Опубликовано в трудах научно-методической конференции "Телематика'2001", СПб.: СПбГИТМО, 2001, С.174-176

ИСПОЛЬЗОВАНИЕ АВТОМАТНОГО ПОДХОДА ДЛЯ РЕАЛИЗАЦИИ ВЫЧИСЛИТЕЛЬНЫХ АЛГОРИТМОВ

А.А.Шалыто, Н.И.Туккель, М.А.Казаков
Государственное унитарное предприятие
"Научно-производственное объединение "Аврора"",
Санкт-Петербургский государственный институт
точной механики и оптики (технический университет)

В [1] была предложена парадигма автоматного программирования, в рамках которой построение и реализация алгоритма выполняется в виде конечного автомата, для которого базовым является понятие "состояние". В указанной работе данный подход был использован применительно к системам логического управления, в которых входные и выходные воздействия являются двоичными переменными.

В [2] был разработан автоматный подход для программной реализации "реактивных" систем, в которых в качестве входных воздействий добавлены события, а входные переменные и выходные воздействия реализуются функциями.

Для указанных классов систем выделение состояний является естественным, так как они в значительной мере определяются физическими состояниями объектов, которыми управляют эти системы [3].

Для вычислительных алгоритмов выделение состояний не столь естественно, однако и для этого класса задач автоматный подход может быть применен [4].

Такой подход позволяет унифицировать процесс построения структурированных программ с визуализацией их состояний, что имеет важное значение особенно для целей обучения [5]. Это не удается обеспечить при традиционном процедурном подходе, так как он в качестве базового использует понятие "действие", а не "состояние".

Программы, указанного класса, могут быть названы автоматными.

Психологические трудности, возникающие при непосредственном построении алгоритмов для вычислительных задач с помощью автоматов, связаны, в частности, с непривычной реализацией циклов. Эти трудности привели к тому, что в [6] автоматный подход был использован только для построения одного из алгоритмов поиска подстрок, в то время как для десятков других алгоритмов, описанных в этой книге, автоматы не применялись ни при алгоритмизации, ни при реализации.

Однако как бы ни была в рамках процедурного программирования построена программа, всегда имеется возможность ее преобразования в автоматную.

Рассмотрим случай, когда автоматный подход используется только для преобразования программы, построенной традиционным путем. При этом преобразование такой программы в автоматную будем проводить в несколько этапов.

На первом этапе по программе строится ее схема, обычно называемая "блок-схема" или "граф-схема" [7]. Если такая схема была построена при проектировании программы (что бывает крайне редко), она может быть использована на этом этапе.

На втором этапе, используя метод, предложенный в [7] для аппаратной реализации схем алгоритмов, в схему программы в зависимости от выбранного для ее замены типа автомата (Мура или Мили) вводятся символы, в которых индексы соответствуют номерам состояний автомата.

При этом для автоматов обеих типов начальная и конечная вершины схемы обозначаются символом S_0. При построении автомата Мура i-я группа соединенных последовательно операторных вершин, которая может состоять также и из одной вершины, обозначается символом S_i. При построении автомата Мили соответствующим символом обозначается точка, следующая за последней из последовательно соединенных операторных

вершин группы, причем указанные точки для различных групп могут совпадать. Это приводит к тому, что автомат Мили имеет число состояний, не превышающее их число в эквивалентном автомате Мура.

На третьем этапе в случае построения автомата Мили в его схеме программы за счет дублирования некоторых операторных вершин уменьшается число точек, следующих за операторными вершинами, что (в соответствии с [8]) обеспечивает сокращение числа состояний автомата рассматриваемого типа.

На четвертом этапе в схеме программы выделяются пути между смежными символами, включая пути начинающиеся и оканчивающиеся одним и тем же символом. Для каждого пути выписываются условия перехода и выполняемые действия. На основе выделенных состояний и переходов строится граф переходов автомата выбранного типа.

На пятом этапе в случае построения автомата Мили его граф переходов, если это возможно, упрощается за счет переноса одинаковых действий с дуг, входящих в вершину, непосредственно в нее саму. При этом строится граф переходов смешанного автомата.

На шестом этапе построенный граф переходов изоморфно реализуется с помощью конструкции switch языка Си или ее аналога из других языков программирования.

На седьмом этапе полученный фрагмент программы используется в качестве тела конструкции do-while, условием выхода из которой является нахождение автомата в состояние S_0.

Из изложенного следует, что произвольная схема программы (алгоритма) всегда может быть реализована одной конструкцией switch, являющейся телом конструкции do-while.

