Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Шаламов Роман Александрович

Создание платформы для визуализации алгоритмов YaccConstructor с помощью технологии WebSharper

Курсовая работа

Научный руководитель:
к. ф.-м. н., ст. преп. Григорьев С. В.

Санкт-Петербург
2017

Оглавление

Введение

Возможность наглядно увидеть результат работы какого-либо алгоритма может быть очень полезной для понимания его конструкции и проведения различных тестов. В частности, очень удобно работать с алгоритмами на графах, если данные графы отрисовываются программой.

Существует множество различных алгоритмов, которые можно визуализировать. В частности, алгоритмы проекта YaccConstructor — платформы для исследования и разработки генераторов лексических и синтаксических анализаторов и других алгоритмов для работы с грамматиками. Так как большинство алгоритмов платформы работают с грамматиками, то очень полезной может быть унифицированная технология, позволяющая выполнять отображение данных алгоритмов в графическом виде на экране пользователя.

На данный момент в YaccConstructor уже существуют веб-приложения, выполняющие визуализацию некоторых отдельно взятых алгоритмов. Поскольку создавать для каждого алгоритма новый проект, который бы реализовывал его визуализацию — неэффективно и трудоемко, то было принято решение о создании такого приложения, в которое можно было бы единообразно добавлять алгоритмы платформы, используя готовую библиотеку компонент и модульную архитектуру, упрощающие разработчику процесс визуализации алгоритма.

За прототип данного приложения было решено взять кодовую базу из уже реализованных проектов (YC.BioGraph и YC.GraphParsingDemo), унифицировать её и интегрировать данные проекты как примеры работы платформы.

1. Постановка задач

Целью данной работы является создание единой платформы в виде веб-сайта, которая бы позволила легко реализовывать визуализацию алгоритмов YaccConstructor [3], а также добавление на данный сайт следующих алгоритмов:

- алгоритм синтаксического анализа графов;
- алгоритм поиска подпоследовательностей РНК в метагеномных сборках.

Данные алгоритмы уже представлены в виде отдельных веб-приложений, поэтому они будут интегрированы на веб-сайт с доработкой.

Для достижения данной цели были поставлены следующие задачи:

- создание нового репозитория в YaccConstructor с названием YC.Web с помощью технологии ProjectScaffold [4];
- изучение особенностей фреймворка WebSharper [5];
- интеграция веб-приложений (синтаксический анализ графов и поиск подпоследовательностей РНК) в YC.Web:
  - объединение кодовой базы всех алгоритмов в один проект;
  - создание масштабируемой модульной архитектуры, позволяющей легко добавлять визуализацию новых алгоритмов;
  - создание библиотеки общих компонент:
    * компоненты ввода грамматик/графов;
    * компонента выбора грамматик/графов из файла;
    * компонента отрисовки графов;
    * компоненты вывода;
- добавление на сайт описания проекта YC.Web и описания работы с ним;

- удаление репозиториев старых веб-приложений.

2. Обзор

YaccConstructor — это платформа для исследования и разработки генераторов лексических и синтаксических анализаторов и других алгоритмов для работы с грамматиками. Также в рамках проекта создаются и другие инструменты для работы с грамматиками, которые можно найти в репозитории на GitHub. По большей части проект реализован на языке программирования F#.

Так как платформа YaccConstructor реализована на языке F#, то лучшим выбором для создания веб-приложения было использование фреймворка WebSharper.

WebSharper — это фреймворк и набор инструментов для разработки веб-приложений полностью на языках программирования C# и F# (или на обоих сразу). Он предоставляет как обширные серверные возможности, так и компилятор на JavaScript. Более того, с помощью него можно писать разметку веб-сайтов, используя только F#.

Все вышесказанное означает, что, используя WebSharper, можно создавать весь комплекс программных компонентов, используя один язык программирования, что существенно упрощает процесс разработки.

Как говорилось ранее, некоторые приложения были реализованы по отдельности различными группами разработчиков.

YC.BioGraph — веб-приложение для поиска подпутей в метагеномных последовательностях, которое также визуализирует полученные последовательности на исходном графе.

YC.GraphParsingDemo — веб-приложение, визуализирующее графы и соответствующие им SPPF по введенным грамматикам. Также в нем реализовано выделение минимального пути между двумя вершинами.

