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.*

1 Построение минимального автомата

ДКА $\mathcal{A}$ называется *минимальным* автоматом, распознающим 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.

Определение 1. Зафиксируем автомат $\mathcal{A}$, распознающий язык L. Пусть $\delta(q_0, x_i) = q_i$, а $\delta(q_0, x_j) = q_j$. 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.

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 для разных объектах, поэтому это не приведёт к путанице.

Теорема 1. *Для каждого регулярного языка, существует минимальный автомат, распознающий его. Состояния минимального автомата соответствуют классам эквивалентности по отношению Майхилла-Нероуда* $\sim_L$

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

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 $\delta_{\mathcal{B}}(q_0, x_i) = \delta_{\mathcal{B}}(q_0, x_j) = q$. 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 $\delta(q, z) \in F$, т.к. $x_i z \in L$, а с другой стороны $\delta(q, z) \notin F$, поскольку $x_j z \notin L$ – приходим к противоречию. □

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 **??**, 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 $(q, \sigma) = \emptyset$, то теперь $(q, \sigma) = q_D$. 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 1. Доказать корректность данного алгоритма.

2 Задачи

Задача 1. Вам сообщили, что язык L имеет конечное число классов эквивалентности Майхилла-Нероуда и вам предоставили доступ к оракулу f, который сообщает лежат ли два слова в одном классе эквивалентности или нет и оракулу g, который сообщает принадлежит ли слово языку L или нет. Можно считать, что вы пишете программу на языке C, которая должна на выходе вывести описание автомата и в ней вы используете функцию $f(x, y)$, которая возвращает 1, если x и y лежат в одном классе эквивалентности и 0 в противном случае. Приведите алгоритм построения ДКА $\mathcal{A}$, который распознаёт язык L.

Задача 2*. Даны два ДКА $\mathcal{A}$ и $\mathcal{B}$. Верно ли, что $L(\mathcal{A}) = L(\mathcal{B})$ тогда и только тогда, когда для всех слов $w : |w| \leqslant |Q_{\mathcal{A}}||Q_{\mathcal{B}}|$, w лежит как в $L(\mathcal{A})$, так и в $L(\mathcal{B})$.

Будем называть алгоритм *эффективным*, если он реализуем за полиномиальное время. Например, программа на C, делает полиномиальное число тактов при исполнении алгоритма.

Задача 3. Предложите алгоритм, который проверяет порождают ли два регулярных выражения один и тот же язык или нет. Будет ли он эффективным?

Задача 4.

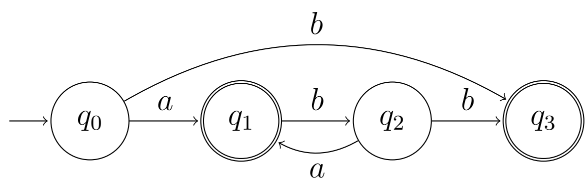
1. Постройте ДКА $\mathcal{A}$ распознающий все слова, в которых чётное число единиц.

2. Постройте ДКА $\mathcal{B}$ распознающий все слова, в которых нечётное число нулей.

3. Постройте автомат $\mathcal{C}$, распознающий все слова, в которых чётное число единиц и нечётное число нулей.

4*. Предложите алгоритм для построения пересечения языков, заданных ДКА.

Задача 5. Язык L задан автоматом $\mathcal{A}$. Построить минимальный автомат для языка $\bar{L}$.



Определим операцию обращения. Язык L^R состоит из всех слов, зеркальных к словам из языка L. То есть

$$L^R = \{w \mid w = w_1 w_2 \dots w_n,\ w_n w_{n-1} \dots w_1 \in L\}.$$

Задача 6†. Замкнуты ли регулярные языки относительно операции обращения? Предложите алгоритм построения ДКА для L^R по ДКА для L.

3 P.S.

На выходе с семинара меня спросили как рисовать в техе стрелочки подобно тем, которые я рисовал на доске. Не знаю не пробовал — можете поискать ответ в книжках или задать вопрос на http://tex.stackexchange.com. Я обычно пишу отдельно по каждому состоянию $q \xrightarrow{\sigma} p$.