ИСПОЛЬЗОВАНИЕ ДИНАМИЧЕСКОЙ СТРУКТУРЫ ИЗ КОНЕЧНЫХ АВТОМАТОВ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПОИСКА ШАБЛОНА В ТЕКСТЕ

Савёнок В.А., Медведев С.А.

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

Медведев С.А. – к.т.н.,доцент

Одной из центральных задача в области обработки текстовой информации является задача поиска шаблона в тексте. На сегодняшний день существует множество подходов и алгоритмов для решения данной задачи. Одним из таких подходов является использование конечного автомата. В данной работе представлен оптимальный по трудоемкости подход к построению эффективной по памяти структуры конечных автоматов для решения задачи поиска шаблона в тексте.

Классическим подходом к решению задачи поиска шаблона в тексте является использование конечного автомата. Данный подход хорошо себя зарекомендовал при работе с регулярными и контекстно-независимыми грамматиками [1]. Такие автоматы содержат относительно небольшое число состояний и позволяют сопоставлять достаточно простые шаблоны, в которых применяются такие выражения, как последовательность, вариация и повторение. В то же время, выявление шаблонов, зависящих от контекста, требует проектирования более сложного автомата, число состояний в котором по отношению к сложности шаблонного выражения растет экспоненциально [2].

Для упрощения процесса разработки автомата предложена декомпозиция на простейшие автоматы, объединенные в многоуровневую динамическую структуру. Пример соответствия между деревом выражения поиска шаблона *#P = (“Microsoft”_ “acquires”) .. [0-3] .. {“Google”, “Amazon”};* и структурой автоматов, создаваемой для сопоставления последовательности текстовых лексем, представлен на рисунке 1.

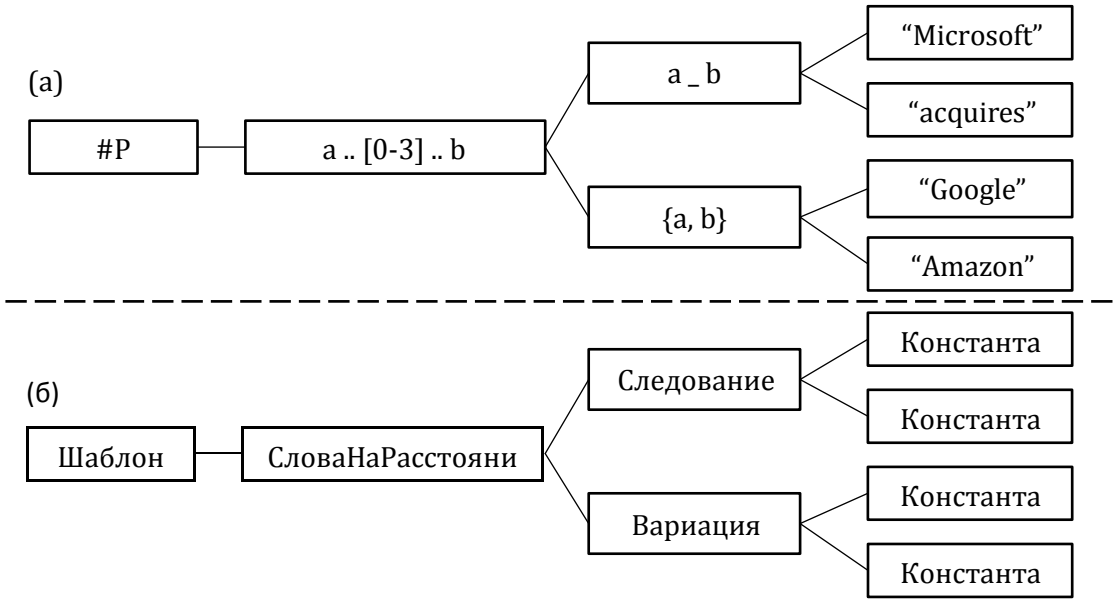

Рисунок 1 – Соответствие между деревом выражения поиска шаблона (а) и структурой автоматов (б)

В разработанной структуре каждый элемент выражения поиска представляется отдельным конечным автоматом с небольшим числом состояний. Такой подход в виде декомпозиции позволяет строить многоуровневые структуры для реализации поиска сложных шаблонов.

Особенность данной структуры заключается в том, что она формируется и модифицируется по мере совпадения отдельных частей выражения шаблона. Автоматы верхнего уровня не будут созданы до тех пор, пока не отработают все автоматы нижнего уровня, что снижает накладные

расходы на проверку шаблонов, которые позже не совпадут. Таким образом повышается эффективность использования памяти и снижается нагрузка на сборщика мусора в языках с автоматическим управлением памятью. В сочетании с событийной моделью взаимодействия автоматов разных уровней, указанный подход позволяет значительно упростить общую структуру поискового движка.

Список использованных источников:

1. John E. Hopcroft and Jeffrey D. Ullman Introduction to Automata Theory, Languages, and Computation / John E. Hopcroft [et al.] // Addison-Wesley, 1979. – P. 217.

2. А.С. Морозов Лекции по конечным автоматам и автоматным структурам [Электронный ресурс]. – Режим доступа: http://math.nsc.ru/~asm256/TA/FANew.pdf. Дата доступа: 20.03.2020.