

Домашняя работа по ТРЯП №9

Автор - Айвазов Денис из 671 группы

1 ноября 2017

1 Построить SLG G_n порождающую слово

$$a^n b a^{n-1} b \dots a^3 b a^2 b a b a b a^2 b a^3 b \dots a^{n-1} b a^n b$$

Грамматика G_n будет описана правилами:

1. $S \rightarrow C_n$ - 1 вывод
2. $A_1 \rightarrow a$; $A_2 \rightarrow A_1 A_1$; $A_3 \rightarrow A_2 A_1$; $A_4 \rightarrow A_3 A_1 \dots A_n \rightarrow A_{n-1} A_1$; - всего n выводов. A_i порождает a^i
3. $C_1 \rightarrow A_1 b A_1 b$; $C_2 \rightarrow A_2 b C_1 A_2 b$; $C_3 \rightarrow A_3 b C_2 A_3 b$; $\dots C_n \rightarrow A_n b C_{n-1} A_n b$;

В итоге получаем, что для какого-то n имеем длину описания: $1+2n+5n+1=7n+2 < 8n$ т.е. $s=8$ и грамматика является SLG.

Само доказательство того, что грамматика порождает именно слово $a^n b a^{n-1} b \dots a^3 b a^2 b a b a b a^2 b a^3 b \dots a^{n-1} b a^n b$. Вывод выглядит так: $S \Rightarrow C_n \Rightarrow A_n b C_{n-1} A_n b \Rightarrow A_n b A_{n-1} b C_{n-2} A_{n-1} b A_n b \Rightarrow^* A_n b A_{n-1} b \dots A_1 b A_1 b \dots A_{n-1} b A_n b \Rightarrow^* a^n b a^{n-1} b \dots a b C_1 a b \dots a_{n-1} b a_n b$

заметим, что все выводы являются детерминированными (т.е. тут однозначно строится дерево выводов) и ни у какого вывода нет двух возможных "толкований".

2 * Построить и доказать алгоритм

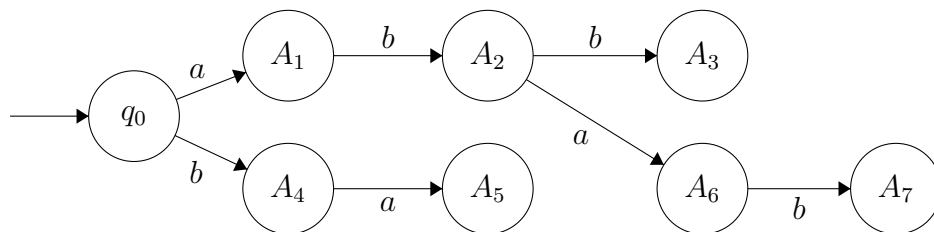
3 Построить LZW-автомат и SLG по слову

3.1 $\omega_1 = a^8$

По приведенному и доказанному на листке алгоритму построим разбиение слова:

a	ab	abb	b	ba	aba	abab
A_1	A_2	A_3	A_4	A_5	A_6	A_7

и LZW- автомат:



SLG задается переходами: $S \rightarrow A_1 A_2 A_3 A_4 A_5 A_6 A_7$; $A_1 \rightarrow a$; $A_2 \rightarrow A_1 b$; $A_3 \rightarrow A_2 b$; $A_6 \rightarrow A_2 a$; $A_7 \rightarrow A_6 b$; $A_4 \rightarrow b$; $A_5 \rightarrow A_4 a$.

