Вариант 3\_2014 Ф.И.О.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_№ группы\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1.** Дана грамматика *G*:  *S* → *Sc|aSc|ε|aAc*  *aAc* → *aEAc|aac*  *aE* → *Ea*  *E* → *a* | (a) Описать язык *L*(*G*) в виде теоретико-множественной формулы:  **Ответ:**  *L*(*G*) = | (б) Каким из перечисленных классов грамматик принадлежит *G*? **Ответ:**   |  |  | | --- | --- | | Класс П | *G* ∈ П? (да/нет) | | регулярные |  | | контекстно-зависимые |  | | контекстно-свободные |  | | грамматики типа 0 |  | | неукорачивающие |  | |

(в) Тип языка: найти такое целое *k*, что *L(G)* является языком типа *k*, но не языком типа *k+*1.

**Ответ:** *k* =

**2.** Является ли однозначной данная грамматика *G*, порождающая язык цепочек в алфавите {*b*,*c*}? Ответ обосновать.

*G: S* → *ASS |SSA| B | A*

*B* → *bB|cB| ε*

|  |  |  |
| --- | --- | --- |
| **3.** | Колобок должен пройти через лесной лабиринт из левого верхнего угла в правый нижний, чтобы спеть песенку лисе. Он умеет за один шаг перемещаться на клетку вниз (след обозначается символом *a*), вверх ( след *b*) или вправо (след *с*). Влево не умеет. Каждая клетка посещается не более одного раза. Постройте грамматику, порождающую язык в алфавите {*a*, *b, с*} всевозможных путей колобка, приводящих к лисе. Например, цепочка *ссaaaacacc* приведет колобка к цели. В грамматике должно быть не более 4-х правил вывода, считая альтернативы. |  |

**4.** Дан список слов: *на*, *правильность*, *проверка*, *программа*, *помощь*, *с, тесты*. Составьте из него, употребив слова в нужном порядке и форме, определение термина (понятия) из раздела СП и назовите сам термин (он не входит в заданный список).

**5.** Можно ли язык, на котором записывается задание для утилиты *make* системы *Unix,*  считать интерпретируемым языком? Обоснуйте ответ.

**6.** По заданной грамматике *G =* {{*d, b*}*,* {*B, S*}*, P, S*} получить эквивалентную неукорачивающую контекстно-свободную грамматику (использовать алгоритм устранения правил с пустой правой частью).

*P: S → dBb | S | ε* **Ответ:**

*B → dB | bS | SSd*

**7.** Даны 1) функция-анализатор на языке Си++ для леволинейной грамматики *GL*:

**bool** scan\_G ()

{ **enum** state { H, С, S, ER }; // множество состояний

state CS= H; // CS —— текущее состояние

**do**  { gc (); // считывает символ в глобальный объект с

**switch** (CS) {

**case** H: **if** ( c == 'a' ) CS = С;

**else** CS = ER; **break**;

**case** С: **if** ( c == 'с' );

**else** **if** ( c == '⊥' ) CS = S;

**else** CS = ER; **break**;

}

}

**while** ( CS != S && CS != ER);

**return** CS == S; // true, если CS != ER, иначе false

}

2) заготовка диаграммы состояний праволинейной грамматики *GR:*

|  |  |
| --- | --- |
| *⊥*  *⊥*  *⊥*  *B*  *F*  *D*  *A*  *S* | (а) Расставить в заготовке диаграммы для *GR*терминальные символы так, чтобы грамматики *GR* и *GL* были эквивалентны.  (б) Восстановить грамматику *GR.* **Ответ:**  (в) Сколько состояний будет иметь конечный автомат, полученный алгоритмом детерминизации диаграммы для *GR* ? **Ответ:\_\_\_\_** |

|  |  |
| --- | --- |
| ­­­**8.** Дана КС-грамматика *G*, порождающая язык *L.*  Вставить в грамматику действия вида **〈***cout <<* ′′символы′′ ;**〉** так, чтобы в процессе рекурсивного спуска был реализован перевод *τ* = {(ω, *a* 2k*b* k-n)| ω∈ *L* , *k* = |ω|, *n* = |ω|*a* }. | *G*:  *S* → *aA |bB |* ε  *A* → *сA | S*  *B* → *сB| S* |

**9.** Дана грамматика *G*. Докажите, что метод рекурсивного спуска непримени́м к ней. Можно ли вычеркнуть один терминальный символ в правилах грамматики так, чтобы к получившейся грамматике метод был бы примени́м?

*G: S* → *aSb | YD | db*

*Y* → *cSY | bd | ε*

*D* → *dd*

**10.** Постройте ПОЛИЗ для фрагмента программы на Си. Префиксный ++ в польской записи обозначается как +#, постфиксный как #+.

i = 0, j = 10; **while** ( i == j ? 0 : 1 ) a += 2 < ++i > j--;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ПОЛИЗ | | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | 11 | | 12 | | 13 | | 14 | | 15 | | 16 | | 17 | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| 18 | | 19 | 20 | | 21 | | 22 | | 23 | | 24 | | 25 | | 26 | | 27 | | 28 | | 29 | | 30 | | 31 | | 32 | | 33 | | 34 | | 35 | | 36 | | 37 | | 38 | 39 |
|  | |  |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  |  |