

**Задачи**  
(Приложение к учебному пособию  
«Формальные грамматики и языки. Элементы теории трансляции»)

**I. Порождающие грамматики. Языки, порождаемые грамматиками.  
Классификация грамматик и языков по Хомскому**

1. Дана грамматика. Построить вывод заданной цепочки.

- |  |  |
|--|--|
| <p>a) <math>S \rightarrow T \mid T+S \mid T-S</math><br/> <math>T \rightarrow F \mid F*T</math><br/> <math>F \rightarrow a \mid b</math><br/>         Цепочка <math>a-b*a+b</math></p> | <p>b) <math>S \rightarrow aSBC \mid abC</math><br/> <math>CB \rightarrow BC</math><br/> <math>bB \rightarrow bb</math><br/> <math>bC \rightarrow bc</math><br/> <math>cC \rightarrow cc</math><br/>         Цепочка <math>aaabbbccc</math></p> |
|--|--|

2. Построить все сентенциальные формы для грамматики с правилами:

$$S \rightarrow A+B \mid B+A$$

$$A \rightarrow a$$

$$B \rightarrow b$$

3. К какому типу по Хомскому относится данная грамматика? Какой язык она порождает? Каков тип языка? Указать максимально возможный номер типа грамматики и языка.

- |  |  |
|--|--|
| <p>a) <math>S \rightarrow APA</math><br/> <math>P \rightarrow + \mid -</math><br/> <math>A \rightarrow a \mid b</math></p>   | <p>b) <math>S \rightarrow A \mid SA \mid SB</math><br/> <math>A \rightarrow a</math><br/> <math>B \rightarrow b</math></p>                             |
| <p>c) <math>S \rightarrow 1B</math><br/> <math>B \rightarrow B0 \mid 1</math></p>  | <p>d) <math>S \rightarrow aQb \mid \varepsilon</math><br/> <math>Q \rightarrow cSc</math></p>  |
| <p>e) <math>S \rightarrow a \mid Ba</math><br/> <math>B \rightarrow Bb \mid b</math></p>   | <p>f) <math>S \rightarrow Ab</math><br/> <math>A \rightarrow Aa \mid ba</math></p>   |
| <p>g) <math>S \rightarrow 0A1 \mid 01</math><br/> <math>0A \rightarrow 00A1</math><br/> <math>A \rightarrow 01</math></p>  | <p>h) <math>S \rightarrow AB</math><br/> <math>AB \rightarrow BA</math><br/> <math>A \rightarrow a</math><br/> <math>B \rightarrow b</math></p>        |
| <p>i) <math>S \rightarrow A \mid B</math><br/> <math>A \rightarrow aAb \mid 0</math><br/> <math>B \rightarrow aBbb \mid 1</math></p>   | <p>j) <math>S \rightarrow 0A \mid 1S</math><br/> <math>A \rightarrow 0A \mid 1B</math><br/> <math>B \rightarrow 0B \mid 1B \mid \perp</math></p>       |
| <p>k) <math>S \rightarrow 0S \mid S0 \mid D</math><br/> <math>D \rightarrow DD \mid 1A \mid \varepsilon</math><br/> <math>A \rightarrow 0B \mid \varepsilon</math><br/> <math>B \rightarrow 0A \mid 0</math></p> | <p>l) <math>S \rightarrow 0A \mid 1S \mid \varepsilon</math><br/> <math>A \rightarrow 1A \mid 0B</math><br/> <math>B \rightarrow 0S \mid 1B</math></p> |

$$\begin{aligned} \text{m) } S &\rightarrow SS \mid A \\ A &\rightarrow a \mid bb \end{aligned}$$

$$\begin{aligned} \text{n) } S &\rightarrow AB \perp \\ A &\rightarrow a \mid cA \\ B &\rightarrow bA \end{aligned}$$

$$\begin{aligned} \text{o) } S &\rightarrow aBA \mid \varepsilon \\ B &\rightarrow bSA \\ AA &\rightarrow c \end{aligned}$$

$$\begin{aligned} \text{p) } S &\rightarrow Ab \mid c \\ A &\rightarrow Ba \\ B &\rightarrow cS \end{aligned}$$

$$\begin{aligned} \text{r) } 1. S &\rightarrow KF \\ 2. K &\rightarrow KB \mid CB \\ 3. C &\rightarrow CA \mid DA \\ 4. DA &\rightarrow aAD \\ 5. Aa &\rightarrow aA \\ 6. DB &\rightarrow bBD \end{aligned}$$

$$\begin{aligned} 7. Bb &\rightarrow bB \\ 8. Ab &\rightarrow bA \\ 9. DF &\rightarrow E \\ 10. BE &\rightarrow Eb \\ 11. AE &\rightarrow Ea \\ 12. bE &\rightarrow b \end{aligned}$$

$$\begin{aligned} \text{s) } 1. S &\rightarrow DC \\ 2. D &\rightarrow aDA \mid bDB \mid aA \mid bB \\ 3. AC &\rightarrow aC \\ 4. BC &\rightarrow bC \\ 5. Aa &\rightarrow aA \end{aligned}$$

$$\begin{aligned} 6. Ba &\rightarrow aB \\ 7. Ab &\rightarrow bA \\ 8. Bb &\rightarrow bB \\ 9. C &\rightarrow \varepsilon \end{aligned}$$

$$\begin{aligned} \text{t) } S &\rightarrow aAc \\ aA &\rightarrow aaBbC \mid ab \\ Bb &\rightarrow bb \mid abbbc \mid aDbbbcc \\ C &\rightarrow c \\ D &\rightarrow ab \end{aligned}$$

$$\begin{aligned} \text{u) } S &\rightarrow 0A1 \\ 0A &\rightarrow 0B1 \mid 0 \\ B1 &\rightarrow 0C11 \mid 01 \\ C &\rightarrow 0D \mid 00D1 \mid 0 \\ D &\rightarrow 01 \end{aligned}$$

4. Построить грамматику, порождающую язык :

$$\text{a) } L = \{ a^n b^m \mid n, m \geq 1 \}$$

$$\text{b) } L = \{ \alpha c \beta \gamma c \mid \alpha, \beta, \gamma \text{ — любые цепочки из } a \text{ и } b \}$$

$$\text{c) } L = \{ a_1 a_2 \dots a_n a_n \dots a_2 a_1 \mid a_i = 0 \text{ или } 1, n \geq 1 \}$$

$$\text{d) } L = \{ 0^n 1^{\lfloor n/2 \rfloor}, n \geq 1 \}$$

$$\text{e) } L = \{ ca^n cb^m c^n, n \geq 0, m \geq 0 \}$$

$$\text{f) } L = \{ a^n b^m \mid n \neq m; n, m \geq 0 \}$$

$$\text{g) } L = \{ \text{цепочки из } 0 \text{ и } 1 \text{ с неравным числом } 0 \text{ и } 1 \}$$

$$\text{h) } L = \{ \alpha \alpha \mid \alpha \in \{a, b\}^+ \}$$

i)  $L = \{ \omega \mid \omega \in \{0, 1\}^+ \text{ и содержит равное количество } 0 \text{ и } 1, \text{ причем любая подцепочка, взятая с левого конца, содержит единиц не меньше, чем нулей} \}$ .

$$\text{j) } L = \{ (a^{2^m} b^m)^n \mid m \geq 1, n \geq 0 \}$$

$$\text{k) } L = \{ a^{3^n+1} \perp \mid n \geq 1 \}$$

$$\text{l) } L = \{ a^{n^2} \mid n \geq 1 \}$$

$$m) L = \{ a^{n^3+1} \mid n \geq 1 \}$$

Каков тип этой грамматики? Каков тип языка?

