ L_{\max} = 6 \end{cases}$

выбираем префикс $p = 5$

$m = 7777\,7000\,0000\,0000\,0123\,4567\,0123\,4343\,7012\,1077$ (40 трёхбитных байтов)

$7\left\{\begin{matrix} S=1 \\ L=4 \\ 7777 \end{matrix}\right\} 0\left\{\begin{matrix} S=1 \\ L=6 \\ 000000 \end{matrix}\right\}\left\{\begin{matrix} S=1 \\ L=5 \\ 00000 \end{matrix}\right\} 1234567\left\{\begin{matrix} S=8 \\ L=5 \\ 01234 \end{matrix}\right\} 343\left\{\begin{matrix} S=9 \\ L=4 \\ 7012 \end{matrix}\right\} 1077$

$7\left\{\begin{matrix} 1-1=0 \\ 4-3=1 \end{matrix}\right\} 0\left\{\begin{matrix} 1-1=0 \\ 6-3=3 \end{matrix}\right\}\left\{\begin{matrix} 1-1=0 \\ 5-3=2 \end{matrix}\right\} 1234\big(5,0\big)67\left\{\begin{matrix} 8-1=7 \\ 5-3=2 \end{matrix}\right\} 343\left\{\begin{matrix} 9-1=8 \\ 4-3=1 \end{matrix}\right\} 1077$

$7\left\{\begin{matrix} 0000 \\ 01 \end{matrix}\right\} 0\left\{\begin{matrix} 0000 \\ 11 \end{matrix}\right\}\left\{\begin{matrix} 0000 \\ 10 \end{matrix}\right\} 1234\big(5,0\big)67\left\{\begin{matrix} 0111 \\ 10 \end{matrix}\right\} 343\left\{\begin{matrix} 1000 \\ 01 \end{matrix}\right\} 1077$

$7\left\{\begin{matrix} 000 \\ 001 \end{matrix}\right\} 0\left\{\begin{matrix} 000 \\ 011 \end{matrix}\right\}\left\{\begin{matrix} 000 \\ 010 \end{matrix}\right\} 1234\big(5,0\big)67\left\{\begin{matrix} 111 \\ 010 \end{matrix}\right\} 343\left\{\begin{matrix} 000 \\ 101 \end{matrix}\right\} 1077$

$code(m) = 7510053052012345067527343540 1077$ (33 трёхбитных байта)

Лучший и худший случаи LZ77 (для октетов: $k = 8$)

Лучший исходный файл — $m_л = \underbrace{0000...0000}_{n}$; худший — ни в одном в окне нет ни одного совпадения $L \geqslant L_{\min}$

Считаем $n \gg L_{\max}$; всё оцениваем приблизительно; в частности, для LZ77-сц $L_{\max}^{\text{несж}} \approx L_{\max}$.
Первые две колонки $|code(m_л)|$ и $|code(m_х)|$ — для любого $L_{\max}$ (то есть любых $|L|$ и $|S|$), далее $|L| = |S| = k$.

$code$	$\lvert code(m_л)\rvert$	$\lvert code(m_х)\rvert$	$L_{\max}^{\lvert L\rvert=\lvert S\rvert=k}$	$L_{\max}^{\lvert L\rvert=\lvert S\rvert=k}\big\rvert_{k=8}$	$\lvert code(m_л)\rvert$	$\lvert code(m_х)\rvert$
LZ77-к	$3 \cdot \frac{n}{L_{\max}+1}$	$3 \cdot n$	$2^k - 1$	255	$3 \cdot \frac{n}{256}$	$3 \cdot n$
LZ77-сс	$(2 + \frac{1}{k}) \cdot \frac{n}{L_{\max}}$	$(1 + \frac{1}{k}) \cdot n$	$2^k + 1$	257	$(2 + \frac{1}{8}) \cdot \frac{n}{257}$	$(1 + \frac{1}{8}) \cdot n$
LZ77-сц	$2 \cdot \frac{n}{L_{\max}}$	$(1 + \frac{1}{L_{\max}}) \cdot n$	$\approx 2^{k-1}$	130 (128 несж)	$\frac{2n}{130} = 4 \cdot \frac{n}{260}$	$(1 + \frac{1}{128}) \cdot n$
LZ77-p	$3 \cdot \frac{n}{L_{\max}}$	$n + count(p) \leqslant n \cdot (1 + \frac{1}{2^k})$	$2^k + 2$	258	$3 \cdot \frac{n}{258}$	$(1 + \frac{1}{256}) \cdot n$

При $|L| = k - 1$ и $|S| = k + 1$: а) $L_{\max}$ примерно вдвое меньше; б) худший случай для LZ77-к не реализуется (окно больше алфавита и $L_{\min} = 1$). При $|L| = k - 2$ и $|S| = k + 2$: $L_{\max}$ вчетверо меньше, а окно больше...

