1 КОНЕЧНЫЕ АВТОМАТЫ И РЕГУЛЯРНЫЕ МНОЖЕСТВА

1.1 Алфавиты, слова, языки

Материал этой и следующей глав относится к теории формальных языков. Базисными понятиями этой теории являются понятия алфавита, слова и языка.

Как и в языке естественном, скажем, русском, алфавитом мы называем множество букв. С точки зрения теории формальных языков, буква есть объект неделимый, атомный, структурой которого мы не интересуемся. Для нас важно уметь как-то отличать разные буквы друг от друга и устанавливать тождество одинаковых.

Более того, мы будем рассматривать только конечные алфавиты, т.е. конечные множества букв.

В дальнейшем *алфавит* — это конечное множество букв.

Примерами алфавитов служат:

- русский алфавит:

  АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

  и соответствующие строчные буквы

- латинский алфавит:

  $$ABCDEFGHIJKLMNOPQRSTUVWXYZ$$

  и соответствующие строчные буквы

- множество десятичных цифр:

  0123456789

В отличие от буквы, слова — объект делимый, составной. *Словом* в алфавите A мы будем называть конечную (возможно, пустую) последовательность букв алфавита A. Таким образом, понятие слова, вообще говоря, зависит от фиксированного алфавита. Однако в частности пустая последовательность букв всегда одинакова и от алфавита не зависит; ее мы будем называть пустым словом и часто обозначать буквой λ или буквой e.

Приведем примеры слов.

- ГРУША — слово в русском алфавите;
- $LEARNING$ — слово в латинском алфавите;
- 945 — слово в алфавите цифр.

Теория формальных языков не занимается смыслом слов. Чтобы не путать наше понятие слова с обычным понятием, скажем, слова в русском языке, приведем еще один пример:

- БЛТПЕРН — также слово в русском алфавите (но не в русском языке!).

Длиной слова называют число букв в этом слове. Длину слова w обозначают $|w|$. При этом разные вхождения одной и той же буквы считаются отдельно, например, длина слова $'fff'$ равна 3.

Сверхслово — это бесконечное слово, т.е. бесконечная счетная последовательность букв алфавита. Если говорить точно, сверхслово — это отображение множества положительных натуральных чисел в алфавит.

Слово (сверхслово) есть упорядоченная последовательность букв, поэтому осмысленны выражения первая буква слова (сверхслова), последняя буква слова в тех случаях, когда слово (сверхслово) не пусто, или пятая буква слова (сверхслова) и т.д. в тех случаях, когда слово содержит нужное число букв. Мы будем считать, что на бумаге слово записывается слева направо, т.е. первой буквой приведенного слова в латинском алфавите является буква латинского алфавита L.

Выше говорилось, что мы умеем устанавливать тождество одинаковых букв из алфавита и различать разные буквы. Тем самым мы умеем отличать друг от друга разные слова и сверхслова алфавита, или же устанавливать тождество двух слов (сверхслов). Например, слова русского алфавита из приведенных примеров различны. Если v и w — два слова (сверхслова) в одном и том же алфавите, то мы пишем $v = w$, если они тождественны, и $v \neq w$, если они различны.

Мы уже использовали выше понятие "язык" ("русский язык"). В понятие "язык" человечество вкладывает много оттенков; из всех них теория формальных языков выделяет один: ***язык*** в алфавите A — это некоторое множество слов в алфавите A. При этом не предполагается, что язык конечен.

Приведем примеры языков.

Следующие три слова, заключенные в фигурные скобки и разделенные запятыми, являются (конечным) языком в латинском алфавите: $\{ASD, FE, ERY\}$.

Пример бесконечного языка — множество всевозможных целых десятичных чисел без знака. Вот некоторые элементы этого множества (они заключены в кавычки):

$$'0',\ '00',\ '018703',\ '1986'.$$

Отметим, что числа $'0'$ и $'00'$ с нашей точки зрения различны.

Аналогичным образом определяется сверх-язык в алфавите A — некоторое множество сверхслов в алфавите A.

Пустой, т.е. не содержащий элементов, язык (сверх-язык) мы будем обозначать символом $\emptyset$.

Введенные нами отношения тождества и различия на словах (сверхсловах) ($=$ и $\neq$) индуцируют отношения тождества и различия на языках и сверх-языках. Именно, мы говорим, что два языка (сверх-языка) L и M равны (и пишем $L = M$), если они состоят из одних и тех же (в смысле равенства) слов (сверхслов). Это означает, что для каждого слова одного языка найдется равное ему слово в другом языке. В противном случае мы говорим, что L и M не равны (и пишем $L \neq M$).

1.2 Операции над словами и языками

Зафиксируем некоторый алфавит (конечное множество букв) A.

Рассмотрим некоторые слова v и w в этом алфавите. Напомним, что, по определению, v и w являются конечными последовательностями букв в алфавите A.

Над этими словами можно определить некоторые операции, которые, по одному слову, или по двум словам, дают некоторое новое слово в том же алфавите A.

Ясно, что таких операций можно определить сколь угодно много. Мы и будем их определять по мере необходимости. В частности, нам будет нужна операция склейки, или конкатенации двух слов. *Конкатенацией* слов v и w мы называем последовательность букв того же алфавита, в которой сначала идут буквы из последовательности v, а затем — буквы из последовательности w. Конкатенацию слов v и w мы обозначаем vw.

Например, конкатенацией слов $'anm'$ и $'ghj'$ в латинском алфавите будет слово $'anmghj'$.

Операция конкатенации неприменима к сверхсловам. Можно, однако, рассмотреть конкатенацию слова w со сверхсловом s, которую мы будем обозначать ws.

Через операцию конкатенации определяется понятие *префикса* слова (сверхслова). Именно, пусть s — слово (сверхслово). Слово w называется префиксом s, если существует слово (сверхслово) t такое, что $s = wt$. Если t — непустое слово, т.е. $|t| \neq 0$, то w называется собственным префиксом s. Слово (сверхслово) t при этом называется *суффиксом* слова s (собственным суффиксом, если $|w| \neq 0$).

Подсловом слова (сверхслова) s называется любой префикс любого суффикса s. Подслово называется собственным, если оно не совпадает с s.

Нам также понадобится операция степени слова. i-я степень слова w, где i — натуральное число, определяется через операцию конкатенации по индукции:

Базис: $w^0 = \lambda$, где, напомним, λ — пустое слово;

Шаг индукции: $w^{i+1} = w^i w$, т.е.является конкатенацией слов w^i и w.

Рассмотрим теперь некоторые языки L и M в алфавите A. На них мы также можем определить операции, которые порождают новые языки в том же алфавите.

