

# Теория языков программирования и методы трансляции

## Часть 8

- Восходящие методы грамматического анализа
- Грамматики предшествования

Гришмановский Павел Валерьевич  
доцент кафедры автоматики и компьютерных систем, к.т.н., доцент

Сургут, 2018

# Грамматики предшествования

- **Грамматиками предшествования** называются такие грамматики, в которых можно однозначным образом установить отношения предшествования для пары любых символов алфавита грамматики
- **Связка (основа)** – подстрока символов, которая совпадает с правой частью одного из правил (т.е. может быть выполнена ее **свертка** к нетерминальному символу в левой части этого правила)
- **Свертка** – операция, обратная шагу вывода (чтение производится слева направо и отыскивается самая левая связка, следовательно, процесс восходящего анализа восстанавливает правосторонний вывод в обратном порядке)
- **Основная задача** при построении транслятора – определение отношений предшествования, необходимых для выявление границ связок

В зависимости от видов отношений, категорий символов, между которыми они устанавливаются, и ограничений на правила

- грамматики **простого** предшествования
- грамматики **слабого** предшествования
- грамматики **операторного** предшествования
- грамматики **расширенного** предшествования
- грамматики **смешанной стратегии**

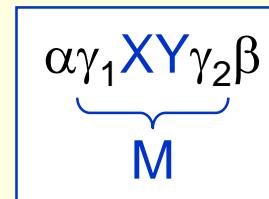
# Грамматики предшествования

Отношение предшествования может быть установлено для пары любых символов, если они могут появиться рядом в какой-либо сентенциальной форме

$$X, Y: S \Rightarrow^* \alpha \gamma_1 XY \gamma_2 \beta, X, Y \in V, \alpha, \gamma_1, \gamma_2, \beta \in V^*$$

- Символы X и Y выводятся одновременно

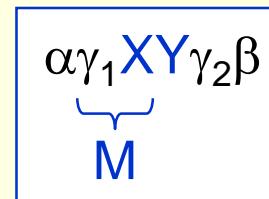
$$\begin{aligned} S &\Rightarrow^* \alpha M \beta \\ M &\rightarrow \gamma_1 X Y \gamma_2 \end{aligned}$$



X и Y сворачиваются одновременно

- Один из символов X и Y выводится позже другого

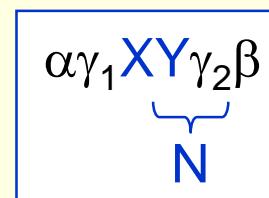
$$\begin{aligned} S &\Rightarrow^* \alpha M Y \gamma_2 \beta \\ M &\rightarrow \gamma_1 X \end{aligned}$$



X сворачивается раньше Y  
 $Y \in V^T$

- Символы X и Y выводятся в разное время

$$\begin{aligned} S &\Rightarrow^* \alpha M N \beta \\ M &\rightarrow \gamma_1 X \\ N &\rightarrow Y \gamma_2 \end{aligned}$$



X сворачивается позже Y  
(Y раньше X)

Правосторонний вывод,  
X раньше Y

# Грамматики предшествования

Распознаватель представляет собой МП-автомат, реализующий **алгоритм «сдвиг-свертка»**. Основа алгоритма:

- Функция  $f(X, Y): V \times V \rightarrow \{ \text{СДВИГ}, \text{СВЕРТКА}, \text{СТОП}, \text{ОШИБКА} \}$ ,  
где  $X \in V$  – символ на вершине стека,  $Y \in V$  – текущий считанный символ
- Функция  $g(\omega): V^+ \rightarrow \{ 1, 2, \dots, p, \text{ОШИБКА} \}$ ,  
где  $\omega \in V^+$  – состояние стека,  $1, 2, \dots, p$  – номера всех правил грамматики

Для каждого считанного символа  $Y$  выполняется:

- Если отношение между символом  $X$  на вершине стека и считанным символом  $Y$  не задано, то  $f(X, Y) = \text{ОШИБКА}$  (работа распознавателя прекращается, а на практике также если  $Y \notin V^T$ )
- Если отношение между символами  $X$  и  $Y$  определено как «предшествует основе» или «составляет основу», то  $f(X, Y) = \text{СДВИГ}$  ( $Y$  помещается в стек)
- Если отношение между символами  $X$  и  $Y$  определено как «следует за основой», то  $f(X, Y) = \text{СВЕРТКА}$ 
  - если  $g(\omega)$  дает номер правила, то выполняется **свертка**: из стека извлекается правая часть  $\omega$ , которая совпадает с правой частью данного правила, и заменяется на символ левой части этого правила (он помещается в стек)
  - если  $g(\omega) = \text{ОШИБКА}$ , то работа распознавателя прекращается
- Если  $X = S$  и  $Y = \perp$ , то  $f(X, Y) = \text{СТОП}$  (работа распознавателя завершается успешным разбором входной цепочки)

