

Задача разбора (синтаксический анализ)

Даны КС-грамматика G и цепочка x .

$x \in L(G)$?

*Если да, то построить дерево вывода для x
(или левый вывод для x , или правый вывод для x).*

Существуют различные методы синтаксического анализа для КС-грамматик;

для некоторых подклассов есть эффективные методы, затрачивающие линейное время $O(n)$ на анализ цепочки длины n .

Каждый метод синтаксического анализа предполагает свой способ построения по грамматике программы-анализатора, которая будет осуществлять разбор цепочек.

В основе анализатора может быть автомат с магазинной памятью. Мы рассмотрим другой способ – метод рекурсивного спуска (система рекурсивных процедур).

Анализатор некорректен, если:

- не распознает хотя бы одну цепочку, принадлежащую языку;
- распознает хотя бы одну цепочку, языку не принадлежащую;
- зацикливается на какой-либо цепочке.

Метод анализа *применím* к данной грамматике, если анализатор, построенный в соответствии с этим методом, корректен.

Метод рекурсивного спуска (РС-метод)

Пример: пусть дана грамматика $G = (\{a, b, c, d\}, \{S, A, B\}, P, S)$,
где

$$P: S \rightarrow ABd$$

$$A \rightarrow a \mid cA$$

$$B \rightarrow bA$$

и надо определить, принадлежит ли цепочка $cabad$ языку $L(G)$.

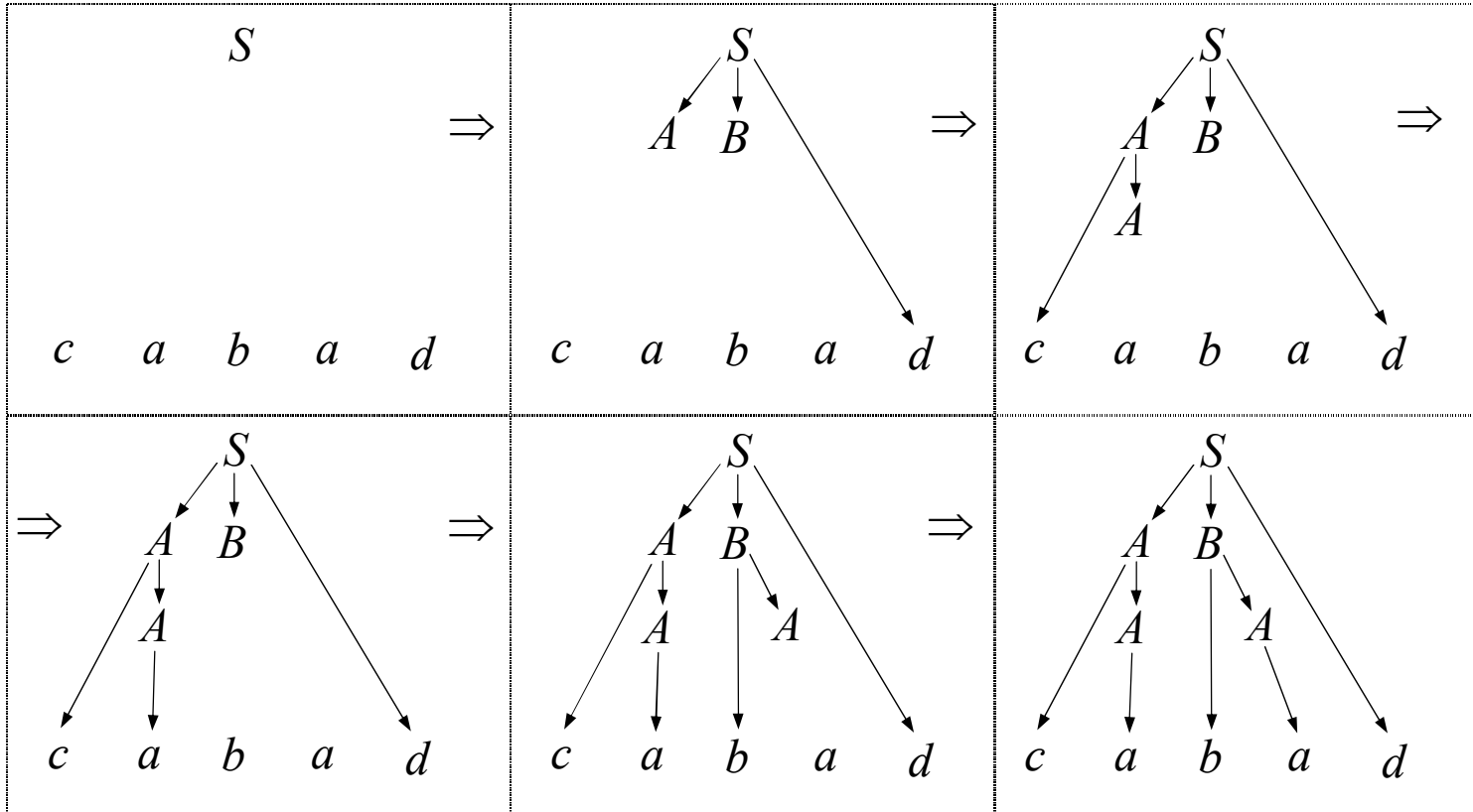
Построим левый вывод этой цепочки:

$$S \rightarrow ABd \rightarrow cABd \rightarrow caBd \rightarrow cabAd \rightarrow cabad$$

Следовательно, цепочка принадлежит языку $L(G)$.

$$S \rightarrow ABd \rightarrow cABd \rightarrow caBd \rightarrow cabAd \rightarrow cabad$$

Построение левого вывода эквивалентно построению дерева вывода методом «сверху вниз» (нисходящим методом) :



Для **каждого нетерминала** грамматики создается своя процедура с **именем** этого нетерминала; ее задача — начиная с указанного места исходной цепочки найти подцепочку, которая выводится из этого нетерминала.

Если подцепочку удалось найти, то работа процедуры считается нормально завершенной и осуществляется возврат в точку вызова, иначе — разбор прекращается и сообщается об ошибке, цепочка не принадлежит языку.

Тело каждой такой процедуры пишется непосредственно по правилам вывода соответствующего нетерминала: терминалы из правой части распознаются самой процедурой, а нетерминалы соответствуют вызовам процедур, носящих их имена.

```
#include <iostream>
using namespace std;
int c; // текущий символ
void A ();
void B ();
void gc ()
{
    cin >> c; // считать символ (лексему) из входного потока
}

void s ()
{
    cout << "S-->ABd, "; // применяемое правило вывода
    A();
    B();
    if ( c == 'd' ) gc();
    else throw c;
}
```

 $G_1:$ $S \rightarrow ABd$ $A \rightarrow a \mid cA$ $B \rightarrow bA$

```

void A ()
{
    if ( c == 'a' )
    {
        cout << "A-->a, ";
        gc ();
    }
    else if ( c == 'c' )
    {
        cout << "A-->cA, ";
        gc ();
        A ();
    }
    else
        throw c;
}

```

G_1 :

$$S \rightarrow ABd$$

$$A \rightarrow a \mid cA$$

$$B \rightarrow bA$$


```
void B ()
{
    if ( c == 'b' )
    {
        cout << "B-->bA, ";
        gc ();
        A ();
    }
    else
        throw c;
}
```

G_1 :

$S \rightarrow ABd$

$A \rightarrow a \mid cA$

$B \rightarrow bA$

```

int main ()
{
    try
    {
        gc ();
        S ();
        if ( c != '⊥' ) // проверяем, что достигнут конец
                        // цепочки
            throw c;
        cout << "SUCCESS !!!" << endl;
        return 0;
    }
    catch ( int c )
    {
        cout << "ERROR on lexeme" << c << endl;
        return 1;
    }
}

```

 $G_1:$
 $S \rightarrow ABd$
 $A \rightarrow a \mid cA$
 $B \rightarrow bA$

Достаточное условие применимости метода рекурсивного спуска

Для применимости метода рекурсивного спуска достаточно, чтобы каждое правило в грамматике имело вид:

(а) либо $X \rightarrow \alpha$,

где $\alpha \in (T \cup N)^*$ и это единственное правило вывода для этого нетерминала;

(б) либо $X \rightarrow a_1\alpha_1 \mid a_2\alpha_2 \mid \dots \mid a_n\alpha_n$,

где $a_i \in T$ для всех $i = 1, 2, \dots, n$; $a_i \neq a_j$ для $i \neq j$; $\alpha_i \in (T \cup N)^*$, т. е. если для нетерминала X правил вывода несколько, то они должны начинаться с терминалов, причем все эти терминалы должны быть различными;

Это условие не является необходимым.

Вопросы и задачи

1. Что означает применимость метода синтаксического анализа к КС-грамматике?
2. Сформулируйте достаточное условие применимости метода рекурсивного спуска.
3. Проверьте, что достаточные условия применимости метода рекурсивного спуска выполнено для данной КС-грамматики:

$$S \rightarrow aAd \mid bC$$

$$A \rightarrow aB \mid b$$

$$C \rightarrow d \mid cC$$

$$B \rightarrow Ab$$

4. Постройте, используя метод рекурсивного спуска, левый вывод в грамматике из задачи 3 для цепочек bcd , $aabbd$.