Нам пока понадобится несколько операций.

Первая операция — теоретико-множественное *объединение*. Так как язык, по определению, является множеством слов, можно рассмотреть объединение двух языков как объединение двух множеств. Объединение языков L и M обозначается $L \bigcup M$ и состоит из тех и только тех слов в алфавите A, которые входят либо в L, либо в M. В обычной математической записи это записывается так:

$$L \bigcup M = \{w \mid \quad w \in L \ \text{ или } \ w \in M\}.$$

Аналогично мы вводим теоретико-множественные операции *разности* $L \setminus M$ и *пересечения* $L \bigcap M$:

$$L \setminus M = \{w \mid \quad w \in L \ \text{ и } \ w \notin M\};$$
$$L \bigcap M = \{w \mid \quad w \in L \ \text{ и } \ w \in M\}.$$

Введенные нами операции на словах дополнительно индуцируют соответствующие операции и на языках.

Под конкатенацией языков L и M, обозначаемой LM, мы понимаем множество слов

$$\{vw \mid \quad v \in L \ \text{ и } \ w \in M\}.$$

Под i-ой степенью языка L, обозначаемой L^i, мы понимаем следующее определенное по индукции множество:

$L^0 = \{\lambda\}$, где λ — пустое слово;

$$L^{i+1} = L^i L.$$

Наконец, итерацией языка L, обозначаемой L^*, мы будем называть бесконечное объединение натуральных степеней языка L:

$$L^* = \{w \mid \text{ существует такое } i \geq 0, \text{ что } w \in L^i\}.$$

Операции объединения, пересечения и разности определяются для сверх-языков так же, как и для языков. Можно также рассмотреть операцию конкатенации языка со сверх-языком.

Существуют важные операции определения сверх-языков через языки.

Пусть, по-прежнему, L — язык.

Пределом языка L называется сверхслово t, которое имеет в L бесконечно много префиксов (т.е.пересечение L с множеством всех префиксов t бесконечно). Отметим, что язык может иметь несколько, или даже бесконечно много, пределов.

Ультрапрефиксом языка L называется сверхслово t, всякий префикс которого является также префиксом некоторого слова из L (т.е. пересечение множества всех префиксов t с множеством всех префиксов всех слов из L бесконечно). Отметим, что язык может иметь несколько, или даже бесконечно много, ультрапрефиксов. Очевидно, что всякий предел языка является также его ультрапрефиксом. Обратное же неверно.

Множество всех пределов языка L обозначается $Lim(L)$, а множество всех ультрапрефиксов L — $Adh(L)$. $Lim(L)$ и $Adh(L)$ являются сверх-языками.

Приведем примеры.

Пример 1.2.1. *Рассмотрим алфавит $A = \{0, 1\}$. A сам по себе является конечным языком. Рассмотрим язык A^*. Ясно, что A^* является множеством всех слов в алфавите A. Очевидно также, что $Lim(A^*)$ и $Adh(A^*)$ равны и каждое из них есть множество всех сверхслов в алфавите A.*

Пример 1.2.2. *Рассмотрим тот же алфавит A и язык L, составленный из всех слов вида* 00...01, *т.е. из слов* 01, 001, 0001, ... *Очевидно, что язык L не имеет ни одного предела. В то же время $Adh(L)$ непусто и состоит в точности из единственного сверхслова в алфавите* $\{0\}$.

Упражнения

Упражнение 1.2.1. *В обозначениях примера 1.2.2 покажите, что* $Adh(LA^*)$ *и* $Adh(A^*L)$ *бесконечны. Что это за множества?*

Упражнение 1.2.2. *Покажите, что итерация языка, который содержит по крайней мере два элемента, всегда является бесконечным языком. Привести (единственный!) пример, когда итерация непустого языка — конечный язык.*

Упражнение 1.2.3 (**). *Приведите пример сверхслова* w *в алфавите* $\{1, 2, 3\}$ *такого, что ни для какого непустого слова* v *в этом же алфавите* v^2 *не является подсловом* w.

Упражнение 1.2.4 (*). *Покажите, что всякий бесконечный язык имеет хотя бы один ультрапрефикс.*

1.3 Регулярные выражения и регулярные множества

До сих пор мы не накладывали на языки никаких ограничений, считая их произвольными множествами слов. Для изучения языков, однако, важно по крайней мере уметь как-то изображать эти языки. Если язык конечен, то можно изобразить его на бумаге путем перечисления всех его элементов. Однако этот способ не годится ни для какого бесконечного языка.

Способ, который мы сейчас собираемся рассмотреть — задание языков при помощи операций над конечными языками. В предыдущем параграфе мы видели, что операция итерации языка позволяет получить из конечного языка бесконечный.

Определение 1.3.1 (регулярности языка). *Язык называется регулярным, если его можно получить из конечного числа конечных языков применением конечного числа операций объединения, конкатенации и итерации языков. Регулярные языки часто называют регулярными множествами.*

Рассмотрим теперь понятие регулярного выражения над алфавитом A.

Определение 1.3.2 (регулярного выражения). *Пусть* A *— алфавит. Регулярное выражение над алфавитом* A *определяется по индукции. Одновременно по индукции мы будем каждому регулярному выражению сопоставлять регулярный язык в рассматриваемом алфавите:*

1. *0 — регулярное выражение; ему соответствует пустой регулярный язык.*
2. *λ — регулярное выражение; ему соответствует одноэлементный регулярный язык, состоящий из одного пустого слова.*
3. *Если $a \in A$, то a — регулярное выражение; ему соответствует регулярный язык $\{a\}$.*
4. *Если r, r' — регулярные выражения, которым соответствуют регулярные языки R, R', то (rr'), $(r+r')$, r^* — регулярные выражения; им соответствуют регулярные языки RR', $R\bigcup R'$, R^*.*

По определению, всякому регулярному выражению r соответствует некоторый регулярный язык R. Мы будем говорить, что регулярное выражение r задает этот регулярный язык R и писать $L(r) = R$. Имеет место очевидная

Теорема 1.1. *Всякий регулярный язык в алфавите A задается некоторым регулярным выражением в алфавите A.*

Доказательство. Следует проводить по индукции. Очевидно, что всякое конечное множество слов может быть задано регулярным выражением с использованием только пунктов 1, 2, 3 определения 1.3.2 и операции + (объединение). Этим доказан базис индукции. Индукционный шаг также очевиден. ∎

Легко проверить, что регулярная операция + коммутативна и ассоциативна, а операция конкатенации ассоциативна и дистрибутивна по отношению к операции +.

