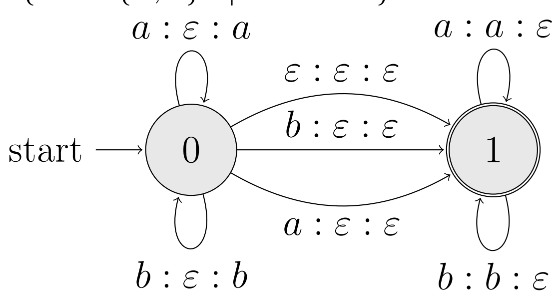
Formal languages HW6

Кушнерюк Сергей, Б05-925

Домашнее задание 6

Задача 1

(a) $\{\ w \in \{a, b\}^* \mid w = w^R\}$



Стековый алфавит совпадает с обычным алфавитом. Рассмотрим автомат.

Доказательство того, что получится автомат, распознавающий палиндромы, проведем по индукции. Действительно, слова длины длины 0 и 1, которые могут получится в этом автомате (ε, a, b) - база.

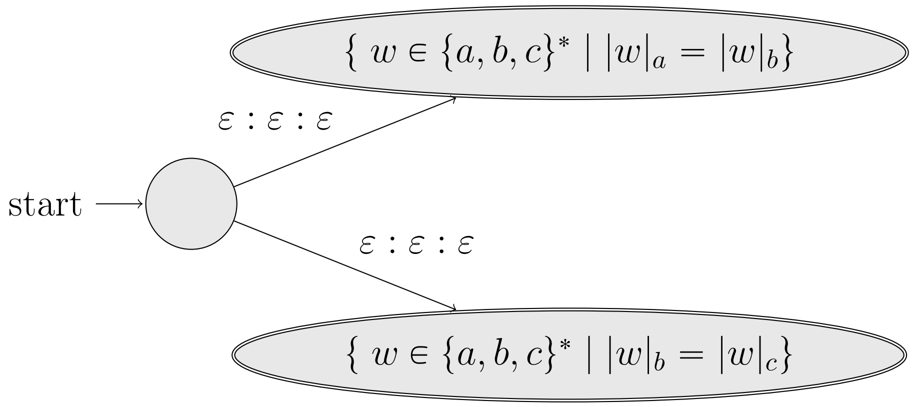
Теперь заметим, что в силу построения автомата, на петлях у 0 всегда кладется буква на стек, при переходе из 0 в 1 стек не меняется, при переходах по циклам на 1 буквы всегда снимаются. Таким образом, первая положенная буква на стек и последняя снятая совпадают. Тогда не рассматривая самый первый и самый последний переход в автомате (и, по сути не учитывая самый первый элемент в стеке) получается слово длины на 2 меньше, для которого предположение верно

Теперь докажем, что любой палиндром из букв a и b можно получить. Рассмотрим нечетный палиндром, четный будет рассматриваться аналогично, только серединный переход будет по ε

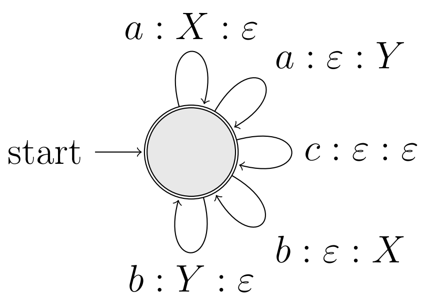
Действительно, по доказательству для предыдущего пункта заметим, что оно никак не зависело от того, какие именно буквы были добавлены в слово до серединного перехода. Будем набирать буквы до тех пор, пока не наберем все буквы кроме средней, возьмем ее, пройдя по переходу между 0 и 1 (стек при этом не изменится). Теперь на стеке у нас находятся все буквы первой части слова (кроме средней буквы), но в обратном порядке, так как каждый раз при добавлении буквы в слово она же добавлялась на стек. Таким образом теперь в слово добавим букву $w_{\frac{n-1}{2}-1}$ (0-индексация), которая равна в силу свойства слова $w_{\frac{n-1}{2}+1}$, затем $w_{\frac{n-1}{2}-2}$ и т.д. ▸

(b) $\{\ w \in \{a, b, c\}^* \mid |w|_a = |w|_b \vee |w|_b = |w|_c\}$

Построим автомат для $\{\ w \in \{a, b, c\}^* \mid |w|_a = |w|_b\}$, автомат для всего языка получится как объединение двух автоматов по следующей схеме

