УДК 004.3.06:62-507

А. А. Ахи, студент,
А. С. Станкевич, доц.,
А. А. Шалыто, д-р техн. наук, проф., зав. каф.,
Санкт-Петербургский государственный
университет информационных технологий,
механики и оптики
e-mail: akhi@rain.ifmo.ru

Алгоритм построения флибов со 100 %-ной точностью предсказания

Предложен алгоритм построения флиба с минимальным числом состояний, который имеет 100 %-ную точность предсказания очередного значения окружающей среды.

Ключевые слова: *конечный автомат, автомат Мили, флиб*

Введение

В работах [1—3] рассмотрена задача о флибах, которая состоит в том, чтобы построить флиб — детерминированный конечный автомат Мили, предсказывающий значение некоторого битового параметра окружающей среды на основе ранее полученных данных.

В указанных работах рассматривалась автоматическая генерация автоматов с помощью генетических алгоритмов. Вопрос о необходимом и достаточном числе состояний автомата для 100 %-ной точности предсказания не рассматривался, хотя в работе [3] благодаря удачному выбору фитнесс-функции было значительно уменьшено их число по сравнению с другими работами.

В настоящей статье предложен алгоритм построения флиба с минимальным числом состояний, имеющего 100 %-ную точность предсказания очередного значения окружающей среды.

Постановка задачи

Задача состоит в моделировании простейшего существа, способного предсказывать изменения параметра среды, обладающего периодичностью. В качестве простейшей модели такого существа можно использовать конечный автомат Мили. В работе [1] такие конечно-автоматные модели были названы флибами (сокращение от *finite living blobs* — конечные живые капельки). На вход флиба подается переменная, которая принимает одно из двух значений — ноль или единица. Эта переменная соответствует состоянию окружающей среды в текущий момент времени. Рассматривается параметр среды, имеющий лишь два возможных значения. Флиб изменяет свое состояние и формирует значение выходной переменной, принимающей одно из двух указанных значений. Это значение соответствует возможному состоянию среды в следующий момент времени. Задача флиба — предсказать, какое на самом деле состояние окружающей среды наступит в следующий момент времени. Это можно выполнить благодаря периодичности изменений состояний среды.

Размер идеального флиба

Будем считать, что битовая маска s, задающая изменение окружающей среды, не является периодичной: $s \neq s_0^n$ ни для какого $n \geqslant 2$. В этом случае обозначим *zeros* число нулей в маске, а *ones* — число единиц.

Утверждение. Флиб, предсказывающий поведение среды с точностью 100 %, имеет не менее max(*zeros*, *ones*) состояний.

Доказательство. Докажем данное утверждение от обратного. Не умаляя общности, пусть *ones* ⩾ *zeros* и пусть также существует автомат из менее чем *ones* состояний, который имеет точность 100 %. Тогда существуют две единицы, угадав которые, автомат приходит в одно и то же состояние. Пусть $|s| = n$, и эти две единицы оказались на i-й и j-й позициях в маске. Так как автомат оказался в одном и том же состоянии, угадав при этом один и тот же символ, то далее автомат будет себя вести одинаково в обоих случаях, так как флиб все время верно угадывает изменения параметра.

Однако в этом случае получаем, что $s[i + k] = s[j + k]$ при $\forall k \in N$. Следовательно, рассмотренная маска s является периодичной.

Таким образом, получаем противоречие. ◀

Теперь известно, что для 100 %-ной точности требуется хотя бы max(*zeros*, *ones*) состояний. Далее будет показано, как построить автомат, на котором достигается эта оценка.

Построение идеального флиба

Рассмотрим построение автомата на примере строки 1111010010. Здесь *zeros* = 4, *ones* = 6. Поэтому на основе приведенного выше утверждения флиб должен иметь не менее шести состояний. Сначала построим тривиальный автомат из $|s|$ состояний (рис. 1).

