Этюды для программистов

Чарльз Уэзерелл

Этюды для программистов

Чарльз Уэзерелл

дата публикации

Содержание

Список иллюстраций

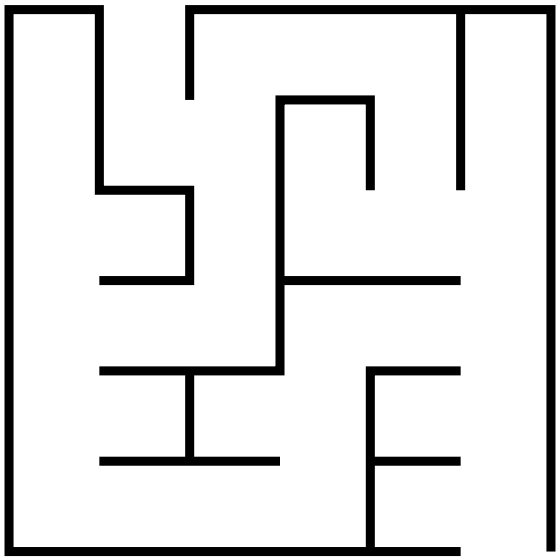
Глава 1. Тезей,

или автоматическое построение лабиринтов

Тезей должен был найти выход из Критского лабиринта или погибнуть от руки Минотавра. Но что поразительно: найти вход в лабиринт— задача не менее трудная.

Здесь не представляется возможным описать все мыслимые лабиринты, да это и не требуется. Мы займемся простыми лабиринтами, построенными на прямоугольнике $m \times n$, где n, m-положительные целые числа. Внутри и на границах прямоугольника поставлены стенки по ребрам покрывающей его единичной квадратной сетки. Чтобы построить из прямоугольника лабиринт, выбьем одну единичную стенку на одной из сторон прямоугольника (получится вход в лабиринт); выбьем одну единичную стенку на противоположной стороне (получится выход) и еще удалим какое-то число строго внутренних стенок. Говорят, что лабиринт имеет решение, если между входом и выходом внутри лабиринта есть путь в виде ломаной, не имеющей общих точек со стенками. Решение единственно, если любые два таких пути проходят через одни и те же внутренние ячейки сетки. На Рисунок 1.1, «Пример лабиринта.» приведен пример лабиринта 6×6 .

Рисунок 1.1. Пример лабиринта.



Тема

Напишите программу, которая по исходным данным т и п строит прямоугольный лабиринт $m \times n$ (проверьте, допустимы ли заданные m и n). Предусмотрите, чтобы программа при каждом обращении к ней порождала разные лабиринты. Лабиринт должен иметь единственное решение, и, чтобы получившийся лабиринт был интересным, все ячейки должны быть соединены с основным путем, дающим решение. Если в вашем распоряжении имеется хорошее графическое устройство, используйте его для изображения лабиринтов, в противном случае придумайте систему обозначений для записи лабиринтов или выводите лабиринты на АЦПУ.

Указания исполнителю

Теоретически нельзя удовлетворить требованию, чтобы любые два лабиринта (даже при одинаковых m и n) были различны, поскольку существует лишь конечное число лабиринтов любого наперед за-

данного размера, а программу можно вызвать большее число раз. Однако число лабиринтов какого-нибудь размера очень велико, и поэтому вероятность повторения лабиринта можно сделать очень маленькой. Практически это достигается, если программа будет производить «случайный» выбор различных вариантов, опираясь на какое-либо доступное ей, но неуправляемое значение (обычно берут дату и время вызова программы). Варианты, между которыми выбирает программа, это, например, положение входа и выхода и положение хотя бы нескольких внутренних разрушаемых стенок. При отладке разумно будет отключить механизм случайного выбора, чтобы изменения результата работы вызывались только изменениями самой программы.

