

# Атрибутные грамматики

Золин Иван, Петрошенко Артём

группа 5030102/00201

Санкт-Петербургский политехнический университет Петра Великого

27.11.2023

# Составные части языка и способы их задания

- *Алфавит* - множество допустимых символов языка.
- *Лексика* - описание элементарных конструкций языка.
- *Синтаксис* - описание правил построения предложений из лексем.
- *Семантика* - описание правил приписывания смысла синтаксически правильным конструкциям языка.
- *Терминал* - элемент алфавита, имеющий конкретное значение (буква, цифра, спецсимвол).
- *Нетерминал* - объект, описывающий сущность языка и не имеющий конкретного значения (формула, арифметическое выражение).

# Атрибутные грамматики: идея Кнута

Д. Кнут, *The Genesis of Attribute Grammars* - "история" и изложение идей о наделении КСГ семантикой

Основные идеи в работе:

- *Атрибут* - значение, смысл ("meaning") объекта, т.е. нетерминальных или терминальных символов;
- Значение атрибута может формироваться с учетом значений атрибутов узлов-потомков дерева разбора. Такой атрибут называется *синтезированным*;
- Значение атрибута может также формироваться с учетом значений атрибутов его предков. Такой атрибут называется *унаследованным*.

# Атрибутные грамматики: формализация (1/6)

Д. Кнут, *Semantics of Context-Free Languages* - формальное описание идей Дональда Кнута и Питера Вегнера

**Определение 1:** Пусть  $G = \langle N, T, R, S \rangle$  -

контекстно-свободная грамматика. ( $N$  - набор нетерминалов,  $T$  - набор терминалов,  $R$  - набор правил,  $S$  - стартовый символ,  $S \in N$ )

Введём для каждого терминального и нетерминального символа  $X$  конечное **множество атрибутов**  $A(X)$  ( $A(X)$  может быть пустым): обозначим множество синтезируемых атрибутов как  $A_0(X)$ , а множество унаследованных атрибутов как  $A_1(X)$ , притом:

- ▶  $A_1(S) = \emptyset$
- ▶  $\forall X \in T : A_0(X) = \emptyset$

**Замечание:** Элемент  $A(X)$  обозначим  $\alpha$ . Не стоит её путать с обозначением цепочек. Также принадлежность определенного атрибута  $\alpha$  определенному символу  $X$  обозначим, как  $X.\alpha$

## Атрибутные грамматики: формализация (2/6)

Рассмотрим бинарную нотацию на примере числа 1101.01:

$B \rightarrow 0$

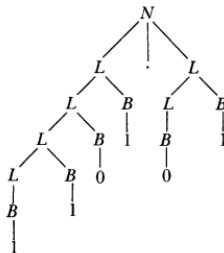
$B \rightarrow 1$

$L \rightarrow B$

$L \rightarrow LB$

$N \rightarrow L$

$N \rightarrow L \cdot L$



Здесь терминалами являются:  $\cdot$  0 1

Нетерминалы:  $B, L, N$

# Атрибутные грамматики: формализация (3/6)

Теперь чтобы нотация имела смысл, нужно, чтобы каждая часть имела смысл. Этого можно добиться, присвоив нетерминалам атрибуты:

$$B \rightarrow 0 \quad v(B) = 0$$

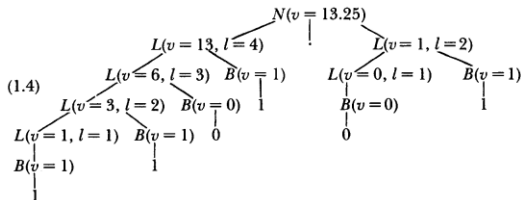
$$B \rightarrow 1 \quad v(B) = 1$$

$$L \rightarrow B \quad v(L) = v(B), \quad l(L) = 1$$

$$L_1 \rightarrow L_2 B \quad v(L_1) = 2v(L_2) + v(B), \quad l(L_1) = l(L_2) + 1$$

$$N \rightarrow L \quad v(N) = v(L)$$

$$N \rightarrow L_1 \cdot L_2 \quad v(N) = v(L_1) + v(L_2)/2^{l(L_2)}$$



*value* - значение, *length* - длина

# Атрибутные грамматики: формализация (4/6)

Мы можем доопределять семантику добавляя новые атрибуты, которые, по нашему мнению, имеют значение. В нашем примере таким атрибутом является позиция:

