Домашнее задание №11

Томинин Ярослав, 778 группа

26 ноября 2018 г.

1.

Рассмотрим СУ, которая делает перевод $w \to \pi_l(w)$. Она была рассмотрена в примере 1. Построим вывод ((a))в грамматике G.

$$
\begin{aligned}
&E \to T\\
&T \to F\\
&F \to (E)\\
&E \to T\\
&T \to F\\
&F \to (E)\\
&E \to T\\
&T \to F\\
&F \to a
\end{aligned}
$$

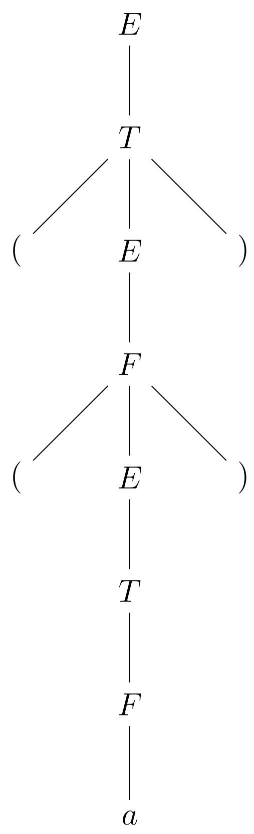
Теперь применим те же правила в грамматике, в которую нас перевел анализатор.

$$
\begin{aligned}
&E \to 2T\\
&T \to 4F\\
&F \to 5E\\
&E \to 2T\\
&T \to 4F\\
&F \to 5E\\
&E \to 2T\\
&T \to 4F\\
&F \to 6
\end{aligned}
$$

Получаем левый разбор-результат выполнение правил:245245246
Для того, чтобы получить правый разбор, достаточно во всех правилах вывода G' перенести цифру в конец вывода. Тогда получаем

$$
\begin{aligned}
&E \to T2\\
&T \to F4\\
&F \to E5\\
&E \to T2\\
&T \to F4\\
&F \to E5\\
&E \to T2\\
&T \to F4\\
&F \to 6
\end{aligned}
$$

Тогда правый разбор: 642542542
Или можно сделать все то же самое, построив дерево разбора. В данном случае дерево разбора для левого и правого вывода одинаковое.



2.

F_i	E	T	F	E'	T'
F_0	$\oslash$	$\oslash$	$\oslash$	ϵ	ϵ
F_1	$\oslash$	$\oslash$	(,id	$\epsilon, +$	$\times, \epsilon$
F_2	$\oslash$	(,id	(,id	$\epsilon, +$	$\times, \epsilon$
F_3	$(, id$	(,id	(,id	$\epsilon, +$	$\times, \epsilon$

Теперь вычислим FOLLOW
Добавим правило $S' \to E\$$

F_i	E	T	F	E'	T'
F_0	\$	$\oslash$	$\oslash$	$\oslash$	$\oslash$
F_1	$\$,)$	$\$, +$	$\times$	\$	$\oslash$
F_2	$\$,)$	$\$, +,)$	$\$, \times, +$	$\$,)$	$\$, +$
F_3	$\$,)$	$\$, +,)$	$\$, \times, +,)$	$\$,)$	$\$, +,)$
F_4	$\$,)$	$\$, +,)$	$\$, \times, +,)$	$\$,)$	$\$, +,)$
F_5	$\$,)$	$\$, +,)$	$\$, \times, +,)$	$\$,)$	$\$, +,)$

Теперь построим анализатор.