3. Основная часть

3.1. Создание репозитория YC.Web

Для унификации кодовой базы алгоритмов и создания модульной архитектуры было необходимо создание общего проекта со всеми необходимыми библиотеками.

Для создания репозитория использовался ProjectScaffold.

ProjectScaffold — инструмент, предоставляющий возможности для начала работы с новом .Net/Mono проектом. Данное решения предоставляет все необходимое для успешной организации файловой системы проекта, возможности для сборки проекта на серверах, автоматическую генерацию технической документации.

В ходе работы был создан репозиторий YC.Web в организации YaccConstrucor, подключены автоматические сборки на серверах Appveyor (для .Net) и Travis CI (для Mono). Также была включена автоматически генерируемая документация к проекту, хостинг которой осуществляется на с помощью серверов GitHub.

3.2. Внешний вид веб-приложения YC.Web

YC.Web представляет собой многостраничный сайт, страницы которого можно раделить на 2 типа: основная страница (рис. 1) и отдельные страницы для каждого из реализованных алгоритмов (рис. 2).

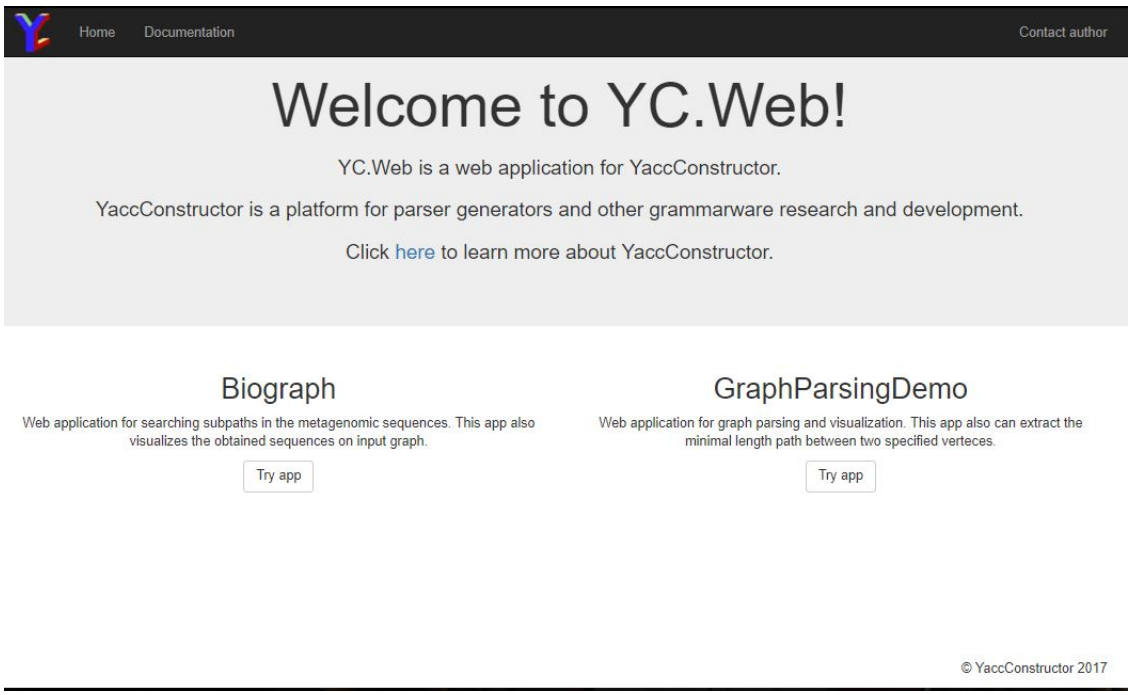

Рис. 1: Главная страница YC.Web

На основной странице присутствует краткое описание проекта YC.Web, полезные ссылки, а также отдельные компоненты для каждого из алгоритмов, которые включают в себя название алгоритма, краткое описание алгоритма и кнопку перехода на страницу алгоритма.