В верхнем слое изображены состояния, из которых существует переход только по единице, а в нижнем — только по нулю. Совместим состояния нижнего слоя с состояниями верхнего слоя так,

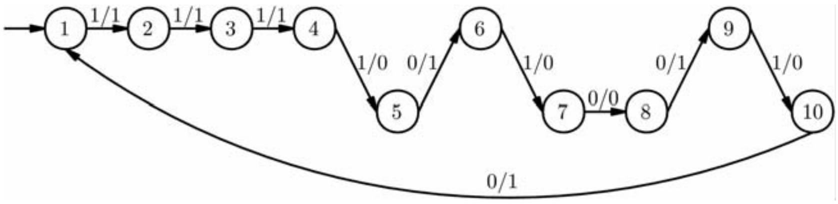

Рис. 1. Тривиальный автомат с |*s*| состояниями

чтобы каждому состоянию из верхнего состояния соответствовало не более одного состояния из нижнего слоя. Ребра из нового состояния будут вести в совмещенные аналоги прежних состояний (рис. 2). Совмещение можно проводить по-разному. На этом рисунке приведены два примера таких совмещений.

Получившиеся автоматы имеют *ones* состояний. С помощью совмещения состояний можно построить автомат для любой маски.

Опишем ***алгоритм*** построения более формально. Пусть $n = \max(zeros, ones)$ — число состояний в автомате. Пронумеруем отдельно нули и единицы в строке. Будем считать, что строка зациклена. Построим автомат таким образом, чтобы i-е состояние отвечало за действия после i-го нуля и/или единицы. Ребра будут строиться по следующим правилам:

1. Если после i-го нуля в строке идет ноль, то проведем из i-го состояния в состояние с номером $i + 1$ ребро с пометкой 0/0.
2. Если после i-го нуля в строке идет j-я единица, то проведем из i-го состояния в состояние с номером j ребро с пометкой 0/1.
3. Если после i-й единицы в строке идет j-й нуль, то проведем из i-го состояния в состояние с номером j ребро с пометкой 1/0.
4. Если после i-й единицы в строке идет единица, то проведем из i-го состояния в состояние с номером $i + 1$ ребро с пометкой 1/1.

Полученный таким образом автомат имеет точность предсказания 100 %.

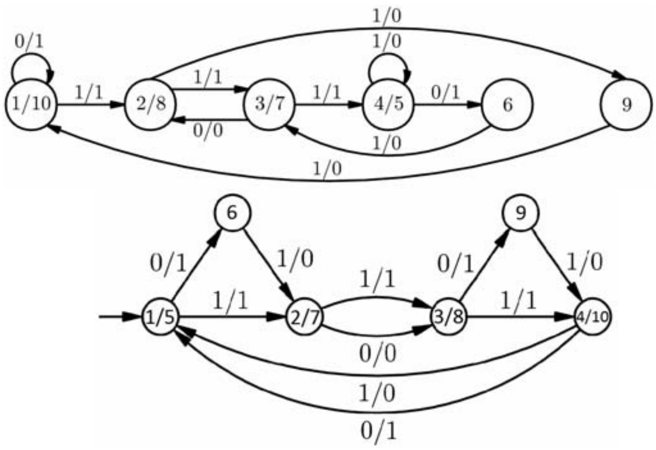

Рис. 2. Два автомата с *ones* состояниями

На рис. 3—7 приведен процесс построения автомата на основе предложенного алгоритма для строки 1111010010.

На рис. 3 показан первый шаг алгоритма — так как после первой единицы в строке стоит вторая единица, то по правилу 4 проводим ребро из вершины 1 в вершину 2 с пометкой 1/1.

