Домашняя работа по ТРЯП №4

Автор - Айвазов Денис из 671 группы

27 сентября 2017

1 $^{+}$Доказать, что REG замкнуты относительно взятия морфизма

Если $L \in REG$ а h-гомоморфизм, то $h(L) \in REG$. Где L задан РВ R. Докажем что $L(h(R)) = h(L(R))$ с помощью индукции по длине регулярного выражения.
Пусть V - какое-то подвыражение нашего РВ R. Будем по индукции доказывать L(h(V))=h(L(V));
База: $V = \varepsilon$ или $V = \emptyset$ то h(V)=V и L(h(V))=L(V)=h(L(V)); Т.е. L(V) либо цепочка без слов, либо не содержит ничего. Если $V = a \in \Sigma$, тогда L(V)={a}; h(L(V))=L(h(V)) т.к. h(V) это цепочка символов h(a);
Индукция: для обьединения двух цепочек F и G, для которых выполнено предположение. Т.е. что для V=F+G выполнено предположение.
Т.к. для F и для G выполнено, то $h(V) = h(F+G) = h(F) + h(G)$; И т.к. $L(h(V)) = L(h(F) + h(G)) = L(h(F)) \cup L(h(G))$ и $L(V) = L(F) \cup L(G)$ получим: $h(L(V)) = h(L(F) \cup L(G)) = h(L(F) \cup h(L(G))$
В итоге получим: L(h(V))=h(L(V)); чтд.
Для итерации получается даже проще. для итерации аналогично с объединением. Таким образом получим, что гомоморфизм языка, заданного РВ(т.е. регулярного языка) так же является регулярным языком.

2 $^+$Верно ли, что для любого языка L и любого морфизма ϕ

2.1 язык $\phi(\phi^{-1}(L))$ совпадает с L?

2.2 язык $\phi^{-1}(\phi(L))$ совпадает с L?

2.3 $\phi^{-1}(\phi(L)) = \phi(\phi^{-1}(L))$

3 $^+$Доказать, что REG замкнуты относительно взятия обратного морфизма

4 * Привести ДКА на котором алгоритм минимизации не работает

5 Показать, что для языка L выполняется лемма о накачке, но он не регулярный

1. Сначала покажем, что язык не принадлежит классу REG.
   $\forall i, j \in PRIMES : i < j,\ b^i \nsim b^j$, (т.е. по определению L $ab^i \in L$, $ab^j \in L$ но в разных классах эквивалентности)
   выполнено, что $\exists m > 0 :\ i + m \in PRIMES, j + m \notin PRIMES$.
   Значит т.к. это выполнено для любых i и j с заданным условием(которых бесконечно много), то у нас бесконечное число классов эквивалентности. А по теореме Майхилла-Нероуда это значит, что язык не регулярен.

2. А теперь покажем, что лемма о накачке выполняется для
   $L = b^* \cup \{ab^p | p \in PRIMES\} \cup aa^+b^*$.
   $\exists p = 4 \geq 1 : \forall \omega \in L : |\omega| > p\ \exists$ разбиение $\omega = xyz :\quad |y| \geq 1; |xy| \leq p; \forall i \geq 0\ xy^iz \in L;$
   Тогда для любого $\omega \in L$ выполнено что-то из трех: $\omega \in b^*$, $\omega\ in\ ab^p$ или $\omega \in aa^+b^*$. Рассмотрим все варианты поотдельности:

   (a) $\omega\ in\ b^*$ т.е. $\omega = b^k bb^j; p = 4 > k; k, j \in \mathbb{N}$;
   тогда y=b и $\forall i \geq 0\ b^k b^i b^j \subset b^* \subseteq L$;

   (b) $\omega \in aa^+b^*$; $\omega = aa^k b^j$; $k, j \in \mathbb{N}_0; k > 0; j \geq 0$; и тогда y=a; Получим, что $\forall i \geq 0\ a^i a^k a^j \in aa^+b^* \in L$

   (c) $\omega\ in\ ab^p$ сделаем $y = a; \omega y b^k : p = 4 \geq k$; тогда

   $$\left[\begin{array}{l} i = 0;\ b^k \in b^* \subseteq L; \\ i = 1;\ ab^k \in ab^p \subseteq L; \\ \forall i \geq 2\ a^i b^k \in aa^+b^* \subseteq L; \end{array}\right.$$

   Т.е. мы рассмотрели все варианты и лемма о накачке действительно выполнена для этого нерегулярного языка. Именно поэтому она просто лемма (т.е. необходимое условие), а не теорема в обе стороны, как теорема Майхилла-Нероуда.

