Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Кафедра информатики

А. А. Волосевич

Основы теории

конечных автоматов

Курс лекций

для студентов специальности I-31 03 04 Информатика

всех форм обучения

Минск 2012

Содержание

[1. Определение конечного автомата 3](#_Toc285193445)

[2. Эквивалентность конечных автоматов 5](#_Toc285193446)

[3. Минимизация конечных автоматов 8](#_Toc285193447)

[4. Конечные автоматы Мили и Мура 12](#_Toc285193448)

[5. Автоматные языки 13](#_Toc285193449)

[6. Лемма о накачке 17](#_Toc285193450)

[7. Эквивалентность и минимизация автоматов-распознавателей 18](#_Toc285193451)

[8. Недетерминированные конечные автоматы 20](#_Toc285193452)

[9. Регулярные множества, выражения и языки 25](#_Toc285193453)

[Литература 29](#_Toc285193454)

# 1. Определение конечного автомата

Любую функцию можно рассматривать как некий преобразователь информации. Аргумент функции – входной сигнал – преобразуется, согласно определённому правилу, в результат функции – выходной сигнал. Функциональные преобразователи обладают важным свойством – их поведение не зависит от предыстории. В реальности, однако, имеется достаточно примеров преобразователей, реакция которых зависит не только от входа в данный момент, но и от того, что было на входе раньше, от входной истории. Такие преобразователи называются автоматами.

Даже если число различных входных сигналов конечно, число возможных входных историй бесконечно (счётно). Если автомат по-разному будет себя вести для каждой возможной предыстории, то такой «бесконечный» автомат должен иметь неограниченный ресурс – память, чтобы все эти предыстории как-то запоминать. Введём на множестве предысторий отношение эквивалентности. А именно, две предыстории будем считать эквивалентными, если они одинаковым образом влияют на дальнейшее поведение автомата. Очевидно, что для правильного функционирования достаточно, чтобы автомат запоминал класс эквивалентности, к которому принадлежит данная история.

Определение 1. (Внутренним) состоянием автомата называется класс эквивалентности его входных историй.

Состояние автомата меняется только при получении очередного входного сигнала. При этом автомат не только выдаёт информацию на выход как функцию входного сигнала и текущего состояния, но и меняет своё состояние, поскольку входной сигнал изменяет предысторию.

Случай, когда количество состояний автомата конечно, является простейшим. Соответствующая формальная модель называется конечным автоматным преобразователем информации, или просто конечным автоматом.

Рассмотрим пример конечного автомата. Опишем поведение родителя, отправившего сына в школу. Сын приносит двойки и пятёрки. Отец не хочет хвататься за ремень каждый раз, как только сын получит очередную двойку, и выбирает более тонкую тактику воспитания. Чтобы описать модель поведения отца, используем граф, в котором вершины соответствуют состояниям, а дуга, помеченная , из состояния в состояние проводится тогда, когда автомат из состояния под воздействием входного сигнала переходит в состояние с выходной реакцией . Граф автомата, моделирующего поведение родителя, представлен на рис. 1.



Рис. 1. Автомат, описывающий поведение «умного» родителя.

Этот автомат имеет четыре состояния и два входных сигнала – оценки, полученные сыном в школе: . Начиная с начального состояния (оно помечено особо), автомат под воздействием входных сигналов переходит из одного состояния в другое и выдаёт выходные сигналы – реакции на входы. Выходы автомата будем интерпретировать как действия родителя так:

– «брать ремень»;

– «ругать»;

– «успокаивать»;

– «надеяться»;

– «радоваться»;

– «ликовать».

Определим конечный автомат формально.

Определение 2. Конечным автоматом (автоматом Мили) называется шестёрка объектов , где:

– конечное непустое множество (множество состояний);

– конечное непустое множество входных сигналов (входной алфавит);

– конечное непустое множество выходных сигналов (выходной алфавит);

– начальное состояние;

– функция переходов;

– функция выходов.

Кроме представления в виде графа, автомат допускает задание в виде таблиц функции переходов и функции выходов.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Реализуется конечный автомат либо аппаратно, либо программно. В последнем случае можно использовать шаблон проектирования State.

## Вопросы и упражнения.

Реализуйте предлагаемые конечные автоматы. Представьте графическую реализацию, табличную и программную.

1. Лифт работает в двухэтажном доме и может выполнять команды: «открыть двери», «закрыть двери», «подняться на второй этаж», «спуститься на первый этаж». Изначально лифт находится на первом этаже, его двери открыты. Построить автомат, выполняющий программу для лифта. Выходные значение – новое состояние или признак некорректности программы.

2. Построить автомат для выдачи кофе. Кофе стоит 15 копеек, автомат принимает монеты по 5 и 10 копеек (выходные значения – «жду», «выдать кофе», «выдать кофе и сдачу», «возврат денег»).