Договоримся о приоритете операций и о правилах опускания скобок в регулярных выражениях. Будем считать, что сначала всегда применяется операция итерации *, затем конкатенация, и лишь затем — +. Например, в выражении $10 + 01^*$ следует расставить скобки так:

$$(10) + (0(1^*)).$$

Для задания языков можно рассмотреть и другие операции над языками, введенные в предыдущем параграфе, скажем, операции пересечения, разности и степени. В соответствии с этим можно рассмотреть другие классы выражений, задающих языки.

В частности, рассматриваются полурасширенные регулярные выражения, задающие языки аналогично регулярным выражениям, но дополнительно использующие операцию взятия пересечения языков &, т.е.

выражению $(r'\&r)$ соответствует язык $R'\bigcap R$ (см. пункт 4 определения 1.3.2).

Рассматриваются также расширенные регулярные выражения, которые, кроме операции пересечения & и регулярных операций, используют еще операцию взятия разности $-$.

Интересно, естественно, насколько широкие классы языков можно задавать регулярными выражениями.

Приведем примеры.

Пример 1.3.1. *Рассмотрим алфавит $A = \{0, 1\}$. В примере 1.2.1 уже были приведены примеры простых регулярных языков над этим алфавитом. Рассмотрим язык L в алфавите A, состоящий из всех слов ровно с двумя буквами 0. Он регулярен. Чтобы показать это, запишем задающее его регулярное выражение:* $1^*01^*01^*$.

Пример 1.3.2. *В том же алфавите множество слов четной длины задается следующим регулярным выражением:* $(01 + 10 + 11 + 00)^*$.

Упражнения

Упражнение 1.3.1. *Приведите примеры регулярных языков и задающих их регулярных выражений.*

Упражнение 1.3.2 (*). *Регулярен ли язык, полученный объединением всевозможных языков вида 0^i1^i?*

Упражнение 1.3.3 (*). *Покажите, что всякий язык, заданный расширенным регулярным выражением, регулярен.*

Упражнение 1.3.4 (*). *Покажите, что никакой регулярный язык не может иметь счетного числа ультрапрефиксов.*

Упражнение 1.3.5. *Расставьте скобки в следующих регулярных выражениях:*

1).

$$1 + 010 + 11^*,$$

2).

$$11^*011^*0^*.$$

Упражнение 1.3.6 (*). *Напишите регулярные выражения, задающие следующие языки:*

1).

$$\{0, 1\}^* \setminus (\{0\}\{1\}\{0\}),$$

2).

$$\{0,1\}^* \setminus (\{1\}\{1\}\{0,1\}).$$

1.4 Конечные автоматы. Распознавание регулярных множеств

Рассмотренные в предыдущем параграфе регулярные выражения решают проблему обозначения некоторых языков, а именно, регулярных языков. Действительно, мы можем записать на бумаге регулярное выражение и считать, что тем самым мы описали язык, о котором идет речь.

Однако такой способ обозначения языков сам по себе практически бессмыслен, так как, вообще говоря, мы не решаем пока главные две задачи:

ЗАДАЧА 1. Допустим, что на бумаге записано регулярное выражение r. Принадлежит ли некоторое конкретное слово w регулярному языку $L(r)$?

ЗАДАЧА 2. Допустим, что на бумаге записаны два регулярных выражения r и r' в алфавите A. Обозначают ли они один и тот же язык? Другими словами, верно ли, что $L(r) = L(r')$?

Ниже мы покажем, что для регулярных языков в действительности обе эти задачи можно решить.

Начнем мы с решения первой задачи. Что значит уметь решать проблему принадлежности произвольного слова некоторому фиксированному языку? Это значит описать процедуру решения этой проблемы таким не допускающим двоякого толкования способом, чтобы можно было всегда по произвольному слову определить его принадлежность данному языку, используя этот способ.

Мы рассмотрим устройства, которые называются конечными автоматами и предназначены для распознавания слов в данном алфавите. Каждое такое устройство имеет конечное число состояний и всегда находится в одном из этих состояний. Автомат обладает разбитой на ячейки односторонней лентой, на которой можно записать слово, и головкой, которая осматривает одну из ячеек ленты. В начальный момент головка обозревает самую левую ячейку ленты.

Такт работы конечного автомата состоит в сдвиге головки вправо на одну букву и переходе в новое состояние. Если головку сдвинуть вправо невозможно из-за того, что кончилось записанное на ленте слово, автомат останавливается. Новое состояние выбирается автоматом в зависимости от старого состояния и буквы, обозревавшейся головкой, в соответствии с конечным набором правил.

Приведем формальное определение.

Определение 1.4.1 (конечного автомата). *Недетерминированным конечным автоматом (сокращенно, НКА) называется упорядоченная совокупность пяти объектов* $\langle A, Q, q_0, F, R\rangle$, *где:*

- A — *конечный алфавит, в котором пишутся слова распознаваемого языка;*
- Q — *конечный алфавит состояний автомата;*
- $q_0 \in Q$ — *начальное состояние;*
- F — *подмножество* Q, *так называемые принимающие или заключительные состояния;*
- R — *конечное множество правил. Правилом называется тройка* $\langle q, a, q'\rangle$, *являющаяся элементом декартова произведения множеств* $Q \times A \times Q$.

Итак, пусть на ленте недетерминированного конечного автомата $J = \langle A, Q, q_0, F, R\rangle$ записано слово w из A^*. Головка автомата в начальный момент времени находится на самой левой букве слова w. Сам автомат находится в состоянии q_0.

Такт работы автомата J состоит в том, что он переходит в новое состояние, и затем сдвигает головку на одну букву вправо. Если сдвинуть головку уже нельзя, то автомат останавливается, в противном случае он делает следующий такт, и т.д.

Пусть головка автомата J обозревает в данный момент на ленте букву a из A, а сам автомат находится в состоянии q из Q. Тогда новое состояние на этом такте, q', выбирается произвольно с условием, что тройка $\langle q, a, q'\rangle$ принадлежит множеству R правил J. В частности, может случиться так, что ни для какого q' тройка $\langle q, a, q'\rangle$ не принадлежит множеству R. Тогда в качестве нового состояния выбирается само q.

Может быть и так, что для данных q, a существует несколько

$$q_1, \ldots, q_k$$

таких, что

$$\langle q, a, q_1\rangle, \ldots, \langle q, a, q_k\rangle$$

все принадлежат R. В таком случае автомат J обладает свободой выбора конкретного q_i в качестве следующего состояния.

Легко видеть, что на слове w автомат J совершит ровно $|w|$ тактов. Состояние, в котором был автомат на последнем из этих тактов и на котором он остановился, будем называть заключительным состоянием.