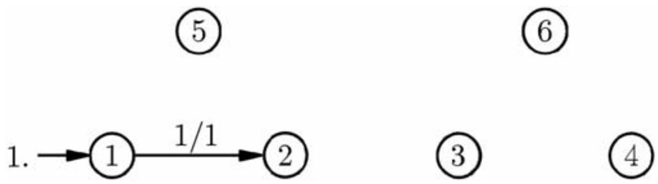

Рис. 3. Первый шаг алгоритма

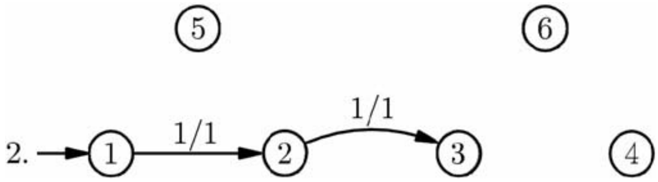

Рис. 4. Второй шаг

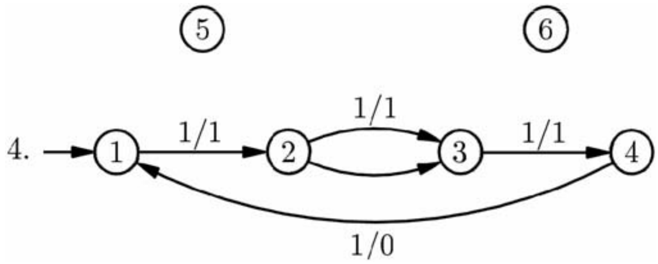

Рис. 5. Четвертый шаг

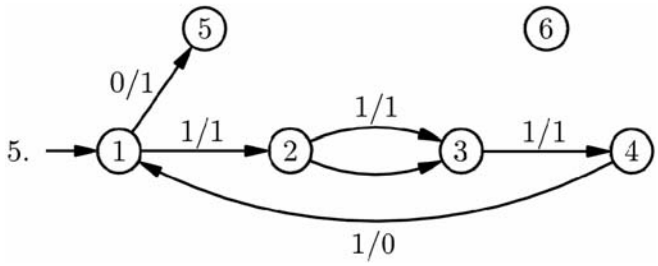

Рис. 6. Пятый шаг

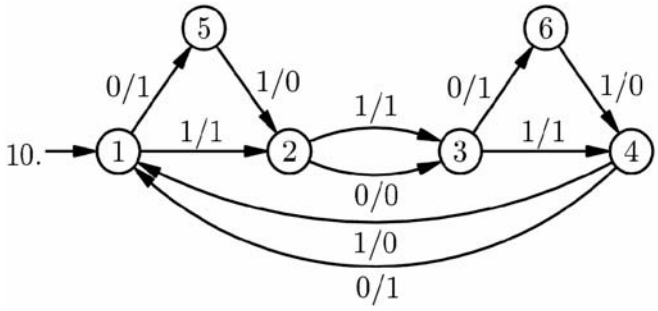

Рис. 7. Десятый шаг

На рис. 4 показан второй шаг алгоритма. При этом, так как после второй единицы в строке стоит третья единица, то по правилу 4 проводим ребро из вершины 2 в вершину 3 с пометкой 1/1.

Третий шаг не приводится, так как он схож с первыми двумя. На рис. 5 показан четвертый шаг алгоритма. При этом, так как после четвертой единицы в строке стоит первый нуль, то по правилу 3 проводим ребро из вершины 4 в вершину 1 с пометкой 1/0.

На рис. 6 показан пятый шаг алгоритма. При этом, так как после первого нуля в строке стоит пятая единица, то по правилу 2 проводим ребро из вершины 1 в вершину 5 с пометкой 0/1.

Шаги с шестого по девятый выполняются аналогичным образом. На рис. 7 показан последний, десятый шаг алгоритма. При этом, так как после четвертого нуля в строке стоит первая единица, то по правилу 2 проводим ребро из вершины 4 в вершину 1 с пометкой 0/1. Искомый автомат построен.

Таблица 1

Алгоритм	Число состояний	Точность предсказания, %
1	20	88
2	20	100
3	12	100
Предложенный	12	100

Таблица 2

Алгоритм	Число состояний	Точность предсказания, %
1	30	87
2	30	97
3	20	100
Предложенный	20	100

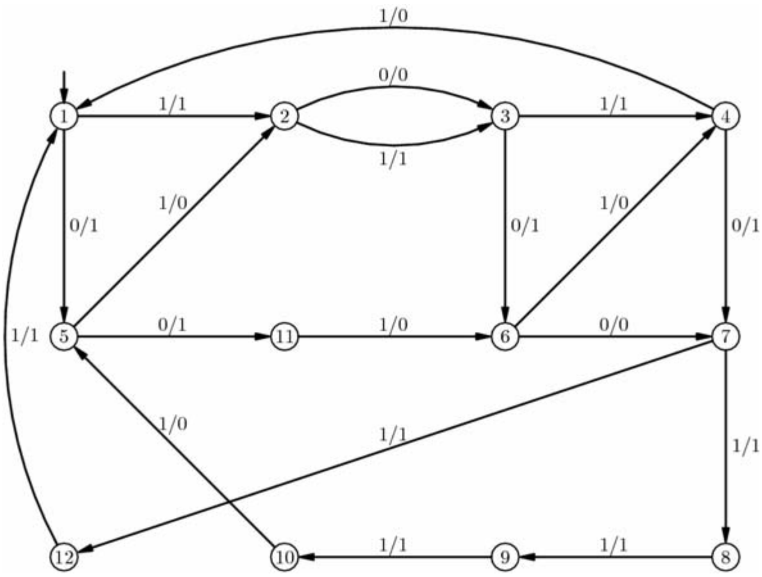

Рис. 8. Автомат с 12 состояниями для строки из 19 символов. Неиспользуемые переходы не изображены