5. К какому типу по Хомскому относится данная грамматика (указать максимально возможный номер)? Какой язык она порождает? Каков тип языка? Выписать грамматику, в состав которой входит только один нетерминал – цель грамматики, подтверждающую ответ.

$$\begin{aligned} a) \quad S &\rightarrow AB \mid ASB \\ A &\rightarrow a \\ aB &\rightarrow b \\ bB &\rightarrow bb \end{aligned}$$

$$\begin{aligned} b) \quad S &\rightarrow 1A0 \\ 1A &\rightarrow 11A0 \mid 01 \end{aligned}$$

6. Эквивалентны ли грамматики с правилами :

$$\begin{aligned} a) \quad S &\rightarrow AB \\ A &\rightarrow a \mid Aa \\ B &\rightarrow b \mid Bb \end{aligned}$$

и

$$\begin{aligned} S &\rightarrow AS \mid SB \mid AB \\ A &\rightarrow a \\ B &\rightarrow b \end{aligned}$$

$$\begin{aligned} b) \quad S &\rightarrow aSL \mid aL \\ L &\rightarrow Kc \\ cK &\rightarrow Kc \\ K &\rightarrow b \end{aligned}$$

и

$$\begin{aligned} S &\rightarrow aSBc \mid abc \\ cB &\rightarrow Bc \\ bB &\rightarrow bb \end{aligned}$$

7. Построить КС-грамматику, эквивалентную грамматике с правилами:

$$\begin{aligned} a) \quad S &\rightarrow aAb \\ aA &\rightarrow aaAb \\ A &\rightarrow \varepsilon \end{aligned}$$

$$\begin{aligned} b) \quad S &\rightarrow AB \mid ABS \\ AB &\rightarrow BA \\ BA &\rightarrow AB \\ A &\rightarrow a \\ B &\rightarrow b \end{aligned}$$

8. Построить регулярную грамматику, эквивалентную грамматике с правилами:

$$\begin{aligned} a) \quad S &\rightarrow A \mid AS \\ A &\rightarrow a \mid bb \end{aligned}$$

$$\begin{aligned} b) \quad S &\rightarrow A.A \\ A &\rightarrow B \mid BA \\ B &\rightarrow 0 \mid 1 \end{aligned}$$

9. Привести пример грамматики, все правила которой имеют вид

$$A \rightarrow Bt, \quad \text{либо} \quad A \rightarrow tB, \quad \text{либо} \quad A \rightarrow t,$$

для которой не существует эквивалентной регулярной грамматики.

10. Доказать, что грамматика с правилами:

$$\begin{aligned} S &\rightarrow aSBC \mid abC \\ CB &\rightarrow BC \\ bB &\rightarrow bb \\ bC &\rightarrow bc \\ cC &\rightarrow cc \end{aligned}$$

порождает язык  $L = \{ a^n b^n c^n \mid n \geq 1 \}$ .

Для этого показать, что в данной грамматике

1. выводится любая цепочка вида  $a^n b^n c^n$  ( $n \geq 1$ ) и
2. не выводятся никакие другие цепочки.

11. Дана грамматика с правилами:

$$\begin{aligned} \text{a) } S &\rightarrow S0 \mid S1 \mid D0 \mid D1 \\ D &\rightarrow H. \\ H &\rightarrow 0 \mid 1 \mid H0 \mid H1 \end{aligned}$$

$$\begin{aligned} \text{b) } S &\rightarrow \text{if } B \text{ then } S \mid B = E \\ E &\rightarrow B \mid B + E \\ B &\rightarrow a \mid b \end{aligned}$$

Построить восходящим и нисходящим методами дерево вывода для цепочки:

a) 10.1001

b) if a then b = a+b+b

12. Определить тип грамматики. Описать язык, порождаемый этой грамматикой. Написать для этого языка КС-грамматику.

$$\begin{aligned} S &\rightarrow P\perp \\ P &\rightarrow 1P1 \mid 0P0 \mid T \\ T &\rightarrow 021 \mid 120R \\ R1 &\rightarrow 0R \\ R0 &\rightarrow 1 \\ R\perp &\rightarrow 1\perp \end{aligned}$$

13. Построить регулярную грамматику, порождающую цепочки в алфавите  $\{a, b\}$ , в которых символ  $a$  не встречается два раза подряд.

14. Написать КС-грамматику для языка  $L$ , построить дерево вывода и левосторонний вывод для цепочки  $aabbbsccc$ .

$$L = \{a^{2n} b^m c^{2k} \mid m=n+k, m>1\}.$$

15. Построить грамматику, порождающую сбалансированные относительно круглых скобок цепочки в алфавите  $\{a, (, ), \perp\}$ . Сбалансированную цепочку  $\alpha$  определим рекуррентно: цепочка  $\alpha$  сбалансирована, если

- a)  $\alpha$  не содержит скобок,
- b)  $\alpha = (\alpha_1)$  или  $\alpha = \alpha_1 \alpha_2$ , где  $\alpha_1$  и  $\alpha_2$  сбалансированы.

16. Написать КС-грамматику, порождающую язык  $L$ , и вывод для цепочки  $aasbbbbsaa$  в этой грамматике.

$$L = \{a^n cb^m ca^n \mid n, m>0\}.$$

17. Написать КС-грамматику, порождающую язык  $L$ , и вывод для цепочки  $110000111$  в этой грамматике.

$$L = \{1^n 0^m 1^p \mid n+p>m; n, p, m>0\}.$$

18. Дан язык  $L = \{1^{3n+2} 0^n \mid n \geq 0\}$ . Определить его тип, написать грамматику, порождающую  $L$ . Построить левосторонний и правосторонний выводы, дерево разбора для цепочки 1111111100.