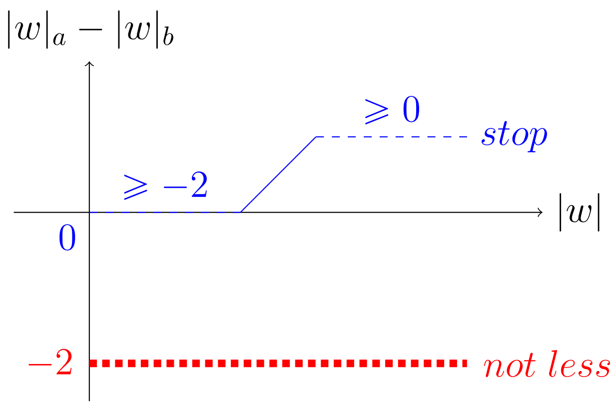


Аналогичный (без c) автомат был на семинаре, но повторим, как он выглядит



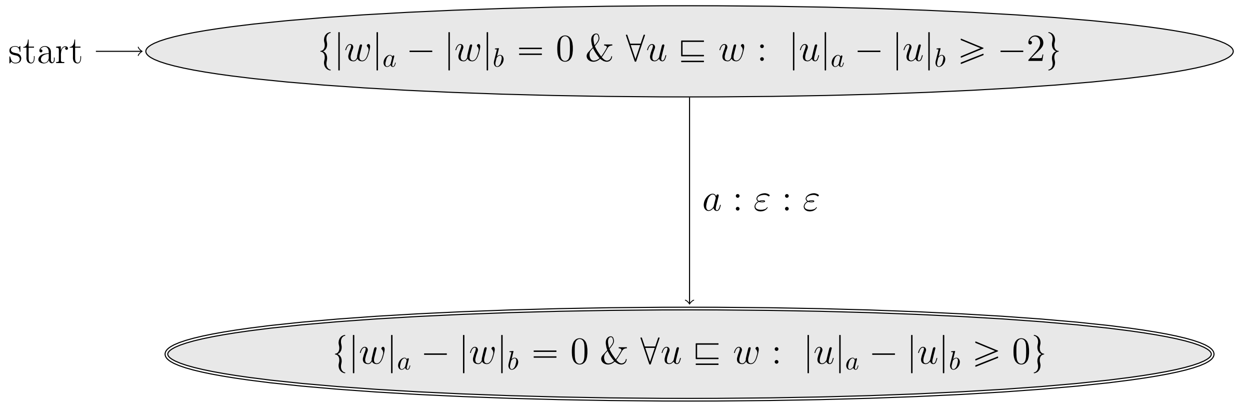
Дейсвительно, когда когда положим X, в слово попадем буква b, когда снимем - a, для Y наоборот $\Longrightarrow$ количество букв a и b будет равным. Кроме того, для каждого слова, определив участки, когда его баланс $|w|_a - |w|_b$ неотрицательный или неположительный по аналогии с ПСП на этом участке пользуясь переходами с X или с Y в зависимости от ситуации получим требуемое слово (в конце стек будет пуст из-за того, что строим, как ПСП). Буква c же никак на баланс не влияет и потому не должна влиять на стек, а также может быть приписана в любом месте слова

(c) Будем в слове находить последний момент, когда баланс 0, после чего он становится 1



(на пунктирах имеется в виду, что на префиксах этих подслов будет баланс со указанным свойством)

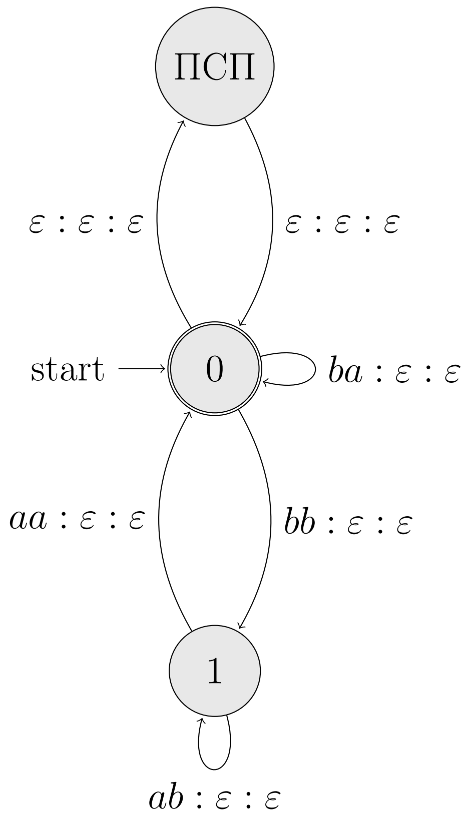
Таким образом, автомат будет строит слова для двух этих частей и выглядеть будет так:



(эти блоки - самостоятельные автоматы, состояния стека до и после входа в них равны)

Последняя вещь - это ПСП, которые мы строить уже умеем. Разберемся с 1 частью

Действительно, разобьем на блоки по моментам, когда баланс нулевой. Баланс неотрицательный $\implies$ можем эти участки выразить как ПСП, иначе баланс неположительный следовательно взяли букву b после момента с нулевым балансом