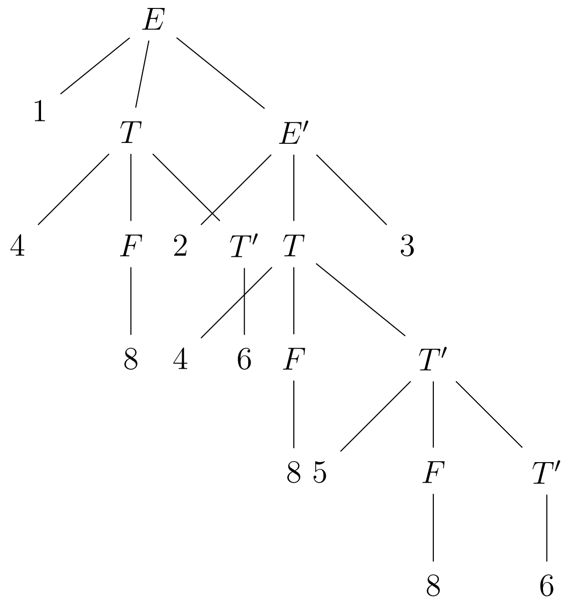
0	E	T	F	E'	T'
id	$E \to TE'$	$T \to FT'$	$F \to id$	$NONE$	$NONE$
$+$	NONE	$T \to FT'$	NONE	$E' \to +TE'$	$T' \to \epsilon$
$\times$	NONE	NONE	NONE	NONE	$T' \to \times FT'$
(	$E \to TE'$	$T \to FT'$	$F \to (E)$	NONE	NONE
)	NONE	NONE	NONE	$E' \to \epsilon$	$T' \to \epsilon$
\$	NONE	NONE	NONE	$E' \to \epsilon$	$T' \to \epsilon$

Для демострации запишем конфигурации
Для начала занумеруем правила

$$
\begin{aligned}
E \to TE'(1) \\
E' \to +TE'|\epsilon(2,3) \\
T \to FT'(4) \\
T' \to \times FT'|\epsilon(5,6) \\
F \to (E)|id(7,8)
\end{aligned}
$$

$$
\begin{aligned}
&(id + id \times id\$, E\$,)\\
&(id + id \times id\$, TE'\$, 1)\\
&(id + id \times id\$, FT'E'\$, 14)\\
&(id + id \times id\$, idT'E'\$, 148)\\
&(+id \times id\$, T'E'\$, 148)\\
&(+id \times id\$, E'\$, 1486)\\
&(+id \times id\$, +TE'\$, 14862)\\
&(id \times id\$, TE'\$, 14862)\\
&(id \times id\$, FT'E'\$, 148624)\\
&(id \times id\$, idT'E'\$, 1486248)\\
&(\times id\$, T'E'\$, 1486248)\\
&(\times id\$, \times FT'E'\$, 14862485)\\
&(id\$, FT'E'\$, 14862485)\\
&(id\$, idT'E'\$, 148624858)\\
&(\$, T'E'\$, 148624858)\\
&(\$, E'\$, 1486248586)\\
&(\$, \$, 14862485863)\\
&(\epsilon, \epsilon, 14862485863)
\end{aligned}
$$

Ура, это слово принимается анализатором!
Дерево разбора:



Это дерево сходится с предыдущей последовательностью выводов.

3.

Построим FIRST

F_i	S	A
F_0	$\oslash$	ϵ
F_1	a, b	ϵ, a, b
F_2	a,b	a, b, ϵ

Из этой таблицы уже видно, что пересечение FIRST(A) и FOLLOW(A) не пустое множество и содержит b, так как b точно содержится в FOLLOW и мы знаем, что оно содержится в FIRST. Так как $\epsilon \in FIRST(A)$, то не выполняется 2 критерий.

Построим $FIRST_2(A), FOLLOW_2(A)$. Для этого воспользуемся алгоритмом

F_i	S	A
F_0	$\oslash$	b, ϵ
F_1	aa, ab, bb	ϵ, b
F_2	aa, ab, bb	b, ϵ

Теперь построим $FOLLOW_2(A)$.

Сделаем грамматику пополненной.