*Syntactic rules*

$B \rightarrow 0$

$B \rightarrow 1$

$L \rightarrow B$

$L_1 \rightarrow L_2 B$

$N \rightarrow L$

$N \rightarrow L_1 \cdot L_2$

*Semantic rules*

$v(B) = 0$

$v(B) = 2^{v(B)}$

$v(L) = v(B), s(B) = s(L), l(L) = 1$

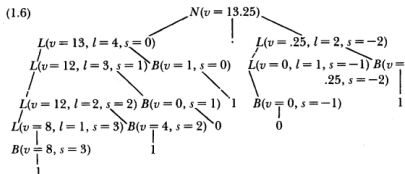
$v(L_1) = v(L_2) + v(B), s(B) = s(L_1),$

$s(L_2) = s(L_1) + 1, l(L_1) = l(L_2) + 1$

$v(N) = v(L), s(L) = 0$

$v(N) = v(L_1) + v(L_2), s(L_1) = 0,$

$s(L_2) = -l(L_2)$



*scale* - позиция

**Замечание:** В предыдущем примере мы имели только синтезированные атрибуты *value* и *length*. Теперь у нас появился унаследованный атрибут *scale*.

## Атрибутные грамматики: формализация (5/6)

Представим каждое  $p$ -ое правило как  $X_0 \rightarrow X_1 X_2 \dots X_{n_p}$ . Свяжем его с соответствующим множеством **семантических правил** из  $F$ , которые имеют форму:

$$X_{p0}.\alpha_i = f_{p,p0,i}(X_{p1}.\alpha_{j_1}, X_{p1}.\alpha_{j_2}, \dots, X_{p1}.\alpha_{j_t}, X_{p2}.\alpha_{j_1}, \dots, X_{pk}.\alpha_{j_t})$$

где:

- ▶  $X_{p0}.\alpha_i$  -  $i$ -ый атрибут символа  $X_{p0}$ :  
 $p0 \in [0..n_p], \alpha_i \in A(X_{p0})$ ;
- ▶  $f_{p,p0,i}$  - функция вычисления  $i$ -ого атрибута символа  $p0$  правила вывода  $p$ ;
- ▶  $p1, \dots, pk$  - индексы терминальных и нетерминальных символов правила  $p$ ;
- ▶  $\alpha_{j_1}, \dots, \alpha_{j_t}$  -  $j_1, \dots, j_t$  атрибуты некоторого символа правой части правила  $p$ ,  $\forall m \in [1..t] : X_{pi}.\alpha_{j_m} \in A(X_{pi})$

**Замечание:** Индексы атрибутов  $j_1, \dots, j_t$  символов правой части не обязаны повторяться от символа к символу! Общая индексация обозначает *выборочность* некоторых атрибутов некоторых символов.



## Атрибутные грамматики: формализация (6/6)

Если над контекстно-свободной грамматикой  $G$  каждому символу сопоставлено множество атрибутов, а каждому правилу вывода - множество семантических правил, то будем называть такую грамматику *атрибутной грамматикой*

$AG = \langle N, T, R, S, A, F \rangle$

- ▶  $A$  - множество атрибутов
- ▶  $F$  - множество семантических правил

*Замечание 1:* В силу введенных определений семантических правил и множества атрибутов, можно говорить, что значение некоторого атрибута  $X.\alpha$  зависит от унаследованных атрибутов символа левой части и от синтезированных атрибутов символов правой части правила вывода.

*Замечание 2:* Для некоторых атрибутов может не быть семантических правил. Например: для атрибутов терминальных символов, которые не зависят от унаследованных атрибутов. Значения таких атрибутов полагаются некоторыми константами.

# Атрибутные грамматики: графы зависимостей (1/3)

**Определение 2:** Пусть задана атрибутная грамматика  $AG$ , выводимая цепочка  $S \Rightarrow^* \omega$ , а также дерево разбора цепочки  $\omega$ . Тогда дерево разбора, с сопоставленными каждой вершине атрибутами называется *атрибутированным деревом разбора*.