Один из возможных подходов к решению таков. Выбираем вход; затем, начав от него, добавляем по одной ячейке к главному пути-решению, пока он не достигнет выходной стороны. После этого удаляем некоторые внутренние стенки так, чтобы все клетки оказались соединенными с главным путем. Чтобы главный путь не получился прямым коридором, следует при его построении предусмотреть случайные повороты. Программа должна также следить за тем, чтобы при построении главного пути или при открытии боковых ячеек не нарушалась единственность решения. Наблюдательный читатель заметит, что определение единственности решения не годится в случае, когда путь заходит в боковой тупик И затем возвращается. Вы можете попробовать разработать в том же духе формально корректное определение.

Инструментовка

Программу можно написать почти на любом из процедурных языков. Используйте эту программу для сравнения языков с точки зрения управляющих структур, встроенных структур данных И эффективности выполнения.

Длительность исполнения

Одному исполнителю на 3 недели.

Глава 2. Секрет фирмы

или математический подход к раскрытию шифров

Представьте себе такую ситуацию. Благодаря выдающимся профессиональным познаниям и незаурядным программистским способностям вас выдвинули на должность руководителя большой группы сотрудников, занимающихся разработкой суперновейшего и пока еще секретного Мини-компилятора для ЭВМ УМ-1 (см. Глава 4, *Мал золотник…* и Глава 3, *Уча — учимся*). Как-то раз, уходя со службы около часу ночи (руководитель должен подавать хороший пример), вы замечаете торчащий в дверях измятый клочок бумаги (содержание которого воспроизведено на Рисунок 2.1, «Таинственная записка»). Сначала вы решаете, что это запись содержимого памяти машины, и уже собираетесь выбросить бумажку. Но, присмотревшись повнимательнее, замечаете, что буквы собраны в группы по пять, — очень странно для УМ-1. Что бы это мло быть?

Рисунок 2.1. Таинственная записка, найденная в вычислительном центре. Случайное вкрапление русских слов, например ШИШ или ОЙ, по-видимому, ничего не означает. Но обратите внимание на повторение сочетаний ЗАЮЪИВУ, ЬЬК, других коротких сочетаний, а особенно повторяющуюся группу букв КНДЙЯГЭ

```
ЖНФЖП ЕЕЫШВ ЛПЖАТ ГФБЦМ КЖЬЗА ЮЪИВУ ЩЖРСЮ БЬЬКЬ ЫЕСУУ ЦТЮБШ УНЖЦМ
ЭЭШЮЗ УЬЕКН АУЕЫЩ ШЖРЬЙ ЛЮПКН ДЙЯГЭ ЪЖЫГЖ ОУШИШ УФГВР ШМАГВ ВУВОС
ЗХЧИУ ГНЛАЯ ЬЬКИЯ РЦЖРЫ АХЪВИ ЖГЭЯЦ СЪУЫФ ЯРМЗФ ЧФЪЩС ЬФШВЕ ОМКТИ
МБЭВЪ КФХЙЦ ХНЬЮЬ МФЛБИ МРУЛМ ЯЗФЧЪ ЪЧЗНК ЗНИВЛ НШГЛЩ ИЛЗНФ ФУЖКН
ДЙЯГЭ ЕУЮЛЛ ЮЖНЯИ ЕМДЙШ ГЯУГВ ЦФЩВЮ МФАГЯ ВХМЭВ ВФПГФ ФЖККГ ЦМЛЫБ
ШМПУЕ ШЖЛЯЮ ЯРЧВЪ ЖУПВМ КЛЫЭС ЭЧИРЫ ГЫЩЗЗ ЗКЖЛЕ ШВРЪЧ ЪААЖЗ ДХЪФС
БРНМЪ КЫБЪФ УНЦЮБ ТЖУНЯ ЕШИМУ КФВГВ ГЧМЭВ ЗРВМЪ ЪЕЕТО ЯЦБЖГ ВИЖМД
КЗЗПА ФЯВНР ЫГЮЩЭ ЯЬЦШЪ ЧНГВЫ АХЪВЛ НШАПВ ЧОЬОЙ КЮАШО КЗЛЩУ ШЯРНЗ
ГХЛТЮ ЖЫШШГ ППЬЫШ АЬФМА ФЕЙЗА ЙПЛУЭ ЖЛЗИЗ НЖККР ЦЯДЧК НДЙЯГ ЭБФЬА
ВБЭКЗ ФКЫТВ ЛЕЪЭЯ ЛЭЩЗН ФХГЧК ТКЫЮЗ ЗЪУЖА ПВЧОЬ ОЙКЕС ЛЗАЮЪ ИВУНЫ
ПКЗВЯ ЪГОСЩ ЛБЬГМ ЯВЗГЬ КШЬГЙ ЕНПСМ ЭВГОГ ЧСОРГ ЩОЦМВ ДГЩКЧ ЮЗВЗК
ЦЧЯРЧ ВЪЖФЫ ЕЛЖАЪ УССХР УОЬЫЕ ЙГЫОТ УЕАГЖ ГЫСЩИ ЯРВТЮ ДЖНЛГ ЦМЗЬЪ
ЯИЦТР ЕМИКЦ ЩВЦОР ЛХМХЖ ВРЬПУ ГВЯРЬ ПМЯЖЖ РЧПШЪ ЧУВГЧ СЕЕГЦ ЬПЗДМ
ОЬОЧЗ КВУФЯ УПОХЪ ГЪЭЯЖ ВЖФ
```

