Лекция 5

Синтаксически управляемая трансляция

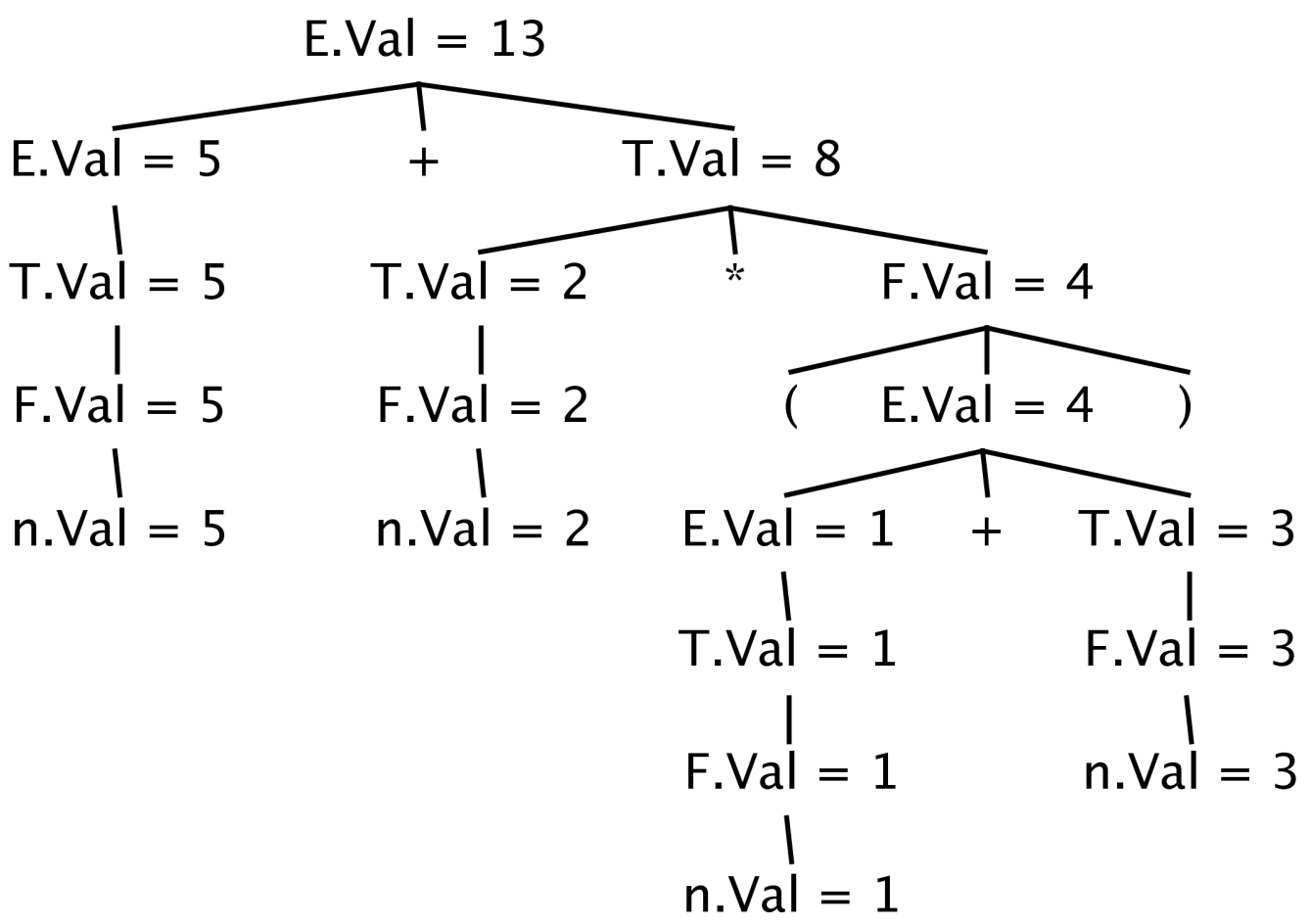
§30. Синтаксически управляемые определения

Синтаксический анализ позволяет ответить на вопрос, является ли цепочка предложением языка, и построить дерево вывода предложения. Можно расширить алгоритмы синтаксического анализа, приспособив их для перевода предложений в другой язык.