# Грамматика простого предшествования

Грамматикой простого предшествования называется такая грамматика, которая удовлетворяет следующим условиям:

1. Является контекстно-свободной  $\lambda$ -свободной грамматикой
2. Множество правил не содержит двух правил с одинаковыми правыми частями
3. Для  $\forall X, Y \in V$  определено не более одного из трех отношений предшествования

$X =\bullet Y$  – составляют основу, если  $(A \rightarrow \alpha XY\beta) \in P$ , где  
 $A \in V^N, \alpha, \beta \in V^*$  (непосредственно соседствуют в правой части правила, свертка  $X$  и  $Y$  происходит одновременно)

$X <\bullet Y$  – предшествует основе (предшествует), если  
 $(A \rightarrow \alpha XM\beta) \in P$  и  $\exists M \Rightarrow^+ Y\gamma$ , где  $A, M \in V^N, \alpha, \beta, \gamma \in V^*$   
(свертка  $X$  к  $A$  происходит позже, чем  $Y$  к  $M$ )

$X \bullet> Y$  – следует за основой (следует), если  
 $(A \rightarrow \alpha MN\beta) \in P$  и  $\exists M \Rightarrow^+ \gamma X$  и  $\exists N \Rightarrow^+ Y\delta$ , где  $A, M, N \in V^N, \alpha, \beta, \gamma, \delta \in V^*$  (свертка  $X$  к  $M$  происходит раньше, чем  $Y$  к  $N$ )

# Грамматика простого предшествования (пример)

Множество правил  $P$  грамматики  $G(V^N, V^T, S, P)$ :

|                                |  |
|--------------------------------|--|
| $S \rightarrow I=E;$           | $I \rightarrow A   AK$   |
| $E \rightarrow E+T   E-T   T$  | $K \rightarrow A   AK   D   DK$                                |
| $T \rightarrow T^*M   T/M   M$ | $C \rightarrow D   CD$   |
| $M \rightarrow (E)   I   C$    | $A \rightarrow a   b   c   d   e   f   \dots   x   y   z   \_$ |
|                                | $D \rightarrow 0   1   2   3   4   5   6   7   8   9$          |

Есть совпадающие правые части правил:  $A, AK, D$

Распознаватель строится на основе видоизмененной грамматики  $G'(V^N', V^T', S', P')$ :

$$V^N' = \{ S', S, E, T, M \},$$

$$V^T' = \{ =, ;, +, -, *, /, (, ), I, C, \_ \},$$

$$P' = \{ S' \rightarrow \underline{\_} S \underline{\_}$$

$$S \rightarrow I=E;$$

$$E \rightarrow E+T | E-T | T$$

$$T \rightarrow T^*M | T/M | M$$

$$M \rightarrow (E) | I | C \}$$

- Символы  $I$  и  $C$  соответствуют лексемам – будем считать их терминальными (для разбора лексем используем сканер)
- Грамматика дополнена символом-ограничителем  $\underline{\_}$

# Грамматика простого предшествования (пример)

Определение отношения  $X =\bullet Y$

$(A \rightarrow \alpha X Y \beta) \in P, A \in V^N, \forall X, Y \in V, \alpha, \beta \in V^*$

Все пары символов, стоящих рядом в правых частях правил (из всех правил грамматики)

|                                      |                                  |
|--------------------------------------|----------------------------------|
| $S' \rightarrow \perp S \perp$       | $\perp S \quad S \perp$          |
| $S \rightarrow I=E;$                 | $I= \quad =E \quad E;$           |
| $E \rightarrow E+T \mid E-T \mid T$  | $E+ \quad +T \quad E- \quad -T$  |
| $T \rightarrow T^*M \mid T/M \mid M$ | $T^* \quad *M \quad T/ \quad /M$ |
| $M \rightarrow (E) \mid I \mid C$    | $(E \quad E)$                    |

# Грамматика простого предшествования (пример с ошибкой;)

Определение отношения  $X <\bullet Y$

$(A \rightarrow \alpha X M \beta) \in P$  и  $\exists M \Rightarrow^+ Y \gamma$ , где  $A, M \in V^N$ ,  $\alpha, \beta, \gamma \in V^*$

Все такие пары символов, в которых  $Y$  является первым в цепочке, выводимой из символа  $M$ , стоящего рядом с  $X$  в правой части правила (из всех правил грамматики, рекурсивно)

|   |                |                                   |
|---|----------------|-----------------------------------|
| $S' \rightarrow \perp S \perp$          | $\perp S$      | $\perp I$                         |
| $S \rightarrow I = E ;$                 | $= E$          | $= T = M = ( = I = C$             |
| $E \rightarrow E + T \mid E - T \mid T$ | $+ T \mid - T$ | $+ M + ( + I + C - M - ( - I - C$ |
| $T \rightarrow T * M \mid T / M \mid M$ | $* M \mid / M$ | $* ( * I * C / ( / I / C$         |
| $M \rightarrow ( E ) \mid I \mid C$     | $( E$          | $( T ( M ( ( I ( C$               |

1. Из « $=\bullet$ » выписываются все пары с **нетерминальным символом справа**

2. Подставляются **первые** символы правых частей правил, если такой символ нетерминальный, то подстановка выполняется относительно него и т.д.