19. Написать общие алгоритмы построения по данным КС-грамматикам  $G_1$  и  $G_2$ , порождающим языки  $L_1$  и  $L_2$ , КС-грамматики для

- a)  $L_1 \cup L_2$
- b)  $L_1 * L_2$
- c)  $L_1^*$

**Замечание:**  $L = L_1 * L_2$  — это конкатенация языков  $L_1$  и  $L_2$ , т.е.  $L = \{ \alpha\beta \mid \alpha \in L_1, \beta \in L_2 \}$ ;  $L = L_1^*$  — это итерация языка  $L_1$ , т.е. объединение  $\{\varepsilon\} \cup L_1 \cup L_1 * L_1 \cup L_1 * L_1 * L_1 \cup \dots$

20. Написать КС-грамматику для  $L = \{\omega_i 2 \omega_{i+1}^R \mid i \in \mathbb{N}, \omega_i = (i)_2\}$  — двоичное представление числа  $i$ ,  $\omega^R$  — обращение цепочки  $\omega$ . Написать КС-грамматику для языка  $L^*$  (см. задачу 19 раздела I).

21. Показать, что грамматика

$$E \rightarrow E+E \mid E * E \mid (E) \mid i$$

неоднозначна. Как описать этот же язык с помощью однозначной грамматики?

22. Показать, что наличие в КС-грамматике правил вида

- a)  $A \rightarrow AA \mid \alpha$
- b)  $A \rightarrow A\alpha A \mid \beta$
- c)  $A \rightarrow \alpha A \mid A\beta \mid \gamma$

где  $\alpha, \beta, \gamma \in (VT \cup VN)^*$ ,  $A \in VN$ , делает ее неоднозначной. Можно ли преобразовать эти правила таким образом, чтобы полученная эквивалентная грамматика была однозначной?

23. Показать, что грамматика  $G$  неоднозначна. Какой язык она порождает? Является ли этот язык однозначным?

$$\begin{aligned} G: S &\rightarrow aAc \mid aB \\ B &\rightarrow bc \\ A &\rightarrow b \end{aligned}$$

24. Дана КС-грамматика  $G = \{VT, VN, P, S\}$ . Предложить алгоритм построения множества  $X = \{A \in VN \mid A \Rightarrow \varepsilon\}$ .

25. Для произвольной КС-грамматики  $G$  предложить алгоритм, определяющий, пуст ли язык  $L(G)$ .

26. Написать приведенную грамматику, эквивалентную данной.

- |                                       |                                      |
|---------------------------------------|--------------------------------------|
| a) $S \rightarrow aABS \mid bCACd$    | b) $S \rightarrow aAB \mid E$        |
| $A \rightarrow bAB \mid cSA \mid cCC$ | $A \rightarrow dDA \mid \varepsilon$ |
| $B \rightarrow bAB \mid cSB$          | $B \rightarrow bE \mid f$            |

$$C \rightarrow cS \mid c$$

$$C \rightarrow cAB \mid dSD \mid a$$

$$D \rightarrow eA$$

$$E \rightarrow fA \mid g$$

27. Язык называется распознаваемым, если существует алгоритм, который за конечное число шагов позволяет получить ответ о принадлежности любой цепочки языку. Если число шагов зависит от длины цепочки и может быть оценено до выполнения алгоритма, язык называется легко распознаваемым. Доказать, что язык, порождаемый неукорачивающей грамматикой, легко распознаваем.

28. Доказать, что любой конечный язык, в который не входит пустая цепочка, является регулярным языком.

29. Доказать, что нециклическая КС-грамматика порождает конечный язык.

**Замечание:** Нетерминальный символ  $A \in VN$  — циклический, если в грамматике существует вывод  $A \Rightarrow \xi_1 A \xi_2$ . КС-грамматика называется циклической, если в ней имеется хотя бы один циклический символ.

30. Показать, что условие цикличности грамматики (см. задачу 29) не является достаточным условием бесконечности порождаемого ею языка.

31. Доказать, что язык, порождаемый циклической приведенной КС-грамматикой, содержащей хотя бы один эффективный циклический символ, бесконечен.

**Замечание:** Циклический символ называется эффективным, если  $A \Rightarrow \alpha A \beta$ , где  $|\alpha A \beta| > 1$ ; иначе циклический символ называется фиктивным.

## II. Регулярные грамматики, ДС, анализаторы по ДС. Преобразование НКА к ДКА

1. Дана регулярная грамматика с правилами:

$$S \rightarrow S0 \mid S1 \mid P0 \mid P1$$

$$P \rightarrow N.$$

$$N \rightarrow 0 \mid 1 \mid N0 \mid N1.$$

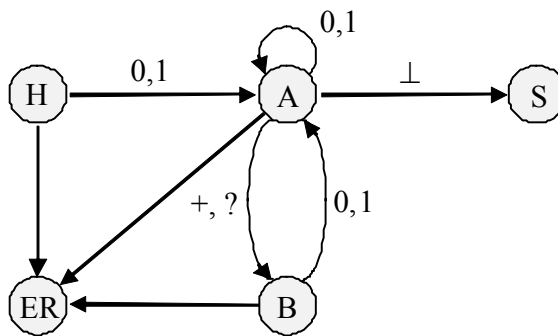
Построить по ней диаграмму состояний и использовать ДС для разбора цепочек: 11.010, 0.1, 01., 100. Какой язык порождает эта грамматика?

2. Дана ДС.

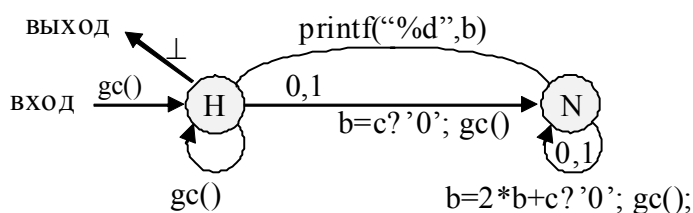
а) Осуществить разбор цепочек 1011 $\perp$ , 10+011 $\perp$  и 0–101+1 $\perp$ .

б) Восстановить регулярную грамматику, по которой была построена данная ДС.

с) Какой язык порождает полученная грамматика?



3. Пусть имеется переменная  $c$  и функция  $gc()$ , считывающая в  $c$  очередной символ анализируемой цепочки. Дана ДС с действиями:



- Определить, что будет выдано на печать при разборе цепочки  $1+101//p11+++1000/5⊥?$
- Написать на Си анализатор по этой ДС.

4. Написать регулярную левостороннюю грамматику для заданного языка, по ней построить ДС, а по ДС — программу анализатора.