3. Построить автомат, выбрасывающий лишние пробелы между словами предложения.

4. Построить конечный автомат, выдающий остаток от деления вводимого троичного числа на 4. Построить два варианта автомата: для случая, когда число вводится со старших разрядов, и для случая, когда число вводится с младших разрядов.

5. Показать, что не существует автономного конечного автомата, выдающего выходную цепочку вида:

# 2. Эквивалентность конечных автоматов

Для булевых функций часто ставится вопрос об эквивалентности. Так как функции работают на конечных множествах, проверку эквивалентности можно провести, сопоставляя таблицы функций. Иная ситуация с конечными автоматами. Два конечных автомата эквивалентны, если реализуемые ими отображения вход-выход эквивалентны. Конечный автомат реализует отображение бесконечного множества входных последовательностей сигналов в бесконечное множество выходных последовательностей сигналов. Поэтому автоматные отображения нельзя сравнить простым перечислением их значений на всех возможных аргументах.

Дадим несколько определений. Пусть – алфавит (конечное множество символов) и – множество цепочек из элементов . Цепочки будем обозначать малыми греческими буквами , , . Введём на множестве цепочек операцию конкатенации, отмечаемую как (точка). Через будем обозначать цепочку, не содержащую символов.

Определение 3. Пусть – конечный автомат. Расширенными функциями перехода и выхода автомата называются функции и , определённые как

Расширенные функции определены на множестве входных последовательностей (входных цепочках) в отличие от обычных функций переходов и выходов, которые определены на множестве одиночных входных сигналов.

Рассмотрим пример построения расширенных функций для автомата из параграфа 1. Пусть имеется цепочка . Тогда, следуя определению 3:

Пусть в некоторое состояние автомата не существует пути из начального состояния. Иными словами, в это состояние автомат не может попасть. Такие состояния автомата называются недостижимыми, остальные – достижимыми. Очевидно, что недостижимые состояния и переходы из них можно отбросить, они не влияют на поведение конечного автомата.

Определение 4. Пусть – конечный автомат. Состояние называется достижимым, если под воздействием какой-либо цепочки входных сигналов автомат попадает в это состояние. Состояние конечного автомата является недостижимым, если под воздействием любой входной цепочки автомат не переходит в это состояние.

– достижимо .

– недостижимо .

Вернёмся теперь к проблеме эквивалентности конечных автоматов.

Определение 5. Конечные автоматы   
и называются эквивалентными, если совпадают их входные алфавиты и реализуемые ими отображения:

;

.

Как было отмечено ранее, проблема эквивалентности конечных автоматов является нетривиальной. Но, оказывается, существует простой метод решения этой проблемы. Этот метод основан на понятии прямого произведения конечных автоматов.

Определение 6. Прямое произведение автоматов  
 и с одинаковым входным алфавитом – это автомат , где:

;

.

Иными словами, конечный автомат, являющийся прямым произведением двух конечных автоматов, в качестве своих состояний имеет пары состояний исходных автоматов, его начальное состояние есть пара их начальных состояний, выходной алфавит – множество пар выходных символов автоматов-множителей, а функции переходов и выходов определены покомпонентно. Таким образом, прямое произведение конечных автоматов – это просто два стоящих рядом невзаимодействующих конечных автомата, синхронно работающих на одном общем входе.

Теорема 1. (Теорема Мура) Два конечных автомата и с одинаковым входным алфавитом являются эквивалентными тогда и только тогда, когда для любого достижимого состояния в их прямом произведении справедливо

Рассмотрим использование теоремы Мура на примере. Пусть имеются два конечных автомата и , графическое представление которых содержит рис. 2.



Рис. 2. Два конечных автомата и .

Действуя по определению 6, построим произведение и (рис. 3).



Рис. 3. Прямое произведение автоматов и .

Выбросим из полученного произведения недостижимые состояния (рис. 4).



Рис. 4. Произведение автоматов и без недостижимых состояний.

По графу переходов видно, что из всех достижимых состояний автомат выдаёт пары одинаковых выходных сигналов. Следовательно, автоматы и эквиваленты.

# 3. Минимизация конечных автоматов

Пример из предыдущего параграфа демонстрирует, что разные автоматы могут функционировать одинаково, даже если у них разное число состояний. Важной задачей является нахождение минимального автомата, который реализует заданное автоматное отображение.

Процедура минимизации конечного автомата базируется на понятии эквивалентных состояний.

Определение 7. Два состояния и конечного автомата  
 называются эквивалентными (обозначается ), если для любой цепочки выполняется .

Если у заданного автомата найти «максимально возможное» разбиение его состояний на классы эквивалентности, то, выбирая эти классы как новые состояния, получим минимальный автомат, эквивалентный исходному.

