

Лекция 8

22 октября

Определение атрибутивной транслирующей грамматики

- *Атрибутивная транслирующая грамматика* – это транслирующая грамматика, к которой добавлены следующие определения:
- 1. Каждый входной символ, символ действия или нетерминальный символ имеет конечное множество атрибутов, и каждый атрибут имеет некоторое множество допустимых значений.
- 2. Все атрибуты нетерминальных символов и символов действия делятся на наследуемые и синтезируемые.
- 3. Правила вычисления наследуемых атрибутов определяются следующим образом:
 - а) каждому вхождению наследуемого атрибута в правую часть продукции сопоставляется правило вычисления этого атрибута как функции некоторых других атрибутов символов, входящих в правую или левую часть данной продукции;
 - б) задаётся начальное значение каждого наследуемого атрибута начального символа грамматики.
- 4. Правила вычисления синтезируемых атрибутов определяются следующим образом:
 - а) каждому вхождению синтезируемого атрибута нетерминального символа в левую часть продукции сопоставляется правило вычисления этого атрибута как функции некоторых других атрибутов символов, входящих в правую или левую часть данной продукции;
 - б) каждому синтезируемому атрибуту символа действия сопоставляется правило вычисления этого атрибута как функции некоторых других атрибутов того же символа действия.

6.3. Примеры атрибутивных транслирующих грамматик

-
-
-
- 1. $\langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle$
- 2. $\langle E \rangle \rightarrow \langle T \rangle$
- 3. $\langle T \rangle \rightarrow \langle T \rangle * \langle P \rangle$
- 4. $\langle T \rangle \rightarrow \langle P \rangle$
- 5. $\langle P \rangle \rightarrow (\langle E \rangle)$
- 6. $\langle P \rangle \rightarrow I$
-
-
-

Транслирующая грамматика

-
-
-
-
-
-
- 1. $\langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle$ { Сложить }
- 2. $\langle E \rangle \rightarrow \langle T \rangle$
- 3. $\langle T \rangle \rightarrow \langle T \rangle * \langle P \rangle$ { Умножить }
- 4. $\langle T \rangle \rightarrow \langle P \rangle$
- 5. $\langle P \rangle \rightarrow (\langle E \rangle)$
- 6. $\langle P \rangle \rightarrow I$
-
-
-

Задание атрибутов

-
-
-
-
-
-
-
-
-
-
-
-
-
-

Сложить(p, q, r)

Сложить _{p, q, r}

Атрибутная транслирующая грамматика

-
-
- 1. $\langle E \rangle_{r2} \rightarrow \langle E \rangle_{p1} + \langle T \rangle_{q1} \{ \text{Сложить}_{p2,q2,r1} \}$
- $p2 := p1, q2 := q1, r1 := \text{Нов}, r2 := r1$
- 2. $\langle E \rangle_{p2} \rightarrow \langle T \rangle_{p1}$
- $p2 := p1$
- 3. $\langle T \rangle_{r2} \rightarrow \langle T \rangle_{p1} * \langle P \rangle_{q1} \{ \text{Умножить}_{p2,q2,r1} \}$
- $p2 := p1, q2 := q1, r1 := \text{Нов}, r2 := r1$
- 4. $\langle T \rangle_{p2} \rightarrow \langle P \rangle_{p1}$
- $p2 := p1$
- 5. $\langle P \rangle_{p2} \rightarrow (\langle E \rangle_{p1})$
- $p2 := p1$
- 6. $\langle P \rangle_{p2} \rightarrow I_{p1}$
- $p2 := p1$
-
-
-

Другой пример

-
-
-
-
- $\langle S \rangle \rightarrow I = \langle E \rangle$
- $\langle E \rangle \rightarrow I \langle R \rangle$
- $\langle R \rangle \rightarrow + I \langle R \rangle$
- $\langle R \rangle \rightarrow \varepsilon$
-
-
-

Транслирующая грамматика

-
-
-
-
- 1. $\langle S \rangle \rightarrow I = \langle E \rangle \{\text{Присвоить}\}$
- 2. $\langle E \rangle \rightarrow I \langle R \rangle$
- 3. $\langle R \rangle \rightarrow + I \{\text{Сложить}\} \langle R \rangle$
- 4. $\langle R \rangle \rightarrow \varepsilon$
-
-
-