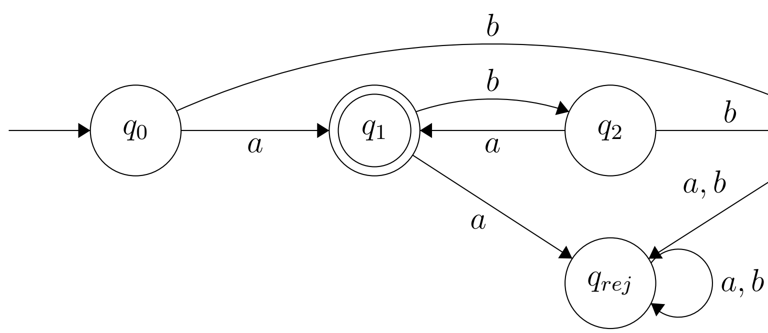
6 К языку добавили конечный и он стал регулярным. Он мог быть нерегулярным?

Ответ: язык L_1 точно был регулярным.
Определим $m = max|v|; v \in R$. Т.е. m - максимальная длина всех слов R
Допустим, что он не регулярен. Тогда для него выполняется:
$\forall p \;\; \exists \omega \in L_1 : \;|\omega| \geq p$
$\forall$ разбиение $\omega = xyz; |xy| \leq p, |y| \geq 0 \; \exists i \geq 0 : xy^i z \notin L_1$
Т.к. L_1 - нерегулярный по предположению, то в нем бесконечное число слов. Возьмем $\omega = xyz : |\omega| > m$. Но т.к. R конечный, то найдется слово, что $\forall i \geq 0 \;\; xy^i z$ не в R.
Теперь у нас отрицание л. о Накачке выполняется не только для L_1, но и для L, что неверно ибо L регулярный по условию. Получим, что наше допущение о нерегулярности неверно и L_1 точно регулярный.

7 Построить min автомат для языка L, заданного автоматом

Для начала сделаем наш ДКА всюду определенным, добавив состояние q_{rej}, попадая в которое, слово точно не примется. Теперь по алгоритму с семинара проведем разделение на классы эквивалентности с последующей проверкой эквивалентных переходов. Для удобства переходы будем смотреть по таблице переходов(она приведена к конечному виду для удобства).

δ	a	b
q_0	q_1	q_3
q_2	q_1	q_3
q_{rej}	q_{rej}	q_{rej}
q_1	q_{rej}	q_2
q_3	q_{rej}	q_{rej}



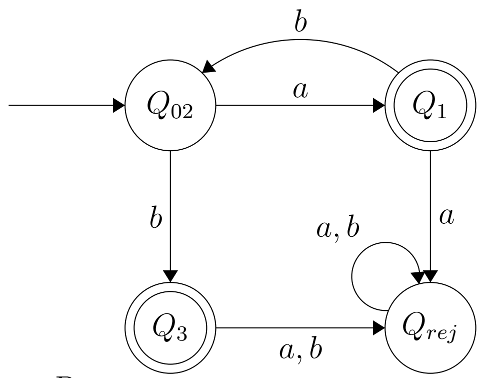
теперь последовательные проверки. Сначала проверка по принципу принимает/не принимает: $\{q_0, q_2, q_{rej} \mid q_1, q_3\}$

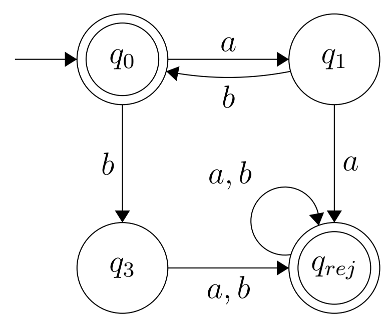
Далее по a: в q_{rej} ведут она сама, q_1, q_3. Но она находится в разных классах с 1 и 3. Тогда получим разделение: $\{q_0, q_2 \mid q_{rej} \mid q_1, q_3\}$

По a у нас все хорошо. Посмотрев по b, получим что q_1(ведет в q_2)не эквивалентна q_3(ведет в q_{rej}). Получим: $\{q_0, q_2 \mid q_{rej} \mid q_1 \mid q_3\}$

Тогда, объединив q_0и q_2 получим минимальный ДКА, изображенный справа.