Снова возвращаетесь в свой кабинет, пытаясь решить загадку. Бумага отменная, слегка пахнет мускусом; почерк явно женский и веет от него этаким французским шармом. Теперь, по здравом размышлении, новая сотрудница мисс Хари начинает казаться вам, пожалуй, немножко слишком экзотичной. Ее французский акцент, неизменное черное платье для коктейля, нитка черного жемчуга, подчеркивающая декольте, и этот будоражащий запах мускуса, наполняющий комнату, когда она туда входит... Она говорит, что работала раньше в региональном вычислительном центре Мак-Дональда в Киокаке. Что-то тут не так. Подождите... Неужели мисс Хари шпионит в пользу знаменитой французской фирмы И Бей Эм? А эта записка-шифровка, в которой все секреты вашего новейшего чудо-компилятора? Чтобы уличить мисс Хари, записку нужно расшифровать. Но как? Может, обратимся за помощью к компьютеру?

Основы шифрования

ЭВМ, безусловно, может оказать помощь, иначе Управление национальной безопасности просто пускает на ветер деньги налогоплательщиков, закупая такое количество техники. Для начала необходимо как следует присмотреться к секретному сообщению. Возможно, что найденная записка была

зашифрована при помощи простой подстановки, т. е. каждая буква первоначального текста была заменена какой-либо другой буквой согласно некоторому правилу шифрования. Сообщение, подвергшееся зашифровке, называется исходным текстом, а в результате получается шифрованный текст. Задача состоит в том, чтобы восстановить исходный текст и правило шифрования (последнее нужно лишь в том случае, если могут появиться другие сообщения, зашифрованные по тому же правилу). Будем предполагать, что исходный текст написан по-русски. Разбиение шифрованного текста на группы по пять букв скрывает, по-видимому, исходную структуру текста, разбитого на слова, которая была бы весьма ценной подсказкой, облегчающей расшифровку.

В простейшем общем классе подстановочных шифров для построения правила шифрования используется некоторый смешанный алфавит, например перестановка обычного алфавита. На рис. 24.2 показан полный исходный алфавит, смешанный алфавит и шифрование короткого сообщения, в котором каждая буква заменяется соответствующей буквой смешанного алфавита. Всякий, кто увлекается головоломками из воскресных газет, знает, что зашифрованные такой подстановкой тексты расшифровываются до смешного просто: сообщения из 30 или 40 букв зачастую оказывается для этого вполне достаточно. Тем не менее, слегка усовершенствовав эту систему, можно сделать ее значительно более надежной.