- $L = \{ x \alpha y \perp \mid \alpha \in \{x, y\}^* \}$
- $L = \{ (x y^3)^n \perp \mid n \geq 1 \}$
- $L = \{ (abb)^k \perp \mid k \geq 1 \}$
- $L = \{ \omega \perp \mid \omega \in \{0,1\}^* \}$ , где за каждой 1 непосредственно следует 0
- $L = \{ 1\omega 1 \perp \mid \omega \in \{0,1\}^+ \}$ , где все подряд идущие 0 – подцепочки нечетной длины

5. Дана регулярная грамматика:

$$\begin{aligned} S &\rightarrow A \perp \\ A &\rightarrow Ab \mid Bb \mid b \\ B &\rightarrow Aa \end{aligned}$$

Определить язык, который она порождает; построить ДС; написать на Си анализатор.

6. Построить ДС, по которой в заданном тексте, оканчивающемся на  $\perp$ , выявляются все парные комбинации  $\langle \rangle$ ,  $\leq$  и  $\geq$  и подсчитывается их общее количество.

7. Написать на Си анализатор, выделяющий из текста вещественные числа без знака (они определены как в Паскале) и преобразующий их из символического представления в числовое.

8. Написать на Си анализатор, выделяющий из текста вещественные числа без знака (они определены как в Паскале) и преобразующий их из символического представления в числовое.

9. Даны две грамматики  $G_1$  и  $G_2$ .

$G_1$ :  $S \rightarrow 0C \mid 1B$   
 $B \rightarrow 0B \mid 1C \mid \varepsilon$   
 $C \rightarrow 0C \mid 1C$

$G_2$ :  $S \rightarrow 0D \mid 1B$   
 $B \rightarrow 0C \mid 1C$   
 $C \rightarrow 0D \mid 1D \mid \varepsilon$   
 $D \rightarrow 0D \mid 1D$

$L_1 = L(G_1)$ ;  $L_2 = L(G_2)$ .

Построить регулярную грамматику для:

- a)  $L_1 \cup L_2$
- b)  $L_1 \cap L_2$
- c)  $L_1^* \setminus \{\varepsilon\}$
- d)  $L_2^* \setminus \{\varepsilon\}$
- e)  $L_1 * L_2$

Если разбор по ней оказался недетерминированным, построить эквивалентную ей грамматику, допускающую детерминированный разбор.

10. Написать левостороннюю регулярную грамматику, эквивалентную данной правосторонней, допускающую детерминированный разбор.

a)  $S \rightarrow 0S \mid 0B$   
 $B \rightarrow 1B \mid 1C$   
 $C \rightarrow 1C \mid \perp$

b)  $S \rightarrow 0B$   
 $B \rightarrow 1C \mid 1S$   
 $C \rightarrow \perp$

c)  $S \rightarrow aB$   
 $B \rightarrow aC \mid aD \mid dB$   
 $C \rightarrow aB$   
 $D \rightarrow \perp$

11. Для данной грамматики

- a) определить ее тип;
- b) какой язык она порождает;
- c) написать Р-грамматику, почти эквивалентную данной;
- d) построить ДС и анализатор на Си.

$S \rightarrow 0S \mid S0 \mid D$   
 $D \rightarrow DD \mid 1A \mid \varepsilon$   
 $A \rightarrow 0B \mid \varepsilon$   
 $B \rightarrow 0A \mid 0$

12. Построить ДС, соответствующую заданной левосторонней регулярной грамматике. Если ДС задает НКА, то преобразовать НКА к ДКА, используя алгоритм преобразования. По получившемуся ДКА написать анализатор. Выписать соответствующую ДКА грамматику.

a)  $S \rightarrow Sa \mid Ab \mid b$   
 $A \rightarrow Ab \mid Sa \mid a$

b)  $S \rightarrow Sb \mid Aa \mid a$   
 $A \rightarrow Sb \mid a \mid b$

c)  $S \rightarrow C\perp$

d)  $S \rightarrow A\perp$



- |    |   |    |   |
|----|---|----|---|
|    | $C \rightarrow A1 \mid B1 \mid 1$<br>$A \rightarrow A1 \mid C0 \mid 0$<br>$B \rightarrow C0 \mid 0$                           |    | $A \rightarrow Bb \mid a$<br>$B \rightarrow Bb \mid b$  |
| e) | $S \rightarrow B0 \mid C0$<br>$B \rightarrow B0 \mid 0$<br>$C \rightarrow C1 \mid A1$<br>$A \rightarrow 0$                    | f) | $S \rightarrow Bb \mid Cc$<br>$B \rightarrow Bb \mid Ab$<br>$C \rightarrow Cc \mid Ab$<br>$A \rightarrow a$ |
| g) | $S \rightarrow S1 \mid A0$<br>$A \rightarrow B1 \mid C1$<br>$B \rightarrow A0$<br>$C \rightarrow C0 \mid 0$                   | h) | $S \rightarrow Sa \mid Cc \mid a$<br>$C \rightarrow Bb$<br>$B \rightarrow Sa \mid a$                        |
| i) | $S \rightarrow C\perp$<br>$C \rightarrow A1 \mid B1 \mid 1$<br>$A \rightarrow A1 \mid C0 \mid 0$<br>$B \rightarrow C0 \mid 0$ | j) | $S \rightarrow A\perp$<br>$A \rightarrow Bb \mid a$<br>$B \rightarrow Bb \mid b$                            |
| k) | $S \rightarrow C\perp$<br>$B \rightarrow B1 \mid 0 \mid D0$<br>$C \rightarrow B1 \mid C1$<br>$D \rightarrow D0 \mid 0$        | l) | $S \rightarrow C\perp$<br>$C \rightarrow B1$<br>$B \rightarrow 0 \mid D0$<br>$D \rightarrow B1$             |
| m) | $S \rightarrow A0$<br>$A \rightarrow A0 \mid S1 \mid 0$   | n) | $S \rightarrow B0 \mid 0$<br>$B \rightarrow B0 \mid C1 \mid 0 \mid 1$<br>$C \rightarrow B0$                 |
| o) | $S \rightarrow A0 \mid A1 \mid B1 \mid 0 \mid 1$<br>$A \rightarrow A1 \mid B1 \mid 1$<br>$B \rightarrow A0$                   | p) | $S \rightarrow S0 \mid A1 \mid 0 \mid 1$<br>$A \rightarrow A1 \mid B0 \mid 0 \mid 1$<br>$B \rightarrow A0$  |
| r) | $S \rightarrow Sb \mid Aa \mid a \mid b$<br>$A \rightarrow Aa \mid Sb \mid a$   |    |   |

13. Грамматика  $G$  определяет язык  $L=L1 \cup L2$ , причем  $L1 \cap L2 = \emptyset$ . Написать регулярную грамматику  $G1$ , которая порождает язык  $L1 * L2$  (см. задачу 19 раздела I.). Для нее построить ДС и анализатор.

$$\begin{aligned}
 S &\rightarrow A\perp \\
 A &\rightarrow A0 \mid A1 \mid B1 \\
 B &\rightarrow B0 \mid C0 \mid 0 \\
 C &\rightarrow C1 \mid 1
 \end{aligned}$$

14. Даны две грамматики  $G1$  и  $G2$ , порождающие языки  $L1$  и  $L2$ . Построить регулярные грамматики для

- $L1 \cup L2$
- $L1 \cap L2$
- $L1 * L2$  (см. задачу 19 раздела I.)

$$G1: S \rightarrow S1 \mid A0 \\ A \rightarrow A1 \mid 0$$

$$G2: S \rightarrow A1 \mid B0 \mid E1 \\ A \rightarrow S1 \\ B \rightarrow C1 \mid D1 \\ C \rightarrow 0 \\ D \rightarrow B1 \\ E \rightarrow E0 \mid 1$$

Для полученной грамматики построить ДС и анализатор.

