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

Савёнок В. А.
Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: savionak@gmail.com

В данной работе представлено уточнение динамической структуры из конечных автоматов для решения задачи поиска шаблона в тексте. Приведено краткое описание структуры и подробно рассмотрен ее динамический аспект. Описана общая реализация отдельных узлов; представлены схемы переходов конечных автоматов, соответствующих каждому типу узла.

Введение

Классическим подходом к решению задачи поиска текста по шаблону является использование конечного автомата. Данный подход широко применяется при работе с регулярными и контекстно-свободными грамматиками [1]. Соответствующие им автоматы содержат небольшое число состояний и позволяют сопоставлять достаточно простые шаблоны, в которых применяются выражения вида «последовательность», «вариация» и «повторение». В то же время, выявление шаблонов, зависящих от контекста, требует проектирования автомата, число состояний в котором по отношению к сложности шаблонного выражения растет экспоненциально [2].

Для упрощения процесса разработки таких автоматов в работе [3] предложена декомпозиция на простейшие автоматы, объединенные в многоуровневую динамическую структуру. В текущей работе подробно описан динамический аспект указанной структуры, а также рассмотрена реализация ее отдельных узлов.

Динамическая структура из конечных автоматов

Рассмотрим пример динамической структуры автоматов для шаблона P = ("Microsoft"_-"acquires") .. [0-3] .. {"Google "Amazon"}, изображенный на рисунке 3. В данной структуре автомат, соответствующий отдельному оператору, называется автоматом-оператором. Поиск совпадения литералов также является операцией, для которой создаются автоматы-операторы. Корневому узлу выражения ставится в соответствие автомат-кандидат совпадения шаблона.

Формируемая структура является динамической и изменяется по мере обработки событий. Проследим данный аспект на примере поиска приведенного шаблона в заданной последовательности лексем: «Microsoft», Space, «acquires», Space, «division», Space, «of», Space, «Google», «.».

Представленный шаблон P начинает совпадать на первой лексеме. При этом создается только два связанных автомата: кандидат и оператор для литерала. После подтверждения совпадения автомат литерала «Microsoft» переходит в заключительное состояние и завершает работу. Осуществляется поиск следующего оператора по дереву выражений, выбирается оператор «последовательность слов» и создается автомат, который привязывается к кандидату. Таким образом, после обработки первой лексемы структура состоит из двух автоматов: кандидата и одного оператора.

При поступлении второй лексемы вызывается текущий автомат-оператор. По лексеме Space принимается решение о продолжении совпадения, но ввиду того, что конструкция не совпала полностью, оператор останется активным.

Лексема «acquires» соответствует литералу в правой части текущего выражения, поэтому для нее создается автомат, который сразу вызывается оператором «последовательность слов». Лексема совпадает. Далее производится поиск следующего оператора по дереву выражений и обнаруживается промежуток в словах. Создается автомат, которому сразу передается событие о совпадении левой части выражения. Поиск в оставшейся последовательности происходит аналогичным образом с постоянной сменой задействованных автоматов.

Таким образом автоматы-операторы верхнего уровня не будут созданы до тех пор, пока не отработают все операторы нижнего уровня. Этот подход снижает накладные расходы на проверку шаблонов, которые позже не смогут продолжить совпадение.

Узлы динамической структуры из конечных автоматов

Автоматы-операторы делятся на два типа: базовые операторы и обработчики отмены совпадения. Диаграмма переходов для базового оператора приведена на рисунке 1.

После создания автомат-оператор находится в состоянии совпадение – Matching. В результате обработки очередного события автомат мо-

жет остаться в том же состоянии либо перейти в одно из заключительных состояний: окончательное совпадение - Final match или отмена совпадения - Rejected.