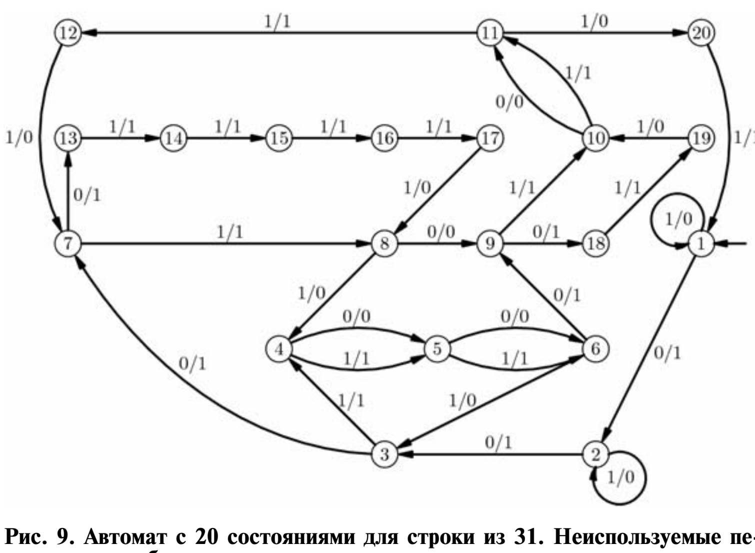

Рис. 9. Автомат с 20 состояниями для строки из 31. Неиспользуемые переходы не изображены

Эксперименты

Ниже приводятся результаты экспериментов, основанных на применении трех алгоритмов (алгоритмы 1—3), изложенных в работах [1—3], и предлагаемого алгоритма.

Эксперимент 1. Битовая маска из 19 символов: 1111010010111101001. Результаты эксперимента приведены в табл. 1.

Автомат, построенный предложенным алгоритмом, приведен на рис. 8.

Эксперимент 2. Битовая маска из 31 символа: 1010111101100011110111110011001.

Результаты эксперимента приведены в табл. 2.

Автомат, построенный предложенным алгоритмом, приведен на рис. 9.

"Слепые" флибы

Описанный выше алгоритм позволяет построить автомат с минимальным числом состояний и точностью предсказания 100 %. Однако остается открытым вопрос о необходимом числе состояний автомата для получения любой другой фиксированной точности. Однако такую задачу можно решить для "слепых" флибов, которая и рассматривается ниже.

Определение и постановка задачи. "Слепой" флиб — автомат Мили, который пытается угадать поведение окружающей среды, однако на вход он получает не текущее состояние окружающей среды, а то значение, которое он выдал на предыдущем шаге.

Получается, что "слепой" флиб всегда считает, что он верно угадал значение битовой переменной и не "смотрит" на окружающую среду. Поведение такого флиба детерминировано. При этом сначала, возможно, флиб выдаст какую-то строку p_0, а затем будет периодически выдавать строку p. Будем называть p — периодом, а p_0 — предпериодом. Для простоты будем считать, что "слепой" флиб не имеет предпериода: $p_0 = \varepsilon$. Задача состоит в том, чтобы для фиксированного числа состояний n и заданной строки s, описывающей поведение окружающей среды, построить "слепой" флиб, который имеет наибольшую точность предсказания.

Решение новой задачи. Из описанного выше следует, что для $n \geqslant \max(zeros, ones)$ максимальная точность составляет 100 % и автомат строится описанным выше алгоритмом (флиб, имеющий 100 %-ную точность ведет себя как "слепой"). Рассмотрим строку p. Заметим, что "слепой" флиб ведет себя в точности как "зрячий" флиб, который смотрит на окружающую среду p и имеет точность 100 %. Из этого следует, что $n \geqslant \max(z_p, o_p)$, где z_p и o_p — число нулей и единиц в p соответственно. Таким образом, получаем, что "слепой" флиб может выдавать любую строку p, содержащую не более n нулей и не более n единиц.

