Редуцируем f_1 относительно $G_0 = \{f_1, f_2, f_3\}$

```
> f4 := NormalForm( f4, G0, plex(a, b, c) );
                        f4 := -b^2 - b c + c^2 - b + c
```

Если остаток не равен 0, добавляем f_4 вG_1= $\{f_1, f_2, f_3, f_4\}$

```
> G1 := [f1, f2, f3, f4];
```

Аналогично предыдущим действиям находим f_5 - зацепление $S(f_2 f_3)$, f_6 - зацепление $S(f_1\ f_3)$, f_7 - зацепление $S(f_1 f_4)$, редуцируем, добавляем к G_1, если остаток не равен нулю.

В итоге получаем базис Грёбнера

```
> G1;
[a b - c^2 - c, a^2 - b c - a, a c - b^2 - b, -b^2
    - b c + c^2 - b + c, 2 b c, -2 c^3 - 2 c^2]
```

Теорема (о количестве конечных решений):

Число решений системы $\langle f_1 \dots, f_n \rangle$конечно тогда и только тогда, когда базис Гребнера идеала $I = \langle f_1 \dots, f_n \rangle$ содержит $f_1 \dots f_n$, старшие члены которых являются степенями переменных $x_1 \dots, x_n$ соответственно.

За конечное число шагов мы получим набор $\{f_1, f_2, f_3, \dots f_n, f_{n+1}, \dots\}$. , f_n где каждое зацепление разрешимо. Это и есть базис Грёбнера идеала $I = \langle f_1 \dots, f_n \rangle$

Тогда базис Грёбнера для данной САУ:$\{f_1, f_2, f_3, f_4, f_5, f_6\}$. Найдем решения из $-2c^3 - 2c^2 = 0 \Rightarrow c = 0$ или $c = -1$. При $c = -1$. $2bc = 0$ $b = 0$. Имеем $a^2 - a - bc = 0 \Rightarrow a^2 = a \Rightarrow a = 0$ или $a = 1$. Наборы (1,0,-1) – не подходят, (0,0,1) – подходят. При $c = 0$, $a^2 = a, a = 0, a = 1, b^2 + b = 0, b = 0, b = -1$. Наборы (0,0,0), (0,-1,0), (1,0,0)- подходят. (1,-1,0) - не подходят.

Ответ: {(0, 0, 0); (1, 0, 0); (0, −1, 0); (0, 0, −1)}.

Список использованных источников:
1. И.В.Аржанцев Базисы Грёбнера и системы алгебраических уравнений
2. https://habrahabr.ru/post/177237/
3. http://halgebra.math.msu.su/wiki/lib/exe/fetch.php/specialcourses:lnd_popovsky.pdf

ПРОГРАММА СБОРА ДАННЫХ О СТРУКТУРЕ ВЕБ-САЙТОВ

Белорусский государственный университет информатики и радиоэлектроники г. Минск, Республика Беларусь

Потехин А.С.

Стержанов М.В. – к.т.н. доцент

В настоящее время всеобщие глобальные тенденции приближаются к тому, что все операции и торговые сделки будут проходить с использованием веб-ресурсов. Для того, чтобы успешно вести бизнес очень важно получать актуальные данные о движения рынка (динамика цен и товаров) и локальные новости, которые порой всецело влияют на формирование спроса, своевременно. Но необходимые данные не всегда легко доступны пользователю и чаще всего они неструктурированны. Рассматривается приложение, которое будет обладать необходимым функционалом для сбора и структурирования данных с различных веб-ресурсов.

Целью исследования, для которого необходим сбор данных из Сети Интернет, является сентимент-анализ данных с различных новостных сайтов. Данные должны содержать полную информацию о новости, включая заголовок, текст, дату и автора новости. Для того, чтобы обеспечить сбор указанной информации, необходимо реализовать инструмент - web-scraper.

В широком понимании web scraping — это сбор данных с различных интернет-ресурсов. Общий принцип его работы можно объяснить следующим образом: автоматизированный код выполняет запросы на целевой сайт и получая ответ, парсит HTML-документ, ищет данные и преобразует их в заданный формат. Т.е. инструменты веб-скрапинга позволяют вручную или автоматически извлекать новые или обновленные данные и сохранять их для последующего использования.

Для того чтобы выполнять эту задачу, инструмент должен поддерживать работу со следующими данными:

HTML, JavaScript, так как большинство сайтов построены с использованием этих технологий;

Plaintext, PDF и другие форматы представления текстовых данных;

URLs, с возможностью построения ни их основе графа веб-ресурсов.

