**Группа ИУ7-29**

|  |  |
| --- | --- |
| **Вариант** | **ФИО** |
| 1 | Волынец Максим Юрьевич |
| 2 | Дерягин Дмитрий Андреевич |
| 3 | Ивлиев Сергей Андреевич |
| 4 | Киселев Ярослав Владимирович |
| 5 | Кладов Станислав Александрович |
| 6 | Кузьмин Артем Юрьевич |
| 7 | Кулаков Дмитрий Леонидович |
| 8 | Малышева Дарья Владиславович |
| 9 | Меньшов Никита Дмитриевич |
| 10 | Могилко Александр Александрович |
| 11 | Орлова Анастасия Андреевна |
| 12 | Стройкова Ксения Александровна |
| 13 | Фроловская Елена Александровна |

**ВАРИАНТ 1**

Показать, что грамматика регулярных выражений G = ({R}, {'|', '\*', '(', ')', **a**, **b**}, P, R) с правилами

P = {R  R '|' R | R R | R '\*' | '(' R ')' | **a** | **b**} неоднозначна. Используя новые нетерминалы, устранить неоднозначность и явно определить приоритет операций. Построить LL(1)-анализатор для преобразованной грамматики и выполнить разбор цепочки w="(**a** | **b**)\***abb**".

**ВАРИАНТ 2**

Для грамматики G = ({S, A, B}, {**if**, **then**, **else**, **true**, **false**, **id**, '**;**', '='}, P, S) с правилами P = {S  **if** B **then** S |

**if** B **then** S **else** S | A;, A  A=**id** | **id**, B  **true** | **false**} построить SLR(1)-анализатор и выполнить разбор цепочки w = "**if true then id=id; else id;**". Конфликт типа «перенос/свертка» разрешать в пользу переноса.

**ВАРИАНТ 3**

Показать, что грамматика G = ({S, L}, {**a**, '(', ')', ','}, P, S) с правилами P = {S  (L) | **a**, L  L, S | S} является грамматикой операторного предшествования, построить для нее анализатор операторного предшествования и выполнить разбор цепочки w = "(**a**, ((**a**, **a**), (**a**, **a**)))".

**ВАРИАНТ 4**

Показать, что грамматика G=({S, A}, {**a**, '(', ')', ','}, P, S) с правилами P={S  ( ) | **a** | ( A ), A  S | A, S} не является LL(1)-грамматикой. Найти эквивалентную ей не леворекурсивную и левофакторизованную LL(1)-грамматику G', построить LL(1)-анализатор для G' и выполнить разбор цепочки w="(( ),**a**,((**a**)))".

**ВАРИАНТ 5**

Для грамматики G=({S, E, V}, {**id**, '=', '\*'}, P, S} с правилами P={S V=E, S  E, E  V, V  **id**, V \*E} построить LR(1)-анализатор и выполнить разбор цепочки w = "\*\***id**=\*\***id**".

**ВАРИАНТ 6**

Показать, что грамматика G = ({B}, {**true**, **false**, **or**, **and**, **not**, '(', ')'}, P, B) с правилами P = {B  B **or** B | B **and** B | **not** B | (B) | **true** | **false**} является грамматикой операторного предшествования, построить для нее анализатор операторного предшествования и выполнить разбор цепочки w = "**not**(**true** **or** **false**)".

**ВАРИАНТ 7**

Показать, что грамматика G = ({E}, {"!", "&&", "(", ")", **id**}, P, R) с правилами P = {E  "!" E, E E "&&" E, E  **id,**

E "(" E ")"} неоднозначна. Используя новые нетерминалы, устранить неоднозначность и явно определить приоритет операций. Построить LL(1)-анализатор для преобразованной грамматики и выполнить разбор цепочки w="!(!**id** && **id**)".

**ВАРИАНТ 8**

Для грамматики G = ({S, A, B, C}, {**a, b, c**}, P, S) с правилами P = {S  ABC, A  **b**A | B | **a**, B  B**a** | **b**C | **ba**, C  B | **cba** | **ca** | **c**} построить каноническую систему множеств LR(0)-ситуаций и соответствующий ей граф goto и ответить на вопрос, какие цепочки являются активными префиксами правых сентенциальных форм грамматики G: A**b**C, **bba**C, **ab**C**ca**, AB**bbbbb**, **bbb**A**b**?

**ВАРИАНТ 9**

Показать, что грамматика G = ({S, L}, {**if**, **then**, **begin**, **end**, **a**, **b**, ';'}, P, S) с правилами P = {S  **if b then** S | **begin** L **end** | **a,** L  L**;** S | S} является грамматикой операторного предшествования, построить для нее анализатор операторного предшествования и выполнить разбор цепочки w = "**begin if b then a; if b then a end"**.

**ВАРИАНТ 10**

Показать, что грамматика G = ({Z, E, F, V, W}, {**i, m, o, c**}, P, Z) с правым концевым маркером $ и правилами P = {E  **m**E, E  **o**E**c**, E  VF, F  **m**E, F  , V  **i**W, W  **o**E**c**, W  , Z  E$} является LL(1)-грамматикой, построить для нее LL(1)-анализатор и выполнить разбор цепочки w = "**immiooicc**$".

**ВАРИАНТ 11**

Для грамматики G = ({S, A, B}, {**a**, **b, x**}, P, S) с правилами P = {S  A | **xb**, A  **a**A**b** | B, B  **x}** построить SLR(1)-анализатор и выполнить разбор цепочки w = "**aaaxbbb**". Если SLR(1)-таблица будет содержать конфликт типа «перенос/свертка», то при разборе конфликт разрешать в пользу переноса символа.

**ВАРИАНТ 12**

Показать, что грамматика G=({S, E, B, L}, {**id, num, print, while**, '(', ')', '{', '}', '>', ';'}, P, S) с правилами P = {S  **print**(E);|**while**(B)S|{L}, E  **id**|**num**, B  E>E, L  SL|} является LL(1)-грамматикой, построить для нее LL(1)-анализатор и выполнить разбор цепочки w="{**while**(**id**> **num**)**print**(**id**);**print**(**num**);}".

**ВАРИАНТ 13**

Для грамматики G = ({S, A, B}, {**a**, **b, x**}, P, S) с правилами P = {S  A | **xb**, A  **a**A**b** | B, B  **x}** построить LR(1)-анализатор и выполнить разбор цепочки w = "**aaaxbbb**".