

1. Построить для данной грамматики $LR(0)$ - и $LR(1)$ -автомат. Является ли данная грамматика $LR(0)$ -, $SLR(1)$ -, $LALR(1)$ -, $LR(1)$ -грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок), проанализировать какую-нибудь из цепочек данного языка.

$$S \rightarrow \text{if } E \text{ then } S \text{ else } S \mid a, E \rightarrow E \text{ or } b \mid b$$

По данной грамматике постройте атрибутивную грамматику, которая возвращает ссылку на тот узел дерева вывода цепочки, который будет исполняться.

2. Построить для данной грамматики $LR(0)$ - и $LR(1)$ -автомат. Является ли данная грамматика $LR(0)$ -, $SLR(1)$ -, $LALR(1)$ -, $LR(1)$ -грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок).

$$S \rightarrow S + A \mid A, A \rightarrow (S) \mid a(S) \mid a$$

По данной грамматике постройте атрибутивную грамматику, которая вычисляет значение выражения следующим образом: значения всех выражений в скобках возводятся в степень числа, стоящего перед этой скобкой (оно равно 1, если там нет ничего) и складывается.

Например, $2 + 3 (1+4(5+1))$ вычисляется как $2 + 1^3 + (5^4 + 1^4)^3$

3. Постройте грамматику (возможно неоднозначную) с одним нетерминалом, которая порождает язык формул логики высказываний (терминалами являются $x, (,), \neg, \wedge, \vee, \Leftrightarrow$). Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок) при помощи этой грамматики.

По данной грамматике постройте атрибутивную грамматику, которая считает сложность вычисления выражения, если известно, что стоимость вычисления дизъюнкции равна a , отрицания — b .

4. Построить для данной грамматики $LR(0)$ - и $LR(1)$ -автомат. Является ли данная грамматика $LR(0)$ -, $SLR(1)$ -, $LALR(1)$ -, $LR(1)$ -грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок).

$$S \rightarrow +A + \mid + B - \mid - A - \mid - B +, A \rightarrow x \mid + A + \mid + B -, B \rightarrow x \mid - A - \mid - B +$$

По данной грамматике постройте атрибутивную грамматику, которая считает количество минусов и плюсов в выражении. Можно ли совмещать семантический и синтаксический анализ в такой грамматике?

5. Построить для данной грамматики $LR(0)$ - и $LR(1)$ -автомат. Является ли данная грамматика $LR(0)$ -, $SLR(1)$ -, $LALR(1)$ -, $LR(1)$ -грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок), проанализировать какую-нибудь из цепочек данного языка.

$$S \rightarrow AB, A \rightarrow 1A0 \mid \lambda, B \rightarrow 0B \mid 0$$

По данной грамматике постройте атрибутивную, которая вычисляет значение данного двузначного числа в десятичном виде.

6. Построить для данной грамматики $LR(0)$ - и $LR(1)$ -автомат. Является ли данная грамматика $LR(0)$ -, $SLR(1)$ -, $LALR(1)$ -, $LR(1)$ -грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок).

$$S \rightarrow SaSb|Sa|c$$

По данной грамматике постройте атрибутивную грамматику, которая возвращает ссылку на первое вхождение буквы b (пустую, если буквы b нет).

7. Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок) для языка всевозможных правильных расстановок скобок трех видов: $()$, $[\]$, $\{ \}$.

По данной грамматике построить атрибутивную, которая вычисляет наибольшую глубину вложения скобок $()$. Например, в выражении $(\{()\})()$ она равна 2, а если выражение не содержит нужных скобок, то 0.

8. Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок) для языка всевозможных арифметических выражений в постфиксной форме записи (терминалами являются x , $($, $)$, $+$, $*$, $-$ (унарный минус)). Пример: $x(xx - +) + x*$.

Написать атрибутивную грамматику, которая вычисляет число отрицательных чисел в выражении (считаем, что x всегда положительное число). Например, в выражении $x - -x - +x +$ одно отрицательное число (значение выражения нас не интересует).

9. Построить для данной грамматики $LR(0)$ - и $LR(1)$ -автомат. Является ли данная грамматика $LR(0)$ -, $SLR(1)$ -, $LALR(1)$ -, $LR(1)$ -грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок).

$$S \rightarrow aA|bB, A \rightarrow 1A0|a, B \rightarrow 1B0|b$$

По данной грамматике построить атрибутивную, которая вычисляет значение данного числа в системе счисления с основанием, равным числу $x + 1$, где x — число, стоящее в начале строки.

10. Построить грамматику (возможно неоднозначную) с одним нетерминалом, которая порождает язык арифметических выражений (терминалами являются x , $($, $)$, знак деления $/$, возведение в степень \ominus , а также операция $*$, определенная по правилу $a * b = \min(a, b)$. Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок) при помощи этой грамматики.

По данной грамматике построить атрибутивную, которая вычисляет количество операций деления вне скобок. Например, в выражении $x/x/(x/x)$ таких операций две.