Определение 7 не является конструктивным: оно не даёт процедуры выяснения того, являются ли два состояния эквивалентными, поскольку мы не можем перебрать все входные цепочки. Однако существует алгоритм определения максимального отношения эквивалентности на множестве состояний конечного автомата, который будет сейчас рассмотрен.

Определение 8. Состояния и конечного автомата называются k-эквивалентными (обозначается ), если они неразличимы любой цепочкой длины .

Алгоритм минимизации состоит в последовательном построении на множестве состояний автомата цепочки разбиений таких, что в один класс разбиения попадают *k*-эквивалентные состояния. Очевидно, что в любом автомате все состояния 0-эквивалентны (при подаче пустой цепочки на вход выходом также является пустая цепочка независимо от состояния автомата). Значит, . Разбиение также легко построить. По определению, в один блок разбиения попадают те состояния, в которых автомат одинаково реагирует на входные сигналы: . Разбиения и являются исходными при построении цепочки разбиений. Если мы сможем определить, как строить следующее разбиение из предыдущего, то начиная с , мы сможем последовательно построить всю цепочку.

Теорема 2. Пусть . Для того чтобы , необходимо и достаточно, чтобы . Иными словами, для того, чтобы два k-эквивалентных состояния были бы -эквивалентными, необходимо и достаточно, чтобы под воздействием любого входного сигнала автомат переходил из этих состояний в пару состояний, которые сами были бы k-эквивалентными.

Очевидно, что если и -эквивалентны, то они *k*-эквивалентны. Иными словами, блоки разбиения являются подблоками разбиения . Поскольку число состояний конечно, может быть только конечное число последовательно уменьшающихся разбиений , начиная с максимального разбиения , содержащего единственный блок. Более того, очевидно, что их не больше, чем число состояний автомата. Однако построение уменьшающихся разбиений можно не продолжать дальше, как только два последовательных разбиения совпадают.

Теорема 3. Пусть . Тогда для любого .

Рассмотрим процедуру минимизации на конкретном примере. Пусть имеется конечный автомат, заданный следующей таблицей функций и .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  | |
|  |  |  |  |
| 1 | 3 | 6 | 1 | 0 |
| 2 | 4 | 8 | 0 | 1 |
| 3 | 1 | 4 | 1 | 0 |
| 4 | 7 | 9 | 0 | 1 |
| 5 | 9 | 1 | 1 | 0 |
| 6 | 3 | 5 | 1 | 0 |
| 7 | 4 | 3 | 0 | 1 |
| 8 | 4 | 2 | 1 | 0 |
| 9 | 5 | 7 | 1 | 0 |

Будем последовательно строить разбиения его состояний.

– . Это тривиальное разбиение, оно состоит из одного элемента, включающего все состояния автомата.

– . Автомат, который находится в состоянии 1, генерирует для входных сигналов комбинацию выходов (1,0), которая совпадает с комбинациями для состояний 3, 5, 6, 8, 9. Поэтому данные состояния являются 1-эквивалентными и объединяются в один класс . Аналогично выделяется класс .

Чтобы вычислить дальнейшие разбиения, применим теорему 2. Для этого построим таблицу переходов автомата, руководствуясь правилом: если принадлежит некому классу, то пишем в соответствующей ячейке имя этого класса. Например, , и так далее.

|  |  |  |
| --- | --- | --- |
|  |  | |
|  |  |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |

Выделяя различные комбинации состояний (не выходов!), получим очередное разбиение:

– .

Обозначим для удобства , , , и построим новую таблицу переходов, используя полученные классы эквивалентности:

|  |  |  |
| --- | --- | --- |
|  |  | |
|  |  |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |

Таким образом, получим:

– .

– .

Так как , на основании теоремы 3 заключаем, что алгоритм минимизации завершён. Новый минимальный автомат будет включать пять состояний (выполните графическое построение исходного и минимизированного автомата самостоятельно).

## Вопросы и упражнения.

1. Минимизировать конечный автомат.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  | | |
| 0 | 1 | 2 | 0 | 1 | 2 |
| 1 | 4 | 6 | 1 | 1 | 0 | 0 |
| 2 | 2 | 2 | 6 | 0 | 0 | 1 |
| 3 | 4 | 2 | 3 | 1 | 0 | 0 |
| 4 | 4 | 3 | 3 | 1 | 0 | 0 |
| 5 | 6 | 6 | 5 | 0 | 0 | 1 |
| 6 | 5 | 6 | 2 | 0 | 0 | 1 |

2. Минимизировать конечный автомат.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  | | |
| 0 | 1 | 2 | 0 | 1 | 2 |
| 1 | 2 | 5 | 5 | 1 | 0 | 1 |
| 2 | 6 | 2 | 5 | 0 | 1 | 1 |
| 3 | 2 | 2 | 7 | 0 | 1 | 1 |
| 4 | 4 | 3 | 1 | 1 | 0 | 1 |
| 5 | 2 | 5 | 5 | 1 | 0 | 1 |
| 6 | 2 | 6 | 5 | 0 | 1 | 1 |
| 7 | 6 | 6 | 3 | 0 | 1 | 1 |

3. Разработать алгоритм минимизации не полностью определённого конечного автомата, то есть такого, у которого некоторые переходы и выходы не определены (считая, что значения функций переходов и выходов на соответствующих парах аргументов являются безразличными).