8 Построить min автомат для языка $\overline{L}$ из предыдущей задачи





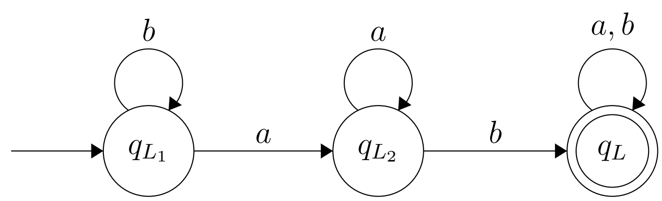
Ранее на семинарах мы определились с тем, что автомат по дополнению языка строится просто изменением принимающих состояний на непринимающие и наоборот.
Проверим его на минимальность. Сначала разделим состояния на классы по принципу принимающее/не принимающее: $\{q_0, q_{rej} \mid q_1, q_3\}$
Теперь проверим переходы по а q_{rej} переходит в себя и поэтому не эквивалентна q_0, которое переходит в q_1. q_3 и q_1 по а вместе переходят в q_{rej}. Получим $\{q_0 \mid q_{rej} \mid q_1, q_3\}$.
Теперь с переходами по а вс; хорошо. Проверим переходы по b: q_1 переходит в q_0, а q_3 переходит в q_{rej}. Значит они тоже не эквивалентны. Таким образом $\{q_0 \mid q_{rej} \mid q_1 \mid q_3\}$. И значит наш ДКА является минимальным.

9 Найдите все классы эквивалентности Майхилла-Нероуда для языка $\Sigma^* ab \Sigma^*$ и постройте ДКА

Выберем три языка, которые в последствии и будут классами эквивалентности. Что мы покажем позже: $L = \Sigma^* ab \Sigma^*, L_1 = b^*, L_2 = b^* a^*$

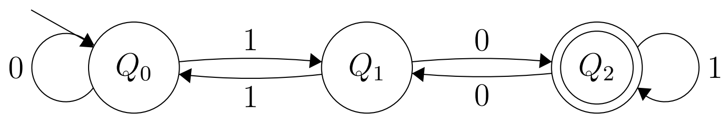
1. Рассмотрим L_1 и слова x и y из него. Тогда дописыванием можем получить:
   - $xa = b^n a, ya = b^m \in L_2 \notin L$
   - $xb = b^{n+1}, yb = b^{m+1} \in L_1 \notin L$
   - $xab = b^n ab, yab = b^m ab \in L$
   - остальные слова будут содержать эти и соответственно принадлежать одному из языков
2. Теперь рассмотрим $L_2 = b^* a^*; x = b^n a^i; y = b^m a^j$
   - $xa = b^n a^{i+1}, ya = b^m a^{j+1} \in L_2 \notin L$
   - $xb = b^n a^i b, yb = b^m a^j b \in L$
   - остальные слова будут содержать эти и соответственно принадлежать одному из языков
3. Теперь рассмотрим $L = \Sigma^* ab \Sigma^*$;. Т.к. слово уже принадлежит, то при дописывании оно и будет принадлежать.

Как мы видим, мы разобрали все случаи и получили классы эквивалентности Майхилла-Нероуда. Теперь построим по ним min ДКА:



10 Регулярен ли язык слов, дающих остаток 2 при делении на 3?

