Лабораторна робота №5

Метод LZ78.

Відомі методи оптимального кодування кодами різної довжини, які базуються на врахуванні різної частоти символів в тексті. Найбільш відомі з них: метод Шеннона-Фано та метод Хаффмана. Ці методи забезпечують достатньо ефективне стиснення без втрат. Для підвищення рівня стиснення даних доцільно перейти від оптимального кодування окремих символів до кодування буквосполучень.

На відміну від LZ77, що працює з уже отриманими даними, LZ78 орієнтується на дані, які тільки будуть отримані (LZ78 не використовує ковзне вікно, він зберігає словник з вже переглянутих фраз). Алгоритм зчитує символи повідомлення до тих пір, поки підстрічка, яка накопичується входить цілком в одну з фраз словника. Як тільки ця підстрічка перестане відповідати хоча б одній фразі словника, алгоритм генерує код, що складається з індексу рядка в словнику, який до останнього введеного символу містив вхідну підстрічку, і символ, який порушив збіг. Потім в словник додається введена підстрічка.

Більш строго, словник починає будуватися з пустого рядка в позиції нуль. У міру надходження і кодування символів, нові рядки додаються в позиції 1, 2 ... Коли наступний символ X читається з вхідного файлу, в словнику шукається рядок з одного символу X. Якщо такого рядка немає, то X додається в словник, а на вихід подається мітка (0,X). Ця мітка означає рядок «нуль X». Якщо входження символу X виявлено (скажімо, в позиції 48), то читається наступний символ Y, і в словнику шукається входження двосимвольних рядків XY. Якщо такого не знайдено, то в словник записується рядок XY, а на вихід подається мітка (48,Y). Така мітка означає рядок XY, так як позицію 48 в словнику займає символ X . Процес продовжується до кінця вхідного файлу.

У загальному випадку поточний символ читається і стає односимвольним рядком. Потім кодер намагається знайти його в словнику. Якщо рядок знайдено, читається наступний символ і приєднується до поточного рядка, утворюючи двосимвольний рядок, який кодер знову намагається знайти в словнику. До тих пір поки такі рядки є в словнику, відбувається читання нових символів і їх приєднання до поточного рядка. В деякий момент такого рядка в словнику не знайдеться. Тоді кодер додає його в словник і будує мітку, в першому полі якої стоїть вказівник на останню знайдену в словнику рядок, а в другому полі записаний останній символ рядка (на якому стався обрив успішних пошуків). У таблиці 1 показані кроки при декодуванні послідовності «sir_sid_eastman_easily_teases_sea_sick_seals».

На кожному кроці рядок, доданий в словник, збігається з кодованим рядком мінус останній символ. У типовому процесі стиснення словник починається з коротких рядків, але в міру просування по кодованому тексту, все більше і більше довгі рядки додаються в словник. Розмір словника може бути фіксованим або визначатися розміром доступної пам'яті. Великий словник дозволяє робити глибокий пошук довгих збігів, але ціною цього служить

довжина поля покажчиків (а, значить, і довжина мітки) і уповільнення процесу словникового пошуку.

Таблиця 1. Кроки кодування LZ78.

Словник	Мітка	Словник	Мітка
0 null			
1 «s»	(0,«s»)	14 «y»	(0, «y»)
2 «i»	(0,«i»)	15 «_t»	(4,«t»)
3 «r»	(0,«r»)	16 «e»	(0,«e»)
4 «_»	(0,«_»)	17 «as»	(8,«s»)
5 «si»	(1,«i»)	18 «es»	(16,«s»)
6 «d»	(0,«d»)	19 «_s»	(4,«s»)
7 «_e»	(4, «e»)	20 «ea»	(4,«a»)
8 «a»	(0,«a»)	21 «_si»	(19,«i»)
9 «st»	(l,«t»)	22 «c»	(0,«c»)
10 «m»	(0,«m»)	23 «k»	(0,«k»)
11 «an»	(8,«n»)	24 «_se»	(19,«e»)
12 «_ea»	(7,«a»)	25 «al»	(8,«l»)
13 «sil»	(5, «l»)	26 «s(eof)»	(l,«(eof)»)

Завдання роботи.

1. Реалізувати програмно алгоритм кодування послідовностей символів на основі методу LZ78.
2. Здійснити кодування:
   а) власного прізвища записаного два рази без пропуску.
   б) довільного фрагмента тексту довжиною від 20 до 30 слів.
   в) випадково згенерованого набору символів у вигляді 20- 30 слів.

Зміст звіту.

1. Текст програми.
2. Вхідна послідовність.
3. Кодовий словник (у вигляді **таблиці 1**). Обчислити середню довжину кодової фрази.
4. Кодована послідовність.
5. Дерево словника (завдання **2.а,б**).