# 4. Конечные автоматы Мили и Мура

Автоматы, рассмотренные в предыдущих параграфах, называются автоматами Мили. Автоматы Мура образуют другой класс моделей, с точки зрения вычислительной мощности эквивалентный классу автоматов Мили.

Определение 9. Конечным автоматом Мура называется шестёрка объектов , где:

– конечное непустое множество состояний;

X – конечное непустое множество входных сигналов;

Y – конечное непустое множество выходных сигналов;

– начальное состояние;

– функция переходов;

– функция выходов.

В автомате Мура выходная функция . определяется не на паре (состояние, входной сигнал), а только на состоянии. То есть, автомат под воздействием входного сигнала переходит в некое состояние, и, выполнив такой переход, генерирует соответствующий выходной сигнал. Пример конечного автомата Мура представлен на рис. 5.



Рис. 5. Автомат Мура и эквивалентный ему автомат Мили.

Очевидно, что по любому автомату Мура легко построить эквивалентный ему автомат Мили. Для этого достаточно перенести значения, генерируемые в состояниях, на стрелки, которые ведут в эти состояния.

Не столь очевидно, что справедливо и обратное утверждение: для любого автомата Мили существует эквивалентный ему автомат Мура. Справедливость этого утверждения легко доказывается конструктивно. Рассмотрим рис. 6. Каждое состояние автомата Мили расщепляется на несколько эквивалентных состояний, с каждым из которых связывается один выходной символ. Построение переходов эквивалентного автомата Мура ясно из рисунка.



Рис. 6. Автомат Мили и эквивалентный ему автомат Мура.

# 5. Автоматные языки

В этом и последующих параграфах конечные автоматы рассматриваются не как преобразователи информации, реагирующие на отдельные входные сигналы, а как распознаватели последовательностей входных сигналов. Последовательности символов часто называют предложениями, а их множества – языками. Таким образом, конечный автомат можно рассматривать как устройство, выполняющее алгоритмические операции над языками.

Определение 10. Конечное множество элементов будем называть словарём, элементы словаря – символами, а последовательности символов словаря – цепочками или предложениями. Множество предложений назовём языком. Язык над словарём V будем обозначать , или просто L, если V очевидно.

Пусть – словарь. Обозначим через множество всех возможных цепочек, составленных из символов словаря . Например, если , то , где – пустая цепочка, не содержащая символов. Очевидно, что хотя конечно, – бесконечное счётное множество. Язык – это просто некоторое подмножество . Всего языков над словарем бесконечное число; множество языков имеет мощность континуум.

Рассмотрим несколько примеров языков, на которые будем ссылаться в дальнейшем.

1. ; .

2. ; .

3. ; .

4. ; .

5. ; .

6. ; .

7. ; .

8. ; .

9. ; .

10. ; .

Определение 11. Любой конечный механизм задания языка называется грамматикой.

Существует два типа грамматик: порождающие и распознающие. Под порождающей грамматикой языка понимается конечный набор правил, позволяющий строить все «правильные» предложения языка и ни одного «неправильного». Распознающая грамматика задаёт критерии принадлежности произвольной цепочки данному языку. Это фактически некоторый алгоритм, принимающий в качестве входа символ за символом произвольную цепочку над словарём и дающий на выходе один из двух возможных ответов: «данная цепочка принадлежит языку » либо «данная цепочка не принадлежит языку ».

Роль распознающей грамматики может выполнить конечный автомат без выхода. Свяжем с некоторыми состояниями автомата метку «Да», а с остальными – метку «Нет». Тогда множество входных цепочек автомата разобьётся на два класса: одни – приводящие автомат в одно из состояний, помеченных «Да», все другие – приводящие автомат в одно из состояний, помеченных «Нет».

Определение 12. Конечным автоматом-распознавателем называется пятёрка объектов , где:

– конечное непустое множество состояний;

– конечное непустое множество входных сигналов;

– начальное состояние;

– функция переходов;

– множество заключительных (финальных) состояний.

Определим обобщённую функцию переходов автомата точно так же, как в определении 3.

Определение 13. Конечный автомат-распознаватель допускает входную цепочку , если переводит его из начального в одно из заключительных состояний, то есть если . Множество всех цепочек, допускаемых автоматом , образует язык, допускаемый .

Определение 14. Язык, для которого существует распознающий его конечный автомат, называется автоматным языком.

