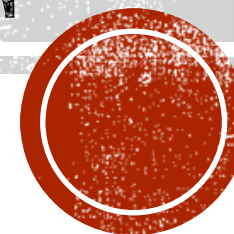


# ЯЗЫКИ ПРОГРАММИРОВАНИЯ И МЕТОДЫ ТРАНСЛЯЦИИ



Лекция 3

*16 февраля 2018 г.*

# ЭТАПЫ КОМПИЛЯЦИИ



# ТРАНСЛЯЦИЯ

**Опр. 1:** *Синтаксически управляемая трансляция* – это трансляция языков, управляемая грамматикой.

## Пример не-синтаксически управляемой трансляции

```
# рекурсивная функция разбора регулярного выражения
def parse():
    global data, currentI
    terms = [] # список термов. Терм - это то, что разделяется символами +
    term = [] # текущий формируемый терм
    while(currentI < len(data) and data[currentI] != ')'):
# если видим скобку, рекурсивно вызываем сами себя, а результат добавляем в качестве части терма
        if (data[currentI] == '('):
            currentI += 1
            term.append(parse())
# если видим +, то добавляем текущий терм к списку terms и начинаем формировать новый терм
        elif (data[currentI] == '+'):
            terms.append(term)
            term = []
# если видим *, то достаем последний автомат из терма, делаем из него замыкание и кладем обратно
        elif (data[currentI] == '*'):
            term.append(makeClosureNSM(term.pop()))
# если текущий символ - буква, то делаем из него автомат и добавляем к терму
        else:
            term.append(makeSymbolNSM(data[currentI]))
            currentI += 1
# Либо дошли до конца строки, либо до закрывающей скобки.
# Результат одинаковый - добавляем последний сформированный терм к списку terms,
# формируем из всего списка термов общее регулярное выражение и возвращаем его
    terms.append(term)
    return makeOr(terms)
```



# СХЕМЫ ТРАНСЛЯЦИИ

**Опр. 2:** *Схема трансляции* – грамматика с семантическими действиями.

**Опр. 3:** *Семантическое действие* – фрагмент любого кода на любом языке программирования, либо на алгоритмическом языке, вставленный в правило грамматики.

$B \rightarrow X \{a\} Y$

- для **рекурсивного спуска** фрагмент  $\{a\}$  выполняется в тот момент, когда мы, полностью обработав  $X$  (если терминал – считав, если нетерминал – вызвав), переходим к обработке  $Y$ .
- для **табличного анализатора** – в момент раскрытия на стеке символа  $B$  на стек кладется правая часть правила, в том числе и фрагмент  $\{a\}$ . В этом случае  $\{a\}$  воспринимается как терминал особого типа, который выполняется в момент его снятия со стека.



# СХЕМЫ ТРАНСЛЯЦИИ

**Грамматика:**

$S \rightarrow \{print\}aS$

$S \rightarrow bB$

$B \rightarrow \{print\}$

**Слово:**

aab

	a	b	←
S	Rep (Sa{print}) Retain	Rep (Bv) Retain	Reject
B	Reject	Reject	Rep ({print}) Retain
a	Pop Adv		
b	Reject	Pop Adv	
{print}	Pop Retain Out ({print})	Pop Retain Out ({print})	Pop Retain Out ({print})
∇	Reject	Reject	Accept



# АТТРИБУТЫ И СУО

**Опр. 4:** *Атрибут* – ячейка памяти, ассоциированная с символом грамматики.

**Опр. 5:** *Аннотированное дерево разбора* – это дерево разбора с указанием значений его атрибутов.

**Опр. 6:** *Синтезируемый атрибут* в узле **N** вычисляется только с использованием значений атрибутов в дочерних по отношению к **N** узлах и в самом узле **N**. **Нетерминал** с вычисляемым **синтезируемым** атрибутом всегда находится в **заголовке** правила.

**Опр. 7:** *Наследуемый атрибут* в узле **N** определяется с использованием значений атрибутов в родительском по отношению к **N** узле, в самом узле **N** и дочерних узлах его родительского узла. **Нетерминал** с вычисляемым **наследуемым** атрибутом всегда находится в **теле** правила.



# АТТРИБУТЫ И СУО

**Опр. 8:** Синтаксически управляемые определения (СУО) – это грамматика с совокупностью семантических правил, связанных с продукциями грамматики.