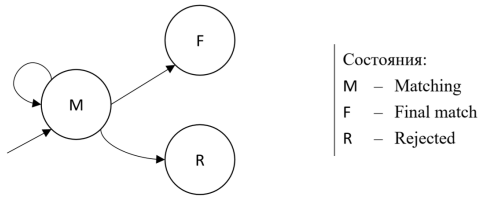

Рис. 1 – Диаграмма переходов базового автомата оператора

Автомат переходит в состояние Final match, если найдено совпадение всего выражения. В этом случае производится поиск вызывающего автомата верхнего уровня по дереву выражений. Если текущий, вызываемый автомат является прямым потомком вызывающего, то последнему передается событие о завершении работы вложенного автомата.

Автомат переходит в состояние Rejected, если не найдено совпадения для следующей лексемы. В такой ситуации генерируется событие отмены совпадения, которое передается ближайшему обработчику в цепочке вызовов автоматов. В динамической структуре обработчиком отмены на самом верхнем уровне выступает кандидат.

Для обработки отмены вводится дополнительное не заключительное состояние неокончательное совпадение – Non-Final match. Автомат переходит в данное состояние в том случае, если обнаружен недостаток данных для принятия решения о совпадении. Таким образом, диаграмма переходов для автомата-обработчика отмены имеет вид, представленный на рисунке 2.

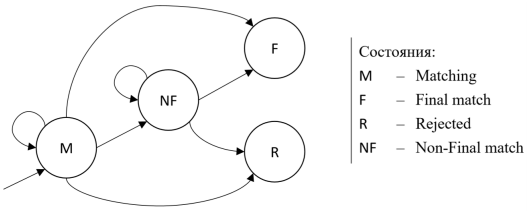

Рис. 2 – Диаграмма переходов автомата обработчика отмены

Автомат переходит в состояние Non-Final match, если вложенные автоматы завершили проверку совпадения, но хотя бы один из них ожидает прохождения дополнительных проверок в контексте поиска совпадений. К таким операторам относятся контекстные операторы, промежуток в словах и исключения.

При переходе в состояние Non-Final match производятся те же действия, что и при окончательном совпадении, однако, автомат продолжает ожидать события об отмене или окончательном совпадении. При этом он перестает быть вложенным относительно вызывающего его автомата и переходит в подчинение главному автомату системы, что позволяет ему получать ожидаемые события.

Выход из состояния Non-Final match осуществляется либо при прохождении всех дополнительных проверок как вложенными операторами, так и самим обработчиком отмены, либо в случае, когда хотя бы один вложенный автомат или сам обработчик не прошел дополнительные проверки на совпадение.

Заключение

Предложен способ реализации элементов динамической структуры из конечных автоматов для решения задачи поиска шаблона в тексте. Каждый элемент представлен отдельным конечным автоматом с небольшим числом состояний, что позволяет строить оптимальные по памяти многоуровневые структуры для реализации поиска сложных шаблонов. Формирование и модификация такой структуры происходит по мере совпадения отдельных частей выражения шаблона, что снижает накладные расходы при сопоставлении шаблонов.

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.
3. Савёнок, В. А. Использование динамической структуры из конечных автоматов для решения задачи поиска шаблона в тексте / В. А. Савёнок // Компьютерные системы и сети: материалы 56-й научной конференции аспирантов, магистрантов и студентов, Минск, 18 – 20 мая 2020 г. / БГУИР. – Минск, 2020. – С. 95-96.

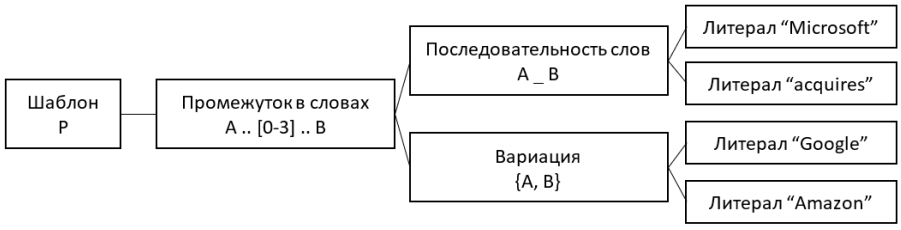

Рис. 3 – Пример структуры из автоматов для дерева выражений