

ТЕОРИЯ АВТОМАТОВ

Глава 1. Введение в теорию конечных автоматов

1.1. Модели вычислений

Большая часть из нас думает о “компьютерах” как о ноутбуках, десктопах и, возможно, суперкомпьютерах. В реальности же вокруг нас намного больше небольших “компьютеров”, которые занимаются вычислениями: часы в микроволновке, сотовые телефоны, кассовые аппараты. О каждом из этих устройств можно думать как о реакционной системе – так как каждая из них реагирует на сигналы или входы из внешнего мира.

В изучении такого большого класса систем, очень полезно иметь какую-нибудь простую модель вычислений, которая абстрагируется от каких-то незначимых деталей программирования или “реальной жизни”. Это позволяет нам сфокусироваться на “вычислительной мощности” конкретной модели вычислений – что возможно вычислить в данной модели вычислений. Это также позволяет нам сфокусироваться на “репрезентативной мощности” конкретной модели вычислений.

1.2. Главная идея

Главное соображение о реакционных системах в том, что ответ на конкретный стимул (сигнал или часть входа) не всегда одинаковый. Например, в случае кассового аппарата он не выдаст чек при нажатии на кнопку, если товар ещё не был оплачен. Таким образом ответ при нажатии на кнопку зависит от предыдущей истории использований системы.

Модель конечного автомата ограничивает количество различных ответов на определенные стимулы конечным числом, которые зафиксированы в описании автомата. Это наибольшее различие между конечными автоматами и другими моделями вычислений. По существу в конечном автомате мы можем строить только автоматы, которые имеют конечное количество состояний. Единственное, что может сделать стимул это, возможно, породить какой-то ответ от автомата и изменить его состояние в какое-то новое.

При дизайне конечного автомата для конкретной задачи обычно вход для автомата представляет собой последовательность стимулов. Мы будем использовать терминологию “алфавита” Σ , чьи символы соответствуют определенному стимулу, который может быть послан в автомат. Тогда наш вход будет состоять из последовательности символов из этого алфавита, где последовательность

обычно называются “строкой” над алфавитом Σ .

1.3. Детали

Мы описываем конечный автомат следующими определениями:

1. Описание стимула для машины. Оно определяется через конечный входной алфавит, состоящий из различных символов. Входной алфавит обычно обозначается как Σ .
2. Описание ответов, которые может выдавать автомат. Оно определяется через конечный выходной алфавит. Выходной алфавит обычно обозначается как Λ .
3. Мы описываем автомат в виде диаграммы, состоящей из кружков, которые обозначают различные состояния автомата. Множество состояний конечного автомата обычно обозначается как Q .
4. Часто есть фиксированное начальное состояние автомата. На диаграмме это состояние помечено стрелкой, которая указывает на начальный кружок, но не соединено ни с каким другим состоянием.
5. Вычислительное поведение автомата описывается переходами, которые являются стрелками, идущими из одного состояния в другое. В случае, если автомат порождает выходы, то каждый переход можно обозначить парой символов a/b , где a берётся из Σ и b из Λ . Идея в том, что получив на вход символ a , то автомат перейдёт из текущего состояния в состояние b . В случае, если автомат не порождает выходы, то каждый переход можно обозначить просто символом из Σ .

Поведение

Определим след конечного автомата M :

1. Конечная последовательность чередующихся состояний и переходов с начальным и конечными состояниями: $s_0, i_1/o_1, s_1, \dots, s_{n-1}, i_n/o_n, s_n$
2. Первое состояние s_0 это начальное состояние M .
3. Для каждой тройки $s_{j-1}, i_j/o_j, s_j$ (состояние, пара входного и выходного символа, состояние), появляющейся в последовательности, в автомате M должен быть переход из состояния s_{j-1} в s_j , обозначенное i_j/o_j .