Рассмотрим примеры распознавателей для некоторых языков, определённых выше. Будем использовать представление автомата в виде графа. При этом финальные состояния будем выделять жирной линией.

1. Автомат-распознаватель для языка представлен на рис. 7.



Рис. 7. Автомат-распознаватель языка .

Входные цепочки и (и только они) переводят автомат из начального состояния в одно из заключительных состояний или . Чтобы не загромождать граф, примем следующее соглашение. Если на графе не указан переход под воздействием некоторого входного сигнала , то это означает, что существует незаключительное состояние, в которое автомат переходит по , причём после этого под воздействием всех входных сигналов автомат не выходит из этого незаключительного состояния.



Рис. 8. Упрощённый граф автомата-распознавателя языка .

Очевидно, что любой конечный язык может быть задан конечным автоматом и поэтому все конечные языки – автоматные.

2. Любой автомат с пустым множеством заключительных состояний допускает .

3. Автомат с единственным состоянием, которое является заключительным, имеющий три перехода (помеченные символами , , ) из этого состояния в него же, допускает .

4. Язык не является автоматным. Предположим противное: существует автомат , который распознает . Пусть число состояний автомата равно . Рассмотрим цепочку . Под воздействием первой части цепочки – последовательных символов – автомат обязательно вернётся в какое-либо из состояний, в котором он уже находился. Пусть это будет состоянием , и длина цепочки , под воздействием которой автомат переходит из в , равна . Тогда цепочка тоже будет переводить автомат из начального состояния в заключительное состояние. Но . Получили противоречие. Значит, автомата, распознающего , не существует.

Более общий критерий, позволяющий сделать заключение, является ли автоматным произвольный язык, даёт так называемая лемма о накачке.

5. В отличие от языка , язык – автоматный. Автомат с двумя состояниями распознает (рис. 9). Простота этого распознавателя объясняется тем, что он не должен отслеживать количества символов и во входных цепочках.



Рис. 9. Автомат, распознающий .

6. Язык – не автоматный язык.

7. Автомат, распознающий , показан на рис. 10.



Рис. 10. Автомат, распознающий .

8. Автомат с тремя состояниями распознает (рис. 11, вместо цифр на переходах поставлена буква «ц», обозначающая любую цифру).



Рис. 11. Автомат, распознающий .

9. Автомат, распознающий , показан на рис. 12.



Рис. 12. Автомат, распознающий .

10. Язык – не автоматный язык.

## Вопросы и упражнения.

Описать автоматы-распознаватели для языков над словарём :

1. Множество цепочек, оканчивающихся на 00;

2. Множество цепочек, содержащих три нуля подряд;

3. Множество цепочек, у которых на третьей позиции справа стоит 1.

# 6. Лемма о накачке

Лемма о накачке (pumping lemma) – это важный теоретический результат, позволяющий во многих случаях проверить, является ли язык автоматным. Поскольку все конечные языки являются автоматными, эту проверку имеет смысл делать только для бесконечных языков.

Теорема 4. (Лемма о накачке) Пусть – автоматный язык. Тогда существует константа (зависящая от ), для которой каждую цепочку из языка , удовлетворяющую неравенству , можно разбить на три цепочки так, что выполняются следующие условия:

1. .

2. .

3. Для любого цепочка также принадлежит .

Это значит, что всегда можно найти такую цепочку недалеко от начала цепочки , которую можно «накачать». Таким образом, если цепочку повторить любое число раз или удалить (при ), то результирующая цепочка всё равно будет принадлежать языку .

Идея доказательства леммы о накачке аналогична рассуждениям, приведённым при доказательстве того, что язык не является автоматным.

Используем лемму о накачке для доказательства неавтоматности некоторых языков.

1. ; .

Предположим, что язык автоматный. Согласно лемме о накачке существует некая константа , при которой выполняются все условия леммы. Рассмотрим цепочку . Пусть – разбиение цепочки по условиям леммы. Так как , то состоит из одних нулей. Цепочка должна принадлежать . Но в цепочке единиц больше, чем нулей. Получили противоречие. Следовательно, гипотеза об автоматности ошибочна.

2. Докажем неавтоматность языка , образованного всеми цепочками из единиц, длины которых – простые числа. Предположим, что язык автоматный. Тогда должна существовать константа , удовлетворяющая условиям леммы о накачке. Рассмотрим любое простое число . Пусть .

Согласно лемме о накачке можно разбить цепочку так, что и . Пусть . Тогда . Рассмотрим цепочку , которая по лемме о накачке должна принадлежать языку , если он действительно является автоматным. Однако

Так как , то . Кроме того, , а , поэтому . Значит число не простое, так как имеет два сомножителя, больших единицы.

Мы начали с предположения, что предлагаемый язык является автоматным, и пришли к противоречию, доказав, что существует некоторая цепочка, которая не принадлежит этому языку, тогда как по лемме о накачке она должна ему принадлежать. Таким образом, язык не является автоматным.