15. По данной грамматике G1 построить регулярную грамматику G2 для языка  $L1^*L1$  (см. задачу 19 раздела I.), где  $L1 = L(G1)$ ; по грамматике G2 — ДС и анализатор.

$$G1: S \rightarrow S1 \mid A1 \\ A \rightarrow A0 \mid 0$$

16. Построить лексический блок (преобразователь) для кода Морзе. Входом служит последовательность «точек», «тире» и «пауз» (например,  $\dots - \dots - \dots - \perp$ ). Выходом являются соответствующие буквы, цифры и знаки пунктуации. Особое внимание обратить на организацию таблицы.

### III. Метод рекурсивного спуска (РС-метод). Применимость РС-метода. КС-грамматики с действиями

1. Определить, применим ли РС-метод грамматике. Ответ обосновать.

a)  $S \rightarrow cA \mid B \mid d \\ A \rightarrow abA \mid c \mid \varepsilon \\ B \rightarrow bSc \mid aAb$

b)  $S \rightarrow aAbc \mid A \\ A \rightarrow bB \mid cBc \\ B \rightarrow bcB \mid a \mid \varepsilon$

c)  $S \rightarrow aSB \mid bAf \mid \varepsilon \\ A \rightarrow bAc \mid cS \\ B \rightarrow cB \mid d$

d)  $S \rightarrow aSB \mid bA \\ A \rightarrow aS \mid cA \mid \varepsilon \\ B \rightarrow bB \mid d$

e)  $S \rightarrow bABCb \mid d \\ A \rightarrow aA \mid cB \mid \varepsilon \\ B \rightarrow Sc \\ C \rightarrow a \{bb\}$

f)  $S \rightarrow aAb \mid cC \\ A \rightarrow a \{bab\} \\ B \rightarrow cAc \mid aB \mid \varepsilon \\ C \rightarrow Bb$

g)  $S \rightarrow aA\{xx\} \\ A \rightarrow bA \mid cBx \mid \varepsilon \\ B \rightarrow bSc$

h)  $S \rightarrow aSc \mid bA \mid \varepsilon \\ A \rightarrow cS\{da\}bA \mid d$

i)  $S \rightarrow bS \mid aAB \\ A \rightarrow bcA \mid ccA \mid \varepsilon \\ B \rightarrow cbB \mid \varepsilon$

j)  $S \rightarrow aASb \mid cfAd \\ A \rightarrow bA \mid c \mid \varepsilon$

2. Восстановить грамматику по функциям, реализующим синтаксический анализ методом рекурсивного спуска. Можно ли было по этой грамматике вести анализ методом рекурсивного спуска?

```

a). #include <iostream.h>
int c;
void A();
void B();

void gc() {cin >> c;}

void S() {
    A();
    if ( c != '\1')
        throw c;
}

void A() {
    B();
    while ( c == 'a' ) {
        gc();
        B();
    }
}

void B() {
    if ( c == 'b' )
        gc();
}

int main() {
    try { gc();
        S();
        cout << "SUCCESS !!!" << endl;
        return 0;
    }
    catch (int c) {
        cout << "ERROR on lexeme" << c << endl;
        return 1;
    }
}

```

```

b). #include <iostream.h>
int c;
void A();
void B();

void gc() {cin >> c;}

void S() {
    if (c == 'a') {
        gc();
        A();
    }
    else
        if (c == 'b' ) {
            gc();
            B();
        }
        else
            throw c;
}

void A() {
    if ( c == 'c' ) {
        gc();
        S();
    }
}

void B() {
    while ( c == ',' ) {
        gc();
    }
}

```

```

        if (c != 'b')
            throw c;
        gc();
    }
}

int main() {
    try { gc();
        S();
        cout << "SUCCESS !!!" << endl;
        return 0;
    }
    catch (int c) {
        cout << "ERROR on lexeme" << c << endl;
        return 1;
    }
}

```

c)

```

#include <iostream.h>
int c;
void A();

void gc() {cin >> c;}

void S (void) {
    if (c == 'a'){
        gc();
        S();
        if (c == 'b')
            gc();
        else
            throw c;
    }
    else A();
}

void A (void) {
    if (c == 'b')
        gc();
    else
        throw c;
    while (c == 'b')
        gc();
}

int main() {
    try { gc();
        S();
        cout << "SUCCESS !!!" << endl;
        return 0;
    }
    catch (int c) {
        cout << "ERROR on lexeme" << c << endl;
        return 1;
    }
}

```

d)

```

#include <iostream.h>
int c;
void A();
void B();

void gc() {cin >> c;}

void S (void) {
    A();
    if (c != '\1')
        throw c;
}

```

```

void A (void) {
    B();
    while ( c == 'a' ) {
        gc();
        B();
    }
    B();
}

void B (void) {
    if ( c == 'b' )
        gc();
}

int main() {
    try { gc();
          S();
          cout << "SUCCESS !!!" << endl;
          return 0;
        }
    catch (int c) {
        cout << "ERROR on lexeme" << c << endl;
        return 1;
    }
}

```

3. Задана КС-грамматика  $G=(VT, VN, P, S)$ . По ней написать синтаксический анализатор, реализующий РС-метод, предварительно преобразовав заданную грамматику, если это требуется для применимости РС-метода и если это возможно.

