Найдем критерий применимости

РС-метод примени́м, если и только если левый вывод (или дерево нисходящим способом) можно построить, начиная с начального символа S, так, что на каждом шаге вывода решение о том, какое правило (альтернативу) применять для замены левого нетерминала, безошибочно принимается по первому символу из непрочитанной части входной цепочки (т. е. по «текущему» символу).

Рассмотрим примеры

G_2:

$S \to aA \mid B \mid d$

$A \to d \mid aA$

$B \to aA \mid a$

G_2 неоднозначна, РС-метод неприменим.

нельзя дать однозначный прогноз, что делать на первом шаге при анализе цепочки, начинающейся с символа a (т. е. по текущему символу a невозможно сделать однозначный выбор: $S \to aA$ или $S \to B$)

G_3 однозначна, но РС-метод неприменим

G_3:

$S \to A \mid B$

$A \to aA \mid d$

$B \to aB \mid b$

Определение: *множество first* (α) — это множество терминальных символов, которыми начинаются цепочки, выводимые из цепочки α в грамматике $G = \langle T, N, P, S \rangle$, т. е.

$$first(\alpha) = \{ a \in T \mid \alpha \Rightarrow a\alpha', \text{ где } \alpha \in (T \cup N)^+, \ \alpha' \in (T \cup N)^* \}.$$

Например: $first(A) = \{ a, d \}$, $first(B) = \{ a, b \}$. Пересечение этих множеств непусто: $first(A) \cap first(B) = \{ a \} \neq \varnothing$, и поэтому метод рекурсивного спуска к G_3 неприменим.

Итак, **наличие в грамматике правил вида** $X \to \alpha \mid \beta$**, таких что** $first(\alpha) \cap first(\beta) \neq \varnothing$**, делает метод рекурсивного спуска неприменимым**.

Рассмотрим еще несколько примеров.

G_4:

$S \to aA \mid BDc$

$A \to BAa \mid aB \mid b$

$B \to \varepsilon$

$D \to B \mid b$

$first(aA) = \{ a \}$, $first(BDc) = \{ b, c \}$;

$first(BAa) = \{ a, b \}$, $first(aB) = \{ a \}$,

$first(b) = \{ b \}$;

$first(\varepsilon) = \varnothing$;

$first(B) = \varnothing$, $first(b) = \{ b \}$.

Метод рекурсивного спуска неприменим к грамматике G_4, так как $first(BAa) \cap first(aB) = \{ a \} \neq \varnothing$.

G_5:

$S \to aA$

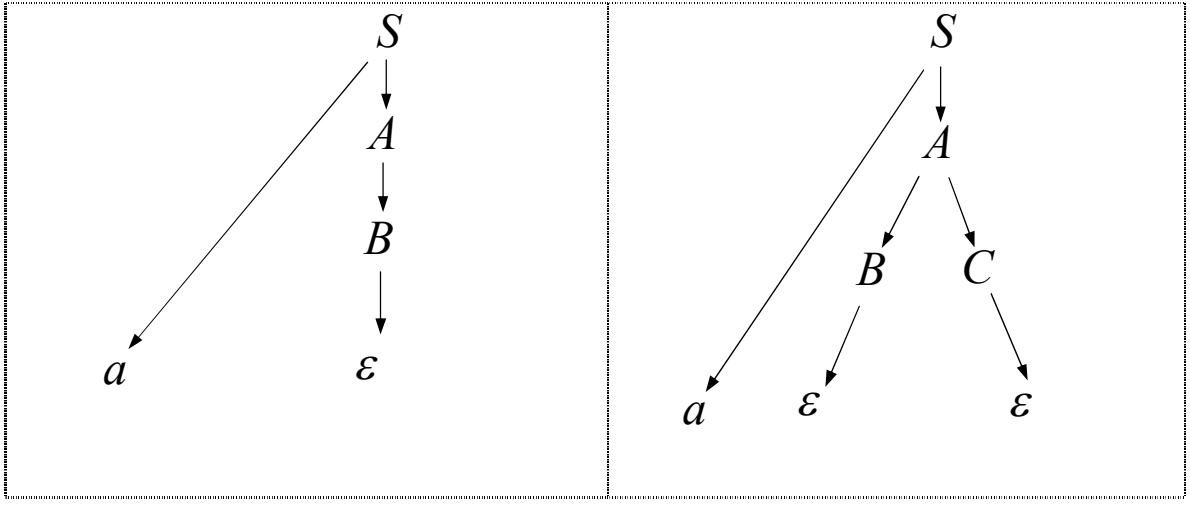
$A \to BC \mid B$

$C \to b \mid \varepsilon$

$B \to \varepsilon$

Пересечение множеств *first* пусто, но РС-метод неприменим.

Действительно, $BC \Rightarrow \varepsilon$ и $B \Rightarrow \varepsilon$. Цепочка a имеет два различных дерева вывода



Таким образом, **если в грамматике для двух различных правил $X \to \alpha \mid \beta$ выполняются соотношения $\alpha \Rightarrow \varepsilon$ и $\beta \Rightarrow \varepsilon$, то метод рекурсивного спуска неприменим**.

Рассмотрим примеры с единственной альтернативой, из которой выводится ε.

G_6:

$S \to cAd \mid d$

$A \to aA \mid \varepsilon$

Метод примен*и*м: если текущий символ a, то выбираем альтернативу $A \to aA$ иначе — $A \to \varepsilon$

G_7:

$S \to Bd$

$B \to cAa \mid a$

$A \to aA \mid \varepsilon$

Неприменим, т.к. для A невозможно правильно выбрать альтернативу без «заглядывания» на символ вперед.

Определение: *множество follow(A)* — это множество терминальных символов, которые могут появляться в сентенциальных формах грамматики непосредственно справа от A, т. е.

$$follow(A) = \{\, a \in T \mid S \Rightarrow \alpha A \beta,\ \beta \Rightarrow a\,\gamma,\ A \in N,\ \alpha, \beta, \gamma \in (T \cup N)^*\}$$

Тогда, **если в грамматике есть пара правил $X \to \alpha \mid \beta$, таких что $\beta \Rightarrow \varepsilon$, $first(X) \cap follow(X) \neq \varnothing$, то метод рекурсивного спуска неприменим к данной грамматике.**

Определение: *множество follow(A)* — это множество терминальных символов, которые могут появляться в сентенциальных формах грамматики непосредственно справа от A, т. е.

$$follow(A) = \{\, a \in T \mid S \Rightarrow \alpha A \beta,\ \beta \Rightarrow a\,\gamma,\ A \in N,\ \alpha, \beta, \gamma \in (T \cup N)^*\}$$

Тогда, **если в грамматике есть пара правил $X \to \alpha \mid \beta$, таких что $\beta \Rightarrow \varepsilon$, $first(X) \cap follow(X) \neq \varnothing$, то метод рекурсивного спуска неприменим к данной грамматике.**

Утверждение. Пусть G — КС-грамматика. Метод рекурсивного спуска применим к G, если и только если для любой пары альтернатив вида $X \to \alpha \mid \beta$ выполняются следующие условия:

(1) $first(\alpha) \cap first(\beta) = \varnothing$;

(2) справедливо не более чем одно из двух соотношений:

$\alpha \Rightarrow \varepsilon$, $\beta \Rightarrow \varepsilon$;

(3) если $\beta \Rightarrow \varepsilon$, то $first(X) \cap follow(X) = \varnothing$.

Канонический вид для РС-метода

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

(2) либо $X \to a_1\alpha_1 \mid a_2\alpha_2 \mid \ldots \mid a_n\alpha_n$,
где $a_i \in T$ для всех $i = 1, 2, \ldots, n$; $a_i \neq a_j$ 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 попарно различными;

(3) либо $X \to a_1\alpha_1 \mid a_2\alpha_2 \mid \ldots \mid a_n\alpha_n \mid \varepsilon$,
где $a_i \in T$ для всех $i = 1, 2, \ldots, n$; $a_i \neq a_j$ для $i \neq j$;
$\alpha_i \in (T \cup N)^*$, и $first(X) \cap follow(X) = \varnothing$.

