

## 2.8. Контекстно-свободные грамматики. Эффективные методы разбора: LL и LR грамматики.

### 1. LL-грамматики

#### 1.1 FIRST

$FIRST(\alpha)$  — множество терминалов, на которые может начинаться строка, выводимая из  $\alpha$ . И  $\varepsilon$ , если из нетерминала выводится  $\varepsilon$ .

Как строить. делаем пока можно:

1. Если  $X$  — терминал, то  $FIRST(X) = \{X\}$ .
2. Если  $X \rightarrow \varepsilon$ , добавить  $\varepsilon$  к  $FIRST(X)$ .
3. Если  $X \rightarrow Y_1 Y_2 \dots Y_k Z \alpha$  и  $\varepsilon \in FIRST(Y_i)$  для всех  $Y_i$ , то добавить в  $FIRST(X)$  все из  $FIRST(Z)$ .

Теперь для строки:  $FIRST(X_1 X_2 \dots X_n)$  — это объединение тех  $FIRST(X_i)$ , для которых  $\forall j < i : \varepsilon \in FIRST(X_j)$

#### 1.2 FOLLOW

$FOLLOW(A)$  — множество терминалов, которые могут появиться в строке после  $A$ . И  $\$$  (специальный символ), если может появиться в конце строки.

Как строить. делаем пока можно:

1. Добавить  $\$$  в  $FOLLOW(S)$  ( $S$  — стартовый нетерминал).
2. Если  $A \rightarrow \alpha B \beta$ , добавить в  $FOLLOW(B)$  все из  $FIRST(\beta)$  кроме  $\varepsilon$ .
3. Если  $A \rightarrow \alpha B$  или  $A \rightarrow \alpha B \beta$  и  $\varepsilon \in FIRST(\beta)$ , то добавить в  $FOLLOW(B)$  все из  $FOLLOW(A)$ .

#### 1.3 Матрица предиктивного анализа

Матрица  $M(A, a)$ ,  $A$  — нетерминал,  $a$  — терминал. Показывает, какую продукцию выбрать, если есть нетерминал  $A$ , а нужно получить строку, которая начинается с  $a$ .

Как строить. Добавляем продукцию  $A \rightarrow \alpha$  в  $M(A, a)$  для всех  $a \in FIRST(\alpha)$ , а если  $\varepsilon \in FIRST(\alpha)$ , то добавляем ее еще и в  $M(A, b)$  для всех  $a \in FOLLOW(\alpha)$ .

Если в каждой ячейке матрицы не более одной продукции, значит грамматика LL(1).

Как разбирать. Смотрим на самый левый нетерминал и на то, что на его месте в разбираемой строке. Смотрим в матрицу, выбираем нужную продукцию, применяем. Повторять пока не получится.

LL( $k$ )-грамматика — это такая грамматика, в которой можно выбрать нужную продукцию, зная  $k$  первых терминалов.

### 2. LR-грамматики

#### 2.1 LR-анализатор

LR-анализатор работает следующим образом. У него есть стек и с ним он может делать две вещи: *перенос* — перенести очередной элемент разбираемой строки в стек или *свертку* — взять несколько элементов с верхушки стека и свернуть их в нетерминал, из которого они получились.

Анализатор решает, какое из двух действий делать и какую продукцию выбрать для свертки, опираясь на то, что у него в стеке и на очередные символы входной строки. LR( $k$ )-анализатор смотрит на  $k$  символов вперед. Интересные случаи  $k = 0$  и  $k = 1$ .

Чтобы быстро опознавать основу в стеке у анализатора есть *состояния*. Состояния также кладутся в стек, между символами грамматики:  $s_0 X_1 s_1 X_2 s_2 \dots X_m s_m$ .

«Программа» для LR-анализатора — это таблицы *action* и *goto*.

Алгоритм работы анализатора:

```

1:   Установить  $ip$  на начало строки.;
2:   while (true) {
3:        $s$  — состояние на вершине стека,  $a$  — символ, на который указывает  $ip$ .;
4:       if ( $action[s, a] = \text{"перенос } s' \text{"}$ ) {
5:           Поместить в стек  $a$ , затем  $s'$ ; увеличить  $ip$ ;
6:       } else if ( $action[s, a] = \text{"свертка } A \rightarrow \beta \text{"}$ ) {
7:           Снять со стека  $2|\beta|$  символов.;
8:           Пусть  $s'$  — состояние на вершине стека.;
9:           Поместить в стек  $A$ , затем  $goto[s', A]$ ;
10:      } else if ( $action[s, a] = \text{"допуск"}$ ) {
11:          return;
12:      } else {
13:          error();
14:      }
15:  }

```

Как строить таблицы? По сути нужно построить автомат, распознающий основы продукций.

## 2.2 SLR-анализ

*Пункт* — продукция с точкой в правой части. Например, продукция  $A \rightarrow XY$  порождает три пункта: « $A \rightarrow .XY$ », « $A \rightarrow X.Y$ » и « $A \rightarrow XY.$ ». Физический смысл: точка — это текущее место разбора, то что слева уже прочитали, то что справа ожидается.