На рис. 24.3 изображен квадрат Виженера, построенный на основе смешанного алфавита, приведенного на рис. 24.2. Сверху и по левому краю квадрата выписан исходный алфавит. В первой строке квадрата представлен смешанный алфавит. Во второй строке тот же алфавит циклически сдвинут на одну позицию, при этом первая буква переместилась в правый конец строки. Квадрат состоит из 32 смешанных алфавитов, полученных из одного смешанного алфавита, каждому из них соответствует та буква исходного алфавита, которая записана слева от него. На рис. 24.4 показано шифрование фразы при помощи ключевого слова ЛИСП и данного квадрата. Ключевое слово многократно записывается под исходным текстом, и каждая буква исходного текста шифруется при помощи смешанного алфавита, соответствующего той букве ключевого слова, которая стоит под данной буквой исходного текста. Эта схема шифрования уже не поддается раскрытию при помощи простого подсчета частот букв, поскольку одна и та же буква исходного текста шифруется по-разному в зависимости от выпавшей на нее буквы ключевого слова. Кроме того, выбрав заранее список ключевых слов и порядок их смены, отправитель И получатель могут повысить секретность переписки, поскольку разным сообщениям будут соответствовать разные ключевые слова, благодаря чему затрудняется анализ, основанный на частотах букв. Тем не менее не так уж все это безнадежно.

Как раскрыть шифр

Будем предполагать, что криптограмма мисс Хари получена при помощи квадрата Виженера, хотя бы по той причине, что он — ее соотечественник. Если наше предположение неверно, методы решения позволят обнаружить это. Если бы сообщение было зашифровано при помощи простой подстановки, то расшифровать его можно было бы, подсчитав количество появлений каждой буквы в шифрованном тексте, поделив это количество на длину сообщения и сравнив полученные величины с частотами букв русского алфавита, приведенными на рис. 24.5. Для сообщений такой длины, как наше, распределения частот, если выписать их в убывающем порядке, почти полностью совпадут, и, таким образом, для каждой буквы исходного текста откроется ее двойник в шифрованном тексте. Но для квадрата Виженера такой простой метод уже не сработает. Необходимо определить не только смешанный алфавит, но и ключевое слово; поскольку каждый из этих элементов искажен другиим, то трудно даже догадаться, с какого конца начать.

Правильной отправной точкой будет нахождение длины ключевого слова. Обратите внимание, что в примере на рис. 24.4 первая, пятая, девятая,... буквы исходного текста зашифрованы при помощи одного и того же смешанного алфавита Л. Если рассматривать лишь каждую четвертую букву шифрованного текста, то получим распределение частот, подобное распределению для букв русского алфавита, поскольку буквы в этих позициях зашифрованы при помощи одного и того же смешанного алфавита, т. е. при помощи простой подстановки. Аналогично если взять каждую четвертую

букву шифрованного текста, начиная со второй, третьей или четвертой позиции, то снова получим распределение частот как для букв русского алфавита. Существует способ измерить, насколько данное распределение частот подобно распределению букв алфавита. Рассмотрим индекс совпадения

$$\text{ИС} = \sum_{i=1}^{32} \frac{f_i(f_i - 1)}{N(N - 1)} \tag{2.1}$$

где N-количество появлений i-й буквы, а N-общее число рассматриваемых букв. Если все буквы рассматриваемого подмножества текста зашифрованы при помощи одного алфавита, то этот индекс совпадения должен иметь значение больше 0.045 и, вероятно, меньше 0.065 (теоретическое значение равно 0.055). Исходя из этого, алгоритм определения длины ключевого слова будет таким:

1. Для i от 1 до 20 предположить, что длина ключевого слова равна 1, и выполнить шаги 2, 3, 4. Мы выбрали верхнюю границу равной 20 лишь для удобства. Разумеется, ключевое слово может быть и длиннее.

2. Для j от 1 до i выполнить шаг 3. В этих двух шагах будут вычислены i различных значений ИС.

3. Построить распределение числа появления букв в позициях j, i+j, 2i+j, ..., т. е. в каждой i-й позиции, начиная с j-й позиции. По формуле, приведенной выше, вычислить ИС_i для полученного распределения. В качестве N в этой формуле нужно использовать число букв в данном подмножестве текста, а не длину всего текста.