Этот вид удобен для построения рекурсивных процедур, но он дает только достаточное условие применимости.

Вопрос*: если грамматика не удовлетворяет критерию применимости РС-метода, то существует ли эквивалентная КС-грамматика, для которой метод рекурсивного спуска применим?*

К сожалению, нет алгоритма, отвечающего на этот вопрос для произвольной КС-грамматики, т.е. это ***алгоритмически неразрешимая проблема.***

Модификация метода для грамматик с итерациями:
Для правил вида $L \to a\ \{,a\}$

то, что в фигурных скобках, может повторяться несколько раз или отсутствовать.

рекурсию заменяем итерацией:

```
void L()
{ if (c != 'a') throw c;
  gc();
 while (c == ',')
   {gc(); if (c != 'a') throw c; else
                                  gc();}
}
```

Важно, чтобы в любой сентенциальной форме после *L* не было запятой, иначе *L* прочитает «не свою» запятую. (Вместо запятой в данном примере может быть любой другой символ.)

Пример, когда анализатор по вышеприведенной схеме не дает корректный ответ:

G:

$S \to LB\perp$

$L \to a\ \{,\ a\}$

$B \to ,b$

Если для этой грамматики написать анализатор, действующий РС-методом, то цепочка *a,a,a,b* будет признана им ошибочной, хотя $a,a,a,b \in L(G)$.

В языках программирования после повторяющихся конструкций обычно идет какой-нибудь новый символ, так что подобных проблем не возникает:

var *a,b,c,d : integer;* или ***int*** *a,b,c,d;*

Если грамматику переписать без итерации { }

$S \to LB\perp$

$L \to a\ M$

$M \to ,\ a\ M \mid \varepsilon$

$B \to ,\ b$

то нетрудно видеть, что $first\ (,\ a) \cap follow\ (M) = \{\ ,\ \} \neq \varnothing$, и поэтому метод рекурсивного спуска неприменим.

Задача разбора (синтаксический анализ) для неоднозначных грамматик

две постановки задачи:

(1) *Даны КС-грамматика G и цепочка x. Требуется проверить: $x \in L(G)$? Если да, то построить все деревья вывода для x (или все левые выводы для x, или все правые выводы для x)*

Для решения этой задачи можно обобщить метод рекурсивного спуска, чтобы он работал с возвратами, пробуя различные подходящие альтернативы.

(2) *Даны КС-грамматика G и цепочка x. Требуется проверить: $x \in L(G)$? Если да, то построить одно дерево вывода для x (возможно, «наиболее подходящее» в некотором смысле).*

Рассмотрим пример. Грамматика неоднозначна. РС-метод неприменим.

