

**ПРЕДСТАВЛЕНИЕ ФУНКЦИИ ПЕРЕХОДОВ ЛИНЕЙНЫМИ БИНАРНЫМИ ГРАФАМИ
ПРИ ГЕНЕРАЦИИ УПРАВЛЯЮЩИХ АВТОМАТОВ С ПОМОЩЬЮ ГЕНЕТИЧЕСКОГО
ПРОГРАММИРОВАНИЯ**

В. Р. Данилов, А. А. Шалыто

УДК 004.4`242

Аннотация. В настоящей статье предлагается новый метод представления функции переходов управляющего автомата, основанный на линейных бинарных графах. Излагаемый метод представления является развитием метода представления функции переходов деревьями решений. Разработан метод генетического программирования, использующий данное представление. Проведено сравнение метода с прототипом на примере задачи об умном муравье.

Ключевые слова: генетическое программирование, автоматное программирование, бинарные линейные графы.

Abstract. This paper presents a new approach for representation of finite state machine transitions function based on linear binary graphs. The method is further development of representation of transitions by decision trees. A genetic programming method for automata generation using presented representation is described. Method is compared with prototype on an artificial ant problem.

Keywords: genetic programming, automata-based programming, linear binary graphs.

1. Введение. Генетическое программирование [1] – метод автоматической генерации программ на основе эволюционных алгоритмов [2], использующий представление программ на высоком уровне абстракции. В работах [3, 4] рассматривалось применение генетического программирования для построения автоматов управления системами со сложным поведением. Автоматы такого рода характеризуются сложностью функции переходов из каждого состояния. Поэтому методы, основанные на представлении функции переходов автоматов полными таблицами, оказываются неприменимыми. Наиболее близким к излагаемому в этой статье подходом является представление функций переходов, основанное на деревьях решений [4]. Недостатком такого подхода является повторное кодирование одинаковых поддеревьев. В настоящей работе предлагается метод представления функций переходов управляющих автоматов с помощью линейных бинарных графов [5].

2. Линейные бинарные графы. Более компактным представлением функций переходов по сравнению с деревьями решений являются разрешающие диаграммы [6]. Каждая из них представляет собой помеченный ациклический ориентированный граф, в котором выделяют вершины двух типов:

- нетерминальные узлы;
- терминальные узлы.

При этом один из нетерминальных узлов является начальным. Все остальные узлы достижимы из него. Из каждого нетерминального узла выходит два ребра. Из терминальных узлов ребер не выходит. Метки в графе расставляются по следующим правилам:

- нетерминальные узлы помечаются названиями переменных;
- терминальные узлы – значениями функции;
- ребра – значениями переменных.

Для определения значения функции по значениям переменных необходимо пройти путь от начального узла до терминального, и сформировать значение, которым помечен этот

терминальный узел. При этом из вершины, помеченной переменной x , переход производится по ребру с пометкой, равной значению указанной переменной. На рис. 1 приведен пример разрешающей диаграммы, реализующей булеву функцию $f = a \oplus b \oplus c$.

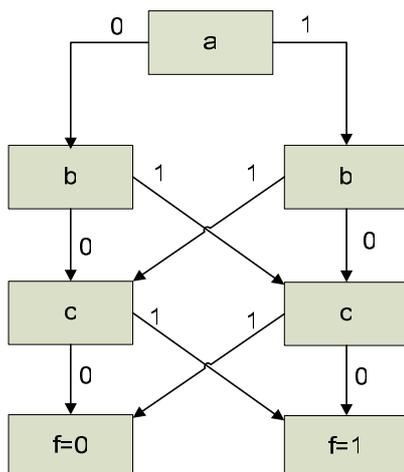


Рис. 1. Пример разрешающей диаграммы

Важным частным случаем разрешающих диаграмм являются линейные разрешающие диаграммы или линейные бинарные графы [5]. Узлы линейного бинарного графа могут быть пронумерованы таким образом, что из каждого нетерминального узла одно из ребер ведет в узел со следующим номером. При этом число элементов, необходимых для реализации булевой формулы бинарным линейным графом, линейно зависит от числа символов формулы.

Пример линейного бинарного графа, реализующего булеву функцию $f = ab \vee c$, приведен на рис. 2.

