

# Арифметическое сжатие

**Напоминание:** Вероятностные методы просты, но эффективны только в том случае, когда вероятности появления символов равны числам  $(1/2)^n$  ( $n$  - любое целое положительное число) – каждому символу алфавита присваивается код с целым числом бит.

**Эту проблему решают арифметические методы сжатия.**

**Идея: Сжимаемая последовательность представляется одним числом (как правило – дробью).**

**Процедура:**

- **последовательно обрабатывается очередной символ сжимаемого текста – на каждом шаге значение дроби уточняется с учетом вероятностного параметра анализируемого символа (нормализация);**
- **последовательно восстанавливается каждый символ сжатой последовательности с учетом последнего (конечного) значения дроби и вероятностного параметра извлекаемого символа**

## Границы интервала

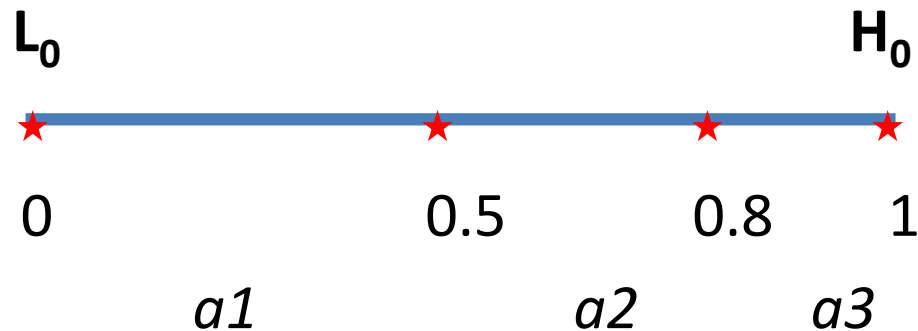
**Пример 1.** Рассмотрим вычисление дроби на интервале **(0, 1)**.  
Интервал  $[0, 1)$  выбран потому, что он удобен для объяснений.  
Разбиваем его на подынтервалы с длинами, равными вероятностям (частотам) появления символов в потоке -

диапазоны соответствующих символов  **$a_i$** . Пусть  $n=3$ .

$$P(a1) = 0.5$$

$$P(a2) = 0.3$$

$$P(a3) = 0.2$$



0. Процесс кодирования символов текста начинается со считывания первого символа входного потока (**шаг 0**) и **присвоения ему своего интервала из начального диапазона (0, 1)**. Этот интервал становится рабочим для следующего этапа – т.е. **происходит сокращение исходного интервала**.
1. Считывается второй символ (**шаг 1**) и **присваивается ему свой интервала из диапазона шага 0**. Этот интервал становится рабочим для следующего этапа – т.е. **опять происходит сокращение исходного интервала (нормализация)**.
- И т.д. Результат – последняя дробь (левая граница интервала)**.

На каждом шаге  $i$  прямого преобразования происходит перерасчет границ интервала:

- $H_i = L_{i-1} + (H_{i-1} - L_{i-1}) * H(a_i)_0$
- $L_i = L_{i-1} + (H_{i-1} - L_{i-1}) * L(a_i)_0$

**(1)**

На каждом шаге  $i$  обратного пре-я:

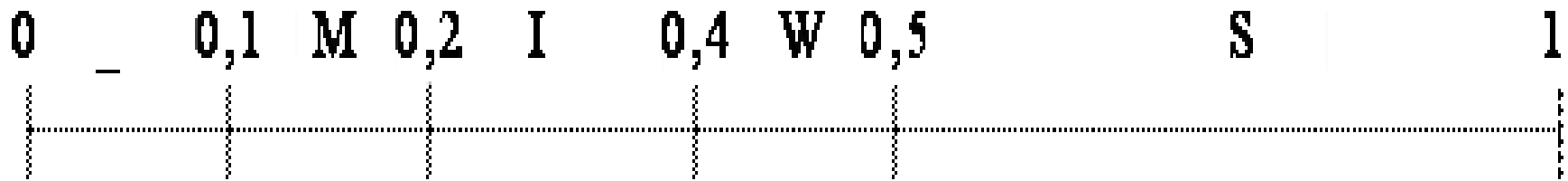
- $\text{код } i = [\text{код } (i-1) - L(a_{i-1})_0] / [H(a_{i-1})_0 - L(a_{i-1})_0]$

**(2)**

**Пример 2.** Известно распределение:  $S - 0,5$ ,  $W - 0,1$ ,  $I - 0,2$ ,  $M - 0,1$ , «-»  $0,1$ .

Входное сообщение: «**SWISS\_MISS**»

Получим следующие наборы интервалов:



