ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 31 1995 Вып. 1

УДК 621.391.15

© 1995 г. Е.В. Курапова, Б.Я. Рябко

ПРИМЕНЕНИЕ ФОРМАЛЬНЫХ ГРАММАТИК ПРИ КОДИРОВАНИИ ИСТОЧНИКОВ ИНФОРМАЦИИ

Для повышения эффективности методов сжатия данных предлагается предварительно описывать их структуру с помощью формальных грамматик. Этот подход позволяет сжимать и файлы небольшого размера, при кодировании которых обычно неэффективны известные адаптивные и неадаптивные коды. На основе этого подхода построены и экспериментально исследованы системы сжатия для баз данных и библиотек программ на нескольких языках программирования. Полученные коды позволяют повысить степень сжатия данных на 10–30% по сравнению с существующими методами.

§ 1. Введение

Задача кодирования источников информации, или сжатия данных, представляет значительный теоретический и практический интерес. Существует множество различных методов сжатия данных, находящих широкое применение при архивации данных в компьютерных системах, в модемах, на цифровых линиях связи и т.п.

Все методы сжатия данных можно разбить на два класса: статические (неадаптивные) и адаптивные (или универсальные) [1]. Статические методы предназначены для сжатия данных с известной статистической структурой. Например, методы, специально предназначенные для текстов на русском языке, используют известные сведения о вероятностях слов, букв и их сочетаний в русском тексте. Статические методы дают значительный эффект только в случае достаточно точной известной статистики исходных данных.

В адаптивных методах при кодировании очередного символа текста используются сведения о ранее закодированной части сообщения, т.е. адаптивные методы настраиваются на статистическую структуру кодируемых данных. Это позволяет адаптивным методам эффективно сжимать данные с неизвестной заранее статистической структурой. Однако у этих методов есть два существенных недостатка: во-первых, они позволяют существенно сжимать только файлы достаточно большой длины; во-вторых, в практических задачах часто встречается ситуация, когда статистическая структура данных точно не известна, но имеются некоторые сведения о ней, которые могли бы быть использованы при сжатии данных. (Например, такие сведения могут быть сформулированы так: “исходный файл – программа на языке Паскаль”). В этом случае адаптивные методы не позволяют достаточно просто учесть имеющиеся сведения о структуре сообщений для повышения эффективности сжатия.

Ситуация, когда необходимо сжимать файлы небольшого объема, и (или) имеются некоторые сведения о структуре сообщений, встречается довольно часто. Например, библиотеки программ, написанных на каком-либо языке программирования,

как правило, состоят из большого числа небольших файлов (от нескольких Кбайт до нескольких десятков Кбайт). Такая же задача возникает и при хранении файлов в различных банках данных, при передаче сообщений электронной почты и т.п.

Прямое использование статических или адаптивных кодов в случае необходимости сжатия небольших файлов с меняющейся статистикой сообщений не позволяет существенно "сжать" данные. Поэтому предлагается промежуточный подход, сочетающий достоинства статических и адаптивных методов сжатия данных. Он основан на описании исходных данных с помощью формальных грамматик и дает возможность учитывать правила и закономерности формирования сообщений. При этом грамматика описывает общую структуру данных, а затем такое описание используется в процессе кодирования.

Для иллюстрации основной идеи предлагаемого метода рассмотрим структуру данных реального библиотечного банка ГПНТБ СО РАН г. Новосибирска, содержащего рефераты печатных работ и их информационные признаки. Каждое сообщение, соответствующее реферату статьи или книги, состоит из трех частей: цифры, представляющие собой информационные признаки реферата (примерно 1 Кбайт), текст реферата на английском языке и на русском языке (2–3 Кбайт). Для описания такой структуры достаточно ввести праволинейную грамматику с тремя состояниями A, B, C, где состояние A соответствует цифрам, состояние B – английскому тексту, состояние C – русскому тексту:

$$\begin{aligned} A &\to 0A\,|\,1A\,|\,\ldots\,|\,9A\,|\,\mathrm{a}B\,|\,\mathrm{b}B\,|\,\ldots\,|\,\mathrm{z}B,\\ B &\to \mathrm{a}B\,|\,\mathrm{b}B\,|\,\ldots\,|\,\mathrm{z}B\,|\,\mathrm{a}C\,|\,\text{б}C\,|\,\ldots\,|\,\text{я}C,\\ C &\to \mathrm{a}C\,|\,\text{б}C\,|\,\ldots\,|\,\text{я}C\,|\,0A\,|\,1A\,|\,\ldots\,|\,9A \end{aligned}$$

(см. описание классов грамматик в [2]). Неформально каждое состояние грамматики можно рассматривать как самостоятельный источник сообщений с алфавитом значительно меньшим, чем у исходного источника. Тогда при кодировании целесообразно использовать свой код для каждого состояния грамматики, так как уменьшение размера алфавита обеспечивает сокращение длин кодовых слов и позволяет использовать более точные оценки вероятностей встречаемости символов.

