Домашнее Задание по ТРЯПу №2

Павливский Сергей Алексеевич , 873

18.09.2019

Задание 1.

Определим язык L $\subseteq \{a, b\}^*$ индуктивными правилами:

1. ε, b, bb $\in$ L;

2. вместе с любым словом x $\in$ L в L также входят слова ax, bax, bbax;

3. никаких других слов в L нет.

Язык T $\subseteq \{a, b\}^*$ состоит из всех слов, в которых нет трёх букв b подряд.

1. Докажите или опровергните, что L = T.

2. Запишите язык T в виде регулярного выражения.

3. Постройте конечный автомат, принимающий T.

Докажите (по индукции), что построенный автомат принимает язык T.

В случае, когда речь идёт об автомате A, для сокращения записи мы будем подразумевать, что данный автомат задан набором A = $(Q_A, \Sigma, {q_0}^A, \delta_A, F_A)$.

Решение

1.

Докажем , что L $\subseteq$ T , то есть , что в словах языка L нет 3 подряд идущих b . Каждое новое слово языка L индуктивно образуется из старого добавлением в его начала слова , заканчивающегося на a , и имеющее в начале не более двух b . Тогда по индукции в словах L нет 3 подряд идущих b : после добавления в начало одной из конструкций в слове не появится 3 подряд идущих b , а в исходных словах L 3 подряд идущих b нет .

Докажем , что $T \subseteq L$, то есть , что язык L содержит все слова , в которых подряд не идут 3 b . Любое слово из T будем строить по следующему правилу : идем с конца по слову , которое надо получить и записываем начало слова , которое строим, данный символ ; если встреченное подслово - a , а следующее после него в считываемом слове - b , то в наше строемое слово добавляем в начало подслово ba , а из считываемого слова выбрасываем это ba из рассмотрения; если встреченное подслово - a , а после него - a , то добавляем в начало строемого слова a , и выбрасываем это a из рассмотрения ; если в самом начале считываемое слово заканчивалось на b или bb , а перед одним из этих двух вариантов было a , то берем в качестве "каркаса"b или bb соответственно , а дальше следуем ранее описанному алгоритму , считывая a; если оно заканчивалось на a, то "каркас ε , а дальше следуем ранее описанному алгоритму , считывая a. Так как слова в T не содержат 3 b подряд , то это все возможные варианты конца слова из T - a, b, ab или ε (тогда слово пустое , а оно содержится в L) . Значит , так как мы описали все варианты окончания слова из T , и у нас есть алгоритм построения любого слова из T для любого из этих окончаний , то любое слово из T строится по правилам языка L , а так как в языке L содержатся все слова , которые строятся по правилам языка L , то $T \subseteq L$ ч.т.д.

Значит $L \subseteq T$, $T \subseteq L$, т.е. $L = T$ ч.т.д.

2.

Используем алгебраический метод Бжозовского для построения РВ по ДКА в пункте 3 .

$$\begin{cases} R_1 = aR_1 + bR_2 + \varepsilon \\ R_2 = aR_1 + bR_3 + \varepsilon \\ R_3 = aR_1 + \varepsilon \end{cases}$$

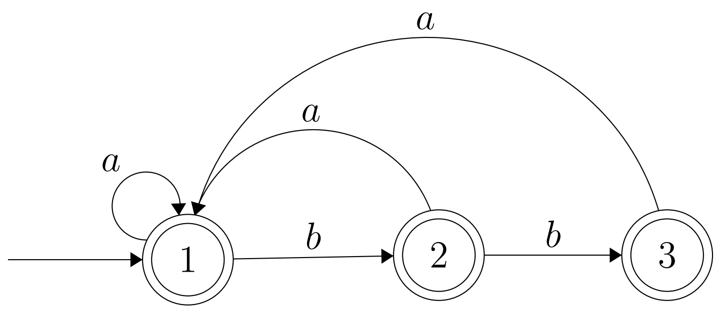
По теореме Ардена :

$$\begin{cases} R_1 = a^*(bR_2 + \varepsilon) \\ R_2 = aR_1 + baR_1 + b + \varepsilon \\ R_3 = aR_1 + \varepsilon \end{cases}$$

$$\begin{cases} R_1 = a^*(baR_1 + bbaR_1 + bb + b + \varepsilon) \\ R_2 = aR_1 + baR_1 + b + \varepsilon \\ R_3 = aR_1 + \varepsilon \end{cases}$$

$R_1 = a^*baR_1 + a^*bbaR_1 + a^*bb + a^*b + a^*$
По теореме Ардена :
$R_1 = (a^*ba|a^*bba)^*(a^*bb|a^*b|a^*) =$ РВ

3.