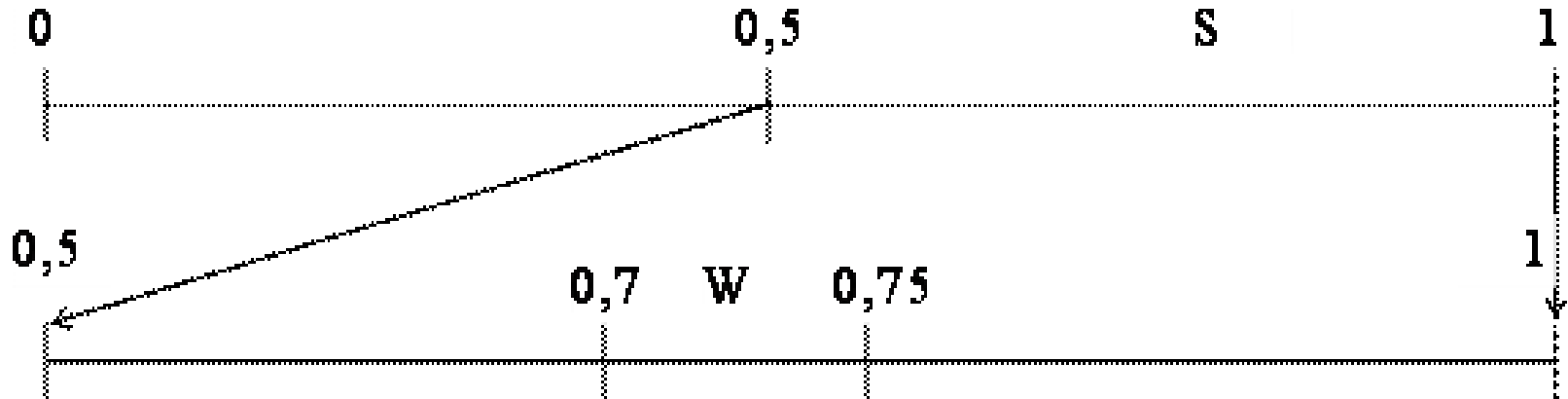
**Шаг 1:** Процесс кодирования начинается со считывания первого символа входного потока и присвоения ему интервала из начального интервала  $(0, 1)$ . В данном случае для первого символа **S** получаем интервал  $(0,5; 1)$ .

**Шаг 2:** Считывается второй символ – **W**, которому соответствует диапазон  $(L(w)_0 = 0,4; H(w)_0 = 0,5)$ . Но исходный диапазон  $(0, 1)$  уже сократился до  $(0,5, 1)$ , поэтому символ **W** необходимо представить в этом новом диапазоне (нормализовать): **вычислить новые нижнюю и верхнюю границы.**

Вычисляем новые нижнюю и верхнюю границы (в соотв. с (1):

- $H_2 = L_1 + (H_1 - L_1) * H(w)_0 = 0.5 + (1 - 0.5) * 0.5 = 0.75$
- $L_2 = L_1 + (H_1 - L_1) * L(w)_0 = 0.5 + (1 - 0.5) * 0.4 = 0.7$ .

Т.о. :значение 0,4 будет соответствовать значению 0,7, а значение 0,5 – значению 0,75



**Шаг 3.** Кодировается символ «I», для которого  $L(I)_0 = 0.2$ ;  $H(I)_0 = 0.4$ :

$$H_3 = L_2 + (H_2 - L_2) * H(I)_0 = 0.7 + (0.75 - 0.7) * 0.4 = 0.72$$

$$L_3 = L_2 + (H_2 - L_2) * L(I)_0 = 0.7 + (0.75 - 0.7) * 0.2 = 0.71$$

В табл. представлены значения границ при кодировании строки  
**SWISS\_MISS**

Символ		Границы
S	L	$0.0+(1.0-0.0)*0.5 = 0.5$
	H	$0.0+(1.0-0.0)*1.0 = 1.0$
W	L	$0.5+(1.0-0.5)*0.4=0.70$
	H	$0.5+(1.0-0.5)*0.5=0.75$
I	L	$0.7+(0.75-0.7)*0.2=0.71$
	H	$0.7+(0.75-0.7)*0.4=0.72$
S	L	$0.71+(0.72-0.71)*0.5=0.715$
	H	$0.71+(0.72-0.71)*1.0=0.72$
S	L	$0.715+(0.72-0.715)*0.5=0.7175$
	H	$0.715+(0.72-0.715)*1.0=0.72$
—	L	$0.7175+(0.72-0.7175)*0.0=0.7175$
	H	$0.7175+(0.72-0.7175)*0.1=0.71775$
M	L	$0.7175+(0.71775-0.7175)*0.1=0.717525$
	H	$0.7175+(0.71775-0.7175)*0.2=0.717550$
I	L	$0.717525+(0.717550-0.717525)*0.4=0.717530$
	H	$0.717525+(0.717550-0.717525)*0.5=0.717535$
S	L	$0.717530+(0.717535-0.717530)*0.5=0.7175325$
	H	$0.717530+(0.717535-0.717530)*1.0=0.717535$
S	L	$0.7175325+(0.717535-0.7175325)*0.5=0.71753375$
	H	$0.7175325+(0.717535-0.7175325)*1.0=0.717535$