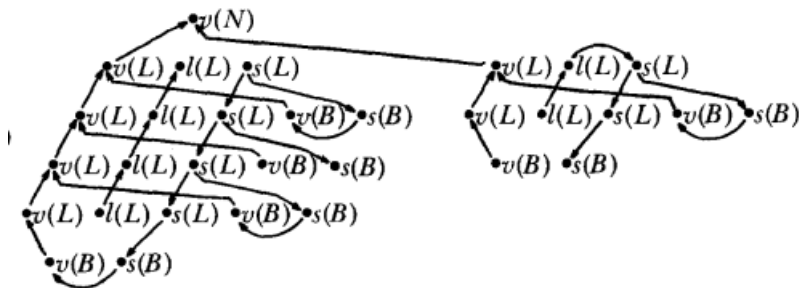
**Определение 3:** Для каждого правила  $p \in R$  определим  $D(p)$  - *граф зависимостей* атрибутов символов, входящих в  $p$ , как ориентированный граф, вершинами которого являются атрибуты символов, входящих в правило вывода  $p$ , и в котором дуга из вершины  $X_{pl}.\alpha_i$  идет в  $X_{pm}.\alpha_j$  тогда и только тогда, когда существует сопоставленное правилу  $p$  семантического правило, в котором:

$$X_{pm}.\alpha_j = f_{p,pm,j}(\dots, X_{pl}.\alpha_i, \dots)$$

**Определение 4:** Граф зависимостей  $D(\omega)$  выводимой цепочки  $S \Rightarrow^* \omega$  определим как ориентированный граф, полученный путем объединения графов зависимостей всех примененных в  $\omega$  правил вывода.

## Атрибутные грамматики: графы зависимостей (2/3)

Пример графа зависимостей:



## Атрибутные грамматики: графы зависимостей (3/3)

*Определение 5:* Атрибутная грамматика называется *незацикленной*, если графы зависимостей деревьев всех цепочек, принадлежащих языку, порождаемому грамматикой  $G$ , не содержат циклов. В противном случае атрибутная грамматика называется *зацикленной*.

*Замечание 1:* Иначе говоря, незацикленная атрибутная грамматика называется *корректной*.

*Замечание 2:* Число всех деревьев всех цепочек, порождаемых грамматикой  $G$  в общем случае бесконечно. Потому важно уметь определять корректность атрибутной грамматики по семантическим зависимостям атрибутов (их число ограничено). Алгоритм проверки на корректность произвольной атрибутной грамматики  $AG = \langle N, T, R, S, A, F \rangle$  предложил Дональд Кнут.

# Атрибутные грамматики: разновидности грамматик

Существуют дополнительные классы атрибутных грамматик:

- ▶ *S-атрибутные грамматики* - грамматики, в которых все атрибуты являются синтезированными;
- ▶ *L-атрибутные грамматики* - грамматики, в которых все наследуемые атрибуты зависят только от наследуемых атрибутов предка и/или от любых атрибутов левых братьев.

Данные грамматики обладают рядом полезных свойств:

- ▶ Всякая S-атрибутная грамматика является L-атрибутной, поскольку в S-атрибутной грамматике нет наследуемых атрибутов;
- ▶ В L-атрибутной грамматике все атрибуты могут быть вычислены за один стандартный обход дерева вывода в глубину.

# Регулярная форма Бэкуса-Наура

## TERMINALS:

number ::= '[1-9]\d\*';

operation ::= '[\+ \\*]';

terminator ::= ', '.

KEYS: '+'; '\*'; ', '.

## NONTERMINALS:

EXPRESSIONS;

EXPRESSION;

TERM.

AXIOM: EXPRESSIONS.

## RULES:

EXPRESSIONS ::= EXPRESSION # , ;

EXPRESSION ::= TERM # + ;

TERM ::= number # \* .

# Диаграмма Вирта

*Синтаксическая диаграмма Вирта* - особый вид орграфов, предназначенных для записи контекстно-свободных грамматик в графической форме.

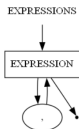


Рис.: Выражения

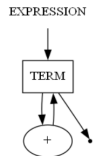


Рис.: Выражение

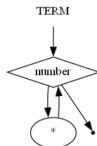


Рис.: Слагаемое

# Калькулятор (1/3)

Пусть **грамматика простых арифметических выражений** таких как сложение и умножение имеет следующий вид:

$$G : E \rightarrow E + T \mid T$$

$$T \rightarrow T \times F \mid F$$

$$F \rightarrow (E) \mid x$$

Семантические правила атрибутной грамматики задаются правилами арифметической операций. В качестве атрибутов нетерминалов  $E, T, F$  выбираем один. Для нетерминала  $N$  параметр  $N.val$  пусть означает его значение.

