Задание 3

Вычислительные возможности конечных автоматов

Ключевые слова [1]:*принцип мат. индукции, язык, регулярные выражения, конкатенация, объединение, итерация, конечные автоматы (КА), детерминированные и недетерминированные КА, регулярные языки.*

0 Ликбез

Задачи помеченные † не являются сложными, однако являются в какой-то мере дополнительными. Я рекомендую их решать, но жёстко этого не требую.

0.1 Отношение эквивалентности

(Бинарным) отношением R на множестве X называется некоторое подмножество R множества $X \times X$. Говорят, что пара элементов x и y удовлетворяют отношению R, если пара (x, y) принадлежит R. Это принято обозначать xRy.

Отношение R называется *рефлексивным*, если для любого $x \in X$ справедливо xRx. Отношение называется симметричным, если из факта xRy следует yRx. Отношение называется транзитивным, если из xRy и yRz следует xRz.

Определение 1. Бинарное отношение называется *отношением эквивалентности*, если оно является рефлексивным, симметричным и транзитивным. Такие отношения обычно обозначаются $\sim_R$ или просто $\sim$, когда ясно о каком отношении идёт речь.

Классом эквивалентности $C(x)$ называется множество элементов эквивалентных x. То есть $C(x) = \{y \,|\, x \sim y\}$.

Упражнение 1. Показать, что классы эквивалентности $C(x)$ и $C(y)$ либо не пересекаются, либо совпадают.

[1] минимальный необходимый объём понятий и навыков по этому разделу)

Множество X, над которым задано отношение эквивалентности $\sim$, можно представить в виде объединения попарно непересекающихся множеств – классов эквивалентности, то есть *факторизовать* по отношению эквивалентности. Фактормножество обозначается как $X\backslash\sim$. То есть, $X\backslash\sim = \{C(x) \mid x \in X\}$. Мощность фактормножества называется *индексом* отношения эквивалентности.

0.2 Морфизмы

Определение 2. Морфизмом называется отображение $\varphi : \Sigma^* \to \Delta^*$, для которого справедливо: $w = uv$, тогда $\varphi(w) = \varphi(u) \cdot \varphi(v)$.

Гомоморфизм, который вы изучали в рамках высшей алгебры, является частным случаем морфизма и в современной терминологии его также называют морфизмом.

Упражнение 2. Показать, что морфизм φ однозначно определён, если для каждой буквы σ алфавита Σ определено значение $\varphi(\sigma)$.

Задача $1^\dagger$. Доказать, что регулярные языки замкнуты относительно взятия морфизма.

Определение 3. Обратным морфизмом φ^{-1} к морфизму $\varphi : \Sigma^* \to \Delta^*$, называется отображение $\varphi^{-1}(w) = \{v \mid \varphi(v) = w\}$

Морфизмы применяются не только к словам, но и к языкам. Запись $\varphi(L)$ означает, что $\varphi(L) = \{\varphi(w) \mid w \in L\}$, то же самое относится и к обратному морфизму: $\varphi^{-1}(L) = \{w \mid \varphi(w) \in L\}$.

Задача $2^\dagger$. Верно ли, что для любого языка L и любого морфизма $\varphi : \Sigma^* \to \Sigma^*$

1. язык $\varphi(\varphi^{-1}(L))$ совпадает с L?
2. язык $\varphi^{-1}(\varphi(L))$ совпадает с L?
3. $\varphi(\varphi^{-1}(L)) \stackrel{?}{=} \varphi^{-1}(\varphi(L))$

Задача $3^\dagger$. Доказать, что регулярные языки замкнуты относительно операции взятия обратного морфизма.

1 Теорема Майхилла-Нероуда

Поскольку мы работаем со словами, то нас будут интересовать бинарные отношения на множестве Σ^*. А именно, нас будет интересовать следующее отношение эквивалентности, задаваемые языком L. Слово x L-эквивалентно слову y, если для любого суффикса z, слова xz и yz либо одновременно лежат в L, либо одновременно не принадлежат L. Формально, $x \sim_L y \iff \forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L$. Это отношение эквивалентности называется отношением Майхилла-Нероуда.

Легко видеть, что это отношение является правоинвариантным, то есть если $x \sim_L y$, то $xz \sim_L yz$ для любого z.