Так как из каждого состояния , кроме 3 , есть ребра a и b , то любое слово из языка корректно обработается , кроме тех , для которых из состояния 3 требуется пойти по ребру b , которого нет . Но в таких словах содержится 3 b подряд , так как после каждого встреченного a автомат переходит в состояние 1 , а такие слова и должны обрабатываться некорректно . Так как все состояния допускающие , то все слова без 3 b подряд закончатся обрабатываться в допускающем состоянии , в то время как слова , в которых есть 3 b подряд не будут обработаны , что и требовалось .

Задание 2.

Постройте ДКА A, B и C, такие что :

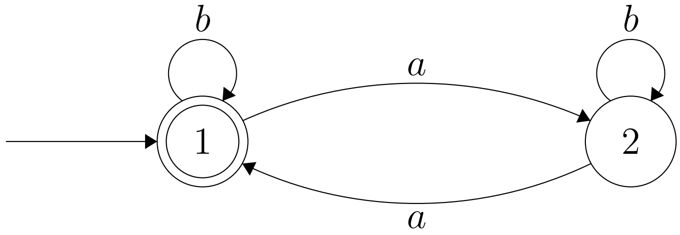
а) ДКА A распознаёт язык из всех слов с чётным числом букв a;

б) ДКА B распознаёт язык из всех слов с нечётным числом букв b;

в) ДКА C распознаёт язык из всех слов с чётным числом букв a и нечётным числом букв b.

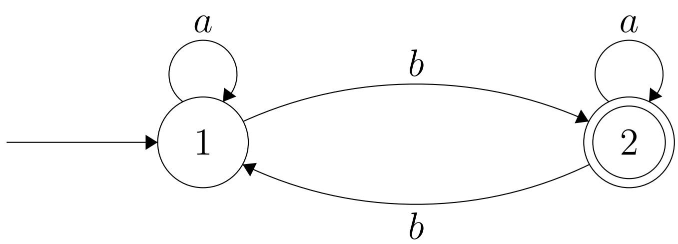
Решение

а)



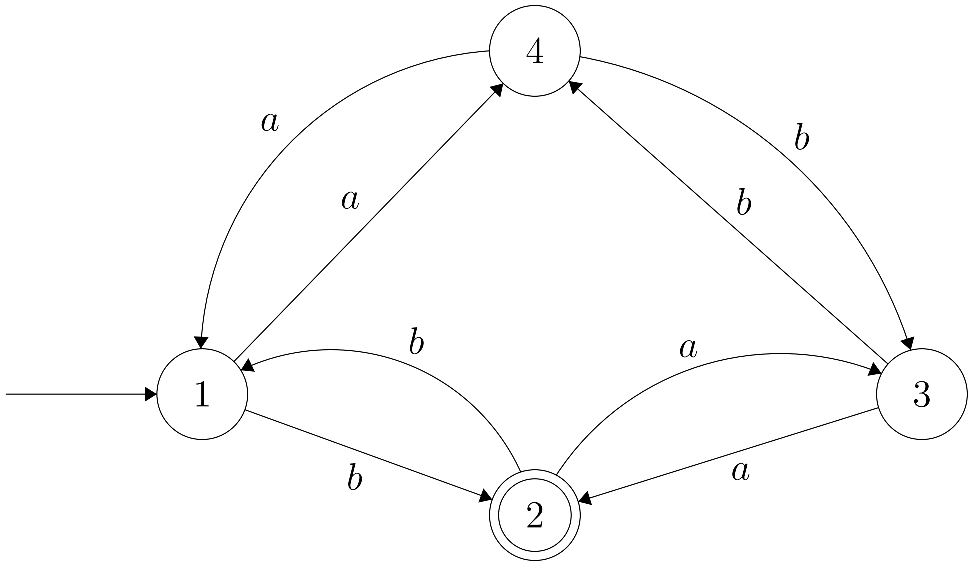
Каждое встреченное b не изменяет состояния автомата . То есть на корректность обработки влияет только количество a . После встреченного a автомат переходит в непринимающее состояние 2 , и будет находиться там до тех пор , пока не будет встреченно a в пару предыдущему . То есть если a разбиваются на пары (то есть их количество четно) , то слово примется , иначе автомат закончит работу в непринимающем состоянии , что и требовалось .