# Грамматика простого предшествования (пример с ошибкой;)

Определение отношения  $X \bullet> Y$

$(A \rightarrow \alpha M N \beta) \in P$  и  $\exists M \Rightarrow^+ \gamma X$  и  $\exists N \Rightarrow^+ Y \delta$ , где  $A, M, N \in V^N$ ,  $\alpha, \beta, \gamma, \delta \in V^*$

Все такие пары символов, в которых  $X$  и  $Y$  являются крайними (первым и последним) в цепочках, выводимых из символов  $N$  и  $M$ , стоящих рядом в правой части правила (все правила, рекурсивно)

|   |                           |   |
|---|---------------------------|---|
| $S' \rightarrow \perp S \perp$          | $S \perp$                 | $\bullet; \perp$                              |
| $S \rightarrow I = E ;$                 | $E ;$                     | $T ; M ; ) ; I ; C ;$                         |
| $E \rightarrow E + T \mid E - T \mid T$ | $E + \quad E -$           | $T + M + ) + I + C + T - M - ) - I - C -$     |
| $T \rightarrow T * M \mid T / M \mid M$ | $T * \quad T / \quad M *$ | $I * \quad C * \quad M / ) / I / C /$         |
| $M \rightarrow ( E ) \mid I \mid C$     | $E )$                     | $T ) \quad M ) \quad ) ) \quad I ) \quad C )$ |

1. Из « $=\bullet$ » выписываются все пары с **нетерминальным символом слева**
2. Если правый символ нетерминальный (в данном примере таких нет), то аналогично « $<\bullet$ ». Эти пары являются исходными, но в результат не входят
3. Подставляются последние символы правых частей правил (рекурсивно)
4. Пары с нетерминальным символом справа исключаются

# Грамматика простого предшествования (пример с ошибкой;)

Отношения фиксируются в матрице предшествования

|   | S | E | T | M | = | ; | + | - | * | / | ( | ) | I | C | L |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S |   |   |   |   |   |   |   |   |   |   |   |   |   |   | = |   |
| E |   |   |   |   |   | = | = | = |   |   |   |   |   | = |   |   |
| T |   |   |   |   |   | > | > | > | = | = |   |   |   | > |   |   |
| M |   |   |   |   |   | > | > | > | > | > |   |   |   | > |   |   |
| = |   |   |   |   | = | < | < |   |   |   |   |   |   | < | < | < |
| ; |   |   |   |   |   |   |   |   |   |   |   |   |   |   | > |   |
| + |   |   |   |   |   |   | = | < |   |   |   |   |   | < | < | < |
| - |   |   |   |   |   |   | = | < |   |   |   |   |   | < | < | < |
| * |   |   |   |   |   |   |   | = |   |   |   |   |   | < | < | < |
| / |   |   |   |   |   |   |   | = |   |   |   |   |   | < | < | < |
| ( |   |   |   |   |   |   |   | = | < | < |   |   |   | < | < | < |
| ) |   |   |   |   |   |   |   |   | > | > | > | > | > | > |   |   |
| I |   |   |   |   |   |   |   |   | = | > | > | > | > | > |   |   |
| C |   |   |   |   |   |   |   |   | > | > | > | > | > | > |   |   |
| L |   |   |   |   | = |   |   |   |   |   |   |   |   |   | < |   |

Должно  
быть <=

- Отношения не должны пересекаться!
- Данная грамматика является грамматикой простого предшествования

# Грамматика простого предшествования (пример)

## Правила грамматики

$$\begin{aligned} S &\rightarrow I=E; \\ E &\rightarrow E+T \mid E-T \mid T \\ T &\rightarrow T^*M \mid T/M \mid M \\ M &\rightarrow (E) \mid I \mid C \end{aligned}$$

При свертке определяются отношения между:

1. символом, оставшимся на вершине стека ("="), и помещаемым в стек нетерминальным символом (M)
2. помещенным в стек (M) и считанным (";") символами