*Замечание:* Здесь и далее продукцией называется синтаксическое правило.



## Калькулятор (2/3)

Номер	Продукция	Семантическое правило
1	$E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
2	$E \rightarrow T$	$E.val = T.val$
3	$T \rightarrow T_1 \times F$	$T.val = T_1.val \times F.val$
4	$T \rightarrow F$	$T.val = F.val$
5	$F \rightarrow (E)$	$F.val = E.val$
6	$F \rightarrow x$	$F.val = x.val$

Таблица: Атрибутная грамматика для арифметических операций

## Калькулятор (3/3)

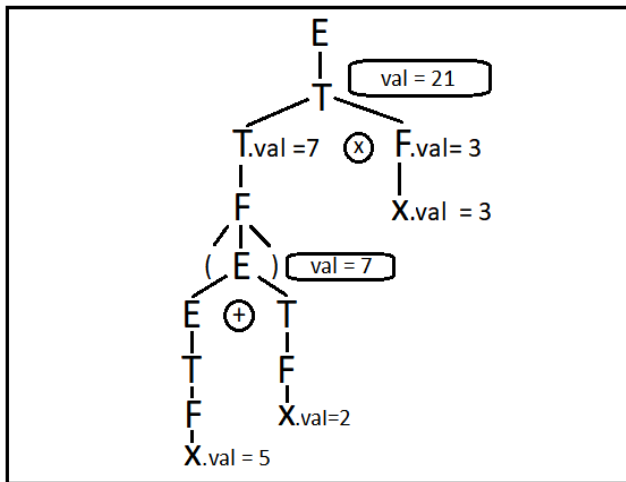


Рис.: Дерево вывода цепочек  $(5 + 2) \times 3$  и значение семантического атрибута *val*

# Символьное дифференцирование (1/4)

Пусть **грамматика арифметических выражений** имеет следующий вид:

$$G : E \rightarrow E + T \mid T$$

$$T \rightarrow T \times P \mid P$$

$$P \rightarrow \text{Sin}(E) \mid \text{Cos}(E) \mid \text{x} \mid \text{C}(E)$$

Правила, по которым производится дифференцирование и задают семантические правила атрибутивной грамматики, порождающей аналитические записи функций. В качестве атрибутов нетерминалов  $E, T, P$  выбираем два. Для нетерминала  $N$  параметр  $N.\psi$  пусть означает цепочку терминальных символов (подвыражение), выведенную из этого нетерминала  $N$  параметр  $N.\phi$  – производную этого подвыражения.

## Символьное дифференцирование (2/4)

№	Продукции	Семантические правила
1	$E_0 \rightarrow E_1 + T$	$E_0.\varphi = E_1.\varphi + T.\varphi;$ <span style="float: right;"><math>E_0.\psi = E_1.\psi + T.\psi</math></span>
2	$E \rightarrow T$	$E.\varphi = T.\varphi;$ <span style="float: right;"><math>E.\psi = T.\psi</math></span>
3	$T_0 \rightarrow T_1 P$	$T_0.\varphi = T_1.\varphi P.\psi + T_1.\psi P.\varphi;$ <span style="float: right;"><math>T_0.\psi = T_1.\psi P.\psi</math></span>
4	$T \rightarrow P$	$T.\varphi = P.\varphi;$ <span style="float: right;"><math>T.\psi = P.\psi</math></span>
5	$P \rightarrow \sin(E)$	$P.\varphi = \cos(E.\psi) (E.\varphi);$ <span style="float: right;"><math>P.\psi = \sin(E.\psi)</math></span>
6	$P \rightarrow \cos(E)$	$P.\varphi = -\sin(E.\psi) (E.\varphi);$ <span style="float: right;"><math>P.\psi = \cos(E.\psi)</math></span>
7	$P \rightarrow x$	$P.\varphi = 1;$ <span style="float: right;"><math>P.\psi = x</math></span>
8	$P \rightarrow C$	$P.\varphi = 0;$ <span style="float: right;"><math>P.\psi = C</math></span>
9	$P \rightarrow (E)$	$P.\varphi = E.\varphi;$ <span style="float: right;"><math>P.\psi = E.\psi</math></span>

