

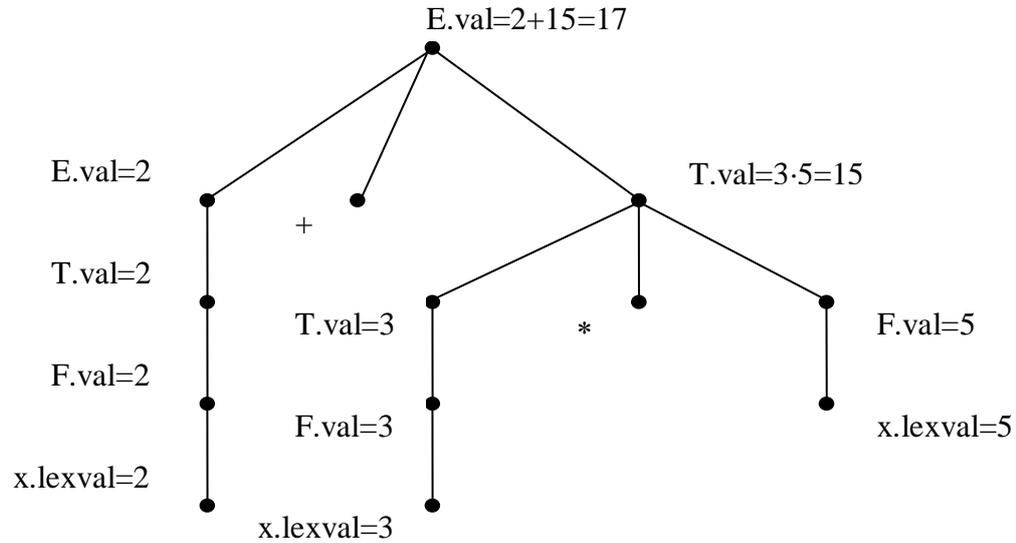
## § 2. Граф зависимости в атрибутивной грамматике. Синтаксическое дерево и даг в операторной грамматике

Опр. Графом зависимости цепочки  $w$  (в атрибутивной грамматике) называется ориентированный граф, полученный из аннотированного дерева вывода цепочки  $w$ . В каждом узле  $X$  аннотированного дерева каждому атрибуту  $X.attr$  сопоставляется вершина графа зависимости. Ориентированное ребро  $(A.attr, B.attr_B)$  соответствует семантическому правилу  $A.attr := f(\dots, B.attr_B, \dots)$ .

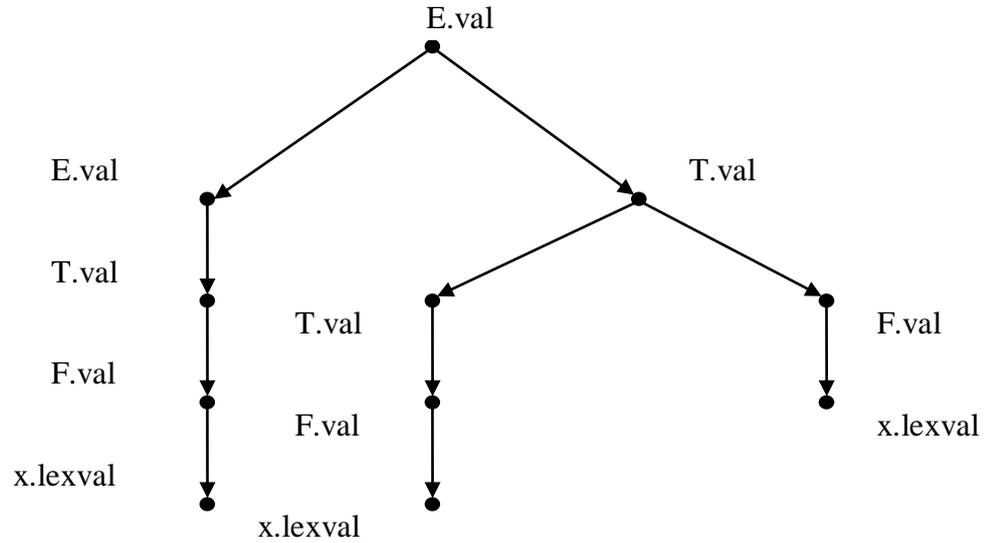
Пример. Аннотированная грамматика из § 1.