б)



Обоснование корректности автомата аналогично пукнту a .

в)



Каждое состояние соответствует одному из четырех случаев : в случае , когда на данный момент было считано четное число a и четное число b , автомат находится в состоянии 1 ; в случае , когда на данный момент было считано нечетное число a и нечетное число b , автомат находится в состоянии 3; в случае , когда на данный момент было считано нечетное число a и четное число b , автомат находится в состоянии 4 ; в случае , когда на данный момент было считано четное число a и нечетное число b , автомат находится в состоянии 2 (это видно из построения автомата , ребра есть только между вершинами , свойство четностей которых выполняется при добавлении новой вершины (например , НН соединено с НЧ и ЧН , но не соединено с ЧЧ , так как четность количества и a и b не могут поменяться при считывании одной вершины) . Поскольку принимающее состояние только 4 , то автомат корректно обрабатывает только слова с четным числом букв a и с нечетным числом букв b , что и требовалось .

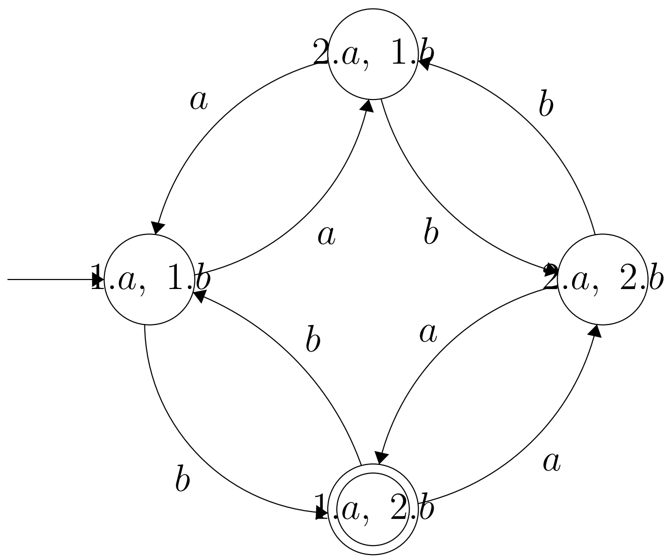
Задание 3.

Постройте автомат C из задачи 2(в) используя автоматы, построенные в первых двух пунктах и конструкцию произведения (при решении этой задачи можно не приводить решение 2(в)).

Решение

A1 = $\langle \Sigma = \{0, 1\}, Q_1 = \{1.a, 2.a\}, 1.a, F_1 = 1.a, \delta_1 \rangle$
A2 = $\langle \Sigma = \{0, 1\}, Q_2 = \{1.b, 2.b\}, 1.b, F_2 = 2.b, \delta_2 \rangle$

A1 x A2 =



Задание 4.

4.

1). Заменим в конструкции произведения множество принимающих состояний на множество $F_C = F_A \times Q_B \bigcup Q_A \times F_B$. Верно ли, что тогда автомат C распознаёт язык L(A) $\bigcup$ L(B)?

2). В случае ответа на первый вопрос «Да» приведите обоснование, в случае ответа на первый вопрос «Нет», приведите исправленную модификацию и докажите её корректность.

Решение

1)Нет , неверно . Даже на примере произведения из задания 3 : в случае нового переопределения $F_C = \{(1.a, 1.b), (2.a, 1.b), (2.a, 2.b), (1.a, 2.b)\}$. Но тогда все состояния становятся принимающими . В частности , они принимают слово состоящее из нечетного числа , среди букв которого только a . Но в этом слове 0 букв b , т.е. нечетное . Тогда данное слово $\notin L(A)$ и $\notin L(B)$, т.е. $\notin L(A) \bigcup L(B)$. Значит автомат С не распознает язык $L(A) \bigcup L(B)$.