11. Построить для данной грамматики $LR(0)$ - и $LR(1)$ -автомат. Является ли данная грамматика $LR(0)$ -, $SLR(1)$ -, $LALR(1)$ -, $LR(1)$ -грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок), проанализировать какую-нибудь из цепочек данного языка.

$$D \rightarrow TL, T \rightarrow int|real, L \rightarrow L; a|a$$

По данной грамматике построить атрибутивную грамматику, которая для каждой объявленной переменной возвращает ссылку на первую ячейку выделенной для нее памяти (считаем, что память выделяется последовательно, номер первой ячейки - 0).

12. Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок) для языка всевозможных регулярных выражений в постфиксной форме записи (терминалами являются $x, (,), \cup, \times, *$, где \times заменяет значок \cdot). Пример: $xx^* \times x^* \cup$.

Постройте атрибутивную грамматику, которая вычисляет звездную высоту языка (максимальное вложенное количество итераций Клини)

13. Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок) для языка всевозможных арифметических выражений в префиксной форме записи (терминалами являются $x, (,), +, *, -$ (унарный минус)). Пример: $* + x(+ - xx)x$.

По данной грамматике постройте атрибутивную грамматику, которая строит дерево вывода данной цепочки.

14. Построить для данной грамматики LR(0)- и LR(1)-автомат. Является ли данная грамматика LR(0)-, SLR(1)-, LALR(1)-, LR(1)-грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок), проанализировать какую-нибудь из цепочек данного языка.

$$S \rightarrow \text{if } E \text{ then } S \text{ else } S \mid a, E \rightarrow E \text{ or } b \mid b$$

По данной грамматике постройте атрибутивную грамматику, которая возвращает ссылку на тот узел дерева вывода цепочки, который будет исполняться.

15. Построить для данной грамматики LR(0)- и LR(1)-автомат. Является ли данная грамматика LR(0)-, SLR(1)-, LALR(1)-, LR(1)-грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок).

$$S \rightarrow S + A \mid A, A \rightarrow (S) \mid a(S) \mid a$$

По данной грамматике постройте атрибутивную грамматику, которая вычисляет значение выражения следующим образом: значения всех выражений в скобках делятся на число, стоящее перед этой скобкой (оно равно 1, если там нет ничего) и складывается.

Например, $2 + 3 (1+4(5+1))$ вычисляется как $2 + 1/3 + (5/4 + 1/4)/3$

16. Построить для данной грамматики LR(0)- и LR(1)-автомат. Является ли данная грамматика LR(0)-, SLR(1)-, LALR(1)-, LR(1)-грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок), проанализировать какую-нибудь из цепочек данного языка.

$$S \rightarrow AB, A \rightarrow 1A0 \mid \lambda, B \rightarrow 0B \mid 0$$

По данной грамматике постройте атрибутивную, которая вычисляет значение данного двузначного числа в десятичном виде.

17. Построить для данной грамматики LR(0)- и LR(1)-автомат. Является ли данная грамматика LR(0)-, SLR(1)-, LALR(1)-, LR(1)-грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок), проанализировать какую-нибудь из цепочек данного языка.

$$D \rightarrow TL, T \rightarrow \text{int} \mid \text{real}, L \rightarrow L; a \mid a$$

По данной грамматике построить атрибутивную грамматику, которая для каждой объявленной переменной возвращает ссылку на первую ячейку выделенной для нее памяти (считаем, что память выделяется последовательно, номер первой ячейки - 0).

18. Построить для данной грамматики $LR(0)$ - и $LR(1)$ -автомат. Является ли данная грамматика $LR(0)$ -, $SLR(1)$ -, $LALR(1)$ -, $LR(1)$ -грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок).

$$S \rightarrow aA|bB, A \rightarrow 1A0|a, B \rightarrow 1B0|b$$

По данной грамматике построить атрибутивную, которая вычисляет значение данного числа в системе счисления с основанием, равным числу $x + 1$, где x — число, стоящее в начале строки.

19. Построить для данной грамматики $LR(0)$ - и $LR(1)$ -автомат. Является ли данная грамматика $LR(0)$ -, $SLR(1)$ -, $LALR(1)$ -, $LR(1)$ -грамматикой? Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок).

$$S \rightarrow SaSb|Sa|c$$

По данной грамматике постройте атрибутивную грамматику, которая возвращает ссылку на последнее вхождение буквы a (пустую, если буквы a нет).

20. Построить грамматику (возможно неоднозначную) с одним нетерминалом, которая порождает язык арифметических выражений (терминалами являются $x, (,)$, знак деления $/$, возведение в степень \ominus , а также операция $*$, определенная по правилу $a * b = \min(a, b)$. Построить LR-распознаватель с обработчиком ошибок (построить таблицы ACTION и GOTO, выписать процедуры обработки ошибок) при помощи этой грамматики.

По данной грамматике построить атрибутивную, которая вычисляет количество операций $*$ внутри скобок. Например, в выражении $x \ominus x \ominus (x \ominus x)$ такая операция одна.