## Протокол грамматического разбора

- входная цепочка: `cnt = 123;`
- цепочка лексем: `I=C ; ⊥`

| Стек ( $\omega$ ) | Y                  | Действие                      |
|-------------------|--------------------|-------------------------------|
| $\perp$           | $<I$               | Сдвиг                         |
| $\perp<I$         | $=$                | Сдвиг                         |
| $\perp<I=$        | $<C$               | Сдвиг                         |
| $\perp<I==$       | $\underbrace{<C>}$ | Свертка $M \leftarrow C$      |
| $\perp<I==$       | $\underbrace{<M>}$ | Свертка $T \leftarrow M$      |
| $\perp<I==$       | $\underbrace{<T>}$ | Свертка $E \leftarrow T$      |
| $\perp<I==$       | $=$                | Сдвиг                         |
| $\perp<I==$       | $=$                | Свертка $S \leftarrow I=E;$   |
| $\perp=S$         | $=\perp$           | <b>Стоп</b> (цепочка принята) |

Пример разбора соответствует приведенной матрице предшествования

# Грамматика простого предшествования (пример)

## Правила грамматики

$S \rightarrow I = E;$   
 $E \rightarrow E + T \mid E - T \mid T$   
 $T \rightarrow T^* M \mid T / M \mid M$   
 $M \rightarrow (E) \mid I \mid C$

## Ошибка разбора (1)

- входная цепочка:  $cnt = +123;$
- цепочка лексем:  $I = + C ; \perp$

| Стек ( $\omega$ ) | Y     | Действие |
|-------------------|-------|----------|
| $\perp$           | $< I$ | Сдвиг    |
| $\perp < I$       | $=$   | Сдвиг    |
| $\perp < I =$     | $? +$ | Ошибка!  |

Между символом на вершине стека ("=") и считанным символом ("+")  
**отношение не определено**, что не позволяет поместить в стек  
считанный символ

Также возможно чтение символа, не принадлежащего алфавиту языка.  
Такую ошибку обнаружит сканер, т.к. окажется невозможна распознать  
очередную лексему

Пример разбора соответствует приведенной матрице предшествования

# Грамматика простого предшествования (пример)

## Правила грамматики

$$\begin{aligned} S &\rightarrow I=E; \\ E &\rightarrow E+T \mid E-T \mid T \\ T &\rightarrow T^*M \mid T/M \mid M \\ M &\rightarrow (E) \mid I \mid C \end{aligned}$$

Отношения между символами позволяют поместить в стек неправильную цепочку

При свертке обнаружено, что **отношение не определено** между символом, оставшимся на вершине стека ("="), и нетерминальным символом, помещаемым в стек (S)

Также возможна ситуация, когда не найдено правило, соответствующее цепочке на вершине стека

## Ошибка разбора (2)

- входная цепочка:  $\text{cnt} = x = 123;$
- цепочка лексем:  $I=I=C; \perp$

| Стек ( $\omega$ ) | Y        | Действие                    |
|-------------------|----------|-----------------------------|
| $\perp$           | $<I$     | Сдвиг                       |
| $\perp<I$         | $=$      | Сдвиг                       |
| $\perp<I=$        | $<I$     | Сдвиг                       |
| $\perp<I==<I$     | $=$      | Сдвиг                       |
| $\perp<I==<I==$   | $<C$     | Сдвиг                       |
| $\perp<I==<I==<C$ | $>$      | Свертка $M \leftarrow C$    |
| $\perp<I==<I==<M$ | $>$      | Свертка $T \leftarrow M$    |
| $\perp<I==<I==<T$ | $>$      | Свертка $E \leftarrow T$    |
| $\perp<I==<I==E$  | $=$      | Сдвиг                       |
| $\perp<I==<I==E$  | $>$      | Свертка $S \leftarrow I=E;$ |
| $\perp<I==?S$     | $=\perp$ | Ошибка!                     |

Пример разбора соответствует приведенной матрице предшествования

# Грамматика слабого предшествования

Определяется аналогично грамматике простого предшествования, за исключением следующего:

- Для  $\forall X, Y \in V$  может быть определено как единственное из трех отношений предшествования, так и одновременно «предшествует основе» и «составляет основу»

Правая часть одного правила может являться суффиксом правой части другого правила

- Если одновременно  $\exists(A \rightarrow \alpha XY\beta) \in P$ ,  $\exists(B \rightarrow \delta XM\gamma) \in P$ ,  $\exists M \Rightarrow^* Y\beta$ , где  $A, B, M \in V^N$ ,  $\alpha, \beta, \gamma, \delta \in V^*$ , то сентенциальная форма может содержать подцепочки  $\alpha XY\beta$  и  $\delta XY\beta\gamma$ , т.е. часть основы  $\alpha XY\beta$  совпадает с основой  $Y\beta$

Функция  $g(\omega)$  должна возвращать номер того правила, правая часть которого имеет максимальную длину из всех правил, правые части которых совпадают с частью на вершине стека

- Целесообразно упорядочить правила грамматики по убыванию длин их правых частей и начинать сопоставление с больших по длине
- В отличие от распознавателя грамматики простого предшествования, извлечение основы из стека следует выполнять только после определения номера правила ( $g(\omega) \neq \text{ОШИБКА}$ ), а не сразу, как только найдено отношение « $\bullet >$ » ( $f(X, Y) = \text{СВЕРТКА}$ ) между считанным символом  $Y$  и вершиной стека  $X$ .

