

Вариант 10

1. Для регулярного выражения над алфавитом $T = \{a, b\}$: $b(ba|b)^*|b$
построить эквивалентный детерминированный конечный автомат.
2. Дана грамматика $G = (\{S, A\}, \{a, b\}, P, S)$ с правилами
 $S \rightarrow ba | A$
 $A \rightarrow a | Aab | Ab$
Построить LL(1)-грамматику, эквивалентную данной.
3. Для логического выражения $A \text{ and not } (B \text{ or } C) \text{ or } (D \text{ and } E)$ сгенерировать код на командах перехода и изобразить атрибутивное дерево.
4. Пусть $G = (N, T, P, S)$ – LR(1)-грамматика, $w \in T^*$.
В каких случаях (в зависимости от G и w) LR(1)-анализатор при анализе цепочки w не сделает ни одного сдвига?
5. Для оператора присваивания $a = b[i] + b[j]$ сгенерировать оптимальный код методом сопоставления образцов.

Вариант 9

1. Для регулярного выражения над алфавитом $T = \{a, b\}$: $(a|b)^* ab(a|b)$
построить эквивалентный детерминированный конечный автомат.
2. Дана грамматика $G = (\{S, A\}, \{a, b\}, P, S)$ с правилами
 $S \rightarrow aaS | abA$
 $A \rightarrow e | Aa | Ab$
Построить LL(1)-грамматику, эквивалентную данной.
3. Для логического выражения $A \text{ and } B \text{ and } C \text{ or not } (D \text{ or } E)$ сгенерировать код на командах перехода и изобразить атрибутивное дерево.
4. Пусть $G = (N, T, P, S)$ – LR(1)-грамматика; $w \in L(G)$, $|w| = n$.
Пусть k – число сдвигов, делаемых LR(1)-анализатором при анализе цепочки w . Привести нижнюю и верхнюю оценку для числа k .
5. Для оператора присваивания $a = b[c[j]]$ сгенерировать оптимальный код методом сопоставления образцов.

Вариант 8

1. Для регулярного выражения над алфавитом $T = \{a, b\}$:
 $(ab|b)^*ba|ab$
построить эквивалентный детерминированный конечный автомат.
2. Дана грамматика $G = (\{S, A\}, \{a, b\}, P, S)$ с правилами
 $S \rightarrow baaA | babA$
 $A \rightarrow e | Aa | Ab$
Построить LL(1)-грамматику, эквивалентную данной.
3. Для логического выражения
 $A \text{ and } (B \text{ or not } (C \text{ and } D) \text{ and } E)$
сгенерировать код на командах перехода и изобразить атрибутивное дерево.
4. Пусть $G = (N, T, P, S)$ – LR(1)-грамматика, $|P| = m \geq 1$; $w \in L(G)$, $|w| = n$.
Пусть k – число сверток, делаемых LR(1)-анализатором при анализе цепочки w . Привести нижнюю оценку для числа k .
5. Для оператора присваивания $a = b[i + j] + 2$ сгенерировать оптимальный код методом сопоставления образцов.

Вариант 7

1. Для регулярного выражения над алфавитом $T=\{a, b\}$: $(a|b)^* ba(a|b)$
построить эквивалентный детерминированный конечный автомат.
2. Дана грамматика $G=(\{S, A\}, \{a, b\}, P, S)$ с правилами
 $S \rightarrow ab \mid baA$
 $A \rightarrow e \mid Aab \mid Ab$
Построить LL(1)-грамматику, эквивалентную данной.
3. Для логического выражения $\text{not} (A \text{ and } B \text{ or } C \text{ or } D) \text{ and } E$ сгенерировать код на командах перехода и изобразить атрибутивное дерево.
4. Пусть $G=(N, T, P, S)$ – LR(1)-грамматика, $|P| = m \geq 1$; $\mathbf{w} \in L(G)$, $|\mathbf{w}|=n$.
Пусть k – число сверток, делаемых LR(1)-анализатором при анализе цепочки \mathbf{w} . Привести нижнюю оценку для числа k .
5. Для оператора присваивания $a = i + 2 + b[j]$ сгенерировать оптимальный код методом сопоставления образцов.

Вариант 6

1. Для регулярного выражения над алфавитом $T=\{a, b\}$:
 $(a|b)^* ab(a|b)^*$
построить эквивалентный детерминированный конечный автомат.
2. Дана грамматика $G=(\{S, A\}, \{a, b\}, P, S)$ с правилами
 $S \rightarrow abaA \mid abba$
 $A \rightarrow e \mid Aa \mid Ab$
Построить LL(1)-грамматику, эквивалентную данной.
3. Для логического выражения
 $A \text{ and } B \text{ or } C \text{ or } D \text{ and not } E$
сгенерировать код на командах перехода и изобразить атрибутивное дерево.
4. Существует ли LR(1)-грамматика, для которой функция действий LR(1)-таблицы не содержит элементов “ошибка”?
5. Для оператора присваивания $a = b[i + b[j]]$ сгенерировать оптимальный код методом сопоставления образцов.