Частина автомата, виділена в метод, володіє наступною семантикою: отримавши вхідний сигнал, автомат виконує деякі дії, починаючи з стану *y*, по завершенні яких повертається у стан *x*. Виконується логіка виконання завдання з наступним поверненням в початковий стан. Саме це і служить підставою для виділення методу.

У результаті перетворення виділяється структурна одиниця автомата – метод, а діаграма, що описує кінцевий автомат, зменшується, що спрощує його розуміння. А виділений метод можна використовувати повторно для зменшення дублювання коду.

Використовуючи описаний метод перетворення виходить схема (послідовність) методів і команд переходів між ними.

Отриману схему в подальшому буде легко виконати перетворення в сценарій тестування – набір тест-кейсів, зібраних в послідовність для досягнення певної мети. Він включає в себе вихідні дані, умови і послідовності виконання дій та очікувані результати.

В. В. Семерков,
студент IV курса,
Харьковский национальный университет радиоэлектроники

ПРИМЕНЕНИЕ АЛГОРИТМОВ ПРИМЕРНОГО СОПОСТАВЛЕНИЯ СЛОВ

Информация представляется в виде набора данных. Современные информационные системы основаны на концепции информации, представленной в виде данных и на концепции алгоритмов, реализованной в виде программного обеспечения. Алгоритмы имеют вспомогательный характер и необходимы для получения, сбора, обработки и преобразования данных. Следовательно, основой информационных систем являются данные и процедуры, организованные в базы данных, адекватно отражающие реалии действительности в той или иной предметной области.

Данный доклад посвящён исследованию методов решения проблемы поиска в базах данных, когда известно произношение текста, но неизвестно как он пишется. Рассматриваются некоторые фонетические алгоритмы. Также описываются метрики похожести текста, используемые в поиске, позволяющем ошибки. Информационная система, использующая фонетические алгоритмы, сможет предложить пользователю правильную выборку данных, даже если он ввёл не совсем корректный запрос для поиска. Данные алгоритмы сопоставляют словам со схожим произношением одинаковые коды, что позволяет осуществлять сравнение множества таких слов на основе их фонетического сходства. Поиск дублирующихся учетных записей, а, следовательно, нечёткое сопоставление записей, является одной из ключевых операций при работе с базами данных персонала крупных организаций [2; 3].

Для того чтобы можно было применять алгоритмы поиска к базам данных, необходимо сначала произвести индексирование баз данных, используя коды, полученные на основании применения фонетических алгоритмов. Рассмотрим некоторые из основных алгоритмов.

Одним из первых был разработан алгоритм Soundex, изобретенный Робертом Расселом и Маргарет Обелл. Его принцип работы основан на разбиении согласных букв на группы с порядковыми номерами, из которых составляется результирующее значение. Первая буква сохраняется, последующие буквы сопоставляются цифрам по таблице (Табл.1). Символы, не представленные в таблице (а это все гласные и некоторые согласные), игнорируются. Смежные символы, или символы, разделенные буквами H или W, входящие в одну и ту же группу, записываются как один. Результат обрезается до 4 символов. Недостающие позиции заполняются нулями [1; 4]. После выполненных процедур остаётся всего лишь 7 тысяч различных вариаций такого кода, что влечет за собой множество совершенно ничем не похожих друг на друга слов, имеющих одинаковый Soundex-код. Таким образом, результат в большинстве случаев включает в себя большое количество «ложноположительных» значений.

Таблица 1
Сопоставление согласных букв с індексами в оригинальном алгоритме Soundex

Согласные буквы	Индексы
B, P, F,V	1
C, S, K, G, J, Q, X, Z	2
D, T	3
L	4
M, N	5
R	6

В улучшенной версии алгоритма (Табл.2), буквы разбиты на большее количество групп. Помимо этого, никакого особого внимания буквам H и W не уделяется, они просто игнорируются. Кроме того, никаких операций с длиной результата не производится – код не имеет фиксированной длины и не обрезается.

