Лекция 5

5.1 Моноид переходов автомата

Напомним несколько основополагающих понятий из теории автоматов и формальных языков.

Детерминированный конечный автомат (кратко ДКА)– это пятерка $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F\rangle$, где Q – конечное *множество состояний*, Σ – конечный *входной алфавит*, $\delta\colon Q \times \Sigma \to Q$ – *функция переходов*, $q_0 \in Q$ – состояние, называемое *начальным*, а F – непустое подмножество в Q, состояния из которого называются *заключительными*.

Конечные автоматы допускают наглядное представление в виде помеченных графов. Мы предполагаем, что читатель знаком с идеей графа; уточним, какие именно графы мы будем использовать. В этих лекциях *граф* – это четвёрка множеств и отображений: множество *вершин* V, множество *ребер* E, отображение $h\colon E \to V$, которое отображает каждое ребро в его *начало* и отображение $t\colon E \to V$, которое отображает каждое ребро в его *конец*. Отметим, что в наших графах допускаются рёбра с общим началом и общим концом; такие рёбра называются *параллельными*. Допускаются и петли (*петлёй* называют ребро, начало и конец которого совпадают). Таким образом, наши графы на самом деле являются ориентированными мультиграфами, но, поскольку другие виды графов нам не встретятся, мы используем короткий термин. *Помеченный граф* оснащён дополнительным отображением $E \to \Lambda$, где Λ – множество, именуемое *алфавитом меток*; это отображение сопоставляет каждому ребру его *метку*. Ребро с началом v, концом v' и меткой a обозначим знакосочетанием $v \xrightarrow{a} v'$.

ДКА $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F\rangle$ изображается как помеченный граф с множеством вершин Q, алфавитом меток Σ и множеством ребер

$$\{q \xrightarrow{a} q' : q, q' \in Q,\ a \in \Sigma,\ \delta(q, a) = q'\}.$$

Итак, переход из состояния q в состояние q', вызванный входной буквой a, изображается ребром с началом q, концом q' и меткой a. Начальное состояние помечается стрелкой, входящей из «ниоткуда», а заключительные состояния – стрелкой, выходящей в «никуда», как показано на рис. 5.1 ниже. Здесь и далее рёбра с несколькими метками заменяют пучки параллельных ребер. Так, ребро $1 \xrightarrow{a,b} 2$ на рис. 5.1 заменяет параллельные рёбра $1 \xrightarrow{a} 2$ и $1 \xrightarrow{b} 2$.

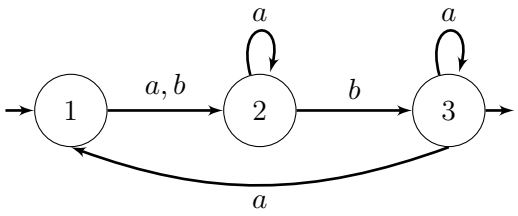

Рис. 5.1: Изображение конечного автомата

Напомним, что Σ^* обозначает множество всех слов над алфавитом Σ (включая пустое слово 1). Функцию переходов ДКА $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F\rangle$ распространяют до функции $Q \times \Sigma^* \to Q$ (по-прежнему обозначаемой δ) следующим образом. Для каждого $q \in Q$ положим $\delta(q, 1) := q$ и если $w = a_1 a_2 \cdots a_\ell$, где $a_1, \ldots, a_\ell \in \Sigma$, то $\delta(q, w) := \delta(...\delta(\delta(q, a_1), a_2), \ldots, a_\ell)$. Иначе говоря, пустое слово никак не изменяет состояние q, а непустое слово $w = a_1 \cdots a_\ell$ действует на q побуквенно: сначала к q применяется первая буква слова w, к тому состоянию, которое после этого получилось, применяется вторая буква этого слова, и т.д., пока не проработают все буквы слова w.

ДКА $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F\rangle$ *принимает* слово $w \in \Sigma^*$, если w – последовательность меток вдоль пути в $\mathcal{A}$, который начинается в состоянии q_0 и заканчивается в одном из состояний из F, т.е. если $\delta(q_0, w) \in F$. Множество всех слов, принимаемых $(\mathscr{A}, q_0, F)$, называется *языком, распознаваемым автоматом* $\mathscr{A}$. *Регулярный язык* над алфавитом Σ – это язык, распознаваемый каким-то ДКА с входным алфавитом Σ.

Сейчас мы покажем, что регулярные языки допускают очень естественную интерпретацию в терминах теории полугрупп. Прежде всего, заметим, что множество Σ^* является моноидом, если определить произведение слов как их *конкатенацию*: чтобы умножить слово u на слово v, просто приписывают v к u. Пустое слово 1 является единицей при таком умножении (что оправдывает наше обозначение для него).