LZ77-сс лучший на $m_л$; используется в LZJB с $|L| = 6$ и $|S| = 10$ бит и изб. данными — худший случай редок.

LZ77-p лучший на $m_х$. Вопрос: всегда ли LZ77-p лучше или равноценен LZ77-к?

Сравнение с RLE

Как RLE, так и LZ77 не требуют отдельного словаря.

Сравниваем реализации с префиксом для $k = 3$: $p = 5$, $RLE: \begin{cases} 4 \leqslant L^{c \neq p} \leqslant 11, \\ 1 \leqslant L^{p} \leqslant 8; \end{cases}$ $LZ77: \begin{cases} 1 \leqslant S \leqslant 16, \\ 4 \leqslant L \leqslant 6. \end{cases}$

1. Одиночная цепочка повторений длины $L \leqslant L_{\max}^{LZ77}$:
   RLE: $L \to 3$ $\quad 77777 \to 527$
   LZ77: $L \to 4$ $\quad 77777 \to 7510$

2. Есть цепочки повторений длины порядка $L_{\max}^{RLE}$:
   RLE: $7777700000000000012345670123434370121077 \to 52757000123450670123434370121077 \quad 40 \to 33$
   LZ77: $7777700000000000012345670123434370121077 \to 751005305201234506752734354 01077 \quad 40 \to 33$

3. Длина повторений не более $L_{\max}^{LZ77}$:
   RLE: $7777700000000123456701234343701210 77 \to 527530001234506701234343701210 77 \quad 36 \to 32$
   LZ77: $7777700000000123456701234343701210 77 \to 75100530012345067527343540 1077 \quad 36 \to 31$

Для большинства файлов LZ77 эффективнее RLE аналогичной реализации; существуют файлы, для которых лучшим будет RLE; для некоторых (модель наивного RLE) лучшим будет наивный RLE.

Модель источника для LZ77

LZ77 строится из предположения, что модель марковская.

Задача: каковы характеристики модели для оценки снизу длины кода алгоритмов семейства LZ77:

- стационарна ли она?
- какова глубина памяти такой модели?

Задача №1

Закодируйте различными реализациями RLE/LZ77 сообщение

$$m = 7770\,0000\,0123\,4567\;\;7770\,7770\,0000\,0000\;\;0011\,2233\,4455\,6677$$

исходная длина m в символах $n = 4 \cdot 12 = 48$ символов

1. в байте $k = 3$ бита;
2. символ кодирования — 3-битный байт $(0 - 7)$; сгруппированы по 4 и 16 штук для удобства.

Задача №2

1. Рассчитать количество информации в сообщении $C =$ «$\underbrace{\text{ля-ля-ля-...-ля}}_{\text{80 раз}}$» (кодировка koi8-r)
2. Закодировать сообщение C алгоритмом LZ77

Спасибо за внимание!

МИЭТ
www.miet.ru

Александра Игоревна Кононова
illinc@mail.ru
gitlab.com/illinc/raspisanie

Информация и модели источника

Заменяя вероятности символов на их оценки по модели X, получаем **оценку** количества информации:

1. Без памяти, $A_1 = \{\text{л}, \text{я}, \text{-}\}$:
   $I_1 = 2 \cdot 80 \cdot \left(-\log_2 \left(\frac{80}{239} \right) \right) + 79 \cdot \left(-\log_2 \left(\frac{79}{239} \right) \right) = 254{,}2 \text{ бит} = 31{,}8 \text{ байт}$
2. Без памяти, $A_1 = \{\text{ля-}, \text{ля}\}$:
   $I_2 = 79 \cdot \left(-\log_2 \left(\frac{79}{80} \right) \right) + 1 \cdot \left(-\log_2 \left(\frac{1}{80} \right) \right) = 7{,}6 \text{ бит} = 0{,}9 \text{ байт}$
3. Маркова, $A_1 = \text{koi8-r}, N = 3$ со следующими оценками вероятностей:
   $p(c_1 = \text{л}) = p(c_2 = \text{я}) = p(c_3 = \text{-}) = \frac{1}{256}$, $p(\text{-}|\text{-ля}) = \frac{79}{80}$,
   $p(\text{л}|\text{ля-}) = p(\text{я}|\text{я-л}) = 1$, $p(eof|\text{-ля}) = \frac{1}{80}$:

$$I_3 = 3 \cdot \log_2 256 + 79 \cdot \left(-\log_2 \left(\frac{79}{80} \right) \right) + 1 \cdot \left(-\log_2 \left(\frac{1}{80} \right) \right) = 31{,}6 \text{ бит} = 3{,}9 \text{ байт}$$