## Вопросы и упражнения.

Докажите, что следующие языки не являются автоматными:

а) множество цепочек, состоящих из «сбалансированных» скобок «(« и «)», которые встречаются в правильно построенном арифметическом выражении;

б) ;

в) ;

г) .

# 7. Эквивалентность и минимизация автоматов-распознавателей

Очевидно, что два распознавателя следует считать эквивалентными, если они распознают один и тот же язык. Проблема эквивалентности двух распознавателей и состоит в нахождении ответа на вопрос: совпадают ли языки, распознаваемые и . Проблема минимизации распознавателя состоит в нахождении такого распознавателя , который, распознавая тот же язык, что и , имеет минимально возможное число состояний. Оказывается, что проблемы эквивалентности и минимизации конечных автоматов-распознавателей только в некоторых чертах отличаются от таковых для конечных автоматов-преобразователей.

Определение 15. Пусть и – два конечных автомата-распознавателя с одним и тем же входным алфавитом. Их прямым произведением называется автомат  
, где

.

Как и для автоматов-преобразователей, прямое произведение автоматов-распознавателей – это просто два автомата с одними и теми же входами, работающие синхронно.

Теорема 5. (Теорема Мура) Два конечных автомата-распознавателя с одинаковым входным алфавитом являются эквивалентными тогда и только тогда, когда при их синхронном функционировании под воздействием одних и тех же входных символов они на каждом шаге всегда попадают либо оба в заключительные, либо оба в незаключительные состояния.

Определение 16. Два состояния и конечного автомата-распознавателя называются эквивалентными, если для любой цепочки функции и попадают либо оба в заключительные, либо оба в незаключительные состояния.

Для автомата, изображённого на рис. 7, состояния и эквивалентны. Любая входная цепочка, поданная на автомат, находящийся в состоянии , будет недопустима, точно так же, как и в случае, когда автомат находится в состоянии . Эквивалентные состояния можно объединить в один класс и построить новый автомат, состояниями которого являются классы эквивалентных состояний. Если мы можем определить на множестве состояний автомата максимально возможное разбиение на классы эквивалентности, то, выбирая его классы эквивалентности как новые состояния, получим минимальный автомат, эквивалентный исходному автомату.

Алгоритм минимизации конечного автомата-распознавателя состоит в построении на множестве его состояний таких разбиений , что в один класс разбиения попадают k-эквивалентные состояния, то есть находящиеся в отношении эквивалентности . При подаче на вход пустой цепочки автомат остается в том же состоянии, в котором он находился. Поэтому разбиение состоит из двух блоков: в один блок попадают все заключительные состояния, а в другой – все незаключительные состояния. Определив, как строить следующее разбиение из предыдущего, мы сможем последовательно построить всю цепочку

Теорема 6. Пусть в конечном автомате-распознавателе , . Для того чтобы , необходимо и достаточно, чтобы .

Теорема 7. Пусть выполняется минимизация конечного автомата-распознавателя и . Тогда для любого .

Рассмотрим пример минимизации конечного автомата, заданного таблицей переходов. Начальное состояние здесь 0, заключительные состояния – 1, 3, 7.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | |  | |
|  | |  | |  | |
|  |  |  |  |  |  |
| 0 | 3 | 2 |  |  |  |  |
|  | 6 | 5 |  |  |  |  |
| 2 | 4 | 7 |  |  |  |  |
|  | 1 | 7 |  |  |  |  |
| 4 | 6 | 3 |  |  |  |  |
| 5 | 3 | 9 |  |  |  |  |
| 6 | 9 | 3 |  |  |  |  |
|  | 1 | 3 |  |  |  |  |
| 8 | 7 | 6 |  |  |  |  |
| 9 | 4 | 7 |  |  |  |  |

Начальное разбиение представляет собой два блока, включающие один заключительные, а другой – незаключительные состояния. Поэтому

Разбиение в один блок объединяет те состояния, которые нельзя различить при подаче цепочек длиной 1, то есть те, которые под воздействием одного и того же входного сигнала переходят в один и тот же блок разбиения .

Следующее разбиение в один блок объединяет те состояния, которые под воздействием одного и того же входного сигнала переходят в один и тот же блок предыдущего разбиения .

Разбиение совпадает с разбиением . Таким образом, минимальный автомат с эквивалентным поведением имеет 4 состояния, представляющих блоки разбиения . Начальным состоянием здесь является состояние , заключительными – два состояния и .

|  |  |  |
| --- | --- | --- |
|  |  | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 8. Недетерминированные конечные автоматы

В основе построения недетерминированного конечного автомата-распознавателя лежит допущение о возможности перехода по определённому входному сигналу сразу в несколько состояний. В такой момент происходит своеобразное «расщепление» реальности: если переход по сигналу возможен в состояния и , то допускается, что в одном из «миров» автомат перешёл в состояние и продолжил свою работу, в другом «мире» состоялся переход в .

