13) Сжатие информации. Алгоритм Хаффмена.

Существующие алгоритмы сжатия данных можно разделить на два больших класса – с потерями, и без. Алгоритмы с **потерями** обычно применяются для сжатия изображений и аудио. Эти алгоритмы позволяют достичь больших степеней сжатия благодаря избирательной потере качества. Однако, по определению, восстановить первоначальные данные из сжатого результата невозможно.  
Алгоритмы сжатия **без потерь** применяются для уменьшения размера данных, и работают таким образом, что возможно восстановить данные в точности такими, какие они были до сжатия. Они применяются в коммуникациях, архиваторах и некоторых алгоритмах сжатии аудио и графической информации. Основной принцип алгоритмов сжатия базируется на том, что в любом файле, содержащем неслучайные данные, информация частично повторяется. Используя статистические математические модели можно определить вероятность повторения определённой комбинации символов. После этого можно создать коды, обозначающие выбранные фразы, и назначить самым часто повторяющимся фразам самые короткие коды. Для этого используются разные техники, например: энтропийное кодирование, кодирование повторов, и сжатие при помощи словаря. С их помощью 8-битный символ, или целая строка, могут быть заменены всего лишь несколькими битами, устраняя таким образом излишнюю информацию.

Один из первых алгоритмов эффективного кодирования информации был предложен Д. А. Хаффманом в 1952 году. Идея алгоритма состоит в следующем: зная вероятности символов в сообщении, можно описать процедуру построения кодов переменной длины, состоящих из целого количества битов. Символам с большей вероятностью ставятся в соответствие более короткие коды. Коды Хаффмана обладают свойством [префиксности](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D1%84%D0%B8%D0%BA%D1%81%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4" \o "Префиксный код) (то есть ни одно кодовое слово не является префиксом другого), что позволяет однозначно их декодировать. Х

Классический алгоритм Хаффмана на входе получает таблицу частот встречаемости символов в сообщении. Далее на основании этой таблицы строится дерево кодирования Хаффмана (Н-дерево).[[1]](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%A5%D0%B0%D1%84%D1%84%D0%BC%D0%B0%D0%BD%D0%B0#cite_note-1)

1. Символы входного алфавита образуют список свободных узлов. Каждый лист имеет вес, который может быть равен либо вероятности, либо количеству вхождений символа в сжимаемое сообщение.
2. Выбираются два свободных узла дерева с наименьшими весами.
3. Создается их родитель с весом, равным их суммарному весу.
4. Родитель добавляется в список свободных узлов, а два его потомка удаляются из этого списка.
5. Одной дуге, выходящей из родителя, ставится в соответствие бит 1, другой — бит 0. Битовые значения ветвей, исходящих от корня, не зависят от весов потомков.
6. Шаги, начиная со второго, повторяются до тех пор, пока в списке свободных узлов не останется только один свободный узел. Он и будет считаться корнем дерева.

14) Информационная зависимость. Совместное распределение. Условная и взаимная информация. Функциональная зависимость.

**Совместное распределение.**

Пусть имеется пара источников информации, которые согласованно выдают последовательности символов, первый из алфавита А, второй – из алфавита В. В качестве примера согласования источников можно рассмотреть следующий: первый источник- это первая буква в слове, а второй вторая буква. В этом случае алфавиты источников одинаковы. Согласованность этих источников заключается в том, что они выдают буквы одного и того же слова. Другой вариант согласования источников – синхронизация: оба источника выдают сообщения в одни и те же моменты времени или дают информацию об одном и том же явлении.

Всякий способ согласования позволяет рассматривать пару источников как одни составной источник, алфавитом которого служит произведение А×В. Частотная характеристика составного источника называется **совместным распределением** пары источников. Изображается он в виде матрицы частот с которой встречается пара (А[i], B[j]), состоящая из i-го символа алфавита А и j-го символа алфавита В.

**Условная и взаимная информация.**

Пусть H1 и H2 представляют собой энтропии двух согласованных источников, а H12 – энтропия их совместного распределения. Энтропия совместного распределения удовлетворяет:

Разность представляет собой кол-во дополнительной информации, которую второй источник несет по отношению к первому. Эта разность называется **условной информацией** первого источника относительно второго. Следовательно:

– называется **взаимной информацией** пары источников

- **условная информация** первого источника от второго

- **условная информация** второго источника от первого

**Функциональная зависимость**

Если условная информация одного источника относительно другого равна нулю, то, зная содержание второго источника, можно, однозначно определить содержание первого.

Следует: источник информации с алфавитом В **функционально зависит** от согласованного с ним источника информации с алфавитом А, если имеется функция , такая , что всякий раз, когда порождает некоторый символ ,источник порождает

15) Защита информации. Избыточность. Контроль четности. Контрольная сумма. Локализация ошибки. Кодирование. Декодирование.

**Избыточность**. При передаче информации даже у самой совершенной техники случаются ошибки. Если же файл не был сжат, то изменение одного бита изменит лишь одну букву текста и эта ошибка скорее всего будет легко исправима и практически не повлияет на смысл текста. Это свойство несжатого текста связано с его избыточностью. Естественный язык обладает значительной информационной избыточностью, которая позволяет восстанавливать слова с несколькими ошибками. В искусственных языках избыточность создается искусственно.

**Контроль четности.** Для уменьшения числа ошибок информацию часто передают с дополнительными данными -- контрольными суммами. Простейший вид контрольной суммы представляет собой так называемый бит четности. При каждой передаче байта внутри компьютера передаются на самом деле не восемь а девять битов. Девятый -- невидимый для пользователя бит четности определяется нулем, если количество единиц в байте четно и единицей в противном случае. Таким образом, сумма цифр расширенного байта (байта с битом четности) всегда четна. И сохранение этой четности контролируется при всех передачах. Если обнаруживается ошибка четности, то есть нечетность у полученного расширенного байта, то, этот байт считывается заново или выдается сообщение об ошибке чтения.