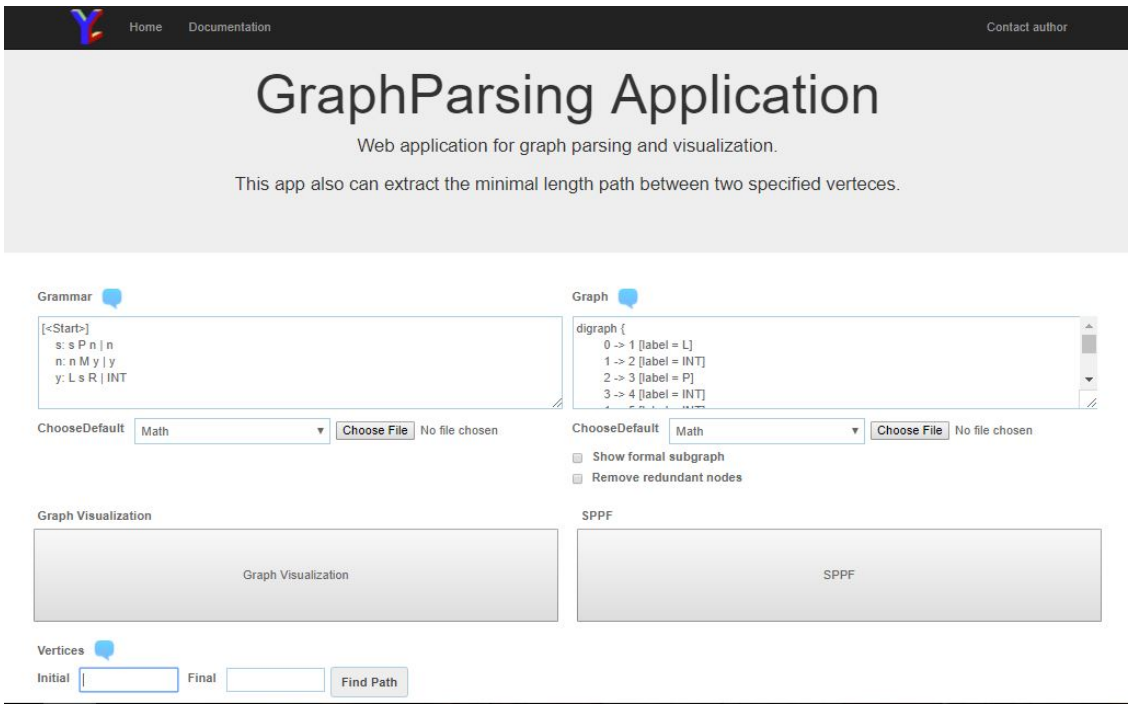

Рис. 2: Пример страницы алгоритма

Страница каждого из алгоритмов включает в себя краткое описание алгоритма и набор из необходимых для визуализации компонент: ввод/вывод, отрисовка графа и т.п.

Для всех страниц в проекте определено общее стилевое решение — компонента Jumbotron библиотеки Bootstrap [1]. Данная компонента также определяет общую шапку для всех страниц веб-приложения, на которой находятся ссылки на главную страницу и документацию к проекту.

3.3. Структура веб-приложения YC.Web

Как уже говорилось ранее, некоторые веб-приложения уже были реализованы как отдельные проекты. Изучив их кодовую базу были выделены следующие особенности, характерные для обоих проектов:

- Все компоненты реализованы с помощью библиотеки WebSharper.Formlets, которая предоставляет набор стандартных веб-компонент (таких как ввод текста и т.д.);
- Компонента отрисовки графа была реализована с помощью JavaScript библиотеки Dracula.js [2].

Так как в каждом из проектов вышеперечисленные технологии использовались исключительно для целей данного проекта, было принято решение реализовать общую библиотеку компонент, которая бы предоставляла возможности для независимого использования одних и тех же компонент разными проектами.

В YC.Web данная библиотека представлена в виде модуля WebComponents, разделенного на два подмодуля: MainPageComponents и AlgorithmsComponents.

Ниже представлена диаграмма данных модулей (рис.3).