Также инструмент должен обладать требованиями [1],[2],[3]:

- Надежность – Веб содержит ресурсы, которые могут вводить скрапер в бесконечный цикл или недоступные сервисы, ожидать выполнения которых он не должен. Скрапер должен быть устойчивым к таким ловушкам;
- Вежливость – интернет-ресурсы имеют явные и неявные политики, регулирующие частоту, с которой скрапер может посетить их. Они описаны в файле robots.txt и эти политики должны соблюдаться;
- Распределенность – скрапер должен иметь возможность выполняться в распределенном режиме на нескольких машинах;

- Масштабируемость – скрапер должен поддерживать возможность увеличения производительности за счет добавления дополнительных вычислительных узлов, на которых он исполняется;
- Производительность и эффективность – скрапер должен обеспечивать эффективное использование системных ресурсов, включая процессор, память и полосу пропускания сети;
- Качество – скрапер должен уметь отделять спам-страницы от полезных и извлекать последние;
- Актуальность – скрапер должен поддерживать обновление собранных данных;
- Расширяемость – скрапер должен быть модульным, т.е. позволять добавлять новую функциональность, для анализа новых форматов данных, протоколов и т.д.

Помимо описанных общих требований для скраперов, можно обозначить основные требования, для конкретной задачи исследования:

Скрапер должен быть кросплатформенным, чтобы его можно было одинаково настраивать и конфигурировать на вычислительных узлах с разными операционными системами;

Скрапер должен обеспечивать производительность обработки порядка 100 стр/сек, чтобы время сбора описанного выше объема данных составляло часы, а не дни. В том случае если окажется, что данных для сбора и анализа больше предполагаемого, скрапер должен предоставлять возможность легко увеличить его производительность путем выделения ему для работы большего числа потоков или добавления дополнительных вычислительных узлов;

Скрапер должен быть интегрирован с базой данных для хранения собранной информации и полнотекстовым индексом, позволяющим быстро извлекать данные для последующего анализа, отвечающие указанным условиям;

Требуется скрапер для сбора данных в ширину и вертикального поиска, так как в указанной задаче необходимо извлечь информацию о конкретной предметной области, а не узкое множество фактов;

В настоящее время существует множество готовых решений веб-скраперов, но готового решения для данной задачи исследования нет, поэтому, для реализации поставленной задачи был разработан собственный веб-скрапер.

Созданный скрапер является эффективным инструметом для поиска в Вебе, ядро написано на C++ с которым взаимодействует Ruby-оболочка., поддерживает граф связей узлов, различные парсеры, фильтры и нормализаторы URL. Он позволяет использовать различные хранилища данных, такие как Cassandra, Hbase и др. Скрапер также является масштабируемым (до 100 узлов в кластере и легко настраивается и расширяется, в полной мере является “вежливым”.

Списокиспользованныхисточников:

1. PAPAVASSILIOU V., PROKOPIDIS P., THURMAIR G. A modular open-source focused crawler for mining monolingual and bilingual corpora from the web // Proceedings of the 6th Workshop on Building and Using Comparable Corpora. — 2013.
2. AHUJA M.S., BAL J.S., VARNICA Web Crawler: Extracting the Web Data // International Journal of Computer Trends and Technology. — 2014.
3. YADAV M., GOYAL N. Comparison of Open Source Crawlers-A Review// International Journal of Scientific & Engineering Research. — 2015.

КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ ШИФРОВ ЗАМЕНЫ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Сидоренко К.А., Приловский Е.В., Усенко Д.В.
Стройникова Е. Д. – ассистент кафедры информатики

Шифры замены являются наиболее часто используемыми шифрами на сегодняшний день. Они характеризуются тем, что отдельные части сообщения (слова, буквы) заменяются на другие буквы, числа, символы и т. д. Но, при этом замена осуществляется так, чтобы через шифрованное сообщение можно было восстановить передаваемое сообщение. Несмотря на вытеснение шифров подстановки блочными шифрами одноразовые блокноты ещё остаются применимыми на государственном уровне. Они используются для обеспечения сверхсекретных каналов связи. Так, по некоторым данным, телефонная линия между главами США и СССР шифровалась при помощи одноразового блокнота и вполне возможно, что подобные линии существуют до сих пор. Одноразовые блокноты применяются шпионами в различных государствах для сокрытия важной информации. Такие сообщения невозможно расшифровать, если отсутствует ключ, записанный в блокноте, независимо от вычислительной мощности ЭВМ.

Шифр Цезаря

Шифр Цезаря является шифром подстановки, который работает следующим образом: все символы циклически заменяются символами, которые расположены на определенном числе позиций в любом направлении от них в алфавите. Рассмотрим следующий пример: в шифре со сдвигом вправо на 3 происходит замена A на D, B на E,…, Z на C и т. д. (рис.1). Способ шифрования с помощью шифра Цезаря есть составляющая часть более сложных шифров, например такого, как шифр Виженера. Но, т. к. шифр Цезаря является моноалфавитным, его легко разгадать и он не практичен в использовании.