Будем говорить, что автомат J останавливается на слове w в алфавите A, если при некотором способе работы заключительным состоянием J на слове w будет состояние, принадлежащее множеству F.

Пусть автомат J находится в состоянии q, обозревает букву a на входной ленте, и справа от его головки на ленте записано слово w.

Определение 1.4.2 (работы конечного автомата). *Если $q' \in Q$ такое, что $\langle q, a, q' \rangle \in R$, то будем писать $(q, aw) \mid\Rightarrow_J (q', w)$. Если же ни для какого q' неверно, что $\langle q, a, q' \rangle \in R$, то будем писать $(q, aw) \mid\Rightarrow_J (q, w)$.*

Если $w = a_0 a_1 ... a_n \in A^*$, $m \le n$ и $q_{i_0}, q_{i_1}, \dots, q_{i_m} \in Q$ таковы, что

$$(q_{i_0}, a_0 \dots a_n) \mid\Rightarrow_J (q_{i_1}, a_1 \dots a_n) \mid\Rightarrow_J \cdots \mid\Rightarrow_J (q_{i_m}, a_m ... a_n),$$

то будем писать $(q_{i_0}, a_0 \dots a_n) \mid\Rightarrow_J^+ (q_{i_m}, a_m ... a_n)$.

Автомат J останавливается на слове $w \in A$ тогда и только тогда, когда существует такое $q \in F$, что $(q_0, w) \mid\Rightarrow_J^+ (q, e)$, где e — пустое слово. Автомат J останавливается на пустом слове e тогда и только тогда, когда $q_0 \in F$.

Определение 1.4.3 (детерминированного конечного автомата). *Пусть $J = \langle A, Q, q_0, F, R \rangle$ — недетерминированный конечный автомат. Будем называть J детерминированным конечным автоматом (ДКА), если для произвольных q из Q и a из A существует самое большее одно состояние q' из Q такое, что тройка $\langle q, a, q' \rangle$ принадлежит R. Тем самым R в случае детерминированного автомата определяет частичное отображение $R : (Q \times A) \to Q$. Именно,*

$R(q, a) = q'$, если $\langle q, a, q' \rangle \in R$.

Детерминированные конечные автоматы являются важным частным случаем недетерминированных, так как по ДКА и слову w можно легко определить, воспринимает ли автомат это слово.

Определение 1.4.4 (языка конечного автомата). *Языком недетерминированного конечного автомата $J = \langle A, Q, q_0, F, R \rangle$ называется множество всех слов в алфавите A, на которых останавливается автомат J. Язык автомата J мы будем обозначать через $L(J)$. Мы* *будем также говорить, что язык $L(J)$ распознается (или задается) автоматом J.*

Определение 1.4.5 (эквивалентности автоматов). *Недетерминированные конечные автоматы $J = \langle A, Q, q_0, F, R\rangle$, $J' = \langle A, Q', q', F', R'\rangle$ называются эквивалентными, если $L(J) = L(J')$.*

Фундаментальным фактом теории конечных автоматов является то, что недетерминизм несуществен для конечного автомата.

Теорема 1.2. *Пусть $J = \langle A, Q, q_0, F, R\rangle$ — недетерминированный конечный автомат. Существует детерминированный конечный автомат $J' = \langle A, Q', q', F', R'\rangle$, распознающий тот же язык.*

Доказательство. Рассмотрим автомат

$$J = \langle A, Q, q_0, F, R\rangle.$$

Нам необходимо построить детерминированный конечный автомат

$$J' = \langle A, Q', q', F', R'\rangle,$$

эквивалентный J.

В качестве множества Q' состояний J' рассмотрим множество

$$\{T \mid\ T \subseteq Q,\ T \neq \emptyset\}.$$

Таким образом, для каждого непустого подмножества Q у нас в Q' есть свое собственное состояние.

В качестве начального состояния q' автомата J' рассмотрим состояние $\{q_0\}$.

В автомате J' мы хотим как бы промоделировать совместное исполнение всех возможных вариантов работы автомата J. Начнем с содержательных пояснений. В начальный момент времени автомат J находится в состоянии q_0 и обозревает первую букву слова, скажем, букву a_0. Поскольку J — недетерминированный автомат, он может сработать в этих условиях по разному. Пусть

$$\langle q_0, a_0, q_{i_1}\rangle, \langle q_0, a_0, q_{i_2}\rangle, \ldots, \langle q_0, a_0, q_{i_r}\rangle$$

будут все элементы R вида $\langle q_0, a_0, q\rangle$. Тогда J может на этом такте перейти в одно из состояний $q_{i_1}, q_{i_2}, \ldots, q_{i_r}$.

Этот первый такт работы мы смоделируем в J', включив в R' одно правило:

$$\langle \{q_0\}, a_0, \{q_{i_1}, q_{i_2}, \ldots, q_{i_r}\}\rangle. \tag{1}$$

По определению, $\{q_{i_1}, q_{i_2}, \ldots, q_{i_r}\}$ является состоянием автомата J. Если же выше $r = 0$, то есть ни для какого $q \in Q$ тройка $\langle q_0, a_0, q\rangle$ не принадлежит R, то включим в R' вместо правила (1) правило

$$\langle\{q_0\}, a_0, \{q_0\}\rangle.$$

Фактически, находясь в состоянии $\{q_{i_1}, q_{i_2}, \ldots, q_{i_r}\}$, мы понимаем, что J мог бы находиться в одном из состояний $q_{i_1}, q_{i_2}, \ldots, q_{i_r}$.

Пусть, в некоторый момент времени, автомат J' обработал буквы $a_0, a_1, \ldots, a_{m-1}$ входного слова, при этом J' находится в состоянии

$$\{q_{i_1}, q_{i_2}, \ldots, q_{i_r}\}$$

и обозревает букву a_m входного слова. Ясно из построения, что автомат J, обработав буквы $a_0, a_1, ...a_{m-1}$ входного слова, мог бы находится в любом из состояний $q_{i_1}, q_{i_2}, \ldots, q_{i_r}$.