Атрибутная транслирующая грамматика

- 1. $\langle S \rangle \rightarrow l_{p1} = \langle E \rangle_{q1} \{ \text{Присвоить}_{p2, q2} \}$
- $p2 := p1, q2 := q1$
- 2. $\langle E \rangle_{t2} \rightarrow l_{p1} \langle R \rangle_{p2, t1}$
- $p2 := p1, t2 := t1$
- 3. $\langle R \rangle_{p1, t2} \rightarrow + l_{q1} \{ \text{Сложить}_{p2, q2, r1} \} \langle R \rangle_{r2, t1}$
- $p2 := p1, q2 := q1, r1 := \text{Нов}, r2 := r1, t2 := t1$
- 4. $\langle R \rangle_{p1, p2} \rightarrow \varepsilon$
- $p2 := p1$

Рассмотрим атрибутное дерево вывода для следующего оператора присваивания:

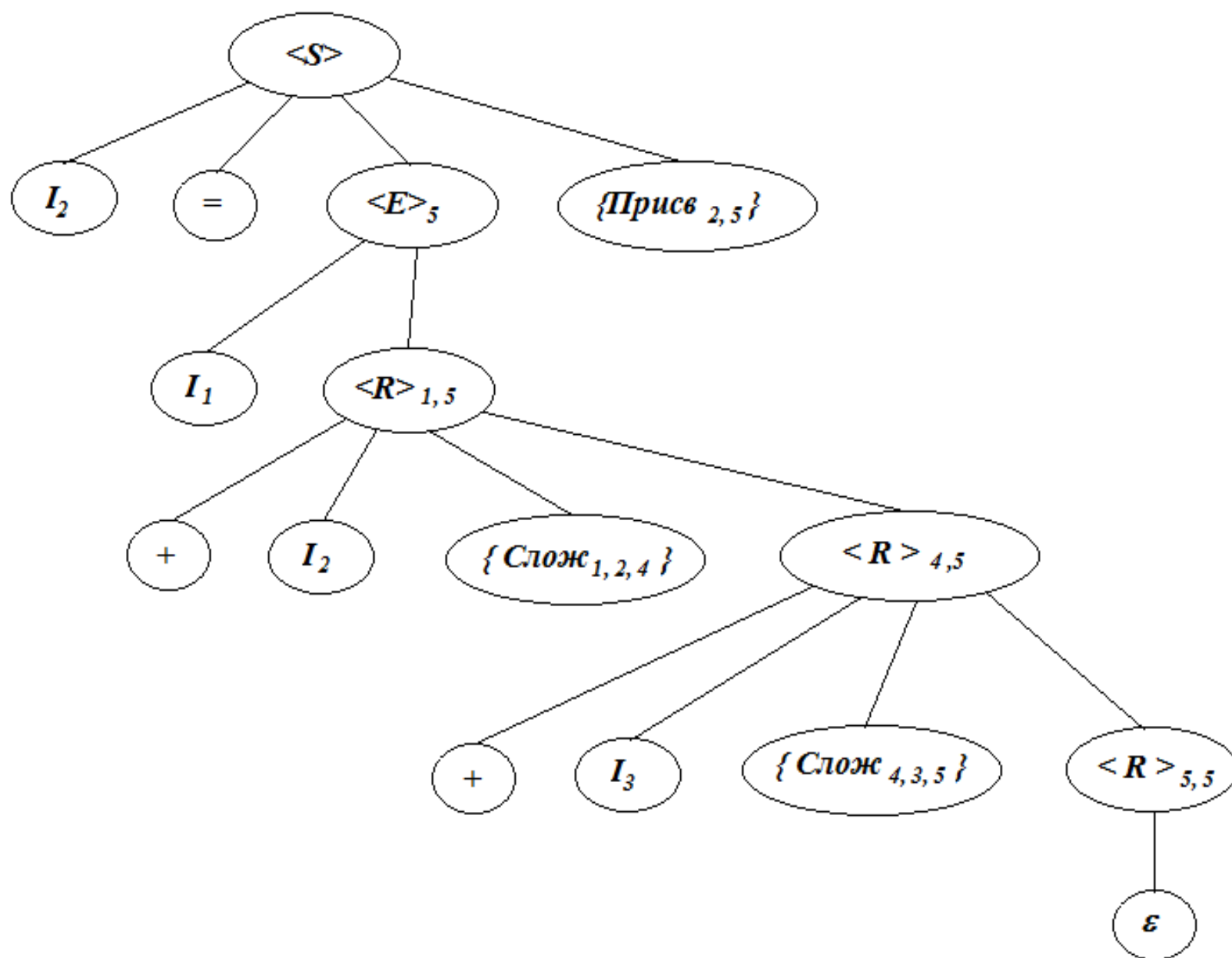
$$b = a + b + c$$

Предположим, что переменные **a**, **b**, **c** размещаются в ячейках с адресами 1, 2, 3 соответственно. Тогда данный оператор присваивания можно записать как

$$l_2 = l_1 + l_2 + l_3$$

Дерево вывода будет выглядеть следующим образом :