Теорема 1 (Майхилл-Нероуд, 1958)**.** *Язык L является регулярным тогда и только тогда, когда Σ^* разбивается на конечное число классов эквивалентности по отношению $\sim_L$. Другими словами, когда $\sim_L$ – отношение конечного индекса.*

Доказательство. Если язык L регулярен, то отношение $\sim_L$ очевидно имеет конечный индекс. Действительно, возьмём произвольный полный[2] ДКА $\mathcal{A}$, распознающий L, в котором все состояния достижимы[3]. Пусть $\mathcal{A}$ имеет n состояний. Рассмотрим слова $x_1, x2, \ldots, x_n$, такие что $\delta(q_0, x_i) = q_i$. По любому слову w, автомат попадает в некоторое состояние q_i, а значит $w \in C(x_i)$, потому что для любого слова z, состояние $q = \delta(q_i, z)$ либо принимающие, либо нет, а $\delta(q_0, x_i z) = \delta(q_0, wz) = \delta(q_i, z) = q$, поэтому $x \sim_L z$. Таким образом, мощность фактормножества $\Sigma^*\backslash\sim_L$ не превосходит n, а значит самих классов эквивалентности конечное число.

В обратную сторону. Пусть таких классов конечное число. Тогда, $\Sigma^*\backslash\sim_L = \{C_1, C_2, \ldots, C_n\}$. Построим, имея такое разбиение, ДКА $\mathcal{A}$, распознающий L.

Построение: Множеством состояний является фактормножество, то есть $Q = \Sigma^*\backslash\sim_L$, в качестве начального состояния q_0 возьмём $C(\varepsilon)$. Функцию переходов δ определим следующим образом. Пусть x_i – представитель класса C_i, тогда $\delta(C_i, \sigma) = C_j$, если $x_i\sigma \in C_j$. Осталось описать множество принимающих состояний: $F = \{C_i \mid x_i \in L\}$.

[2] ДКА является полным, если в нём определены все переходы, т.е. $\forall q \in Q, \forall \sigma \in \Sigma : \delta(q, \sigma) \neq \emptyset$.

[3] формально в Q могут быть состояния, в которые невозможно попасть из q_0. Обратите внимание, что они могут возникать при применении конструкции произведения.

Корректность: По построению, автомат $\mathcal{A}$ при обработке слова w на i-ом шаге оказывается в состоянии[4] $C(w[1,i])$. Таким образом, обработав слово, автомат перейдёт в состояние $C(w)$, которое будет принимающим тогда и только тогда, когда $w \in L$, поскольку если $x_i \sim_L w$, и $x_i \in L$, а $w \notin L$, то отношение $\sim_L$ не является правоинвариантным: $\varepsilon \sim_L \varepsilon$, но $\varepsilon \cdot x_i \not\sim_L \varepsilon \cdot w$, приходим к противоречию. □

Эта теорема очень часто вызывает непонимание: почему мы можем построить автомат, если существует конечное разбиение. Да, допустим, что разбиение есть, но кто же нам его дал? При доказательстве теорем, мы можем использовать факты из логики вида «утверждение всегда либо истинно, либо ложно» и используем оракул, который отвечает на наши вопросы – если Оракул ответил «истинно», то мы начинаем одну ветвь рассуждений, если «ложно», то другую. Если во всех ветках ответа оракула, мы доказали правильность нашего утверждения, то утверждение считается доказанным. Так, мы пользовались тем, что оракул сообщал нам конечное ли у нас число классов эквивалентности или нет, лежат ли два слова в одном классе эквивалентности или нет и мы успешно *построили* автомат в доказательстве – это означает, что мы доказали, что если классов эквивалентности конечное число, то такой автомат есть. На практике же, зная только то, что классов эквивалентности конечное число, автомат мы можем и не построить – для того, чтобы построить автомат, оракул должен быть вычислимой функцией, то есть мы должны уметь строить такую машину Тьюринга[5], которая отвечала бы на наши вопросы. Доказательства, в которых оракул вычислим, называются *конструктивными*.

2 Лемма о накачке[6]

В данном разделе мы поговорим о лемме о накачке – одном из способов доказательства нерегулярности языка.

[4] Напомним, что $w[i,j]$ есть подслово слова w, начинающееся с i-го символа w и заканчивающееся j-ым.

[5] Или, например, мы можем написать программу на языке C.

[6] Также известна как лемма о разрастании – неудачный перевод неудачного термина «Pumping Lemma».

Лемма 1. *Для любого регулярного языка L существует такая константа $p \geqslant 1$, что для любого слова w из L длиннее p, справедливо:*

- $w = xyz$
- $|y| \geqslant 1$
- $|xy| \leqslant p$
- $\forall i \geqslant 0, xy^i z \in L$.