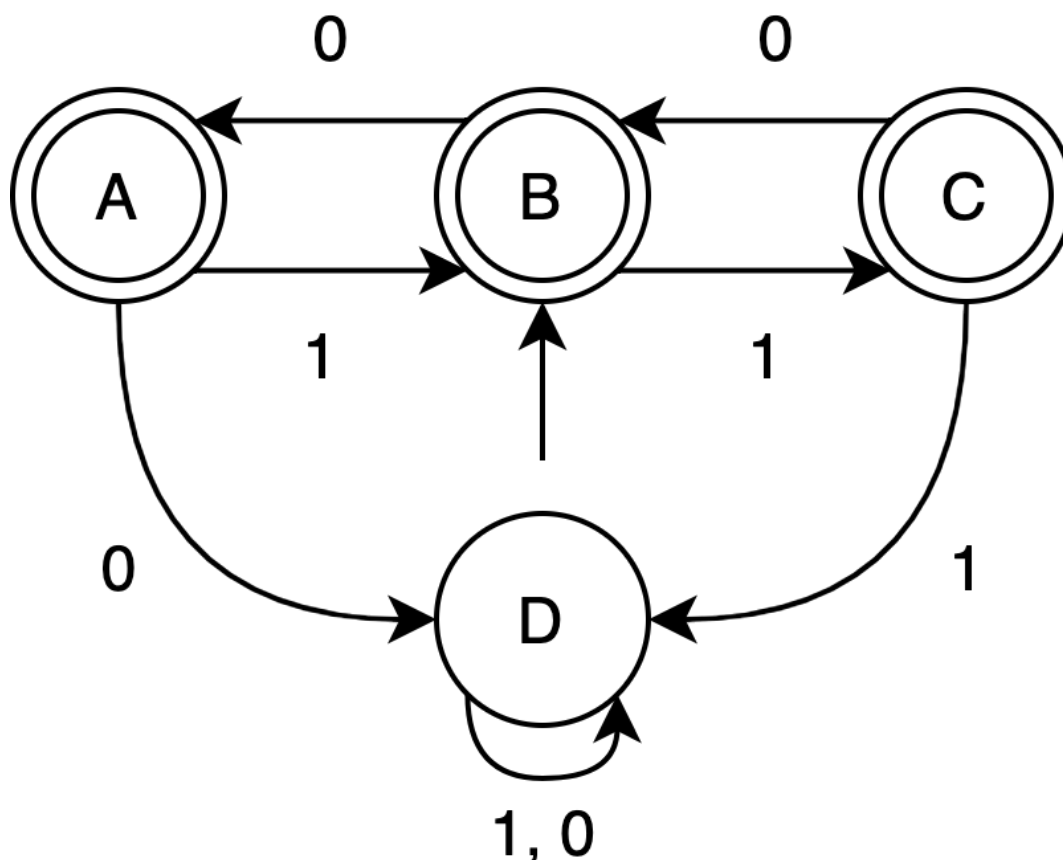


Рисунок 1. Пример конечного автомата

Пример

Конечный автомат на рисунке 1 принимает последовательности нулей и единиц, в которых количество нулей никогда не превосходит количество единиц более чем на 1, и наоборот. В данном случае последовательность 0101101 допустима, так как соответствующий след $B, 0, A, 1, B, 0, A, 1, B, 1, C, 0, B, 1, C$ имеет последним состоянием конечное состояние C .

Напротив, последовательность 0101001 не допустима, так как соответствующий след $B, 0, A, 1, B, 0, A, 1, B, 0, A, 0, D, 1, D$ имеет последним состоянием D , которое не является конечным.

Детерминированность

В обсуждении мы ни разу не говорил о том, что из одного состояния можно перейти только в одно состояние. Если это так, то конечный автомат называется “детерминированным”. Автоматы, в которых из одного состояния можно перейти в несколько, называются “недетерминированными”.