№	Правило вывода	Семантические правила
1	$E \rightarrow E_1 + T$	$E.val := E_1.val + T.val$
2	$E \rightarrow T$	$E.val := T.val$
3	$T \rightarrow T_1 * F$	$T.val := T_1.val \cdot F.val$
4	$T \rightarrow F$	$T.val := F.val$
5	$F \rightarrow (E)$	$F.val := E.val$
6	$F \rightarrow x$	$F.val := x.lexval$

Аннотированное дерево вывода для  $w = 2 + 5 * 3$ :



Граф зависимости:



Для более сложных примеров атрибутивной грамматики допускаются «фиктивные» атрибуты, которым также будет соответствовать вершина графа зависимости.

Теорема.

Все атрибуты в аннотированном дереве цепочки  $w$  могут быть вычислены  $\Leftrightarrow$  граф зависимости цепочки  $w$  не содержит циклов.

Один способ «сжатого» дерева вывода – синтаксическое дерево, применяется в случае операторной грамматики.

Напоминание:

Опр. Грамматика  $G$  называется операторной, если она  $\varepsilon$ -свободная, и ее неаннулирующие правила вывода имеют вид  $A \rightarrow \gamma_0 a_1 \gamma_1 a_2 \gamma_2 \dots a_n \gamma_n$ , где  $n \geq 0$ ,  $\gamma_i \in \Gamma \cup \{\varepsilon\}$ ,  $a_i \in \Sigma$ .

( $a_1, a_2, \dots, a_n$  – операторы,  $\gamma_0, \gamma_1, \gamma_2, \dots, \gamma_n$  – операнды.)

Пример. Грамматика из § 1:

$G = \{E \rightarrow E+T \mid T, T \rightarrow T * F \mid F, F \rightarrow (E) \mid x\}$  является операторной.

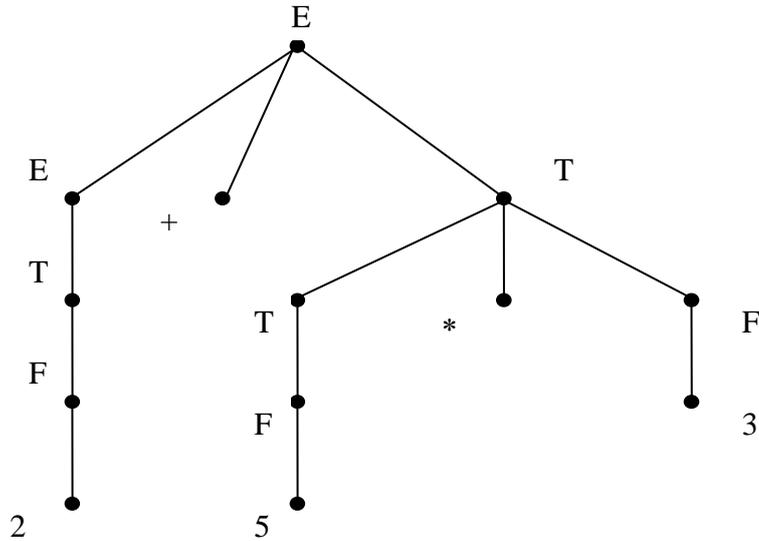
Алгоритм перехода от дерева вывода к синтаксическому дереву.

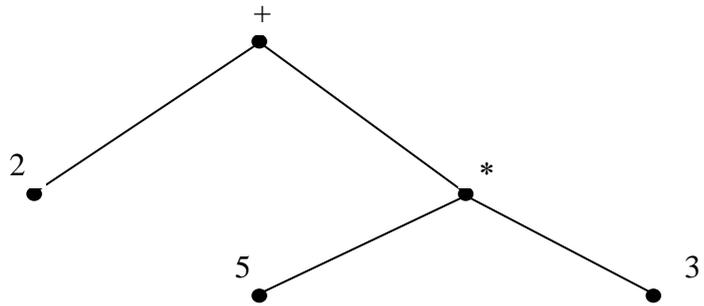
1) Листья, соответствующие терминалам, равным оператору, удаляются. Соответствующие терминалы связываются с отцом.