Применяя предлагаемый подход для реализации алгоритма поиска максимального значения в заданном одномерном массиве, удается построенную традиционным путем структурированную программу, содержащую один цикл for,

```
void main ()
{
 enum    { n = 8 } ;
 int     a[n] = { 44, 55, 12, 42, 94, 18, 6, 67 } ;
 int     max = a [0] ;
 int     i = 0 ;

 for( i = 1 ; i < n ; i++ )
  if( a[i] > max )
    max = a[i] ;
}
```

преобразовать в автоматную программу, состоящую из одной конструкции do-while, телом которой является конструкция switch, реализующая автомат Мили с двумя состояниями:

```
void main ()
{
 int     y0 = 0 ;
 enum    { n = 8 } ;
 int     a[n] = { 44, 55, 12, 42, 94, 18, 6, 67 } ;
 int     max = a[0] ;
 int     i = 0 ;

 do
 {
  switch( y0 )
```

```
    {
     case 0:
       max = a[0] ; i = 1 ;   y0 = 1 ;
     break ;

     case 1:
       if( i >= n )              y0 = 0 ;
       else
       if( a[i] > max )
         { max = a[i] ; i++ ; }
       else
         i ++ ;
       break ;
    }
   }
   while( y0 != 0 )  ;
}
```

Применяя предлагаемый подход для реализации алгоритма Дейкстры, предназначенного для поиска кратчайших расстояний в положительном взвешенном графе от вершины с номером ноль до всех остальных вершин, удается построенную традиционно структурированную программу, содержащую цикл for, в который вложены два таких же цикла, преобразовать в автоматную программу, состоящую из одной конструкции do-while, телом которой является конструкция switch, реализующая автомат Мили с четырьмя состояниями.

Изложенный подход напоминает известный метод Ашкрофта и Манны [9], предназначенный для структурирования неструктурированных программ, который по этой причине к структурированным программам не применяется.

Предлагаемый подход при структурировании неструктурированных программ является более эффективным по сравнению с указанным методом ввиду явного использования автоматов и возможности выбора разных их типов, а для структурированных программ он обеспечивает их преобразование с целью унификации структуры и обеспечения визуализации состояний.

Изложенный подход позволяет использовать не только отдельные автоматы, но и системы взаимосвязанных автоматов. Так, например, алгоритм поиска максимального значения в двумерном массиве может быть реализован автоматной программой, содержащей конструкцию do-while, тело которой реализует автомат Мили с тремя состояниями или систему из двух (по числу вложенных циклов) взаимосвязанных автоматов с двумя состояниями каждый, в которой второй автомат вложен во второе состояние первого.

Эта система автоматов может быть реализована вложенными конструкциями switch [1] или вызовом функции второго автомата, реализованного конструкцией switch, из одной из меток case первого автомата, также реализованного конструкцией switch.

Другие аспекты взаимосвязи схем алгоритмов и программ с графами переходов рассмотрены в [10] (www.softcraft.ru (раздел "Программирование/Автоматные модели")), а также в [1,3].

Литература

1. Шалыто А.А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998.
2. Шалыто А.А., Туккель Н.И. SWITCH-технология - автоматный поход к созданию программного обеспечения "реактивных" систем //Промышленные АСУ и контроллеры. 2000. N10.

3. Шалыто А.А. Логическое управление. Методы аппаратной и программной реализации алгоритмов. СПб.: Наука, 2000.
4. Кузнецов Б.П. Психология автоматного программирования //BYTE/ Россия. 2000. N11.
5. Казаков М.А., Столяр С.Е. Визуализаторы алгоритмов как элемент технологии преподавания дискретной математики и программирования //Телематика 2000. Тез. докл. Международной научно-метод. конф. СПб.: СПбГИТМО (ТУ), 2000.
6. Кормен Т., Лейзерсон Ч., Ривеста Р. Алгоритмы. Построение и анализ. М.: МЦНМО, 1999.
7. Баранов С.И. Синтез микропрограммных автоматов (граф-схемы и автоматы). Л.: Энергия, 1979.
8. Aschcroft E, Manna Z. The translation of "goto" programm into "while" programm //Proceeding of 1971 IFIP Congress.
9. Йодан Э. Структурное проектирование и конструирование программ. М.: Мир, 1979.
10. Шалыто А.А. Использование граф-схем алгоритмов и графов переходов при программной реализации алгоритмов логического управления //Автоматика и телемеханика. 1996. N6,7.