Глава 2. Детерминированные конечные автоматы

2.1. Определение

Алфавит Σ это некоторый набор конечного числа различных символов. Когда мы говорим об алфавите, мы обычно заинтересованы в строках над этим алфавитом. Строка над Σ это просто последовательность любого количества символов, взятых из Σ . Мы обозначаем Σ^* множество всех строк над Σ . Для любого алфавита обязательно есть пустая строка, которые обозначают ϵ . Для любого алфавита Σ язык над Σ^* это подмножество $L \subset \Sigma^*$.

Определение 2.1. *Детерминированный конечный автомат это автомат M , определенный как набор*

$$M = (Q, \Sigma, s_0, F, \Delta) \quad (1)$$

, состоящий из:

1. *конечного множества состояний Q*
2. *конечного алфавита Σ*
3. *выделенного начального состояния $s_0 \in Q$*
4. *множества $F \subseteq Q$ конечных состояний и*
5. *функции переходов $\Delta : Q \times \Sigma \rightarrow Q$.*

Определение 2.2. *Допустимый язык L конечного автомата $M = (Q, \Sigma, s_0, F, \Delta)$ это множество строк $x \in \Sigma^*$ такое, что:*

- *есть след x в автомате M ;*
- *след заканчивается в конечном состоянии автомата M .*

Заметим, что для любого детерминированного конечного автомата, чей набор переходов является функцией, всегда выполняется первое условие (существует след для любой строки $x \in \Sigma^*$). Таким образом для любого $x \in \Sigma^*$, x не допустима тогда и только тогда, когда последнее состояние его следа это не конечное состояние (последнее состояние не находится в F).

Теорема 1. *Каждый детерминированный конечный автомат эквивалентен детерминированному конечному автомату, который удовлетворяет условиям из определения 2.1.*

Доказательство 1. Пусть M это детерминированный конечный автомат, чей набор переходов не является строго определенной функцией. Тогда может существовать некоторая строка $x \in \Sigma^*$, которая не допустима автоматом, потому что не существует следа для этой строки. Добавим в автомат M новое состояние Q , которое мы будем называть “удаленным” состоянием. Удаленное состояние не является конечным состоянием. Теперь для каждого состояния q и каждого символа a такого, что нет перехода из q помеченного a , мы добавим переход, используя удаленное состояние как результат этого перехода. Таким образом мы направим все строки, у которых нет следа в автомате M в это новое удаленное состояние, которое не является конечным. Мы также добавим переход из удаленного состояния в него же по всем символам из алфавита Σ . Это обеспечит нам то, что функция переходов является строгой на расширенном наборе состояний, и что единожды направленная в удаленное состояние строка будет оставаться в нём. Таким образом каждая строка, у которой нет следа в автомате M теперь имеет след в расширенном конечном детерминированном автомате, и каждая такая строка будет недопустимой для нового автомата. Тем самым язык нового автомата идентичен $L(M)$.

2.2. Детерминированный конечный автомат для распознавания нечетных чисел

Теперь рассмотрим задачу создания детерминированного конечного автомата для распознавания нечетных натуральных чисел.