Определение. *Аннотированное дерево синтаксического разбора* – это дерево разбора для некоторого предложения языка, с каждым узлом которого связана структура, состоящая из именованных полей – *атрибутов*.

Значениями атрибутов может быть всё, что угодно – числа, строки, деревья, и т.д. Заставив алгоритм синтаксического анализатора вычислять значения атрибутов по специальным *семантическим правилам*, ассоциированным с правилами грамматики, мы можем добиться перевода предложений на другой язык. Это называется *синтаксически управляемой трансляцией*.

Пример.



$$
\begin{aligned}
E &\rightarrow E + T \\
E &\rightarrow T \\
T &\rightarrow T \star F \\
T &\rightarrow F \\
F &\rightarrow (E) \\
F &\rightarrow n
\end{aligned}
$$

Значения атрибутов в узлах аннотированного дерева вычисляются с помощью синтаксически управляемого определения.

Определение. *Синтаксически управляемое определение* (syntax-directed definition) – это тройка $\langle G, A, R\rangle$, в которой:
$G = \langle N, T, S, P\rangle$ – КС-грамматика;
A – конечное множество атрибутов, при этом с каждым символом $x \in N \cup T$ связано 0 или более атрибутов;
R – конечное множество семантических правил, при этом с каждым правилом грамматики связано 0 или более семантических правил.
Семантическое правило, связанное с правилом грамматики
$X \to x_1 x_2 \dots x_n$, имеет вид $b := f(c_1, c_2, \dots, c_k)$,
где b – атрибут символа, входящего в правило,
$c_1, c_2, \dots, c_k$ – атрибуты любых других символов, входящих в правило,
f – функция, вычисляющая значение атрибута b по значениям атрибутов $c_1, c_2, \dots, c_k$.

Определение. *Атрибутная грамматика* (attribute grammar) – это синтаксически управляемое определение, в котором функции в семантических правилах не имеют побочных эффектов.

Пример. Атрибутная грамматика арифметических выражений.

Правила грамматики	Семантические правила
$E \to E_1 + T$	$E.Val := E_1.Val + T.Val$
$E \to T$	$E.Val := T.Val$
$T \to T_1 \star F$	$T.Val := T_1.Val \times F.Val$
$T \to F$	$T.Val := F.Val$
$F \to (E)$	$F.Val := E.Val$
$F \to n$	$F.Val := n.LexVal$

Говорят, что атрибут b одного из символов правила грамматики p, *зависит* от атрибутов $c_1, c_2, \ldots, c_k$ других символов правила p, если с правилом p связано семантическое правило $b := f(c_1, c_2, \ldots, c_k)$.

Определение. *Синтезируемый атрибут*:

– для терминального символа это атрибут, вычисляемый на этапе лексического анализа;

– для нетерминального символа это атрибут, вычисляемый на основе синтезируемых атрибутов дочерних узлов дерева синтаксического разбора.

Определение. *Наследуемый атрибут* – это атрибут символа, вычисляемый на основе атрибутов родительского узла или соседних узлов (то есть, узлов, имеющих того же родителя).

Пример. Атрибутная грамматика объявлений переменных.

Правила грамматики	Семантические правила
$D \to T\ L$	$L.In := T.Type$
$T \to$ int	$T.Type := integer$
$T \to$ float	$T.Type := real$
$L \to L_1,\ v$	$L_1.In := L.In$ $v.Type = L.In$
$L \to v$	$v.Type := L.In$

Атрибут $T.Type$ – синтезируемый, атрибуты $L.In$ и $v.Type$ – наследуемые.

§31. Порядок выполнения семантических правил

Преимущество синтаксически управляемых определений – компактность, достигаемая за счёт того, что не указывается порядок вычисления семантических правил.

Методы выбора порядка выполнения семантических правил:
– метод, связанный с анализом зависимостей атрибутов в дереве разбора;
– метод, основанный на внедрении семантических действий в правила грамматики.