4. Если все значения ИС_1, ИС_2, ..., ИС_i больше 0.045, то, вероятно, i кратно длине ключевого слова. Если только один из ИС меньше 0.045, то i также может быть кратно длине ключевого слова.

Проверить длину ключевого слова можно И другим способом. Найдите два места в шифрованном тексте, где две одинаковые буквы идут в том же порядке, например ЦМ в позициях 19 И 54 на рис. 24.1. Такое повторение могло произойти по двум разным причинам. Возможно, в соответствующих местах исходного текста были различные сочетания букв, которым отвечали разные части ключевого слова, и они случайно отобразились в одинаковые сочетания букв, либо в исходном тексте были повторения, которые попали на одинаковые части ключевого слова, И, таким образом, оказались зашифрованными дважды одним и тем же способом. Во втором случае расстояние между началами повторяющихся сочетаний букв должно быть кратно длине ключевого слова. Ксожалению, невозможно определить, по какой из двух причин произошло повторение данного сочетания букв: случайное повторение пар букв в шифрованном тексте довольно частое явление. Но если в шифрованном тексте повторяются сочетания из трех или более букв, то вероятность того, что это повторение произошло случайно, а не в результате повторения ключа, очень мала (для сочетаний из четырех и более букв она практически нулевая). Таким образом, другой способ выявления длины ключевого слова — отыскать в шифрованном тексте все пары повторяющихся групп из трех и более букв и измерить расстояния между ними. Число, которое делит 90% или более из этих расстояний, — прекрасный претендент на роль длины ключевого слова. Данная проверка вместе с вычислением значений ИС однозначно определяет длину ключевого слова.

Предположим, нам удалось выяснить, что длина ключевого слова равна k. Тогда первоначальный шифрованный текст можно разбить на k групп G1, G2, ..., Gk, где каждая группа начинается с позиции i, $1 \le i \le k$, и содержит каждую k-ю букву текста, начиная с i-й буквы. Каждая из этих k групп была зашифрована при помощи только одного алфавита, т. е. при помощи простой подстановки. Остается в каждой группе для каждой шифрованной буквы определить ее эквивалент в исходном тексте. Но здесь у нас имеется хорошее подспорье. Если бы был известен алфавит, по которому была зашифрована какая-нибудь из групп, то алфавит, по которому была зашифрована любая другая группа, можно было бы найти путем циклического сдвига уже известного алфавита на некоторое число букв. С другой стороны, определить исходные эквиваленты букв было бы проще, если бы удалось распределения числа появлений букв для различных групп скомбинировать в одно обобщенное

распределение, поскольку, чем больше данных было использовано для построения какого-либо распределения, тем достовернее будут сделанные на его основе статистические выводы. Для построения такой комбинации необходимо знать относительные сдвиги между алфавитами, использованными для шифрования различных групп.

Относительные сдвиги находятся при помощи некой модификации индекса совпадения. Построим для каждой группы G_i распределение числа появлений букв И запишем его в алфавитном порядке шифрованных букв. В табл. 24.1 показаны распределения для сообщения, приведенного на рис. 24.1, в предположении, что k=7. Пусть $f_{i\alpha}$-количество появлений буквы α алфавита i. Определим функцию

$$R_{i,\ j,\ r} = \sum_{\beta=1}^{32} f_{i,\ \beta}\ f_{j,\ \beta\ +\ r} \tag{2.2}$$

Считается, что если β+r больше 32, то происходит циклический возврат к началу алфавита. Чем больше значение $R_{i,j,r}$, тем больше вероятность того, что алфавит для группы j в квадрате Виженера находится на r позиций ниже алфавита для группы i. Вычислим все значения $R_{i,j,r}$ (для $j \le i$ их можно не вычислять благодаря свойству симметрии) и выберем i и j, которые дают максимальное значение $R_{i,j,r}$. Вероятно, группа j сдвинута на r позиций относительно группы i.