Решение

$F_C = F_A \times Q_B \bigcap Q_A \times F_B$

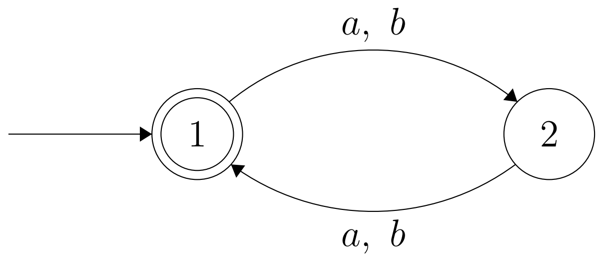
Левое произведение дает пары вида (не принимающие из А, принимающие из В) и (принимающие из А, принимающие из В); правое произведение дает пары вида (принимающие из А, не принимающие из В) и (принимающие из А, принимающие из В) . Тогда в пересечении имеем (принимающие из А , принимающие из В) , что и было нужно .

Задание 5.

Постройте ДКА A, распознающий язык из всех слов чётной длины. Пусть B — ДКА из задачи 2(б). Постройте ДКА C, распознающий язык L(A) △ L(B), модифицировав конструкцию произведения.

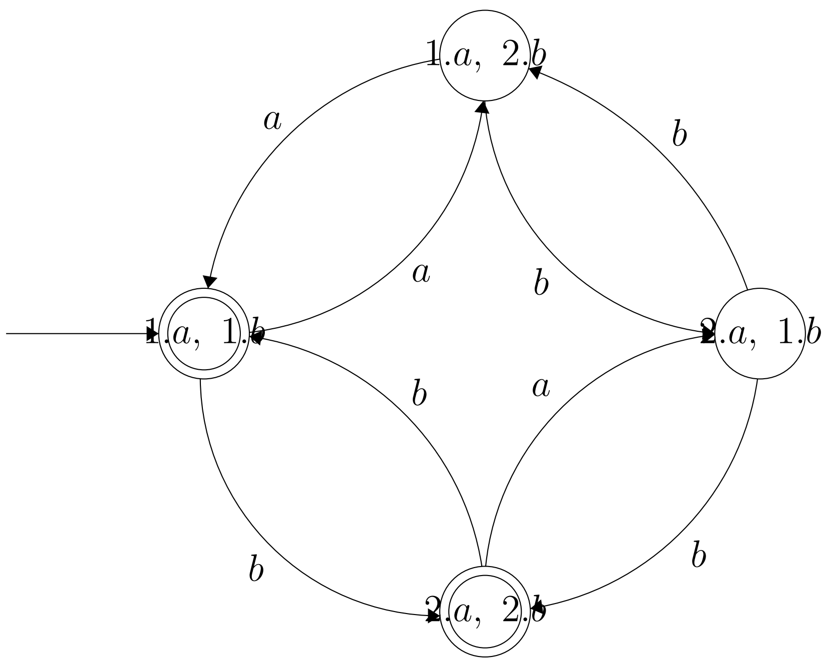
Решение

Автомат A =



Автомат , который мы должны построить , принимает слова , когда только одно из условий выполняется (либо четное количество букв в слове , либо нечетное количество букв b) . Тогда принимающими надо сделать состояния автомата , в которых ровно одно состояние принимающее , а второе нет .

Построим такой автомат :



Задание 6.

Постройте полиномиальный алгоритм, который, получив на вход ДКА A и B, проверяет, совпадают ли языки L(A) и L(B).

Решение

Учитывая предыдущую задачу , проверить совпадение языков можно построив их симметрическую разность . В случае их совпадения , она равна и в ДКА не будет состояний , в которых состояние ДКА одного языка принимающее , а состояние ДКА другого языка нет (свойство 1). Тогда за полиномиальное время (это обычный граф , заданный списком своих ребер и своими вершинами , который строится за полином) построим ДКА L(A) $\triangle$ L(B) , а дальше одним из алгоритмов обхода , например, BFS , пройдемся по всем вершинам и проверим их на свойство 1 . Обход , очевидно , выполняется за полиномиальное время . Значит все действия выполняются за полиномиальное время , то есть весь алгоритм полиномиальный , что и требовалось .