УДК 004.021 **Информационные технологии**

***Макаров Олег Сергеевич**, студент магистратуры,*
ФГБОУ ВО «Национальный исследовательский Мордовский государственный университет им. Н. П. Огарёва», Россия, г. Саранск
***Куляшова Наталья Михайловна**, доцент, канд. физ.-мат. наук,*
доцент кафедры фундаментальной информатики
факультета математики и информационных технологий
ФГБОУ ВО «Национальный исследовательский Мордовский государственный университет им. Н. П. Огарёва», Россия, г. Саранск

АЛГОРИТМЫ РАСПОЗНАВАНИЯ ПОЧТОВЫХ АДРЕСОВ

Аннотация: В статье описываются алгоритмы распознавания почтовых адресов из строк произвольного формата. Алгоритмы базируются на сравнении вводимой информации с адресными объектами из эталонных справочников и не зависят от выбора технологий разработки и базы данных. На основе разработанных алгоритмов было создано программное обеспечение для распознавания адресов на языке программирования C# со степенью ошибки не более 15%.

Ключевые слова: Алгоритмы, почтовый адрес, распознавание, адресные структуры, дерево адресных объектов, сравнение строк, обход дерева.

Abstract: The article describes algorithms for recognizing mail addresses from strings of arbitrary format. The algorithms are based on a comparison of the input information with the address objects from the standard dictionaries and do not depend on development technologies or databases. Based on the developed algorithms, software was created for mail address recognition in the C# programming language with an error rate less than 15%.

Key words: Algorithms, mail address, recognition, address structures, address object tree, string comparison, tree traversal.

Введение

В современном мире очень много приложений сталкиваются с проблемой распознавания почтовых адресов из строк, записанных пользователями в произвольном формате. Программирование таких алгоритмов является нетривиальной задачей для программиста [3], поэтому на рынке информационных технологий присутствуют программные решения данной задачи. Однако сравнительный анализ таких технологий показывает [2, с. 356], что в настоящее время отсутствуют качественные и свободные в использовании программные сервисы для распознавания почтовых адресов. В данной статье описываются принципиально новые алгоритмы построения собственного программного обеспечения для распознавания адресных структур, независимо от технологий для реализации.

Начальные условия

Независимо от применяемых алгоритмов в процессе работы программы для оптимизации скорости выполнения может происходить распараллеливание процесса распознавания по доступным ядрам компьютера. Все алгоритмы заканчивают работу в двух случаях: если были разобраны все символы из входной строки поиска либо в процессе обработки были исключены все варианты распознавания. Алгоритмы работают с деревом адресных объектов, фрагмент которого представлен на рисунке 1.

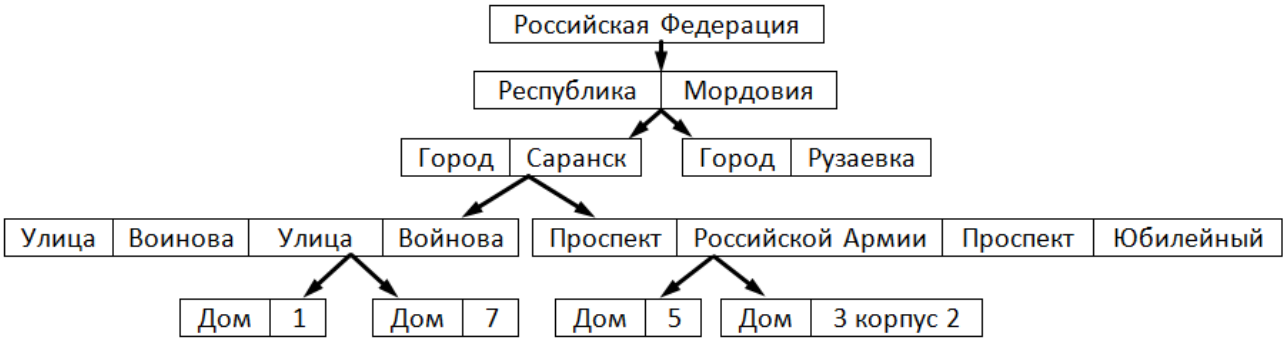

Рисунок 1 – Дерево адресных объектов.

Дерево начинается от самого большого адресного объекта (РФ) и заканчивается листьями-домами. Адресные объекты в узлах дерева формируются из эталонного справочника, который выбирается в зависимости от требований к программному обеспечению. Примерами таких справочников могут открытые данные ФИАС или КЛАДР [1]. Алгоритмы не зависят от регистра символов в строке поиска или наименованиях адресных объектов в дереве.

Алгоритм 1. Нисходящий обход дерева

Алгоритм распознает адреса, записанные в порядке убывания административных единиц. Строка адреса разбивается на отдельные элементы («адресное слово») по разделителям (обычно это пробел и запятая), и затем разбирается слева направо. Одновременно с этим запускается обход дерева адресных элементов, начиная с корневого элемента. В простейшей реализации сравнение адресного слова с адресным элементом в узле дерева осуществляется посимвольно, до тех пор, пока строка наименования адресного элемента не будет полностью содержать в себе подстроку адресного слова либо встретятся несовпадающие символы. Проще всего объяснить работу программы на примере анализа строки «Саранск, Дачная, д. 17».