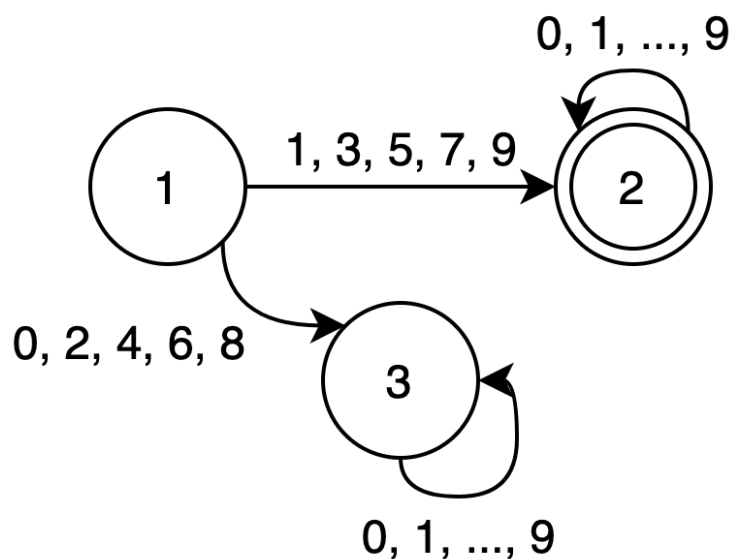


Рисунок 2. Автомат для распознавания нечетных натуральных чисел

Заметим, что автомат на рисунке 2 детерминированный. В нём нет множественных переходов, и все переходы исходящие из одного состояния имеют разные символы.

Глава 3. Недетерминированные конечные автоматы

3.1. Определение

Главное отличие между недетерминированными и детерминированными конечными автоматами в том, что входные строки могут иметь больше одного следа.

Концепт недетерминированности даёт нам больше гибкости, когда мы моделируем реакционные системы: мы можем использовать его, чтобы помочь нам описать системы, чьё поведение мы не можем полностью предсказать. За такую гибкость приходится платить – у нас больше нет приятного свойства, что каждая входная строка имеет в точности один вычислительный путь (след), который можно предсказать из определения конечного автомата. Это значит, например, что реализация недетерминированного конечного автомата на языках программирования не такая прямолинейная задача.

Однако главный интерес в недетерминированных конечных автоматах это изучение формальных языков, для которых они очень ценны.

Определение 3.1. *Недетерминированный конечный автомат это автомат M , определенный как набор*

$$M = (Q, \Sigma, s_0, F, \Delta) \quad (2)$$

, состоящий из:

1. *конечного множества состояний Q*
2. *конечного алфавита Σ*
3. *выделенного начального состояния $s_0 \in Q$*
4. *множества $F \subseteq Q$ конечных состояний и*
5. *описания Δ всех возможных переходов. Формально Δ это функция вида $\Delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow P(Q)$, где $P(Q)$ – множество всех подмножеств множества Q .*

Пример

Рисунок 3 демонстрирует недетерминированный конечный автомат M , который принимает строки над алфавитом $\Sigma = \{a, b\}$, которые оканчиваются на aba . Автомат M имеет формальное определение:

$$M = (\{0, 1, 2, 3\}, \{a, b\}, 0, \{3\}, \Delta) \quad (3)$$

,

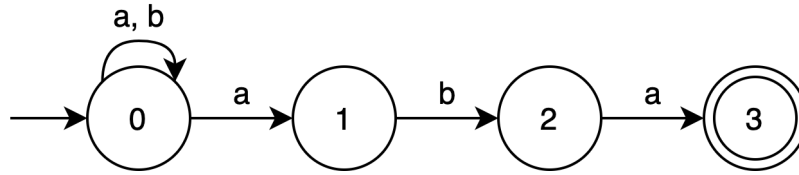


Рисунок 3. Автомат, принимающий строки, оканчивающиеся на aba

Где описание Δ всех возможных переходов задаётся таблицей:

Δ	a	b
0	$\{ 0, 1 \}$	$\{ 0 \}$
1	\emptyset	$\{ 2 \}$
2	$\{ 3 \}$	\emptyset
3	\emptyset	\emptyset

Когда мы видим в таблице символ пустого множества \emptyset , это означает, что для данного состояния и символа не существует перехода ни в одно состояние. Таким образом не может быть следа, проходящего через данное состояние и считывающего данный символ.

Теперь заметим, что любая строка из a и b будет принята автоматом, если она проходит через состояния 1, 2 и 3 в самом конце своего следа. Это возможно тогда и только тогда, когда строка заканчивается на aba . Таким образом допустимый язык для автомата на рисунке 3:

$$L(M) = \{xaba | x \in \{a, b\}^*\} \quad (4)$$