$S' \to S\$$

F_i	S	A	S'
F_0	\$	aa, ba	$\oslash$
F_1	\$	aa, ba	$\oslash$

Отлично, теперь проверим выполняется ли критерий.
Рассмотрим первые два правила
$FIRST_2(aAaa) = FIRST_2(a) \oplus_2 FIRST_2(A) \oplus_2 FIRST_2(a) = \{a\} \oplus_2 \{b, \epsilon\} \oplus_2 \{a\} = \{aa, ab\}$
$FIRST_2(babA) == \{ba\}$
Так как нет пересечений, то здесь нет противоречияя критерию.
Рассмотрим последние два правила.
$FIRST_2(b\alpha) = FIRST_2(b) \oplus_2 FOLLOW_2(A) = \{a\} \oplus \{aa, ba\} = \{ba, bb\}$
$FIRST_2(\epsilon\alpha) = FOLLOW_2(A) = \{aa, ba\}$
Заметим, что последние два множества имеют пересечение, поэтому это грамматика не является LL(2)-грамматикой.
4.

$$
\begin{aligned}
&S \to baaA|babA \\
&A \to \epsilon|Aa|Ab
\end{aligned}
$$

Преобразуем грамматику

$$
\begin{aligned}
&S \to baQ(1) \\
&Q \to aA|bA(2,3) \\
&A \to A'(4) \\
&A' \to aA'|bA'|\epsilon(5,6,7)
\end{aligned}
$$

Построим FIRST

F_i	S	A	A'	Q
F_0	$\oslash$	ϵ	$\oslash$	$\oslash$
F_1	b	ϵ	a, b	a, b
F_2	b	ϵ, a, b	a, b	a, b

Построим FOLLOW

F_i	S	A	A'	Q
F_0	\$	$\oslash$	$\oslash$	$\oslash$
F_1	\$	$\oslash$	$\oslash$	\$
F_2	\$	\$	\$	\$
F_3	\$	\$	\$	\$

Построим таблицу анализатора.

0	S	A	A'	Q
a	NONE	4	5	2
b	1	4	6	3
\$	NONE	4	7	NONE

Демонстрация с помощью конфигураций.

$$
\begin{aligned}
&(baab\$, S\$,)\\
&(baab\$, baQ\$, 1)\\
&(aab\$, aQ\$, 1)\\
&(ab\$, Q\$, 1)\\
&(ab\$, aA\$, 12)\\
&(b\$, A\$, 12)\\
&(b\$, A'\$, 124)\\
&(b\$, bA'\$, 1246)\\
&(\$, A'\$, 1246)\\
&(\$, \$, 12467)\\
&(\epsilon, \epsilon, 12467)
\end{aligned}
$$