Поочередно рассмотрим все эти возможности. Если J' находится в состоянии q_{i_n} , то, обозревая букву a_m, он может перейти в одно из состояний $q_{k_1}, q_{k_2}, \ldots, q_{k_s} \in Q$, где

$$\langle q_{i_n}, a_m, q_{k_1}\rangle, \langle q_{i_n}, a_m, q_{k_2}\rangle, \ldots, \langle q_{i_n}, a_m, q_{k_s}\rangle$$

представляют все правила из R вида $\langle q_{i_n}, a_m, q\rangle$. Если же в R нет правил вида $\langle q_{i_n}, a_m, q\rangle$, то J может перейти только в состояние q_{i_n}. Таким образом, единственное правило из R' для состояния $\{q_{i_1}, q_{i_2}, \ldots, q_{i_r}\}$ и буквы a_m должно учитывать все эти возможности для всех входящих в него состояний первоначального автомата, то есть иметь вид:

$$\langle\{q_{i_1}, q_{i_2}, \ldots, q_{i_r}\}, a_m, X\rangle,$$

где X содержит все такие q, что $\langle q_{i_j}, a_m, q\rangle \in R$ для некоторого $j \in \{1, \ldots, r\}$, а также все такие q_{i_j} для $j \in \{1, \ldots, r\}$, что $\langle q_{i_j}, a_m, q\rangle \notin R$ для всех $q \in Q$, и не содержит никаких других элементов.

Формально, построим множество R' следующим образом.

Пусть a — буква из A и $T = \{q_1, \ldots, q_n\}$ — некоторое непустое подмножество Q. Пусть T' содержит все такие q, что $\langle q_j, a, q\rangle \in R$ для некоторого $j \in \{1, \ldots, n\}$, а также все такие q_j для $j \in \{1, \ldots, n\}$, что $\langle q_j, a, q\rangle \notin R$ для всех $q \in Q$, и не содержит никаких других элементов. Тогда включим тройку $\langle T, a, T'\rangle$ в R'.

Теперь положим

$$F' = \{T \in Q' \mid T \bigcap F \neq \emptyset\}.$$

Докажем утверждение теоремы формально.

1. Пусть $w = a_m a_{m+1} ... a_k \in A^*$ — некоторое слово, $T \subseteq Q$, $q \in T$. Индукцией по длине слова w покажем, что если

$$(q, w) \mid\Rightarrow_J^+ (q', e),$$

то для некоторого $T' \subseteq Q$ такого, что $q' \in T'$ имеет место

$$(T, w) \mid\Rightarrow_{J'}^+ (T', e),$$

где e — пустое слово.

Если $m = k$, то есть $w = a_m$, то либо q' таково, что $\langle q, a_m, q' \rangle \in R$, либо ни для какого q'' неверно, что $\langle q, a_m, q'' \rangle \in R$, и тогда $q' = q$. В обоих случаях ясно, что если рассмотреть то единственное T', что $\langle T, a_m, T' \rangle \in R'$, то $q' \in T'$. Тогда $(T, a_m) \mid\Rightarrow_{J'} (T', e)$, то есть $(T, w) \mid\Rightarrow_{J'}^+ (T', e)$ и утверждение доказано.

Пусть $m < k$. Так как $(q, w) \mid\Rightarrow_J^+ (q', e)$, то существует в Q такое состояние q_{m+1}, что

$$(q, w) \mid\Rightarrow_J (q_{m+1}, a_{m+1} a_{m+2} ... a_k),$$

$$(q_{m+1}, a_{m+1} a_{m+2} ... a_k) \mid\Rightarrow_J^+ (q', e).$$

Рассуждая как и в предыдущем случае, можно предполагать, что существует $T'' \subseteq Q$ такое, что $q_{m+1} \in T''$ и $(T, w) \mid\Rightarrow_{J'} (T'', a_{m+1} a_{m+2} ... a_k)$.

По индукции, существует $T' \subseteq Q, q' \in T'$, такие что

$$(T'', a_{m+1} ... a_k) \mid\Rightarrow_{J'}^+ (T', e).$$

Но тогда

$$(T, w) \mid\Rightarrow_{J'}^+ (T', e).$$

Утверждение доказано.

В частности, для произвольного $w \in L(J)$ существует $q' \in F$ такое, что $(q_0, w) \mid\Rightarrow_J^+ (q', e)$. Тогда, как доказано, существует $T' \subseteq Q$ такое, что $(\{q_0\}, w) \mid\Rightarrow_{J'}^+ (T', e)$ и $q' \in T'$. Так как $q' \in F$, то $T' \in F'$, по определению. Значит, $w \in L(J')$.

Итак, $L(J) \subseteq L(J')$.

Теперь докажем обратное, то есть что $L(J') \subseteq L(J)$.

2. Пусть $w = a_m a_{m+1} ... a_k \in A^*$ — некоторое слово, $T, T' \subseteq Q$, $q' \in T'$. Пусть

$$(T, w) \mid\Rightarrow_{J'}^+ (T', e).$$

Докажем, что существует $q \in T$ такое, что

$$(q, w) \models^{+}_{J} (q', e).$$

Пусть $m = k$. Тогда $(T, a_m) \models_{J'} (T', e)$ и по определению, существует такое $q \in T$, что либо $\langle q, a_m, q' \rangle \in R$, либо $q = q'$ и не существует никакого такого $q'' \in Q$, для которого $\langle q', a_m, q'' \rangle \in R$.

Ясно, что тогда существует $q \in T$, что $(q, a_m) \models_{J} (q', e)$, то есть $(q, w) \models_{J} (q', e)$, и утверждение доказано.

Пусть $k > m$. Так как

$$(T, w) \models^{+}_{J'} (T', e),$$

то

$$(T, w) \models^{+}_{J'} (T'', a_k),$$
$$(T'', a_k) \models_{J'} (T', e)$$

для некоторого $T'' \subseteq Q$. Но тогда

$$(T, a_m \dots a_{k-1}) \models^{+}_{J'} (T'', e).$$

По индукции, существует $q'' \in T''$ такое, что

$$(q'', a_k) \models_{J} (q', e).$$

По индукции также существует такое $q \in T$, что

$$(q, a_m \dots a_{k-1}) \models^{+}_{J} (q'', e).$$

Но это значит, что

$$(q, a_m \dots a_k) \models^{+}_{J} (q'', a_k).$$

Сопоставив, получим, что

$$(q, a_m \dots a_k) \models^{+}_{J} (q', e),$$

чем доказано утверждение.

В частности, пусть $w \in L(J')$. Тогда существует $T \in F'$ такое, что

$$(\{q_0\}, w) \models^{+}_{J'} (T, e).$$

Так как $T \in F'$, то по определению найдется $q' \in F$ такое, что $q' \in T$. Но тогда, по доказанному, существует $q \in \{q_0\}$ такое, что $(q, w) \models^{+}_{J} (q', e)$, то есть $(q_0, w) \models^{+}_{J} (q', e)$, и $q' \in F$. Значит, $w \in L(J)$.

Итак, $L(J') \subseteq L(J)$. ∎

Из доказательства видно, что детерминированный автомат, который моделирует недетерминированный, имеет более сложное описание (в частности, большее число состояний). Это можно было бы доказать и формально.

Вернемся к регулярным множествам. Следующая теорема показывает, что задача 1 о распознавании принадлежности слова регулярному множеству решается уже конечными автоматами.