Эффективность использования грамматического описания при сжатии данных исследовалась нами по представительному набору данных различного типа. При этом для исследований были использованы тексты программ на языках Бейсик и Ассемблер, данные библиотечного банка, а также командные файлы базы данных FOXBASE. Для описания структуры исходных данных использовались простые классы грамматик – праволинейные и $LL(k)$ грамматики (см. определение, например, в [2]). Это дает возможность достаточно легко осуществлять анализ сообщений за один просмотр, не замедляя тем самым процесс кодирования. Как показали результаты исследований, использование формальных грамматик позволяет значительно повысить степень сжатия данных по сравнению с существующими методами.

§ 2. Использование формальных грамматик при кодировании источников с известной статистикой

В случаях кодирования файлов, содержащих рефераты статей, или файлов с текстами программ в библиотеках программ, при построении метода сжатия можно наряду с формально-грамматическим описанием использовать и сведения о частотах встречаемости различных букв и их сочетаний, получив их статистические оценки по представительному набору данных. Мы будем условно называть эту ситуацию задачей использования формальных грамматик при кодировании источников с известной статистикой. При этом процесс построения кода можно разбить на три этапа. Сначала общая структура исходных сообщений описывается с помощью формальной грамматики (такое грамматическое описание дает возможность

Таблица 1

Результаты сжатия текстов программ на языке Бейсик

Название файла	Длина файла	Предлагаемый метод	Стандартные архиваторы		
			ARJ 2,30	LHARC 1,13	PKZIP 1,02
B1	3154	40,4	47,6	50	51
B2	1995	38,8	46,7	50	52
B3	1503	34,8	48,9	52	55
B4	1378	35,9	47,6	50	52
B5	1107	38,3	51,8	56	57

при кодировании учесть общие закономерности формирования сообщений). Второй этап построения кода – исследование статистической структуры исходных данных. Для этого предлагается использовать представительную выборку, которая позволяет проанализировать состав сообщений и оценить вероятности появления символов и их сочетаний в различных грамматических состояниях источника. Затем на основе найденных оценок вероятностей осуществляется построение кодера и декодера. Мы использовали упрощенный вариант кода Хаффмена, позволяющий кодировать и декодировать сообщения с высокой скоростью и достаточно эффективно. Однако можно использовать и другие коды, например, арифметический.

Сущность предложенного алгоритма кодирования и декодирования заключается в использовании нескольких различных вариантов кода в зависимости от состояния грамматики. Точнее, для каждого грамматического состояния построен свой код, основанный на полученных статистических данных. При кодировании грамматика, описывающая структуру исходных данных, позволяет выяснить текущее состояние источника сообщений. По состоянию источника определяется код, а затем кодовое слово для текущего символа. При декодировании грамматическое состояние определяется по декодированной части сообщения так же, как и при кодировании, а затем по кодовой комбинации восстанавливается исходный символ.

С помощью разработанного метода были построены системы сжатия для библиотек программ, написанных на языке Бейсик и Ассемблер для IBM PC. Для системы сжатия текстов на языке Бейсик структура программ была описана с помощью грамматики $LL(10)$, состоящей из 15 правил. Использование грамматики именно этого класса объясняется тем, что названия команд и ключевых слов в языке Бейсик состоят не более, чем из 10 символов. Поэтому возникает необходимость "заглянуть" на несколько символов вправо от кодируемой буквы, чтобы идентифицировать команду или ключевое слово. Другими словами, при использовании грамматики $LL(k)$ "задержка" не превосходит k символов. По представительной выборке из 17 файлов, содержащих тексты программ на языке Бейсик, были оценены вероятности различных символов и команд в разных состояниях грамматики. Затем программным путем для каждого состояния грамматики были построены коды, обеспечивающие сжатие, близкое к максимальному. Аналогичным образом строилась система сжатия для программ на языке Ассемблер.