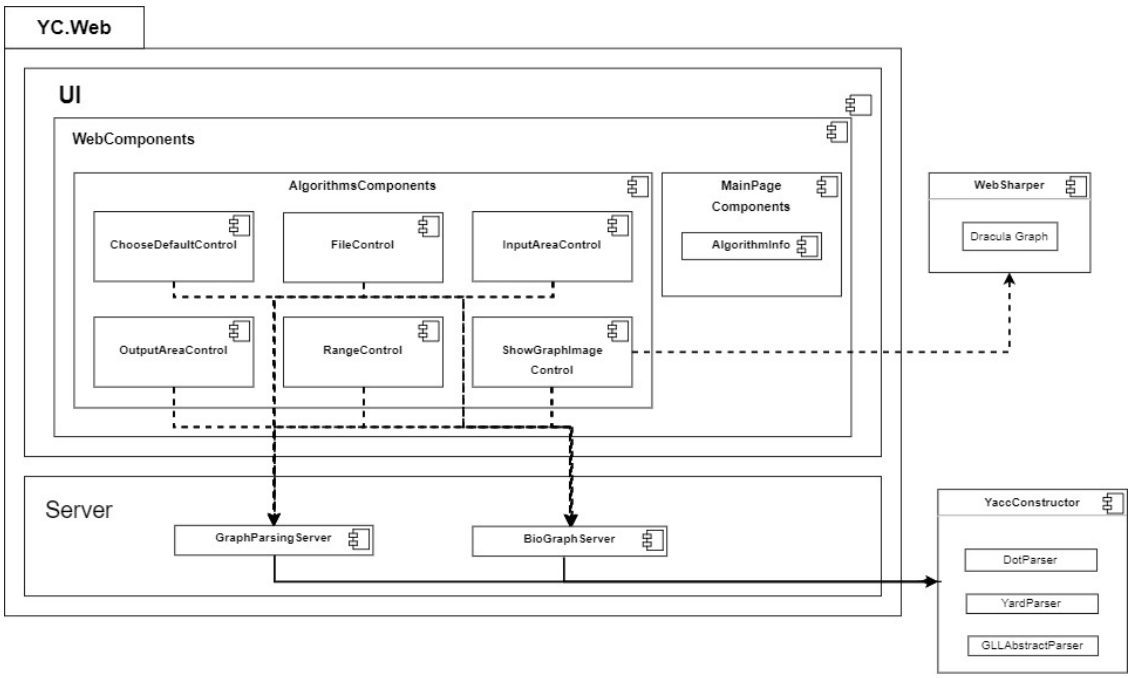

Рис. 3: Диаграмма модулей

AlgorithmComponents — библиотека компонент, используемых для визуализации алгоритмов.

AlgorithmInfo — интерфейс для добавления нового алгоритма на стартовую страницу проекта

Модули *AlgorithmName*Functions и *AlgorithmName*Remote создаются разработчиком в отдельном файле.

*AlgorithmName*Functions — модуль, включающий в себя все необходимые для правильной работы алгоритма функции, выполняемые на сервере.

*AlgorithmName*Remote — модуль, отвечающий за клиент-серверную взаимосвязь между формами, отображаемыми на странице и функциями алгоритма.

Заключение

В ходе данной работы были получены следующие результаты:

- создан новый репозиторий с помощью технологии ProjectScaffold под названием YC.Web;
- изучены особенности фреймворка WebSharper;
- создана модульная архитектура для добавления новых алгоритмов в YC.Web;
- создана библиотека общих компонент для визуализации алгоритмов;
- произведена интеграция следующих веб-приложений: YC.BioGraph, YC.GraphParsingDemo и удаление их старых репозиториев;
- на сайт с документацией было добавлено описание по работе с проектом.

В качестве дальнейшего развития возможны следующие направления:

- добавление разработчиками новых алгоритмов;
- добавление новых компонент для визуализации алгоритмов;
- создание мобильной версии веб-приложения;
- интеграция репозитория с сервисом Appharbor или другим хостинг-сервисом.

Список литературы

[1] Bootstrap. — URL: https://getbootstrap.com (online; accessed: 17.10.2017).

[2] Dracula.js. — URL: https://www.graphdracula.net (online; accessed: 17.10.2017).

[3] Project YaccConstructor. — URL: https://github.com/YaccConstructor/YaccConstructor (online; accessed: 17.10.2017).

[4] ProjectScaffold. — URL: https://fsprojects.github.io/ProjectScaffold (online; accessed: 17.10.2017).

[5] WebSharper. — URL: https://websharper.com (online; accessed: 17.10.2017).