2) Ветви дерева, соответствующие цепным правилам, сворачиваются в одну вершину, помеченную последним терминалом. (Напоминание, цепными правилами называются:  $A_1 \rightarrow A_2$ ,  $A_2 \rightarrow A_3$ , ...,  $A_k \rightarrow a$ .)

Как следствие – все узлы помечаются терминалами.

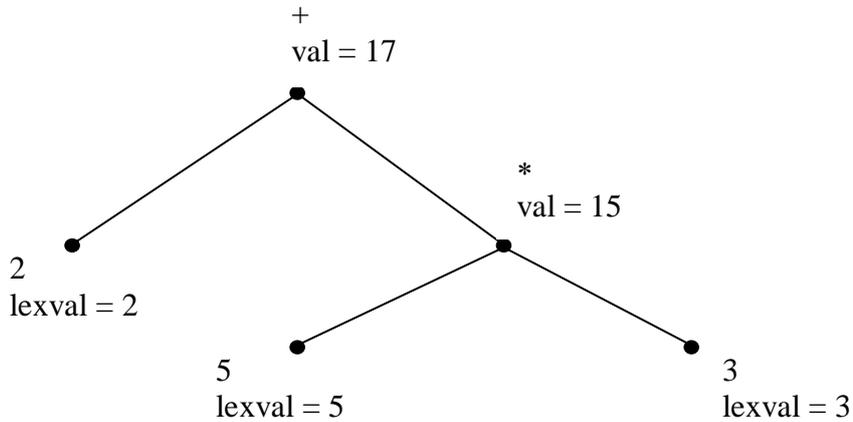
Пример. Преобразуем дерево вывода для  $w = 2 + 5 * 3$  в синтаксическое дерево:





Комментарий: значения атрибутов записываются в синтаксическом дереве возле узлов (операторов) или возле листьев (операндов).

Пример.

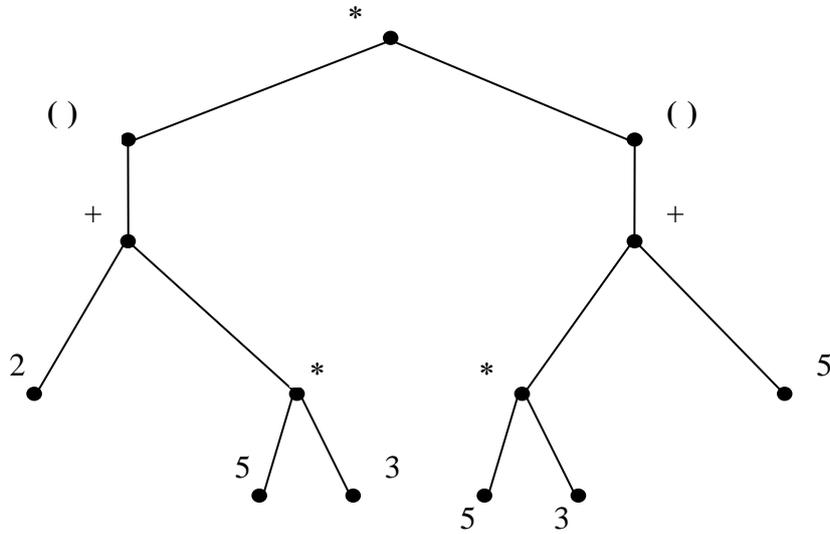


Опр. Синтаксический даг (Directed Acyclic Graph) для цепочки  $w$  в операторной грамматике – граф, полученный из синтаксического дерева отождествлением изоморфных поддеревьев.

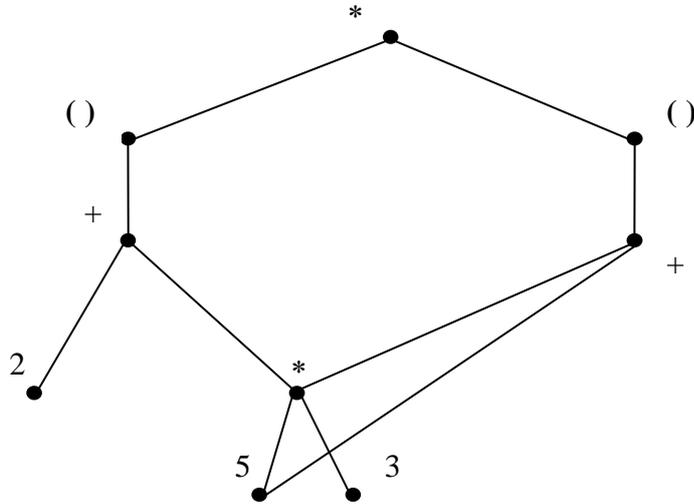
Пример. Рассмотрим цепочку  $w = (2 + 5 * 3) * (5 * 3 + 5)$  в грамматике из предыдущего примера.

