

## ПРИМЕНЕНИЕ SWITCH-ТЕХНОЛОГИИ ДЛЯ РЕШЕНИЯ КЛАССИЧЕСКИХ ЗАДАЧ РАСПОЗНАВАНИЯ ЦЕПОЧЕК СИМВОЛОВ

А.А.Шальто, Н.И.Туккель

Государственное унитарное предприятие  
"Научно-производственное объединение "Аврора",  
Санкт-Петербургский государственный институт  
точной механики и оптики (технический университет)

Известно [1,2], что теория построения компиляторов базируется на иерархии моделей теории автоматов. При этом для задач распознавания цепочек символов с усложнением языка усложняется также модель применяемого автомата.

Рассмотрены три классические [2] задачи распознавания цепочек для:

- скобок произвольной глубины;
- языка  $\{1^n 0^n \mid n \geq 0\}$ ;
- языка  $\{1^n 0^n 1^n \mid n \geq 0\}$ .

Несмотря на то, что на первый взгляд кажется, что первая задача сложнее второй, это не так. В первой задаче проверяется только парность скобок, а во второй аналогичная задача решается после определения правильного порядка символов.

Распознаватели, построенные на базе конечных автоматов, являются автоматами без выхода и позволяют, в частности, решить задачу распознавания четности числа символов в цепочке. Однако даже возможность формирования выходных цепочек в классической модели конечного автомата не позволяет на ее основе решить указанные задачи. При этом первые два языка, задаваемые контекстно-свободными грамматиками, требуют использования модели автомата с магазинной памятью, а для третьего языка, задаваемого контекстно-зависимой грамматикой, необходим переход к линейно-ограниченным автоматам [2].

В [3] на основе теории автоматов была предложена SWITCH-технология для программной реализации алгоритмов логического управления. При этом каждый автомат имел несколько параллельных двоичных входов и выходов и мог принадлежать к одному из следующих классов автоматов: автомат без выходов, автомат Мура, автомат Мили, смешанный автомат. В отличие от [2], где применяются таблицы переходов, в [3] в качестве основного способа задания автоматов используются графы переходов.

Ограниченность применяемой в [3] модели конечного автомата для "реактивных" (событийных) систем привела к ее расширению [4]. При этом вместо двоичных входов и выходов было предложено использовать входные и выходные воздействия. Входные воздействия могут быть событиями и входными переменными, а выходные воздействия – действиями, которые могут выполняться в вершинах, а также на дугах и петлях графа переходов. Для обеспечения универсальности входные переменные и действия реализуются функциями, число и вид которых не фиксировано, а определяется схемой связей [3], содержащей автомат и объекты управления.

В первых двух рассматриваемых задачах в качестве объектов управления выступают счетчик, заменяющий магазин, и два индикатора "Допустить" и "Отвергнуть". Счетчик может быть обнулен, а его значение – увеличено или уменьшено на единицу. С выхода счетчика на вход автомата поступает информация о том, равно ли значение счетчика нулю.

В третьей задаче в схему связей введен дополнительный объект управления – переменная, запоминающая значение счетчика. На вход автомата дополнительно поступает информация о том, что значение счетчика равно  $n$ .

Это позволило решить все три перечисленные выше задачи на основе одной и той же модели автомата, предложенной в SWITCH-технологии.

В рамках предлагаемой технологии преобразование графа переходов в текст программы выполняется формально и изоморфно. Отметим, что граф переходов автомата Мили может быть реализован одной конструкцией switch. Если на всех дугах, входящих в некоторую вершину графа, присутствуют одинаковые действия, то они могут быть перенесены в нее. При этом автомат Мили преобразуется в более компактный смешанный автомат, который, однако, реализуется двумя конструкциями switch.

В классической теории трансляторов в описание автомата попадают действия, выполняемые с входной лентой. В предлагаемом подходе работа с лентой может быть заменена работой с массивом, обрабатываемым в цикле, телом которого является функция, реализующая автомат.

Первая задача была решена двумя способами. Первый из них базируется на применении автомата без памяти, для которого может быть построен граф переходов с одной вершиной и четырьмя петлями. Это решение совпадает с приведенным в [2] с точностью до работы с лентой и замены магазина счетчиком. По мнению авторов, более естественным является второй способ, при использовании которого распознавание выполняется не на петлях, а в соответствующих состояниях смешанного автомата: "Начальное", "Анализ", "Допустить", "Отвергнуть".

Вторая задача также была решена двумя способами. Первый из них базируется на применении автомата Мили с двумя состояниями, в котором отсутствует достижимость начального состояния. Это решение совпадает с приведенным в [2] с точностью до работы с лентой и замены магазина счетчиком. Второе решение основывается на применении смешанного автомата с пятью состояниями: "Начальное", "Считать `1`", "Считать `0`", "Допустить", "Отвергнуть".

Третья задача, которая не может быть решена с использованием магазинного автомата, реализуется смешанным автоматом с шестью состояниями: "Начальное", "Считать `1`", "Считать `0`", "Повторно считать `1`", "Допустить", "Отвергнуть".

Кроме того, на основе излагаемого подхода была решена задача распознавания цепочек символов языков:  $\{0^n 1^n \mid n \geq 1\}$ ,  $\{0^n 1^n \dots 0^n 1^n 0^n \mid n \geq 1\}$ ,  $\{1^n 0^n \dots \mid n \geq 1\}$  и  $\{1^n 0^n \dots 1^n 0^n 1^n \mid n \geq 1\}$ . Усложнение этой задачи по сравнению с предыдущей привело к дальнейшему увеличению числа объектов управления в схеме связей автомата. Новыми объектами в этой схеме являются переменные a и b, определяющие порядок следования единиц и нулей. Эта задача реализуется смешанным автоматом с шестью состояниями: "Начальное", "Вычисление n", "Символы `a`", "Символы `b`", "Допустить", "Отвергнуть".

Предложенные в [4] автоматы могут быть вложенными. Благодаря этому имеется возможность создавать автоматы, объединяющие несколько распознавателей.

В заключение отметим, что в отличие от специфического применения автоматов в теории трансляторов, при использовании предлагаемого подхода автоматы могут применяться при программировании различных задач и, в частности, приведенных в настоящей работе.

## Литература

1. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Том 1. Синтаксический анализ. М.: Мир, 1978.
2. Льюис Ф., Розенкранц Д., Стирнз Р. Теоретические основы проектирования компиляторов. М.: Мир, 1979.
3. Шалыто А.А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998.
4. Шалыто А.А., Туккель Н.И. SWITCH-технология - автоматный поход к созданию программного обеспечения "реактивных" систем //Промышленные АСУ и контроллеры. 2000. N10.