Пусть M – моноид. Говорят, что язык $L \subseteq \Sigma^*$ распознаётся моноидом M, если существуют такой гомоморфизм $\varphi\colon \Sigma^* \to M$ и такое подмножество $P \subseteq M$, что для любого слова $w \in \Sigma^*$ выполнена эквивалентность

$$w \in L \iff w\varphi \in P.$$

Теорема 5.1. *Для любого языка L следующие условия эквивалентны:*
(1) L распознаётся конечным моноидом;
(2) L распознаётся детерминированным конечным автоматом.

Доказательство. (1) $\Rightarrow$ (2). Покажем, как построить ДКА, распознающий язык $L \subseteq \Sigma^*$, из конечного моноида M, распознающего этот язык с помощью гомоморфизма $\varphi\colon \Sigma^* \to M$ и подмножества $P \subseteq M$. Рассмотрим пятерку $\langle M, \Sigma, \delta, 1, P\rangle$, где 1 – единица моноида M, а функция $\delta\colon M \times \Sigma \to M$ определена правилом $\delta(m, a) := m(a\varphi)$. Проверим, что язык$L$ состоит в точности из слов, принимаемых так построенным ДКА.

Действительно, рассмотрим произвольное слово $w = a_1 a_2 \cdots a_n$, где $a_i \in \Sigma$. Тогда $\delta(1, w) = 1 \cdot (a_1\varphi)(a_2\varphi) \cdot \ldots \cdot (a_n\varphi)$. Поскольку φ — гомоморфизм, имеем $1 \cdot (a_1\varphi)(a_2\varphi) \cdot \ldots \cdot (a_n\varphi)(1 \cdot a_1 \cdot \ldots \cdot a_n)\varphi = 1 \cdot w\varphi = w\varphi$. Поэтому $\delta(1, w) \in P$ тогда и только тогда, когда $w\varphi \in P$, т.е. тогда и только тогда, когда $w \in L$.

(2) $\Rightarrow$ (1). Покажем, как построить конечный моноид, распознающий язык $L \subseteq \Sigma^*$ из ДКА $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, распознающего этот язык. Каждое слово $w \in \Sigma^*$ вызывает преобразование $q \mapsto \delta(q, w)$ множества Q. Мы обозначим это преобразование $\delta(\_, w)$, имея в виду, что элементы множества Q поставляются вместо пробела. Множество $M(\mathcal{A})$ всех преобразований вида $\delta(\_, w)$ замкнуто относительно умножения преобразований, поскольку, как легко понять, $\delta(\_, u)\delta(\_, v) = \delta(\_, uv)$, и содержит тождественное преобразование $\delta(\_, 1)$. Поэтому $M(\mathcal{A})$ есть подмоноид в моноиде T_Q всех преобразований множества Q; он называется *моноидом переходов автомата* $\mathcal{A}$. Поскольку множество Q конечно, моноид T_Q конечен, а следовательно, и $M(\mathcal{A})$ – конечный моноид.

Теперь нужно определить гомоморфизм $\varphi\colon \Sigma^* \to M(\mathcal{A})$ и подмножество $P \subseteq M(\mathcal{A})$, такие, что для любого слова $w \in \Sigma^*$

$$w \in L \Longleftrightarrow w\varphi \in P.$$

В качестве φ возьмем отображение, сопоставляющее каждому слову $w \in \Sigma^*$ вызванное им преобразование $\delta(\_, w)$. Тогда уже упомянутое равенство $\delta(\_, u)\delta(\_, v) = \delta(\_, uv)$ означает, что $(u\varphi)(v\varphi) = (uv)\varphi$, т.е. что φ – гомоморфизм. В качестве P возьмём множество $L\varphi$. Тогда если $w \in L$, то по определению $w\varphi \in P$. Обратно, пусть $w\varphi \in P$. Тогда существует такое слово $v \in L$, что $v\varphi = w\varphi$, т.е. $\delta(\_, w) = \delta(\_, v)$. Если преобразования $\delta(\_, w)$ и $\delta(\_, v)$ равны, то они одинаково действуют на все состояния автомата $\mathcal{A}$ и, в частности, на его начальное состояние q_0. Итак, $\delta(q_0, w) = \delta(q_0, v)$, но $\delta(q_0, v) \in F$, так как $v \in L$. Отсюда $\delta(q_0, w) \in F$, т.е. $w \in L$. □

5.2 Алгоритм построения моноида переходов

В доказательстве теоремы 5.1 введена важная конструкция моноида переходов автомата. Именно на этой конструкции основан алгебраический подход к задачам теории формальных языков, о котором упоминалось в §1.4, – она позволяет переводить эти задачи на язык алгебры. Впрочем, значение моноида переходов понятно и безотносительно к приложениям этой конструкции: если рассматривать автомат как вычислительное устройство, то моноид переходов этого автомата – не что иное как библиотека программ, содержащая описания всех доступных автомату вычислений.