На первом этапе при поиске подстроки «саранск» в наименованиях адресных объектов программа не найдет соответствия на первом уровне дерева поиска (РФ). Затем программа станет искать среди всех адресных объектов второго уровня (субъекты РФ). На этапе обработки первого символа «с», программа распараллелит поиск по следующим субъектам:

- Санкт-Петербург (город федерального значения);
- Севастополь (город федерального значения);
- Ставропольский (Край);
- Самарская (Область);
- Саратовская (Область);
- Сахалинская (Область);

- Свердловская (Область);
- Смоленская (Область);
- Саха (Республика);
- Северная Осетия (Республика);

Однако после сравнения третьего символа алгоритм вновь не найдет соответствия, и программа продолжит поиск подстроки «саранск» на нижележащих уровнях. Как только программа обнаружит полное вхождение данной подстроки в наименовании какого-либо адресного элемента, параметры этой ветви поиска изменятся. В последующих итерациях для данной ветви программа извлечет следующее адресное слово «дачная» и будет осуществлять поиск среди дочерних элементов данного узла и так далее. Например, для указанной входной строки могут встречаться такие элементы, как «улица Саранская» в Москве или «поселок Саранское» в Калининградской Области. Данные варианты будут находиться в результатах поиска до тех пор, пока не исключаться при сравнении нижестоящих элементов в последующих сравнениях. Если отсеивание каких-либо вариантов не произойдет по мере обработки адреса, они будут включены в конечный результат распознавания адреса с последующей сортировкой по мере удовлетворения входным условиям. Вероятнее всего, «улица Саранская» в Москве исключится, поскольку дочерние элементы этого узла будут представлены домами, а «поселок Саранское» в Калининградской области будет удален из результирующей выборки, потому что не найдутся дочерние адресные объекты, содержащие подстроку «дачная». Если адрес указан максимально полно, без ошибок и сокращений, с указанием всех административных единиц, вероятность того, что он распознается однозначно, значительно выше.

Преимущество алгоритма: высокая скорость вследствие простоты реализации.

Недостаток алгоритма: сильно зависит от порядка записи административных единиц. Адрес, записанный в «американском» стиле (начиная от номера дома, к наименованию субъекта РФ), можно распознавать,

только «перевернув» входную строку, однако распознавание такого адреса не всегда приводит к ожидаемому и корректному результату.

Алгоритм 2

Вычленение адресных объектов. Перед осуществлением поиска из входной строки убираются все префиксы адресных структур (например, «улица», «район» и т. д.). Входная строка делится по разделителям на несколько подстрок, которые формируют массив адресных слов.

На первом этапе работы алгоритма берется первое адресное слово из массива и сравнивается с наименованиями адресных структур во всех узлах дерева поиска. Узлы, в которых сравнение завершилось успешно, отбираются для последующих этапов алгоритма. Они образуют набор отдельных групп конечных адресов с общими родительскими элементами (выше, чем рассматриваемый узел в иерархии дерева поиска), из которых в дальнейшем будут отбираться варианты распознавания адреса. Варианты в такой группе могут только отсеиваться.

На следующей стадии алгоритма запускается цикл по всем оставшимся элементам отсортированного массива адресных слов, начиная со второго, и для каждой группы выполняется следующее:

- если текущий элемент из отсортированного массива совпадает с каким-либо родительским элементом либо его частью, не использованной ранее для сравнения, вся текущая группа адресов переходит в следующую итерацию неизменной;
- если текущий элемент из отсортированного массива совпадает с другой частью текущего элемента, не использованной ранее для сравнения, вся текущая группа адресов переходит в следующую итерацию неизменной;
- если текущий элемент из отсортированного массива совпадает с каким-либо дочерним элементом любого уровня либо его частью, не использованной ранее для сравнения, происходит замена: этот дочерний узел становится узлом группы, и вся группа копируется для участия в следующих итерациях (разветвление алгоритма поиска);

- в противном случае группа отсеивается из результатов поиска.

Когда массив отсортированных элементов будет исчерпан, в результате останутся только адреса, удовлетворяющие критериям поиска.

Преимущество Алгоритма 2: не зависит от порядка записи административных единиц во входной строке.

Недостатки Алгоритма 2: требует большего количества времени на распознавание, не учитывает наименования префиксов административных единиц (улица, район), поэтому имеет большую погрешность распознавания.

Заключение

Вышеприведенные алгоритмы были реализованы на языке программирования C# 8.0 с использованием технологии ASP.NET Core [4] на основе эталонного справочника ФИАС. Исходный код программы расположен на сервере GitHub по адресу https://github.com/bajitumop/AddressRecognizer. Тестирование показало, что программа может корректно распознавать более 85% тестовых адресов.

Библиографический список:

1. База данных ФИАС [Электронный ресурс]. – [Б. м. : Б. и.], [2020]. – Режим доступа: https://fias.nalog.ru/Updates. – Загл. с экрана (дата обращения: 11.07.2020).

2. Макаров О. С. Обзор online-сервисов по формализации адресов / О. С. Макаров // XLVII Огарёвские чтения. – Саранск, 2019. – С. 352–357.

3. Павлюц А. Работаем с почтовыми адресами и их качеством. Как правильно [Электронный ресурс] / А. Павлюц. – [Б. м. : Б. и.], [201–]. – Режим доступа: https://iqsystems.ru/postaddress-do-it-right/. – Загл. с экрана (дата обращения: 10.06.2020).

4. ASP.NET Core – Краткое руководство [Электронный ресурс]. – [Б. м. : Б. и.], [201–]. – Режим доступа: https://coderlessons.com/tutorials/microsoft-technologies/izuchite-asp-net-core/asp-net-core-kratkoe-rukovodstvo. – Загл. с экрана (дата обращения: 25.06.2020).