Из групп G_i и G_j построим новую супергруппу G_{ij}, положив величину $f_{ij,\alpha}$, равной $f_{i,\alpha} + f_{j,\alpha+r}$. Отбросим из рассмотрения группы G_i и G_j, заменив их группой G_{ij}, и повторим описанный в последних двух абзацах процесс. После k-1 повторений станут известны относительные сдвиги для всех k алфавитов. Кроме того, будет найдено обобщенное распределение частот. Для того чтобы найти исходные эквиваленты букв шифрованного текста, переупорядочим последние согласно их частотам. В результате буквы шифрованного текста должны расположиться в том же порядке, что и буквы русского алфавита (см. рис. 24.5). Теперь нетрудно восстановить весь квадрат Виженера и расшифровать текст. Ключевое слово можно найти, перебрав 32 набора из букв, относительные расстояния между которыми соответствуют найденным сдвигам алфавитов. Возможно, что некоторые редко встречающиеся буквы окажутся не на своих местах. Эту ситуацию можно поправить при помощи визуального исследования полученного текста. Следует восстановить и смешанный алфавит, и ключевое слово, поскольку они оба могут иметь некоторую психологическую связь с содержанием сообщения и их выявление поможет дополнительно убедиться в правильности решения. Между прочим, что же написала мисс Хари?

Тема.

Напишите программу, которая в качестве входных данных воспринимает шифрованное сообщение и, в предположении, что оно зашифровано по схеме Виженера, печатает расшифрованный текст. Программа должна также печатать квадрат Виженера и ключевое слово, которые она вычисляет в процессе решения задачи. Специальные входные параметры должны управлять выводом промежуточных результатов, таких, как, например, все возможные длины ключевого слова, распределения частот букв для отдельных алфавитов, значения ИС и т. д., которые нужны для контроля. Эти результаты могут быть полезны при отладке, а также в тех, к сожалению, вполне реальных ситуациях, когда предложенное машиной решение оказалось не совсем точным. Четкость оформления выводных данных имеет большое значение: бестолковые распечатки лишь затрудняют работу интуиции специалиста по расшифровке сообщений.

Указания исполнителю.

Описанные здесь алгоритмы вполне понятны и легко реализуются, но обладают одним неприятным свойством — они не дают однозначного результата. Длина ключевого слова, например, будет лишь «вероятной», так что необходимо еще сделать обоснованный выбор одной из возможных длин. Аналогично алгоритмическое определение исходных эквивалентов для редко встречающихся букв ши-

фрованного текста следует проверить, убедившись, что при расшифровке получаются правильные русские слова. Увеличивая статистическую информацию, доступную программе, мы получим более надежное основание для алгоритмических решений, но все равно эти решения должен проверить человек. Помимо указанных алгоритмов в вашей программе должны быть реализованы средства, позволяющие подтвердить обоснованность выводов, которые делает программа. Один хороший способ обеспечить такую оценочную функцию — написать программу в рамках какой-либо диалоговой системы, чтобы программа и пользователь смогли совместно обсудить качество каждого решения до того, как оно будет окончательно принято. «Обсуждение» обычно состоит в том, что программа сообщает человеку факты, говорящие в пользу того или иного возможного решения, а человек либо принимает его, либо отвергает, после чего вычисление может быть продолжено.

Несмотря на то что алгоритмы неоднозначны и такая расплывчатость обычно порождает у программиста чувство неуверенности, эту программу легко проверить. Первой частью работы, по-видимому, должна быть программа шифровки, которая воспринимает в качестве исходных данных русский текст и, выбрав некоторым случайным образом смешанный алфавит и ключевое слово, выдает квадрат Виженера и печатает зашифрованный текст в стандартном пятибуквенном формате. Пробелы и пунктуация должны убираться из текста автоматически. Эта программа должна уметь также воспринимать в качестве возможных параметров квадрат Виженера и ключевое слово, чтобы можно было повторно проверять отдельные особенности работы программы расшифровки. Помните о том, что для хорошего статистического поведения алгоритмов необходимо, чтобы сообщение было в 30-40 раз длиннее ключевого слова.

Инструментовка.