В данном параграфе демонстрируется, что любой недетерминированный конечный автомат (НКА) эквивалентен некоторому обычному детерминированному автомату (ДКА). Основная причина интереса к НКА – их более простой и компактный вид для многих задач распознавания.

Определение 17. Недетерминированным конечным автоматом-распознавателем называется пятёрка объектов , где:

– конечное непустое множество состояний;

– конечное непустое множество входных сигналов;

– множество начальных состояний;

– функция переходов ( – булеан множества );

– множество финальных состояний.

Как видим, разница в определениях НКА и ДКА заключается в виде функции переходов.

Если разрешить в автомате не только переход по одному сигналу во множество состояний, но и переход в новое состояние по пустой строке, получим недетерминированный автомат с ε-переходами.

Определение 18. НКА с ε-переходами называется пятёрка объектов , где:

– конечное непустое множество состояний;

– конечное непустое множество входных сигналов;

– множество начальных состояний;

– функция переходов;

– множество финальных состояний.

Как определить для недетерминированных автоматов основное понятие – «распознавание входной цепочки»? Подход, который принят в теории формальных языков, состоит в следующем.

Определение 19. Недетерминированный конечный автомат распознает входную цепочку , если существует путь, помеченный символами цепочки , из начального в одно из заключительных состояний автомата, возможно, с учетом спонтанных переходов. Недетерминированный конечный автомат распознает язык , если он распознает все цепочки этого языка, и только их.

Рассмотрим примеры НКА. Пусть , – все цепочки, содержащие подстроку . НКА, распознающий язык , представлен на рис. 13.



Рис. 13. Недетерминированный конечный автомат.

Следующий НКА с ε-переходами будет использоваться в дальнейших примерах.



Рис. 14. Недетерминированный конечный автомат с ε-переходами.

Теорема 8. Для любого недетерминированного конечного автомата-распознавателя существует эквивалентный ему детерминированный конечный автомат-распознаватель.

Доказательство этой теоремы проведём конструктивно, то есть предложим алгоритм построения по заданному НКА эквивалентного ему ДКА. Сначала покажем, как НКА с ε-переходами привести к НКА без ε-переходов, а потом – как для НКА без ε-переходов построить эквивалентный ему детерминированный автомат, допускающий тот же язык.

Пусть – НКА с ε-переходами. Будем строить НКА – эквивалент , но без ε-переходов.

Определение 20. ε-замыканием состояния , обозначаемым , называется множество всех состояний, которые достижимы из без подачи входного сигнала.

Множеству принадлежит само состояние и все те состояния, которые достижимы из по ε-переходами. Для автомата, изображённого на рис. 14:

Множеством состояний автомата являются ε-замыкания состояний . Начальными состояниями будут ε-замыкания начальных состояний . Множеством заключительных состояний будут такие ε-замыкания, в которые входят заключительные состояния .

Функция переходов автомата определяется по следующей формуле:

С учётом вышесказанного, определим для автомата , эквивалентного автомату, изображённому на рис. 14, таблицу переходов. Начальным состоянием нового автомата будет состояние , заключительным состоянием – .

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |



Рис. 15. Недетерминированный конечный автомат без ε-переходов.

Рассмотрим теперь алгоритм построения по заданному недетерминированному автомату без ε-переходов эквивалентного ему детерминированного автомата. В качестве состояний детерминированного автомата следует выбрать булеан множества . Начальным состоянием будет элемент из , соответствующий . Множество заключительных состояний искомого автомата состоит из всех таких множеств состояний, которые включают хотя бы одно заключительное состояние исходного автомата. Функция переходов детерминированного автомата определяется следующим образом:

В случае нашего примера множество состояний будет содержать 16 элементов: , , , , , , , …, . Начальным состоянием будет состояние . Финальными будут все состояния, которые содержать . И, например,

После того, как детерминированный автомат построен, можно выполнить его минимизацию. Для нашего примера после минимизации получим автомат, показанный на рис. 16.



Рис. 16. ДКА, построенный по недетерминированному автомату.

## Вопросы и упражнения.

Преобразуйте следующие НКА в ДКА. Начальные состояния помечены стрелкой, конечные – звёздочками.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 9. Регулярные множества, выражения и языки

В данном параграфе рассматриваются множества цепочек над конечным словарём, которые легко описать формулами особого вида. Эти множества называются регулярными.

Пусть и – множества цепочек. Определим для таких множеств три операции.

1. Объединение: .

2. Конкатенация (произведение): . Знак операции конкатенации обычно опускается.

Пример: , , .

Обозначим произведение множеств ( – пустая цепочка).