- |  |  |
|--|--|
| a) $S \rightarrow bS \mid aAB$<br>$A \rightarrow bcA \mid ccA \mid \varepsilon$<br>$B \rightarrow cbB \mid \varepsilon$                          | b) $S \rightarrow aASb \mid cfAd$<br>$A \rightarrow bA \mid c \mid \varepsilon$  |
| c) $S \rightarrow Sa \mid Sbb \mid fAc$<br>$A \rightarrow aB \mid d$<br>$B \rightarrow abB \mid Sb$  | d) $S \rightarrow cAd$<br>$A \rightarrow Aa \mid bB$<br>$B \rightarrow abB \mid \varepsilon$   |
| e) $S \rightarrow E \perp$<br>$E \rightarrow () \mid (E \{, E\}) \mid A$<br>$A \rightarrow a \mid b$   | f) $S \rightarrow P := E \mid \text{if } E \text{ then } S \mid \text{if } E \text{ then } S \text{ else } S$<br>$P \rightarrow I \mid I(E)$<br>$E \rightarrow T \{+T\}$<br>$T \rightarrow F \{*F\}$<br>$F \rightarrow P \mid (E)$<br>$I \rightarrow a \mid b$ |
| g) $F \rightarrow \text{function } I(I) S; I:=E \text{ end}$<br>$S \rightarrow ; I:=E S \mid \varepsilon$<br>$E \rightarrow E*I \mid E+I \mid I$ | h) $S \rightarrow SaAb \mid Sb \mid bABa$<br>$A \rightarrow acAb \mid cA \mid \varepsilon$<br>$B \rightarrow bB \mid \varepsilon$  |
| i) $S \rightarrow Ac \mid dBea$<br>$A \rightarrow Aa \mid Ab \mid daBc$<br>$B \rightarrow cB \mid \varepsilon$                                   | j) $S \rightarrow fASd \mid \varepsilon$<br>$A \rightarrow Aa \mid Ab \mid dB \mid f$<br>$B \rightarrow bcB \mid \varepsilon$  |

4. Какой язык порождает заданная грамматика? Провести анализ цепочки  $(a,(b,a),(a,(b)),b)\perp$ .

$$S \rightarrow \langle k = 0 \rangle E \perp$$

$$E \rightarrow A \mid (\langle k = k + 1; \text{if } (k == 3) \text{ ERROR();} \rangle E \{, E\}) \langle k = k - 1 \rangle$$

$$A \rightarrow a \mid b$$

5. Есть грамматика, описывающая цепочки в алфавите  $\{0, 1, 2, \perp\}$ :

$$\begin{aligned} S &\rightarrow A\perp \\ A &\rightarrow 0A \mid 1A \mid 2A \mid \varepsilon \end{aligned}$$

Дополнить эту грамматику действиями, исключающими из языка все цепочки, содержащие подцепочки 002.

6. Дана грамматика, описывающая цепочки в алфавите  $\{a, b, c, \perp\}$ :

$$\begin{aligned} S &\rightarrow A\perp \\ A &\rightarrow aA \mid bA \mid cA \mid \varepsilon \end{aligned}$$

Дополнить эту грамматику действиями, исключающими из языка все цепочки, в которых не выполняется хотя бы одно из условий:

- в цепочку должно входить не менее трех букв c ;
- если встречаются подряд две буквы a, то за ними обязательно должна идти буква b.

7. Есть грамматика, описывающая цепочки в алфавите  $\{0, 1\}$ :

$$S \rightarrow 0S \mid 1S \mid \varepsilon$$

Дополнить эту грамматику действиями, исключающими из языка любые цепочки, содержащие подцепочку 101.

8. Написать КС-грамматику с действиями для порождения

$$L = \{a^m b^n c^k \mid m+k = n \text{ либо } m-k = n\}.$$

9. Написать КС-грамматику с действиями для порождения

$$L = \{1^n 0^m 1^p \mid n+p > m, m \geq 0\}.$$

10. Дана грамматика с семантическими действиями:

$$\begin{aligned} S &\rightarrow \langle A = 0; B = 0 \rangle L \{L\} \langle \text{if } (A > 5) \text{ ERROR()} \rangle \perp \\ L &\rightarrow a \langle A = A+1 \rangle \mid b \langle B = B+1; \text{if } (B > 2) \text{ ERROR()} \rangle \mid \\ &\quad c \langle \text{if } (B == 1) \text{ ERROR()} \rangle \end{aligned}$$

Какой язык описывает эта грамматика ?

11. Дана грамматика:

$$\begin{aligned} S &\rightarrow E\perp \\ E &\rightarrow () \mid (E \{, E\}) \mid A \\ A &\rightarrow a \mid b \end{aligned}$$

Вставить в заданную грамматику действия, контролирующие соблюдение следующих условий:

- уровень вложенности скобок не больше четырех;
- на каждом уровне вложенности происходит чередование скобочных и бесскобочных элементов.

12. Включить в правила вывода действия, проверяющие выполнение следующих контекстных условий:

a) Пусть в языке  $L$  есть переменные и константы целого, вещественного и логического типов, а также есть оператор цикла

$$S \rightarrow \text{for } I = E \text{ step } E \text{ to } E \text{ do } S$$

Включить в это правило вывода действия, проверяющие выполнение следующих ограничений:

- тип  $I$  и всех вхождений  $E$  должен быть одинаковым;
- переменная логического типа недопустима в качестве параметра цикла.

Для каждой используемой процедуры привести ее текст на Си.

b) Дан фрагмент грамматики

$$P \rightarrow \text{program } D; \text{ begin } S \{ ; S \} \text{ end}$$

$$D \rightarrow \dots | \text{label } L \{ , L \} | \dots$$

$$S \rightarrow L \{ , L \} : S' | S'$$

$$S' \rightarrow \dots | \text{goto } L | \dots$$

$$L \rightarrow I$$

где  $I$  — идентификатор.

Вставить в грамматику действия, контролирующие выполнение следующих условий:

- каждая метка, используемая в программе, должна быть описана и только один раз;
- не должно быть одинаковых меток у различных операторов;
- если метка используется в операторе `goto`, то обязательно должен быть оператор, помеченный такой меткой.

Для каждой используемой процедуры привести ее текст на Си.

#### IV. Синтаксически управляемый перевод

1. Написать грамматику арифметического выражения, использующего операции  $+$ ,  $-$ ,  $*$ ,  $/$  и круглые скобки (приоритет стандартный), аргументы операций — переменные  $a$  и  $b$ , например:  $a+(b-a)*b/a+b$ . Предполагая, что анализ грамматики будет производиться РС-методом, вставить в нее действия (в виде `cout << ...`) по переводу таких выражений в ПОЛИЗ.