Теорема 1.3. *Пусть r — регулярное выражение в алфавите A, $R = L(r)$. Существует такой НКА*

$$J = \langle A, Q, q_0, F, P \rangle,$$

что $R = L(J)$.

Доказательство. Мы явно построим J индукцией по сложности построения r.

Базис.

1. Если r есть регулярное выражение 0, положим $Q = \{q_0\}$, $F = P = \emptyset$.

2. Если r есть регулярное выражение λ, положим

$$Q = \{q_0, q_1\},$$

$$F = \{q_0\},$$

$$P = \{\langle q_0, a, q_1 \rangle \mid a \in A\}.$$

3. Если r есть регулярное выражение $a \in A$, положим

$$Q = \{q_0, q_1, q_2\},$$

$$F = \{q_1\},$$

$$P = \{\langle q_0, a, q_1 \rangle\} \bigcup \{\langle q_0, b, q_2 \rangle \mid \ b \in A, \ b \neq a\} \bigcup \{\langle q_1, b, q_2 \rangle \mid \ b \in A\}.$$

Шаг индукции. Пусть r', r'' — регулярные выражения и пусть по индукции им уже сопоставлены такие недетерминированные конечные автоматы

$$J' = \langle A, Q', q', F', P' \rangle,$$

$$J'' = \langle A, Q'', q'', F'', P'' \rangle,$$

что $L(J') = L(r')$ и $L(J'') = L(r'')$. Считаем, что $Q' \bigcap Q'' = \emptyset$.

Без потери общности можно считать, что в J' для всякого состояния $q' \in Q'$ и всякого $a \in A$ существует $q \in Q'$, что $\langle q', a, q\rangle \in P'$. Того же будем требовать и от автомата J''.

1. Пусть r есть регулярное выражение $(r'r'')$. Положим

$$J = \langle A, Q, q', F, P\rangle,$$

где $Q = Q' \bigcup Q''$, $F = F''$,

$$P = P' \bigcup P'' \bigcup \{\langle q, a, q''\rangle \mid \; q \in Q', \; a \in A, \; q' \in F', \; \langle q, a, q'\rangle \in P'\}.$$

Если $q'' \in F''$, добавим еще в F все состояния из F', то есть в этом случае $F = F' \bigcup F''$.

Другими словами, автомат J представляет собой объединение автоматов J' и J'' и работает так. Сначала он работает как автомат J', с тем добавлением, что вместо перехода в любое из заключительных состояний автомата J', может перейти в начальное состояние автомата J''. Перейдя в это состояние, он далее работает как автомат J''.

2. Пусть r есть регулярное множество $(r' + r'')$. Положим

$$J = \langle A, Q, q_0, F, P\rangle,$$

где

$$Q = Q' \bigcup Q'' \bigcup \{q_0\},$$

$$q_0 \notin (Q' \bigcup Q''),$$

$$F = F' \bigcup F'',$$

$$P = P' \bigcup P'' \bigcup \{\langle q_0, a, q\rangle \mid \; a \in A, (\langle q', a, q\rangle \in P' \;\text{ или }\; \langle q'', a, q\rangle \in P'')\}.$$

Если $q' \in F'$ или $q'' \in F''$, то дополнительно включим q_0 в F (для учета пустого слова).

Другими словами, из начального состояния q_0 автомата J разрешены в точности все переходы, разрешенные из начального состояния q' автомата J' и из начального состояния q'' автомата J''.

3. Пусть r есть регулярное множество $(r')^*$. Положим

$$J = \langle A, Q, q_0, F, P\rangle,$$

где

$$Q = Q' \bigcup \{q_0\},$$

$$q_0 \notin Q',$$

$$F = F' \bigcup \{q_0\},$$

$$P = P' \bigcup \{\langle q, a, q' \rangle \mid \; q_1 \in F',\; a \in A,\; \langle q, a, q_1 \rangle \in P'\} \bigcup$$

$$\{\langle q_0, a, q \rangle \mid \; a \in A,\; q \in Q',\; \langle q', a, q \rangle \in P'\}.$$

Построение здесь аналогично случаю конкатенации, только произведенному с самим собой, что обеспечивает "зацикливание". Новое состояние q_0 введено для обеспечения распознавания пустого слова.

Очевидно, что для так построенного автомата J будет иметь место равенство $L(J) = L(r)$. ■

Итак, мы показали, что всякое регулярное множество может быть распознано конечным автоматом.

Оказывается, что и наоборот, всякое множество, которое может быть распознано конечным автоматом, является регулярным. Мы отложим формальное доказательство этой теоремы. Однако в следующем параграфе сам этот факт (об эквивалентности конечных автоматов и регулярных множеств) будет нами использован.

Упражнения

Упражнение 1.4.1 (**). *Показать, что для всякого n существует язык, который распознается НКА с n состояниями, но для распознавания которого ДКА требуется 2^n состояний.*

Упражнение 1.4.2. *Пусть дан недетерминированный конечный автомат $J = \langle A, Q, q_0, F, P \rangle$. Постройте НКА $J' = \langle A, Q', q', F', P' \rangle$ такой, что $L(J) = L(J')$, причем в J' для всякого состояния $q \in Q'$ и всякого $a \in A$ существует $q'' \in Q'$, что $\langle q, a, q'' \rangle \in P'$.*

Упражнение 1.4.3. *Постройте недетерминированные конечные автоматы, распознающие регулярные множества, заданные следующими регулярными выражениями:*

1).
$(0, 1)^* 1 + 011^*$,
2).
$((0 + 1)^* 1 + 01)^* 0$,
3).
$((01 + 10)^* + 11 + 00)^*$.

Упражнение 1.4.4. *Постройте детерминированные конечные автоматы, распознающие регулярные множества, заданные регулярными выражениями из предыдущего упражнения.*

1.5 Разрешимые проблемы и свойства замкнутости

В предыдущем параграфе мы столкнулись с проблемой определения принадлежности слова регулярному множеству, заданному регулярным выражением. Для решения этой проблемы мы предложили способ, состоящий в построении по данному регулярному выражению (детерминированного) конечного автомата, распознающего соответствующее регулярное множество. Процесс же проверки того, воспринимает ли данный ДКА данное слово, описан нами ясно и однозначно.

Ясное и однозначное описание некоторого процесса называется алгоритмом. Таким образом, мы предложили алгоритм решения задачи о принадлежности данного слова языку, заданному данным регулярным выражением.

Алгоритмы изучаются в математике давно. Достаточно привести в пример алгоритм Евклида нахождения наибольшего общего делителя двух натуральных чисел.

