Броновицький А.Р.

До питання про методи збереження інформації (архівація даних).

Архівація зменшує об'єм пам'яті чи диска, потрібного для збереження файлів в ЕОМ, а також кількість часу, необхідного для передачі інформації по каналах встановленої ширини пропускання. Це є форма кодування. Іншими цілями кодування є пошук та виправлення помилок, а також шифрування. Процес пошуку і виправлення помилок протилежний архівації - він збільшує надлишковість даних, коли їх не потрібно подавати у зручній для сприйняття людиною формі. Видаляючи з тексту надлишковість, архівація сприяє шифруванню, що ускладнює пошук шифру доступним статистичним методом.

У цій статті ми розглянемо зворотню архівацію або архівацію без наявності похибок, коли початковий текст може бути повністю відновлений зі стиснутого стану. Архівація із втратою даних використовується для цифрового запису аналогових сигналів, таких як звук, графіка чи відео. Зворотня архівація особливо потрібна для текстів, записаних на природних чи штучних мовах, оскільки в цьому випадку помилки неприпустимі.

Існує багато вагомих причин виділяти ресурси ЕОМ в розрахунку на їх стиснене представлення, тому що більш швидка передача даних і зменшення простору для їх збереження дозволяють зберегти значні кошти та поліпшити показники ЕОМ. Архівація даних, мабуть, буде залишатися у сфері уваги через всезростаючі обсяги інформації, що

зберігається та передається в ЕОМ, крім того, її можна використовувати для подолання деяких фізичних обмежень, таких як, наприклад, порівняно низька ширина пропускання телефонних каналів.

Одним із найбільш ранніх і добре відомих методів архівації є алгоритм Хоффмана (1), що був і залишається предметом більшості досліджень. Проте, наприкінці 70-х років дві нових ідеї поповнили список таких алгоритмів. Одна полягала у відкритті методу АРИФМЕТИЧНОГО КОДУВАННЯ (2,3,4,5,6, 7,8,9), що має схожу з кодуванням Хаффмана функцію, але також має декілька важливих властивостей, що дають можливість досягти значної переваги при архівації. Іншим нововведенням був метод Зіва-Лемпела (10,11), що дає ефективний стиск, але має підхід, цілком відмінний від хоффманівського й арифметичного. Обидві ці техніки з часу своєї першої публікації значно вдосконалилися, розвинулися і лягли в основу практичних високоефективних алгоритмів.

Існують два основних способи проведення стиснення: статистичний і словниковий. Кращі статистичні методи застосовують арифметичне кодування, кращі словникові - метод Зіва-Лемпела. У статистичному стиску кожному символу присвоюється код, що грунтується на ймовірності його появи в тексті. Ті символи, що частіше зустрічаються, отримують короткі коди, і навпаки. У словниковому методі групи послідовних символів або "фраз" замінюються кодом. Змінена фраза може бути знайдена в деякому "словнику". Тільки останнім часом було показано, що будь-яка практична схема словникового стиснення може бути зведена до відповідної статистичної схеми стиснення, і знайдено загальний алгоритм перетворення словникового методу у статистичний (12,13). Тому при відшукуванні кращого стиснення статистичне кодування обіцяє бути найбільше плідним, хоча словникові методи і привабливі своєю швидкістю.

На даний час існує багато практичних реалізацій цих алгоритмів. Одні з найбільш поширених архіваторів - це ZIP, RAR, ARJ. При створенні власної програми був використаний видозмінений префіксний алгоритм Хоффмана. Діюча експериментальна програма поки що давала такий результат: файли, заархівовані RAR, ZIP чи ARJ, архівувались нашою програмою, при цьому новостворений архів займав на 2-3% менші розміри. Але алгоритм, закладений в основу цієї програми, передбачає неоднократне стискання, тобто прогнозується порушити залишковість інформації і відповідно зменшити розміри стиснутого файла.

Література

1.Huffman D.A. 1952. A method for the construction of minimum redundancy codes.In Proceedings of the Institute of Electrical and Radio Engineers 40,9(Sept.),pp.1098-1101.

2.Guazzo M. 1980.A general minimum-redundancy source-coding algorithm.IEEE Trans.Inf.Theory IT-26, 1(Jan.),15-25.

3.Langdon G.G. 1984.An introduction to arithmetic coding.IBM J.Res.Dev. 28,2(Mar.),135-149.

4.Langdon G.G. and Rissanen J.J. 1982. A simple general binary source code.IEEE Trans.Inf.Theory IT-28 (Sept),800-803.

5.Pasco R. 1976. Source coding algorithms for fast data compression. Ph.D. dissertation.Dept.of Electrical Engineering, Stanford Univ.

6.Rissanen J.J. 1976. Generalized Kraft inequality and arithmetic coding.IBM J.Res.Dev.20,(May.),198-203.

7.Rissanen J.J. 1979. Arithmetic codings as number representations. Acta Polytechnic Scandinavica, Math 31(Dec.),44-51.

8.Rissanen J.J. and Langdon G.G. 1979. Arithmetic coding. IBM J.Res.Dev. 23,2 (Mar.), 149-162.Describes a broad class of arithmetic codes.

9.Rubin F. 1979. Arithmetic stream coding using fixed precision registers. IEEE Trans.Inf.Theory IT-25,6(Nov.),672-675

10.Ziv J. and Lempel A.1977.A universal algorithms for sequental data compression.IEEE Trans.Inf.Theory IT-23,3,3(May),337-343.

11.Ziv J. and Lempel A.1978.Compression of individual sequences via variable-rate coding. IEEE Trans.Inf.Theory IT-24,5(Sept.),530-536.

12.Bell T.C. 1987.A unifying theory and improvements for existing approaches to text compression.Ph.D. dissertation, Dept. of Computer Science, Univ. of Canterbury, New Zealand.

13.Bell T.C. and Witten I.H. 1987.Greedy macro text compression. Res. Rept.87/285/33. Department of Computers Science, University of Calgary.