# Грамматика слабого предшествования (пример)

Распознаватель строится на основе грамматики

$G'(V^N', V^T', S', P')$ :

$V^N' = \{ S', S, E, T, M \}$ ,

$V^T' = \{ =, ;, +, -, *, /, (, ), I, C, \perp \}$ ,

$P' = \{ S' \rightarrow \perp S \perp$

$S \rightarrow I = E;$

$E \rightarrow E+T \mid E-T \mid T$

$T \rightarrow T^*M \mid T/M \mid M$

$M \rightarrow (E) \mid -M \mid I \mid C \}$

Добавлено правило  $M \rightarrow -M$   
(см. пример «Грамматики  
простого предшествования»)

Найдены следующие отношения (дополнительно к примеру выше)

|                  |  |                               |
|------------------|--|-------------------------------|
| $X = \bullet Y:$ | $M \rightarrow -M$   | $-M$                          |
| $X < \bullet Y:$ | $E \rightarrow E+T \mid E-T \mid T$<br>$T \rightarrow T^*M \mid T/M \mid M$<br>$M \rightarrow (E)$ | $+ - -$<br>$* - / -$<br>$( -$ |

Для пары «-M» ранее также было определено отношение «<•»  
(см. пример «Грамматики простого предшествования»)

# Грамматика слабого предшествования (пример)

Отношения фиксируются в матрице предшествования

|   | S | E  | T  | M  | =  | ;  | + | - | * | / | ( | ) | I | C | ⊥ |
|---|---|----|----|----|----|----|---|---|---|---|---|---|---|---|---|
| S |   |    |    |    |    |    |   |   |   |   |   |   |   |   | = |
| E |   |    |    |    |    | =  | = | = |   |   |   |   |   | = |   |
| T |   |    |    |    |    | >  | > | > | = | = |   |   |   | > |   |
| M |   |    |    |    |    | >  | > | > | > | > |   |   |   | > |   |
| = |   | <= | <  | <  |    |    | < |   |   | < |   |   | < | < |   |
| ; |   |    |    |    |    |    |   |   |   |   |   |   |   |   | > |
| + |   |    | <= | <  |    |    | < |   |   | < |   |   | < | < |   |
| - |   |    |    | <= | <= |    |   | < |   |   | < |   | < | < |   |
| * |   |    |    |    | =  |    |   | < |   |   | < |   | < | < |   |
| / |   |    |    |    |    | =  |   |   | < |   |   | < |   | < |   |
| ( |   |    |    |    |    | <= | < | < |   |   | < |   | < | < |   |
| ) |   |    |    |    |    |    |   | > | > | > | > | > |   | > |   |
| I |   |    |    |    |    |    |   | = | > | > | > | > | > |   |   |
| C |   |    |    |    |    |    |   | > | > | > | > | > |   |   |   |
| ⊥ |   |    |    |    |    |    |   |   |   |   |   |   |   |   | < |

- Имеет место пересечение отношений
- Данная грамматика является грамматикой слабого предшествования

# Грамматика слабого предшествования (пример)

## Правила грамматики

$$\begin{aligned}S &\rightarrow I=E; \\E &\rightarrow E+T \mid E-T \mid T \\T &\rightarrow T^*M \mid T/M \mid M \\M &\rightarrow (E) \mid -M \mid I \mid C\end{aligned}$$

Свертка дает символ  $M$ ,  
который должен быть  
помещен в стек. Между " $-$ " и  $M$   
определенны одновременно  
отношения " $<$ " и " $=$ ". В стек  
помещаются оба или признак,  
обозначающий их сочетание

На вершине стека обнаружены одновременно  
две основы " $M$ " и " $-M$ ". Возможна свертка по  
правилам  $T \rightarrow M$  и  $M \rightarrow -M$ . Выбирается  
правило, имеющее большую длину правой  
части (из альтернатив " $<$ " и " $=$ " выбрана " $=$ ")

## Фрагмент грамматического разбора

- входная цепочка содержит унарный «-»

| Стек ( $\omega$ ) | Y           | Действие                  |
|-------------------|-------------|---------------------------|
| ...               | ( $<-$      | Сдвиг                     |
| ...               | $<-$ $<C$   | Сдвиг                     |
| ...               | $<-<C$ $>$  | Свертка $M \leftarrow C$  |
| ...               | $<-<=M$ $>$ | Свертка $M \leftarrow -M$ |
| ...               | $<M$ $>$    | Свертка $T \leftarrow M$  |
| ...               | $<T$ $>$    | Свертка $E \leftarrow T$  |