Эта задача прямо-таки создана для языка типа Снобол, в котором средства работы с текстовыми данными сочетаются с простыми арифметическими операциями. Хорошим кандидатом может бьхть и какой-нибудь другой язык, с более широким диапазоном алгебраических вычислений и с достаточными средствами обработки текстовых данных, например PL/I, Паскаль или XPL. Но какой бы язык вы ни выбрали, постарайтесь избежать представления литер целыми числами: требования машинного представления не должны навязывать некрасивое, путаное решение задачи.

Длительность исполнения.

Одному исполнителю на 2 недели.

*Партия переводчика.

При переводе на русский язык зашифрованного примера надо было сначала расшифровать его. Попытка сделать это с помощью описанной процедуры не привела к успеху. После небольшого размышления стало ясно, что наш ключ не подходит потому, что он от другого замка! Действительно, предлагаемый автором способ определения относительных сдвигов столбцов с помощью величин $R_{i,j,r}$ исходит из того, что два столбца отличаются, кроме случайных отклонений, циклическим сдвигом на величину, равную разности номеров двух букв ключевого слова. Это свойство будет иметь место, если несколько изменить способ шифрования. В нашем случае вместо $R_{i,j,r}$ следует использовать числа $p_{i,j,r}$, вычисляемые, как описано ниже.

Пусть число букв алфавита равно n. Будем обозначать i-ю букву алфавита x_i или y_i, в зависимости от того, идет речь об исходном тексте или о зашифрованном. Нам известны средняя частота $p_i=p(x_i)$ появления i-й буквы в русском языке, число $f_{k,j}$ появлений i-й буквы в k-й группе зашифрованного текста, общее число N_k букв в k-й группе. Определим вероятности $p_k(y_j|x_i)$ появления фактического числа букв $f_{k,j}$, если буква y_i в k-й группе обозначает букву x_i исходного текста. Эти вероятности подчиняются биномиальному распределению.

$$p_k(y_j|x_i) = C_{N_k}^{f_{k,j}} \; p_i^{f_{k,j}} \left(1 - p_i\right)^{N_k - f_{k,j}} \tag{2.3}$$

Далее найдем по формуле Байеса вероятности $p_k(y_j|x_i)$ того, что буква y_i в k-й группе означает букву x_i исходного текста. Априорные вероятности гипотез примем равными 1/n.

$$p_k(y_j|x_i) = \frac{(1/n)\; p_k(y_j|x_i)}{\sum_{m=1}^{n}(1/n)\; p_k(y_j|x_m)} = \frac{p_k(y_j|x_i)}{\sum_{m=1}^{n} p_k(y_j|x_m)} \tag{2.4}$$

Рассмотрим теперь пару групп (столбцов табл. 24.1) k и l. Будем говорить, что между ними имеется сдвиг r, если каждой букве y_j зашифрованного текста в l-й группе соответствует буква исходного текста на r большая (по модулю n), чем в k-й группе. Это означает, что в ключевом слове l-я буква на r меньше k-й. Для оценки вероятностей $p_{k,l,r}$ того, что между k-й и l-й групнами имеется сдвиг r, вычислим величины

$$\tilde{p}_{\mathrm{k,l,r}} = \prod_{j=1}^{n}\left(\sum_{i=1}^{n} p_k(x_i|y_j) \times p_l(x_{i\oplus r}|y_j)\right) \tag{2.5}$$

Символы $\oplus$, $\ominus$ означают сложение и вычитание по модулю n. Величина $p_{k,l,r}$ есть вероятность фактического распределения числа появлений букв при условии, что имеет место сдвиг r. Здесь не учитывается, что разные y_j соответствуют разным x_i. Значения $p_{k,l,r}$ получаются по формуле Байеса

$$p_{\mathrm{k,l,r}} = \frac{\tilde{p}_{\mathrm{k,l,r}}}{\sum_{s=0}^{n-1}\tilde{p}_{\mathrm{k,l,s}}} \tag{2.6}$$

Глава 3. Уча — учимся

или Моделирование большого компьютера

Глава 4. Мал золотник…

или Компилятор для алгебраического языка