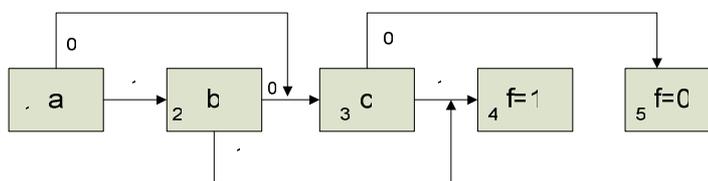


Рис. 2. Пример линейного бинарного графа

3. Представление автомата линейными бинарными графами. Опишем предлагаемое представление автомата с помощью линейных бинарных графов. Для задания автомата необходимо выразить его функции переходов и действия. Осуществим следующее преобразование автомата: вместо задания функций переходов и действия для автомата в целом, определим эти функции в каждом состоянии. Более формально это выглядит следующим образом: зададим для каждого состояния $q \in Q$ функцию $\sigma_q : X \rightarrow Q \times Y$, такую что $\sigma_q(x) = (\delta(q, x), \lambda(q, x))$ для $\forall x \in X$.

Функции σ_q соответствуют функциям переходов и действия в состоянии q . Каждая из таких функций может быть выражена с помощью некоторого линейного бинарного графа. Переменными этих графов являются входные переменные автомата, а множеством значений – все возможные пары (Новое Состояние, Действие). Таким образом, автомат в целом задается упорядоченным набором линейных бинарных графов.

Для использования представления автоматов в виде набора линейных бинарных графов в генетическом программировании необходимо определить *генетические операции над автоматами*. В качестве таких операций будем использовать:

- случайное порождение автомата – в каждом состоянии создается случайный линейный бинарный граф;
- скрещивание автоматов – линейные бинарные графы в соответствующих состояниях скрещиваются;
- мутация автомата – в линейном бинарном графе, который соответствует случайному состоянию, производится мутация.

При этом считается, что число состояний в автомате фиксировано. Поэтому противоречий при выполнении определенных таким образом операций не возникнет.

Теперь определим *представление линейных бинарных графов в виде хромосом*:

- граф представляет собой упорядоченный список узлов. При этом считается, что узлы перечисляются в таком порядке, что из каждого узла одно из выходящих ребер ведет в следующий узел;
- в каждом нетерминальном узле хранится переменная, которой помечен данный узел (переменная расщепления), значение этой переменной, соответствующее переходу в следующий узел, и номер узла, в который производится переход при инверсном значении переменной расщепления;
- в каждом из терминальных узлов хранится значение соответствующей функции переходов.

Описанное представление хромосомы может быть представлено в виде строки. При этом терминальные узлы являются последними узлами линейного бинарного графа. Поэтому возможно отдельно выделить части, кодирующие терминальные и нетерминальные узлы.

Продемонстрируем это на примере. Закодируем в виде строки граф, приведенный на рис. 2. Заметим, что узлы этого графа пронумерованы в требуемом порядке слева направо. Тогда этому графу будет соответствовать следующая строка:

a13b04c15 10,

в которой указанные выше части бинарного графа разделены пробелом.

Определим теперь *генетические операции над линейными бинарными графами*. Для скрещивания линейных бинарных графов можно применить любой из известных методов скрещивания строк, отдельно скрещивая части, описывающие нетерминальные и терминальные узлы. Например, это может быть одноточечный или двуточечный кроссовер [2]. При этом часть, соответствующая терминальным узлам, должна быть скопирована полностью из одной родительской хромосомы. Операция мутации может быть определена следующим образом: выбирается произвольный нетерминальный узел, после этого в него вносится одно из следующих изменений:

- замена переменной, которой помечен модифицируемый узел;
- замена значения переменной, которым помечен переход в следующий в порядке нумерации узел;
- замена номера узла, в который производится переход при несовпадении значения переменной.

4. Апробация. Разработанный подход был протестирован на задаче «Умный муравей-3» [7]. Приведем описание задачи. Муравей находится на случайном игровом поле. Поле представляет собой тор размером 32×32 клетки. При этом муравей видит перед собой некоторую область (рис. 3).

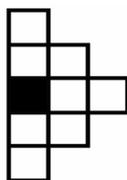


Рис. 3. Видимая муравью область

Еда (яблоки) в каждой клетке располагается с некоторой вероятностью μ . Значение μ является параметром задачи. Игра длится 200 ходов, за каждый из которых муравей может сделать одно из трех действий: поворот налево, поворот направо, шаг вперед. Если после хода муравей попадает на клетку, где есть яблоко, то оно съедается. Целью задачи является построение стратегии поведения муравья, при которой математическое ожидание числа съеденных яблок максимально.

