

АЛГОРИТМ LZW

Виконали студенти групи І-34

Віятник Христина

Шимків Назар

ОПИС ПРОЦЕСУ

Процес стиснення виглядає наступним чином. Послідовно зчитуються символи вхідного потоку і відбувається перевірка, чи існує в створеній таблиці рядків такий рядок. Якщо такий рядок існує, зчитується наступний символ, а якщо рядок не існує, в потік заноситься код для попередньої знайденої рядка, рядок заноситься в таблицю, а пошук починається знову.

ПСЕВДОКОД АЛГОРИТМУ

Ініціалізація словника усіма можливими односимвольними фразами.
Ініціалізація вхідної фрази ω першим символом повідомлення.

Зчитати наступний символ K з кодованого повідомлення.

Якщо кінець повідомлення, то видати код для ω , інакше:

Якщо фраза $\omega (K)$ вже є в словнику, привласнити вхідній фразі значення $\omega (K)$ і перейти до Кроку 2, інакше видати код ω , додати $\omega (K)$ в словник, привласнити вхідній фразі значення K і перейти до Кроку 2.

кінець

LZ78 орієнтується на дані, які тільки будуть отримані (LZ78 не використовує ковзне вікно, він зберігає словник з вже переглянутих фраз). Алгоритм зчитує символи повідомлення доти, поки що накопичується подстрока входить цілком в одну з фраз словника. Як тільки цей рядок перестане відповідати хоча б одній фразі словника, алгоритм генерує код, що складається з індексу рядка в словнику, яка до останнього введеного символу містила вхідну рядок, і символу, що порушив збіг. Потім в словник додається введена подстрока. Якщо словник вже заповнений, то з нього попередньо видаляють менш всіх використовувану в порівняннях фразу. Якщо наприкінці алгоритму ми не знаходимо символ, який порушив збіги, то тоді ми видаємо код у вигляді (індекс рядка в словнику без останнього символу, останній символ).

| Содержимое окна | Содержимое буфера | КОД |
|---|-------------------|-------------------------|
| <i>kabababababz</i> | <i>k</i> | $\langle 0,0,k \rangle$ |
| <i>kabababababz</i> | <i>a</i> | $\langle 0,0,a \rangle$ |
| <i>kabababababz</i> | <i>b</i> | $\langle 0,0,b \rangle$ |
| <i>k</i> <i>ab</i> <i>ababababz</i> | <i>aba</i> | $\langle 2,2,a \rangle$ |
| <i>kaba</i> <i>babab</i> <i>abz</i> | <i>bababz</i> | $\langle 2,5,z \rangle$ |

| Текущая строка | Текущий символ | Следующий символ | Вывод | | Словарь |
|----------------|----------------|------------------|-------|------|----------|
| | | | Код | Биты | |
| ab | a | b | 0 | 000 | 5: ab |
| ba | b | a | 1 | 001 | 6: ba |
| ac | a | c | 0 | 000 | 7: ac |
| ca | c | a | 2 | 010 | 8: ca |
| ab | a | b | - | - | - - |
| aba | b | a | 5 | 101 | 9: aba |
| ad | a | d | 0 | 000 | 10: ad |
| da | d | a | 3 | 011 | 11: da |
| ab | a | b | - | - | - - |
| aba | b | a | - | - | - - |
| abac | a | c | 9 | 1001 | 12: abac |
| ca | c | a | - | - | - - |
| cab | a | b | 8 | 1000 | 13: cab |
| ba | b | a | - | - | - - |
| bae | a | e | 6 | 0110 | 14: bae |
| e | e | - | 4 | 0100 | - - |

| Содержимое словаря | Содержимое считываемой строки | КОД |
|---|-------------------------------|-----------------------|
| | <i>k</i> | $\langle 0,k \rangle$ |
| <i>k</i> | <i>a</i> | $\langle 0,a \rangle$ |
| <i>k, a</i> | <i>b</i> | $\langle 0,b \rangle$ |
| <i>k, </i> <i>a</i> <i>, b</i> | <i>ab</i> | $\langle 2,b \rangle$ |
| <i>k, a, b, </i> <i>ab</i> | <i>aba</i> | $\langle 4,a \rangle$ |
| <i>k, a, </i> <i>b</i> <i>, ab, aba</i> | <i>ba</i> | $\langle 3,a \rangle$ |
| <i>k, a, b, ab, </i> <i>aba</i> <i>, ba</i> | <i>abab</i> | $\langle 5,b \rangle$ |
| <i>k, a, b, ab, aba, ba, abab</i> | <i>z</i> | $\langle 0,z \rangle$ |

ДЕКОДУВАННЯ

Особливість LZW полягає в тому, що для декомпресії нам не треба зберігати таблицю рядків у файл для розпакування. Алгоритм побудований таким чином, що ми в змозі відновити таблицю рядків, користуючись тільки потоком кодів.

Тепер уявімо, що ми отримали закодоване повідомлення, наведене вище, і нам потрібно його декодувати. Перш за все, нам потрібно знати початковий словник, а наступні записи словника ми можемо реконструювати вже на ходу, оскільки вони є просто конкатенацією попередніх записів.