Есть несложный по идее алгоритм моноида переходов, который проще всего объяснить на примере. Рассмотрим ДКА, изображенный на рис. 5.2. (Нетрудно видеть, что это минимальный по числу состояний автомат, распознающий язык $\Sigma^* aba \Sigma^*$ над алфавитом $\Sigma = \{a, b\}$. Впрочем, для алгоритма «происхождение» автомата, который он обрабатывает, не имеет значения.)

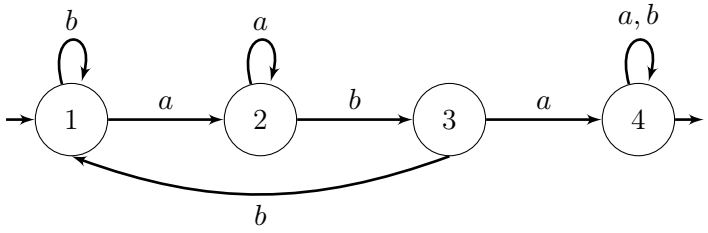

Рис. 5.2: Минимальный автомат языка $\Sigma^* aba \Sigma^*$

Построение моноида переходов начнем с таблицы, столбцы которой проиндексированы состояниями автомата, а строки – буквами алфавита. В ячейку на пересечении строки, отвечающей букве $c \in \Sigma$, и столбца, отвечающего состоянию $q \in \{1, 2, 3, 4\}$, запишем то состояние, в которое автомат перейдет из q под буквы c:

	1	2	3	4
a	2	2	4	4
b	1	3	1	4

Таким образом, таблица показывает, какие преобразования множества состояний вызваны буквами, т.е. словами длины 1. Допишем к таблице строки, показывающие преобразования множества состояний, вызванные словами длины 2. Для этого подействуем на строки таблицы буквами a и b. Действуя на первую строку буквой a, видим, что строка не изменяется, т.е. слово a^2 действует на каждое состояние так же, как буква a. Поэтому дописывать строку, отвечающую слову a^2, не будем, а вместо этого запишем в отдельную таблицу (*таблицу соотношений*) соотношение $a^2 = a$. Действуя на вторую строку буквой a, а также действуя на обе строки буквой b, получим новые строки; их мы допишем в таблицу. (Мы располагаем строки так, что индексирующие их слова идут в алфавитном порядке по возрастанию.)

	1	2	3	4
a	2	2	4	4
b	1	3	1	4
ab	3	3	4	4
ba	2	4	2	4
b^2	1	1	1	4

Продолжая процесс, допишем к таблице строки, показывающие преобразования множества состояний, вызванные словами длины 3. Для этого подействуем буквами a и b на три строки, добавленные на предыдущем шаге. Заметим, что в силу соотношения $a^2 = a$ сразу ясно, что применение буквы a к строке, отвечающей слову ba, не даст ничего нового. Кроме того, видно, что применение буквы b не меняет строку, отвечающую слову b^2. Это значит, что в таблицу соотношений надо добавить соотношение $b^3 = b^2$.

	1	2	3	4
a	2	2	4	4
b	1	3	1	4
ab	3	3	4	4
ba	2	4	2	4
b^2	1	1	1	4
aba	4	4	4	4
ab^2	1	1	4	4
bab	3	4	3	4
b^2a	2	2	2	4

Теперь допишем к таблице строки, показывающие преобразования множества состояний, вызванные словами длины 4. Для этого подействуем буквами a и b на четыре строки, добавленные на предыдущем шаге. Использование ранее зафиксированных соотношений позволяет не рассматривать применение буквы a к строкам, отвечающим словам aba и b^2a, и применение буквы b к строке, отвечающей слову ab^2. Кроме того, появятся новые соотношения: $(ab)^2 = aba, ab^2a = a$ и $(ba)^2 = aba$. Поэтому к таблице допишутся только две строки:

	1	2	3	4
a	2	2	4	4
b	1	3	1	4
ab	3	3	4	4
ba	2	4	2	4
b^2	1	1	1	4
aba	4	4	4	4
ab^2	1	1	4	4
bab	3	4	3	4
b^2a	2	2	2	4
bab^2	1	4	1	4
b^2ab	3	3	3	4

Попытка применить буквы a и b к этим двум строкам не даст новых строк, но даст еще одно соотношение, а именно, $b^2ab^2 = b^2$.

Тем самым, построение моноида переходов автомата, изображенного на рис. 5.2, закончено. Видим, что этот моноид состоит из 12 преобразований: тождественного преобразования и преобразований, отвечающих строкам таблицы, на которой закончилось построение.

Упражнение 5.1. *Вычислить отношения Грина в моноиде переходов автомата, изображенного на рис. 5.2.*

Важно понимать, что хотя описанный алгоритм концептуально прост, его реализация может оказаться весьма трудоемкой, поскольку моноид переходов даже небольшого автомата может оказаться очень большим. Так, для каждого n существует ДКА с n состояниями и тремя входными буквами, моноид переходов которого состоит из n^n преобразований.