Существует единственное  
подходящее правило  $T \rightarrow M$

# Грамматика операторного предшествования

Грамматикой операторного предшествования называется такая грамматика, которая удовлетворяет следующим условиям:

1. Является контекстно-свободной  $\lambda$ -свободной грамматикой
2. Множество правил не содержит правил со смежными нетерминальными символами в правой части

$$\forall A, B, C \in V^N: (A \rightarrow \alpha BC\beta) \notin P, \alpha, \beta \in V^*$$

3. Для  $\forall x, y \in V^T$  определено не более одного из трех отношений предшествования

$X = \bullet Y$  – составляют основу, если  $(A \rightarrow \alpha xy\beta) \in P$  или  $(A \rightarrow \alpha xBy\beta) \in P$ , где  $A, B \in V^N, \alpha, \beta \in V^*$

$X < \bullet Y$  – предшествует основе (предшествует), если  $(A \rightarrow \alpha xB\beta) \in P$  и  $\exists (B \Rightarrow^* y\gamma \text{ или } B \Rightarrow^* Cy\gamma)$ , где  $A, B, C \in V^N, \alpha, \beta, \gamma \in V^*$

$X \bullet > Y$  – следует за основой (следует), если  $(A \rightarrow \alpha By\beta) \in P$  и  $\exists (B \Rightarrow^* \gamma x \text{ или } B \Rightarrow^* \gamma xC)$ , где  $A, B, C \in V^N, \alpha, \beta, \gamma \in V^*$

# Грамматика операторного предшествования

Матрица предшествования для грамматики операторного предшествования более компактна, чем для грамматики простого или слабого предшествования, т.к. строится только для терминальных символов, но класс грамматик операторного предшествования уже, чем простого.

Построение распознавателя аналогично алгоритмам простого или слабого предшествования с учетом:

- при чтении очередного символа  $Y$  (терминального) он сравнивается с символом  $X$  – **ближайшим к вершине стека терминальным символом** (нетерминальные символы во внимание не принимаются)
- после отыскания основы выполняется свертка (с учетом содержащихся в ней нетерминальных символов) и полученный нетерминальный символ помещается в стек **без указания отношения** (или с отношением «составляет основу» - это упрощает практическую реализацию)

# Атрибутные трансляции

**Атрибутная трансляция** – это семантически ориентированная трансляция на основе **атрибутной грамматики**, в которой для каждого символа определены атрибуты (**числовые или нечисловые**), а каждое синтаксическое правило грамматики дополнено одним или несколькими **семантическими правилами**

**Семантическое правило** вычисляет значение атрибута символа, входящего в синтаксическое правило, как функцию атрибутов любых символов, входящих в то же правило:

- при выборе правила для дальнейшего анализа
- в момент **свертки** подцепочки к нетерминальному символу

Атрибуты могут нести **любую** семантическую информацию:

- вычисленное значение;
- тип значения (**константа, значение выражения, идентификатор и т.п.**);
- область определения (**локальная, глобальная, класс и т.п.**);
- эквивалентные фрагменты генерируемого кода и др.

Для атрибутов должна быть задана **область определения значений** (**целое, вещественное, строка, множество значений и т.п.**)

Значения атрибутов используются для интерпретации программы или генерации программы на другом языке