**Контрольная сумма.** Важнейшим программным способом защиты информации являются контрольное суммирование. Всякий файл можно рассматривать как последовательность чисел, и контрольная сумма определяется как поразрядная сумма этих чисел. После чтения файла вычисляется контрольная сумма у считанного файла и сравнивается с записанной контрольной суммой. Этот метод применяется также для защиты от вирусов.

**Локализация ошибки**. Если мы точно знаем в каком бите ошибка, то мы можем ее исправить, достаточно изменить значение этого бита на противоположное.

Код Хэмминга позволяет найти ошибку в последовательности битов при условии, что ошибок не больше одной. Если требуется передать п -- битов, а передается N > п. В переданном тексте, кроме п битов информации должно оставаться место для информации о позиции ошибки -- одном из N мест, то есть logN + 1 бит (нулевая возможность -- отсутствие ошибок). Таким образом . Видно, что количество дополнительных битов логарифмически зависит от длины кодовой

**Кодирование.** Кодирование информации -- процесс преобразования сигнала из формы, удобной для непосредственного использования информации, в форму, удобную для передачи, хранения или автоматической переработки.

Энтропийное кодирование -- кодирование последовательности значений с возможностью однозначного восстановления с целью уменьшения объёма данных (длины последовательности) с помощью усреднения вероятностей появления элементов в закодированной последовательности.

**Декодирование** - преобразование зашифрованной информации в понятный, пригодный для непосредственного использования вид.

16) Агрегатное описание систем.

Агрегат - унифицированная схема, получаемая наложением дополнительных ограничений на множества состояний, сигналов и сообщений и на операторы перехода, а так же выходов.

t ∈ T - моменты времени; x ∈ X - входные сигналы; u ∈ U - управляющие сигналы; y ∈ Y - выходные сигналы; z ∈ Z - состояния, x(t), u(t), y(t), z(t) - функции времени.

Агрегат - объект, определенный множествами T, X, U, Y, Z и операторами H и G реализующими функции z(t) и y(t). Структура операторов H и G является определяющей для понятия агрегата.

Вводится пространство параметров агрегата b=(b1, b2, ...,bn) ∈ B.

Оператор выходов G реализуется как совокупность операторов G` и G``. Оператор G` выбирает очередные моменты выдачи выходных сигналов, а оператор G`` - содержание сигналов.

у=G``{t, z(t),u(t),b}.

В общем случае оператор G`` является случайным оператором, т.е. t, z(t), u(t) и b ставится в соответствие множество y с функцией распределения G``. Оператор G` определяет момент выдачи следующего выходного сигнала.

Операторы переходов агрегата. Рассмотрим состояние агрегата z(t) и z(t+0).

Оператор V реализуется в моменты времени tn , поступления в агрегат сигналов xn(t). Оператор V1 описывает изменение состояний агрегата между моментами поступления сигналов.

z(t’n + 0) = V{ t’n, z(t’n), x(t’n), b}.

z(t) = V1(t, tn, z(t+0),b}.

Особенность описания некоторых реальных систем приводит к так называемым агрегатам с обрывающимся процессом функционирования. Для этих агрегатов характерно наличие переменной соответствующей времени оставшемуся до прекращения функционирования агрегата.

Все процессы функционирования реальных сложных систем по существу носят случайный характер, по этому в моменты поступления входных сигналов происходит регенерация случайного процесса. То есть развитие процессов в таких системах после поступления входных сигналов не зависит от предыстории.

Автономный агрегат - агрегат который не может воспринимать входных и управляющих сигналов.

Неавтономный агрегат - общий случай.

Частные случаи агрегата:

- кусочно-марковский агрегат - агрегат процессы, в котором являются обрывающими марковскими процессами. Любой агрегат можно свести к марковскому;

- кусочно-непрерывный агрегат - в промежутках между подачей сигналов функционирует как автономный агрегат;

- кусочно-линейный агрегат. dzv(t)/dt = F(v)(zv).

Представление реальных систем в виде агрегатов неоднозначно, вследствие неоднозначности выбора фазовых переменных.

Иерархические системы. Иерархический принцип построения модели как одно из определений структурной сложности. Иерархический и составной характер построения системы.

Вертикальная соподчиняемость. Право вмешательства. Обязательность действий вышестоящих подсистем. Страты - уровни описания или абстрагирования. Система представляется комплексом моделей - технологические, информационные и т.п. со своими наборами переменных.

Слои - уровни сложности принимаемого решения:

1) срочное решение;

2) неопределенность или неоднозначность выбора.

Разбитие сложной проблемы на более простые: слой выбора способа действия, слой адаптации, слой самоорганизации.

Многоэшелонные системы. Состоят из четко выраженных подсистем, некоторые из них являются принимающими решения, иерархия подсистем и принятия решений.

Декомпозиция на подсистемы - функционально-целевой принцип, декомпозиция по принципу сильных связей.

17) Дать определение дополнительного множества к множеству Χ по отношению к множеству А.

Дополнительным к множеству Χ по отношению к универсальному множеству А, если Хhttp://lib.kstu.kz:8300/tb/books/Informatcionnoe_obespechenie_sotcial@mno-@ekonomicheskih_protcessov/labas/laba8.files/image008.gifА, называется множество, состоящее из элементов множества А, не принадлежащих множеству X. Символически обозначается

Ca(x)

18) Теорема Шеннона для эффективных кодов. Теорема Шеннона для корректирующих кодов