Задание 4

Замкнутость регулярных языков, теорема Майхилла-Нероуда и минимальные автоматы

Ключевые слова[1]:*язык, регулярный язык, ДКА, полный ДКА, НКА, отношение эквивалентности, декартово произведение.*

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 объем 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: если $w = uv$, то $\varphi(w) = \varphi(u) \cdot \varphi(v)$.

Операция взятия морфизма переносится на множества естественным образом: $\varphi(X) = \{\varphi(x) \mid x \in X\}$.

Морфизм является частным случаем понятия гомоморфизма, которое вы изучали в рамках курса высшей алгебры. В современной терминологии гомоморфизмы также называют просто морфизмами.

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. Обратите внимание, что отношение эквивалентности задано на множестве всех слов Σ^*, а не только на словах языка L: таким образом, x и y в определении – произвольные слова. 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 $x_1, x_2, \ldots, x_n$, такие что $\delta(q_0, x_i) = q_i$. По любому слову w автомат переходит из начального состояния 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 $w \sim_L x_i$. 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: Можеством состояний является фактормножество, то есть

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

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

$Q = \Sigma^* \backslash \sim_L$, 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: $F = \{C_i \,|\, x_i \in L\}$.
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 слово x_i лежит в языке L, то и слово w тогда лежит в языке L, иначе различающим для них будет пустое слово.

□

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*.

---

[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.

2 Построение минимального автомата

ДКА $\mathcal{A}$ называется *минимальным* автоматом, распознающим язык L, если $L(\mathcal{A}) = L$ и не существует ДКА $\mathcal{B}$, такого что $L(\mathcal{B}) = L$ и число состояний автомата $\mathcal{B}$ меньше числа состояний автомата $\mathcal{A}$.

Для доказательства существования и корректности построения минимального автомата мы будем использовать теорему Майхилла-Нероуда. Нам потребуется аналогичное отношению М.Н. отношение эквивалентности, определённое на состояниях. Будем использовать обозначение $q \xrightarrow{w} p$, если из состояния q обрабатывая слово w, автомат переходит в состояние p или что то же самое $(q, w) \vdash^* (p, \varepsilon)$.

Определение 4. Зафиксируем автомат $\mathcal{A}$, распознающий язык L. Пусть $q_0 \xrightarrow{x_i} q_i$, а $q_0 \xrightarrow{x_j} q_j$. Тогда $q_i \sim_L q_j$ тогда и только тогда, когда $x_i \sim_L x_j$. Данное отношение мы назовём соответствующим отношению Майхилла-Нероуда.

Неэквивалентные состояния называются *различимыми*. Обратите внимание, что это *два разных* отношения эквивалентности, потому что одно из них определено на множестве всех слов, а другое на множестве состояний. Мы обозначаем одинаково два разных отношения, потому что они схожи, но определены на разных объектах, поэтому это не приведёт к путанице.

Теорема 2. *Для каждого регулярного языка, существует минимальный полный автомат, распознающий его. Состояния минимального автомата соответствуют классам эквивалентности по отношению Майхилла-Нероуда* $\sim_L$

Доказательство. В силу теоремы Майхилла-Нероуда, язык L регулярен тогда и только тогда, когда отношение $\sim_L$ имеет конечный индекс. Рассмотрим автомат $\mathcal{A}$, построенный в доказательстве теоремы Майхилла-Нероуда и покажем, что он является минимальным. Допустим противное – пусть автомат $\mathcal{B}$ имеет меньшее число состояний, чем $\mathcal{A}$. Тогда, по принципу Дирихле, существует два слова x_i и x_j, такие что $x_i \not\sim_L x_j$ и в автомате $\mathcal{B}$ при этом $q_0 \xrightarrow{x_i} q$ и $q_0 \xrightarrow{x_j} q$. Так как $x_i \not\sim_L x_j$, то найдётся такое слово z, что $x_i z \in L$, а $x_j z \notin L$. Но тогда с одной стороны $q \xrightarrow{z} q_f \in F$, т.к. $x_i z \in L$, а с другой стороны $q \xrightarrow{z} q_{\bar{f}} \notin F$, поскольку $x_j z \notin L$ – приходим к противоречию. $\square$