3. Итерация:

Определение 21. Класс регулярных множеств над конечным словарём определяется так:

1. – регулярное множество;

2. – регулярное множество;

3. – регулярное множество;

4. Если и – регулярные множества, то регулярны:

– объединение ;

– конкатенация ;

– итерации и .

5. Если множество не может быть построено конечным числом применения правил 1-4, то оно нерегулярно.

Регулярные множества обладают полезным свойством – их можно очень просто описать формулами, которые называются регулярными выражениями.

Определение 22. Класс регулярных выражений над конечным словарём определяется так:

1. и ε – регулярные выражения;

2. – регулярное выражение;

3. Если и — регулярные выражения, то регулярными выражениями являются:

– их сумма ;

– их произведение ;

– их итерации и .

4. Если выражение не построено конечным числом применения правил 1–3, то оно не является регулярным.

Знак произведения можно опускать. Для уменьшения числа скобок, как и в любой алгебре, используются приоритеты операций: итерация самая приоритетная; менее приоритетно произведение; самый низкий приоритет у сложения.

Примеры регулярных выражений: , .

Регулярные множества и регулярные выражения весьма близки. Но они представляют собой разные сущности: регулярное множество – это множество цепочек (в общем случае бесконечное), а регулярное выражение – это формула, схематично показывающая, как было построено соответствующее ей регулярное множество с помощью перечисленных выше операций (и эта формула конечна). Таким образом, любое регулярное выражение задаёт некий язык.

Очевидно, что множество цепочек регулярно тогда и только тогда, когда оно может быть представлено регулярным выражением. Однако одно и то же множество цепочек может быть представлено различными регулярными выражениями. Если два регулярных выражения задают одно и то же регулярное множество, будем называть такие выражения эквивалентными (). Возникает вопрос, как определить эквивалентность двух регулярных выражений.

Теорема 9. Для любых регулярных выражений , и справедливо:

1. ; ; ; ;

2. ; ; ; (в общем случае );

3. ; ;

4. ; ; ;

5. ; ; .

Эти соотношения можно доказать, проверяя равенство соответствующих множеств цепочек. Их можно использовать для упрощения регулярных выражений. Например: .

Регулярные выражения – это конечные формулы, задающие регулярные языки. Но таким же свойством обладают и конечные автоматы – они тоже задают языки. Возникает вопрос: как соотносятся между собой классы языков, задаваемые конечными автоматами и регулярными выражениями? Стефан Клини, американский математик, доказал следующую теорему.

Теорема 10. (Теорема Клини) Классы регулярных множеств и автоматных языков совпадают.

Иными словами, каждый автоматный язык может быть задан формулой (регулярным выражением) и каждое регулярное множество может быть распознано конечным автоматом. Доказательство теоремы Клини конструктивно. На первом шаге доказывается, что для любого конечного автомата можно построить регулярное выражение, задающее распознаваемый этим автоматом язык. На втором шаге показывается, что по любому регулярному выражению можно построить конечный автомат, допускающий в точности цепочки соответствующего регулярного множества.

Покажем, что для каждого регулярного выражения может быть построен конечный автомат (возможно, недетерминированный), распознающий язык, задаваемый . Определение таких автоматов дадим рекурсивно.

1. Если , то  (два несвязанных состояния).

2. Если , то .

3. Если , то .

Пусть имеется автомат  с одним начальным и одним заключительным состоянием, допускающий язык, задаваемый . Тогда

4. Если , то



(начальные и заключительные состояния и совмещаются).

5. Если , то



(начальное состояние и заключительное состояние совмещаются).

6. Если , то



Рассмотрим пример. Пусть задано выражение . Построим соответствующий автомат.

По правилу 4:



По правилам 4,5 и 6:



По правилам 5 и 6:



Построенный автомат недетерминированный, с ε-переходами. После построения эквивалентного детерминированного автомата и проведения минимизации получим автомат, изображённый на рис. 17.



Рис. 17. Автомат, соответствующий .

## Вопросы и упражнения.

Задайте следующие языки над словарём в виде регулярных выражений. Затем выполните переход от регулярного выражения к недетерминированному конечному автомату и детерминированному конечному автомату.

1. Множество цепочек, оканчивающихся на 01;

2. Множество цепочек, содержащих два нуля подряд;

3. Множество цепочек, у которых на третьей позиции справа стоит 1.

# Литература

1. Карпов, Ю. Г. Теория автоматов / Ю. Г. Карпов. – СПб. : Питер, 2002. – 224 с.: ил.

2. Мозговой, М. В. Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход / М. В. Мозговой. – СПб. : Наука и техника, 2006. – 320 с.: ил.

3. Хопкрофт, Джон, Э. Введение в теорию автоматов, языков и вычислений, 2-е изд. / Дж. Э. Хопкрофт, Р. Мотвани, Дж. Д. Ульман ; пер. с англ. – М. : Издательский дом «Вильямс», 2008. – 528 с.: ил.