Задача свелась к задаче со строками: построить такую строку p, состоящую из не более n нулей и не более n единиц для того, чтобы величина $\frac{same(s^{|p|}, p^{|s|})}{|s||p|}$ была максимальна. Здесь $same(s^{|p|}, p^{|s|})$ — число совпадающих символов в строках $s^{|p|}$ и $p^{|s|}$.

Решение задачи со строками. Эта задача решается с помощью динамического программирования [4]. Сначала решим эту задачу для фиксированной длины p.

Решение при фиксированной длине. Положим $m = |p|$. Будем вычислять функцию a_{ij} — максимальное число совпадений, которое можно получить, зафиксировав первые $i + j$ символов p, среди которых i нулей и j единиц. При подсчете числа совпадений остальные символы p не рассматриваются.

Для вычисления значений этой функции понадобятся некоторые вспомогательные величины:

- b_i — число нулей, которые встречаются в s^m на позициях с номерами $mk + i$, где $0 \leqslant k < |s|$ — целое число. При этом $b_i = \sum\limits_{k=0}^{|s|} [s^m[mk + i] = 0]$;
- $c_i = |s| - b_i$ — число единиц, которые встречаются в s^m на позициях с номерами $mk + i$.

База динамики — $a_{00} = 0$, $a_{ij} = +\infty \ \forall i, j$: $i \neq 0 \vee \vee j \neq 0$.

Переход осуществляется следующим образом: пусть известно значение a_{ij}, тогда:

- если $i < n$, то $a_{i+1,j} = \max(a_{i+1,j}, a_{ij} + b_{i+j+1})$. Этот переход соответствует попытке приписать к уже имеющейся части p нуль в конец;
- если $j < n$, то $a_{i,j+1} = \max(a_{i,j+1}, a_{ij} + c_{i+j+1})$. Этот переход соответствует попытке приписать к уже имеющейся части p единицу в конец.

В результате $x = \max\limits_{|p| - n \leqslant i \leqslant n} a_{i,|p|-i}$ — максимальное число совпадающих символов, которое можно получить при фиксированной длине p. По результатам вычислений можно получить саму строку p, на которой достигается такой результат.

Решение задачи в общем случае. В общем случае необходимо перебрать все возможные длины p от 1 до $2n$. Среди всех полученных решений для разных длин следует выбрать то, для которого величина $\frac{x(p)}{|p|}$ максимальна.

Воспользовавшись алгоритмом построения идеального флиба для строки p, получим желаемый оптимальный "слепой" флиб. Указанным образом для любого n можно узнать, с какой наибольшей точностью "слепой" флиб с не более чем n состояниями может угадывать поведение окружающей среды, задаваемой строкой s.

Заключение

Для некоторых строк автомат с минимальным числом состояний и 100 %-ной точностью предсказания был построен в работе [3] с помощью генетических алгоритмов. При этом на вычисления уходило много времени и не было гарантии достижения оптимального результата для произвольных строк. Алгоритм, предложенный в этой статье, строит автомат с минимальным числом состояний и 100 %-ной точностью предсказания с существенно меньшими временными затратами. Остается открытым вопрос о необходимом числе состояний автомата для получения любой другой фиксированной точности. Однако для "слепых" флибов эта задача в настоящей работе решена.

Список литературы

1. **Воронин О., Дьюдни А.** Дарвинизм в программировании // Мой компьютер. 2004. № 35. URL: http://www.mycomp.kiev.ua/text/7458
2. **Лобанов П. Г., Шалыто А. А.** Использование генетических алгоритмов для автоматического построения конечных автоматов в задаче о "Флибах" // Матер. 1-й Российской мультиконференции по проблемам управления. Сб. докл. 4-й Всероссийской научной конференции "Управление и информационные технологии" (УИТ-2006). СПб.: Изд. СПбГЭТУ "ЛЭТИ". 2006. URL: http://is.ifmo.ru/works/flib/
3. **Мандриков Е. А., Кулев В. А., Шалыто А. А.** Применение генетических алгоритмов для создания управляющих автоматов в задаче о "Флибах" // Информационные технологии. 2008. № 1. URL: http://is.ifmo.ru/download/2008-02-23_flibs.pdf
4. **Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.** Алгоритмы: построение и анализ. Гл. 15. Динамическое программирование. М.: Вильямс, 2005.