Проблема, для решения которой существует алгоритм, называется разрешимой проблемой.

Мы ограничимся пока данным выше неформальным понятием алгоритма. Неформального понятия достаточно для доказательства разрешимости тех или иных проблем, поскольку, когда мы предъявляем некоторый конкретный алгоритм решения конкретной проблемы, осуществимость его обычно не вызывает сомнений. Позже мы столкнемся еще с понятием неразрешимой проблемы, то есть проблемы, для решения которой не существует алгоритма. Определение неразрешимой проблемы связано уже с формальным определением алгоритма. Такое определение будет дано в главе 4. Неразрешимые проблемы изучаются в главе 5.

В данном параграфе мы установим разрешимость еще нескольких проблем для регулярных множеств, заданных регулярными выражениями. Для этого нам понадобятся результаты о замкнутости класса регулярных множеств. Они интересны и сами по себе.

Определение 1.5.1 (замкнутости семейства языков). *Пусть S — семейство языков (например, регулярные языки), $f^{(n)}$ — n-местная* *операция над языками, то есть отображение, которое языкам*

$$L_1, L_2, \ldots, L_n$$

в некотором алфавите A ставит в соответствие язык

$$L = f(L_1, L_2, \ldots, L_n)$$

в алфавите A. Будем говорить, что семейство S замкнуто относительно операции f, если из того, что языки $L_1, L_2, \ldots, L_n$ принадлежат семейству S следует, что и язык $f(L_1, L_2, \ldots, L_n)$ принадлежит семейству S.

Из определения регулярных множеств следует, что класс регулярных множеств замкнут относительно операций объединения, конкатенации и итерации (см. п.1.2).

Теорема 1.4. *Дополнение всякого регулярного множества R в алфавите A (то есть язык $A^* \setminus R$) тоже является регулярным множеством.*

Доказательство. Пусть r — задающее R регулярное выражение. По теореме 1.3 существует детерминированный конечный автомат $J = \langle A, Q, q_0, F, P \rangle$ такой, что $L(J) = R$.

Рассмотрим ДКА $J' = \langle A, Q, q_0, Q \setminus F, P \rangle$. Очевидно, что $L(J') = A^* \setminus R$.

Как мы говорили, позже мы формально докажем, что всякое множество, распознаваемое конечным автоматом, регулярно. Использованием этого факта завершается доказательство теоремы. ∎

Теорема 1.5. *Пусть R, R' — регулярные множества в алфавите A. Тогда их пересечение $R \bigcap R'$ тоже является регулярным множеством в алфавите A.*

Доказательство. Имеет место очевидное тождество

$$(R \bigcap R') = (A^* \setminus ((A^* \setminus R) \bigcup (A^* \setminus R'))).$$

В силу теоремы 1.4 настоящего параграфа, в левой части тождества стоит регулярное множество. ∎

Теорема 1.6. *Пусть R, R' — регулярные множества в алфавите A. Тогда их разность $R \setminus R'$ тоже является регулярным множеством в алфавите A.*

Доказательство. Имеет место очевидное тождество

$$(R \setminus R') = (R \bigcap (A^* \setminus R')).$$

В силу теорем 1.4 и 1.5 настоящего параграфа, в левой части тождества стоит регулярное множество. ∎

Рассмотрим теперь некоторые разрешимые проблемы для регулярных множеств, заданных регулярными выражениями.

Проблема пустоты. Дано регулярное выражение r. Пуст ли регулярный язык $L(r)$?

Проблема эквивалентности. Даны два регулярных выражения r и r'. Совпадают ли регулярные языки $L(r)$ и $L(r')$?

Обе эти проблемы разрешимы.

Теорема 1.7. *Существует алгоритм, который по данному регулярному выражению r определяет, пуст ли язык $L(r)$.*

Доказательство. Возможно несколько вариантов доказательства. Один из них — индукцией по сложности r. Здесь можно исходить из того, что

(1) $L(r^*)$ всегда непусто, ибо по крайней мере содержит пустое слово;

(2) $L(rr')$ пусто тогда и только тогда, когда либо $L(r)$ пусто, либо $L(r')$ пусто;

(3) $L(r + r')$ пусто тогда и только тогда, когда одновременно $L(r)$ пусто и $L(r')$ пусто.

Мы рассмотрим доказательство, основанное на конечных автоматах.

Пусть $J = \langle A, Q, q_0, F, P\rangle$ — задающий $L(r)$ конечный автомат (все равно, детерминированный или нет). Он может быть явно построен по r.

Построим содержащиеся в Q множества $G_i, i = 0, 1, \ldots, |Q|$, где $|Q|$ обозначает мощность множества Q. Пусть $n = |Q|$.

Положим $G_0 = \{q_0\}$.

Если G_i уже определено, положим

$$G_{i+1} = (\{q \in Q \mid \ a \in A,\ q' \in G_i,\ \langle q', a, q\rangle \in P\} \bigcup G_i).$$

Фактически G_n — это множество состояний, в которых автомат J может находиться при работе с какими бы то ни было словами в алфавите A, то есть всякое состояние из G_n достижимо при работе с некоторым словом во входном алфавите, и наоборот, если некоторое состояние достижимо при работе с некоторым словом, то это состояние входит в G_n. Таким образом, проблема пустоты языка, задаваемого автоматом J, сводится к проблеме достижимости хотя бы одного заключительного состояния. ∎

Фактически мы доказали и теорему о разрешимости проблемы пустоты языка, заданного конечным автоматом.

Теорема 1.8. *Существует алгоритм, который по данным регулярным выражениям r, r' определяет, задают ли эти выражения один и тот же язык.*

Доказательство. Пусть $R = L(r)$, $R' = L(r')$. Очевидно, что $R = R'$ тогда и только тогда, когда одновременно $R \setminus R'$ и $R' \setminus R$ пусты. Как $R \setminus R'$, так и $R' \setminus R$ — регулярные множества (теорема 1.6).

Рассмотрим ДКА J, J', задающие $L(r)$ и $L(r')$. Такие автоматы могут быть явно построены (см. теорему 1.3).

Используя формулу из теоремы 1.6 и методы конструирования конечных автоматов из теоремы 1.4 настоящего параграфа и теоремы 1.3, можно явно построить НКА J_1 и J' такие, что $L(J_1) = R \setminus R'$, $L(J') = R' \setminus R$.

Проблемы пустоты $L(J_1)$ и $L(J')$ разрешимы, как замечено в доказательстве теоремы 1.7. ∎

Упражнения

Упражнение 1.5.1. *Пусть J, J' — два ДКА в некотором алфавите.*

1). Запишите явно ДКА J'' такой, что $L(J'') = L(J) \bigcup L(J')$.