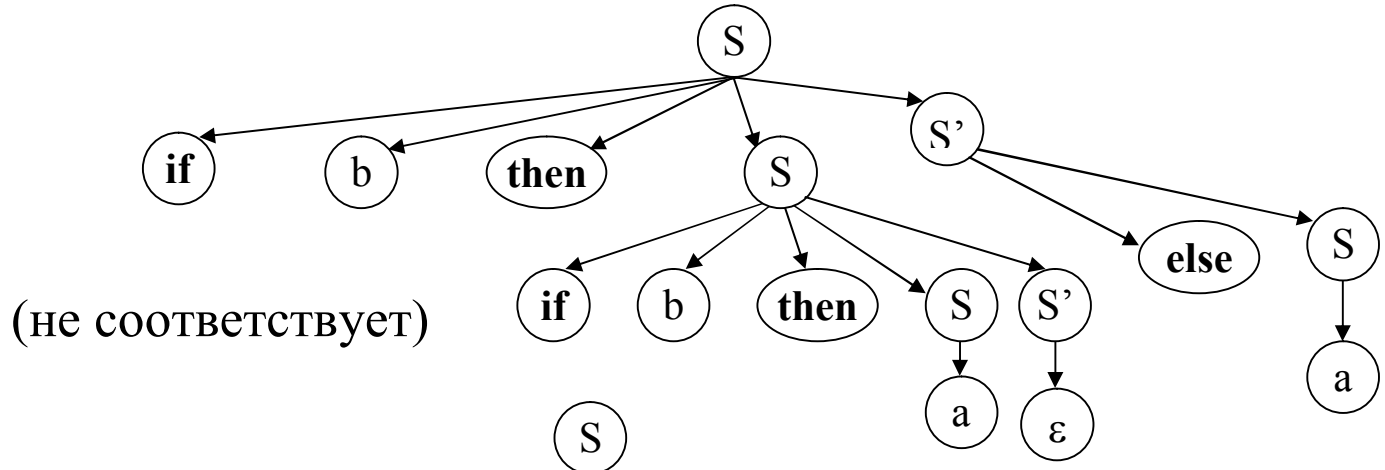
$G = (\{\mathbf{if}, \mathbf{then}, \mathbf{else}, a, b\}, \{S\}, P, S, S')$,

где $P = \{S \to \mathbf{if}\ b\ \mathbf{then}\ S\ S' \mid a;$

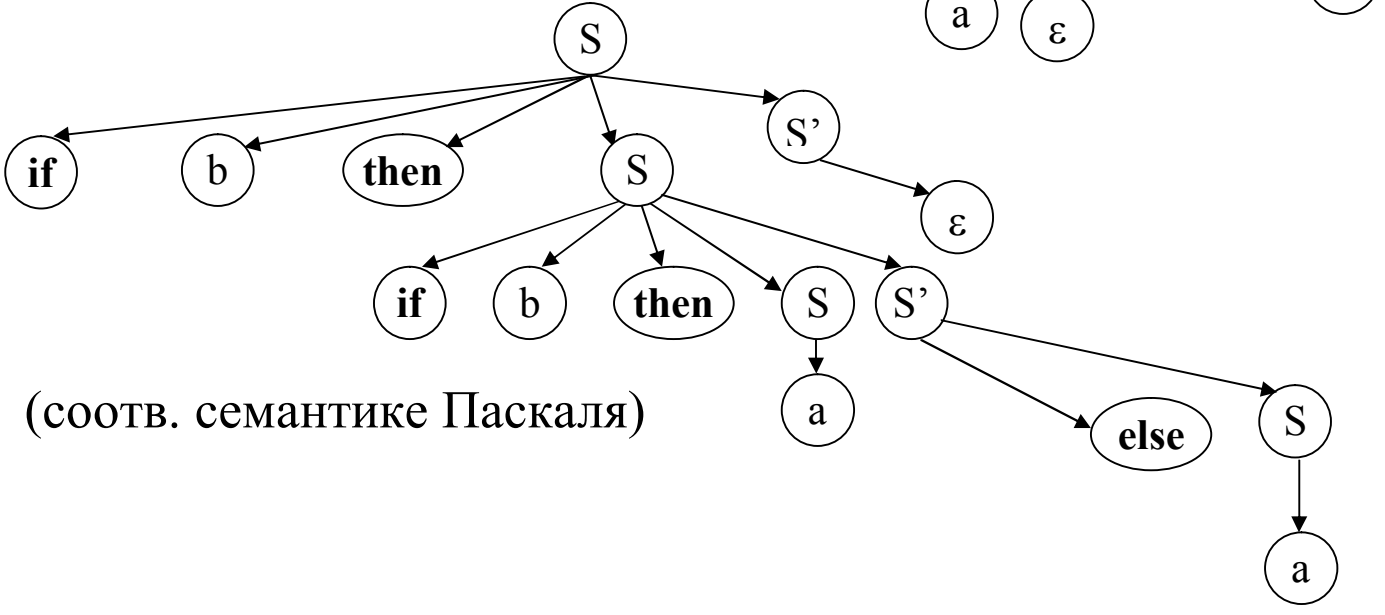
$S' \to \mathbf{else}\ S \mid \varepsilon\ \}.$