Таблица 2
Сопоставление согласных букв с индексами в улучшенном алгоритме Soundex

Согласные буквы	Индексы
B, P	1
F, V	2
C, S, K	3
G, J	4
Q, X, Z	5
D, T	6
L	7
M, N	8
R	9

В качестве альтернативы алгоритму Soundex был разработан алгоритм Metaphone. Указанный алгоритм является более точным, чем Soundex, потому что использует больший набор правил английского произношения [1; 4]. Исходное слово преобразуется с учетом правил английского языка, используя заметно более сложные правила, и при этом теряется значительно меньше информации, так как буквы не разбиваются на группы. Итоговый код представляет собой набор символов из множества {0, B, F, H, J, K, L, M, N, P, R, S, T, W, X, Y}, в начале слова также могут быть гласные из множества {A, E, I, O, U}.

В 2000 году появился улучшенный алгоритм Double Metaphone, который отличается от других фонетических алгоритмов, генерируя из исходного слова не один, а два кода длиной до 4 символов каждый. Первый отражает основной вариант произношения слова, второй – альтернативную версию. Он имеет большое количество различных правил, учитывающих, помимо всего прочего, различное происхождение слов, уделяя внимание восточно-европейским, итальянским и китайским словам.

Алгоритмы нечёткого поиска характеризуются метрикой – функцией расстояния между двумя словами, позволяющей оценить степень их сходства в данном контексте. Наиболее известными метриками являются расстояния Хемминга, Левенштейна и Дамерау-Левенштейна.

Расстояние Хемминга указывает число позиций, в которых соответствующие символы двух слов одинаковой длины различны, т.е. является метрикой только на множестве слов одинаковой длины, что сильно ограничивает область его применения.

Наиболее часто применяемой метрикой является расстояние Левенштейна или расстояние редактирования. Исходный вариант этого алгоритма имеет временную сложность $O(nm)$ и потребляет $O(nm)$ памяти, где n и m являются длинами сравниваемых строк. Расстояние Левенштейна – это минимальное количество правок одной строки, чтобы превратить её во вторую. Под правками подразумеваются три возможные операции: стирание символа, замена символа и вставка символа.

Приведём несколько примеров. Пусть функция levenshtein является функцией, находящей расстояние Левенштейна для двух строк.

levenshtein(‘ABC’,’ABC’) = 0

levenshtein(‘ABC’,’ABCDEF’) = 3

Расстояние Левенштейна позволяет субъективно оценить, насколько строки не похожи друг на друга.

Алгоритм Дамерау-Левенштейна является модификацией алгоритма Левенштейна. В алгоритм Дамерау-Левенштейна добавлена еще одна важная операция – транспозиция, по сравнению с алгоритмом Левенштейна. Транспозиция – это операция по перестановке местами двух соседних символов. Фредерик Дамерау доказал, что 80% ошибок при наборе текста человеком являются транспозициями, что делает эту модификацию намного эффективнее в некоторых случаях.

В докладе были рассмотрены такие фонетические алгоритмы как Soundex, улучшенный алгоритм Soundex, Metaphone, Double Metaphone, а также метрики

расстояния. Большая часть фонетических алгоритмов реализована на множестве языков программирования.

Фонетические алгоритмы ставят в соответствие каждой из строк фонетический код, а затем применяют операции сравнения, обычно основанные на фонетических свойствах, иногда в сочетании с методами вычисления дистанции редактирования [2; 3]. Для использования в конкретной предметной области часто имеет смысл провести оптимизацию выбранных алгоритмов либо использовать их комбинацию. Фонетические алгоритмы находят практическое применение во многих областях науки и информационных технологий: сжатие данных, криптография, распознавание речи, генетика и молекулярная биология.

ЛИТЕРАТУРА

1. Кнут Д. Искусство программирования. Т.3. Сортировка и поиск – М.: Издат. дом «Вильямс», 2003 – 801 с.
2. Цыганов Н.Л. Обзор алгоритмов нечёткого сопоставления записей применительно к задаче исключения дублирования персональных данных – Московский инженерно-физический институт, 2006.
3. Цыганов Н.Л. Проблемы очистки иизбежания дублирования персональных данных с помощью методики нечеткого сопоставления в практике Европейской Организации Ядерных Исследований // Науч.сессияМИФИ-2005.Сб. науч. тр. М.: МИФИ, 2005. Т.12. – 193 с.
4. Википедия. Metaphone – Режим доступа: http://ru.wikipedia.org/wiki/Metaphone