Сначала построим синтаксическое дерево:

Синтаксическое дерево цепочки  $w = (2 + 5 * 3) * (5 * 3 + 5)$ :



Синтаксический даг цепочки  $w = (2 + 5 * 3) * (5 * 3 + 5)$ :



Построение синтаксических деревьев можно реализовать при помощи атрибутивной грамматики с единственным синтезируемым атрибутом `.ptr`. Значение `A.ptr` равно указателю на узел синтаксического дерева для нетерминала `A`.

Пусть операторная грамматика использует только двухместные операторы, а группировка скобками не считается оператором.

Для терминала `t` используется атрибут `.entry`, равный указателю на запись в таблице символов для символа `t`.

В семантических правилах применяются две функции:

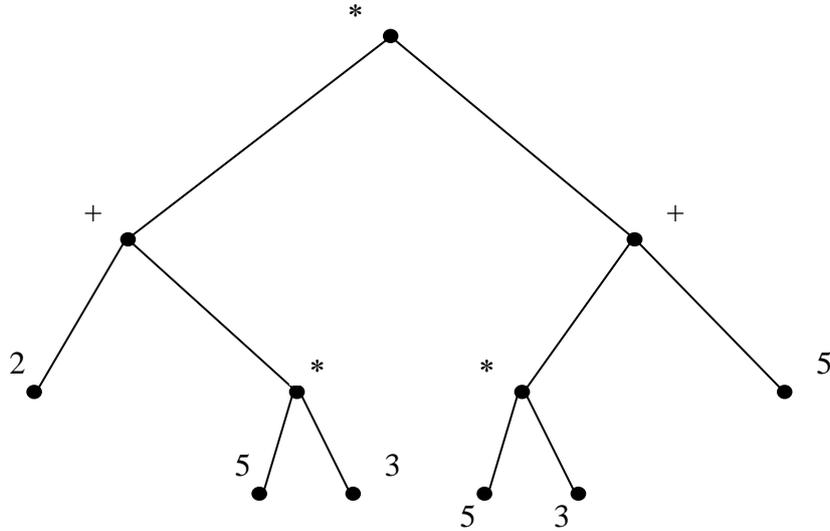
`MKLEAF(t, t.entry)` создает лист, помеченный операндом `t` со значением `t.entry`.

`MKNODE(op, left, right)` создает узел оператора `op`, содержащий указатели на левого и правого сыновей (операнды).

Для грамматики арифметических выражений  $G = \{E \rightarrow E+T \mid T, T \rightarrow T * F \mid F, F \rightarrow (E) \mid x\}$  атрибутная грамматика построения синтаксического дерева представлена в следующей таблице:

№	Правило вывода	Семантические правила
1	$E \rightarrow E_1 + T$	$E.ptr := \text{MKNODE}(+, E_1.ptr, T.ptr)$
2	$E \rightarrow T$	$E.ptr := T.ptr$
3	$T \rightarrow T_1 * F$	$T.ptr := \text{MKNODE}(*, T_1.ptr, F.ptr)$
4	$T \rightarrow F$	$T.ptr := F.ptr$
5	$F \rightarrow (E)$	$F.ptr := E.ptr$
6	$F \rightarrow x$	$F.ptr := \text{MKLEAF}(x, x.entry)$

Синтаксическое дерево цепочки  $w = (2 + 5 * 3) * (5 * 3 + 5)$ , построенное по предложенной атрибутивной грамматике:



Атрибутную грамматику построения дага можно получить, если внести изменения в функции MKLEAF(...) и MKNODE(...). Пусть перед созданием листа или узла функция проверяет, существует ли идентичный лист или узел. Если существует, то функция должна вернуть ссылку на существующий лист или узел. Если нет – то функция создает новый.