- Выходной код (сжатое сообщение) – это последнее значение переменной  $L = 0.71753375$ .
- Результат кодирования – это вещественное число с очень большой точностью.
- Нп, файл объемом 1Мб сжат до 500 Кб, в котором будет записано одно число. Арифметические операции с такими числами реализовать сложно и долго. Поэтому **практическая реализация арифметического кодера должна основываться на операциях с целыми числами, которые не должны быть слишком длинными**.

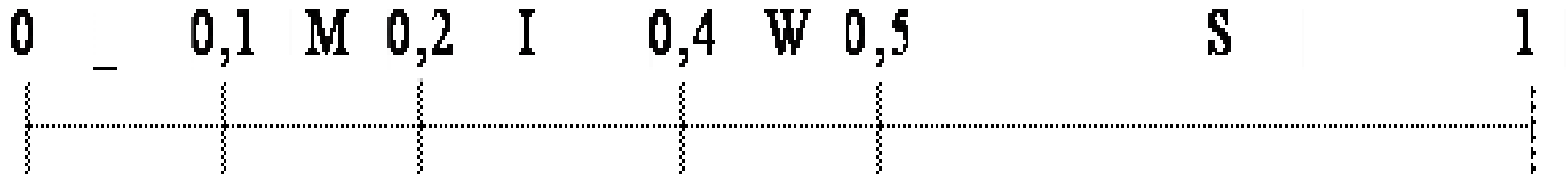


# Обратное преобразование

- На входе:

Статистика:  $p(S) = 0,5$ ;  $p(W) = 0,1$ ;  $p(I) = 0,2$ ;  $p(M) = 0,1$ ;  $p(\_) = 0,1$

Шкала (интервалы)



Число: **0.71753375**

- Преобразование

Шаг 1. Число **0.71753375** принадлежит инт-лу (0.5-1)  $\rightarrow S$

Шаг 2. Интервал (0.5-1) следует преобразовать в стандартный (0-1) и вычислить код второго символа (код 2; в соотв. с (2)):

$$\begin{aligned} \text{код 2} &= [\text{код 1} - L(S)_0] / [H(S)_0 - L(S)_0] = [0.71753375 - 0.5] / [1 - 0/5] = \\ &= 0.4350675 \rightarrow W \end{aligned}$$

Шаг 3. Выполняются вычисления по вышеуказ-му алгоритму:

$$\text{код } 3 = [\text{код } 2 - L(W)_0] / [H(W)_0 - L(W)_0] = [4350675 - 0.4] / [0.5 - 0.4] = 0.350675 \rightarrow I,$$

Т.е.

$$\text{код } i = [\text{код } (i-1) - L(a_{i-1})_0] / [H(a_{i-1})_0 - L(a_{i-1})_0]$$

Символ	код i-1	код i		
S	<b>0.71753375</b> – 0.5	= 0.21753375	/ 0.5 = <b>0.4350675</b>	(w)
W	<b>0.4350675</b> – 0.4	= 0.0350675	/ 0.1 = <b>0.350675</b>	(I)
I	<b>0.350675</b> – 0.2	= 0.150675	/ 0.2 = <b>0.753375</b>	(S)
S	<b>0.753375</b> – 0.5	= 0.253375	/ 0.5 = <b>0.50675</b>	(S)
S	<b>0.50675</b> – 0.5	= 0.00675	/ 0.5 = <b>0.0135</b>	(-)
–	<b>0.0135</b> – 0	= 0.0135	/ 0.1 = <b>0.135</b>	(M)
M	<b>0.135</b> – 0.1	= 0.035	/ 0.1 = <b>0.35</b>	(I)
I	<b>0.35</b> – 0.2	= 0.15	/ 0.2 = <b>0.75</b>	(S)
S	<b>0.75</b> – 0.5	= 0.25	/ 0.5 = <b>0.5</b>	(S)
S	<b>0.5</b> – 0.5	= 0	/ 0.5 = 0	