Доказательство. Поскольку $L \in \mathsf{REG}$, то существует ДКА $\mathcal{A}$ распознающий L. Пусть $\mathcal{A}$ имеет N состояний. Возьмём $p = N + 1$. Тогда, если если слово w принадлежит L и $|w| \geqslant p$, то это означает, что при обработке w автомат $\mathcal{A}$ оказался в некотором сотоянии q дважды. Пусть в первый раз автомат оказался в q после прочтения префикса x, а второй раз, при прочтении префикса xy. Тогда $\delta(q, y) = q$, но поскольку $w = xyz$ принадлежит L, то это означает, что $\delta(q, z) = q_f \in F$, а значит все слова вида $xy^i z$, $i \geqslant 0$ лежат в L. □

Обратите внимание, что при доказательстве леммы, я использовал те же трюки, что и в доказательстве на семинаре того, что $a^n b^n$ – нерегуляный язык.

Пример 1. Используем лемму о накачке для доказательства христоматийного примера нерегулярности языка $L = \{0^n 1^n \mid n \geqslant 0\}$.

Доказательство. Допустим, что язык L регулярный. Тогда, по лемме о накачке, существует константа p, что для любого слова w длиннее p, существует такое разбиение xyz, что $|xy| \leqslant p$ и слова $xy^i z$, $i \geqslant 0$ принадлежат L.

Рассмотрим $w = 0^p 1^p$. Если такое разбиение существует, то y имеет вид 0^k или 1^k, $k \geqslant 1$ – в противном случае, если $y = 0^k 1^l$, то $y^2 = 0^k 1^l 0^k 1^l$, но в L нет слов, в которых за 1 следует 0. Допустим, что $y = 0^k$. Тогда $x = 0^m$, $z = 0^l 1^p$, $k + m + l = p$. Но тогда, по лемме о накачке $xy^2 z \in L$, а значит, слово $0^{m+2k+l} 1^l \in L$, но $m + 2k + l > p$, т.к. $m + k + l = p$ и $k > 0$, поскольку $|y| \geqslant 1$. Аналогично приходим к противоречию когда $y = 1^k$. □

У этой леммы слишком много минусов. Во-первых, она работает не всегда: если язык нерегулярен, это ещё не означает, что лемма о накачке для него не выполняется. Во-вторых, она слишком громоздкая. Даже для такого простого примера как $L = \{0^n 1^n\}$, потребовалось много писанины, а в более сложных случаев перебор возможных y куда шире. Как показывает наблюдение, руками нерегулярность доказать быстрее, да и работает техника, обсужденная на семинаре в тех случаях, когда применима лемма о накачке. Но тем не менее, у этой леммы есть и плюсы – учебные. Во-первых, лемма о накачке показывает структуру регулярного языка: разность длин двух последовательных слов из регулярного языка ограниченна линейной функцией. Во-вторых, сущетсвует ещё лемма о накачке для КС-языков, для понимания которой стоит изучить более простую лемму о накачке для регулярных языков. В случае КС-языков, доказательство непринадлежности языка классу КС уже куда менее очевидно, так что лемма о накачке становится мощным и одним из основных инструментов.

3 Задачи

Задача 4. Доказать, что регулярные языки замкнуты относительно операций объединения, пересечения и дополнения.

Задача 5. Применить лемму для доказательства нерегулярности языка $L = \{a^{2^n} \mid n \geqslant 0\}$.

Задача 6 (№6 из задания). Будут ли регулярными следующие языки:

1. $L_1 = \{a^{2013n+5 \,|\, n=0,1,\ldots}\} \cap \{a^{509k+29} \,|\, k = 401, 402, \ldots\} \subseteq \{a^*\}$
2. $L_2 = \{a^{200n^2+1} \,|\, n = 1000, 1001, \ldots\} \subseteq \{a^*\}$
3. Язык L_3 всех слов в алфавите $0, 1$, которые представляют числа в двоичной записи, дающие остаток два при делении на три (слово читается со старших разрядов). Например, $001010(10102 = 1010 = 3 \times 3 + 1) \notin L_3$, а $10001(100012 = 1710 = 5 \times 3 + 2) \in L_3$.
4. Построить ДКА, распознающий язык L_3.

Задача 7. Показать, что лемма о накачке выполняется для языка $L = \{uvwxy \mid u, y \in \{0, 1, 2, 3\}^*; v, w, x \in \{0, 1, 2, 3\}$, причём $v = w$ или $v = x$

или $x = w$ $\} \cup \{w \mid w \in \{0,1,2,3\}^*$, причём $\frac{1}{7}$ букв в w есть 3 $\}$. Более формально:
$L = \{uvwxy \mid u, y \in \{0,1,2,3\}^*; v, w, x \in \{0,1,2,3\} \wedge (v = w \vee v = x \vee x = w)\} \cup \{w \mid w \in \{0,1,2,3\}^*, \left\lceil \frac{|w|}{7} \right\rceil = |w|_3\}$

1. Доказать, что $L \notin \mathsf{REG}$.