В этой грамматике для цепочки **if** *b* **then if** *b* **then** *a* **else** *a* можно построить два различных дерева вывода:

Одно из них соответствует семантике Паскаля: **else** относится к ближайшему i**f**. Такое дерево можно получить, написав РС-процедуры, где S’ по возможности отдает предпочтение непустой альтернативе.



(не соответствует)



(соотв. семантике Паскаля)

ПОЛИЗ – польская инверсная запись **(постфиксная запись)**

Пример. Обычной (инфиксной) записи выражения

a*(b+c)-(d-e)/f

соответствует такая постфиксная запись:

abc+*de-f/-

- порядок операндов остался таким же, как и в инфиксной записи,
- учтено старшинство операций,
- нет скобок.

Простым будем называть выражение, состоящее из одной константы или имени переменной.

Приоритет и ассоциативность операций в инфиксных выражениях позволяют четко установить границы операндов:

a — простое выражение ;

$a + b * c \sim a + (b * c)$;

$a - b + c - d \sim ((a - b) + c) - d$.

ПОЛИЗ выражений

(1) если E является простым выражением, то ПОЛИЗ выражения E — это само выражение E;

(2) ПОЛИЗом выражения $\boldsymbol{E_1\ \theta\ E_2}$,
где θ — знак бинарной операции, E_1 и E_2 операнды для θ,
является запись $\boldsymbol{E_1'\ E_2'\ \theta}$,
где E_1' и E_2' — ПОЛИЗ выражений E_1 и E_2 соответственно;

(3) ПОЛИЗом выражения $\boldsymbol{\theta\ E}$, где θ — знак унарной операции, а E — операнд θ,
является запись $\boldsymbol{E'\ \theta}$,
где E' — ПОЛИЗ выражения E;

(4) ПОЛИЗом выражения (E) является ПОЛИЗ выражения E.

Алгоритм интерпретации с помощью стека

ПОЛИЗ просматривается поэлементно слева направо. В стеке хранятся значения промежуточных вычислений и результат.

(1) если очередной элемент — операнд, то его значение заносится в стек;

(2) если очередной элемент — операция, то на "вершине" стека сейчас находятся ее операнды (это следует из определения ПОЛИЗа и предшествующих действий алгоритма); они извлекаются из стека, над ними выполняется операция, результат снова заносится в стек;

(3) когда выражение, записанное в ПОЛИЗе, прочитано, в стеке останется один элемент — это значение всего выражения.

Примечание: для интерпретации, кроме ПОЛИЗа выражения, необходима дополнительная информация об операндах, хранящаяся в таблицах.

Вопросы и задачи

1. Применим ли метод рекурсивного спуска к неоднозначным КС-грамматикам?
2. Сформулируйте критерий применимости метода рекурсивного спуска.
3. Проверьте, выполняется ли критерий применимости для для КС-грамматики:

$S \to A \mid C$	$S \to aAb \mid bC$	$S \to aA \mid C$	$S \to BA \mid \varepsilon$
$A \to aB \mid b$	$A \to aB \mid \varepsilon$	$A \to aBa \mid b$	$A \to aBa \mid \varepsilon$
$C \to d \mid cCc$	$C \to d \mid cC$	$C \to a \mid cC$	$C \to a \mid cC$
$B \to Ab$	$B \to Ab$	$B \to Ab$	$B \to \varepsilon$

4. Постройте польскую инверсную запись (ПОЛИЗ) для выражения:
   $(a+b)*((1-2)/b+3)$

5. Вычислите, используя стек, выражение в ПОЛИЗе : 6 7+8 7 - * 5 + 1* 2 1 - +