Язык регулярен потому что мы можем построить для него ДКА. Разберем, как он работает и докажем корректность. Собственно, вот он:



Сначала немного рассуждений об остатках. Арифметика остатков от деления на 3 работает как и обычная, только по модулю. Умножив число с остатком 0, получим число с остатком 0. Если был 1, стал 2. Если был 2, стал 1. В нашем случае есть только операции умножения на два - получение 0 справа. и умножения на 2+1.(Индексы автомата соответствуют остаткам при делениина 3). Докажем, что наш ДКА принимает слова-двоичные числа, дающие остаток 2 при делении на 3.

База: Пустое число имеет остаток 0. Куча нулей в начале(или их отсутствие) (т.е. число 0^*) никак не влияют на остаток. А в начале он 0. Если мы получили сначала число вида $0^*1 = 1$, то его остаток 1 и мы в Q_1. Если же число число было $0^*10 = 2$ то его остаток 2 и мы в Q_2. Число $0^*11 = 3 \equiv 0$

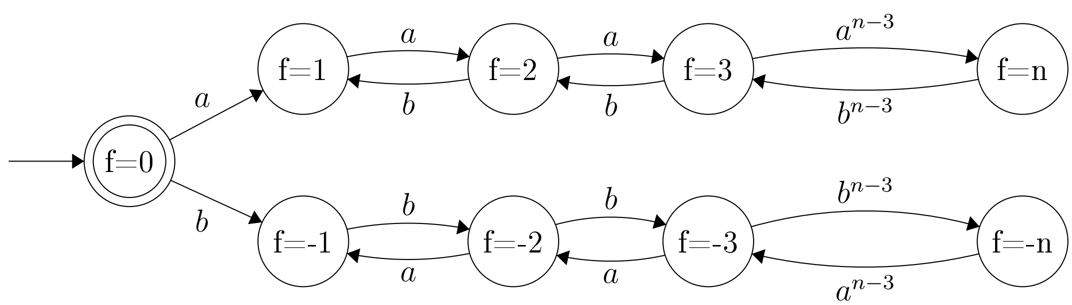
Шаг: Получив единичку, остаток +1, а мы переходим в Q_1. Далее, два варианта: либо мы получаем 1 и возвращаемся в Q_0 что значит, что наше число было умножено на два и была прибавлена 1. Т.е. остаток стал (1+1*2) mod 3=0. Либо мы получаем 0. Тогда наше число просто *2, как и его остаток, которые становится равным двум. И мы попадаем в принимающее Q_2. Переходы по 1 не меняют остатка, т.к. мы число *2 и +1. Т.е. остаток сохраняется и мы остаемся в принимающем состоянии. А если получаем 0, то просто умножаем его на два, и остаток станвится 1. Переходим в непринимающее Q_1

Таким образом, мы доказали корректность нашего автомата, который по сути имитирует арифметику по модулю. Заметим так же, что этот автомат минимальный (следует сразу из переходов) и разделяет все числа на три класса эквивалентности Майхилла-Нероуда множество всех чисел по их остатку от деления.