Просто из факта того, что язык L имеет конечное число классов эквивалентности М.-Н., не совсем ясно как построить автомат $\mathcal{A}$, распознающий L. Однако, имея любой ДКА, распознающий L можно конструктивно построить по нему минимальный автомат. Рассмотрим регулярный язык L и распознающий его полный ДКА $\mathcal{A}$. Идея построения по автомату $\mathcal{A}$ минимального состоит в склейке эквивалентных состояний. Под склейкой состояний $q_i \sim_L q_j$ мы понимаем удаление состояния q_j из автомата $\mathcal{A}$ и направление всех переходов ведущих в него в состояние q_i.

Утверждение 1. *Склеив все эквивалентные состояния автомата $\mathcal{A}$, мы получим минимальный автомат.*

Доказательство. Склейка состояний $q_i \sim_L q_j$ не изменит язык $L(\mathcal{A})$, потому что из состояния q_j по слову z, автомат $\mathcal{A}$ попадает в принимающее состояние тогда и только тогда, когда он попадает в принимающее состояние по слову z из состояния q_i. Таким образом, склеив все эквивалентные состояния автомата $\mathcal{A}$, мы получим минимальный автомат, потому что в силу определения эквивалентности на состояниях каждое состояние соответствует классу эквивалентности М.-Н. и по теореме 2, он является минимальным. □

Осталось привести алгоритм склейки состояний.

Алгоритм. На вход алгоритма подаётся ДКА $\mathcal{A}$.
1. В случае, если автомат $\mathcal{A}$ не является полным, пополним автомат $\mathcal{A}$, добавив состояние q_D, такое что $q_D \xrightarrow{\Sigma} q_D$, и если $\delta(q, \sigma) = \varnothing$, то теперь $\delta(q, \sigma) = q_D$. Если в $\mathcal{A}$ есть состояния, не достижимые из начального состояния, удалим их.

2. Разделим множество состояний на два подмножества: множество принимающих состояний $F = Q_1$ и его дополнение $Q \setminus F = Q_2$.

$\mathbf{i+1}$. Пусть после i-ого шага алгоритма множество состояний Q разбито на j подмножеств $Q_1, \ldots, Q_j$. Зафиксируем символ $\sigma \in \Sigma$ сделаем следующее. Если для $q_k \in Q_k$ $q_k \xrightarrow{\sigma} q_l \in Q_l$, поместим состояние q_k в множество $Q_{k,l}$. Получили новое разбиение Q на подмножества $Q_{k,l}$ и повторяем для него эту процедуру для оставшихся символов $\sigma \in \Sigma$. Если после $|\Sigma|$ разбиений мы получили разбиение, в котором j подмножеств, то алгоритм останавливается, в противном случае он продолжает работу.

Склеив все состояния, попавшие в одно подмножество, получим минимальный автомат.

Упражнение 3. Доказать корректность данного алгоритма.

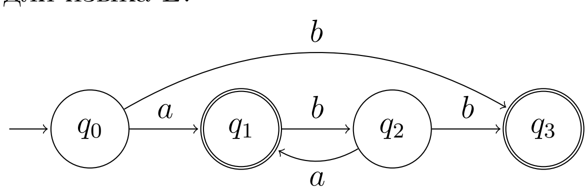
3 Задачи

Задача 4. Доказать, что для языка L выполняется лемма о накачке, но он не является регулярным. Обозначим за PRIMES множество простых чисел. Напомним, что $A^+ = AA^*$.

$$L = b^* \cup \{ab^p \mid p \in \text{PRIMES}\} \cup aa^+b^*.$$

Задача 5. К языку L_1 добавили конечный язык R и получили язык L ($L = L_1 \cup R$). Язык L оказался регулярным. Верно ли, что язык L_1 мог быть нерегулярным?

Задача 6. Язык L задан автоматом $\mathcal{A}$. Построить минимальный автомат для языка L.



Задача 7. Постройте минимальный автомат для языка $\bar{L}$ из предыдущей задачи.

Задача 8. Найдите все классы эквивалентности Майхилла-Нероуда для языка $\Sigma^*ab\Sigma^*$ и постройте по ним ДКА.