# Атрибутные трансляции

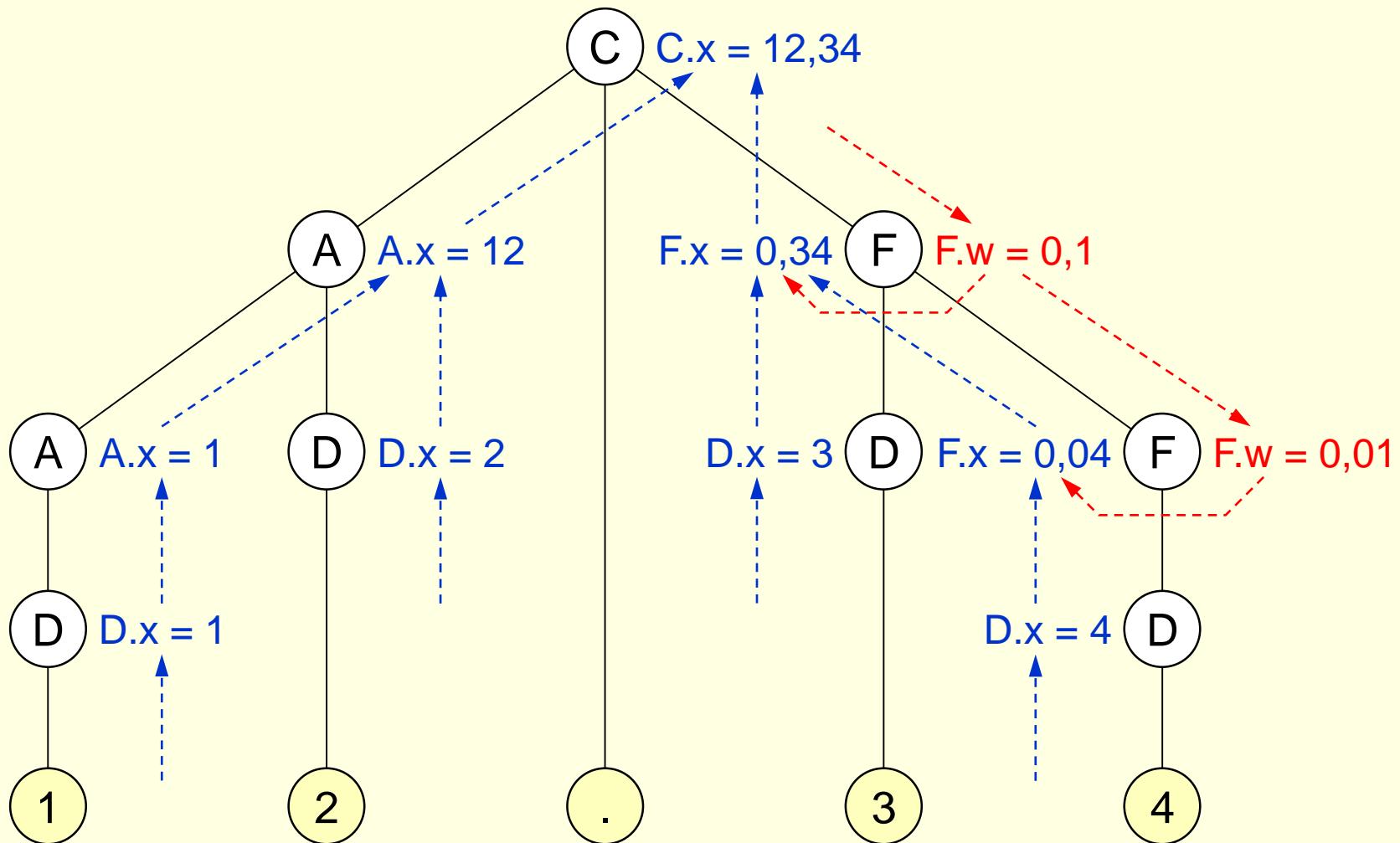
- **Синтезированный атрибут** – атрибут символа в **левой** части правила, значение которого вычисляется как функция от значений атрибутов символов правой части правила  
**(информация передается вверх по дереву грамматического разбора)**
- **Унаследованный атрибут** – атрибут символа в **правой** части правила, значение которого вычисляется как функция от значений атрибутов символов как в левой, так и в правой частях правила  
**(информация передается вниз по дереву грамматического разбора или между соседними узлами)**
- Семантическое правило может быть записано в виде выражения (например, формулы), псевдокода и т.п.
- Если один нетерминальный символ встречается неоднократно в одном правиле грамматики, то в семантическом правиле этому символу присваивается индекс в соответствии с порядковым номером его вхождения в правой части (без индекса – в левой)

# Атрибутные трансляции (пример 1)

| Синтаксические правила   | Семантические правила                          |                         |
|--|--|-------------------------|
|  | Синтезированные атрибуты                       | Унаследованные атрибуты |
| $C \rightarrow A$<br>$C \rightarrow A.F$                           | $C.x = A.x$<br>$C.x = A.x + F.x$               | $F.w = 0,1$             |
| $A \rightarrow A$<br>$A \rightarrow AD$                            | $A.x = D.x$<br>$A.x = A_1.x * 10 + D.x$        |                         |
| $F \rightarrow D$<br>$F \rightarrow DF$                            | $F.x = D.x * F.w$<br>$F.x = F_1.x + D.x * F.w$ | $F_1.w = F.w * 0,1$     |
| $D \rightarrow 0$<br>$D \rightarrow 1$<br>...<br>$D \rightarrow 9$ | $D.x = 0$<br>$D.x = 1$<br>...<br>$D.x = 9$     |                         |

Атрибуты:  $x$  – значение константы или ее части  
 $w$  – множитель разряда («вес» цифры)