В табл. 1 приведены коэффициенты сжатия для текстов программ на языке Бейсик, полученные при использовании разработанной системы сжатия и наиболее известных стандартных архиваторов. Коэффициенты сжатия здесь выражают процентное отношение длины закодированного файла к длине исходного. Из табл. 1 видно, что в этом случае эффективность разработанной системы выше, чем у известных стандартных архиваторов. Аналогичные результаты были получены для системы сжатия программ на языке Ассемблер.

§ 3. Использование формальных грамматик при кодировании источников с неизвестной статистикой

Часто статистические характеристики источника не могут быть получены до построения метода сжатия данных. Эта ситуация возникает в случае, когда априорных сведений нет, или в случаях, когда статистические характеристики могут сильно меняться от файла к файлу. Например, при сжатии текстов программ на каком-либо языке программирования или сообщений электронной почты мы можем описать структуру сообщений с помощью формальных грамматик и использовать это описание для повышения эффективности сжатия, но не можем получить достаточно надежных оценок частоты встречаемости отдельных букв и слов, так как эти частоты сильно меняются от файла к файлу. Эту ситуацию мы будем называть задачей использования формальных грамматик при кодировании сообщений с неизвестной статистикой.

В этом случае целесообразно при кодировании наряду с грамматическим описанием использовать адаптивные коды. Мы исследовали эффективность грамматического описания в сочетании с различными наиболее известными адаптивными методами сжатия. К ним прежде всего относятся адаптивные словарные коды из класса Лемпела – Зива (LZ) и арифметический код (см. описание и обзор в [1]). Мы использовали при исследовании методы из каждого из этих двух классов, а также широко распространенный стандартный архиватор PKZIP для IBM PC.

В процессе исследований определялись коэффициенты сжатия данных указанными адаптивными методами с использованием грамматического описания и без него. По полученным результатам для всех методов был вычислен процент улучшения коэффициента сжатия данных при использовании грамматики по сравнению с обычным применением адаптивного метода. Например, использование арифметического кода для библиотечных данных позволяет уменьшить длину файла до 27% по отношению к исходной длине файла. Применение этого же метода совместно с грамматикой из двух состояний позволяет уменьшить длину файла до 14%, т.е. улучшить сжатие на 54% по сравнению с исходным арифметическим кодом.

Значения процента улучшения сжатия для всех исследованных данных и методов приведены в табл. 2. В ней графа LZW11 соответствует методу Лемпела – Зива – Велча [3] с длиной кодового слова 11 бит. В графе CODER приведены результаты использования быстрого словарного кода, работающего с байтами данных [4]. В последней графе представлены результаты для быстрого арифметического кода [5].

Таблица 2

Процент улучшения сжатия данных при использовании грамматик

Тип данных	Длина файла в байтах	Количество состояний грамматики	Методы сжатия			
			PKZIP	LZW11	CODER	ARIFM
Данные библиотечного банка ГПНТБ СО РАН	1001524	2	12	23	8	54
Командные файлы FOXBASE	102444	18	5	35	0	4
Тексты программ на языке Бейсик	70748	15	9	15	7	6

Как видно из табл. 2, практически во всех случаях применение формально-грамматического описания улучшает сжатие данных адаптивными методами. Величина процента улучшения сжатия различна для разных классов методов и типов данных. Это объясняется как особенностями самих методов, так и различиями в структуре исходных данных.

§ 4. Заключение

Таким образом, использование грамматического описания при кодировании позволяет значительно повысить эффективность методов сжатия данных. Важно отметить, что при этом кодирование и декодирование осуществляется за один просмотр, а выбор простых классов грамматик дает возможность легко осуществлять анализ сообщений, практически не замедляя процесс кодирования и декодирования.

СПИСОК ЛИТЕРАТУРЫ

1. *Bell T.C., Cleary J.G., Witten I.H.* Text Compression. New Jersey: Prentice Hall, Englewood Cliffs, 1990.
2. *Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции. Т. 1. М.: Мир, 1978.
3. *Welch T.A.* A Technique for High-Performance Data Compression // IEEE Computer. 1984. V. 17. № 6. P. 8–19.
4. *Перцев И.В., Рябко Б.Я., Ситняковская Е.И.* Эффективные методы сжатия данных для аппаратной реализации. // Тр. учебных институтов связи (в печати).
5. *Ryabko B.Ya.* Fast and Efficient Coding of Information Sources // IEEE Trans. Inform. Theory. 1994. V. 40. № 1. P. 96–99.

Поступила в редакцию
26.04.94