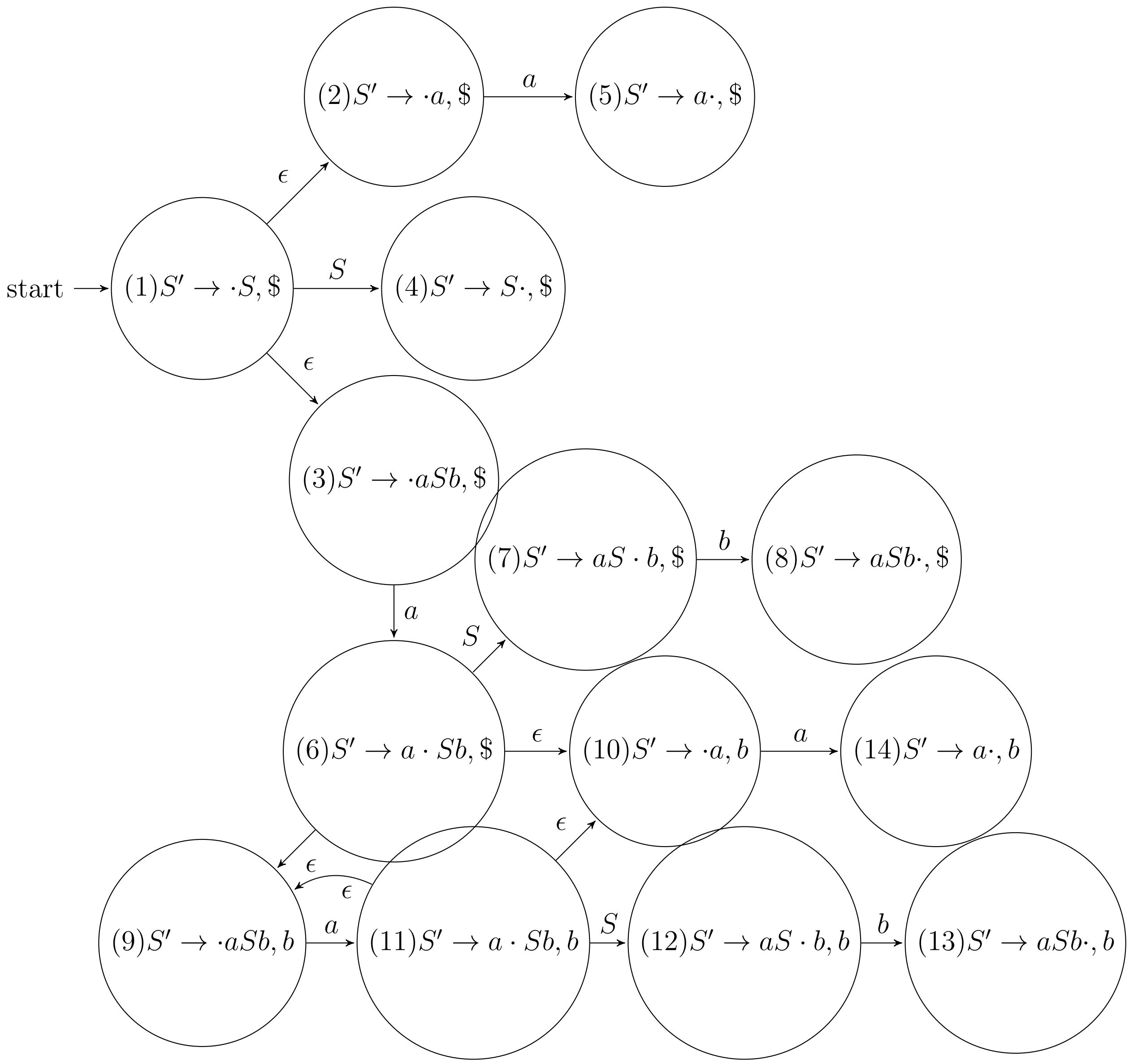
Слово принимается нашей грамматикой.

5.

$$
\begin{aligned}
&S' \to S\$\\
&S \to aSb|a
\end{aligned}
$$

Рассмотрим начальное состояние $[S' \to \cdot S\$,$
$epsilon]$

Проведем поиск в ширину, то есть из достижимых вершин будем проводить все возможные переходы. Закончим, когда не появится ничего нового на k-ой итеррации.



Построим ДКА по НКА

Q_i	a	b	S
$Q_0 = \{1, 2, 3\}$	$Q_1 =$ 5,6,9,10	-	$Q_2 =$4
$Q_1 = \{5, 6, 9, 10\}$	$Q_3 =$11,10,9,14	-	$Q_4 =$7,12
$Q_2 = \{4\}$	-	-	-
$Q_3 = \{9, 10, 11, 14\}$	$Q_3 =$9,10,11,14,	-	$Q_5 =$12
$Q_4 = \{7, 12\}$	-	$Q_6 =$8,13	-
$Q_5 = \{12\}$	-	$Q_7 =$13	-
$Q_6 = \{8, 13\}$	-	-	-
$Q_7 = \{13\}$	-	-	-

Эта таблица определяет ДКА.

6.

Поймем, что в произвольной грамматике, которая порождает наш язык есть два правила вида

$$S \to \alpha$$
$$S \to \beta$$

Причем мы знаем, что $a^k \in FIRST_k(\alpha)$ и $a^k \in FIRST_k(\beta)$.
Если мы докажем этот факт, то мы получим непустое пересечение, следовательно, наша грамматика не будет LL. Так как мы можем сказать это для произвольной грамматика, то наш язык не является LL языком.
Поймем, что из S должен выводиться a^k, поэтому в одном из FIRST точно содержится a^k.
Так же поймем, что у нас точно будет два правила такого вида. Докажем это от противного. Допустим, что существует только одно правило такого вида, тогда наша грамматика бужет допускать слова $a^k b^i$, но этого быть не должно. Поэтому у нас есть два правила, где второе правило должно "отвечать "за вывод $a^n b^n$, поэтому в FIRST второго правила тоже содержится a^k.