2. Дана грамматика языка  $L1$ , в которую вставлены действия по переводу цепочек языка  $L1$  в цепочки языка  $L2$ . Определить языки  $L1$  и  $L2$ .

$$\begin{aligned} \text{a) } S &\rightarrow a \langle a = 1; b = 0; \rangle A \perp \\ A &\rightarrow a \langle \text{if } ( a ) \{ \text{cout} \ll \text{'a'}; a = 0; \} \text{ else } a++; \rangle A | \\ &\quad bA \langle \text{if } ( b ) \{ \text{cout} \ll \text{'b'}; b = 0; \} \text{ else } b++; \rangle | \varepsilon \end{aligned}$$

$$\begin{aligned} \text{b) } S &\rightarrow \langle a = 0; \rangle E \perp \langle \text{cout} \ll \text{'\perp'}; \rangle \\ E &\rightarrow a \langle a = 1; \rangle E \langle \text{cout} \ll \text{'a'}; \rangle | \\ &\quad b \langle \text{if } ( a == 0 ) \text{cout} \ll \text{'b'}; \rangle E \langle \text{cout} \ll \text{'b'}; \rangle | \varepsilon \end{aligned}$$

3. Написать грамматику для языка  $L1$ . Вставить в нее действия по переводу цепочек языка  $L1$  в соответствующие цепочки языка  $L2$ .

В качестве действий можно использовать только оператор cout << ... .  
 СУ-перевод происходит во время анализа **методом рекурсивного спуска**.

- a)  $L1 = \{ a^n c^m b^n, n \geq 0, m \geq 1 \}$   
 $L2 = \{ 0^n 1^{n+m}, n \geq 0, m \geq 1 \}$
- b)  $L1 = \{ \alpha c^n, \alpha \in (a,b)^*, n \geq 1 \}$   
 $L2 = \{ a^n c^m, \text{ где } m - \text{ количество символов } a \text{ в цепочке } \alpha \}$
- c)  $L1 = \{ \omega \in \{a,b\}^+, \text{ где содержится } n \text{ символов } a \text{ и } m \text{ символов } b, \text{ расположенных в произвольном порядке; } n, m \geq 0; n+m > 0 \}$   
 $L2 = \{ 1^{n+m} 0^n \mid n, m \geq 0; n+m > 0 \}$
- d)  $L1 = \{ \omega \in \{a,b\}^+, \text{ где содержится } n \text{ символов } a, \text{ расположенных в произвольном порядке; } n \geq 0 \}$   
 $L2 = \{ 2^n \omega^R \mid n \geq 0, \omega^R - \text{ реверс цепочки } \omega \}$
- e)  $L1 = \{ 1^n 0^m 1^m 0^n \mid m, n > 0 \}$   
 $L2 = \{ 1^m 0^{n+m} \mid m, n > 0 \}$
- f)  $L1 = \{ \omega \perp \mid \omega \in \{a, b\}^+, \omega = \alpha^n, \text{ где } \alpha = ab \mid ba, n \geq 1 \}$   
 $L2 = \{ \omega \perp \mid \omega = \beta^n, \text{ где } \beta = \{ b, \text{ если } \alpha = ab; \text{ либо } a, \text{ если } \alpha = ba \} \}$

4. Написать грамматику для языка L1. Вставить в нее действия по переводу цепочек языка L1 в соответствующие цепочки языка L2.

В качестве действий можно использовать любые операторы.

СУ-перевод происходит во время анализа **методом рекурсивного спуска**.

- a)  $L1 = \{ 1^m 0^n \mid n, m > 0 \}$   
 $L2 = \{ 1^{m-n} \mid \text{если } m > n; \}$   
 $0^{n-m} \mid \text{если } m < n; \}$   
 $\varepsilon \mid \text{если } m = n \}$
- b)  $L1 = \{ b_i \mid b_i = (i)_2, \text{ т.е. } b_i - \text{ это двоичное представление числа } i \in \mathbb{N} \}$   
 $L2 = \{ (b_{i+1})^R \mid b_{i+1} = (i+1)_2, \omega^R - \text{ перевернутая цепочка } \omega \}$
- c)  $L1 = \{ \alpha \perp \mid \alpha \in \{a,b\}^* \}$   
 $L2 = \{ \beta \perp \mid \beta = b^n \alpha^R, \text{ где } n - \text{ количество символов } b \text{ в цепочке } \alpha, \text{ предшествующих первому вхождению символа } a; \alpha^R - \text{ реверс цепочки } \alpha \}$
- d)  $L1 = \{ \omega \perp \mid \omega \in \{a,b\}^+, \text{ где содержится } n \text{ символов } a \text{ и } m \text{ символов } b, \text{ расположенных в произвольном порядке} \}$   
 $L2 = \{ \omega \in \{a,b\}^* \mid \omega = a^{\lfloor n/2 \rfloor} b^{\lfloor m/2 \rfloor} \}$
- e)  $L1 = \{ \omega \perp \mid \omega \in \{0, 1\}^+ \text{ и представляет } (b_i)^R, \text{ т.е. реверс двоичного числа } i \}$   
 $L2 = \{ \omega \in \{/\}^*, \omega = /^i, \text{ т.е. количество } /, \text{ равное значению } i \}$



5. Построить грамматику, описывающую целые двоичные числа (количество 0 и 1 четно, допускаются незначащие нули). Вставить в нее действия по переводу этих целых чисел в четверичную систему счисления.

6. Написать грамматику для выражений, содержащих переменные, знаки операций +, -, \*, /, \*\* и скобки ( ) с обычным приоритетом операций и скобок. Включить в эту грамматику действия по переводу этих выражений в префиксную запись (операции предшествуют операндам). Предложить интерпретатор префиксной записи выражений.

7. В грамматику, описывающую выражения, включить действия по переводу выражений из инфиксной формы (операция между операндами) в функциональную запись.

Например,

$$\begin{array}{lcl} a+b & \implies & +(a, b) \\ a+b*c & \implies & +(a, *(b, c)) \end{array}$$

## V. ПОЛИЗ, перевод в ПОЛИЗ

1. Представить в ПОЛИЗе следующие выражения:

- |                       |                                  |
|-----------------------|----------------------------------|
| a) $a+b-c$            | b) $a*b+c/a$                     |
| c) $a/(b+c)*a$        | d) $(a+b)/(c+a*b)$               |
| e) $a$ and $b$ or $c$ | f) not $a$ or $b$ and $a$        |
| g) $x+y=x/y$          | h) $(x*x+y*y < 1)$ and $(x > 0)$ |

2. Для следующих выражений в ПОЛИЗе дать обычную инфиксную запись:

- |                |                        |                      |
|----------------|------------------------|----------------------|
| a) $ab*c+$     | b) $abc*/$             | c) $ab+c*$           |
| d) $ab+bc-/a+$ | e) $a$ not $b$ and not | f) $abca$ and or and |
| g) $2x+2x*<$   |                        |                      |

3. Используя стек, вычислить следующие выражения в ПОЛИЗе:

- |                              |                             |
|------------------------------|-----------------------------|
| a) $x y * x y / +$           | при $x = 8, y = 2$ ;        |
| b) $a 2 + b / b 4 * +$       | при $a = 4, b = 3$ ;        |
| c) $a b$ not and $a$ or not  | при $a = b = \text{true}$ ; |
| d) $x y * 0 > y 2 x - <$ and | при $x = y = 1$ .           |