Автомат управления муравьем в этой задаче имеет восемь (число видимых клеток) булевых входных переменных. Каждая из них определяет, есть ли яблоко в клетке, соответствующей переменной.

Для определения эффективности предложенного подхода было выполнено его сравнение с методом представления функций переходов деревьями решений [4] и представлением функций переходов полными таблицами.

Сравнение методов производилось при следующих настройках: стратегия отбора – элитизм [2], функция приспособленности – среднее значение числа съеденных яблок на 200 полях со случайным расположением яблок; для размножения отбираются 25 % популяции, имеющих наибольшее значение функции приспособленности; частота мутации – 2 %; размер популяции – 200 особей; число популяций – 100, $\mu = 0.04$. Игровые поля внутри одной популяции совпадают, поля для различных популяций – различны. Последнее измерение функции приспособленности осуществлялось на случайном наборе из 2000 полей. Результаты эксперимента приведены в таблице.

Таблица. Результаты экспериментов

μ	0.04			
	Функция приспособленности			
Число состояний	2	4	8	16
Таблица переходов	17.18	15.94	15.03	13.68
Деревья решений	18.28	20.28	18.60	20.18
Предложенный метод	18.82	16.44	19.75	15.46

Анализ результатов показывает, что предложенный метод всегда является более эффективным по сравнению с представлением функции переходов полными таблицами, чего и следовало ожидать. Сравнение с прототипом показывает, что предложенный метод конкурентоспособен, так как для некоторых значений числа состояний генерируемого автомата, он обеспечивает более высокое значение функции приспособленности по сравнению с методом представления функции переходов деревьями решений. Авторы предполагают, что изложенный метод работает хуже прототипа только в тех случаях, когда некоторые переходы помечены гораздо более простыми формулами, чем другие. Причина

этого в том, что в отличие от метода представления деревьями решений предложенный метод существенно медленнее упрощает формулы в процессе эволюции. Устранение этого недостатка является предметом дальнейших исследований.

Литература

1. *Koza J.* Genetic programming: On the Programming of Computers by Means of Natural Selection. MA: The MIT Press, 1992.
2. *Гладков Л. А., Курейчик В. В., Курейчик В. М.* Генетические алгоритмы. М.: Физматлит, 2006.
3. *Поликарпова Н. И., Точилин В. Н., Шалыто А. А.* Применение генетического программирования для генерации автоматов с большим числом входных переменных // Научно-технический вестник СПбГУ ИТМО. Выпуск 53. Автоматное программирование, с. 24–42.
4. *Данилов В. Р.* Метод представления автоматов деревьями решений для использования в генетическом программировании // Научно-технический вестник СПбГУ ИТМО. Автоматное программирование. 2008. Вып. 53, с. 103–108.
5. *Шалыто А. А.* Логическое управление. Методы аппаратной и программной реализации. СПб.: Наука. 2000, 780 с.
6. *Bryant R. E.* Graph-based algorithms for boolean function manipulation // IEEE Transactions on Computers. 1986. № 8, pp. 677–691.
7. *Бедный Ю. Д., Шалыто А. А.* Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей». СПбГУИТМО, 2007. <http://is.ifmo.ru/works/ant>

Данилов Владимир Рюрикович — магистрант факультета информационных технологий и программирования Санкт-Петербургского университета информационных технологий, механики и оптики (СПбГУ ИТМО). Область научных интересов: генетическое программирование, автоматное программирование. Число научных публикаций — 5. vdaniloff@rain.ifmo.ru; СПбГУ ИТМО, Кронверкский пр., 49, Санкт-Петербург, 197101, РФ. Научный руководитель – А.А. Шалыто.

Шалыто Анатолий Абрамович — докт. техн. наук., проф.; заведующий кафедрой технологий программирования факультета информационных технологий и программирования Санкт-Петербургского государственного университета информационных технологий, механики и оптики (СПбГУ ИТМО). Область научных интересов: автоматное программирование, генетическое программирование, верификация программ, логическое управление. Число научных публикаций — 300. shalyto@mail.ifmo.ru, <http://is.ifmo.ru>; СПбГУ ИТМО. Кронверкский пр., 49, Санкт-Петербург, 197101, РФ.