Атрибутное дерево вывода



6.4. *L*-атрибутные грамматики

- Определим класс грамматик, называемых *L*-атрибутными грамматиками. Свойство *L*-атрибутности предназначено для того, чтобы можно было вычислять атрибуты в порядке, пригодном для нисходящих процессов обработки.
- Атрибутная транслирующая грамматика называется *L*-атрибутой, если выполняются следующие условия:
- 1. Для каждого правила вычисления наследуемого атрибута символа из правой части продукции каждый аргумент этого правила – это или наследуемый атрибут символа левой части, или произвольный атрибут символа правой части, находящегося левее данного символа.
- 2. Для каждого правила вычисления синтезируемого атрибута символа левой части продукции каждый аргумент этого правила – это или наследуемый атрибут символа левой части, или произвольный атрибут символа правой части данной продукции.
- 3. Для каждого правила вычисления синтезируемого атрибута символа действия любой аргумент этого правила – это наследуемый атрибут данного символа действия.

Копирующие правила

- Введём ещё ограничение на правила вычисления атрибутов.
- В общем случае правило вычисления некоторого атрибута имеет вид:
 -
 - $y := f(x_1, x_2, \dots, x_n)$
 -
- Введём следующее определение.
- Правило вычисления атрибутов называется *копирующим*, если оно имеет одну из следующих форм:
 -
 - $y := x$ или $(y_1, y_2, \dots, y_m) := x$
 -
- где левая часть – это атрибут или список атрибутов, а правая часть – либо атрибут, либо константа.
- Правая часть правила называется *источником* копирующего правила.

Независимое множество копирующих правил

- Иногда можно объединить два или более копирующих правила в одно.
- Например, правила:
 - $(x, y) := z$ и $(a, b) := y$
- можно записать как одно правило:
 - $(a, b, x, y) := z$
- Множество копирующих правил называется *независимым*, если источник каждого правила из этого множества не входит ни в одно из других правил множества.

Грамматика вида *простого присваивания*

- И ещё одно определение.
- Атрибутная транслирующая грамматика имеет *форму простого присваивания*, если:
 - а) не копирующими могут быть только правила вычисления синтезируемых атрибутов символов действия;
 - б) для каждой продукции соответствующее множество копирующих правил является независимым.
- Необходимо сделать замечание относительно копирующих правил.
- Каждый вызов процедуры **Нов** даёт другое значение, поэтому объединять правила, источником которой является **Нов** нельзя.

6.5. Атрибутный автомат с магазинной памятью

- Перед нами стоит следующая задача.
- Для данной L -атрибутной грамматики в форме простого присваивания с входной $LL(1)$ -грамматикой построить МП-автомат, выполняющий описываемый этой атрибутной грамматикой перевод.
- Для решения этой задачи необходимо построить по исходной $LL(1)$ -грамматике транслирующую грамматику цепочечного перевода путём включения в неё необходимых символов действия.
- Затем строится соответствующая L -атрибутная грамматика в форме простого присваивания.
- Атрибутный МП-автомат строится как обычный магазинный автомат, но к каждому магазинному символу добавляются поля для атрибутов.
- Правила обработки переходов дополняются правилами обработки атрибутов (атрибутных полей).
- Обсудим теперь схему использования атрибутивных магазинных символов.

Схема использования атрибутивных магазинных символов

- К тому моменту, когда магазинный символ оказывается на верху магазина, его поля будут заполнены следующим образом:
- 1. Если соответствующий атрибут наследуемый, то поле будет содержать значение этого атрибута.
- 2. Если соответствующий атрибут синтезируемый, то поле будет содержать указатель на связанный список наследуемых полей, где этот атрибут должен быть запомнен.
- 3. Атрибуты входного символа трактуются как синтезируемые.
- В поле синтезируемого атрибута и атрибута входного символа заносится указатель в тот момент, когда символ вталкивается в магазин; содержимое поля не меняется в течение времени существования соответствующего магазинного символа.
- Поле для наследуемого атрибута заполняется либо значением этого атрибута, либо указателем на список (или маркером конца списка).
- В последнем случае поле является элементом списка, на который указывает поле какого-нибудь синтезируемого атрибута или атрибута входного символа, и будет заполнено некоторым значением перед тем, как соответствующий символ окажется на верху магазина.