4. Записать на ПОЛИЗе фрагмент программы на C:

- a) `S=0; i=1; while( i<10) { S = S*(i+i); i++;}`
- b) `if( (x+1) > (2*y) ) x = y; else y = (x+y)*2;`
- c) `i = 1; S = 0; while ( i < 10 && S < 40 ) { S = S + f(i); ++i; };`
- d) `if (z<x*y+5) a = x<y, z = (x+6)/(a-y); else z = y<<2;`
- e) `a = x + y < z*(t + x) ? - (a + b)/(c-d)*2 : ++x+5;`
- f) `S = x+y; i = 1; for (j = 0; j < n; j++) { S = S + i*j*S; i = i*x; }`
- g) `i = 1; S = 0; while ( i < 10 && S < 40 ) { S = S + f(i); ++i; }`
- h) `if (z<x*y+5) a = x<y, z = (x+6)/(a-y); else z = y<<2;`
- i) `do {x = y; y = 2*y;} while (x < k);`
- j) `S = 0; for (i = 1; i <= k; i = i + 1) S += i*i;`
- k) `switch (k) {  
    case 1:       a = not a; break;  
    case 2:       b = a or not b ;  
    case 3:       a = b ;  
}`

5. Выражение на ПОЛИЗе записать в инфиксной форме ( на C )

a)  $\underline{x} \underline{y} \underline{z} a \times 5 y / + * z 6 + 8 * - = ==$

b)  $\underline{x} a \times z y / + * z 6 a - * + =$

6. Является ли запись

$\underline{y}, 15, =, \underline{x}, x, a, b, c, 2, /, 1, +, *, -, *, a, -, =, \underline{y}, y, 2, -, =, y, 10, <=, 4, !F$   
 правильной записью в ПОЛИЗе следующего фрагмента программы на С (считаем, что элементы ПОЛИЗа нумеруются с 1):

$y = 15; do \{x = x*(a-b*(c/2+1)) -a; y = y-2;\} while y>10;$

Если не является, то объясните почему и предложите свой вариант ПОЛИЗа для этого фрагмента программы.

7. Является ли запись

$\underline{i}, 1, =, \underline{i}, n, <, 33, !F, \underline{a}, a, b, +, 1, -, x, y, y, 2, +, /, -, *, 5, +, =, \underline{i}, i, 2, +, =, 4, !$   
 правильной записью в ПОЛИЗе следующего фрагмента программы на С (считаем, что элементы ПОЛИЗа нумеруются с 1):

$for ( i = 1; i < n; i = i+2 ) a = (a+b-1)*(x-y/(y+2))+5;$

Если не является, то объясните почему и предложите свой вариант ПОЛИЗа для этого фрагмента программы.

8. a) записать на ПОЛИЗе фрагмент программы на С:

$i = 1; S = 0; while ( i < 10 \&\& S < 40 ) \{ S = S + f(i); ++i; \};$

b) выражение на ПОЛИЗе

$\underline{x} \underline{y} \underline{z} a \times 5 y / + * z 6 + 8 * - = ==$

записать в инфиксной форме ( на С ).

9. a) записать на ПОЛИЗе фрагмент программы на С:

$if (z < x*y+5) a = x < y, z = (x+6)/(a-y); else z = y << 2;$

b) выражение на ПОЛИЗе

$\underline{x} a \times z y / + * z 6 a - * + =$

записать в инфиксной форме ( на С ).

10. Предложить ПОЛИЗ для следующих операторов:

a)  $if ( E ) S_1; S_2; S_3$

семантика этого оператора такова: вычисляется значение выражения E; если его значение меньше 0, то выполняется оператор  $S_1$  ; если равно 0 — оператор  $S_2$  , иначе — оператор  $S_3$

f)  $choice ( S_1; S_2; S_3 ), E$

семантика этого оператора такова: вычисляется значение выражения E; если его значение равно i, то выполняется оператор  $S_i$  для  $i = 1, 2, 3$ ; иначе оператор choice эквивалентен пустому оператору.

g)  $cycle ( E_1; E_2; E_3 ), S$

семантика этого оператора отличается от семантики оператора for в языке Си только тем, что оператор S выполняется, по крайней мере, один раз (т.е. после вычисления выражения  $E_1$  сразу выполняется оператор S, затем вычисляется значение  $E_3$ , потом — значение  $E_2$ , которое используется для контроля за количеством повторений цикла также, как и в цикле for).

11. Написать грамматику для выражений, содержащих переменные, знаки операций +, -, \*, / и скобки ( ), где операции должны выполняться строго слева направо, но приоритет скобок сохраняется. Определить действия по переводу таких выражений в ПОЛИЗ.

12. Изменить приоритет операций отношения в М-языке ( сделать его наивысшим). Построить соответствующую грамматику, отражающую этот приоритет. Написать синтаксический анализатор, обеспечить контроль типов, задать перевод в ПОЛИЗ.

13. Написать КС-грамматику, аналогичную данной,

$$\begin{aligned} E &\rightarrow T \{+T\} \\ T &\rightarrow F \{*F\} \\ F &\rightarrow (E) \mid i \end{aligned}$$

с той лишь разницей, что в новом языке будет допускаться унарный минус перед идентификатором, имеющий наивысший приоритет (например,  $a * -b + -c$  допускается и означает  $a * (-b) + (-c)$ ).

В созданную грамматику вставить действия по переводу такого выражения в ПОЛИЗ. Для каждой используемой процедуры привести ее текст на Си.

14. Дана грамматика, описывающая выражения:

$$\begin{aligned} E &\rightarrow TE' & E' &\rightarrow +TE' \mid \varepsilon \\ T &\rightarrow FT' & T' &\rightarrow *FT' \mid \varepsilon \\ F &\rightarrow PF' & F' &\rightarrow \wedge PF' \mid \varepsilon \\ P &\rightarrow (E) \mid i \end{aligned}$$

Включить в эту грамматику действия по переводу этих выражений в ПОЛИЗ. Для каждой используемой процедуры привести ее текст на Си.

## СОДЕРЖАНИЕ

Задачи (Приложение к учебному пособию "Формальные грамматики и языки. Элементы теории трансляции") .....	1
I. Порождающие грамматики. Языки, порождаемые грамматиками. Классификация грамматик и языков по Хомскому .....	1
II. Регулярные грамматики, ДС, анализаторы по ДС. Преобразование НКА к ДКА .....	6
III. Метод рекурсивного спуска (РС-метод). Применимость РС-метода. КС-грамматики с действиями .....	10
IV. Синтаксически управляемый перевод.....	15
V. ПОЛИЗ, перевод в ПОЛИЗ .....	18