Рис.: Атрибутная грамматика для символьного дифференцирования

## Символьное дифференцирование (3/4)

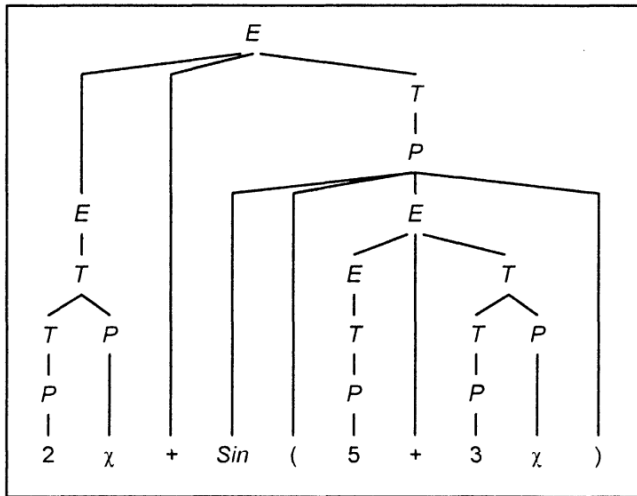


Рис.: Дерево вывода цепочек  $2x + \text{Sin}(5 + 3x)$

## Символьное дифференцирование (4/4)

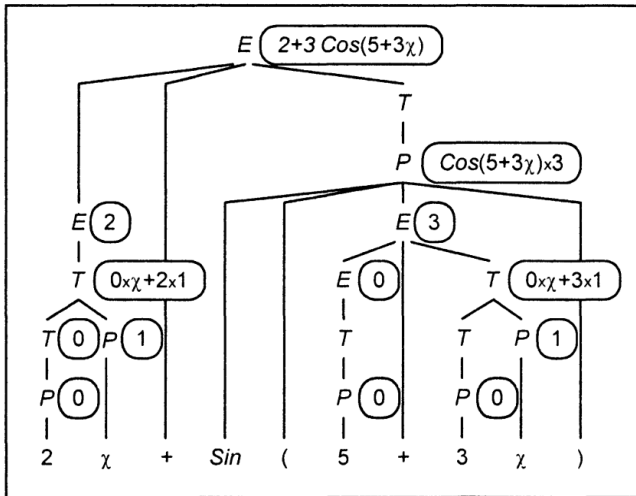
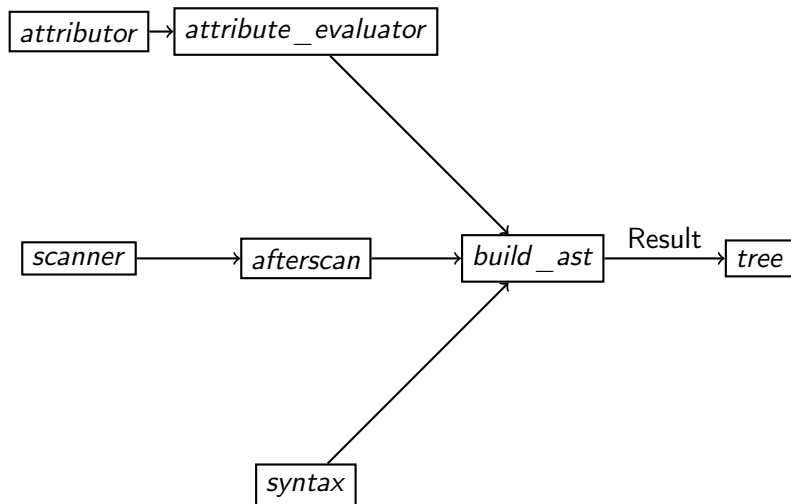


Рис.: Значение семантического атрибута  $\phi$

# Описание работы программы



Исходный код:

[https://github.com/aVorotnikov/dsl\\_generator](https://github.com/aVorotnikov/dsl_generator)

# Список литературы

1. Д. Кнут "The Genesis of Attribute Grammars"
2. Д. Кнут "Semantics of Context-Free Languages"
3. Ю.Г. Карпов, "Основы построения трансляторов 2005г., стр.91-97; стр.126-130"
4. А.А. Воротников, Ф.А. Новиков "ВКР Учебное инструментальное средство задания синтаксиса и семантики предметно ориентированных языков"