

**РАЗРАБОТКА ДЕМОСТРАЦИОННОГО ПРИЛОЖЕНИЯ ПО
ТЕМЕ "МЕТОДЫ ВЫВОДА В КОНТЕКСТНО-СВОБОДНЫХ
ГРАММАТИКАХ"
DEVELOPMENT OF A DEMO APPLICATION ON THE TOPIC
"DERIVATION METHODS IN CONTEXT-FREE GRAMMARS".**

*Рамизова Карина Бахадуровна
Кириченко Константин Дмитриевич*

Ключевые слова: контекстно-свободные грамматики, LL(1) грамматика, МП-автомат, дерево разбора слова.

В данной работе описывается приложение, предназначенное для демонстрации разбора заданного слова в контекстно свободной LL(1) грамматике. Контекстно-свободные (КС) грамматики являются распространенным способом задания искусственных языков, в частности, языков программирования. Разбор слова в произвольной КС-грамматике в общем случае может быть достаточно трудоемким, поэтому для практических целей выделяют подклассы КС-грамматик, такие как LL(k) или LR(k) грамматики. Наиболее простыми представителями этих подклассов являются LL(1) грамматики, которые часто рассматриваются в учебных целях. Вместе с тем, алгоритмы разбора даже для таких грамматик уже являются достаточно сложными и вызывают затруднения при изучении.

Для КС языков распознавателем является односторонний автомат со стековой внешней памятью – МП-автомат. МП-автоматы имеют стек, в который можно помещать специальные символы, и эти символы будут влиять на переход от одного состояния к другому. Таким образом, конфигурация МП-автомата зависит от текущего состояния автомата, символа входной цепочки и одного или нескольких символов верхушки стека.

Грамматика обладает свойством LL(k), $k > 0$, если на каждом шаге вывода для однозначного выбора очередной альтернативы МП-автомату достаточно знать символ на верхушке стека и рассмотреть первые k символов от текущего положения считывающей головки во входной цепочке символов[1]. В нашем случае рассматриваются LL(1) грамматики, то есть для однозначного определения правила перехода при чтении анализируемой цепочки, находящейся на входной ленте, считывающая головка может заглядывать на один символ вперед.

Работой алгоритма разбора слова LL(1) грамматики управляет управляющая таблица, которая строится путем анализа правил грамматики. Таким образом, построение таблицы разбора слова

происходит по следующему алгоритму: в зависимости от текущей конфигурации, состоящей из состояния входной цепочки, магазина и выходной цепочки, находится соответствующая запись в управляющей таблице и происходит переход в следующую конфигурацию, до тех пор пока не будет достигнута конфигурация, состоящая из пустой цепочки, маркера дна магазина и некоторой выходной цепочки.

По таблице разбора слова несложно построить дерево разбора, которое отображает шаги работы алгоритма разбора. Динамическое построение дерева разбора в ходе распознавания слова позволяет увидеть связи между работой стекового автомата, применением грамматических правил и формальным выводом слова в грамматике

В данной работе реализовывался алгоритм нисходящего разбора для LL(1) грамматик. Для реализации демонстрационного приложения были выделены следующие этапы работы:

- написание алгоритмов чтения и анализа введенных пользователем грамматик;
- реализация веб-приложения;
- добавление графического отображения дерева разбора слова

Для построения дерева разбора слова LL(1) грамматики необходимо было реализовать следующие алгоритмы:

- алгоритм определения множеств FIRST и FOLLOW заданной грамматики;
- алгоритм построения управляющей таблицы;
- алгоритм построения таблицы разбора введенного слова;
- алгоритм построения дерева по таблице разбора.

Далее для оформления всех реализованных алгоритмов было создано веб-приложение с серверной частью на технологиях Node.js и MySQL и клиентской частью на Vue.js.

Для графического отображения дерева разбора была выбрана библиотека D3.js, так как она обладает рядом преимуществ перед другими библиотеками.

Разработанное приложение содержит две роли: “Администратор” и “Пользователь”. Пользователь может зарегистрироваться и добавлять, обновлять и удалять разные грамматики в своём профиле. Также он может просмотреть основные характеристики своей грамматики, а именно множества терминальных и нетерминальных символов, множества FIRST и FOLLOW. Далее пользователь может ввести слово в поле для ввода, после чего строится таблица разбора данного слова и

выводятся 2 вида деревьев разбора: интерактивное дерево разбора и дерево разбора, отображающее шаги построения.

Также реализована обработка ошибок построения таблицы разбора, а именно: “неожиданный символ”, “ожидался другой символ”, “неожиданный конец ввода” и “неожиданные символы в конце”. При возникновении одной из этих ошибок в процессе построения таблицы разбора, таблица разбора всё равно отображается, но только до момента появления ошибки, а деревья разбора слова не отображаются.

Список литературы

Гордеев А. В., Молчанов А. Ю. Системное программное обеспечение. СПб.:Питер : 2001. 736 с.