**Опр. 9:** Семантическое правило указывает порядок вычисления атрибутов грамматики.

ПРОДУКЦИЯ	СЕМАНТИЧЕСКОЕ ПРАВИЛО
1) $L \rightarrow E \mathbf{n}$	$L.val = E.val$
2) $E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
3) $E \rightarrow T$	$E.val = T.val$
4) $T \rightarrow T_1 * F$	$T.val = T_1.val \times F.val$
5) $T \rightarrow F$	$T.val = F.val$
6) $F \rightarrow (E)$	$F.val = E.val$
7) $F \rightarrow \mathbf{digit}$	$F.val = \mathbf{digit.lexval}$

**Опр. 10:** СУО, содержащее только синтезируемые атрибуты, называется *S-атрибутным*.

**Опр. 11:** Побочные действия в СУО – инструкции, работающие с внешними по отношению к грамматике объектами (таблица символов, файлы, экран монитора, ...)

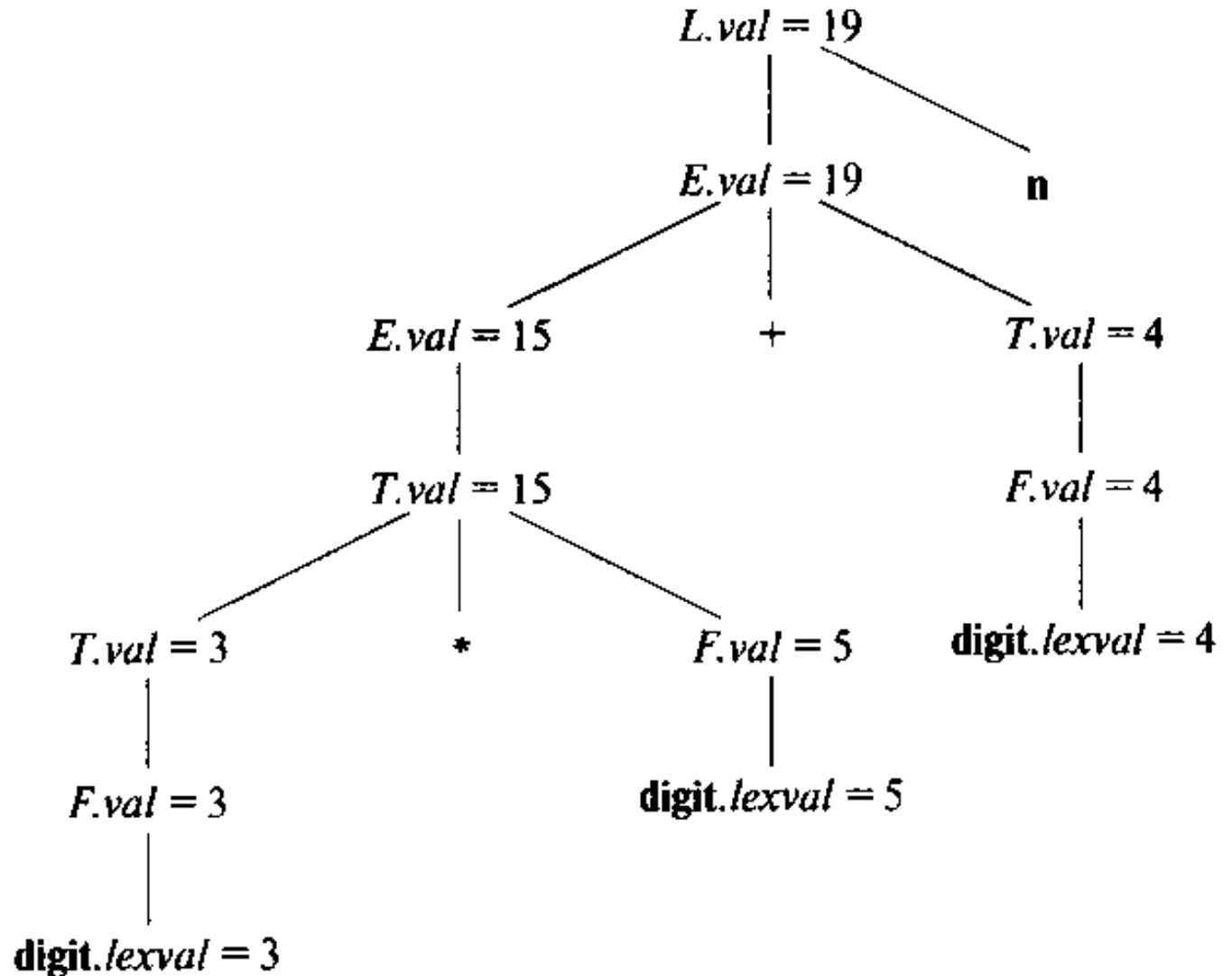
**Опр. 12:** СУО без побочных действий называется *атрибутной грамматикой*.