Тогда, после того, как взяли букву b есть ограниченное число случаев:

1. После этого взяли a - баланс снова 0, блок закончен

2. После этого взяли b, тогда в конце блока взяли aa, иначе блок был бы меньше, ибо в конце взяли ba, а значит баланс был перед b равен 0. Значит между этими моментами единственная возможность - это брать ab: aa - не может быть, ибо опять противоречие с выбором блока - нулевой баланс случился раньше; начинаться на b не может, так как баланс не меньше -2

Таким образом строгий разбор случаев гарантирует еще и однозначный порядок прохождения состояний и переходов для любого слова $\{|w|_a - |w|_b = 0 \ \& \ \forall u \sqsubseteq w: \ |w|_a - |w|_b \geqslant -2\}$

Задача 2.

0	1	2
		S_1 → AT
		S_1 → XA
S → aT	S → AT	S → AT
S → aSa	S → ASA	S → XA
		X → AS
T → bU	T → BU	T → BU
T → cT	T → CT	T → CT
U → Va	U → VA	U → VA
U → cU	U → CU	U → CU
V → a	V → a	V → a
V → bcVc	V → BCVC	V → YZ
		Y → BC
		Z → VC
	A → a	A → a
	B → b	B → b
	C → c	C → c

0. Исходная грамматика. Бесполезных правил нет
1. Убрали все смешанные правила
2. Убрали длинные правила. ε-правил нет. S не ε-порождающий.

   Пропустим шаги с добавлением правил $S_1 \to S$ и затем удалением одинарных правил и сразу добавим правила $S_1 \to AT \mid XA$

   Унарных правил (теперь) нет

Итого, получили грамматику в нормальной форме Хомского

Задача 3.

Алгоритм с лекции муторный, сделаем по своему

0	1	2	3	4
S → bT S → aS	S_1 → bT S_1 → aS S → bT S → aS	S_1 → bT S_1 → aS S → bT S → aS	S_1 → bT S_1 → aS S → bT S → aS	S_1 → bT S_1 → aS S → bT S → aS
T → cT T → Uc	T → cT T → UC	T → cT T → bVAC T → cUCC	T → cT T → bVC T → cUQ	T → cT T → bVC T → cUQ
U → bVa U → cUc	U → bVA U → cUC			
V → Vab V → b	V → VAB V → b	V → VAB V → b	V → VBA V → bA	V → bAV V → bA
	A → a B → b C → c	A → a B → b C → c	A → a B → b C → c	A → a B → b C → c
			Q → cC	Q → cC

0. Исходная грамматика

1. Исправим все смешанные правила, при этом оставляя первый символ неизменным, если это буква. Также добавим новый стартовый нетерминал аналогично тому, как он бы добавлялся в нормальной форме Хомского

2. Нетерминал U используется только в правилах нетерминала T, поэтому просто вставим правые части правил для U в правила для T. Действительно, подстановка всевозможных правых частей правил для нетерминалов вместо них самих не изменит грамматику: если некоторое слово выводилось ранее с использованием этого нетерминала, то будет выводится и сейчас, так как он уже раскрытый "вшит"всеми способами в те нетерминалы, которые его использовали, верно и обратное, так как "вшитые"правила для этих нетерминалов можно вновь вытащить и вернуть этот нетерминал в грамматику

3. $V \to Vab, V \to b$. Т.е. $V \vdash \{b(ab)^n \mid n \in \mathbb{N} \bigcup \{0\}\}$ (легко доказыается по индукции аналогично размышлениям в 4 пункте). Заметим теперь, что V используется только в нетерминале T, поэтому его можно исправить как нам удобно, что не повлияет на выводимость других частей грамматики. Сделаем так, что V теперь выводит исходные слова, к которым в конце приписано a, т.е. $V \vdash \{(ba)^n \mid n \in \mathbb{N}\}$

   Кроме этого, исправим оставшееся правило для нетерминала T так, чтобы в конце осталось ровно два нетерминала, добавив новый нетерминал в том виду, в котором он удовлетворяет виду для НФ Грейбах. Таким образом, правила для нетерминала T все будут в хорошей форме

4. Теперь просто приведем V в хорошую форму. Представим V как $V \to ba, V \to bAV$. Действительно, $V \vdash \{(ba)^n \mid n \in \mathbb{N}\}$, т.е. то, что раньше: по индукции, мы можем вывести только одно слово длины 2 - ba (база), а если $V \vdash (ba)^n$, то сначала применив единственно возможное теперь второе правило, затем вывод $bAV \to baV \vdash ba(ba)^n = (ba)^{n+1}$. В обратную сторону, вывод любого слова такого вида также доказывается по индукции: база очевидна, а применив аналогичные рассуждения, как в предыдущем пункте докажем требуемое