**Вариант 12 — Пути шахматного короля.**

Пути шахматного короля по бесконечной доске с возвращением по пройденным клеткам в обратном порядке: ARRUULBRDDLL.

1. **Набор граничных примеров и контрпримеров, демонстрирующих уточнение языка.**

|  |  |
| --- | --- |
| Верные примеры | Неверные примеры |
| ARRUULBRDDLL, ARDLBRUL | ARRUDBLLDU, ALUUDDR |

1. **КС-грамматика языка**

A={U,D,R,L,A,B}(U-шаг вверх, D-шаг вниз, R-шаг вправо, L-шаг влево, A-начало, B-конец)

<язык> ::= <q0>

<q0> ::= A<q1>

<q1> ::= U<q1>D | R<q1>L | D<q1>U |L<q1>R | B

1. **Грамматический разбор одного примера.**

<q0>

AR<q1>L

ARR<q1>LL

ARRU<q1>DLL

ARRUU<q1>DDLL

ARRUUL<q1>RDDLL

ARRUULBRDDLL

1. **Проверка того, что грамматика удовлетворяет однозначности ветвления по первому символу (принадлежит классу LL(1)**

<q0> ::= A<q1>

L1 = {A}

<q1> ::= U<q1>D | R<q1>L | D<q1>U |L<q1>R | B

L2={U}

L3={R}

L4={D}

L5={L}

L7={B}

L2∩L3∩L4∩L5∩L6∩L7={}

*Так как нетерминальные символы не могут начинаться с одинаковых терминальных символов,* исходная грамматика удовлетворяет однозначности ветвления по первому символу.

1. **Модифицированная грамматика.**

Исходная КС-грамматика удовлетворяет условию однозначности ветвления. Модифицировать грамматику не нужно.

1. **Таблица перевода языка в диаграммы.**

|  |  |
| --- | --- |
| <q0> |  |
| <q1> |  |

1. **Таблица перевода синтаксических диаграмм в алгоритм синтаксического анализа.**

|  |  |
| --- | --- |
|  | procedure CHESS  {  if ch==’A’ then read(ch) else error;  ACTION;  } |
|  | procedure ACTION  {  if c==’U’ then  {  read(c) ;  ACTION;  if c=='D' read(c);  else error;  }  else  {  if c==’D’ then  {  read(c) ;  q1;  if c=='U' read(c);  else error;  }  else  {  if c==’L’ then  {  read(c) ;  ACTION;  if c=='R' read(c);  else error;  }  else  {  if c==’R’ then  {  read(c) ;  ACTION;  if c=='L' read(c);  else error;  }  else  {  if c!='B' error;  }  }  }  }  } |

1. **Таблица перевода алгоритма в программу**

|  |  |
| --- | --- |
| procedure CHESS  {  if ch==’A’ then read(ch) else error;  ACTION;  } | bool q0(char c)  {  if (c=='A') return true;  else return false;  } |
| procedure ACTION  {  if c==’U’ then  {  read(c) ;  ACTION;  if c=='D' read(c);  else error;  }  else  {  if c==’D’ then  {  read(c) ;  q1;  if c=='U' read(c);  else error;  }  else  {  if c==’L’ then  {  read(c) ;  ACTION;  if c=='R' read(c);  else error;  }  else  {  if c==’R’ then  {  read(c) ;  ACTION;  if c=='L' read(c);  else error;  }  else  {  if c!='B' error;  }  }  }  }  } | bool q1(char ch1, char ch2)  {  if (ch1=='L' && ch2=='R') return true;  else  {  if (ch1=='R' && ch2=='L') return true;  else  {  if (ch1=='D' && ch2=='U') return true;  else  {  if (ch1=='U' && ch2=='D') return true;  else  {  if (ch1=='B') return true;  else return false;  }  }  }  }  } |

**Исходный код**

1. #include <iostream>
2. #include <cstring>
3. #include <windows.h>
5. using namespace std;
7. bool q0(char);
8. bool q1(char, char);
10. int main()
11. {
12. string str;
13. int menu=1;
14. SetConsoleCP(1251);
15. SetConsoleOutputCP(1251);
16. while (menu)
17. {
18. bool f=true;
19. system("CLS");
20. cout<<"Задание: Пути шахматного короля по бесконечной доске с возвращением по пройденным клеткам в обратном порядке."<<endl;
21. cout<<"Правила: A-начало, R-вправо, L-влево, D-вниз, U-вверх, B-конец."<<endl;
22. cout<<"Вводить символы необходимо строкой"<<endl<<endl;
24. cout<<"Верные примеры: ARRUULBRDDLL, ARDLBRUL." <<endl;
25. cout<<"Неверные примеры: ARRUDBLLDU, ALUUDDR."<<endl<<endl;
27. cout<<"Введите путь для проверки на корректность:"<<endl;
28. cin>>str;
29. int l=str.length();
30. if (l%2==0)
31. {
32. for(int i=0, m=l; i<l; ++i, --m)
33. {
34. if (i==0)
35. f\*=q0(str[i]);
36. else
37. f\*=q1(str[i],str[m]);
38. }
39. if (f)
40. cout<<"Путь составлен ВЕРНО."<<endl;
41. else
42. cout<<"Путь составлен НЕВЕРНО."<<endl;
43. }
44. else
45. cout<<"Путь составлен НЕВЕРНО."<<endl;
46. cout<<endl<<"Введите 0, если хотите выйти из программы, иначе введите любое другое число"<<endl<<">";
47. cin>>menu;
48. }
50. }
52. bool q0(char ch)
53. {
54. if (ch=='A') return true;
55. else return false;
56. }
58. bool q1(char ch1, char ch2)
59. {
60. if (ch1=='L' && ch2=='R') return true;
61. else
62. {
63. if (ch1=='R' && ch2=='L') return true;
64. else
65. {
66. if (ch1=='D' && ch2=='U') return true;
67. else
68. {
69. if (ch1=='U' && ch2=='D') return true;
70. else
71. {
72. if (ch1=='B') return true;
73. else return false;
74. }
75. }
76. }
77. }
78. }