Метод выбора порядка выполнения семантических правил, связанный с анализом зависимостей атрибутов в дереве разбора

Определение. *Граф зависимостей атрибутов* для дерева синтаксического разбора – это направленный граф, в узлах которого расположены атрибуты узлов дерева, а дуги обозначают зависимость атрибутов (дуга ведёт из узла c в узел b, если атрибут b зависит от атрибута c).

Если синтаксически управляемое определение не содержит ошибок, то граф зависимостей атрибутов будет ациклическим для любого дерева синтаксического разбора.

Любая топологическая сортировка графа зависимостей атрибутов даёт правильный порядок выполнения семантических правил, вычисляющих атрибуты. (Топологическая сортировка – это упорядочивание узлов направленного ациклического графа согласно частичному порядку, заданному дугами этого графа на множестве его узлов.)

Метод задания порядка выполнения семантических правил с помощью схемы трансляции

Определение. *Семантическое действие* для правила p грамматики – это семантическое правило для p, привязанное к определённой позиции в правой части p.

Пример.

```
Loop ::= 'while' { Expr.SymT := Loop.SymT; } Expr
         'do' { Block.SymT := Loop.SymT; } Block.
```

Семантическое действие разбивает правую часть правила на две подцепочки, которые мы будем называть *префиксом и суффиксом семантического действия*.

Подразумевается, что семантическое действие должно выполняться в тот момент синтаксического анализа, когда полностью распознана часть входной цепочки, соответствующая префиксу этого действия.

Определение. *Схема трансляции* – это синтаксически управляемое определение, семантические правила которого являются семантическими действиями и удовлетворяют условиям:

1. наследуемый атрибут для символа x вычисляется в действии, предшествующем символу x;
2. действие не должно обращаться к синтезируемому атрибуту символа, расположенного справа от действия;
3. синтезируемый атрибут для нетерминала в левой части правила вычисляется в действии, расположенном справа от последнего символа правила.

Пример. Схема трансляции объявлений переменных.

```
D ::= T { L.In := T.Type; } L.
T ::= 'int' { T.Type := integer; }
    | 'float' { T.Type := real; }.
L ::= { L1.In := L.In; } L1 ',' { v.Type := L.In } v
    | { v.Type := L.In } v.
```

§32. Восходящее выполнение S-атрибутных определений

Определение. *S-атрибутное определение* – это синтаксически управляемое определение, в котором применяются только синтезируемые атрибуты.

Синтезируемые атрибуты могут быть вычислены восходящим синтаксическим анализатором в процессе разбора входной цепочки.

K символам грамматики (или состояниям SLR-распознавателя), содержащимся в стеке ПС-анализатора, добавляется дополнительная информация: значения синтезируемых атрибутов.

Возможно выделение специального стека для синтезируемых атрибутов.

При выполнении свёртки по правилу $X \to u$ со стека снимается $|u|$ символов (или состояний), и на основе значений их атрибутов вычисляется значение синтезируемого атрибута символа X.

Пример. Разбор цепочки $3 * (4 + 5)$.

№	Стек	Атрибуты	Вход	Действие
1	$		3*(4+5)$	Перенос
2	$n	3	*(4+5)$	Свёртка по правилу F → n
3	$F	3	*(4+5)$	Свёртка по правилу T → F
4	$T	3	*(4+5)$	Перенос ×3
7	$T*(n	3, , , 4	+5)$	Свёртка по правилу F → n
8	$T*(F	3, , , 4	+5)$	Свёртка по правилу T → F
9	$T*(T	3, , , 4	+5)$	Свёртка по правилу E → T
10	$T*(E	3, , , 4	+5)$	Перенос ×2
12	$T*(E+n	3, , , 4, , 5	)$	Свёртка по правилу F → n
13	$T*(E+F	3, , , 4, , 5	)$	Свёртка по правилу T → F
14	$T*(E+T	3, , , 4, , 5	)$	Свёртка по правилу E → E + T
15	$T*(E	3, , , 9	)$	Перенос
16	$T*(E)	3, , , 9 ,	$	Свёртка по правилу F → (E)
17	$T*F	3, , 9	$	Свёртка по правилу T → T * F
18	$T	27	$	Свёртка по правилу E → T
19	$E	27	$	Допуск