

Решение задачи №4. Вариант 18

Условие: Построить SLR-анализатор для грамматики:

1. $S \rightarrow BS$
2. $S \rightarrow c$
3. $A \rightarrow cB$
4. $A \rightarrow c$
5. $B \rightarrow Ba$
6. $B \rightarrow Aa$

1 Определение функций FIRST и FOLLOW

Для построения SLR-таблицы необходимо вычислить множества FIRST (первые терминалы) и FOLLOW (терминалы, которые могут следовать за нетерминалом). Добавим пополняющее правило $S' \rightarrow S$ (правило 0).

- **FIRST:**

- $FIRST(A) = \{c\}$
- $FIRST(B) = FIRST(B) \cup FIRST(A) \implies \{c\}$
- $FIRST(S) = FIRST(B) \cup \{c\} \implies \{c\}$

- **FOLLOW:**

- $FOLLOW(S') = \{\$\}$
- $FOLLOW(S) = FOLLOW(S') \cup FOLLOW(S) = \{\$\}$
- $FOLLOW(B) = FIRST(S) \cup \{a\} \cup FOLLOW(S) = \{c, a, \$\}$ (из $S \rightarrow BS$ и $B \rightarrow Ba$)
- $FOLLOW(A) = \{a\}$ (из $B \rightarrow Aa$)

Нетерминал	FIRST	FOLLOW
S	$\{c\}$	$\{\$\}$
A	$\{c\}$	$\{a\}$
B	$\{c\}$	$\{a, c, \$\}$

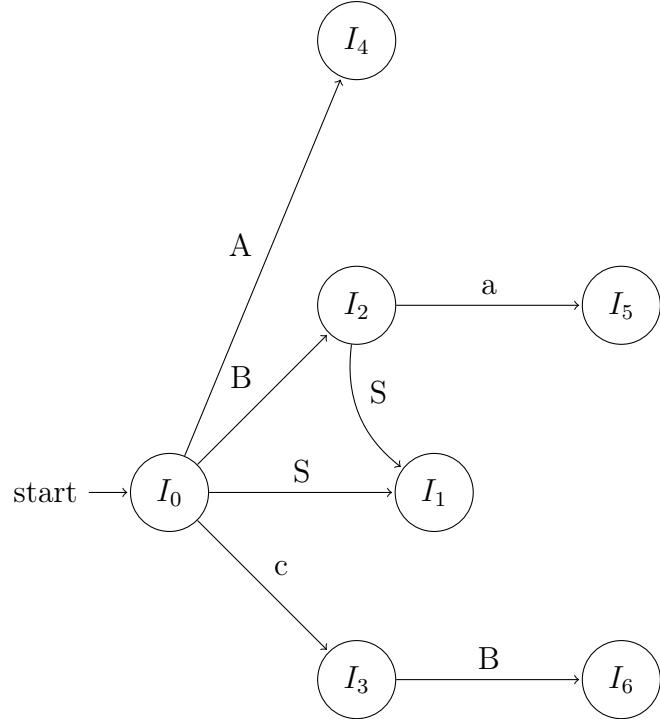
2 Построение множества пунктов (состояний)

Сформируем состояния автомата с помощью замыкания (Closure) множеств LR(0)-пунктов.

- I_0 : Замыкание($S' \rightarrow \bullet S$)
 $S' \rightarrow \bullet S, S \rightarrow \bullet BS, S \rightarrow \bullet c, B \rightarrow \bullet Ba, B \rightarrow \bullet Aa, A \rightarrow \bullet cB, A \rightarrow \bullet c$
- $I_1 = \text{Goto}(I_0, S)$: $S' \rightarrow S \bullet$ (Принимающее состояние)
- $I_2 = \text{Goto}(I_0, B)$: $S \rightarrow B \bullet S, B \rightarrow B \bullet a, S \rightarrow \bullet BS, S \rightarrow \bullet c, \dots$ (Замыкание)
- $I_3 = \text{Goto}(I_0, c)$: $S \rightarrow c \bullet, A \rightarrow c \bullet B, A \rightarrow c \bullet, B \rightarrow \bullet Ba, \dots$ (Конфликты в LR(0), разрешаемые в SLR)

- $I_4 = \text{Goto}(I_0, A): B \rightarrow Aa\bullet$
- $I_5 = \text{Goto}(I_2, a): B \rightarrow Ba\bullet$
- $I_6 = \text{Goto}(I_3, B): A \rightarrow cB\bullet, B \rightarrow B \bullet a$

3 Диаграмма переходов



4 Таблица SLR-анализатора

Обозначения: sN — перенос в состояние N , rN — свёртка по правилу N , acc — допуск. В пустых ячейках подразумевается обработка ошибок (например, $e1$: пропуск символа, $e2$: вставка ожидаемого токена).

Сост.	Action (Терминалы)			Goto (Нетерминалы)		
	a	c	\$	S	A	B
0	$e1$	$s3$	$e2$	1	4	2
1			acc			
2	$s5$	$s3$	$e2$	1	4	2
3	$r4$	$s3$	$r2$			6
4	$s5$					
5	$r5$	$r5$	$r5$			
6	$s5$	$r3$	$r3$			

5 Проверка на примерах

Пример 1 (Правильный): $c\$$

- Стек: [0], Вход: $c\$$. Action(0, c) = $s3$.

- Стек: $[0, c, 3]$, Вход: $\$$. Action(3, $\$$) = $r2 (S \rightarrow c)$.
- Стек: $[0]$, S на входе. Goto(0, S) = 1. Стек: $[0, S, 1]$.
- Action(1, $\$$) = **Accept**.

Пример 2 (Неправильный): a\$

- Стек: $[0]$, Вход: a $\$$. Action(0, a) = $e1$ (Ошибка: ожидалось c или нетерминал).
- Восстановление: Пропуск a, попытка разбора дальше.

Пример 3 (Неправильный): cc\$

- Стек: $[0, c, 3]$, Вход: c $\$$. Action(3, c) = $s3$ (перенос).
- Стек: $[0, c, 3, c, 3]$, Вход: $\$$. Action(3, $\$$) = $r2 (S \rightarrow c)$.
- Стек: $[0, c, 3, S, 1]$, Вход: $\$$.
- **Ошибка:** После $S \rightarrow c$ не может быть другого S без предшествующего B .