3.2 $\omega_2 = \text{tobeornottobeortobeornot}$

По тому же алгоритму построим:

t	o	b	e	or	n	ot	to	be	ort	ob	eo	r	no	t
A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	$A_{15} \equiv A_1$

Сама грамматика будет иметь вид:

$S \rightarrow A_1 A_2 A_3 A_4 \dots A_{14} A_{15};$

$A_1 \rightarrow t; A_2 \rightarrow o;$

$A_3 \rightarrow b; A_4 \rightarrow e;$

$A_5 \rightarrow A_2 r; A_6 \rightarrow n;$

$A_7 \rightarrow A_2 t; A_8 \rightarrow A_1 o;$

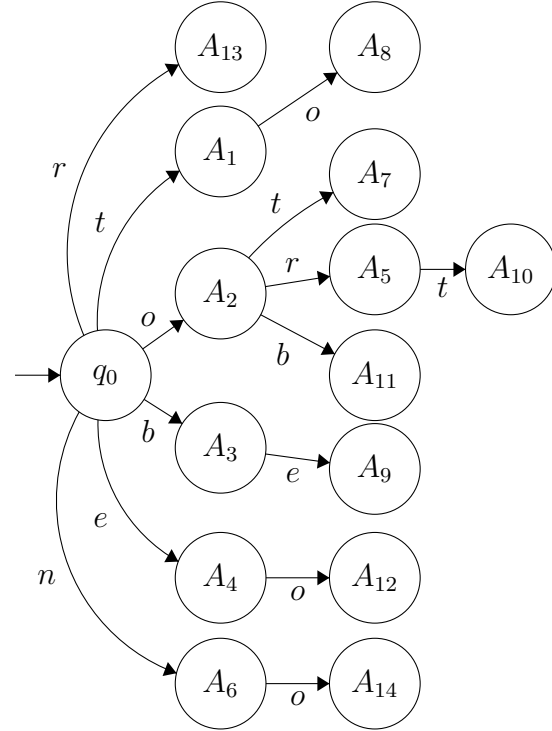
$A_9 \rightarrow A_3 e; A_{10} \rightarrow A_5 t;$

$A_{11} \rightarrow A_2 b; A_{12} \rightarrow A_4 o;$

$A_{13} \rightarrow r; A_{14} \rightarrow A_5 t;$

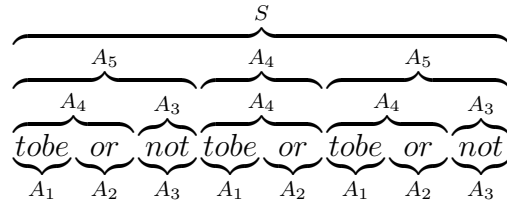
$A_{15} \rightarrow A_1;$

Заметим, что хоть эта грамматика построена по данному алгоритму абсолютно верно, с точки зрения сжатия она не оптимальна. Сумма длин правых частей правил у нее: $15+7+8*2=38$. При длине слова 24. В следующем задании мы ее улучшим.



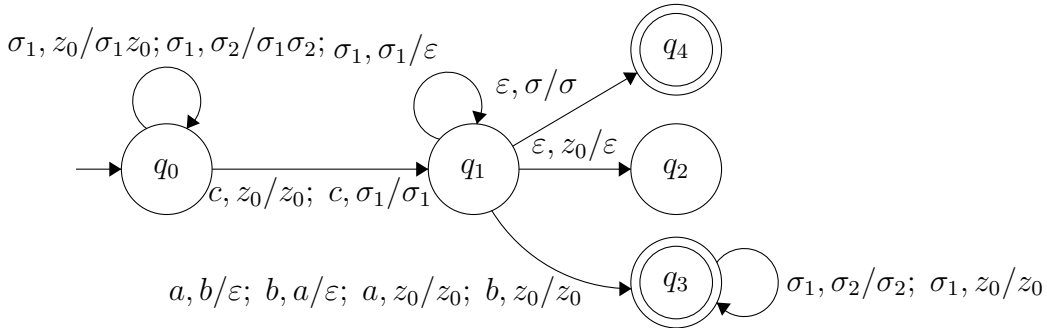
4 Построить для ω_2 более оптимальную SLG

Заметим, что по данному слову грамматику можно построить проще:



$S \rightarrow A_5 A_4 A_5$; $A_5 \rightarrow A_4 A_3$; $A_4 \rightarrow A_1 A_2$; $A_3 \rightarrow not$; $A_2 \rightarrow tobe$; $A_1 \rightarrow or$;
 Выводимость также однозначна и показана на рисунке, который можно интерпретировать как дерево вывода слова ω_2 из нетерминала S .
 В данном случае сумма длин правых частей равна: $3+2+2+3+4+2=16$, что меньше длины слова (24) и тем более меньше результата из прошлого задания (38)

5 Построить ДМП-автомат для языка L ; $L = \{xscy \mid x, y \in \{a, b\}^*; x \neq y^R\}$



Вот построенный по языку L МП-автомат, принимающий по принимающему состоянию.