11 Опишите классы эквивалентности Майхилла-Нероуда для L и построить min ДКА

11.1 $L = \{\omega \mid |\omega|_a = |\omega|_b\}$

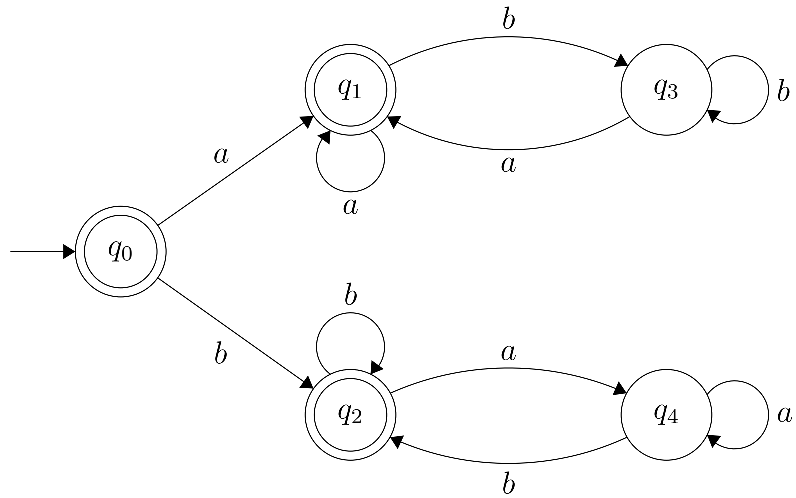
Введем функцию на множестве слов $f(\omega) = |\omega|_a - |\omega|_b$. Ее результат - целое число - разность количества букв а и б. Причем из задания функции сразу видны ее свойства: $f(\omega a) = f(\omega)+1; f(\omega b) = f(\omega)-1; f(\omega_1\omega_2) = f(\omega_1) + f(\omega_2)$. Просто схематично изобразим то, как она считается для КОНЕЧНОГО n(а дальше работает индукция. Но схему для счетного числа элементов построить уже затруднительно). Сразу говорю, в общем случае который мы разбираем ЭТО НЕ ДКА. Это просто схема того, как считается f. Хотя для конечного n(т.е. для слов у которых разность числа букв ограничена) это можно интерпретировать как ДКА, каждое из состояний которого принимает слова, с заданной разностью букв. Но в реальности n стремится к бесконечности, и мы не можем построить такой автомат. (что значит нерегулярность языка, доказанную в предыдущем задании)
Теперь само решение: классами эквивалентности будут слова с различной f (т.е. на каждый класс свое значение. Тогда будем обозначать их как $C_n, n \in \mathbb{Z}$). Тогда $\forall x, y \in C_n$ т.е.$f(x) = f(y)$ и $\forall \omega \in \Sigma^*$ получим $f(x\omega) = f(x) + f(\omega) = f(y) + f(omega) = f(y\omega)$ что и значит, что $x, y \in C_{n+f(\omega)}$. Значит все классы между собой различны, а внутри класса все элементы эквивалентны.



11.2 $L = \{\omega \mid |\omega|_{ab} = |\omega|_{ba}\}$

Вот мой автомат из предыдущего дз. Проверим его на минимальность. Заметим, что он всюду определен. Сначала разделим по принципу

принимающее/не принимающее $\{q_0, q_1, q_2 \mid q_3, q_4\}$
По a: $q_2 \nsim q_1, q_2 \nsim q_0, q_3 \nsim q_4$. Получим:$\{q_0, q_1 \mid q_2 \mid q_3 \mid q_4\}$
Теперь проверим по b: $q_0 \nsim q_1$. В итоге все различны: $\{q_0 \mid q_1 \mid q_2 \mid q_3 \mid q_4\}$
Значит каждое из состояний задает класс эквивалентности Майхилла-Нероуда. Посмотрим эти классы, как если бы мы не знали об этом ДКА.



1. $C_0 = \{\varepsilon\}; \forall x, y \in C_0$ дописывание любой цепочки переводит их в слова других классов, которые мы и разберем ниже.

2. $C_1 = \{a(a^*bb^*a)^*\};\ \forall x, y \in C_1 : x = a(a^ibb^ja)^k; i, j, k \geq 0; y = a(a^nbb^ma)^l; n, m, l \geq 0$; Рассмотрим дописывания: $xaa^* \in C_1; yaa^* \in C_1$, $xbb^* \in C_3; ybb^* \in C_3$, $xba \in C_1; yba \in C_1$. Остальные выражаются через эти, если обновлять x и y.

3. $C_3 = \{aa^*bb^*(aa^*bb^*)^*\};\ \forall x, y \in C_2 :$ аналогично. Рассмотрим дописывания: любое дописывание b оставляет нас в C_3. А любое дописывание букв a переводит в C_1. Остальные выражаются. И заметим, что из классов C_1 и C_3 мы никак не выйдем куда-то еще.

4. $C_2 = \{b(b^*aa^*b)^*\}$ полностью аналогично C_1, только меняем буквы a на b, а цифры 1 на 2 и 3 на 4.

5. $C_4 = \{bb^*aa^*(bb^*aa^*)^*\}$ полностью аналогично C_3, только меняем буквы a на b, а цифры 1 на 2 и 3 на 4.

Все классы рассмотрены. Теперь заметим, что они описывают наш ДКА и наоборот. Т.е. эти способы задания регулярного языка эквивалентны.