Основные конфигурации

- Рассмотрим теперь основные ситуации, которые возникают при работе атрибутного МП-автомата.
- ***1. Начальная конфигурация.***
- Процесс обработки начинается, когда в магазине находится начальный символ грамматики и маркер дна.
- Инициализация полей начального символа является частью общей инициализации магазина.
- Поля, соответствующие наследуемым атрибутам, заполняются начальными значениями, которые задаются атрибутной грамматикой.
- Поля, соответствующие синтезируемым атрибутам, заполняются указателями на списки наследуемых полей, в которые впоследствии должны быть занесены значения этих синтезируемых атрибутов.

Основные конфигурации (продолжение)

- **2. На верху магазина входной символ.**
- Когда верхним магазинным символом является входной символ, и магазинный символ совпадает с текущим входным символом (случай, когда нет ошибки), атрибуты текущего входного символа отображаются на соответствующие поля верхнего магазинного символа (напомним, что эти поля синтезируемые).
- Каждое из этих полей содержит указатель на список наследуемых полей магазина, в которые требуется занести значение соответствующего атрибута входного символа.
- Обычное в таких случаях действие автомата (*Вытолкнуть, Сдвиг*) расширяется так, что каждый атрибут текущего входного символа копируется во все поля списка полей, на который указывает соответствующее поле верхнего магазинного символа.

Основные конфигурации (продолжение)

- **3. На верху магазина символ действия.** Когда на верху магазина оказывается символ действия, обычный переход МП-автомата дополняется процессами обработки атрибутов следующим образом:
 - а) значения наследуемых атрибутов извлекаются из соответствующих полей верхнего магазинного символа;
 - б) значения синтезируемых атрибутов вычисляются по наследуемым атрибутам с помощью правил вычисления атрибутов, связанных с символом действия;
 - в) значение каждого синтезируемого атрибута помещается во все поля списка полей, на которые указывает соответствующее поле верхнего магазинного символа.
- Кроме того, атрибуты также используются для того, чтобы выдать соответствующую выходную команду.
- Естественно, что переходная процедура для символа действия выполняет также обычные операции автомата без атрибутов – *Вытолкнуть, Держать*.

Основные конфигурации (продолжение)

- ***4. На верху магазина нетерминал.***
- Когда верхний магазинный символ является нетерминалом, автомат без атрибутов выполняет переход, основанный на продукционном правиле.
- Этот переход заменяет верхний магазинный символ цепочкой символов из правой части продукции и, возможно, производит выдачу, связанную с символами действия, которые входят в правую часть продукции, но не вталкиваются в магазин.
- Атрибутный автомат вталкивает в магазин ту же цепочку символов и выдаёт те же выходные символы.
- Кроме того, атрибутный переход может потребовать некоторых вычислений атрибутов и некоторых действий по заполнению полей помещаемых в магазин символов и действий по занесению значений атрибутов в некоторые поля символов, хранящихся в магазине.

6.6. Трансляция оператора присваивания

- Исходная грамматика:
-
- 1. $\langle S \rangle \rightarrow I = \langle E \rangle \{ \text{Присвоить} \}$
- 2. $\langle E \rangle \rightarrow I \langle R \rangle$
- 3. $\langle R \rangle \rightarrow + I \{ \text{Сложить} \} \langle R \rangle$
- 4. $\langle R \rangle \rightarrow \varepsilon$
-

- Множества выбора:
-
- $\text{Выбор}(1) = \{ I \}$
- $\text{Выбор}(2) = \{ I \}$
- $\text{Выбор}(3) = \{ + \}$
- $\text{Выбор}(4) = \text{След}(\langle R \rangle) = \{ \# \}$

Управляющая таблица МП-автомата

Магазинные символы	I	=	+	⊥
<S>	#1	Отвергнуть	Отвергнуть	Отвергнуть
<E>	#2	Отвергнуть	Отвергнуть	Отвергнуть
<R>	Отвергнуть	Отвергнуть	#3	#4
I	Вытолкнуть Сдвиг	Отвергнуть	Отвергнуть	Отвергнуть
=	Отвергнуть	Вытолкнуть Сдвиг	Отвергнуть	Отвергнуть
▽	Отвергнуть	Отвергнуть	Отвергнуть	Допустить
Начальное содержимое магазина <E>▽				