

Рекурсия

## (допълнение)

### 1. Синтактичен анализ и намиране на стойност на изрази

В много случаи синтаксисът на различни езикови конструкции има рекурсивна структура. За формалното описание на такива конструкции се използва широко разпространения език на Бекус-Наур. Например, цяло число без знак можем да опишем по следните два начина:

$::= |$

или

$::= |$

Да се направи синтактичен анализ на символен низ според някакви правила означава да се провери дали низът е получен според тези правила и се определи видът на съставящите го части и връзките между тях. За целта се конструира т. нар. дърво на синтактичния разбор.

Например символните низове (9+7) и ((8-5)/2) са изрази според правилата:

$::= ()$

$::= +|-|*|/;$

$::= 0|1|\dots|9$

и съответните дървета на синтактичен разбор са:



Задачата на синтактичния анализ не се състои в рисуване на такива дървета, а в анализ на текста и ако е правилен, да се конструира някаква структура, която определя вида на съставящите низа части и връзките между тях. В горния случай следните структури определят дърветата на синтактичния разбор:

`expr(sign(+), expr(digit(9)), expr(digit(7)))`

`expr(sign(/), expr(sign(-), expr(digit(8)), expr(digit(5))),`

`expr(digit(2))).`

В тази част няма да разгледаме в пълнота задачата за синтактичния анализ на изрази. Ще дискутираме само проверката дали даден символен низ е израз в смисъла на дадени правила. Тъй като дефинициите на тези правила обикновено са рекурсивни, най-естествените решения са рекурсивните.

Задача 110. Да се състави рекурсивна програма, която проверява дали низ, въведен от клавиатурата *започва* с израз, определен от правилата:

`::= |()`

`::= +|-|*|/;`

`::= 0|1|...|9.`

Програма `Zad110.cpp` решава задачата. В нея са дефинирани булевите функции:

`bool formula1();` - проверява дали въведеният низ започва с формула;

`bool digit(char);` - проверява дали символ е цифра;

bool sign(char); - проверява дали символ е знак, според указаното правило.

Program Zad110.cpp

```
#include
```

```
bool formula1();
```

```
bool digit(char);
```

```
bool sign(char);
```

```
int main()
```

```
{if (formula1()) cout << c;
```

```
if (c != '(') return digit(c);
```

```
bool yes = formula1();
```

```
if (!yes) return false;
```

```
cin >> c;
```

```
if (!sign(c)) return false;
```

```
yes = formula1();
```

```
cin >> c;
```

```
return yes && c=='\n');
```

```
}
```

Забележка: Ще отбележим още веднаж, че програмата отговаря положително, както за низа 8, така и за низовете 88 и 8a, както за низа (3+9), така и за низа (3+9)a-5.

Задачата за правилността на символен низ относно дадени правила ще решаваме по аналогичен начин, но чрез индекс ще следим до коя позиция е разпозната търсената форма и отговорът ще е положителен само ако низът е изчерпен.

Задача 111. Да се състави програма, която определя дали символен низ е израз в смисъла на следните правила:

::= +|

-|

```
;
```

```
::= *|
```

```
/|
```

```
;
```

```
::= 0|1| ... |9.
```

Във функцията main се въвежда символен низ, в който се “изтриват” интервалите. Четенето на отделните символи от низа се осъществява чрез символната функция getchar:

```
char getchar()
```

```
{i++;
```

```
if (i==len) return ' ';
```

```
else return s[i];
```

```
}
```

която използва глобалните променливи

```
int i, // индекс на текущия символ
```

```
len; // дължина на символния низ s и
```

```
char s[100]; // символния низ, анализиран за израз
```

и връща сочения от индекса  $i$  символ, ако  $0 \leq i$