# АТТРИБУТЫ И СУО

**Опр. 13:** Аннотированное дерево разбора с указанием значений его

**Опр. 14:** Граф зависимостей изображенными атрибутами в определенном разборе. Ребро от одного атрибута к другому атрибуту необходимо для

ПРОДУКЦИЯ	СЕМАНТИЧЕСКОЕ ПРАВИЛО
1) $L \rightarrow E n$	$L.val = E.val$
2) $E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
3) $E \rightarrow T$	$E.val = T.val$
4) $T \rightarrow T_1 * F$	$T.val = T_1.val \times F.val$
5) $T \rightarrow F$	$T.val = F.val$
6) $F \rightarrow (E)$	$F.val = E.val$
7) $F \rightarrow \mathbf{digit}$	$F.val = \mathbf{digit.lexval}$





# АТРИБУТЫ И СУО

Построим аннотированное дерево разбора и граф зависимостей для следующей LL(1)-грамматики для выражения  $3*5$ :  $\text{digit}_3 * \text{digit}_5 \$$

ПРОДУКЦИЯ	СЕМАНТИЧЕСКИЕ ПРАВИЛА
1) $T \rightarrow F T'$	$T'.inh = F.val$ $T.val = T'.syn$
2) $T' \rightarrow *F T'_1$	$T'_1.inh = T'.inh \times F.val$ $T'.syn = T'_1.syn$
3) $T' \rightarrow \epsilon$	$T'.syn = T'.inh$
4) $F \rightarrow \mathbf{digit}$	$F.val = \mathbf{digit.lexval}$



# АТРИБУТЫ И СУО

Построим аннотированное дерево разбора и граф зависимостей для следующей LL(1)-грамматики для выражения `float id1 , id2 , id3`

ПРОДУКЦИЯ	СЕМАНТИЧЕСКИЕ ПРАВИЛА
1) $D \rightarrow T L$	$L.inh = T.type$
2) $T \rightarrow \mathbf{int}$	$T.type = \mathbf{integer}$
3) $T \rightarrow \mathbf{float}$	$T.type = \mathbf{float}$
4) $L \rightarrow L_1 , \mathbf{id}$	$L_1.inh = L.inh$ $addType(\mathbf{id.entry}, L.inh)$
5) $L \rightarrow \mathbf{id}$	$addType(\mathbf{id.entry}, L.inh)$



# УПРАЖНЕНИЯ



# УПРАЖНЕНИЯ

## Упражнение 1

*Нарисовать  
синтаксическое дерево  
разбора и строку вывода  
для входа  $bacb$ :*

1.  $S \rightarrow A b B$
2.  $A \rightarrow \{w\} a c$
3.  $A \rightarrow b A \{x\}$
4.  $B \rightarrow \{y\}$



# УПРАЖНЕНИЯ

## Упражнение 2

Нарисовать аннотированное дерево разбора для выражения  $(3+4) * (5+6) \$$

ПРОДУКЦИЯ	СЕМАНТИЧЕСКОЕ ПРАВИЛО
1) $L \rightarrow E \mathbf{n}$	$L.val = E.val$
2) $E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
3) $E \rightarrow T$	$E.val = T.val$
4) $T \rightarrow T_1 * F$	$T.val = T_1.val \times F.val$
5) $T \rightarrow F$	$T.val = F.val$
6) $F \rightarrow (E)$	$F.val = E.val$
7) $F \rightarrow \mathbf{digit}$	$F.val = \mathbf{digit.lexval}$



# УПРАЖНЕНИЯ

## Упражнение 3

Нарисовать аннотированное дерево разбора для строки  $a_1cb_2b_3a_4c$

1.  $S_p \rightarrow A_q b_r A_t \quad p \leftarrow r+t$
2.  $A_p \rightarrow a_p \{w\}_p c$
3.  $A_p \rightarrow b_q A_r \{x\}_p \quad p \leftarrow q+r$



# УПРАЖНЕНИЯ

## Упражнение 4

Используя следующие грамматику с СУО нарисовать аннотированное дерево разбора и граф зависимостей для строк

(a)  $a_2$                       (b)  $b_1ca_3$                       (c)  $b_2b_3cca_4$ :

- |  |                          |
|--|--------------------------|
| 1. $S_p \rightarrow A_{q,r} B_t$       | $p \leftarrow q * t$     |
|  | $r \leftarrow q + t$     |
| 2. $A_{p,q} \rightarrow b_r A_{t,u} c$ | $u \leftarrow r$         |
|  | $p \leftarrow r + t + u$ |
| 3. $A_{p,q} \rightarrow \varepsilon$   | $p \leftarrow \emptyset$ |
| 4. $B_p \rightarrow a_p$               |                          |

## Упражнение 5

Можно ли написать LL(1)-анализатор для грамматики из упр. 4. Ответ обоснуйте.