σ_1, σ_2 на схеме это буквы из $\{a, b\}$. Причем σ_1 может быть равно σ_2 , а может и не быть. Сразу опишем что значат состояния: МП прокручивает слово x и кладет его в стек в состоянии q_0 . Как только он встретит c , то по переходу $q_0 \rightarrow q_1$ попадет в состояние q_1 , уже будет прокручивать y и сравнивать его побуквенно со стеком. Тогда возможны варианты:

1. $y^R = x$ тогда он будет крутиться в q_1 пока буквы в стеке и поступающие на вход не закончатся одновременно. И он перейдет по переходу $\varepsilon, z_0/\varepsilon$ в состояние q_2 которое говорит, что наше слово - палиндром с s посередине. А это слово не принимается и не принадлежит L
2. $\exists i : x_i^R \neq y_i$ - значит мы нашли несовпадение. И слово уже принадлежит языку L . А как только автомат получил несовпадение, то ему ничего не остается кроме как пойти по $a, b/\varepsilon$ или $b, a/\varepsilon$ в состояние q_3 которое значит, что несовпадение найдено и слово уже принимается (хотя и может продолжить крутиться до конца стека и y)
3. $|y| > |x|$ (при равных предыдущих символах) т.е. стек опустеет быстрее, хотя буквы на входе еще будут. Этому соответствуют переходы $a, z_0/z_0$; $b, z_0/z_0$ из q_1 в q_3 . Слово принимается как языком так и автоматом.
4. $|y| < |x|$ (при равных предыдущих символах) т.е. слова на входе закончатся при непустом стеке. Значит наш автомат перейдет в состояние q_4 по переходу $\varepsilon, \sigma/\sigma$ и остановится в этом принимающем состоянии. Считаем, что он переходит по эpsilon-переходам только с пустым входом. В данном случае слово тоже принимается как автоматом так и языком.

Из рассмотрения всех возможных случаев следует, что построенный ДМП распознает язык L .

Он является детерминированным по определению

6 * язык префиксов всех слов языка

7 Построить приведенную грамматику

Грамматика выглядела так:

$S \rightarrow$	A	B	C	E	AG
$A \rightarrow$	C	aABC	ε		
$B \rightarrow$	bABa	aCbDaGb	ε		
$C \rightarrow$	BaAbC	aGD	ε		
$F \rightarrow$	aBaaCbA	aGE			
$E \rightarrow$	A				

Теперь удалим из нее все бесплодные символы действуя по алгоритмы:

$V_0 = \{a, b\}$; $V_1 = \{a, b, A, B, C\}$; $V_2 = \{a, b, A, B, C, S, F, E\}$;

$V_3 = \{a, b, A, B, C, S, F, E\} = V_2$ $N = \{S, A, B, C, E, F\} \Rightarrow G$ бесплод.

Удалим его и все, что с ним связано:

$S \rightarrow$	A	B	C	E
$A \rightarrow$	C	aABC	ε	
$B \rightarrow$	bABa	ε		
$C \rightarrow$	BaAbC	ε		
$F \rightarrow$	aBaaCbA			
$E \rightarrow$	A			

Займемся поиском недоступных символов в точности следуя алгоритму:

$V_0 = \{S\}$; $V_1 = \{S, A, B, C, E\}$; $V_2 = \{S, A, B, C, E\} = V_1$. Значит F

-недоступен. Расстрелять.

$S \rightarrow$	A	B	C	E
$A \rightarrow$	C	aABC	ε	
$B \rightarrow$	bABa	ε		
$C \rightarrow$	BaAbC	ε		
$E \rightarrow$	A			