# Атрибутные трансляции (пример 1)



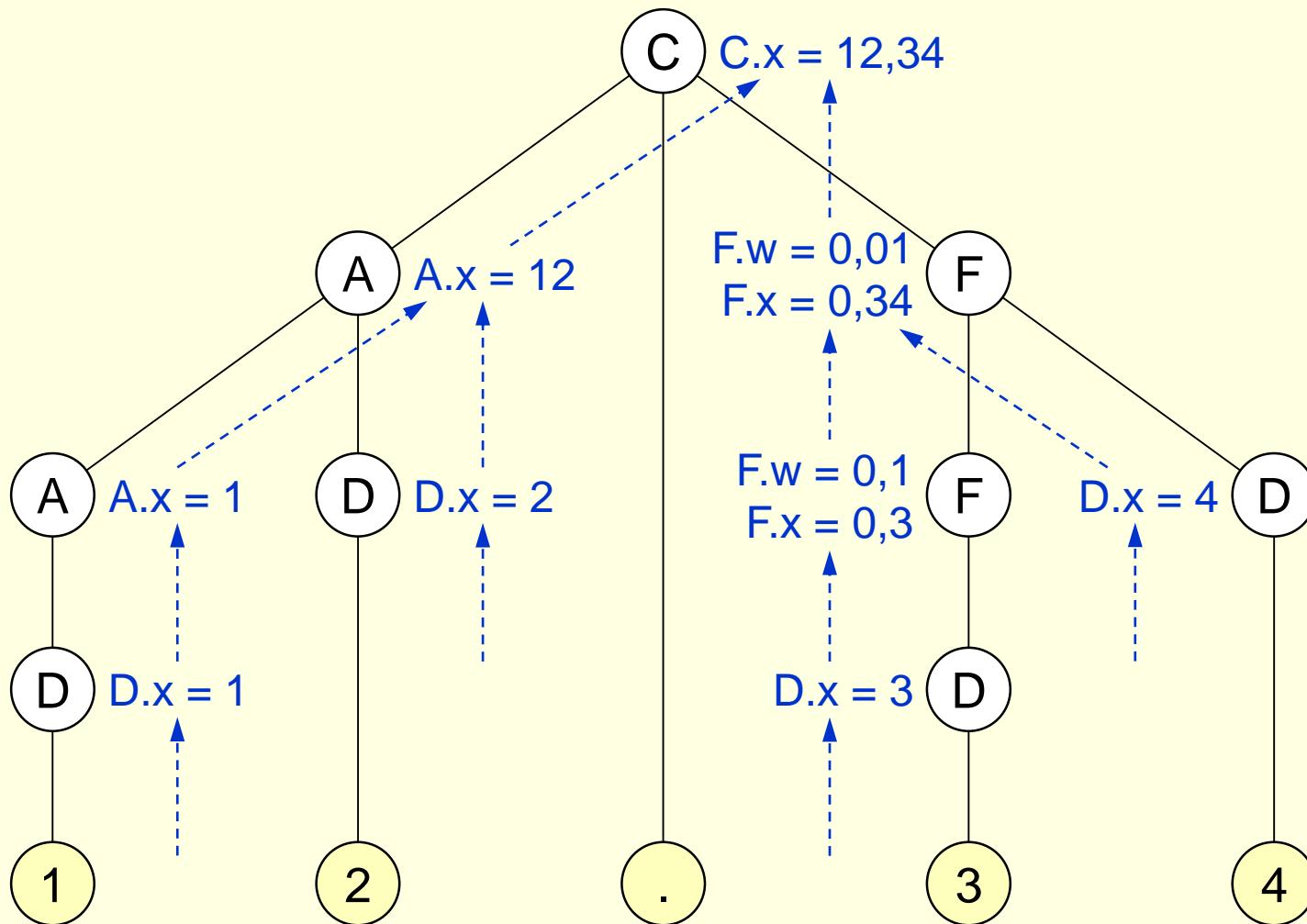
Унаследованные атрибуты затруднительно использовать при восходящем разборе

# Атрибутные трансляции (пример 2)

| Синтаксические правила | Семантические правила<br>(синтезированные атрибуты) |
|------------------------|---|
| $C \rightarrow A$      | $C.x = A.x$   |
| $C \rightarrow A.F$    | $C.x = A.x + F.x$                                   |
| $A \rightarrow D$      | $A.x = D.x$   |
| $A \rightarrow AD$     | $A.x = A_1.x * 10 + D.x$                            |
| $F \rightarrow D$      | $F.w = 0,1 , \quad F.x = D.x * F.w$                 |
| $F \rightarrow FD$     | $F.w = F_1.w * 0,1 , \quad F.x = F_1.x + D.x * F.w$ |
| $D \rightarrow 0$      | $D.x = 0$   |
| $D \rightarrow 1$      | $D.x = 1$   |
| ...                    | ...   |
| $D \rightarrow 9$      | $D.x = 9$   |

Атрибуты:  $x$  – значение константы или ее части  
 $w$  – множитель разряда («вес» цифры)

# Атрибутные трансляции (пример 2)



Унаследованные атрибуты отсутствуют, применим как  
нисходящий, так и восходящий разбор