2). Запишите явно ДКА J'' такой, что $L(J'') = L(J) \bigcap L(J')$.

Упражнение 1.5.2. *В обозначениях предыдущего упражнения запишите явно ДКА J'' такой, что $L(J'') = L(J)L(J')$.*

Упражнение 1.5.3 (**). *Обращением языка R называется язык R', состоящий из всех слов языка R, записанных в обратном порядке (справа налево). Покажите, что класс регулярных множеств замкнут относительно одноместной операции обращения.*

Упражнение 1.5.4. *Проведите доказательство теоремы 1.7 индукцией по сложности регулярного выражения.*

Упражнение 1.5.5 (**). *Пусть r, r' — регулярные выражения в алфавите A длины n и n' соответственно (т.е. число символов в r, например, равно n). Какова минимальная длина регулярных выражений r'', r''', задающих множества $L(r) \bigcap L(r')$ и $A^* \setminus L(r)$?*

Упражнение 1.5.6. *Пусть r, r' — регулярные выражения в алфавите A. Покажите, что проблема включения ($L(r)$ содержится в $L(r')$) разрешима.*

Упражнение 1.5.7. *Проверьте, пусты ли языки, задаваемые следующими регулярными выражениями:*

1).
$(01^* + \lambda)^*(010^*\lambda + \lambda)$,
2).
$(01^*\lambda + \lambda^*)(010^*\lambda + \lambda)^*$,
3).
$(01^*\lambda + \lambda)(010^*\lambda + \lambda)$.

Упражнение 1.5.8. *Проверьте, эквивалентны ли пары регулярных выражений:*

$((0 + 1)^*1 + 101)^*$ *и* $0^*1 + 1^*1 + ((0 + 1)^* + 1)^*1$;
$(01^* + \lambda)^*(010^*\lambda + \lambda)^*$ *и* $(01^*\lambda + \lambda)(0 + 1)^*$.

1.6 Лемма о разрастании и доказательства нерегулярности

Как выясняется, класс регулярных множеств довольно беден. Однако внимание к этому классу объясняется, в частности, тем, что регулярные языки можно очень быстро и просто распознавать при помощи конечных автоматов.

Когда мы имеем дело с каким-то языком (заданным некоторым способом), нам важно знать, регулярен ли он. Для доказательства регулярности достаточно записать задающее язык регулярное выражение. Однако у нас пока нет способа доказывать нерегулярность языка.

Следующая лемма дает нам мощный инструмент для доказательства нерегулярности языков. Фактически эта лемма описывает строение регулярного языка.

Лемма 1.9 (о разрастании). *Пусть r — регулярное выражение в алфавите A. Существует такое натуральное число $k > 0$, что всякое такое слово $w \in L(r)$, что $|w| > k$, можно представить в виде $w = xyz$, где x, y и z — слова в алфавите A, так что выполняются следующие условия:*

1. $0 < |y| < k$;
2. $xy^iz \in L(r)$ *для всякого* $i \geq 0$;

3. $0 \leq |x| < k$.

Доказательство. Пусть $J = \langle A, Q, q_0, F, P\rangle$ — задающий $L(r)$ детерминированный конечный автомат.

Положим $k = |Q| + 1$, где $|Q|$ обозначает число элементов Q,

$$Q = \{q_0, q_1, \ldots, q_{|Q|-1}\}$$

.

Пусть $w \in L(r) = L(J)$, $|w| > k$. Рассмотрим работу автомата J на слове w. Пусть при этой работе автомат проходит состояния

$$q_0 = q_{i_1}, q_{i_2}, \ldots, q_{i_k}, q_{i_{k+1}}, \ldots, q_{i_t},$$

где $t = |w| + 1$, $q_{i_t} \in F$. Значит, среди $i_1, i_2, \ldots, i_k$ найдутся два одинаковых индекса, поскольку $0 \leq i_j \leq |Q| - 1$, а $k = |Q| + 1$. Пусть, например, $i_j = i_m$, $1 \leq j < m \leq k$. Представим слово w в виде xyz, где x имеет длину $j - 1$, y — длину $m - j$, z — длину $t - m$.

Покажем, что такое представление удовлетворяет условиям леммы. Действительно,

$$0 < |y| < m \leq k = |Q| + 1,$$

т.е. условие (1) леммы выполнено. Затем,

$$0 \leq |x| < j < m \leq k = |Q| + 1,$$

т.е. условие (3) леммы также выполнено. Далее, видно, что автомат J распознает и слово xz, и все слова вида xy^iz. Для слова xz автомат J в силу детерминизма, пройдет последовательность состояний

$$q_0 = q_{i_1}, \ldots, q_{i_j}, q_{i_{m+1}}, \ldots, q_{i_t}.$$

Для слова xy^iz автомат J в силу детерминизма, пройдет

$$q_0 = q_{i_1}, \ldots, q_{i_j},$$

затем i раз последовательность состояний

$$q_{i_{j+1}}, q_{i_{j+2}}, \ldots, q_{i_m} = q_{i_j},$$

и затем последовательность

$$q_{i_{m+1}}, \ldots, q_{i_t}.$$

$\blacksquare$

Применим лемму для доказательства нерегулярности языков. Начнем с решения упражнения 1.3.2.

Следствие 1.10. *Язык L, полученный объединением всевозможных одноэлементных языков вида 0^i1^i, не является регулярным.*

Доказательство. Пусть, напротив, язык L регулярен. По лемме, существует тогда константа $k > 0$, что всякое такое слово $w \in L$, что $|w| > k$, можно представить в виде $w = xyz$, где $0 < |y| < k$, что, в частности, слово $xyyz$ тоже принадлежит L.

1. y составлено только из нулей. Тогда слово $xyyz$ содержит больше нулей, чем единиц, и не принадлежит L по определению.

2. y составлено только из единиц. Аналогично предыдущему.

3. y составлено и из нулей, и из единиц. Тогда, так как xyz содержит в начале только нули, а потом только единицы, то y начинается с 0 и кончается на 1. Значит, слово $xyyz$ содержит подслово 10. Но тогда слово $xyyz$ не принадлежит L, ибо никакое слово с подсловом 10 не принадлежит L.

Во всех трех случаях получили противоречие. ∎

Упражнения

Упражнение 1.6.1. *Приведите пример языка L, который не регулярен, но для которого выполняются условия леммы о разрастании.*

Упражнение 1.6.2. *Регулярны ли следующие языки:*

1).
$$\{0^i1^j0^k \mid i, j, k \geq 0\};$$

2).
$$\{0^i1^j0^i \mid i, j \geq 0\}.$$

Предметный указатель